

Version 10.1.0



User Guide

Product Information

This document applies to IBM Cognos BI Version 10.1.0 and may also apply to subsequent releases. To check for newer versions of this document, visit the IBM Cognos Information Centers (<http://publib.boulder.ibm.com/infocenter/cogic/v1r0m0/index.jsp>).

Copyright

Licensed Materials - Property of IBM

© Copyright IBM Corp. 2005, 2010.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

IBM, the IBM logo, ibm.com, PowerPlay, ReportNet, TM1, and Cognos are trademarks or registered trademarks of International Business Machines Corp., in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Table of Contents

Introduction	51
Chapter 1: What's New?	53
New Features in Version 10.1.0	53
Use Your Own External Data	53
Active Reports	53
Statistical Analysis	54
Mozilla Firefox Support	54
Chart Enhancements	54
Enhancements to the User Interface	56
Improved Set Operations	57
Preview Reports when Opening and Saving	58
Report Name Used for the Exported Output File Name	58
Enhancements for SAP BW Data Sources	58
Column Titles When Drilling up or Down	58
Custom Properties for Prompts	58
Simplified Calculations	58
New Report Style	59
List Enhancements	59
Accessible Report Output	59
Curly Brackets and Parentheses are Breakable	60
Changed Features in Version 10.1.0	60
Report Studio Express Authoring Mode Replaced by Business Insight Advanced	60
Order of Drawing Bars in Bar Charts	60
Removed Features in Version 10.1.0	60
Analyze with Analysis Studio	60
Conformed Dimensions on SAP BW Data Sources	60
Excel 2000 and Excel 2000 Single Sheet Report Outputs	60
What's New in Version 8.4	61
New Features in Version 8.4	61
Changed Features in Version 8.4	64
What's New in Version 8.3	64
New Features in Version 8.3	65
Changed Features in Version 8.3	68
Chapter 2: Understanding Report Studio	73
Building IBM Cognos Business Intelligence Applications	73
Relational vs. Dimensional Reporting Styles	74
The User Interface	76
Report Authoring Views	79
Page Design View	80
Page Structure View	80
Report Layout and Queries	80
Layout	80

Table of Contents

Queries	81
Report Objects	82
Find Objects in a Report	83
Options	83
Web Browser Settings	87
The IBM Cognos Software Development Kit	88
Chapter 3: Creating a Report	89
Specify the Data Package	89
Refresh the Package	90
Choose a Basic Report Layout	90
Add Data	90
Data Source Icons	91
Insert a Single Data Item	92
Validate a Report	93
Save a Report	94
Run a Report	94
Running a Report Against a Dimensional Data Source	97
Specify Not to Render a Page If It Does Not Contain Data	98
Controlling the Rows Per Page for Multiple Containers in HTML and PDF	98
Producing a Report in CSV Format	98
Producing a Report in Excel Format	99
Producing a Report in XML Format	100
Set PDF Page Options	100
View Lineage Information for a Data Item	101
The IBM Cognos BI Lineage Tool	102
Access the IBM WebSphere Business Glossary	103
Creating Reports for Business Insight Dashboards	104
Prompts in Business Insight Dashboards	105
Enable Filters in Business Insight Dashboards	105
Creating Reports for IBM Cognos for Microsoft Office	106
Creating Reports for Mobile Devices	107
Creating location-aware reports	107
Unsupported features	108
Chapter 4: Lists	111
Set List Properties	111
Hide Columns in List Reports	112
Limitations When Using Set Expressions in List Reports	112
Using Repeaters	113
Convert a List into a Repeater	113
Example - Create Mailing Labels	114
Chapter 5: Crosstabs	117
Crosstab Nodes and Crosstab Node Members	117
Moving Crosstab Nodes	118
Crosstab Node Creation Option	118
Set Crosstab Properties	118
Create a Single-Edge Crosstab	120
Create a Nested Crosstab	121

Specify the Default Measure	122
Swap Columns and Rows	122
Indent Data	123
Change a List into a Crosstab	123
Example - Add Aggregate Data to a Crosstab	123
Example - Create a Discontinuous Crosstab	125
Chapter 6: Charts	127
Current Default Charts and Legacy Charts	127
Creating Charts	127
Chart Objects	127
Chart Types	130
Choosing a Chart Type and Configuration	130
Column Charts	131
Line Charts	131
Pie Charts	132
Bar Charts	133
Area Charts	133
Point Charts	134
Combination Charts	134
Scatter Charts	135
Bubble Charts	136
Quadrant Charts	136
Bullet Charts	137
Gauge Charts	138
Pareto Charts	138
Progressive Column Charts	139
Microcharts	139
Marimekko Charts	140
Radar Charts	141
Polar Charts	141
Range Indicator Charts	142
Chart Configurations	142
Convert Charts From One Type to Another	145
Convert Legacy Charts to Current Default Charts	145
Limitations When Converting Legacy Charts	145
Customizing Chart Properties	147
Resize a Current Default Chart	150
Resize a Legacy Chart	150
Customizing the Color Palette of a Chart	151
Creating a Conditional Color Palette in a Chart	153
Change a Chart Background in a Current Default Chart	157
Change a Chart Background in a Legacy Chart	157
Add Background Effects to a Chart Object in a Legacy Chart	158
Add Colored Regions in a Current Default Chart	159
Specify Custom Label Text	161
Customizing the Axes of a Chart	161
Showing Gridlines in a Chart	165
Customize the Legend in a Current Default Chart	166

Table of Contents

Customize the Items in the Legend of a Current Default Chart	167
Customize the Items in the Legend of a Legacy Chart	168
Adding a Note to a Chart	169
Adding a Baseline to a Chart	170
Adding Markers to Charts	173
Display Trendlines in Current Default Charts	176
Display Regression Lines in Legacy Scatter or Bubble Charts	177
Show Data Values or Labels in a Current Default Pie or Donut Chart	178
Show Data Values or Labels in Other Current Default Charts	179
Define Query Context When Customizing Legend Entries, Legend Titles, or Axis Labels	180
Summarize Small Slices, Bars, or Columns in Current Default Charts	181
Customizing Lines and Data Points in a Line Chart	181
Customize a Current Default Combination Chart	183
Create a Matrix of Current Default Charts	184
Customize a Current Default Gauge Chart	186
Customize a Legacy Gauge Chart	187
Define Color by Value in Current Default Scatter and Bubble Charts	188
Specify the Bubble Size in a Current Default Bubble Chart	190
Set the Position of the First Slice in a Current Default Pie Chart	190
Create a Donut Chart from a Current Default Pie Chart	191
Pull Out Pie Slices in a Current Default Pie Chart	191
Define the Cumulation Line in a Current Default Pareto Chart	192
Define the Cumulation Line in a Legacy Pareto Chart	193
Insert a Microchart into a Crosstab	193
Customize a Current Default Bullet Chart	194
Create a Drill-up and Drill-down Chart	194
Example - Creating Drill-through Access in a Legacy Chart	195
Create the target report	195
Create the Source Report	196
Define the Drill Behavior	197
Example - Creating and Customizing a Metrics Range Legacy Chart	197
Create a Metrics Range Chart	198
Customize the Chart	198
Customize the Chart Legend Labels	199
Adding a Metric Studio Diagram to a Report	200
Add a Static Metric Studio Diagram to a Report	200
Add a Data-driven Metric Studio Diagram to a Report	201
Chapter 7: Maps	203
The Report Studio Map Object	203
Example - Create a Map Report	204
Set Map Properties	205
Example - Define Data Values for the Region Layer	207
Match Data Values to Names in the Map File	208
Define Data Values for the Point Layer	209
Add Colors to the Region or Point Layer	210
Add a Legend Title	210
Add a Note to a Map	211
Drill Through to Another Report From a Map	211

Edit a Map	212
Additional Maps	213
Location Intelligence	213
Chapter 8: Active Reports	215
Active Report Controls	215
Active Report Variables	216
Convert an Existing Report to an Active Report	216
Specify Active Report Properties	216
Adding Interactivity to a Report	217
Create and Manage Active Report Variables	217
Adding Controls to a Report	218
Add Data to a Control	220
Define a Connection Between Controls	221
Specify the Selection Behavior of a Control	222
Specify the Reaction Behavior of a Control	223
Set the No Value List Item	225
Show or Hide a Column or Row	225
Sort Data in a List	226
Summarize Data in the Active Report Output	226
Run or Save an Active Report	226
Example - Create an Active Report Showing Revenue Data for Core Products	227
Example - Create an Active Report Showing Revenue Data for New Order Methods	231
Chapter 9: Statistical Analysis	235
IBM Cognos Statistics	235
Common Concepts in Statistical Analysis	236
Building Cases in IBM Cognos Statistics	240
Descriptive Statistics	241
Basic Descriptive Statistics	242
Create a Descriptives Table	244
Histograms	246
Create a Histogram	247
Boxplots	248
Create a Boxplot	250
Create a Q-Q Plot	252
Q-Q Plot Options	254
Means Comparison	255
t-Tests	255
Create a One-Sample t-Test	256
Using One-Way ANOVA to Compare Means	258
ANOVA Options	262
Nonparametric Tests	265
Create A One-Way Chi-Square Test	266
Example: Specify the Proportion for Expected Values in a Chi-Square Test	268
Create a Two-Way Chi-Square Test	270
Basic Correlation and Linear Regression	273
Showing Correlation Between Variables	274
Correlation Options	277
Use Linear Regression	278

Table of Contents

Linear Regression Options	282
Create a Curve Estimate	285
Curve Estimation Options	287
Statistical Process Control	291
Control Charts	292
Create X-Bar, R, and S Charts	294
Create Individuals and Moving Range Charts	299
Create p,np Charts	300
Create c,u Charts	301
Cases are Units or Cases are Subgroups	302
Specification Limits	303
Control Rules	303
Process Capability Indices	304
Process Performance Indices	305
Chapter 10: Relational Reporting Style	307
Add Relational Data to a Report	307
Group Relational Data	308
Perform Advanced Grouping	309
Set the Group Span for a Column	310
Divide Data into Sections	313
Summarizing Data Relationally	314
Add a Simple Summary	315
Set the Auto Group & Summarize Property	316
Specify Detail or Summary Aggregation in the Model	317
Specify Aggregation Properties for a Data Item	317
Use Summary Functions in Data Item Expressions	317
Summary Functions	320
Focusing Relational Data	324
Create a Detail or Summary Filter	324
Sorting Relational Data	328
Perform Advanced Layout Sorting	329
Working with Relational Queries	329
Specify a List of Data Items for an Object	330
Relate a Query to a Layout	330
Connecting Queries Between Different Data Sources	330
Add a Query to a Relational Report	331
Create a Union Query	332
Create a Join Relationship	336
Reference a Package Item in a Child Query	337
Create a Master Detail Relationship	337
Working with Queries in SQL	340
Using Relational Calculations	342
Create a Simple Calculation	343
Insert a Query Calculation	344
Create a Layout Calculation	346
Chapter 11: Dimensional Reporting Style	347
Add Dimensional Data to a Report	347
Customize the Source Tree	349

Insert a Member	349
Search for a Member	350
Nest Members	351
Insert a Hierarchy	351
Insert a Member Property	352
Create a Set of Members	352
Edit a Set	353
Extended Data Items	356
Tips for Working with Ragged or Unbalanced Hierarchies	357
Limited Support for Relational Functions When Used with OLAP Data Sources	357
Summarizing Data Dimensionally	358
Add a Simple Summary	360
Summarizing Values in Crosstabs	361
Summarizing Member Sets	363
Limitations When Summarizing Measures in DMR Data Sources	363
Rolling and Moving Averages	364
Show Data for Specific Time Periods	365
Summary Functions	365
Focusing Dimensional Data	371
Create a Context Filter	371
Limitations When Filtering Dimensional Data Sources	373
Sorting Dimensional Data	375
Perform Advanced Set Sorting	377
Working with Dimensional Queries	377
Specify a List of Data Items for an Object	378
Relate a Query to a Layout	378
Using Dimensional Data Sources with Queries	379
Connecting Queries Between Different Data Sources	379
Multiple-fact Queries	380
Add a Query to a Dimensional Report	383
Add Dimension Information to a Query	384
Create a Master Detail Relationship	386
Working with Queries in SQL or MDX	389
Example - Create a Dynamic Report That Shows Period-to-date Data	395
Using Dimensional Calculations	396
Create a Simple Member Calculation	397
Insert a Query Calculation	398
Create a Layout Calculation	399
Create an Intersection (Tuple)	400
Assigning a Hierarchy or Dimension	400
Browse or Search the Values of a Data Item	401
Considerations when Creating Calculations	402
Drilling Up and Drilling Down in Dimensional Reports	413
Member Sets	414
Create a Drill-up and Drill-down Report	414
Example - Create a Dashboard Report	417
Chapter 12: Adding Prompts to Filter Data	421
Use the Build Prompt Page Tool	422

Table of Contents

Build Your Own Prompt and Prompt Page	423
Example - Create a Report Showing Products Shipped for a Specific Time Interval	426
Create a Parameter to Produce a Prompt	428
Create a Prompt Directly in a Report Page	429
Modifying Prompts	430
Change the Prompt Control Interface	430
Specify That a Prompt Requires User Input	431
Allow Users to Select Multiple Values in a Prompt	431
Show or Hide Prompt Status	432
Specify a Default Selection for a Prompt	432
Customize Prompt Text	433
Specify Prompt Values	433
Add a Prompt Button	434
Create a Cascading Prompt	435
Control the Data That Appears in a Tree Prompt	435
Using in_range Filters with Character Data	436
Chapter 13: Laying Out a Report	437
Report Layout Recommendations	437
The Page Structure View	437
Copy Object Formatting	438
Add a Header or Footer to a Report or List	439
Add Borders to an Object	440
Add a Text Item to a Report	441
Specify the Font for a Report	441
Insert an Image in a Report	441
Insert a Background Image in an Object	442
Add Background Effects	443
Insert a Formatting Object in a Report	444
Elements Supported in Rich Text Items	447
Example - Add a Multimedia File to a Report	448
Align an Object in a Report	448
Use a Table to Control Where Objects Appear	448
Apply a Table Style	449
Apply Padding to an Object	450
Set Margins for an Object	450
Add Multiple Items to a Single Column	451
Example - Create a Report with Multiple Items in One Column	451
Reuse a Layout Object	453
Change a Reused Object	454
Update Reused Objects	455
Specify Text Flow in an Object	455
Specify Line Spacing and Breaking	456
Specify the Height and Width of an Object	456
Control How Other Objects Flow Around an Object	457
Specify Report Properties	458
Create and Modify Report and Object Styles	459
(Don't Print) Class	460
Modify Classes to Format Query Studio Reports	461

Modifying the Default Layout Style Sheet	461
Add Color to an Object	461
Chapter 14: Working With Your External Data	463
Preparing to Work with your External Data	465
Import Your External Data	466
Edit Your External Data	469
Delete Your External Data	470
Running a Report That Contains External Data	470
Making your Reports Public	470
Chapter 15: Using Conditions	473
Highlight Data Using a Conditional Style	473
Manage Conditional Styles	476
Example - Add a Conditional Style to an Existing Report	476
Highlight Data Using a Style Variable	477
Adding Conditional Rendering to a Report	478
Add a Variable	478
Hide or Show an Object	480
Add Conditional Rendering	481
Example - Create a Conditional Report	481
Add Multiple Layouts	484
Setting up a Multilingual Reporting Environment	484
Create a Multilingual Report in Report Studio	485
Chapter 16: Bursting Reports	487
Defining Burst Recipients	487
Create a Calculated Field	487
Creating a Burst Table	488
Specify a Burst Group	490
Set Burst Options	490
Enable Bursting	492
Creating Burst Reports Using Dimensional Data Sources	492
Example - Burst a Report	493
Chapter 17: Format Data	495
Set the Default Data Formats	495
Specify the Data Format for an Object	496
Specifying the Number of Decimals in Numbers	496
Locale-sensitive Properties	497
Suppress Empty Cells	498
Suppress Null Cells Using Filters	498
Limitations When Formatting Empty Cells in SAP BW Data Sources	499
Example - Suppress Zeros in Rows and Columns in an Existing Report	499
Specify What Appears for Data Containers that Contain No Data	500
Using Patterns to Format Data	500
Pattern Guidelines	500
Date and Time Symbols	501
Decimal Format Symbols	509

Table of Contents

Chapter 18: Using Query Macros	511
Add a Query Macro	511
Creating Prompts Using Query Macros	511
Example - Select a Country Prompt	514
Chapter 19: Using Drill-through Access	515
Understanding Drill-through Concepts	516
Drill-through Paths	516
Selection Contexts	517
Drilling Through to Different Report Formats	517
Drilling Through Between Packages	518
Bookmark References	518
Members and Values	519
Member Unique Names	519
Conformed Dimensions	520
Business Keys	521
Scope	521
Mapped Parameters	522
Drilling Through on Dates Between PowerCubes and Relational Packages	523
Set Up Drill-through Access in a Report	523
Specify the Drill-through Text	526
Example - Drill Through to a Hidden Report from a Report Studio Report	527
Check the Target Report	527
Create and Test the Drill-Through Definition	528
Try It Yourself - Create a Report with Drill-through Access to Itself	529
Drilling Through from IBM Cognos Series 7 to IBM Cognos BI	533
Setting Up Drill-through Access from IBM Cognos Visualizer	533
Setting Up Drill-through Access from PowerPlay Web	533
Create and Test the Target for a Series 7 Report	534
Chapter 20: Working with Multiple Pages	535
Add a Page to a Report	535
Create a Page Break or Page Set	536
Join Nested Page Sets	537
Example - Preparing a Product List Report	538
Add a Bookmark	540
Create a Table of Contents	542
Example - Add a Table of Contents to a Report	543
Insert Page Numbers in a Report	545
Control Page Breaks and Page Numbering	545
Create Page Layers	547
Chapter 21: Creating Report Templates	549
Convert a Report to a Template	549
Create a New Template	549
Chapter 22: Managing Existing Reports	551
Open and Save a Report on Your Computer	551
Copy a Report to the Clipboard	552
Open a Report from the Clipboard	552

Open a File from Another Studio	552
Analysis Studio Query Specification	553
Managing Changes in the Package	557
Update a Name Reference	558
Change the Package	558
Chapter 23: Upgrading Reports	561
Upgrading Reports from IBM Cognos BI Version 8.4	561
Upgrading Report Styles	561
Upgrading Legacy Charts	562
New Default for Hidden or Deleted Axis Titles	562
Upgrading Reports with SAP BW Prompt Variables	562
Upgrading Reports from IBM Cognos BI Version 8.1 or 8.2	563
Members Containing No Data Are Not Suppressed for SAP BW	563
Thousands Separators Missing	563
RQP-DEF-0177 Error When Upgrading Reports	563
Upgrading Reports from IBM Cognos ReportNet	564
Undocumented and Unsupported Features	564
No Sort Order	564
Layout Errors	564
Report Format	564
Customized Style Sheets	564
PDF Reports	565
IF-THEN-ELSE Statements	565
Solve Order	565
Chart Behavior	565
Database Only Processing of Queries	565
Changes in the Behavior of Functions Between ReportNet 1.1 and IBM Cognos BI	565
Chapter 24: Sample Reports and Packages	567
The Great Outdoors Company Samples	567
The Great Outdoors Group of Companies	568
Employees	569
Sales and Marketing	569
Great Outdoors Database, Models, and Packages	570
Samples in the Sales and Marketing (Cube) Package	573
Actual vs. Planned Revenue	573
Historical Revenue	573
Revenue by Date Range	573
Revenue by Product Brand (2005)	574
Running Total For Promotion	574
Sales Revenue Expectation	574
Same Month Prior Year	575
Selected Retailer Country	575
Top Retailers by Country	575
Tree Prompt Retailers Set	576
Samples in the GO Data Warehouse (analysis) Package	576
Budget vs. Actual	576
Customer Returns and Satisfaction	576
Employee Satisfaction 2006	577

Table of Contents

Employee Training by Year	577
Eyewear Revenue by Brand and Size	577
Global Bonus Report	578
GO Balance Sheet as at Dec 31 2006	578
Great Outdoors Company Balance Sheet as at Dec 31 2006	579
Manager Profile	579
Planned Headcount	579
Positions to Fill	579
Promotion Success	580
Quantity Sold vs. Shipped and Inventory	580
Recruitment Report	580
Return Quantity by Order Method	580
Returned Items	581
Returns by Damage, Failed Orders and Complaints in 2006	581
Returns by Failed Orders in 2006	581
Returns by Order Method	582
Revenue by GO Subsidiary 2005	582
Sales Commissions for Central Europe	583
Sales Growth Year Over Year	583
Succession Report	583
Top 10 Retailers for 2005	584
Samples in the GO Data Warehouse (query) Package	584
Bursted Sales Performance Report	584
Employee Expenses	585
Health Insurance	585
Pension Plan	585
Regular Salary	585
TOC Report	585
Total Revenue by Country	586
Samples in the GO Sales (analysis) Package	586
2005 Quarterly Sales Forecast	586
2005 Sales Summary	586
Briefing Book	586
Horizontal Pagination	587
No Data	587
PDF Page Properties	587
Singletons on Page Body	588
Table of Contents	588
Samples in the GO Sales (query) Package	588
Order Invoices - Donald Chow, Sales Person	588
Samples in the IBM Cognos Statistics Package	589
Advertising Costs on Sales Revenue	589
Catalog Sales	589
Clothing Manufacturer Quality	589
Discharged Patients	590
DVD Score	590
Income Growth	590
Shampoo pH Level	590
Interactive Samples	590

Bursted Sales Performance Report	591
Percentage Calculation (by year)	591
Recruitment Report	591
Revenue by GO Subsidiary 2005	591
Rolling and Moving Averages	592
Chapter 25: Limitations When Producing Reports in Microsoft Excel Format	593
Unable to Load Images from the IBM Cognos BI Content Store in a Report	593
Blank Worksheet Appears	593
Warning Message Appears When Excel Opens an IBM Cognos Business Intelligence Report	593
Spreadsheet Content Not Saved for Reports Saved in XLS Format	593
Unable to Load Excel Report in Netscape 7.01	594
Unable to Nest Labels in Charts	594
Data Series Are Truncated	594
Colors Different From Those in HTML or PDF	594
Repeating Pie Charts	594
Unable to Skip Discrete Axis Labels	594
Unsupported IBM Cognos Business Intelligence Formatting	595
Cells Contain Series of #	596
Excel Cannot Render Reports with More Than 256 Columns	596
Table and Column Widths	596
Secure Socket Layer (SSL) Is Not Supported in Some Excel Formats and Versions	596
Number Formats Become Currency Formats in Japanese Excel	596
Reports Show Data in Wrong Columns	597
Unable to Access Reports on Remote Servers	597
Drill-through Reports Are Not Supported in Excel	597
Map Reports Are Not Supported in Excel	597
Unsupported Excel Formatting	597
Hyperlink Buttons Are Not Supported in Excel	597
Unable to View Reports in Excel Format Sent as Email Attachments	598
Unsupported Chart Properties in Excel	598
Unsupported Chart Types in Excel	599
Cell Height and Width Are Incorrect	600
Chapter 26: Report Studio Object and Property Reference	601
Report Studio Objects	601
3-D Area	601
3-D Bar	601
3-D Combination Chart	602
3-D Line	602
3-D Scatter Chart	602
Angular Axis	602
Angular Measure	603
Area	603
Area	603
As of Time Expression	603
Axis Labels	603
Axis Title	603
Axis Title	604
Bar	604

Table of Contents

Bar 604
Baseline 604
Baseline 604
Baseline 605
Baseline 605
Block 605
Bookmark 605
Bubble Chart 606
Bubble Chart 606
Bubble Measure 606
Bubble Measure 606
Bullet Chart 607
Bullet Measure 607
Button 607
Button 607
Button Bar 607
Button Icon 608
Button Label 608
Button Value 608
Calculated Measure 608
Calculated Member 608
Calculated Member 608
Caption 609
Card 609
Card Label 609
Card Value 609
Category axis 609
Category Axis 609
Category Baseline 610
Chart Body 610
Chart Body 610
Chart Footer 610
Chart Footer 610
Chart Measure 611
Chart Node Member 611
Chart Subtitle 611
Chart Subtitle 611
Chart Text Item 611
Chart Text Item 612
Chart Title 612
Chart Title 612
Check Box 612
Check Box Group 612
Check Box Icon 612
Check Box Label 613
Check Box Value 613
Class 613
Color by Value Measure 613
Colored Region 613

Colored Region 614
Combination Chart 614
Combination Chart 614
Component Override 615
Conditional Block 615
Conditional Block 615
Context Item Text 615
Crosstab 615
Crosstab Columns 616
Crosstab Columns 616
Crosstab Corner 616
Crosstab Fact Cells 616
Crosstab Intersection 617
Crosstab Member Fact Cells 617
Crosstab Node Member 618
Crosstab Rows 618
Crosstab Rows 618
Crosstab Space 618
Crosstab Summary 619
Cumulation Label 619
Cumulation Line 619
Cumulation Line 619
Cumulation Line 619
Cumulation Line Axis 619
Cumulation Line Label 620
Data Button Bar 620
Data Check Box Group 620
Data Deck 620
Data Drop-Down List 620
Data Item 621
Data List Box 621
Data Radio Button Group 621
Data Tab Control 621
Data Toggle Button Bar 621
Date 622
Date & Time Prompt 622
Date Prompt 622
Deck 622
Default Measure 622
Default Measure 623
Detail Filter 623
Dimension 623
Dimensional Edge Summary 623
Display Layer 623
Drop-Down List 623
Edge Summary 624
Equation Label 624
Explicit Member Set 624
Fact 624

Table of Contents

Field Set 624
Gauge Axis 624
Gauge Chart 625
Gauge Chart 625
Gauge Labels 625
Gauge Numerical Axis 625
Generated Prompt 626
Hierarchy Set 626
HTML Item 626
Hyperlink 626
Hyperlink Button 627
Image 627
Intersection (Tuple) 627
Interval Prompt 627
Join 628
Key 628
Label 628
Layout Component Reference 628
Legend 628
Legend 628
Legend Title 629
Legend Title 629
Level 629
Level Hierarchy 629
Level Set 629
Line 630
Line 630
List 630
List Box 630
List Cell 630
List Column 631
List Column Body 631
List Columns 631
List Columns 632
List Columns Body Style 632
List Columns Title Style 632
List Column Title 632
List Footer 632
List Header 633
List Item 633
List Item Icon 633
List Item Label 633
List Item Value 633
List Page Footer 634
List Page Header 634
List Row 634
List Row Cells Style 634
List Summary 634
List Summary 634

Map 635
Map Location 635
Map Location 635
Map Refinement Location 635
Map Refinement Location 636
Marimekko Chart 636
Marker 636
Marker 636
Marker 637
MDX 637
Measure 637
Member 637
Member Children Set 637
Member Hierarchy 637
Member Property 638
Member Property 638
Member Set 638
Metrics Range Chart 638
Metric Studio Diagram 638
Named Set 639
Note 639
Note 639
Note Content 639
No Value List Item 639
Numerical Axis 639
Numerical Axis 640
Numeric Axis 640
Numeric Baseline 640
Page 640
Page Body 640
Page Break Text 641
Page Footer 641
Page Header 641
Page Number 641
Page Set 641
Pareto Bars 642
Pareto Chart 642
Pareto Chart 642
Pie Chart 643
Pie Chart 643
Pie Labels 643
Point Color Legend Title 643
Point Layer 644
Point Measure 644
Point Size Legend Title 644
Point Size Measure 644
Polar Chart 644
Primary Axis 645
Primary Bottom Axis 645

Table of Contents

Progressive Chart 645
Progressive Chart 645
Prompt Button 646
Query 646
Query Operation 646
Query Reference 646
Radar Chart 647
Radial Axis 647
Radial Measure 647
Radio Button 647
Radio Button Group 647
Radio Button Icon 648
Radio Button Label 648
Radio Button Value 648
Region Color Legend Title 648
Region Layer 648
Region Measure 648
Regression Line 649
Repeater 649
Repeater Table 649
Repeater Table Cell 649
Rich Text Item 649
Row Number 650
Scatter Chart 650
Scatter Chart 650
Scatter Marker 650
Secondary Axis 651
Secondary Bottom Axis 651
Select & Search Prompt 651
Set Expression 651
Singleton 652
Slicer Member Set 652
SQL 652
Summary Filter 652
Tab 652
Tab Control 652
Tab Icon 653
Tab Label 653
Table 653
Table Cell 653
Table of Contents 653
Table of Contents Entry 654
Table Row 654
Tab Value 654
Target Measure 654
Target Measure 654
Text Box Prompt 654
Text Item 655
Time 655

Time Prompt 655
Toggle Button Bar 655
Tolerance Measure 656
Total Column 656
Total Column Label 656
Tree Prompt 656
Trendline 656
Trendline Label 657
Value Prompt 657
Variable 657
Variable Text Item 657
Win-Loss Chart 657
Win-Loss Measure 658
X Axis 658
X Axis 658
X Axis 658
X Axis Measure 658
Y1 Axis 659
Y2 Axis 659
Y Axis 659
Y Axis 659
Y Axis 659
Y Axis Measure 659
Z Axis 660
Z Axis 660
Z Axis Measure 660
Report Studio Properties 660
3-D Viewing Angle 660
Absolute Position 661
Across 661
Active Report Variable 661
Aggregate Data Item 661
Aggregate Function 661
Aggregation Method 662
Allow 45° Rotation 662
Allow 90° Rotation 662
Allow Skip 662
Allow Sorting 662
Allow Stagger 662
Allow Ties 662
Alternate Text 663
Angular Axis 663
Application 663
Apply Single Class 663
Area Shape 664
Auto Font Size 664
Auto Group & Summarize 664
Auto-Sort 664
Auto-Submit 664

Table of Contents

Auto Truncation 664
Avoid Division by Zero 664
Avoid Label Collision 665
Axis Angles 665
Axis Assignment 665
Axis Assignment 665
Axis Direction 665
Axis Justification 665
Axis Labels 665
Axis Labels 666
Axis Line 666
Axis Line 666
Axis Line 666
Axis Range 666
Axis Title 666
Axis Title 667
Axis Title 667
Background Color 667
Background Effects 667
Background Image 668
Bar, Line, or Area Index 668
Bar Shape 668
Bar Width 669
Based on 669
Baselines 669
Bevel 669
Block Variable 669
Border 669
Border Color 670
Border Color 670
Borders 670
Borders 670
Borders 670
Bottom Position 671
Bottom Position (px) 671
Bottom Position (px) 671
Box Type 671
Box Type 671
Bubble Size 672
Bubble Size 672
Bubble Size 672
Bubble Size 672
Bullet Indicators 672
Button Definition 672
Button Gap 673
Buttons Definition 673
Button Width 673
Calculation Intersection 673
Calendar Type 673

[Caption](#) 673
[Caption](#) 673
[Cardinality](#) 674
[Cascade Source](#) 674
[Case Insensitive](#) 674
[Catalog](#) 674
[Category Axis](#) 674
[Category Baselines](#) 674
[Chart Orientation](#) 675
[Chart Type](#) 675
[Chart Type](#) 675
[Chart Type](#) 675
[Check Boxes Definition](#) 675
[Choices Deselect All Text](#) 675
[Choices Select All Text](#) 676
[Choices Text](#) 676
[Classes](#) 676
[Clickable Region](#) 676
[Clickable Region](#) 677
[Clickable Regions](#) 677
[Clock Mode](#) 677
[Color by Value](#) 677
[Colored Regions](#) 677
[Color Legend Title](#) 677
[Color Legend Title](#) 678
[Column Coordinate](#) 678
[Column Titles](#) 678
[Column Visibility](#) 678
[Combinations](#) 678
[Component Reference](#) 678
[Component Reference](#) 679
[Conditional Palette](#) 679
[Conditional Palette](#) 679
[Conditional Styles](#) 679
[Connecting Lines](#) 680
[Connecting Lines](#) 680
[Connections](#) 680
[Container Filter](#) 680
[Container Select](#) 680
[Contents Height](#) 681
[Control Enable](#) 681
[Cross Product Allowed](#) 681
[Cumulation Axis](#) 681
[Cumulation Axis](#) 681
[Cumulation Label](#) 681
[Cumulation Label](#) 682
[Cumulative Line](#) 682
[Cumulative Line](#) 682
[Current Block](#) 682

Table of Contents

Current Card	682
Current Tab	682
Custom Label	682
Custom Label	683
Data Format	683
Data Item	683
Data Item	683
Data Item Label	684
Data Item Value	684
Data Item Value	685
Data Language	685
Data Points	686
Data Source	686
Days Text	686
Deck Cards Definition	686
Default Card	686
Default Measure	686
Default Selections	687
Default Title	687
Default Title	687
Define Contents	687
Define Member Sets	687
Definition	687
Depth	687
Description	688
Deselect Text	688
Detail	688
Diagram Identifier	688
Dial Face Fill	688
Dial Outline Color	689
Dictionary	689
Dimension	689
Display After Overall Header	689
Display Frequency	689
Display Milliseconds	689
Display Seconds	690
Display Value	690
Down	690
Drill-Through Definitions	690
Drop Shadow	690
Drop Shadow	690
Duplicates	691

Embed 691
End Position 691
Execution Method 691
Execution Optimization 691
Expand Features 692
Exploded Slices 692
Expression 692
Expression 692
Expression 692
Extend Width 693
Face Color 693
Fact Cells Precedence 693
Fill Effects 693
First Column Color 693
First Date 693
First Label Index 694
First Slice Angle 694
Floating 694
Font 694
Font Auto-Sizing 695
Footer 695
Footer 695
Foreground Color 695
From Text 696
Gauge Axes 696
Gauge Axis Colors 696
Gauge Axis Inner Radius 696
Gauge Axis Outer Radius 696
Gauge Border 697
Gauge Labels 697
Gauge Needle 697
Gauge Palette 697
Gauge Pivot 697
Generated SQL 697
Gradient 697
Gridlines 698
Gridlines 698
Gridlines 698
Grouping & Sorting 698
Grouping Type 698
Group Span 698
Has Fact Cells 698
Header Text 699
Heading Level 699
Height (px) 699
Hide Adornments 699
Hide Text 699
Hierarchy 699
Hierarchy Unique Name 700

Table of Contents

Highest Value Text 700
Hole Size (%) 700
Horizontal Alignment 700
Hours Text 701
HTML 701
HTML Source Variable 701
Icon 701
Ignore Data with No Features 702
Include Zero For Auto Scale 702
Insert Text 702
Join Relationships 702
Keywords Text 702
Label 702
Label 703
Label 704
Label Control 704
Label Location 704
Labels 704
Labels 704
Last Date 705
Left Position 705
Left Position (px) 705
Left Position (px) 705
Legend 705
Legend 705
Legend Label 705
Legend Separator 706
Legend Title 706
Legend Title 706
Level 706
Level Indentation 706
Level Unique Name 707
Line and Markers 707
Line Shape 707
Line Style 707
Line Style 707
Line Style 707
Line Styles 708
Line Type 708
Line Weight (pt) 708
List Items Definition 708
Loss Color 708
Lowest Value Text 708
Map & Layers 709

Map Drills 709
Margin 709
Marimekko Totals 709
Marker Color 709
Marker Color 709
Marker Label 710
Markers 710
Markers 710
Marker Shape 710
Marker Shape 710
Marker Size (pt) 710
Marker Text Location 711
Master Detail Relationships 711
Material Effects 711
Material Effects 711
Matrix rows and columns 711
Maximum Characters 712
Maximum Execution Time 712
Maximum Hotspots 712
Maximum Rows Retrieved 712
Maximum Size (pt) 712
Maximum Tables 713
Maximum Text Blob Characters 713
Maximum Truncation Characters 713
Maximum Value 713
MDX 713
Member 714
Member Offset (%) 714
Members 714
Members 714
Member Unique Name 714
Milliseconds Text 715
Minimum Size (pt) 715
Minimum Value 715
Minor Gridlines 715
Minor Gridlines 715
Minor Gridlines 716
Minutes Text 716
Multi-Line 716
Multi-Select 716
Name 716
Name 717
Name 717
Name 717
Name 717
Name 718
Negative Column Color 718
Nested Label Display 718
No Data Contents 718

Table of Contents

No Data Features Size (pt) 718
Node Coordinate 719
Note Border 719
Notes 719
Notes 719
Note Text 719
No Value List Item 719
Number of Regression Lines 720
Numbers Only 720
Number Style 720
Numerical Axis 720
Numerical Axis 720
Numeric Baselines 720
Numeric Value 720
Numeric Value 721
Options Text 721
Orientation 721
Outer Join Allowed 721
Output Aggregation Method 721
Override Dimension Info 721
Overrides 721
Padding 722
Pagination 722
Palette 722
Palette 722
Palette 723
Parameter 723
Pareto Axis 723
PDF Page Setup 723
Percentile 723
Percent of Axis 723
Performance Pattern 724
Pie Labels 724
Plot Area Fill 724
Point Shape 724
Polynomial Exponent 724
Position 724
Position 724
Position in List 725
Position type 725
Position type 725
Position type 725
Positive Column Color 725
Pre-populate 725
Pre-populate Levels 726
Pre-Sort 726
Processing 726
Progressive Axis 726
Progressive Palette 726

Projection List 726
Properties 727
Property 727
Property Unique Name 727
Push To Bottom 727
Query 727
Radar Type 728
Radial Axis 728
Radio Buttons Definition 728
Range 728
Range Label 729
Regression Line 729
Regression Type 729
Relative Alignment 729
Remove Text 729
Render 730
Render Fact Cells 730
Render Page when Empty 730
Render Variable 730
Repeater Direction 731
Report Expression 732
Required 732
Results Deselect All Text 733
Results Select All Text 733
Results Text 733
Reverse Category order 733
Right Position 733
Right Position (px) 733
Rollup Aggregate Function 734
Rollup Processing 734
Root Members Only 734
Rotate Values 735
Row Coordinate 735
Rows Per Page 735
Scale 735
Scale Interval 735
Scope 735
Search Instructions Text 736
Search Text 736
Seconds Text 736
Select UI 736

Table of Contents

Separator 736
Series Color 736
Series Type 737
Set Operation 737
Set Sorting 737
Set Variable Values 737
Share Result Set 738
Show Caption 738
Show Data Points 738
Show Data Range in Legend 738
Show Features with No Data 738
Show Feelers 738
Show Hover 739
Show Icon 739
Show Legend Values 739
Show line 739
Show Pointer Cursor 739
Show Values 739
Show Values 740
Size 740
Size & Overflow 740
Size & Overflow 741
Size Legend Title 741
Slice Direction 741
Slide Animation Direction 741
Solve Order 742
Solve Order 742
Sorting 742
Source Type 742
Source Type 742
Source Type 743
Source Type 743
Spacing & Breaking 743
Spider Effects 743
SQL 743
SQL Join Syntax 744
SQL Syntax 744
Standard Deviations 744
Start Position 744
Static Choices 744
Style Variable 745
Subtitle 745
Subtitle 745
Summarize Small Items 745
Summarize Small Slices 746

Summary Text 746
Suppress 746
Suppression 746
Table Header 746
Table of Contents 747
Table of Contents Name 747
Table Properties 747
Tab Orientation 747
Tabs Definition 747
Tab Width 747
Target Color 748
Target Marker 748
Target Marker Border Color 748
Target Marker Position 748
Target Range (%) 748
Text 748
Text Color 749
Text Flow & Justification 749
Text Orientation 749
Text Source Variable 749
Text Truncation 749
Title 749
Title 750
Tolerance Color 750
Tolerance Label 750
Tooltips 750
Tooltips 750
Tooltips 751
Tooltips 751
Top Position 751
Top Position (px) 751
Total Column 751
Total Column 751
Total Column Color 751
To Text 752
Trendline Label 752
Trendlines 752
Truncation 752
Truncation Text 752
Type 752
Type 752
Type 753
Type 753
Upper Range Skew (%) 753
URL 753
URL Source Variable 753
Usage 754
Use 1.x Behavior 754
Use Aggregate Cache 754

Table of Contents

Use Detail Value on Page	754
Use for Parameter Info	754
Use Local Cache	755
User SAP Member Cache	755
Use Same Range For All Instances	755
Use SAP MUN as Business Key	755
Use Set Aggregation	755
Use SQL Parameters	756
Use SQL With Clause	756
Use Thousands Separator	756
Use Value	756
Use Value	756
Value Location	756
Value Markers	757
Value Representation	757
Values	757
Values	757
Values	757
Value Type	757
Vertical Alignment	758
Visible	758
Visible	758
Visual Angle	758
White Space	759
Width (px)	759
Win Color	759
Win-Loss Threshold	759
X Axis	759
X-Axis Data Item Value	759
Y1 Axis	760
Y2 Axis	760
Y2 Axis Position	760
Y Axis	760
Y-Axis Data Item Value	760
Data Formatting Properties	760
"Not Applicable" Characters	761
Any Error Characters	761
Calendar Type	761
Clock	761
Currency	761
Currency Display	761
Currency Symbol	761
Currency Symbol Position	761
Date Ordering	762
Date Separator	762
Date Style	762
Decimal Separator	762
Display AM / PM Symbols	762
Display As Exponent	762

Display Days	762
Display Eras	762
Display Hours	762
Display Milliseconds	763
Display Minutes	763
Display Months	763
Display Months	763
Display Seconds	763
Display Time Zone	763
Display Weekdays	763
Display Years	763
Display Years	763
Divide By Zero Characters	763
Exponent Symbol	764
Group Size (digits)	764
International Currency Symbol	764
Mantissa (digits)	764
Maximum No. of Digits	764
Minimum No. of Digits	764
Missing Value Characters	764
Negative Pattern	764
Negative Sign Position	765
Negative Sign Symbol	765
No. of Decimal Places	765
Numeric Overflow Characters	765
Padding Character	765
Pattern	765
Percentage Symbol	765
Percent Scale (integer)	765
Scale	766
Secondary Group Size (digits)	766
Security Error Characters	766
Thousands Separator	766
Time Separator	766
Time Style	766
Time Unit	766
Use Thousands Separator	766
Zero Value Characters	767
Appendix A: Calculation Components	769

Operators 769

(769
)	769
*	769
,	769
/	769
	770
+	770
-	770

Table of Contents

< 770
<= 770
<> 770
= 770
> 771
-> 771
>= 771
and 771
auto 771
between 771
case 772
contains 772
currentMeasure 772
default 772
distinct 772
else 773
end 773
ends with 773
for 773
for all 773
for any 773
for report 774
if 774
in 774
in_range 774
is missing 775
is null 775
is not missing 775
is not null 775
like 775
lookup 775
not 776
or 776
prefilter 776
rows 776
starts with 776
then 776
when 776
Summaries 777
aggregate 777
average 777
count 777
maximum 777
median 778
minimum 778
moving-average 778
moving-total 779
percentage 779
percentile 780

quantile	781
quartile	781
rank	782
running-average	783
running-count	783
running-difference	784
running-maximum	784
running-minimum	785
running-total	786
standard-deviation	786
standard-deviation-pop	787
total	787
variance	787
variance-pop	787
Member Summaries	788
aggregate	788
average	788
count	788
maximum	789
median	789
minimum	789
percentage	789
percentile	789
quantile	790
quartile	790
rank	790
standard-deviation	790
standard-deviation-pop	790
total	791
variance	791
variance-pop	791
Constants	791
date	791
date-time	791
time with time zone	791
timestamp with time zone	791
interval	792
interval year	792
interval month	792
interval year to month	792
interval day	792
interval hour	792
interval minute	792
interval second	792
interval day to hour	792
interval day to minute	792
interval day to second	792
interval hour to minute	792
interval hour to second	792

Table of Contents

- interval minute to second 793
- null 793
- number 793
- string 793
- time 793
- Constructs 793
 - if then else 793
 - in_range 793
 - search case 794
 - simple case 794
- Business Date/Time Functions 794
 - _add_days 794
 - _add_months 795
 - _add_years 795
 - _age 795
 - _day_of_week 796
 - _day_of_year 796
 - _days_between 796
 - _days_to_end_of_month 796
 - _first_of_month 797
 - _last_of_month 797
 - _make_timestamp 797
 - _months_between 798
 - _week_of_year 798
 - _years_between 798
 - _ymdint_between 799
- Block Functions 799
 - _firstFromSet 799
 - _remainderSet 800
- Macro Functions 800
 - + 800
 - _add_days 800
 - _add_months 801
 - _add_years 801
 - array 802
 - csv 802
 - dq 802
 - getConfigurationEntry 803
 - grep 803
 - _first_of_month 803
 - _last_of_month 803
 - join 804
 - prompt 804
 - promptmany 805
 - sb 805
 - sq 806
 - sort 806
 - split 806
 - substitute 806

timestampMask	807
toLocal	807
toUTC	808
unique	808
urlencode	809
CSVIdentityName	809
CSVIdentityNameList	809
CAMPassport	809
CAMIDList	810
CAMIDListForType	810
Common Functions	810
abs	810
cast	811
ceil	811
ceiling	811
char_length	812
character_length	812
coalesce	812
current_date	812
current_time	813
current_timestamp	813
exp	813
extract	813
floor	814
ln	814
localtime	814
localtimestamp	815
lower	815
mod	815
nullif	815
octet_length	815
position	816
power	816
_round	816
sqrt	817
substring	817
trim	817
upper	817
Dimensional Functions	818
ancestor	818
ancestors	818
bottomCount	819
bottomPercent	819
bottomSum	819
caption	820
children	820
closingPeriod	820
cousin	821
completeTuple	821

Table of Contents

currentMember 822
defaultMember 822
descendants 823
except 824
filter 824
firstChild 825
firstSibling 825
_format 825
emptySet 826
generate 826
head 826
hierarchize 827
hierarchy 827
item 828
intersect 828
lag 828
lastChild 828
lastPeriods 829
lastSibling 829
lead 829
level 830
levels 830
linkMember 831
members 831
nextMember 831
openingPeriod 831
order 832
ordinal 833
parallelPeriod 833
parent 834
periodsToDate 834
prevMember 834
member 835
nestedSet 835
set 835
siblings 836
tail 836
topCount 836
topPercent 837
topSum 837
tuple 837
union 838
roleValue 838
rootMember 839
rootMembers 839
subset 839
unique 840
value 840
DB2 840

ascii 840
ceiling 840
char 841
chr 841
concat 841
date 841
day 842
dayname 842
dayofweek 842
dayofweek_iso 842
dayofyear 843
days 843
dec 843
decimal 843
difference 843
digits 844
double 844
event_mon_state 844
float 844
hex 844
hour 845
insert 845
integer 845
int 845
julian_day 846
lcase 846
left 846
length 846
locate 847
long_varchar 847
ltrim 847
microsecond 847
midnight_seconds 847
minute 848
month 848
monthname 848
quarter 848
radians 849
repeat 849
replace 849
right 849
round 850
rtrim 850
second 850
sign 850
smallint 851
soundex 851
space 851
substr 851

Table of Contents

table_name 851
table_schema 852
time 852
timestamp 852
timestamp_iso 852
timestampdiff 852
to_char 853
translate 853
trunc 853
truncate 853
ucase 854
value 854
varchar 854
week 854
year 854
DB2 Math 855
DB2 Trigonometry 855
Informix 857
 cardinality 857
 char_length 857
 concat 858
 date 858
 day 858
 extend 858
 hex 858
 initcap 859
 length 859
 lpad 859
 mdy 859
 month 859
 nvl 859
 octet_length 860
 replace 860
 round 860
 rpad 860
 substr 861
 to_char 861
 to_date 861
 trunc 861
 weekday 861
 year 862
Informix Math 862
Informix Trigonometry 862
MS Access 863
 ascii 863
 ceiling 864
 chr 864
 concat 864
 curdate 864

curtime 864
dayname 864
dayofmonth 865
dayofweek 865
dayofyear 865
hour 865
instr 865
lcase 865
left 866
length 866
locate 866
ltrim 866
minute 866
month 866
monthname 867
now 867
position 867
quarter 867
right 867
round 867
rtrim 868
sign 868
space 868
substr 868
substring 869
truncate 869
ucase 869
week 869
year 869
MS Access Cast 869
MS Access Math 870
MS Access Trigonometry 871
Netezza 871
 ascii 871
 chr 872
 decode 872
 initcap 872
 instr 872
 lpad 872
 ltrim 873
 months_between 873
 next_day 873
 nvl 873
 round 873
 rpad 873
 rtrim 874
 substr 874
 {current_db} 874
 {current_user} 874

Table of Contents

{session_user} 874
to_char 874
to_date 874
to_number 875
translate 875
date_trunc 875
trunc 875
version 875
Netezza Math 875
Netezza Trigonometry 876
Netezza Fuzzy 877
Netezza Phonetic 877
Oracle 878
add_months 878
ascii 878
ceil 878
char_length 879
chr 879
concat 879
decode 879
dump 879
greatest 880
initcap 880
instr 880
instrb 880
least 880
length 880
lengthb 881
lpad 881
ltrim 881
months_between 881
new_time 881
next_day 882
nls_initcap 882
nls_lower 882
nls_upper 882
nvl 882
replace 883
round 883
rpad 883
rtrim 883
sign 883
soundex 884
substr 884
substrb 884
{sysdate} 884
to_char 884
to_date 885
to_number 885

translate	885
trunc	885
trunc	885
{user}	885
vsize	886
Oracle Math	886
Oracle Trigonometry	886
Red Brick	888
ceil	888
concat	888
{current_user}	888
date	888
dateadd	888
datediff	889
datename	889
dec	889
decimal	889
decode	889
float	889
ifnull	890
int	890
integer	890
length	890
lengthb	890
ltrim	890
nullif	891
positionb	891
real	891
round	891
rtrim	891
sign	892
string	892
substr	892
substrb	892
time	892
timestamp	893
timestamp	893
to_char	893
SQL Server	893
ascii	893
char	893
charindex	894
{current_user}	894
datalength	894
dateadd	894
datediff	894
datename	894
datepart	895
day	895

Table of Contents

difference	895
getdate	896
left	896
ltrim	896
month	896
patindex	896
replace	897
replicate	897
reverse	897
right	897
round	897
rtrim	897
sign	898
soundex	898
space	898
str	898
stuff	898
year	898
SQL Server Math	899
SQL Server Trigonometry	899
Teradata	901
account	901
add_months	901
bytes	901
case_n	901
char2hexint	902
characters	902
database	902
date	902
format	902
index	902
log	902
nullif	903
nullifzero	903
profile	903
random	903
role	903
session	903
soundex	903
substr	904
time	904
type	904
user	904
vargraphic	904
zeroifnull	904
Teradata Trigonometry	905
SAP BW	907
SAP BW Trigonometry	907
SAP BW Math	908

Sybase 908
 ascii 908
 char 909
 charindex 909
 datalength 909
 dateadd 909
 datediff 909
 datename 910
 datepart 910
 day 910
 difference 910
 getdate 911
 left 911
 ltrim 911
 month 911
 patindex 911
 rand 912
 replicate 912
 reverse 912
 right 912
 round 912
 rtrim 912
 soundex 913
 space 913
 str 913
 stuff 913
 substring 913
 to_unichar 914
 uhighsur 914
 ulowsurr 914
 uscalar 914
 year 915
 Sybase Math 915
 Sybase Trigonometry 915

Postgres 917
 Postgres String 917
 ascii 919
 chr 919
 {current_catalog} 919
 {current_db} 919
 {current_schema} 919
 {current_user} 919
 {session_user} 920
 Postgres Data type formatting 920
 translate 920
 date_trunc 920
 version 921
 Postgres Math 921
 Postgres Trigonometry 921

Table of Contents

Vertica	923
Vertica String	923
ascii	925
chr	925
current_database	925
current_schema	925
{current_user}	925
{session_user}	925
Vertica Data type formatting	925
translate	926
date_trunc	926
version	926
Vertica Math	926
Vertica Trigonometry	927
Paraccel	929
Paraccel String	929
current_database	929
current_schema	929
{current_user}	930
{session_user}	930
Paraccel Data type formatting	930
translate	930
version	930
Paraccel Math	931
MySQL	931
MySQL String	931
ascii	932
database	933
schema	933
session_user	933
system_user	933
version	933
MySQL Math	933
MySQL Trigonometry	934
Greenplum	935
Greenplum String	935
ascii	937
chr	937
current_database	937
current_schema	938
{current_user}	938
{session_user}	938
Greenplum Data type formatting	938
translate	939
date_trunc	939
version	939
Greenplum Math	939
Greenplum Trigonometry	940
Report Functions	941

_add_days 941
_add_months 941
_add_years 942
_age 942
_day_of_week 942
_day_of_year 942
_days_between 942
_days_to_end_of_month 943
_first_of_month 943
_last_of_month 943
_make_timestamp 943
_months_between 943
_week_of_year 944
_years_between 944
_ymdint_between 944
abs 944
AsOfDate 944
AsOfTime 945
BurstKey 945
BurstRecipients 945
ceiling 945
CellValue 945
character_length 945
ColumnNumber 945
CubeCreatedOn 946
CubeCurrentPeriod 946
CubeDataUpdatedOn 946
CubeDefaultMeasure 946
CubeDescription 946
CubeIsOptimized 946
CubeName 947
CubeSchemaUpdatedOn 947
exp 947
extract 947
floor 947
GetLocale 948
HorizontalPageCount 948
HorizontalPageNumber 948
InScope 948
IsAccessible 948
IsBursting 948
IsCrosstabColumnNameMember 948
IsCrosstabRowNodeMember 949
IsFirstColumn 949
IsInnerMostCrosstabColumnNameMember 949
IsInnerMostCrosstabRowNodeMember 949
IsLastColumn 949
IsLastInnerMostCrosstabColumnNameMember 949
IsLastInnerMostCrosstabRowNodeMember 949

Table of Contents

IsOuterMostCrosstabColumnNameMember	950
IsOuterMostCrosstabRowNodeMember	950
IsPageCountAvailable	950
ln	950
Locale	950
lower	950
mapNumberToLetter	950
mod	951
ModelPath	951
Now	951
nullif	951
octet_length	951
PageCount	951
PageName	952
PageNumber	952
ParamCount	952
ParamDisplayValue	952
ParamName	952
ParamNames	952
ParamValue	952
position	953
power	953
ReportAuthorLocale	953
ReportCreateDate	953
ReportDate	953
ReportDescription	953
ReportID	954
ReportLocale	954
ReportName	954
ReportOption	954
ReportOutput	954
ReportPath	954
ReportProductLocale	955
ReportSaveDate	955
round	955
RowNumber	955
ServerLocale	955
ServerName	955
sqrt	956
substring	956
TOCHeadingCount	956
Today	956
trim	956
upper	956
URLEncode	957
Data Type Casting Functions	957
Appendix B: Troubleshooting	963
Problems Creating Reports	963

Division by Zero Operation Appears Differently in Lists and Crosstabs	963
Application Error Appears When Upgrading a Report	963
Nested List Report Containing a Data Item That is Grouped More Than Once Does Not Run After Upgrade	963
Background Color in Template Does not Appear	964
Subtotals in Grouped Lists	964
Chart Labels Overwrite One Another	964
Chart Shows Only Every Second Label	964
Chart Gradient Backgrounds Appear Gray in Internet Explorer	965
Metadata Change in Oracle Essbase Not Reflected in Reports and in the Studios	965
Relationships Not Maintained in a Report With Overlapping Set Levels	965
Summaries in Query Calculations Include Nulls with SAP BW Data Sources	966
Creating Sections on Reports That Access SAP BW Data Sources	966
Error Characters (--) Appear in Reports	966
Columns, Rows, or Data Disappear With SSAS 2005 Cubes	967
Function Unreliable with Sets	968
Report Differences Between TM1 Executive Viewer and IBM Cognos BI with TM1 Data Sources	968
Order of Metadata Tree Differs for TM1 Data Sources	968
Problems Calculating Data	968
Unexpected Summary Values in Nested Sets	968
Null Results for Calculations Using SAP BW Data Sources	969
Incorrect Results with IBM Cognos PowerCubes and Time Measures	969
Incorrect Results in Summaries When Using OLAP Data Sources	970
Problems Filtering Data	971
Unexplained Discrepancies in Number Calculations	971
HRESULT= DB_E_CANTCONVERTVALUE Error When Filtering on a _make_timestamp Column	973
Problems Running Reports	973
Report Runs Slowly	973
Summaries in Report Do not Correspond to the Visible Members	976
Cannot Find the Database in the Content Store (Error QE-DEF-0288)	977
Parse Errors When Opening or Running an Upgraded Report	978
Overflow Error Occurs When a Value in a Crosstab Is More Than 19 Characters	978
The ORA-00907 Error Appears When Running a Report	978
A Report or Analysis Does Not Run Because of Missing Items	978
Cannot View Burst Report	978
A report upgraded from reportNet does not retain its original look	979
Measure Format Disappears in SSAS 2005	980
Drill-through Links are Not Active in the Safari Browser	980
Data Does Not Appear in a Target Report or the Wrong Data Appears	980
A Running Total in Grouped Reports Gives Unexpected Results	981
PCA-ERR-0057 Recursive Evaluation Error	981
Arithmetic Overflow Error When Running a Report in PDF Format	981
RQP-DEF-0177 An error occurred while performing operation 'sqlPrepareWithOptions' status='-69' UDA-SQL-0043 Error	982
Problems When Drilling Through	982
Cannot Drill Through Between PowerCubes Because MUNs Do Not Match	982
Unexpected or Empty Results When Drilling Through	983

Table of Contents

Cannot Drill Through From a Relational Source to a Cube	984
Appendix C: Accessibility Features	985
Keyboard Shortcuts	985
Considerations to Improve Report Accessibility	985
Enable Accessible Report Outputs	986
Add Alternate Text to Images and Charts	987
Add Summary Text to Tables	987
Designate Cells Headers in Tables	988
Example - Conditionally Show a List Below a Chart for an Accessible Report	988
IBM and accessibility	990
Glossary	991
Index	997

Introduction

This document is intended for use with IBM® Cognos® Report Studio. It provides step-by-step procedures and background information to help you create standard and complex reports. Report Studio is a Web product for creating reports that analyze corporate data according to specific information needs.

Audience

To use this guide, you should have

- knowledge of your business requirements
- experience using a Web browser, developing Web applications, and writing reports
- knowledge of databases and data modeling concepts

Finding information

To find IBM® Cognos® product documentation on the web, including all translated documentation, access one of the IBM Cognos Information Centers at <http://publib.boulder.ibm.com/infocenter/cogic/v1r0m0/index.jsp>. Updates to Release Notes are published directly to Information Centers.

You can also read PDF versions of the product release notes and installation guides directly from IBM Cognos product disks.

Using quick tours

Quick tours are short online tutorials that illustrate key features in IBM Cognos product components. To view a quick tour, start IBM Cognos Connection and click the **Quick Tour** link in the lower-right corner of the Welcome page. Quick Tours are also available in IBM Cognos Information Centers.

Accessibility features

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use information technology products. This product has accessibility features. For information on these features, see the accessibility section in this document.

Forward-looking statements

This documentation describes the current functionality of the product. References to items that are not currently available may be included. No implication of any future availability should be inferred. Any such references are not a commitment, promise, or legal obligation to deliver any material, code, or functionality. The development, release, and timing of features or functionality remain at the sole discretion of IBM.

Samples disclaimer

The Great Outdoors Company, GO Sales, any variation of the Great Outdoors name, and Planning Sample depict fictitious business operations with sample data used to develop sample applications for IBM and IBM customers. These fictitious records include sample data for sales transactions, product distribution, finance, and human resources. Any resemblance to actual names, addresses, contact numbers, or transaction values is coincidental. Other sample files may contain fictional data manually or machine generated, factual data compiled from academic or public sources, or data used with permission of the copyright holder, for use as sample data to develop sample applications. Product names referenced may be the trademarks of their respective owners. Unauthorized duplication is prohibited.

Chapter 1: What's New?

This section contains a list of **new**, **changed**, and **removed** features for this release. It will help you plan your upgrade and application deployment strategies and the training requirements for your users.

For information about upgrading, see the IBM® Cognos® Business Intelligence *Installation and Configuration Guide* for your product.

For information about other new features for this release, see the IBM Cognos Business Intelligence *New Features Guide*.

For changes to previous versions, see

- "What's New in Version 8.4" (p. 61)
- "What's New in Version 8.3" (p. 64)

To review an up-to-date list of environments supported by IBM Cognos products, such as operating systems, patches, browsers, Web servers, directory servers, database servers, and application servers, visit www.ibm.com.

New Features in Version 10.1.0

Listed below are new features since the last release.

Use Your Own External Data

You can now supplement your enterprise data with your own external data file. You can import an .xls, .txt, .csv, or .xml file into IBM® Cognos® and create reports that contain your own data. This allows you to leverage the IBM Cognos platform, including the security, data integrity, and full range of reporting functionality.

You can perform self-service reporting without the need to ask your IT department or administrator to set up a data source connection to your file. You import your own data file and start reporting on it right away.

After importing, your external data file is protected by the same IBM Cognos security as your enterprise data, thus allowing you to report on your data in a secure environment.

For more information, see "[Working With Your External Data](#)" (p. 463).

Active Reports

You can now use IBM® Cognos® Report Studio to create active reports. IBM Cognos Active Report is a report output type that provides a highly interactive and easy-to-use managed report. Active reports are built for business users, allowing them to explore their data and derive additional insight.

Active reports make business intelligence easier for the casual user. Report authors build reports targeted at their users' needs, keeping the user experience simple and engaging. Active reports can

be consumed by users who are offline, making them an ideal solution for remote users such as the sales force.

For more information, see "[Active Reports](#)" (p. 215).

Statistical Analysis

IBM Cognos Report Studio now offers statistical functionality in an add-on package called IBM Cognos Statistics.

Report Studio integrates statistical analysis and reporting, providing you with the capability to distribute reports that now can include statistical information. Unlike specialized statistical tools, this functionality is easy to use within the IBM Cognos Business Intelligence authoring environment.

This functionality covers three main areas in statistical analysis:

- the distribution of data
- data analysis and testing
- statistical process control

For more information, see "[IBM Cognos Statistics](#)" (p. 235).

Mozilla Firefox Support

Versions 3.5 and 3.6 of the Mozilla Firefox Web browser are now supported for Report Studio. For a full list of supported software environments, see www.ibm.com.

Chart Enhancements

Report Studio includes many enhancements to charts.

Current Default Charts and Legacy Charts

Report Studio has a new default chart technology. Any new charts added to a report use the current default charts.

The current default charts use more properties that allow you to customize more aspects of the chart. In addition, the current default charts offer you a preview of the style changes that you make to your chart.

If you prefer to continue working with the legacy charts, you can select the **Use legacy chart authoring** option.

You can also convert legacy charts to the current default charts. When you convert a chart to a different chart type, Report Studio maps the chart properties that exist in the new chart type. For more information, see "[Upgrading Legacy Charts](#)" (p. 562) and "[Convert Charts From One Type to Another](#)" (p. 145).

Bullet Charts

You can now create bullet charts to complement your dashboard reports and as an alternative to gauge charts. A bullet chart features a single primary measure, such as current revenue, compares that measure to one or more other measures to enrich its meaning, such as the target or planned

revenue, and displays it in the context of a qualitative range of performance, such as poor, satisfactory, or good. For more information, see "[Bullet Charts](#)" (p. 137).

Chart Matrix

You can now convert a complex nested chart into a matrix configuration that shows multiple small charts arranged in rows and columns. The numeric scale of all the charts is the same so that you can easily compare and analyze values from each chart. For more information, see "[Create a Matrix of Current Default Charts](#)" (p. 184).

Customize the Legend

You can hide or show the legend and change its position relative to the chart object, chart area, or using a specific report expression. For more information, see "[Customize the Legend in a Current Default Chart](#)" (p. 166).

If your legend includes items that are too long, you can truncate long legend items at a specific number of characters. You can also show the values of the data items in your legend. For more information, see "[Customize the Items in the Legend of a Current Default Chart](#)" (p. 167).

Colored Regions

You can now define colored regions in the body of a chart to make your chart more informative. For example, you can divide the background of a scatter chart into quadrants and color each quadrant. For more information, see "[Add Colored Regions in a Current Default Chart](#)" (p. 159).

Enhanced Pie Charts

You can customize how data labels appear in a pie chart, including their format and placement. For more information, see "[Show Data Values or Labels in a Current Default Pie or Donut Chart](#)" (p. 178).

To make your pie charts easier to read and analyze, you can now:

- pull out pie slices from the rest of the pie to highlight them. For more information, see "[Pull Out Pie Slices in a Current Default Pie Chart](#)" (p. 191).
- set the position at which the first slice in a pie chart appears. For more information, see "[Set the Position of the First Slice in a Current Default Pie Chart](#)" (p. 190).
- summarize the smaller slices in a pie chart to avoid having many tiny slices in your pie. For more information, see "[Summarize Small Slices, Bars, or Columns in Current Default Charts](#)" (p. 181).

Enhanced Chart Styles

You can create a new chart palette or edit an existing chart palette, including specifying colors, fill types, and patterns. For more information, see "[Customizing the Color Palette of a Chart](#)" (p. 151).

You can add drop shadows to chart elements, such as lines in line charts. For more information, please see "[Add Background Effects to a Chart Object in a Legacy Chart](#)" (p. 158).

You can customize the color, style, and weight of chart gridlines or set the background of your chart to display bands of color. For more information, see "[Show Gridlines in a Current Default Chart](#)" (p. 165).

You can change the bar shape in a bar or column chart to one of the following two dimensional and three dimensional shapes: box, cylinder, cone, pyramid, rectangle, and triangle. You can also change the width of the bars or columns in a bar or column chart to a percentage of the chart body, such as 80% or 25%.

Chart Axes

To make your charts easier to read, you can now customize the color, style, weight, and transparency of chart axis lines and specify where to display major and minor gridlines on the chart. For more information, see "[Customize the Axes of a Current Default Chart](#)" (p. 162).

Color by Value in Scatter and Bubble Charts

In scatter and bubble charts, you can now specify data to appear in different colors based on the data values. The color of the points or bubbles adds an additional aspect to the chart to assist you in finding relationships in large amounts of data. An entry appears in the legend to identify the color and its value. For more information, see "[Define Color by Value in Current Default Scatter and Bubble Charts](#)" (p. 188).

Bubble Size

In a bubble chart, you can now set the maximum and minimum bubble size and specify the smallest bubble to show in the chart. For more information, see "[Specify the Bubble Size in a Current Default Bubble Chart](#)" (p. 190).

Positioning and Formatting Notes

When you add a note to a chart, you can now set the note's position relative to the sides of the chart area or chart body. You can also use a report expression to position the note next to a data item within the chart. For example, in a bar chart showing revenue for each product line by country, you could type [Country]='Canada' to position the note relative to the Canada bar. You can also customize the background of the note with color and background images. For more information, see "[Add a Note to a Current Default Chart](#)" (p. 169).

Insert Charts With the Toolbar Button

You can now insert a specific chart type from any report with the insert chart button  on the toolbar. When you insert a chart into a report that already contains a list, the list data is added to the chart automatically.

Enhancements to the User Interface

Report Studio includes the following features to enhance the user interface:

- You can now position the **Insertable Objects** and **Properties** panes to the right of the work area, to match the user interface of IBM Cognos Business Insight and IBM Cognos Business Insight Advanced. For more information, see "[Options](#)" (p. 83).

- When working with dimensional data sources, the insert individual members button  and the insert member with children button  have moved to the **Insertable Objects** pane. These buttons have the same function as they did in the previous release. For more information, see "[Insert a Member](#)" (p. 349).
- When working with dimensional and mixed model data sources, you can now switch between viewing the full data tree and the dimensional-only data tree by clicking the view package tree button  and the view members tree button . For more information, see "[Add Dimensional Data to a Report](#)" (p. 347).

These same buttons are available in the expression editor.

- You can now remove all of the filters at once from your report with new options in the filters button . For more information, see "[Focusing Dimensional Data](#)" (p. 371) and "[Focusing Relational Data](#)" (p. 324).
- Expanded tooltips for toolbar buttons now include a description of what the button does. To see a tooltip, pause the pointer over a button in the toolbar. You can turn off the expanded tooltips ([Tools, Options](#)).
- A new **Show Container Selectors** visual aid allows you to select list, crosstab, repeater table, table of contents, active report application objects, and table container objects with one click. When enabled, a small clickable region (three orange dots) appears in the top left corner of these report objects. This selector provides easier access to the container objects' properties and allows you to easily move containers with one mouse gesture. For more information, see "[Visual Aids Button](#)" (p. 78).

Improved Set Operations

You can now perform more operations on a set of members when working with dimensional data sources. You can do the following by right-clicking a set, and clicking **Edit Set**:

- Exclude members from the initial set or from the set as it is currently shown.
For more information, see "[Exclude Members from a Set](#)" (p. 353).
- Move members to the top or bottom of the set.
For more information, see "[Move Members Within a Set](#)" (p. 354).
- Show only the members with the top or bottom values.
For more information, see "[Limit Data to Top or Bottom Values](#)" (p. 354).
- Filter the members in a set.
For more information, see "[Filter the Members Within a Set](#)" (p. 355).
- Expand and collapse members to add its child members below it as new rows.
For more information, see "[Expand and Collapse a Member Within a Set](#)" (p. 356).

Preview Reports when Opening and Saving

You can now see a preview of the report when you open or save it (Tools, Options, Display report preview).

For more information, see "[Options](#)" (p. 83).

Report Name Used for the Exported Output File Name

When you run a report in an export format such as PDF, delimited text (CSV), Microsoft® Excel (XLS), the IBM Cognos report name is now used as the exported file name. This allows you to save the report output using the same name as the original report. For more information, see "[Run a Report](#)" (p. 94).

Enhancements for SAP BW Data Sources

IBM Cognos Business Intelligence includes the following enhancements when creating reports with SAP BW data sources.

Prompting for SAP BW Variables

SAP BW variables are parameters in a SAP BW query that your business warehouse administrator sets up. When the queries run, the SAP BW variables are filled with values. SAP BW variables are automatically exposed as prompts when you run a report or when you add data to a report.

Prompting for SAP BW variables in IBM Cognos BI has improved. You are prompted for optional variables when you initially drag and drop data items.

Time-dependent Hierarchies

Time-dependant hierarchies now automatically reflect hierarchy or structure changes. When a structure is imported into IBM Cognos Framework Manager, each SAP BW time hierarchy is depicted as an individual level. Report Studio users can use these structures to report on and compare levels that are valid for a specific time period.

Column Titles When Drilling up or Down

When you drill down or up, you can now specify whether the column title shows the member caption value or the level label value. For more information, see "[Create a Drill-up and Drill-down Report](#)" (p. 414).

Custom Properties for Prompts

You can now customize the text that instructs your report users how to use report prompts. For example, you can now change the default text that appears above a search text box from **Keywords** to the text of your choice. For more information, see "[Customize Prompt Text](#)" (p. 433).

Simplified Calculations

To add a calculation, you now choose to add either a **Query Calculation** or **Layout Calculation** object from the toolbox tab.

For more information, see "[Using Relational Calculations](#)" (p. 342) and "[Using Dimensional Calculations](#)" (p. 396).

Function Description Improvements

The functions that you can use to create calculations now include improved descriptions and more examples. The descriptions and examples appear in the Report Studio user interface and in this user guide. For more information about functions, see "[Calculation Components](#)" (p. 769).

New Vendor-specific Functions

Functions that are specific to Postgres, Vertica, Netezza, Paraccel, MySQL, and Greenplum now appear in the expression editor in the **Vendor Specific Functions** folder. For more information, see "[Calculation Components](#)" (p. 769).

New Report Style

IBM Cognos BI includes a new default report style with updated colors and gradients. If your report uses a custom report template, your report will appear the same in this version of Report Studio as it did in previous versions. By default, new reports appear in the new report style. For more information, see "[Create and Modify Report and Object Styles](#)" (p. 459).

If you want to create new reports with the custom template style, you can edit the default template for new reports to match your custom template. For more information about custom templates, see "[Creating Report Templates](#)" (p. 549).

List Enhancements

You can now specify whether to automatically include an overall aggregate summary in the list footer. For more information, see "[Lists](#)" (p. 111).

You can also specify whether grouping a column automatically makes it the first column in the list. For more information, see "[Group Relational Data](#)" (p. 308).

You enable these behaviors in the Report Studio options, by selecting the **Group and automatic summary behavior for lists** check box.

A new **Render** property allows you to hide columns from the output of list reports. For more information, see "[Hide Columns in List Reports](#)" (p. 112).

Accessible Report Output

Report Studio now includes features that allow you to create accessible report output. Accessible reports contain features that allow users with disabilities to access report content using assistive technologies, such as screen readers. You can

- add alternative text for non-text objects, such as images, charts, and maps.
- add summary text for crosstabs, lists, repeater tables, and tables.
- specify whether table cells are table headers.

In addition, the documentation now includes alternate text for all graphics so that screen readers can interpret graphics.

Curly Brackets and Parentheses are Breakable

When you run a report in PDF format, curly brackets {} and parentheses () no longer stay on the same line as the text before them. For example, **Products(2004)** may now break to a new line between **Products** and **(2004)**.

Changed Features in Version 10.1.0

Listed below are changes to features since the last release.

Report Studio Express Authoring Mode Replaced by Business Insight Advanced

The two authoring modes in IBM® Cognos® Report Studio, Professional and Express, no longer exist. Report Studio is still the tool that professional report authors use to create advanced and managed reports. The Express authoring mode is now replaced by IBM Cognos Business Insight Advanced. Business Insight Advanced is part of a new report consumption experience that provides an integrated business intelligence experience for business users. It offers much greater capability, such as full support for list reports, charts, and relational data sources. It offers an entirely different user experience that is designed for data exploration.

For more information, see the Business Insight Advanced *User Guide*.

Order of Drawing Bars in Bar Charts

In a horizontal bar chart, the new default chart type draws bars from bottom to top. Legacy charts draw the bars or columns in the reverse order. To control the order of the bars, use the [Reverse Category order](#) property for the category axis.

Removed Features in Version 10.1.0

Listed below are features that are removed since the last release.

Analyze with Analysis Studio

The ability to open a report with IBM® Cognos® Analysis Studio from within IBM Cognos Report Studio (Tools or right-click, **Analyze**) was removed in this release.

Conformed Dimensions on SAP BW Data Sources

Support for conformed dimensions generated by IBM Cognos Framework Manager for SAP BW data sources was removed in this release.

Excel 2000 and Excel 2000 Single Sheet Report Outputs

The Microsoft® Excel 2000 spreadsheet software and Excel 2000 Single Sheet spreadsheet software report outputs are no longer supported in this release.

What's New in Version 8.4

This section contains a list of [new](#) and [changed](#) features for past releases. It also contains a cumulative list of similar information for previous releases. Knowing this information will help you plan your upgrade and application deployment strategies and the training requirements for your users.

To review an up-to-date list of environments supported by IBM® Cognos® products, such as operating systems, patches, browsers, Web servers, directory servers, database servers, and application servers, visit www.ibm.com.

For information about upgrading, see the IBM Cognos Business Intelligence *Installation and Configuration Guide* for your product.

For an overview of new features for this release, see the IBM Cognos Business Intelligence *New Features Guide*.

New Features in Version 8.4

Listed below are new features since the last release. Links to directly-related topics are included.

Query Reuse Cache Enabled by Default

In earlier versions of IBM Cognos Business Intelligence, reports did not reuse queries by default. Query reuse is now the default setting in IBM Cognos Framework Manager models and IBM Cognos Report Studio reports. You can disable query reuse for each individual query in Report Studio, or you can disable query reuse for the entire package in Framework Manager.

To disable query reuse in Report Studio, in Query Explorer, click the query and change the **Use Local Cache** property.

For more information about query reuse, see the Framework Manager *User Guide*.

Enhanced Drill-through Capabilities

In earlier versions of IBM Cognos BI, model-based drill-through supported only reports created in IBM Cognos Analysis Studio, IBM Cognos Query Studio, or IBM Cognos Report Studio as targets. Other types of drill-through targets are now supported. For example, you can now drill through to IBM Cognos PowerPlay® Studio reports saved in the content store or to a package that contains an IBM Cognos PowerCube.

You can now perform dynamic filtering of data in Report Studio reports. When you apply a filter to a source report and then drill through to a target report, the target report is automatically filtered if the item you selected is in a query in the target report.

In Report Studio, you can now define a drill-through definition that allows a report to be shown on the Go To page only when a specific measure, dimension, hierarchy, alternate hierarchy, level, or query item exists in the source report. This allows you to restrict when the drill path is available to users. For more information, see "[Using Drill-through Access](#)" (p. 515).

When the source report is based on a dimensional package and you choose to pass parameter values, you can now choose what property of the source metadata item to map to the target item. For example, you can map the member caption instead of the business key to the target. By default, the business key is used.

New Chart Types

The following new chart types are now available:

- Microcharts are miniature line and column charts. Win-loss charts are microcharts where the value of each bar is either one or minus one. This chart type is often used to represent a win or loss.
- Marimekko charts are 100 percent stacked charts in which the width of a column is proportional to the total of the column's values. Individual segment height is a percentage of the respective column total value.

Line charts now include an option for creating step line charts. A new **Line Type** property for the **Line** object in the chart allows you to use vertical and horizontal lines to connect the data points, forming a step-like progression.

Line charts can now also include opening, closing, minimum, and maximum value markers.

In bubble charts, a new **Bubble Size** property allows you to control the size of bubbles in relation to the actual values of the Z-axis.

Extended Suppression Capabilities

You can now suppress rows, columns, or rows and columns based on zero, null, divide by zero, and overflow values in reports created with relational, dimensionally-modeled relational, and OLAP data sources. For more information, see "["Suppress Empty Cells" \(p. 498\)](#)".

Crosstab Measures Can be Formatted Based on Other Crosstab Measures

You can apply conditional styles to crosstab cells based on an adjacent value in the crosstab. For example, you can conditionally color the bubbles in a bubble chart based on any combination of the measures, and you can conditionally color an entire row of the crosstab based on any measure in the columns. For more information, see "["Highlight Data Using a Conditional Style" \(p. 473\)](#)".

New Color Palettes for Charts

You can now easily match the palette of Report Studio charts with charts from Microsoft® Excel. The **Palette** property provides palettes that match those from various versions of Excel.

You can also use the new Jazz palette, which is compatible with the IBM Cognos Business Insight palette.

For more information about color palettes, see "["Customizing the Color Palette of a Chart" \(p. 151\)](#)".

Enhanced Report Graphics

A greater variety of colorful graphics and backgrounds are now available for dashboards and reports. For example, you can now include borders with rounded corners, transparent background fills and images, drop shadows, and background gradients. For more information, see "["Add Background Effects" \(p. 443\)](#)".

Synchronized Drilling Up and Drilling Down with Multiple Queries

You can now link groups of data items from different queries in a report so that when you drill up or down on one query, data items in the other query also drill up or down. For more information, see "[Create a Drill-up and Drill-down Report](#)" (p. 414).

Lineage Information for Data

You can now view lineage information of a data item to see what the item represents before you add it to a report. A data item's lineage information traces the item's metadata back through the package and the package's data sources. Viewing lineage information ensures that you add the correct data items to a report. For example, you can view the lineage information of a model calculation to see how it was created.

You can view this information from IBM® Cognos® Report Studio, IBM Cognos Query Studio, IBM Cognos Analysis Studio, and IBM Cognos Viewer. For more information, see "[View Lineage Information for a Data Item](#)" (p. 101).

Access to the IBM WebSphere Business Glossary

If you use the IBM WebSphere® Business Glossary, you can now access the glossary from any of the following data objects in Report Studio:

- Query subject
- Query item
- Measure
- Dimension
- Hierarchy
- Level
- Property/attribute
- Top node member
- Member
- Level item

Extended Support for Dimensionally-modeled Relational Data Sources

There is now support for multiple hierarchies of a single dimension for dimensionally-modeled relational data sources.

In addition, the sorting of data by member is enhanced. A member can appear sorted in the metadata tree and on the report without slowing performance. Also, members of a level are sorted in the context of their ancestor levels. This means that the hierarchical relationship of a level is considered when member sorting is applied.

Support for Multiple Hierarchies per Dimension for IBM Cognos and SSAS Cubes

If you use IBM Cognos PowerCubes or Microsoft® SQL Server Analysis Services (SSAS) 2005 cubes, you can now insert multiple hierarchies from a single dimension in the same query in your report. In previous releases, if a query contained multiple hierarchies from the same dimension, you encountered errors or no data appeared in your report. For more information, see "[Insert a Hierarchy](#)" (p. 351).

Intersections (Tuples) Can Include Calculated Members and Measures

You can now insert calculated members and measures in an intersection (tuple). This allows you to sort using calculated members and measures. For more information, see "[Create an Intersection \(Tuple\)](#)" (p. 400).

New Query Properties

The following two new properties allow you to customize queries:

- **User SAP Member Cache** allows you to specify whether to cache members from the hierarchy for SAP BW data sources.
- **Use SQL Parameters** allows you to specify whether generated SQL uses parameters. For information about working with SQL, see "[Working with Queries in SQL or MDX](#)" (p. 389).

For more information about queries in the relational reporting style, see "[Working with Relational Queries](#)" (p. 329). For more information about queries in the dimensional reporting style, see "[Working with Dimensional Queries](#)" (p. 377).

Changed Features in Version 8.4

Listed below are changes to features since the last release. Links to directly-related topics are included.

Shared Queries Between Lists and Repeaters

The Share Query property was renamed to **Share Result Set**.

This property allows you to share an identical query between data containers that use it. To share a query, data containers must be lists, repeaters, or repeater tables and must use the same grouping structure and list of properties. The data containers cannot be part of a master detail relationship. Sharing queries improves performance by minimizing the number of queries executed against the database. For more information, see "[Share Result Set](#)" (p. 738).

Editing SQL

If you edit the SQL of a query, you must now set the Processing property of the query to **Limited Local**. For more information see "[Working with Queries in SQL or MDX](#)" (p. 389).

What's New in Version 8.3

This section contains a list of **new** and **changed** features for past releases. It also contains a cumulative list of similar information for previous releases. Knowing this information will help you plan your upgrade and application deployment strategies and the training requirements for your users.

To review an up-to-date list of environments supported by IBM® Cognos® products, such as operating systems, patches, browsers, Web servers, directory servers, database servers, and application servers, visit www.ibm.com.

For information about upgrading, see the *Installation and Configuration Guide* for your product.

For an overview of new features for this release, see the *New Features Guide*.

New Features in Version 8.3

Listed below are new features since the last release. Links to directly-related topics are included.

Enhanced User Interface When Working With Dimensional Data Sources

IBM Cognos Report Studio now offers several enhancements for when you work with a dimensional data source. You can now

- view a members-oriented source tree in the **Insertable Objects** pane and customize the tree to show the information you want.
- create sets of members by selecting multiple members from within the same hierarchy in the source tree.
- select whether to insert only members, only their children, or both into your report.
- create sectioned page breaks by dropping members or sets in the **Page layers** area.
- create slicer filters by dropping members or sets in the **Context filter** area.

Enhanced Conditional Styles

Conditional styles are now easier to use due to a new streamlined user interface and more options.

You can now apply multiple conditional styles in reports. For example, you can apply one style for data formats on specific objects and apply a second style for conditional highlighting throughout the report.

For more information, see "[Highlight Data Using a Conditional Style](#)" (p. 473).

Extended Crosstab Indentation

Crosstab indentation is extended with new properties to support both outdenting and relative indenting. Relative indenting adds an indent based on the levels as they appear in the report rather than in the cube data source. You can now avoid double-indenting, by which members are taken from both the parent and grandchild levels.

For more information, see "[Indent Data](#)" (p. 123).

Customizable Data Tree

You can now customize the source tree in the **Insertable Objects** pane. You can select options for a metadata view with dimensions, hierarchies, levels, and an optional members folder; or you can select a member-oriented tree view. You can also select a combination of these views. For more information, see "[Customize the Source Tree](#)" (p. 349).

Table of Contents in PDF Reports

You can now add one or more book-like table of contents in your PDF report to summarize all or part of the content of the report. The table of contents provides page numbers and dynamic links to the content.

For more information, see "[Create a Table of Contents](#)" (p. 542).

No Data Handling

When a query returns no data, you can now choose to provide a message or to remove the data item from the report. The new **No Data Contents** property was added to many data containers such as lists, crosstabs, and charts. For more information, see "[Specify What Appears for Data Containers that Contain No Data](#)" (p. 500).

You can also choose not to render the entire report page if all the data containers on the page do not contain any data. For more information, see "[Specify What Appears for Data Containers that Contain No Data](#)" (p. 500).

Simplified Report Style

You can now use a Report Studio style class named **Simplified** styles to remove the default styles defined in the GlobalReportStyles.css file. This class can be used in any report to present a style more appropriate to the financial crosstab style of report. This new class is used in the financial report template.

For more information, see "[Create and Modify Report and Object Styles](#)" (p. 459).

Enhanced Report Expressions

A series of new functions was added in the expression editor to make report expressions more powerful. The new functions give you the flexibility to create report expressions for reporting and conditional processing.

The new functions include a new mathematical function named `mod`, time-based functions, and conversion functions. Of interest to IBM Cognos PowerCube users are the cube variables that can now be added to report output.

The `double2string` function can be used in data type conversions.

The following functions return the named PowerCube properties:

- `CubeName`
- `CubeDescription`
- `CubeCreatedOn`
- `CubeDataUpdatedOn`
- `CubeSchemaUpdatedOn`
- `CubeIsOptimized`
- `CubeDefaultMeasure`
- `CubeCurrentPeriod`

- `CellValue`

The following functions are used to determine where the current cell is within a data frame, such as a list or crosstab:

- `GetColumnNumber`
- `IsFirstColumn`
- `IsLastColumn`
- `GetColumnNumber`
- `GetRowNumber`
- `IsColumnNameMember`
- `IsRowNameMember`
- `IsInnerMostColumnNameMember`
- `IsOuterMostColumnNameMember`
- `IsInnerMostRowNameMember`
- `IsOuterMostRowNameMember`
- `IsLastInnerMostColumnNameMember`
- `IsLastInnerMostRowNameMember`
- `IsFirstColumn`
- `IsLastColumn`
- `IsFirstRow`
- `IsLastRow`
- `GetTableRowNumber`
- `GetTableColumnNumber`
- `InScope`

For more information, see "[Calculation Components](#)" (p. 769).

Singleton Results in Layout

You can now insert single data items anywhere in your report using the new singleton object.

For more information, see "[Insert a Single Data Item](#)" (p. 92).

PDF Horizontal Pagination

Reports can now span two or more page widths in a PDF. This is useful when you want wide data in lists and crosstabs to show in its original size. The fit-to-page option from previous releases is still available. You decide for each list or crosstab which items span across pages and which items shrink to fit a single page.

Several options for horizontal page numbering are provided. For more information, see "[Control Page Breaks and Page Numbering](#)" (p. 545) and "[Insert Page Numbers in a Report](#)" (p. 545).

Page-by-Page PDF Page Layout

You can now set the page orientation (portrait or landscape) and paper size independently on each page of PDF reports. This is useful when some pages within the same report present information that is better suited to a different layout.

The page orientation and size are set as properties of the page. For more information, see "[Set PDF Page Options](#)" (p. 100).

Zooming on Chart Region

You can now zoom into any area of a map report. Because the area of interest is expanded to fit the full report frame, it is no longer necessary to create a separate map in MapInfo for each area of interest or the corresponding drill-through and report definitions.

A single map with multiple layers can be used as the target of authored reports that use a drill-through or a prompt to define the area of interest. The new **Expand Features** property indicates whether the map should be expanded to focus on the filtered features.

For example, a single map of Europe first shows revenue by country and then, when drilling on a country, expands the selected country to fit the report frame.

For more information, see "[Expand Features](#)" (p. 692).

Other New Features

Other new features include the following:

- A new style toolbar allows you to easily apply styles, such as fonts, colors, alignment, borders, and data formats.
- You can now copy and reuse the style information from one object to another.
- You can now save your report output to a file system, such as a local area network.
- A new financial report template is now available with default formatting for financial reports.
- The **Select & Search Prompt** and **Value Prompt** now let you browse through large numbers of prompt values.
- You can now perform a search to find objects with specific characteristics in your report.
- You can now specify a default measure for map, point, scatter, bubble, and polar charts.
- Microsoft® Excel 2007 native spreadsheets are now supported as a report format in addition to the existing Excel HTML formats.

Changed Features in Version 8.3

Listed below are changes to features since the last release. Links to directly-related topics are included.

Special Cell Values Shown Differently

The default formatting for special cell values in reports originally created in IBM Cognos Analysis Studio has changed from 8.2 to 8.3. Special cell values include nulls and missing values, division by zero values, values denied by security, N/A, overflow, or other error values.

You can change the characters for special cell values. For more information, see "Format Data" (p. 495).

Cell Values	IBM Cognos 8.3	IBM Cognos 8.2 and Earlier
Nulls and missing values	(blank)	(blank)
Overflow or underflow	--	#!Overflow
Division by zero	/0	/0
Security (for example, insufficient access rights)	#!Security	#!Security
N/A	--	(blank)
Error	--	#!Error
Casting error	--	#!Casting Error

Slicer Filters Now Affect All Edge Values

Slicer filters now apply to all edge expressions as they do to cell values. For more information, see *Upgrading to Cognos 8 BI 8.3 - Changes in Product Behavior* on www.ibm.com.

Default Calculation Position

The default position for calculations in Analysis Studio has changed from 8.2 to 8.3. In previous releases, calculations were positioned at the end of a set. In a crosstab that contains many items, the calculation may not be visible on the screen. The default position of calculations is now after the last item used as an operand in the calculation.

The position of a calculation in a saved analysis created in Analysis Studio 8.2 may appear in a new position in the crosstab when viewed in Analysis Studio, IBM Cognos Report Studio, or IBM Cognos Viewer in IBM Cognos 8.3. This does not apply to reports originally created in Analysis Studio and then saved in Report Studio.

Changes to Default Data Formats for OLAP Data Sources

The default formats for calculated values have changed when you use the following OLAP data sources:

- Dimensionally-modeled relational data sources
- IBM Cognos PowerCube

- IBM Cognos 8 Planning - Contributor
- Oracle Essbase
- SAP BW

The changes may affect the number of decimal places, the number of group separators, or the rounding precision used in your reports. The following table lists some examples of the changes.

Data format or calculation	Description of change
Group separators	<p>Large percentage results now include group separators (typically a comma at every three digits). For example, 1234.00% becomes 1,234.00%.</p> <p>If none of the values in a calculation or summary use group separators, the results also do not use group separators.</p>
\$0 format	Measures that were formatted with the IBM Cognos format string \$0 (possible with Oracle Essbase) now have decimal places, since this format string does not specify that decimals should be suppressed.
Division	<p>If the numerator is a currency with at least one decimal and the denominator is a query constant, a number, or a percentage, the result now has as many decimals as the currency. In all other cases, the result of a division always has three decimals.</p> <p>If the denominator is a query constant, a number, or a percentage, the result now has the same format type as the numerator. In all other cases, the result of a division is a number.</p>
Square root	If the operand is a currency with decimals, its decimal precision is now used for the result. Otherwise the result has three decimal places.
Rounding	The results now have as many decimals as the rounding precision.

You can override these default formats using IBM Cognos Report Studio or IBM Cognos Query Studio. For more information, see www.ibm.com.

Summaries on Non-additive Measures Give Incorrect Results for SAP BW

If your list or grouped list report includes a summary, such as a total, average, or variance, on a non-additive measure (such as the price of an item or the quantity on hand in an inventory system), error cells, which typically show two dashes (--), are now returned. However, when using an SAP BW data source, the first value is returned as the summary for the non-additive measure instead of two dashes.

In Query Studio, a meaningless number may now appear instead of two dashes.

To obtain correct results, in Report Studio, set the Suppress property for the query to **None**. This option may cause more empty rows to appear in the report. You can also remove the summaries from your report. The report will no longer include a footer with two dashes or empty cells as the summary.

Clause in Aggregate Expressions is No Longer Context-dependent for OLAP

If a crosstab report against a relational data source uses a `for` clause in an aggregate expression, the results do not depend on whether the expression is projected on another edge.

In IBM Cognos 8.1 and 8.2, in many cases for OLAP data sources, the expression was calculated in the context of the opposite edge. As a result, some charts in Query Studio often did not match the corresponding list report.

In IBM Cognos 8.3, the OLAP behavior is now consistent with the relational behavior. If you require context-dependence, use either the bare measure or the member aggregate syntax (`within set` or `within detail`).

For more information, see *Upgrading to Cognos 8 BI 8.3 - Changes in Product Behavior* on www.ibm.com.

Rounding with Equality Filters for SSAS Cubes

In previous versions of IBM Cognos 8, when using a Microsoft SQL Server 2000 or 2005 Analysis Services (SSAS) cube, adding an equality filter with a specific constant implicitly rounded the cells to the number of decimal places for the constant. This caused some inconsistencies when using the following operators: `<`, `>`, `<=`, `>=`, and `<>`.

For example, the following filter uses the not equal (`<>`) operator: `[measures].[BigInt] <> 3.14159`. It now returns the raw values 3.141586 and 3.141592 and no longer rounds the values to five decimal places.

For example, the following filter uses the `=` operator: `[MEASURES].[BigInt] = 3.14159`. In previous releases, the left side of the expression was rounded to 5 digits and raw values of 3.141586 and 3.141592 were returned. In this release, raw values of 3.141586 and 3.141592 are no longer returned, consistent with the behavior for `<>`.

Other data sources were always consistent and did not round values.

If you want to keep the old behavior, use the `round` function or use `([MEASURES].[BigInt] < 3.141595 and [MEASURES].[BigInt] >= 3.141585)`.

Oracle Essbase Calculations on Missing Facts

In IBM Cognos 8 version 8.2, arithmetic query calculations for Oracle Essbase data sources treated missing fact values as zero. In version 8.3, such calculations now have no value (null).

This new behavior is consistent with the behavior of calculated members in Oracle Essbase cubes and with IBM Cognos PowerCubes and SAP BW behavior. In addition, it improves the behavior of null suppression based on such calculations.

If you want to keep these values as zero, you can explicitly convert the null to zero with an expression such as `if ([Discount Percentage] is null) then (0) else ([Discount Percentage])`.

For more information, see *Upgrading to Cognos 8 BI 8.3 - Changes in Product Behavior* on www.ibm.com.

Changes to the Data Tree for SAP BW

If you use an SAP BW data source, you will notice the following changes when working with the data tree:

- If you expand the member folder, you now see all the members.
- If you expand the member folder and the data source uses variables, you are no longer prompted to satisfy the variables.
- If you search in the member folder, you are no longer prompted to satisfy the data source variables.
- If you search for members, the search retrieves all the members that meet the search criteria despite the variable conditions.
- If your data source uses variables and the report includes an IBM Cognos 8 prompt, you are now prompted to satisfy the variables.

Chapter 2: Understanding Report Studio

IBM® Cognos® Report Studio is a Web-based report authoring tool that professional report authors and developers use to build sophisticated, multiple-page, multiple-query reports against multiple databases. With Report Studio, you can create any reports that your organization requires, such as invoices, statements, and weekly sales and inventory reports.

Use Report Studio for reports that

- are intended for a wide audience
- exist long enough to require maintenance for changing requirements and data
- require detailed control over the appearance

Your reports can contain any number of report objects, such as charts, crosstabs, and lists, as well as non-BI components such as images, logos, and live embedded applications that you can link to other information.

Building IBM Cognos Business Intelligence Applications

You use the IBM® Cognos® Business Intelligence components to build reporting and analysis applications.

The lifetime of an IBM Cognos Business Intelligence application can be months, or even years. During that time, data may change and new requirements appear. As the underlying data changes, authors must modify existing content and develop new content. Administrators must also update models and data sources over time. For more information about using data sources, see the *IBM Cognos Administration and Security Guide* and the *IBM Cognos Framework Manager User Guide*.

In a working application, the technical and security infrastructure and the portal are in place, as well as processes for change management, data control, and so on. For information about the workflow associated with creating IBM Cognos BI content, see the *IBM Cognos Architecture and Deployment Guide*. For additional information, see the IBM Cognos Solutions Implementation Methodology toolkit, which includes implementation roadmaps and supporting documents. Information about the toolkit is available on www.ibm.com.

The following graphic provides an overview for how to use IBM Cognos BI to build applications across all of your IBM Cognos BI components.



- ❑ Locate and prepare data sources and models

IBM Cognos BI can report from a wide variety of data sources, both relational and dimensional. Database connections are created in the Web administration interface, and are used for modeling, for authoring, and for running the application.

To use data for authoring and viewing, the business intelligence studios need a subset of a model of the metadata (called a package). The metadata may need extensive modeling in Framework Manager.

Build and publish the content

Reports, scorecards, analysis, dashboards and more are created in the business intelligence studios of IBM Cognos BI. Which studio you use depends on the content, lifespan, and audience of the report, and whether the data is modeled dimensionally or relationally. For example, self-service reporting and analysis are done through IBM Cognos Business Insight Advanced, IBM Cognos Query Studio, and IBM Cognos Analysis Studio, and scheduled reports are created in IBM Cognos Report Studio. Report Studio reports and scorecards are usually prepared for a wider audience, published to IBM Cognos Connection or another portal, and scheduled there for bursting, distribution, and so on. You can also use Report Studio to prepare templates for self-service reporting.

Deliver and view the information

You deliver content from the IBM Cognos portal or other supported portals, and view information that has been saved to portals, or delivered by other mechanisms. You can also run reports, analyses, scorecards, and more from within the business intelligence studio in which they were created.

For information about tuning and performance, see the *IBM Cognos Administration and Security Guide* and www.ibm.com.

Relational vs. Dimensional Reporting Styles

You can create reports in IBM® Cognos® Report Studio using either a relational reporting style or a dimensional reporting style depending on the type of data source you use. The Report Studio tools and query language are the same. However, it is important to choose a reporting style to ensure that you are making the most of your data and to avoid mixing dimensional and relational concepts.

If you are not sure which reporting style to use, ask your Administrator what type of data source you are using. The full list of supported data sources can be found on www.ibm.com.

This user guide is divided into relational and dimensional reporting sections so that you can follow the best practices for using Report Studio with the reporting style you have chosen.

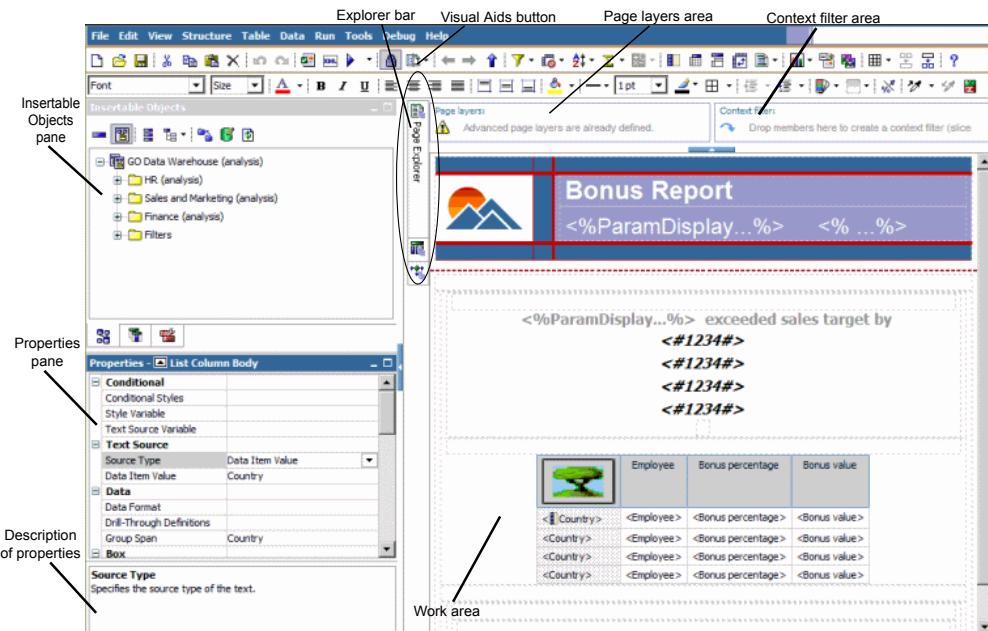
The following table outlines the best practices for both reporting styles. For more information about using the relational reporting style, see "[Relational Reporting Style](#)" (p. 307). For more information about using the dimensional reporting style, see "[Dimensional Reporting Style](#)" (p. 347).

Item	Relational reporting	Dimensional reporting
Report type	Lists Crosstabs Charts Maps	Crosstabs Charts Maps
Data tree	package folder namespace query subject query item measure level	A customizable source tree, including the following: package folder namespace fact measure dimension hierarchy level level attribute member
Data items	Data items using relational and common constructs	Data items using dimensional and common constructs Extended data items
Inserting data	Report Studio groups data automatically by query item. For example, when you add the Product type query item to a list, all product types appear when you run the report.	Report Studio groups data automatically by level. For example, when you add the Product type level to a crosstab, all product types appear when you run the report. You can also create sets of members from different levels within the same hierarchy. For example, you can create a set of members that includes only Cooking Gear and Lanterns and then add that data to a crosstab. You can also choose to insert just the member, just the children of the member, or the member and its children.

Item	Relational reporting	Dimensional reporting
Summarizing data	Member summaries: Aggregate within detail	Member summaries: Aggregate within set
Focusing data	Add a query item and then add a detail or summary filter to view only the data you want to see. For example, add the Quarter query item to a list and filter by Q3.	Add only the relevant members to an edge of the crosstab or to the context filter. For example, only add the Q3 member to your report.
Drilling	Drilling through by value	Drilling through by member Drilling up and down
Page and section breaks	Simple page breaks Page sets Page layers Master detail relationships using parameterized edge or slicer expressions	Simple page breaks Page sets Sections Master detail relationships using parameterized filters
Alternative Studios	IBM Cognos Query Studio	IBM Cognos Analysis Studio

The User Interface

The IBM® Cognos® Report Studio user interface has two panes, an explorer bar, and a work area to help you create reports.



We recommend that you use a screen resolution of at least 1024 by 768 pixels.

Insertable Objects Pane

The **Insertable Objects** pane contains objects that you can add to a report. You add objects to a report by dragging them to the work area.

The **Insertable Objects** pane contains these tabs:

- The **Source** tab contains items from the package selected for the report, such as data items and calculations.
- The **Data Items** tab describes the queries created in the report.
- The **Toolbox** tab contains a variety of objects that you can add to the report, such as text and graphics.

Properties Pane

The **Properties** pane lists the properties that you can set for an object in a report.

You can obtain additional information about a property by selecting it and pressing F1. For example, you can view the list of objects that use each property.

When you specify a value for a property, press Enter, click another property, or save the report to ensure that the value is saved.

Tip: To view a description of the currently selected property at the bottom of the pane, from the View menu, click **Property Descriptions**.

Explorer Bar

Pause the pointer over the following buttons on the Explorer bar to work with different parts of a report:

- the page explorer button 

You use Page Explorer to view or create new report pages and prompt pages or to create and modify classes.

- the query explorer button 

You use Query Explorer to create or modify queries in relational reporting or dimensional reporting and to perform complex tasks, such as defining union joins and writing SQL statements.

- the condition explorer button 

You use Condition Explorer to work with variables to define conditions in a report.

Page Layers Area

Use the **Page layers** area to create sections, or page breaks, in a report to show values for each member on a separate page. For example, you can drag **Northern Europe** sales territory from the **Insertable Objects** pane to the **Page layers** area. The report is broken into a separate page for each territory within northern Europe. Each page's context appears in the report header.

For more information, see ["Create Page Layers" \(p. 547\)](#).

Context Filter Area

When working with dimensional data, use the **Context filter** area to filter your report to show values, or context, for only a specific data item. This technique is also known as a slicer filter. For example, you can drag **Sales Territory** from the **Insertable Objects** pane to the **Context filter** area. When you click a specific territory from the list, the values in the crosstab change to represent data for that territory.

For more information, see ["Create a Context Filter" \(p. 371\)](#).

Visual Aids Button

The visual aids button  provides the following options to help you when you are designing reports in the layout.

Option	Description
Show Boundary Lines	Shows all boundary lines around objects.
Force Boundary Lines	Overrides all boundary line settings by replacing them with default (dotted line) lines. For example, if you set the Border property for an object, the border lines are replaced with dotted lines.
Show Repeating	Repeats objects when you insert them. For example, when you insert a data item in a crosstab, the data item appears in each row or in each column of the crosstab.
Show Page Header & Footer	Shows the page header and page footer.
Show Drag & Drop Padding	Shows drag-and-drop zone when the Padding property for an object is set to 0. If the Padding property is set to a value that is greater than the minimum padding that Report Studio uses to show drag-and-drop zones, only the minimum padding is shown.
Show Hidden Objects	Shows objects for which the Box Type property was set to None or for which the Visible property was set to No .
Show Sorting	Shows the sorting icon for data items for which a sort order was specified. For more information about sorting data, see "Sorting Relational Data" (p. 328) or "Sorting Dimensional Data" (p. 375) .
Show Grouping	Shows the grouping icon for grouped data items.

Option	Description
Show Source Type	Shows the icon for the source type of objects, such as layout calculation.
Show Data Item Type	Shows the icon for the type of data item, such as query item, member, or measure.
Show Drill-through Definitions	Shows data items for which the drill-through definition was defined as a hyperlink.
Show Table of Contents Entries	Shows table of contents entries inserted in the report.
Show Bookmarks	Shows bookmarks inserted in the report.
Show Master Detail Relationships	Shows master detail relationships defined in the report.
	<p>Tip: Pausing the pointer over the master detail relationship icon shows the relationship.</p>
Show No Data Contents Tab Control	Shows tabs if the data container's No Data Contents property is set to Yes .
Show Repeater and Singleton Containers	Shows repeater and singleton containers inserted in the report.
Show Helper Controls	Shows controls in the top right corner of active report application objects.
Show Container Selectors	<p>Shows a small clickable region (three orange dots) in the top left corner of the following container objects and allows you to select them:</p> <ul style="list-style-type: none"> • list • crosstab • repeater table • table of contents • table • active report application objects

Report Authoring Views

IBM® Cognos® Report Studio has two views in which you can author reports: **Page Design** view and **Page Structure** view. You can choose a report authoring view on the **View** menu. Different

options are available in each view, so you often need to use both views. For example, you must use both views to remove sections in relational reporting.

Page Design View

Page Design view is the default view in Report Studio. In this view, you can see what your report will look like after you run it.

Page Structure View

Page Structure view displays an overview of all of the report objects in your report in a tree structure, which is organized by page. For more information, see "[The Page Structure View](#)" (p. 437).

Report Layout and Queries

All reports have two components: a layout component that defines the report appearance and a query component that defines report data. Understanding these components will help you design effective reports.

Layout

A layout is a set of pages that defines the appearance and formatting of a report. When you design the layout of a report, you

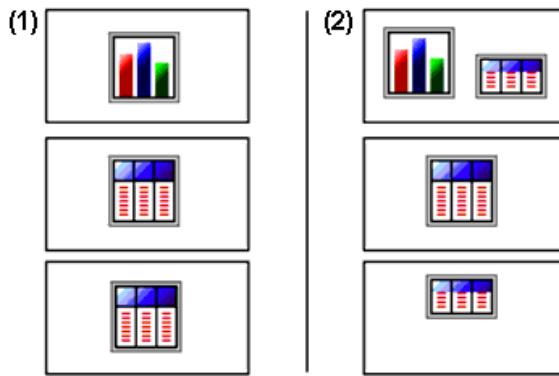
- present the data in a meaningful way by using lists, crosstabs, charts, and maps
- add formatting, such as borders, color, images, and page numbers
- specify how the data flows from one page to the next

Pages

Pages are containers for the layout objects that you use to build a report. A page is made up of the following mandatory and optional components:

- page header (optional)
- page body (mandatory)
- page footer (optional)

When you run a report, the amount of data queried often exceeds one page. As a result, a page will repeat until all the data is shown. You have control over how data flows from one page to the next. For example, here are alternative representations of a report that contains a chart and a lengthy list.



1. The chart appears on its own page. The list begins on the next page and fills subsequent pages until all rows appear.
2. The chart and the first few rows of the list appear on the first page. The rest of the data in the list appears on the subsequent pages.

Objects

You add layout objects to a page when you create a report. Below are objects that you will use often when building reports in IBM® Cognos® Report Studio:

- **list**
Add a list to show data in rows and columns.
- **crosstab**
Add a crosstab to show data in a grid with dimensions along the rows and columns and measures in the cells or intersection points.
- **chart**
- **map**
- **repeater**
Add a repeater to show each instance of a certain column or data item in a separate frame.
- **text**
- **block**
Add a block to hold text or other information. Blocks are often used to lay out horizontal bands of information.
- **table**

For information about other layout objects you can add to a report, see "[Insert a Formatting Object in a Report](#)" (p. 444).

Queries

Queries determine what data items appear in the report. Sometimes you want detailed rows of data, which you obtain by using a simple SELECT statement. Other times you must calculate totals or averages using summary functions and grouped columns or must apply filters to show only the data you want.

Report Studio automatically creates the queries you need as you build reports. However, you can modify these queries or create your own custom queries to get the results you want.

For more information about working with queries in the relational reporting style, see "[Working with Relational Queries](#)" (p. 329). For more information about working with queries in the dimensional reporting style, see "[Working with Dimensional Queries](#)" (p. 377).

Report Objects

You build reports by adding objects and manipulating them to obtain the results you want. To understand how to work with objects in IBM® Cognos® Report Studio, you must be familiar with the following concepts:

- [object types](#)
- [objects as containers](#)
- [locking and unlocking objects](#)
- [hierarchy of objects](#)

For information about Report Studio objects, see "[Report Studio Object and Property Reference](#)" (p. 601).

Object Types

In Report Studio, layout objects are either inline or block. You can insert other objects on the same line as an inline object, but not on the same line as a block object. When you insert an object to the left or to the right of a block object, the object appears on the line above or below the block object, respectively. Examples of inline objects include graphics and text items. Examples of block objects include any report type (list, crosstab, chart, map, or repeater) and tables.

Note: You can also use an object's floating property to define how other objects flow around the object. For example, you can specify how text flows around an image.

Objects as Containers

Objects, such as tables, blocks, and any report frame, are containers in which you can insert other objects. For example, you can insert a list in one cell of a table and a chart in another.

Tip: You can also nest objects to create a sophisticated layout. For example, you can insert a table in a cell of another table.

Locking and Unlocking Objects

To manipulate the contents of some objects, you must first unlock the object. For example, you have a list that contains the column Product Name. You want to insert a graphic inside the Product Name column to show an image of each product. Unlocking the list allows you to insert the image object inside a list column.

Tip: From the **Structure** menu, click **Lock Page Objects**. Toggling this menu item locks and unlocks all layout objects in a report. However, this setting is not saved with the report.

Hierarchy of Objects

In Report Studio, objects are organized hierarchically. For example, a list contains list columns, and each list column contains a text item, which is the name of the inserted data item.

The hierarchy of objects is useful to remember when you apply formatting because formatting is applied to the child objects of the object. For example, you can specify that all list column titles in a list have red as the background color. The formatting is automatically applied to any new columns you add to the list because the formatting is applied to the list and is therefore applied to the objects in the list. If you apply formatting to a specific object, it will override the same formatting specified for the parent object.

Find Objects in a Report

You can quickly locate specific objects in a report by using **Find**.

Steps

1. From the **Edit** menu, click **Find**.
2. Click **Find what** and choose the type of object to find.
3. In the **Options** box, specify the search criteria.
For example, if you want to find the objects that reference a specific data item, you must specify the query that contains the data item and the data item name.
4. Click **Find Next**.

The first object that meets the search criteria is selected in the report. Continue clicking **Find Next** to show all other objects that meet the search criteria.

Note: In some cases, such as searching for a data item that is used to define a conditional style, Report Studio cannot directly select the object in the report. Instead, Report Studio selects the object that uses the object for which you are searching.

Options

You can set various options that control the appearance and behaviors of IBM® Cognos® Report Studio (**Tools**, **Options**).

View Options

Option	Description
Use Windows skin	Replaces the current appearance of the interface with the display scheme specified by the Windows settings.
Show startup dialog	Shows the Welcome dialog box at startup.

Option	Description
Reuse Cognos Viewer window	Reuses the same IBM Cognos Viewer window when you rerun a report without first closing the window.
Resize Cognos Viewer window	Maximizes the IBM Cognos Viewer window when you run a report.
Enable animation	Animates the appearance of dialog boxes, menus, and panes.
Window startup size	Specifies the size of the Report Studio window at startup.
Position pane on the right (requires restart)	Moves the Insertable Objects pane and Properties pane to the right of the work area. This check box is cleared by default. For the change to take effect, you must close and then restart Report Studio.
Show rich tooltips (requires restart)	Specifies whether to show descriptive tooltips when you hover over a button in the toolbar. This check box is selected by default. For the change to take effect, you must close and then restart Report Studio.
Display report preview	Shows a preview of the report when you open or save a report, within the Open, Save, and Save As dialog boxes.

Edit Options

Option	Description
Wrap text in editors	Automatically wraps text in all editors where you can define expressions.
Automatically populate values list	When building expressions in the expression editor, automatically shows values when you browse the data of a data item.
Automatically validate expressions	Automatically validates calculations, such as filters, created in the expression editor. For more information, see " Using Relational Calculations " (p. 342) or " Using Dimensional Calculations " (p. 396).
In-place edit	Enables the editing of text in place when double-clicking.

Option	Description
Use chart flyouts in Design View	<p>When in Page Design view, shows the chart drop zones as flyouts that appear only when you hover your mouse over the chart. When cleared, the chart drop zones are always displayed.</p> <p>By default, this option is not enabled.</p>
Drop replace on crosstab and chart nodes	<p>Specifies what the existing members are replaced with when you drag a new member onto a report.</p>
Double click on member action	<p>When working with dimensional data, specifies what happens when you double-click a member data item.</p> <p>By default, you drill down or up on the item that you double-click.</p>
Layout dimensions	<p>Specifies the width and height of the area where you will create reports.</p>

Report Options

Option	Description
Alias member unique names	<p>When working with a dimensional data source, creates an alias when you add a member to the report or to an expression.</p>
Delete unreferenced query objects	<p>Automatically deletes query objects linked to another object. For example, if you delete a list, the query linked to the list is deleted as well.</p>
Delete unreferenced conditional styles and palettes	<p>Automatically deletes conditional styles or palettes when the last data item that refers to the conditional style or palette is also deleted.</p>

Option	Description
Always create extended data items for new reports	<p>When working with dimensional data sources, determines whether Report Studio creates extended data items or expression-based data items for new reports.</p>
	<p>Expression-based data items allow you to view or edit the expression by double-clicking the Expression property for the item.</p>
	<p>If you are working with a dimensional data source and this option is not selected, the appropriate dimensional data item is added when you insert items such as members, levels, or calculated members. For example, if you insert a level, a level set is created. This makes it easier to work with dimensional data items because Report Studio knows the data item type of the items that are in the report.</p>
Automatic group and summary behavior for lists	<p>When working with lists, automatically adds an overall aggregate summary in the list footer and a summary for any groups in the list. When grouping a column, automatically makes it the first column in the list.</p>
Limit on inserted members	<p>When working with a dimensional data source, limits the number of child members that are inserted. For example, you specify 3 for this option and, in the toolbar, you specify the option to insert children when you drag a member to a data container. You then drag the Camping Equipment member to the rows of a crosstab. What you see as rows are the child members Cooking Gear, Tents, and Sleeping Bags and a row named Others (Camping Equipment) for the remaining child members of Camping Equipment.</p>
Aggregation mode	<p>When working with a dimensional data source, specifies the aggregation type to use when aggregating values in crosstabs and charts.</p>
	<p>Within detail aggregates the visible details.</p>
	<p>Within aggregate aggregates the visible aggregates at the next lower level of detail.</p>
	<p>Within set aggregates the member sets.</p>
	<p>For more information about aggregating values in crosstabs and charts, see ""Summarizing Values in Crosstabs" (p. 361).</p>

Advanced Options

Option	Description
Use operating system clipboard	Uses the Microsoft® Windows® (or other operating system) clipboard instead of the internal Report Studio clipboard.
Allow local file access	Enables the opening and saving of reports on your computer.
Use legacy chart authoring	Enables you to create new reports using the legacy charts instead of the default charts and disables the automatic upgrade of charts in existing reports to the current default charts. Select this check box if you do not want to upgrade the charts in your existing reports.
Disable chart preview in design view	When in Page Design view, displays a static image for a chart instead of updating the chart with a preview of your chart, using simulated data.
Override 10.x styles with 8.x styles on new reports	Specifies whether to use the Version 8.x report styles by default when creating new reports. For more information about report styles, see " Create and Modify Report and Object Styles " (p. 459).
Map feature display limit	When working with maps, specifies the maximum number of features that can appear in a map.
Member display count limit (in source tree)	When working with dimensional data, specifies the maximum number of members that can appear in the Insertable Objects pane when performing a search.

Web Browser Settings

IBM® Cognos® Report Studio can be used in the Microsoft® Internet Explorer and Mozilla Firefox Web browsers. For a full list of supported software environments, see www.ibm.com.

IBM Cognos Business Intelligence uses the default browser configurations provided by Internet Explorer and Firefox. Additional required settings are specific to the browser.

For Internet Explorer, the following settings are required:

- Allow Cookies
- Active Scripting
- Allow META REFRESH
- Run ActiveX controls and plug-ins

- Script ActiveX controls marked safe for scripting
- Binary and Script Behaviors
- Allow programmatic clipboard access
- Enable pop-ups for the IBM Cognos BI server

For Firefox, the following settings are required:

- Enable JavaScript™
- Accept Cookies
- Allow Scripts to disable or replace context menus
- Enable pop-ups for the IBM Cognos BI server

Report Studio uses the native Microsoft Internet Explorer XML support, which is a component of the browser. ActiveX support must be enabled because Microsoft implements XML using ActiveX. IBM Cognos BI does not provide or download ActiveX controls. Only the ActiveX controls that are installed as part of Internet Explorer are enabled through this configuration.

For more information about the Web browser configuration and cookies used by IBM Cognos BI, see the IBM Cognos Business Intelligence *Installation and Configuration Guide*.

The IBM Cognos Software Development Kit

When you create a report, you are creating a report specification. A report specification is an XML file that you can view (**Tools, Show Specification** ). In addition, you can view the specification for a selected object with the **Show Specification (Selection)** menu option.

You can also programmatically create or modify reports by using an editing tool to work with report specifications. You then use the IBM® Cognos® Software Development Kit to implement the reports in your IBM Cognos BI environment. This is useful if, for example, you must make the same modification in many reports. Rather than opening each report and making the change, you can automate the process using the Software Development Kit, thereby saving you time. For more information about the Software Development Kit, contact your local sales office.

Tips

- When you are viewing the report specification, you cannot modify or copy parts of it.
- You can also modify the XML code in a report specification by saving the report specification on your computer.

Chapter 3: Creating a Report

When you create a report, you are actually creating a report specification. The report specification defines the queries and prompts that are used to retrieve data and the layouts and styles used to present the data. For simplicity, the report specification is named by the same name as the report.

Specify the Data Package

Specify the package that will provide items for the report.

The packages that you use to generate reports are based on models that are created in the modeling tool, IBM® Cognos® Framework Manager. A model is a set of related objects, such as query subjects, dimensions, filters, and calculations. When you open a package in IBM Cognos Business Intelligence, these model objects are visible in the left frame.

The package must be previously created and published to the IBM Cognos Connection portal. For more information, see the Framework Manager *User Guide*.

Steps

1. Open IBM Cognos Report Studio with the package you want to use.
2. In the **Welcome** dialog box, choose whether to open a new or existing report or template:
 - To create a new report or template, click **Create a new report or template** and [choose a basic report layout](#).

Tip: You can specify a language other than the default language for your package by clicking on the ellipsis (...) button to the right of the **Package** field, clicking the **Language** ellipsis button in the **Report Package** dialog box, highlighting the desired language, and then clicking OK. As stated on the language selection dialog box, items such as separators and decimals may need to be manually updated for language-specific syntax rules as a result of your selection.
 - To open an existing report or template, click **Open an existing report or template** and select a report.

Objects from the selected package, such as query items, appear on the **Source** tab of the **Insertable Objects** pane.

Tip: You can later change packages.

Refresh the Package

If the package that a report is using has changed, refresh it to ensure that you are working with the latest version.

Steps

1. In the Insertable Objects pane, click the Source tab .
2. Click the refresh button .

Tip: You can also close and reopen the report to upgrade it to the latest version of the package.

Choose a Basic Report Layout

Report Studio includes several basic report layouts that include report objects, such as lists, crosstabs, maps, repeaters, and headers and footers. You can also choose to start with a blank report or open an existing report.

Tip: The Budget vs. Actual sample report in the GO Data Warehouse (analysis) package is based on a basic report layout. For more information about The Great Outdoors Company samples, see "["Sample Reports and Packages" \(p. 567\)](#)".

Steps

1. From the File menu, click New .
2. If you want to change the package, click the ellipsis (...) button and click a different package.
3. Choose a basic report layout:
 - To create a report from a blank layout, double-click **Blank**.
 - To create a report using one of Report Studio's predefined report layouts, double-click the layout.
 - To create a new report template that can be applied to reports created in IBM Cognos Query Studio or IBM Cognos Analysis Studio, double-click **Report Template**.
 - To create a financial report using the basic financial report layout, double-click **Financial Report**. For more information about creating report templates, see "["Creating Report Templates" \(p. 549\)](#)"
 - To create a new report using another report, double-click **Existing**, locate the report, and click **Open**.

Tip: In the Type box, click **Templates** to see only existing templates.

The basic report layout appears in the report page.

Add Data

Select the data items you want to appear in your report.

For more information about adding data to a relational style report, see "[Add Relational Data to a Report](#)" (p. 307). For more information about adding data to a dimensional style report, see "[Add Dimensional Data to a Report](#)" (p. 347).

Step

- In the Insertable Objects pane, on the Source tab , drag data items to the report object. A flashing black bar indicates where you can drop a data item. Data items in the report appear on the Data Items tab .

Data Source Icons

Each object in the data source has a representative icon. You can insert all of the following objects in a report, except for packages and dimensions.

Icon	Object
	Package, which contains the objects you can insert in a report.
	Namespace, used to organize objects.
	Query subject, which represents a table in the database.
	In relational data sources, query item, which represents a column of qualitative data in the database, such as product name or country. In dimensional data sources, level attribute, which represents a property of a level.
	A member is a unique item within a hierarchy. For example, Camping Equipment and 4 Man tent are members of the Products Hierarchy.
	Dimension, which represents a broad grouping of descriptive data about a major aspect of a business, such as products, dates, or markets.
	Hierarchy, which represents a collection of dimensional members organized into a tree structure.

Icon	Object
	<p>Level, which is a set of members that have common attributes. For example, a geographical dimension might contain levels for country, region, and city.</p> <p>Multiple levels can exist within a level hierarchy, beginning with the root level. The root level is the parent and rollup of all members in the first level. It is used to obtain a rollup of all values across the hierarchy and to provide a convenient point to start drilling.</p> <p>For example, a Years level hierarchy may contain the following levels:</p> <ul style="list-style-type: none"> • Root level Years • First level Year • Second level Quarter • Third level Month
	Measure or fact, a query item that represents a column of quantitative data in the database, such as revenue or quantity.
	Measures folder, which contains hierarchical measures.
	Model filter.
	Model calculation.
	Folder, used to organize data items. You cannot import a folder into your report.

Insert a Single Data Item

You can insert a single data item anywhere in your report using the singleton object. The singleton object retrieves only the first row value for that query. Inserting a single data item is useful when you want to show a value that is independent from the rest of the values in the report or when you want to insert some boilerplate text, such as a company name and address. For example, you can add the total revenue value in the header of each page in a report.

You can associate multiple singleton objects with a single query in relational reporting and dimensional reporting to optimize performance, such as when all the data items in the singleton are from the same database table. In addition, two or more singletons can reference data items from the same query. This is useful when using a single query is more efficient to display a set of single values than using multiple queries.

You can also filter the data item in the singleton. For example, you can show the total revenue for only the year 2007.

Queries that are associated to a singleton object are not supported when producing report output in delimited text (CSV) format.

Tip: The Returns by Damage, Failed Orders and Complaints in 2006 sample report in the GO Data Warehouse (analysis) package includes a singleton. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Steps

1. In the **Insertable Objects** pane, on the **Toolbox** tab , drag **Singleton** to the report.
An empty data container is created.
2. From the **Insertable Objects** pane, on the **Source** tab , drag a data item into the **Singleton** container.
Tip: To create a singleton, you can also drag a data item anywhere in your report layout.
3. To change the query associated to the singleton object, in the **Properties** pane, double-click the **Query** property and make changes.

When the report is run, the first row value for the data item is retrieved.

Validate a Report

Validate your report to ensure that it contains no errors.

When you open a report created in a previous version of IBM Cognos BI, it is automatically upgraded and validated. For more information, see "[Upgrading Reports](#)" (p. 561).

Steps

1. From the **Tools** menu, click **Validate Report** .
2. If you require more detail from the validation process, from the **Tools** menu, click **Validate Options** and do the following:
 - Click one of the following validation levels.

Validation level	Description
Error	Retrieves all errors returned from the query.
Warning	Retrieves all errors and warnings returned from the query. This is the default validation level.
Key Transformation	In addition to errors and warnings, retrieves informational messages describing important transformation steps from the report specification to the native query sent to the data source. These messages can show the cause of errors and warnings returned from the query. This feature is only partially implemented at present.

Validation level	Description
Information	Retrieves errors, warnings, key transformations, and other information related to query planning and execution.

- Select the **Interactive data** check box to specify that no query optimization is to be used. The **Interactive data** check box controls how queries will be processed during validation. Clear the **Interactive data** check box to set the **Execution Optimization** property to **All Rows**.
Tip: For more information about the **Execution Optimization** property, see **Execution Optimization** in "[Report Studio Object and Property Reference](#)" (p. 601). A message box appears indicating whether any errors were found in the report.
 - Revalidate your report.
3. If you encounter validation errors and want Report Studio to identify incorrect objects in your report, from the **Tools** menu, click **Auto Correct**. Report Studio provides a list of such objects that you can remove individually to successfully run your report. In some cases, the information or error message is linked to the location of the issue in your report. To go to the location of the issue, click the message, and then click **Select**. If only warnings and information appear in the dialog box, these will disappear when you click **OK**.

Save a Report

Save your report to preserve the modifications you made.

Reports are saved to the IBM Cognos BI server. You can also save your report on your computer.

Steps

1. From the **File** menu, click **Save**  or click **Save As** to save a copy of the report under a different name.
2. If you are saving the report for the first time, specify where to save the report and type a file name.
For information about setting up folders in IBM Cognos Connection for your reports, see the *IBM Cognos Connection User Guide*.
3. Click **Save**.

Run a Report

Run your report to see the data that is retrieved. Save time by validating it first to check for errors.

You can also run a report or a group of reports in IBM Cognos Connection. Furthermore, you can save report outputs to a file system. For more information, see the *IBM Cognos Business Intelligence Administration and Security Guide*.

You can specify not to render a report page if it does not contain any data.

When you run a report in an export format such as PDF, delimited text (CSV), Microsoft® Excel spreadsheet software (XLS), the IBM Cognos report name is used as the exported file name.

If you run a report that uses functions or features not supported by the data source, an error message appears. We recommend that you periodically test your reports while you author them in Report Studio to ensure that you do not encounter multiple error messages when you run the report.

Steps

1. Open a report.
2. If you want to clear parameter values stored on the IBM Cognos BI server, from the **File** menu, click **Clear Parameter Values**.

Parameter values stored by the IBM Cognos BI server include signon, validation, and prompt information. For example, if you define two data source connections in IBM Cognos Connection that point to the same data source, you are prompted to choose one when you run a report.

This information is stored so that you are not prompted each time you run the report.

3. If you want to view only the tabular data, from the **Run** menu, click **View Tabular Data** .

If the report contains multiple queries, you must first click an object, such as a list or crosstab, that uses the query for which you want to view the tabular data.

Use this command to ensure that the right results appear. For example, you create a calculation and you want to ensure it is giving you the results you want.

Tip: You can also view tabular data in Query Explorer, which is useful when you are building queries.

4. If you want to set run options, from the **Run** menu, click **Run Options**.

The default value is the value of the selected corresponding run option in IBM Cognos Connection.

The run options that you set apply only to the current session. When you close Report Studio, the options return to the default settings.

5. Change any values for the current session.

Option	Description
Format	Specify to change the default format from HTML.
Paper size	Specify only if the output format is PDF.
Paper orientation	Specify only if the output format is PDF.

Option	Description
Data mode	<p>Specify how much data is returned:</p> <ul style="list-style-type: none"> • All Data returns all data. • Limited Data limits the amount of data returned based on design mode filters defined in the package. • No Data returns artificial data instead of actual data from the data source. <p>For more information about design mode filters, see the Framework Manager <i>User Guide</i>.</p>
Language	<p>The content language sets the preferred language for the data, IBM Cognos Viewer, dates, and so on.</p>
Rows per page	<p>Specifies the number of rows to appear on each page.</p> <p>A Rows Per Page property exists in the Properties pane for lists and crosstabs. If you set this property, the setting overrides the same-named run option. This property applies to both HTML and PDF outputs. For more information about this property, see "Controlling the Rows Per Page for Multiple Containers in HTML and PDF" (p. 98).</p>
Prompt	<p>Select to be prompted for each prompt defined unless the prompt is defined in a report page.</p> <p>If you clear the Prompt check box, you are prompted only if the report cannot run without user intervention. For example, if a report has a single parameterized filter that is optional, you are not prompted when you run the report.</p>
Include accessibility features	<p>Specifies whether accessibility features, such as alternate text on images and charts and summary text in tables, are included in the report output.</p> <p>This option is also used for conditional layouts in Report Studio. When the option is selected, you can specify that a crosstab is rendered as an accessible alternate to a chart.</p> <p>For more information about how to create accessible reports, see "Considerations to Improve Report Accessibility" (p. 985).</p>

- From the **Run** menu, click one of the options to produce the report in the format you want. You can produce a report in HTML, PDF, **CSV**, various **Microsoft Excel formats**, and **XML**. You cannot produce a report in CSV format if you have more than one query defined in the report unless the additional queries are used for prompts.

The report runs in IBM Cognos Viewer. Once the report has finished running, you can run the report again in the same format or in a different format. If you run the report again in CSV or XLS format, the report will appear in a new browser window.

The options available in IBM Cognos Viewer depend on the capabilities set by the administrator for each user. For more information, see the *IBM Cognos Business Intelligence Administration and Security Guide*.

Running a Report Against a Dimensional Data Source

You can cancel a report that is running against Microsoft SQL Server Analysis Services only during the initial portion of its execution. After this time, the report runs to completion.

The same behavior applies to SAP BW data sources.

In Framework Manager, you can also control the number of levels within a hierarchy from which members, or values, are extracted from the hierarchy to populate a tree prompt.

For SAP BW, you can reduce the number of hierarchy levels to limit the number of nodes by setting the SAP BW variable property `trimHierarchyLevels` to 1. This removes the lowest level from the hierarchy prior to creating the list of nodes.

Units of Measure Notation

When running a report against an SAP BW data source, units of measure are included in the same column as the data values, separated by one space. For example, Celsius and Fahrenheit notations are appended to the end of the value.

If you see an asterisk character (*), one of the following was detected:

- an unknown currency
- a value with an unknown or questionable unit of measure, such as a mixed currency calculation or rollup

Mixed currency values occur when you calculate values with different currencies.

This behavior occurs when you are using an IBM Cognos cube as a data source.

This behavior also occurs for SAP BW data sources.

Unsupported SAP Variable Properties

The following SAP variable properties are not supported:

- Exclusionary ranges appear as an inclusionary prompt.
- Mandatory not initial appears as a mandatory prompt.

When using Business Explorer (BEx) to define variables in your SAP data source, avoid using exclusionary ranges and the mandatory not initial property.

Specify Not to Render a Page If It Does Not Contain Data

You can specify not to render a report page if the page does not contain any data when the report is run.

Steps

1. In the report page, click a data container.
2. In the Properties pane, click the select ancestor button  and click the data container type.
For example, if the data container is a list, click **List**.
3. Set the **Render Page when Empty** property to **No**.
4. Repeat steps 1 to 3 for all other data containers in the page and any table of contents objects.

When you run the report, if no data is produced in all data containers and table of contents objects in a page, the page is not rendered. The page is not rendered even if the page contains other objects, such as text items or images.

Controlling the Rows Per Page for Multiple Containers in HTML and PDF

If you have more than one data container in a report, such as a list and a crosstab, you can control how the report is rendered in HTML and PDF by setting the **Rows Per Page** property for each container.

For HTML output, the report property option **Page break by data container for interactive HTML** controls whether the default number of rows is rendered for each data container on each page.

IBM Cognos BI uses the following rules when rendering reports in HTML and PDF:

- If the **Rows Per Page** property is not set for any of the data containers, 20 rows per page are rendered in HTML and each page is completely filled in PDF. The first data container is rendered until there is no more data, followed by the next container, and so on.

Tip: The number of rows that appear on a PDF page depends on the font size set in the report.

- If the **Rows Per Page** property is set for each data container, the specified numbers of rows are rendered in HTML and PDF on each page until there is no more data.
- If the property is set for only some of the containers, the specified numbers of rows are rendered in HTML and PDF on each page until there is no more data. For the remaining containers, 20 rows per page are rendered on each page in HTML and each page is completely filled in PDF.

For example, you have two lists, List1 and List2. You set the **Rows Per Page** property to 5 for List1. When you run the report in HTML, the first page contains the first 5 rows from List1 followed by the first 15 rows of List2.

- If no data is returned for a data container, an empty container is rendered.

Producing a Report in CSV Format

IBM Cognos BI can produce reports in CSV format so you can open them in other applications, such as Microsoft Excel spreadsheet software. Reports saved in CSV format

- support Unicode data across many client operating systems
- are UTF-16 Little Endian data encoded
- include a BOM (Byte Order Mark) at the beginning of the file
- are tab-delimited
- do not enclose strings in quotation marks
- use a new line character to delimit rows

You can open reports saved in CSV format using a variety of spreadsheet software applications. By default, reports produced in CSV format will appear in the application associated with the .csv file type.

You cannot produce the following in CSV format:

- maps
- charts that do not have at least one category or series
- reports that have more than one query defined in the report, unless the additional queries are used for prompts

In IBM Cognos Connection, you can configure the CSV output to suit your environment. For example, you can specify the character used to delimit fields. For more information, see the IBM Cognos Business Intelligence *Administration and Security Guide*.

Producing a Report in Excel Format

IBM® Cognos® Business Intelligence can produce reports in Microsoft® Excel spreadsheet software format.

The following Excel format options are available:

- **Excel 2007** will produce reports that you can view in Microsoft Excel version 2007.
- **Excel 2002** will produce reports that you can view in Microsoft Excel versions earlier than 2007.

Excel 2007 format renders report output in native Excel XML format, also known as XLSX, that provides a fast way to deliver native Excel spreadsheets to Microsoft Excel 2002, Microsoft Excel 2003, and Microsoft Excel 2007. Users of Microsoft Excel 2002 and Microsoft Excel 2003 must install the Microsoft Office Compatibility Pack, which provides file open and save capabilities for the new format. The output is similar to other Excel formats, with the following exceptions:

- Charts are rendered as static images.
- Row height can change in the rendered report to achieve greater fidelity.
- Column width that are explicitly specified in reports are ignored in Microsoft Excel 2007.
- Merged cells are used to improve the appearance of reports.

Excel 2002 format also offers the following benefits:

- Works with SSL protocol.
- Works with a single signon.

Secure reports can be accessed without subsequent signons because the system automatically identifies users and provides security information.

- Works with Netscape 7.01.
- Spreadsheets are contained in a single file for reliable spreadsheet navigation.

Limitations exist when producing reports in XLS format. For more information, see "["Limitations When Producing Reports in Microsoft Excel Format" \(p. 593\)](#).

Producing a Report in XML Format

XML report outputs save the report data in a format that conforms to an internal schema, xmldata.xsd. You can find this schema file in c10_location/bin.

This format consists of a dataset element, which contains a metadata element and a data element. The metadata element contains the data item information in item elements. The data element contains all the row and value elements.

You can create models from reports and other data that conform to the xmldata.xsd schema. This is useful if you want to use a report as a data source for another report, or if you use a database that cannot be read by Framework Manager. In this case, export the data from the data source to an XML file, in conformance with the xmldata schema, and then open the XML file in Framework Manager.

For more information, see the Framework Manager *User Guide*.

For more information, see the Framework Manager *User Guide*.

You cannot produce the following in XML format:

- maps
- charts that do not have at least one category or series
- reports that have more than one query defined in the report, unless the additional queries are used for prompts

If a report contains more than one data container, such as a crosstab and a list, and both containers use the same query, only the output for the list is produced. If a report contains multiple lists, only the output for the first list is produced. If a report contains multiple crosstabs and multiple lists, only the output for the first list is produced.

Set PDF Page Options

Set PDF page options, such as page orientation and paper size, to control how report pages appear in PDF. You can set PDF page options for individual report pages or for all report pages in a layout.

Tip: The PDF Page Properties sample report in the GO Sales (analysis) package includes PDF page options. For more information about The Great Outdoors Company samples, see "["Sample Reports and Packages" \(p. 567\)](#).

You can also set PDF page options in IBM Cognos Connection. For more information, see the IBM Cognos Connection *User Guide*.

The paper size for PDF output can be a maximum of 200 x 200 inches or 500 x 500 centimeters. A larger paper size produces a blank page.

Steps

1. To set PDF page options for all report pages in a layout, do the following:
 - From the File menu, click **PDF Page Setup** and set the page options.
2. To set PDF page options for an individual page, do the following:
 - Pause the pointer over the page explorer button  and click the page.
 - In the work area, click anywhere on the page.
 - In the **Properties** pane, click the select ancestor button  and click **Page**.
 - Double-click the **PDF Page Setup** property, select the **Override the page setup for this page** check box, and set the page options.

View Lineage Information for a Data Item

View lineage information of a data item to see what the item represents before you add it to a report. Lineage information traces the metadata of an item back through the package and the data sources used by the package. Lineage also displays any data item filters that were added by the report author or that were defined in the data model. Viewing lineage information ensures that you add the correct data items to a report. For example, you can view the lineage information of a model calculation to see how it was created.

Lineage is available only after your administrator has configured it. For more information, see the IBM® Cognos® Business Intelligence *Administration and Security Guide*.

Note: Lineage is not supported in reports that are not linked to packages.

You can use the lineage tool that comes with IBM Cognos BI, or you can use another lineage tool by specifying the URL to the tool in IBM Cognos Administration. Note that if the URL source is secured, the source must be able to prompt users for a password because IBM Cognos BI does not pass security information. IBM Cognos BI also supports the IBM Metadata Workbench as a lineage tool. For more information about configuring other lineage tools, see the IBM Cognos Business Intelligence *Administration and Security Guide*.

Tip: The Customer Returns and Satisfaction sample report in the GO Data Warehouse (analysis) package includes lineage information. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

You cannot use lineage information to troubleshoot queries. For example, lineage information will not explain why a data item is double counted. Also, you cannot view lineage information when running a report from a mobile device.

Before you can access lineage information for a report, your administrator must configure lineage in IBM Cognos Administration. Also, the administrator must enable the lineage capability and grant read permission for you on the report.

Note: The IBM Cognos BI lineage tool shows lineage on a report at its highest level. The lineage does not change after you drill down on a report. Because the selection context used to launch lineage can be affected by drill-down operations, we recommend that you always launch lineage at the highest report level before drilling down on the report. Otherwise, the lineage may not start properly.

Step

- In the **Insertable Objects** pane, on the **Source** tab , right-click the data item and click **Lineage**.

Tip: You can view lineage information for multiple data items at the same time by first Ctrl+clicking the items. The IBM Metadata Workbench does not support viewing lineage for multiple data items at once.

The lineage tool opens showing the lineage information of the selected data item.

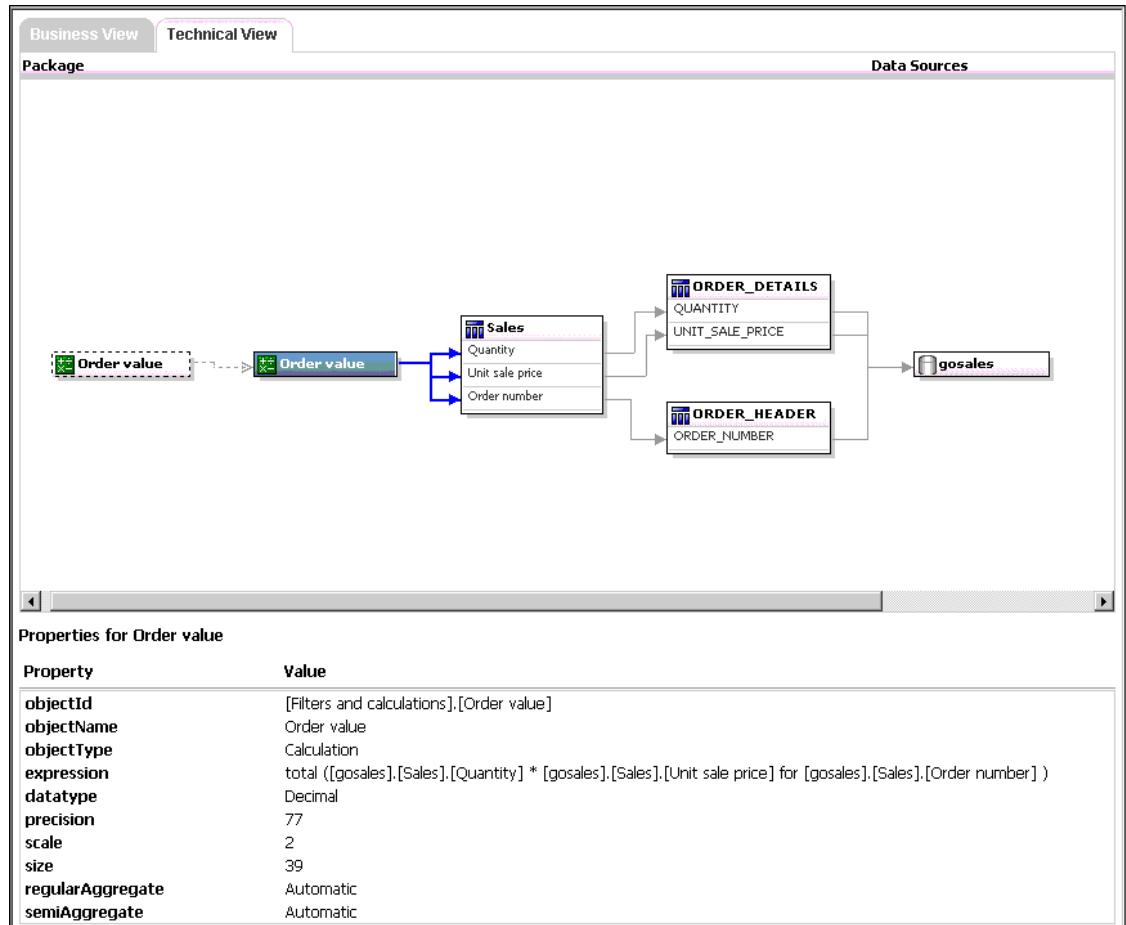
The IBM Cognos BI Lineage Tool

The IBM Cognos BI lineage tool includes two views: the business view and the technical view.

The business view displays high-level textual information that describes the data item and the package from which it comes. This information is taken from IBM Cognos Connection and the Framework Manager model.

The technical view is a graphical representation of the lineage of the selected data item. The lineage traces the data item from the package to the data sources used by the package.

When you click an item, its properties appear below it. If you click an item in the **Package** area, you see the model properties of the item. If you click an item in the **Data Sources** area, you see the data source properties of the item.



You can also view lineage information in IBM Cognos Viewer after you run a report. For example, you can click a cell in a crosstab to see how the cell value was calculated. To view lineage information in IBM Cognos Viewer, right-click an item in the report and then click **Lineage**. If you or an administrator runs a saved report with the IBM Cognos BI lineage tool, both the business view and the technical view are visible. Report consumers can see only the business view. In addition to the **Package** and **Data Sources** areas, a **Report** area exists when looking at the technical view.

Access the IBM WebSphere Business Glossary

Business glossaries help you manage and share an enterprise vocabulary and classification system. If you use the IBM® WebSphere® Business Glossary, you can access the glossary from any of the following data objects:

- Query subject
- Query item
- Measure
- Dimension
- Hierarchy
- Level

- Property/attribute
- Top node member
- Member
- Level item

Step

- Right-click the data item and click **Glossary**.

The IBM WebSphere Business Glossary appears.

Creating Reports for Business Insight Dashboards

When building a dashboard in IBM® Cognos® Business Insight, business users can insert an entire IBM Cognos Report Studio report or insert only individual objects from within a report.

To create effective reports for use in IBM Cognos Business Insight dashboards:

- Ensure that all the objects and pages within the report have meaningful names.
Report Studio gives each object in a report a default name such as List 1, List2, and so on. In Business Insight, these names appear in the **Content** tree. To help business users recognize the report objects, rename them to something more meaningful (**Properties** pane, **Name** property). If you do not want an object to be added in a dashboard, clear the **Name** property for that object. The object will then not appear in the **Content** tree of Business Insight.
- Create a container report that includes all the report objects that dashboard users might want to add in a report. Such a report is not meant to be run and does not need to show report objects in a proper layout. It is only used as a container for the various report parts. Each page of a report appears as a folder within the Business Insight **Content** pane.

For example, on one page named Prompts you can insert all the possible prompts that users might want to add to a dashboard, and give each prompt a useful business name, such as Value Prompt for Region. On a second page named Charts, you can insert a number of charts useful for dashboards.

- Ensure that the titles and labels used in your report are understandable by business users.
- Ensure that you use report objects that are well suited for dashboards. They convey the most information in as little space as possible. For example, if you include a chart, use bullet charts, microcharts, line charts, or column and bar charts. Use conditional blocks to show business indicators.
- If your report includes charts, customize the font sizes so that the charts appear correctly in a dashboard. You may need to reduce the font sizes.
- If your report includes embedded objects, such as a list with an embedded chart and crosstab, and business users insert the list object in their dashboard, the embedded objects appear as blank. Users must insert the entire report. Therefore, you may want to hide the list and

embedded objects from business users by clearing the **Name** property and allowing them to only insert the entire report.

- If your report includes a map, customize the colors of the map in Report Studio so that they match the color palette of the dashboard. You cannot edit the color palette of a map from within a Business Insight dashboard.

Prompts in Business Insight Dashboards

A prompt control, when assigned a **Name** property in Report Studio, appears as an object that you can insert from the **Content** pane in a Business Insight dashboard.

If a dashboard includes reports and report parts from the same model, prompts automatically filter the appropriate data item in all the widgets in the dashboard. For example, a prompt filters on Product Line. Any widget in the dashboard that includes the Product Line data item is filtered when you select a value from this prompt control. If you do not want a prompt to control a widget in the dashboard, you can disable communication between the two widgets.

If a dashboard includes prompted reports or report parts from different models, ensure that the prompts use the same parameter names. For example, a year prompt should use the same parameter **p_Years** in reports from both models. Otherwise, the prompt control is unable to control reports from another package.

Enable Filters in Business Insight Dashboards

You can include a data item in the query of the report, but not show that data item in the actual report. This is useful for creating calculations and filters. If you want Business Insight users to filter content in a dashboard based on a data item that is not visible in the report, you must include this data item in its own query and name it **_BusinessInsight_** within the Report Studio report. Therefore, the data item must exist in both the query used for the data container (such as the list, crosstab, or chart) and in this separate query.

For example, a chart shows the revenue for the product lines for each region. You want to filter the chart to show only data for the year 2008. When you author the report, in addition to including Year in the Query1 used for the chart data container, you create a new query named **_BusinessInsight_** that includes Year.

For more information, see "[Creating Reports for Business Insight Dashboards](#)" (p. 104).

Steps

1. Pause your pointer over the **Query Explorer** and click **Queries**.
2. From the **Insertable Objects** pane, drag a **Query** object to the work area.
3. Click the **Query** object, and in the **Properties** pane, in the **Name** box, type **_BusinessInsight_**.
4. Double-click the **Query** object, and from the **Source** tab of the **Insertable Objects** pane, add the data item to use as a filter.

Ensure that the same data item also exists in the query used for the data container.

Creating Reports for IBM Cognos for Microsoft Office

IBM® Cognos® for Microsoft® Office provides an integrated environment for IBM Cognos products and Microsoft Office. You can use IBM Cognos for Microsoft Office to select pieces of reports to embed in Microsoft Excel workbooks, Microsoft Word documents, or Microsoft PowerPoint presentations, including data, metadata, headers, footers, and charts. You can use predefined reports or you can create new content using IBM Cognos PowerPlay® Web, IBM Cognos Query Studio, IBM Cognos Business Insight Advanced, or IBM Cognos Report Studio.

Note that to access PowerPlay content, your administrator must configure PowerPlay to work with IBM Cognos BI. PowerPlay content that is published only to Upfront is not available to IBM Cognos for Microsoft Office.

Because IBM Cognos for Microsoft Office cannot fully convert highly formatted reports into Excel or other Microsoft document output types, you may not get the results that you want.

To create effective reports for IBM Cognos for Microsoft Office, follow these recommendations:

- Create content to meet specific Microsoft Office integration needs.

For example, in IBM Cognos Business Intelligence, many options are available to format data. Use less formatting to make data more accessible to Office applications.

- Organize reports.

You can publish workbooks to IBM Cognos Connection and organize them with your reports in **Public Folders** or **My Folders**. For more information, see the *IBM Cognos Connection User Guide*. By organizing your content, you can quickly retrieve the information that you want.

Tip: Workbooks, documents, and presentations that are enabled for IBM Cognos for Microsoft Office are identified by their own unique icons, helping you to distinguish them from other types of files.

- Optimize report templates for Microsoft Office.

If you rely on IT personnel or other report authors to create content, request report templates that are optimized for your Microsoft Office integration needs. You may want to request only the data elements or queries that you need and request minimal formatting so that you can more easily use Microsoft Office formatting capabilities with the IBM Cognos content. For example, reports authored in Report Studio can contain list objects embedded within list objects with specific formatting applied. When converted to the tabular representation available in Excel, these reports may not be rendered in the same way in which they appear in IBM Cognos BI.

- Format elements in the Office application.

Instead of formatting objects in IBM Cognos BI, add the formatting in the Office application. By applying less formatting in IBM Cognos BI, you can import more data into the desired locations.

- Label report elements using descriptive names.

This practice makes them more easier to find after you import them. Examples of report elements include lists, crosstabs, and charts.

- Do not nest report objects.

If you nested report objects, some objects may not appear in the correct location, or they may not appear at all. In addition, nesting report objects may cause the following error message to appear:

RDS-ERR-1000 Report Data Service could not process from the content provider.

For example, this error occurs if a repeater or repeater table is inside a block or table. It also appears when layout objects, such as lists, crosstabs, and charts, are in a conditional block that is inside another block or table.

- Keep table sizes small.

For example, because of the size of slides, the maximum number of rows and columns that you can have in Microsoft PowerPoint tables is 25. Although Word and Excel permit larger tables, it takes more time to download and render them.

- Use images with transparent backgrounds.

The background will show through the image in the Office application, making the image look like part of the presentation. If you want, you can then supply your own background color.

- Specify the height and width of images in a list.

This practice ensures that the image appears in the correct size in the Office application.

- Remember that graphs and charts are imported as images.

Images in IBM Cognos BI have image maps associated with them to enable tooltips and hotspots. IBM Cognos BI for Microsoft Office cannot import tooltips and hotspots into Office applications.

- Consider the additional limitations that exist when producing reports in Excel format.

Creating Reports for Mobile Devices

You can send IBM® Cognos® Report Studio reports to mobile devices that have IBM Cognos Mobile installed, or that are using the IBM Cognos Mobile Web portal.

Creating location-aware reports

A location-aware report filters report data based on the user's current location as determined by the GPS coordinates of the mobile device. For example, a Vice President of Sales travels frequently to visit his Regional Sales Managers. When he arrives at an airport, he wants to filter his sales report to show information about the region that he is visiting.

You can create location-aware reports in Report Studio if your IBM Cognos Mobile administrator has set up location-aware parameters. For more information, check with your IBM Cognos Mobile administrator or see the *IBM Cognos Mobile Installation and Administration Guide*.

Location-aware reports must include at least one prompt that requests the location information. This location information, as defined by your IBM Cognos Mobile administrator, could use either coordinates or defined regions, such as states, provinces, cities, or voter regions. Your administrator

also defines the parameter names for the prompts, which you must provide when adding the location prompt.

Unsupported features

In addition, the following report features are not supported for reports that are viewed on mobile devices with IBM Cognos Mobile.

Formatting

- Font size
Font size is supported for reports that are viewed on the Web portal.
- Line-through font style
Line-through font style is supported for reports that are viewed on the Web portal.
- HTML items
HTML items are supported for reports that are viewed on the Web portal.
- Rich text items
- HTML links
HTML links are supported for reports that are viewed on the Web portal.
- Chart border colors, patterns (such as dotted lines), and widths
- Borders on any objects other than crosstabs or lists
- Padding
- Chart frames inside blocks, table cells, or field sets
- Vertical text alignment
- Line height
- Letter spacing
- Page number background
- Text rotation
- Whitespace for formatting, such as tabs

Report Features

- Multi-page widgets in IBM Cognos Business Insight dashboards
Only the first page is shown for reports that are viewed on mobile devices or on the Web portal.
- Static images that are protected with NTLM in reports
Set NTLM security only on the cgi-bin directory. If you choose to enforce NTLM across your static content as well, you will encounter missing static images in your reports.

- Image backgrounds in cells
- Image animations
- Hyperlink buttons
- Bookmarks
- JavaScript™ (or any kind of client-side scripting)
- Field sets
- Drill definitions
- Drilling through on multiple values

Authentication

- Symbian
- JavaScript

Prompting

- Inline prompts ("in-page" prompts)

Prompts are not shown within the report, but the user is prompted instead when they run the report.

- In-page cascading prompts

In-page cascading prompts are supported for reports that are viewed on the Web portal.

- Value prompts displayed as checkboxes

- Customized prompt pages and conditional display of prompts

- Prompts that are directly on a report page

- Multi-select range prompts, such as a prompt that allows users to view data for January 1 to February 15 and February 17 to March 2

Multi-select range prompts are supported for reports that are viewed on the Web portal.

- Japanese Imperial calendar control

- Prompt page layout

Prompt page layout is supported for reports that are viewed on the Web portal.

- Select & Search prompt options

IBM Cognos hardcodes the includes and case insensitive options which are the least restrictive options.

- Seconds and milliseconds in time prompts

- Prompts for data source connections, such as to provide credentials or choose a database

Other Report Features

- IBM Cognos PowerPlay 8 reports used as a source report
- Symbian animated application icons

When using the IBM J9 Java™ Virtual Machine, the buildSettings tool is not supported. In addition, the buildSettings tool for BlackBerry and Windows® Mobile devices is only supported on the Microsoft Windows operating system.

For more information about IBM Cognos Mobile, see the IBM Cognos Mobile *Installation and Administration Guide* and *User Guide*.

Chapter 4: Lists

Use lists to show detailed information from your database, such as product lists and customer lists.

A list shows data in rows and columns. Each column shows all the values for a data item in the database or a calculation based on data items in the database.

Product line	Quantity
Camping Equipment	27,301,149
Golf Equipment	5,113,701
Mountaineering Equipment	9,900,091
Outdoor Protection	12,014,445
Personal Accessories	34,907,705
Overall - Summary	89,237,091

You can specify whether to automatically add an overall aggregate summary in the list footer and a summary for any groups in the list by selecting **Automatic group and summary behavior for lists** in the IBM® Cognos® Report Studio [options](#). The default aggregation as specified in your enterprise data source is used.

For information about converting a list into a crosstab, see "[Change a List into a Crosstab](#)" (p. 123).

Tip: The Order Invoices - Donald Chow, Sales Person sample report in the GO Sales (query) package includes a list. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Set List Properties

Format lists to give them the appearance you want. You can specify formatting for individual column titles or bodies or for all columns in a list. When you specify formatting for all columns in a list, the formatting is automatically applied to new columns you subsequently add.

You can also quickly format lists by applying table styles.

Steps

1. To format an individual column, do the following:

- Click the column title or column body.

To format the text in the column title or the data in the column body, click the unlock button  in the toolbar and then click the title or body.

To format the whole column, click the select ancestor button  in the title bar of the Properties pane and click **List Column**.

- In the Properties pane, set the property value.

For example, to specify a background color, click **Background Color** and choose the color.

2. To format all columns, do the following:
 - Click a column in the list.
 - To format list column bodies, click the select ancestor button in the title bar of the **Properties** pane and click **List Columns Body Style**.
 - To format list column titles, click the select ancestor button in the title bar of the **Properties** pane and click **List Columns Title Style**.
 - To format entire columns, click the select ancestor button in the title bar of the **Properties** pane and click **List Columns**.
 - To format the entire list, click the select ancestor button in the title bar of the **Properties** pane and click **List**.
 - In the **Properties** pane, set the property value.

For example, to specify a background color, click **Background Color** and choose the color.

Formatting for specific columns overrides formatting for entire columns. For example, you specify red as the background color for a specific column body and green as the background color for all columns. When you run the report, the specific column body is red and the remaining columns in the list are green. New columns added to the list will be green.

For more information about formatting reports, see "[Laying Out a Report](#)" (p. 437).

Hide Columns in List Reports

You can hide columns from the output of list reports. If you select to hide a column, the query for that column is still executed.

Steps

1. Select the column that you want to hide.
2. Click the select ancestor button  in the title bar of the **Properties** pane and click **List Column**.
3. Set the **Render** property to **Yes**.

When you run your report, the column is hidden in the report output.

Limitations When Using Set Expressions in List Reports

In list reports, we recommend that you avoid using set expressions. When in a list, set expressions, such as `TopCount`, may produce fewer rows than in the corresponding crosstab.

For example, the following list report includes Year in the first column, followed by a column containing an expression that returns the top three months by revenue. Only three rows appear in the report and they correspond to the top three months across all years. If you group by Year, the report still shows only three months. However, the corresponding crosstab report, you see three months for each year.

Year	Top 3 Months	Revenue
2007	March	171,457,960.73
	July	166,441,982.56
	June	161,995,729.54
	Revenue	Revenue
2007	March	171,457,960.73
	July	166,441,982.56
	June	161,995,729.54
2006	June	141,875,242.63
	December	130,525,197.76
	August	129,235,970.16
2005	December	109,868,142.48
	November	103,650,656.99
	June	100,066,807.17
2004	December	84,197,463.09
	June	82,169,806.98
	July	80,723,457.3

In crosstab reports in IBM® Cognos® BI, set evaluation is always done in the context of what the set is nested under in the report. However, in list reports set evaluation is done independently of the grouping. For example, if the first column contains Country instead of Year, you see the top three months for each country (across all years) in both cases.

In the case of different dimensions, you can force context independence by replacing [Revenue] in the `topCount` expression with `tuple ([Revenue], X)`, where X is the default member of the hierarchy that contains Month.

However, for nested levels in the same hierarchy, there is no such workaround at this time.

Using Repeaters

Use repeaters to repeat items when you run the report. For example, you can use repeaters to create mailing labels, including customer names and addresses.

To build a repeater, drag the **Repeater** or **Repeater Table** object from the **Toolbox** tab to the work area. Use repeaters to repeat items across a single row without a particular structure. For example, you want to create a list that contains Year, and Product line. For each year, you want all product lines to appear in a single row. To do this, create a list with Year as a column and with a repeater as a second column. Then insert Product line into the repeater. Use repeater tables to repeat items in a table structure. Drop the items in the repeater, and modify the properties of the repeater to obtain the results you want. For example, you can specify how many frames appear per page in a repeater table by typing values in the **Across** and **Down** properties.

Convert a List into a Repeater

You can convert a list into a repeater table to take advantage of an existing list.

Steps

1. Click any part of the list.
2. From the **Structure** menu, click **Convert List to Repeater**.

Example - Create Mailing Labels

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create mailing labels for all of the company's retailers.

Steps

1. Open IBM® Cognos® Report Studio with the GO Data Warehouse (query) package.
2. In the Welcome dialog box, click **Create a new report or template**.
3. In the New dialog box, click **Repeater Table** and click **OK**.
4. Click the repeater, click the select ancestor button  in the Properties pane title bar, and click **Repeater Table**.
5. In the Properties pane, set the following properties:
 - Set the **Across** property to 2.
 - Set the **Down** property to 5.
 - Double-click the **Table Properties** property, select the **Fixed size** check box, and click **OK**.
6. In the **Insertable Objects** pane, on the **Toolbox** tab, drag the **Table** object to the repeater.
The **Insert Table** dialog box appears.
7. In the **Number of columns** box, type 1 and click **OK**.
8. Click the table, ensure that you see **Table Cell** in the Properties pane title bar, and modify the following properties:
 - Double-click **Background Image**, click **Specified** and click **Browse**.
 - Click **logo.jpg** and click **OK**.
 - In the **Background Image** dialog box, under **Position**, click the align top right button.
 - Under **Tiling**, click **Do not tile** and click **OK**.
 - Double-click **Size & Overflow**, and in the **Height** box, type **175**, and click **OK**.
9. Click the table, click the select ancestor button in the Properties pane title bar, and click **Table**.
10. In the Properties pane, specify properties for the table:
 - Double-click **Border**.
 - In the **Style** box, click **Solid line**.
 - In the **Width** box, click **1 pt**.
 - In the **Color** box, click **Black**.
 - Under **Preview**, click the apply all borders button  and click **OK**.
 - Double-click **Font**, and under the **Size** box, click **8pt**, and click **OK**.

11. In the **Insertable Objects** pane, on the **Toolbox** tab, drag the **Block** object to the repeater 8 times to create 8 blocks.
12. Drag the **Text Item** object  to the first block:
 - In the **Text** dialog box, type **To:** and click **OK**.
 - Select the text item.
 - Double-click the **Font** property, set the weight to **Bold**, and click **OK**.
13. Click the first block, and, in the **Properties** pane, specify properties for the first block:
 - Double-click the **Padding** property, type **35** in the box on the right, click **mm** as the unit, and click **OK**.
 - Set the **Horizontal Alignment** property to **Center**.
14. In the **Insertable Objects** pane, on the **Source** tab , expand **Sales and Marketing (query)**, **Sales (query)**, and **Retailer site** and drag seven data items to the remaining seven blocks.
15. Ctrl+click the seven blocks to select them, and in the **Properties** pane, specify properties for the seven blocks:
 - Double-click the **Padding** property, type **25** in the box on the left, click **mm** as the unit, and click **OK**.
 - Set the **Horizontal Alignment** property to **Left**.

When you run the report, each page contains 10 mailing labels in two columns.

Chapter 5: Crosstabs

Use crosstab reports, also known as matrix reports, to show the relationships between three or more query items. Crosstab reports show data in rows and columns with information summarized at the intersection points.

For example, the crosstab below shows the gross profit and revenue by product line for each year.

Profitability by Product Line						
		2007			2006	
		Gross profit	Revenue	Cost of Goods Sold	Gross profit	Revenue
Personal Accessories	Glacier Deluxe	1,741,826.22	3,928,757.22	2,186,931	1,669,390.15	3,958
	Glacier GPS Extreme	5,639,265.95	12,034,715.22	6,395,449.27	5,921,887.95	12,249
	TX	8,306,303.78	18,172,476.5	9,866,172.72	14,701,866.32	32,35
	Pocket Gizmo	2,675,373.35	4,380,582	1,705,208.65	3,454,153.34	5,68
	Seeker 35	2,235,849.84	7,754,925.78	5,519,075.94	2,574,835.2	8,93
	Zone	9,434,315.06	28,021,707.95	18,587,392.89	16,263,229.54	49,62
	Mountain Man Deluxe	2,269,799.48	4,563,194.48	2,293,395	2,493,495.99	4,972
	Polar Extreme	347,639.66	701,512.16	353,872.5	499,545.53	998
	Ranger Vision	5,622,674.4	10,525,086.6	4,902,412.2	5,891,875.71	11,71

Tip: The Same Month Prior Year sample report in the Sales and Marketing (cube) package includes a crosstab. For more information about The Great Outdoors Company samples, see "["Sample Reports and Packages" \(p. 567\)](#).

For information about adding microcharts in a crosstab, see "["Insert a Microchart into a Crosstab" \(p. 193\)](#).

Crosstab Nodes and Crosstab Node Members

When you add data items to crosstabs, you create crosstab nodes and crosstab node members. These objects allow you to easily create crosstabs, by dragging data items to crosstabs.

Crosstabs are dimensional objects that have row edges and column edges. Each edge is composed of a set of crosstab nodes. Each crosstab node contains the following:

- One or more crosstab node members.
- Zero or one nested crosstab node, which contains one or more crosstab node members or nested crosstab nodes.

Each crosstab node member refers to a data item that contains an expression to define the members in the crosstab.

The following crosstab contains four crosstab nodes.

Revenue		<#Order year#>	<#Order month#>	<#Total(Order month)#>	<#Order month#>	<#Total(Order month)#>
(1)	<#Total(Product line)#>	<#1234#>		<#1234#>	<#1234#>	<#1234#>
(2)	<#Product line#>	<#Product type#>		<#1234#>	<#1234#>	<#1234#>
		<#Product type#>		<#1234#>	<#1234#>	<#1234#>
	<#Average(Product line) #>		<#1234#>	<#1234#>	<#1234#>	<#1234#>

Crosstab node 1 contains a single node member for the total. This node refers to the data item Total(Product line).

Crosstab node 2 contains a crosstab node member that refers to the data item Product line. This member has a nested crosstab node containing a crosstab node member that refers to the data item Product type.

Tip: Nodes are also created when you add data items to charts.

Moving Crosstab Nodes

Crosstab nodes can be placed anywhere in the crosstab. For example, in the previous diagram, you can drag Order month under Average(Product line) to create a row edge.

To move nested items from one edge to another, ensure that you select the crosstab node and not the crosstab node member. For example, in the previous diagram, you want to move Product line and Product type to the column edge. To do this, click Product line and, in the Properties pane, click the select ancestor button  and click Crosstab Node. Both Product line and Product type are selected.

Crosstab Node Creation Option

In the Structure menu, the **Create Crosstab Nodes** option affects the drag-and-drop behavior in crosstabs. When the option is turned on and you add a data item to a crosstab, the item is created as a crosstab node. If the option is turned off, the item is created as a crosstab node member to an existing crosstab node.

For example, in the previous diagram, if the option is turned on and you drag Country from the Insertable Objects pane to beneath Product line, Country becomes a new peer node to Product line. If the option is turned off, Country becomes a new peer node to Product line and has Product type as a nested crosstab node. This happens because instead of adding a new node, you are adding a new member to the existing node that already contains Product line.

Tip: To create discontinuous crosstabs, turn the **Create Crosstab Nodes** option on. If you want the items on the edges of crosstabs to be related (contain the same nested items), turn the **Create Crosstab Nodes** option off.

Set Crosstab Properties

Format crosstabs to give them the appearance you want. You can specify formatting for rows, columns, and fact cells in a crosstab or for the entire crosstab. When you specify formatting for all rows, columns, fact cells, or the crosstab, the formatting is automatically applied to any new items you add.

Order of Styles Applied

If you apply the same styles, such as font color, to crosstab rows, columns, and crosstab intersections, the styles are applied in the following order:

- crosstab fact cells
- fact cells in the outermost rows
- fact cells in the innermost rows
- fact cells in the outermost columns
- fact cells in the innermost columns
- crosstab intersections

The style applied to the last object in the list overrides styles applied to previous objects. In addition, class styles are applied before styles that you apply manually.

You can also quickly format crosstabs by applying table styles and add white space to a crosstab by inserting crosstab space objects.

Steps

1. To format the entire crosstab, do the following:
 - Click anywhere in the crosstab.
 - Click the select ancestor button  in the title bar of the **Properties** pane and click **Crosstab**.
 - In the **Properties** pane, set the property value.
For example, to specify a background color, click **Background Color** and choose a color.
2. To format all rows, columns, or fact cells, click a row, column, or fact cell in the crosstab and then do the following:
 - If you clicked a row, click the select ancestor button in the title bar of the **Properties** pane and click **Crosstab Rows**.
 - If you clicked a column, click the select ancestor button in the title bar of the **Properties** pane and click **Crosstab Columns**.
 - If you clicked a fact cell, click the select ancestor button in the title bar of the **Properties** pane and click **Crosstab Fact Cells**.
 - In the **Properties** pane, set the property value.
For example, to specify a background color, click **Background Color** and choose the color.
Tip: You can also right-click the row or column and click **Select Fact Cells**.
3. To format all crosstab cells for a specific row or column, do the following:
 - Click the row or column.
 - Click the select ancestor button in the title bar of the **Properties** pane and click **Crosstab Member Fact Cells**.

Tip: You can also right-click the row or column and click **Select Member Fact Cells**.

- In the **Properties** pane, set the property value.

For example, to specify a background color, click **Background Color** and choose the color.

4. To format all row or column titles, do the following:

- Click a row or column title.
- Click the select ancestor button in the title bar of the **Properties** pane and click **Crosstab Rows** or **Crosstab Columns**.
- In the **Properties** pane, set the property value.

For example, to specify a background color, click **Background Color** and choose the color.

5. To format an individual row, column, or intersection, do the following:

- Click the row, column, or intersection.

Tip: To format the data in a row, column, or intersection, click the unlock button  in the toolbar and then click the text item to format.

- In the **Properties** pane, set the property value.

For example, to specify a background color, click **Background Color** and choose the color.

Tip: In cases where fact cell formatting applied to rows conflicts with fact cell formatting applied to columns, you can set the **Fact Cells Precedence** property in the **Properties** pane to determine whether the row formatting or the column formatting has precedence. To set this property, click anywhere in the crosstab, click the select ancestor button in the title bar of the **Properties** pane, and click **Crosstab**.

For more information about formatting reports, see "["Laying Out a Report" \(p. 437\)](#)".

Create a Single-Edge Crosstab

Create a single-edge crosstab report to show data in a list-like form. For example, to show the quantity of products sold for each year and for each order method, you could create a crosstab with **Order Year** and **Order Method** as rows and **Quantity** as the measure.

Steps

1. In the **Insertable Objects** pane, from the toolbox tab , insert a crosstab object to the work area.
2. From the **Source** tab , drag data items to the **Rows** or **Columns** drop zone.
A black bar indicates where you can drop the data item.
3. Repeat step 2 to insert additional data items:
 - If you dragged the data item in step 2 to the **Rows** drop zone, drag the additional items above or below the first item.

- If you dragged the data item in step 2 to the **Columns** drop zone, drag the additional items to the left or right of the first item.
4. To add measures to the crosstab, drag the measures to the **Measures** drop zone.

When you run the report, a crosstab is produced that has only one edge.

Create a Nested Crosstab

Nest data in a crosstab report to compare information by using more than one data item in a column or row. For example, a report shows the number of sales by product line for the past fiscal year. You decide to add a data item to further break down the number of sales by quarter.

When nesting columns in a crosstab report, there are four distinct drop zones where you can insert a new data item. The drop zone you choose will define the relationship between the data item and the column.

Rows

The following relationships are created when you insert a data item as a row:

- Inserting a data item to the left or right of a column creates a parent-child relationship between them.
- When you insert a data item to the left of a column, the data item becomes a parent to the column. When you insert a data item to the right of a column, the data item becomes a child of the column.
- Inserting a data item above or below a column creates a union relationship between them.

Columns

The following relationships are created when you insert a data item as a column:

- Inserting a data item to the left or right of a column creates a union relationship between them.
- Inserting a data item above or below a column creates a parent-child relationship between them.

When you insert a data item above a column, the data item becomes a parent to the column. When you insert a data item below a column, the data item becomes a child of the column.

For example, you have a crosstab with Product line as rows and Quantity and Revenue as nested rows. For columns, you have Order method with Country as a nested column. In this crosstab,

- Product line is a parent to Quantity and Revenue.
- Quantity and Revenue are peers.
- Order method is a parent to Country.

Steps

1. In the **Insertable Objects** pane, on the **Source** tab , click the data item to add.

2. Drag the data item to the report as a nested column or nested row.
A black bar indicates where you can drop the data item.
3. Repeat steps 1 to 2 to add other nested columns or rows.

Tip: If you add more than one measure to a crosstab, all measures appear as columns. You cannot have one measure appear as a row and another as a column. To make all measures appear as rows, swap columns and rows.

Specify the Default Measure

You can specify the default measure for crosstabs and certain chart types. Specify the default measure to be used when the measures cannot be determined by what is on the edges. For example, you create a crosstab with Order method as rows and Product line as columns. You add Quantity and Revenue as nested rows, making Order method their parent. You then add Country under Order method. Since there is no measure specified for Country, you specify the default measure so that data is returned for each country.

In crosstabs, IBM® Cognos® Report Studio automatically sets the default measure when you insert a measure into the crosstab cells.

Steps

1. To specify the default measure for a crosstab, do the following:
 - Click any part of the crosstab, and then click the select ancestor button  in the title bar of the Properties pane.
 - Click **Crosstab**.
 - Set the **Default Measure** property to the default measure.
2. To specify the default measure for a chart, drag the measure to the **Default measure** box in the chart.

Swap Columns and Rows

Swap columns and rows to look at information from a different perspective. This may help you discover high and low points in the data that you hadn't previously noted.

You can only swap columns and rows in a crosstab or chart. In a chart, you swap the x- and y-axes.

Step

- From the toolbar, click the swap rows and columns button .

In the report, the rows become the columns and the columns become the rows.

Indent Data

You can indent crosstab node members to set them apart from surrounding data.

When you insert a hierarchy in rows, all members are automatically indented according to their level. By default, the first member in a set is not indented. If you insert a hierarchy in columns, the members are not automatically indented. You can change the indentation properties of members in the report.

Relative indentation means that the member will shift by one tab when the member's level increases relative to the previous member in the hierarchy.

Indenting based on the level in the hierarchy means that the member will shift the number of tabs equivalent to the level in the hierarchy.

You can also indent objects by applying padding.

Steps

1. Click the crosstab node member to indent.
2. In the **Properties** pane, set the **Level Indentation** property to your indentation type, length, and direction.

Change a List into a Crosstab

Change a list into a crosstab to view your data from a different perspective.

Steps

1. Click the columns to appear as columns or nested columns in the crosstab.
2. From the **Structure** menu, click **Pivot List to Crosstab**.

The list becomes a crosstab with the columns you selected in step 2 appearing as columns and nested columns. The unselected columns, except for measures, appear as rows and nested rows. If you have one measure, it becomes the cells of the crosstab. If you have more than one measure, they appear as columns.

Tip: To make all measures appear as rows, swap columns and rows.

Example - Add Aggregate Data to a Crosstab

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a report showing sales by order method to determine which methods are generating the most revenue and the highest sales volume.

Steps

1. Open IBM® Cognos® Report Studio with the GO Data Warehouse (query) package.
2. In the **Welcome** dialog box, click **Create a new report or template**.

3. In the New dialog box, click **Crosstab** and click **OK**.
4. In the Insertable Objects pane, on the **Source** tab :
 - Expand **Sales and Marketing (query)** and **Sales (query)**.
 - Expand **Product** and double-click **Product line** to add it as rows.
 - Expand **Order method** and double-click **Order method** to add it as columns.
 - Click **Product type** and drag it just to the right of **Product line**.
Product type is now nested in **Product line**.
 - Expand **Sales fact** and drag **Quantity** to the **Measures** drop zone to add it as a measure.
 - Drag **Revenue** to the right of **Quantity** to add it as a second measure.
 - **Tip:** Drop **Revenue** into the crosstab when you see a vertical bar between the **Product type** column and the **Order method** column.
5. Click any part of the crosstab, and then click the select ancestor button  in the title bar of the **Properties** pane.
6. Click **Crosstab**.
7. In the **Properties** pane, double-click the **Font** property.
8. In the **Size** box, click **8pt** and click **OK**.
9. Click one of the measures.
10. Click the summarize button  and click **Maximum**.

When you run the report, you can see that for camping equipment, **Cooking Gear** generated the highest sales volume for the **Special** order method and **Tents** generated the most revenue. **Tents** ordered by the **Web** order method generated the largest revenue.

		E-mail		Fax		Mail		Sales visit		Special	
		Quantity	Revenue	Quantity	Revenue	Quantity	Revenue	Quantity	Revenue	Quantity	Revenue
Camping Equipment	Cooking Gear	667,254	12,660,160.31	222,711	4,375,855.61	171,358	3,349,528.79	1,435,032	29,955,819.16	115,031	2,344,882
	Lanterns	263,370	6,786,981.1	61,991	1,724,287.33	56,107	1,486,107.02	535,633	14,243,626.26	30,150	835,346.38
	Packs	151,719	18,578,756.65	38,840	5,737,253.68	34,452	4,260,591.65	276,738	35,504,440.57	14,940	2,659,194.73
	Sleeping Bags	174,310	14,711,086.99	48,580	5,894,604.37	39,916	3,997,819.92	325,265	35,363,204.56	20,638	2,806,666.82
	Tents	156,431	23,162,109.58	41,836	5,322,397.49	46,225	8,254,596.71	327,086	53,544,871.32	22,769	3,742,899.51
	Maximum (Product type)	667,254	23,162,109.58	222,711	5,894,604.37	171,358	8,254,596.71	1,435,032	53,544,871.32	115,031	3,742,899.51
Golf Equipment	Golf Accessories	194,588	3,446,977.94	60,831	1,064,185.81	44,488	835,353.31	157,959	2,594,946.04	27,454	398,608.96
	Irons	23,707	15,888,260.28	8,281	5,247,384.18	6,986	4,592,152.49	21,167	13,820,312.42	2,873	1,843,743.54
	Putters	95,756	8,490,684.25	26,600	2,213,771.51	23,720	1,921,430.83	67,412	5,645,827.58	6,309	629,444.44
	Woods	19,249	20,108,010.69	6,939	6,715,961.77	5,238	5,344,350.85	17,250	17,179,832.69	1,949	2,092,966.03
	Maximum (Product type)	194,588	20,108,010.69	60,831	6,715,961.77	44,488	5,344,350.85	157,959	17,179,832.69	27,454	2,092,966.03
Mountaineering Equipment	Climbing Accessories	138,242	2,038,855.77	172,803	2,390,328.78	46,488	679,297.57	598,362	8,432,511.02	63,772	802,217.59
	Rope	8,118	2,237,177.54	10,804	3,006,587.53	4,209	1,133,910	45,383	12,623,472.19	5,350	1,395,807.43
	Safety	17,881	1,283,131.17	36,195	2,658,809.76	8,065	592,500.66	125,179	9,265,767.61	8,386	625,119.3
	Tools	34,973	1,917,287.48	72,606	3,792,644.01	22,497	1,125,950.43	272,313	14,294,875.82	16,348	850,863.79
	Maximum (Product type)	138,242	2,237,177.54	172,803	3,792,644.01	46,488	1,133,910	598,362	14,294,875.82	63,772	1,395,807.43

Example - Create a Discontinuous Crosstab

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a report showing sales for each product line by quarter and by order method. Since the report will have columns with data from different dimensions, you create a discontinuous crosstab report. Discontinuous crosstabs are also known as disconnected or disjoint crosstabs or crosstabs with unrelated columns.

Steps

1. Open IBM® Cognos® Report Studio with the GO Data Warehouse (query) package.
 2. In the Welcome dialog box, click **Create a new report or template**.
 3. In the New dialog box, click **Crosstab** and click **OK**.
 4. In the Insertable Objects pane, on the Source tab :
 - Expand Sales and Marketing (query) and Sales (query).
 - Expand Product and drag Product line to the Rows area.
 - Expand Order method and drag Order method to the Columns area.
 - Expand Sales fact and drag Revenue to the Measures area.
 5. Pause the pointer over the query explorer button  and click **Query1**.
 6. On the Toolbox tab , drag **Data Item** to the Data Items pane.
- The **Data Item Expression** dialog box appears.
7. In the Expression Definition box, type the following and click **OK**:

[Employee summary (query)].[Employee by region].[Region]

8. In the Properties pane, set the Name property to Region.
9. Pause the pointer over the page explorer button  and click Page1.
10. On the Source tab, expand Retailer site and drag Region to the left of Order method.

Tip: Drop Region into the crosstab when a flashing vertical bar appears between the Product line column and the Order method column. Otherwise, Region may appear as a nested row instead of a column.

11. Click Region.
12. In the Properties pane, double-click the Sorting property.
13. From the Data Items pane, drag Region to the Sort List pane.
14. Click the sort order button  to sort quarters in ascending order and click OK.
15. Run the report.

Revenue	Americas	Asia Pacific	Central Europe	Northern Europe	Southern Europe	E-mail	Fax	Mail
Camping Equipment	481,445,781.04	421,639,391.62	343,645,848.36	180,851,396.88	161,454,246.13	75,899,094.63	23,054,398.48	21,348,644.09
Golf Equipment	217,262,995.22	193,677,873.68	153,632,833.39	84,424,300.9	77,413,364.7	47,933,933.16	15,241,303.27	12,693,287.48
Mountaineering Equipment	123,127,397.88	107,505,775.01	88,051,532.89	46,091,108.04	44,884,319.08	7,476,451.96	11,848,370.08	3,531,658.66
Outdoor Protection	23,002,647.68	19,716,018.32	17,488,870.77	8,346,431.17	7,440,328.31	5,882,477.87	1,966,484.72	2,098,391.71
Personal Accessories	132,249,058.98	116,715,219.51	1,540,675,699.15	49,825,913.97	46,207,416.17	42,651,086.54	17,962,985.46	6,419,357.03

Chapter 6: Charts

You can use IBM® Cognos® Report Studio to create many chart types, such as column, bar, area, and line charts. For information, see "[Chart Types](#)" (p. 130).

Current Default Charts and Legacy Charts

IBM® Cognos® Report Studio has a new default chart technology as of version 10.1.0. You will find procedures in this chapter for using both the legacy charts and the current default charts. To use the legacy chart type, set the **Use legacy chart authoring** option. For information about converting a legacy chart to a new default chart, see "[Convert Charts From One Type to Another](#)" (p. 145).

Creating Charts

Before creating charts, review the available chart types to select the best chart for your needs. Also review the chart objects that make up charts.

Tip: To view the available chart types, from the File menu, click New, and then double-click the chart icon . You can also view the available chart types by adding a **Chart** object from the **Toolbox** tab in the **Insertable Objects** pane to an existing report.

To create charts, drag data items from the **Source** tab of the **Insertable Objects** pane to the measures, data series and categories drop zones.

To help you when creating charts, you can do the following:

- Preview your chart style changes.

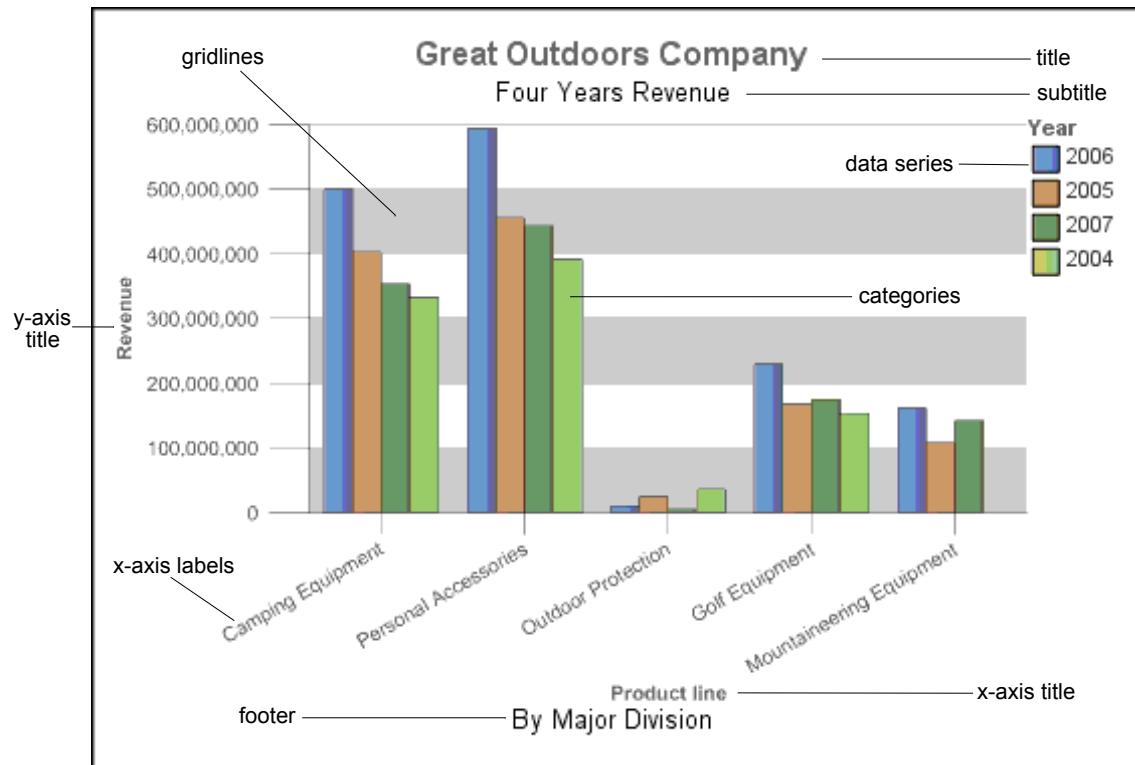
As you author your chart and edit the chart properties, IBM® Cognos® Report Studio provides you with a preview of your chart, using simulated data. This allows you to view your style changes without running your report. You need to run your report to see the chart with your actual data. You can disable the chart preview in the options.

- Resize your charts.
- Move the chart drop zones into flyouts so that they appear only when you pause your pointer over the chart.

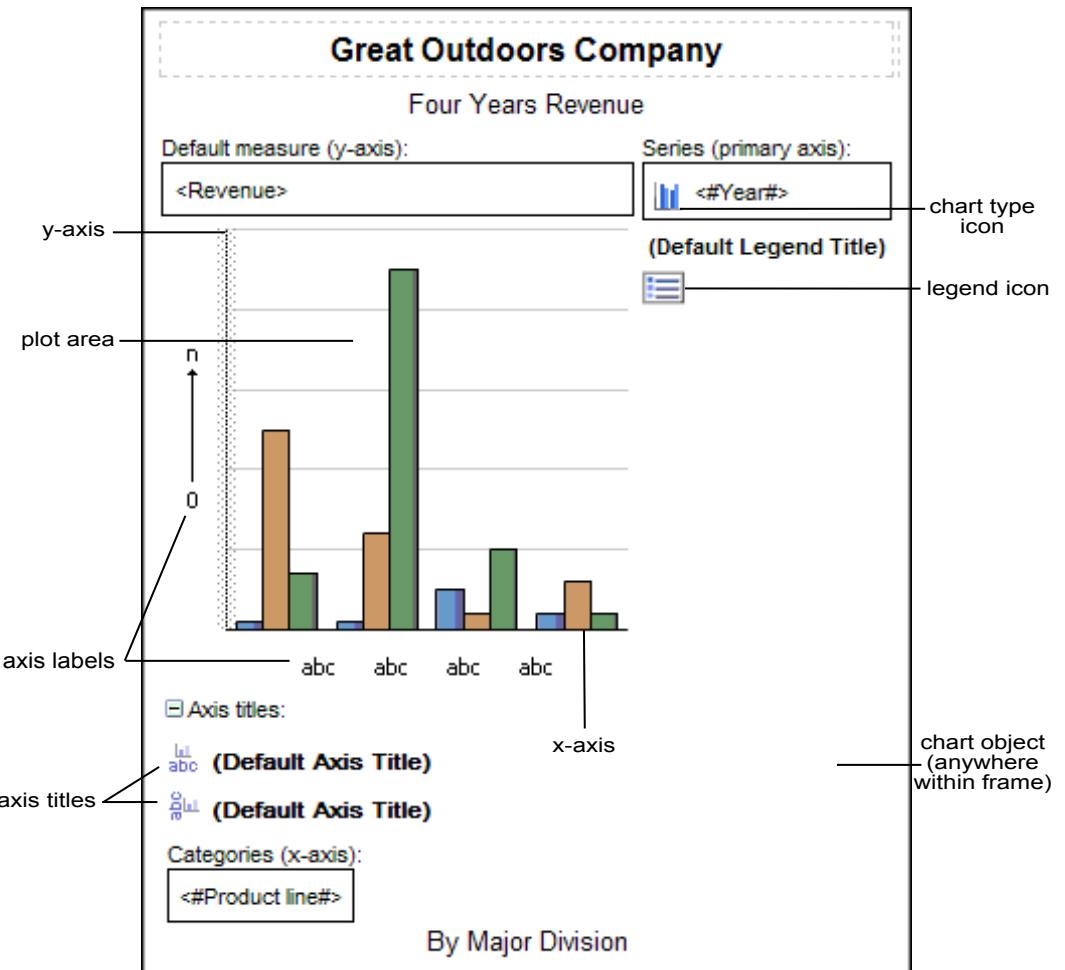
After you create a basic chart, modify the various chart objects to customize your chart.

Chart Objects

The following shows the most common chart objects as they appear in your report output in IBM® Cognos® Viewer.



The following shows the same chart as it appears in the IBM Cognos Report Studio user interface. The Y-axis is selected, and the axis titles are displayed.



Data Series

A data series is a group of related data points that are plotted in a chart. Each series has a unique color or pattern and is described in the legend. You can plot one or more data series in a chart; pie charts have only one data series.

In the example chart, the data series are order years 2004, 2005, 2006, and 2007.

Categories

Categories are groups of related data from the data series that are plotted on the X-axis. Categories of multiple data series are shown together using clustered and stacked data markers.

In the example chart, the categories are the product lines of The Great Outdoors Company in clustered columns.

Axes

Axes are lines that provide references for measurement or comparison.

The primary axis (or Y-axis) refers to measures of quantitative data, such as sales figures or quantities. Charts can have more than one primary axis.

The category axis (X-axis or ordinal axis) plots qualitative data, such as products or regions. It runs horizontally, except in bar charts.

The z-axis is the vertical axis in a 3-D chart.

Major gridlines extend from the tick marks on an axis and run behind the data markers.

Legend

A legend is a key to the patterns or colors assigned to the data series or categories in a chart.

Columns, Lines, and Areas

Charts use graphical elements such as columns, horizontal bars, points, bubbles, lines, and areas as visual representations of data points.

Chart Types

IBM® Cognos® Report Studio provides many types of charts for presenting your data in a way that is meaningful for your users.

You can select from a variety of chart types (such as pie, bar, line, gauge, scatter, and so on) and you can select from a variety of chart configurations (such as stacked columns, 3-D pies). Combination charts allow you to use more than one chart type within your chart.

Some chart types are not supported for Microsoft® Excel output or appear differently in Excel. For more information, see "[Limitations When Producing Reports in Microsoft Excel Format](#)" (p. 593).

Choosing a Chart Type and Configuration

To choose a chart type, consider what you want the chart to illustrate. Different chart types and configurations emphasize different things.

Purpose	Chart type or configuration
Show contributions of parts to a whole	Bar Charts Pie Charts Stacked Charts , when you want to display measures of the whole, as well as the parts 100 Percent Stacked Charts
Show trends in time or contrast values across different categories	Line Charts Area Charts Bar Charts Column Charts Always place time in the horizontal axis.
Compare groups of related information against actual values	Bar Charts Radar Charts

Purpose	Chart type or configuration
Compare different kinds of quantitative information	Combination Charts
Rank values in descending or ascending order	Bar Charts
	Column Charts
Show correlation between two sets of measures	Point Charts
Show key performance indicators in an executive dashboard	Gauge Charts
	Bullet Charts

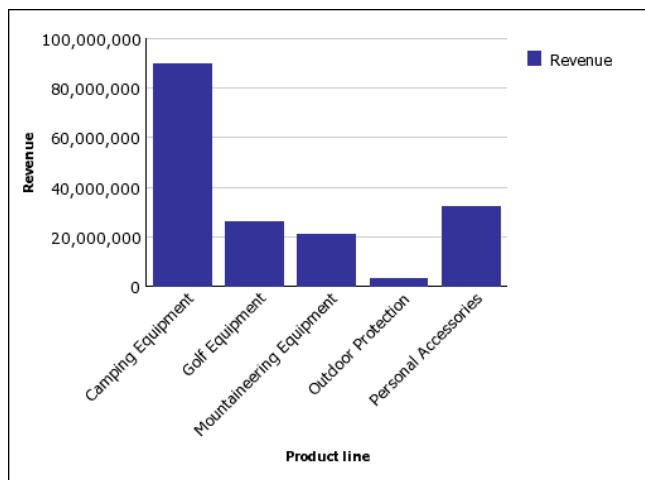
Column Charts

Column charts are useful for comparing discrete data or showing trends over time.

Column charts use vertical data markers to compare individual values.

The Sales Growth Year Over Year sample report in the GO Data Warehouse (analysis) package includes a column chart. For more information about The Great Outdoors Company samples, see ["Sample Reports and Packages" \(p. 567\)](#).

The following example shows the revenue for each product line.



Column charts can plot data using [standard](#), [stacked](#), [100 percent stacked](#), and [three-dimensional](#) configurations.

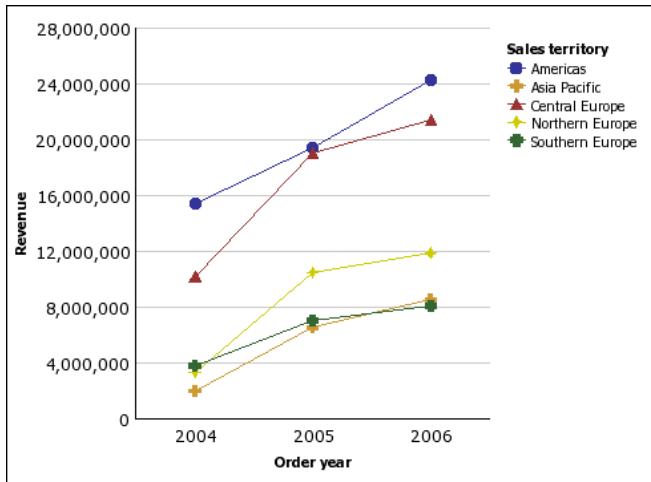
Line Charts

Line charts are useful for showing trends over time and comparing many data series.

Line charts plot data at regular points connected by lines.

Line charts can plot data using [standard](#), [stacked](#), [100 percent stacked](#), and [three-dimensional](#) configurations. It is best not to use stacked line charts because they are difficult to distinguish from unstacked line charts with multiple data series.

The following example shows a rising revenue trend in every territory.



The Top Retailers by Country sample report in the Sales and Marketing (cube) package includes a line chart. For more information about The Great Outdoors Company samples, see "["Sample Reports and Packages" \(p. 567\)](#)".

Pie Charts

Pie charts are useful for highlighting proportions.

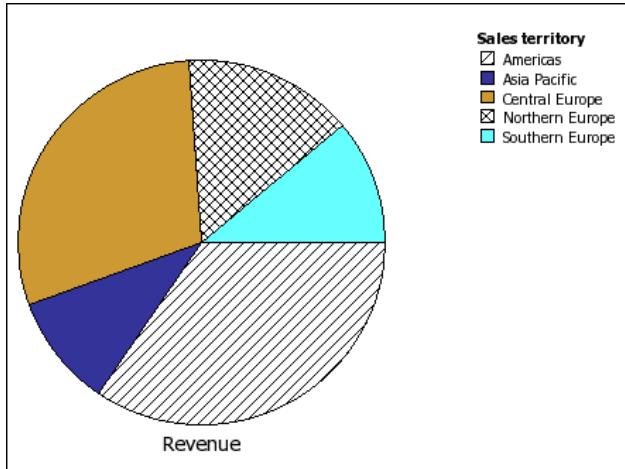
They use segments of a circle to show the relationship of parts to the whole. To highlight actual values, use another chart type, such as a stacked chart.

Pie charts plot a single data series. If you need to plot multiple data series, use a 100 percent stacked chart.

Reports in PDF or HTML format show a maximum of 16 pies or gauges per chart. If you need to see more, run the report in Excel Single Sheet format and they all appear in the report.

The Returns by Failed Orders in 2006 sample report in the GO Data Warehouse (analysis) package includes a pie chart. For more information about The Great Outdoors Company samples, see "["Sample Reports and Packages" \(p. 567\)](#)".

The following example shows that the largest proportion of revenue comes from the Americas, followed closely by the Asia Pacific region.



Pie charts can plot data using [standard](#), [100 percent](#), and [three-dimensional](#) configurations.

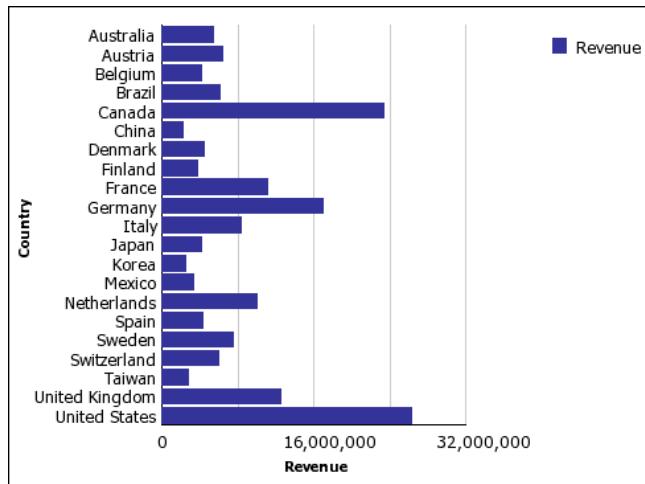
Bar Charts

Bar charts are useful for showing trends over time and plotting many data series.

Bar charts use horizontal data markers to compare individual values.

The Manager Profile sample report in the GO Data Warehouse (analysis) package includes a bar chart. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

The following example shows revenue for every country.



Bar charts can plot data using [standard](#), [stacked](#), and [100 percent stacked](#) configurations.

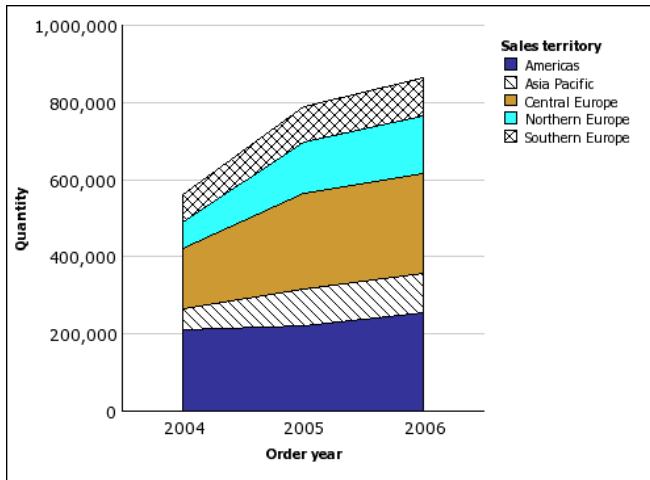
Area Charts

Area charts are useful for emphasizing the magnitude of change over time. Stacked area charts are also used to show the relationship of parts to the whole.

Area charts are like line charts, but the areas below the lines are filled with colors or patterns.

Do not use standard area charts to show multiple data series because it is possible for areas with lower values to be covered by others. For multiple data series, use a stacked area chart.

The following example is a stacked area chart showing the quantity of products sold over a two-year period in multiple territories.



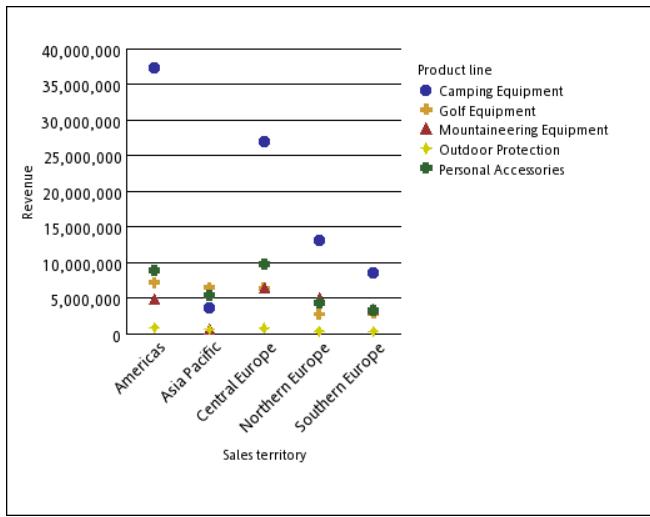
Area charts can plot data using [standard](#), [stacked](#), [100 percent stacked](#), and [three-dimensional](#) configurations.

Point Charts

Point charts are useful for showing quantitative data in an uncluttered fashion.

Point charts use multiple points to plot data along an ordinal axis. A point chart is the same as a line chart without the lines. Only the data points are shown.

The following example shows the revenue for each product line.

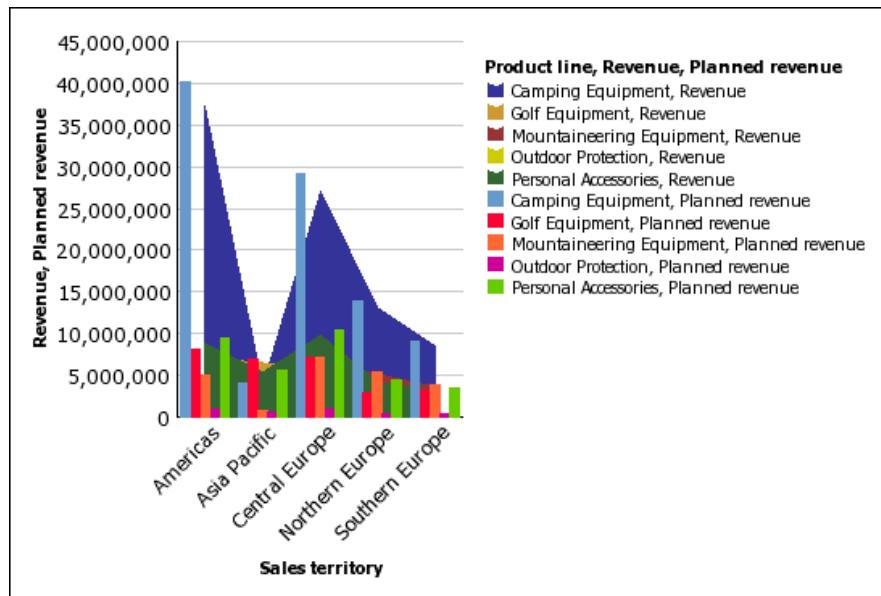


Combination Charts

Combination charts plot multiple data series by using combinations of columns, areas, and lines within one chart. They are useful for highlighting relationships between the various data series.

The Quantity Sold vs. Shipped and Inventory sample report in the GO Data Warehouse (analysis) package includes a combination chart. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

The following example shows a combination chart that includes planned revenue as a column chart and actual revenue as an area chart.



Combination charts can plot data using [standard](#), [stacked](#), [100 percent stacked](#), and [three-dimensional](#) configurations.

Scatter Charts

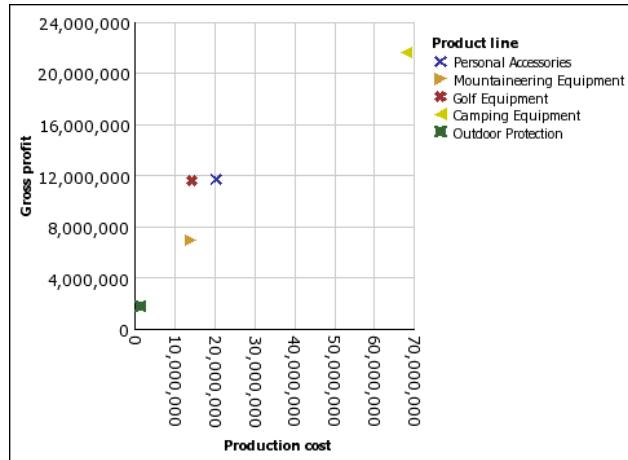
Scatter charts use data points to plot two measures anywhere along a scale, not only at regular tick marks.

You can also specify a default measure. For example, you might need to specify a default measure to give context to a calculated measure in the chart. For more information about the default measure, see "[Specify the Default Measure](#)" (p. 122).

Scatter charts are useful for exploring correlations between different sets of data.

The Employee Satisfaction 2006 sample report in the GO Data Warehouse (analysis) package includes a scatter chart. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

The following example shows the correlation between production cost and gross profit for each product line.



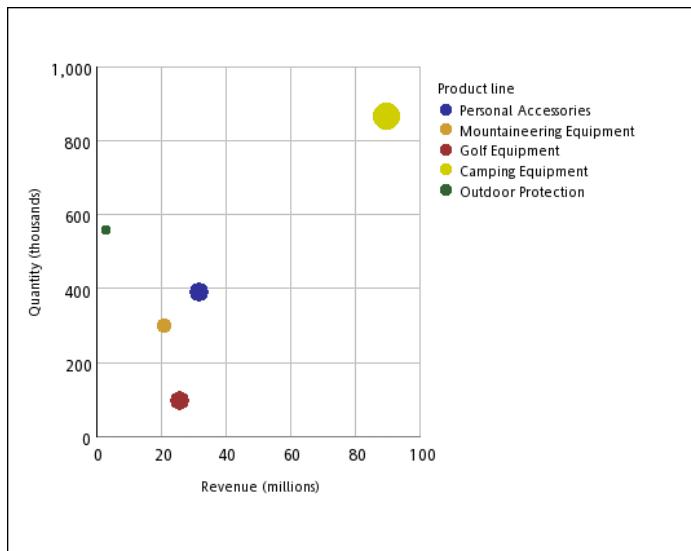
Bubble Charts

Bubble charts, like scatter charts, use data points and bubbles to plot measures anywhere along a scale. The size of the bubble represents a third measure.

You can also specify a default measure. For example, you might need to specify a default measure to give context to a calculated measure in the chart. For more information about the default measure, see "[Specify the Default Measure](#)" (p. 122).

Bubble charts are useful for representing financial data. These charts are not supported for Microsoft Excel output.

The following example plots quantity and revenue by product line. The size of the bubble represents the amount of gross profit.



Quadrant Charts

Quadrant charts are bubble charts with a background that is divided into four equal sections.

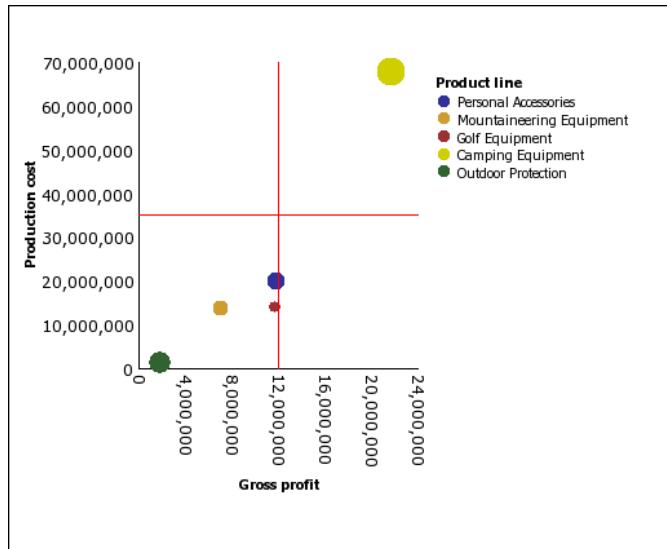
Quadrant charts are useful for plotting data that contains three measures using an X-axis, a Y-axis, and a bubble size that represents the value of the third measure.

You can also specify a default measure. For example, you might need to specify a default measure to give context to a calculated measure in the chart. For more information about the default measure, see "[Specify the Default Measure](#)" (p. 122).

Legacy quadrant charts use baselines to create the quadrants. Current default charts use colored regions.

Use a quadrant chart to present data that can be categorized into quadrants, such as a SWOT (strengths, weaknesses, opportunities, and threats) analysis.

The following example shows the relationship between production cost and gross profit. The size of the bubble represents the quantity of units sold.



Bullet Charts

Bullet charts are a variation of bar charts. They compare a featured measure (the bullet) to a targeted measure (the target). They also relate the compared measures against colored regions in the background that provide additional qualitative measurements, such as good, satisfactory, and poor.

Bullet charts are often used instead of gauge charts in executive dashboards. Bullet charts can be horizontal or vertical.

Note: This chart type applies only to the current default charts, and does not apply to the legacy charts.

A bullet chart contains the following components:

- A bullet measure.

The bullet measure, Revenue, appears as the blue bar in the chart below.

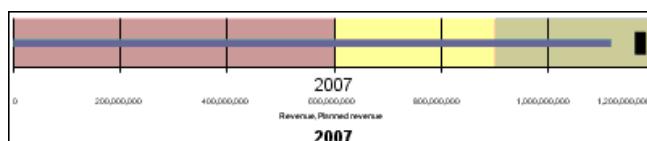
- A target measure.

The target measure, Planned revenue, appears as the black indicator in the chart below.

- From zero to five colored regions along the numeric scale to provide information about the featured measures' qualitative state.

The chart below includes three colored regions: 0-50%, 50-75%, and 75-100%.

- A label that identifies the measures.
- A numeric scale.



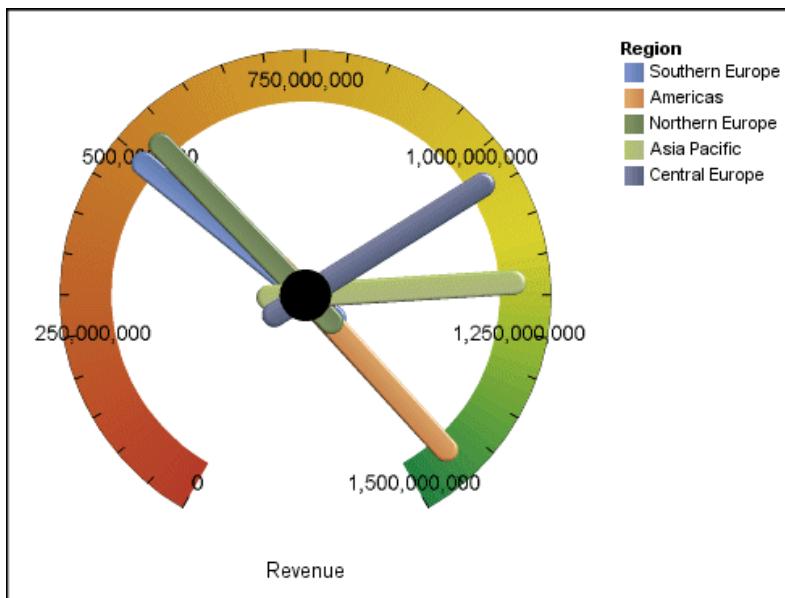
Gauge Charts

Gauge charts, also known as dial charts or speedometer charts, use needles to show information as a reading on a dial. The value for each needle is easily read against the colored data range or chart axis. This chart type is often used in executive dashboard reports to show key business indicators.

Gauge charts are useful for comparing values between a small number of variables either by using multiple needles on the same gauge or by using multiple gauges.

Reports in PDF or HTML format are limited to show a maximum of 16 pies or gauges per chart. These charts are not supported for Microsoft Excel output.

A gauge chart consists of a gauge axis (which contains the data range, color ranges, and intervals markers), needles, and a center pivot point. The following example shows a basic gauge chart with default attributes. It is a degree dial chart with two axes.



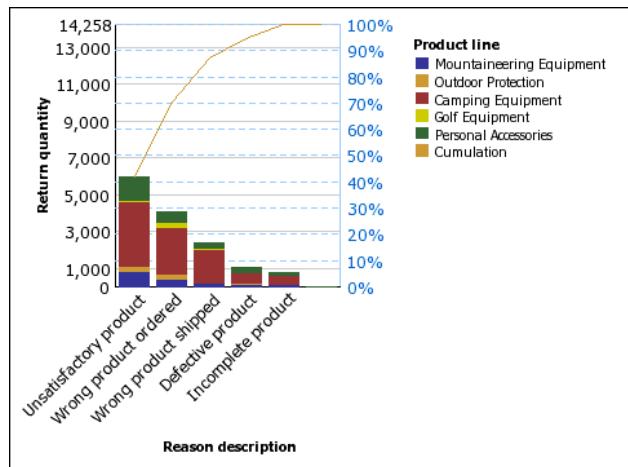
Pareto Charts

Pareto charts help you to improve processes by identifying the primary causes of an event. They rank categories from the most frequent to the least frequent. These charts are frequently used for quality control data, so that you can identify and reduce the primary cause of problems.

Pareto charts include a cumulation line, which shows the percentage of the accumulated total of all the columns or bars.

You can create before and after comparisons of Pareto charts to show the impact of corrective actions. These charts are not supported for Microsoft Excel output.

The following example shows that the most frequent reason for product returns is unsatisfactory product.



You can also create Pareto charts using horizontal bars.

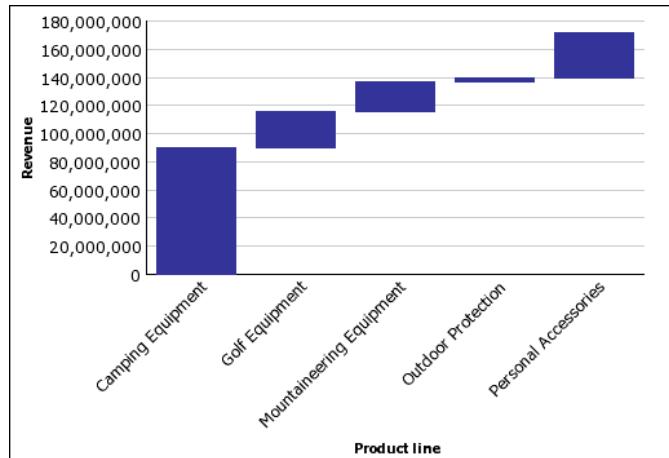
Progressive Column Charts

Progressive column charts, also known as waterfall charts, are like stacked charts with each segment of a single stack displaced vertically from the next segment.

Progressive column charts are useful for emphasizing the contribution of the individual segments to the whole.

These charts are not supported for Microsoft Excel output.

The following example analyzes the contribution of each product line to revenue.



Progressive column charts can plot data using [standard](#) and [three-dimensional](#) configurations. You can also create progressive charts using horizontal bars.

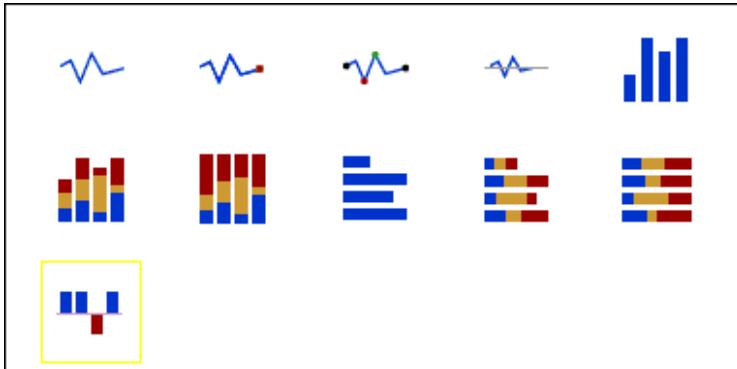
Microcharts

Microcharts are smaller versions of column charts, bar charts, and line charts that you can use in crosstabs and dashboards. Microcharts include column microcharts and bar microcharts, which are available in stacked and 100 percent stacked configurations, and win-loss charts and modified line charts that contain the following:

- A closing value marker.

- Opening, closing, high, and low value markers.
- A reference line.

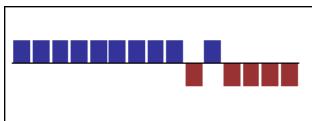
If you specify custom labels for a microchart, the custom labels appear in the tooltips, not on the chart itself. For more information about custom labels, see "[Specify Custom Label Text](#)" (p. 161).



Win-loss charts are microcharts in which the value of each column is either 1 or -1, often denoting a win or loss.

Win-loss charts use two measures (the default and the win-loss measure) and no series. The win-loss measure is the measure or calculation that you define. You can also specify a default measure. For example, you might need to specify a default measure to give context to a calculated measure in the chart. For more information about the default measure, see "[Specify the Default Measure](#)" (p. 122).

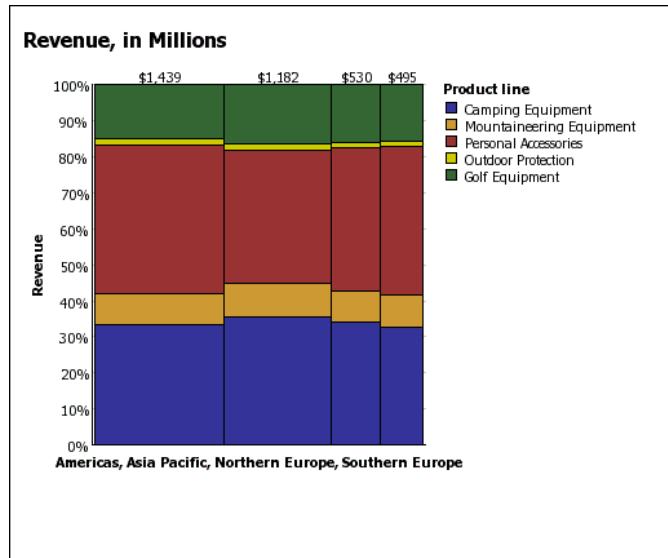
The following example shows the quarters that have a margin of less than 10,000 in red.



Marimekko Charts

Marimekko charts are 100 percent stacked charts in which the width of a column is proportional to the total of the column's values. Individual segment height is a percentage of the respective column total value.

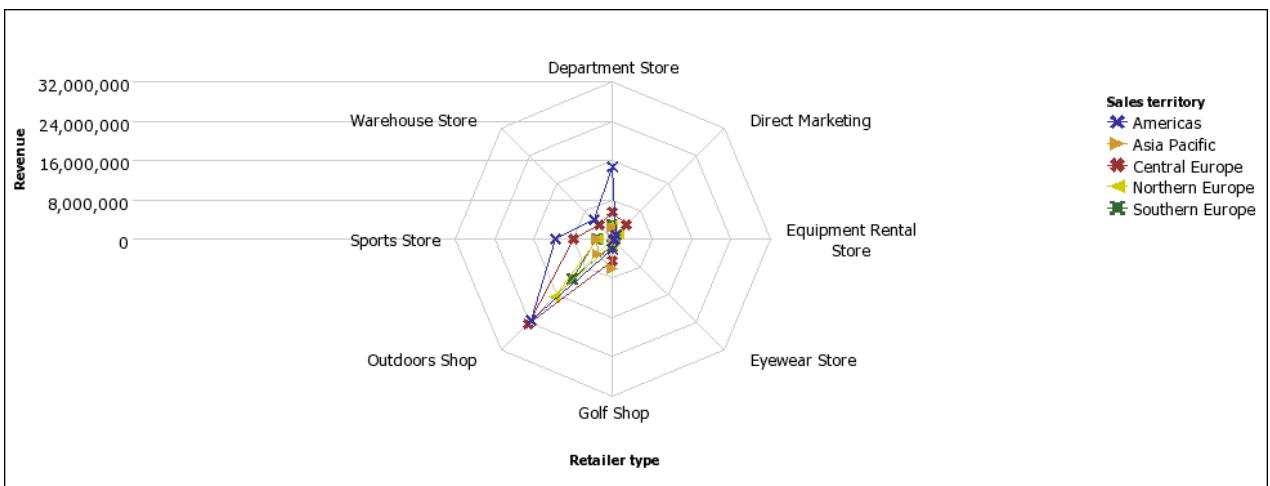
The following example shows the contribution of revenues for product lines in different regions.



Radar Charts

Radar charts integrate multiple axes into a single radial figure. For each figure, data is plotted along a separate axis that starts at the center of the chart.

The following example shows the revenue from multiple retailer types in multiple territories.



Radar charts can plot data using [standard](#) and [stacked](#) configurations.

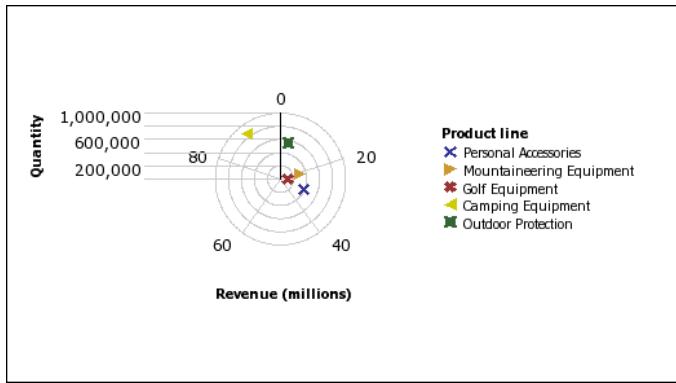
Polar Charts

Polar charts are useful for showing scientific data.

They are circular charts that use values and angles to show information as polar coordinates.

You can specify a default measure. For example, you might need to specify a default measure to give context to a calculated measure in the chart. For more information about the default measure, see "[Specify the Default Measure](#)" (p. 122).

The following example shows the revenue and quantity for each product line. The distance along the radial axis represents quantity, and the angle around the polar axis represents revenue.

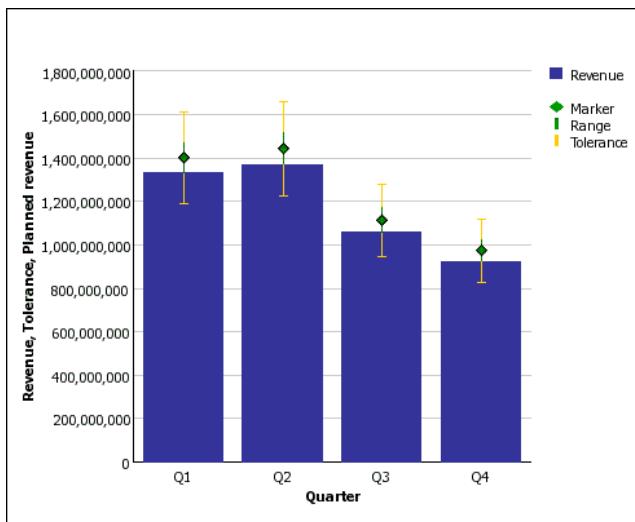


Range Indicator Charts

Range indicator, or metrics range, charts are useful for showing a target range and a tolerance range.

A range indicator chart adds a target and range marker to a column, line, or area chart.

The following example shows actual revenue versus planned revenue. The markers indicate the planned revenue, and the range and tolerance lines are defined by the chart author.



You can also insert IBM Cognos Metric Studio diagrams in your report to show the relationship between metrics. For more information, see "["Add a Static Metric Studio Diagram to a Report" \(p. 200\)](#)" and "["Add a Data-driven Metric Studio Diagram to a Report" \(p. 201\)](#)".

Chart Configurations

Chart configurations specify the grouping type of the columns, bars, lines, and areas in a chart. Some examples are standard, stacked, and 100 percent stacked charts.

Standard Charts

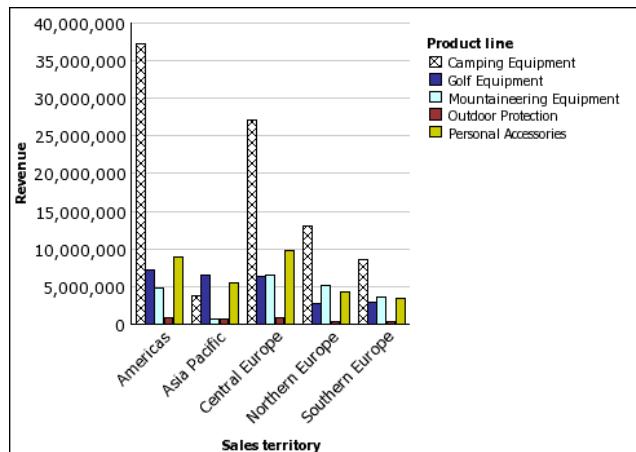
Standard or absolute charts are useful for comparing specific values and for representing discrete data, such as data for different regions or individual employees. For example, a standard column chart that plots regional sales emphasizes the actual value that each region achieves in sales.

Standard charts plot the actual value of each data series from a common axis.

When you create charts using multiple data series, you can distinguish each series by the color or pattern of its data marker. Related data series are shown together in clusters for easy comparison.

In standard area and radar charts that have multiple data series, the colored areas that represent lower values might be covered by the larger colored areas that represent higher values. Use the stacked configuration for area and radar charts with multiple data series.

The following example shows the revenue values for each product line within each territory.



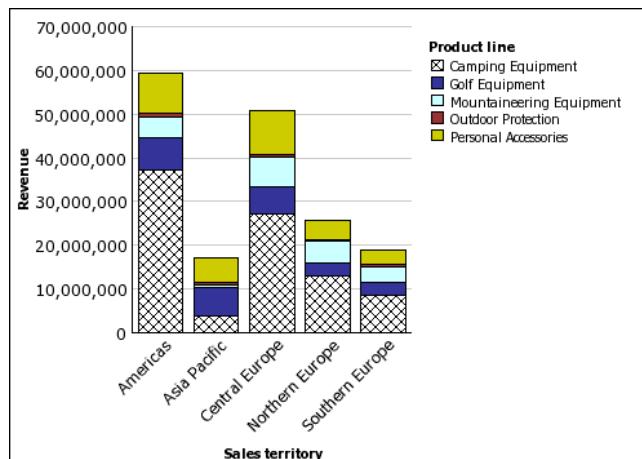
Stacked Charts

Stacked charts are useful for comparing proportional contributions within a category. They plot the relative value that each data series contributes to the total. For example, a stacked column chart that plots product line sales will emphasize the proportion that each product line contributes to the total in each territory.

You can distinguish each data series by the color or pattern of its section in the stack. The top of each stack represents the accumulated totals for each category.

Do not use the stacked configuration in line charts that have multiple data series because it is difficult to distinguish between unstacked and stacked configurations, and your chart consumers might misunderstand your data.

The following example shows that camping equipment contributed a large proportion of the actual revenue in most sales territories.



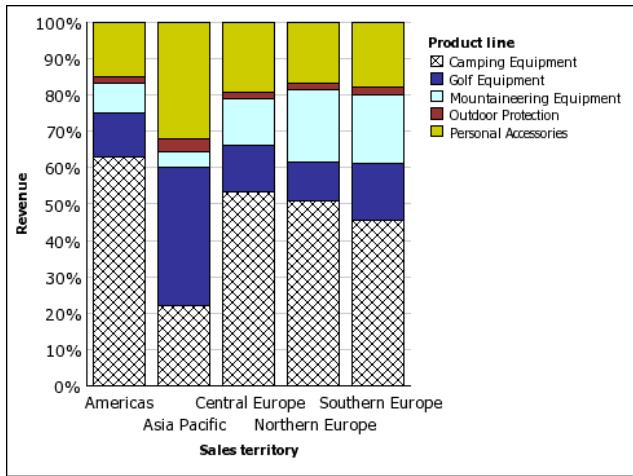
100 Percent Stacked Charts

100 percent stacked charts are useful for comparing proportional contributions across all categories. They plot the relative contribution of each data series to the total as a percentage. For example, a 100 percent stacked column chart that plots product line sales emphasizes the percentage within each region without referring to actual values.

You can distinguish each data series by the color or pattern of its section in the stack. Each stack represents 100 percent.

100 percent stacked charts highlight proportions. When actual values are important, use another chart configuration.

The following example shows the percentage of sales for each product line in each region.



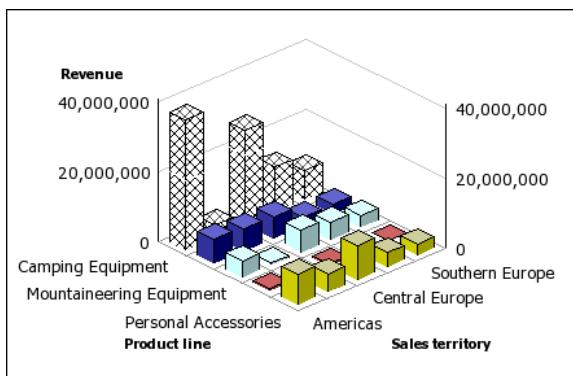
Three-dimensional Charts

Three-dimensional charts provide a visually effective display that is suitable for presentations.

Three-dimensional column, bar, line, and area charts plot data by using three axes.

Three-dimensional pie charts have a three-dimensional visual effect.

Do not use three-dimensional charts when you need to show exact values, such as for control or monitoring purposes. The distortion in three-dimensional charts can make them difficult to read accurately. For example, the following chart shows actual revenue for each product line in each territory, but some data labels are omitted because there is not enough room to display them all.



Convert Charts From One Type to Another

You can convert a chart from one type (for example, a bar chart) to another type (for example, a line chart).

When you convert an existing chart to a new chart type, IBM® Cognos® Report Studio keeps the properties from the existing chart if those properties exist in the new chart type. For example, if you convert a pie chart to a bar chart, Report Studio maps your chart palette to the new chart, but does not map the exploding slices, because the exploding slices property does not exist in a bar chart.

Note: Report Studio converts legacy charts to the current default chart types automatically if you change the chart type unless you select the **Use legacy chart authoring** option. You can convert legacy charts to current default charts, but you cannot convert current default charts back to legacy charts.

Steps

1. Right-click a chart and click **Convert Chart**.
2. Select a new chart type, and click **OK**.

If some chart properties cannot be converted, the **Chart Property Changes** dialog box appears notifying you of the properties that cannot be converted.

Convert Legacy Charts to Current Default Charts

You can convert charts from the legacy charts to the current default charts. When you convert a legacy chart, select the template that most closely matches your legacy chart in order to preserve as many settings as possible.

Note: You can convert legacy charts to current default charts, but you cannot convert current default charts back to legacy charts.

Steps

1. From the **Tools** menu, click **Options**.
2. From the **Advanced** tab, clear the **Use legacy chart authoring** check box.
3. Right-click the legacy chart and click **Convert Chart**.
4. Select a new chart type, and click **OK**.

If some chart properties cannot be converted, the **Chart Property Changes** dialog box appears notifying you of the properties that cannot be converted.

Limitations When Converting Legacy Charts

When you convert a legacy chart to the current default chart, some chart types or chart properties might not be migrated properly. For example, clustered area charts and 100% stacked line charts are not available in the current default charts. If your exact chart configuration is not available, select the closest matching template.

The following legacy chart properties are not migrated to the current default charts:

- **Conditional Palette**
- **Style Variable**
- **Display Frequency and First Label Index** on the category axis
- **Visual Angle**
- **Rotate Values**
- **Value Location**
- **Conditional Style, Style Variable, and Drill-Through Definitions** on pie labels and gauge labels
- **Connecting Lines** on combination charts
- line styles; only four line styles are supported in the current default charts

The following table outlines some behavior differences between legacy charts and the current default charts.

Legacy Charts	Current Default Charts
Combination charts are always ordered from back to front - area, bar, and line.	Combination charts support any order.
In bar charts, multiple bar definitions for the Y1-axis are placed side-by-side.	In bar charts, multiple bars are placed on top of one another and you can control the width of the bars.
In combination charts, the color palette is continued over multiple bars, lines, and areas.	In combination charts, color palettes are restarted for each bar, line, and area.
The legend order respects the order in which bars, lines, and areas are specified. For example, bars can appear before areas in the legend.	The legend order is the same as the order specified in the chart.

The following chart types continue to use the legacy chart technology and therefore cannot be upgraded to the current default chart technology. If you want to convert these chart types to the current default charts, upgrade them to a different chart type, such as a column or line chart.

- Win loss
- Radar
- Polar
- 3-D Bar, 3-D Line, 3-D Area, and 3-D Combination
- 3-D Scatter

- Metrics range
- Marimekko

Customizing Chart Properties

After you create a chart, you can customize it by changing its properties.

You select the chart object in IBM® Cognos® Report Studio to view its properties. Some properties are dependent on the existence of other properties.

If you are familiar with using conditional variables, you can customize the chart to change its appearance or provide information in response to expressions or conditions.

As you edit the chart properties, Report Studio provides you with a preview of your chart, using simulated data. This allows you to view your style changes without running the report. You can disable the chart preview in the options.

The following table shows some of the properties you can change in charts. These properties are available when you select the chart object unless specified otherwise in the **Action to perform** in the **Properties pane** column.

Goal	Action to perform in the Properties pane
Hide or show the title, subtitle, or footer	Select the chart. Under Chart Titles , set the Title , Subtitle , or Footer property.
Hide or show the legend	Select the chart. Under Chart Annotations , set the Legend property.
	For more information, see " Customize the Legend in a Current Default Chart " (p. 166).
Hide or show baselines	Select the chart. Under Chart Annotations , set the Baselines property.
	For more information, see " Add a Baseline to a Current Default Chart " (p. 172).
Hide or show trendlines or regression lines	Select the chart. Under Chart Annotations , set the Trendlines or Regression Line property.
	For more information, see " Display Trendlines in Current Default Charts " (p. 176) or " Display Regression Lines in Legacy Scatter or Bubble Charts " (p. 177).
Hide or show markers	Select the chart. Under Chart Annotations , set the Markers property.
	For more information, see " Add a Marker to a Current Default Chart " (p. 174).

Goal	Action to perform in the Properties pane
Hide or show notes	<p>Select the chart. Under Chart Annotations, set the Notes property.</p> <p>For more information, see "Add a Note to a Current Default Chart" (p. 169).</p>
Hide or show the axes	<p>Select the chart. Under Axes, set the Y1 Axis, Y2 Axis, or Category Axis property.</p> <p>For more information, see "Customize the Axes of a Current Default Chart" (p. 162).</p>
Hide or show the axis title or axis line	<p>Select the axis. Under General, set the Axis Line or Axis Title property.</p> <p>For more information, see "Customize the Axes of a Current Default Chart" (p. 162).</p>
Change an axis title	<p>Select the axis title. Under General, set the Default Title property to No, and then double-click the axis title and type a new title.</p> <p>For more information, see "Customize the Axes of a Current Default Chart" (p. 162).</p>
Change axis properties, such as range, scale interval, and so on	<p>Select the numeric axis. Under General, set the Minimum Value, Maximum Value, Scale Interval, or Scale property.</p> <p>For more information, see "Change the Axis Scale of a Current Default Chart" (p. 163).</p>
Hide or show the tooltips	<p>Select the chart. Under Chart Labels, set the Tooltips property.</p> <p>When you pause your pointer over a data marker in the report output, the corresponding absolute or cumulative value appears in a tooltip.</p> <p>Tooltips are not supported in PDF output.</p>
Change the data format	<p>Select the numeric axis. Under Data, set the Data Format property.</p>
Change the white space around the chart	<p>Select the chart. Under Box, set the Padding or Margin property.</p> <p>For more information, see "Apply Padding to an Object" (p. 450) and "Set Margins for an Object" (p. 450).</p>

Goal	Action to perform in the Properties pane
Change the color or pattern in the palette for columns, lines, and areas	<p>Select the chart object. Under Color & Background, set the Palette or Conditional Palette property.</p> <p>For more information, see "Customizing the Color Palette of a Chart" (p. 151) and "Create a Simple Conditional Palette for a Current Default Chart" (p. 153).</p>
Change the chart orientation	<p>Select the chart. In the Properties pane, under Chart Orientation, select Vertical or Horizontal.</p>
Change the default color or font for all chart objects	<p>Select the chart. Under Color & Background, set the Background Color, Foreground Color, or Fill Effects property.</p> <p>For more information, see "Change a Chart Background in a Legacy Chart" (p. 157).</p>
	<p>Under Font & Text, set the Font or Relative Alignment property.</p>
	<p>Tip: The 2005 Quarterly Sales Forecast sample report in the GO Sales (analysis) package includes text alignment. For more information about The Great Outdoors Company samples, see "Sample Reports and Packages" (p. 567).</p>
Override the default font or color for axes and chart values	<p>Click the unlock button  in the toolbar to unlock the chart object, select the chart body by clicking between the axes, and set the Font, Background Color, Foreground Color, or Fill Effects property.</p>
Change the three-dimensional appearance of a chart	<p>Select the chart. Under General, set the Depth or Visual Angle property.</p> <p>For example, for a pie chart, a Depth value of 0 (zero) leaves the pie flat, or two-dimensional. A value of 100 raises the pie almost to a round ball shape.</p>
	<p>Tip: You can select a value from the Depth list or type a value that is not listed.</p>
Insert a background image or watermark in the chart body.	<p>Click the lock button in the toolbar to unlock the chart object, select the chart body by clicking between the axes.</p> <p>Under Color & Background, set the Background Image property.</p> <p>For more information, see "Change a Chart Background in a Legacy Chart" (p. 157).</p>

Goal	Action to perform in the Properties pane
Go to another report	<p>Select the chart. Under Data, set the Drill-Through Definitions property.</p> <p>For more information, see "Using Drill-through Access" (p. 515).</p>
Specify which labels and values to use when generating the text shown on the chart	<p>Select the chart. Under Chart Labels, set the Values property.</p> <p>Note: When you show all the labels and values on some chart types, such as scatter charts, bubble charts, and polar charts, the text shown might be too long.</p>

Resize a Current Default Chart

You can change the size of a current default chart by dragging the resize handle in the lower right corner of the chart.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the [chart object](#).
2. Click the resize handle in the lower right corner of the chart . Use Shift+drag to maintain the aspect ratio of the chart.
3. To see the actual size of the chart as it will appear in the report output, collapse the chart objects with the minus (-) button in the upper left corner of the chart.

The new chart size appears in the **Size & Overflow** property. You can also edit the chart size with this property.

Resize a Legacy Chart

You can resize a legacy chart using the **Size & Overflow** property. The options available with this property are detailed in "[Specify the Height and Width of an Object](#)" (p. 456).

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the [chart object](#).
2. In the **Properties** pane, double-click the **Size & Overflow** property and specify the height and width.

Customizing the Color Palette of a Chart

You can use the chart palette to control the colors or patterns used in the columns, lines, data markers or areas in a chart. For example, if "Telephone" is the first in a data series of order methods, and you want it to appear in blue, use the palette to make the first item in the series blue.

You can customize the color palette for area, bar, bubble, gauge, Pareto, pie, and scatter charts.

You can also apply background effects to chart objects or change the colors of specific chart elements.

Customize the Color Palette of a Current Default Chart

Progressive charts and legacy charts have different steps. For more information about current default charts and legacy charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the [chart object](#).
2. To choose a preset color palette, click the chart palette presets button  on the style toolbar, and then click a palette.
3. To change the existing color palette:
 - In the Properties pane, under **Color & Background**, double-click the **Palette** property.
 - On the **Entries** tab, click the palette entry that you want to change.
 - Under **Fill**, select fill type, color, transparency, and other fill settings. The options in the **Fill type** list depend on the chart type you are using.
 - Linear Gradient** displays a gradient that changes color horizontally. You can specify the position of each color in the gradient and add or remove colors from the gradient.
 - Radial Line Gradient** displays a gradient that changes color outwards from a central point, or focus. You can specify the position of each color in the gradient and the size and position of the focus. You can also add or remove colors from the gradient.
 - Radial Rectangle Gradient** displays a gradient that changes color outwards from a rectangle. You can specify the position of each color in the gradient and the size of the rectangle. You can also add or remove colors from the gradient.
 - Pattern** displays a colored pattern that you choose from a preset list. You can specify the foreground and background colors.
 - Color** displays a solid color. You can specify the transparency.
 - Image** displays an image. You can specify the image to use.
- To change the color of this palette definition, click the color in the **Colors** list, click **Color**, specify the color properties, and click **OK**.
4. Under **Style**, depending on the chart type you are customizing and the fill type you chose in step 2, you can change the marker shape, line style, default color, and line weight.
5. To add a new palette entry, click the new button  and specify the new palette settings.

6. To change the order in which the colors, gradients, or patterns appear in the chart, use the arrow buttons under the **Palette** box to change their position.
7. To copy and paste a palette, open the palette dialog box and press Ctrl+C to copy the palette to the clipboard. Close the palette dialog box. Select another chart, open the palette dialog box, and press Ctrl+V to paste the palette from the clipboard.
8. To customize the data markers, on the **Properties** tab, depending on the chart type you are customizing and the fill type you chose in step 2, you can specify whether data markers are shown and change their color and size.

Customize the Color Palette of a Current Default Progressive Chart

Legacy charts and other types of current default charts have different steps. For more information about current default charts and legacy charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the progressive chart object.
2. In the **Properties** pane, double-click the **Progressive Palette** property.

You can specify the following:

- **Positive value fill** defines the appearance of the positive bars or columns on the chart.
- **Negative value fill** defines the appearance of the negative bars or columns on the chart.
- **First value fill** defines the appearance of the first bar or column on the chart.
- **Total value fill** defines the appearance of the total bar or column on the chart, if a total bar or column is present.

3. To change the appearance of a fill:

- Click the colored box beside the fill name.
- Click a fill type from the **Fill type** list.

Linear Gradient displays a gradient that changes color horizontally. You can specify the position of each color in the gradient and add or remove colors from the gradient.

Radial Line Gradient displays a gradient that changes color outwards from a central point, or focus. You can specify the position of each color in the gradient and the size and position of the focus. You can also add or remove colors from the gradient.

Radial Rectangle Gradient displays a gradient that changes color outwards from a rectangle. You can specify the position of each color in the gradient and the size of the rectangle. You can also add or remove colors from the gradient.

Pattern displays a colored pattern that you choose from a preset list. You can specify the foreground and background colors.

Color displays a solid color. You can specify the transparency.

Image displays an image. You can specify the image to use.

- To change the color of this fill, click the color in the **Colors** list, click **Color**, specify the color properties, and click **OK**.

- 4. To change the text of the label that appears next to a bar, double-click the text box beside the fill name and specify the text.

Customize the Color Palette of a Legacy Chart

Current default charts have different steps. For more information about current default charts and legacy charts, see "["Current Default Charts and Legacy Charts" \(p. 127\)](#)".

Steps

1. Select the [chart object](#).

2. In the **Color & Background** section of the **Properties** pane, double-click the **Palette** property.

3. In the **Palette** dialog box, click the new palette entry button to define a new palette entry:
 - To apply color, click **Color**, specify the color properties, and click **OK**.
 - To apply a gradient, click **Gradient**, specify the **Direction**, **From color**, and **To color** properties, and click **OK**.
 - To apply a pattern, click **Pattern**, specify the **Pattern**, **Foreground color**, and **Background color** properties, and click **OK**.

- Tip:** You can also select a predefined palette from the **Palettes** box.

4. To change the order in which the colors, gradients, or patterns appear in the chart, use the arrow buttons under the **Palette** box to change their position.

5. To copy and paste a palette, open the palette dialog box and press **Ctrl+C** to copy the palette to the clipboard. Close the palette dialog box. Select another chart, open the palette dialog box, and press **Ctrl+V** to paste the palette from the clipboard.

6. Repeat step 3 for each chart series.

Creating a Conditional Color Palette in a Chart

You can create a conditional palette to color data items in your chart in different ways depending on a condition. For example, in a column chart that shows revenue per month, you want to make the columns for the months that have a revenue greater than \$1000000 green.

You can also use conditional styles to highlight exceptional data and use conditions to control the layout of your report. For more information, see "["Using Conditions" \(p. 473\)](#)".

Create a Simple Conditional Palette for a Current Default Chart

You can create a simple conditional palette based on a string data item or a numeric data item. For example, if you want to create a condition to display only Revenues over one million dollars, you can create a numeric condition; whereas, if you want to create a condition to display only product codes with the letter A in them, you can create a string condition.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the chart series to which you want to apply the conditional palette.
 2. In the Properties pane, under Color & Background, double-click the Conditional Palette property.
 3. Click the new button  and click New Condition.
 4. If you want to create a conditional palette based on a string, do the following:
 - Select the string data item to determine the condition.
 - In the Type of conditional style list, click String, and click OK.
 - Click the new button and select how to define the condition:
 - To select more than one individual value, click Select Multiple Values and click the values.
 - To type specific values, click Enter Values and type the values.
 - To specify your own criteria, such as values that begin with the letter A, click Enter String Criteria and specify the condition.
 - Under Palette Entry, beside the new condition, click the edit button .
 - Define the palette entry fill type, colors, and angle, and click OK.
 5. If you want to create a conditional palette based on a numeric range, date/time range, interval range, date range, or time range, do the following:
 - Select a numeric data item to determine the condition.
 - In the Type of conditional style list, click the type of condition you want to create, and click OK.
 - Click the new button  and type a value to define a threshold.
The value appears in the Range column, and two ranges are created.
 - For each range, under Style, click the edit button  to define the palette entry fill type, colors, and angle, and then click OK.
 - Repeat the steps above to add other conditions.
- Tip:** Under Style, pause the pointer over each range  to see the condition produced for that range.
- If you want to use the default palette for the lowest range instead of specifying a conditional palette select the Use default palette for lowest range check box.

- To move a value above or below a threshold, click the arrow button  next to the value. For example, you insert a threshold value of five million. By default, the ranges are less than or equal to five million and greater than five million. Moving the five million value above the threshold changes the ranges to less than five million and greater than or equal to five million.
6. To customize the label for a conditional palette, do the following:
 - Click **Label** beside the palette entry.
 - In the **Conditional Palette Entry Label** dialog box, click **Specified text** and click the ellipsis (...) button.
 - Type the default label text in the **Label** text box.
 - To add customized labels for other languages, click the add button , select the other languages, and click the language to define the customized label for that language.
 7. Specify the order in which to evaluate the conditions by clicking a condition and then clicking the move up or move down arrow.
Conditions are evaluated from top to bottom, and the first condition that is met is applied.

Create an Advanced Conditional Palette for Current Default Charts

Advanced conditions allow you to use a calculations or expression to evaluate the condition. If multiple advanced conditions are met, only the first conditional style is applied.

For more information about legacy and current default charts, see "["Current Default Charts and Legacy Charts" \(p. 127\)](#)".

Steps

1. Select the chart series to which you want to apply the conditional palette.
2. In the **Properties** pane, under **Color & Background**, double-click the **Conditional Palette** property.
3. Click the add button  and click **Advanced Condition**.
4. Click the new advanced condition button  and specify the expression that defines the condition.
The new advanced condition appears in the **Conditional Palette - Advanced** dialog box.
5. Under **Palette Entry**, beside the new advanced condition, click the edit button .
6. Define the palette entry fill type, colors, and angle, and click **OK**.
7. Specify the order in which to evaluate the conditions by clicking a condition and then clicking the move up or move down arrow.
Conditions are evaluated from top to bottom, and the first condition that is met is applied.

Create a Conditional Color Palette for a Legacy Chart

You can create a color palette that appears differently depending on a yes or no answer, an expression or calculation, or the report language.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Click the [chart object](#).
2. In the Properties pane, under **Color & Background**, double-click the **Conditional Palette** property.
3. If you want to create a variable that has only two possible values, Yes and No, do the following:
 - Under **Variable**, click **New boolean variable**.
 - In the **New Variable** dialog box, type a name for the variable.
 - In the **Expression Definition** box, define the condition and click **OK**.

For example, the following expression returns the value Yes if revenue is less than one million and the value No if revenue is greater than or equal to one million:

`[Revenue]<1000000`

For information about creating expressions, see "[Using Relational Calculations](#)" (p. 342) or "[Using Dimensional Calculations](#)" (p. 396) and "[Calculation Components](#)" (p. 769).

- In the **Value** box, select whether the expression evaluates to Yes or No.
 - Under **Effects**, specify the color palette to use for the condition.
4. If you want to create a variable whose values are string-based, do the following:
 - Under **Variable**, select **New string variable**.
 - In the **Expression Definition** box, define the condition and click **OK**.

For example, the following expression returns the value **high** if revenue is greater than one million and the value **low** if revenue is less than or equal to one million:

`if ([Revenue]>1000000) then ('high') else ('low')`

For information about creating expressions, see "[Using Relational Calculations](#)" (p. 342) or "[Using Dimensional Calculations](#)" (p. 396) and "[Calculation Components](#)" (p. 769).

- Click the add button  in the **Values** pane.
- For each value that the variable can assume, type the name of the value that corresponds with the possible outcomes defined in the expression.

For example, in the previous expression, you must create two values for the variable, high and low.

Tip: You can create a group by clicking two or more values and then clicking the group values button . For example, you can create a group that includes the available French languages.

5. If you want to create a variable whose values are different languages, do the following:
 - Under **Variable**, select **Report Language Variable**.
 - In the **Languages** dialog box, select the languages to support.
6. Click **OK** to close the expression editor and leave the **Conditional Palette** dialog box open.
7. Under **Effect**, click **Pattern**.
8. Under **Pattern**, select a pattern and click **OK**.
9. In the **Box** section of the **Properties** pane, set the **Borders** property to **Show**.

Change a Chart Background in a Current Default Chart

You can use a gradient, pattern, color, or image to customize the chart background.

For more information about legacy and current default charts, see "["Current Default Charts and Legacy Charts" \(p. 127\)](#)".

Steps

1. Click the [chart object](#).
2. Click the unlock button , select the chart body by clicking between the axes, and, in the **Properties** pane, under **Color & Background**, double-click the **Plot Area Fill** property.
3. Click a fill type from the **Fill type** list.

Linear Gradient displays a gradient that changes color horizontally. You can specify the position of each color in the gradient and add or remove colors from the gradient.

Radial Line Gradient displays a gradient that changes color outwards from a central point, or focus. You can specify the position of each color in the gradient and the size and position of the focus. You can also add or remove colors from the gradient.

Radial Rectangle Gradient displays a gradient that changes color outwards from a rectangle. You can specify the position of each color in the gradient and the size of the rectangle. You can also add or remove colors from the gradient.

Pattern displays a colored pattern that you choose from a preset list. You can specify the foreground and background colors.

Color displays a solid color. You can specify the transparency.

Image displays an image. You can specify the image to use.

Change a Chart Background in a Legacy Chart

You can use a solid color, a pattern, or a gradient fill effect to customize the chart background.

You can also use an image as a background for a chart. For more information, see "[Insert a Background Image in an Object](#)" (p. 442).

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Click the [chart object](#).
2. Click the unlock button , select the chart body by clicking between the axes, and, in the **Properties** pane, under **Color & Background**, double-click the **Fill Effects** property.
3. In the **Effect** box, choose a color, a gradient fill effect, or a pattern:
 - To apply a color, click **Color** and specify the color properties.
 - To apply a gradient fill effect, click **Gradient** and then specify the **Direction**, **From color**, and **To color** properties.
 - To apply a pattern effect, click **Pattern** and specify the **Pattern**, **Foreground Color**, and **Background Color** properties.The foreground color is the color of the selected pattern. The background color is the color of the area behind the pattern.

Tip: To remove a background fill effect, click **None**.

Add Background Effects to a Chart Object in a Legacy Chart

You can change the look of certain charts and chart objects by applying visual effects such as drop shadows, borders, fills, texture effects, and bevel effects.

For information about adding color to chart objects, see "[Add Color to an Object](#)" (p. 461) and "[Customizing the Color Palette of a Chart](#)" (p. 151).

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the [chart object](#).
2. To apply a preset background fill effect, from the toolbar, click the background effects presets  button.
3. To add a custom fill effect and border do the following:
 - In the **Properties** pane, under **Color & Background**, double-click the **Background Effects** property.
 - Click **Border** and specify settings for border style, width, color, corner radius for rounded rectangles, and transparency.

If the object also includes a fill with a transparency setting, select the **Allow transparent bleed** check box to apply the same transparency to the border.

- Click **Fill** and specify the settings.
The fill effect can either be a solid color, a gradient, or a pattern.
4. To add a texture effect, in the **Properties** pane, under **Color & Background**, select an option under **Material Effects**.
 5. To add a drop shadow
 - In the **Properties** pane, under **Color & Background**, double-click the **Drop Shadow** property.
 - Select the **Drop shadow** check box.
 - From the **Shadow color** list, click a color.
 - Under **Blur**, click an intensity to set the drop shadow's blur.
 - Under **Offset**, enter a value and a unit to define the width and height of the drop shadow.
 - Under **Transparency**, type a percentage to determine the transparency of the shadow.
 6. To add a bevel
 - In the **Properties** pane, under **Color & Background**, double-click the **Bevel** property and click a bevel effect.

Add Colored Regions in a Current Default Chart

You can define colored regions in the body of a chart. For example, you can divide the background of a scatter chart into quadrants and color each quadrant.

You can add colored regions to bubble, bullet, combination, Pareto, progressive column, and scatter charts.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

In bullet charts, colored regions are automatically added.

You can use the following criteria to position the colored regions.

Option	Position
Percent on Axis (%)	Uses a percentage of the full range of the axis. For example, if the axis range is -500 to 1100, a Percent on Axis value of 25% puts the baseline at -100 (25% of the range, 1600).
Numeric Value	Uses a static numeric value.

Option	Position
Mean	Uses the statistical mean plus or minus a number of standard deviations based on all charted data values on the specified axis.
Percentile (%)	Uses a specified percentile.
Statistical Maximum	Uses the following expression: $25\text{th percentile value} - 1.5 * (75\text{th percentile value} - 25\text{th percentile value})$ <p>For example, if 2.5 is the 25th percentile and 7.5 is the 75th percentile, the statistical minimum is -5 [2.5 -1.5(5) = -5].</p>
Statistical Minimum	Uses the following expression: $75\text{th percentile value} + 1.5 * (75\text{th percentile value} - 25\text{th percentile value})$ <p>For example, if 2.5 is the 25th percentile and 7.5 is the 75th percentile, the statistical maximum is 15 [7.5 +1.5(5) = 15].</p>
Query Calculation	Uses a query calculation from the same query or from a different query. For more information, see " Using Relational Calculations " (p. 342).
Layout Calculation	Uses a layout calculation. For more information, see " Using Relational Calculations " (p. 342).
Category Index	Specifies a position based on the index value of the data item in the categories axis. The value must be between 0 and 100. For example, a Category index value of 1 indicates that the position is located at the first data item. This is the default.

You can also combine colored regions with gridlines.

Steps

1. Click the [chart object](#).
2. In the Properties pane, under **Color & Background**, double-click the **Colored Regions** property.
3. Click the new button 
4. Under **Region label**, type a name for the region.
5. Under **Label location**, specify whether the label should appear in the legend or in the chart.
6. If your chart includes multiple charts, such as in a combination chart, under **Based on**, select the chart to which the colored regions apply.

7. Under **Region fill**, click the color box to specify the color and fill effects.
8. Specify the start and end of the category and numeric positions.
9. If you want to add more colored regions, repeat steps 3 to 5.
10. If you add more than one colored region, specify their order using the up and down arrows.

The regions are drawn in the same order that they appear in this list. The first region is drawn first, on the bottom, and the last region is drawn on top of the other regions.

Tip: To delete a colored region, click the colored region icon and text and click the delete button .

The new region appears in the **Colored regions** box.

Specify Custom Label Text

By default, chart labels use the name of the underlying data item label. You can change the label text to make it more meaningful. For example, you can rename a chart item labeled **ISO_3_Letter_Code to Country**.

If you specify custom labels for a microchart or win-loss charts, the custom labels appear in the tooltips, not on the chart itself.

Steps

1. Click the chart node member or measure.
2. In the **Properties** pane, under the **Chart Labels**, set the **Custom Label** property to **Show**.
A new chart text item appears above the default measure or below the chart node member.
3. Do one of the following:
 - To change the text of the label, double-click the chart text item and type the text.
 - To change the source of the label text, select the new chart text item. In the **Text Source** section of the **Properties** pane, set the **Source Type** property.

If nothing is specified in the new chart text item, the label appears blank.

Tip: To remove a custom label, set the **Custom Label** property to **Hide**.

Customizing the Axes of a Chart

Chart axes are lines that border the chart area and provide a reference for measurements. They act as a vehicle for tick marks and scales and form a frame around the chart. The Y-axis is usually the vertical axis and contains data. The X-axis is usually the horizontal axis and contains categories.

To make your chart data easier to understand, you can do the following to customize each axis:

- show or hide the axis labels (data values)
- show or hide the axis line
- change the color, style, and weight of the axis line

- show or hide major and minor tick marks and specify where to display them

In HTML/PDF, if you have a report with a line chart, you can use the **Include Zero For Auto Scale** property to adjust the scale of the Y-axis of the chart. If the difference between your measure's maximum/minimum values is much less than the difference between your measure's minimum value and zero, using this property will make the chart Y-axis begin at something closer to your measure's minimum value, making the line utilize all of the chart's vertical space. In some cases this is significant in improving the readability of the chart.

Customize the Axes of a Current Default Chart

You can customize the axis labels, axis lines, and minor and major tick marks. You can change the scale for the major and minor tick mark using the **Axis Range** property. For more information, see ["Change the Axis Scale of a Current Default Chart" \(p. 163\)](#).

For more information about legacy and current default charts, see ["Current Default Charts and Legacy Charts" \(p. 127\)](#).

Steps

1. Select the Y-axis or the X-axis of the chart.
2. To show or hide the axis labels, in the **Properties** pane, set the **Axis Label** property.
3. If your chart includes nested categories, to change how the category axis labels are displayed, set the **Nested Label Display** property.
4. To show or hide axis lines, in the **Properties** pane, double-click the **Axis Line** property, and select or clear the **Axis line** check box.
5. To change the color, style, and weight of the axis line, in the **Properties** pane, set the **Axis Line** property.
6. To show or hide minor and major tick marks and specify their location, in the **Properties** pane, set the **Axis Line** property.

Customize the Axes of a Legacy Chart

You can customize the axis labels and axis line.

For more information about legacy and current default charts, see ["Current Default Charts and Legacy Charts" \(p. 127\)](#).

Steps

1. Select the Y-axis or the X-axis of the chart.
2. To show or hide the axis labels, in the **Properties** pane, set the **Axis Label** property.
3. Select the **Axis line** check box.
4. Click **Line color** and choose a color for this axis line.
5. To show or hide axis lines, in the **Properties** pane, double-click the **Axis Line** property, and select or clear the **Axis line** check box.

6. To change the color, style, and weight of the axis line, in the **Properties** pane, double-click the **Axis Line** property.

Change the Axis Scale of a Current Default Chart

By default, Report Studio automatically determines the minimum and maximum scale values for the axes in a chart. For example, a Y-axis showing revenue values might have an axis range of zero dollars to one million dollars. You can customize the axis scale, or range, to make your chart easier to understand. You can specify the following for each axis:

- the maximum and minimum values of the range
- whether to include zero in an automatic axis range
- whether to use a logarithmic scale, if the values in the chart cover a very large range
- how often major and minor gridlines appear

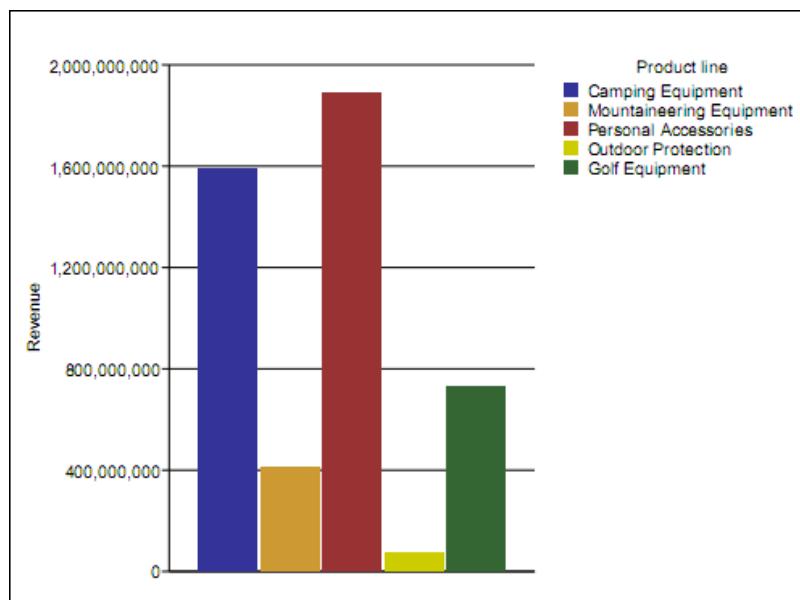
For more information about legacy and current default charts, see "["Current Default Charts and Legacy Charts" \(p. 127\)](#).

Logarithmic Scales

Logarithmic scales can be useful when some of the data you are displaying is much less or much more than the rest of the data or when the percentage or ratio differences between values are important.

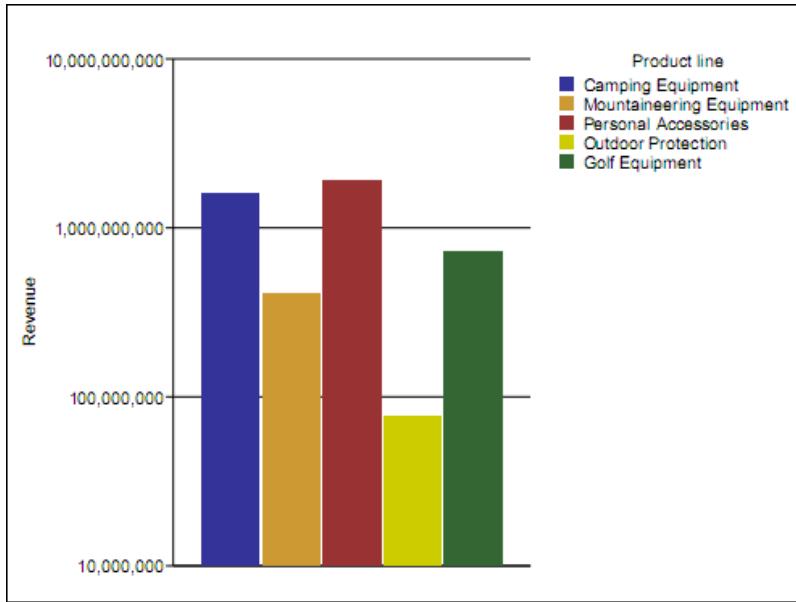
A logarithmic scale in Report Studio shows the base value of 10 raised to the power of a value. For example, 10 has a logarithm of 1 because 10 raised to the power of 1 is 10, 100 has a logarithm of 2 because 10 raised to the power of 2 is 100, and so on.

For example, the chart below is using a normal scale:



In this chart, the Y-axis shows values up to 2,000,000,000 with evenly spaced intervals of 400,000,000. However, the Outdoor Protection column is so much less than the other columns that it is difficult to compare them.

The following is the same chart using a logarithmic scale:



In this chart, the Y-axis still shows the same revenue values, but the intervals reflect a logarithmic scale, which increases exponentially. You can now compare all the product lines because none of the product lines have much higher or much lower values than the others.

Logarithmic scales can also be useful when the percentage or ratio differences between values are important. For example, if the logarithmic scale chart above represents data from 2008, and you add data from 2009 as a second set of colored bars, you could see the differences between the 2008 revenue and 2009 revenue. In a logarithmic scale, differences between values on the Y-axis represent the same percentage for each bar. So if the 2008 and 2009 data differs by the same distance for each product line, you could deduce that your revenue went up by the same percentage for each product line. This would not be clear on a normal scale.

Steps

1. Select the Y-axis or the X-axis of the chart.
2. In the Properties pane, under **General**, double-click the **Axis Range** property.
3. To set a maximum value for this axis, under **Maximum**, click **Manual**, and then type a maximum value in the **Manual** box.

Note: You can use either a positive or negative value as the maximum value.

4. To set a minimum value for this axis, under **Minimum**, click **Manual**, and then type a minimum value in the **Manual** box.

Note: You can use either a positive or negative value as the minimum value.

5. To include zero on this axis, select the **Include zero** check box.
6. To use a logarithmic scale for this axis, select the **Logarithmic scale** check box.

Note: If you choose to use a logarithmic scale, ensure that your chart consumers know that the scale is logarithmic by including that information in the axis title or in a note on the chart.

7. To set the position of major gridlines and tick marks, under **Major interval**, click **Manual**, and then type the distance between major gridlines and tick marks in the **Manual** box.

The distance between major gridlines and tick marks is measured in the units of that axis. For example, if the axis is revenue in dollars, type the dollar value in the **Manual** box.

8. To add minor gridlines, type the number of minor gridlines that you want to see between each major gridline in the **Number of minor intervals** box.

Showing Gridlines in a Chart

To make the data in a chart that includes axes easier to read, you can show horizontal and vertical gridlines. You can show gridlines for the major or minor intervals on the axes.

You cannot show gridlines for chart types that do not display axes, such as pie and donut charts.

Show Gridlines in a Current Default Chart

You can also show alternating bands of color in the chart background that correspond to your axis gridlines.

You can change the scale for the major and minor gridlines using the **Axis Range** property. For more information, see "[Change the Axis Scale of a Current Default Chart](#)" (p. 163).

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the Y-axis or the X-axis of the chart.
2. To show alternating bands of color as your chart background, do the following:
 - In the **Properties** pane, double-click the **Gridlines** property.
 - Select the **Show alternating color bands** check box.
 - Set the color and transparency of the first and second colors.
3. To show major gridlines as your chart background, do the following:
 - In the **Properties** pane, double-click the **Gridlines** property.
 - Select the **Show major gridlines** check box.
 - Set the color, style, and weight of the major gridlines.
 - If you are working with a gauge chart, specify the length of the gridline.
4. To show minor gridlines as your chart background, do the following:
 - In the **Properties** pane, double-click the **Minor gridlines** property.
 - Select the **Show minor gridlines** check box.
 - Set the color, style, and weight of the minor gridlines.

- If you are working with a gauge chart, specify the length of the gridline.

Show Gridlines in a Legacy Chart

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the Y-axis or the X-axis of the chart.
2. In the Properties pane, under General, double-click the Gridlines property.
3. To specify the color, style, and weight of major gridlines do the following:
 - Select the Show major gridlines check box.
 - Click Line color and choose a color for the major gridlines.
 - To specify the thickness of the gridlines, click Point size and select a line weight from the list.
 - Click a line style in the Line style list.
4. To specify the color, style, and weight of minor gridlines:
 - In the Properties pane, under General, double-click the Minor Gridlines property.
 - Select the Show minor gridlines check box.
 - Click Line color and choose a color for the minor gridlines.
 - To specify the thickness of the gridlines, click Point size and select a line weight from the list.
 - Click a line style in the Line style list.
 - To change the transparency of the gridlines, type a percentage in the Transparency box.

Customize the Legend in a Current Default Chart

You can hide or show the legend and change its position relative to the chart area, chart body, or a data item. For example, in a bar chart showing revenue for each product line by country, you could use the expression [Country] = 'Canada' to position the legend relative to the Canada bar.

When you choose a preset legend position, that position appears inside any padding that you have added to the chart object. Customized legend positions do not include any chart padding.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the [chart object](#).
2. In the Properties pane, under Chart Annotations, double-click the Legend property.

3. To show the legend on the chart, select the **Show Legend** check box.
4. To choose a preset legend position, click **Preset** and click a position from the diagram.
5. To choose a customized legend position, do the following:
 - Click **Advanced** and click the ellipsis (...) button.
 - To display the legend at a set distance from the sides of the chart area, from the **Anchor** list, click **Relative to Chart**.
 - To display the legend at a set distance from the sides of the chart body, from the **Anchor** list, click **Relative to Chart Body**.
 - To display the legend at a set distance from a data item, such as a specific pie slice in a pie chart, from the **Anchor** list, click **Report Expression**, click the ellipsis (...) button beside **Expression**, and enter an expression in the **Report Expression** dialog box.
 - Set the horizontal and vertical distances from the anchor.

Customize the Items in the Legend of a Current Default Chart

If your legend includes items that are too long, you can truncate long legend items at a specific number of characters. For example, if you want an ellipsis (...) to appear at the end of each truncated legend item, type ... in the **Truncation text** box.

You can also show the values of the data items in your legend.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the legend.
2. To truncate the legend
 - In the **Properties** pane, under **General**, double-click the **Text Truncation** property.
 - To specify the number of characters at which the legend items are truncated, click **Manual** and type the number of characters in the **Maximum characters** box.
 - To shrink the font of the legend item text until all the text fits in the legend, select the **Shrink font as needed** check box.
 - To specify some text to appear at the end of truncated legend items, in the **Truncation text** box, type the text that you want to appear at the end of truncated items.
3. To show the values of legend items within the legend
 - In the **Properties** pane, under **General**, set the **Show Values** property.

First Value and **Last Value** refer to the first and last item in the child set under the legend data item. **Minimum Value** and **Maximum Value** refer to the lowest and highest value in the child set under the legend data item.

- If you want to change the separator between the legend item and value, type a new separator in the **Legend Separator** property.
4. To customize the title of the legend
- Click the default legend title area in the work area.
 - In the **Properties** pane, under **General**, set the **Default Title** property to **No**.
 - Double-click the default legend title area in the work area and type the new title.

Customize the Items in the Legend of a Legacy Chart

If your legend includes items that are too long, you can truncate long legend items at a specific number of characters. For example, if you want an ellipsis (...) to appear at the end of each truncated legend item, type ... in the **Truncation** text box.

You can also show the values of the data items in your legend and customize the legend title.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the legend.
2. To truncate the legend text, do the following:
 - In the **Properties** pane, under **General**, set the **Auto Truncation** property to **Yes**.
 - In the **Maximum Characters** property, type the number of characters to appear before the text is truncated.
 - In the **Truncation Text** property, type the characters to append when the text is truncated.
 - To shrink the font of the legend item text until all the text fits in the legend, select the **Shrink font as needed** check box.
 - To specify text to appear at the end of truncated legend items, type it in the **Truncation text** box.
3. To show the values of legend items within the legend, do the following:
 - In the **Properties** pane, under **General**, set the **Show Legend Values** property.

First Value and **Last Value** refer to the first and last item in the child set under the legend data item. **Minimum Value** and **Maximum Value** refer to the lowest and highest value in the child set under the legend data item.
 - If you show multiple values and want to change the separator between the values, type a new separator in the **Separator** property.
4. To customize the title of the legend, do the following:
 - Click the default legend title area in the work area.

- In the Properties pane, under General, set the Default Title property to No.
- Double-click the default legend title area in the work area and type the new title.

Adding a Note to a Chart

Add a note to a chart to provide additional detail. Notes appear as text in a chart.

By default, notes are aligned with the upper left corner of the chart object. When you add a note to a chart, you can set the note's position relative to the sides of the chart area or chart body. You can also use a report expression to position the note next to a data item within the chart. For example, in a bar chart showing revenue for each product line by country, you could type [Country]='Canada' to position the note relative to the Canada bar.

Notes overwrite whatever is under them so you must position them properly.

If you apply more than one note, ensure that each note has a different position in the report so that they do not overwrite each other. You can also specify the order that they should be drawn in when the report runs. If you have two notes with the same coordinates, the first one in the list is drawn first and the next one is drawn on top of the first.

Tip: The Top 10 Retailers for 2005 sample report in the GO Data Warehouse (analysis) package includes a note. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Add a Note to a Current Default Chart

You can add and position multiple notes on your chart.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the [chart object](#).
2. In the Properties pane, under **Chart Annotations**, double-click the **Notes** property.
3. Click the new button .
4. Type the note text in the Text box and click OK.

Tips: To delete a note, click the note and click the delete button . To move a note up or down in the list, click the up or down arrows.

5. In the Notes dialog box, click OK.
6. To position the note:
 - Click the note icon  in the list of notes.
 - In the Properties pane, under **Positioning**, double-click the **Position** property.
 - To align the note horizontally, click the left, center, or right alignment button.
 - To align the note vertically, click the top, middle, or bottom alignment button.

- To set the margins around the chart body, type margin values and choose margin units.
 - To display the note at a set distance from the sides of the chart area, from the **Anchor** list, click **Relative to Chart**.
 - To display the note at a set distance from the sides of the chart body, from the **Anchor** list, click **Relative to Chart Body**.
 - To display the note at a set distance from a data item, from the **Anchor** list, click **Report Expression**, click the ellipsis (...) button beside **Expression**, and enter an expression in the **Report Expression** dialog box.
7. To edit the text, double-click the text next to the note object  in the chart.

Add a Note to a Legacy Chart

You can add and position multiple notes on your chart.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps for Legacy Charts

1. Click the [chart object](#).
2. In the Properties pane, under **Chart Annotations**, double-click the **Notes** property.
3. Click the new button .
4. Type the note text in the **Text** box and click **OK**.

Tips: To delete a note, click the note and click the delete button . To move a note up or down in the list, click the up or down arrows.

5. In the **Notes** dialog box, click **OK**.
6. Click the note icon  and set the note position, size, and border in the Properties pane.
You set the location of the note by defining number of pixels from the bottom left corner of the chart area.

Adding a Baseline to a Chart

Baselines are horizontal or vertical lines that cut through the chart to indicate major divisions in the data. For example, you can add a baseline to show a sales quota or break-even point.

Each baseline represents a value on an axis.

Depending on the type of chart, you can use the following options to position the baseline.

Option	Description
Numeric Value	Uses a static numeric value.

Option	Description
Query Calculation	Uses a query calculation from the chart query or from a different query. For more information, see " Using Relational Calculations " (p. 342).
Layout Calculation	Uses a layout calculation. For more information, see " Using Relational Calculations " (p. 342).
Category Index	<p>Uses the index value of the data item in the categories axis.</p> <p>For example, a Category index value of 1 indicates that the baseline is located at the first data item. This is the default.</p> <p>Note: This option applies only to the current default charts, and does not apply to the legacy charts.</p>
Member Value	<p>When working with dimensional data sources, uses a position relative to a member. For example, a member calculation that uses an expression similar to [Query1].[Current year]=2006 places the baseline in the middle of the chart object that represents 2006. To place the baseline between 2006 and 2007 on the chart, a half member width to the right, set the Member Offset property to 50%. To place the baseline between 2005 and 2006, set the Member Offset property to -50%. You can also type in 100, -200, and so on to place the baseline on the chart.</p>
Statistical Limit	<p>Statistical maximum uses the following expression:</p> $25\text{th percentile value} - 1.5 * (75\text{th percentile value} - 25\text{th percentile value})$ <p>For example, if 2.5 is the 25th percentile and 7.5 is the 75th percentile, the statistical minimum is -5 [2.5 -1.5(5) = -5].</p>
	<p>Statistical minimum uses the following expression:</p> $75\text{th percentile value} + 1.5 * (75\text{th percentile value} - 25\text{th percentile value})$ <p>For example, if 2.5 is the 25th percentile and 7.5 is the 75th percentile, the statistical maximum is 15 [7.5 +1.5(5) = 15].</p>
	<p>Statistical minimum and Statistical maximum use percentiles to determine values, so the baseline might not appear on the chart if its value is off the axis.</p>
Mean	Uses the statistical mean plus or minus a number of standard deviations based on all charted data values on the specified axis.
Percentile (%)	Uses a specified percentile.

Option	Description
Percent on Axis (%)	Uses a percentage of the full range of the axis. For example, if the axis range is -500 to 1100, a Percent on Axis value of 25% puts the baseline at -100 (25% of the range, 1600).

If you apply more than one baseline, you can specify the order in which they should be drawn when the report runs. If you have two baselines with the same coordinates, the first one in the list is drawn first and the next one is drawn on top of the first one.

By default, the baseline and its label appear in the legend.

Tip: The Positions to Fill sample report in the GO Data Warehouse (analysis) package includes a baseline. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Baselines do not support color transparency.

Add a Baseline to a Current Default Chart

When working with current default charts, you can define baselines by values on the numeric axis or the category axis.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Click the [chart object](#).
2. Depending on your chart and the type of baseline you want to add, in the **Properties** pane, under **Chart Annotations**, double-click the **Numeric Baselines** or **Category Baselines** property.
3. Click the new button  and choose the type of baseline from the list.
4. Specify the necessary criteria for the baseline position
5. Under **Baseline Properties**, type a label for the baseline and specify the line style.
Tip: To remove the baseline from the legend, delete the label.
6. If you add more than one baseline, specify their order using the up and down arrows.
The new baselines appear in the **Markers, notes, baselines, and trendlines** box.
7. To change the label that appears next to the baseline, under the **Markers, notes, baselines, and trendlines** box, double-click the placeholder text next to the baseline icon   and type your text.
8. To change the line style, select the chart and in the **Properties** pane, under **Chart Annotations**, double-click the **Baselines**, **Numeric Baselines**, or **Category Baselines** property.
9. To delete a baseline, select the baseline icon   and click the delete button .

Add a Baseline to a Legacy Chart

When working with legacy charts, you can define baselines by the values on the numeric axis.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Click the [chart object](#).
2. In the **Properties** pane, under **Chart Annotations**, double-click the **Baselines** property.
3. Click the new button  and choose the type of baseline from the list.
4. Specify the necessary criteria for the baseline position.
If you use a calculation to determine the baseline, define an expression.
5. If you add more than one baseline, specify their order using the up and down arrows.
6. Click **OK**.

A baseline icon appears in the **Markers, notes, and baselines** box.

7. To define the line style, click the baseline icon  and set the **Line Styles** property.
8. To define the text style, click the baseline text and make changes in the **Properties** pane.

Tip: To delete a baseline, click its baseline icon and click the delete button .

Adding Markers to Charts

Markers are symbols that you add to a chart to designate points of significance that can help you analyze or understand the data.

For example, you might want to add a marker to designate the time when a significant event happened, such as the date when a new product was launched.

You can add a marker to combination, progressive column, Pareto, scatter, and bubble charts.

When you define the position of the marker, you define the X- and Y-axes coordinates for the placement of the symbol. You can position the marker in the chart using the following options.

Option	Position
Numeric Value	Uses a static numeric value.
Query Calculation	Uses a query calculation from the same query or from a different query. For more information, see " Using Relational Calculations " (p. 342).
Layout Calculation	Uses a layout calculation. For more information, see " Using Relational Calculations " (p. 342).

Option	Position
Statistical Minimum	<p>Uses the following expression:</p> $25\text{th percentile value} - 1.5 * (75\text{th percentile value} - 25\text{th percentile value})$ <p>For example, if 2.5 is the 25th percentile and 7.5 is the 75th percentile, the statistical minimum is -5 [2.5 -1.5(5) = -5].</p> <p>Statistical minimum uses percentiles to determine values, and might not always appear in the chart if the values are off the axis.</p>
Statistical Maximum	<p>Uses the following expression:</p> $75\text{th percentile value} + 1.5 * (75\text{th percentile value} - 25\text{th percentile value})$ <p>For example, if 2.5 is the 25th percentile and 7.5 is the 75th percentile, the statistical maximum is 15 [7.5 +1.5(5) = 15]. Statistical maximum uses percentiles to determine values, so the marker might not always appear on the chart if its value is off the axis.</p>
Mean	Uses the statistical mean plus or minus a number of standard deviations based on all charted data values on the specified axis.
Percentile	Uses a specified percentile.
Percent on Axis	Uses a percentage of the maximum axis value.

If you apply more than one marker, you can specify the order in which they should be drawn when the report runs. If you have two markers with the same coordinates, the first one in the list is drawn first and the next one is drawn on top of the first.

Add a Marker to a Current Default Chart

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Note: Current default scatter charts have a different set of steps.

Steps

1. Select the [chart object](#).
2. In the **Properties** pane, under **Chart Annotations**, double-click the **Markers** property.
3. Click the new button , and then set the marker properties
4. In the **Based on** box, select the chart object that will determine the marker position.
5. In the **Numeric position** box, specify how you want to define the position on the numeric (Y) axis.

6. In the **Category position** box, specify how you want to define the position on the category (X) axis.
7. In the **Marker label** box, type the label that you want to give to the marker.
8. In the **Marker size** box, specify the size of the marker symbol.
9. To specify the color, fill, and shape of the marker symbol, click the **Marker color and style** box.
10. If you add more than one marker, specify their order by using the up and down arrows.

The new markers appear in the **Markers, notes, and baselines** box.

Tip: To delete a marker, click the marker icon and text and click the delete button .

11. Run the report.

Add a Marker to a Current Default Scatter Chart

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps for Current Default Scatter Charts

1. Select the scatter [chart object](#).
2. In the **Properties** pane, under **Chart Annotations**, double-click the **Markers** property.
3. Click the new button , and then set the marker properties.
4. In the **X-axis position** box, specify how you want to define the position on the X-axis.
5. In the **Y-axis position** box, specify how you want to define the position on the Y-axis.
6. In the **Marker label** box, type the label that you want to give to the marker.
7. In the **Marker size** box, specify the size of the marker symbol
8. To specify the color, fill, and shape of the marker symbol, click the **Marker color and style** box.
9. If you add more than one marker, specify their order using the up and down arrows.

The new markers appear in the **Markers, notes, and baselines** box.

Tip: To delete a marker, click the marker icon and text and click the delete button .

Add a Marker to a Legacy Chart

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the [chart object](#).
2. In the **Properties** pane, under **Chart Annotations**, double-click the **Markers** property.

3. Click the new button , and then set the **Numeric position type**, **Marker label**, **Marker shape**, **Numeric value**, and **Color** properties.
4. If you add more than one marker, specify their order by using the up and down arrows.

The new markers appear in the **Markers, notes, and baselines** box.

Tip: To delete a marker, click the marker icon or marker label and click the delete button .

Display Trendlines in Current Default Charts

You can display or hide trendlines in the current default bar, line, area, bubble, or scatter charts. Trendlines, also known as lines of best fit or regression lines, graphically illustrate trends in data series and are commonly used when charting predictions. A trendline is typically a line or curve that connects or passes through two or more points in the series, showing a trend.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

You can specify the following types of trendlines:

- **Linear**

Use a linear trendline when your data increases or decreases along a straight line at a constant rate. For example, if your chart displays a steady increase in revenue by product line over time, you could use a linear trendline.

- **Exponential**

Use an exponential trendline when your data values increase or decrease exponentially, or at an increasingly higher or lower rate. For example, if your chart displays an exponential increase in revenue by product line over time, you could use an exponential trendline.

- **Polynomial**

Use a polynomial trendline when your data values both increase and decrease. For example, if your chart displays both increases and decreases in revenue by product line over time, you could use a polynomial trendline.

- **Logarithm or Natural Logarithm**

Use a logarithmic trendline when your data values increase or decrease rapidly and then level out. For example, if your chart displays a rapid decrease in revenue by product line over time and then a plateau, you could use a logarithmic trendline.

- **Power**

Use a power trendline when your data values increase or decrease in a curve at a steady rate. For example, if your chart displays a steady increase in revenue by product line over time and your data points will fit a curved line, you could use a power trendline.

- **Moving Average**

Use a moving average trendline when your data values fluctuate and you want to smooth out the exceptions to see trends. For example, if your chart displays wild fluctuations in revenue

by product line over time, but you know that some data points are exceptions, you could use a moving average trendline.

If you are not sure which trendline type to use, try each type to see which one best fits most of your data points. For example, a linear trendline will not fit most points on a scatter chart with widely spread data points.

Trendlines do not support color transparency.

Steps

1. Select the [chart object](#).
2. In the **Properties** pane, under **Chart Annotations**, double-click the **Trendlines** property.
3. Click the new button  and click a trendline type.
4. Define the trendline by specifying the following options.

The options that are available depend on the type of trendline you chose.

- To set the order, or degree, of a polynomial trendline, in the **Order** box, type a value between 2 and 6.
- To set the number of periods to go back in a moving average trendline, in the **Periods** box, type a value.
- If you have more than one series on your chart, in the **Based on** list, click the data you want to use for the trendline.
- Click either **One trendline for all series items** or **A trendline for each series item**.
- To customize the style of the trendlines, click **Line Styles** and customize the line color, weight, and style.
- To customize the trendline labels in the legend, click **Label** and choose **None**, **Automatic**, or **Custom**.
- To display the trendline equation, click **Show equation**, and then click **Show in legend** or **Show on chart**, and then click **Position** to define the equation's position on the chart.
- To display the R-squared value of the trendline, click **Show R-squared value**, and then click **Show in legend** or click **Show on chart** and then click **Position** to define the position of the value on the chart.

Display Regression Lines in Legacy Scatter or Bubble Charts

When using legacy charts, you can display or hide regression lines on scatter charts and bubble charts. Regression lines, also known as lines of best fit or trend lines, graphically illustrate trends in data series. Regression lines are commonly used when charting predictions. A regression line is typically a line or curve that connects or passes through two or more points in the series, showing a trend. Regression lines always start at the zero intercept of the x-axis and y-axis.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

You can specify the following types of regression lines to determine the position and slope of the line:

- **Linear**

Use a linear regression line when your data increases or decreases at a consistent rate. For example, if your chart displays a steady increase in revenue by product line over time, you could use a linear regression line.

- **Common Log or Natural Log**

Use a logarithmic regression line when your data increases or decreases rapidly and then levels out. For example, if your chart displays a rapid decrease in revenue by product line over time and then a plateau, you could use a logarithmic regression line.

- **Exponential**

Use an exponential regression line when your data increases or decreases exponentially, or at steadily increasing or decreasing rate. For example, if your chart displays an exponential increase in revenue by product line over time, you could use an exponential regression line.

- **Polynomial Fit**

Use a polynomial fit regression line when your data both increases and decreases. For example, if your chart displays both increases and decreases in revenue by product line over time, you could use a polynomial fit regression line.

If you are not sure which regression line type to use, try each type to see which one best fits most of your data points. For example, a linear regression line is not going to fit most points on a scatter chart with widely spread data points.

Steps

1. Select the scatter or bubble chart.
2. In the Properties pane, under **Chart Annotations**, set the **Regression Line** property to **Show**.
3. Click the regression line icon  in the report layout.
4. In the Properties pane, under **General**, set the **Line Styles**, **Regression Type**, and **Number of Regression Lines** properties.

Show Data Values or Labels in a Current Default Pie or Donut Chart

You can show the data labels or data values within the slices or dots of pie or donut charts. For example, in a pie chart, show the data values within each pie slice, so that you know the exact size of each pie slice.

There are separate steps for other current default charts.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the pie or donut [chart object](#).
2. In the **Properties** pane, under **Chart Labels**, double-click the **Show Values** property.
3. To display data labels for each slice of the pie, under **Show**, select the **Slice names** check box.
4. To show lines pointing from data labels to the slices they apply to, select the **Show leader lines** check box.
5. To specify the data label format, in the **Values** list, click **Hide**, **Absolute**, **Percentage**, or **Absolute and Percentage**.
 - **Hide** does not display data values.
 - **Absolute** displays the absolute value of the data.
 - **Percentage** displays the slice's percentage of the whole pie.
 - **Absolute and Percentage** displays the slice's percentage of the whole pie as an absolute value.
6. To specify the placement of data labels, in the **Position** list, click **All around the pie** or **Aligned left and right of the pie**.
7. To specify how to display the labels if their positions on the chart overlap, in the **Collision mode** list, click one of the following modes:
 - **None** specifies that labels appear in default positions and might overlap.
 - **Normal** specifies that labels are placed just above their corresponding data markers or chart objects. There is no collision detection, so labels can overlap.
 - **Coarse Stagger** specifies that labels are placed close to their data markers and staggered so that they do not overlap. This collision mode takes less time to render than **Fine Stagger** but might result in labels being farther away from their corresponding data markers.
 - **Fine Stagger** specifies that labels are staggered so that they do not overlap. The labels are as close to the data markers as possible without overlapping. This collision mode takes more time to render than **Coarse Stagger** but might result in labels being closer to their corresponding data markers.

Show Data Values or Labels in Other Current Default Charts

There are separate steps for current default pie and donut charts.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the [chart object](#).
2. In the Properties pane, under **Chart Labels**, double-click the **Data Labels** property.
3. To show lines pointing from data labels to the data marker they apply to, select the **Leader lines** check box.
4. For each of the categories, series, and measures, select whether to show only values, values and labels, or neither.
5. To specify how to display the labels if their positions on the chart overlap, in the **Collision Mode** list, click one of the following modes:
 - **None** specifies that labels appear in default positions and might overlap.
 - **Normal** specifies that labels are placed just above their corresponding data markers or chart objects. There is no collision detection, so labels can overlap.
 - **Coarse Stagger** specifies that labels are placed close to their data markers and staggered so that they do not overlap. This collision mode takes less time to render than **Fine Stagger** but might result in labels being farther away from their corresponding data markers.
 - **Fine Stagger** specifies that labels are staggered so that they do not overlap. The labels are as close to the data markers as possible without overlapping. This collision mode takes more time to render than **Coarse Stagger** but might result in labels being closer to their corresponding data markers.

Define Query Context When Customizing Legend Entries, Legend Titles, or Axis Labels

You want to use a revenue expression as a chart legend title. If you get an error message saying that the query context of a layout object cannot be determined, you must define the property list for the item to which it refers. You must first add the desired data item to the query before you can define its property list. For more information, see "[Specify a List of Data Items for an Object](#)" (p. 378).

Steps

1. Open the chart to customize.
2. Pause the pointer over the query explorer button  and click the query.
3. In the **Insertable Objects** pane, drag the desired item to the **Data Items** window to add it to the query.
4. Pause the pointer over the page explorer button  and click the chart page.
5. In the **Insertable Objects** pane, drag the desired item to the layout object.

6. In the **Properties** pane, under **Data**, double-click the **Properties** property.
7. Select the check box for the data item to define.

Summarize Small Slices, Bars, or Columns in Current Default Charts

You can summarize the smaller slices or bars in charts to avoid having many tiny slices or bars. For example, if your pie chart shows revenue by product and 10 of your products have less than 1% of the pie, you can summarize these 10 slices into one larger slice and name that slice Other.

Similarly, you can summarize small items in a column, bar, area, and line chart.

You cannot summarize small slices or items in charts that have matrix edges or in charts that have multiple numeric axes.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the [chart object](#).
2. In the **Properties** pane, under **General**, double-click the **Summarize Small Slices** or **Summarize Small Items** property.
3. To summarize small slices or items up to a maximum number, select the **Maximum number of slices** or **Maximum number of items** check box and type the maximum number.
4. To summarize all the slices or bars that are smaller than a specific value, select the **Summarize slices smaller than a value** or **Summarize items smaller than a value** check box, type a value that represents the upper size limit, and choose whether the value is a percentage or absolute.
5. Under **Small slice summarization type** or **Small item summarization type**, choose whether to summarize the slices or items as a **Total** or **Average**.
Note: Averages are calculated using only the summarized items in the report.
6. Under **Slice label** or **Item label**, type a label for the one large slice, bar, area, or line that summarizes the smaller ones.

Customizing Lines and Data Points in a Line Chart

You can customize the lines in a line chart to show only lines, only data points, or both. Data points represent series values for each category on the Y-axis. You can show special data markers that represent statistically significant values, such as open, high, low, and close values.

You can also add markers at other positions on the chart. For more information, see "[Add a Marker to a Current Default Chart](#)" (p. 174).

You can also change the shape of the line that connects the data points to one of the following:

- **Point to Point** shows data points connected by straight lines.
- **Step at Point** shows data points connected by steps that start and end at the data points.

- **Step Between Points** shows data points connected by steps that start and end between the data points.
- **Smooth** shows data points connected by smooth curves.

Customize Lines and Data Points in a Current Default Line Chart

You cannot show value markers under the following conditions:

- The **Data points** option is selected.
- The configuration of the line chart is stacked or 100 percent stacked.

For more information about legacy and current default charts, see "["Current Default Charts and Legacy Charts" \(p. 127\)](#)".

Steps

1. Select the [chart object](#).
2. In the chart area, under **Series**, click the line chart icon to view the line properties.
3. To select whether to show only the line, the line and data markers, or only the data markers, in the **Properties** pane, select an option from the **Line and Markers** list.
4. To show special data markers, double-click the **Value Markers** property. Select the check box for the special data markers that you want to add and for each marker, specify the color and shape.

The options you specify in **Value Markers** overwrite the options for **Line and Markers**.

5. To change the shape of the line that connects data markers, select an option from the **Line Shape** list.

6. To show data labels for the data points, double-click the **Data Labels** property.

For more information, see "["Show Data Values or Labels in a Current Default Pie or Donut Chart" \(p. 178\)](#)".

7. To change the color of the lines, double-click the **Palette** property.

For more information, see "["Customizing the Color Palette of a Chart" \(p. 151\)](#)".

Customize Lines and Data Points in a Legacy Line Chart

You cannot show value markers under the following conditions:

- The **Data points** option is selected.
- The configuration of the line chart is stacked or 100 percent stacked.
- The **Show line** property is set to **No**, creating a point chart.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the [chart object](#).
2. In the chart area, under **Series**, click the line chart icon to view the line properties.
3. To select whether to show only the line, the line and data markers, or only the data markers, in the **Properties** pane, under **General**, select an option from the **Line Type** list.
4. To show or hide the line, select an option from the **Show line** list.
5. To show data points or value markers, double-click the **Show Data Points** property:
 - To show or format data points, click **Data points** and specify the point shape and point size.

The point color is defined by the chart palette. For more information, see "[Customizing the Color Palette of a Chart](#)" (p. 151).

- To show or format value markers, click **Value markers** and specify the markers to show. For each marker, specify the shape and marker size. Click **Color** to change the marker color.

Tip: To remove all data points and value markers, in the **Show Data Points** dialog box, select **None**.

Customize a Current Default Combination Chart

Combination charts show data series using two or more types of charts - area, bar, and line. The different charts are overlaid on top of each other. You can customize the order in which the charts appear along with the type of charts and their configurations.

You can also customize which numeric axes to show and which chart to show on each axes.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the combination [chart object](#).
2. In the **Properties** pane, under **General**, double-click the **Combinations** property.
3. Under **Numeric axes**, select which axes to show.
4. Under **Combinations**, add or remove data series.
5. If you want to change the order in which the series appear, use the up and down arrows.

The series appear in the order they are listed. Each chart appears in the foreground of any previous charts.

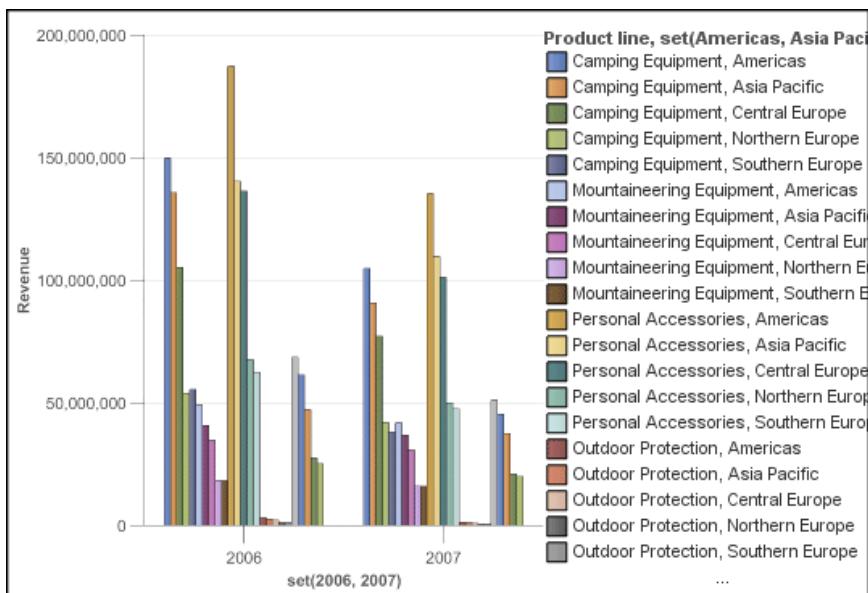
6. If you want to change the configuration type of the series, such as changing clustered bars to stacked bars, under **Combinations**, select the series, click the edit button  and select the type.
7. If you use the same data series for multiple charts and want to synchronize the data marker colors, under **Color & Background**, set the **Series Color** property to **Match**.

Create a Matrix of Current Default Charts

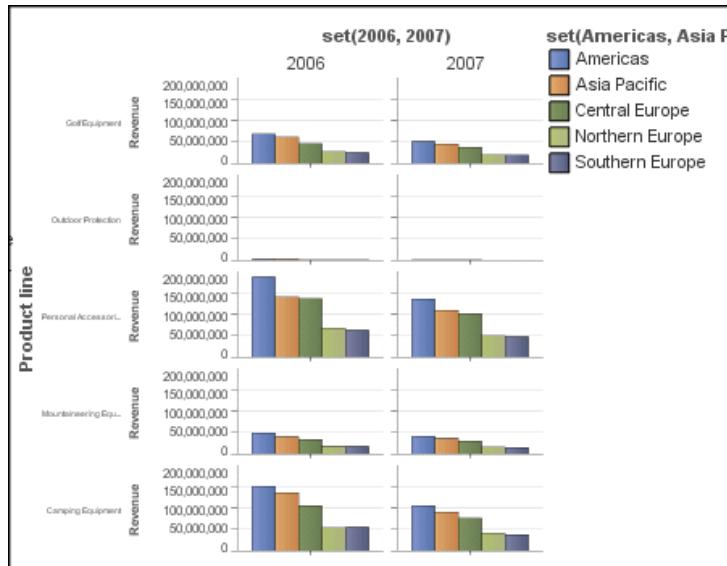
You can view a complex chart that includes nested series or categories into a matrix, or crosstab, that shows multiple small charts arranged in rows and columns. The charts in the rows represent the outer nested levels of the series and categories and, the charts in the columns represent the categories. Each data item in the outer nested levels of the series and categories becomes a separate chart. The numeric scale of all the charts is the same so that you can easily compare.

When working with pie, gauge, and bullet charts, if you include a data item in the categories, you automatically create a matrix of charts. One chart appears for each data item in the category. When working with progressive and bullet charts, if you include a data item in the series, you also automatically create a matrix of charts.

For example, the following column chart shows the revenue for each year (in the categories or X-axis) for all the regions and product lines (in the series or Y-axis). This chart is very complex and difficult to understand.



The following chart contains the same series and categories. However, when converted to a matrix of charts, the information is much easier to analyze. The columns show charts for each year and the rows show charts for each the product lines. The bars represent the revenue for each region.



For more information about legacy and current default charts, see ["Current Default Charts and Legacy Charts" \(p. 127\)](#).

Steps

1. Select the [chart object](#).
2. In the **Properties** pane, under **General**, double-click the **Matrix rows and columns** property.
3. If your matrix chart includes only rows or columns, and you want them to wrap, select the **Wrap rows or columns if possible** check box.
4. If you want to show the axes and axis labels for each small chart in the matrix, select the **Repeat row and column axis labels** check box.
When this option is cleared, axes and axis labels appear only along the outer edge of the matrix.
5. Select the **Show outer nested series as matrix rows** or **Show outer nested categories as matrix columns** check boxes (or both).
6. In the **Matrix levels** box, select the number of nested levels to include in the matrix columns or rows.

The remaining nested levels are represented in the chart bodies in the matrix if the chart supports additional categories. The pie, gauge, and bullet charts do not support additional categories. The bullet and progressive column charts do not support additional series.

For example, in the above chart, Regions are nested under Product Line. In the matrix chart, a **Matrix level** of 1 is specified. Therefore, Product Line appears as the matrix rows (series) and Regions appear within the chart bodies.

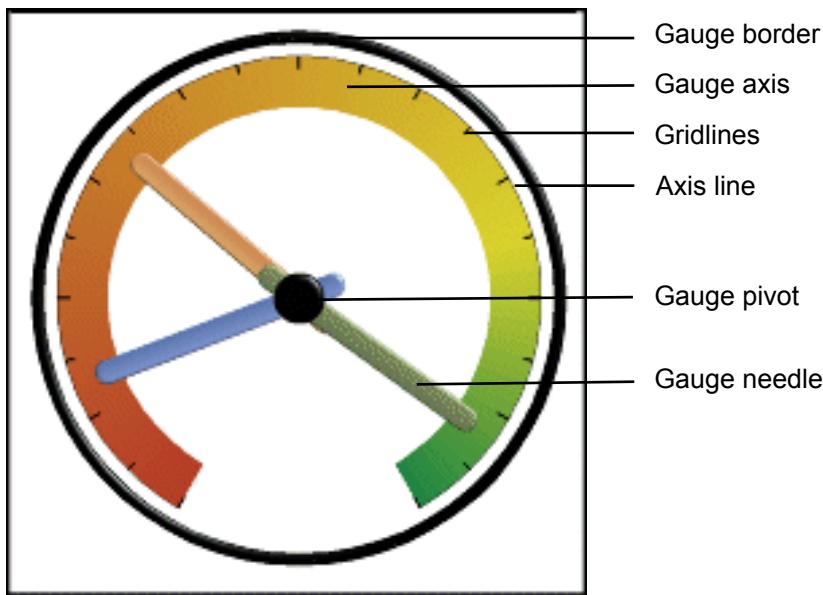
7. If you want labels for the matrix rows and columns to appear on each chart, select the **Show row labels** or **Show column labels** check box.
8. In the **Labels location** list, select where the nested labels should appear on each chart.
The default position is on the left for rows and on the bottom for columns.

9. If the labels are too long, click **Truncation**.
10. To change the font, color, and data format of the labels, click **Style**.
11. To show the title in the row or column axis, select the **Show row title** or **Show column title** check box.
12. If you want to hide or show the axes, select the axis object, and in the **Properties** pane, under **Miscellaneous**, set the **Show in Matrix** property.

Customize a Current Default Gauge Chart

When you create a gauge chart, you can choose from a variety of chart templates, which offer different shape, axis, and border options.

You can customize the following aspects of your gauge chart. As you modify properties, the chart preview shows you what your chart will look like.



Unless indicated in the user interface, all sizes are a percentage of the maximum allowed.

If your gauge chart includes a border, long axis labels, such as 250,000,000, may overlap the gauge border and be difficult to read. To avoid this problem, customize the data format of your gauge chart measure and reduce the scale so that less zeros are displayed. Or change the gauge border color or size or remove the border.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Click the gauge [chart object](#).
2. If you want to change the shape of the gauge, modify the start and end angles of the gauge axes and border, as follows:

- In the **Properties** pane, under **General**, double-click the **Gauge Axes** property and specify the start and end angles and the direction of the axes.
- Double-click the **Gauge Border** property and specify the style and the start and end angles of the border.

For example, a border start angle of 0 degrees and end angle of 180 degrees produces a semi-circular gauge.

3. If you want to add an additional axes to a gauge chart, do the following:
 - In the **Properties** pane, under **General**, double-click the **Gauge Axes** property.
 - Click the new button  and specify the start and end angles and the direction of the new axis.
4. If you want to change the size, shape, and color of the center pivot point, double-click the **Gauge Pivot** property and specify the style.
5. If you want to change the indicators in the gauge axis, click the **Gauge Axis** object in the chart and do the following:
 - To change the needle, under **Axes**, double-click the **Gauge Needle** property and specify the style.
 - To change, add, or remove color bands that indicate the data range positions, under **Color & Background**, double-click the **Gauge Axis Colors** property and specify the color palette.
 - If you want to change the size or thickness of the gauge axis and color bands, under **Axes**, specify a percentage size for the **Gauge Axis Inner Radius** and **Gauge Axis Outer Radius** properties.
 - If you want to change the gridlines, under **General**, double-click the **Gridlines** or **Minor Gridlines** properties and specify the style.
 - If you want to change the appearance of the gauge axis line, under **General**, double-click **Axis Line**.

Customize a Legacy Gauge Chart

You can customize the colors and threshold boundaries of the gauge axis areas and whether they use numbers instead of percentages. You can also change the colors of the gauge face and outline and hide gauge labels. By default, the legacy gauge chart uses a band divided into thirds that runs from green to red.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Click the gauge chart object.

2. If you want to customize the color of the gauge face or outline, in the **General** section of the **Properties** pane, double-click the **Face Color** or **Dial Outline Color** property.
3. If you want to hide the gauge labels, in the **Chart Labels** section of the **Properties** pane, set the **Gauge Labels** property to **Hide**.
4. If you want to customize the colors and boundaries of the gauge areas, do the following:
 - In the **Color & Background** section of the **Properties** pane, double-click the **Gauge Palette** property.
 - To change the color of a boundary area, under **Palette**, select the color, click **Color**, specify the color properties, and click **OK**.
 - To change the value of a boundary area, under **Palette**, select the boundary value, and type a new value.
 - To add a new boundary area, click the new button .
 - To choose a pre-defined threshold style, click the **Palette** drop-down menu.

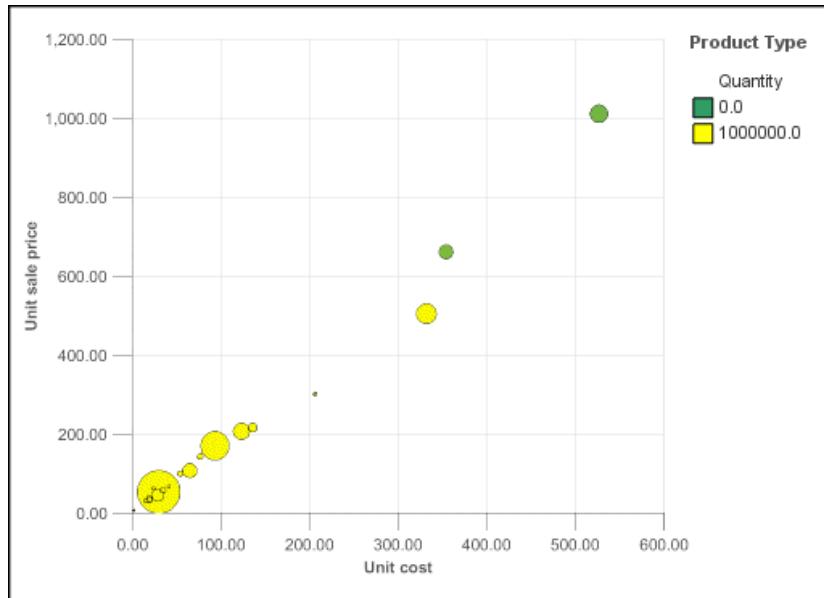
Tips

- You can choose whether to use discrete colors or continuous colors.
- To use a number instead of a percentage as a boundary, clear the **Percentage** check box, and then type a number in the **Numeric boundary** box.

Define Color by Value in Current Default Scatter and Bubble Charts

In a scatter or bubble chart you can specify that data points or bubbles appear in different colors based on an additional measure or fact. The color of the points or bubbles adds an additional aspect to the chart to assist you in finding relationships in large amounts of data.

For example, the following bubble chart shows the relationship between the unit cost and unit sale price. The size of the bubbles shows the gross profit and the color of the bubbles shows whether the quantity sold is above (yellow) or below (green) 1,000,000 units.



In a bubble chart, you can combine the color by value and bubble size to create a more meaningful chart.

For more information about legacy and current default charts, see ["Current Default Charts and Legacy Charts" \(p. 127\)](#).

Steps

1. Drag a measure to the Color drop zone under **Measures** and then select the measure.
2. In the Properties pane, under **Color & Background**, double-click the **Color by Value** property.
3. To define the colors by percentages instead of actual values, select the **Percentage** check box. For example, if the **Percentage** check box is selected and your values range from 25 (red) to 50 (green), then the bottom 25 percent of values will be red, the top 50 percent of values will be green, and the values between 25 and 50 percent will be an interpolated color, such as yellow.
4. If you want to use a preset color palette, click **Palette**, and select the palette that you want to use.
5. If you want to customize a palette color or boundary value, select the palette entry in the **Palette** box and specify the color, transparency, and type a new boundary value.
6. To add a new palette entry, select the palette entry below which you want to add the new entry, click the new palette entry button , and click **Color**.
7. If your chart has lines or markers, under **Style**, set the marker shape, line style, and line weight.
8. To choose a color and transparency for missing or null values, under **Missing Values**, click **Color** and type a value in the **Transparency** box.

Specify the Bubble Size in a Current Default Bubble Chart

In a bubble chart, you use a measure or fact to determine the size of the bubbles (the **Bubble Measure** in the **Measures** drop zone of the chart). You can then specify the size range of the bubbles in the chart. You can also specify what value the smallest bubbles represent.

For example, your bubble measure is revenue, and you set the minimum and maximum bubble size to 5 pt and 20 pt, respectively. You set the smallest bubble to represent the value zero. All the bubbles on your chart will be between 5 and 20 pts and any bubbles between 0 and 5 pts on this scale will appear at 5 pt.

You can combine bubble size with colors by value to create a chart that shows multiple dimensions.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the [chart object](#).
2. In Properties pane, under **General**, double-click the **Bubble Size** property.
3. Under **Smallest bubble**, select what value the smallest bubbles represent:
 - To show data items with a value of zero at the minimum bubble size, click **Zero**.
Tip: This setting is consistent with Microsoft® Excel 2003.
 - To show data items with a value of zero at the minimum bubble size and show negative bubbles as hollow, select the **Zero. Negatives shown as hollow** check box.
Tip: This setting is consistent with Microsoft Excel 2007.
 - To set the minimum bubble size to the minimum data value, click **Minimum data value**, and type a size for the minimum and maximum bubble size.
Note: The minimum data value can be positive or negative.

Set the Position of the First Slice in a Current Default Pie Chart

The default starting position of the first slice is 0 (zero), which displays the first slice beginning at the 3 o'clock position. By default, the slices appear counterclockwise around the pie, so a starting position of 90 displays the first slice at the 12 o'clock position, a starting position of 180 displays the first slice at the 9 o'clock position, and so on. You can change the direction of the slices so that they appear clockwise.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the pie [chart object](#).
2. In the **Properties** pane, under **General**, beside the **First Slice Angle** property, enter the angle at which you want the first slice to appear.

- To change the directions in which slices appear, change the Slice Direction property.

Create a Donut Chart from a Current Default Pie Chart

You can add a hole to the middle of your pie chart to create a donut chart. You can then display something in the hole, such as a company logo, a calculation, or the legend.

For more information about legacy and current default charts, see ["Current Default Charts and Legacy Charts" \(p. 127\)](#).

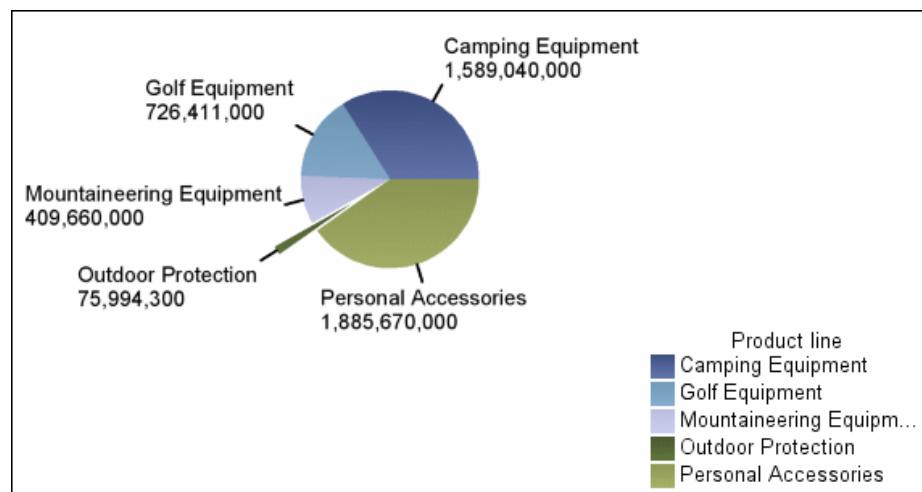
Steps

- Select the pie [chart object](#).
- In the Properties pane, under General, set the Hole size (%) property to the percentage of the pie that you want the hole to take up.

Tip: You can select a value from the Hole size (%) list or type a value that is not listed.

Pull Out Pie Slices in a Current Default Pie Chart

You can highlight pie slices by pulling them out from the rest of the pie. For example, the following chart shows revenue by product line with slices of less than 1,000,000,000 in revenue pulled out by 25%.



For more information about legacy and current default charts, see ["Current Default Charts and Legacy Charts" \(p. 127\)](#).

Steps

- Select the pie [chart object](#).
- In the Properties pane, under General, double-click the Exploded Slices property.
- In the Exploded Slice dialog box, click the new button .
- In the Exploded amount box, type the percentage by which the slice should appear away from the rest of the pie.

0% displays the pie slice in the pie; 100% displays the pie slice as far out from the rest of the pie as possible.

5. Define which slice to pull out:

- To select a slice to pull out using its slice index, under **Exploded slice**, click **Slice number** and type the slice number.

The slice number refers to the order of the slice in the legend. In the example above, Camping Equipment has a slice number of 1 and Personal Accessories has a slice number of 5.

- To select a slice to pull out using a calculation, under **Exploded slice**, click **Expression** and click the ellipsis (...) beside the **Expression** box to define an expression.

In the example above, the expression that defines which slices to pull out is as follows:
`[Query1].[Revenue] < 100000000`.

The slice definitions appear in the **Exploded Slices** list.

6. To pull out other slices, repeat steps 3 to 5.

Define the Cumulation Line in a Current Default Pareto Chart

The cumulation line on a Pareto chart displays the percentage of the accumulated total of all the columns or bars. For example, if your chart displays revenue by product line by year, the cumulation line at the second year's column would be the total revenue of the first and second years.

You can customize the appearance of the cumulation line and its data points.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the Pareto chart.
2. To hide or show the cumulation line, in the **Properties** pane, under **Chart Annotations**, set the **Cumulation Line** property.
3. To customize the cumulation line, click the cumulation line icon  , and in the **Properties** pane, under **General**, set the following properties:
 - **Line Styles** defines the cumulation line color, style, and weight.
 - **Cumulation Label** defines the label in the legend for the cumulation line. You can use the default label from the data source or type a custom label. The color and marker shape of the cumulation line still appears in the legend when this property is set to **None**.
 - **Data Points** defines whether to show or hide data points along the cumulation line, whether to show or hide data point borders, the color of data point borders, and the data point size and shape.
 - **Data Labels** specifies whether to show or hide the labels for the data points along the cumulation line.

Define the Cumulation Line in a Legacy Pareto Chart

The cumulation line on a Pareto chart displays the percentage of the accumulated total of all the columns or bars. For example, if your chart displays revenue by product line by year, the cumulation line at the second year's column would be the total revenue of the first and second years.

You can customize the appearance of the cumulation line and its data points.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the Pareto chart.
2. To hide or show the cumulation line, in the Properties pane, under **Chart Annotations**, set the **Cumulative Line** property to **Show**.
3. To customize the cumulation line, click the cumulation line icon  and in the Properties pane, under **General**, set the following properties:
 - **Cumulation Axis** displays or hides the cumulation line's axis on the right of the Pareto chart.
 - **Cumulation Label** displays or hides the cumulation line label in the legend. The color and marker shape of the cumulation line still appears in the legend when this property is set to **No**.
 - **Line Styles** defines the cumulation line color, style, and weight.
 - **Marker Size (pt)** defines the size of the markers along the cumulation line in points.
 - **Marker Shape** defines the shape of the markers along the cumulation line.
 - **Values** specifies whether to show or hide the values for the markers along the cumulation line.
 - **Value Location** defines the location of the marker values.

Insert a Microchart into a Crosstab

You can use microcharts to improve the visualization of data in crosstabs.

Steps

1. Select a crosstab row or column.
2. From the right-click menu, click **Insert Chart for Row Data** or **Insert Chart for Column Data**.
3. From the **Insert Chart** dialog box, select a chart and click **OK**.
4. Specify the data to plot in the microchart.

The chart automatically plots the data in the specified rows or columns. You can change this if necessary.

Customize a Current Default Bullet Chart

After you create a bullet chart, you can customize the shape, color, and size of the bullet and target indicators.

By default, the bullet chart includes three gray colored regions in the background. You can edit the colored regions (**Properties** pane, **Colored Regions**). For more information, see "[Add Colored Regions in a Current Default Chart](#)" (p. 159).

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

Steps

1. Select the bullet [chart object](#).
2. To change the shape, color, and size of the bullet or target
 - In the **Properties** pane, under **General**, double-click the **Bullet Indicators** property.
 - Under **Bullet**, specify how to show the bullet.

The **Bar width** setting specifies the width of the bullet bar as a percentage of the space available. For example, if you specify 50 percent, the bar uses half of the space available. If you specify 100 percent, the bar uses all the available space.

 - Under **Target**, specify how to show the target.
3. To change the chart orientation, in the **Properties** pane, under **General**, set the **Chart Orientation** property.

Create a Drill-up and Drill-down Chart

If you use a dimensionally-modeled data source, you can create a chart that allows you to drill down to lower-level data or drill up to higher-level data.

Drilling up and down allows you to view more general or more detailed information on your data within a predefined dimensional hierarchy.

This is an example of a dimensional hierarchy:

Years - Year - Quarter - Month

Before you begin, ensure that you are using a dimensionally-modeled data source.

For more information about using drill-up and drill-down reporting in Report Studio, see "[Create a Drill-up and Drill-down Report](#)" (p. 414).

Steps

1. Open a chart that uses a dimensionally-modeled data source.
2. From the **Data** menu, click **Drill Behavior**.
3. On the **Basic** tab, under **Report output drill capabilities**, select the **Allow drill-up and drill-down** check box.

By default, the system determines which items can be drilled on based on the dimensional structure.

On the **Basic** tab, you can make drilling unavailable for any data item by selecting the item in either the **Disable drill-up for** box or the **Disable drill-down for** box.

On the **Advanced** tab, you can change the drill-up or drill-down behavior for any parameter by selecting the parameter and then choosing one of the desired behaviors.

The chart generates links for any item that can be drilled down on.

You can drill down or drill up by right-clicking and choosing the action from the context menu. The menu items are unavailable if an item cannot be drilled up or down on.

Example - Creating Drill-through Access in a Legacy Chart

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a chart that shows the revenue for each product line and allows the reader to drill through from the revenue chart to view the product details for any item selected. You create a drill-through report to link two reports containing related information. You can then access related or more detailed information in one report by selecting a value in the chart. You create two reports: a target list report that contains the details for the item and a source report that contains the chart that shows the product line revenue.

For more information about using drill-through reporting in IBM® Cognos® Report Studio, see "[Set Up Drill-through Access in a Report](#)" (p. 523).

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

This example has three parts:

- [Create the target report.](#)
- [Create the source report.](#)
- [Define the drill behavior.](#)

Create the target report

This example has three parts:

1. [Create the target report.](#)
2. [Create the source report](#)
3. [Define the drill behavior.](#)

Steps

1. Open Report Studio with the GO Data Warehouse (query) package.
2. In the Welcome dialog box, click **Create a new report or template**.
3. In the New dialog box, click **List** and click **OK**.

4. In the Insertable Objects pane, on the Source tab , expand Sales and Marketing (query), Sales (query), and Product.
5. Double-click the following data items to add them to the list:
 - Product line
 - Product type
 - Product name
 - Introduction date
 - Product image

Now you must create a filter to use as a drill-through parameter. A drill-through parameter begins and ends with a question mark (?) symbol.

6. Click the filters button .
7. Click the add button  and type the following in the Expression Definition window:
[Sales (query)].[Product].[Product line]=?p_PL?
8. Save the report as **Product Line Details**.

Create the Source Report

This example has three parts:

1. [Create the target report](#).
2. [Create the source report](#)
3. [Define the drill behavior](#).

Steps

1. Create a new report.
2. In the New dialog box, click **Chart** and click **OK**.
3. In the **Chart group** pane, click **Column**.
4. In the **Chart type** pane, click **Column**.
5. Click **OK**.
6. In the Insertable Objects pane, on the Source tab , expand Sales and Marketing (query) and then Sales (query).
7. Expand Sales fact and drag Revenue to the **Measure (Y-axis)** drop zone.
8. Expand Order method and drag Order method to the **Series** drop zone.
9. Expand Product and drag Product Line to the **Categories (X-axis)** drop zone.

Define the Drill Behavior

This example has three parts:

1. [Create the target report.](#)
2. [Create the source report](#)
3. [Define the drill behavior.](#)

Steps

1. From the **Data** menu, click **Drill Behavior**.
2. On the **Basic** tab, under **Report output drill capabilities**, select the **Allow this report to be a package-based drill-through source** check box and click **OK**.
3. Right-click the [chart object](#) and click **Drill-Through Definitions**.
4. Click **New Drill Through**.
5. Under **Report**, click the ellipsis (...) button.
6. Select the **Product Line Details** report you created and click **Open**.
7. Under **Action**, click **Run the Report**.
8. Under **Format**, click **HTML**.
9. Click the edit button .

Any existing drill-through parameters appear. You see the parameter you created for **Product Line Details**.

10. For item **p_PL**, under **Method**, click **Pass data item value**, and under **Value**, click **Product line**.
11. Save the chart as **Product Revenue**.
12. Run the report.

The chart shows the product lines as clickable links. When you click a product line, the second report runs for that product line.

Example - Creating and Customizing a Metrics Range Legacy Chart

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are asked to create a chart that shows revenue compared to planned revenue by quarter by product line. You also want to highlight revenues that were below target.

For more information about legacy and current default charts, see "[Current Default Charts and Legacy Charts](#)" (p. 127).

This example has three parts:

- [Create a metrics range chart.](#)

- [Customize the chart.](#)
- [Customize the chart legend labels.](#)

Create a Metrics Range Chart

This example has three parts:

1. [Create a metrics range chart.](#)
2. [Customize the chart.](#)
3. [Customize the chart legend labels.](#)

Steps

1. Open IBM® Cognos® Report Studio with the GO Data Warehouse (query) package.
2. In the Welcome dialog box, click **Create a new report or template**.
3. In the New dialog box, click **Chart** and click **OK**.
4. In the **Chart group** pane, click **Metrics Range**.
5. In the **Chart type** pane, click **Column Chart with Range Indicators** and click **OK**.
6. In the Insertable Objects pane, on the Source tab , expand Sales and Marketing (query), Sales (query), and Sales fact.
7. Drag **Revenue** to the **Actual (Y-axis)** drop zone.
8. Drag **Planned Revenue** to the **Target (marker)** drop zone.
9. Expand **Time dimension**, and drag **Quarter** to the **Categories (X-axis)** drop zone.
10. On the Toolbox tab , drag a query calculation to the **Tolerance (marker)** drop zone.
11. In the Create Calculation dialog box, type **Tolerance**.
12. In the Expression Definition box, type **[Planned revenue] * 0.1** and click **OK**.
13. Run the report and then pause the pointer over each bar to see the revenue for that quarter.

Customize the Chart

This example has three parts:

1. [Create a metrics range chart.](#)
2. [Customize the chart.](#)
3. [Customize the chart legend labels.](#)

Steps

1. Select the chart body.

2. In the Properties pane, under Target Markers, ensure that the Performance Pattern property is set to **On Target**.
3. Double-click the **Marker Color** property, click **Lime** and click **OK**
4. Set the **Upper Range Skew (%)** property to **0%**.

This option removes the upper tolerance line. In this example, you do not need to see the upper tolerance line because the revenue is always below the planned revenue target.

5. Ensure that the **Target Range (%)** property is set to **50%**.

This option sets the size of the range around the planned revenue target. The percentage you choose is the percentage of the tolerance measure. In this example, a 50% target range would display 50% of the tolerance calculation you created, which is 50% of 10% of planned revenue.

6. Double-click the **Target Color** property, click **Red** and click **OK**.
7. Double-click the **Target Marker Border Color** property, click **Green** and click **OK**.
8. Run the report.

Customize the Chart Legend Labels

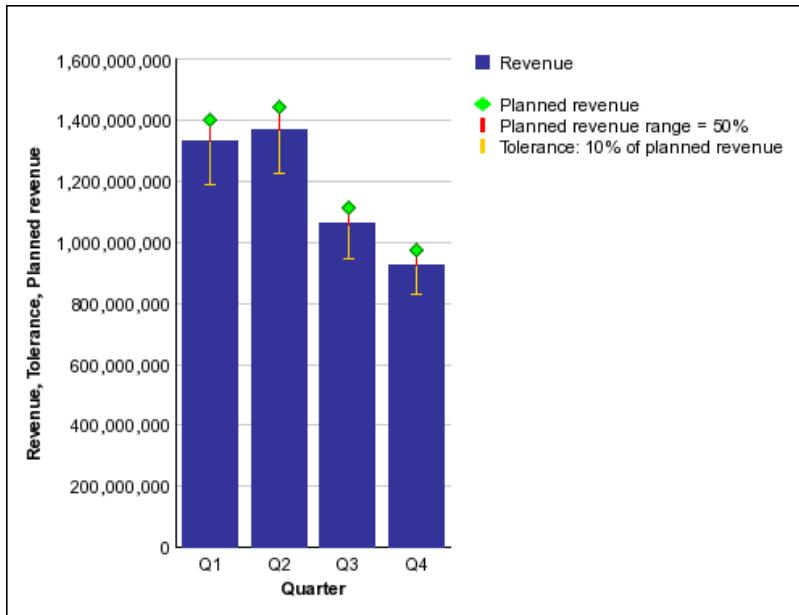
This example has three parts:

1. [Create a metrics range chart](#).
2. [Customize the chart](#).
3. [Customize the chart legend labels](#).

Steps

1. Under the chart legend select **Marker**.
2. In the Properties pane, under **Text Source**, double-click the **Text** property.
3. Replace the text in the **Text** dialog box with **Planned revenue** and click **OK**.
4. Under the chart legend select **Tolerance**.
5. In the Properties pane, under **Text Source**, double-click the **Text** property.
6. Replace the text in the **Text** dialog box with **Tolerance: 10% of planned revenue** and click **OK**.
7. Under the chart legend, select **Range**.
8. In the Properties pane, under **Text Source**, double-click the **Text** property.
9. Replace the text in the **Text** dialog box with **Planned revenue range = 50%** and click **OK**.
10. Run the report.

The report now shows customized legend labels for the markers, tolerances, and ranges as shown below.



Adding a Metric Studio Diagram to a Report

You can add IBM® Cognos® Metric Studio impact or custom diagrams in your reports. Impact diagrams show cause-and-effect relationships between metrics. Custom diagrams allow you to monitor your metrics using a predefined visual representation, such as a process diagram or strategy map.

The diagrams are added as images within your report.

You can add a diagram in the following ways:

- as a static image with the **Metric Studio Diagram** object
- as a dynamic image from a metrics package

You can also add range indicator charts.

You must first create custom diagrams in Metric Studio before you can add them in an IBM Cognos Report Studio report. Impact diagrams are automatically created in the metrics package. For information about creating Metric Studio diagrams, see the Metric Studio *User Guide*.

Add a Static Metric Studio Diagram to a Report

You insert the image using the diagram identifier from Metric Studio. The identifier is converted to an image URL and the image appears in your report output.

Because the diagram is a static image, any changes to the metrics will not appear in the diagram image within your report.

Steps

1. In the **Insertable Objects** pane, on the **Toolbox** tab , drag a **Metric Studio Diagram** object to the report.
2. Select the Metric Studio diagram object.

3. In the **Properties** pane, paste the diagram identifier in the **Diagram Identifier** box.
Tip: Copy the identifier from Metric Studio (**Diagrams** tab, **View the Diagram Identifier** button in the **Actions** column).
4. In the **Description** box, type a description for the diagram.
5. If you want to make the diagram image accessible, in the **Alternate Text** box, type a description for the image.

For more information, see "[Add Alternate Text to Images and Charts](#)" (p. 987).

Add a Data-driven Metric Studio Diagram to a Report

If you use a metrics package (which is a relational data source), you can insert a Metric Studio diagram directly from the source tree in the **Insertable Objects** pane. Because you add the diagram from the metrics store, any changes to the metrics will appear in the diagram image when you re-run your report.

If you add diagrams from the **Metric History** table of the metrics package, one diagram is added for each metric per time history. As a result, your report could contain a large number of diagrams. To improve the performance of your report, consider adding a filter to limit the time period.

Steps

1. Open Report Studio with a metrics package.
2. In the **Insertable Objects** pane, from the **Source** tab , insert a diagram query item to the report.
3. If you want to make the diagram image accessible, select the image object for the diagram, and in the **Alternate Text** box, type a description for the image.

For more information, see "[Add Alternate Text to Images and Charts](#)" (p. 987).

Chapter 7: Maps

IBM® Cognos® Report Studio provides a set of maps that you can use to represent tabular data in a spatial context. For example, on a map of the world, countries can be colored to represent the level of revenue.

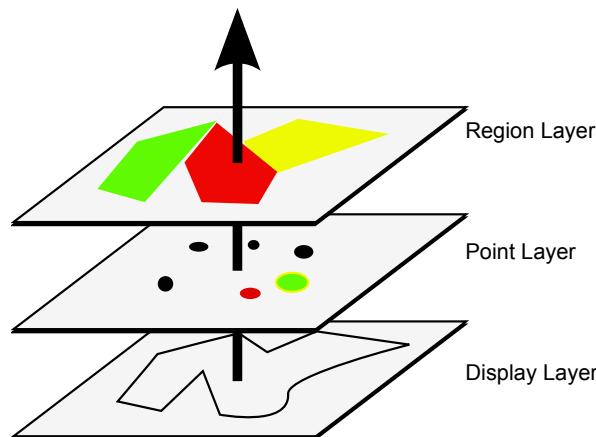
Maps are not supported for reports run in Microsoft® Excel format.

The Report Studio Map Object

Maps are most often used to show geographical areas, but they can be used to show other spatial information, such as a floor plan in a building, seats in an airplane, or parts of the human body.

Maps are similar to crosstabs in the way their data is organized. The display is different, but maps show the intersection of data the same ways as crosstabs; for example, you can see the revenue for golf equipment in Canada.

A map in IBM® Cognos® BI consists of a collection of layers. Each layer contains different information and acts like a transparency that is placed on top of the map. Each layer adds more information to the map. For example a map of the world may contain information related to countries on one layer and information related to cities on another level.



IBM Cognos Report Studio maps provide the following three types of layers:

- Region layer

Specifies the regions on a map to be differentiated according to values in the data source. For example, to show the revenue level for each country on a map of the world, choose Country as the region layer and then specify that the color of each country is based on the revenue value for that country. Regions can be set up for drilling through to other reports.

- Point layer

Specifies the points to be placed on a map. The color and size of the points is based on the data that you select. For example, you choose to show cities as points on a map and set the color

of each point by revenue and the size of each point by profit. Points can be set up for drilling through to other reports.

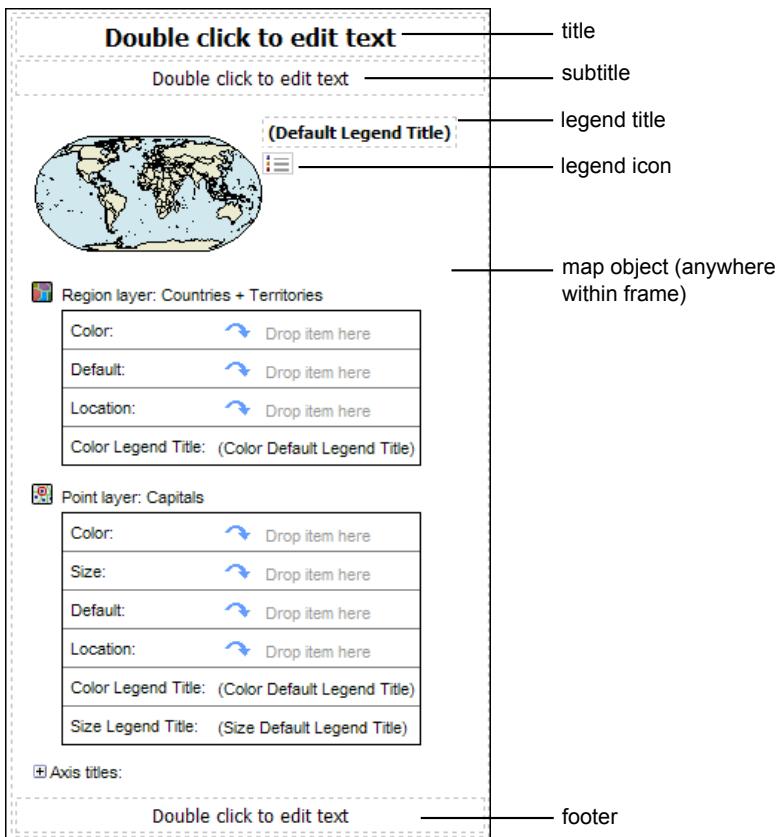
- Display layer

You can show or hide items such as grid lines or capital cities. This layer is determined in the map file and not in the data source.

Note: If you intend to create CSV or XML output from your map, use only a point layer or a region layer. CSV and XML do not support the simultaneous use of both layers in the same map. Only one layer will be rendered in the output.

Parts of Map Reports

The following shows the parts of a map as they appear in the Report Studio interface.



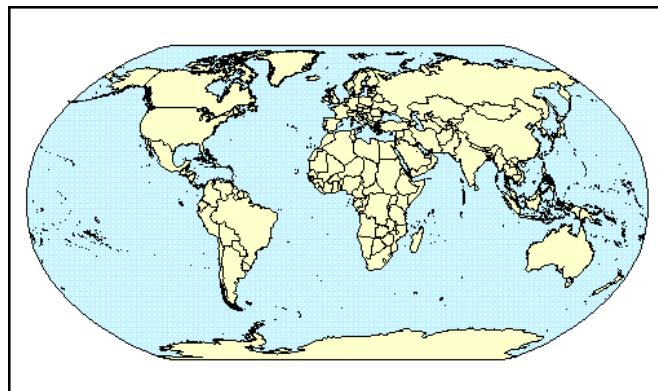
Example - Create a Map Report

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are asked to show how revenue for the company is distributed throughout the world. This information can be shown in tabular format using a list report, but a map will create a more meaningful presentation. You decide to create a report that contains a map of the world showing the distribution of revenue by country.

Steps

1. Open IBM® Cognos® Report Studio with the GO Data Warehouse (query) package.

2. In the Welcome dialog box, click Create a new report or template.
 3. In the New dialog box, click Map and click OK.
 4. In the Choose Map dialog box, in the Maps pane, expand the World folder and click World.
 5. In the Region layers box, click Countries + Territories.
 6. In the Point layers box, click None.
 7. In the Display layers box, click Oceans.
- Tip:** You can select or deselect multiple display layers by Ctrl+clicking.
- Tip:** You can return to the Choose Map dialog box at any time by double-clicking the map background.
8. Drag the following data items to the map:
 - Revenue (in Sales fact) to the Color drop zone.
 - Retailer country (in Retailer site) to the Location drop zone.
 9. Run the report.



Set Map Properties

When you select an element in a map, you can view its properties in the **Properties** pane. Some properties are dependent on the existence of other properties.

If you are familiar with using conditional variables, you can customize the map to change appearance or provide information in response to expressions or conditions.

When you change a map property, you usually do not see the change until you run the report. Changes to the properties of labels and titles are reflected immediately.

These are some of the properties you can change in maps. These properties are available when you select the map object, unless specified otherwise in the **Action to perform in the Properties pane** column.

Goal	Action to perform in the Properties pane
Hide or show the title, subtitle, footer, or axis title	Under Chart Titles , set the Title , Subtitle , Footer , or Axis Title property.
Hide or show the legend	Under Chart Annotations , set the Legend property.
Hide or show map labels	Select the region or point layer. Under Chart Labels , set the Labels property.
Hide or show values on the map	Select the region or points layer. Under Chart Labels , set the Values property.
Hide or show the border around the legend	Select the legend icon. Under Box , set the Borders property.
Change the border around the map object	Under Box , set the Border property.
Hide or show the tooltips	Under Chart Labels , set the Tooltips property.
Note: Some versions of Acrobat Reader do not support tooltips.	
Change the amount of white space around the map	Under Box , set the Padding or Margin property.
Change the default colors for all map elements	Under Color & Background , set the Background Color , Foreground Color , or Fill Effects property.
Change the font and the alignment of text	Under Font & Text , set the Font or Relative Alignment property.
	Tip: The 2005 Quarterly Sales Forecast sample report in the GO Sales (analysis) package includes text alignment. For more information about The Great Outdoors Company samples, see " Sample Reports and Packages " (p. 567).
Resize the map	Under Positioning , set the Size & Overflow property.
Change the font for the legend	Select the legend icon. Under Font & Text , set the Font property.
Change the format of values in the legend	Select the value in the region or point layer. Under Data , set the Data Format property.
Ignore data with no features	Under Data , set the Ignore Data with No Features property.

Goal	Action to perform in the Properties pane
Specify the size of points	In the point layer, click the measure in the Size drop zone and set the Minimum Size and Maximum Size properties.
Add titles, subtitles, footers, or axis titles	Set the Title , Subtitle , Footer , or Axis Title property.

Steps

1. Select the map object or map element to change:
 - To change general properties, such as size and color, click the map object.
 - To change specific map elements, such as a layer or title, click the element itself.

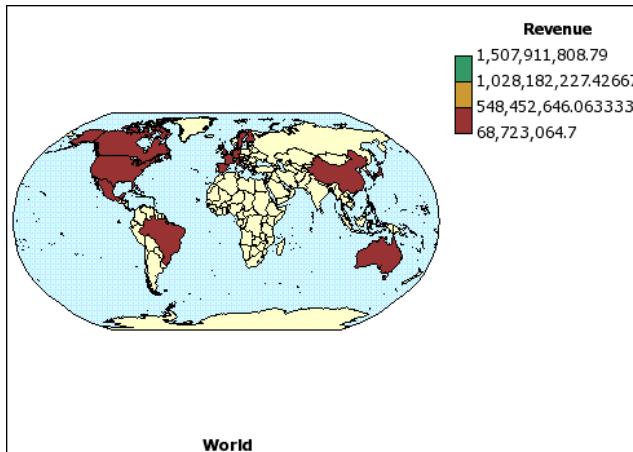
Tip: To cancel a selection, press the Esc key.
2. In the **Properties** pane, set the property value.
An ellipsis (...) button indicates that a dialog box provides further options.
You may have to scroll to see all the properties.

Example - Define Data Values for the Region Layer

The map that you created in the previous topic is not yet linked to a data source. You will now specify the data values from your data source that will determine the color of each region.

Steps

1. In the **Insertable Objects** pane, expand **Sales and Marketing (query)**, **Sales (query)**, and **Sales fact**.
2. Drag **Revenue** to the **Color** drop zone.
3. Expand **Employee by organization**.
4. Drag **Country** to the **Location** drop zone.
5. Run the report.



Match Data Values to Names in the Map File

If the **Ignore Data with No Features** property is set to **No**, then each object that is called from the data source must have a matching label in the specified layer of the map file. For example, if your data source has a country named United States and the layer in the map file labels the same country USA, then there is a mismatch that must be corrected. IBM® Cognos® Report Studio only makes you aware of a mismatch if each object in your data source does not have a corresponding label in the map file. If there are extra labels in the map file that do not have a match in the data source, the report will run without an error message.

A mismatch between your data and the map file must be corrected by the report author. It can not be corrected by a consumer of the map report at run time. There are two ways to correct a mismatch between your data and the labels in the map files. You can use IBM Cognos Map Manager to edit the labels in the layers of the map file, or you can use the dictionary property to create an alias for each mismatched object. When you use the dictionary property, it resolves the mismatch only for a single report, and is not shared with other reports. If you intend to continue using a map with the same data source, it is best to edit the map in Map Manager so that the labels match the objects in your data source.

For information about using Map Manager, see the *Map Manager Installation and User Guide*.

Steps to Create an Alias

1. Select the map object.

The title bar of the **Properties** pane now shows the word **Map**.

2. In the **General** section of the **Properties** pane, double-click the **Dictionary** property.

3. Click the new button .

4. In the **Dictionary Entry** dialog box, click **Search**.

5. In the **Search string** box, type a word or part of a word for which to search.

For example, if you are searching for United States, type in part or all of the name.

6. In the **Search map layer** box, click the layer to search and click **Search**.

7. In the **Matching features** box, click the label to which to match your data source and click **OK**.
8. In the **Alias** box, type the name as it appears in the data source and click **OK**.

For example, if the country in your data source is named USA, type USA as the alias.

To find out the name for the objects in your data source, run a list report. For example, you can run a list report to show the names of all the countries in your data source. For more information see "[Lists](#)" (p. 111).

Define Data Values for the Point Layer

The Point layer in a map is used to visually represent data for point locations, such as cities or sales outlets. Both the color and size of points can be based on data from your data source.

Steps to Set the Color and Size of Points

1. Open the **Choose Map** dialog box:
 - When you create a new map report, this dialog box appears automatically.
 - If you are already in a map report, double-click the map background.
2. In the **Point Layers** box, select the layer containing the points to show on the map.
For example, on a map of the world, you may want the points to represent cities.
3. In the **Insertable Objects** pane, drag an object to the **Color** drop zone in the **Point Layer**.
For example, to have the color of the point based on revenue, drag **Revenue** from the **Insertable Objects** pane to the **Color** drop zone.
4. In the **Insertable Objects** pane, drag an object to the **Size** drop zone in the **Point Layer**.

Steps to Set the Location of Points

1. In the **Insertable Objects** pane, drag an object to the **Location** drop zone in the **Point Layer**.
The object must be supported in the map file as a point location. For example, in the **World** sample map, city is supported as a point location but country is not.
2. If you need to refine the location, drag an object to the **Refine Location** drop zone.
Use this drop zone when there is more than one location with the same name. For example, if you try to run a report with cities in the point layer, and there is more than one city in your data source with the same name, the report does not run. An error message indicates that there are cities in your data source with duplicate names. You can differentiate the cities by using the data object **Region** to refine the location.

Add Colors to the Region or Point Layer

You can add colors for regions or points and specify values to determine when those colors are shown.

Steps

1. In the report, click the **Region Layer** or **Point Layer**.
2. In the **Color & Background** section of the **Properties** pane, double-click the **Palette** property.
3. Click the new button  and click **Color**.
A new color is added to the list of colors.
4. To view the palette colors as a continuous spectrum in which colors blend into one another, click **Continuous Colors**.
5. With the new color selected, click **Color** in the right pane of the dialog box and select a color.
6. Change the percentage boundaries for the colors.

Tip: To specify absolute values rather than percentages, clear the **Percentage** check box.

Add a Legend Title

There are legend titles for the entire legend, for the color of the regions, for the color of the points, and for the size of the points.

Steps to Change the Legend Title for the Entire Map

1. If the legend title is not showing, click the legend icon .
2. In the **General** section of the **Properties** pane, set the **Legend Title** to **Show**.
3. In the report, double-click the legend title and type the title.

Step to Change the Legend Title for the Region Color, Point Color, or Point Size

- By default the legend titles are taken from the object you have selected from the data source. To change a title, do one of the following:
 - In the **Insertable Objects** pane, drag a text or calculation object to the **Color Legend Title** drop zone in the Region Layer or Point Layer, or the **Size Legend Title** drop zone in the Point Layer.
 - Double-click the **Color Legend Title** or **Size Legend Title** drop zone, to change from the default legend title, then double-click the drop zone again. In the **Text** dialog box, type text for the legend title.

Add a Note to a Map

You can add one or more notes, determine their position in the map report, and specify borders around notes.

Steps to Add a Note

1. Select the map object.
2. In the Chart Annotations section of the Properties pane, double-click the Notes property.
3. Click the new button , and click OK twice.
A note icon with the words New Note appears in the report.
4. Click New Note next to the note icon .
5. In the Text Source section of the Properties pane, double-click the Text property.
6. Type the text to appear in the note and click OK.

Steps to Position a Note and Add a Border

1. In the report, click the note icon .
2. In the Positioning section of the Properties pane, type values to specify the bottom position, left position, height, and width for the note.
The location of the note is defined by the number of pixels.
3. Set the Note Border property to specify a border around the note.
4. Run the report to view the note.
If necessary, change the position again.

Drill Through to Another Report From a Map

You can link regions or points on a map to another report. For example, on a map of the world, you can specify that when you click on China, a map of China opens.

Steps to Set Up a Filter in the Target Report

1. Open the target report.
2. From the Data menu, click Filters.
3. On the Detail Filters tab, click the add button .
4. In the Available Components box, click the Source or Data Items tab to select the data item to use.

For example, to open the target report when Canada is clicked in the source report, expand Countries and double-click Country.

5. In the **Expression Definition** box, type an operator after the data item or select an operator from the **Functions** tab and then enter a value.

For example, to open the report when Canada is clicked in the source report, the expression would be as follows:

[Country]='Canada', where [Country] is the name of the data item in the package.

6. Save the target report.

Steps to Set Up a Drill-through Link in the Source Report

1. Open the source report.
2. Select the region layer or the point layer.
3. In the **Data** section of the **Properties** pane, double-click the **Map Drills** property.
4. In the **Map Drills** dialog box, click the new button .
5. In the **Search string** box, type the name of the feature to use for the drill-through link.

For example, to open a report when you click on Canada in the map, search on all or part of the word **Canada**.
6. Click **Starts with** to search for features that start with the search string or click **Contains** to search for feature names that include the search string.
7. To include a parent layer's name in the results of your search, select a layer in the **Include parent layer** list.

For example, if you searched for **Oslo** and you included the parent layer of **Countries + Territories**, the search result would be **Oslo (Norway)**.
8. Click **Search**.
9. Select a feature from **Matching features** box and click **OK**
10. In the **Drill-Through Definitions** dialog box, click the new button.
11. Click the ellipsis (...) button beside the **Report** box, and select the target report.
12. Run the report.

When you click the selected feature in the source report, the target report will open.

Edit a Map

Administrators and modelers use a Microsoft® Windows® operating system utility named Map Manager to import maps and update labels for maps in Report Studio. For map features such as country and city names, administrators and modelers can define alternative names to provide multilingual versions of text that appears on the map.

With IBM® Cognos® Map Manager you can edit the labels in maps to be consistent with the object names in your database.

For instructions, see the Map Manager *Installation and User Guide*.

Additional Maps

IBM® Cognos® supplies a set of standard maps that can be used directly with IBM Cognos Report Studio as well as IBM Cognos Map Manager.

If you already have proprietary geographic data, you can also create your own custom .GST files and then import these into Map Manager. To create custom .GST files, you must use an application that produces MapInfo files, such as MapInfo Professional.

Location Intelligence

Sophisticated mapping functionality, known as location intelligence, can be used for a broad range of business applications that extend the mapping capability of IBM® Cognos® BI. MapInfo provides solutions that can be directly integrated with IBM Cognos BI. These include the ability to dynamically create geographic filters and custom areas for aggregating data for ad-hoc analysis. Examples of business applications of location intelligence are listed in the following table.

Business application	Benefits
Target marketing	Learn who your best clients are and find more like them.
Network optimization and site location analysis	Put stores near your customers and look for gaps in geographical coverage.
Routing and work force optimization	Reduce the number of trucks you need and make your drivers more efficient.
e-government	Provide citizens with self-service opportunities.
Sales territory creation	Create balanced sales territories.
Economic development	Plan the development of your community.
Communications network planning	Avoid costly mistakes by putting cell towers in the right locations. Identify the locations of clients in your service area.

You can contact MapInfo for both data and location intelligence solutions through their Web site: www.mapinfo.com.

Chapter 8: Active Reports

You can use IBM® Cognos® Report Studio to create active reports. IBM Cognos Active Report is a report output type that provides a highly interactive and easy-to-use managed report. Active reports are built for business users, allowing them to explore their data and derive additional insight.

Active reports make business intelligence easier for the casual user. Report authors build reports targeted at their users' needs, keeping the user experience simple and engaging. Active reports can be consumed by users who are offline, making them an ideal solution for remote users such as the sales force.

Active reports are an extension of the traditional IBM Cognos report. You can leverage existing reports and convert them to active reports by adding interactive behavior, providing end users with an easy-to-consume interface.

Like existing IBM Cognos reports, you can execute active reports from IBM Cognos Connection as well as schedule and burst them to users.

You build active reports with the same objects that you use to build other report types. However, there are objects that are specific to active reports. These objects fall into two categories:

- [active report controls](#)
- [active report variables](#)

For tips about building active reports, see the Proven Practices section of the Cognos Customer Center (<http://www-01.ibm.com/software/data/cognos/customercenter>).

Active Report Controls

You use active report controls to create the layout of an active report as well as filter and sort data in the report.

Layout

The following controls are used to build the layout of an active report:

- Tab controls, which are used for grouping similar report items.
- Decks of cards, which are used for layering report items.
- Hiding or showing list columns, which allows users to control the data they see by using check boxes.

Filtering and Sorting

To help report authors deliver the content in the most consumable way possible, IBM® Cognos® Report Studio provides several new filtering controls:

- List and drop-down list controls

- Interactions with charts
For example, clicking a category in a chart filters the data in a list.
- Radial buttons
- Check boxes
- Toggle buttons
- Push button controls

Data Containers

Data containers, such as lists and crosstabs, are also considered controls in an active report, as you can add interactive behavior to them.

Active Report Variables

Active report variables work in conjunction with active report controls to add interactivity to a report. Actions performed on a control, such as selecting an item in a control or selecting a control itself, can set the value of a variable. In turn, controls can respond to changes in the value of a variable, such as filtering data in a control.

Convert an Existing Report to an Active Report

You can leverage existing reports by converting them to active reports.

Steps

1. Open the report that you want to convert.
2. From the File menu, click **Convert to Active Report**.

IBM® Cognos® Active Report menu items, objects, and properties become available in IBM Cognos Report Studio. Objects in the report that are not supported in active reports, such as prompt controls in report pages, are removed.

Specify Active Report Properties

You can change the default properties of an active report.

Steps

1. From the File menu, click **Active Report Properties**.
2. To specify the title that will appear in the browser window title bar when the report is viewed, under **Window Title**, type the title that you want.
3. If you want to specify the window title in other languages, do the following:
 - Under **Window Title**, click the ellipsis button (...).

- Click the add button .
 - Select the languages that you want.
 - In the **Language** column, click one of the languages that you selected.
 - Click the edit button  and type the window title for that language.
4. If you want to show an icon beside the window title when a report is viewed, in the **Window Icon** box, type the URL of the icon that you want.
 5. In the **Window Startup Size** box, choose among the available options to specify the size of the browser window when an active report is viewed.
 6. In the **Maximum query rows** box, specify the maximum number of rows of data that the report can contain.
The value that you specify will influence the performance when users interact with the report.
 7. To reuse the **Window Startup Size** and **Maximum query rows** properties when you create other active reports, select the **Use as default for new active reports** check box.

Adding Interactivity to a Report

You create an active report by adding interactive behavior to controls in the report. Adding interactive behavior links controls to each other, allowing an action performed on a control to change the state of another control.

You add interactivity to a report by:

- [creating and managing active report variables](#)
- [adding controls](#)
- [adding data to controls](#)
- [defining a connection between controls](#)
- [specifying the selection behavior of controls](#)
- [specifying the reaction behavior of controls](#)

Create and Manage Active Report Variables

Create active report variables to allow controls in a report to interact with each other. Active report variables pass information from one control to another control.

IBM® Cognos® Report Studio automatically creates active report variables when you define connections between controls. However, you may want to make changes to these variables, such as

changing the name or defining default values. For more information about defining connections between controls, see ["Define a Connection Between Controls" \(p. 221\)](#).

Steps

1. From the Tools menu, click **Manage Active Report Variables**.
2. To create a new variable, do the following:
 - Under the **Active Report Variables** box, click the new button  and specify the name of the variable.
 - If you want to specify a default value for the variable, under the **Default Variable Values** box, click the new button  and define the value.

Tip: You can define more than one default value for a variable. Define multiple default values when you want more than one item selected in a control that allows multiple selections, such as a check box group.
3. To modify an existing variable or its values, under **Active Report Variables** or **Default Variable Values**, click the edit button  and make the changes that you want.

Tip: After you add interactivity to a report, the **Variable References** box shows how variables are used in the report.

Adding Controls to a Report

IBM® Cognos® Active Report uses the following active report controls. The controls are available on the **Toolbox** tab . For many controls, two different versions are available, a static version and a data-driven version. Use the static version of a control when you want to manually define the values that the control can contain. Use the data-driven version of a control when you want the values that the control can contain to come from a data item.

Tip: To view only toolbox items that are specific to active reports, right-click anywhere in the **Toolbox** tab and click **Active Report Toolbox Items**.

Variable Text Items

Use variable text items to insert active report variables in a report. Variable text items are useful to show the value of a variable when an item in another control is selected. For example, you can use a variable text item as a title for a list that is linked to a drop-down list control. When an item is selected from the drop-down list, the item appears as the title of the list.

Decks and Data Decks

Use decks and data decks to show different objects and different data respectively based on a selection in another control. For example, clicking a radio button in a radio button group control shows a list object while clicking a different radio button shows a chart object.

Decks are composed of cards. In static decks, you define the number of cards in the deck and you insert the objects that you want in each card. This allows you to create cards that can contain different objects, such as pie chart in one card and a crosstab in another card. In data decks, the cards

are defined by a data container or data items inserted in the deck, and a data item determines which card appears.

Tab Controls and Data Tab Controls

Use tab controls and data tab controls to define multiple pages for the same area of a report. In data tab controls, the tabs are driven by a data item that you insert in the control.

Button Bars and Data Button Bars

Use button bars and data button bars to add a group of push buttons. In data button bars, the buttons are driven by a data item that you insert in the control.

In reports, users can click only one button at a time.

Toggle Button Bars and Data Toggle Button Bars

Use toggle button bars and data toggle button bars to add a group of buttons that change appearance when pressed. In data toggle button bars, the buttons are driven by a data item that you insert in the control.

In reports, users can click one or more buttons simultaneously.

Radio Button Groups and Data Radio Button Groups

Use radio button groups and data radio button groups to group a set of buttons that have a common purpose. In data radio button groups, the radio buttons are driven by a data item that you insert in the control.

In reports, users can click only one radio button at a time.

Check Box Groups and Data Check Box Groups

Use check box groups and data check box groups to group a set of check boxes. In data check box groups, the check boxes are driven by a data item that you insert in the control.

In reports, users can select one or more check boxes simultaneously.

Drop-Down Lists and Data Drop-Down Lists

Use drop-down lists and data drop-down lists to provide a list of items that users can choose from. In data drop-down lists, the lists are driven by a data item that you insert in the control.

In reports, users can select only one item at a time.

List Boxes and Data List Boxes

Use list boxes and data list boxes to provide a list of items that users can choose from. In data list boxes, the lists are driven by a data item that you insert in the control.

In reports, users can select one or more items in a list box.

Buttons

Use buttons to add individual push buttons to a report.

Data Containers

Lists, crosstabs, charts, and maps are also controls in an active report, as you can add interactive behavior to them.

Add Data to a Control

For every control that you insert in a report, you must add the data that you want to appear in the control. For data-driven controls, you insert data items from the package. For static value driven controls, you manually define the data in a data table.

Steps to Add Data to a Data-Driven Control

1. In the Insertable Objects pane, on the Source tab , drag query subjects or query items to the control.
2. To add an image to the data in the control, drag the appropriate data item to the **Icon** box.
3. If the data that will appear in the control depends on other data that you do not want to show, drag the appropriate data item to the **Values** box.

For example, you add Product type to a data list box control and you want to filter the product types that appear in the control based on a product line selected in another control. To filter the data, you must drag Product line to the **Values** box.

Tip: IBM® Cognos® Report Studio can automatically copy data items from other controls when you create connections between controls. For more information, see "[Define a Connection Between Controls](#)" (p. 221).

4. If the control is a data deck and you want to create a default card that will appear when no other card matches the variable value passed to the deck, in the **Properties** pane, click Yes for the **Default Card** property. Then click the No Data Contents tab  for the deck and insert the objects that you want to appear on the default card.

Steps to Add Data to a Static Control

1. In the report, click the definition icon  of the control.

Tip: The definition icon is visible only when the visual aid **Show Active Object Controls** is enabled. For more information, see "[Visual Aids Button](#)" (p. 78).

2. To add a new object, such as a card in a deck or a button in a button bar, under **Data Table**, click the new button.
3. In **Data Table**, define the values that you want to appear in the control.

Values are organized by data item. In addition to the default data item (Label) and values provided, you can create your own data items and values. For example, to add an icon to each value, you must create a data item for the icons.

- To define a new data item, click the new button  and type the name of the data item that you want to create.
- Click inside each table row and type the name of the value.

- To add translations for the values, to support users in multiple languages, click the ellipsis (...) button, choose the languages that you want to support by clicking the add button , and type the translated text for each language added.
 - If you created new data items, in the **Definition** box, click the **Label** drop-down list and select the data item that you want to use in the control.
 - To add an icon to each value, select the **Icon** check box, click the drop-down list, and select the data item that contains the icons that you want to use.
4. If the control is a deck, click the left and right arrows to navigate to each card of the deck and insert the objects that you want to appear on each card.

To create a default card that will appear when no other card matches the variable value passed to the deck, in the **Properties** pane, click **Yes** for the **Default Card** property. Then click the **No Data Contents** tab  for the deck and insert the objects that you want to appear on the default card.

Define a Connection Between Controls

Define a connection between controls to link them together. When you link controls, an action performed on one control affects the behavior of the other control.

When you define a connection between controls, IBM® Cognos® Report Studio creates the active report variables required for the connection. Alternatively, you can define connections between controls by creating your own variables and using them to specify the selection behavior and the reaction behavior of the controls. For more information about creating variables, see "[Create and Manage Active Report Variables](#)" (p. 217). For more information about specifying the selection behavior of a control, see "[Specify the Selection Behavior of a Control](#)" (p. 222). For information about specifying the reaction behavior of a control, see "[Specify the Reaction Behavior of a Control](#)" (p. 223).

Steps

1. Click the interactive behavior icon  of a control that you added to the report.
Tip: The interactive behavior icon is visible only when the visual aid **Show Active Object Controls** is enabled. For more information, see "[Visual Aids Button](#)" (p. 78).
2. In the **Control** drop-down list, select the control for which you want to define a connection.
3. Click **Create a New Connection**.
4. In the top left drop-down list, select the control that you want to specify as the source.
5. In the top right drop-down list, select the control that you want to specify as the target.
6. Click the **Connection Method** drop-down list between the source and target controls and specify the type of relationship that you want to define between the two controls:
 - To filter data in the target control based on what will be selected in the source control, select **Filter**.

- To select a target control based on what will be selected in the source control, click **Select**.
7. In the **Active Report Variable** section, choose to create a new variable for the connection or use an existing variable.
- Tip:** If you create a new variable, you can change the default name generated by Report Studio.
8. In the **Data Item** drop-down list at the bottom left of the dialog box, select the data item that will drive the behavior of the target control.
9. In the **Data Item** drop-down list at the bottom right of the dialog box, select the item that will determine what appears in the target control.
- If the data item that you want to use is not in the query of the target control, select **Copy Data Item from Source** to copy the source data item specified in the previous step to the target control query. If the target control is a data container, the copied data item appears shaded in the container to indicate that the column visibility property for the item is set to **Hidden**. For more information about the column visibility property, see "[Show or Hide a Column or Row](#)" (p. 225).
10. Click **Connect**.
11. To create or manage other active report variables, click **Manage Active Report Variables**.

When a connection is defined, Report Studio creates the appropriate conditions for the selection behavior and reaction behavior of the source and target controls.

Tip: The interactive behavior icon  of a control changes to indicate that a connection was defined.

Specify the Selection Behavior of a Control

Specify the selection behavior of a control to determine what interactive behavior should occur when the control is clicked or when an object in the control, such as a button, check box, or data item, is selected. Specifying the selection behavior of a control sets the value of active report variables defined in the report. The active report variable values are then used to specify the reaction behavior of controls. For more information about specifying the reaction behavior of controls, see "[Specify the Reaction Behavior of a Control](#)" (p. 223).

For example, you add two drop-down list controls to a report and you create a connection between them. The first drop-down list shows product lines and the second drop-down list shows product types. When a product line is selected in the first drop-down list box, you want to pass that information to the second drop-down list box so that it shows only the product types of the selected product line.

You can specify a selection behavior for any control, except for decks and data decks.

IBM® Cognos® Report Studio automatically sets active report variable values when you define a connection between controls. However, you may need to set the value of other variables to specify additional interactive behavior. For more information about defining a connection between controls, see "[Define a Connection Between Controls](#)" (p. 221).

Steps

1. Click the interactive behavior icon  of the source control.

Tip: The interactive behavior icon is visible only when the visual aid **Show Active Object Controls** is enabled. For more information, see "["Visual Aids Button" \(p. 78\)](#)".

2. In the **Behavior on Select** box, to the right of **Set Variable Values**, click the edit button .
3. To create or manage active report variables, click **Manage Active Report Variables**. For more information about creating or managing active report variables, see "["Create and Manage Active Report Variables" \(p. 217\)](#)".
4. For each active report variable that you want to set as part of the selection behavior of the control, click the **Value** drop-down list and select one of the following choices:
 - If you want to clear the current value of the variable, click **Set to Empty**.

Tip: Clearing the current value of a variable is useful when you have cascading controls. For example, if the parent control is reset to a default value, you want to reset the child control to a default value as well.
 - If you want to specify your own value, click **Type in a value** and type the value that you want to use.
 - If you want the variable values to be driven by a data item in the report, click the data item that you want to use.
 - If you do not want the variable to be used for the control, click **Do not set**.

Specify the Reaction Behavior of a Control

Specify the reaction behavior of a control to determine how a control will react when another control, or an object in the control, is selected. You can specify the following reaction behaviors:

- Select an item in a control.

For example, you specify a default value for a variable, and the variable drives a drop-down list control. When the report is run, you want the default value to appear in the control.

You can specify this reaction behavior for any control, except for single button controls.

- Filter data in a control.

For example, you want to filter the product types in a drop-down list control based on the product line selected in another drop-down list control.

You can specify this reaction behavior for drop-down lists, data drop-down lists, list boxes, data list boxes, radio button groups, data radio button groups, check box groups, data check box groups, lists, and crosstabs.

- Enable a control.

For example, you want to make a control interactive only when a selection is first made in another control.

You can specify this reaction behavior for any control, except for decks, data decks, tabs, data tabs, lists, crosstabs, charts, and maps.

In addition to the above behaviors, you can specify the item to appear when nothing is selected in a list control, hide or show a column, and whether to allow data to be sorted in a list. For more information about specifying the item to appear when nothing is selected in a list control, see "[Set the No Value List Item](#)" (p. 225). For more information about hiding or showing a column, see "[Show or Hide a Column or Row](#)" (p. 225). For more information about sorting data in a list, see "[Sort Data in a List](#)" (p. 226).

Steps

1. Click the interactive behavior icon  of the control.
 2. In the **Reaction Behavior** box, to the right of the reaction behavior that you want to specify, click the edit button .

If a condition has not yet been defined, two dialog boxes appear. The second dialog box prompts you to define the condition expression that you want to use to specify the reaction behavior.

 3. In the **Item** box on the left, define the left side of the condition expression.
 - To use a data item, click the **Data Item** radio button and then click the item that you want to use.

The data items that are available in the drop-down list depend on the data items that you inserted into the control.
 - To use a static item, click the **Value** radio button and type the value that you want to use. The value must be defined as a value of the selected active report variable in the next step.
 - If you want to set the value of an active report variable to null, click the **Empty Variable** radio button.
 4. In the **Item** box on the right, define the right side of the condition expression.
 - To use a value in an active report variable, click the **Active Report Variable** radio button and then click the variable that you want to use.

The variables that are available in the drop-down list depend on the variables that you defined in the report.
 - To use a static item, click the **Value** radio button and type the value that you want to use.
 - To use null as the value, click the **Empty Data Item** radio button.

Use null as the value to specify the reaction behavior when no data exists for the value selected in the source control.
 5. If you want to apply the condition when no item is selected in the source control, clear the **Drop this condition if the variable is empty** check box.
- Clearing the check box can improve performance and prompt users to perform an action. For example, a report contains a list with many rows of data, and the list is filtered by another control. When the check box is cleared, no data will appear in the list until an item is selected in the other control.

Set the No Value List Item

Set the no value list item to show a specific item in a drop-down list or list box control when no value is selected. For example, a report contains a drop-down list control that contains product lines. You create a no value list item named (All) to show all product lines when the report is initially viewed.

Steps

1. Select the control.
2. In the **Properties** pane, for the **No Value List Item** property, click **Show**.
A text item representing the no value list item appears in the control.
3. In the **Properties** pane, double-click the **Label** property.
4. In the **Default text** box, type the name of the no value list item.
5. To specify the text for other languages, do the following:
 - Click the add button .
 - Select the languages that you want.
 - In the **Language** column, click one of the languages that you selected.
 - Click the edit button  and type the name of the no value list item for that language.
 - Repeat for all remaining languages.
6. In the **Properties** pane, click the **Position in List** property and specify where you want the no value list item to appear in the control.

Show or Hide a Column or Row

You can show or hide a column in a list or a column or row in a crosstab when the report is viewed.

Steps

1. Click the title of the column or row that you want to show or hide.
2. If the data container is a crosstab, in the **Properties** pane, click the **Visibility** property and click **Yes** or **No**.
3. If the data container is a list, in the **Properties** pane, double-click the **Column Visibility** property.
4. To make the list column visible when the report is viewed, click the **Visible** radio button.
5. To hide the list column when the report is viewed, click the **Hidden** radio button.
6. To make the list column visible based on a condition, click the **Visible based on condition** radio button, click the edit condition button , and specify the condition.
 - In the left **Item** box, type the value that you want to use for the condition.

If you want to use null as the value for the condition, click the **Empty Variable** radio button instead.

- Click the **Operator** drop-down list and select the operator that you want to use for the condition.
- In the right **Item** box, select the active report variable that you want to use for the condition.
- If you want to apply the condition when no item is selected in a control that is connected to the list, clear the **Drop this condition if the variable is empty** check box.

Sort Data in a List

You can specify whether users can sort data in a list in the report output.

Steps

1. Select the column that you want users to be able to sort.
2. In the Properties pane, for the **Allow Sorting** property, click **Yes**.

Summarize Data in the Active Report Output

You can add an output summary to a list or crosstab control that reflects the data that appears in the control when an active report is viewed. For example, the data in a list is driven by selecting an item in a data drop-down list control. An output summary specified for the list is recalculated when a different item is selected in the drop-down list.

Steps

1. In the list or crosstab, click the column or row for which you want to add an output summary.
2. In the toolbar, click the summarize button  and then, under **Summarize in Output**, choose the summary that you want to add the active report output.

Run or Save an Active Report

You can run an active report in HTML format or save it as a MIME HTML (MHT) file so that you can send the report to your consumers. The MHT file is viewable in the Microsoft® Internet Explorer or Mozilla Firefox browsers.

To assist you in resolving problems when you are authoring an active report, you can select a run option that shows the state of active report variables in the report as you click various controls.

If you are using Microsoft Internet Explorer 6.0, you cannot open an active report in MHT format as a file. You can view the report only by using a URL.

To view an active report in MHT format in Mozilla Firefox, you must first download an UnMHT add-on.

Steps

1. To show the state of active report variables when the report is viewed, from the **Run** menu, click **Run Options**, and select the **Enable right-click debug menu** check box.
2. To run a report, from the **Run** menu, click **Run Active Report**.
3. To save a report, from the **Run** menu, click **Download Active Report**, and when prompted, choose to save the report as an MHT file.

When running a report with the **Enable right-click debug menu** check box selected, right-clicking in the report output window allows you to choose to view the variables in the report and to clear the variable values.

Example - Create an Active Report Showing Revenue Data for Core Products

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create an active report showing revenue data for the core products Camping Equipment and Golf Equipment. The revenue data is specific to countries in the Americas. The report will allow users to filter data by product or by country.

To create this report, you will need to add two data drop-down list controls and a list container. The first data drop-down list control will contain the core products that users can select. The second data drop-down list control will contain countries that users can select. The list will show revenue data for each core product and country. When users click different products or countries, the data in the list will be filtered based on the selections made.

Steps to Create the Active Report

1. Open IBM® Cognos® Report Studio with the GO Data Warehouse (analysis) package.
2. In the Welcome page, click **Create new**.
3. In the New dialog box, click **Active Report**.
4. Double-click the report title and type
Core Products Revenue
5. In the Insertable Objects pane, on the **Toolbox** tab , drag the **Table** object to the report.
6. Clear the **Maximize width** check box and then click **OK** to create a table with two columns and one row.
7. Drag the **Data Drop-Down List** object to each cell in the table.

8. Drag the List object to the report so that it appears under the table.

Steps to Add Data to the First Data Drop-Down List Control

1. Right-click the first data drop-down list control and click Go to Query.
Query1 opens in the Query Explorer.
 2. In the Insertable Objects pane, on the Source tab , do the following:
 - Expand the Sales and Marketing (analysis) folder and the Sales namespace.
 - Expand the Products dimension and then the Products hierarchy.
 - Drag Product line to the Data Items area.
 3. Click Product line and in the Properties pane, do the following:
 - Click the Name property, delete the existing name, and type
Core products
 - Double-click the Set Definition property.
 - Click the new button  and then click **Exclude**.
 - In the Available members box, expand Sales and Marketing (analysis), Sales, and Products.
 - Drag the Mountaineering Equipment, Personal Accessories, and Outdoor Protection members to the Members box.
- The Core products member set now contains only Camping Equipment and Golf Equipment.
4. Click OK twice.
 5. Pause the pointer over the page explorer button  and click Page1.
 6. In the Insertable Objects pane, on the Data Items tab , drag Core products from Query1 to the drop-down list box in the control.
 7. Click the control and in the Properties pane, for the No Value List Item property, click Show.
The Properties pane for the no value list item property appears.
 8. Double-click the Label property, type Core products in the Default text box, and click OK.

Steps to Add Data to the Second Data Drop-Down List Control

1. Right-click the second data drop-down list control and click Go to Query.
Query2 opens in the Query Explorer.
2. In the Insertable Objects pane, on the Toolbox tab , drag Data Item to the Data Items area.
3. In the Expression Definition box, type
`children ([Sales].[Retailers].[Retailers].[Region]->[Retailers].[710])`

Tip: This expression returns the children of the **Americas** member. The MUN for Americas is used instead of the member name.

4. In the **Properties** pane, click the **Name** property, delete the name and type **Countries**
5. Pause the pointer over the page explorer button  and click **Page1**.
6. In the **Insertable Objects** pane, on the **Data Items** tab , drag **Countries** from **Query2** to the drop-down list box in the control.
7. Click the control and in the **Properties** pane, for the **No Value List Item** property, click **Show**. The **Properties** pane for the no value list item appears.
8. Double-click the **Label** property, type **Countries** in the **Default text** box, and click **OK**.

Steps to Add Data to the List

1. Pause the pointer over the query explorer button  and click **Query1**.
2. Copy the **Core products** data item.
3. Pause the pointer over the query explorer button  and click **Query3**.
4. Paste the **Core products** data item into **Query3**.
5. Repeat steps 1 to 4 to copy the **Countries** data item from **Query2** to **Query3**.
6. Pause the pointer over the page explorer button  and click **Page1**.
7. In the **Insertable Objects** pane, on the **Data Items** tab , drag **Core products** and **Countries** from **Query3** to the list.
8. In the **Insertable Objects** pane, on the **Source** tab , do the following:
 - Expand the **Products** dimension and then the **Products** hierarchy.
 - Drag **Product type** to the list, inserting it between **Core products** and **Countries**.
 - Expand **Sales fact**.
 - Drag **Revenue** and **Planned revenue** to the list, to the right of **Countries**.

Steps to Add Interactivity to the Report

1. Click the interactive behavior icon  of the first data drop-down list control.
2. Click **Create a New Connection**.
3. Click the upper right drop-down list and click **List1**.
4. Ensure that the following information appears in the dialog box and click **Connect**.
 - In the **Connection Method** drop-down list, **Filter** is selected.

- In the Data Item drop-down list for both controls, Core products is selected.
5. In the Control drop-down list, click Data Drop-Down List2.
 6. Click Create a New Connection.
 7. Click the upper right drop-down list and click List1.
 8. Ensure that the following information appears in the dialog box and click Connect.
 - In the Connection Method drop-down list, Filter is selected.
 - In the Data Item drop-down list for both controls, Countries is selected.
 9. In the Control drop-down list, click Data Drop-Down List1.
 10. In the Behavior on Select section, click the edit button .
 11. Set Countries Variable 1 to Set to Empty and click OK twice.

When users select a different product in the first data drop-down list control, the second data drop-down list value will reset to the no value list item.

Steps to Specify Default Values for the Data Drop-Down List Controls

1. From the Tools menu, click Manage Active Report Variables.
2. In the Active Report Variables box, ensure that Core products Variable 1 is selected.
3. Under the Default Variable Values box, click the new button , type Camping Equipment, and click OK.
4. In the Active Report Variables box, click Countries Variable 1.
5. Under the Default Variable Values box, click the new button , type United States, and click OK twice.

When users either run the report from IBM Cognos Connection or open the report in MHT format, the following data appears in the list. Users can click different products or countries and see the revenue data that they want to see in the list.

Camping Equipment		Core products	Product type	Countries	Revenue	Planned revenue
Camping Equipment	Cooking Gear			United States	43,152,191	47,952,594.23
Camping Equipment	Sleeping Bags			United States	50,338,776.21	54,345,028.43
Camping Equipment	Packs			United States	57,520,074.37	61,360,321.97
Camping Equipment	Tents			United States	90,663,037.65	95,695,990.84
Camping Equipment	Lanterns			United States	21,597,711.83	23,090,549.81

Example - Create an Active Report Showing Revenue Data for New Order Methods

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create an active report showing revenue data for the new order methods Web, E-mail, and Telephone. The report will allow users to filter data by order method.

To create this report, you will need to add a data check box group control and a list container. The data check box group control will contain the new order methods that users can select. When the report is run, users can select one or more order methods, and the data in the list container will be filtered based on the selections made.

Steps to Create the Active Report

1. Open IBM® Cognos® Report Studio with the GO Data Warehouse (analysis) package.
2. In the Welcome page, click **Create new**.
3. In the **New** dialog box, click **Active Report**.
4. Double-click the report title and type
New Order Methods
5. In the **Insertable Objects** pane, on the **Toolbox** tab , drag the **Data Check Box Group** object to the report.
6. Drag the **Block** object under the data check box group control.
7. Drag the **List** object to the report so that it appears under the block.

Steps to Add Data to the Data Check Box Group Control

1. Right-click the data check box group control and click **Go to Query**.
Query1 opens in the Query Explorer.
2. In the **Insertable Objects** pane, on the **Source** tab , do the following:
 - Expand the **Sales and Marketing (analysis)** folder and the **Sales** namespace.
 - Expand the **Order method** dimension and then the **Order method** hierarchy.
 - Drag **Order method type** to the **Data Items** area.
3. Click **Order method type** and in the **Properties** pane, do the following:
 - Click the **Name** property, delete the existing name and type
New order methods
 - Double-click the **Set Definition** property.
 - Click the new button  and then click **Exclude**.

- In the Available members box, expand Sales and Marketing (analysis), Sales, and Order method.
- Drag the Fax, Mail, Sales visit, and Special members to the Members box.

The New order methods member set now contains only Telephone, E-mail, and Web.

4. Click OK twice.
5. Pause the pointer over the page explorer button  and click Page1.
6. In the Insertable Objects pane, on the Data Items tab , drag New order methods from Query1 to the data check box group control.

Steps to Add Data to the List

1. Pause the pointer over the query explorer button  and click Query1.
2. Copy the New order methods data item.
3. Pause the pointer over the query explorer button  and click Query2.
4. Paste the New order methods data item into Query2.
5. Pause the pointer over the page explorer button  and click Page1.
6. In the Insertable Objects pane, on the Data Items tab , drag New order methods from Query2 to the list.
7. In the Insertable Objects pane, on the Source tab , do the following:
 - Expand the Time dimension and then the Time hierarchy.
 - Drag Year to the list, and drop it to the left of New order methods.
 - Expand Sales fact.
 - Drag Revenue and Gross profit to the list, and drop them to the right of New order methods.

Steps to Add Interactivity to the Report

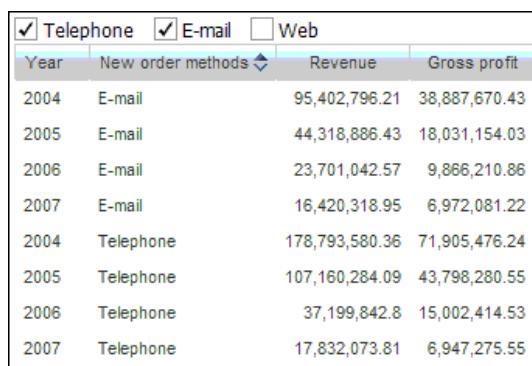
1. Click the interactive behavior icon  of the data check box group control.
2. Click Create a New Connection.
3. Ensure that the following information appears in the dialog box, click Connect, and then click OK.
 - In the top right drop-down list, List1 appears.
 - In the Connection Method drop-down list, Filter is selected.
 - In the Data Item drop-down list for both controls, New order methods is selected.

4. Click the Year column title and in the Properties pane, for the Allow Sorting property, click Yes.
5. Repeat step 4 for the New order methods column.

Steps to Specify Default Values for the Data Check Box Group Control

1. From the Tools menu, click Manage Active Report Variables.
2. In the Active Report Variables box, ensure that New order methods Variable 1 is selected.
3. Under the Default Variable Values, click the new button , type Telephone, and click OK.
4. Repeat step 3 to create a second default value, E-mail.
5. Click OK.

When users either run the report from IBM Cognos Connection or open the report in MHT format, Telephone and E-mail are selected in the data check box group control and the list shows data for these two order methods. Users can select or clear one or more check boxes and see the revenue data that they want to see in the list. In addition, they can click the Year or New order methods column title to sort the data in the list.



<input checked="" type="checkbox"/> Telephone	<input checked="" type="checkbox"/> E-mail	<input type="checkbox"/> Web	Year	New order methods	Revenue	Gross profit
			2004	E-mail	95,402,796.21	38,887,670.43
			2005	E-mail	44,318,886.43	18,031,154.03
			2006	E-mail	23,701,042.57	9,866,210.86
			2007	E-mail	16,420,318.95	6,972,081.22
			2004	Telephone	178,793,580.36	71,905,476.24
			2005	Telephone	107,160,284.09	43,798,280.55
			2006	Telephone	37,199,842.8	15,002,414.53
			2007	Telephone	17,832,073.81	6,947,275.55

Chapter 9: Statistical Analysis

If your installation of IBM® Cognos® Business Intelligence contains IBM Cognos Statistics, and if your administrator has granted you the required capabilities, you can use statistical objects in your reports. For more information, see the *IBM Cognos Statistics Installation and Configuration Guide*.

IBM Cognos Statistics

IBM® Cognos® Business Intelligence integrates statistical objects in IBM Cognos Report Studio. This enables financial and business analysts to gain and share richer insights through statistical reporting and analysis that can be distributed to all users.

You can create multiple statistical objects. After you select a statistical object, you follow the steps in a wizard format and drag items to the appropriate drop zones, adding cases and variables as needed. Not all types of data are suitable for all drop zones: for best results, you should be familiar with your data.

IBM Cognos Statistics covers three main areas:

Distribution of Data

You can use the following to explore the distribution, variability, and percentile information of your data.

- [Descriptive statistics](#)
- [Histograms](#)
- [Boxplots](#)
- [Q-Q Plots](#)

To see how you can view data to learn more about your customers, see "[Example: Create a Report Showing Descriptive Statistics](#)" (p. 245).

Data Analysis and Testing

You can use the following to show how variables are related, or to compare the means between groups of numeric or [categorical](#) data to test a hypothesis.

- [Means comparison](#)
- [Nonparametric tests](#)
- [Correlation and regression](#)

To see how you can test a hypothesis, see "[Example: Using a One-Sample t-Test to Compare Household Income Against a Hypothetical Value](#)" (p. 257).

Statistical Process Control (SPC)

You can use the following to monitor, control, and improve a process through statistical analysis.

- [Control charts](#)
- [Control rules](#)
- [Process capability](#)
- [Process performance](#)

To see how you can help ensure the quality of a manufacturing process, see "[Example: Create a Report Showing Shampoo Manufacturing Quality Control](#)" (p. 295).

The statistical object you need depends on your data and what business question you want to answer.

Cases and IBM Cognos Statistics

To create a unique identifier for the cases variable, you can use an item from the data tree or you can define a query. For more information, see "[Building Cases in IBM Cognos Statistics](#)" (p. 240).

Common Concepts in Statistical Analysis

Here are some common concepts used in statistical analysis.

Samples and Populations

In statistics, a population refers to the entire data set of objects, individuals, observations, and so on, that you have sampled from. A sample is a subset of the population that you work with in your analysis. One goal of statistics is to test if the results of the sample can be generalized to the population.

The Normal Distribution

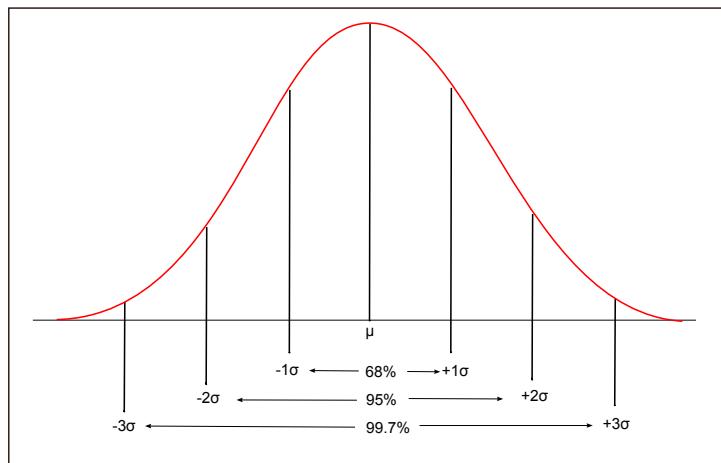
The normal distribution describes a symmetric bell-shaped distribution of values. For example, the distribution of people's IQ scores is bell-shaped and symmetrical around a mean.

Many statistical tests, such as the t-test and one-way ANOVA, assume that the data is normally distributed. If the data is not normally distributed, then use a nonparametric method, such as a chi-square test.

You can check for normal distribution using statistical objects such as [Q-Q plots](#).

Standard Deviation

Standard deviation, often denoted as a sigma or small Greek s, is a measure of variability that describes how data points vary from the mean. The portion of a distribution lying between plus and minus one standard deviation from the mean contains approximately 68% of cases, and the portion lying between plus and minus 1.96 standard deviations contains approximately 95% of cases. That is, if a variable is normally distributed, you can expect 95% of the cases to be within 2 standard deviations from the mean.



Standard Error

The standard error is the estimated standard deviation of a sample statistic. For example, the standard error of a sample mean is an estimate of the likely standard deviation that the means from an infinite number of samples would have.

The standard error gives an indication of the accuracy of the sample mean as an estimate of the population mean. The smaller the standard error, the less the spread and the more likely it is that any sample mean is close to the population mean, in other words, that there is a 68% chance that the true population mean is within + or - 1 standard error of the sample mean.

Confidence Intervals

Because of random errors in samples, information from a sample contains a level of uncertainty. A confidence interval gives an estimated range of values as a measure of this uncertainty. The upper and lower bounds refer to the end points of the confidence interval.

For example, a confidence interval of 95% means that if you repeat a test multiple times, 95% of the time the mean - if that is what you are comparing - will be within the upper and lower bound of the confidence interval.

Quantitative and Categorical Data

The type of data determines what you can do with it. The two basic divisions of data are qualitative (categorical) data and quantitative (numeric) data.

Examples of categorical data include ID numbers, gender, and survey responses such as Yes or No.

Examples of quantitative data include weight in kilograms, time in seconds, and the number of respondents to a survey.

Quantitative data can be either continuous or discrete. Discrete data contains finite values that you can count, such as the number of employees in a division. Continuous data forms a continuum of infinite steps, such as the height, weight, or time.

Not all numbers are quantitative. For example, telephone numbers, ID numbers, and account numbers are numeric, but the average phone number or median ID number is not meaningful.

Levels of Measurement

Data can be nominal, ordinal, interval, or ratio.

Nominal data refers to categorically discrete data, such as customer names or product types.

Ordinal data refers to quantities that have a natural ordering, such as a rating scale from 1 to 5.

Interval data refers to data that you can measure along a scale with equal intervals, such as temperature and dates. The zero is arbitrary because 20F is not twice as hot as 10F.

Ratio data is similar to interval data that you can measure along a scale with equal intervals, but ratio data has a natural zero. For example, height, weight, and so on, have a natural zero at no height or no weight.

Null Hypothesis and Alternative Hypothesis

In hypothesis tests, such as t-tests and chi-square tests, you begin with the null hypothesis, often written as H₀. This states that there is no difference between the [populations](#) from which the samples were taken, or that no correlation exists between the variables in the population.

The alternative hypothesis, often written as H_a, states that a difference or relationship does exist.

For an example of how a hypothesis is tested, see "[Example: Using a One-Sample t-Test to Compare Household Income Against a Hypothetical Value](#)" (p. 257).

Types of Error and Statistical Power

There are two types of errors possible when conducting a statistical analysis.

- Type I error, also known as a false positive, is the error of rejecting a null hypothesis when it is true. In other words, you conclude that there is a difference where none exists.
- Type II error, also known as a false negative, is the error of failing to reject a null hypothesis when it is not true. In other words, you fail to conclude that there is a difference when in fact there is one.

The power of a test is the probability of correctly rejecting a null hypothesis when the null hypothesis is false. This probability is inversely related to the probability of making a type II error, not rejecting the null hypothesis when it is false.

For example, ANOVA has higher statistical power. There is less chance of a type II error, thinking there is no relationship when there is.

Significance in Statistics

IBM Cognos Statistics uses significance levels (or p-values) with many contexts.

The statistical significance of a result is the probability that the observed relationship (between variables) or difference (between means) in a sample would occur by pure chance if no such relationship or differences existed in the [population](#) from which the sample was drawn.

Significance is expressed as a decimal. A significant result is usually considered to exist where the p-value (or probability) is <0.05, meaning that the results obtained would occur by chance alone less than 5% of the time.

Degrees of Freedom

Degrees of freedom refers to the number of independent observations in a set of data that are free to vary if a parameter, such as the mean, is to remain constant.

When estimating a mean score from a sample, the number of independent observations is equal to the sample size minus one. You describe a sample size of 50 as having 49 (that is, $50 - 1$) degrees of freedom.

Treatment of Missing Values

As a method for handling missing values, listwise deletion excludes an entire record from analysis if any single value is missing.

For example, consider the following records for 10 customers:

Customer ID	Age in years	Previously defaulted	Household income in thousands
1	22	yes	31
2	21	no	44
3	65	no	42
4	24	no	missing
5	25	missing	35
6	32	yes	56
7	21	yes	36
8	missing	yes	34
9	40	no	120
10	44	no	50

Listwise deletion removes customers 4, 5, and 8 before performing any further analysis.

Pairwise deletion ignores records only as needed. For example, in calculating the means variables in the table above, pairwise deletion removes customer 8 from the calculation for the mean of Age in years, but not from any other calculation. Likewise, customer 5 is removed from the count of Previously defaulted customers, but not from any other calculation.

There is another alternative when using linear regression. The **Substitute with mean** option uses all cases, but substitutes missing observations with the mean of the variable.

One-tailed or Two-Tailed Test of Significance

You can test the [significance](#) of a correlation using one tail of a distribution or two tails.

The most common test of significance is the two-tailed test, where you test if a difference between groups or relationships goes either way. For example, using a two-tailed test, your hypothesis might look like this:

The null hypothesis (H_0): the correlation is 0

The alternative hypothesis (H_a): the correlation is not 0. The correlation can be either positive or negative, as long as it is not 0.

Using a one-tailed test, your hypothesis might look like this:

The null hypothesis (H_0): the correlation is 0 or negative.

The alternative hypothesis (H_a): the correlation is positive.

The two-tailed significance test is more conservative. Use a one-tailed test only when you are interested in detecting results in a particular direction, and the results in the other direction are of no interest.

Building Cases in IBM Cognos Statistics

As part of the IBM® Cognos® Statistics guided wizard, you supply a case variable for the analysis. A case uniquely identifies each data point being analyzed, such as a unique ID, order number, or part name. Using Time, for example, cases could either be defined as years or months. Depending on the analysis you are conducting, you might wish to use as cases for your analysis the aggregated form, for example Years, or the details, for example Months.

Cases can also be built by nesting query items or member sets. For example, you might nest a level-based set named Retailer Country beneath the level-based set Months to create a crossjoin where a case is created for the intersection tuple of each country within each month.

You can test out the nesting that you intend to use for statistical analysis by constructing a crosstab and performing nesting on the rows: the rows will be the cases.

There are many types of nesting behavior, such as single dimension nesting or multi dimension crossjoins. Behavior differs between dimensional model and relational models.

Using Dimensional Data

Because dimensional data is structured differently, you might have to use a slightly different technique to add [cases](#) to your statistical object. The items of interest that you need to create cases are often the children of a dimension member, rather than the member itself, which is a aggregate sum of its children.

For more information about this, see "[Add Dimensional Data to a Report](#)" (p. 347).

The following topics might be of particular interest:

- ["Insert a Member"](#) (p. 349)
- ["Create a Set of Members"](#) (p. 352)
- ["Insert a Hierarchy"](#) (p. 351)

For an example of how to insert data from a dimensional data source, see "[Example: Create A Profit Analysis Using Boxplots](#)" (p. 251).

Descriptive Statistics

Descriptive statistics quantitatively summarize a data set. For an overall sense of the data being analyzed, you can show descriptive statistics along with more formal analyses.

You can use descriptive statistics

- to look at averages, such as the mean or median
- to obtain information, such as the mean for groups of interest, that you might need to interpret other statistical tests
- to provide graphical representations of data, such as [histograms](#) and [boxplots](#)

Grouped and Ungrouped Statistics

You can create a table of summarized statistics, such as mean, median, and so on, for one or more numeric variables, based on all the cases. The table below shows the descriptive statistics for household income.

Descriptive Statistics	
	Household income in thousands
Mean	59.59
Std. Deviation	67.130
N	1500
Median	40.00
Minimum	12
Maximum	1,079

You can also place the measures adjacent to each other to create multiple measure statistics. The table below shows the descriptive statistics for three measures.

Descriptive Statistics			
	Household income in thousands	Credit card debt in thousands	Other debt in thousands
Mean	59.59	1.93	3.84
Std. Deviation	67.130	2.974	5.333
N	1500	1500	1500
Median	40.00	.99	2.21
Minimum	12	0	0
Maximum	1,079	36	63

You can also create a table that displays summarized statistics for cases grouped by [categorical](#) data based on a single measure, such as household income in thousands. The table below shows the mean household income for customers grouped by education level.

Descriptive Statistics					
	High school degree	Post-undergraduate degree	Did not complete high school	Some college	College degree
Mean	52.00	99.71	51.48	56.90	70.94
Std. Deviation	56.370	147.769	51.855	53.836	67.940
N	527	84	246	333	310
Median	35.00	59.50	36.00	39.00	49.00
Minimum	12	16	15	13	15
Maximum	533	1,079	497	403	512

Basic Descriptive Statistics

Descriptives tables describe the basic features of data in quantitative terms. You can choose one or more of the following statistics.

Measures of Central Tendency

Measures of central tendency provide a single number that summarizes the average distribution of a variable.

Statistic	Description
Mean	The arithmetic mean is the sum of samples divided by the number of cases. Note: Outlying values can affect the mean.
Harmonic mean	The total number of measurements divided by the sum or the reciprocals of the measurements. It is useful when averaging measurements that are rates or ratios, such as price/earnings ratios.
Geometric mean	The n th root of the product of the sample values, where n represents the number of cases. A common use for the geometric mean is in averaging rates of return on investments. For example, averaging the rate of return on an investment that earns 10% the first year, 50% the second year, and 30% the third year.
Median	Half of the cases fall above the median and half of the cases fall below the median. 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 less sensitive to outlying values than the mean.

Statistic	Description
Count (N)	The number of cases, observations, or records.
Sum	The total of the sample values, across all cases with non-missing values.

Measures of Dispersion

Measures of dispersion describe the degree of spread around the central tendency measure, that is, the extent to which observations cluster within the distribution. Measures of dispersion feature prominently in calculating [control charts](#).

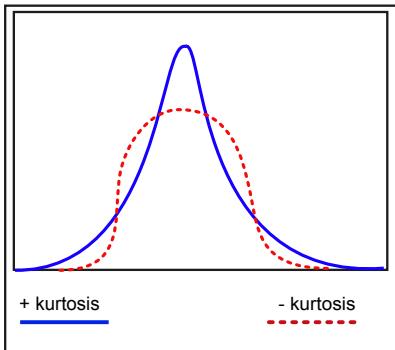
Measures of dispersion include simple measures such as maximum, minimum, and range; common statistical measures such as standard deviation and variance; and measures such as the interquartile range (IQR). For information about the interquartile range, see "[Boxplots](#)" (p. 248).

Statistic	Description
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.
Standard error	A measure of how much the value of the mean may vary from sample to sample taken from the same distribution.
Minimum	The smallest value of a numeric variable.
Maximum	The largest value of a numeric variable.
Range	The difference between the largest and smallest values of a numeric variable (the maximum minus the minimum).

Measures of Distribution

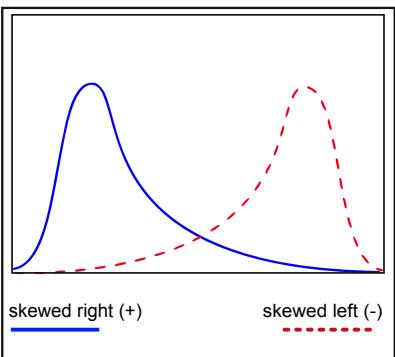
Measures of distribution, such as kurtosis and skewness, characterize the shape and symmetry of the distribution.

Kurtosis is a measure of the extent to which observations cluster around a central point. For the [normal distribution](#), the value of the kurtosis statistic is zero.



Positive kurtosis indicates that the observations show greater peakedness and longer tails than those in the normal distribution, and negative kurtosis indicates that the observations show less peakedness and have shorter tails.

Skewness is 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.

Create a Descriptives Table

You can use descriptive statistics to describe the main features of a set of data in quantitative terms.

Notes

- You must have IBM® Cognos® Statistics installed and configured to create statistical objects.
- For information about working with dimensional data, see "[Building Cases in IBM Cognos Statistics](#)" (p. 240).

Steps

1. From the File menu, click New.
If you want to change the package, click the ellipsis (...) and click a different package.
2. In the New dialog box, click **Statistics**. Click OK.
3. In the Select Statistic dialog box, expand **Descriptive Statistics**, and click **Basic Descriptives Statistics**. Click OK.

4. Select whether to create a table of standard descriptive statistics or descriptive statistics grouped by categorical variables. Click OK.
5. In the statistical object wizard, expand the package in the metadata tree if necessary and drag items to the following drop zones:
 - Drag one or more measure items to the Analysis variables drop zone. Click Next.
 - If you chose to summarize statistics grouped by categorical data, drag a non-measure item to the Grouping variables drop zone. Click Next.
6. Drag an item to the Cases variable drop zone to define a set of cases. Click Next.
7. Select the descriptive statistics that you need. Click Finish.
8. Run the report.

A table appears showing the selected descriptive statistics.

See the following example for a simple demonstration.

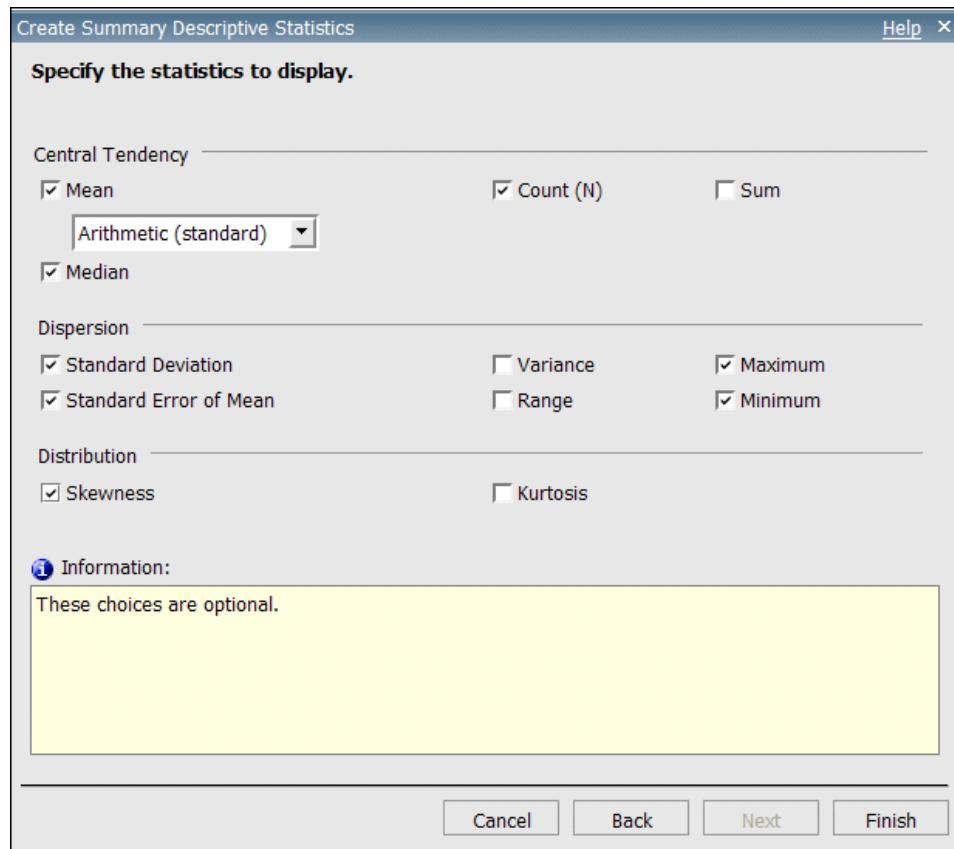
Example: Create a Report Showing Descriptive Statistics

You are a report author at a bank. A bank officer requests a report that provides customer statistics such as age, income, and credit card debt.

Note: You must have IBM Cognos Statistics installed and configured to create this example.

Steps

1. Open the BANKLOAN_CS package in IBM Cognos Report Studio.
2. In the Welcome dialog box, click Create a new report or template.
3. In the New dialog box, click Statistics and then click OK.
4. In the Select Statistic dialog box, expand Descriptive Statistics, click Basic Descriptives Statistics, and then click OK.
5. Select Summary descriptive statistics. Click OK.
6. In the statistical object wizard, expand BANKLOAN_CS in the metadata tree.
 - Drag the items for age, household income, and credit card debt to the Analysis variables drop zone. Click Next.
 - Drag ID to the Cases variable drop zone to define a set of cases. Click Next.



7. Select the **descriptive statistics** that you need. Click **Finish**.

8. Run the report.

A table appears showing descriptive statistics for the bank customers.

Descriptive Statistics			
	Age in years	Household income in thousands	Credit card debt in thousands
Mean	34.17	59.59	1.93
Std. Deviation	13.142	67.130	2.974
N	1500	1500	1500
Median	31.00	40.00	.99
Std. Error of Mean	.339	1.733	.077
Skewness	.952	5.543	4.877
Std. Error of Skewness	.063	.063	.063
Minimum	18	12	0
Maximum	79	1,079	36

Histograms

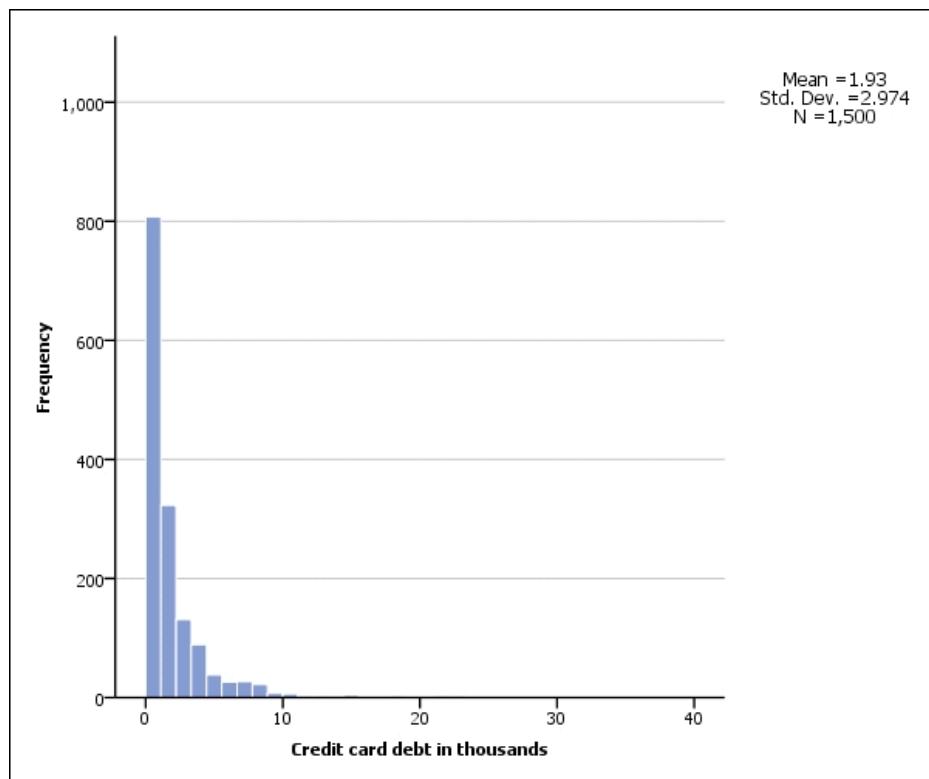
Histograms display the range of variable values in intervals of equal length. You can use a histogram to graphically summarize the frequency of observations using bins.

For information about how you can create a histogram and modify its properties, see "[Create a Histogram](#)" (p. 247).

For example, a bank examines the credit card debt of its clients. Each bin represents the number of clients that have credit card debt of zero to one thousand dollars, one thousand to two thousand dollars, and so on. Frequency refers to the number of clients in each specific bin. For example, a little over 800 clients have a credit card debt of one thousand dollars or less.

You can see by the count (N) that the results represent 1,500 clients. The mean credit card debt is 1,930 dollars.

This histogram shows the number of cases per unit interval: the height of each bar is equal to the number of people who fall into that category.



Create a Histogram

You can create a histogram to graphically summarize and display the distribution of a set of data. Histograms use bins to group individual data values into one instance of a graphic element, such as a bar. The bar height indicates the number of cases in the bin.

By default, the histogram divides the values of the variable into several evenly spaced intervals to size the bins. You can change the binning method to use

- a specified number of bins
- a specified number of units per bin

You can also modify the bar color, style, and chart size from the **Properties** pane.

Notes

- You must have IBM Cognos Statistics installed and configured to create statistical objects.
- If you are working with dimensional data, see "[Building Cases in IBM Cognos Statistics](#)" (p. 240).

Steps

1. From the File menu, click **New**.
If you want to change the package, click the ellipsis (...) and click a different package.
2. In the **New** dialog box, click **Statistics**. Click **OK**.
3. In the **Select Statistic** dialog box, expand **Descriptive Statistics**, click **Histogram**. Click **OK**.
4. In the statistical object wizard, expand the package in the metadata tree if necessary and drag items to the following drop zones:
 - Drag a measure item to the **Analysis Variable** drop zone. Click **Next**.
 - Drag an item to the **Cases variable** drop zone to define a set of [cases](#). Click **Next**.
5. Specify any output options, such as titles or footnotes. Click **Finish**.
You also have the option to change the histogram properties.
6. In the work area, click the histogram to display its properties.
7. On the **Properties** pane, under **General Settings**, you can specify
 - whether to display a curve
 - binning method
 - bar style
 - bar color
 - chart size
8. Run the report.

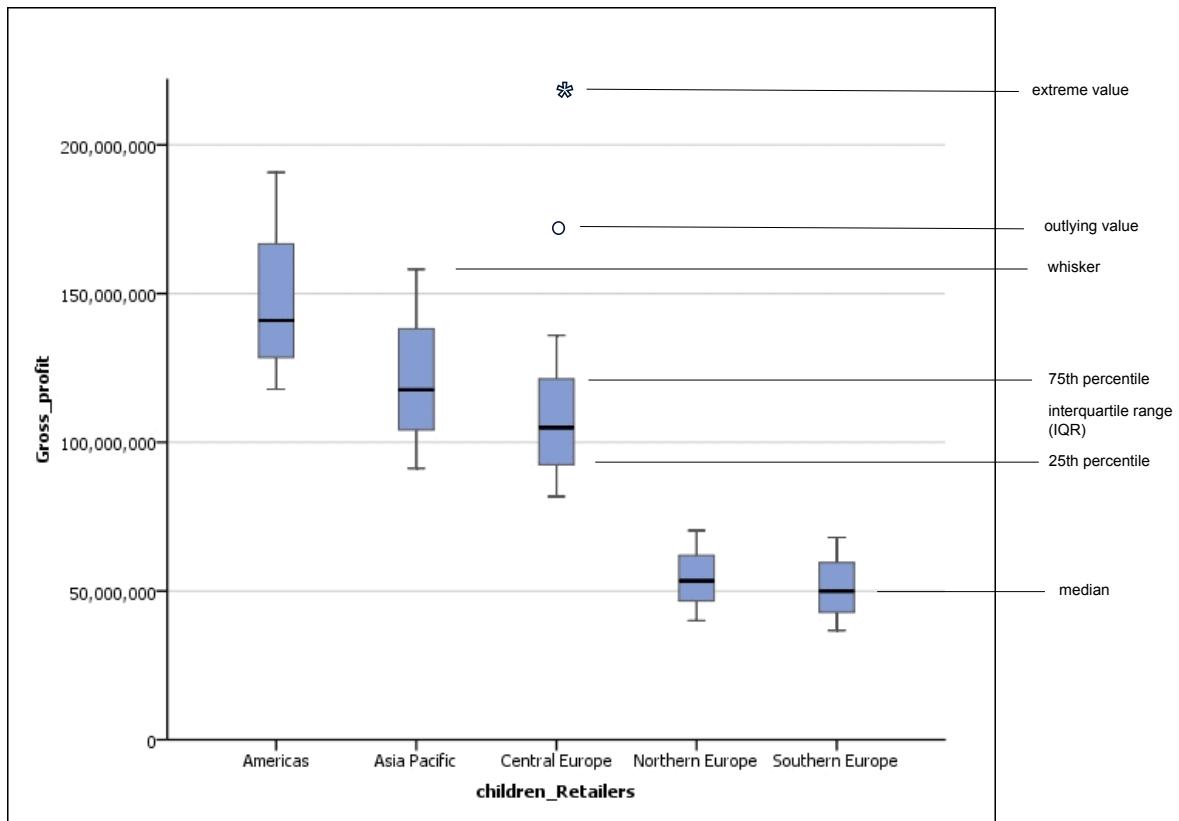
Boxplots

A boxplot (also known as a box-and-whisker chart) is a convenient way to show groups of numerical data, such as

- minimum and maximum values
- upper and lower quartiles
- median values
- outlying and extreme values

The spacings between the different parts of the boxplot indicate the degree of [dispersion](#) (spread) and [skewness](#) in the data and identify outliers.

Here are the different parts of the boxplot:



- The line in the middle of the boxes is the median. Half of the cases 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 with extreme values.
- The bottom of the box indicates the 25th percentile. Twenty-five percent of cases have values below the 25th percentile. The top of the box represents the 75th percentile. Twenty-five percent of case have values above the 75th percentile. This means that 50% of the cases lie within the box.

The interquartile range (IQR) is the difference between the 75th and 25th percentiles and corresponds to the length of the box.

- The T-bars that extend from the boxes are named inner fences or whiskers. These extend to the minimum and maximum value that is equal to or less than 1.5 times the IQR from the end of a box.

If the data is distributed [normally](#), approximately 95% of the data is expected to lie between the inner fences.

- A boxplot may contain outlying values marked as circles, and extreme values marked as asterisks.

Outlying values are values between 1.5 IQRs and 3 IQRs from the end of a box.

Extreme values are more than 3 IQRs from the end of a box.

Note: Outlying values can cause the mean to be different than the median.

Create a Boxplot

You can create a boxplot to show the median, quartiles, and outlier and extreme values for a variable. The interquartile range (IQR) is the difference between the 75th and 25th percentiles and corresponds to the length of the box.

You can display a boxplot for each category of a **categorical** variable, or you can leave the **Grouping variable** drop zone empty to display a single boxplot that summarizes all cases in the data.

Tip: The Income Growth report contains a boxplot. For more information about IBM Cognos Statistics samples, see "["Sample Reports and Packages" \(p. 567\)](#)".

Notes

- You must have IBM Cognos Statistics installed and configured to create statistical objects.
- For information about cases, see "["Building Cases in IBM Cognos Statistics" \(p. 240\)](#)".

Steps

1. From the **File** menu, click **New**.
If you want to change the package, click the ellipsis (...) and click a different package.
2. In the **New** dialog box, click **Statistics**. Click **OK**.
3. In the **Select Statistic** dialog box, expand **Descriptive Statistics** and click **Boxplot**. Click **OK**.
4. In the statistical object wizard, expand the package in the metadata tree if necessary and drag items to the following drop zones:
 - Drag a measure item to the **Analysis variable** drop zone. Click **Next**.
 - Drag an item to the **Cases variable** drop zone to define a set of **cases**. Click **Next**.
 - If you want to create boxplots for a grouped categorical variable, drag a non-measure item to the **Grouping variable** drop zone. Click **Next**.
5. Specify how to handle **missing values** and whether to exclude or include data when the group identifier is missing. Click **Next**.
6. Specify any output options, such as titles or footnotes. Click **Finish**.
You also have the option to change the boxplot properties.
7. In the work area, click the boxplot to display its properties.
8. On the **Properties** pane, under **General Settings**, you can specify
 - bar color
 - chart size
9. Run the report.

See the following example for a demonstration.

Example: Create A Profit Analysis Using Boxplots

You are a report author. You are asked to create a report showing the profitability of retailers in all sales territories.

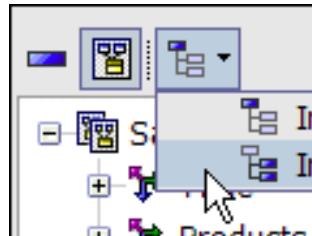
You use boxplots to compare the results of each sales territory using a five-number summary: the median, the 25th and 75th percentiles, and the minimum and maximum observed values.

Note: You must have IBM Cognos Statistics installed and configured to create this example.

Steps

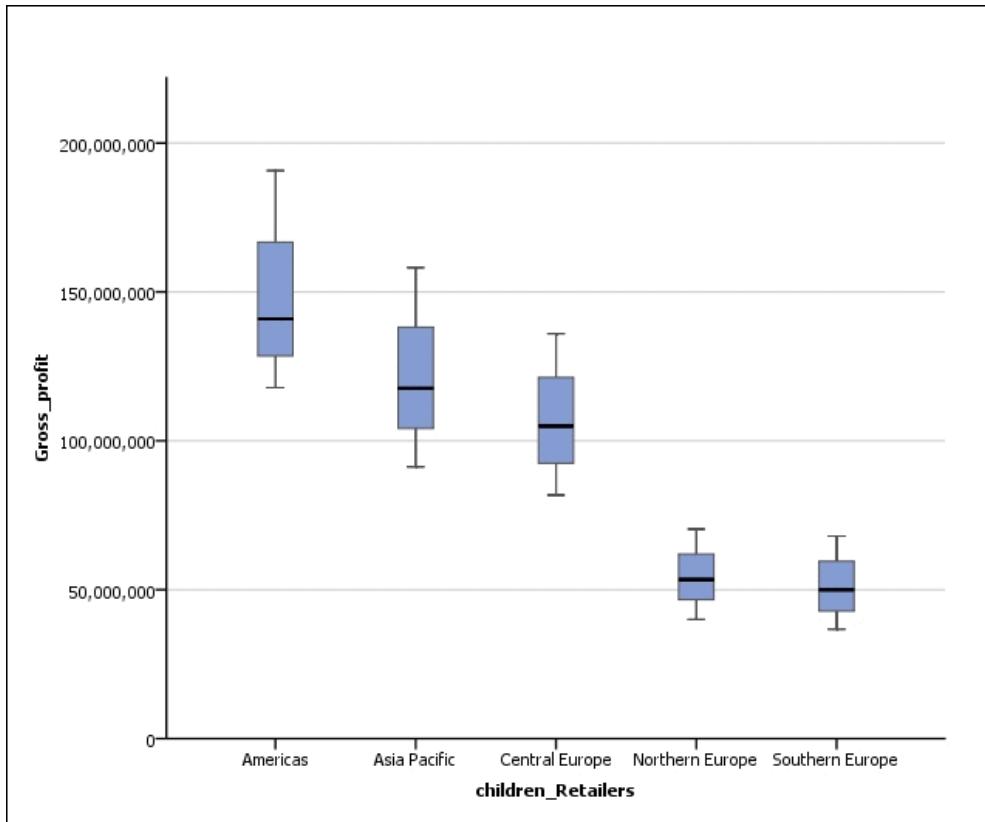
1. Open the Sales and Marketing cube in Report Studio.
2. In the **Welcome** dialog box, click **Create a new report or template**.
3. In the **New** dialog box, click **Statistics**. Click **OK**.
4. In the **Select Statistic** dialog box, expand **Descriptive Statistics**, and click **Boxplot**. Click **OK**.
5. In the statistical object wizard, expand Sales and Marketing in the metadata tree and drag items to the following drop zones:
 - Expand the Measures folder and drag the gross profit item to the **Analysis variable** drop zone. Click **Next**.

Because you are using a dimensional data source, you must insert the children of the members for the next two items by using the **Insert children** button. These buttons are available when you use a dimensional data source.



- Click the **Insert children** button. Expand the Time dimension. Under the Members folder, click the Time level. Insert the children of the Time member in the **Cases variable** drop zone to define a set of **cases**. Click **Next**.
- Click the **Insert children** button. Expand the Retailer dimension. Under the Members folder, click the Retailer member. Insert the children of the Retailers level in the **Grouping variable** drop zone. Click **Finish**.
6. Run the report.

This boxplot shows the gross profit statistics for various markets.



Create a Q-Q Plot

You can create a Q-Q (quantile-quantile) plot to chart the quantiles of a variable's distribution against a distribution of your choice, including the [normal distribution](#).

Quantiles are values that divide the cases into a number of equal-sized groups.

If the selected variable matches the test distribution, the points cluster around a straight line. Outliers appear as points that are farther from the overall pattern of points.

Notes

- You must have IBM Cognos Statistics installed and configured to create statistical objects.

Steps

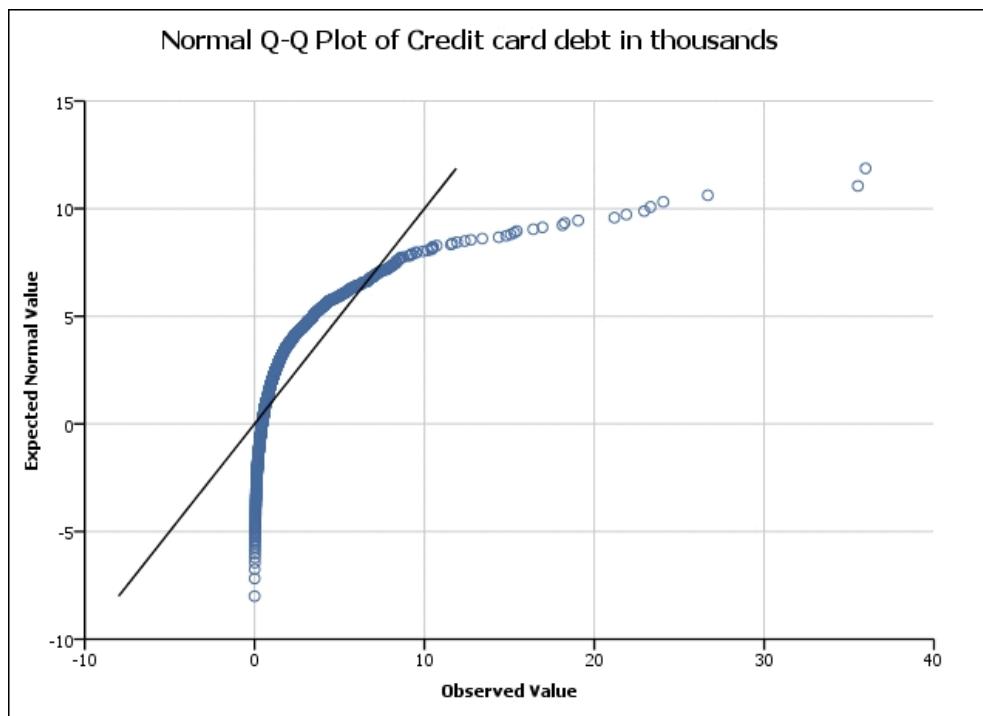
1. From the File menu, click New.
If you want to change the package, click the ellipsis (...) and click a different package.
2. In the New dialog box, click Statistics and then click OK.
3. In the Select Statistic dialog box, expand Descriptive Statistics, click Q-Q Plot, and then click OK.
4. In the statistical object wizard, expand the package in the metadata tree if necessary and drag items to the following drop zones:
 - Drag a measure item to the Analysis variable drop zone. Click Next.

- Drag an item to the **Cases** drop zone to define a set of **cases**. Click **Next**.
5. Specify any output options. Click **Next**.
 6. Specify any transformation options, proportional estimate options, and rank options. Click **Finish**.
 7. Run the report.

By default, the results include

- a Q-Q plot
- a detrended Q-Q plot
- a table of estimated distribution parameters

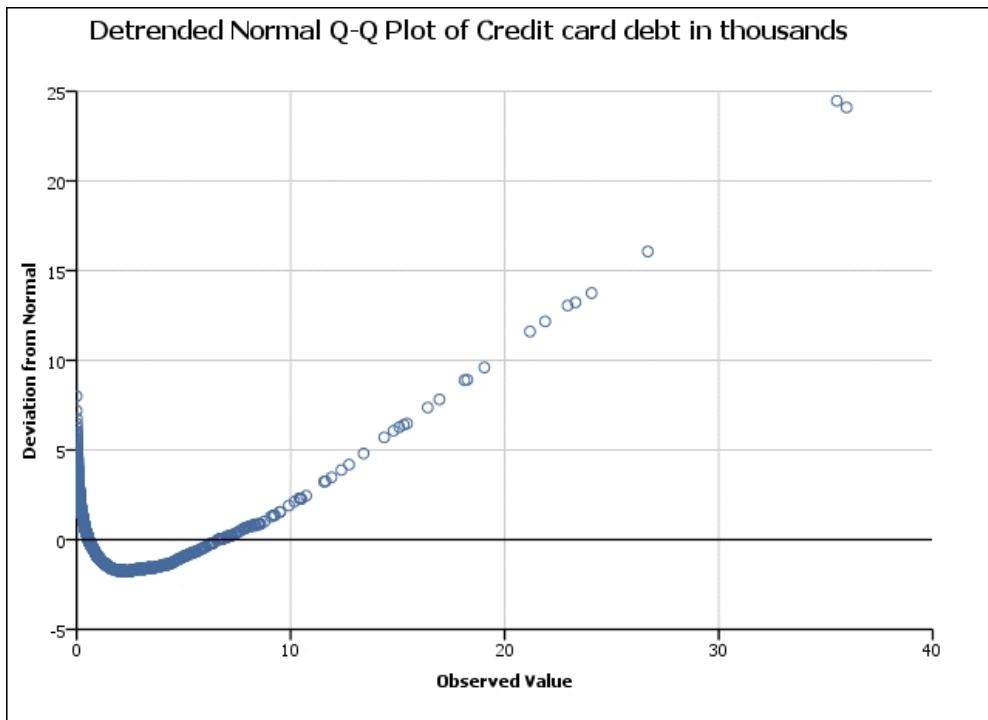
Q-Q plots produce a scatterplot with the quantiles of the variable on the horizontal axis and the expected normal distribution on the vertical axis. A plot of variables against the expected normal variables reveals a straight line. Curvature of the points indicates departures from normality. Outliers appear as points that appear far from the overall pattern of points.



The detrended Q-Q plot below is another way of looking at the data.

This plot, based on the same data as the Q-Q plot above, displays the deviation of each point in the normal probability plot from the straight line corresponding to the normal.

The vertical axis represents the difference between each point in the normal probability plot and the straight line representing the perfect normal. The horizontal axis represents the observed value. This visually clarifies the areas where there is greatest deviation from the normal. If the data in the sample were normal, all the data points in the detrended normal plot would appear near the horizontal line centered at 0.



Q-Q Plot Options

Test Distributions

Aside from the normal distribution, other available test distributions include beta, chi-square, exponential, gamma, half-normal, Laplace, logistic, lognormal, normal, Pareto, Student's t, Weibull, and uniform.

Depending on the distribution selected, you can specify the [degrees of freedom](#) and other parameters.

Customize Output

You can specify to show the following:

- a case processing summary table
- an estimated distribution parameters table
- a Q-Q plot
- a detrended Q-Q plot

Other Q-Q Plot Options

You can also

- change the proportion estimate formula
- specify how to assign ranks to ties
- specify different transformation options

Means Comparison

You can compare the means of two or more groups to determine if the difference between the groups is statistically significant, that is, if the difference is due to something other than random chance.

You use the t-test to test the differences between a group and a hypothetical test value. If the test involves more than one group, you can use [analysis of variance](#) (ANOVA).

These tests use parametric data, that is, numeric data from a normal distribution. For nonparametric data, such as nominal classifications or data that is not normally distributed, you use [nonparametric tests](#).

t-Tests

The one-sample t-test tests the probability that the difference between the sample mean and a test value is due to chance. Probabilities of .05 or less are typically considered significant.

Grouped or Ungrouped

You can create a t-test to compare means based on all the cases. Shown below is a typical t-test report, showing tables for descriptive statistics and t-test statistics.

One-Sample Statistics				
	N	Mean	Std. Deviation	Std. Error Mean
Household income in thousands	1500	59.59	67.130	1.733
One-Sample Test				
	Test Value = 50.74			
	t	df	Sig. (2-tailed)	Mean Difference
Household income in thousands	5.105	1499	.000	8.849
			95% Confidence Interval of the Difference	
			Lower	Upper
			5.45	12.25

You can create t-tests that give results for cases grouped by [categorical data](#) for a single measure, such as household income in thousands. The example below shows the t-test results of mean household income for customers grouped by education level.

One-Sample Statistics

	N	Mean	Std. Deviation	Std. Error Mean
College degree	310	70.94	67.940	3.859
High school degree	527	52.00	56.370	2.456
Some college	333	56.90	53.836	2.950
Did not complete high school	246	51.48	51.855	3.306
Post-undergraduate degree	84	99.71	147.769	16.123

One-Sample Test

	Test Value = 50.74						
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference		
					Lower	Upper	
College degree	5.234	309	.000	20.195	12.60	27.79	
High school degree	.514	526	.608	1.262	-3.56	6.09	
Some college	2.089	332	.037	6.164	.36	11.97	
Did not complete high school	.222	245	.824	.736	-5.78	7.25	
Post-undergraduate degree	3.038	83	.003	48.974	16.91	81.04	

Create a One-Sample t-Test

You can use a one-sample t-test to test whether the mean of a variable differs from a specified test value. For example, you might want to test whether the mean household income for the customers of a bank differs from the national average.

Note

- You must have IBM® Cognos® Statistics installed and configured to create statistical objects.

Steps

1. From the **File** menu, click **New**.

If you want to change the package, click the ellipsis (...) and click a different package.
2. In the **New** dialog box, click **Statistics**. Click **OK**.
3. In the **Select Statistics** dialog box, expand **Means Comparison** and click **One-Sample t-Test**. Click **OK**.
4. Select whether to create a summary t-test or a t-test **grouped by categorical variables**. Click **OK**.
5. In the statistical object wizard, expand the package in the metadata tree if necessary and drag items to the following drop zones:
 - Drag a measure item to the **Analysis variables** drop zone. Click **Next**.

If you chose to summarize statistics for cases grouped by **categorical data**, drag a single measure to the drop zone.

- Drag an item to the **Cases variable** drop zone. Click **Next**.
 - If you chose to summarize statistics for cases grouped by categorical data, you must drag one or more categorical items to the **Grouping variables** drop zone. Click **Next**.
6. Specify a numeric test value against which to compare.
- You must enter the test value in the same format that the variable uses.
7. Specify any other output options. You can:
- change the [confidence interval](#)
 - specify how to treat [missing values](#)
 - choose whether to show the one-sample statistics table
8. Click **Finish**.
9. Run the report.

You see two tables:

- A table showing the sample count, mean, standard deviation, and [standard error](#) of the mean.
- A table showing the t-value, [degrees of freedom](#), significance, mean difference between the data value and the test value, and a confidence interval for this difference.

The p-value tells you if there is a significant difference between the test value and the sample mean.

The t-statistic represents the number of standard errors that separate the sample mean from your hypothesized test value.

See the following example for a demonstration.

Example: Using a One-Sample t-Test to Compare Household Income Against a Hypothetical Value

You are a report author for a bank. Your research tells you that the mean national household income is \$50,740. The mean household income of the bank's customers is above the national average. You want to know if the difference in mean household incomes is genuinely higher than average, or if this the result of chance.

The null hypothesis is that there is no difference between the mean household income of the bank's customers and the hypothesized national mean household income. The alternative hypothesis is that there is a significant difference between the two.

Note: You must have IBM Cognos Statistics installed and configured to create this example.

Steps

1. Open the BANKLOAN_CS package in IBM Cognos Report Studio.
2. In the **Welcome** dialog box, click **Create a new report or template**.
3. In the **New** dialog box, click **Statistics** and then click **OK**.

4. In the Select Statistic dialog box, expand Means Comparison, click One-Sample t-Test, and then click OK.
5. Select Summary One-Sample t-Test. Click OK.
6. In the statistical object wizard, expand the BANKLOAN_CS package if necessary and drag items to the following drop zones:
 - Drag the Household income in thousands item to the Analysis variables drop zone. Click Next.
 - Drag the Customer ID item to the Cases variable drop zone to define a set of cases and click Next.
7. Enter 50.74 in Test value. Leave the Confidence interval, Missing values, and Customize output at the default settings.
8. Click Finish.
9. Run the report.

You see a statistics table and a t-test table, as shown below.

One-Sample Statistics					
	N	Mean	Std. Deviation	Std. Error Mean	
Household income in thousands	1500	59.59	67.130	1.733	
One-Sample Test					
	Test Value = 50.74				
	t	df	Sig. (2-tailed)	Mean Difference	95% Confidence Interval of the Difference
Household income in thousands	5.105	1499	.000	8.849	5.45 12.25

The statistic of interest is the p-value for **significance**. This value is .000, rounded to three decimal places. Because this value is <0.05, you reject the null hypothesis. You can state that the mean household income of bank customers is significantly different from the national average of \$50,740.

The t-statistic, 5.105, is the number of **standard errors** (1.733) that separate the sample mean (\$59,590) from the hypothesized value (\$50,740).

The Mean Difference is obtained by subtracting the test value (50.74 in this example) from the sample mean.

The 95% **confidence interval** of the difference tells you that, repeated multiple times, there is a 95% chance that the mean difference between household income of your customers and that of the national average will lie between \$5,450 and \$12,250.

Using One-Way ANOVA to Compare Means

You can use one-way ANOVA to assess whether groups of means differ significantly.

ANOVA assumes that there is homogeneity of variance, that is, that the variance within each of the groups is equal. You can check for homogeneity of variance by using the [Levene's test](#).

ANOVA does not tell you which groups are different from the others, only whether a significant difference exists. After finding a significant difference, you can do [post hoc tests](#) on the independent variable to examine the differences between groups.

Tip: The DVD Score report uses one-way ANOVA. For more information about IBM Cognos Statistics samples, see "[Sample Reports and Packages](#)" (p. 567).

Note

- You must have IBM Cognos Statistics installed and configured to create statistical objects.

Steps

1. From the File menu, click New.

If you want to change the package, click the ellipsis (...) and click a different package.

2. In the New dialog box, click Statistics. Click OK.

3. In the Select Statistics dialog box, expand Means Comparison and click One-Way ANOVA. Click OK.

4. In the statistical object wizard, expand the package in the metadata tree if necessary and drag items to the following drop zones:

5. • Drag one or more items to the Dependent variables drop zone. Click Next.
 - Drag a nonmeasure item to the Independent variable drop zone. Click Next.
 - Drag an item to the Cases variable drop zone to define a set of [cases](#). Click Next.

6. Optionally, you can specify [contrasts](#), how to treat [missing values](#), and which [output options](#) to display. Click Next.

7. Specify a [significance](#) level and any [post hoc tests](#). Click Finish.

8. Run the report.

By default, you see:

- A table of ANOVA statistics

The statistic of interest is the significance.

- A chart of the observed means

You also see any other selected output options.

See the following example for a demonstration.

Example: Using a One-Way ANOVA to Compare Mean Household Income Between Groups by Educational Level

You are a report author for a bank. You want to know how the mean household income of your customers varies by education level. What is the probability that the differences occur by random chance?

Note: You must have IBM Cognos Statistics installed and configured to create this example.

Steps

1. Open the BANKLOAN_CS package in Report Studio.
2. In the Welcome dialog box, click **Create a new report or template**.
3. In the New dialog box, click **Statistics**. Click **OK**.
4. In the Select Statistic dialog box, expand **Means Comparison** and click **One-Way ANOVA**. Click **OK**.
5. In the statistical object wizard, expand the BANKLOAN_CS package and drag items to the following drop zones:
 - Drag the Household income in thousands item to the **Dependent variables** drop zone. Click **Next**.
 - Drag the Level of education item to the **Independent variable** drop zone. Click **Next**.
 - Drag the Customer ID item to the **Cases variable** drop zone to define a set of **cases** and click **Next**.
6. Leave the default output options. Click **Next**.
7. Select the post hoc test **Tukey**.
8. Click **Finish**.
9. Run the report.

You see an ANOVA table as shown below, the results of the Tukey post hoc test, and a chart of means. Look at the ANOVA table below.

Household income in thousands					
ANOVA					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	224084.078	4	56021.020	12.823	.000
Within Groups	6531097.129	1495	4368.627		
Total	6755181.207	1499			

The statistic of interest is the **significance**. The p-value is .000, rounded to three decimal places. Because the significance <.05, you reject the null hypothesis. You conclude that there is a significant difference between the household income of customers with different levels of education.

This does not tell you which level of education might account for the difference. Look at the post hoc test results shown below. The asterisk (*) marks which differences are significant. You can see that the college degree and post-undergraduate degree groups are associated with higher household income more significantly than the other levels of education.

Household income in thousands								
		Multiple Comparisons		Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
		(I) Level of education	(J) Level of education				Lower Bound	Upper Bound
Tukey HSD	High school degree	Post-undergraduate degree	Post-undergraduate degree	-47.712(*)	7.765	.000	-68.92	-26.51
			Did not complete high school	.526	5.104	1.000	-13.41	14.47
			Some college	-4.902	4.627	.827	-17.54	7.73
			College degree	-18.934(*)	4.731	.001	-31.85	-6.01
	Post-undergraduate degree	High school degree	High school degree	47.712(*)	7.765	.000	26.51	68.92
			Did not complete high school	48.239(*)	8.353	.000	25.43	71.05
			Some college	42.810(*)	8.070	.000	20.77	64.85
			College degree	28.779(*)	8.130	.004	6.57	50.98
	Did not complete high school	High school degree	High school degree	-.526	5.104	1.000	-14.47	13.41
			Post-undergraduate degree	-48.239(*)	8.353	.000	-71.05	-25.43
			Some college	-5.428	5.557	.866	-20.60	9.75
			College degree	-19.460(*)	5.644	.005	-34.87	-4.05
	Some college	Post-undergraduate degree	High school degree	4.902	4.627	.827	-7.73	17.54
			Post-undergraduate degree	-42.810(*)	8.070	.000	-64.85	-20.77
			Did not complete high school	5.428	5.557	.866	-9.75	20.60
			College degree	-14.032	5.216	.056	-28.28	.22
	College degree	Post-undergraduate degree	High school degree	18.934(*)	4.731	.001	6.01	31.85
			Post-undergraduate degree	-28.779(*)	8.130	.004	-50.98	-6.57
			Did not complete high school	19.460(*)	5.644	.005	4.05	34.87
			Some college	14.032	5.216	.056	-.22	28.28

* The mean difference is significant at the 0.05 level.

You also see a table for homogeneous subsets as shown below. In each subset column, only the means of the variables in that subset are shown. The subsets can overlap, that is, a group can belong to more than one subset.

Homogeneous Subsets				
Tukey HSD				
Household income in thousands				
Means for groups in homogeneous subsets are displayed.				
Level of education	N	Subset for alpha = 0.05	1	2
Did not complete high school	246	51.48		
High school degree	527	52.00		
Some college	333	56.90	56.90	
College degree	310		70.94	
Post-undergraduate degree	84			99.71
Sig.		.919	.195	1.000

ANOVA Options

Aside from specifying how to treat [missing values](#) and whether to show a table of [descriptive](#) statistics for the independent variable, you can specify the following options in your report.

Contrasts

If the groups you compare are defined along a continuum, you can use polynomial contrasts to test for a trend in the means. The first degree of freedom contains the linear effect across the levels of the factor, the second degree of freedom contains the quadratic effect, and so on.

For example, as a factor you have three age groups. You can test for a linear and a quadratic trend across the age groups. The linear contrast compares the lowest age with the highest age group, and the quadratic contrast compares the middle group with the lowest and highest age groups together.

Fixed and Random Effects

Use a fixed effects model when the groups in your analysis are selected and ordered in some meaningful way. Random effects in ANOVA assume that the groups are a random sampling of many potential groups. For example, with groups defined by Age or by Gender use the fixed effects model, whereas with groups defined by City you should use the random effects model.

Homogeneity of Variance Tests

You can use Levene's test for homogeneity of variance to test the ANOVA assumption that each group of the independent variable has the same variance.

If the Levene statistic is significant at the .05 level, you reject the null hypothesis that the groups have equal variances.

Test of Homogeneity of Variances

	Household income in thousands
Levene Statistic	13.403
df1	4
df2	1495
Sig.	.000

Equality of Means Tests

You can use either the Brown-Forsythe test or the Welch test to assess the equality of means when groups are unequal in size. These tests do not assume homogeneity of variance.

If the statistic is significant at the .05 level, you reject the null hypothesis that the groups have equal means.

Robust Tests of Equality of Means

Tests	Statistic	df1	df2	Sig.
Welch	6.683	4	418.020	.000
Brown-Forsythe	7.266	4	182.659	.000

Post Hoc Tests

ANOVA does not tell you which groups are different from the others, only that a difference exists. After finding a significant difference, you can do [post hoc tests](#) on the factor to examine the differences between levels.

This table presents the tests available, ordered from the most liberal (greater statistical [power](#) and greater false-positive rate) to the most conservative (smaller false-positive rate and less statistical power).

These post hoc tests assume equal variances.

Post hoc test	Description
LSD	The LSD (least significant difference) method applies standard t-tests to all possible pairs of group means.

Post hoc test	Description
SNK, REGWF, REGWQ, and Duncan	<p>The SNK (Student-Newman-Keuls), REGWF (Ryan-Einot-Gabriel-Welsh F), REGWQ (Ryan-Einot-Gabriel-Welsh Q), and Duncan methods involve sequential testing.</p>
	<p>After ordering the group means from lowest to highest, the two most extreme means are tested for a significant difference using a critical value that is adjusted for the fact that these are the extremes from a larger set of means.</p>
	<p>If these means are found to not be significantly different, the testing stops. If they are different, then the testing continues with the next most extreme set, and so on.</p>
Bonferroni, Sidak	<p>The Bonferroni test (also called the Dunn procedure) and Sidak test (also called Dunn-Sidak) are performed at a stringent significance level to ensure that the family-wise (that is, applying to the set of tests) false-positive rate does not exceed the specified value.</p>
Tukey (b)	<p>The Tukey (b) test is a compromise test that combines the Tukey test (see next test) and the SNK criterion to produce a test result that falls between the two.</p>
Tukey	<p>Tukey's HSD (Honestly Significant Difference), also called Tukey HSD, WSD, or the Tukey(a) test, controls the false-positive rate family-wise.</p>
	<p>This means, if you are testing at the .05 level, that when performing all pairwise comparisons the probability of obtaining one or more false positives is .05.</p>
Scheffe	<p>Scheffe's method also controls the family-wise error rate. It adjusts not only for the pairwise comparisons, but also for any possible comparison that you specify.</p>
Hochberg's GT2, Gabriel	<p>Most post hoc procedures mentioned above (excepting LSD, Bonferroni, and Sidak tests) are derived by assuming equal group sample sizes in addition to homogeneity of variance and normality of error. When the subgroup sizes are unequal, Report Studio substitutes a single value (the harmonic mean) for the sample size. Hochberg's GT2 and Gabriel's post hoc test explicitly allow for unequal sample sizes.</p>

Post hoc test	Description
Waller-Duncan	The Waller-Duncan test takes a Bayesian approach that adjusts the criterion value based on the size of the overall F statistic in order to be sensitive to the types of group differences associated with the F statistic (for example, large or small).
Dunnett	A pairwise multiple comparison t -test that compares a set of treatments against a single control mean. You can choose the first category or the last category as the default control category. 2-sided tests that the mean at any level (except the control category) of the factor is not equal to that of the control category. < Control tests if the mean at any level of the factor is smaller than that of the control category. > Control tests if the mean at any level of the factor is greater than that of the control category.

These post hoc tests adjust for unequal variances and sample sizes in the groups.

Post hoc test	Description
Games-Howell	The Games-Howell test is designed for unequal variances and unequal sample sizes, and is based on the q-statistic distribution.
Tamhane's T2	Tamhane's T2 is a conservative test. It is considered more appropriate than Tukey's HSD when cell sizes are unequal, or when homogeneity of variances is violated.
Dunnett's T3, Dunnett's C	Use these tests instead of Games-Howell when it is essential to maintain control over the significance level across multiple tests.

Nonparametric Tests

You use nonparametric tests to compare frequencies in **categorical** data, such as payment methods or gender. You test for **significant** differences between observed frequencies and expected frequencies in data that does not have a normal distribution. If the observed frequencies and expected frequencies are significantly different, you reject the **null hypothesis**.

To analyze data from a single categorical variable, you can use a one-way chi-square test. When analysis of categorical data involves two variables, you can use a two-way chi-square test.

Create A One-Way Chi-Square Test

One-way chi-square tests (also known as chi-square goodness-of-fit tests) compare observed frequencies against expected frequencies using data from a single [categorical](#) variable. By default, this test compares the observed and expected frequencies in the category to test that all categories contain the same proportion of values. You can also specify different proportions for the expected values.

Use the **One-Way Chi-Square** option if the data is a simple list of categorical values, such as payment method, where the frequencies come from the count for each unique value.

ID	Method of Payment
1	credit card
2	cash
3	credit card

Use the **One-Way Chi-Square with counts measure** option if the data is organized in unique category & count pairs.

ID	Method of Payment	Count
1	Credit card	132
2	Cash	267
3	Debit	133

When your analysis involves two categorical variables, you can use a two-way chi-square test.

Note

- You must have IBM® Cognos® Statistics installed and configured to create statistical objects.

Steps

1. From the **File** menu, click **New**.
If you want to change the package, click the ellipsis (...) and click a different package.
2. In the **New** dialog box, click **Statistics**. Click **OK**.
3. In the **Select Statistics** dialog box, expand **Nonparametric Tests** and click **One-Way Chi-Square Test**. Click **OK**.
4. Select the type of test, **One-Way Chi-Square** or **One-Way Chi-Square with counts measure**. Click **OK**.
5. In the statistical object wizard, expand the package in the metadata tree if necessary and drag items to the following drop zones:

- Drag a nonmeasure item to the **Analysis variable** drop zone. Click **Next**.
 - If you chose **One-Way Chi-Square with counts measure**, drag the item that contains the counts measure to the **Counts variable** drop zone. Click **Next**.
 - Drag an item to the **Cases variable** drop zone to define a set of **cases**. Click **Next**.
6. Define the output. By default, a frequency table and a statistics table display. You can also display a table of quartiles. Click **Finish**.
7. Run the report.

You see two tables:

- A table showing the frequency distribution, expected frequencies, and residuals. A residual is the difference between the observed value and the expected value.
- A table showing the statistics for chi-square, [degrees of freedom](#), and [significance](#).

See the following example for a simple demonstration.

Example: Using a One-Way Chi-Square Test to Compare Daily Hospital Discharge Rates

You are a report author in a hospital. To deal with staff shortages, the administration must determine if the number of patients that the hospital discharges varies significantly by the day of the week.

You begin with the null hypothesis that there is no difference between the expected frequencies of discharged patients for the different days of the week and the observed frequencies. The alternative hypothesis is that there is a significant difference between the frequency of discharged patients and the day of the week.

Notes

- You must have IBM Cognos Statistics installed and configured to create statistical objects.

Steps

1. Open the DISCHARGEDATA package in IBM Cognos Report Studio.
2. From the **File** menu, click **New**.
3. In the **New** dialog box, click **Statistics**. Click **OK**.
4. In the **Select Statistics** dialog box, expand **Nonparametric Tests**, click **One-Way Chi-Square Test**. Click **OK**.
5. Select **One-way Chi-Square with counts measure**. Click **OK**.
6. From the metadata tree, expand the package if necessary and drag items to the following drop zones:
 - Drag the Day of the week item to the **Analysis variable** drop zone. Click **Next**.
 - Drag the Average daily discharge item to the **Counts variable** drop zone. Click **Next**.
 - Drag the ID item to the **Cases variable** drop zone. Click **Next**.

7. Leave the output set to the default. Click Finish.

8. Run the report.

You see two tables:

- A table showing the observed frequencies, expected frequencies, and residuals.
- A table showing the statistics for chi-square, [degrees of freedom](#), and [significance](#).

Frequencies			
Day of the week			
Day of the week	Observed N	Expected N	Residual
Sunday	44	84.1	-40.1
Monday	78	84.1	-6.1
Tuesday	90	84.1	5.9
Wednesday	94	84.1	9.9
Thursday	89	84.1	4.9
Friday	110	84.1	25.9
Saturday	84	84.1	-.1
Total	589		

Test Statistics	
	Day of the week
Chi-Square	29.389(1)
df	6
Asymp. Sig.	.000

1 0 cells (.0%) have expected frequencies less than 5. The minimum expected cell frequency is 84.1.

The statistic of interest is the significance. The p-value is .000, rounded to three decimal places. Because the significance is <.05, you reject the null hypothesis. You conclude that there is a significant difference between the expected and observed frequencies, that is, discharge rates differ significantly according to the day of the week.

Asymptotic significance means significance that is most accurate with large samples and less accurate with small samples.

Example: Specify the Proportion for Expected Values in a Chi-Square Test

By default, the one-way chi-square test uses equal proportions to evenly calculate the expected values. You can also specify different proportions. For example, you hypothesize that 20% of patients are discharged on Friday, 20% on Saturday, and 20% on Sunday. The remainder are dis-

charged throughout the rest of the week, 10% per day. You want to test whether the observed proportions from your sample differ significantly from these hypothesized proportions.

It is important that you add the numbers in the order that the variables appear in the data. In the example, the days of the week appear like this.

ID	Day of the week
1	Sunday
2	Monday
3	Tuesday
4	Wednesday
5	Thursday
6	Friday
7	Saturday

If you want to specify 20% for Friday, Saturday, and Sunday, you must add the values in the **Add values** dialog box in this order: 20, 10,10,10,10,20,20.

Steps

1. Open the DISCHARGEDATA package in Report Studio.
2. From the **File** menu, click **New**.
3. In the **New** dialog box, click **Statistics**. Click **OK**.
4. In the **Select Statistics** dialog box, expand **Nonparametric Tests**, click **One-Way Chi-Square Test**. Click **OK**.
5. Select **One-way Chi-Square with counts measure**. Click **OK**.
6. In the statistical object wizard, expand the package in the metadata tree if necessary and drag items to the following drop zones:
 - Drag the Day of the week item to the **Analysis variable** drop zone. Click **Next**.
 - Drag the Average daily discharge item to the **Counts variable** drop zone. Click **Next**.
 - Drag the ID item to the **Cases variable** drop zone. Click **Next**.
7. Leave the output set to the default. Click **Finish**.
8. In the work area, click the chi-square statistical object to display its properties.

9. On the Properties pane, under General Settings, change Expected Values from All categories are equal to values.
10. In the Expected Values dialog box, click the add icon .
11. Under Numbers only, type 20. Click OK.
12. Repeat steps 10 and 11 for each of these proportion figures: 10,10,10,10,20, and 20.
13. Run the report.

You see two tables, as in the preceding example. The number of expected observations is now calculated using your specified proportions. For example, the number of expected discharged patients for Sunday is now 117.8, that is, 20% of the total count of 589.

Create a Two-Way Chi-Square Test

Two-way chi-square tests (also known as chi-square tests of independence) compare observed frequencies against expected frequencies using data from two **categorical** variables.

For example, you can compare the observed number of transactions that use different methods of payment (cash, check, credit card, and so on) identified by gender against an expected number to answer the question: do methods of payment by different genders vary significantly?

Use the **Two-Way Chi-Square** option if your data contains pairs of categorical values where the frequencies come from the count for each unique pairing.

ID	Gender	Method of Payment
1	male	cash
2	male	cash
3	female	credit card

Use the **Two-Way Chi-Square with counts measure** option if your data contains unique pairs, plus a count.

ID	Gender	Method of Payment	Count
1	Female	cash	321
2	Male	cash	237

Note

- You must have IBM Cognos Statistics installed and configured to create statistical objects.

Steps

1. From the File menu, click New.

If you want to change the package, click the ellipsis (...) and click a different package.

2. In the New dialog box, click **Statistics**. Click **OK**.
3. In the Select Statistics dialog box, expand Nonparametric Tests and click Two-Way Chi-Square Test. Click **OK**.
4. Select the type of test, Two-Way Chi-Square or Two-Way Chi-Square with counts measure. Click **OK**.
5. In the statistical object wizard, expand the package in the metadata tree if necessary and drag items to the following drop zones:
 - Drag an nonmeasure item to the **Analysis variable 1** drop zones. Click **Next**.
 - Drag another nonmeasure item to the **Analysis variable 2** drop zones. Click **Next**.
 - Drag an item to the **Cases variable** drop zone to define a set of [cases](#). Click **Next**.
6. Specify the crosstab statistics that you want to show. Click **Finish**.

The default is to show the observed count and the case processing summary. You can also show the expected count and the percentage of row, column, and total for each category, as well as the residual, the standardized residual, and the adjusted residual.

7. Run the report.

You see three tables:

- A case processing summary.
- A contingency table showing the frequency distribution.
- A table showing the statistics for chi-square, likelihood ratio, association, and the number of valid cases.

The chi-square statistic is determined by both the results of the cells in the contingency table and the [degrees of freedom](#).

The likelihood ratio is based on maximum likelihood estimation, and is usually similar to the chi-square results.

Valid cases are all cases that are not missing data, as described in the case processing summary.

See the following example for a simple demonstration.

Example: Using a Two-way Chi-Square Test to Compare Default Rates and Level of Education

You are a report author at a bank. You use a two-way chi-square test to determine whether level of education is related to the rate of default.

The null hypothesis is that the two variables are independent - in this case, that there is no significant difference in the rate of default for customers with different levels of education. The alternative hypothesis is that the rate of default differs significantly according to the level of education.

Note: You must have IBM Cognos Statistics installed and configured to create this example.

Steps

1. Open the BANKLOAN_CS package in Report Studio.
2. From the File menu, click New.
3. In the New dialog box, click Statistics. Click OK.
4. In the Select Statistics dialog box, expand Nonparametric Tests, and click Two-way Chi-Square. Click OK.
5. Select Two-Way Chi-Square. Click OK.
6. In the statistical object wizard, expand the package in the metadata tree if necessary and drag items to the following drop zones:
 - Drag the Level of education item to the Category 1 drop zone. Click Next.
 - Drag the Previously defaulted item to the Category 2 drop zone. Click Next.
 - Drag the Customer ID item to the Cases variable drop zone. Click Finish.
7. Run the report.

Case Processing Summary

Crosstabulation	Valid		Missing		Total	
	N	Percent	N	Percent	N	Percent
Level of education * Previously defaulted	1500	100.0%	0	.0%	1500	100.0%

Count

Level of education * Previously defaulted Crosstabulation

		Previously defaulted		Total
		No	Yes	
Level of education	High school degree	330	197	527
	Post-undergraduate degree	57	27	84
	Did not complete high school	182	64	246
	Some college	213	120	333
	College degree	170	140	310
Total		952	548	1500

Chi-Square Tests

	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	22.585 ⁽¹⁾	4	.000
Likelihood Ratio	22.950	4	.000
Linear-by-Linear Association	2.500	1	.114
N of Valid Cases	1500		

¹ 0 cells (.0%) have expected count less than 5. The minimum expected count is 30.69.

The statistic of interest is the Pearson Chi-Square **significance**. The p-value is .000, rounded to three decimal places. Because the significance <.05, you reject the null hypothesis. You conclude that there is a significant difference between the default rates of customers with different levels of education.

Asymptotic significance means significance that is most accurate with large samples and less accurate with small samples.

Basic Correlation and Linear Regression

Correlation and regression analysis let you examine relationships between variables.

Basic correlation is a measure of association between two variables. The existence of a correlation does not imply causality, but simply helps you to understand the relationship.

Linear regression, on the other hand, examines the relationship between one dependent variable and one or more independent variables. You can use linear regression to predict the dependent variable when the independent variables are known.

Showing Correlation Between Variables

You can use correlation to measure the strength of a relationship between two variables using the correlation coefficient.

The correlation coefficient can range from -1 to 1 , where -1 or 1 indicates a perfect relationship. The farther the coefficient is from 0 , either positive or negative, the stronger the relationship. For example, a coefficient of $.494$ is exactly as strong as a coefficient of $-.494$.

Positive coefficients indicate a direct relationship, that is, when one variable increases, the other increases. Negative coefficients indicate an inverse relationship, that is, when one variable increases, the other one decreases.

If you square the coefficient, you get the R square statistic. Multiplied by 100 , this statistic tells you the percentage of the variation in the dependent variable that is explained by the independent variable. For an example of this, see "[Example: Show the Correlation Between Age and Household Income](#)" (p. 275).

Tip: The Catalog Sales report is based on correlation. For more information about IBM® Cognos® Statistics samples, see "[Sample Reports and Packages](#)" (p. 567).

Notes

- You must have IBM Cognos Statistics installed and configured to create statistical objects.

Steps

1. From the File menu, click New.
If you want to change the package, click the ellipsis (...) and click a different package.
2. In the New dialog box, click **Statistics**. Click OK.
3. In the **Statistics** dialog box, expand **Correlation and Regression** and click **Basic Correlation**. Click OK.
4. In the statistical object wizard, expand the package in the metadata tree if necessary and drag items to the following drop zones:
 - Drag a measure item to the **Analysis variable 1** drop zone. Click Next.
 - Drag another measure item to the **Analysis variable 2** drop zone. Click Next.
 - Drag an item to the **Cases variable** drop zone to define a set of [cases](#). Click Next.
5. Specify the correlation options, such as
 - which correlation coefficient to use
 - which statistics to display
 - how to treat [missing values](#)

- the significance option
- whether to show a scatterplot

For more information, see "[Correlation Options](#)" (p. 277).

6. Click **Next**.
7. You can specify output options, such as titles or footnotes. Otherwise, click **Finish**.
8. Run the report.

By default, you see a scatterplot, a statistics table, and a correlations table.

See the following example for a simple demonstration.

Example: Show the Correlation Between Age and Household Income

You are a report author at a bank. You are asked to show if household income is related to age and, if so, how strongly.

The null hypothesis is that there is no relationship between the two variables.

Note: You must have IBM Cognos Statistics installed and configured to create this example.

Steps

1. Open the BANKLOAN_CS package in IBM Cognos Report Studio.
2. In the **Welcome** dialog box, click **Create a new report or template**.
3. In the **New** dialog box, click **Statistics** and then click **OK**.
4. In the **Select Statistic** dialog box, expand **Correlation and Regression**, click **Basic Correlation**, and then click **OK**.
The statistical object wizard opens.
5. In the statistical object wizard, expand the BANKLOAN_CS package in the metadata tree if necessary and drag items to the following drop zones:
 - Drag the Age in years item to the **Analysis variable 1** drop zone. Click **Next**.
 - Drag the Household income in thousands item to the **Analysis variable 2** drop zone. Click **Next**.
 - Drag the customer ID item to the **Cases variable** drop zone to define a set of **cases**. Click **Next**.
6. Leave the default correlation options. Click **Finish**.
7. Run the report.

You see a scatter chart, a descriptive statistics table, and a correlations table.

First, look at the correlations table to find the statistic of interest in this case, the correlation coefficient.

Descriptive Statistics			
	Mean	Std. Deviation	N
Age in years	34.17	13.142	1500
Household income in thousands	59.59	67.130	1500
Correlations			
		Age in years	Household income in thousands
Age in years	Pearson Correlation	1	.494(**)
	Sig. (2-tailed)		.000
	N	1500	1500
Household income in thousands	Pearson Correlation	.494(**)	1
	Sig. (2-tailed)	.000	
	N	1500	1500

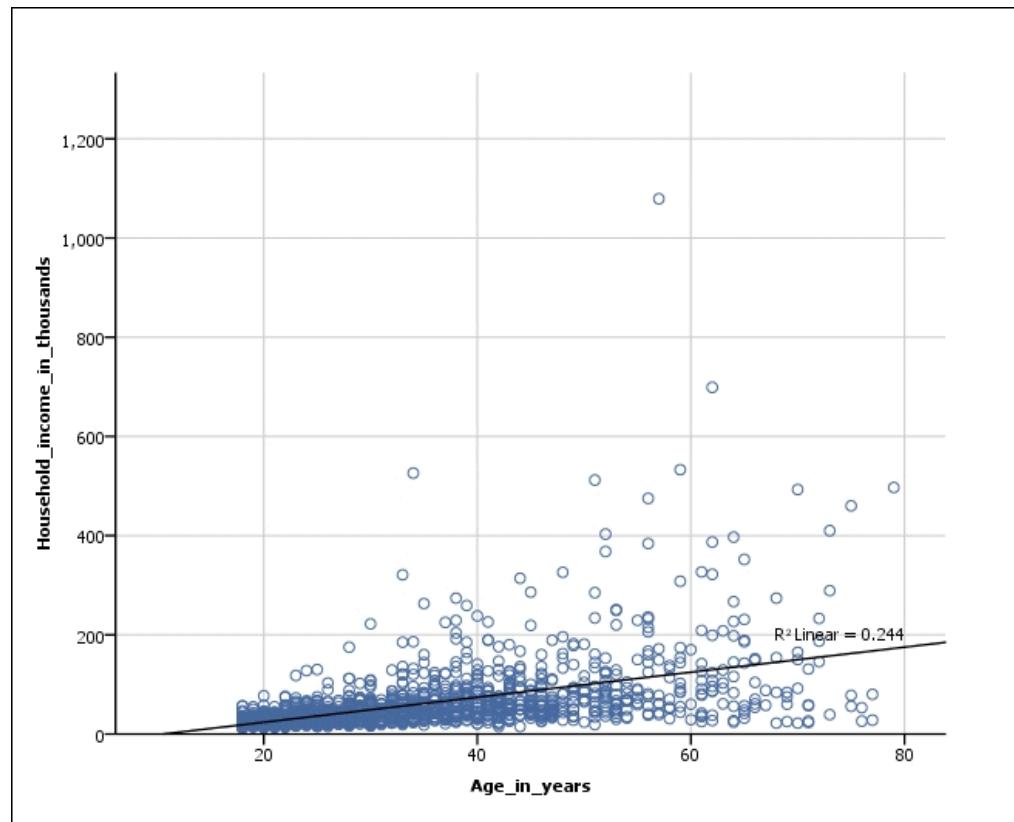
** Correlation is significant at the 0.01 level (2-tailed).

The statistic of interest is the Pearson correlation coefficient. This statistic, .494, shows a positive relationship between age in years and household income.

The p-value for 2-tailed significance is .000, rounded to three decimal places.

Because the significance <.05, you reject the null hypothesis. You conclude that there is a significant relationship between age in years and household income, and that the relationship is positive.

To view the relationship graphically, look at the scatterplot.



Note the R square statistic (0.244) listed to the right of the graph. In this case, the Pearson coefficient is .494. If you square the Pearson coefficient (.494 * .494), you get .244 – the same as the R square statistic. Multiplied by 100, this statistic tells you the percentage of the variation in the household income variable that is explained by the Age in years variable.

You can then conclude that age explains 24.40% of the variation in household income in our sample.

Correlation Options

Aside from specifying how to treat [missing values](#), you can specify the following options in your report.

Correlation Coefficients

Correlations coefficients measure how variables are related. The coefficient that you use depends on the type of data with which you are working.

Coefficient	Description
Pearson's	Measures the strength of the linear relationship between two scale (ratio or interval) variables. This common coefficient is the default.

Coefficient	Description
Kendall's Tau b	<p>Measures the strength of the relationship between ordinal data. This coefficient is most appropriate for square tables, that is, tables that contain the same number of rows and columns.</p> <p>Use this coefficient for nonparametric data.</p>
Spearman	<p>Uses the same calculations as Pearson's, but uses ranks of scores instead of actual values. Use when one or both variables are ordinal and not scalar, or if there are outlying values.</p> <p>Use this coefficient for nonparametric data.</p>

Use Pearson's if there is a linear relationship. Use Kendall's Tau b or Spearman for ranked data. You can also choose whether to show a [one-tailed](#) or [two-tailed](#) test of significance.

Use Linear Regression

Linear regression displays the regression coefficients of the linear equation involving one or more independent variables that best predict the value of the dependent variable. When there is more than one independent variable, this is known as multiple regression.

For example, you can explain how much of a salesperson's total yearly sales (the dependent variable) are due to independent variables such as education and years of experience (the independent variables).

You can use different regression models. A regression model relates y (the dependent variable) to a function of x (the independent variables) and a set of unknown parameters.

Tip: The Advertising Costs on Sales Revenue report is based on multiple regression. For more information about IBM Cognos Statistics samples, see "[Sample Reports and Packages](#)" (p. 567).

Note

- You must have IBM Cognos Statistics installed and configured to create statistical objects.

Steps

1. From the File menu, click New.
If you want to change the package, click the ellipsis (...) and click a different package.
2. In the Welcome dialog box, click **Create a new report or template**.
3. In the New dialog box, click **Statistics**. Click **OK**.
4. In the Select Statistic dialog box, expand **Correlation and Regression** and click **Linear Regression**. Click **OK**.
5. In the statistical object wizard, expand the package in the metadata tree if necessary and drag items to the following drop zones:

- Drag an item to the **Dependent variable** drop zone. Click **Next**.
 - Drag one or more items to the **Independent variables** drop zone. Click **Next**.
If you specify more than one variable, you are performing multiple regression.
 - Drag an item to the **Cases variable** drop zone to define a set of **cases**. Click **Next**.
6. Select an **regression method** and how to treat **missing values**. Click **Next**.
 7. Select your desired output options.
For more information, see "[Linear Regression Options](#)" (p. 282).
 8. Click **Finish**.
 9. The default output includes a variable entry table, a model summary, an ANOVA table, and a coefficients table.
The model summary table appears even if you select no other output options. The statistic of interest in the model summary table is the R square value. R square is the percent of the dependent variable that the independent variables can explain.
The coefficients table gives you predictive information. It shows you how to predict the dependent variable from the value of the independent variables.
See the following example for a simple demonstration.

Example: Use Linear Regression to Analyze the Relation Between Age and Household Income

You are a report author at a bank. You are asked to analyze the relationship between age and household income.

The null hypothesis is that there is no relationship between the two variables.

Note: You must have IBM Cognos Statistics installed and configured to create this example.

Steps

1. Open the BANKLOAN_CS package in Report Studio.
2. In the **Welcome** dialog box, click **Create a new report or template**.
3. In the **New** dialog box, click **Statistics** and then click **OK**.
4. In the **Select Statistic** dialog box, expand **Correlation and Regression**, click **Linear Regression**, and then click **OK**.
The statistical object wizard opens.
5. From the metadata tree, expand BANKLOAN_CS and drag items to the following drop zones:
 - Drag the Household income in thousands item to the **Dependent variable** drop zone.
 - Drag the Age in years item to the **Independent variables** drop zone.
 - Drag the Customer ID item to the **Cases variable** drop zone to define a set of **cases** and click **Next**.

6. Leave the default entry method and missing values options, and click **Next**.
7. Leave the default output options, and click **Finish**.
8. Run the report.

By default, you see a variable entry table, a model summary, an ANOVA table, and a regression coefficients table.

In the model summary table, the statistic of interest is the R square statistic, in this case .244.

- The coefficient of determination (R square) is the square of the correlation coefficient (R). It represents the proportion of variance in the dependent variable that can be accounted for by the regression equation. For example, an R square value of .24 means that the regression equation can explain 24% of the variance in the dependent variable. The other 76% is unexplained.
- Adjusted R square is a standard downward adjustment to correct for the possibility that, if there are many independent variables, some of the variance might be due to chance. The more independents, the greater the adjustment.
- The standard error of the estimate is the standard deviation of the data points as they are distributed around the regression line.
- The R square change refers to the amount R square increases or decreases when a variable is added to or deleted from the equation, as is done in stepwise regression.
- The F change statistic shows the significance level associated with adding or removing the variable for each step. You can change this in the [regression method](#) area of the wizard. Steps that are not significant are not modeled.

Variables Entered/Removed⁽²⁾

Model	Variables Entered	Variables Removed	Method
1	Age in years ⁽¹⁾		Enter

¹ All requested variables entered.² Dependent Variable: Household income in thousands**Model Summary**

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Change Statistics				
					R Square Change	F Change	df1	df2	Sig. F Change
1	.494 ⁽¹⁾	.244	.243	58.388	.244	483.473	1	1498	.000

¹ Predictors: (Constant), Age in years**ANOVA⁽²⁾**

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1648242.793	1	1648242.793	483.473	.000 ⁽¹⁾
	Residual	5106938.414	1498	3409.171		
	Total	6755181.207	1499			

¹ Predictors: (Constant), Age in years² Dependent Variable: Household income in thousands

The variables table shows the variables that have been included in the analysis and the regression method that is used to enter the variables.

The ANOVA table tests the acceptability of the model. The Regression row displays information about the variation accounted for by your model. The Residual row displays information about the variation that is not accounted for by your model. If the significance value of the F statistic is less than 0.05, then the variation that the model explains is not due to chance.

Next, look at the coefficients table.

Coefficients⁽¹⁾

Model	Unstandardized Coefficients		t	Sig.	95.0% Confidence Interval for B		Standardized Coefficients	
	B	Std. Error			Lower Bound	Upper Bound	Beta	Std. Error
1	(Constant)	-26.636	4.201	-6.340	.000	-34.877	-18.396	
	Age in years	2.523	.115	21.988	.000	2.298	2.748	.494

¹ Dependent Variable: Household income in thousands

The main statistic of interest in the coefficients table is the unstandardized regression coefficient, Age in years 2.523.

The regression equation is

dependent variable = slope * independent variable + constant

The slope is how steep the regression line is, based on a scatterplot. The constant is where the regression line strikes the y-axis when the independent variable has a value of 0.

In this example, the slope is 2.523, and the constant is -26.636. So the regression equation is
predicted value of household income = $2.523 * \text{age in years} - 26.636$.

That is, for the average person, we would estimate that their household income at age 30 would be

$$2.523 * 30 - 26.63 = 49.06 \text{ (in thousands)}$$

Note: For multiple regression, the regression equation is similar. If you have 3 independent variables (IV1, IV2, and IV3, the regression equation is

dependent variable = $B(\text{IV1}) * \text{IV1} + B(\text{IV2}) * \text{IV2} + B(\text{IV3}) * \text{IV3} + \text{constant}$

The coefficients table also includes the [confidence interval](#) for B and the standardized coefficients.

Linear Regression Options

Aside from how to treat [missing values](#) and whether to show a table of [descriptive](#) statistics for the variables, you can specify the following options in your report.

Regression Methods

You can specify how independent variables are entered into the analysis. Using different methods, you can construct a variety of regression models from the same set of variables.

In multiple regression, the variable elimination criteria options apply when you specify the forward, backward, or stepwise variable selection method. You can include or remove variables from the model depending on the significance (probability) of the F value that is shown in the ANOVA table.

Therefore, 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.

Method	Description
Enter	Enters all variables in a single step.

Method	Description
Stepwise	<p>At each step, enters the independent variable that is not in the equation and has the smallest probability of F, if that probability is sufficiently small.</p>
	<p>Variables already in the regression equation are removed if their probability of F becomes sufficiently large.</p>
	<p>The method terminates when no more variables are eligible for inclusion or removal.</p>
Forward	<p>A stepwise variable selection procedure in which variables are sequentially entered into the model.</p>
	<p>The procedure stops when there are no variables that meet the entry criterion.</p>
Backward	<p>All variables are entered into the equation and then sequentially removed.</p>
	<p>The procedure stops when there are no variables that meet the removal criterion.</p>

Customized Output

You can specify what to show in your output.

- Show R square change

Use this to show the change in the R square statistic that is produced by adding or deleting an independent variable. If the R square value change associated with a variable is large, the variable is a good predictor of the dependent variable.

- Show ANOVA table

Use this to show an ANOVA table that provides information about levels of variability within a regression model and forms a basis for tests of significance.

- Show coefficients

Use this to show the regression coefficient B, as well as its standard error and specified level of confidence. Also shows the t value for B, the significance level of t, and standardized coefficients (beta).

In a multiple regression analysis, standardized coefficients help to determine which of the independent variables have a greater effect on the dependent variable when the variables have different units of measurement, such as income measured in dollars and age measured in years.

- Show variable processing summary

Use this to show the variables that are included in the analysis and the entry method used to enter the variables.

- Show descriptives

Use this to show the number of valid cases, the mean, and the standard deviation for each variable in the analysis.

- Show correlations

Use this to show the Pearson correlation, the 1-tailed significance, and the sample size (N) for each variable.

- Show part and partial correlation

In multiple regression, part correlation displays the correlation between the dependent variable and an independent variable when the linear effects of the other independent variables in the model have been removed from the independent variable.

It is related to the change in R square when a variable is added to an equation. Sometimes called the semipartial correlation.

Partial correlation shows the correlation that remains between two variables after removing the correlation that is due to their mutual association with the other variables.

- Show covariance matrix

Use this to show a variance-covariance matrix of regression coefficients with covariances off the diagonal and variances on the diagonal. Also shows a correlation matrix.

- Show confidence interval

Use this to show the confidence interval that you specify.

- Show model selection statistics

Use this to show model selection statistics based on the Akaike information criterion, the Amemiya prediction criterion, Mallows' prediction criterion, and the Schwarz Bayesian criterion.

- Show excluded variable statistics

In multiple regression, shows the betas and other coefficients related to variables that are not included in the model for the step listed.

Beta in is the beta weight that would result if the variable were put back into the model for the listed step. Likewise, t, significance, tolerance, and VIF (variance inflation factor), are the coefficients that would result from adding that variable back in. A variance inflation factor greater than 2 is usually considered problematic.

The excluded variable with the largest partial correlation is usually the best candidate to add back in.

- Show collinearity diagnostics

In multiple regression, displays a table that assesses if there is too much collinearity in the model. Collinearity (or multicollinearity) is the undesirable situation when one independent variable is a linear function of one of the other independent variables.

Crossproducts of the independent variables are factored in a matrix. High eigenvalues indicate dimensions that account for a large amount of the variance in the matrix. Eigenvalues close to

0 indicate that the predictors are highly intercorrelated and that small changes in the data values might lead to large changes in the estimates of the coefficients.

The condition indices are computed as the square roots of the ratios of the largest eigenvalue to each successive eigenvalue.

A condition index over 15 indicates a possible collinearity problem. A condition index over 30 suggests a serious collinearity problem. If a factor of the crossproduct matrix has a high condition index, look in the variance proportions column to see if it accounts for a large proportion of variance in two or more variables. If so, collinearity is a problem.

- Show residuals statistics

Use this to show summary data regarding the residuals, that is, the difference between each individual value in the sample and the sample mean, and the standardized residuals.

A standardized residual is the raw residual value divided by the square root of the residual mean square.

- Show standardized residuals histogram

Use this to show a histogram of the residual to check whether the variance is normally distributed.

A symmetric histogram that is evenly distributed around zero indicates that the underlying assumption of normality for a model is likely to be true.

- Show standardized residuals PP plots

The PP plot is another test of normally distributed residual error.

Under perfect normality, the plot would be a 45-degree line.

- Show scatterplot of residuals versus dependent variables

Use this to show the difference between the calculated and measured values of the dependent variable as a function of the measured values.

If the regression model represents the data correctly, the residuals are randomly distributed around the horizontal line.

- Show partial regression plot

Use this to show the effect of adding an additional variable to a model that already contains one or more independent variables.

Create a Curve Estimate

You can use curve estimation to plot a curve through a set of points to examine the relationship between one independent variable and one or more dependent variables.

The aim of curve estimation is to find the best fit for your data, expressed as the correlation coefficient R square. In general, the closer the R square value is to 1.00, the better the fit.

Notes

- You must have IBM Cognos Statistics installed and configured to create statistical objects.

Steps

1. From the File menu, click **New**.
If you want to change the package, click the ellipsis (...) and click a different package.
2. In the **Welcome** dialog box, click **Create a new report or template**.
3. In the **New** dialog box, click **Statistics** and then click **OK**.
4. In the **Select Statistic** dialog box, expand **Correlation and Regression**, click **Curve Estimation**, and then click **OK**.
5. In the statistical object wizard, expand the package in the metadata tree if necessary and drag items to the following drop zones:
 - Drag an item to the **Dependent Variables** drop zone.
 - Drag an item to the **Independent Variable** drop zone.
 - Drag an item to the **Cases variable** drop zone to define a set of [cases](#). Click **Next**.
6. Select the [curve models](#) to use. The default model is linear. Click **Next**.
7. Select your desired output [options](#). Click **Next**.
8. Run the report.

The default output includes a scatterplot and model summary tables, ANOVA tables, and regression coefficients for each of the curve models selected.

See the following example for a simple demonstration.

Example: Create a Curve Estimation

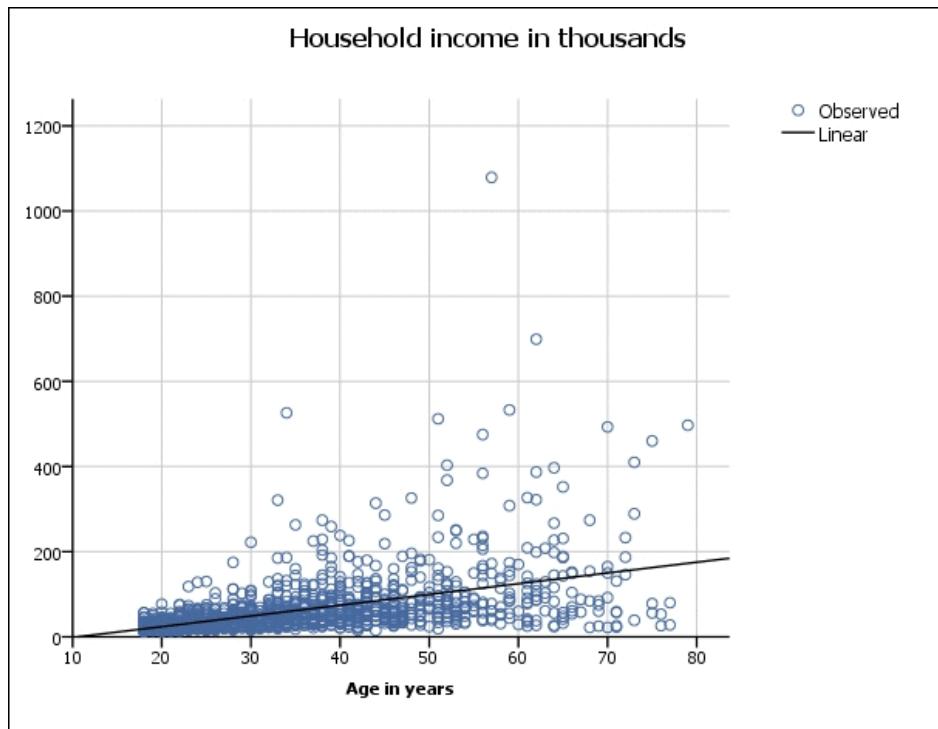
You are a report author at a bank. In this scenario, you analyze whether household income increases with a customer's age and observe if there is a linear relationship between these two variables.

Steps

1. Open the BANKLOAN_CS package in Report Studio.
2. In the **Welcome** dialog box, click **Create a new report or template**.
3. In the **New** dialog box, click **Statistics** and then click **OK**.
4. In the **Select Statistic** dialog box, expand **Regression**, click **Curve Estimation**, and then click **OK**.
5. In the statistical object wizard, expand the BANKLOAN_CS package in the metadata tree if necessary and drag items to the following drop zones:
 - Drag the Household income in thousands item to the **Dependent Variables** drop zone. Click **Next**.

- Drag the Age in years item to the **Independent Variable** drop zone. Click **Next**.
 - Drag the Customer ID item to the **Cases** drop zone to define a set of **cases**. Click **Next**.
6. Leave linear as the default **model**. Click **Next**.
 7. Leave the default output options. Click **Finish**.
 8. Run the report.

The default output includes a scatterplot, model summary table, ANOVA table, and coefficients table.



The scatterplot suggests that the linear model is a good fit for your data. Look at the model summary table. How close is the R square value to 1.00?

You can try using different models with your data to help you find the model with the optimum fit.

Curve Estimation Options

You can specify the following options in your report.

Include Constant in Equation

By default, the regression model includes a constant term. Deselecting this option forces regression through the origin. Results of regression through the origin are not comparable to results of regression that do include a constant. For example, the R square statistic cannot be interpreted in the usual way.

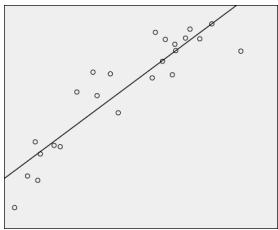
Curve Models

You can choose one or more curve estimation regression models. First, view a scatterplot of your data. If your variables appear to be related linearly, use a simple linear regression model. If the plot resembles a mathematical function you recognize, fit your data to that type of model. For example, if your data resembles an exponential function, use an exponential model.

Linear

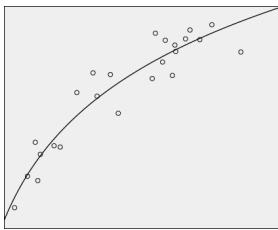
A model whose equation is $Y = b_0 + (b_1 * X)$, where b_0 is the constant, b_1 the regression coefficient for x , the independent variable.

The series values are modeled as a linear function of X .



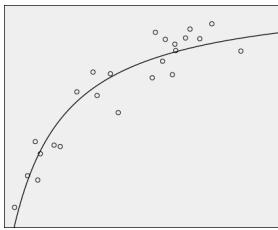
Logarithmic

A model whose equation is $Y = b_0 + (b_1 * \ln(X))$, where $\ln()$ is the natural log function.



Inverse

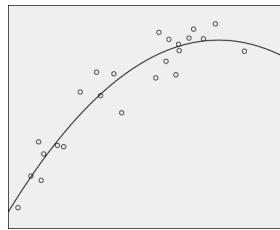
A model whose equation is $Y = b_0 + (b_1 / X)$.



Quadratic

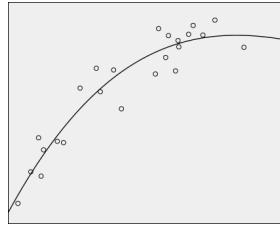
A model whose equation is $Y = b_0 + (b_1 * X) + (b_2 * X^{**2})$, where ** is the exponentiation operator. If b_2 is positive, the slope is upward; if negative, downward.

You can use the quadratic model to model a series that "takes off" or a series that dampens.



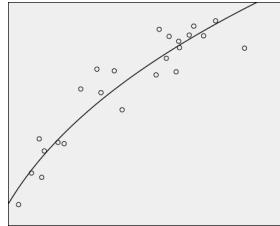
Cubic

A model that is defined by the equation $Y = b_0 + (b_1 * x) + (b_2 * X^{**2}) + (b_3 * X^{***3})$. If b_3 is positive, the slope is upward; if negative, downward.



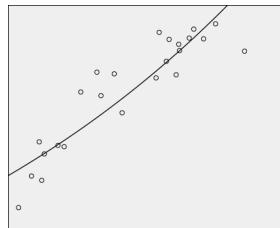
Power

A model whose equation is $Y = b_0 * (X^{**b_1})$ or $\ln(Y) = \ln(b_0) + (b_1 * \ln(X))$. If b_0 is positive, the slope is upward; if negative, downward.



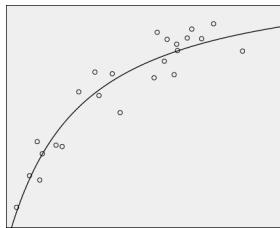
Compound

A model whose equation is $Y = b_0 * (b_1^{**}X)$ or $\ln(Y) = \ln(b_0) + (\ln(b_1) * X)$. If b_0 is positive, the slope is upward; if negative, downward.



S

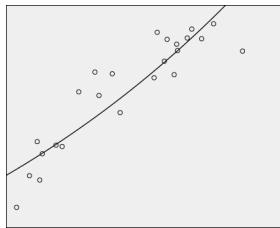
A model whose equation is $Y = e^{**}(b_0 + (b_1/X))$ or $\ln(Y) = b_0 + (b_1/X)$, where e is the base of the natural logarithm. If b_1 is positive, the slope is upward; if negative, downward.



Logistic

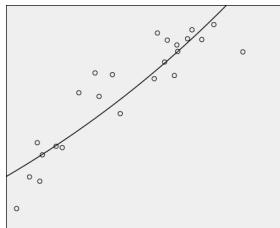
A model whose equation is $Y = 1 / (1/u + (b0 * (b1**X)))$ or $\ln(1/y-1/u) = \ln(b0) + (\ln(b1) * X)$ where u is the upper boundary value. If b1 is negative, the slope is upward; if positive, downward.

After selecting Logistic, specify the upper boundary value to use in the regression equation. The value must be a positive number that is greater than the largest dependent variable value.



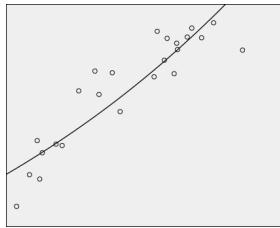
Growth

A model whose equation is $Y = e^{**}(b0 + (b1 * X))$ or $\ln(Y) = b0 + (b1 * X)$. If b1 is negative, the slope is downward; if positive, upward.



Exponential

A model whose equation is $Y = b0 * (e^{**}(b1 * X))$ or $\ln(Y) = \ln(b0) + (b1 * X)$. If b0 is negative, the slope is downward; if positive, upward.



Customized Output

You can specify what to include in your output.

Statistic	Description
Show case processing summary	Shows the number of total cases, as well as any excluded cases, forecasted cases, and newly created cases. Cases are excluded from the analysis when there is a missing value in any variable.
Show variable processing summary	Shows the number of positive, negative, and missing values. Also shows the number of zeros.
Show chart	Shows a scatterplot.
Show model summary	Shows R, the R square value and adjusted R square value, and the standard error of the estimate.
Show ANOVA table	Shows an ANOVA table that provides information about levels of variability within a regression model and forms a basis for tests of significance.
Show coefficients	Shows the regression coefficient B, standard error of B, confidence intervals with the specified level of confidence for B, t value for B, the two-tailed significance level of t, and standardized coefficient beta.

Statistical Process Control

All processes show variation, but excessive variation can produce undesirable or unpredictable results.

You use statistical process control (SPC) to monitor critical manufacturing and other business processes that must be within specified limits. For example, a control chart can inform you if samples from a shampoo manufacturing process exceed a specified pH level.

The basic steps of statistical process control include

- measuring the process
- monitoring the process
- finding and eliminating undesirable variations
- continuously improving the process

You can use [control charts](#) to monitor the variations in a process.

You can use [specification limits](#) to determine if your product conforms to specifications.

You can specify [control rules](#) to highlight when a process is going out of control.

Control Charts

Control charts plot samples of your process output collected over time to show you whether a process is in control or out of control. The type of control chart that you choose depends on the type of output that your process produces and the sampling plan that you use to collect the data.

Control Charts for Variables

Use one of the following charts for variable data, that is, data that you can measure, such as temperature, diameter, and so on. You use these charts to compare actual observations to a target value to see if your product meets specifications.

In situations where you take several measurements, then you should use an X-bar chart with either the R chart or S chart.

Chart	Description
X-bar charts	Plot the average of each subgroup. An X-bar chart is often accompanied by either the R chart or S chart.
R charts	Plot the data range for each subgroup.
S charts	Plot the data standard deviation for each subgroup.

In situations where you take a single measurement at each collection period, you should use an individuals or moving range chart.

Chart	Description
Individuals charts	Plot individual observations that cannot be put into subgroups, such as single-measurement samples.
Moving range charts	Plot the spread of measures in terms of the differences between consecutive samples.

For variable data, your data should be set up as in the table below.

ID	Time of measurement	Batch	Shift	pH level
1	1	1	day	4.98
2	1	2	day	4.65
3	1	3	day	4.91
4	1	4	day	4.62
5	1	5	day	4.84

ID	Time of measurement	Batch	Shift	pH level
6	1	6	day	4.69
7	2	1	day	4.52
8	2	2	day	4.73
9	2	3	day	5.05

To see how a control chart plots this data, look at the X-bar chart example. You can see that the first data marker in the chart represents the mean pH level (4.78) for the 6 batches that make up a sample for the subgroup variable Time of measurement (1).

Likewise, if you look at the R bar chart in the example, you can see that the first data marker represents the range (subtract the smallest value 4.62 from the largest value 4.98) for the 6 batches that make up a sample for the subgroup variable Time of measurement (1).

Control Charts for Attributes

Use one of the following charts for attribute data, such as the number of defective parts or late shipments. You can track defects using different methods.

In situations where you are tracking whole units that are defective, you should use a p chart or np chart.

Chart	Description
p chart	Plots the percentage of defective units, such as the percent of automobiles with defects per shift. Sample sizes do not need to be equal. They can vary between collection periods.
np chart	Plots the number of defective units, such as the number of automobiles with defects per shift. Sample sizes must be equal.

In situations where you are tracking the number of defects per unit, you should use a c chart or u chart.

Chart	Description
u chart	Plots the number of defects per unit, such as the number of defects per automobile per shift. Sample sizes do not need to be equal. They can vary between collection periods.

Chart	Description
c chart	Plots the number of defects, such as the total number of defects per shift. Sample sizes must be equal.

For attribute data, your data should be set up similar to the table below. Because the sample sizes are not equal in this case, you would have to use a p chart or a u chart.

ID	Lot	Sampled	Defects
1	1.00	193.00	6.00
2	2.00	198.00	1.00
3	3.00	211.00	1.00
4	4.00	210.00	.00
5	5.00	204.00	2.00
6	6.00	214.00	1.00
7	7.00	208.00	1.00
8	8.00	210.00	3.00
9	9.00	198.00	2.00

Create X-Bar, R, and S Charts

You can create X-bar, R, and S charts to chart [variable data](#) where the sample size is a made up of several measurements.

For example, you might use X-bar charts to monitor the variation in the lengths of cut material. If you detect a shift towards a control limit, this might indicate that the cutting tool is wearing and must be replaced.

X-bar charts are often used in conjunction with S charts and R charts.

R charts plot range values by subtracting the smallest value in a subgroup from the largest value in the same subgroup. The center line on the chart represents the mean of the ranges of all the subgroups.

S charts plot the standard deviations for each subgroup. The center line on the chart represents the mean of the standard deviations of all the subgroups.

Note

- You must have IBM® Cognos® Statistics installed and configured to create statistical objects.

Steps

1. From the File menu, click New.

If you want to change the package, click the ellipsis (...) and click a different package.

2. In the New dialog box, click Statistics and then click OK.
3. In the Select Statistic dialog box, expand Control Charts, click X-bar, R, or S and then click OK.
4. From the metadata tree, expand the package if necessary and drag items to the following drop zones:
 - Drag an item to the Process measurement variables drop zone to represent your measure of interest. Click Next.
 - Drag an item to the Subgroups variable drop zone to represent the time or sample lot. Click Next.
 - Drag an item to the Cases variable drop zone to define a set of [cases](#). Click Next.
5. Specify the number of standard deviations and the minimum subgroup size.
The number of standard deviations determines the control limit bands.
The default settings are ± 3 standard deviations and a minimum subgroup size of 2.
Select **Display subgroups defined by missing values** to display each missing value for the subgroup variable as a separate group in the chart.
6. For X-bar charts, you can specify the following:
 - Specify [control rules](#) to identify out-of-control points. Click Next.
 - Specify the [specification limits](#) to define the upper and lower specification limits for the process measurement variable.
 - Specify any desired [process capability indices](#).
 - Specify any desired [process performance indices](#). Click Next.
7. For X-bar charts, R charts, and S charts, specify any output options such as titles or footnotes. Click Finish.
8. Run the report.

See the following example for a simple demonstration.

Example: Create a Report Showing Shampoo Manufacturing Quality Control

You are a report author at a shampoo manufacturer. You create an R chart and an X-bar chart to see if the manufacturing process runs according to the specification limits.

During each shift, you record pH levels for separate output batches at regular time intervals. You set the [specification limits](#) to allow a fluctuation of 4.8-5.2 in the pH level, with a target value of 5.

Steps

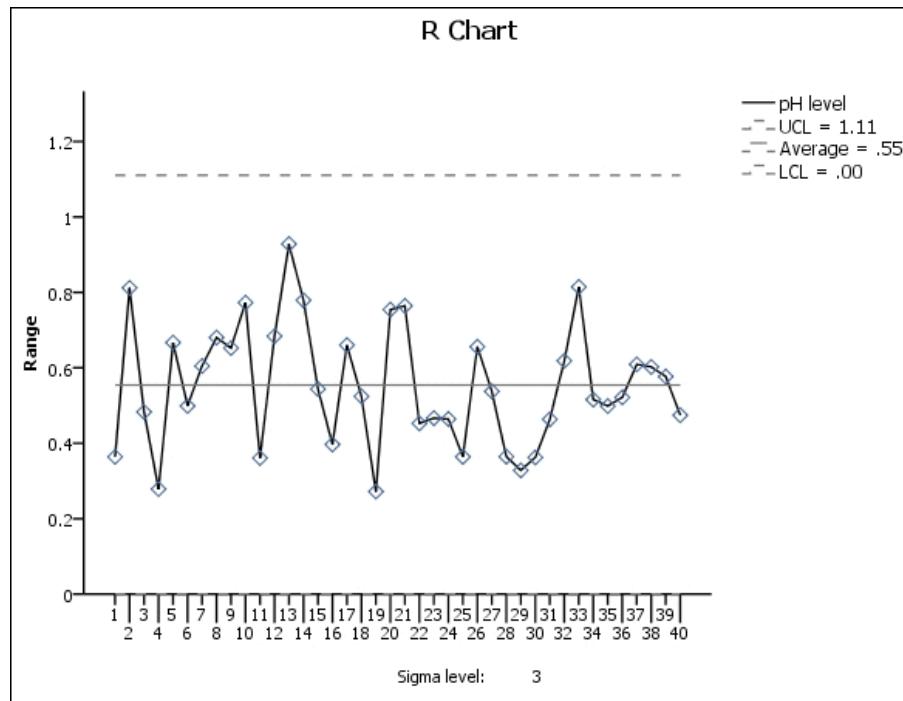
1. Open the SHAMPOO_PH package in IBM Cognos Report Studio.
2. In the **Welcome** dialog box, click **Create a new report or template**.
3. In the **New** dialog box, click **Statistics** and then click **OK**.
4. In the **Select Statistic** dialog box, expand **Control Charts**, click **R**, and then click **OK**.
5. From the metadata tree, expand the SHAMPOO_PH package if necessary and drag items to the following drop zones:
 - Drag the pH level item to the **Process measurement variables** drop zone. Click **Next**.
 - Drag the Time of measurement item to the **Subgroups variable** drop zone to represent the sequential parameter. Click **Next**.
 - Drag the ID item to the **Cases variable** drop zone to define a set of [cases](#).
6. Leave the number of standard deviations at 3 and the minimum subgroup size as 2.
7. Click **Finish**.

Now you want to add an X-bar chart to the report.

8. In the **Insertable objects** pane, click on the toolbox tab. Scroll down to **Statistics**.
9. Drag a statistical object onto the work area.
10. Repeat steps 4 through 6 to create an X-bar chart, with the following specifications:
 - Leave the default [control rules](#) and click **Next**.
 - Specify the process upper specification limit as 5.2, the lower specification limit as 4.8, and the specification target as 5.0.
 - Select CP and CpU as [process capability indices](#).
 - Select PP and PpU as [process performance indices](#). Click **Next**.
 - Type **Shampoo Quality Measurements** as a report title. Click **Finish**.
11. Run the report.

The output includes the R chart and X-bar chart, as well as tables containing rule violations and process statistics.

Examine the R chart first. The R chart uses sample ranges to monitor changes in the spread of a process. The sample ranges are plotted as points on the chart and connected with lines. The solid centerline is the average of the sample ranges. The two dashed lines are the upper control limit (UCL) and the lower control limit (LCL). Note that the value of the LCL, 0.0, is not visible because it is flush with the horizontal axis.

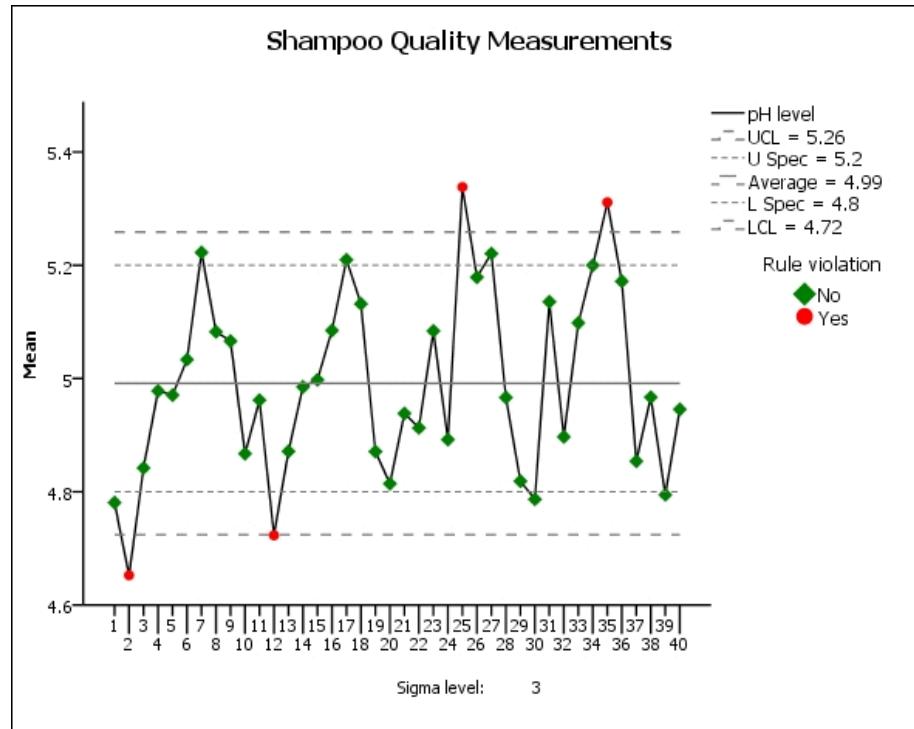


The control limits are calculated so that if the process is in control, nearly all of the sample points will fall between the limits. Control limits reflect the expected amount of variation in the sample ranges when only common causes of variation are present.

All the points in the R chart are within the control limits. The process variability is in control. For a different view - that of the process itself, rather than the variability - you look at the X-bar chart.

If an R chart indicates that the process variability is out of control, then you should disregard the X-bar chart. The average of the sample ranges is used in the calculation of the control limits for the X-bar chart. So if the R chart does not display control, the control limits on the X-bar chart are questionable. Because the process variability appears to be in control, you now examine the X-bar chart.

The X-bar chart below shows the results for a specified time period.



The X-bar chart uses sample means to monitor changes in a process. The sample means (or averages) are plotted as points and connected with lines. The solid centerline is the average of the sample means. The two inner dashed lines are the upper control limit (UCL) and the lower control limit (LCL). The control limits reflect the expected amount of variation in the sample means when only common causes of variation are present.

To see how this data is set up, see the section Control Charts for Variables in "[Control Charts](#)" (p. 292).

The average is 4.99, which is on target, but this plot shows that certain points in the process are out of control.

The Rule Violations table shown below displays points that violate any rule. The table uses the subgroups variable to identify the points. In this case, the subgroups variable is Time of measurement.

Rule Violations

4 points violate control rules.

Time of measurement	Violations for Points
2	Less than -3 sigma
12	Less than -3 sigma
25	Greater than +3 sigma
35	Greater than +3 sigma

Process Statistics

The normal distribution is assumed. LSL = 4.8 and USL = 5.2.

Process Statistics		
Capability Indices	CP ⁽¹⁾	.305
	CpU ⁽¹⁾	.318
Performance Indices	PP	.261
	PpU	.272

¹ The estimated capability sigma is based on the mean of the sample group ranges.

In the Process Statistics table, the CP and PP value is less than 1. This indicates that the process shows excess variability. The process variability indices CpU and PpU show that the process behaves in a relatively symmetric way. For more information about these indices, see ["Process Capability Indices" \(p. 304\)](#) and ["Process Performance Indices" \(p. 305\)](#).

Using the X-bar, you determine that certain points in the process are out of control. You might want to investigate the differences in the pH of shampoo that different shifts produce.

Create Individuals and Moving Range Charts

You can create individuals and moving range charts to chart [variable data](#) where the sample size is a single measurement.

Individuals charts plot the measured value of each individual sample. The center line on the chart represents the average of all individual samples in the chart.

Moving range charts plot the difference between each sample value and the preceding sample value. The center line on the chart represents the average change from one sample to another.

If you want to chart variable data for data where the sample size is made up of several measurements, you can use an [x-bar](#), [R](#), or [S](#) chart.

Note

- You must have IBM Cognos Statistics installed and configured to create statistical objects.

Steps

1. From the File menu, click New.

If you want to change the package, click the ellipsis (...) and click a different package.

2. In the New dialog box, click **Statistics**. Click **OK**.
3. In the Select Statistic dialog box, expand **Control Charts** and click **Moving Range or Individuals**. Click **OK**.
4. From the metadata tree, expand the package if necessary and drag items to the following drop zones:
 - Drag an item to the **Process measurement variable** drop zone to represent your measure of interest. Click **Next**.
 - Drag an item to the **Subgroups variable** drop zone to represent the sequential parameter. Click **Next**.
 - Drag an item to the **Cases variable** drop zone to define a set of **cases**. Click **Next**.
5. Specify the number of standard deviations and the minimum subgroup size.
Select **Display subgroups defined by missing values** to display each missing value for the subgroup variable as a separate group in the chart.
In moving range charts, the subgroup size specifies the span that calculates the moving range.
In individuals charts, you can also specify the number of standard deviations, the minimum subgroup size, the **control rules**, and the **specification limits**.
6. Specify any output options such as titles or footnotes. Click **Finish**.
7. Run the report.

Create p,np Charts

You can use a p chart or an np chart to plot **attribute data** where you are tracking whole units that are defective.

Use np charts plot the quantity of units that have defects based on samples of equal size.

Use p charts plot the percent of units that have defects based on samples of equal or unequal size.

Tip: The Clothing Manufacturer Quality report is based on a p chart. For more information about IBM Cognos Statistics samples, see "["Sample Reports and Packages" \(p. 567\)](#)".

Notes:

- You must have IBM Cognos Statistics installed and configured to create statistical objects.
- When there are unequal sample sizes for an np chart, a p chart is recommended.

Steps

1. From the File menu, click **New**.
If you want to change the package, click the ellipsis (...) and click a different package.
2. In the New dialog box, click **Statistics** and then click **OK**.

3. In the Select Statistic dialog box, expand Control Charts, click p, np (Cases are Units) or p, np (Cases are Subgroups) and then click OK.
4. From the metadata tree, expand the package if necessary and drag items to the following drop zones:
 - If using **Cases are Units**, drag an item to the Characteristic variable drop zone. Click Next. Drag an item to the Subgroups variable drop zone. Click Next.
 - If using **Cases are Subgroups**, drag an item to the Sample size variable drop zone. Click Next. Drag an item to the Number of nonconforming variable drop zone. Click Next.
5. Drag an item to the Cases variable drop zone to define a set of **cases**. Click Next.
6. Under Chart, choose either p (Proportion nonconforming) to produce a p chart, or np (Number of nonconforming) to produce an np chart.
If using **Cases are Units**, under Value to count, choose either Conforming or Nonconforming and enter a value.
7. Click Next.
8. Specify the number of standard deviations to show and the **control rules**.
Select **Display subgroups defined by missing values** to display each missing value for the subgroup variable as a separate group in the chart.
9. Click Next.
10. Specify any output options such as titles or footnotes. Click **Finish**.
11. Run the report.

Create c,u Charts

You can create c,u charts to plot **attribute data** where you are tracking the number of defects per unit.

Use c charts to plot the total quantity of defects based on samples of equal size.

Use u charts to plot the defects per unit based on samples of equal or unequal size.

Notes:

- You must have IBM Cognos Statistics installed and configured to create statistical objects.
- When there are unequal sample sizes, a u chart is recommended.

Steps

1. From the File menu, click New.
If you want to change the package, click the ellipsis (...) and click a different package.
2. In the New dialog box, click **Statistics** and then click OK.

3. In the Select Statistic dialog box, expand Control Charts, click c, u (Cases are Units) or c, u (Cases are Subgroups) and then click OK.
4. From the metadata tree, expand the package if necessary and drag items to the following drop zones:
 - If using Cases are Units, drag an item to the Characteristic variable drop zone. Click Next. Drag an item to Subgroups variable. Click Next.
 - If using Cases are Subgroups, drag an item to the Sample size variable drop zone. Click Next. Drag an item to the Number of nonconforming variable drop zone. Click Next.
5. Drag an item to the Cases variable drop zone to define a set of cases. Click Next.
6. Under Chart, choose either u (Nonconformities per unit) to produce a u chart, or c (Number of nonconformities) to produce a c chart. Click Next.
7. Specify the number of standard deviations to show and the control rules.
Select Display subgroups defined by missing values to display each missing value for the subgroup variable as a separate group in the chart.
Click Next.
8. Specify any output options, such as titles or footnotes, and click Finish.
9. Run the report.

Cases are Units or Cases are Subgroups

In p,np charts and c,u charts, you can choose between cases are units and cases are subgroups.
What is the difference?

Cases are Units

If each case is a unit with a subgroup identifier, such as time of measurement or batch number, cases are assigned to a category according to the value of the identifier. In the following table, each case is a unit.

Time of measurement	Value
8:50	6.35
8:50	6.40
8:50	6.32
8:50	6.39
11:30	6.39
11:30	6.46

Time of measurement	Value
11:30	6.37
11:30	6.36

Cases are Subgroups

If each case is a subgroup, there are as many variables as individuals within the sample. In the following table, each case is a subgroup.

Time of measurement	N1	N2	N3	N4
8:50	6.35	6.40	6.32	6.39
11:30	6.39	6.46	6.37	6.36

Specification Limits

In X-bar charts and individuals charts, you can specify the limits that determine if your measure of interest meets the intended specifications. These specification limits are shown on the chart along with the calculated limits that the [control rules](#) determine.

- To show a fixed upper limit, enter the value in [Upper specification limit](#).
- To show a fixed lower limit, enter the value in [Lower specification limit](#).
- To show a target value for the process, enter the target value in [Specification target](#).

You can also specify how to estimate the capability sigma, which is a measure of variation used in the computation of [capability indices](#). You can estimate the capability sigma based on the R-bar values (the mean range), on the S-bar values (the mean standard deviation), or on the variation within a subgroup.

For an example of setting specification limits, see "[Example: Create a Report Showing Shampoo Manufacturing Quality Control](#)" (p. 295).

You can also specify whether to show the percentage of individual observations in the process that lie outside the specification limits by clicking [Actual % outside specification limits](#).

Control Rules

Control rules are decision rules for detecting out-of-control conditions in control charts. The location of data points relative to the control limits and the central line indicate whether the process performs as expected. The number of specified standard deviations determine the control limits. The default is ± 3 standard deviations.

Note: Keep in mind the difference between control limits and specification limits. Control limits are calculated automatically by the number of standard deviations, and determine if the process is

in control. **Specification limits** are specified manually, and determine if your measure of interest meets the intended specifications.

In X-bar charts, individuals charts, p,np charts, and c,u charts, you can use control rules to supplement the default rule of displaying points only when they exceed the specified control limits. Random variation within those limits indicates the process is in control. Nonrandom variation might indicate that something is wrong.

For example, you can choose to display when two of three consecutive points fall outside the 2-sigma warning limits, but are still within the 3-sigma control limits. This might signify issues in the future that have not yet actually gone out of control.

If a point violates any rule, it appears in the chart with a different shape and color compared to in-control points.

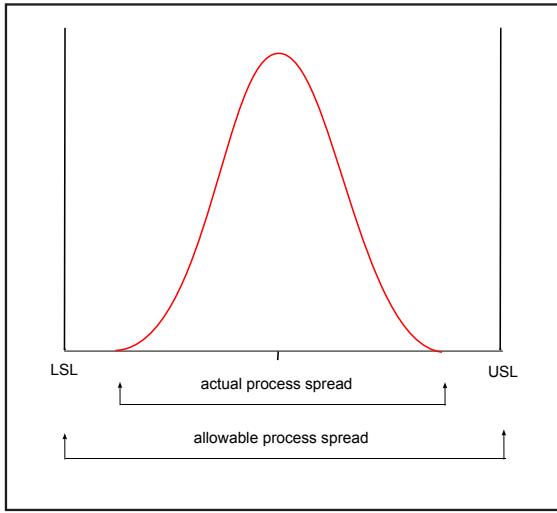
The output includes a table of rule violations. If a point violates more than one rule, it is listed multiple times for each rule violation. By default, the subgroup variable is used to identify points in the table.

Process Capability Indices

The process capability indices measure the capability of the process.

Process capability compares the actual output of a process to the specified **specification limits** by calculating the ratio of the spread between the process specifications (the specification width) to the spread of the process values (the process width).

The curve in the graphic below represents the process width. The upper specification limit and the lower specification limits represent the specification width.



Each statistic measures a different aspect of the process. Most of these indices are based on the **capability sigma** (which is computed using the sample mean moving range), the specification limits, or both.

For an example of setting capability indices, see "[Example: Create a Report Showing Shampoo Manufacturing Quality Control](#)" (p. 295).

Statistic	Description
CP	Measures the overall capability of the process, using the ratio of the difference between the specification limits to the observed process variation. CP values greater than or equal to 1 indicate a capable process. Values less than 1 indicate that the process is too variable.
CpU CpL	Determines whether or not the process variability is symmetric, using the distance between the process mean and the upper specification limit (CpU) or the lower specification limit (CpL) scaled by capability sigma. If this value is close to CP, then the process variability is relatively symmetric.
K	Measures the deviation of the process mean from the midpoint of the specification limits. This measurement is computed independently of the estimated capability sigma.
CpK	Capability of the process related to both dispersion and centeredness. It is the minimum of CpU and CpL. If only one specification limit is provided, we compute and report a unilateral CpK instead of taking the minimum.
CR	The reciprocal of CP.
CpM	An index relating capability sigma and the difference between the process mean and the target value.
Z-upper Z-lower	The number of capability sigma between the process mean and the upper specification limit (Z-upper) or lower specification limit (Z-lower).
Z-max Z-min	The minimum number (Z-min) or maximum number (Z-max) of capability sigma between the process mean and the specification limits.
Z-out	The estimated percentage outside the specification limits. The standard normal approximation is based on the Z-upper and Z-lower indices.

Process Performance Indices

The process performance indices measure the performance of the process. Most of these indices are based on the standard deviation of the process, or the [specification limits](#) chosen in the statistical object wizard, or both.

These measures are similar to process capability indices, but are based on the overall process variability rather than the sample variability.

For an example of setting performance indices, see "[Example: Create a Report Showing Shampoo Manufacturing Quality Control](#)" (p. 295).

Statistic	Description
PP	Measures the overall performance of the process.
PpU	The distance between the process mean and the upper specification limit scaled by process standard deviation.
PpL	The distance between the process mean and the lower specification limit scaled by process standard deviation.
PpK	Measures the performance of the process related to both dispersion and centeredness. It is the minimum of PpU and PpL. If only one specification limit is provided, we compute and report a unilateral PpK instead of taking the minimum.
PR	The reciprocal of PP.
PpM	An index relating process variance and the difference between the process mean and the target value.
Z-upper Z-lower	The number of standard deviations between the process mean and the upper specification limit (Z-upper) or the lower specification limit (Z-lower).
Z-max Z-min	The minimum number (Z-min) or maximum number (Z-max) of standard deviations between the process mean and the specification limits
Z-out	The estimated percentage outside the specification limits. The standard normal approximation is based on the Z-upper and Z-lower.

Chapter 10: Relational Reporting Style

The relational reporting style is recommended for relational data sources. Relational data is best represented by lists. This data is organized in IBM® Cognos® Report Studio by query items.

In relational reporting, you summarize data by using headers and footers in lists, summary functions, and within detail aggregation. You focus data in relational reporting with summary or detail filters.

IBM Cognos Query Studio is also available to work with relational data.

You can perform the following tasks in relational reporting:

- [add data](#)
- [group data](#)
- [summarize data](#)
- [focus data with filters](#)
- [sort data](#)
- [add queries](#)
- [use calculations](#)

Add Relational Data to a Report

Select the data items to appear in the report.

We recommend using relational data in the relational reporting style. However, if you are using dimensional data, see "[Add Dimensional Data to a Report](#)" (p. 347).

If you frequently use items from different query subjects or dimensions in the same reports, ask your modeler to organize these items into a folder or model query subject and then to republish the relevant package. For example, if you use the product code item in sales reports, the modeler can create a folder that contains the product code item and the sales items you need.

Steps

1. In the **Insertable Objects** pane, on the **Source** tab , drag each query item to the report.
A flashing black bar indicates where you can drop an item. Items inserted in the report appear on the **Data Items** tab .
- Other ways to select data items are to double-click each item or to right-click each item and click **Insert**.

2. If you want to remove a data item from the report, select it and click the delete button . To remove the data item from the report but keep it on the **Data Items** tab, click the cut button instead.
3. For more information about a query item, right-click the item in the **Source** tab and click **Properties**. For example, when working with relational data sources, the **Usage** property identifies the intended use for the data represented by the data item. You can also obtain more information by clicking **Lineage**.

Group Relational Data

Group data items in a list report to remove duplicate values. For example, you have a report that shows all the products purchased and their product type. You group the Product type column so that each product type cell spans the products purchased cells.

You can specify whether grouping a column automatically makes it the first column in the list by selecting **Group and automatic summary behavior for lists** in the IBM® Cognos® Report Studio [options](#). With this option selected, if the list contains any fact or measure columns, footers that show the summary values for the columns are automatically created for each group. Grouping a second column makes it the second column. When you ungroup a column, that column becomes the first column after the grouped column.

After a column is grouped, you can move it elsewhere in the report.

You can also group data items in repeaters and in page sets.

In addition to grouping, you can specify the sort order of data items. In lists, you can sort data items within groups as well as ungrouped items. In crosstabs, you can sort row and column items. For more information, see "[Perform Advanced Layout Sorting](#)" (p. 329).

Tip: The Succession Report sample report in the GO Data Warehouse (analysis) package includes grouping. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Limitations When Grouping Data

When working with dimensional data sources, you cannot group two levels or hierarchies from different dimensions in the same list.

For example, a list contains the following levels:

- Product Line
- Region
- Product Type
- Country
- Revenue

Product Line and Product Type are from one dimension and Region and Country are from another dimension. You group by the first three columns. You may encounter the following errors.

OP-ERR-0199: *The query is not supported. The dimensions on the edge are inconsistent.*

OP-ERR-0049. *Unable to support a grouping that breaks the hierarchical order of the query items within it.*

When working with dimensional data sources, you can group or sort only in the order of the existing hierarchy. If you change the order, the above error also occurs.

You can omit columns from the grouping. For example, if a level hierarchy contains the levels Country, State, and City, and another level hierarchy contains the level Product:

- Country, State, City is valid.
- Country, Product, and State is not valid because a level from another level hierarchy was inserted between two levels from another level hierarchy.
- Country, City, and Product is valid even though State is omitted.
- Country, Country code is valid where Country code is a member property of Country.

To override the hierarchical order that the data source specifies, you can override the dimension information of the query. For more information, see "["Add Dimension Information to a Query" \(p. 384\)](#).

We recommend that you do not use member properties of a dimension to group data. Grouping by using member properties may give unexpected results, such as duplicates.

Steps

1. Click the column on which to group.

You can click either the column title or one of the column cells.

Tip: To perform multiple groupings at once, use Ctrl+click or Shift+click.

2. From the Structure menu, click **Group/Ungroup** .

A symbol appears indicating that the column is grouped. You can control when its values appear by [setting the group span](#), and the column appears in the list of headers and footers that you can add to the report.

Note: Grouping all columns in a list is equivalent to creating a single-edge crosstab.

Perform Advanced Grouping

You can perform more advanced groupings in a list to get the results you want. For example, you can

- view the entire grouping structure and make changes to the grouping order
- group a data item that appears in a query but not in the layout
- specify a list of properties for a group

You can also perform advanced sorting.

Steps

1. Click a column in the list.
2. Click the select ancestor button  in the title bar of the Properties pane and click List.
3. In the Properties pane, double-click the **Grouping & Sorting** property.
4. To change the grouping order, in the Groups pane, click the grouped item in the Groups folder and drag it to the grouping hierarchy.
5. To create a new grouping, in the Data Items pane, click the data item and drag it to the Groups folder in the Groups pane.
6. To specify a list of data items for a group, do the following:
 - Click the group.

Tip: Click **Overall** to specify a list of data items for the overall group. The overall group is the grouping of all data into one group. You can specify a list of data items for the overall group to add overall totals to the list.

- Click the properties button .
- Select the appropriate check boxes.

Set the Group Span for a Column

When columns are grouped, you can choose how often to show column names by changing the group spanning. For example, when country and city are both grouped, you can choose to show the country name each time

- the country changes, by spanning Country by Country
- the city changes, by spanning Country by City
- there is a new record, by specifying no spanning

Spanning one grouped column by another column is helpful if the second column contains many items.

Steps

1. Click the column that will span the other column(s).
2. In the Properties pane, set the **Group Span** property to the column to span.

Example - Remove Identical Values in a List

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a report that lists all product orders organized by order number. To make the report easy to read, you group the Order number column so that only one instance of each order number appears. However, because each order contains different products, the same order

date appears for each product. You decide to show the order date only when the order number changes.

Steps

1. Open Report Studio with the **GO Data Warehouse (query)** package.
2. In the **Welcome** dialog box, click **Create a new report or template**.
3. In the **New** dialog box, click **List** and click **OK**.
4. In the **Insertable Objects** pane, on the **Source** tab , expand **Sales and Marketing (query)** and **Sales (query)**.
5. Expand **Sales order** and add **Order number** by double-clicking it.
6. Expand **Time dimension** and add **Date** by double-clicking it.
7. Expand **Product** and add **Product name** by double-clicking it.
8. Expand **Sales fact** and add **Quantity** and **Revenue** by double-clicking them.
9. **Group** the **Order number** column.
10. Click the **Date** column.
11. In the **Properties** pane, set the **Group Span** property to **Order number**.

When you run the report, the first row of the **Order date** column appears for each **Order number** row.

Order number	Date	Product name	Quantity	Revenue
100001	Jan 12, 2004	Flicker Lantern	256	8,624.64
		Polar Ice	92	9,411.6
100002	Jan 12, 2004	Bear Edge	172	6,690.8
		Edge Extreme	162	18,032.22
		Glacier GPS Extreme	74	24,747.82
		Insect Bite Relief	422	2,532
		Mountain Man Deluxe	90	6,825.6
100003	Jan 12, 2004	BugShield Extreme	3,252	21,170.52
		Polar Extreme	19	2,733.15
		Seeker 50	88	10,975.36
		Sun Shield	1,107	6,376.32
100004	Jan 12, 2004	Hibernator Lite	354	29,658.12
		Star Gazer 2	139	75,289.35
		Star Lite	261	89,841.42
		TrailChef Deluxe Cook Set	279	33,658.56
100005	Jan 12, 2004	Bear Survival Edge	120	10,299.6
		Double Edge	382	6,165.48
		Lady Hailstorm Titanium Woods Set	26	32,878.04
		Mountain Man Combination	44	4,139.96
		Mountain Man Extreme	24	6,711.84

Example - Repeat a Column Value in a List

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a report that lists all products sold by the company, organized by the product line and product type. To make the report easier to read, you group the Product line and Product type columns so that only one instance of each column appears. However, because some product lines contain many different product types and products, you decide to show the product line for each product type.

Steps

1. Open Report Studio with the **GO Data Warehouse (query)** package.
2. In the **Welcome** dialog box, click **Create a new report or template**.
3. In the **New** dialog box, click **List** and click **OK**.
4. In the **Insertable Objects** pane, on the **Source** tab , expand **Sales and Marketing (query)**.
5. Expand **Sales (query)** and **Product** and add the following data items by double-clicking them:
 - **Product line**
 - **Product type**
 - **Product name**
6. Expand **Sales fact** and add **Product cost** by double-clicking it.
7. **Group** the **Product line** and **Product type** columns.
8. Click the **Product line** column.
9. In the **Properties** pane, set the **Group Span** property to **Product type**.
10. Click any part of the list and then click the select ancestor button  in the title bar of the **Properties** pane.
11. Click **List**.
12. In the **Properties** pane, double-click the **Font** property.
13. In the **Size** box, click **8pt**.

When you run the report, the product line appears whenever the product type changes.

Product line	Product type	Product name	Product cost
Camping Equipment	Cooking Gear	TrailChef Canteen	6,607,904.78
		TrailChef Cook Set	28,305,454.21
		TrailChef Cup	1,545,089.31
		TrailChef Deluxe Cook Set	35,527,805.84
		TrailChef Double Flame	18,416,925.00
		TrailChef Kettle	11,373,514.16
		TrailChef Kitchen Kit	13,608,439.00
		TrailChef Single Flame	31,839,545.34
		TrailChef Utensils	8,611,463.50
		TrailChef Water Bag	11,292,005.38
Camping Equipment	Lanterns	EverGlow Butane	4,792,227.24
		EverGlow Double	1,421,083.75
		EverGlow Kerosene	6,304,040.00
		EverGlow Lamp	13,459,561.68
		EverGlow Single	10,363,144.02
		Firefly 2	9,088,038.23
		Firefly 4	4,968,302.40
		Firefly Extreme	4,838,898.69
		Firefly Lite	3,717,927.00
		Firefly Mapreader	5,734,965.00

Divide Data into Sections

Create sections in a report to show a data item as the heading of a section. When you run the report, separate sections appear for each value.

Creating sections is similar to creating headers by grouping on a data item. The difference is that section headers appear outside the list, crosstab, chart, or repeater. In addition, you can group data items only in lists.

When working with dimensional data, you can also create page layers to show values on a separate page for each member.

Steps

1. Click the column to make a section heading.

2. Do one of the following:

- From the **Structure** menu, click **Section** .

If the column is in a list or repeater, this menu option will create sections without creating a master detail relationship. This can improve performance when running the report.

If the column is in a crosstab or chart, this menu option will create sections using a master detail relationship.

- From the **Structure** menu, click **Section Using Master/Detail**.

This menu option creates sections using a master detail relationship.

Sections are created, and a section header appears. The data container that contains the column used to create sections is embedded in a list.

3. To add or remove section headers and footers, from the **Structure** menu, click **Headers & Footers**, click **List Headers & Footers**, and select or clear the appropriate check boxes.

The section header and footer check boxes appear under **Overall header**.

Remove Sections

You can remove sections and reinsert the data item that was used to create the sections in the data container.

Steps

1. From the **View** menu, click **Page Structure**.
2. Expand the page containing the data container with the section.
3. Expand **Page Body** until you see the data container in which you added the section.

The data container is nested in the **List Column Body** object of the **List** object that is created when you add a section.

4. Drag the data container to **Page Body**.

The data container appears as a node of **Page Body**.

5. Delete the **List** object.
6. From the **View** menu, click **Page Design**.

7. In the **Insertable Objects** pane, click the **Data Items** tab .

8. Drag the data item that was used as a section header back into the data container.

9. If you created sections in a list without [creating a master detail relationship](#), click the data item, and then click the group/ungroup button  to ungroup the item.

For information about the page structure view, see "[The Page Structure View](#)" (p. 437).

Summarizing Data Relationally

Summarize data in your reports to obtain totals, averages, and so on.

In a report, you can add both detail and summary aggregation. Detail aggregation, which is supported only for relational data sources, specifies how a data item is totaled at the lowest level in a report. In lists, detail aggregation specifies how the values that appear in the rows are totaled. In crosstabs, detail aggregation specifies how the values in the cells are totaled. For example, detail aggregation for a measure like Revenue might be Total in both lists and crosstabs. In the following list report, this means that the values you see for the Revenue column represent the total revenue for each product type.

Product line	Product type	Revenue
Camping Equipment	Cooking Gear	\$5,401,132.08
	Tents	\$47,456,718.10
	Sleeping Bags	\$11,688,944.32
	Packs	\$14,627,996.26
	Lanterns	\$10,539,200.16
Golf Equipment	Putters	\$2,310,367.24
	Golf Accessories	\$811,516.72
	Irons	\$8,801,464.98
	Woods	\$13,982,116.64

Summaries, which are supported for all data sources, specify how data items are totaled in the headers and footers of a list and in the total rows and columns of a crosstab. For list reports, these summaries only summarize the data that is visible on that page of the report.

You can specify detail and summary aggregation in different ways by using any of the following:

- aggregation properties that are specified in the model
- the **Auto Group & Summarize** property
- the summarize button in the toolbar
- aggregation properties for a data item
- summary functions in data item expressions
- the solve order of calculations

Limitation

If a summary is applied to a report that contains binary large object (BLOB) data, such as images or multimedia objects, you cannot also perform grouping or sorting.

The Type of Data

How data is aggregated also depends on the type of data that you are aggregating. Aggregation rules are applied differently to facts, identifiers, and attributes. For example, if you aggregate a data item that represents part numbers, the only aggregate rules that apply are count, count distinct, count non-zero, maximum, and minimum. For information about how to determine the type of data that a data item represents, see "[Add Relational Data to a Report](#)" (p. 307) and "[Add Dimensional Data to a Report](#)" (p. 347).

Add a Simple Summary

You can add simple summaries to the groups in a report by using the summarize button in the toolbar. This button provides a subset of the summary functions available in IBM® Cognos® Report Studio. For list reports, a **Custom** option is also available so that you can add your own summary function in the expression of the data item.

The summarize button sets the rollup aggregate property for the data item to the selected summary, and places the data item into an appropriate footer. A footer is created for each group as well as an overall footer, unless they already exist.

For information about adding a rolling or moving average, see "[Rolling and Moving Averages](#)" (p. 364).

Steps

1. Click the column to which to add a summary.
2. Click the summarize button  and click a summary type.
3. To change the summary label, do the following:
 - Click the label.
 - In the Properties pane, under Text Source, set the Source Type property to the source type to define the label.
For example, set it as **Data Item Value** to produce a dynamic label for the summary based on data item values.
 - Set the property below Source Type to specify the label.

This property depends on the source type you chose. For example, if you chose **Data Item Value** as the source type, set the **Data Item Value** property to the data item to use to define the label.

In lists, the summary appears as a footer. If the column to which you added a summary is grouped, group and overall summaries appear. In crosstabs and charts, the summary appears as a node.

Tips

- To change a summary, select it and, in the Properties pane, under Data Item, click Rollup Aggregate Function and choose a different function.
- In crosstabs, you can add multiple summaries at the same level. For example, you have a crosstab with Product line as rows, Order year as columns, and Revenue as the measure. For Product line, you can add the **Total** summary as a header, which will total all revenue for each order year. You can then add the **Average** summary as a footer, which will give the average revenue of all product lines for each order year.

Set the Auto Group & Summarize Property

Set the **Auto Group & Summarize** query property to specify whether Report Studio should group non-fact data items (identifiers and attributes) and apply aggregate functions to aggregate fact data items in lists.

If you are using an OLAP data source, data is always summarized regardless of how this property is set.

Steps

1. Pause the pointer over the query explorer button  and click a query.
2. In the Properties pane, set the **Auto Group & Summarize** property:

- To group non-aggregate fact data items and apply aggregate functions to aggregate fact data items in lists, set this property to **Yes**.
- To render detail rows, set this property to **No**.

Specify Detail or Summary Aggregation in the Model

When working with relational or dimensionally-modeled relational (DMR) data sources, you can use the aggregation properties specified for the query item in the IBM Cognos Framework Manager model instead of specifying detail or summary aggregation in the report. The model specifies the default summary function for each query item.

Aggregation functions are mapped between data sources, Framework Manager, and Report Studio.

Specify Aggregation Properties for a Data Item

When working with relational data sources and list reports, for each data item in a report, you can specify detail and summary aggregation properties to manage summaries without having to create complex data item expressions.

Steps

1. Click the data item for which to set detail or summary aggregation.
2. In the **Properties** pane, set the **Aggregate Function** or the **Rollup Aggregate Function** property to a function.

Use Summary Functions in Data Item Expressions

You can use summary functions in data item expressions. The summary functions in the expression editor that have the same name as the summary functions available through the **Aggregate Function** and **Rollup Aggregate Function** properties operate the same way. For example, in a relational report, setting the **Aggregate Function** property to **Total** is the same as changing the expression of the data item to `total ([Revenue])`.

In general, report maintenance is easier if the **Aggregate Function** and **Rollup Aggregate Function** properties are used rather than adding summary functions to data item expressions. Use summary functions in expressions if the required summary is not supported as an aggregate or rollup aggregate property or if the complexity of the expression cannot be managed using the data item properties. Add a summary function to a data item expression if one of the following conditions applies:

- The underlying data source is relational, and you want to use database vendor-specific summary functions.
- You want to use summary functions that require more than one parameter, such as percentile.
- You require summary expressions that are not available in the aggregation properties, such as a `for` clause. You should use `for` clauses only in relational style reports (list reports).

For example, your report uses Product line, Product type, and Quantity. You want to calculate the percentage of the parent quantity that is represented by each product type. This requires you to compare the total quantity for each product type to the total quantity for the parent product line.

In other words, your expression requires summaries at different group levels. You use summary functions, such as `aggregate`, in the data item expression with a `for` clause to specify the group level as follows:

`[Quantity] / aggregate ([Quantity] for [Product line])`

Product line	Product type	Quantity	% of Product Line
Camping Equipment	Cooking Gear	13,400,351	49%
	Lanterns	4,826,755	18%
	Packs	2,756,540	10%
	Sleeping Bags	3,153,218	12%
	Tents	3,164,285	12%
Camping Equipment - Summary		27,301,149	100%
Golf Equipment	Golf Accessories	3,119,747	61%
	Irons	391,445	8%
	Putters	1,284,570	25%
	Woods	317,939	6%
Golf Equipment - Summary		5,113,701	100%

In some cases, you may need a summary that is different from the default. For example, if the default summary for Quantity is `total`, you can calculate the average as follows:

`[Quantity] / average (aggregate ([Quantity]) for [Product line])`

Product line	Product type	Quantity	% of Product Line Average
Camping Equipment	Cooking Gear	13,400,351	245%
	Lanterns	4,826,755	88%
	Packs	2,756,540	50%
	Sleeping Bags	3,153,218	58%
	Tents	3,164,285	58%
Camping Equipment - Summary		27,301,149	100%
Golf Equipment	Golf Accessories	3,119,747	244%
	Irons	391,445	31%
	Putters	1,284,570	100%
	Woods	317,939	25%
Golf Equipment - Summary		5,113,701	100%

The inner summary gives you the expected Quantity values at the Product Type level, which are then averaged to the Product Line level. This two-stage aggregation is required for OLAP data sources and recommended for relational data sources also.

Why is this necessary? Here's what would happen if you simply averaged Quantity, as follows:

`[Quantity] / average([Quantity] for [Product line])`

Product line	Product type	Quantity	% of Product Line Average
Camping Equipment	Cooking Gear	13,400,351	2,538,352%
	Lanterns	4,826,755	914,305%
	Packs	2,756,540	522,156%
	Sleeping Bags	3,153,218	597,296%
	Tents	3,164,285	599,392%
Camping Equipment - Summary		27,301,149	5,171,500%
Golf Equipment	Golf Accessories	3,119,747	1,205,327%
	Irons	391,445	151,236%
	Putters	1,284,570	496,299%
	Woods	317,939	122,837%
Golf Equipment - Summary		5,113,701	1,975,700%

The denominator is now computed as the average of the detail Quantity rows in the underlying fact table. Since these values are much smaller than the pre-summarized Quantity values you see in the report, the resulting percentage is far too big, and not what we wanted.

Tip: The Total Revenue by Country sample report in the GO Data Warehouse (query) package includes a total summary function. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Steps

1. Click a data item.
2. In the Properties pane, double-click the **Expression** property and add summary functions to the expression for the data item.

Limitations When Using Clauses in Summary Functions

A summary function that uses a `for` clause may yield unexpected results. These may include error messages, warnings, incorrect numbers, and more or fewer than expected rows, columns, or chart points and lines.

To avoid these problems, ensure that the parameters that follow the `for` clause adhere to the following constraints:

- Parameters must be simple data item references.
- All data items in the parameter list must appear on every list, crosstab, or chart that uses that summary.
- For any edge used in the `for` clause, data items listed in the `for` clause must start with the first data item on that edge.
- Data items must be listed in the order in which they appear on each edge of the report with no gaps.
- In crosstabs and charts, there must be no sibling data items that are considered details. Summaries are normally not considered details.
- Section headers must not be included in the parameter list.

If following these constraints does not resolve the problems and your report uses dimensional data with no detail or summary filters, consider using the `within set` clause instead of the `for` clause.

Summary Functions

This section describes the summary functions that are available in Report Studio. Some functions, such as **Custom**, are available only when you click the summarize button  on the toolbar. Additional summary functions are available in the expression editor.

Automatic Summary

Depending on the type of data item, applies the function **None**, **Summarize**, or **Calculated** based on the context in which the data item appears.

Calculated is applied if the data item expression

- contains a summary function
- is an `ifthenelse` or `case` expression that contains a reference to at least a modeled measure in its condition
- contains a reference to a model calculation or to a measure that has the **Regular Aggregate** property set to a value other than **Unsupported**
- contains a reference to at least one data item that has the **Rollup Aggregate Function** property set to a value other than **None**

If the underlying data source is relational and if the data item expression contains no summary functions and a reference to at least one fact that has the **Regular Aggregate** property set to a value other than **Unsupported**, **Summarize** is applied.

In all other contexts, **None** is applied.

For relational and dimensionally modeled relational (DMR) data sources, if this function is applied to a data item expression that has the `average` function, weighted averages are computed based on the lowest level detail values in the data source.

This is the default function.

Average

Adds all existing values and then divides by the count of existing values.

Calculated

Specifies that all the terms within the expression for a data item are aggregated according to their own rollup rules, and then the results of those aggregations are computed within the overall expression.

For example, a list contains the data item `Quantity` with the **Aggregation** property set to **Total**. You add a query calculation named `Quantity Calculated`. You define its expression as `Quantity + 100` and you set its **Aggregation** property to **Calculated**. When you run the report, the values for `Quantity Calculated` are computed by first adding all values for `Quantity` and then adding 100 to each value.

In crosstabs, this function overrides any solve orders that are specified.

Count

Counts all existing values.

If the underlying data source is dimensionally modeled relational (DMR), Count behaves as follows if it is specified in the **Aggregate Function** or **Rollup Aggregate Function** property for a data item.

Object	Behavior
Level	<code>Count distinct</code> is used. A warning appears when you validate the report.
Member set	<code>Count distinct</code> is used. A warning appears when you validate the report.
Attribute	Supported.
Measure	Supported.

If you add the `count` summary to a non-fact column in a list and then group the column, the column will not be grouped when you run the report. To resolve this issue, group the column first before adding the `count` summary.

Count Distinct

Returns the total number of unique non-null records.

If the underlying data source is dimensionally modeled relational (DMR), `count distinct` is supported for levels, member sets, attributes, and measures when it is specified in the **Aggregate Function** or **Rollup Aggregate Function** property for a data item.

Custom

Summarizes data based on an expression that you define.

Maximum

Selects the largest existing value.

Median

Returns the median value of the selected data item.

Minimum

Selects the smallest existing value.

None

Does not aggregate values.

If the underlying data source is relational, the data item is grouped when the query property **Auto Group & Summarize** is set to Yes.

Not Applicable

This function is available only for the **Aggregate Function** property. It specifies that the aggregate attribute is ignored. For example, the calculation will be applied after the data is aggregated.

This function differs from the **Calculated** function, which applies aggregation to the rows and then performs the calculation.

For example, for the `average ([Quantity]) + 5` expression, when the **Aggregate Function** property is set to **Not Applicable**, five is added to the average of the single Quantity value. When the function is set to **Calculated**, five is added to the average of a set of numbers.

This setting should not be applied to a simple model reference.

This setting is relevant only for reports that are upgraded from IBM Cognos ReportNet® 1.x.

Standard Deviation

Returns the standard deviation of the selected data item.

From a mathematical perspective, this function is not useful for small numbers of items and is not supported if the query property **Auto Group & Summarize** is set to **Yes**.

Summarize

Aggregates data based on model or data type information. This function can be thought of as a calculate and then aggregate rule. Data is aggregated using the following rules:

If the underlying data source type is relational and the data item or calculation is a reference to a single fact query item in the model that has the **Regular Aggregate** property set to **Sum**, **Maximum**, **Minimum**, **Average**, or **Count**, aggregate data using this function. Otherwise, data is aggregated according to the data type of the data item as follows:

- **Total** is applied for numeric and interval values.
- **Maximum** is applied for date, time, and date-time values.
- **Count** is applied for everything else.

The underlying data source type can be dimensional and the data item or calculation a reference to a single item in the model. In this case, if the **Regular Aggregate** property of the model item is not **unknown**, the summary function that corresponds to the **Regular Aggregate** property is applied. If the **Regular Aggregate** property is **unknown**, set the function to **Calculated**.

For example, a list contains the data item **Quantity** with the **Aggregation** property set to **Average**. You add a query calculation named **Quantity Summarize**. You define its expression as **Quantity + 100** and you set its **Aggregation** property to **Summarize**. When you run the report, the values for **Quantity Summarize** are computed by first adding 100 to each value for **Quantity** and then calculating the total. The total is calculated because **Quantity** contains numeric values and the expression for **Quantity Summarize** is not a simple data item reference. If **Quantity Summarize** is defined as **Quantity**, the function **Average** is applied to each value.

Total

Adds all existing values.

Variance

Returns the variance of the selected data item.

Note: From a mathematical perspective, this function is not useful for small numbers of items and is not supported if the query property **Auto Group & Summarize** is set to **Yes**.

Mapping Aggregation Functions From Data Sources to Framework Manager and Report Studio

The summary functions available in Framework Manager and Report Studio reflect summary functions supported by relational and dimensional data sources. The following table shows how summary functions in data sources are mapped to Framework Manager and Report Studio.

Data source	Framework Manager	Report Studio
None specified, or none	Unsupported	None
average (avg)	Average	Average
count	Count	Count
count distinct	Count Distinct	Count distinct
maximum (max)	Maximum	Maximum
minimum (min)	Minimum	Minimum
median	Median	Median
standard deviation (stddev)	Standard Deviation	Standard Deviation
sum	Sum	Total
variance (var)	Variance	Variance
count non zero	Count Non-Zero	Automatic
average non zero	unknown	Automatic
external	unknown	Automatic
any	unknown	Automatic
calculated	unknown	Automatic
unknown	unknown	Automatic
first_period	unknown	Automatic

Data source	Framework Manager	Report Studio
last_period	unknown	Automatic

The following summary functions exist only in either Framework Manager or Report Studio. There is no corresponding summary function in data sources.

Framework Manager	Report Studio
Automatic	Automatic
Calculated	Calculated
No corresponding summary function	Summarize
No corresponding summary function	Not Applicable

Focusing Relational Data

You can limit the data in your report with filters or query parameters. The method you choose should depend on the capabilities of the data source, the performance requirements, the persistence of the dataset, and how complex you want the report to be.

In IBM® Cognos® Report Studio, summary and detail filters are best for list reports in the relational reporting style. Summary filters are applied to summaries and detail filters are applied to details. These filters remove rows and columns from your reports.

You can also use prompts, or prompt controls, to allow report users to provide input for the filter.

Create a Detail or Summary Filter

Add a filter expression to focus a report and minimize processing time by excluding unwanted data. For example, you can filter data to show only customers who placed purchase orders that were valued at over one thousand dollars during the past year. When you run the report, you see only the filtered data.

Detail filters are applied only to detail data items from the data source, and not to summary data items. Examples of detail data items include customer names, product numbers, and order numbers. Detail filters are Boolean expressions that apply one or more conditions and evaluate to either true or false. If the conditions are true for a row of data, the row is retrieved from the data source and appears in your report. If the conditions are false, the row is not retrieved and does not appear in your report. You can specify whether detail filters are applied before or after summaries using the **Application** property.

Tip: The 2005 Sales Summary sample report in the GO Sales (analysis) package includes a detail filter. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Summary filters are applied to summary data items in a report. They are also Boolean expressions, however they are used to exclude specific groups of data items based on the group summaries. They are based on summary calculations.

For relational data sources, if an expression is used in multiple reports or by different report authors, ask your modeler to create the expression as a standalone object in the model and include it in the relevant package. For information about creating filters in the package, see the IBM Cognos Framework Manager *User Guide*.

If you are filtering a multiple-fact query, you should understand the differences between using a detail filter and summary filter.

If you filter values that are floating-point numbers, you may encounter imprecise filtering results due to rounding or data truncation. Floating-point numbers may appear differently in the report than how they are stored in the data source. To ensure accurate results, your filter criteria should account for any rounding issues.

If you are working with a Microsoft® SQL Server 2005 Analysis Services (SSAS) data source, we recommend that you not use OR filters in summaries. Doing so causes error characters (--) to appear in summaries instead of values.

Steps to Create a Filter

1. To add a filter that was created in the package, in the **Insertable Objects** pane, on the **Source** tab , drag the filter to the report.

You must drag the filter to an object that is linked to a query. If there is more than one query defined in the report, drag the filter to an object that is linked to the query.

2. From the **Data** menu, click **Filters** , and click **Edit Filters**.

If you have more than one query defined in the report, you must first click an object that is linked to a query.

Tip: You can also define filters in **Query Explorer**.

3. In the **Filters** dialog box, decide what type of filter to create:

- To add a filter that will apply to detail values, click the **Detail Filters** tab.
- To add a filter that will apply to summary values, click the **Summary Filters** tab.

4. Click the add button .

5. In the **Available Components** box, define the filter expression:

- To add data items that are not shown in the report, on the **Source** tab, double-click data items.
- To add data items that are in the report but not necessarily in the model (such as calculations), on the **Data Items** tab , double-click data items.
- To add data items from a specific query in the report, on the **Queries** tab , double-click data items.

- To add functions, summaries, and operators, on the **Functions** tab , double-click elements.

Note: You can insert only functions that return a Boolean value (true or false). For example, you cannot insert the function `topCount` because it returns a set of data. Filter expressions must resolve to a Boolean in order to be valid.

- To add a value that is derived from a parameter, on the **Parameters** tab , double-click the parameter.

Parameters define prompts, drill-through reports, and master detail relationships.

Tip: You can also type the filter expression directly in the **Expression Definition** box. When typing date values, use the YYYY-MM-DD format.

For more information about creating expressions, see "[Using Relational Calculations](#)" (p. 342).

6. Click the validate button  and click **OK**.

7. In the **Usage** box, specify whether the filter is required, optional, or not to be used.

8. If you created a detail filter, in the **Application** box, click one of the following options:

- To apply the filter to individual records in the data source, click **Before auto aggregation**.

For example, for a specific product type within a product line, you want to filter out individual orders that generated less than one thousand dollars in revenue.

- To apply the filter to data after the query has grouped and summarized at the lowest level of detail, click **After auto aggregation**.

For example, you want to filter out the product types that generated less than ten thousand dollars in revenue within a product line.

For example, you have a list that contains the data items Product line, Product type, and Revenue. The revenue values you see are aggregated to the product type level. If you create a detail filter on Revenue and you choose to filter values before auto aggregation, you are filtering non-aggregated revenue values. If you choose to filter values after auto aggregation, you are filtering revenue values aggregated to the product type level.

9. If you created a summary filter, click the ellipsis (...) button under **Scope** and select the grouping level at which to apply the filter.

For example, a revenue report is grouped on product line and product type. You can choose to filter total revenue for either the product line or product type level.

If the query in which you are creating the summary filter is linked to more than one data container, the data item that you select as the grouping level must be grouped in all data containers linked to the query. Otherwise, the report will not run successfully.

Tip: To filter at the overall level, do not click a level. For example, if a report has an overall footer that shows the total revenue for all products, by not choosing a level you will apply the

filter to the footer. In addition, if you are using a dimensional data source, excluding a parent level excludes its children, and excluding all children excludes the parent.

Steps to Edit a Filter

1. Click an object that is linked to a query that contains a filter.
2. To remove all filters, from the Data menu, click Filters , and click Remove all Filters.
3. To edit a filter, from the Data menu, click Filters , and click Edit Filters.

Tip: You can also edit filters from **Query Explorer**. Pause the pointer over the query explorer button  and click the query that contains your filter.

Ignoring the Time Component in Date Columns

Database systems use Date, Time, and Timestamp to represent date and time values. While a timestamp type holds a date and time component, an application may allow the RDBMS to default the time component. When rows are inserted, updated, or queried, the application may specify only a date value and leave the RDBMS to extend the value to include a default time (usually 00:00:00.000).

The challenge with a timestamp is when the application has no immediate interest in the time component. For example, the business question How many orders were taken today? implies all orders taken irrespective of what time the order was booked. If the application defaulted the time component as it stored rows, the query that was used to answer the question returns the count of orders taken today. If the application stored the actual time component, the query likely returns no data, because the number of orders entered at midnight is probably zero.

Relying on the time defaults can be dangerous if the application changes and starts to capture actual times. To avoid this problem, you can

- [truncate the time](#) by creating a derived column
- [convert](#) the timestamp to a date
- [create a hi-low filter](#)

Truncating the Time by Creating a Derived Column

In the Framework Manager model, create a derived column using a calculated expression that truncates the time from a timestamp, and returns a timestamp containing the original date and a default time. For example, if you use an Oracle data source, the following expression creates the derived column DATEONLY from COL1, where COL1 contains the values as stored by the application while DATEONLY contains the dates with the default time of 12:00:00 AM:

```
Select COL1,trunc(COL1) as DATEONLY from [SCOTT_TIGER].DATES
```

Tip: You can change the data format of the column to show only the date value by setting the **Date Style** property to **Short**.

You can then apply filters to the DATEONLY column that would return the correct results. If you create a parameter in Report Studio that filters on this column, the default prompt will present a date and time control because the data type is still a timestamp.

Converting the Timestamp to a Date

In the Framework Manager model, define a calculation that uses the `cast` function to convert the timestamp to a date. For example, the following expression converts the data type of the column COL1 to date:

```
cast ([SCOTT_TIGER].[DATES].[COL1],DATE)
```

If you create a parameter in Report Studio that filters on this calculation, the default prompt presents a date control.

Ignoring the Time by Creating a Hi-Low Filter

You can create a hi-low filter to ignore the time. In the Framework Manager model, create a filter in the form date-column between date-lowtime and date-hightime. For example, the following expression returns all values between 00:00:00:000 and 23:59:59:000 for a given day:

```
[SCOTT_TIGER].[DATES].[COL1] between?p1? and cast(substring(?p1?,1,10),'23:59.59.000',timestamp)
```

Sorting Relational Data

You can sort items to view them in your preferred order. By default, IBM® Cognos® Report Studio retrieves items in the order defined in the data source. OLAP data sources always have a defined order. Relational and dimensionally-modeled relational data sources may not always have a defined order. The data modeler defines the sorting options in the model. For more information about data modeling, see the IBM Cognos Framework Manager *User Guide*. Sorting applied in Report Studio applies only to the layout within the report.

You can sort items in a list in ascending or descending order based on a value or a label, such as revenue or employee name. You can also perform advanced sorting to sort columns within groups or to sort a row or column using another data item.

Tip: The Sales Growth Year Over Year sample report in the GO Data Warehouse (analysis) package includes sorting. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Steps

1. Click the column or row on which to sort.
2. Click the sort button  and click **Ascending** or **Descending**.

An arrow appears beside the data item to indicate that a sort order was set.

When you specify a sort order for more than one column, the columns are sorted in the order in which they were inserted in the report. For example, you add columns A, B, and C to a report and specify a sort order for each. When you run the report, column A is sorted first, then column B, and then column C. You can change the order in which the columns are sorted in the **Edit Layout** **Sorting** options.

Tip: To remove a sort order, click **Don't Sort**.

Perform Advanced Layout Sorting

If you use relational data, in a list, you can sort columns within groups and change the sort order of columns. In a crosstab, you can sort a row or column by another item, such as Order year by Revenue.

Steps

1. Click a column.

In a crosstab, click the sorted row or column on which to perform advanced sorting.

2. Click the sort button  and click **Edit Layout Sorting**.
 3. To sort a list column within a group, do the following:
 - In the **Groups** pane, under the **Groups** folder, expand the folder of the grouped column.
 - In the **Data Items** pane, drag the data items to sort to the **Sort List** folder.

Tip: You can also drag data items from the **Detail Sort List** folder.

 - Click the sort order button  to specify ascending or descending order.
 4. To change the sort order of columns, in the **Groups** pane, change the order of columns in the **Sort List** folder of a group or in the **Detail Sort List** folder.
- Tip:** Add items to the **Detail Sort List** folder to sort items that are not groups.
5. If you are sorting a crosstab, do the following:
 - From the **Data Items** pane, drag the item on which to sort to the **Sort List** pane.

For example, you have a crosstab with Product line as rows, Order year as columns, and Revenue as the measure. To sort Order year by Revenue, drag Revenue to the **Sort List** pane.

 - Click the sort order button to specify ascending or descending order.

Working with Relational Queries

Queries specify what data appears in the report. In IBM® Cognos® Report Studio, you create and modify queries using **Query Explorer**. **Query Explorer** provides an alternative way to modify existing reports or to author new reports. You can use **Query Explorer** to perform complex tasks and other tasks that are more difficult to do in the report layout. For example, use **Query Explorer** to

- improve performance by changing the order in which items are queried from the database, or by changing query properties to allow the report server to execute queries concurrently where possible

Note: By default, queries run sequentially. Your administrator must enable the concurrent query execution feature. For more information, see the IBM Cognos Business Intelligence *Administration and Security Guide*.

- view or add filters and parameters and modify their properties
- view or add dimensions, levels, and facts
- incorporate SQL statements that come from other reports or reports that you write
- create complex queries using set operations and joins

Specify a List of Data Items for an Object

Specify the list of data items for an object when you must reference a data item that is in a query but is not in the layout. For example, you want to add a layout calculation to a list that uses a data item that is in the query definition. If the data item does not appear in the list, you must reference it in order to make the layout calculation work.

You must also specify the list of data items if you apply conditional formatting that uses a data item that is not in the query.

Steps

1. Click a layout object.

Tip: For a list of layout objects for which you can specify a list of properties, see the **Properties** property in "Report Studio Object and Property Reference" (p. 601).

2. In the Properties pane, double-click the **Properties** property and select data items.

Relate a Query to a Layout

Queries and layouts work together. After you decide the type of data that you need, you must create a layout in which to show the results. Each column of data must be both selected for the query and shown in a layout unless there are some columns that you do not want to show. The query and layout portions of a report must be linked to produce a valid report.

Report Studio automatically links query and layout. For example, when you use Report Studio and the list report layout, query and layout are automatically linked.

Steps to Manually Link a Query to a Data Container

1. Select a data container.
2. In the Properties pane, set the **Query** property to a query.
3. In the Insertable Objects pane, on the **Data Items** tab , drag data items from the query to the data container.

Connecting Queries Between Different Data Sources

IBM Cognos BI supports the following:

- RDBMS to RDBMS joins
- set operations of any two queries

- master detail relationships between any two queries
- drill from any query to any other query

You cannot create the following types of joins:

- cube-to-cube (homogeneous)
- cube-to-cube (heterogeneous)
- cube-to-RDBMS
- cube-to-SAP BW
- SAP-BW-to-RDBMS

Add a Query to a Relational Report

You can create multiple queries in **Query Explorer** to suit your particular needs. For example, you can create a separate query for each data container in a report to show different data.

Tip: The Briefing Book sample report in the GO Sales (analysis) package and the Top 10 Retailers for 2005 sample report in the GO Data Warehouse (analysis) package include multiple queries. For more information about The Great Outdoors Company samples, see "["Sample Reports and Packages" \(p. 567\)](#)".

Steps

1. Pause the pointer over the query explorer button  and click **Queries**.
2. In the **Insertable Objects** pane, drag one of the following objects to the work area.

Object	Description
Query	Adds a query.
Join	Adds a join relationship .
Union	Adds a union operator .
Intersect	Adds an intersect operator .
Except	Adds an except (minus) operator .
SQL	Adds SQL commands .

3. In the **Properties** pane, set the object properties.

For example, if you added a [join](#), set the **Join Relationships** property to define the join.

4. Double-click a query.

5. In the **Insertable Objects** pane, on the **Source** tab , drag data items to the **Data Items** pane.

Tip: You can add data items to the query that you do not want to appear in the layout. For example, to filter on Product line code and show Product line in the layout, you must add both data items to the query.

6. To create a new data item, in the **Insertable Objects** pane, on the **Toolbox** tab , drag **Data Item** to the **Data Item** pane.
7. To add a filter, in the **Insertable Objects** pane, on the **Toolbox** tab, drag **Filter** to the **Detail Filters** or **Summary Filters** pane and define the filter expression.

Tip: You can also create a filter by dragging a data item from the **Source** tab to one of the filters panes and completing the filter expression. If you are working with a dimensional data source, you can quickly filter data by dragging a member to a filter pane instead of dragging the level to which the member belongs. For example, dragging the member 2006 from the Years level will filter data for the year 2006. This method is quicker than dragging the Years level and specifying data only for the year 2006 in the filter expression.

When adding queries to the report

- right-click the work area and click **Show Package Sources** to see the queries that use data items from a package
- right-click the work area and click **Expand References** to see the relationships that exist between queries in the report, which is useful when you are creating complex queries

Create a Union Query

Create a union query to combine two or more queries into one result set.

You can combine queries that use different data sources. For example, you can combine a query that returns data from a dimensional data source with a query that returns data from a relational data source.

To combine two queries, the following conditions must be met:

- The two queries must have the same number of data items.
- The data items must have compatible types and appear in the same order.
For numeric data types, integer, float, double, and decimal are compatible.
For string data types, char, varChar, and longVarChar are compatible.
For binary data types, binary and varBinary are compatible.
Date data types must match exactly.

Steps

1. Pause the pointer over the query explorer button  and click **Queries**.
2. In the **Insertable Objects** pane, do the following:
 - Drag **Query** to the work area.

- Drag **Union**, **Intersect**, or **Except** to the right of the query.

Two drop zones appear to the right of the operator.

- Drag a **Query** object to each drop zone.

Two queries are created in the work area, and a shortcut to each query appears in the drop zones.

3. Double-click each query that makes up the union query and **add data items** to the query.

4. Return to the Queries work area.

5. Click the set operator that you added in step 2.

6. In the **Properties** pane, set the **Duplicates** property to remove or preserve duplicate rows.

7. Double-click the **Projection List** property.

The projection list shows the list of projected data items for the set operation.

8. To automatically produce the list of projected data items, click **Automatically generated**.

Report Studio generates the projection list using only one of the two queries in the union.

9. To add, delete, move, or rename data items in the projection list, click **Manual** and make the changes.

10. Double-click the union query.

11. In the **Insertable Objects** pane, on the **Source** tab , drag data items to the **Data Items** pane.

The union query is complete. You can now **link** the union query to a data container in the layout.

Example - Create a Two-column List Report for Three Data Items

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a list report that shows revenue for all product lines and order methods. However, you want the product lines and order methods to appear in a single column. To create this report, you use a union query to join the Product line and Order method data items.

Steps to Create a Report

1. Open Report Studio with the GO Data Warehouse (query) package.

2. In the Welcome dialog box, click **Create a new report or template**.

3. In the New dialog box, click **Blank**.

4. Pause the pointer over the query explorer button  and click **Queries**.

5. In the **Insertable Objects** pane, do the following:

- Drag **Query** to the work area.

- Drag **Union** to the right of the query.

Two drop zones appear to the right of the operator.

- Drag a **Query** object to each drop zone.

Query2 and **Query3** are created in the work area, and a shortcut to each query appears in the drop zones.

Steps to Define the Queries

1. Double-click **Query2**.
2. In the **Insertable Objects** pane, on the **Source** tab , drag the following data items to the **Data Items** pane:
 - from the **Product** folder, drag **Product line**
 - from the **Sales fact** folder, drag **Revenue**
3. In the **Insertable Objects** pane, on the **Toolbox** tab , drag **Data Item** to the **Data Items** pane.
The data item will be used to sort product lines and order methods in the report.
4. In the **Expression Definition** box, type '**A**' and click **OK**.
5. In the **Properties** pane, set the **Name** property to
Sort key
6. On the toolbar, press the back button  to return to the **Queries** work area.
7. Double-click **Query3**.
8. In the **Insertable Objects** pane, on the **Source** tab, drag the following data items to the **Data Items** pane:
 - from the **Order method** folder, drag **Order method**
 - from the **Sales fact** folder, drag **Revenue**
9. Repeat steps 3 to 5 to create a Sort key data item in **Query3** with '**B**' as its definition.

Steps to Define the Operator

1. On the toolbar, press the back button  to return to the **Queries** work area.
2. Click the **Union** operator.
3. Double-click the **Projection List** property.
The **Product line** item in the projection list contains both product lines and order methods.
4. Click **Manual**.
5. Click **Product line**, and then click the edit button .
6. In the **Edit** box, type the following after **Product line**:
& Order method

7. Click **OK** twice.

Steps to Define the List Properties

1. Double-click **Query1**.
2. In the **Insertable Objects** pane, on the **Source** tab, drag the following data items to the **Data Items** pane:
 - **Product line & Order method**
 - **Revenue**
 - **Sort key**
3. Select the **Sort key** data item and, in the **Properties** pane, set the **Pre-Sort** property to **Sort ascending**.

When you run the report, all product lines will appear first followed by all order methods.

4. Pause the pointer over the page explorer button  and click **Page1**.
 5. In the **Insertable Objects** pane, on the **Toolbox** tab, drag **List** to the work area.
 6. Click the list.
 7. Click the select ancestor button  in the title bar of the **Properties** pane and click **List**.
 8. In the **Properties** pane, set the **Query** property to **Query1**.
- The list is linked to the union query.
9. Double-click the **Properties** property.
 10. Select the **Sort key** check box and click **OK**.

Since the **Sort key** data item does not appear in the list, you must make it a property of the list before it can sort product lines and order methods.

11. In the **Insertable Objects** pane, on the **Data Items** tab , drag the following items from **Query1** to the list:
 - **Product line & Order method**
 - **Revenue**
12. Run the report.

A list report with two columns is produced. All product lines and order methods appear in the first column.

Product line: & Order Method	Revenue
Camping Equipment	1,589,036,664.03
Golf Equipment	726,411,367.89
Mountaineering Equipment	409,660,132.9
Outdoor Protection	75,994,296.25
Personal Accessories	1,885,673,307.78
E-mail	179,843,044.16
Fax	70,073,542.01
Mail	46,091,338.97
Sales visit	310,194,834
Special	27,351,320.25
Telephone	340,985,781.06
Web	3,712,235,908.4

Create a Join Relationship

You can create a join relationship to join two queries.

In general, join relationships should be created in the IBM Cognos Framework Manager model. Create a join relationship in Report Studio if what you are trying to do cannot be modeled in Framework Manager.

Steps

1. Pause the pointer over the query explorer button  and click **Queries**.
2. In the **Insertable Objects** pane, do the following:
 - Drag **Query** to the work area.
 - Drag **Join** to the right of the query.
 Two drop zones appear to the right of the operator.
 - Drag a **Query** object to each drop zone.
 Two queries are created in the work area, and a shortcut to each query appears in the drop zones.
3. Double-click each query that makes up the join query and **add data items** to the query.
4. On the toolbar, press the back button  to return to the Queries work area.
5. Click **Join**.
6. In the **Properties** pane, double-click the **Join Relationships** property.
7. Click **New Link**.
8. To create the link, click a data item in the left query, and then click a data item in the right query.
9. For each query, click **Cardinality** and click a cardinality option.
10. Click **Operator** and click an operator.
11. To convert the join relationship to an expression, click **Convert to expression**.

Convert the join relationship to an expression to make changes to the join definition.

Note: After you convert the relationship to an expression, you cannot change it back to a relationship.

12. Double-click the join query.
13. In the Insertable Objects pane, on the Source tab , drag data items to the Data Items pane.

The join query is complete. You can now [link](#) the join query to a data container in the layout.

Reference a Package Item in a Child Query

When you create a child query in Report Studio, you can only reference items from its parent or from other queries. For example, if you add a filter to a child query, the only items that you can insert into the expression are items that exist in other queries defined in the report. To add an item from the package, you must unlink the child query from its parent.

Steps

1. Pause the pointer over the query explorer button  and click **Queries**.
2. Unlink the child query from its parent by selecting the parent query shortcut to the right of the child query and clicking the delete button .
3. Double-click the child query.

Package items are now available on the Source tab  of the Insertable Objects pane.

4. Add package items.

For example, to create a detail filter that references a package item, drag the filter object from the Toolbox tab to the Detail Filters pane, and then add the items in the Expression Definition box.

5. Click the up arrow button in the toolbar to return to the queries work area.
6. Recreate the link between the child query and the parent query by dragging the parent query to the right of the child query.
7. If necessary, double-click the child query to complete it.

For example, if you created a filter, you may need to complete the filter expression.

Create a Master Detail Relationship

Create a master detail relationship to deliver information that would otherwise require two or more reports. For example, you can combine a list with a chart. The list can contain product lines and the chart can show details for each product line.

Master detail relationships must appear in nested frames to produce the correct results. You can create a master detail relationship in two ways:

- Use a parent frame for the master query and a nested frame for the detail query.

- Associate a report page with the master query and use a data container, such as a list or crosstab, for the detail query.

You cannot display parent data in the child frame or child data in the parent frame. Also, you cannot perform calculations across master detail queries.

You can use a master detail relationship to show data from separate data sources in a single report. However, the data sources must be contained in the same package.

If you are working with an SAP BW data source, you cannot use a data item from the master query that contains non-ASCII values.

To create a master detail relationship using queries that reference two different dimensional data sources, you must first [link the members](#) by aliasing the levels that contain the members.

Steps

- To use a parent frame for the master query and a nested frame for the detail query:

- In the **Insertable Objects** pane, on the **Toolbox** tab , drag a **List**, **Repeater Table**, or **Repeater** to the report.
- Add a second data container to the object you inserted.
You can insert a list, crosstab, chart, repeater table, or repeater into a list. You can add a list to a repeater table or repeater.
- Add data items to both data containers.

- To associate a report page with the master query and use a data container for the detail query:

- Click anywhere in the report page.
- In the **Properties** pane, click the select ancestor button  and click **Page**.
- Set the **Query** property.
- In the **Insertable Objects** pane, on the **Toolbox** tab, drag a data container to the report.

- To link a data item in the master query to a parameter in the detail query instead of to another data item, create the parameter in the detail query.

Use parameters to filter values at a lower level in the detail query.

- Pause the pointer over the query explorer button  and click the detail query.
- In the **Insertable Objects** pane, on the **Toolbox** tab, drag the **Filter** object to the **Detail Filters** box.
- In the **Expression Definition** box, create the parameter.

- Pause the pointer over the page explorer button  and click the report page.
- Click anywhere in the report page.
- In the **Properties** pane, click the select ancestor button and click **Page**.

7. Click the data container containing the details.
8. From the **Data** menu, click **Master Detail Relationships**.
9. Click **New Link**.
10. In the **Master** box, click the data item that will provide the primary information.
11. To link the master query to a data item, in the **Detail** box, click the data item that will provide the detailed information.
12. To link the master query to a parameter, in the **Parameters** box, click the parameter that will provide the detailed information.

Tip: To delete a link, select the link and press the Delete key.

13. If your detail query object is a chart, you can disconnect a chart title from the master query.
 - Click the chart title, and then, from the **Data** menu, click **Master Detail Relationships**.
 - Clear the **Use the master detail relationships from the chart** check box.
14. If your master detail relationship includes a chart as the detail and you are using automatic axis ranges, you can set the same axis range for all the detail charts:
 - Select the axis.
 - In the **Properties** pane, under **General**, double-click the **Axis Range** property.
 - Under **Maximum** and **Minimum**, click **Automatic**.

The master detail report option is only available with automatic axis ranges.

 - Under **Master detail reports**, select the **Same range for all instances of chart** check box.

For more information about axis ranges, see "[Change the Axis Scale of a Current Default Chart](#)" (p. 163).

Tip: To avoid seeing the same data item twice in the report, click the data item in the data container driven by the detail query and click the cut button. This removes the item from the report display but keeps it in the query.

Master Detail or Burst Reports with Charts or Crosstabs May Result in Denial of Service

When running a master detail or burst report that includes a chart or crosstab, disk space exhaustion may cause the report or other requests to fail. A large set of burst keys or master rows may produce one or more charts per detail, resulting in many master detail executions. This may cause the temp folder to accumulate many gigabytes of temporary files containing data required for successful chart rendering.

To avoid this issue, we recommend that you test large master detail or burst reports that include charts or crosstabs to determine the potential peak disk requirements for the report.

Working with Queries in SQL

For each query in a report, you can work with the SQL that is executed when you run a report.

You can

- [view the SQL for an entire report or a query](#)
- [build a report using your own SQL](#)
- [convert a report to use SQL directly](#)
- [edit the SQL](#)

When working with a relational data source, the SQL that is produced by Report Studio depends on the [report output selected](#).

Note: MDX is not available with relational or DMR data sources.

View the SQL for an Entire Report or a Query

View the SQL to see what is passed to the database when you run a report.

Steps

1. To view the SQL for the entire report, from the **Tools** menu, click **Show Generated SQL/MDX**.

This option shows the SQL that will be executed in the data source. The SQL is organized by query and by query result. If a query is used in more than one data container, a query result is generated for each data container.

2. To view the SQL for a specific query, do the following:

- Pause the pointer over the query explorer button  and click the query.
- In the **Properties** pane, double-click the **Generated SQL/MDX** property.

The **Generated SQL/MDX** property shows the SQL that is executed when you view tabular data (from the **Run** menu, click **View Tabular Data**). Tabular data shows the data that is produced by the query in the form of a list. You can use this property to help you build advanced queries.

The SQL for the query appears in the **Generated SQL/MDX** dialog box. You can choose to view native SQL, which is the SQL that is passed to the database when you execute the query, or Cognos SQL, which is a generic form of SQL that Report Studio uses. Cognos SQL is converted to native SQL before the query is executed.

Build a Report Using Your Own SQL

You can build a report by adding SQL from an external source, such as another report.

If you edit the SQL of a query, you must change the **Processing** property for that query to **Limited Local**.

Steps

1. From the **File** menu, click **New**.

2. Click Blank.
3. Pause the pointer over the query explorer button  and click Queries.
4. In the **Insertable Objects** pane, drag **Query** to the work area.
5. In the **Insertable Objects** pane, to build an SQL query, drag **SQL** to the right of the query
Tip: You can drag **SQL** anywhere in the work area, and Report Studio will automatically create a query.
6. In the **Properties** pane, double-click the **Data Source** property and click a data source.
7. If required, set the **Catalog** property to the name of the catalog.
8. Double-click the **SQL** property and type the SQL.
9. Click **Validate** to check for errors.
10. In the **Properties** pane, set the **Processing** property to **Limited Local**.
11. Double-click the query.
If the SQL is valid, the data items defined in the SQL appear in the **Data Items** pane.
12. Pause the pointer over the page explorer button  and click a report page.
13. In the **Insertable Objects** pane, on the **Toolbox** tab, drag an object to the work area.
For example, drag a list, crosstab, chart, or repeater.
14. Click the data container.
15. In the **Properties** pane, click the select ancestor button  and click the container you just created.
For example, if you created a list, click **List**.
16. Set the **Query** property to the query for the report type.
17. In the **Insertable Objects** pane, on the **Data Items** tab , drag the items from the query you chose in the previous step to the data container.

Convert a Query to SQL

You can convert a query to SQL to edit it. You may want to do this to improve performance or to use SQL features that are not supported directly by Report Studio.

Converting a query to SQL is an irreversible process.

Steps

1. Pause the pointer over the query explorer button  and click the query.
2. In the **Properties** pane, double-click the **Generated SQL** property.
3. Click **Convert**.
4. Click **Validate** to check for errors.

5. In the Properties pane, set the Processing property to Limited Local.

Edit the SQL

You can edit the SQL for a query that has been created as an SQL query or that has been converted to SQL.

Steps

1. Pause the pointer over the query explorer button  and click the SQL item under the query.
2. Double-click the SQL item.
3. Make changes in the text box.
4. Click **Validate** to check for errors.

First-Rows Optimization

The SQL produced by Report Studio depends on the report format you choose. For example, if you specify HTML format, first-rows optimization is requested. All-rows is requested if you specify PDF.

It is important for database administrators and programmers to remember that Report Studio does not always use first-rows optimization. If you assume first-rows optimization is always requested, this can cause the RDBMS optimizer to process the query differently than you expect.

Using Relational Calculations

Insert a calculation to make your report more meaningful by deriving additional information from the data source. For example, you create an invoice report, and you want to see the total sale amount for each product ordered. Create a calculated column that multiplies the product price by the quantity ordered.

You build calculations in the expression editor using functions.

If a calculation is used in multiple reports, or by different report authors, ask your modeler to create the calculation as a standalone object in the model and include it in the relevant package.

You can add calculations to lists, crosstabs, and all other data containers. You can also add calculations directly to a page header, body, or footer. However, you must first associate a query to the page. For more information, see "[Add a Page to a Report](#)" (p. 535)

Units of Measure

When creating calculations in IBM® Cognos® Report Studio and IBM Cognos Query Studio, you may encounter problems with the units of measure. For example, the calculation Cost*Cost returns the unit of measure * instead of a currency unit of measure. To avoid this problem, change the format of the corresponding column to obtain the desired unit of measure.

Calculation Solve Order

When calculations in the rows and columns of a report intersect, calculations are performed in the following order:

- addition or subtraction
- multiplication or division
- aggregation (rollup)
- remaining arithmetic functions: absolute, round, round down, average, minimum, maximum, medium, count
- percentage, % difference (growth) or % of total
- rank, quartile, quantile, or percentile

If both calculations have the same precedence, for example, if they are both business functions, then the row calculation takes precedence.

Limitations of Calculations

You should use only the expressions and functions available in Report Studio, and follow their syntax.

Minimal checking exists for calculations. If your calculation uses an invalid expression, your report results may contain unexpected values.

In addition, you should define member summaries as follows:

summary_function (currentMeasurewithin setset_reference)

where *set_reference* is a level or set inserted from the **Source** or **Query** tabs of the **Insertable Objects** pane.

Unless otherwise required, *summary_function* should be the **aggregate** function. If you use an explicit summary function, you may encounter problems with measures and scenario or account dimension members (such as profit margin, distinct count, and so on) that have complex rollup rules, or members that do not roll up.

Know your data, and confirm with the owner of the cube where overriding the automatic aggregation is safe.

Because of the above limitations, summaries of calculations may not provide reliable values. For convenience, you may have to build reports where row summaries and calculated member columns intersect. In such reports, these intersections may contain unexpected values. In contrast, row calculations intersecting with column aggregates using the **aggregate** function are safe because the calculation is performed on the reliably summarized values.

Create a Simple Calculation

You can select items in your report and create simple calculations. In addition to simple arithmetic calculations, you can perform the following calculations:

Calculation	Description
Rollup	Summarizes, or rolls up, all the values in a hierarchy.
%	Calculates the value of a selected item as a percentage of another item.
% Growth	Calculates the difference between two items as a percentage.
% Of Base	This calculation is only available if you select two members from different hierarchies, each from a different edge. This calculation takes the first selected member from edge A and the second selected member from edge B. The calculation result of a percent of base should be such that all the values for member A will compare itself to the intersect value of member A and B.
Custom	Allows you to specify your own value when performing a simple arithmetic calculation. Also allows you to change the order of operands or type a custom name for the new calculated row or column.

Steps

1. Select the items in your report to calculate.
 2. Click the insert calculation button  and select the calculation to perform.
- Tip:** Calculations that are not applicable to the items you selected are greyed out.
3. To change the order of the operands or the name of the calculated item added to the report, click **Custom**.

The calculation appears as a new row or column in your report.

Insert a Query Calculation

Insert a query calculation into your report to add a new row or column with values that are based on a calculation. For example, you create a query calculation named Euros that converts dollars to euros by multiplying an existing dollar measure by a conversion rate. Euros can then be displayed to end users in a separate row or column.

Note: When creating an expression that will be used in a double-byte environment, such as Japanese, the only special characters that will work are ASCII-7 and ~ -- || - \$ ¢ £ ¬.

Quality of Service Indicators

Not all data sources support functions the same way. The data modeler can set a quality of service indicator on functions to give a visual clue about the behavior of the functions. Report authors can

use the quality of service indicators to determine which functions to use in a report. The quality of service indicators are:

- not available (X)

The function is not available for any data source in the package.

- limited availability (!!)

The function is not available for some data sources in the package.

- limited support (!)

The function is available for all data sources in the package but is not naturally supported for that data source. IBM Cognos BI uses a local approximation for that function. Because an approximation is used, performance can be poor and the results may not be what you expect.

- unconstrained (check mark)

The function is available for all data sources.

Using Quotation Marks in Literal Strings

When inserting literal strings in an expression, you must enclose the string in single quotation marks. If the string contains a quotation mark, it must be escaped. For example, if you want to insert the string ab'c, you must type 'ab'"c'.

Steps

1. In the Insertable Objects pane, click the Toolbox tab .

2. Drag Query Calculation to the report.

3. Type a meaningful name for your expression in the Name box.

For example, if you are calculating the difference between 2007 revenue and 2006 revenue, you could name your expression **2007 - 2006 Revenue**.

4. In the Available Components pane, define the calculation:

- To add data items that are not shown in the report, on the source tab , double-click the data items.
- To add data items that are in the report but not necessarily in the model, such as calculations, on the data items tab , double-click the data items.
- To add functions, summaries, and operators, on the functions tab , double-click elements.

Tip: To filter the visible functions, summaries, and operators in the Available Components pane, click the filter button  and filter by the function type, what the function returns, or what the function acts on.

Tip: You can also type the calculation directly in the Expression Definition box. When typing date values, ensure that the date format is correct for your database type.

For information about specific functions, see "[Calculation Components](#)" (p. 769).

5. Click the validate button .

Any validation errors appear on the **Errors** tab of the **Information** pane.

Tips:

- To show or hide the **Available Components** pane, click the available components button .
- To show or hide the **Information** pane, click the information button .
- To copy and paste expression components in the **Expression Definition** pane, you can use the copy button  and the paste button .

Create a Layout Calculation

Create a layout calculation to add run-time information to your report, such as current date, current time, and user name.

When creating layout calculations, only report functions are available to you in the expression editor. These functions are executed in the report layout, as opposed to the other functions which are executed in the query.

Steps

1. In the **Insertable Objects** pane, click the **Toolbox** tab .
2. Drag **Layout Calculation** to the report.
3. In the **Available Components** box, define the calculation:
 - To add data items that are not shown in the report, on the source tab , double-click data items.
 - To add data items that are in the report but not necessarily in the model, such as calculations, on the data items tab , double-click data items.
 - To add data items from a specific query, on the queries tab , double-click data items.
 - To add functions, summaries, and operators, on the functions tab , double-click elements.
 - To add a value that is derived from a parameter, on the parameters tab , double-click a parameter.

Parameters define prompts, drill-through reports, and master detail relationships.

Tip: You can also type the expression directly in the **Expression Definition** box.

For more information about creating expressions, see "["Calculation Components" \(p. 769\)](#)".

4. Click the validate button .

Chapter 11: Dimensional Reporting Style

The dimensional reporting style is recommended for dimensionally-modeled relational (DMR) and Online Analytical Processing (OLAP) data sources. Dimensional data is best represented by crosstabs, maps, and charts. This data is shown in IBM® Cognos® Report Studio in dimensions, hierarchies, levels, and members.

In dimensional reporting, you summarize data by using member summaries and within set aggregation. You focus data in dimensional reporting by adding only the relevant members to the edge of a crosstab or to the context filter. You can also enable drilling up and drilling down in dimensional reports.

IBM Cognos Analysis Studio and IBM Cognos Business Insight Advanced are also available to work with dimensional data.

You can perform the following tasks in dimensional reporting:

- [add data](#)
- [summarize data](#)
- [focus data with filters](#)
- [sort data](#)
- [add queries](#)
- [use calculations](#)
- [enable drilling up and down](#)

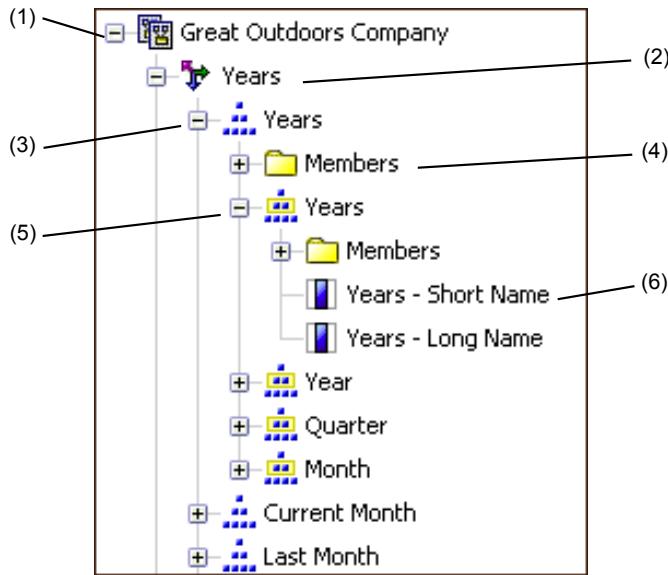
Add Dimensional Data to a Report

We recommend using dimensional data in the dimensional reporting style. However, if you are using relational data, see "[Add Relational Data to a Report](#)" (p. 307).

For dimensional and mixed model data sources, you can view the full data tree by clicking the view package tree button  in the Insertable Objects pane. You can switch to the dimensional-only data tree by clicking the view members tree button .

If you are working with a dimensional data source, data items are organized hierarchically. Dimensional data sources include OLAP and dimensionally-modeled relational (DMR) data sources.

The Source tab  in the Insertable Objects pane shows a metadata-based view of the data.



Note: The names of levels and members in a dimension come from the model. It is the responsibility of the modeler to provide meaningful names.

1. Package

Packages are subsets of a model containing items that you can insert in a report.

2. Dimension

Dimensions are broad groupings of descriptive data about a major aspect of a business, such as products, dates, or markets.

3. Level hierarchy

Level hierarchies are more specific groupings within a dimension. For example, for the **Years** dimension, data can be organized into smaller groups, such as **Years**, **Current Month**, and **Last Month**.

4. Members folder

Members folders contain the available members for a hierarchy or level. For example, the **Members** folder for the **Years** level hierarchy contains everything found in the **Year**, **Quarter**, and **Month** levels.

5. Level

Levels are positions within the dimensional hierarchy that contain information at the same order of detail and have attributes in common. Multiple levels can exist within a level hierarchy, beginning with a root level. For example, the **Years** level hierarchy has the following related levels.

Level	Level name	Description
Root	Years	The root level.

Level	Level name	Description
First	Year	Years in the Years root level. For example, 2004, 2003, and 2002.
Second	Quarter	Quarters for each year in the Year level. For example, 2004 Q1, 2004 Q2, and 2004 Q3.
Third	Month	Months for each quarter in the Quarter level. For example, Jan., Feb., and Mar.

Tip: The **Measures** dimension contains the measures available in the data source.

6. Member property

Member properties are attributes that each member possesses. For example, gender could be a property for all employee members. For more information about member properties, see ["Insert a Member Property" \(p. 352\)](#).

Customize the Source Tree

When using dimensional data, you can customize how you see data in the source tree. You can select a full package tree view that shows dimensions, hierarchies, levels, an optional members folder, and member properties.

Or you can select a members-only tree view, such as the one in IBM® Cognos® Analysis Studio. This view allows you to add only members in your report.

Steps

1. In the Insertable Objects pane, click the **Source** tab .
2. To view the members-only tree view, click the view members tree button .
3. To view the full package tree, click the view package tree button .
4. To change what is included in the tree, right-click and click **Package Tree Settings**.

Use the **Preview** pane to choose the appropriate settings.

Insert a Member

By default, when you insert members from the source tree into your report with IBM® Cognos® Report Studio, single members are inserted without any details or children. You can change how members are inserted. For example, you may want to insert a member along with all its children or insert only the member's children. Or you may want to insert members as a set.

When you double-click a member that you have already inserted in your report, by default, its children are inserted after the parent member. You can control whether the children are inserted

before or after, nested, or not inserted at all: from the **Tools** menu, click **Options**, and then choose a setting on the **Edit** tab.

When inserting members into a crosstab, ensure that you insert members from the same hierarchy on only one of the crosstab edges. If you insert members from the same hierarchy on both edges of the crosstab, you may encounter unexpected results. For example, a report that uses members from years in the rows and Quarters in the columns is very difficult to read because the useful numbers are distributed over a large area of mostly blank cells.

Calculated members from the data source also appear in the source tree. However, calculated members in Microsoft® SQL Server Analysis Services (SSAS) cubes do not appear in the list of descendants in the query. If you want these calculated members to appear in a report or a prompt, you must insert them explicitly.

Steps

1. In the **Insertable Objects** pane, from the **Source** tab , click the insert single member button  and select how to insert members.
2. If you want to insert members within a set instead of inserting individual members, click the create sets for members button .
3. Do one of the following:
 - To insert a member above or below another member, drag the new member above or below a cell.
A flashing black bar appears where you can drop the new member.
 - To replace a member, drag the new member to the cell with the member to replace.
A flashing black box appears where you can drop the new member.

Search for a Member

You can perform a member search to quickly find data.

You can control the number of members that are returned in a search by specifying a value for the **Member display count limit (in source tree)** option (**Tools**, **Options**, **Advanced** tab).

Steps

1. In the **Insertable Objects** pane, on the **Source** tab , select and then right-click a hierarchy  or level  and click **Search**.
2. In the **Words** box, type the words or characters for which to search.
3. Click the search parameter to use.
4. To search all descendants instead of just the immediate children, select the **Search all descendants** check box.

For example, when performing a search in a hierarchy, selecting this check box returns members found in all the levels of the hierarchy.

5. Click Search.

The search results appear in a hierarchical structure on the **Search tab**  in the **Insertable Objects** pane. You can browse the hierarchy to explore members at lower levels.

Tip: You can insert members directly from the **Search tab** into a report. This can save you time, because you do not have to define a filter. For example, instead of inserting **Product line** from the **Source tab** and adding a filter for **Camping Equipment**, you can insert **Camping Equipment** from the **Search tab**.

Nest Members

When you insert members in your report, you may want to nest them in another row or column to make your report easier to use. You can nest members from different dimensions. You can also nest sets.

For example, in the following report, the quarters (Q1 to Q4) were selected from the Time dimension and nested only for the GO Americas member, which is from a different dimension.

Amount year to date	<#2004#>
<#GO Americas#>	<#Q1#> <#1234#> <#Q2#> <#1234#> <#Q3#> <#1234#> <#Q4#> <#1234#>
<#GO Consolidated corporate#>	<#1234#>
<#GO Consolidated eliminations#>	<#1234#>
<#GO Consolidated#>	<#1234#>
<#GO Asia Pacific#>	<#1234#>

When you filter out children from nested sets, the parent set still appears in your report. To avoid this, filter only the top level set and nest only the complete set of descendants at the desired levels.

Step

- In the **Insertable Objects** pane, on the **Source tab** , drag the members to the work area.
A flashing black bar indicates where you can drop an item.

Insert a Hierarchy

You can quickly insert entire hierarchies in a report.

When using an IBM® Cognos® PowerCube, a SSAS cube, or a dimensionally-modeled relational data source, you can insert multiple hierarchies from the same dimension in a crosstab. For example, you can place one hierarchy from a dimension on an edge of the crosstab and nest another hierarchy from the same dimension on the same edge, on another edge, or in the **Context filter** area.

You can also perform arithmetic, percentage, and ranking calculations using multiple hierarchies.

Limitations When Inserting Multiple Hierarchies from the Same Dimension

Summary or analytic calculations that refer to different hierarchies from the same dimension, such as total, count, average, minimum, and maximum, are not supported. These calculations cause an OP-ERR-0250 error.

Steps

1. In the **Insertable Objects** pane, on the **Source** tab , drag the hierarchy  to the report.
2. In the **Insert Hierarchy** dialog box, choose which members to insert:
 - To insert only the root members of the hierarchy, click **Root members**.
 - To insert all members of the hierarchy, click **All members**.

Tip: The default **Level Indentation** property for all members is to show the hierarchical structure when the report is run.

Insert a Member Property

You can insert member properties in a report. For example, an employee level may have a property named gender. In the **Insertable Objects** tree, member properties are identified by the icon .

Step

- In the **Insertable Objects** pane, on the **Source** tab , drag the member property  to the report.
You can only insert properties. You cannot group them.

Create a Set of Members

Use sets to group members that are logically related for various actions, such as formatting, nesting, and sorting. Creating sets is also useful when members may be dynamic over time. For example, the child accounts of a total assets account may change from year to year. By creating a set, you do not have to modify the report each time accounts are added or removed.

After you create a set of members, you can add or remove members within the set.

You can perform operations on the set such as excluding specific members, moving members within the set, showing only the top or bottom members, filtering the set, and expanding or collapsing the members within a set. You can view the set definition to see and edit operations that are performed on the set.

Steps

1. In the **Insertable Objects** pane, click the create sets for members button  to toggle between adding individual members and creating sets for members.
2. In the **Insertable Objects** pane, select the items to include in the set and drag them to the work area.
3. to add or remove members from the set, right-click the set and click **Edit Members**.

Tip: you can also select the set and, in the **Properties** pane, double-click the **Members** property.

Edit a Set

After you create a set, you can view its definition to see, change, add, or delete operations, such as the following, that are performed on the set:

- excluding a member
- moving members to the top or bottom of a set
- applying a top or bottom filter
- filtering the set
- expanding or collapsing members in a set

The definition of a set shows you, in the graphical tree, the history of all the operations that are performed on the set.

You can see and change the order in which the operations are performed. For example, you apply a top three filter and then exclude one member. Your set now contains only two members. You can choose to exclude the member before the top three filter so that your set still contains three members.

Steps

1. Right-click a set and click **Edit Set**.

A graphical tree shows all the operations that are performed on the set of members.

2. To see the details of an operation, hover your mouse over an operation node.
3. To change the order of an operation, click the operation node, and click the right or left arrow.
4. To edit an operation, click the operation node and click the edit button .
5. To add a new operation, click the new button .

Exclude Members from a Set

You can choose to remove individual items that are not needed in your analysis.

Steps

1. Right-click the set and click **Edit Set**.
2. Click the new button  and click **Exclude**.
3. In the **Exclude** box, choose the members to exclude and use the right arrow to move them to the **Members** pane.

Move Members Within a Set

You can move one or more members to the top or bottom of a set.

Steps

1. Right-click the set and click **Edit Set**.
2. Click the new button  and click **Move to Top** or **Move to Bottom**.
3. In the **Available members** list, select the member to move to the top or bottom and click the right arrow.

The members appear in the order shown in the **Members** list.

Limit Data to Top or Bottom Values

You want to focus your report on the items of greatest significance to your business question. For example, you want to identify your top 100 customers and what that group of customers is worth.

You can limit the data to the top or bottom values of a set. This keeps the amount of data shown in the work area small, even when using large data sources.

You can define a top or bottom rule by specifying

- a number, such as the 50 top or bottom performing sales people
- a percentage, such as customers who contribute to the top 10% of overall revenue
- a cumulative sum, such as customers who contribute to the first ten million dollars of overall revenue

If the selected set contains a user-defined filter, the top or bottom rule applies only to the included values. For example, if you apply a filter to show only retailers with revenue greater than one million dollars, the bottom rule applies to the lowest values within those results.

You can filter a set of members to show only those at the top or bottom and base the filter on the measure you are using and the set of members on the opposite edge. You can also specify a custom filter.

Steps

1. Right-click the set and click **Edit Set**.
2. Click the new button  and click **Top or Bottom**.
3. Under **Type**, select whether to filter the **Top** or **Bottom** values.
4. To choose how and how much you want to filter, under **Number of items**, click **Count**, **Percent**, or **Sum** and type the number of data items to show, the percent of data items to show, or the value to sum.
5. Choose the intersection on which to base your filter by clicking the ellipsis (...) button and selecting the members and measure for the intersection from either the **Source** tab or the **Calculated Members and Measures** tab.

For more information, see "[Create an Intersection \(Tuple\)](#)" (p. 400).

Filter the Members Within a Set

You can remove members from within a set by specifying filter conditions so that only the members that you require remain. You can filter a set based on

- member captions

For example, your set includes employee names and you want to keep only the employees with names that begin with a specific letter.

- properties that are numeric or strings

For example, your set includes employee names and you want to filter employees using the gender property.

- intersection of values (tuple)

For example, you want to keep only the employees who used less than ten sick days for the year 2008.

If you want to create a more complex filter, you can combine multiple conditions using AND, OR, and NOT operators. By default, multiple conditions are combined with an AND operator, which means that all conditions must be met for the filter to take effect.

Filtering the members in a set is not the same as relation detail or summary filters.

Steps

1. Right-click the set and click **Edit Set**.
2. Click the new button  and click **Set Filter**.
3. Select how you want to filter.
 - If you want to filter the set using a caption, click **Caption**.
 - If you want to filter the set using a member property, click **Property** and select from the drop-down list.
 - If you want to filter the set using an intersection of members, or tuple, click **Intersection (tuple)** and click the ellipsis (...) button. Then, from **Available members and measures**, select the desired items and click the right arrow to move them to the **Intersection members and measures** list.
4. Specify the **Operator** and **Value** to complete your condition and click **OK**.
5. To add additional conditions, click the new button .
6. If you include multiple conditions in your filter, use the AND, OR, and NOT operators to combine them.
 - To add an operator, select the conditions that you want to combine, and click an operator.
 - To change an operator, double-click the operator to toggle among available ones or select the operator and then select a different one from the list.

- To remove an operator, select it in the condition string, and then click the delete button .
7. To change a condition, click the edit button .

Expand and Collapse a Member Within a Set

You can expand a member to add its child members below it as new rows, and you can collapse expanded members to remove the expanded member's child members.

Steps

1. Right-click the set and click **Edit Set**.
2. Click the new button  and click **Expand or Collapse**.
3. In the **Available members** list, select the member to expand or collapse and click the right arrow to move it to the **Members** list.

Extended Data Items

Note: Extended data items do not apply to relational packages or non-dimensionally modeled packages.

Extended data items differ from the traditional, expression-based text strings by enabling you to do the following:

- Add an extended data item's child members

You can double-click an extended data item to insert its child members beside it in the report object.

- View an extended data item's details

You can select an extended data item and see where the data item appears in the package in the **Properties** pane under **Data Item**.

- Select member sets

You can select a member in a set to highlight it as the primary selection and the other members as secondary selections.

- Sort sets

You can select an extended data item, click the sort button , and click **Advanced Set Sorting** to sort a set in ascending or descending order, to sort hierarchically, and to sort by caption, by a property, or by an intersection (tuple).

- Apply contextual calculations

You can select an extended data item and then, from the **Data** menu, click **Calculate** to see a contextual list of calculations that are available for this extended data item.

Extended data items are optional. If you use extended data items in a report, we recommend that you use them for the entire report instead of mixing them with expression-based data items. For information about turning on extended data items, see "[Specify Report Properties](#)" (p. 458).

Note: In IBM® Cognos® Business Insight Advanced, extended data items are always used. Therefore, if you are authoring a report in IBM Cognos Report Studio for report consumers who are working in Business Insight Advanced, you should use extended data items.

Tips for Working with Ragged or Unbalanced Hierarchies

In ragged or unbalanced hierarchies, some members that are not at the lowest level of the hierarchy may have no descendants at one or more lower levels. Support for these hierarchy gaps in relational data sources is limited. More complete support is provided for OLAP data sources, but some reports may still result in unexpected behavior. For example, the following may occur:

- Groups corresponding to missing members may appear or disappear when grouped list reports are pivoted to a crosstab. This happens with set expressions using the `filter` function, and detail filters on members.
- Ragged and unbalanced sections of the hierarchy are suppressed when set expressions in that hierarchy are used on an edge.
- When a crosstab is sectioned or is split into a master detail report, sections corresponding to missing members become empty.
- Cells that were suppressed may still appear in the report output for reports with ragged or unbalanced hierarchies.

Some of these behaviors may be corrected in a future release, while others may be codified as supported behavior. To avoid these behaviors, do not use levels from ragged or unbalanced hierarchies. Instead of using levels, use the descendants, children, or ancestors.

We consider the following scenarios to be safe:

- One or more nested level references on an edge with no modifying expression.
- A hierarchy reference on only one level of one edge.
- One or more explicit members or sets of explicit members as siblings on only one level of one edge.
- Summaries of the previous three scenarios.

In all cases, you should test reports based on ragged and unbalanced hierarchies to confirm that hierarchy gaps are handled correctly.

For more information about ragged or unbalanced hierarchies, see the IBM® Cognos® Framework Manager *User Guide*.

Limited Support for Relational Functions When Used with OLAP Data Sources

When working with an OLAP data source, we recommend that you not use relational functions, such as `substring` and concatenation functions, in a report that also contains a measure with the

Aggregate Function property set to **Calculated** or **Automatic** in the model. If you do so, you may encounter unexpected results. For example, some summaries are calculated using the **Minimum** function instead of the aggregate function derived from the individual query items.

In the expression editor, an exclamation mark (!) that precedes a function indicates that the function is not naturally supported for that data source. IBM® Cognos® BI uses a local approximation for that function. Because an approximation is used, performance can be degraded and the results may not be what you expect.

For example, you create an IBM Cognos Query Studio report that contains the data items Product line and Retailer site count. The footer summary is set to **Calculated**. You then insert a calculated column that returns the first three characters of the Product line item, which uses the relational concatenation function. The footer summary now shows the lowest gross margin value.

Product line	First Three Characters	Retailer site count
Camping Equipment	Cam	632
Mountaineering Equipment	Mou	265
Personal Accessories	Per	810
Outdoor Protection	Out	639
Golf Equipment	Gol	367
Summary		265

For more information about aggregation functions, see the *Query Studio User Guide* or the IBM Cognos Report Studio *User Guide*.

Summarizing Data Dimensionally

Summarize data in your reports to obtain totals, averages, and so on.

The summary options that you can use depend on the type of data source that you are using. If you are querying an OLAP data source, all measure values that appear in reports are pre-summarized because the data source contains rolled up values. The type of summary that is used is specified in the data source itself. As a result, we recommend that you use the Automatic summary when creating dimensional style reports. This ensures that the report always uses the type of summary that the data modeler specified in the data source.

For example, the modeler may have specified that the rollup for revenue is total and the rollup for stock prices is average.

If you use other types of summaries with dimensional style reports, you may encounter unexpected results.

You can also add summaries, which are supported for any data source. Summaries specify how data items are totaled in the headers and footers of a list and in the total rows and columns of a crosstab. For list reports, these summaries only summarize the data that is visible on that page of the report.

Revenue	2004	2005	2006	Total(Order year)
Camping Equipment	\$20,471,328.88	\$31,373,606.46	\$37,869,055.58	\$89,713,990.92
Golf Equipment	\$5,597,980.86	\$9,598,268.88	\$10,709,215.84	\$25,905,465.58
Mountaineering Equipment		\$9,642,674.54	\$11,248,676.06	\$20,891,350.60
Outdoor Protection	\$1,536,456.24	\$988,230.64	\$646,428.04	\$3,171,114.92
Personal Accessories	\$7,144,797.52	\$10,955,708.04	\$13,793,960.30	\$31,894,465.86
Total(Product line)	\$34,750,563.50	\$62,558,488.56	\$74,267,335.82	\$171,576,387.88

You can specify summary aggregation in different ways by using any of the following:

- aggregation properties that are specified in the model
- the **Auto Group & Summarize** property
- the summarize button in the toolbar
- aggregation properties for a data item
- the solve order of calculations

Limitation

If a summary is applied to a report that contains binary large object (BLOB) data, such as images or multimedia objects, you cannot also perform grouping or sorting.

The Type of Data

How data is summarized also depends on the type of data that you are summarizing. Summary rules are applied differently to facts, identifiers, and attributes. For example, if you summarize a data item that represents part numbers, the only summary rules that apply are count, count distinct, count non-zero, maximum, and minimum. For information about how to determine the type of data that a data item represents, see "[Add Relational Data to a Report](#)" (p. 307) and "[Add Dimensional Data to a Report](#)" (p. 347).

For information about how null or missing values are handled in summaries, see "[Null \(Missing\) Values in Calculations and Summaries](#)" (p. 407).

Limitations on Measure Rollups

For all OLAP data sources except IBM® Cognos® PowerCube and Microsoft® SQL Server 2005 Analysis Services (SSAS), aggregation and re-aggregation are supported only for calculations and measures that use the following rollups: Sum (Total), Maximum, Minimum, First, Last, and Count.

All other types of rollup either fail or return error cells, which typically appear as two dash characters (--) .

This problem occurs in, but is not limited to, the following:

- footers
- aggregate function
- context filters that select more than one member of a hierarchy that is used elsewhere on the report

Summarizing Sets

When working with Microsoft SQL Server Analysis Services (SSAS) data sources, We recommend that you not summarize values for sets which contain members that are descendants of other members in the same set. If you do so, SSAS double-counts values for automatic summaries and all data sources double-counts values for explicit summaries.

Add a Simple Summary

You can add simple summaries in a report by using the summarize button . This button provides a subset of the summary functions available in IBM Cognos Report Studio.

The summarize button sets the rollup aggregate property for the data item to the selected summary, and places the data item into an appropriate footer. A footer is created for each set, hierarchy, or level.

For information about adding a rolling or moving average, see "["Rolling and Moving Averages" \(p. 364\)](#)".

Steps

1. Click the item to which to add a summary.
2. Click the summarize button  and click a summary type.
3. To change the summary label, do the following:
 - Click the label.
 - In the **Properties** pane, under **Text Source**, set the **Source Type** property to the source type to define the label.
For example, set it as **Data Item Value** to produce a dynamic label for the summary based on data item values.
 - Set the property below **Source Type** to specify the label.

This property depends on the source type you chose. For example, if you chose **Data Item Value** as the source type, set the **Data Item Value** property to the data item to use to define the label.

In crosstabs and charts, the summary appears as a node.

Tips

- To change a summary, select it and, in the **Properties** pane, under **Data Item**, click **Aggregation Method** and choose a different function.
- In crosstabs, you can add multiple summaries at the same level. For example, you have a crosstab with Product line as rows, Order year as columns, and Revenue as the measure. For Product line, you can add the **Total** summary as a header, which will total all revenue for each order year. You can then add the **Average** summary as a footer, which will give the average revenue of all product lines for each order year.

Summarizing Values in Crosstabs

In crosstabs, summaries are calculated using one of the following summary expressions:

- `aggregate ([measure] within set [data item])` summarizes the member values from the data source within the current content.
- `aggregate ([measure] within detail [data item])` summarizes the lowest level of details in the report.
- `aggregate ([measure] within aggregate [data item])` summarizes each level of details in the report.

For example, in the following crosstab, if you specified **Total** as the summary, these aggregate expressions are produced for each aggregation mode:

	<#Revenue#>	<#Revenue#>
<#Quarter#>	<#Month#>	<#1234#>
	<#Month#>	<#1234#>
<#Total(Quarter)#>	<#1234#>	<#1234#>

- `Total ([Revenue] within set [Quarter])`

This expression totals the quarter values from the data source at the intersecting product line. At the bottom right corner, it totals the aggregate over all product lines for each quarter.

- `Total ([Revenue] within detail [Quarter])`

This expression totals the month values visible in the report at the intersecting product line. At the bottom right corner, it totals all of the intersecting month - product line values visible in the report.

- `Total ([Revenue] within aggregate [Quarter])`

This expression totals the month values visible in the report at the intersecting product line into quarters. At the bottom right corner, it does the same, but starting with the aggregate over all product lines for each month.

In simple cases, the members and values visible in the report and the aggregate rules in the report are the same as those in the data source and all of these expressions produce the same results.

For example, for the quarter and month values, if you are totaling the values for all months in all quarters, it makes no difference whether the visible values, the values in the cube, or the month and quarter values are used. The result is the same.

		Revenue
2005 Q 3	2005/Sep	\$5,300,599.08
2005 Q 4	2005/Oct	\$11,778,348.92
	2005/Nov	\$3,353,399.48
	2005/Dec	\$4,645,069.28
2006 Q 1	2006/Jan	\$3,119,227.22
	2006/Feb	\$3,045,478.88
	2006/Mar	\$6,867,919.18
2006 Q 2	2006/Apr	\$8,306,586.26
	2006/May	\$5,129,112.24
	2006/Jun	\$6,927,447.24
2006 Q 3	2006/Jul	\$5,078,789.96
	2006/Aug	\$5,180,166.38
	2006/Sep	\$4,892,742.38
2006 Q 4	2006/Oct	\$15,887,237.28
	2006/Nov	\$3,609,599.42
	2006/Dec	\$6,240,652.88
Total(Quarter)		\$171,576,387.88

Tip: The Total Revenue by Country sample report in the GO Data Warehouse (query) package includes a total summary function. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Different results appear when you start filtering, changing aggregation types, or using set expressions or unions.

For example, the following crosstab shows the quantity of products sold across all product types for each product line. The bottom of the crosstab has three summary values that show the average quantity of products sold by product line.

		Quantity
Camping Equipment	Cooking Gear	13,400,351
	Tents	3,164,285
	Sleeping Bags	3,153,218
	Packs	2,756,540
	Lanterns	4,826,755
Mountaineering Equipment	Rope	418,560
	Safety	1,135,194
	Climbing Accessories	5,850,251
	Tools	2,496,086
Personal Accessories	Watches	4,299,195
	Eyewear	20,311,396
	Knives	7,229,314
	Binoculars	1,075,087
	Navigation	1,992,713
Outdoor Protection	Insect Repellents	5,800,964
	Sunscreen	5,384,127
	First Aid	829,354
Golf Equipment	Irons	391,445
	Woods	317,939
	Putters	1,284,570
	Golf Accessories	3,119,747
Average(Product line) - within detail		4,249,385.29
Average(Product line) - within aggregate		4,040,006.76
Average(Product line) - within set		17,847,418.2

Each summary value uses a different aggregation mode that is indicated in the summary name.

- Average (Product line) - within detail

This summary is the average of the detail values in the crosstab.

- Average (Product line) - within aggregate

For each product line, the average of the detail values is calculated. This is equivalent to applying the average aggregation function to the Product type column. This summary is the average of the product type averages.

- Average (Product line) - within set

This summary is the average of the product type values rolled up into sets at the Product line level. The values are obtained from the data source. If filters or slicers exist, the values are recomputed using the aggregation rules defined in the data source.

In most cases, you should use the within detail aggregation mode because the results are easiest to understand and the same as the results for footers in a grouped list report. In more complex cases, you may consider the within aggregate aggregation mode. The within set aggregation mode should be reserved for reports with a purely dimensional focus such as when there are no detail or summary filters defined in the report.

Summarizing Member Sets

When you work with dimensional data sources and you summarize member sets, if an explicit summary function such as **Total** is used and the set contains duplicates, the result is double-counted. If you are using an OLAP data source, the result produced for the automatic summary function depends on the data source.

For example, the product line rows below were defined using the expression

```
union ([Product line], [Camping Equipment], ALL)
```

where [Product line] is the level that contains Camping Equipment.

	Revenue
Camping Equipment	1,589,036,664.03
Mountaineering Equipment	409,660,132.9
Personal Accessories	1,885,673,307.78
Outdoor Protection	75,994,296.25
Golf Equipment	726,411,367.89
Camping Equipment	1,589,036,664.03
Aggregate(Product line)	4,686,775,768.85
Total(Product line)	6,275,812,432.88

For PowerCubes, Aggregate (Product line) is the sum of the product lines excluding duplicates.

For more information about how the aggregate function is processed, see "["Summary Functions" \(p. 365\)](#).

Limitations When Summarizing Measures in DMR Data Sources

There are limitations when summarizing dimensionally-modeled relational (DMR) measures and semi-additive measures in crosstabs using the aggregation function count distinct, median,

standard-deviation, or variance. The following limitations can produce empty or error cells when the report is run:

- The aggregation function must apply to all members of a level or all children of a member.
- To use OLAP functions in detail filters that are applied to a dimension that is not in the report, or is at a level below what is being reported, only the functions `children`, `level`, `members`, `roleValue`, and `rootMembers` will work.
- You cannot define detail filters that reference one or more measures and are set to **After auto aggregation**.
- You cannot define context filters that have more than one member from a dimension that does not appear in the report.
- Context filters that have more than one member from a dimension that appears in the report produce errors in all cells that are ancestors of the slicer members.
- If a crosstab has a row that is a set of members from one dimension (dimension A) and another row that is a set of members from another dimension (dimension B), and a context filter containing members from dimension A is defined, error cells are produced in the row that contains members from dimension B.
- If a context filter contains members from a dimension, and a crosstab has a row that is a set of members from a higher level than the slicer members, error cells are produced for that row.
- Error cells are produced when drilling down on a crosstab that has two nested levels.

If you do not consider these limitations in a calculation, the report may return inaccurate results.

If there is a non-measure calculation that returns a constant or contains a summary function, and the calculation has a lower solve order than the measure being aggregated, error cells are returned for the aggregated measure.

In list reports, error cells are produced as a result of these limitations if the list uses an OLAP function other than `children`, `filter`, `level`, `members`, `roleValue`, and `rootMembers`.

Rolling and Moving Averages

Use rolling and moving averages to analyze data for specific time series and to spot trends in that data. When viewing these averages on a line chart, use a longer period of time to reveal long-term trends.

In IBM® Cognos® Report Studio, to add a rolling or moving average, you must create summary and custom calculations using layout expressions "[Insert a Query Calculation](#)" (p. 398).

The Rolling and Moving Averages interactive sample report includes rolling and moving calculations. For more information about The Great Outdoors Company, see "[Sample Reports and Packages](#)" (p. 567).

Rolling Average

A rolling average continuously updates the average of a data set to include all the data in the set until that point. For example, the rolling average of return quantities at March 2004 would be

calculated by adding the return quantities in January, February, and March, and then dividing that sum by three.

Tip: The Historical Revenue sample report in the Sales and Marketing (Cube) package also includes a rolling average. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Moving Average

A moving average calculates the average of a data set for a specified period. For example, the moving average of return quantities at March 2004 with a specified period of two would be calculated by adding the return quantities in February and March and then dividing that sum by two. In Report Studio, you can use a prompt to specify the period.

Show Data for Specific Time Periods

Show data for a specific time period to associate a report with a business time period rather than the execution time. For example, you have a monthly report that you run at the beginning of each month, and you want the last business day of the previous month to appear rather than the day on which you run the report.

If the report functions `AsOfDate` and `AsOfTime` are added to the report, they will return a value based on the results of the **As of Time Expression** object. If the **As of Time Expression** object is not added to the report, these two functions return the date and time at which the report is run.

If the **As of Time Expression** object is added more than once in the report, the first occurrence of the object in the layout that returns a valid value is used.

Steps

1. In the Insertable Objects pane, on the **Toolbox** tab , drag **As of Time Expression** to the report.
2. Double-click **As of Time Expression**.
3. In the **Expression Definition** box, type the expression.

The expression must return a date-time value.

Tip: You can drag a function that returns a constant from the **Constants** folder in the **Functions** tab. You can then change the constant to the value to use.

If you do not specify a time, the default time 12:00:00.000 AM is used.

Summary Functions

This section describes the summary functions that are available in Report Studio. Some functions, such as **Custom**, are available only when you click the summarize button  on the toolbar. Additional summary functions are available in the expression editor.

Automatic Summary

Depending on the type of data item, applies the function **None**, **Summarize**, or **Calculated** based on the context in which the data item appears.

Calculated is applied if the underlying data source is OLAP. It is also applied if the data item expression

- contains a summary function
- is an `ifthenelse` or `case` expression that contains a reference to at least a modeled measure in its condition
- contains a reference to a model calculation or to a measure that has the **Regular Aggregate** property set to a value other than **Unsupported**
- contains a reference to at least one data item that has the **Rollup Aggregate Function** property set to a value other than **None**

If the underlying data source is relational and if the data item expression contains no summary functions and a reference to at least one fact that has the **Regular Aggregate** property set to a value other than **Unsupported**, **Summarize** is applied.

If the underlying data source is OLAP and if the data item expression contains no summary functions and a reference to at least one fact that has the **Regular Aggregate** property set to a value other than **Unsupported**, **Calculated** is applied.

If the underlying data source is SAP BW, reports containing aggregations and summaries run more efficiently if the aggregation applied to a query item matches the aggregation rule for the underlying key figure on the SAP BW server. In Report Studio, the easiest way to accomplish this is to change the value of the **Aggregate Function** property to **Automatic**.

In all other contexts, **None** is applied.

For relational and dimensionally modeled relational (DMR) data sources, if this function is applied to a data item expression that has the `average` function, weighted averages are computed based on the lowest level detail values in the data source.

This is the default function.

Average

Adds all existing values and then divides by the count of existing values.

Calculated

Specifies that all the terms within the expression for a data item are aggregated according to their own rollup rules, and then the results of those aggregations are computed within the overall expression.

For example, a list contains the data item **Quantity** with the **Aggregation** property set to **Total**. You add a query calculation named **Quantity Calculated**. You define its expression as **Quantity + 100** and you set its **Aggregation** property to **Calculated**. When you run the report, the values for **Quantity Calculated** are computed by first adding all values for **Quantity** and then adding 100 to each value.

In crosstabs, this function overrides any solve orders that are specified.

Count

Counts all existing values.

If the underlying data source is OLAP, Count behaves as follows if it is specified in the **Rollup Aggregate Function** property for a data item.

Object	Behavior
Level	<code>Count distinct</code> is used. A warning appears when you validate the report.
Member set	<code>Count distinct</code> is used. A warning appears when you validate the report.
Attribute	Not supported. An error is returned when you run the report.
Measure	Supported.

If the underlying data source is dimensionally modeled relational (DMR), Count behaves as follows if it is specified in the **Aggregate Function** or **Rollup Aggregate Function** property for a data item.

Object	Behavior
Level	<code>Count distinct</code> is used. A warning appears when you validate the report.
Member set	<code>Count distinct</code> is used. A warning appears when you validate the report.
Attribute	Supported.
Measure	Supported.

If you add the `count` summary to a non-fact column in a list and then group the column, the column will not be grouped when you run the report. To resolve this issue, group the column first before adding the `count` summary.

Count Distinct

Returns the total number of unique non-null records.

If the underlying data source is OLAP, `count distinct` behaves as follows if it is specified in the **Rollup Aggregate Function** property for a data item.

Object	Behavior
Level	Supported.
Member set	Supported.

Object	Behavior
Attribute	Not supported. An error is returned when you run the report.
Measure	Not supported. An error is returned when you run the report.

If the underlying data source is dimensionally modeled relational (DMR), `count distinct` is supported for levels, member sets, attributes, and measures when it is specified in the **Aggregate Function** or **Rollup Aggregate Function** property for a data item.

Custom

Summarizes data based on an expression that you define.

Maximum

Selects the largest existing value.

Median

Returns the median value of the selected data item.

Minimum

Selects the smallest existing value.

None

Does not aggregate values.

If the underlying data source is relational, the data item is grouped when the query property **Auto Group & Summarize** is set to **Yes**.

Not Applicable

This function is available only for the **Aggregate Function** property. It specifies that the aggregate attribute is ignored. For example, the calculation will be applied after the data is aggregated.

This function differs from the **Calculated** function, which applies aggregation to the rows and then performs the calculation.

For example, for the `average ([Quantity]) + 5` expression, when the **Aggregate Function** property is set to **Not Applicable**, five is added to the average of the single Quantity value. When the function is set to **Calculated**, five is added to the average of a set of numbers.

This setting should not be applied to a simple model reference.

This setting is relevant only for reports that are upgraded from IBM Cognos ReportNet® 1.x.

Standard Deviation

Returns the standard deviation of the selected data item.

From a mathematical perspective, this function is not useful for small numbers of items and is not supported if the query property **Auto Group & Summarize** is set to **Yes**.

Summarize

Aggregates data based on model or data type information. This function can be thought of as a calculate and then aggregate rule. Data is aggregated using the following rules:

If the underlying data source type is relational and the data item or calculation is a reference to a single fact query item in the model that has the **Regular Aggregate** property set to **Sum**, **Maximum**, **Minimum**, **Average**, or **Count**, aggregate data using this function. Otherwise, data is aggregated according to the data type of the data item as follows:

- **Total** is applied for numeric and interval values.
- **Maximum** is applied for date, time, and date-time values.
- **Count** is applied for everything else.

The underlying data source type can be dimensional and the data item or calculation a reference to a single item in the model. In this case, if the **Regular Aggregate** property of the model item is not **unknown**, the summary function that corresponds to the **Regular Aggregate** property is applied. If the **Regular Aggregate** property is **unknown**, set the function to **Calculated**.

For example, a list contains the data item **Quantity** with the **Aggregation** property set to **Average**. You add a query calculation named **Quantity Summarize**. You define its expression as **Quantity + 100** and you set its **Aggregation** property to **Summarize**. When you run the report, the values for **Quantity Summarize** are computed by first adding 100 to each value for **Quantity** and then calculating the total. The total is calculated because **Quantity** contains numeric values and the expression for **Quantity Summarize** is not a simple data item reference. If **Quantity Summarize** is defined as **Quantity**, the function **Average** is applied to each value.

Total

Adds all existing values.

Variance

Returns the variance of the selected data item.

Note: From a mathematical perspective, this function is not useful for small numbers of items and is not supported if the query property **Auto Group & Summarize** is set to **Yes**.

Mapping Summary Functions From Data Sources to Framework Manager and Report Studio

The summary functions available in IBM Cognos Framework Manager and Report Studio reflect summary functions supported by relational and dimensional data sources. The following table shows how summary functions in data sources are mapped to Framework Manager and Report Studio.

Data source	Framework Manager	Report Studio
None specified, or none	Unsupported	None
average (avg)	Average	Average

Data source	Framework Manager	Report Studio
count	Count	Count
count distinct	Count Distinct	Count distinct
maximum (max)	Maximum	Maximum
minimum (min)	Minimum	Minimum
median	Median	Median
standard deviation (stddev)	Standard Deviation	Standard Deviation
sum	Sum	Total
variance (var)	Variance	Variance
count non zero	Count Non-Zero	Automatic
average non zero	unknown	Automatic
external	unknown	Automatic
any	unknown	Automatic
calculated	unknown	Automatic
unknown	unknown	Automatic
first_period	unknown	Automatic
last_period	unknown	Automatic

The following summary functions exist only in either Framework Manager or Report Studio. There is no corresponding summary function in data sources.

Framework Manager	Report Studio
Automatic	Automatic
Calculated	Calculated
No corresponding summary function	Summarize
No corresponding summary function	Not Applicable

Focusing Dimensional Data

In IBM® Cognos® Report Studio, there are three approaches to focusing dimensional data in a crosstab:

- drag data items to the edges of your crosstab
- drag data items to the context filter area
- create set expressions on the edges of your crosstab that return members or sets of members, such as the `filter` function

Note: Summary and detail filters are recommended only for focusing data in relational reporting.

The approach you choose should depend on the capabilities of the data source, the performance requirements, the persistence of the dataset, and how complex you want the report to be.

If you filter values that are floating-point numbers, you may encounter imprecise filtering results due to rounding or data truncation. Floating-point numbers may appear differently in the report than how they are stored in the data source. To ensure accurate results, your filter criteria should account for any rounding issues.

Consider the limitations when filtering dimensional data and when filtering data from SAP BW data sources.

You can also use prompts, or prompt controls, to allow users to provide input for the filter.

Create a Context Filter

When working with dimensional data, you can use context filters, or slicer filters, to quickly focus your report on a particular view of the data.

For example, the following crosstab contains product lines in the rows, years in the columns, and revenue as the measure. We want to filter the values to show us the revenue for only Web orders from Asia Pacific. To change the context, you drag Asia Pacific and Web from the source tree to the **Context filter** section of the overview area. The crosstab then shows the revenue for only Asia Pacific and Web. Changing context changes only the values that appear. It does not limit or change the items in the rows or columns.

Asia Pacific				
Web				
Revenue	2004	2005	2006	2007
Camping Equipment	36,415,479.91	77,222,359.67	121,312,260.99	83,023,522.78
Mountaineering Equipment		18,115,550.83	37,258,799.5	31,810,456.11
Personal Accessories	67,562,845.5	94,998,084.39	136,872,087.64	106,879,947.27
Outdoor Protection	4,135,482.66	4,388,590.73	2,241,092.7	1,107,629.48
Golf Equipment	8,586,746.5	27,061,602.93	49,822,888.09	39,248,971.12

The members that are used as the context filter appear in the report header when you run the report.

Any summary values in the report are recomputed to reflect the results returned by the context filter.

You can create multiple context filters to filter across two or more different hierarchies.

Tip: The Budget vs. Actual sample report in the Sales and Marketing (cube) package includes a context filter. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Note: There are limitations when using context filters:

- Calculated measures do not get suppressed by the filter. To avoid this, drag the desired members from the **Insertable Objects** pane directly onto the row or column edge instead of using the members in the context filter.
- Suppression can be slow on large cubes, because all of the data is retrieved and then filtered.
If the desired data in the context filter is from a hierarchy that appears on the crosstab edge, drag the desired members from the **Insertable Objects** pane directly onto the crosstab edge.
If the desired data is from a hierarchy that does not appear on the crosstab edge, use a context filter in conjunction with other means to limit the data, such as a top or bottom filter.

To use a calculation or expression to define a context filter, create a slicer member set using the query explorer. A slicer member set is an expression that returns a set of members from the same dimension. For example, you can define a slicer member set that filters for the top five products with the greatest revenue.

Context filters are different from other filters. When you filter data, members that do not meet the filter criteria are removed from the report. A context filter does not remove members from a report. Instead, their values are filtered or and you see blank cells.

You can also use context filters as prompts when the report is run.

When creating context filters,

- use only members from hierarchies that are not already projected on an edge of the crosstab
- use only one member per hierarchy

Steps to Create a Simple Context Filter

- If you do not see the **Context filter** area, show it using the **View** menu (**Panes, Context Area**).
- In the **Insertable Objects** pane, from the **Source** tab , select or search for one or more items on which to filter.
- Drag the item on which to filter into the **Context filter** section of the overview area.
- To change context, select a new item from the **Context filter** box.

Steps to Create an Advanced Slicer Filter

- Pause the pointer over the query explorer button  and click the query to which to add a slicer.
- On the **Toolbox** tab , drag **Slicer Member Set** to the **Slicer** pane.
- Drag the members to the **Expression Definition** box.

You must join the members in the expression by using the `set` function. For example, the following expression contains the Fax and Telephone members from the Order Method dimension:

```
set ([Fax],[Telephone])
```

4. To add members from another dimension, repeat step 2 to create a separate slicer member set.

Define a Prompt Using a Context Filter

If context filters (or slicers) are defined, you can use these filters to create prompts in the report. Prompts provide questions that help users to customize the information in a report to suit their own needs. For example, you create a prompt so that users can select a region. Only data for the specified region is retrieved and shown in the report.

Before you define a prompt, you must create context filters.

Steps

1. In the **Context filter** section of the overview area, click the down arrow next to the filter to use as a prompt.
2. Click **Prompt** and do one of the following:
 - To remove the prompt, click **No Prompt**.
 - To allow users who run the report to select any member from the hierarchy, click **Prompt on Hierarchy**.
 - To allow users who run the report to select from a list of members in the current level, click **Prompt on Level**.

Limitations When Filtering Dimensional Data Sources

Avoid combining context filters (slicers) with dimensional constructs that involve members from hierarchies that are used elsewhere in the report. This combination gives results that are often not what you might expect and that may change in a future release.

Dimensional data sources provide implicit rollup at all levels of each dimensional hierarchy. Context filters and slicers with more than one member invalidate any pre-computed rollup of members at hierarchy levels above the level at which the filter applies.

The following types of filtering are safe for dimensional reporting:

- Context filters (slicers) with one or more explicit, non-calculated members per hierarchy, if those hierarchies are not used elsewhere in the report.
- Expressions that use the `filter` function (not in a slicer), comparing measure, tuple, or attribute values.

Nesting Incomplete Sets

When you nest sets (especially sets of explicit members, and sets that result from the functions such as `filter` and `intersect`), there may be members at one nesting level that have no corresponding

members that can appear nested below them. In such cases, filtering results may not be what you expect: the higher-level members may or may not appear in the result.

While this is most commonly seen within a single hierarchy, it can also happen across different hierarchies and dimensions.

To avoid this problem, ensure that the lower-level sets are constructed to ensure that they are never empty for any of the members in the higher-level set. For example, you could filter only the top level set and nest only the complete set of descendants at the desired levels below.

Error Characters (--) When Filtering

When you use slicers or context filters with a calculated fact or measure, you may see two dashes (--) as values for some or all of the cells in the report that represent aggregates that are calculated in the database (aggregation is set to **Automatic**). This means that the data source is unable to compute these values.

You can avoid this error by using an explicit rollup rule or aggregation function such as **Total**. However, do this only if you are familiar with the data and absolutely certain that this is the appropriate answer for that report.

Time Hierarchies

Error characters also appear in summaries if your report includes a time hierarchy and you use two or more members from a different time hierarchy as a context filter. To avoid the error, use only one member as the context filter.

Filters in Queries and Sub-queries

For dimensional data sources, filters in queries and sub-queries are considered equivalent. The same applies to slicers.

Limitations When Specifying the Scope of Summary Filters with Dimensional Data Sources

When you specify the scope for a summary filter, consider the following limitations. The scope of a summary filter

- must refer to a grouped data item for list reports. Otherwise, you encounter the following error:

OP-ERR-0212 The summary filter scope (Scope1) found in the query (Query1) is not valid. It must refer to a grouped query item.

- can refer to multiple data items, but it can refer to only one data item per edge of the crosstab. Otherwise, you encounter the following error:

OP-ERR-0213 The summary filter scope (Scope1) found in the query (Query1) is not valid. Referencing more than one query item from the same edge is not supported.

- must refer to a level and cannot refer to a measure. Otherwise, you encounter the following error:

OP-ERR-0209 The summary filter scope (Scope1) found in the query (Query1) is not valid. It must refer to a level.

- must refer to a data item that is projected on an edge of the crosstab. If the scope of a summary filter refers to a data item that was deleted from the report layout, you may encounter unexpected results. Otherwise, you encounter the following error:

OP-ERR-0213 The summary filter scope (Scope1) found in the query (Query1) is not valid. Referencing more than one query item from the same edge is not supported.

- must refer to unique data items. For example, the summary filter cannot refer to the same data item more than once.

Limitations When Filtering Data Using an SAP BW Data Source

If you are working with an SAP BW data source, you must consider additional exceptions when applying filters.

If you apply a filter and a sort to an item that corresponds to the leaf-level of a recursive hierarchy, siblings may appear in the report even though the aggregated values are correct. Siblings are characteristic values with the same parent as the filtered member.

Each level in an SAP BW hierarchy has an item with the same name as the level and a role of `_businessKey`. Such items are known as level identifiers. The level identifier must be an exact value for the operators `=`, `<`, and `>` to work. For example, for the filter [Office] > 'Chicago' to work, the value 'Chicago' must exist in the data source. If you do not know the exact values, you can apply the filter to one of the attribute items associated with the level, such as [OfficeLongName] > 'C'. Filters on non-identifiers are possible, but they are slower because SAP BW data sources are optimized for queries based on level identifiers.

When filtering time-related data, only the level identifier items of the time-related characteristics in SAP BW, such as 0CALDAY and 0CALMONTH, should be used for performing anything other than equality filters. All other (attribute) items in these hierarchies are formatted string representations of the characteristic values with which they are associated. These formatted values sort alphanumerically and not chronologically.

Using Prompt Expressions in Filters

If you create a filter expression using report item attributes, such as the following, no data is returned.

`[report item attribute] = ?prompt?`

To resolve the problem, do one of the following:

- Fully qualify the item by using items from the source tree instead of data items from the report.
- Use prompt controls in the report instead of hand coded prompts.

Sorting Dimensional Data

You can sort items to view them in your preferred order. By default, IBM® Cognos® Report Studio retrieves items in the order defined in the data source. OLAP data sources always have a defined order. Relational and dimensionally-modeled relational data sources may not always have a defined order. The data modeler defines the sorting options in the model. For more information about data modeling, see the IBM Cognos Framework Manager *User Guide*.

Tip: The Sales Growth Year Over Year sample report in the GO Data Warehouse (analysis) package includes sorting. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Sorting Sets by Labels

You can sort sets of members based on their label (or caption). By default, when you select **Sort by Label, Ascending** or **Descending**, the members within the set are sorted only by their labels. You can also perform advanced sorting to sort sets using an intersection (tuple) or a member property.

For example, a set contains regions and you sort the regions within the set in descending alphabetical order. Only extended data items that represent sets support set sorting.

Before you can perform set sorting, you must create sets for the members in your report.

You cannot sort data items from different dimensions that are intermixed. For example, if you have years and products on the same edge and at the same level in your crosstab, you can sort the years, and you can sort the product lines, but you cannot sort them as combined set because years and products come from different dimensions.

Sorting Sets in the Opposite Axis by Value

You can sort members in the sets on the opposite axis based on the value of a member or measure that you select.

For example, a crosstab contains years inserted as individual members in the columns, product lines inserted as a member set in the rows, and revenue as the measure. You select the 2004 column and then select **Sort by Value, Descending**. The values in the 2004 column are sorted.

Revenue	2004	2005
Personal Accessories	391,647,093.61	456,323,355.9
Camping Equipment	332,986,338.06	402,757,573.17
Golf Equipment	153,553,850.98	168,006,427.07
Outdoor Protection	36,165,521.07	25,008,574.08
Mountaineering Equipment		107,099,659.94

Before you can perform this sorting, you must create sets for the members in your report.

In nested crosstabs, items are sorted based on the values of the innermost nested row or column of the opposite axis. For example, a crosstab contains years in the columns and retailers nested within product line in the rows. Select 2004, sort in descending order, and you see retailers listed from the largest value to the smallest value for each product line.

Limitations When Sorting SAP BW Data

If you use a SAP BW data source, each level in a hierarchy has an item with the same name as the level and a role of `_businessKey`. Such items are known as level identifiers. You should use only the level identifier of all the time-related characteristics, such as `0CALDAY` and `0CALMONTH`, for sorting. All other (attribute) items in these hierarchies are formatted string representations of the characteristic values with which they are associated. These formatted values sort alphanumerically and not chronologically.

Tip: By level identifier, we are referring to a query item with the same name as the level that has the role of _businessKey. Each level in a SAP BW hierarchy has a level identifier.

Steps to Sort Sets by Label

1. In a crosstab, select a set to sort.
2. Click the sort button  and under Sort by Label, click Ascending or Descending.

Tip: To remove a sort order, click Don't Sort.

Steps to Sort Sets in the Opposite Axis by Value

1. In a crosstab, select a member or measure to sort.

For example, to sort a set in the row axis, select a member or measure in the column axis.

2. Click the sort button  and under Sort by Value, click Ascending or Descending.

Tip: To remove a sort order, click Don't Sort.

Perform Advanced Set Sorting

If you use dimensional data, you can sort using the property of a member. For example, your report includes employee names and you want to sort the employees using the gender property. You can also sort using an intersection of members, also known as a tuple. For example, you want to sort the employees using the value of sick days taken for the year 2006.

Steps to Sort Sets of Dimensional Data

1. Select a set.
2. Click the sort button  and under Other Sort Options, click Edit Set Sorting.
3. Specify the sorting options.
4. If you are sorting members from different levels and want to preserve the hierarchy, select the **Hierarchized sorting** check box.
5. To sort items using a member property, under Sort by, double-click **Property**, navigate the tree, and select the member property to use.
6. To sort items using an intersection of members, or a tuple, under Sort by, click **Intersection (tuple)** and click the ellipsis (...) button. Then, from the Available members and measures box, select the items to use and click the right arrow to move them to the Intersection members and measures box. You can also select items from the Calculated Members and Measures tab.

Working with Dimensional Queries

Queries specify what data appears in the report. In IBM® Cognos® Report Studio, you create and modify queries using Query Explorer. Query Explorer provides an alternative way to modify existing reports or to author new reports. You can use Query Explorer to perform complex tasks

and other tasks that are more difficult to do in the report layout. For example, use Query Explorer to

- improve performance by changing the order in which items are queried from the database, or by changing query properties to allow the report server to execute queries concurrently where possible

Note: By default, queries run sequentially. Your administrator must enable the concurrent query execution feature. For more information, see the *IBM Cognos Business Intelligence Administration and Security Guide*.

- view or add filters and parameters and modify their properties
- view or add dimensions, levels, and facts
- incorporate SQL statements that come from other reports or reports that you write
- create complex queries using set operations and joins

Specify a List of Data Items for an Object

Specify the list of data items for an object when you must reference a data item that is in a query but is not in the layout. For example, you want to add a layout calculation to a list that uses a data item that is in the query definition. If the data item does not appear in the list, you must reference it in order to make the layout calculation work.

You must also specify the list of data items if you apply conditional formatting that uses a data item that is not in the query.

Steps

1. Click a layout object.

Tip: For a list of layout objects for which you can specify a list of properties, see the **Properties** property in "[Report Studio Object and Property Reference](#)" (p. 601).

2. In the **Properties** pane, double-click the **Properties** property and select data items.

Relate a Query to a Layout

Queries and layouts work together. After you decide the type of data that you need, you must create a layout in which to show the results. Each column of data must be both selected for the query and shown in a layout unless there are some columns that you do not want to show. The query and layout portions of a report must be linked to produce a valid report.

Report Studio automatically links query and layout. For example, when you use Report Studio and the list report layout, query and layout are automatically linked.

Steps to Manually Link a Query to a Data Container

1. Select a data container.
2. In the **Properties** pane, set the **Query** property to a query.

3. In the **Insertable Objects** pane, on the **Data Items** tab , drag data items from the query to the data container.

Using Dimensional Data Sources with Queries

When you are working with SAP BW data sources, you can use only a single hierarchy in a query.

Creating queries using a mix of OLAP and relational data is not supported. If you create queries using a database for which you do not know the type, consult your database administrator or modeler.

When performing multi-cube queries using dimensional data sources, the following restrictions apply:

- Only basic operators (+, *, /,-) are available for cross-cube calculations.
- Inner joins are not supported.
- All joins for multi-cube queries are outer joins.
- Viewing MDX using the **Generated SQL/MDX** query property may not show you the actual MDX that is executed. This is because when a multi-cube query is executed, a number of smaller queries are actually executed.

Running Large SAP BW Queries

When working with a SAP BW data source, if your report includes a data set greater than one million cells, you may encounter the following error:

XQE-CON-0007 XQE error encountered: XSB-ERR-0022 Execution of MDX failed: XSB-ERR-0038 SAP Error executing BAPI BAPI_MDDATASET_SELECT_DATA: com.cognos.xqe.data.providers.olap.sbwodp.bapi.JBapiSAPException: RFC Error: MDX result contains too many cells (more than 1 million).

To avoid this error, set the Processing property for the query to Limited Local. This allows the report to process locally.

Connecting Queries Between Different Data Sources

IBM Cognos BI supports the following:

- RDBMS to RDBMS joins
- set operations of any two queries
- master detail relationships between any two queries
- drill from any query to any other query

You cannot create the following types of joins:

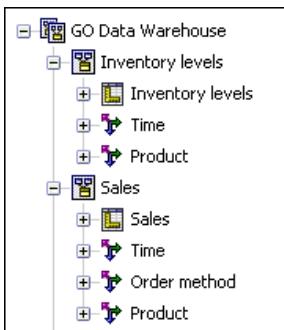
- cube-to-cube (homogeneous)
- cube-to-cube (heterogeneous)
- cube-to-RDBMS

- cube-to-SAP BW
- SAP-BW-to-RDBMS

Multiple-fact Queries

You must understand the result of a query on more than one fact table to achieve the results that you want. The result of a multiple-fact query varies depending on whether you are working with conformed or non-conformed dimensions, on the level of granularity, and on the additive nature of the data.

The following data source has characteristics that affect the results if you use a multiple-fact query with Inventory levels and Sales. Granularity for time differs in that inventory levels are recorded monthly and sales are recorded daily. Also, Sales includes a non-conformed dimension, Order method.



The following examples will help you interpret the results of a multiple-fact query and understand the options for changing a query to obtain the results that you want.

Conformed Dimensions

Individual queries on Inventory levels and Sales by Quarter and Product yield the following results.

Quarter	Product	Opening inventory	Closing inventory	Quantity
200501	Husky Rope 50	720	727	608
	Aloe Relief	1,128	1,236	1,256
	Course Pro Umbrella	1,077	1,217	902
200501		975	1,060	2,766
200502	Husky Rope 50	667	632	1,246
	Aloe Relief	988	874	1,710
	Course Pro Umbrella	969	1,025	2,158
200502		874	843	5,114

A query on multiple facts and conformed dimensions respects the cardinality between each fact table and its dimensions and returns all the rows from each fact table. The fact tables are matched on their common keys, Product and Time.

Product and Time apply to both Inventory levels and Sales. However, inventory levels are recorded monthly and sales are recorded daily. In this example, results are automatically aggregated to the lowest common level of granularity. Quantity, which comes from Sales, is rolled up to months.

Nulls are often returned for this type of query because a combination of dimensional elements in one fact table may not exist in the other. For example, if Husky Rope 50 was available in inventory

in 200501, but there were no sales of this product in the same time period, Quantity would show a null in the Husky Rope 50 row.

IBM Cognos BI does not support conformed dimensions generated by Framework Manager for SAP BW data sources.

Non-conformed Dimensions

If a non-conformed dimension is added to the query, the nature of the results returned by the query is changed.

Quarter	Product	Order method	Opening inventory	Closing inventory	Quantity
200501	Husky Rope 50	Telephone	720	727	254
	Husky Rope 50	Mail	720	727	70
	Husky Rope 50	E-mail	720	727	50
	Husky Rope 50	Web	720	727	98
	Husky Rope 50	Sales visit	720	727	136
	Aloe Relief	Telephone	1,128	1,236	400
	Aloe Relief	E-mail	1,128	1,236	138
	Aloe Relief	Web	1,128	1,236	378
	Aloe Relief	Sales visit	1,128	1,236	340
	Course Pro Umbrella	Fax	1,077	1,217	72
	Course Pro Umbrella	Telephone	1,077	1,217	140
	Course Pro Umbrella	E-mail	1,077	1,217	202
	Course Pro Umbrella	Web	1,077	1,217	80
	Course Pro Umbrella	Sales visit	1,077	1,217	224
	Course Pro Umbrella	Special	1,077	1,217	184
200501			975	1,060	2,766

Order Method exists only in Sales. Therefore, it is no longer possible to aggregate records to a lowest common level of granularity because one side of the query has dimensionality that is not common to the other side of the query. Opening inventory and Closing inventory results are repeated because it is no longer possible to relate a single value from these columns to one value from Quantity.

Grouping on the Quarter key demonstrates that the result in this example is based on the same data set as the query on conformed dimensions. Summary values are the same. For example, the total quantity for 200501 is 2,766 in both examples.

Filters on Non-conformed Dimensions

By default, to improve performance, filters are applied at the database level. The default behavior can result in unexpected nulls when you add a filter to a multiple-fact query. If you create a filter on the Order method dimension to show only the fax order method and apply the filter at the data source, the report includes nulls.

Quarter	Product	Order method	Opening inventory	Closing inventory	Quantity
200501	Husky Rope 50		720	727	
	Aloe Relief		1,128	1,236	
	Course Pro Umbrella	Fax	1,077	1,217	72
200501			975	1,060	72
200502	Husky Rope 50	Fax	667	632	82
	Aloe Relief	Fax	988	874	94
	Course Pro Umbrella	Fax	969	1,025	314
200502			874	843	490
200601	Husky Rope 50	Fax	722	741	132
	Aloe Relief	Fax	1,175	1,233	40
	Course Pro Umbrella	Fax	1,463	1,764	130
200601			1,120	1,246	302
200602	Husky Rope 50	Fax	725	714	86
	Aloe Relief	Fax	1,114	1,024	36
	Course Pro Umbrella		1,520	1,388	
200602			1,120	1,042	122
Summary			1,022	1,048	986

The filter is applied only to one subject area, Sales. Because Order method does not exist in Inventory levels, all products still appear in the report. For example, Course Pro Umbrella was in inventory in 200602. Because there were no sales using the fax order method for this product in 200602, Order method and Quantity are null.

To remove the nulls, change the filter so that it is applied to the result set instead of the data source. By applying the filter to the result set, you get the same results that you would get if you were working with conformed dimensions. Only the products that were ordered using the fax order method appear in the report.

Quarter	Product	Order method	Opening inventory	Closing inventory	Quantity
200501	Course Pro Umbrella	Fax	1,077	1,217	72
			1,077	1,217	72
	Husky Rope 50	Fax	667	632	82
200502	Aloe Relief	Fax	988	874	94
	Course Pro Umbrella	Fax	969	1,025	314
			874	843	490
200601	Husky Rope 50	Fax	722	741	132
	Aloe Relief	Fax	1,175	1,233	40
	Course Pro Umbrella	Fax	1,463	1,764	130
200601			1,120	1,246	302
200602	Husky Rope 50	Fax	725	714	86
	Aloe Relief	Fax	1,114	1,024	36
			919	869	122
Summary			989	1,025	986

The summary for quantity is 986 using either filter method, which shows that the results are based on the same data set.

In Report Studio, there are two types of filters. A detail filter is applied to the data source. A summary filter is applied to the result set. For more information about how to apply detail and summary filters, see ["Create a Detail or Summary Filter" \(p. 324\)](#).

Add a Query to a Dimensional Report

You can create multiple queries in Query Explorer to suit your particular needs. For example, you can create a separate query for each data container in a report to show different data.

Tip: The Briefing Book sample report in the GO Sales (analysis) package and the Top 10 Retailers for 2005 sample report in the GO Data Warehouse (analysis) package include multiple queries. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Steps

1. Pause the pointer over the query explorer button  and click **Queries**.
2. In the **Insertable Objects** pane, drag one of the following objects to the work area.

Object	Description
Query	Adds a query.
SQL	Adds SQL commands.
MDX	Adds MDX commands.

3. In the **Properties** pane, set the object properties.
 4. Double-click a query.
 5. In the **Insertable Objects** pane, on the **Source** tab , drag data items to the **Data Items** pane.
- Tip:** You can add data items to the query that you do not want to appear in the layout. For example, to filter on Product line code and show Product line in the layout, you must add both data items to the query.
6. To create a new data item, in the **Insertable Objects** pane, on the **Toolbox** tab , drag **Data Item** to the **Data Item** pane.
 7. To add a filter, in the **Insertable Objects** pane, on the **Toolbox** tab, drag **Filter** to the **Detail Filters** or **Summary Filters** pane and define the filter expression.

Tip: You can also create a filter by dragging a data item from the **Source** tab to one of the filters panes and completing the filter expression. If you are working with a dimensional data source, you can quickly filter data by dragging a member to a filter pane instead of dragging the level to which the member belongs. For example, dragging the member 2006 from the Years level will filter data for the year 2006. This method is quicker than dragging the Years level and specifying data only for the year 2006 in the filter expression.

When adding queries to the report

- right-click the work area and click **Show Package Sources** to see the queries that use data items from a package

- right-click the work area and click **Expand References** to see the relationships that exist between queries in the report, which is useful when you are creating complex queries

Add Dimension Information to a Query

You can add dimension information to a query if any of the following apply:

- There is no dimension information available in the data source.
For example, the data source contains flat data that you want to model dimensionally.
- You want to override the dimension information in the data source.
- You want to extend or restrict dimension information in the data source.

If you do not add dimension information and the data source contains dimensions, then the dimension structure of the data source is used. If the data source contains no dimensions, IBM Cognos BI creates a default dimension structure.

Dimension information is not intended to define the presentation of information, but to help query planning. Dimension information can be considered as a form of query hint. In addition, users will not be able to drill down on dimensions that you add to a report.

Steps

1. Pause the pointer over the query explorer button  and click a query.
2. In the Properties pane, set the **Override Dimension Info** property to Yes.
The **Dimension Info** tab appears in the work area.
3. Click the **Dimension Info** tab.
4. To create a dimension from an existing data item, in the **Insertable Objects** pane, on the **Source** tab , drag the data item to the **Dimensions** pane.
Report Studio automatically generates the entire dimension.
5. To create a new dimension, in the **Insertable Objects** pane, on the **Toolbox** tab , drag **Dimension** to the **Dimensions** pane, and then build the new dimension by adding objects to the dimension:
 - To create a level, drag **Level**.
 - To create a level hierarchy, drag **Level Hierarchy**.
 - To create a member hierarchy, drag **Member Hierarchy**.
For information about these objects, see "[Add Dimensional Data to a Report](#)" (p. 347).
6. In the **Insertable Objects** pane, on the **Source** tab, drag data items to the objects you added in the previous step.
For example, if you created a level, define a key by dragging a data item to the **Keys** folder.

7. To create a fact, in the **Insertable Objects** pane, on the **Source** tab, drag a data item to the **Facts** pane.

Example - Create a Crosstab Report that Shows Empty Rows

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a crosstab report that shows the total revenue per year for each product line broken down by order method. By default, if no revenue was produced for a particular product line in a specific year, no row appears in the crosstab for that product line and year. You override the dimension information of the crosstab so that empty rows appear in the report.

Steps

1. Open Report Studio with the **GO Data Warehouse (query)** package.
2. In the **Welcome** dialog box, click **Create a new report or template**.
3. In the **New** dialog box, click **Crosstab** and click **OK**.
4. In the **Insertable Objects** pane, on the **Source** tab , drag the following data items from **Sales and Marketing (query)**, **Sales (query)** to the crosstab:
 - from the **Product** folder, drag **Product line** as rows
 - from the **Time dimension** folder, drag **Year** as nested rows
 - from the **Order method** folder, drag **Order method** as columns
 - from the **Sales fact** folder, drag **Revenue** as the measure
5. Right-click the crosstab and click **Go to Query**.
6. In the **Properties** pane, set the **Override Dimension Info** property to **Yes**.
The **Dimension Info** tab appears at the bottom of the work area.
7. Click the **Dimension Info** tab.
8. In the **Insertable Objects** pane, on the **Source** tab, drag the following items to the **Dimensions** pane:
 - **Product line**
 - **Year**

Product line and **Year** become separate dimensions in the query.
9. Pause the pointer over the page explorer button  and click **Page1**.
10. Click **Year**.
11. In the **Properties** pane, double-click the **Sorting** property.
12. In the **Data Items** box, drag **Year** to the **Sort List** box.
13. Run the report.

All order years appear for all product lines, even if no revenue was produced.

Revenue		Web	Telephone	Fax	Mail	Special	E-mail	Sales visit
Mountaineering Equipment	2004							
	2005	65,855,489.46	10,626,292.36	6,129,791.95	2,497,183.28	2,838,822.56	2,517,063.13	16,635,017.2
	2006	132,736,443.67	6,586,124.67	3,538,047.25	1,034,475.38	517,010.05	1,829,100.61	14,798,621.63
	2007	117,010,256.92	5,698,410.37	2,180,530.88		318,175.5	3,130,288.22	13,182,987.81
Personal Accessories	2004	284,622,826.47	45,940,692.79	11,313,266.47	2,952,291.35	3,497,802.5	22,819,708.21	20,500,505.82
	2005	411,577,877.16	18,428,095.15	3,613,228.75	1,719,745.82	1,390,381.2	10,253,053.91	9,340,973.91
	2006	568,668,077.83	5,979,547.46	2,149,810.49	1,346,338.05	146,195.56	5,568,561.15	10,150,877.88
	2007	427,367,391.98	3,173,298.96	886,679.75	400,981.81	152,249.24	4,009,763.27	7,703,084.84
Golf Equipment	2004	49,583,401.41	44,244,120.93	6,255,930.08	8,396,407.78	3,705,129.72	29,322,537.92	12,046,323.14
	2005	116,939,694.38	27,340,352.57	3,539,563.59	2,355,735	1,259,633.25	8,851,232.61	7,720,215.67
	2006	203,385,896.61	6,411,233.64	2,408,222.14	1,941,144.7		5,401,733.78	10,562,039.68
	2007	157,698,057.23	734,405.51	3,037,587.46			4,358,428.85	8,912,340.24
Outdoor Protection	2004	13,735,716.85	8,141,169.76	1,435,512.2	1,402,530.23	858,775.41	4,135,915.35	6,455,901.27
	2005	16,479,270.8	3,203,287.7	385,329.2	594,355.43	267,356.86	1,406,531.47	2,672,442.62
	2006	8,570,078.91	507,485.63	123,028.48	101,506.05	10,798.96	289,343.01	746,934.8
	2007	4,166,745.33	76,371.43	22,614.84			50,688.04	154,605.62
Camping Equipment	2004	125,829,519.92	80,467,596.88	9,634,763.39	10,015,621.15	5,844,211.12	39,124,634.73	62,069,990.87
	2005	270,463,415.88	47,562,256.31	6,228,274.27	8,846,759.96	5,012,986.47	21,291,005.31	43,352,874.97
	2006	426,353,675.75	17,715,451.4	5,226,451.57	2,482,266.26	332,095.44	10,612,304.02	37,660,178.39
	2007	311,192,071.84	8,149,587.54	1,964,909.25	3,996.72	1,199,696.41	4,871,150.57	25,528,917.64

Create a Master Detail Relationship

Create a master detail relationship to deliver information that would otherwise require two or more reports. For example, you can combine a list with a chart. The list can contain product lines and the chart can show details for each product line.

Master detail relationships must appear in nested frames to produce the correct results. You can create a master detail relationship in two ways:

- Use a parent frame for the master query and a nested frame for the detail query.
- Associate a report page with the master query and use a data container, such as a list or crosstab, for the detail query.

You cannot display parent data in the child frame or child data in the parent frame. Also, you cannot perform calculations across master detail queries.

You can use a master detail relationship to show data from separate data sources in a single report. However, the data sources must be contained in the same package.

If you are working with an SAP BW data source, you cannot use a data item from the master query that contains non-ASCII values.

To create a master detail relationship using queries that reference two different dimensional data sources, you must first [link the members](#) by aliasing the levels that contain the members.

Steps

1. To use a parent frame for the master query and a nested frame for the detail query:

- In the Insertable Objects pane, on the Toolbox tab , drag a List, Repeater Table, or Repeater to the report.

- Add a second data container to the object you inserted.
You can insert a list, crosstab, chart, repeater table, or repeater into a list. You can add a list to a repeater table or repeater.
 - Add data items to both data containers.
2. To associate a report page with the master query and use a data container for the detail query:
- Click anywhere in the report page.
 - In the **Properties** pane, click the select ancestor button  and click **Page**.
 - Set the **Query** property.
 - In the **Insertable Objects** pane, on the **Toolbox** tab, drag a data container to the report.
3. To link a data item in the master query to a parameter in the detail query instead of to another data item, create the parameter in the detail query.
Use parameters to filter values at a lower level in the detail query.
- Pause the pointer over the query explorer button  and click the detail query.
 - In the **Insertable Objects** pane, on the **Toolbox** tab, drag the **Filter** object to the **Detail Filters** box.
 - In the **Expression Definition** box, create the parameter.
4. Pause the pointer over the page explorer button  and click the report page.
5. Click anywhere in the report page.
6. In the **Properties** pane, click the select ancestor button and click **Page**.
7. Click the data container containing the details.
8. From the **Data** menu, click **Master Detail Relationships**.
9. Click **New Link**.
10. In the **Master** box, click the data item that will provide the primary information.
11. To link the master query to a data item, in the **Detail** box, click the data item that will provide the detailed information.
12. To link the master query to a parameter, in the **Parameters** box, click the parameter that will provide the detailed information.
- Tip:** To delete a link, select the link and press the Delete key.
13. If your detail query object is a chart, you can disconnect a chart title from the master query.
- Click the chart title, and then, from the **Data** menu, click **Master Detail Relationships**.
 - Clear the **Use the master detail relationships from the chart** check box.

14. If your master detail relationship includes a chart as the detail and you are using automatic axis ranges, you can set the same axis range for all the detail charts:

- Select the axis.
- In the **Properties** pane, under **General**, double-click the **Axis Range** property.
- Under **Maximum** and **Minimum**, click **Automatic**.

The master detail report option is only available with automatic axis ranges.

- Under **Master detail reports**, select the **Same range for all instances of chart** check box.

For more information about axis ranges, see "[Change the Axis Scale of a Current Default Chart](#)" (p. 163).

Tip: To avoid seeing the same data item twice in the report, click the data item in the data container driven by the detail query and click the cut button. This removes the item from the report display but keeps it in the query.

Link Members from Two Data Sources

If you create a master detail relationship using queries that reference two different dimensional data sources, you may need to create a relationship between levels with the same member captions but different Member Unique Names (MUNs). For more information about MUNs, see "[Recommendation - Use Member Unique Name \(MUN\) Aliases](#)" (p. 412).

For example, you may want to link the Americas member in the Sales territory level in the Great Outdoors cube with Americas in the GO Subsidiary level in the Great Outdoors Finance cube. To create the relationship, you must alias the levels that contain the members to link.

Steps

1. In the parent frame or the report page, double-click the level that contains the member that will provide the primary information.

The **Data Item Expression** dialog box appears.

2. In the **Expression Definition** box, use the expression in the following function:

`caption (expression)`

For example, `caption ([great_outdoors_company].[Sales Territory].[Sales Territory].[Sales territory])`

3. Repeat steps 1 to 2 for the level in the data container that contains the details.

A caption alias is created for each level. You can now use the caption alias for each level to create a master detail relationship using member captions, which are the same, instead of MUNs, which are different.

You can now create the master detail relationship between the two levels.

Master Detail or Burst Reports with Charts or Crosstabs May Result in Denial of Service

When running a master detail or burst report that includes a chart or crosstab, disk space exhaustion may cause the report or other requests to fail. A large set of burst keys or master rows may produce

one or more charts per detail, resulting in many master detail executions. This may cause the temp folder to accumulate many gigabytes of temporary files containing data required for successful chart rendering.

To avoid this issue, we recommend that you test large master detail or burst reports that include charts or crosstabs to determine the potential peak disk requirements for the report.

Working with Queries in SQL or MDX

For each query in a report, you can work with the SQL or MDX that is executed when you run a report. You can

- [view the SQL or MDX for an entire report or a query](#)
- [build a report using your own SQL or MDX](#)
- [convert a report to use SQL](#)
- [edit the SQL or MDX](#)

Note: MDX is not available with DMR data sources.

View the SQL or MDX for an Entire Report or a Query

View the SQL or MDX to see what is passed to the database when you run a report.

Note: Only SQL is available when using DMR data sources because these data sources were relational.

Steps

1. To view the SQL or MDX for the entire report, from the Tools menu, click **Show Generated SQL/MDX**.

This option shows the SQL or MDX that will be executed in the data source. The SQL or MDX is organized by query and by query result. If a query is used in more than one data container, a query result is generated for each data container.

2. To view the SQL or MDX for a specific query, do the following:

- Pause the pointer over the query explorer button  and click the query.
- In the Properties pane, double-click the **Generated SQL/MDX** property.

The **Generated SQL/MDX** property shows the SQL or MDX that is executed when you view tabular data (from the Run menu, click **View Tabular Data**). Tabular data shows the data that is produced by the query in the form of a list. You can use this property to help you build advanced queries.

The SQL or MDX for the query appears in the **Generated SQL/MDX** dialog box. For SQL, you can choose to view native SQL, which is the SQL that is passed to the database when you execute the query, or Cognos SQL, which is a generic form of SQL that Report Studio uses. Cognos SQL is converted to native SQL before the query is executed.

Build a Report Using Your Own SQL or MDX

You can build a report by adding SQL or MDX from an external source, such as another report.

If you are working with MDX, you must be aware of the [MDX syntax](#) that Report Studio supports.

If you edit the SQL of a query, you must change the **Processing** property for that query to **Limited Local**.

Steps

1. From the **File** menu, click **New**.
2. Click **Blank**.
3. Pause the pointer over the query explorer button  and click **Queries**.
4. In the **Insertable Objects** pane, drag **Query** to the work area.
5. In the **Insertable Objects** pane, do one of the following:
 - To build an SQL query, drag **SQL** to the right of the query.
 - To build an MDX query, drag **MDX** to the right of the query.
- Tip:** You can drag **SQL** or **MDX** anywhere in the work area, and Report Studio will automatically create a query.
6. In the **Properties** pane, double-click the **Data Source** property and click a data source.
7. If required, set the **Catalog** property to the name of the catalog.
8. Double-click the **SQL** or **MDX** property and type the SQL or MDX.
9. Click **Validate** to check for errors.
10. If you are working with SQL, in the **Properties** pane, set the **Processing** property to **Limited Local**.
11. Double-click the query.
If the SQL or MDX is valid, the data items defined in the SQL or MDX appear in the **Data Items** pane.
12. Pause the pointer over the page explorer button  and click a report page.
13. In the **Insertable Objects** pane, on the **Toolbox** tab, drag an object to the work area.
For example, drag a list, crosstab, chart, or repeater.
14. Click the data container.
15. In the **Properties** pane, click the select ancestor button  and click the container you just created.
For example, if you created a list, click **List**.
16. Set the **Query** property to the query for the report type.

17. In the **Insertable Objects** pane, on the **Data Items** tab , drag the items from the query you chose in the previous step to the data container.

Convert a Query to SQL

You can convert a query to SQL to edit it. You may want to do this to improve performance or to use SQL features that are not supported directly by Report Studio.

Converting a query to SQL is an irreversible process.

Steps

1. Pause the pointer over the query explorer button  and click the query.
2. In the **Properties** pane, double-click the **Generated SQL** property.
3. Click **Convert**.
4. Click **Validate** to check for errors.
5. In the **Properties** pane, set the **Processing** property to **Limited Local**.

Edit the SQL or MDX

You can edit the SQL or MDX for a query that has been created as an SQL or MDX query or that has been converted to SQL.

Steps

1. Pause the pointer over the query explorer button  and click the SQL or MDX item under the query.
2. Double-click the SQL or MDX item.
3. Make changes in the text box.
If you are working with MDX, you must be aware of the [MDX syntax](#) that Report Studio supports.
4. Click **Validate** to check for errors.

First-Rows Optimization

The SQL produced by Report Studio depends on the report format you choose. For example, if you specify HTML format, first-rows optimization is requested. All-rows is requested if you specify PDF.

It is important for database administrators and programmers to remember that Report Studio does not always use first-rows optimization. If you assume first-rows optimization is always requested, this can cause the RDBMS optimizer to process the query differently than you expect.

Supported MDX Syntax

If you are working with MDX, you must be aware of the syntax that Report Studio supports. Report Studio supports the MDX grammar as specified in Microsoft® Data Access SDK version 2.8.

Report Studio supports the following MDX functions. This is a partial list and applies only when you are working with the following data sources:

- cubes
- SAP BW
- IBM Cognos TM1®
- DB2® OLAP
- IBM Cognos Consolidation
- IBM Cognos Contributor

For these data sources, IBM Cognos BI uses a variation of the Microsoft SQL Server Analysis Services syntax.

- Acos
- Aggregate
- Ancestor
- Ancestors
- Asin
- Atan
- Avg
- BottomCount
- BottomPercent
- BottomSum
- Children
- ClosingPeriod
- CoalesceEmpty
- Convert
- Cos
- Cosh
- Count
- Cousin
- Crossjoin
- CurrentMember
- DefaultMember
- Descendants

- Dimension
- Distinct
- Except
- Exp
- Filter
- FirstChild
- FirstSibling
- Generate
- Head
- Hierarchize
- Hierarchy
- IIf
- Intersect
- Is
- IsAncestor
- IsEmpty
- Item
- Lag
- LastChild
- LastPeriods
- LastSibling
- Lead
- Level
- Levels
- LinkMember
- Ln
- Log10
- Max
- Median
- Members
- Min

- Mtd
- Name
- NextMember
- OpeningPeriod
- Order
- Ordinal
- ParallelPeriod
- Parent
- PeriodsToDate
- PrevMember
- PreviousMember
- Properties
- Qtd
- Rank
- Siblings
- Sin
- Sinh
- Stddev
- Subset
- Sum
- Tail
- Tan
- Tanh
- TopCount
- TopPercent
- TopSum
- Union
- Value
- Var
- Wtd
- Ytd

Example - Create a Dynamic Report That Shows Period-to-date Data

When you are working with a dimensional data source, you can use dimensional functions to retrieve data for a specific period-to-date.

In this topic, you learn how to create a dynamic report that retrieves year-to-date revenue for each product line. The report also shows the percentage of the revenue generated in the month users select when they run the report.

It should take 15-20 minutes to complete this topic, and your report will look like this.

Revenue	April	YTD Revenue for April	Percent of YTD Revenue
Camping Equipment	24,764,220	24,764,220	100.00%
Personal Accessories	26,057,935.06	26,057,935.06	100.00%
Outdoor Protection	2,713,436.3	2,713,436.3	100.00%
Golf Equipment	11,742,767.4	11,742,767.4	100.00%
Mountaineering Equipment			

Steps

1. Create a crosstab report that uses the **Go Data Warehouse (analysis)** package.
2. Add the following data items to the report:
 - From **Product**, drag **Product line** to the **Rows** drop zone.
 - From **Sales fact**, drag **Revenue** to the **Measures** drop zone.

Tip: Use the **Source** tab in the **Insertable Objects** pane.
3. Create the following query calculation named **Selected Month** in the **Columns** drop zone:
`[Sales].[Time dimension].[Time dimension].[Month]->?Month?`
4. Create this query calculation named **Year to Date Set** next to the **Selected Month** column:
`periodsToDate([Sales].[Time dimension].[Time dimension].[Year],[Selected Month])`

Tip: Information about the `periodsToDate` function appears in the **Information** pane when you click the function in the **Functions** tab. The `periodsToDate` function is located in the **M-Q** folder in the **Common Functions** folder.
5. Click the **Year to Date Set** column and click the cut button .
6. Create the following query calculation named **YTD Revenue** next to the **Selected Month** column:
`total (currentMeasure within set [Year to Date Set])`
7. Create the following query calculation named **Percent of YTD Revenue** next to the **YTD Revenue** column:
`[Selected Month]/[YTD Revenue]`
8. Click the unlock button  to unlock the report.
9. Add this text item to the left of the text in the **YTD Revenue** column, and add a blank space after the text:

YTD Revenue for

10. Set the font style of the text to italic.
11. In the YTD Revenue column, click <#YTD Revenue#> and change the Source Type property to Report Expression.
12. Double-click the Report Expression property and drag the Month parameter to the Expression Definition box.
Tip: You can find the Month parameter in the Parameters tab.
13. Click the lock/unlock button to lock the report.
14. Right-click the Percent of YTD Revenue column title and click Fact Cells for "Percent of YTD Revenue".
15. In the Properties pane, double-click the Data Format property.
16. Under Format type, click Percent.
17. Set the No of Decimals property to 2.
18. Run the report to view what it will look like for your users.

Users are prompted to select a month. When they click OK, the report shows year-to-date revenue for each product line and the percentage of revenue generated for the selected month.

Need More Help?

- [Creating Report Templates](#)
- [Add Relational Data to a Report](#) or [Add Dimensional Data to a Report](#)
- [Using Dimensional Calculations](#)
- [Set Crosstab Properties](#)
- [Specify the Font for a Report](#)
- [Format Data](#)

Using Dimensional Calculations

Insert a calculation to make your report more meaningful by deriving additional information from the data source. For example, you create an invoice report, and you want to see the total sale amount for each product ordered. Create a calculated column that multiplies the product price by the quantity ordered.

You build calculations in the expression editor using functions.

If a calculation is used in multiple reports, or by different report authors, ask your modeler to create the calculation as a standalone object in the model and include it in the relevant package.

Create a Simple Member Calculation

When working with dimensional data sources, you can select members in your report and create simple calculations. In addition to simple arithmetic calculations, you can perform the following calculations:

Calculation	Description
Rollup	Summarizes, or rolls up, all the values of the selected items in a hierarchy.
%	Calculates the value of a selected item as a percentage of another item.
% Difference	Calculates the difference between two items as a percentage.
% Of Base	<p>This calculation is only available if you select two members from different hierarchies, each from a different edge.</p> <p>This calculation takes the first selected member from edge A and the second selected member from edge B. The calculation result of a percent of base should be such that all the values for member A will compare itself to the intersect value of member A and B.</p> <p>For example, you have a crosstab showing quantity sold by year by product line. You select 2004 and Camping Equipment, right-click the selection, click Calculate, and then click % of Base (2004, Camping Equipment). The calculated row shows the percentage value of each product line for 2004 as compared to Camping Equipment for 2004.</p>
Custom	Allows you to specify your own value when performing a simple arithmetic calculation. Also allows you to change the order of operands or type a custom name for the new calculated row or column.

Steps

1. Select the items in your report to calculate.
2. Click the insert calculation button  and select the calculation to perform.
Tip: Calculations that are not applicable to the items you selected are greyed out.
3. To change the order of the operands or the name of the calculated item added to the report, click **Custom**.

The calculation appears as a new row or column in your report.

Insert a Query Calculation

When working with dimensional data, insert a query calculation into your report to add a new row or column with values that are based on a calculation. For example, you create a query calculation named Euros that converts dollars to euros by multiplying an existing dollar measure by a conversion rate. Euros can then be displayed to end users in a separate row or column.

In IBM® Cognos® Report Studio, you can create the following types of calculations:

- Create calculated members or calculated measures where the expression is a member or a tuple-based (as opposed to property-based) value expression. For example, [2005] – [2004] is a calculated member and [Revenue] – [Planned Revenue] is a calculated measure.
- Use set expressions to create calculated sets of members. For example, children ([2004]) is a set expression that displays the child members of 2004.
You must base each set expression on a hierarchy and the set expression must contain only members from this hierarchy.
- Create value expressions when you want to insert a string, number, date, or interval value.

You can also create a calculation that uses an intersection (tuple) that you already defined.

Notes:

- If you are using a DB2® data source, the subtract operator is invalid if you combine the datatypes timestamp2 and packed decimal.
- When creating an expression that will be used in a double-byte environment, such as Japanese, the only special characters that will work are ASCII-7 and ~ -- || - \$ ¢ £ ¬.

Steps

1. In the Insertable Objects pane, click the Toolbox tab .
2. Drag **Query Calculation** to the report.
3. In the Name box, type a meaningful name for your calculation.
For example, if you are calculating the difference between 2007 revenue and 2006 revenue, you could name your expression **2007 - 2006 Revenue**.
4. Click the type of calculation you want to create and select the hierarchy that contains the data that is the focus of your calculation.
5. In the Available Components pane, define the calculation:
 - Specify how you want to view the available data items in your data package by clicking the view member tree button  or the view package tree button .
 - To add data items that are not shown in the report, on the source tab 398 IBM Cognos Report Studio

- To add functions, summaries, and operators, on the functions tab , double-click elements.

Tip: To filter the visible functions, summaries, and operators in the **Available Components** pane, click the filter button  and filter by the function type, what the function returns, or what the function acts on.

Tip: You can also type the calculation directly in the **Expression Definition** box. When typing date values, ensure that the date format is correct for your database type.

For information about specific functions, see "[Calculation Components](#)" (p. 769).

6. Click the validate button .

Any validation errors appear on the **Errors** tab of the **Information** pane.

Tips:

- To show or hide the **Available Components** pane, click the available components button .
- To show or hide the **Information** pane, click the information button .
- To copy and paste expression components in the **Expression Definition** pane, you can use the copy button  and the paste button .

Create a Layout Calculation

Create a layout calculation to add run-time information to your report, such as current date, current time, and user name.

When you create layout calculations, unsupported functions do not appear in the **Functions** tab of the expression editor. Specifically, there is no **Summaries** folder, and some operators, constants, and constructs are also unavailable. These functions are not available because only the database can perform them. Report expressions and calculations based on layout expressions are performed in Report Studio.

To see the complete list of functions available in the expression editor, except for report functions, create a query calculation. All functions are available when you create a query calculation because these calculations are performed in the database and not in Report Studio.

Steps

1. In the **Insertable Objects** pane, click the **Toolbox** tab .
2. Drag **Layout Calculation** to the report.
3. In the **Available Components** box, define the calculation:
 - To add data items that are not shown in the report, on the source tab , double-click data items.
 - To add data items that are in the report but not necessarily in the model, such as calculations, on the data items tab , double-click data items.

- To add data items from a specific query, on the queries tab , double-click data items.
- To add functions, summaries, and operators, on the functions tab , double-click elements.
- To add a value that is derived from a parameter, on the parameters tab , double-click a parameter.

Parameters define prompts, drill-through reports, and master detail relationships.

Tip: You can also type the expression directly in the Expression Definition box.

For more information about creating expressions, see "[Calculation Components](#)" (p. 769).

4. Click the validate button .

Create an Intersection (Tuple)

When working with dimensional data, an intersection, also known as a tuple, is useful for obtaining a value from the combination of two or more members that you specify. Each member must be from a different hierarchy. The intersection can include only one measure.

For example, the intersection (Revenue, 2004, Cooking Gear) shows the revenue value for the year 2004 and for the product line Cooking Gear.

Use tuples to avoid report summaries that do not reflect the visible data. For more information about this issue, see "[Summaries in Report Do not Correspond to the Visible Members](#)" (p. 976).

Steps

1. In the Insertable Objects pane, click the Toolbox tab .
2. Drag the Intersection (Tuple) object to the report.
3. From the Available members and measures pane, select items and click the right arrow to move them to the Intersection members and measures box. You can also use items from the Calculated Members and Measures tab .
4. To define the hierarchy for this intersection, click a parent object in the Intersection hierarchy box.

Assigning a Hierarchy or Dimension

You must assign each calculated member, set expression, and intersection (tuple) to a hierarchy or dimension. We recommend that you select the hierarchy or dimension upon which your calculation focuses. For example, if your calculated member is based on years, select the Time hierarchy. If you create a calculated measure, select the Measures dimension.

Select only a hierarchy that cannot affect the value of the calculation. For example, the hierarchy Camping Equipment has the same value no matter what Products context it appears in the report. Therefore,

- the calculation ([Camping Equipment]-[Mountaineering Equipment]) has a well-defined meaning only in the Products hierarchy, so select that hierarchy.

- the calculation tuple ([Revenue], [Camping Equipment]) can be assigned to either the Products or Measures hierarchy, as it is not affected by either context.
- the calculations ([1999] - [Camping Equipment]) and ([Fiscal Year 1999] - [Calendar Year 1999]) do not have a well-defined meaning in any hierarchy, and therefore cannot be used reliably at all.

Browse or Search the Values of a Data Item

When building expressions in the expression editor, you can browse the data of a data item. This is useful when you do not know how a particular value is stored in the database. For example, you want to filter a report so that it shows data for only China. The actual value in the database for China is The People's Republic of China, and this is what you must insert in the filter expression to obtain the correct results.

You can also browse the data of a data item when creating conditional styles.

Steps

- In the source, data items, or query tab, right-click the data item you want to browse.

Note: For dimensional and mixed model data sources, you can view the full data tree by clicking the view package tree button  or the dimensional-only data tree by clicking the view members tree button .

- If you want to insert a single value, click **Select Value**.
- If you want to insert multiple values, click **Select Multiple Values** .
- In the **Values** box, click the value you want to insert in the expression.

Tip: Use the **Words** box to search for specific values. If the data source is case sensitive and you want to perform a case insensitive search, click the **Search** arrow and then click **Case Insensitive**.

- If you clicked multiple values, click the right arrow button  to add them to the **Selected values** box.
- Click **Insert**.

Tip: To control where values appear in the **Expression Definition** box, you can click the copy button  if you selected a single value or the copy all button if you selected multiple values instead. The values are copied to the clipboard, and you can then paste them where you want.

Searching for Values May Return Unexpected Results

In the expression editor, when searching for values for a data item, the results you obtain may contain unexpected results if the data item is not a string data type. Because users can edit the expression for a data item, IBM Cognos BI cannot determine with certainty what the data type is.

Therefore, IBM Cognos BI guesses the data type of the data item by looking at its aggregate and rollup aggregate set.

Considerations when Creating Calculations

Consider the following information and limitations when creating dimensional calculations.

Calculation Solve Order

When calculations in the rows and columns of a report intersect, calculations are performed in the following order:

- addition or subtraction
- multiplication or division
- aggregation (rollup)
- remaining arithmetic functions: absolute, round, average, minimum, maximum, medium, count
- percentage, % difference (growth), or % of total
- rank, quartile, quantile, percentile

If both calculations have the same precedence, for example, if they are both business functions, then the row calculation takes precedence.

You can override the order of precedence by changing the solve order property. For more information, see "[Resolve Multiple Calculations for Crosstabs and Charts](#)" (p. 405).

Length of Expressions

When creating layout calculations that use complex expressions, limit the length of the expression to less than 1,000 tokens. A token is a keyword, identifier, constant, or special character symbol that the server identifies when it parses the SQL. For example, the expression `1+1+1+1+1+1+1+1+1` contains approximately 28 tokens, whereas the expression `1+1+1...+1` (400 times) contains more than 1000 tokens.

Tip: The Sales Commissions for Central Europe sample report in the GO Data Warehouse (analysis) package includes expressions. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Quality of Service Indicators

Not all data sources support functions the same way. The data modeler can set a quality of service indicator on functions to give a visual clue about the behavior of the functions. Report authors can use the quality of service indicators to determine which functions to use in a report. The quality of service indicators are:

- not available (X)
The function is not available for any data source in the package.
- limited availability (!!)
The function is not available for some data sources in the package.
- limited support (!)

The function is available for all data sources in the package but is not naturally supported for that data source. IBM Cognos BI uses a local approximation for that function. Because an approximation is used, performance can be poor and the results may not be what you expect.

- unconstrained (check mark)

The function is available for all data sources.

Using Quotation Marks in Literal Strings

When inserting literal strings in an expression, you must enclose the string in single quotation marks. If the string contains a quotation mark, it must be escaped. For example, if you want to insert the string ab'c, you must type 'ab'"c'.

Limitations of Calculations

You should use only the expressions and functions available in Report Studio, and follow their syntax.

Minimal checking exists for calculations. If your calculation uses an invalid expression, your report results may contain unexpected values.

In addition, you should define member summaries as follows:

summary_function (currentMeasurewithin setset_reference)

where *set_reference* is a level or set inserted from the Source or Query tabs of the Insertable Objects pane.

Unless otherwise required, *summary_function* should be the aggregate function. If you use an explicit summary function, you may encounter problems with measures (such as profit margin, distinct count, and so on) that have complex rollup values, and/or scenario or account dimension members that do not roll up.

Know your data, and confirm with the owner of the cube where overriding the automatic aggregation is safe.

Because of the above limitations, summaries of calculations may not provide reliable values. For convenience, you may have to build reports where row summaries and calculated member columns intersect. In such reports, these intersections may contain unexpected values. In contrast, row calculations intersecting with column aggregates using the aggregate function are safe because the calculation is performed on the reliably summarized values.

Units of Measure

When creating calculations in Report Studio and IBM Cognos Query Studio, you may encounter problems with the units of measure. For example, the calculation Cost*Cost returns the unit of measure * instead of a currency unit of measure. To avoid this problem, change the format of the corresponding column to obtain the desired unit of measure.

Limitation When Using Dimensional Functions with Running and Moving Summaries

IBM Cognos BI does not currently support queries that contain both dimensional functions and running summaries or moving summaries. For example, when you drill down in a report, the query

uses the dimensional function `children`, which is not compatible with running and moving summaries. Running and moving summaries are supported for only the overall level of a dimension.

Creating Expressions Using SAP BW Data Sources

You must consider the following when creating expressions using an SAP BW data source, or you may not get the results you expect:

- The case and if/then/else constructs are not supported in filters.
- The query item identifier of the leaf-level of the 0CALDAY characteristic and its presentation hierarchies is of type date. When the values for the query item identifier are presented in Report Studio, they are formatted as dates. These formatted values should not be used in filter expressions. The correct date constant format for use in expressions is YYYY-MM-DD.
- You can apply a comparison expression with an operator other than equals to a query item that represents a level identifier. However, level identifiers are more efficient for identifying specific values. Range comparisons must be performed on the IBM Cognos application server, which slows down the performance of the report.

Using Microsoft Excel Functions with SSAS 2005 Data Sources

If you are working with a Microsoft® SQL Server 2005 Analysis Services (SSAS) data source and you want to use Microsoft Excel VBA functions such as ROUNDDOWN in MDX queries, the following Microsoft Office features must be installed on the SSAS server:

- Microsoft Office Excel
- Microsoft Visual Basic for Applications

To see a list of Excel VBA functions, see the Microsoft Web site (www.microsoft.com).

If these Microsoft Office features are not installed on the SSAS 2005 server, and you use Excel VBA functions, you encounter an error such as the following: YK-ERR-0008 The data provider returned an error message: "The '[Excel].[ROUNDDOWN]' function does not exist.".

This error occurs because the OLAP server cannot process the function since Excel function libraries are missing.

For more information, see the Microsoft Knowledge Base article # 932214 at <http://support.microsoft.com>.

Concatenating Strings

When IBM Cognos BI concatenates strings locally and if any of the involved strings contain null values, the result of the concatenation is an empty cell or a null value. This occurs because IBM Cognos BI requires that an expression that involves a null value returns a null value. Many databases ignore null strings when they perform concatenations. For example, if you concatenate strings A, B, and C, and if string B is a null value, the database may concatenate only strings A and C.

Intersecting Calculations in Crosstabs and Charts

An intersection point in a crosstab or chart can contain a value that is derived from multiple calculations.

If the query expressions for the row and column that intersect in a crosstab both include calculations, the intersecting value is the result of performing both calculations. The second calculation is performed on the result of the first calculation. If you change the order in which the calculations are performed, the resulting value at the intersection point changes.

The solve order is a positive numeric value that you assign to data items. Negative values are not supported. The data item with the lowest value is calculated first, followed by the next higher value, and so on. The data item with the highest solve order is computed last. If you do not specify a solve order, the default value of 0 is used. In crosstabs, if more than one data item has the same solve order, column items are calculated first and row items are calculated second.

For example, a crosstab contains the columns Actual revenue, Sales target, and a column that calculates the percentage by which actual revenue exceeds target revenue. The crosstab contains a row for each Sales year and a summary row that calculates the overall totals for each of the three columns. The cell where the percentage calculation and the overall total calculation intersect contains only one value. By default, IBM Cognos BI calculates the column percentage first and the summary row last. The value that appears at the intersection is therefore a sum of the percentages, which is not a meaningful result. The value is more meaningful if the overall percentage is calculated after the actual and sales target revenues are summed. Specifying a solve order for the percentage calculation that is higher than the overall total calculation gives the following result:

	Actual revenue	Sales target	<i>Percentage by which Actual Exceeds Target</i>
2004	\$43,025,310.66	\$15,051,551.00	186%
2005	\$86,096,447.76	\$29,408,080.00	193%
2006	\$102,253,419.64	\$36,068,756.00	183%
Total(Sales year)	\$231,375,178.06	\$80,528,387.00	187%

It is not necessary to specify the solve order for the overall total calculation. Because the default solve order is 0, setting the solve order for the percentage calculation to a value higher than 0 is sufficient.

Nested Calculations in Crosstabs

If a crosstab contains nested calculations, the calculations are performed in the following order by default:

1. outermost calculation on the columns
2. innermost calculation on the columns
3. outermost calculation on the rows
4. innermost calculation on the rows

If there are solve order ties, column items are calculated before row items and outer nesting items are calculated before inner nesting items.

Resolve Multiple Calculations for Crosstabs and Charts

If an intersection in a crosstab or chart contains a value that is derived from multiple calculations, you can specify the order in which to perform the calculations. The solve order must be a positive numeric value. The calculations are performed from lowest to highest solve order value.

Setting the **Rollup Aggregate Function** property to **Calculated** for a data item assigns a solve order that is higher than any other data item in a query unless a solve order is explicitly specified. If there are multiple data items that have the **Rollup Aggregate Function** property set to **Calculated**, the normal rules for solve order apply.

Steps

1. Click the data item for which to specify the solve order.

A data item can have only one solve order value. Consequently, if a data item is in a query that is used in more than one context and you want the data item to have a different solve order in each context, you must use a separate data item.

Tip: In crosstabs, to specify the solve order for the default measure, click the crosstab and, in the **Properties** pane, click the select ancestor button  and click **Crosstab**.

2. In the **Properties** pane, set the **Solve Order** property to an integer value.

Tip: To specify the solve order for the default measure, click **Default Measure Solve Order**.

Resolve Query Calculations that Intersect with Calculated Measures Defined in Microsoft Analysis Services

Microsoft SQL Server Analysis Services (SSAS) data sources may include data source calculations. Because IBM Cognos BI cannot predict the presence of a solve order on such calculations, setting the **Solve Order** property may not resolve the solve order correctly. You can suppress potentially meaningless results when querying dimensional data sources that include data source calculations. In the following example, the columns **YTD Change** and **YTD Growth** are data source calculated members and the summary values for these two columns were suppressed.

Revenue	Prior YTD	YTD	YTD Change	YTD Growth
Camping Equipment	290,355,466.50	352,910,329.97	62,554,863.47	21.54%
Personal Accessories	329,883,402.11	443,693,449.85	113,810,047.74	34.50%
Outdoor Protection	5,942,944.77	4,471,025.26	-1,471,919.51	-24.77%
Golf Equipment	138,993,560.79	174,740,819.29	35,747,258.50	25.72%
Mountaineering Equipment	93,654,629.48	141,520,649.70	47,866,020.22	51.11%
Total(Product line)	858,830,003.65	1,117,336,274.07	--	--

When a query-defined calculated member and a data source calculated member intersect, the value is the result of performing both calculations. The second calculation is performed on the result of the first calculation. Whether the final value is meaningful depends on the order in which the calculations are performed. Because IBM Cognos BI cannot determine the presence of these members ahead of time, it cannot automatically determine the correct solve order.

This issue occurs mostly with **Aggregate** summaries in SSAS 2000. In most cases, SSAS 2005 gives the expected results when no solve order is defined in the cube. Therefore, when using SSAS 2005, you do not need to set either the **Solve Order** property in the query or the **Calculation Intersection** property. If your SSAS 2000 cubes include calculated members, we recommend that you migrate to SSAS 2005.

Note: Dimensionally-modeled relational (DMR) data sources query one or more underlying relational databases, and therefore cannot contain data source calculations. They are not considered dimensional data sources in the context of this topic.

Steps to Suppress Values for Specific Data Items

1. Pause the pointer over the query explorer button  and click the query that contains the data item for which to suppress values.
2. In the **Data Items** pane, click the data item.
3. In the **Properties** pane, set the **Calculation Intersection** property to **Show "Not applicable"**.

When a query-defined calculated member and a data source calculated member intersect, the resulting value is suppressed. If the intersecting members do not contain data source calculations, the value is not suppressed.

Null (Missing) Values in Calculations and Summaries

The data source that you use determines how null (or missing) values are handled in calculations, summaries, and other expressions.

Null values represent missing data values. For example, a new eyewear product, called Abby, is introduced in 2009. Therefore there are no sales in 2008. If you create a report with the years 2008 and 2009, the values for sales of Abby for the year 2008 are null (missing).

By default, null values appear as blank cells in a crosstab or list report. You can change the default to specify the display of other characters, such as the word null, by changing the data format of the **Missing Value Characters** property for the cells or report. For more information, see "[Set the Default Data Formats](#)" (p. 495).

Arithmetic Operations

Null values in arithmetic operations yield different results depending on the type of data source.

In Microsoft SQL Server Analysis Services (SSAS) and Cubing Services data sources, a null value in arithmetic operations is treated as follows:

- In operations such as addition (+), a null value yields the same result as zero ($9 + \text{NULL} = 9$)
- In operations such as multiplication (*), a null value yields a null result ($9 * \text{NULL} = \text{NULL}$)

You can accommodate this behavior by using more complex expressions such as the following:

- `if ([M] is not NULL, then ([M]) else (0))`
- `if ([M] is not NULL, then (<expression involving M>) else NULL)`

In relational and other OLAP data sources the result of an operation that includes a null value is always null. For example, $9 + \text{NULL} = \text{NULL}$.

Equality Comparisons

Equality comparisons, such as equal to (=) and not equal to (<>), compare two values and return a Boolean value of true or false.

For all data sources, equality comparisons are treated as tests for missing data. Inequality comparisons are treated as tests for non-missing data.

For example, `NULL=NULL` is true and `NULL=0` is false.

Ordered Comparisons

In ordered comparisons, such as `rank` and `quantile` functions, null values are handled as follows.

For relational data sources, the result of the comparison is always false. For example, `NULL < 1` is false.

For Microsoft SSAS, Cubing Services, and other OLAP data sources, the null value is treated as zero. For example, `NULL < 1` is true.

Boolean Functions

In Boolean operations, such as `or`, null values are handled as follows.

For relational and dimensionally-modeled relational data sources, the result of a Boolean operation is always null. Boolean functions are used in conditions and the null result is treated as a false condition. For example, `NULL OR TRUE = NULL (FALSE)`

For Microsoft SSAS, Cubing Services, and other OLAP data sources, the null value is treated as if it is false. For example, `NULL OR TRUE = TRUE`.

Summary Functions

All summary functions, such as `aggregate`, `average`, and `maximum`, ignore null values. For all functions except `count`, if all values in the expression are null, the result is null. For `count`, if all values in the expression are null, the result is zero.

For example, if the value for `Quantity` for 2003 is 10 and the values for 2001 and 2002 are null, then the average is as follows:

- `average ([Quantity] within setset([2002], [2003])) = 10`
- `average ([Quantity] within setset([2002], [2001])) = NULL`

The count is as follows:

- `count ([Quantity] within setset([2002], [2003])) = 1`
- `count ([Quantity] within setset([2002], [2001])) = 0`

String Operations

For all data sources, the result of string operations, such as concatenation (`||`), that include a null value is always null.

For example, `'A' || NULL = NULL` and `NULL || NULL = NULL`.

Dimensional Coercion Rules

IBM Cognos BI uses dimensional coercion rules to automatically convert dimensional types to better match other types. These conversions help you build simpler expressions, making them easier

to understand. In addition to the implicit rules that IBM Cognos BI provides, you can make explicit data type conversions using various functions, such as `children`.

IBM Cognos BI supports the following types of coercion:

- coercion of an object of one dimensional type into another, such as a level into a member set
- coercion of a dimensional object into a numeric, date, time, or string value, such as a numeric measure into a numeric value

Coercion rules are applied to expressions and data items. In expressions, an operator or function may require operands to be of a particular dimensional type. When an operand is not of the required type, one or more coercion rules may be applied to coerce the operand to the appropriate type.

Coercion rules may also be applied to data items to coerce the data item to a collection of members or values.

Function Operands

The following describes how coercion rules are applied to function operands:

- If the operand is of the required type, no coercion is required.
- If the function operand is supposed to be a numeric value, it is coerced into one.
A coercion rule exists for each type of dimensional object coercion.
- If the function operand is supposed to be a dimensional object and a coercion exists to coerce the operand to the required type, the coercion is applied.
- If no coercion exists, an error message with the code QE-DEF-0478 appears to indicate that an unsupported conversion of a dimensional object from the source to target type occurred.

Comparison and Other Symmetric Operators

Binary operators that accept operands of more than one type, such as equals (=) and `in`, need both operands to be the same dimensional type.

No coercion is possible between value domains (numeric, date, time, and string) or between members and values. Consequently, if either operand is a value type, both must be in the same value domain and converted explicitly using a function. Otherwise, the query fails.

Members and member set operands are valid only with the operators equals (=), not equals (<>), `in`, and `not in`, where the right side of the expression is a member, member set, or prompt. Only the following are valid:

- `[member/member set] = [member]`
- `[member/member set] <> [member]`
- `[member/member set] = ?p?`
- `[member/member set] <> ?p?`
- `[member/member set] in ([member], ...)[member/member set] not in ([member], ...)`

- $[member/member\ set] \text{ in } ([member\ set])[member/member\ set] \text{ not in } ([member\ set])[member/member\ set]$ in ?p? $[member/member\ set] \text{ not in } ?p?$

Exceptions

For the left operand, member sets are supported in detail and summary filter expressions, but not in expressions that use the `filter` function. Members are not supported in detail and summary filters, but they may be used in expressions that use the `filter` function.

The `in_range` operator is not supported for members and member sets.

Normal coercion rules apply to these operands.

NULL operands are considered values, rather than members.

Examples

The following examples illustrate how coercion is applied to levels in expressions with operators.

`[Sales].[Products].[].[Product Line] = [Sales].[Products].[].[Product Line]->[Outdoor Equipment]`

The left operand is coerced to the following member set:

`members ([Sales].[Products].[].[Product Line])`

The following expressions are invalid:

- `[Sales].[Products].[].[Product Line] = NULL`
- `[Sales].[Products].[].[Product Line] + 1`
- `[Sales].[Products].[].[Product Line] = 'Canada'`
- `[Sales].[Products].[].[Product Line] > 2000`

The following examples illustrate how coercion is applied to members in expressions with operators.

In each of the examples below, the left operand is coerced to a value of the current cell at this member:

`(tuple ([Sales].[Products].[].[Product Line]->[Outdoor Equipment]))`

- `[Sales].[Products].[].[Product Line]->[Outdoor Equipment] = NULL`
- `[Sales].[Products].[].[Product Line]->[Outdoor Equipment] + 1`
- `[Sales].[Products].[].[Product Line]->[Outdoor Equipment] > 2000`
- `[Sales].[Products].[].[Product Line]->[Outdoor Equipment] + 'Canada'`

Coercion Rules for Dimensional Objects

The following table describes coercion rules that are applied when coercing an object of one dimensional type into another.

Note: Measures and Member Unique Names (MUNs) are considered members.

Source type	Target type	Result
Measure dimension	Hierarchy	A measure dimension can be used without any conversion as a hierarchy.
Measure dimension	Member set	The member set contains the members of the measure dimension. This coercion is done in two steps. First, measure dimensions are coerced to hierarchies, and then hierarchies are coerced to member sets.
Hierarchy	Member set	A set containing the members of the hierarchy.
Level	Member set	A set containing the members of the level. The equivalent expression is <code>members (level)</code> .
Measure	Member	A measure can be used without any conversion as a member.
Measure	Tuple	This coercion is done in two steps. First, measures are coerced to members, and then members are coerced to tuples.
Measure	Member set	This coercion is done in two steps. First, measures are coerced to members, and then members are coerced to member sets.
Member	Tuple	The tuple consists of the member. The equivalent expression is <code>tuple (member)</code> .
Member	Member set	The set consists of the member. This coercion is done in two steps. First, members are coerced to tuples, and then tuples are coerced to member sets.

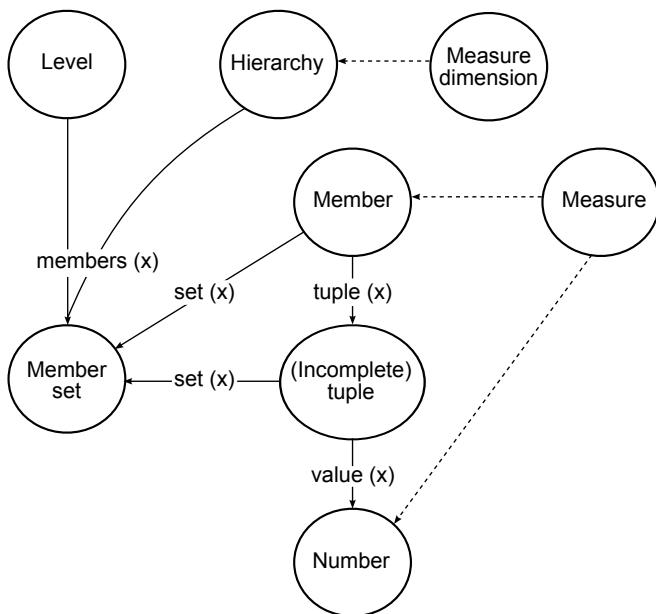
Coercion Rules for Numeric and String Values

The following table describes the coercion rules for expressions or prompts that return a numeric or string value. These rules apply to functions and operators that require an operand of a particular data type, as well as to binary operators that require operands of the same type.

Source type	Target type	Result
Measure	Number, Date/Time, or String	The result is the value of the measure at the current cube intersection.
Member	Number	<p>The result is the value of the cell at the cube intersection specified by the member.</p> <p>This coercion is done in two steps. First, members are coerced to tuples, and then tuples are coerced to the target type.</p>
Tuple	Number	<p>The value of the cell at the cube intersection specified by the tuple.</p> <p>The equivalent expression is <code>value (operand or parameter)</code>.</p>

Coercion Paths

The following diagram shows how coercion rules are connected. The expression that coerces a source node type to a target node type is built by following the path that connects them. A dotted line indicates a coercion that does not require an explicit conversion function. This occurs whenever the source type is a specialization of the target type. For example, every measure is a member so you can use measures without conversion wherever a member is required.



Recommendation - Use Member Unique Name (MUN) Aliases

If you are working with a dimensional data source, use MUN aliases to simplify building reports and expressions. In addition, you can tie the MUN back to the member in the package.

When working with dimensional data sources, IBM Cognos BI uses MUNs to uniquely identify members. MUNs are very complex. For example, the MUN for the member 2004 might appear as

[great_outdoors_company].[Years].[Years].[Year]->: [PC] . [@MEMBER] . [20040101-20041231].

When you are building expressions that involve members, the use of MUNs makes these expressions difficult to read and maintain.

Tip: To view the MUN of a member, in the **Insertable Objects** pane, on the **Source** tab , right-click the member and click **Properties**.

Report Studio has an option that automatically generates an alias for MUNs. The MUN alias is a data item that has the MUN as its definition. For the previous example, you would see 2004 as the data item, and the MUN would appear in the data item's **Expression** property. Enable this option to produce a MUN alias whenever you add a member to a report or expression.

Note: Do not modify or rename MUN aliases, because they will lose their connection to their respective members in the package.

Running a Report That Contains Members

If you are using members in a report and your modeler has updated the data source in a way that changes member unique names, the following problems may arise when you run the report.

- If the report queries an OLAP data source, you receive an error message explaining that some specific members cannot be found.
- If the report queries a dimensionally modeled relational (DMR) data source, data items whose member unique names have changed do not appear in the report. Calculations that refer to members that have changed no longer contain values from the members.

To resolve these problems, you must update the member unique names in the report. Open the query that contains the members that you need to update in Query Explorer. Delete the member from the pane in which it appears and reinsert it from the **Source** tab. For example, if you inserted the member as a detail filter, delete it from the **Detail Filters** pane and reinsert it.

Drilling Up and Drilling Down in Dimensional Reports

When working with dimensional or dimensionally-modeled relational (DMR) data sources, you can create reports that allow the reader to view more general or more detailed information on your data within a predefined dimensional hierarchy (such as Years - Year - Quarter - Month) without having to create different reports.

You can link groups of data items from different queries so that when you drill up or drill down in one query, the data item also drills up or drills down in the linked queries. Linking data items is useful if your report includes more than one query with the same data item. For example, if your report shows a list and a crosstab that each include the Product Line data item, you can link the Product Line data item from the crosstab query to the Product Line data item from the list query so that when you drill up in the crosstab Product Line data item, the list Product Line data item also drills up.

Member Sets

Member sets are used to group data items that are logically related for various actions, such as drill actions, zero suppression, and ranking. They can be defined as a flat list or as a tree structure in which member sets in the same parent chain are considered related.

For example, for drill operations, a member set defines the set of items that can potentially change when a given item in the set is drilled on. The values of other items in the query or even those in the same hierarchy are preserved when any item in this set is drilled on. Usually, a member set references items that have logical roles in a drill action, such as a detail, a summary of the detail, or a parent of the detail. A single data item can belong to only one member set.

If you do not define member sets for items, the IBM® Cognos® BI server associates items into default member sets and behaviors using simple dimension rules on item expressions. You can override the behavior for a particular item while other items continue to use the default.

When you define a member set, you must explicitly define behaviors for each item in the set. Items in the set that have no behaviors have their values preserved.

Drill behaviors always act from a root member set. This means that when an item is drilled on, the root member of its member set is found and all items from the root down are processed. Although calculations and aggregates are not directly related by hierarchy, they respond because of their dependence on the items upon which they are based.

Create Member Sets

Create a member set when you want to define a non-default drill behavior. You specify what items respond to a drill action by adding them to the member set.

Steps

1. Pause the pointer over the query explorer button  and click the query in which to create a member set.
2. In the Properties pane, set the Define Member Sets property to Yes.
3. Double-click the query and click the Member Sets tab.
4. In the Insertable Objects pane, drag items to the work area.

To define a member set as a tree structure, drag the item that will serve as the root item of the set to the work area, and then drag other items over the root item to create the tree structure.

You can also nest member sets.

Create a Drill-up and Drill-down Report

You can link groups of data items from different queries so that when you drill up or drill down in one query, the data item also drills up or drills down in the linked queries.

Before you begin, ensure that you are using a dimensional data source.

Tip: The Returned Items sample report in the GO Data Warehouse (analysis) package includes drilling. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Column Titles When Drilling up or Down

When you drill down or up, you can specify whether the column title shows the member caption value or the level label value. By default, titles use member captions. Using level labels may be more meaningful. For example, within the Country column, you drill down on Australia. By default, the column title becomes Australia, the member caption. It may be more meaningful to use Country, the level label, as the column title.

Steps

1. From the **Data** menu, click **Drill Behavior**.
2. On the **Basic** tab, in the **Report output** drill capabilities box, select the **Allow drill-up and drill-down** check box.

By default, IBM Cognos Report Studio determines which items can be drilled on based on the dimensional structure.

Tip: To enable the report to be used as the source during a package drill-through, select the **Allow this report to be a package-based drill-through source** check box and click **OK**. For more information about package drill-through, see the *Administration and Security Guide*.

3. To disable drill-up or drill-down for a data item, select the data item in either the **Disable drill-up for** box or the **Disable drill-down for** box.
4. From the **Advanced** tab, you can change the drill-up or drill-down behavior for any data item by selecting the data item and then choosing one of the following behaviors.

Behavior name	Drill-up behavior	Drill-down behavior
Preserve	The data item's value will remain unchanged.	The data item's value will remain unchanged.
Empty Set	The set of values associated with this data item is set to be the empty set (novalues). For crosstabs, the data item will effectively be removed from the report.	The data item will be removed from the report.
Replace Item	The data item's value will change to become the parent (if a lowest-level summary was drilled up on) or the grandparent (if a lowest-level detail of a dimension was drilled up on) of the item that was drilled on.	The data item's value will change to become the item that was drilled on.

Behavior name	Drill-up behavior	Drill-down behavior
Replace Expression	The data item's value will change to become the children of the parent (if a lowest-level summary was drilled up on) or the children of the grandparent (if a lowest-level detail of a dimension was drilled up on) of the item that was drilled on.	The data item's value will change to become the children of the item that was drilled on.
Change Expression	The data item's value will change to become the value of some other data item in the query.	The data item's value will change to become the value of some other data item in the query.
Depth Based Expression	The data item's value will change to become all data items n generations higher in the dimensional hierarchy than the item that was drilled on. The number of generations or levels is determined by the Depth value.	The data item's value will change to become all data items n generations higher in the dimensional hierarchy than the item that was drilled on. The number of generations or levels is determined by the Depth value.
Ancestor	The data item's value will change to become the data value of the ancestor n generations higher in the dimensional hierarchy than the item that was drilled on. The number of generations or levels is determined by the Depth value.	The data item's value will change to become the data value of the ancestor n generations higher in the dimensional hierarchy than the item that was drilled on. The number of generations or levels is determined by the Depth value.

5. In the **Column title drill behavior** box, specify whether to use member captions or level labels as the columns titles for each data item.

6. From the **Data Items Linking** tab, you can link groups of data items:

- Click the new button .
- Type a name for this group of linked data items and click **OK**.

To rename a group of linked data items, click the group, click the rename button , type a name, and click **OK**.

- Use the arrows to move data items from the queries on the left to the groups on the right.

Note: You can only add a data item to one group.

The report will generate links for any item that can be drilled up or down on.

When you run a report, you can drill down or drill up by right-clicking the data item and choosing the action from the context menu. The menu items will be disabled if an item cannot be drilled up or down on.

Recommendation - Drilling Down in Very Large Data Sources

When you drill down in a report that uses a very large data sources query results can be much larger than what can be held in memory. For example, if you insert the lowest level of four hierarchies that each contain 1000 members, the report can contain 1000,000,000,000 cells. At best, this query will take a very long time to run. While this large query executes, all other queries for the same server process will likely be blocked. For most data sources, the query will likely fail due to insufficient memory or timing out. Then the memory is released and normal services resume.

However, when using a Microsoft® SQL Server 2005 Analysis Services (SSAS) cube, the memory consumed by the SSAS client is not always released in a way that other queries can effectively reuse. The server continues to run with insufficient memory, causing new queries to either run very slowly or fail completely. You may encounter an error, and, to continue, the system administrator must manually stop and restart the IBM Cognos BI service.

To avoid these problems, consider the size of hierarchy levels and sets when creating reports and avoid combining them in ways that create large queries. To determine the size of a set, create and run a very small report that includes only a count of the members within that set. You can also use filtering techniques to focus your data.

Example - Create a Dashboard Report

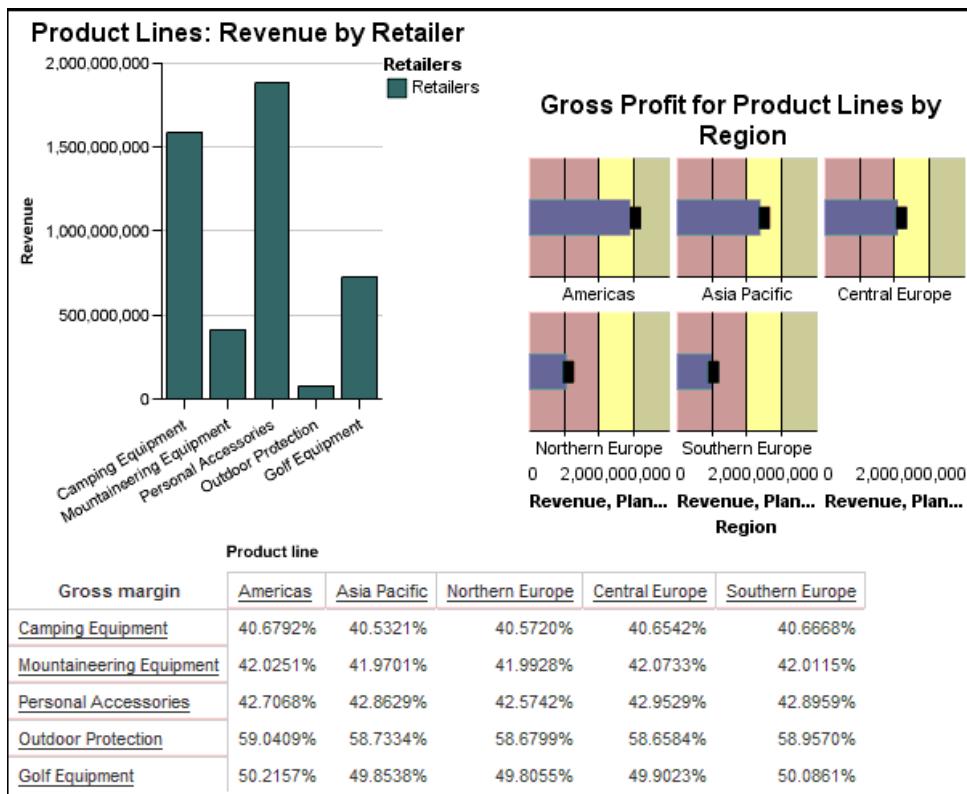
Create a dashboard report to show a high-level graphical view of company performance indicators. By using dashboard reports, users can

- drill up and drill down to see higher and lower levels of detail
- navigate to other targets, including Report Studio reports, IBM Cognos Query Studio reports, and IBM Cognos Analysis Studio analyses
- include multiple charts derived from different data sources in a single report

Dashboard reports are not the same as dashboards created in IBM Cognos Business Insight. Dashboard reports display several different report objects in one report, but the information in the report is static. Dashboards created in Business Insight are a collection of IBM Cognos content that you assemble into an interactive dashboard. Content is displayed in widgets. For information about creating interactive and sophisticated dashboards, see the IBM Cognos Business Insight *User Guide*. You can also create dashboards within IBM Cognos Connection. In these dashboards, content is displayed in portlets within pages. For more information, see the IBM Cognos Connection *User Guide*.

In this topic, you learn how to create a dashboard report so that users can examine important sales data in a single view.

It should take 15 to 20 minutes to complete this topic, and your dashboard report will look like this.



Steps

1. Create a blank report with the GO Data Warehouse (analysis) sample package.
2. Add a column chart, a bullet chart, and a crosstab.
3. Drag the following items to the bullet chart:
 - Revenue (in Sales and Marketing (analysis), Sales, Sales fact) to the Bullet Measure drop zone
 - Planned Revenue to the Target Measure drop zone
 - Product line level (in Products) to the Series (matrix rows) drop zone
 - Region level (in Retailers) to the Categories (matrix columns) drop zone
4. Drag the following items to the column chart:
 - Revenue (in Sales fact) to the Default measure (y-axis) drop zone
 - Product line level (in Product) to the Categories (x-axis) drop zone
 - Retailers level (in Retailers) to the Series (primary axis) drop zone
5. Drag the following items to the crosstab:
 - Gross margin (in Sales fact) to the Measures drop zone

- Product line level (in **Product**) to the **Rows** drop zone
 - Region level (in **Retailer site**) to the **Columns** drop zone
6. Click the bullet chart.
 7. Set the **Size & Overflow** property width to 300 px and height to 300 px.
 8. Set the **Title** property to **Show**.
 9. Double-click the chart title and type the following text:
Gross Profit for Product Lines by Region
 10. Set the **Font** property for the chart title to **Arial, 12pt, Bold**.
 11. Set the **Padding** property for the left and bottom padding to 20 px.
 12. In the **Bullet Indicators** property, set the bar width to 30%.
 13. Click the chart, click the palette button , and click **Contemporary** from the available palettes.
 14. Set the **Size & Overflow** property width to 300 px.
 15. Set the **Title** property to **Show**.
 16. Double-click the chart title and type the following text:
Product Lines: Revenue by Retailer
 17. Set the **Font** property for the chart title to **Arial, 12pt, Bold**.
 18. From the **Data** menu, click **Drill Behavior**.
 19. Select the **Allow drill-up and drill-down** check box.
 20. Run the dashboard report to view what it will look like for your users.
- Users can drill up or drill down on values in the report to view related information.

Need More Help?

- [Creating Report Templates](#)
- [Add Relational Data to a Report](#) or [Add Dimensional Data to a Report](#)
- [Customize a Current Default Gauge Chart](#)
- [Specify the Height and Width of an Object](#)
- [Specify the Font for a Report](#)
- [Create a Drill-up and Drill-down Report](#)

Chapter 12: Adding Prompts to Filter Data

You can add prompts to a report to add interactivity for users. Prompts act as questions that help users to customize the information in a report to suit their own needs. For example, you create a prompt so that users can select a product type. Only products belonging to the selected product type are retrieved and shown in the report.

Prompts are composed of three interrelated components: parameters, prompt controls, and parameter values. Parameters are based on parameterized filters and form the questions to ask users. Prompt controls provide the user interface in which the questions are asked. Parameter values provide the answers to the questions.

IBM® Cognos® Report Studio provides several ways to create prompts. You can

- [use the Build Prompt Page tool](#)
- [build your own prompt and prompt page](#)
- [create a parameter to produce a prompt](#)
- [insert prompts directly into the report page](#)
- in dimensional reporting, you can also [define prompts using context filters](#)

If you include reports from different packages in an interactive dashboard that uses global filters, ensure that you use the same parameter name for the prompt in all the reports. For more information about creating interactive dashboards in IBM Cognos Connection, see the *IBM Cognos Business Intelligence Administration and Security Guide*.

You can also create prompts in the package. For more information, see the *IBM Cognos Framework Manager User Guide*.

You can also use context filters, also known as slicer filters, to focus the data in your report.

Syntax of Prompt Expressions

Prompt expressions use the following syntax, where p represents the parameter name.

You can also use these expressions to create parameterized data items that you can then use in master detail relationships.

The type of expression you use depends on whether you have chosen relational or dimensional reporting style. For more information about reporting styles, see "["Relational vs. Dimensional Reporting Styles" \(p. 74\)](#)".

Relational: Detail Filter Expressions	Dimensional: Context (Slicer) or Edge Expressions
<p>[<i>data item</i>] = ?p?</p> <p>for single-select prompts and master detail relationships</p>	<p>[<i>level or hierarchy</i>]->?p?</p> <p>For single-select member prompts and master detail relationships</p>
<p>[<i>data item</i>] in ?p?</p> <p>for multi-select prompts</p>	<p>set([<i>level or hierarchy</i>]->?p?)</p> <p>For multi-select member prompts</p>
	<p>filter [<i>set</i>], [<i>property expression</i>] = ?p?)</p> <p>For single-select value prompts and master detail relationships</p> <p>A property expression is an expression that is assigned to a property to enable dynamic update of the property when the report runs.</p>
	<p>filter [<i>set</i>], [<i>property expression</i>] IN ?p?)</p> <p>For multi-select value prompts</p> <p>A property expression is an expression that is assigned to a property to enable dynamic update of the property when the report runs.</p>

Use the Build Prompt Page Tool

Use the **Build Prompt Page** tool to quickly add filters and prompts to a report. This tool creates filters in the query using the prompt information properties that are set in the IBM® Cognos® Framework Manager model. For more information about the prompt information properties, see the Framework Manager *User Guide*.

Note: The **Build Prompt Page** tool creates a static report specification, so if the prompt information properties are updated in the Framework Manager model, those updates will not be reflected in your report.

Tips: The Promotion Success sample report in the GO Data Warehouse (analysis) package includes a prompt page. The Historical Revenue sample report in the Sales and Marketing (cube) package and the Eyewear Revenue by Brand and Size sample report in the GO Data Warehouse (analysis) package include multiple prompts. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Steps

1. Click the column or row on which users will be prompted.

To create multiple prompts or a [cascading prompt](#), click more than one column or row.

2. From the Tools menu, click Build Prompt Page.

A prompt page is created that has

- a page header
- a prompt control for each selected column
- a page footer containing Cancel, Back, Next, and Finish buttons

You can add more objects or modify existing elements. For example, you can [change the prompt control](#) chosen by Report Studio.

Build Your Own Prompt and Prompt Page

Create your own prompt and prompt page to control how they appear in a report.

Tip: The Promotion Success sample report in the GO Data Warehouse (analysis) package includes a prompt page. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Steps

1. Pause the pointer over the page explorer button  and click **Prompt Pages**.
2. In the **Insertable Objects** pane, on the **Toolbox** tab, drag **Page** to the **Prompt Pages** box.
3. Double-click the page you just created.
4. In the **Insertable Objects** pane, on the **Toolbox** tab , drag one of the following prompt controls to the prompt page.

Prompt control	Description
Text Box Prompt	<p>Retrieves data based on a value that users type.</p> <p>Use this control when users know exactly what value they want to enter, such as a name or account number.</p>
Value Prompt	<p>Retrieves data based on values that users select from a list.</p> <p>Use this control to show a list of possible values from which users can choose.</p> <p>Note: The maximum number of items that can appear in a list is 5000.</p>

Prompt control	Description
Select & Search Prompt	<p>Retrieves values based on search criteria that users specify. Data is then retrieved based on values users select from the search results.</p> <p>Use this control instead of a value prompt if the list of values is very long, which can slow down performance.</p> <p>Tip: Users have the option of performing a case sensitive or case insensitive search. A case sensitive search is faster, while a case insensitive search usually returns more values.</p> <p>You cannot use this control if you are working with SAP BW data sources.</p>
Date Prompt	<p>Retrieves data based on a date that users select.</p> <p>Use this control to filter a date column. Users can retrieve data for a specific day, a set of days, or a range of days.</p>
Time Prompt	<p>Retrieves data based on a time that users select.</p> <p>Use this control to restrict a report to a particular time or time range. For example, users can see how many orders are received after business hours. Users can then use this information to determine the number of staff needed to work after hours.</p>
Date & Time Prompt	<p>Retrieves data based on a date and time that users select.</p> <p>Use this control to filter a datetime or timestamp column. This control is useful for specifying ranges. For example, users can retrieve all orders received from Monday at 12:00 a.m. to Friday at 5:00 p.m.</p>
Interval Prompt	<p>Retrieves data based on a time interval that users specify.</p> <p>Use this control to retrieve data that is related to the passage of time. For example, users can retrieve a list of products that were returned 30 or more days after they were purchased.</p>

Prompt control	Description
Tree Prompt	<p>Retrieves data based on values that users select from a list. Values are organized hierarchically.</p> <p>This control is useful when you are working with dimensional data sources. Data is shown from the top of a dimension hierarchy to the most detailed member, and users can choose the level of detail they want to view in the report. For more information about tree prompts, see "Control the Data That Appears in a Tree Prompt" (p. 435).</p> <p>Tree prompts let you page through large numbers of prompt values with a More link at the bottom of the tree.</p>
Generated Prompt	<p>Selects a prompt control based on the data type of the data item.</p> <p>This control acts like a placeholder. When users run the report, the control is replaced by the appropriate prompt control. For example, if users are prompted for date values, the control is replaced by a date & time prompt.</p>

The **Prompt Wizard** dialog box appears.

5. If you are creating a text box, date, time, date and time, interval, or generated prompt, do the following:
 - Create a new parameter for the prompt or use an existing parameter.
 - Click **Next**.
 - If you created a new parameter, define the expression by selecting a data item from the package and the operator to use.

Tip: Make the prompt optional by selecting the **Make the filter optional** check box.

 - Go to step 7.
6. If you are creating a value, select & search, or tree prompt, do the following:
 - Create a new parameter for the prompt or use an existing parameter.
 - Click **Next**.
 - If you created a new parameter and you want to use the parameter to filter data, select the **Create a parameterized filter** check box and define the expression by selecting a data item from the package and the operator to use.

If you are creating a tree prompt, you must choose **in** in the **Operator** box.

You can also use a parameter to provide a value for a layout calculation, such as showing a user's name in the report. When the report is run, you can use a parameter to prompt the user to type his name and have it appear in the report.

Tip: Make the prompt optional by selecting the **Make the filter optional** check box.

- Click **Next**.
- If you created a parameterized filter and you have more than one query defined in the report, select the check box for the query on which to filter and click **Next**.
- Select the **Create new query** check box to create the query that will be used to build the list of data values shown when the report is run.

Tip: Do not create a new query if you want to link the prompt to an existing query or if you intend to create the query at a later time.

- Click the ellipsis (...) button beside **Values to use** and click the data item on which to prompt.
- To choose a data item that is different than what users will see when they are prompted, click the ellipsis (...) button beside **Values to display** and click the data item.
- To create a **cascading prompt**, in the **Cascading source** box, click the parameter that represents the cascade source.

7. Click **Finish**.

The prompt control is added to the prompt page. A prompt page is like a report page. You can insert graphics and text and apply formatting.

You can also **modify** the properties of the prompt control by clicking it and making changes in the **Properties** pane.

Example - Create a Report Showing Products Shipped for a Specific Time Interval

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a report that shows the time interval between closing and shipping dates.

Steps

1. Open IBM® Cognos® Report Studio with the GO Data Warehouse (query) package.
2. In the **Welcome** dialog box, click **Create a new report or template**.
3. In the **New** dialog box, click **List** and click **OK**.
4. In the **Insertable Objects** pane, on the **Source** tab , expand **Sales and Marketing (query)** and **Sales (query)**.
5. Expand **Sales fact** and double-click **Quantity**.
6. Expand **Products** and double-click **Product type**.
7. Expand **Sales order** and double-click **Order number**.
8. Create a time interval calculation:

- In the **Insertable Objects** pane, on the **Toolbox** tab , drag **Query Calculation** to the right of **Quantity** in the list.
- In the **Name** box, type the following and click **OK**:
Time Interval
- In the **Expression Definition** box, type the following and click **OK**:
[Sales (query)].[Time (close date)].[Date (close date)]-[Sales (query)].[Time (ship date)].[Date (ship date)]

9. Create a filter:

- From the **Data** menu, click **Filters** .
- Click the add button .
- In the **Expression Definition** box, from the **Data Items** tab , double-click **Time Interval** calculation you created.
 - Add **> ?p1?** after **Time Interval** to create the following expression:
[Time Interval] > ?p1?
- Click **OK** twice.

A parameterized filter is created that will return data when the difference between the return date and the order date is greater than the value specified by the user.

10. Create a prompt page:

- Pause the pointer over the page explorer button  and click **Prompt Pages**.
- In the **Insertable Objects** pane, drag **Page** to the **Prompt Pages** pane, and then double-click it.
- In the **Insertable Objects** pane, on the **Toolbox** tab, drag **Interval Prompt** to the work area.
The **Prompt Wizard** dialog box appears.
- Click **Use existing parameter**, and then click **p1**.
- Click **Finish**.

11. Run the report.

An interval prompt appears.

12. In the **Days** box, type a value and click **Finish**.

Tip: You can also type values for the **Hrs** and **Mins** boxes.

A list report appears showing all products that were shipped after the time interval you specified. For example, if you typed 5, the list will show products that were shipped more than 5 days after the order date.

Order number	Product type	Quantity	Time Interval
100001	Lanterns	256	29 days
100005	Binoculars	65	9 days
100005	Knives	502	9 days
100005	Watches	24	9 days
100005	Woods	26	9 days
100011	Cooking Gear	2,172	10 days
100011	First Aid	312	10 days
100011	Insect Repellents	1,792	10 days
100011	Knives	2,490	10 days
100011	Navigation	98	10 days
100011	Sleeping Bags	242	10 days
100011	Sunscreen	1,164	10 days

Create a Parameter to Produce a Prompt

IBM® Cognos® Report Studio can automatically generate prompted reports based on parameters you create. When you run the report, Report Studio can generate a prompt page for each parameter not associated to an existing prompt page depending on whether the prompt run option is selected or not.

Note: When Report Studio automatically generates a prompt, it creates filters in the query using the prompt information properties that are set in the IBM Cognos Framework Manager model. For more information about the prompt information properties, see the Framework Manager *User Guide*.

You can also define parameters when you want to create a drill-through report or define master detail relationships.

Steps

- From the Data menu, click Filters .
 - On the Detail Filters tab, click the add button .
- The Detail Filter dialog box appears.
- In the Available Components box, click the Source tab  or the Data Items tab  to select the data item to use for the prompt:
 - To filter data based on data items not shown in the report, double-click a data item on the Source tab.
 - To filter data that appears in the report but not necessarily in the model, such as calculations, double-click a data item on the Data Items tab.
- The data item appears in the Expression Definition box.
- In the Expression Definition box, type an operator after the data item or select an operator from the Functions tab .

The operator sets some of the default properties of the prompt. For example, if the operator is equals (=), users will be able to select only a single prompt value and the prompt's Multi-Select property is set to No.

For more information about creating expressions, see "[Using Relational Calculations](#)" (p. 342) or "[Using Dimensional Calculations](#)" (p. 396).

5. Type a name after the operator to define the prompt parameter. A question mark must precede and follow the name.
6. To specify whether the prompt is mandatory, in the Usage box, click Required, Optional, or Disabled.

Create a Prompt Directly in a Report Page

You can add prompt controls directly in a report page instead of creating a prompt page.

Prompt controls that are added to report pages will not appear in the following:

- saved reports
- PDF reports
- reports that are sent to users by email
- scheduled reports

Prompt controls are interactive. They are used to satisfy parameter values before running a report. As a result, prompt controls added to a report page only appear when you run the report in HTML format. When you run a report in HTML format, users select which values they want to see, and the report is refreshed, producing a new report.

For the non-interactive reports listed above, prompt parameter values must be collected and satisfied before the report is run. You provide the parameter values using the **Run options** tab in IBM® Cognos® Connection. If you do not provide all the required values, the report will fail to run. You can access the Run options tab by clicking the set properties button  for the report.

Tip: The Revenue by Product Brand (2005) sample report in the Sales and Marketing (cube) package and the Rolling and Moving Averages interactive sample report include value prompts. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Steps

1. In the Insertable Objects pane, on the **Toolbox** tab , drag a [prompt control](#) to the report. The **Prompt Wizard** dialog box appears.
2. [Provide the information](#) necessary to create the prompt.

Tip: Make the prompt optional by selecting the **Make the filter optional** check box. Otherwise, when you run the report for the first time, the prompt appears in a generated prompt page rather than in the report page. Alternatively, you can specify a default selection for the prompt.

3. In the work area, click the prompt.
4. In the **Properties** pane, set the **Auto-Submit** property to **Yes**.

If you do not set the **Auto-Submit** property to **Yes**, the report will not refresh when users select a different prompt value.

Tip: An alternative to setting the prompt to auto-submit is to add a prompt button from the **Toolbox** tab and set its **Type** property to **Finish**.

The prompt control is added to the report page. You can modify its properties by clicking it and making changes in the **Properties** pane.

Modifying Prompts

For each prompt you create, you can modify its properties by specifying values in the **Properties** pane. For example, you can

- [change the prompt control interface](#)
- [require user input](#)
- [enable multiple selections](#)
- [show or hide prompt statuses](#)
- [specify default selections](#)
- [specify prompt values](#)
- [add prompt buttons](#)
- [create cascading prompts](#)

Some properties you set for a prompt may be overridden under some conditions. For example, some properties set for the filter associated with a prompt may override the corresponding prompt property.

Change the Prompt Control Interface

By default, when you create a prompt, IBM® Cognos® Report Studio selects the prompt control interface. You can change the prompt control interface depending on the type of prompt you created. For example, for a value prompt, you can choose a drop-down list, a list box, or a radio button group.

Steps

1. Click the prompt control.
2. In the **Properties** pane, set the **Select UI** property to the interface.

Specify That a Prompt Requires User Input

You can specify that a prompt requires user input before the report can run.

Steps

1. Click the prompt control.
2. In the **Properties** pane, set the **Required** property to Yes.
3. Pause the pointer over the page explorer button  and click a report page.
4. From the **Data** menu, click **Filters** .
5. Click the filter associated with the prompt.
6. In the **Usage** box, click **Required**.

When you run the report, a star appears next to the prompt indicating that the user must select or type a value.

If you have a [cascading prompt](#) and the parent prompt control is required, the child prompt control is disabled. This ensures that users choose at least one value in the parent prompt before they can choose a value in the child prompt. Conversely, if the parent control is optional, then the child control is populated. This gives users the ability to choose values in the child prompt without having to choose a value in the parent prompt.

Allow Users to Select Multiple Values in a Prompt

You can allow users to select more than one value in a prompt. For example, you have a prompt for which users must select a product line. You can modify the prompt so that users can select more than one product line.

If you enable multiple selections, the **Auto-Submit** property is always set to no.

Steps

1. Click the prompt control.
 2. In the **Properties** pane, choose whether to allow users to specify more than one value or a range of values:
 - To allow users to specify more than one value, set the **Multi-Select** property to Yes.
 - To allow users to specify a range of values, set the **Range** property to Yes.
 3. Pause the pointer over the page explorer button  and click a report page.
 4. From the **Data** menu, click **Filters** .
- If you have more than one query defined in the report, you must first click an object linked to a query.
5. Double-click the filter associated with the prompt.

6. Change the operator to one of the following:

- If you are creating a multi-select prompt, change the operator to `in`.

For example, `[Product_line] in ?Product line?` where `[Product_Line]` is the name of the data item allows users to select multiple product lines.

- If you are creating a range prompt, change the operator to `in_range`.

For example, `[Margin] in_range ?Margin?` where `[Margin]` is the name of the data item allows users to specify a margin range.

Show or Hide Prompt Status

Each prompt you create in a report provides dynamic validation when the report is run. Validity checks are performed to ensure that the data is correct and that required values are supplied. For example, a star appears next to each required prompt. An arrow appears next to a prompt if you must select or type a value. If you type an incorrect value, a dotted line appears. You can choose whether to show the star and arrow for each prompt.

Steps

1. Click the prompt control.
2. In the **Properties** pane, set the **Hide Adornments** property to **Yes** to hide the prompt characters or **No** to show them.

Specify a Default Selection for a Prompt

You can specify a default selection for a prompt so that users do not have to select or type a value when they run the report.

Steps

1. Click the prompt control.
2. To define a range of values, in the **Properties** pane, set the **Range** property to **Yes**.
3. To specify more than one default selection, in the **Properties** pane, set the **Multi-Select** property to **Yes**.
4. In the **Properties** pane, double-click the **Default Selections** property.
5. Click the add button  and do one of the following:
 - If you chose to define a single value, type the value as the default selection.
 - If you chose to define a range of values, type the minimum and maximum values of the range in the **Minimum value** and **Maximum value** boxes, respectively.
6. Repeat step 4 to specify other default selections.

Customize Prompt Text

You can customize the instructional text that appears around prompts. For example, a value prompt with multiple selections includes a **Select all** link below the choices that you can customize to text other than **Select all**. You can also specify translated text in many languages for international report users.

Steps

1. Click the prompt control.
 2. To change the default prompt text, in the **Properties** pane, double-click any of the properties under **Prompt Text**.
- Note:** When you select a property in the **Properties** pane, its description appears in the information pane below the **Properties** pane.
3. Click **Specified text**, and then click the ellipsis (...) button.
 4. In the **Localized Text** dialog box, type the text that you want to appear.
 5. To add customized text for other languages:
 - Click **Add**.
 - Select the countries for which you want to specify localized text and click **OK**.
 - To specify text for a country, select the country and click **Edit**, and then type the localized text in the text box.

Specify Prompt Values

Provide your own values in a prompt to

- show something different from what is in the database
- improve performance by not accessing the database
- provide text for optional prompts, such as **Select a value**
- restrict the number of values available

For example, you have a prompt in which users choose a country. For the database value United States, you want USA to appear in the prompt.

Tip: The Rolling and Moving Averages interactive sample report includes a value prompt. For more information about The Great Outdoors Company samples, see "["Sample Reports and Packages" \(p. 567\)](#)".

Steps

1. Click the prompt control.
2. In the **Properties** pane, double-click the **Static Choices** property.
3. Click the add button .

4. In the **Use** box, type the prompt value to add.
5. In the **Display** box, type the value that will appear in the prompt.
6. Repeat steps 3 to 5 to add other prompt values.
7. To link a prompt value to a condition, do the following:
 - In the **Variable** box, choose the variable to use or create your own.
For information about creating variables, see "[Add a Variable](#)" (p. 478).
 - In the **Value** box, click one of the possible values for the variable.
 - Click the static value to link to the variable and click the edit button .
 - In the **Display** box, type the value to appear in the prompt.
 - Repeat this procedure for each additional value.

Add a Prompt Button

Add prompt buttons so that users can submit selected items, cancel reports, or navigate between pages.

When you are building prompts and prompt pages, you may have to add prompt buttons to submit selections. Some prompt controls, such as the value prompt, can be set to submit selections automatically. Other prompt controls, such as the date prompt, require a prompt button.

Steps

1. Pause the pointer over the page explorer button  and click the page to which to add a prompt button.
2. In the **Insertable Objects** pane, on the **Toolbox** tab, drag **Prompt Button**  to the work area.
3. Click the prompt button and, in the **Properties** pane, set the **Type** property to one of the following actions.

Goal	Action
Cancel the report	Cancel
Go to the previous prompt page	Back
Go to the next prompt page	Next
Run the report	Finish
Reprompt the user	Reprompt
Tip: Reprompting users is useful when you have cascading prompts .	

Create a Cascading Prompt

Create a cascading prompt to use values from one prompt to filter values in another prompt. For example, a report contains the columns Product line and Product type. You create prompts for these columns, and you specify that the Product type prompt is a cascading prompt that uses Product line as the source. When users select a product line, they see only the product types related to the selected product line.

Tip: The Employee Training by Year sample report in the GO Data Warehouse (analysis) package includes cascading prompts. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Steps

1. To make the cascading source a required prompt, select it and, in the **Properties** pane, set the **Required** property to Yes.
2. Click the prompt control to use as a cascading prompt.
3. In the **Properties** pane, double-click the **Cascade Source** property.
4. Click the parameter that represents the cascade source.
5. If the prompt allows users to select **multiple values**, add a prompt button to the cascade source to provide the cascading prompt with the appropriate values:
 - In the **Insertable Objects** pane, on the **Toolbox** tab, drag **Prompt Button**  to the report.
 - Click the prompt button and, in the **Properties** pane, set the **Type** property to **Reprompt**.
 - To change the text in the prompt button, in the **Insertable Objects** pane, on the **Toolbox** tab, drag **Text Item**  to the prompt button and type the text.

For more information about how to create cascading prompts, see the Report Studio Quick Tour.

Control the Data That Appears in a Tree Prompt

You can control what data appears in a tree prompt and how the data is structured to get the results that you want. To do this, you add various functions to the filter expression.

In addition, the operator that you chose in the **Prompt Wizard** dialog box controls what appears next to each prompt value. If the operator is in or not in, check boxes appear next to each prompt value. If the operator is equals (=), no check boxes appear.

Steps

1. Pause the pointer over the query explorer button  and click the query that is associated with the prompt.
2. In the **Data Items** pane, double-click the data item on which you are prompting.
3. In the **Expression Definition** box, type one of the following functions.

Goal	Function
Show the hierarchical structure of all members in the hierarchy. If this function is not used, all members are shown in a flat list.	<code>rootmembers (data_item)</code>
Show the descendants of the data item in a hierarchical structure where x represents the level. For example, if you are prompting on the Year hierarchy and x=1, you will see 2004, 2005, and 2006 in the tree. If x=2, you will see 2004 Q1, 2004 Q2, and so on.	<code>descendants (rootmembers (data_item, x))</code>
Show the children of a member. For example, 2004 Q1, 2004 Q2, 2004 Q3, and 2004 Q4 appear for the member 2004.	<code>children (member)</code>

Using `in_range` Filters with Character Data

If you use an `in_range` filter with character data, and the **From** value is greater than the **To** value, the filter returns no results. For example, if the **From** value is "Zone" and the **To** value is "Aloe Relief", the report returns no data.

To allow results within a range regardless of whether the **From** value is greater than the **To** value, your IBM® Cognos® administrator can enable a prompting setting. For more information, see the *IBM Cognos Business Intelligence Administration and Security Guide*.

Chapter 13: Laying Out a Report

When creating a report, a good layout is essential to ensure that the information in the report is presented in a clear and effective manner.

Report Layout Recommendations

- ❑ Define the page structure.

Determine what goes into the page header, body, and footer. The page header contains information that appears at the top of each page. The page body contains information that starts on the first instance of the page. If there is too much data to fit on a single page, it continues across all instances of the page. The page footer contains information that appears at the bottom of each page.

- ❑ Identify horizontal bands of information.

Look for natural bands of information running across the page. Each of these bands typically translates into a block.

- ❑ Identify vertical bands of information.

In each horizontal band of information, look for bands that run up and down the page. Each of these bands typically translates into table cells.

- ❑ Decide which data frames to use to lay out the data.

Choose a list, crosstab, chart, repeater, or text frame.

- ❑ Set properties at the highest level item possible.

By setting properties at the highest level, you set them once instead of setting them for each child object. For example, if you want all text items in a report to appear in a specific font, set the font for the page.

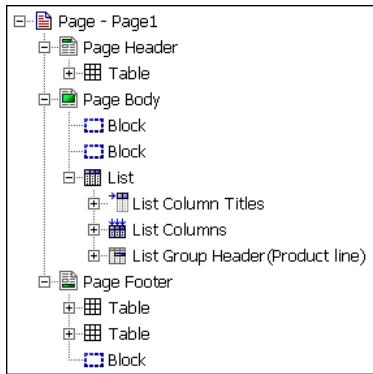
Tip: When setting properties for an object, click the select ancestor button  in the Properties pane title bar to see the different levels available.

- ❑ Use padding and margins to create white space.

Do not use fixed object sizing unless it is absolutely necessary. When you specify that an object has a fixed size, your layout becomes less flexible.

The Page Structure View

When you add objects to a report, you usually work in the layout. From the View menu, click **Page Structure** to view the report in a different way.



Use the page structure view

- to view the entire contents of a report page in a tree structure

Using a tree structure is useful for locating the objects in a page and troubleshooting problems with nested objects.

- to quickly move objects from one area of a page to another

If you have a complex layout, it may be difficult to select, cut, and paste objects in the layout view. Objects are easier to locate in the page structure view.

- to modify object properties

You can modify object properties in the layout or in the page structure view.

Tip: To switch back to the report layout, from the View menu, click **Page Design**.

Copy Object Formatting

You can quickly copy the formatting of items in your report, such as fonts, colors, borders, and number formats, and apply that formatting to other items.

Steps

1. Click an item that has the formatting to copy.
2. Do one of the following:
 - To copy all the formatting applied to the item, click the pick up style button , click the item to format, and then click the apply style button .
 - To copy only one of the formatting styles, click the down arrow to the right of the pick up style button and click the style to copy. Then click the item to format and click the apply style button.
3. If you want to make changes to a style that you copied, click the down arrow to the right of the pick up style button and click **Edit Dropper Style**.
4. In the **Style** dialog box, specify basic and advanced style characteristics.

Add a Header or Footer to a Report or List

Add a header or footer to make a report easier to read. Headers and footers are containers in which you can add objects like text, images, and report expressions such as the current date and page numbers. You can add headers and footers to pages and lists.

Tip: The Revenue by Product Brand (2005) sample report in the Sales and Marketing (cube) package includes a customized header and footer. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Pages

Add a page header or footer when you want information to appear on every page in the report, such as a title or page numbers. You can use combinations of text, data items, and report expressions in titles.

Lists

You can add the following headers and footers to lists to organize data into logical sections or to identify every change in value of a column.

Type	Description
List page header	Adds a header that appears at the top of the list on every page in which list data appears.
Overall header	Adds a header that appears once at the top of the list.
Group or section header	Adds a header that appears for each group of a grouped column or each section.
Group or section footer	Adds a footer that appears for each group of a grouped column or each section.
Overall footer	Adds a footer that appears once at the bottom of the list.
List page footer	Adds a footer that appears at the bottom of the list on every page in which list data appears. Note that summary calculations in list page footers only summarize the data that is visible on that page of the list report.

You can also add section footers by adding a summary.

To see the headers and footers, ensure that the visual aids are turned on.

Steps

1. If you want to add a page header or footer, from the **Structure** menu, click **Headers & Footers**, **Page Header & Footer**, select the appropriate check boxes, and click **OK**.

Tip: You can also click the headers & footers button  on the toolbar.

2. If you want to add a list header or footer, from the **Structure** menu, click **Headers & Footers**, **List Headers & Footers**, select the appropriate check boxes, and click **OK**.

Tip: You can also click the headers & footers button on the toolbar and then click the list headers & footers button .

3. If you want to add objects to a header or footer, drag the objects from the **Insertable Objects** pane to the appropriate location.

To add objects to a list header or footer, you must first unlock the report. From the **Structure** menu, click **Lock Page Objects**.

Tip: To add data items to the page header or footer, you must associate a query to the page.

4. To split a header or footer into multiple cells that span the list, from the **Structure** menu, click **Headers & Footers** and click **Split List Row Cell**.
5. To merge multiple cells in a header or footer, from the **Structure** menu, click **Headers & Footers** and click **Merge List Row Cells**.
6. To add rows to a header or footer, from the **Structure** menu, click **Headers & Footers** and click **Insert List Row Cells Above** or **Insert List Row Cells Below**.

Add Borders to an Object

You can add borders to objects in a report such as a column, a header, or a footer or to the whole report.

Steps

1. Click the object to which to add a border.

Tip: To quickly select the parent of an object, click the object, and then click the select ancestor button  in the title bar of the **Properties** pane.

2. In the **Properties** pane, double-click the **Border** property and select values in the **Color**, **Style**, and **Width** boxes.

Note: Specifying a width using % as the unit of measure is not supported when producing reports in PDF.

3. Apply the selected values to the top, bottom, left side, right side, or all sides of the object by clicking the appropriate button in the **Preview** section.

Tip: To clear the border applied, click the clear borders button .

Tip: You can also add borders using the IBM® Cognos® Report Studio toolbar.

Add a Text Item to a Report

You can add text to a report. You can insert text in other objects, such as a block or table cell, or directly in the report page.

You can also add multilingual text to a report.

Steps

1. In the **Insertable Objects** pane, click the **Toolbox** tab.
2. Drag the **Text Item** object  to the report.

The **Text** dialog box appears.

3. Type the text and click **OK**.

Tip: You can also paste text from another part of the report.

You can now format the text by changing the font, color, size, and so on. Select the text and make the appropriate changes in the **Properties** pane.

Specify the Font for a Report

You can specify the font for text in a report.

Steps

1. Click the object.

Tip: To specify the default font for the report, click the page.

2. Do one of the following:

- From the toolbar, specify the font properties.
- In the **Properties** pane, double-click the **Font** property and specify the font properties.

Tip: Type a list of fonts in the **Family** box if you are not sure whether a specific font is installed on a user's computer. For example, if you type **Times New Roman, Arial, monospace**, IBM® Cognos® Report Studio checks to see if Times New Roman is installed. If it is not, Report Studio checks for Arial. If Arial is not installed, the monospace font used by the computer is used.

If you clicked **(Default)** for any of the font properties, the default value for the property is used. Default values are stored in a style sheet that is used across all IBM Cognos Business Intelligence tools. You can modify default values by modifying classes.

Insert an Image in a Report

You can insert an image in a report. You can insert images in other objects, such as blocks or table cells, directly in the report page or as the background image of another object.

The images that you insert must first be uploaded to the IBM® Cognos® BI server or another Web server and must be .gif or .jpg format.

Steps

1. In the Insertable Objects pane, click the Toolbox tab .
2. Drag the **Image** object  to the report and then double-click it.
3. In the **Image URL** dialog box, type the URL of the image to insert or click **Browse** to go to the location containing the image.

To browse images on a Web server, you must enable Web-based Distributed Authoring and Versioning (WebDAV) on your Web server. For more information about configuring Web servers, see the *IBM Cognos Business Intelligence Installation and Configuration Guide*.

Insert a Background Image in an Object

You can insert a background image for objects in a report. For example, use a background image to add a watermark to a page.

You can also create your own background visual effects, such as drop shadows and gradient fills. For more information, see "[Add Background Effects](#)" (p. 443).

The images that you insert must first be uploaded to the IBM Cognos BI server or another Web server and must be .gif or .jpg format.

Steps

1. Select the object.
2. In the Properties pane, under **Color & Background**, double-click the **Background Image** property.
3. To use the background from the default style for the object (Cascading Style Sheet (CSS) class), click **Default**.
4. To explicitly not insert a background image, click **None**.

This option overrides any default background images including background gradients on objects that appear in the 10.x report styles.

5. To insert a specific image, click **Specified** and in the **Image URL** box, type the URL of the image to insert or click **Browse** to go to the location containing the image.

To browse images on a Web server, you must enable Web-based Distributed Authoring and Versioning (WebDAV) on your Web server. For more information about configuring Web servers, see the *IBM Cognos Business Intelligence Installation and Configuration Guide*.

6. In the **Position** box, choose how to align the image in the object.
7. In the **Tiling** box, click a tiling option.

Add Background Effects

You can add background effects to enhance the appearance of your report. You can enhance objects, such as data containers (lists, crosstabs, and charts), headers, footers, page bodies, and so on with borders, gradient fill effects, drop shadows, and background images. You can also apply background effects as a class style.

The background effect is rendered within the rectangle area that bounds the object. To use a generated background in a chart, the chart must have a fixed size.

For information about applying background effects to charts, see "[Add Background Effects to a Chart Object in a Legacy Chart](#)" (p. 158).

Notes

- If the background image is complex and large, the size of the report output may be affected proportionately.
- Background effects are rendered only if the data container has a fixed height and width; if a percentage size is used, the effects are ignored.
- Resizing or overflow behavior is ignored for generated images in HTML reports.

Steps to Add Background Effects to a Chart

1. Click the chart object.
2. To choose a preset background effect, click the background effects presets button  on the style toolbar, and then click a background.
3. In the Properties pane, under Color & Background, double-click the **Background Effects** property.
4. Select one or more of the following:
 - To apply a border, click **Border** and specify settings for border style, width, color, corner radius for rounded rectangles, and transparency.
If the element also includes a fill with a transparency setting, select the **Allow transparent bleed** check box to apply the same transparency to the border.
 - To apply a fill effect, click **Fill** and specify the settings. The fill effect can either be a solid color, a gradient, or a pattern. You can define a gradient fill effect as a linear, radial line, or radial rectangle gradient.
 - To apply a drop shadow effect, click **Drop Shadow** and specify the shadow color, transparency value, color, and offset settings. The default horizontal and vertical offset is 5 pixels.
 - To specify one or more images as a background, click **Images**. You can specify the transparency value and the position for each defined image. You can also specify a custom position for each image.

Tip: To remove the effect, clear its check box.

Steps to Add a Background Gradient to a Page

1. Click anywhere in the report page.
2. In the **Properties** pane title bar, click the select ancestor button  and click **Page**.
3. Under **Color & Background**, double-click the **Gradient** property.
4. Select the **Gradient** check box and then select the colors and direction for the gradient.

Background gradients and drop shadows do not appear in Microsoft® Excel spreadsheet software output. They are also supported only for report outputs run in the Microsoft Internet Explorer Web browser.

Tip: To remove the effect, clear the **Gradient** check box.

Steps to Add a Drop Shadow to a Container

1. Click the crosstab, list, repeater table, or table in the report.
2. In the **Properties** pane title bar, click the select ancestor button  and click **Crosstab, List, Repeater Table, or Table**.
3. Under **Color & Background**, double-click the **Drop Shadow** property.
4. Select the **Drop shadow** check box and then select the color, offset, and transparency values.

Background gradients and drop shadows do not appear in Excel output. They are also supported only for report outputs run in the Internet Explorer Web browsers.

If you add a drop shadow, ensure that you also specify a background color for the object. Otherwise, if you keep the default transparent background, data values also have a drop shadow and are difficult to read.

Tip: To remove the effect, clear the **Drop shadow** check box.

Insert a Formatting Object in a Report

In addition to text and images, the **Toolbox** tab  in the **Insertable Objects** pane contains other objects that you can add to the report layout.

Object	Description
Block	<p>Adds an empty block, which is a container in which you can insert other objects. This is useful for controlling where objects appear.</p> <p>Tip: You can use blocks to add space between objects. However, empty blocks are not rendered. You must insert an object or specify the height and width.</p> <p>Tip: The Revenue by GO Subsidiary 2005 sample report in the GO Data Warehouse (analysis) package includes an empty block. For more information about The Great Outdoors Company samples, see "Sample Reports and Packages" (p. 567).</p>
Table	<p>Adds a table, which is a container in which you can insert other objects. This is useful for controlling where objects appear.</p>
Field Set	<p>Adds an empty block that has a caption. This is similar to the Block object, but with a caption.</p>
Intersection (Tuple)	<p>Adds an intersection (tuple). For more information, see "Create an Intersection (Tuple)" (p. 400).</p>
Query Calculation	<p>Adds a calculated column. For more information, see "Using Relational Calculations" (p. 342) or "Using Dimensional Calculations" (p. 396).</p>
Crosstab Space	<p>Inserts an empty cell on a crosstab edge. Allows for the insertion of non-data cells on an edge. Blank cells appear for the edge when the report is run.</p> <p>Insert this object when a crosstab edge does not produce useful data and you want blanks to appear in the cells instead.</p>
Crosstab Space (with fact cells)	<p>Inserts an empty cell on a crosstab edge. Allows for the insertion of non-data cells on an edge. The contents of the fact cells for the edge are rendered when a measure is added or the default measure is specified.</p> <ul style="list-style-type: none"> • If the crosstab space is nested, the scope of the fact cells is the scope of the item that is at the level before the space. • If the crosstab space is not nested and there are no items nested below it, the scope of the fact cells is the default measure.
Singleton	<p>Inserts a single data item.</p>
Conditional Blocks	<p>Adds an empty block that you can use for conditional formatting.</p>

Object	Description
HTML Item	<p>Adds a container in which you can insert HTML code. HTML items can be anything that your browser will execute, including links, images, multimedia, tooltips, or JavaScript™.</p> <p>HTML items appear only when you run the report in HTML format.</p> <p>Note: You cannot include <form> tags in HTML items.</p> <p>Tip: The Table of Contents sample report in the GO Sales (analysis) package includes an HTML item. For more information about The Great Outdoors Company samples, see "Sample Reports and Packages" (p. 567).</p> <p>When you upgrade to the next version of IBM® Cognos® BI, the report upgrade processes do not account for the use of undocumented and unsupported mechanisms or features, such as JavaScript, that refers to IBM Cognos HTML objects.</p>
Rich Text Item	<p>Inserts an object that is used to render HTML in the layout. This object is similar to the HTML Item, except that rich text items also render in PDF output. Using rich text items is useful when you want to add annotations defined in a data source to a report.</p> <p>Note: Rich text items support only a restricted set of well-formed XHTML.</p>
Hyperlink	<p>Adds a hyperlink so that users can jump to another place, such as a Web site.</p>
Hyperlink Button	<p>Adds a hyperlink in the form of a button.</p>
As of Time Expression	<p>Adds an As of Time expression so that you can show data for a specific time period.</p>
Page Number	<p>Inserts page numbers that you can customize.</p>
Row Number	<p>Numbers each row of data returned when the report is run.</p> <p>Note: You can add row numbers only to lists and repeaters.</p>
Layout Component Reference	<p>Adds a reference to another object. Useful when you want to reuse an object.</p>
Table of Contents	<p>Creates a table of contents that is rendered in the output of a report.</p>

Object	Description
Table of Contents Entry	Adds a table of contents marker.
Bookmark	Inserts a bookmark so that users can move from one part of a report to another.

For information about each prompt control and prompt buttons, see "[Adding Prompts to Filter Data](#)" ([p. 421](#)). The **Toolbox** tab contains other objects when you work in other areas such as Query Explorer and Condition Explorer. You can also add your own objects to the toolbox, such as a company logo that you use often. For more information, see the IBM Cognos Business Intelligence *Administration and Security Guide*.

Before you can add a hyperlink, HTML item, or hyperlink button, you must have the **HTML Items in Report** capability. For more information, see the IBM Cognos Business Intelligence *Administration and Security Guide*.

Step

- To add an object, drag or double-click it from the **Toolbox** tab .

Elements Supported in Rich Text Items

Rich text items support all XHTML character entities, such as (non-breaking space), and the following elements:

- div
- span
- ul
- ol
- li

Each element only supports the style attribute, which must contain a valid CSS style. In addition, ul and ol elements support list-style attributes. Specifically, the ol element supports decimal, and the ul element supports circle, disc, and square, as well as list-style-image.

For example, the following code produces an unordered list entitled List: with three items. Each list item is in a different color, and the list-style attribute used is circle.

```
<div style="font-size:14pt; text-decoration:underline">List:</div>
<ul style="list-style-type:circle">
<li style="color:green">Item <span style="font-weight:bold">A</span></li>
<li style="color:red">Item B</li>
<li style="color:blue">Item C</li>
</ul>
```

Example - Add a Multimedia File to a Report

You are a report author at The Great Outdoors Company, which sells sporting equipment. You want to insert a Microsoft® Windows® Media Audio/Video file named GO.wmv in a template that serves as a cover page for all reports.

You must have Windows Media Player installed on your computer.

Steps

1. Open IBM Cognos Report Studio with the **GO Data Warehouse (query)** package.
2. In the **Welcome** dialog box, click **Create a new report or template** and in the **New** dialog box, click **Blank**.
3. In the **Insertable Objects** pane, on the **Toolbox** tab, drag the **HTML Item** object  to the report.
4. Select the **HTML Item**.
5. In the **Properties** pane, double-click the **HTML** property.
6. In the **HTML** dialog box, type the following:

```
<OBJECT classid="CLSID:6BF52A52-394A-11D3-B153-00C04F79FAA6"><PARAM NAME="URL" VALUE="/c10/webcontent/samples/images/GO.wmv"/></OBJECT>
```

When you run the report in HTML format, the multimedia file plays in Windows Media Player.

Align an Object in a Report

You can specify a horizontal and vertical alignment for an object in a report to determine where they appear.

[Tables](#) can also be used to determine where objects appear in a report.

Note: The **Justify** horizontal alignment works with HTML output but does not apply to PDF output.

Steps

1. Select the object to align.
2. From the toolbar, click one of the available horizontal or vertical alignment buttons.

Use a Table to Control Where Objects Appear

You can use tables in your report to control where objects appear. Tables can be inserted anywhere in a report, such as a header, a footer, or the page body. After you create a table, insert objects in the cells.

You can also apply a predefined [table style](#) to tables.

The [alignment buttons](#) can also be used to determine where objects appear in a report.

Tip: The Singletons on Page Body sample report in the GO Sales (analysis) package includes a table to control the report layout. For more information about The Great Outdoors Company samples, see "Sample Reports and Packages" (p. 567).

Steps to Create a Table

1. In the **Insertable Objects** pane, click the **Toolbox** tab.
2. Drag **Table**  to the report.
The **Insert Table** dialog box appears.
3. In the **Number of columns** and **Number of rows** boxes, type the number of columns and rows for the table.
4. If you want to have the table span the width of the report page, select the **Maximize width** check box.
5. If you want to add borders to the table, select the **Show borders** check box.
6. If you want to merge cells in the table, select the cells and click the merge cells  button.

Steps to Customize the Table Borders and Size

1. Select the table object.
2. In the **Properties** pane, under **Positioning**, double-click the **Table Properties** property.
3. To display both the inside and outside borders, do the following:
 - Clear the **Collapse borders** check box.
 - Type a number in the **Cell spacing** text box to define how much space to add between the table cells.
 - Select the **Show empty cell borders** check box to display the borders even for empty cells.
4. To fix the size of the columns in the table, check the **Fixed size** check box.

When this check box is cleared, the table columns expand to fit the text.

Other table properties are described in "Report Studio Object and Property Reference" (p. 601).

Apply a Table Style

Apply a table style to quickly format tables. You can also apply a table style to lists, crosstabs, and repeater tables.

Steps

1. Click the table, list, or crosstab.
2. From the **Table** menu, click **Apply Table Style**.
3. In the **Table styles** box, click a table style.

Tip: Some styles are unique to tables, lists, or crosstabs.

4. If you are applying a table style to a table or list, in the **Apply special styles to** section, select or clear the various column and row check boxes based on how you want to treat the first and last columns and rows.

Some check boxes may not be available for particular table styles, or to particular columns or rows.

5. If you are applying a table style to a list or crosstab, select the **Set this style as the default for this report** check box to set the style as the default for all lists and crosstabs.

For a list, you may need to clear the **First column** and **Last column** check boxes in the **Apply special styles to** section before you can select this check box. In addition, some table styles cannot be set as the default.

Apply Padding to an Object

Apply padding to an object to add white space between the object and its margin or, if there is a border, between the object and its border.

Tip: The GO Balance Sheet as at Dec 31 2006 sample report in the GO Data Warehouse (analysis) package includes padding. For more information about The Great Outdoors Company samples, see "["Sample Reports and Packages" \(p. 567\)](#)".

Steps

1. Select the object to which to apply padding.
2. In the **Properties** pane, double-click the **Padding** property.
3. Specify top, bottom, left, and right padding by typing values in the corresponding boxes and choosing the unit of measure.

Note: Specifying padding using % as the unit of measure is not supported when producing reports in PDF.

Tip: You can quickly apply left or right padding by either pressing Tab and Shift+Tab or by clicking the increase indent  and the decrease indent buttons in the toolbar. When using the toolbar buttons, you can indent an object by up to nine times the indentation length. You can specify the indentation length to use by clicking the arrow beside either button and clicking **Set Default Indent Length**.

Set Margins for an Object

Set the margins for objects in a report to add white space around them.

For Date, Time, Row Number, and Page Number objects, you can only set the left and right margins. If you want to set the top or bottom margins for these objects, place them in a table or a block. Then set the margin or padding properties on the table or block object.

Steps

1. Select the object.

2. In the **Properties** pane, double-click the **Margin** property.
3. Specify the top, bottom, left, and right margins by typing values in the corresponding boxes and choosing the unit of measure.

Note: Specifying margins using % as the unit of measure is not supported when producing reports in PDF.

Add Multiple Items to a Single Column

You can add multiple items to a single column to condense a report. For example, you have a list report that contains many columns. You can reduce the number of columns in the list by putting related information in a single column.

Steps

1. From the **Structure** menu, click **Lock Page Objects** to unlock the report.
2. In the **Insertable Objects** pane, drag the items to the column.

For example, you can add data items from the **Source** tab  or text items from the **Toolbox** tab .

Example - Create a Report with Multiple Items in One Column

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a list report showing the name, position, email address, phone number, extension, and fax number for each sales representative in each city organized by country. To reduce the number of columns in the list, you show email addresses, phone numbers, extensions, and fax numbers in a single column.

Steps

1. Open IBM® Cognos® Report Studio with the **GO Data Warehouse (query)** package.
2. In the **Welcome** dialog box, click **Create a new report or template**.
3. In the **New** dialog box, click **List** and click **OK**.
4. In the **Insertable Objects** pane, on the **Source** tab , expand **HR (query)**, **Employee summary (query)**, and **Employee by region**. Add the following data items to the list by double-clicking them:
 - **Country**
 - **City**
 - **Employee name**
 - **Position name**
 - **Email**

- Work phone
 - Extension
 - Fax
5. Click the **Country** column and, from the **Structure** menu, click **Section** .
 6. From the **Structure** menu, ensure that the report is unlocked
 7. Click **Country** and, in the **Properties** pane, double-click the **Font** property.
 8. Change the font to **Arial Black, 11 pt, and Bold**, and then click **OK**.
 9. In the **Insertable Objects** pane, on the **Toolbox** tab , drag **Table** to the right of the **Work phone** text item in the first row of **Work phone** column and create a table that has one column and three rows.
 10. Drag the following data items to the table:
 - **Email** to the first row
 - **Work phone** to the second row
 - **Fax** to the third row
 11. Drag **Extension** to the right of **Work phone** in the table.
 12. In the **Insertable Objects** pane, on the **Toolbox** tab, drag **Text Item** to the left of each item in the table and type the following for each item, putting a blank space before and after the text:
 - **Email:**
 - **Work phone:**
 - **ext:**
 - **Fax:**
 13. Ctrl+click the **Extension**, **Email**, and **Fax** column titles and click the delete button .
 14. Click the **Work phone** column title.
 15. In the **Properties** pane, click the **Source Type** property and click **Text**.
 16. Double-click the **Text** property.
 17. Type the following and click **OK**:

Contact Information
 18. Select the **Work phone** data item in the first row of the list above the table and click the delete button.
 19. Double-click the text item in the page header, type the following, and click **OK**:

Sales Representatives Contact List

20. From the **Structure** menu, click **Lock Page Objects**.

The report is locked.

21. Run the report.

Contact information for each sales representative appears in a single column.

<u>Sales Representative Contact List</u>			
Australia			
City	Employee name	Position name	Contact Information
Melbourne	Alice Walter	Branch Sales Manager	Email: AWalter@grtd123.com Work phone: +(61) 03 2982 4242 ext: 8910 Fax: +(61) 03 2982 4243
Melbourne	Alice Walter	Level 3 Sales Representative	Email: AWalter@grtd123.com Work phone: +(61) 03 2982 4242 ext: 8910 Fax: +(61) 03 2982 4243
Melbourne	Andrea Samuel	Payroll Clerk	Email: ASamuel@grtd123.com Work phone: +(61) 03 2982 4242 ext: 8224 Fax: +(61) 03 2982 4243
Melbourne	Catherine Fowlie	Warehouse Worker	Email: CFowlie@grtd123.com Work phone: +(61) 03 2982 4242 ext: 8465 Fax: +(61) 03 2982 4243
Melbourne	Cindy Sandles	Product Technician	Email: CSandles@grtd123.com Work phone: +(61) 03 2982 4242 ext: 8225 Fax: +(61) 03 2982 4243
Melbourne	Dan Michaels	Budget Analyst	Email: DMichaels@grtd123.com Work phone: +(61) 03 2982 4242 ext: 8480 Fax: +(61) 03 2982 4243
Melbourne	Dave Smythe	Level 1 Sales Representative	Email: DSmythe@grtd123.com Work phone: +(61) 03 2982 4242 ext: 8909 Fax: +(61) 03 2982 4243
Melbourne	David Dolby	District Sales Manager	Email: DDolby@grtd123.com Work phone: +(61) 03 2982 4242 ext: 8288 Fax: +(61) 03 2982 4243

Reuse a Layout Object

You can save time by reusing layout objects that you add to a report instead of re-creating them. For example, you have a multiple-page report and you want to show the company logo in the page header of each page. Insert the logo once and reuse it on all other pages.

Steps

1. Click the object to reuse.

Tip: To quickly select the parent of an object, click the object, and then click the select ancestor button  in the title bar of the **Properties** pane.

2. In the **Properties** pane, set the **Name** property to a value beginning with a letter to uniquely identify the object and press the Enter key.

IBM® Cognos® Report Studio may have already specified a name for the object.

3. To reuse the object in another report, open that report.

4. In the **Insertable Objects** pane, on the **Toolbox** tab , drag the **Layout Component Reference** object to the location in which it will be reused.
5. In the **Component Location** box, do one of the following:
 - To reference an object in the current report, click **This report**.
 - To reference an object in another report, click **Another report**, click the ellipsis (...) button, and open the report.
6. In the **Available components to reference** box, click the object and click **OK**.
7. If the referenced object is in another report, click the **Layout Component Reference** object and in the **Properties** pane, set the **Embed** property to specify how to store the referenced object in the report:
 - Click **Copy** to store a copy of the object.
The copy is not automatically **updated** if the source object is modified.
 - Click **Reference** to store a reference, or pointer, of the object.
The reference of the object is automatically updated if the source object is modified. For example, if you open or run the report, you see the modified object.

A copy or reference of the object appears where you placed the **Layout Component Reference** object. If a source object is changed, you can **update reused objects**.

Tip: You can also create a new report or template and add all the objects to share. All your shared objects then reside in a single location, like a library.

Change a Reused Object

If you reuse an object that contains other objects, you can change the child objects to something different. For example, you have a **block** object containing a text item in the page header and you decide to reuse the block in the page footer. However, you want the text item in the page footer block to show different text from that in the page header.

Steps

1. In the parent object to reuse, click the child object to change.
2. In the **Properties** pane, set the **Name** property to a value beginning with a letter to uniquely identify the object.
Report Studio may have already specified a name for the object.
3. Select the copy of the parent object you created with the **Layout Component Reference** object.
4. In the **Properties** pane, double-click the **Overrides** property.
The child object in the copy of the parent object is replaced by the following text:
Drop item to override component child.
5. In the **Overrides** dialog box, select the child object to change and click **OK**.

6. Drag an object to replace the child object.

You can replace the child object with any other object, not just an object of the same type. For example, if the child object is a text item, you can replace it with an image.

Update Reused Objects

If a report contains objects referenced in another report, you can quickly update the referenced objects if the source objects have changed. Shared objects are stored in the layout component cache.

Steps

1. Open the report to update.

2. From the Tools menu, click **Layout Component Cache**.

The **Layout Component Cache** dialog box appears, showing all reports that contain referenced objects and the source reports where the objects exist.

3. To view which components are reused, click a source report.

The components that are reused appear in the **Components used** pane.

4. Click **Reload Components** to refresh all referenced objects.

Although referenced objects are automatically refreshed when you open or run a report, clicking this button updates components that were changed while the report is open.

5. Click **Update All Component Copies** to refresh all copied objects.

6. Click **Close**.

Specify Text Flow in an Object

You can specify text flow properties by choosing any of these options.

Option	Description
Direction	Sets the reading order of an object, such as right to left.
Writing mode	Sets the direction and flow of content in an object.
Bi-directional	Sets the level of embedding in an object.
Type	Sets the type of alignment used to justify text in an object.
Kashida space	Sets the ratio of kashida expansion to white space expansion when justifying lines of text in the object. This property is used in Arabic writing systems.

Note: All the text flow properties work with HTML output but do not apply to PDF output.

Steps

1. Click an object.
2. In the Properties pane, double-click the **Text Flow & Justification** property.
3. Specify the text flow options.

Specify Line Spacing and Breaking

You can specify text properties by choosing any of these options.

Option	Description
Line Height	Sets the distance between lines of text in an object.
Letter Spacing	Sets the amount of additional space between letters in an object.
Text Indent	Sets the indentation of the first line of text in an object. Note: This property works with HTML output but does not apply to PDF output.
Word Break	Sets line-breaking behavior within words.
Break words when necessary	Sets whether to break words when the content exceeds the boundaries of an object.
Enforce stricter line-breaking rules for Japanese Text	Sets line-breaking rules for Japanese text.

Steps

1. Click an object.
2. In the Properties pane, double-click the **Spacing & Breaking** property.
3. Specify the text properties.

Specify the Height and Width of an Object

You can specify the height and width of objects using various units of measurement. In addition, if the object is a **field set**, **text box prompt**, **prompt button**, **hyperlink button**, **block**, or a **conditional block**, you can specify how to handle content overflow. Specify the height and width by choosing any of these options.

Option	Description
Height	Sets the height of the object.
Width	Sets the width of the object.
Content is not clipped	If the contents of the block exceed the height or width of the block, the block automatically resizes to fit the contents.
Content is clipped	If the contents of the block exceed the height or width of the block, the content is clipped. Note: The clipped content still exists. It is just not visible in the block.
Use scrollbars only when necessary	If the contents of the block exceed the height or width of the block, scrollbars are added to the block.
Always use scrollbars	Scrollbars are added to the block.

Do not use percentages to resize charts and maps that contain interactive elements that are activated when you pause the pointer over them, such as tooltips or drill-through links, because the browser is unable to realign the hard-coded hot spots after an image is resized.

When you use a percentage to specify the size of an object, the percentage is relative to the object's parent. In some cases, setting the size of an object using percentages will not give you the results that you want unless you also specify the size of the parent container.

Steps

1. Click an object.
2. In the Properties pane, double-click the Size & Overflow property and specify the height and width.

Control How Other Objects Flow Around an Object

You can control how objects flow around other objects by choosing any of the following options.

Option	Description
Float	Sets how other objects flow around the object.
Allow floating objects on both sides	Allows other objects to flow on both sides. The Float property must be set.

Option	Description
Move below any floating object on the left side	If there are other objects to the left of the object, the object moves below those objects. The Float property must be set.
Move below any floating object on the right side	If there are other objects to the right of the object, the object moves under those objects. The Float property must be set.
Move below any floating object	Moves the object under any other object in which the Float property was set.

Steps

1. Click an object.
2. In the **Properties** pane, double-click the **Floating** property.
3. Specify how other objects flow around the object by specifying the floating options.

Specify Report Properties

You can change the default report property settings.

Steps

1. From the **File** menu, click **Report Properties**.
2. To specify the classes that IBM® Cognos® Report Studio uses to format objects, click **Report styles** and choose one of the available options.
For more information, see "[Create and Modify Report and Object Styles](#)" (p. 459).
3. For reports with multiple data containers, to render the default number of rows of each data container on each HTML page, you must set the **Page break by data container for interactive HTML** option to **Yes**. The default value is **No**.
For more information, see "[Controlling the Rows Per Page for Multiple Containers in HTML and PDF](#)" (p. 98).
4. To automatically create extended data items every time a data item is inserted, select the **Always create extended data items** check box.
Report Studio creates extended data items by default. For more information about extended data items, see "[Extended Data Items](#)" (p. 356).
5. To create a single, scrollable HTML page when you run and save this report as HTML from IBM Cognos Connection, clear the **Paginate saved HTML output** check box.
The default is to paginate HTML reports with the same page breaks as PDF reports. The single, scrollable HTML page will be available from IBM Cognos Connection when you click the view output versions for this report button .

6. If you are working with reports created in IBM Cognos ReportNet® and you want to create CSV report output, click **Use 1.x CSV export**.

This option ensures that all the query columns are exported. In IBM Cognos ReportNet, if a data item was referenced using the **Properties** property of a list, it was included in the CSV output. In IBM Cognos BI, the default is to export only the columns in the list.

Create and Modify Report and Object Styles

Create your own classes or modify existing classes in a report to format objects across a report according to your particular needs. In IBM® Cognos® Report Studio, objects in reports are assigned a Cascading Style Sheet (CSS) class that provides a default style for the object. For example, when you create a new report, the report title has the class property **Report title text** assigned to it. In addition, objects inherit the classes set on their parent objects.

You can use classes to highlight data using conditional styles.

Classes you create or modify can be applied only to the current report. To create or modify classes for all reports, you must [modify a layout style sheet](#). In addition, some classes can be used to [format Query Studio reports](#).

If you use the Microsoft® Internet Explorer Web browser, version 6, some color gradients used in the 10.x default report styles are not supported.

Steps to Modify the Report Style

1. From the File menu, click **Report Properties**.
2. Click **Report styles** and select one of the following options:
 - To work with classes in the default style sheet, click **10.x styles**.
Use 8.x report styles when you are working with reports created in IBM Cognos 8 and you want to preserve their original appearance.
 - To work with classes that were used in IBM Cognos ReportNet®, click **1.x styles**.
Use 1.x report styles when you are working with reports created in ReportNet and you want to preserve their original appearance.
 - To work with classes that have minimal styling defined, click **Simplified styles**.
This option is useful when creating financial reports.

Steps to Modify Object Styles

1. Pause the pointer over the page explorer button  and click **Classes**.
2. To create a new class, in the **Insertable Objects** pane, drag **Class** to the **Local Classes** pane.
3. To modify an existing class, in the **Local Classes** or **Global Class Extensions** pane, click the class.

Modify a global class to apply a change to all objects that use that class. For example, if you modified the style **List column title cell**, all column titles in lists will reflect your modifications.

Tip: Ctrl+click classes to make the same change to more than one class.

4. In the **Properties** pane, modify the properties to specify your desired formatting.

Tip: Look at the **Preview** pane to preview your changes for different report objects, such as blocks, table cells, and text items.

If you modify a global class, a pencil symbol appears beside the global class icon to indicate that the class was modified.

5. Apply the class to objects:

- Pause the pointer over the page explorer button and click a report page.
- Click an object to which to apply a class.
- In the **Properties** pane, double-click the **Class** property.
- Click the classes to apply from the **Local classes** and **Global classes** panes and click the right arrow button .
- If you applied more than one class, in the **Selected classes** pane, specify the order in which the classes are applied using the up and down arrow buttons.

Classes in the **Selected classes** pane are applied from top to bottom. The style properties from all classes are merged together when they are applied. However, if the classes have style properties in common, the properties from the last class applied override those from previous classes.

(Don't Print) Class

The **(Don't Print)** class allows HTML items to display in the Web browser but not to print. The **(Don't Print)** class behaves as follows in the various report output formats:

- **HTML**

The Web browser defines the behavior. The HTML standard is that the item appears on the screen in the Web browser but is excluded by the print operation of the Web browser.

- **PDF**

The item is excluded from the output.

- **Microsoft Excel 2002 spreadsheet software**

The class is specified in the HTML that IBM Cognos uses to render Microsoft Excel 2002 output. However, Microsoft Excel 2002 does not appear to honor it and displays the item as missing some or all other style definitions.

- **Microsoft Excel 2007 spreadsheet software**

The item is excluded from the output.

- **CSV**

The item is included in the output.

- XML

The item is included in the output.

An alternative way to consistently exclude a report object from HTML, PDF and Microsoft Excel output is to set the **Box Type** property for the object to **None**. CSV and XML report outputs will still contain the object.

Modify Classes to Format Query Studio Reports

Some global classes are specific to IBM Cognos Query Studio or can be applied to Query Studio reports. You can modify the following classes to format Query Studio reports.

Class	Description
List column body cell	Style used to format list data.
List column title cell	Style used to format list column titles.

Steps

1. Create a Query Studio template.
2. Modify the global classes.
3. Save the template.

To format a Query Studio report using the modified classes, the template must be applied to the report. For more information about applying a template to a Query Studio report, see the *Query Studio User Guide*.

Modifying the Default Layout Style Sheet

In addition to creating or modifying classes in a report, you can create and modify classes that will apply to all reports. Default styles are stored in a style sheet named `GlobalReportStyles.css`. For information about modifying the style sheet, see the *IBM Cognos Business Intelligence Administration and Security Guide*.

Add Color to an Object

You can add background and foreground color to objects in the report. The foreground color applies to the text within objects.

Tip: The TOC Report sample report in the GO Data Warehouse (query) package includes objects with color. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Steps

1. Select the object.

Tip: To quickly select the parent of an object, click the object, and then click the select ancestor button  in the title bar of the **Properties** pane.

2. Do one of the following:
 - Click the background color button  or the foreground color button .
 - In the **Properties** pane, double-click the **Background Color** or **Foreground Color** property.
3. To apply an existing color, click the **Named Colors** tab or **Web Safe Colors** tab and choose one of the available colors.

Use a named color to select a color from a small set of colors. Use Web safe colors to select from 216 available colors.
4. To apply a custom color, click the **Custom Color** tab and type values in the **Red**, **Green**, and **Blue** boxes.

The values must be hexadecimal.

Chapter 14: Working With Your External Data

You can supplement your enterprise data with your own external or personal data file. This allows you to create a report using an external file that contains data which is not part of your IBM® Cognos® Business Intelligence enterprise data. For example, you are an analyst and you receive a spreadsheet that contains what-if data about opening new branches of your retail store. You are asked to analyze the impact of these possible new branches on existing sales volumes. You link this scenario data with your enterprise data and create a professional report using IBM Cognos BI.

You import your own data file and start reporting on it right away. After importing, your external data file is protected by the same IBM Cognos security as your enterprise data, thus allowing you to report on your data in a secure and private environment.

You can use the following types of files:

- Microsoft® Excel (.xls) spreadsheet software files

The supported versions are the same versions supported for report output in IBM Cognos BI. For an up-to-date list of environments supported by IBM Cognos products, such as operating systems, patches, browsers, Web servers, directory servers, database servers, and application servers, visit www.ibm.com.

- tab-delimited text (.txt) files
- comma-separated (.csv) files
- XML (*.xml) files

Your XML files must adhere to the IBM Cognos schema specified at *c10_location/bin/xmldata.xsd*. For more information, contact your IBM Cognos administrator.

This schema consists of a dataset element, which contains a metadata element and a data element. The metadata element contains the data item information in item elements. The data element contains all the row and value elements.

For example, the following simple XML code produces a table with two columns (Product Number and Color) and two rows of data.

```
<?xml version="1.0" ?> <dataset xmlns="http://developer.cognos.com/schemas/xmldata/1/" xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"> <metadata> <item name="Product Number" type="xs:string" length="6" scale="0" precision="2" /> <item name="Color" type="xs:string" length="18" scale="0" precision="8" /> </metadata> <data> <row> <value>1</value> <value>Red</value> </row> <row> <value>2</value> <value>Blue</value> </row> </data> </dataset>
```

To work with your external data, you start with an existing IBM Cognos package. You import data from your external file into the package and create links between data items in your file and data items in your enterprise data source. You then publish a new package that allows you to create reports that use your data and your enterprise data, or reports that use only your data. You can link your external data with both dimensional and relational data sources.

Before you can import your own external data file, your IBM Cognos administrator must grant you permission for the **Allow External Data** capability found within the **Report Studio** capability, and you must have permission to use IBM Cognos Report Studio or IBM Cognos Business Insight Advanced. For more information, see the *IBM Cognos Business Intelligence Administration and Security Guide*.

External Data Packages

When you import external data into a package, you do not overwrite the original package. You create a new package that includes the original package, the new external data, and any links or relationships that you defined between the two. By default, the new package is saved in the **My Folders** area of IBM Cognos Connection appends **External Data** to the original package name.

You can change where your package is published.

IBM Cognos Samples

Sample external data sources in Microsoft Excel (.xls) format are provided with IBM Cognos BI. You can import these sample files into the Great Outdoors Company sample reports found within the **Business Insight Advanced Report Samples** folder of the **GO Data Warehouse (analysis)** or **GO Data Warehouse (query)** package.

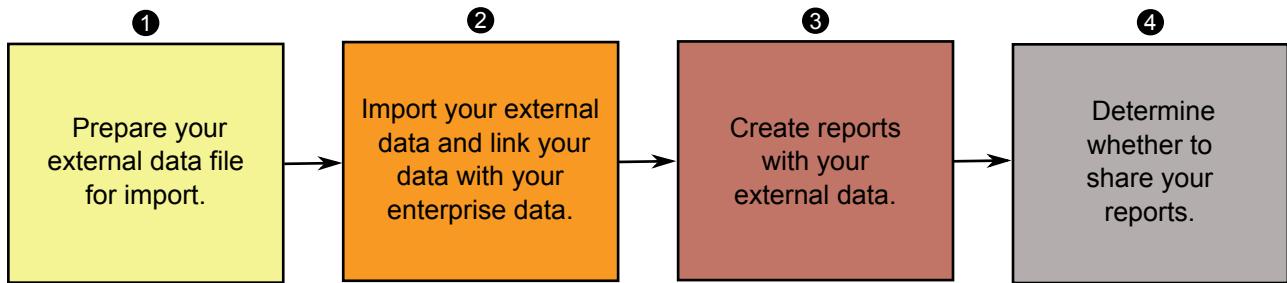
You can find the following external data source files on the server where IBM Cognos BI is installed in the *c10_location/webcontent/samples/db/other* directory.

- accounts.xls
- employee.xls
- organization.xls
- product_brand.xls
- product_color.xls
- product_line.xls
- product_name.xls
- product_size.xls
- product_type.xls
- products.xls
- promo_sets.xls
- promotions.xls
- region.xls
- retailers_site.xls
- time.xls

To obtain these files, contact your IBM Cognos administrator.

Preparing to Work with your External Data

To work with your external data, follow this process.



1. Prepare your external data file for import.

Ensure that your external data file matches your enterprise data for your reporting needs. For example, if your external data file contains sales values by month, ensure that the formatting of months in your file matches the formatting used in your enterprise data source. Ensure that you can uniquely link at least one column from your external data file, such as product codes or years, with your enterprise data source.

The maximum file size that you can import is 2.5 MB, with a maximum of 20000 rows. You can import a maximum of one external data file per package. Your IBM® Cognos® modeler can override these governors in IBM Cognos Framework Manager.

2. Import your external data.

You import your external data file from your own computer or from a location on your network into an existing IBM Cognos package.

A step-by-step wizard guides you through importing your data. If you want to create reports that contain data from both your external data file and your enterprise data source, you must link data items from the two data sources. You can import all or a subset of data columns from your external file.

By adding external data, you extend an existing IBM Cognos package definition with the new data items from your file and you create a new package.

3. Create reports with your external data file.

After you import and link your external data, it appears as a new namespace in the data tree of the **Source** tab in the **Insertable Objects** pane and is integrated with the IBM Cognos content.

You can then create reports with your data and perform any operation, such as filtering, sorting, grouping, or adding calculations. When you run the report, it uses data items from your external data file.

You can save reports that contain your external data within the **My Folders** area of the IBM Cognos portal.

4. Determine whether you want to share your reports that use external data with other people in your organization.

If you decide to share, take into account these considerations.

Working with Date Data

If the data that you import contains dates, ensure that the dates use the format yyyy-mm-dd.

Working with Dimensional Data Sources

If your enterprise data source is dimensional, such as OLAP or dimensionally-modeled relational, and you want to link your external data with your enterprise data, you must first create a tabular or list report. Create and save a list report with your enterprise data that contains the data items that you want to use to link with your external data. Ensure that you remove the aggregate rows that are automatically added in the footer of the list. This list is a projection of your dimensional data source.

When you import your external data, use the list report that you created to link your external data with the query subject from your enterprise data.

Data in your external file is relational by nature because it consists of tables and rows. If your enterprise data source is dimensional, you can still import and work with your external data. However, you cannot mix relational data from your external data files, and dimensional data from your enterprise data source within the same query. For example, a data container, such as a list, crosstab, or chart, uses one query and you cannot mix relational and dimensional data within the same list, crosstab, or chart. Doing so will cause an error.

If you want to use data from both the external data file and the original package within the same query, you must link the external data to a query subject within the current package instead of another report.

Working with External Data in an Unsecured IBM Cognos Application

If your IBM Cognos application is not secured, and users can log on anonymously, you may encounter issues if multiple people import external data in the same package.

For example, Robert imports his external data into package A and saves the package and reports he created in **My Folders**. Then, Valerie also imports her external data into the same package A and saves the package in **My Folders**. Valerie has therefore overwritten Robert's external data in package A. Now, if Robert tries to run one of his reports, he encounters errors because his external data is no longer in package A.

To avoid this problem,

- save packages that contain external data with a unique name.
- apply security to your IBM Cognos applications so that users do not share the same **My Folders** area.

Import Your External Data

A wizard walks you through the following steps to import your external data.

Selecting Your Data

You select the file to import from your own computer, or from your local area network.

You select which columns to import.

You then specify a namespace to use. The namespace provides a unique name to associate with the data items that you import. The namespace appears in the data tree of the **Source** tab in the **Insertable Objects** and is used to organize the data items. By default, the namespace is the imported file name without the extension.

If you change the default name for the namespace, you are prompted to select the external data file each time you run the report. To avoid this, select the **Allow server to automatically load file** check box.

Mapping Your Data to Enterprise Data

If you want to create reports that contain data from both your external file and from your enterprise data, you must link at least one query subject from your package or from an existing report to a data item in your external data. This mapping creates a relationships between your external data

and your enterprise data. For example, your external data contains information about employees, including an employee number. You map the employee number from your external data file to the employee number in your enterprise data. This ensures that your data is integrated smoothly.

Specifying Data Attributes

You can change how query items from your external data file appear after they are imported into IBM® Cognos® BI. For example, you can change the number of decimal places or the default summary.

If you want to use numeric data from your external data source as a measure in a crosstab, you must assign that data item a default summary other than **Unsupported**. A specified default summary makes the data item appear as a measure in the **Insertable Objects** tree. Otherwise, if you add the data item with an **Unsupported** default summary as the measure in a crosstab, no values appear.

Attribute	Description
Data type	<p>Identifies whether data in the file is one of the following type:</p> <ul style="list-style-type: none"> Integer, which represents numeric values. Decimal, which represents integer values that are scaled by a variable power of 10. Text, which represents values that contain letters and symbols. Date or Date Time, which represent dates and times.
Default summary	<p>Identifies Sum, Average, Max, Min, Count or Unsupported as the default type of summary for the data item.</p> <p>Applies only to Integer and Decimal data types.</p>
Decimal places	<p>Specifies the number of decimal places for the data item.</p> <p>Applies only to the Decimal data type.</p>

Specify the Mapping Options

If you mapped links between data items in your external data and data items in your enterprise data, specify the options that define the relationships between the data items.

For each data item that you import and link, specify whether values are unique or exist more than once in both your external data and in your enterprise data. You can also specify how to handle rows that contain missing values in the report results.

Steps to Import Data

- From the Tools menu, click **Manage External Data**.

Tip: You can also click the manage external data button  at the top of the **Source** tab in the **Insertable Objects** pane.

2. On the **Select Data** page of the wizard, under **External data file**, click **Browse** and select your external data file to import.

If you want the server to load the file without prompting users when they run the report, select the **Allow the server to automatically load the file** check box.

If selected, you must use the Universal Naming Convention (UNC) path, such as `\servername\filename` and you must ensure that the IBM Cognos server has access to the file.

3. Under **Data items**, select the check box for the data items that you want to import.

4. Type a name for the namespace and click **Next**.

The namespace appears in the **Source tree**, and identifies the external data within the package. By default, the name is the name of your imported external data file.

5. If you do not want to link your data or change the data attributes, click **Finish** now.

Steps to Map Data

1. On the **Data Mapping** page, link existing query subjects in your enterprise data to data items in your external data file

2. Under **Existing query subject / report**, click the ellipsis (...) button and do one of the following:

- If you want to select a query subject from the data tree, click **Choose Query Subject** and select a query subject.
- If you want to select from the query subjects included in a report, click **Choose a Report** and select a report.

3. Click the two data items that you want to link and click the **New link** button.

A link appears linking two data items.

Tip: You can create multiple links for multiple data items. To delete a link, select the link and click **Delete Link**.

4. Click **Next**.

Steps to Finish Importing Data

1. On the **Data Attributes** page, specify the attributes for each data item after it is imported and click **Next**.

For example, if you import numeric data items, you can change the default summary and number of decimal places.

2. On the **Mapping Options** page, specify the relationships between the linked data items and how to process the results in the report output.

3. Click **Finish**.

Steps to Publish the Package

1. If you want to rename the package that will be published or change the location where it is published, do the following:
 - In the **Manage External Data** dialog, under **Package name**, click the ellipsis (...) button.
 - Type a new name for the package and select the location where to save it.
 - Click **Save**.

2. Click **Publish**.

IBM Cognos BI imports your external data into a new package. A namespace with the data items from your external file appears in the **Source** tree of the **Insertable Objects** pane.

You can now create reports with your external data.

Edit Your External Data

After you import your data, you can change the data mappings and options that you originally specified and republish the package. You can

- rename the namespace, which renames the organizational folder that appears in the data tree of the **Source** tab in the **Insertable Objects** pane. If you change the namespace, you are prompted for the external data file when you run the report.
- change which columns to import
- change the data mapping links
- change the data attributes
- change the mapping options

You can also import multiple external data files into the same package. To do this, your data modeler must modify governors in the model and republish the package that contains your enterprise data. For more information, see the IBM® Cognos® Framework Manager *User Guide*.

Steps

1. From the **Tools** menu, click **Manage External Data** .
2. In the **Manage External Data** dialog box, select the external data to edit and click the edit button .
3. In the left pane, select the options that you want to change.
4. Click **OK** and the re-publish the package.

IBM Cognos BI re-imports your external data and updates the data items that appears in the data tree of the **Insertable Objects** pane.

You can now create and update reports with your external data.

Delete Your External Data

You can delete your external data from within the package that you created if you no longer need it.

Steps

1. From the Tools menu, click **Manage External Data** .
2. In the **Manage External Data** dialog box, select the external data package to delete and click the delete button .
3. Click **Publish**.

The external data namespace is removed from the package.

If you also no longer require the external data package or any reports created with the package, you can delete it from within IBM® Cognos® Connection.

Running a Report That Contains External Data

Reports that contain external data run the same way as reports that contain only enterprise data. If you have access to the report, you will also have access to the external data included within the report.

You may be prompted to select the location of the external data file when you run the report if either

- the report author did not specify to automatically load the file.
- the IBM® Cognos® BI server can not locate the file.

However, you are not re-prompted within the same Web browser session.

You can determine whether data in a report uses external data by tracing its lineage. For more information, see "[View Lineage Information for a Data Item](#)" (p. 101).

Making your Reports Public

After you created a report that uses your external data, you may want to make it public to share it with coworkers. They can run your report using your external data file that you made available on a public network drive that the IBM® Cognos® server can access. They can also use their own version of the file. If they use their own version, the file must contain the same columns as your original external data file that you used to import the data and create the report. In addition, you must clear the **Allow server to automatically load file** check box in the **Select Data** page of the **Manage External Data** wizard.

To make reports public, you must save them in the **Public Folders** area of the IBM Cognos portal. To save content in **Public Folders**, you must have the appropriate permissions. Contact your IBM

Cognos administrator to obtain permissions and to inform him or her that you are sharing a package or files.

If you share your reports, ensure that you maintain the reports.

Chapter 15: Using Conditions

You can define conditions to control what users see when they run a report. Conditions can apply to specific items in a report. For example, you can define a conditional style to highlight exceptional data, such as product revenue that exceeds your target.

Conditions can also apply at the report layout level. Conditional layouts are useful for delivering reports to a multilingual audience. For example, you can have text items, such as titles and cover pages, appear in the same language as the data in the report.

You can use conditions to

- [highlight data using conditional styles](#)
- [highlight data using style variables](#)
- [specify which objects are rendered when a report is run](#)

Highlight Data Using a Conditional Style

Add conditional styles to your report to better identify exceptional or unexpected results. A conditional style is a format, such as cell shading or font color, that is applied to objects if a specified condition is true.

For example, you want to automatically highlight in green the departments in your organization that meet their budget quotas and highlight in red the departments that go over budget. Creating conditional styles color-codes information in your reports so that you can find areas that need attention.

You can apply multiple conditional styles to objects. For example, you can apply one style in specific cells and another style for the overall report. If multiple styles set the same property, such as font color, the last style in the list is applied.

You can apply conditional styles based on any data item in your report.

You can create a conditional style once and reuse it on multiple objects in your report. You can specify the order in which conditional styles are applied. You can also use existing local classes as your conditional styles.

You can create the following types of conditional styles.

Type	Description
Numeric Range	Highlights straight numerical data, such as revenues and losses.
Date/Time Range	Highlights data from specific dates and times.
Date Range	Highlights data from specific dates.

Type	Description
Time Range	Highlights data from specific times.
Interval	Highlights data falling between set intervals.
String	Highlights specific alphanumeric items in a report. For example, you can highlight all instances of a specific word or phrase, such as Equipment. String criteria are case-sensitive. If multiple string conditions are met, only the first conditional style is applied.
Advanced	Creates conditional styles that use calculations or expressions. If multiple advanced conditions are met, only the first conditional style is applied.

You can also apply a conditional color palette to a chart.

You can perform a search to find objects in your report that use conditional styles. You can also view all the conditional styles used in your report to delete or modify them.

You can also use variables to highlight data. If a report contains both conditional styles and style variables, the style variables are applied first and then the conditional styles are applied.

Tip: The Return Quantity by Order Method sample report in the GO Data Warehouse (analysis) package includes conditional highlighting. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Steps to Create a New Conditional Style

1. Click the object for which you want to define a conditional style and click the conditional styles button .

Tip: You can also right-click the object and click **Style, Conditional Styles** or click the object, and then, in the **Properties** pane, set the **Conditional Styles** property.

2. Click the add button  and click **New Conditional Style**.

3. Select the data item to determine the condition and click **OK**.

The type of conditional style that you can use depends on the type of data item you select.

4. In the **Name** box, type a name for the conditional style.

5. To define a numeric value, date/time, date, time, or interval condition:

- Click the new button  and select a value to define a threshold.

The value appears in the **Range** column, and two ranges are created.

- For each range, under **Style**, click one of the predefined styles to apply to the range or click the edit button  and create a new style.

Tip: You can also define a style for the cells in your report that have missing values.

- Repeat the steps above to add other conditions.

Tip: Under **Style**, pause the pointer over each range  to see the condition produced for that range.

- To move a value above or below a threshold, click the arrow button  next to the value.

For example, you insert a threshold value of five million. By default, the ranges are less than or equal to five million and greater than five million. Moving the five million value above the threshold changes the ranges to less than five million and greater than or equal to five million.

6. To define a string condition:

- Click the new button and select how to define the condition.
- To select more than one individual value, click **Select Multiple Values** and click the values.
- To type specific values, click **Enter Values** and type the values.
- To specify your own criteria, such as values that begin with the letter A, click **Enter String Criteria** and specify the condition.
- For each condition, under **Style**, click one of the predefined styles to apply or click the edit style button and create a new style. Specify the style to apply to remaining values by clicking one of the predefined styles beside **Remaining values (including future values)**.
- Specify the order in which to evaluate the conditions.

Conditions are evaluated from top to bottom, and the first condition that is met is applied.

Steps to Reuse an Existing Conditional Style

1. Click the data item for which you want to define a conditional style, and then click the conditional styles button .

Tip: You can also right-click the data item and click **Style**, **Conditional Styles** or click the data item, and then, in the **Properties** pane, set the **Conditional Styles** property.

2. Click the add button , click **Use Existing Conditional Style**, and select the style.

Steps to Create an Advanced Conditional Style

1. Click the data item for which you want to define a conditional style, and then click the conditional styles button .

Tip: You can also right-click the data item and click **Style**, **Conditional Styles** or click the data item, and then, in the **Properties** pane, set the **Conditional Styles** property.

2. Click the add button  and click **Advanced Conditional Style**.
3. Type a name for the conditional style.

4. Click the add button  and specify the expression that defines the condition.
5. For each condition, under **Style**, click one of the predefined styles to apply or click the edit button  and create a new style. Specify the style to apply to remaining values by clicking one of the predefined styles beside **Remaining values (including future values)**.
6. Specify the order in which to evaluate the conditions by clicking a condition and then clicking the move up or move down arrow.

Conditions are evaluated from top to bottom, and the first condition that is met is applied.

Manage Conditional Styles

You can view, modify, or delete the conditional styles that are applied to your report. You can also define a new conditional style.

In the IBM® Cognos® Report Studio options, you can specify whether to automatically delete conditional styles that are no longer used in a report.

Step

- From the Tools menu, click **Manage Conditional Styles**.

Example - Add a Conditional Style to an Existing Report

You are a report writer at The Great Outdoors Company, which sells sporting equipment. You have a report that compares current year data to previous year data and highlights negative percentage variances in red (Poor) and positive percentage variances in green (Excellent). You want to add a third conditional style to indicate percentage variances above 0 but less than 10. You create a conditional style that highlights percentage variances between 0 and 10% in yellow (Average).

Steps

1. Open the GO Balance Sheet as at Dec 31 2006 report from the GO Data Warehouse (analysis) package.
2. Right-click any cell in the % Variance column and click **Style, Conditional Styles**.
3. Select **Conditional Style 1** and click the edit button .
4. Select the first advanced condition listed and click the edit button.
5. Delete **=0** from the expression definition.
6. Click the **Functions** tab, and then expand the **Operators** folder.
7. Double-click **between**, and then click after **between** in the expression, add a space, and type **0**.
8. Double-click **and**, and then click after **and** in the expression, add a space, type **.1**, and click **OK**.
9. From the **Style** box associated with this condition, click **Average**.
10. Select the second advanced condition listed and click the edit button.

11. Replace >0 in the expression with >.1 and click OK.
12. Leave the style associated with this condition as is.
13. Run the report.

The new conditional style appears in the % Variance column.

Year to date (USD \$000's)	Current Year (2006)		Previous Year (2005)	
	December	December	Variance	% Variance
Assets (total)	9,459,048	10,717,093	-1,258,045	-11.74%
Current assets (total)	5,709,903	5,960,900	-250,996	-4.21%
Operating assets (total)	3,319,279	4,301,658	-982,378	-22.84%
Other assets (total)	429,865	454,536	-24,671	-5.43%
Liabilities (total)	-4,255,662	-3,856,126	-399,536	10.36%
Current liabilities (total)	-3,364,742	-3,059,376	-305,366	9.98%
Long-term and other liabilities (total)	-890,920	-796,749	-94,171	11.82%
Equity (total)	-5,203,386	-6,860,968	1,657,582	-24.16%
Common stock	-1,871,226	-1,780,558	-90,668	5.09%
Other capital	-991,916	-615,202	-376,714	61.23%
Retained earnings - net	-2,327,374	-4,450,170	2,122,797	-47.70%
Currency translation gain (or loss)	-12,870	-15,038	2,168	-14.41%
Declared dividends	0	0	0	-
BALANCE				

Highlight Data Using a Style Variable

Highlight data in your report to better identify exceptional results. For example, you want to identify sales representatives who have exceeded their quota. You create a condition that checks whether each representative's sales for the year is greater than their quota for the year.

Style variables are useful if you are working with reports that were created in a previous version of IBM® Cognos® BI or if you want to use language variables to specify conditional styles.

You can also use conditional styles to highlight data. If a report contains both conditional styles and style variables, the style variables are applied before the conditional styles.

Steps

1. [Create a variable](#) and define the condition that determines if the data will be highlighted.
2. In the work area, click the column to highlight based on the condition.
3. In the Properties pane, double-click the Style Variable property.
4. Click Variable, click the variable to assign to the object, and click OK.
5. If you assigned a string variable, in the Values box, select the values for the condition to support.
Tip: A default value exists for the variable, and it is always selected.
6. If you assigned a language variable, in the Values box, select the languages for the condition to support.
Tip: A default value exists for the variable, and it is always selected.
7. Click OK.

8. Pause the pointer over the condition explorer button  and click a value other than the default value.

Tip: When you select a value in the condition explorer, the Explorer bar becomes green to indicate that conditional formatting is turned on and that any changes you make to the report apply only to the variable value.

For example, if you created a Boolean variable, click Yes.

9. In the **Properties** pane, specify the formatting with which to highlight the column when the condition is satisfied.

For example, click the **Border** property to create a thicker border around the column.

10. Repeat steps 8 to 9 for other possible values defined for the variable.

Tip: To view the report with no variables applied, pause the pointer over the condition explorer button and click **(No variable)** or triple-click the Explorer bar.

When you run the report, the report objects to which you applied the variable are highlighted when the condition is satisfied. For example, if you created a Boolean variable, the objects are highlighted when the condition is met. If the condition is not satisfied for any object, no conditional formatting is applied.

Adding Conditional Rendering to a Report

You can specify which objects are rendered when a report is run. You can

- [add variables](#)
- [hide and show objects](#)
- [add conditional rendering](#)
- [add multiple layouts](#)
- [create multilingual reports](#)

Add a Variable

Before you can add conditional formatting or conditional rendering to your report, you must add a variable. You can create a variable in the condition explorer or in the **Properties** pane.

Steps to Add a Variable from the Condition Explorer

1. Pause the pointer over the condition explorer button  and click **Variables**.
2. In the **Insertable Objects** pane, drag one of the following variables to the **Variables** pane:
 - To create a variable that has only two possible values, Yes and No, drag **Boolean Variable**.
 - To create a variable whose values are string-based, drag **String Variable**.
 - To create a variable whose values are different languages, drag **Report Language Variable**.

3. If you created a Boolean variable, in the **Expression Definition** box, define the condition and click **OK**.

For example, the following expression returns the value **Yes** if revenue is less than one million and the value **No** if revenue is greater than or equal to one million:

`[Revenue]<1000000`

For information about creating expressions, see "[Using Relational Calculations](#)" (p. 342) or "[Using Dimensional Calculations](#)" (p. 396) and "[Calculation Components](#)" (p. 769).

4. If you created a string variable, do the following:

- In the **Expression Definition** box, define the condition and click **OK**.

For example, the following expression returns the value **high** if revenue is greater than one million and the value **low** if revenue is less than or equal to one million:

`if ([Revenue]>1000000) then ('high') else ('low')`

For information about creating expressions, see "[Using Relational Calculations](#)" (p. 342) or "[Using Dimensional Calculations](#)" (p. 396) and "[Calculation Components](#)" (p. 769).

- Click the add button  in the **Values** pane.
- For each value that the variable can assume, type the name of the value that corresponds with the possible outcomes defined in the expression.

For example, in the previous expression, you must create two values for the variable, **high** and **low**.

Tip: You can create a group by clicking two or more values and then clicking the group values button . For example, you can create a group that includes the available French languages.

5. If you created a language-specific variable, in the **Languages** dialog box, select the languages to support.

Steps to Add a Variable from the Properties Pane

1. Select the report object.
2. In the **Properties** pane, under **Conditional**, double-click the conditional property to which to assign the variable.

The following conditional properties are available:

Goal	Conditional Property
<p>Specify a variable based on which text can be conditionally Text Source Variable shown.</p> <p>For example, you want different text to appear when a report is run in a different language.</p>	

Goal	Conditional Property
<p>Specify a variable based on which object can be conditionally rendered.</p> <p>For example, you want to make a revenue report smaller by not rendering rows that are below a threshold.</p>	Render Variable
<p>Specify a variable based on which object can be conditionally styled.</p> <p>For example, you want data that meets some criterion to appear in a different color.</p>	Style Variable
<p>Specify a variable based on which objects in a block can be conditionally rendered. Applies only to conditional block objects that you insert in a report.</p>	Block Variable

3. In the **Variable** box, click an existing variable or one of the following variable types:
 - <New language variable>
 - <New string variable>
 - <New boolean variable>
4. In the **New Variable** dialog box, in the **Name** box, type the name of the variable.
5. If you created a string variable, click the add button , type the string values to define, and click **OK**.
6. If you created a language variable, select the languages to support and click **OK**.
7. In the **Expression Definition** box, define the condition.

Hide or Show an Object

You can hide and show objects in a report based on a condition you define.

You can also specify that an object should not be rendered based on a condition.

Tip: The Global Bonus Report sample report in the GO Data Warehouse (analysis) package includes hidden objects. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

Steps

1. [Create a variable](#), and define the condition that determines if the object is shown or hidden.
- Tip:** Create a Boolean variable to show and hide objects, as this type of variable has only two possible values.

2. In the **Insertable Objects** pane, on the **Toolbox** tab , drag a **Conditional Blocks** object to the work area.
3. Select the conditional block.
4. In the **Properties** pane, double-click the **Block Variable** property.
5. In the **Variable** box, click the variable you created and click **OK**.
6. Set the **Current Block** property to **Yes**.
7. In the **Insertable Objects** pane, drag the object to show or hide to the conditional block.

For example, drag a data item from the **Source** tab or from the **Data Items** tab.

You may need to link the report page to a query before you can add a data item to the block.

When you run the report, the report objects to which you applied the variable are visible when the condition is satisfied and invisible when it is not.

Add Conditional Rendering

Add conditional rendering to specify which objects are rendered when a report is run. This is useful when your report contains sensitive data.

Conditional rendering is not the same as [hiding objects](#). When you hide an object, the object exists but is transparent. If an object is not rendered, it is not in the report.

For a list of objects that can be rendered conditionally, see the **Render Variable** property in "[Report Studio Object and Property Reference](#)" (p. 601).

Steps

1. Select the list column to be rendered conditionally.

Tip: You must select the list column, not the list column body or the list column title. If the body or title is selected, as indicated in the **Properties** pane, click the select ancestor button  and click the list column.

2. In the **Properties** pane, double-click the **Render Variable** property.
3. Click **Variable** and click the variable that will determine if the column will be rendered.
4. In the **Render for** box, select the values that the condition will support.

Tip: A default value exists for the variable, and it is always selected.

Example - Create a Conditional Report

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a report that shows orders after a date specified by the user. The report will prompt the user for a date and ask whether the user wants to see a description for each order.

Steps

1. Open IBM® Cognos® Report Studio with the **GO Data Warehouse (query)** package.

2. In the Welcome dialog box, click **Create a new report or template**.
3. In the New dialog box, click **List** and click **OK**.
4. In the Insertable Objects pane, on the Source tab , expand **Sales and Marketing (query)**, and **Sales (query)** and add data items to the list:
5.
 - Expand **Time dimension** and add **Date**.
 - Expand **Sales order** and add **Order number**.
 - Expand **Product** and add **Product name** and **Product description**.
 - Expand **Sales fact** and add **Quantity**, **Unit price**, and **Revenue**.
6. Click **Date**, and then click the section button .
7. Select the **Order Number** column and click the group button .
8. Click **Revenue**, and then click the summarize button  and click **Total**.
9. Change the title of the report to **New Orders**.
10. Pause the pointer over the page explorer button  and click **Prompt Pages**.
11. Create a new prompt page by double-clicking **Page** in the Insertable Objects pane.
12. Double-click the new prompt page.
13. In the Insertable Objects pane, on the Toolbox tab , double-click **Text Item** and type the following text:

Enter the start date, and select if descriptions will be shown.
14. Insert a 2 by 2 table into the prompt page by clicking the insert table button  and moving the pointer until four squares are highlighted in a 2 by 2 pattern.
15. In the Insertable Objects pane, on the Toolbox tab, drag a **Text Item** into the upper-left cell and type the following text:

Starting Date
16. In the Insertable Objects pane, drag a **Text Item** into the lower-left cell and type the following text:

Show Descriptions
17. In the Insertable Objects pane, drag a **Date Prompt** into the upper-right cell.
18. In the **Prompt Wizard** window, select **Create a new parameter**, type **p_Date** in the space provided, and then click **Next**.
19. In the **Create Filter** window, select **Create a parameterized filter** with the following entries:

- For **Package item**, click the ellipsis (...) button and open **Sales (query)** and **Time dimension**, and then click **Date**.
 - For **Operator**, click >.
20. Click **Finish**.
21. In the **Insertable Objects** pane, drag a **Value Prompt** into the lower-right cell.
22. In the **Prompt Wizard**, in the **Choose Parameter** window, select **Create a new parameter**, type **p_ShowDesc** in the space provided, and then click **Finish**.
23. Select the **Value Prompt** and, in the **Properties** pane, double-click **Static Choices**.
24. Click the add button .
25. In the **Edit** dialog box, type **Yes** in both the **Use** and **Display** boxes.
26. Click the add button.
27. In the **Edit** dialog box, type **No** in both the **Use** and **Display** boxes.
28. Click **OK**.
29. Pause the pointer over the condition explorer button  and click **Variables**.
30. In the **Insertable Objects** pane, create a new Boolean variable by double-clicking **Boolean Variable**.
31. In the **Report Expression** dialog box, type the following in the **Expression Definition** window and click **OK**:
- ```
ParamDisplayValue("p_ShowDesc") = 'Yes'
```
32. In the **Properties** pane, set the **Name** property to **showDesc**.
33. Pause the pointer over the page explorer button  and click the report page.
34. Click the **Product descriptions** column.
35. In the **Properties** pane, click the select ancestor button  and click **List Column**.
36. In the **Properties** pane, set the **Render Variable** property to the **showDesc** Boolean variable you created.
37. Run the report.

The report prompts you for a date provides orders that occur after the date you entered. The report also asks whether to show the **Descriptions** column, and the column is rendered only if you choose **Yes**.

## Add Multiple Layouts

Add multiple layouts to show a report in different ways. For example, you can define a different layout for each language in a [multilingual report](#). This allows you to create a single report that can be viewed by report consumers that use different regional settings.

### Steps

1. [Create a variable](#) and define the condition that will be used for each layout.

For example, create a report language variable that includes each language that requires a conditional layout.

**Note:** Expressions used in a conditional layout cannot reference a query.

2. From the File menu, click **Conditional Layouts**.
3. Select a variable, and then select the values that require a separate layout.

A layout is created for each value you selected. Use the page explorer to navigate the different layouts. For each layout, click **Report Pages** to create a report page or **Prompt Pages** to create a prompt page and add objects.

**Tip:** You can create new variables from the **Conditional Layouts** dialog. The variables are added to the condition explorer. For more information, see "[Add a Variable](#)" (p. 478).

## Setting up a Multilingual Reporting Environment

You can create reports that show data in more than one language and use different regional settings. This means that you can create a single report that can be used by report consumers anywhere in the world.

The samples databases provided with IBM® Cognos® store a selection of text fields, such as names and descriptions, in more than 25 languages to demonstrate a multilingual reporting environment. For information about how data is stored in the samples databases and how the samples databases are set up to use multilingual data, see the *Administration and Security Guide*.

Here is the process for creating a multilingual reporting environment:

- Use multilingual metadata.

The data source administrator can store multilingual data in either individual tables, rows, or columns.

For more information about configuring your database for multilingual reporting, see the *Administration and Security Guide*.

- Create a multilingual model.

Modelers use Framework Manager to add multilingual metadata to the model from any data source type except OLAP. They add multilingual metadata by defining which languages the model supports, translating text strings in the model for things such as object names and descriptions, and defining which languages are exported in each package. If the data source contains multilingual data, modelers can define queries that retrieve data in the default language for the report user.

For more information, see the Framework Manager *User Guide*.

Create multilingual maps.

Administrators and modelers use a Microsoft® Windows® operating system utility named Map Manager to import maps and update labels for maps in Report Studio. For map features such as country and city names, administrators and modelers can define alternative names to provide multilingual versions of text that appears on the map.

For more information, see the Map Manager *Installation and User Guide*.

Create a multilingual report.

The report author uses Report Studio to create a report that can be viewed in different languages. For example, you can specify that text, such as the title, appears in German when the report is opened by a German user. You can also add translations for text objects, and create other language-dependent objects.

For more information, see the Report Studio *User Guide*.

Specify the language in which a report is viewed.

You can use IBM Cognos Connection to do the following:

- Define multilingual properties, such as a name, screen tip, and description, for each entry in the portal.
- Specify the default language to be used when a report is run.

**Tip:** You can specify the default language on the run options page, in the report properties, or in your preferences.

- Specify a language, other than the default, to be used when a report is run.

For more information, see the IBM Cognos Connection *User Guide*.

The data then appears in the language and with the regional settings specified in

- the user's Web browser options
- the run options
- the IBM Cognos Connection preferences

Any text that users or authors add appears in the language in which they typed it.

## Create a Multilingual Report in Report Studio

You can create a report in Report Studio that can be viewed in different languages. For example, you can specify that text such as the title appears in German when the report is opened by a German user. You can also add translations for text objects and create other language-dependent objects.

If you want the report to show data in different languages, the model must also be multilingual.

### Steps

1. Create a report language **variable**.

2. In the work area, select the object to modify based on a language.

3. In the **Properties** pane, double-click the **Style Variable** property.

If you are changing the language of a text string, click **Text Source Variable** instead.

4. Click **Variable** and click the language variable you created.

5. In the **Values** box, select the languages for the condition to support and click **OK**.

**Tip:** A default value exists for the variable, and it is always selected.

6. Pause the pointer over the condition explorer button  and a language for the variable.

**Tip:** When you select a value in the condition explorer, the Explorer bar becomes green to indicate that conditional formatting is turned on and that any changes you make to the report apply only to the variable value.

7. In the **Properties** pane, specify the formatting for the language.

For example, to change the language of a text string, double-click the **Text** property and select the new string.

8. Press Enter when you are done.

9. Repeat steps 6 to 8 for all other languages specified for the variable.

**Tip:** To view the report with no variables applied, pause the pointer over the condition explorer button and click **(No variable)** or triple-click the Explorer bar.

When you run the report, the report objects to which you applied the variable are formatted according to the browser's language.

# Chapter 16: Bursting Reports

Burst a report to distribute its contents to various recipients. Bursting is the process of running a report once and then dividing the results for recipients who each view a subset of the data. For example, salespeople in different regions each need a report showing the sales target for their country. You use burst reports to send each salesperson only the information they need. Burst reports can be distributed by email or saved to a folder for viewing in IBM® Cognos® Connection.

**Tip:** The Bursted Sales Performance Report sample report in the GO Data Warehouse (analysis) package includes bursting. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

To burst a report against a dimensional data source, see "[Creating Burst Reports Using Dimensional Data Sources](#)" (p. 492).

You cannot burst crosstab or chart reports. However you can burst a report where a container includes a crosstab or chart that is part of a master detail relationship. In this situation, you can burst only HTML, PDF, and XLS output formats; you cannot burst CSV or XML output formats.

For information about avoiding disk space exhaustion when bursting charts or crosstabs, see "[Master Detail or Burst Reports with Charts or Crosstabs May Result in Denial of Service](#)" (p. 388).

To create a burst report, you

- [define burst recipients](#)
- [specify burst groups](#)
- [set burst options](#)
- [enable bursting](#)

## Defining Burst Recipients

Define the recipients who will receive data when the report is run. You can distribute burst reports to individual users, groups, roles, distribution lists, and contacts. To define the recipients, you

- [create a calculated field](#) in the report
- [create a burst table](#) in the data source

## Create a Calculated Field

You can use a calculated field to dynamically create burst report recipients.

### Steps

1. Pause the pointer over the query explorer button  and click the query that will produce the data to distribute.

2. In the **Insertable Objects** pane, on the **Toolbox** tab , drag **Data Item** to the **Data Items** pane.
3. In the **Expression Definition** box, type the expression that will generate the list of recipients and click **OK**.  
For example, typing the following expression builds the list of the employees of The Great Outdoors Company. The expression concatenates the first letter of each employee's first name with their last name.  
`lower(substring([Employee summary (query)].[Employee by organization].[First name],1,1) + [Employee summary (query)].[Employee by organization].[Last name])`
4. To give the data item a more meaningful name, in the **Properties** pane, set the **Name** property to a different name and press the **Enter** key.

## Creating a Burst Table

You can create a burst table to specify the list of recipients.

You can also use an existing table as the burst table.

Creating a burst table involves

- creating the table in the source database
- importing the table into a package

### Creating the Burst Table in the Source Database

You can create a burst table in the source database for the list of recipients. The steps you must follow depend on the database system you are using. The burst table must contain the following columns:

- A unique identifier  
**Tip:** Some database systems do not require a unique identifier for each table.
- A recipient column
- The data item on which to burst

You can also include other columns that provide additional information. For example, if you plan to distribute reports by email, you can add a column for the email address of each recipient.

After you create the table, add the recipients who will receive the report. You can create a mixed recipients list that includes individual users, groups, roles, contacts, distribution lists, or email addresses. For example, a burst table may contain the following recipients.

| Recipient example                             | Recipient type |
|-----------------------------------------------|----------------|
| CAMID(":Canada")                              | Group          |
| CAMID(":")/contact[@name='Silvano Allessori'] | Contact        |

| Recipient example                                      | Recipient type                                                                                                                                   |
|--------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| CAMID(":")/distributionList[@name='European Partners'] | Distribution list                                                                                                                                |
| CAMID("LDAP_Local_ID:u:uid=gbelding,ou=people")        | Authentication provider user or group, where LDAP_Local_ID is the name of an LDAP namespace ID, and people is the name of an organizational unit |
| c10@ibmcognos99.com                                    | Email address                                                                                                                                    |

CAMID stands for Cognos® Access Manager ID, and it represents an internal search path to the recipients. Specify search paths when you want to save burst reports in a folder. You can obtain the search path in IBM® Cognos Connection by opening the **Set properties** page for each recipient and clicking **View the search path**. Ensure that you use the proper syntax when adding recipients to the burst table.

In the case of NTLM namespaces, user IDs in the search path use alphanumeric or numeric characters that make them difficult to read. You can use the following alternate search path syntax:

```
directory/namespace[@name="Local NT"]//account[@userName="gbelding"]
```

where Local NT is the name of a namespace and gbelding is the name of a user. The double slash before the account element indicates that you are searching all accounts under the specified namespace.

**Note:** If you have a mixed recipients list, do not mix email address recipients and alternate path recipients. Because the alternate path syntax contains the @ symbol, it will be mistaken for an email address.

For more information about users, groups, roles, contacts, and distribution lists, see the IBM Cognos Business Intelligence *Administration and Security Guide*.

## Importing the Table into a Package

After you create the burst table in the source database, you must add it to the package that you will use to create the report.

In IBM Cognos Framework Manager, do the following:

- ❑ Open the package.
- ❑ Import the table.
- ❑ Define the relationship between the burst table and the table containing the data item on which to burst.

For example, you are bursting on country code. You define a relationship between country code in the burst table and country code in the Country table.

- ❑ Save and publish the package.

For more information about importing tables and creating relationships, see the Framework Manager *User Guide*.

## Specify a Burst Group

Specify burst groups to set how the report will be distributed. Burst groups are defined by a data item that you create in the report or that you add from the burst table.

### Steps

1. Pause the pointer over the query explorer button  and click the query that will produce the data to distribute.
2. If you are creating a data item, do the following:
  - In the Insertable Objects pane, on the Toolbox tab , drag Data Item to the Data Items pane.
  - In the Expression Definition box, type the expression that defines the burst key.

For example, the following expression builds an email address for each sales representative in The Great Outdoors Company. The expression incorporates the calculated field that was [previously created](#), which is named `userID` below, with `ibmcognos99.com` as the domain name.

```
[userID] + '@ibmcognos99.com'
```

**Tip:** To give the data item a more meaningful name, in the Properties pane, set the Name property to a different name and press Enter.
3. To specify a burst table column as the data item, do the following:
  - In the Insertable Objects pane, on the Source tab , expand the burst table.
  - Drag the data item to the Data Items pane.

For example, if you are bursting reports by email, drag the data item containing email addresses.

## Set Burst Options

Set burst options for the report to indicate the data item on which to burst and the recipients.

Before you set burst options, ensure that the data item on which you intend to burst is in the report and is grouped. The grouped column will create the appropriate subsets of data. In addition, you must associate the burst key with this level of grouping.

### Steps

1. From the File menu, click **Burst Options**.
2. Select the **Make report available for bursting** check box.

3. Under **Burst Groups**, in the **Query** box, click the query that contains the data item on which to burst.

**Tip:** You can choose a query that does not appear in the layout. This is useful to distribute the same report to all burst recipients.

4. In the **Label** box, click the data item with which to label each burst report.
5. Click the edit button .
6. In the **Data Items** box, drag the data item on which to burst to the **Groups** folder and click **OK**.

**Tip:** You can specify the sort order of data within each group by dragging data items to the **Sort List** folder and then clicking the sort order button .

7. Under **Burst Recipient**, in the **Query** box, click the query that contains the data item to be used as the distribution list.
8. In the **Data Item** box, click the data item that contains the recipients.

9. In the **Type** box, choose the method to burst the report:

- Click **Automatic** to let IBM® Cognos® BI determine from the data item whether to email reports or send them to folders in IBM Cognos Connection.
- Click **Email addresses** to distribute reports by email.
- Click **Directory entries** to distribute reports to folders that recipients can access in IBM Cognos Connection.
- **Note:** To burst reports to multiple mobile device users, you must choose to distribute reports to folders. You can choose **Directory entries**, or you can choose **Automatic** if the data item returns directory entries instead of email addresses. To view the reports, the recipients must have IBM Cognos Mobile installed on their mobile devices. For more information about IBM Cognos Mobile, see the IBM Cognos Mobile *Installation and Administration Guide* and *User Guide*.

When recipients log into IBM Cognos BI, they will see only the report that is specific to them.

10. If the report contains two nested data containers, such as a list and a chart, click the ellipsis (...) button beside **Master detail relationships** and define the relationship between the data containers.

For information about master detail relationships, see "[Create a Master Detail Relationship](#)" (p. 386).

## Enable Bursting

When the report is ready to be distributed, enable bursting for the report in IBM® Cognos® Connection.

### Steps

1. Locate the report in IBM Cognos Connection.
2. Under **Actions**, click the run with options button .
3. Click **advanced options** on the right of the screen.
4. Under **Time and mode**, click **Run in the background**.
5. Select the **Burst the report** check box.
6. If you are distributing reports by email, select the **Send the report by email** check box.

**Tip:** If you are bursting the report to a folder, you can also send the report by email if the **Send the report by email** check box is selected. Reports will be emailed if the recipient's email address is stored in the authentication source you are using or if they are entered in the recipient's personal information in IBM Cognos Connection.

7. If the burst report contains a drill-through link to another report and you are distributing the burst report by email, do the following:
  - Click **Edit the options**.
  - Select the **Include a link to the report** check box.

If you do not select the check box, the drill-through links in the burst report will not work.

8. Run the report.

Allow a few moments for the report to run. If you are an administrator, you can view all outputs for the report. Under **Actions**, click **View the output versions for this report**. When burst recipients log into IBM Cognos Connection or access their email accounts, they will see only the data that is meant for them.

## Creating Burst Reports Using Dimensional Data Sources

You can burst a report using a dimensional data source by using burst information that is stored in the data source. Because you do not want to append bursting information to existing dimensional data sources, you can create a relational data source that contains the burst information.

Report bursting is limited when the underlying data source is a cube (MOLAP data source such as IBM® Cognos® PowerCube, Microsoft® Analysis Services, Oracle Essbase, or IBM DB2®/OLAP). The burst report must be a grouped report, and the burst is restricted to the outermost grouping in the report. For example, if you have a report grouped on Country and State, then you can burst the report only on Country.

To burst a report using a dimensional data source, do the following:

- In IBM Cognos Framework Manager, include both the dimensional data source that is the basis for reporting and the relational burst table in the model.

For more information about models, see the Framework Manager *User Guide*.

- In IBM Cognos Report Studio, create a master detail report in which the master query drives the report and the detail query contains the burst information.

You must group the master query on the data item on which you are bursting. This data item must have a corresponding data item in the relational burst table.

Author the detail query against the relational burst table. The burst table must contain two columns: the data item corresponding to the data item used in the master report for bursting and the data item that contains the recipient information. The recipient can be an email address or an expression that results in a search path to an object in IBM Content Manager, such as an account, group, role, contact, or distribution list.

For more information about master detail queries, see "[Create a Master Detail Relationship](#)" (p. 386).

- Ensure that the detail query, which must be evaluated by IBM Cognos BI when the report is executed, is not visible:

- Place a list that is based on the detail query in a conditional block with a box type of **None**.
- Link the master and detail queries using the following expression:

```
[Master Burst Key] = [Detail Burst Key]
```

When you set the burst options for the report, the master query provides the data items for the burst key and the detail report provides the data items for the burst recipients.

## Example - Burst a Report

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a report that lists product sales for each sales representative. The report is to be emailed to each sales representative, but they only need to see the data that pertains to them. You create a list report that you burst to each sales representative.

### Steps

1. Open IBM® Cognos® Report Studio with the GO Data Warehouse (query) package.
2. In the Welcome dialog box, click **Create a new report or template**.
3. In the New dialog box, click **List** and click **OK**.
4. In the Insertable Objects pane, on the **Source** tab , expand **Sales and marketing (query)** and **Sales (query)**. Add the following data items:
  - Expand Employee by region and add **Employee name**.
  - Expand Products and add **Product line**, **Product type**, and **Product**.
  - Expand Sales fact and add **Quantity** and **Revenue**.

5. Group the Employee name, Product line, and Product type columns.
6. Click the Employee name column, click the headers and footers button , and then click Create Header.

Employee name appears as a header in the list. You no longer need to keep the data item as a list column.
7. In the list, click the Employee name column and click the delete button .
8. Click Revenue, and then click the summarize button  and click Total.
9. Pause the pointer over the query explorer button  and click Query1.
10. In the Insertable Objects pane, on the Source tab, drag Email from the Employee by region folder to the Data Items pane.
11. From the File menu, click Burst Options.
12. Select the Make report available for bursting check box.
13. Under Burst Groups, in the Query box, click Query1.
14. In the Label box, click Employee name.
15. Click the edit button .
16. In the Data items box, drag Employee name to the Groups folder and click OK.
17. Under Burst Recipient, in the Query box, click Query1.
18. In the Data Item box, click Email.
19. In the Type box, click Email addresses.
20. Save the report.
21. Locate the report in IBM Cognos Connection.
22. Under Actions, click Run with options .
23. Click advanced options on the right of the screen.
24. Under Time and mode, click Run in the background.
25. Select the Burst the report check box.
26. Select the Send the report by email check box.
27. Run the report.

When sales representatives access their email accounts, they will see reports with only the data that is meant for them.

# Chapter 17: Format Data

Format data in a report to improve readability. For example, you can show all date values in the order year, month, and day. If you do not set **Data Format** properties here, data is formatted according to the properties set in the model. If the properties were not set in the model, data is formatted according to the International Components for Unicode (ICU) formats.

Special cells, such as overflow or underflow, errors, or not applicable values, appear with two dash (--) characters unless you override them.

You can also format data based on a condition or specify the format for a particular object.

Formatting data includes

- [setting default data formats](#)
- [specify the data format for objects](#)
- [understanding locale-sensitive properties](#)
- [suppressing empty cells](#)
- [specify what appears for data containers that have no data](#)
- [use patterns to format data](#)

If you create a calculation that uses mixed currency values, an asterisk character (\*) appears as the unit of measure. To remove the asterisk character, change the format of the corresponding row or column.

## Set the Default Data Formats

Set the default data properties for each type of data, including text, number, currency, percent, date, time, date/time, and time interval.

### Steps

1. From the **Data** menu, click **Default Data Formats**.
2. In the **Format type** box, click a format type.

The properties that you can set for the selected format type appear in the **Properties** box.

3. If you clicked the **Currency** format type and require different currencies in the report, click the add button  and select currency check boxes.

For example, you may have one column with values are in euros and another column whose values are in US dollars.

4. Set the properties.

If you added currencies in step 3, click each one and set the properties. If you did not add any currencies, any properties you set will apply to all currencies.

For properties in which you type meta-characters that represent certain types of information, such as YYYY-MM-DD for dates, the meta-characters that are required depend on the authoring language specified for the report. For more information, see "[Using Patterns to Format Data](#)" (p. 500).

If you set a value for the **Pattern** property, all other formatting properties are ignored with the following exceptions:

- **Missing Value Characters**
- **Zero Value Characters**
- **Negative Pattern**

Some properties are [language-sensitive](#) and should be changed only with caution.

The data formatting properties you set are applied to objects only in the current layout. If a data item contains values in multiple currencies but only a subset of those currencies have defined formats, the default format for the locale in which you are working is applied to values with no specified format.

## Specify the Data Format for an Object

Specify the format for a particular object if you are not getting the results you want. For example, you add a measure to a report and you want to see two decimals when you run the report. You set the number of decimals to two for the **Number** format type for the current layout. However, when you run the report, you see more than two decimals for the measure. To get the results you want, you must map the measure to the **Number** format type.

Data formats are not applied in delimited text (CSV) and XML report outputs.

### Steps

1. Click the object.
2. In the Properties pane, double-click the **Data Format** property.
3. Under **Format type**, click the format type to apply to the object.
4. To override any of the properties of the format type that were defined for the current layout, in the **Properties** box, click the property and specify its value.

## Specifying the Number of Decimals in Numbers

When specifying the number of decimals, IBM® Cognos® BI uses the IEEE 754 default rounding mode known as half even. With half even rounding, numbers are rounded toward the nearest truncated value, unless both truncated values are equidistant, in which case the value ending in an even digit is chosen, as follows:

- If the digit immediately after the precision to be shown is greater than 5, the number is rounded up.
- If the digit immediately after the precision to be shown is less than 5, the number is rounded down.
- If the digit immediately after the precision is a 5, the number is rounded down when the preceding digit is even and rounded up when the preceding digit is odd.

For example, the number 78.5 is rounded to 78, while the number 73.5 is rounded to 74.

In addition, if the maximum number of decimals is lower than the actual number of decimals in the number, the number is rounded to the maximum number of decimals.

## Locale-sensitive Properties

IBM® Cognos® Report Studio contains an extensive library of customized properties adapted to users from different regions who speak different languages. For example, if a modeler specifies that a given data item is a currency, only the proper currency symbol must be specified. When reports are created, Report Studio automatically adapts the format of the currency numbers to each user according to the content language specified in IBM Cognos Connection.

When modelers or report authors specify properties, these properties override user preferences and risk creating inconsistent formatting for users of other cultures. It is usually much easier and safer to let Report Studio take care of formatting. For example, for the date format type, different regions use different characters to represent the date separator. If you specify a date separator, you may confuse users in other regions.

The following data formatting properties are locale-sensitive:

- **Currency Symbol Position**
- **Date Separator**
- **Date Ordering**
- **Calendar Type**
- **Time Separator**
- **Display AM/PM Symbols**
- **Clock**
- **Decimal Symbol**
- **Negative Sign Position**
- **Thousands Separator**
- **Group Size (digits)**
- **Secondary Group Size (digits)**
- **Missing Value Characters**

- Zero Value Characters
- Pattern
- Negative Pattern

## Suppress Empty Cells

Sparse data may result in crosstabs showing empty cells. For example, a crosstab that matches employees with products results in many rows of empty values for the revenue measure if the employee does not sell those products.

You can suppress rows, columns, or rows and columns based on divide by zero, missing, and overflow values. Suppressing rows or columns without data gives you a more concise view of your report.

Calculations are performed before suppression is applied. If you have multiple crosstabs or charts, you must select one in order to access suppression options.

Within a list report, suppression applies only to rows and is based on non-grouped data items. If details of a group are null but the header or footer is not null, the group value is not suppressed.

Suppression can also be applied to charts, repeaters, and repeater tables.

Access to the suppression feature depends on the settings in your modeling component, IBM® Cognos® Transformer, IBM Cognos Framework Manager, and IBM Cognos Administration.

### Steps

1. From the Data menu, click **Suppress**  and click **Suppression Options**.
2. Under **Suppress**, choose what sections to suppress.
3. Under **Suppress the following**, choose which values to suppress.

## Suppress Null Cells Using Filters

You can also use filters to suppress null cells in your reports. Using filters ensures that calculations take suppression into account. You may also obtain better report performance because the filtering is done at the data source.

If your report includes more than one measure or fact expression, it is best to base the filter on only one underlying fact or measure.

### Relational-style Reports

In relational-style reports, you can use detail and summary filters.

### Dimensional-style Reports

In dimensional-style reports, you can use the `filter` function.

For example, insert a set expression in your crosstab and use the expression `filter ( descendants ([Set]) is not null)`. If your crosstab includes three or more levels within the same dimension on an edge, use the expression `filter ( descendants ( currentMember([Hierarchy]) is not null)`.

If your crosstab includes nested sets, filter the sets using a cascading approach to improve performance. For example, first filter the outermost (or highest nested level) set, and then filter the remaining sets proceeding inward.

## Limitations When Formatting Empty Cells in SAP BW Data Sources

When working with SAP BW data sources, if the SAP BW server administrator configured custom formatting for empty cells on the SAP BW server, this custom format does not appear in IBM Cognos BI reports. Ask your administrator to configure the formatting of empty cells in IBM Cognos BI.

## Example - Suppress Zeros in Rows and Columns in an Existing Report

You are a report writer at The Great Outdoors Company, which sells sporting equipment. You have a report that compares current year data to previous year data. You want to suppress zeros in the report to make the report more concise. You use the zero suppression tool to set the level of suppression.

### Steps

1. Open the Great Outdoors Company Balance Sheet as at Dec. 31, 2006 report.
2. From the Data menu, click **Suppress**  and click **Suppression Options**.
3. Under **Suppress the following**, click **Rows and columns**.
4. Under **Suppress the following type of values**, select the **Zero values** check box.
5. Run the report.

Rows and columns containing zeros are hidden.

| Year to date (USD \$ 000's)             | 2006 Actual results in USD |  | 2005 Actual results in USD |  |
|-----------------------------------------|----------------------------|--|----------------------------|--|
|                                         | December                   |  | December                   |  |
| <b>Assets (total)</b>                   | \$4,353,363,798            |  | \$3,804,483,176            |  |
| Current assets (total)                  | \$2,336,392,807            |  | \$2,139,066,575            |  |
| Operating assets (total)                | \$1,778,012,688            |  | \$1,484,257,572            |  |
| Other assets (total)                    | \$238,958,303              |  | \$181,159,029              |  |
| <b>Liabilities (total)</b>              | (\$1,602,289,213)          |  | (\$1,754,874,006)          |  |
| Current liabilities (total)             | (\$1,144,967,294)          |  | (\$1,206,574,775)          |  |
| Long-term and other liabilities (total) | (\$457,321,919)            |  | (\$548,299,231)            |  |
| <b>Equity (total)</b>                   | (\$2,751,074,586)          |  | (\$2,049,609,170)          |  |
| Common stock                            | (\$719,183,000)            |  | (\$749,178,000)            |  |
| Other capital                           | (\$69,921,200)             |  | (\$183,898,457)            |  |
| Retained earnings - net                 | (\$1,961,541,522)          |  | (\$1,116,128,225)          |  |
| Currency translation gain (or loss)     | (\$428,864)                |  | (\$404,488)                |  |

## Specify What Appears for Data Containers that Contain No Data

You can specify what appears for a data container when there is no data available. For example, you can have text appear that states that there is no data available for a list object in a report. This can be done for lists, crosstabs, charts, maps, repeaters, repeater tables, and tables of contents.

**Tip:** The No Data sample report in the GO Sales (analysis) package includes data containers that have no data. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

### Steps

1. Click a data container.
2. In the Properties pane, click the select ancestor button  and click the data container type.  
For example, if the data container is a list, click List.
3. Set the **No Data Contents** property to Yes.  
Two tabs appear at the top of the data container.
4. Click the **No Data Contents** tab .
5. In the Insertable Objects pane, on the Toolbox tab, insert the objects to appear when there is no data available.

## Using Patterns to Format Data

You can format data so that it matches any pattern of text and numbers when default formats are not appropriate. For example, you can format dates to use full text including the era, or you can format them to only use numbers and show the last two digits of years to save space.

Using symbols and patterns can provide similar results as basic data formatting tasks. For example, you can set how many digits appear after the decimal point. You can achieve these types of results with a pattern, or you can set the **No. of Decimal Places** property. Patterns allow flexibility for more complex requirements.

Each supported content language code requires a specific set of symbols to be used in patterns. For each language code, there are two tables you will need; one for date and time symbols, and one for decimal symbols. The decimal symbols are the same for all locales, however, date and time symbols are grouped into six locale groups. Check the Date and Time Symbol section to see which locale group is used for your locale.

To define patterns, open the **Data Format** dialog box, and edit the **Pattern** property for each format type. Use the symbols that are defined in the language code tables, and follow these guidelines.

### Pattern Guidelines

When you define a pattern, the number of symbols you use affects how the data will be shown. There are different rules for text, numbers, and values that can take the form of text or numbers.

## Text

You can specify whether text is produced in full or abbreviated form.

| Number of symbols | Meaning          | Example              |
|-------------------|------------------|----------------------|
| 4 or more         | Full text form   | EEEE produces Monday |
| Less than 4       | Abbreviated form | EEE produces Mon     |

## Numbers

The number of symbols you use in a pattern sets the minimum number of digits that are produced in a report. Numbers that have fewer digits than specified are zero-padded. For example, if you specify mm for minutes, and the database value is 6, the report will show 06.

**Note:** The year value is handled differently. If you specify two symbols for year, the last two digits of the year value is produced. For example, yyyy produces 1997, and yy produces 97.

## Text and Numbers

For values that can produce text or numbers, such as months, you can specify whether text or numbers are produced, and whether words are abbreviated.

| Number of symbols | Meaning | Example               |
|-------------------|---------|-----------------------|
| 3 or more         | Text    | MMMM produces January |
|                   |         | MMM produces Jan      |
| Less than 3       | Numbers | MM produces 01        |
|                   |         | M produces 1          |

## Date and Time Symbols

Date and time symbols are divided into locales, each of which is detailed below.

### Locale Group A

Locales: af-za, en, en-au, en-be, en-bw, en-ca, en-gb, en-hk, en-ie, en-in, en-mt, en-nz, en-ph, en-sg, en-us, en-vi, en-za, fo-fo, gl-es, id, id-id, is, is-is, it, it-ch, it-it, kk-kz, ms, ms-bn, ms-my, nb-no, nl, nl-be, nl-nl, no, no-no, om-et, om-so, pl, pl-pl, pt, pt-br, pt-pt, so-dj, so-et, so-ke, so-so, sv, sv-fi, sv-se, sw-ke, sw-tz

| Meaning | Symbol | Presentation | Example |
|---------|--------|--------------|---------|
| Era     | G      | Text         | AD      |
| Year    | y      | Number       | 1996    |

| Meaning                        | Symbol | Presentation    | Example               |
|--------------------------------|--------|-----------------|-----------------------|
| Year (of 'Week of Year')       | Y      | Number          | 1996                  |
| Month in year                  | M      | Text and number | July and 07           |
| Week in year                   | w      | Number          | 27                    |
| Week in month                  | W      | Number          | 2                     |
| Day in year                    | D      | Number          | 189                   |
| Day in month                   | d      | Number          | 10                    |
| Day of week in month           | F      | Number          | 2 (2nd Wed in July)   |
| Day of Week (1=first day)      | e      | Number          | 2                     |
| Day in week                    | E      | Text            | Tuesday               |
| a.m. or p.m. marker            | a      | Text            | pm                    |
| Hour in day (1 to 24)          | k      | Number          | 24                    |
| Hour in a.m. or p.m. (0 to 11) | K      | Number          | 0                     |
| Hour in a.m. or p.m. (1 to 12) | h      | Number          | 12                    |
| Hour in day (0 to 23)          | H      | Number          | 0                     |
| Minute in hour                 | m      | Number          | 30                    |
| Second in minute               | s      | Number          | 55                    |
| Millisecond                    | S      | Number          | 978                   |
| Time zone                      | z      | Text            | Pacific Standard Time |
| Escape used in text            | '      | n/a             | n/a                   |
| Single quote                   | "      | n/a             | '                     |

## Locale Group B

Locales: be-by, bg-bg, el, el-gr, fi, fi-fi, hr, hr-hr, hu, hu-hu, ja, ja-jp, ko, ko-kr, ro, ro-ro, ru, ru-ua, ru-ru, sh-yu, sk, sk-sk, sl-si, sq-al, sr-sp, th, tr, tr-tr, uk-ua, zh, zh-cn, zh-hk, zh-mo, zh-sg, zh-tw

| Meaning                        | Symbol | Presentation    | Example             |
|--------------------------------|--------|-----------------|---------------------|
| Era                            | G      | Text            | AD                  |
| Year                           | a      | Number          | 1996                |
| Year (of 'Week of Year')       | A      | Number          | 1996                |
| Month in year                  | n      | Text and number | July and 07         |
| Week in year                   | w      | Number          | 27                  |
| Week in month                  | W      | Number          | 2                   |
| Day in year                    | D      | Number          | 189                 |
| Day in month                   | j      | Number          | 10                  |
| Day of week in month           | F      | Number          | 2 (2nd Wed in July) |
| Day of Week (1=first day)      | e      | Number          | 2                   |
| Day in week                    | E      | Text            | Tuesday             |
| a.m. or p.m. marker            | x      | Text            | pm                  |
| Hour in day (1 to 24)          | h      | Number          | 24                  |
| Hour in a.m. or p.m. (0 to 11) | K      | Number          | 0                   |
| Hour in a.m. or p.m. (1 to 12) | k      | Number          | 12                  |
| Hour in day (0 to 23)          | H      | Number          | 0                   |
| Minute in hour                 | m      | Number          | 30                  |
| Second in minute               | s      | Number          | 55                  |

| Meaning             | Symbol | Presentation | Example               |
|---------------------|--------|--------------|-----------------------|
| Millisecond         | S      | Number       | 978                   |
| Time zone           | z      | Text         | Pacific Standard Time |
| Escape used in text | '      | n/a          | n/a                   |
| Single quote        | "      | n/a          | '                     |

### Locale Group C

Locales: ca-es, cs, cs-cz, da, da-dk, es, es-ar, es-bo, es-cl, es-co, es-cr, es-do, es-ec, es-es, es-gt, es-hn, es-mx, es-ni, es-pa, es-pe, es-pr, es-py, es-sv, es-us, es-uy, es-ve, eu-es, mk-mk

| Meaning                   | Symbol | Presentation    | Example             |
|---------------------------|--------|-----------------|---------------------|
| Era                       | G      | Text            | AD                  |
| Year                      | u      | Number          | 1996                |
| Year (of 'Week of Year')  | U      | Number          | 1996                |
| Month in year             | M      | Text and number | July and 07         |
| Week in year              | w      | Number          | 27                  |
| Week in month             | W      | Number          | 2                   |
| Day in year               | D      | Number          | 189                 |
| Day in month              | t      | Number          | 10                  |
| Day of week in month      | F      | Number          | 2 (2nd Wed in July) |
| Day of Week (1=first day) | e      | Number          | 2                   |
| Day in week               | E      | Text            | Tuesday             |
| a.m. or p.m. marker       | a      | Text            | pm                  |
| Hour in day (1 to 24)     | h      | Number          | 24                  |

| Meaning                           | Symbol | Presentation | Example               |
|-----------------------------------|--------|--------------|-----------------------|
| Hour in a.m. or p.m.<br>(0 to 11) | K      | Number       | 0                     |
| Hour in a.m. or p.m.<br>(1 to 12) | k      | Number       | 12                    |
| Hour in day (0 to 23)             | H      | Number       | 0                     |
| Minute in hour                    | m      | Number       | 30                    |
| Second in minute                  | s      | Number       | 55                    |
| Millisecond                       | S      | Number       | 978                   |
| Time zone                         | z      | Text         | Pacific Standard Time |
| Escape used in text               | '      | n/a          | n/a                   |
| Single quote                      | ''     | n/a          | '                     |

## Locale Group D

Locales: de, de-at, de-be, de-ch, de-de, de-lu

| Meaning                  | Symbol | Presentation    | Example             |
|--------------------------|--------|-----------------|---------------------|
| Era                      | G      | Text            | AD                  |
| Year                     | j      | Number          | 1996                |
| Year (of 'Week of Year') | J      | Number          | 1996                |
| Month in year            | M      | Text and number | July and 07         |
| Week in year             | w      | Number          | 27                  |
| Week in month            | W      | Number          | 2                   |
| Day in year              | D      | Number          | 189                 |
| Day in month             | t      | Number          | 10                  |
| Day of week in month     | F      | Number          | 2 (2nd Wed in July) |

| Meaning                        | Symbol | Presentation | Example               |
|--------------------------------|--------|--------------|-----------------------|
| Day of Week (1=first day)      | e      | Number       | 2                     |
| Day in week                    | E      | Text         | Tuesday               |
| a.m. or p.m. marker            | a      | Text         | pm                    |
| Hour in day (1 to 24)          | h      | Number       | 24                    |
| Hour in a.m. or p.m. (0 to 11) | K      | Number       | 0                     |
| Hour in a.m. or p.m. (1 to 12) | k      | Number       | 12                    |
| Hour in day (0 to 23)          | H      | Number       | 0                     |
| Minute in hour                 | m      | Number       | 30                    |
| Second in minute               | s      | Number       | 55                    |
| Millisecond                    | S      | Number       | 978                   |
| Time zone                      | z      | Text         | Pacific Standard Time |
| Escape used in text            | '      | n/a          | n/a                   |
| Single quote                   | "      | n/a          | '                     |

### Locale Group E

Locales: fr, fr-be, fr-ca, fr-ch, fr-fr, fr-lu

| Meaning                  | Symbol | Presentation    | Example     |
|--------------------------|--------|-----------------|-------------|
| Era                      | G      | Text            | AD          |
| Year                     | a      | Number          | 1996        |
| Year (of 'Week of Year') | A      | Number          | 1996        |
| Month in year            | M      | Text and number | July and 07 |

| Meaning                        | Symbol | Presentation | Example               |
|--------------------------------|--------|--------------|-----------------------|
| Week in year                   | w      | Number       | 27                    |
| Week in month                  | W      | Number       | 2                     |
| Day in year                    | D      | Number       | 189                   |
| Day in month                   | j      | Number       | 10                    |
| Day of week in month           | F      | Number       | 2 (2nd Wed in July)   |
| Day of Week (1=first e day)    |        | Number       | 2                     |
| Day in week                    | E      | Text         | Tuesday               |
| a.m. or p.m. marker            | x      | Text         | pm                    |
| Hour in day (1 to 24)          | h      | Number       | 24                    |
| Hour in a.m. or p.m. (0 to 11) | K      | Number       | 0                     |
| Hour in a.m. or p.m. (1 to 12) | k      | Number       | 12                    |
| Hour in day (0 to 23)          | H      | Number       | 0                     |
| Minute in hour                 | m      | Number       | 30                    |
| Second in minute               | s      | Number       | 55                    |
| Millisecond                    | S      | Number       | 978                   |
| Time zone                      | z      | Text         | Pacific Standard Time |
| Escape used in text            | '      | n/a          | n/a                   |
| Single quote                   | "      | n/a          | '                     |

## Locale Group F

Locales: ga-ie

| Meaning                        | Symbol | Presentation    | Example               |
|--------------------------------|--------|-----------------|-----------------------|
| Era                            | R      | Text            | AD                    |
| Year                           | b      | Number          | 1996                  |
| Year (of 'Week of Year')       | B      | Number          | 1996                  |
| Month in year                  | M      | Text and number | July and 07           |
| Week in year                   | t      | Number          | 27                    |
| Week in month                  | T      | Number          | 2                     |
| Day in year                    | l      | Number          | 189                   |
| Day in month                   | L      | Number          | 10                    |
| Day of week in month           | F      | Number          | 2 (2nd Wed in July)   |
| Day of Week (1=first day)      | e      | Number          | 2                     |
| Day in week                    | E      | Text            | Tuesday               |
| a.m. or p.m. marker            | a      | Text            | pm                    |
| Hour in day (1 to 24)          | u      | Number          | 24                    |
| Hour in a.m. or p.m. (0 to 11) | K      | Number          | 0                     |
| Hour in a.m. or p.m. (1 to 12) | k      | Number          | 12                    |
| Hour in day (0 to 23)          | U      | Number          | 0                     |
| Minute in hour                 | n      | Number          | 30                    |
| Second in minute               | s      | Number          | 55                    |
| Millisecond                    | S      | Number          | 978                   |
| Time zone                      | c      | Text            | Pacific Standard Time |

| Meaning             | Symbol | Presentation | Example |
|---------------------|--------|--------------|---------|
| Escape used in text | '      | n/a          | n/a     |
| Single quote        | "      | n/a          | '       |

## Decimal Format Symbols

All locales

| Symbol | Meaning                                                                                                                                                                                           |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0      | A digit that is shown even if the value is zero.                                                                                                                                                  |
| #      | A digit that is suppressed if the value is zero.                                                                                                                                                  |
| .      | A placeholder for decimal separator.                                                                                                                                                              |
| ,      | A placeholder for thousands grouping separator.                                                                                                                                                   |
| E      | Separates mantissa and exponent for exponential formats.                                                                                                                                          |
| ;      | Separates formats for positive numbers and formats for negative numbers.                                                                                                                          |
| -      | The default negative prefix.                                                                                                                                                                      |
| %      | Multiplied by 100, as percentage.                                                                                                                                                                 |
| ‰      | Multiplied by 1000, as per mille.                                                                                                                                                                 |
| ¤      | The currency symbol. If this symbol is present in a pattern, the monetary decimal separator is used instead of the decimal separator.                                                             |
| ¤¤     | The international currency sign. It will be replaced by an international currency symbol. If it is present in a pattern, the monetary decimal separator is used instead of the decimal separator. |
| X      | Other characters that can be used in the prefix or suffix.                                                                                                                                        |
| '      | Used to quote special characters in a prefix or suffix.                                                                                                                                           |
| /u221E | Infinity symbol.                                                                                                                                                                                  |
| /uFFFD | Not a Number symbol.                                                                                                                                                                              |



---

# Chapter 18: Using Query Macros

---

A macro is a fragment of code that you can insert in the Select statement of a query or in an expression. For example, add a macro to insert a new data item containing the user's name.

You can also add query macros to the IBM® Cognos® Framework Manager model. For more information, including examples, see the Framework Manager *User Guide*.

Support for query macros in IBM Cognos Report Studio includes the same capabilities as macros used in Framework Manager, including support for parameter maps defined within the Framework Manager model. However, Report Studio query macros do not extend to the layout. Therefore, when making changes to the query using macros, you must bear in mind the side-effects on the layout. For example, if a macro removes a column from the query that the layout refers to, a run-time error will occur.

## Add a Query Macro

Add a query macro to allow run-time changes to be made to SQL queries.

### Steps

1. Pause the pointer over the query explorer button  and click Queries.
2. If the query to which you are adding a macro was [built by using your own SQL](#), do the following:
  - Click the SQL object that is linked to the query.
  - In the Properties pane, double-click the SQL property.
  - In the SQL dialog box, type the macro.
3. If the query to which you are adding a macro was not built by using your own SQL, do the following:
  - Click the query.
  - In the Properties pane, double-click the Generated SQL/MDX property.
  - In the Generated SQL/MDX dialog box, click Convert.
  - In the SQL dialog box, type the macro.

**Tip:** For a list of supported macro functions, see "[Calculation Components](#)" (p. 769).

## Creating Prompts Using Query Macros

You can create mandatory and optional prompts in reports using query macros. Use the prompt macro functions `prompt` and `promptmany` to create single-value and multiple-value prompts. You

can use prompt macro functions when working with a relational data source or a dimensionally-modeled relational (DMR) data source.

The `prompt` and `promptmany` functions have the following mandatory and optional parameters. All argument values must be specified as strings.

### Name

This mandatory parameter is the name of the prompt. Name can also refer to the name of a parameter on a user-created prompt page, in which case the user-created prompt page appears when the report is run instead of the default prompt page that the macro would generate.

### Datatype

This optional parameter is the prompt value data type. The default value is string. Prompt values are validated. In the case of strings, the provided value is enclosed in single quotation marks and embedded single quotation marks are doubled.

Values include the following:

- boolean
- date
- datetime
- decimal
- double
- float
- int
- integer
- interval
- long
- memberuniquename

Memberuniquename is not an actual data type. This value must be used when the data type of the parameter is member unique name (MUN).

- numeric
- real
- short
- string
- time
- timeinterval
- timestamp

- token
- xsddate
- xsddatetime
- xsddecimal
- xsddouble
- xsdduration
- xsdfloat
- xsdint
- xsdlong
- xsdshort
- xsdstring
- xsdtme

### **DefaultText**

This optional parameter is the text to be used by default. If a value is specified, the prompt is optional.

If you use a space and no values are provided in the **Prompt Value** dialog box, a `Where` clause is usually not generated.

If you use text and no values are provided in the **Prompt Value** dialog box, a `Where` clause is usually generated using the default value.

Ensure that the text you provide results in a valid SQL statement.

**Note:** If the data type is `memberuniqueName`, a value for the `DefaultText` parameter must be provided. For example:

```
(#prompt('WhichLevel', 'memberuniqueName', '[goSalesAgain].[PRODUCT1].[PRODUCT].[PRODUCT(All)]->[all]')#)
```

### **Text**

This optional parameter is text that precedes any user-provided values, such as 'and column1 = '.

### **QueryItem**

This parameter is optional. The prompt engine can take advantage of the **Prompt Info** properties of the query item. Descriptive information can be shown, although the prompt value is a code.

### TextFollowing

This optional parameter is the closing parenthesis that is used most often for the `promptmany` function. This parameter is also useful when the prompt is optional and is followed by hardcoded filters in the SQL statement.

## Example - Select a Country Prompt

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a report that will prompt users to choose the country for which they want to see data.

The following code shows how to use macros to create a prompt:

```
select
 COUNTRY_MULTILINGUAL.COUNTRY_CODE
 as COUNTRY_CODE,
 COUNTRY_MULTILINGUAL.COUNTRY as COUNTRY,
 COUNTRY_MULTILINGUAL."LANGUAGE" as LANGUAGE1,
 COUNTRY_MULTILINGUAL.CURRENCY_NAME
 as CURRENCY_NAME
from
 gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_MULTILINGUAL
where COUNTRY_MULTILINGUAL.COUNTRY = #prompt('CountryName')#
```

### Notes

- Because the `<Datatype>` argument is not specified, valid prompt values must be strings, which is correct in this case.
- Because the `<DefaultText>` argument is not specified, the `CountryName` prompt is a required prompt.

---

# Chapter 19: Using Drill-through Access

---

Using drill-through access, you can move from one report to another within a session while maintaining your focus on the same piece of data. For example, you select a product in a sales report and move to an inventory report about that product.

Drill-through access helps you to build business intelligence applications that are bigger than a single report. Drill-through applications are a network of linked reports that users can navigate, retaining their context and focus, to explore and analyze information.

Drill-through access works by defining the target report or object and then using information from the source report to filter the target. IBM® Cognos® BI can match data from the source to metadata in the target report, or you can define parameters in the target for greater control.

**Note:** In the dimensional reporting style, you can also enable drilling up and drilling down. For more information about drilling up and drilling down, see "[Create a Drill-up and Drill-down Report](#)" (p. 414).

## What You Should Know

For a drill-through link to work, it is necessary to know

- the source
- the target
- how the data in the packages that contain these objects is related

Depending on the underlying data, you may create a drill-through definition and have IBM Cognos BI match the data (dynamic drill-through) or create parameters in the target (parameterized drill-through).

- whether to run the target report or to open it

The target of drill-through access is usually a saved report definition. The report can be created in IBM Cognos Report Studio, IBM Cognos Query Studio, IBM Cognos PowerPlay® Studio, or IBM Cognos Analysis Studio. The target of drill-through access can also be a package that contains an IBM Cognos PowerCube, in which case a default view of the PowerCube is created.

- whether the users of the drill-through link in the source report have the appropriate permissions to view or run the target report
- if the target is being run, in what format to run it, and what filters to use on the target

If you do not want to run the target report on demand, you may link to a bookmark in the saved output instead of running the report. For more information, see the *IBM Cognos Business Intelligence Administration and Security Guide*.

## Sources and Targets

In IBM Cognos BI, there are many different examples of source and target. For example, you can drill through

- between reports created in different packages against different data source types, such as from an analysis against a package that contains a PowerCube to a detailed report against a package based on a relational data source
- from one existing report to another report using definitions created in Report Studio
- between IBM Cognos Viewer reports authored in Report Studio, Query Studio, and Analysis Studio
- to and from a package built on a PowerCube
- from IBM Cognos Series 7 to IBM Cognos BI reports
- from Metric Studio to other IBM Cognos BI reports by passing parameters using URLs

For more information, see the *IBM Cognos Metric Studio User Guide*.

**Tip:** In the GO Data Warehouse (analysis) package, the Positions to Fill sample report is a drill-through target report and the Recruitment Report sample report is a drill-through source report. For more information about The Great Outdoors Company samples, see "["Sample Reports and Packages" \(p. 567\)](#)".

## Understanding Drill-through Concepts

Before you set up drill-through access, you must understand the key concepts about drilling through. Knowing these concepts will help you to avoid errors so that report consumers drill through as efficiently as possible.

### Drill-through Paths

You can create a drill-through path in a source report in Report Studio, or using Drill-through Definitions in IBM® Cognos® Connection. A drill-through path is the definition of the path that is taken when moving from one report to another, including how the data values are passed between the reports.

Using **Drill-through Definitions**, you can create a drill-through path from any report in the source package to any target report in any other package in IBM Cognos Connection. This type of drill-through definition is stored in the source package. Users of any report in the package can use the drill-through definition to drill between any combination of Analysis Studio, Query Studio, PowerPlay® Studio, or IBM Cognos Viewer reports in any package.

For any target report that contains parameters, you should map the target parameters to the correct metadata in the drill-through path. This ensures that the values from the source report are passed to the correct parameter values, and that the target report is filtered correctly. If you do not map parameters, then the users may be prompted for values when the target report is run.

A report-based drill-through path refers to a path created and stored in a Report Studio source report. This type of drill-through path is also called authored drill through. The path is associated

with a specific data column, chart, or cross tab in the source report, and is available only when users select that area of the report. If an authored drill-through definition is available, a hyperlink appears in the source report when it is run.

Report-based drill-through is limited to Report Studio source reports and any target reports. Use this type of drill-through access when you want to pass data item values or parameter results from within a source report to the target report, pass the results of a report expression to a target report, or a use URL link as a part of the drill-through definition.

## Selection Contexts

The selection context represents the structure of the values selected by the user in the source. In Analysis Studio, this includes the context area. When a package drill-through definition is used, the selection context is used to give values for mapped parameters (parameterized drill-through) or also to map the appropriate data items and values.

Drill-through links can also be defined to open the target object at a bookmark. The content of this bookmark may also specified by the selection context.

Drill-through access is possible between most combinations of the IBM® Cognos® Business Intelligence studios. Each studio has been optimized for the goals and skills of the audience that uses it, and in some cases for the type of data source it is designed for. Therefore, you may need to consider how the various studios manage the selection context when you drill through between objects created in different studios, and how the data sources are conformed. During testing or debugging, you can see how source values are being mapped in different contexts using the drill-through assistant.

## Drilling Through to Different Report Formats

The settings in the drill-through definition determine how users see the report results. For example, the users may see the reports in IBM® Cognos® Viewer as an HTML Web page, or the reports may open in Query Studio, PowerPlay® Studio, or Analysis Studio. If your users have PowerPlay Studio, then they may also see the default view of a PowerCube.

Reports can be opened as HTML pages, or as PDF, XML, CSV, or Microsoft® Excel spreadsheet software formats. When you define a drill-through path, you can choose the output format. This can be useful if the expected use of the target report is something other than online viewing. If the report will be printed, output it as PDF; if it will be exported to Excel for further processing, output it as Excel or CSV, and so on.

If you define a drill-through path to a report that is created in Analysis Studio, PowerPlay Studio, or Query Studio, consumers can open the report in its studio instead of in IBM Cognos Viewer. This can be useful if you expect a consumer to use the drill-through target report as the start of an analysis or query session to find more information.

For example, if an application contains a dashboard style report of high-level data, you can define a drill-through link to Analysis Studio to investigate items of interest. The Analysis Studio view can then be drilled through to a PDF report for printing.

**Note:** Report Studio does not display data results.

## Drilling Through Between Packages

You can set up drill-through access between different packages. The two packages can be based on different types of data source, but there are some limits.

The following table shows the data source mappings that support drill-through access.

| Source data source               | Target data source                                                                                                                    |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| OLAP                             | OLAP<br><br><b>Note:</b> OLAP to OLAP drill through is supported only if the data source type is the same, for example, SSAS to SSAS. |
| OLAP                             | Dimensionally modeled relational                                                                                                      |
| OLAP                             | Relational data<br><br><b>Note:</b> For more information, see <a href="#">"Business Keys" (p. 521)</a> .                              |
| Dimensionally modeled relational | Dimensionally modeled relational                                                                                                      |
| Dimensionally modeled relational | Relational                                                                                                                            |
| Relational                       | Relational                                                                                                                            |

## Bookmark References

When you drill through, the values that you pass are usually, but not always, used to filter the report. IBM® Cognos® Business Intelligence supports bookmarks within saved PDF and HTML reports so that a user can scroll a report to view the relevant part based on a URL parameter.

For example, you have a large inventory report scheduled to run daily or weekly during off hours because of resource considerations. Your users may want to view this report as a target because it contains detailed information, but you want them to view the saved output rather than run this large report. Using this Action option and bookmark settings, users can drill through from another source location based on products to open the saved report to the page that shows the product they want to focus on.

When a bookmark in the source report is used in a drill-through definition, it provides the value for the URL parameter. When report consumers drill through using this definition, they see the relevant section of the target report.

Bookmark references are limited to previously run reports that are output as PDF or HTML and contain bookmark objects.

## Members and Values

Dimensionally modeled data, whether stored in cubes or stored as dimensionally-modeled relational (DMR) data, organizes data into dimensions. These dimensions contain hierarchies. The hierarchies contain levels. And the levels contain members.

An example of a dimension is Locations. A Locations dimension may contain two hierarchies: Locations by Organization Structure and Locations by Geography. Either of these hierarchies may contain levels like Country and City.

Members are the instances in a level. For example, New York and London are members in the City level. A member may have multiple properties, such as Population, Latitude, and Longitude. Internally, a member is identified by a Member Unique Name (MUN). For more information, see . The method by which a MUN is derived depends on the cube vendor.

Relational data models are made up of data subjects, such as Employees, which are made up of data items, such as Name or Extension. These data items have values, such as Peter Smith.

In IBM® Cognos® Business Intelligence, the methods of drilling through available are

- Dimensional (member) to Dimensional (member)
- Dimensional (member) to Relational (data item value)
- Relational (data item value) to Relational (data item value)

If the target parameter is a member, the source must be a member. The source and target should usually be from a conformed dimension. For more information, see "[Conformed Dimensions](#)" (p. 520). However, if the data will support it, you may also choose to define a mapping using different properties of the source metadata item.

If the target parameter is a value, the source can be either a value or a member. If the source is a dimensional member, you must ensure that the level or dimension is mapped to the target data item correctly in the drill-through definition. The business key from which the member is sourced should usually match the relational target value, which is most often the business key. For more information, see "[Business Keys](#)" (p. 521). However, if the data will support it, you may also choose to define a mapping from the caption of the source metadata item.

## Member Unique Names

The member unique name (MUN) is a unique identifier for a member in IBM® Cognos® reports. It is stored in the report specification when the member is referenced in the report directly. The MUN is used in drill-through between OLAP data sources. The member keys in the MUN for the different OLAP data sources must match.

The MUN is used to find the member in the data source, which is similar to how business keys are used to find records in a table. For example, when you create OLAP dimension Products, you use the Product Line database column as a label for the members in your Product Line level. However, you use the Product Line Code business key from the database table to ensure that all the Product lines are unique in that level. The source value that you used to create the members is used in combination with the data source name, hierarchy, and level information in the member unique name.

If the MUN changes, members that are directly referenced in expressions, filters, or reports are no longer found. Changes to the MUN may be related to other changes. For example, changes to the hierarchy and level structures may change the level unique name, and changes to the business key values may change the member key path. Other factors that can affect the MUN are application changes during the design stage or over time, IBM Cognos PowerCube category codes that are unpredictably unique, the production environment that has more members than the test environment, or removing the member from the data source.

To avoid potential problems, we recommend the following best practices when you build OLAP data sources:

- Use unique codes and keys within a dimension for the member keys.
- Define your OLAP and relational packages using unique conformed values for the source values (business keys) within similar dimensions or data values where drill-through between applications may be required.
- Ensure that the business keys and dimension metadata structure are the same in the production and test environments.
- Do not change the business keys in Framework Manager in the production environment.
- Resolve the non-unique keys in a dimension in the data source before you build the cube.

Ensure that there are no duplicate source values in all levels of a dimension before you build a PowerCube. We do not recommend using the tilde character (~) in the category codes.

For more information, see the section about uniqueness in the IBM Cognos Series 7 *Step-by-Step Transformer*.

For information about PowerCubes migrated from IBM Cognos Series 7, see the IBM Cognos PowerPlay® *Migration and Administration Guide* or the *Migration Assistant User Guide*.

## Conformed Dimensions

If you work with more than one dimensional data source, you may notice that some dimensions are structured the same, and some are not. The reason that dimensions can be structured differently is that the data sources may serve different purposes.

For example, a Customer dimension appears in a Revenue data store, but not in an Inventory data store. However, the Products dimension and the Time dimension appear in both data stores.

Dimensions that appear in multiple data stores are conformed if their structure is identical for all of the following:

- hierarchy names
- level names
- level order
- internal keys

Drilling through is possible between different dimensional data stores only if the dimensions are conformed, and if the dimension data store is of the same vendor type, such as IBM® Cognos®

PowerCube as the source and the target. For example, in two data stores for Revenue and Inventory that contain Products and Time dimensions, it is possible to define the Products and Time dimensions differently for each data store. However, for drill-through between the Products and Time dimensions to work, their structures must be identical in each data store.

If you are not sure whether your dimensions are conformed, then you should check with the data modeler to ensure that the drilling through will produce meaningful results.

IBM Cognos Business Intelligence does not support conformed dimensions generated by Framework Manager for SAP BW data sources.

### **Dimensionally-modeled Relational Data Sources**

Ensure that each level contains a business key that has values that match your PowerCube or other DMR models. Also, you must also ensure that the **Root Business Key** property is set and uses the business key of the first level in the hierarchy. This helps to ensure that you have a conformed member unique name when attempting to drill through using members from this dimension.

## **Business Keys**

When drill-through access is defined from a member to a relational value, the business key of the member is passed by default. This means that your relational target parameter must be set up using the data item with a matching value, which is most often the business key data item. You can also choose to pass the caption of the source metadata item.

For example, employees are usually uniquely identified by an employee number, not by their name, because their name is not necessarily unique. When you drill through from a dimensional member to a relational data item, the value provided is the business key. Therefore, the parameter in the target report must be defined to accept a business key value. The exact logic used to define the business key value supplied depends on the cube vendor. For IBM® Cognos® PowerCubes, the business key value is the **Source** property defined for the level in IBM® Cognos® Transformer. IBM Cognos Series 7 Transformer PowerCubes pass the source value if the drill-through flag was enabled before the cube was built. Otherwise, the category code is used.

In Report Studio, you can determine what the member business key is using an expression such as `roleValue('_businessKey', [Camping Equipment])`. This expression is case sensitive.

SSAS 2005 multi-part business keys are not supported in drill-through operations.

**Tip:** When other users run your drill-through report, you may not want them to be prompted for a business key. In Report Studio, you can build a prompt page with a text that is familiar to the users, but filters on the business key. Your Framework Manager modeler can also set the **Display Item Reference** option for the **Prompt Info** property to use the business key when the data item is used in a prompt.

## **Scope**

Scope is specific to drill-through definitions created using Drill-through Definitions in IBM Cognos Connection (package drill-through definitions). The scope you set defines when the target report is shown to the users, based on the items they have in the source report.

Usually, you define the scope of a drill-through path to match a parameter that it passes. For example, if a target report contains a list of employees, typically you only want to display the report as an available drill-through choice when a user is viewing employee names in a source report. If employee names are not in the source report and the scope was set on the employee name in the drill-through definition, the employee report does not appear on the list of available drill-through target reports in the **Go To** page. You can set the scope to a measure or to an item in the report.

In report-based drill-through access, where the drill-through path is associated with a specific report column, the column serves as the scope.

## Mapped Parameters

Drill-through targets may contain existing parameters, or you may choose to add parameters to the target for greater control over the drill-through link. You usually map all parameters in a drill-through target to items from the source.

When you map source items that are OLAP or DMR members to target parameters, you can select from a set of related member properties to satisfy the requirements of the target parameter. For a dimensional target, a dimensional source item uses the member unique name by default. For a relational target, a dimensional source item uses the business key by default.

For example, you could change the source member property that is used for a mapping to the member caption instead of the business key to match the parameter in a relational target. For a dimensional target, you could define a parameter that accepts a particular property (such as business key or parent unique name), then pass the appropriate source property to satisfy that target.

Note that if you define drill through between non-conformed dimensions, you should test carefully to ensure that the results behave as expected.

If you do not specify parameter mappings, then by default, you will be prompted for any parameters required in the target when you use the drill-through link. To customize this behavior, use the display prompt pages setting.

When the action is set to **Run using dynamic filtering**, then additional filtering is applied if names from the context in the source report match names of items in the target. Use this action as well when there are no parameters defined in the target.

If parameters are not mapped correctly, then you may receive an empty report, the wrong results, or an error message.

The source and target cannot contain identical parameter names when they are from different packages, even if the data structure is conformed. If the source and target are from the same package, there is no restriction.

If you have the necessary permissions, you can use the drill-through assistant to look at what source parameters are passed, and what target parameters are mapped for a given drill-through link.

You can change the dynamic drill-through filter behaviour if you want drill-through to generate a filter using the Member Business Key instead of the default Member Caption. For more information, see *Changing Drill-Through Filter Behavior* in the *IBM® Cognos® Administration and Security Guide*.

## Drilling Through on Dates Between PowerCubes and Relational Packages

Usually, drilling through from OLAP to relational packages requires that the target report parameter is set using the business key in the relational data. However, this method does not work well for dates. OLAP data sources typically view dates as members, such as Quarter 1 2006, while relational data sources view dates as ranges, such as 1/Jan/2006 to 31/March/2006.

A special feature exists for drilling through between PowerCubes and relational packages. Ensure that the target report parameter is set up using `in_range`. Note that the parameter must be of type date-time, and not integer.

An example follows:

```
[gosales_goretailers].[Orders].[Order date] in_range
?Date?
```

Also ensure that the drill-through definition maps the parameter at the dimension level and that the PowerCube date level is not set to suppress blank categories. Enabling the option to suppress blank categories in the Transformer model before you build the cube may cause the drill-through on dates to be unsuccessful. This happens because there are missing values in the range.

## Set Up Drill-through Access in a Report

Set up drill-through access in a source report to link two reports containing related information. You can then access related or more detailed information in one report (the target) by selecting one or more data item values from another report (the source). IBM® Cognos® passes values from the source to the target and uses the passed values to filter the target object. If the data in the source and target is conformed or if the data item names are the same, then the system may map the source values to filter the target (dynamic drill-through). If you require greater control, you may define parameters in the target object (parameterized drill-through). For instructions about creating parameters in other kinds of targets, see the *IBM Cognos Business Intelligence Administration and Security Guide*.

When you define the drill-through path in IBM Cognos Report Studio, you can pass a value from a different data item that is hidden from the user for display purposes but still in the query. For example, users see the Product Name data item and can drill through on that item, but the drill-through definition passes the Product Number value for the product name the user chose. You can also define the target parameter to accept multiple values or a range of values from the data item in the drill-through source report or to accept a value from a parameter in the source report.

Drill-through definitions that have been authored inside a Report Studio report appear in the source report as blue underlined links. Users click the hyperlink to select the value they want passed to the target and to drill through to the target report. If a user selects multiple items within a single column then when the drill-through target report is run, the values from each selected row are passed to the target parameters. This occurs as an *and* condition.

You can also drill through within the same report by creating bookmarks. As well, you can create drill-through definitions in the source package instead of the Report Studio report. Users can use package drill-through definitions to navigate to a target report from an IBM Cognos Analysis Studio analysis, an IBM Cognos Query Studio report, an IBM Cognos PowerPlay Studio report or IBM

Cognos PowerCube package, or a Report Studio report. For more information, see the *Administration and Security Guide*.

If you have the IBM Cognos Software Development Kit, you can use URLs to set up drill-through access to and from third-party sources.

You can also drill through from a map.

If you are using an SAP BW data source for the target report and the target report contains a variable for a hierarchy node, values from the source report can be values only of the data item representing the leaf-level identifier of the hierarchy.

Before you begin, ensure that you have a report that will serve as the source report and another report that will serve as the target report.

### Steps to Create a Parameter in the Target

1. In Report Studio, open the target report.
2. Create a parameter that will serve as the drill-through column or that will filter the report.

For example, to drill through or filter Product line, create the following parameter:

[Product line]=?prodlne\_p?

**Tip:** Use the operators `in` or `in_range` to enable the target report to accept multiple values or a range of values.

3. In the **Usage** box, specify what to do when a value for the target parameter is not passed as part of a drill through:
    - To specify that users must select a value, click **Required**.  
If a value for the target parameter is not passed, users are prompted to choose a value.
    - To specify that users do not need to select a value, click **Optional**.  
Users are not prompted to choose a value and so the value is unfiltered.
    - To specify not to use the parameter, click **Disabled**.  
The parameter is not used during the drill-through. It will also not be used in the report for any other purposes.
- Tip:** If the parameter is needed in the report for other reasons, then you can also specify not to use it in the drill-through definition (**Parameters** table, **Method**, **Do not use parameter**).

### Steps to Create the Drill-through Definition

1. Check the drill-through target:
  - Confirm that the drill-through users have access to the target.
  - If necessary, check what parameters exist in the target.
2. Open the source report.

3. Click the element in the report that will serve as the starting point for the drill-through link.  
You can select a data item that your report users are likely to choose to drill on for more detailed information, such as an Employee Name data item.

**Tip:** If you are passing only parameter values to the target report, you do not have to drill on a data item. Instead, you can drill on any object in the report, such as the report title. Therefore, you can drill from outside the context of a query.

4. Click the drill-through definitions button, or, from the Properties pane, double-click the **Drill-Through Definitions** property.

5. Click the new drill-through definition button.

A drill-through definition is created.

**Tip:** To change the drill-through name, click the rename button, type the new name, and click **OK**.

6. On the **Target Report** tab, click the ellipsis (...) button next to the **Report** box and select the drill-through target report.

7. In the **Action** box, decide how users will view the target report when they click the drill-through link in the parent report:

- To view the latest data in IBM Cognos Viewer, select **Run the report**.
- To edit a Query Studio, Analysis Studio, or PowerPlay Studio target, select **Edit the report**.
- To have IBM Cognos BI match values from the selection context with data in the target (dynamic filtering), select **Run the report using dynamic filtering**.

**Note:** If you chose this option, any parameters in the target report are still used for the drill-through access.

- To view the most recently saved output version of the target report from IBM Cognos Connection, select **View the most recent report**.
- To use the default action specified for the report in IBM Cognos Connection, select **(Default)**.

8. If the target report contains parameters, then, under the **Parameters** box, click the edit button.

Each required and optional parameter defined in the target report appears in the **Parameters** dialog box.

9. For each parameter, specify the **Method of treatment**:

- To specify not to pass any value, click **(Default)**.
- To specify not to use this parameter, click **Do not use parameter**.

The target report will not be filtered by this parameter.

- To pass values from a data item, click **Pass data item value**, click **Value**, and then click the data item.

Values for the selected data item are passed to the target report.

- To pass values from a source report parameter, click **Pass parameter value**, click **Value**, and then click the parameter.  
Values for the selected parameter are passed to the target report.
- If you choose to pass parameters, and the source report is based on a dimensional package, then click an item from the drop-down list in the **Source metadata item properties** column. You can pass the default property of the source metadata item (business key) or another property, such as the member caption.

10. Click **OK**.

11. If you chose to run the target report, in the **Format** box, click the output format for your report.

**Tip:** Click **(Default)** to run the report using the default format specified for the report in IBM Cognos Connection.

12. To open the target report in a new window, select the **Open in new window** check box.

13. In the **Display prompt pages** box, choose how to display prompt pages:

- To always display prompt pages when the drill-through action occurs regardless of whether values were specified for the required parameter values, click **Always**.
- To use the prompt settings of the target report (specified by the **Prompt for Values** check box in IBM Cognos Connection, **Report Properties**, **Report** tab) to determine whether to show the prompt pages, click **Based on the default prompt settings of the target report**.
- To not display prompt pages when the required parameter values are provided by the drill-through definition, click **Only when required parameter values are missing**. This is the default setting.

The drill-through text appears as a blue hyperlink in text items in the non-chart areas of the report. Report consumers can also start the drill-through action by clicking the **Go To** button or by right-clicking the item and clicking **Go To, Related links**. If you have the necessary permissions, you can view which parameters were passed from the source and how they are mapped in the target object from the **Go To** page using the drill-through assistant.

For more information, see the *Administration and Security Guide*.

## Specify the Drill-through Text

You can specify the drill-through text that appears when users can drill through to more than one target. For example, if users from different regions view the report, you can show text in a different language for each region.

### Steps

1. Right-click the drill-through object and click **Drill-Through Definitions**.
2. If more than one drill-through definition exists for the object, in the **Drill-Through Definitions** box, click a drill-through definition.

3. Click the **Label** tab.
4. To link the label to a condition, in the **Condition** box, do the following:
  - Click **Variable** and click an existing variable or create a new one.
  - Click **Value** and click one of the possible values for the variable.
5. In the **Source type** box, click the source type to use.
6. If the source type is **Text**, click the ellipsis (...) button beside the **Text** box and type text.
7. If the source type is **Data Item Value** or **Data Item Label**, click **Data Item** and click a data item.
8. If the source type is **Report Expression**, click the ellipsis (...) button beside the **Report Expression** box and define the expression.
9. If the label is linked to a condition, repeat steps 5 to 8 for the remaining possible values.

When users run the source report and click a drill-through link, the **Go to** page appears. The drill-through text you specified appears for each target. If you did not specify the drill-through text for a target, the drill-through name is used.

## Example - Drill Through to a Hidden Report from a Report Studio Report

You want to set up a drill-through link from an employee satisfaction report created in IBM® Cognos® Report Studio to a hidden list report about compensation, also created in Report Studio.

The source report (**Employee Satisfaction 2006**) is based on the package GO Data Warehouse (analysis) which is modeled on a DMR data source. The target report (**Compensation (hidden)**) is based on the package GO Data Warehouse (query). You set up this drill-through connection from within Report Studio (report-based, or authored drill through) because you do not want to make a report about compensation available for drill through from any source report in the package. The target report is already hidden in the portal, so that it is unlikely to be run by anyone who does not use the drill through link.

You must have the IBM Cognos Business Intelligence samples from the deployment zip file Cognos\_DrillThroughSamples installed to follow this exercise, and you must have access to Report Studio.

The **Compensation** report is a hidden report. You may be able to set whether hidden reports are visible (**My Preferences**, **General** tab) and whether you can hide reports. This capability is set by your administrator.

### Check the Target Report

The procedure to check the target report is as follows.

#### Steps to Check the Target Report

1. Open the target report:

- In IBM Cognos Connection, go to Public Folders, GO Data Warehouse (query), Report Studio Report Samples.
  - Locate the report Compensation (hidden) and open it in Report Studio.  
**Tip:** If you do not see the report, go to IBM Cognos Connection and confirm that you can view hidden reports (My Preferences, General tab).
2. In Report Studio, from the **Data** menu, click **Filters** and check what filter parameters are available.  
You want to filter from the source report on department, not time, so you will only use the **pPosition** parameter in the drill-through definition.
3. In the report body, select the list column body **Position-department (level 3)** and review the data item properties.  
Because the drill-through definition goes from DMR to relational, the data item values will need to match.
4. Close the Compensation (hidden) report.

## Create and Test the Drill-Through Definition

The procedure to create and test the drill-through definition is as follows.

### Steps to Create and Test the Drill-Through Definition

1. Open the source report:
  - In IBM Cognos Connection, go to Public Folders, GO Data Warehouse (analysis), Report Studio Report Samples.
  - Locate the Employee Satisfaction 2006 report and open it in Report Studio.
2. Save the Employee Satisfaction 2006 report with a new name, such as Employee Satisfaction 2006 New.  
This is to keep the original report and drill-through definition intact for comparison.
3. In the table **Employee rankings and terminations by department**, select the column **Position-department (level 3)**.
4. In the properties pane, review the data item properties, to confirm that the data item names match values in the target report.
5. In the properties pane, under **Data**, double-click **Drill-through definitions**.
6. Select the definition **DrilltoHiddenRep** and delete it.  
**Note:** In the following steps, you recreate the drill-through definition. For comparison, use the original sample report.
7. In the **Drill-through Definitions** box, click the new drill-through definition button.
8. Click the rename button, and type a name for the drill-through definition.

**Tip:** This is the name that consumers see when they select from a list of possible drill-through definitions in the final report.

9. In the **Target Report** tab, select the target report:

- Under **Report**, click the ellipsis button (...).
- Navigate to **GO Data Warehouse (query)**, **Report Studio Report Samples**, and select the **Compensation (hidden)** report.

**Tip:** If you do not see the report, go to IBM Cognos Connection and confirm that you can see hidden reports (**My Preferences**, **General** tab).

10. Under **Action**, select **Run the report**.

11. Under **Parameters**, click the edit button.

A table of parameters available in the target report appears, showing the parameter **pPosition**.

12. Map the parameter from the **Compensation (Hidden)** report to the metadata in the **Employee Satisfaction 2006** report:

- In the **Method** column, select **Pass data item value**, because the target report is based on a relational data source.
- In the **Value** column, select **Position-department (level 3)**.

**Tip:** In this report, you pass values from the column where the drill-through is defined. In other cases, you might pass a related parameter. For example, you could drill through on employee name, but pass the employee number.

13. Save the report.

14. Run the report, and click a department to test the drill-through definition.

When you test the drill-through link, the **Compensation (hidden)** report appears, filtered by the department you selected. The report appears as a drill-through target whether or not it is hidden in IBM Cognos Connection.

If your administrator has given you the **Drill Through Assistant** capability, then you can see additional information you right-click on the link and select **Go To** see a list of drill-through targets.

From the **Go To** page, you can see what source values are passed, and what target parameters are mapped.

## Try It Yourself - Create a Report with Drill-through Access to Itself

Create a report that drills to itself so that users can view detailed information in the same report.

In this topic, you learn how to create a report that shows revenue by each retailer. Users can access detailed order information for a particular retailer.

It should take 20-25 minutes to complete this topic, and your report will look like this.

**Order Details for All Retailers**

| Retailer name                                       | Revenue       |
|-----------------------------------------------------|---------------|
| <a href="#">1 for 1 Sports shop</a>                 | 6,432,250.32  |
| <a href="#">4.Golf only</a>                         | 5,015,375.54  |
| <a href="#">4 Your Eves</a>                         | 873,022.36    |
| <a href="#">Aarhus Sport</a>                        | 5,999,906.89  |
| <a href="#">Accapamento</a>                         | 6,077,377.8   |
| <a href="#">Accesorios Importados, S.A. de C.V.</a> | 7,945,373.51  |
| <a href="#">AcquaVerde</a>                          | 12,696,927.89 |
| <a href="#">ActiForme</a>                           | 2,005,590.67  |
| <a href="#">Action Factory</a>                      | 8,613,520.15  |
| <a href="#">Act'NUp Fitness</a>                     | 13,661,733.09 |
| <a href="#">Acute Sight</a>                         | 1,662,290.71  |
| <a href="#">Advanced Climbing Ltd</a>               | 1,008,869.51  |
| <a href="#">Air Frais</a>                           | 3,225,980.69  |
| <a href="#">Air marin</a>                           | 8,248,477.5   |
| <a href="#">Aktiv Markt</a>                         | 16,262,466.37 |
| <a href="#">Aktive Freizeit</a>                     | 558,286.8     |
| <a href="#">Aktive Markt</a>                        | 12,013,789.68 |
| <a href="#">Algonquin Camping Supplies</a>          | 12,105,107.18 |
| <a href="#">All Season Camping Goods</a>            | 6,133,822.91  |
| <a href="#">All'aperto esperienza</a>               | 6,756,555.8   |

Select a retailer in the list on the left to view order details below

| Order number | Date | Product name | Quantity | Revenue |
|--------------|------|--------------|----------|---------|
|--------------|------|--------------|----------|---------|

[☶ Top](#) [⤵ Page up](#) [⤵ Page down](#) [⤷ Bottom](#)

## Steps to Create the Report and Add Data

1. Create a new blank report that uses the **GO Data Warehouse (query)** package.
  2. Add a table with two columns and one row.
  3. Add a list object to each column in the table.
  4. Add these data items to the first list:
    - **Retailer name** (in **Retailer**)
    - **Revenue** (in **Sales fact**)
- Tip:** Use the **Source** tab in the **Insertable Objects** pane.
5. Set the sort order for the **Retailer name** column to **Sort Ascending**.
  6. Add **Retailer name** in **Retailer** to the second list.
  7. Add these data items to the second list:
    - **Order number** in **Sales order**
    - **Date** in **Time Dimension**
    - **Product name** in **Product**
    - **Quantity** in **Sales fact**
    - **Revenue** in **Sales fact**

**Tip:** To simultaneously add all the data items to the list, Ctrl+click the items before dragging them to the list.

8. In the second list, click the **Retailer name** column and click the cut button.

9. In the second list, create the following parameterized filter:  
**[Retailer name]=?Selected retailer?**
10. Save the report.

### Steps to Set up Drill-through Access

1. In the left list, right-click a **Retailer name** column (and not the column title) and click **Drill-through Definitions**.
2. Create a new drill-through definition.
3. Specify the report as the target report.
4. Under **Parameters**, click the edit button.
5. Set the **Selected retailer** parameter to pass data item values using the **Retailer name** data item.

### Steps to Add a Prompt

1. In the **Toolbox** tab, add a value prompt under the table.

Because you are setting up drill-through access to the same report, you must add a prompt with a default value so that users are not prompted when they run the report.

2. In the **Prompt Wizard**, specify that the prompt is to use the existing parameter named **Selected retailer**, and click **Finish**.
3. Click the value prompt.
4. In the **Properties** pane, double-click the **Default Selections** property and add a simple selection named **NoRetailer**.
5. Set the following properties:
  - **Required to No**
  - **Hide Adornments to Yes**
  - **Visible to No**

### Steps to Add a Report Title

1. Double-click the report title and type the following text, adding a blank space at the end:  
**Order Details for**  
**if(ParamDisplayValue('Selected retailer')='NoRetailer') then 'All Retailers' else (ParamDisplayValue('Selected retailer'))**
2. In the **Toolbox** tab, add the following layout calculation to the right of the report title:

**if(ParamDisplayValue('Selected retailer')='NoRetailer') then 'All Retailers' else (ParamDisplayValue('Selected retailer'))**

When the report is run, the report title changes to reflect the retailer selected by the user. If no retailer is selected, 'All Retailers' appears.

3. Click the layout calculation and, in the Properties pane, set the class to **Report title text**.

### Steps to Add Conditional Formatting

1. Pause the pointer over the condition explorer button and click **Variables**.

Use conditional formatting to show a message above the second list when users have not selected a retailer from the first list.

2. Create this Boolean variable named **HighLevel**:

```
ParamDisplayValue('Selected retailer')<>'NoRetailer'
```

3. Pause the pointer over the page explorer button and click **Page1**.

4. In the **Toolbox** tab, add a block above the second list.

5. Insert a text item in the block with the following text:

Select a retailer in the left list to view order details below

6. Set the font style for the text to bold.

7. Click the text item and, in the **Properties** pane, set the **Style Variable** property to the **HighLevel** variable.

8. Pause the pointer over the condition explorer and click the **Yes** value for the **HighLevel** variable.

9. Set the **Visible** property for the text item to **No**.

10. Triple-click the explorer bar to turn off conditional formatting.

11. Run the report to view what it will look like for your users.

A list appears on the left that shows revenue for each retailer. When users click a retailer, order information for the selected retailer appears in the second list.

### Need More Help?

- [Use a Table to Control Where Objects Appear](#)
- [Add Relational Data to a Report](#)
- [Create a Parameter to Produce a Prompt](#)
- [Set Up Drill-through Access in a Report](#)
- [Create a Prompt Directly in a Report Page](#)
- [Using Relational Calculations](#)
- [Calculation Components](#)
- [Add a Variable](#)
- [Hide or Show an Object](#)

## Drilling Through from IBM Cognos Series 7 to IBM Cognos BI

You can set up drill-through access from IBM® Cognos® Series 7 to IBM Cognos BI. Specifically, you can drill through to an IBM Cognos BI report from an IBM Cognos PowerPlay® Web report or IBM Cognos Visualizer report. Drill through is supported for both PowerCubes and other cubes.

You must complete steps in both IBM Cognos Series 7 and IBM Cognos BI to enable drill through. Refer to the IBM Cognos Series 7 documentation for the specific steps you must complete in the IBM Cognos Series 7 components. Cross references to the appropriate IBM Cognos Series 7 documents are included below.

### Setting Up Drill-through Access from IBM Cognos Visualizer

Setting up drill-through access from IBM® Cognos® Visualizer to IBM Cognos Business Intelligence involves setting up the target report.

To set up drill-through access, you must do the following:

- specifying the IBM Cognos BI target and selecting the filters to add to the target report

You must configure drill through to IBM Cognos BI for individual IBM Cognos Visualizer reports. For more information, see the *IBM Cognos Visualizer User Guide*.

- creating and testing the target report

For more information, see ["Create and Test the Target for a Series 7 Report" \(p. 534\)](#).

### Setting Up Drill-through Access from PowerPlay Web

Setting up drill-through access from PowerPlay® Web to IBM® Cognos® Business Intelligence involves setting up target reports.

To set up drill-through access, you must do the following:

- for PowerCubes, specifying drill-through targets for IBM Cognos BI reports in the Transformer model

For more information, see the Transformer documentation.

- for other cubes, specifying drill-through targets for IBM Cognos BI reports in PowerPlay Connect.

For more information, see the PowerPlay OLAP Server Connection Guide.

- configuring drill-through access in PowerPlay Server Administration

In addition to enabling drill-through access to IBM Cognos BI, you must specify the location of the IBM Cognos BI server and the IBM Cognos BI folder that contains the target reports.

For more information, see the *PowerPlay Enterprise Server Guide*.

- selecting the filters to add to the target report

In PowerPlay Enterprise Server Administration, enable and use IBM Cognos BI Assistance to identify the filter expressions required in the target report. For more information, see the *PowerPlay Enterprise Server Guide*.

- creating and testing the target report

For more information, see "[Create and Test the Target for a Series 7 Report](#)" (p. 534).

## Create and Test the Target for a Series 7 Report

The target report must be based on a published package that contains the metadata items that you want to filter on, or contains items that are mapped to those metadata items.

When you create the target report, ensure that the names of the parameters you add are identical to the parameter names listed in the **Drill Through Assistant** page in IBM® Cognos® Series 7.

However, the metadata item that you use in the target report for that parameter name does not have to be the identical label. The data values between the target parameter and the source value shown in the drill assistant must match. You may also need to change the type of operator in the target parameter from what is recommended in the **Drill Through Assistant**. For example, if the assistant recommends an = operator but you want to pass a date range, you should change the parameter operator in the target to in\_range.

### Steps

1. Start Report Studio and create a new report.
2. Add the data items and other objects you want.
3. From the **Data** menu, click **Filters**.
4. In the **Detail Filters** tab, click the add button.
5. In the **Expression Definition** box, create the parameterized filter you want by typing the filter expression.
6. Click **OK**.
7. In the **Usage** box, click **Optional**.

If you do not make the filter optional, a prompt page appears when you drill through to the report.

8. Repeat steps 4 to 7 for other parameterized filters you want to add.
9. Save the report.

The report name must match what you specified as a target in the PowerCube, other cube, or IBM Cognos Visualizer report.

10. Test the drill through in the PowerPlay® report or IBM Cognos Visualizer report.

# Chapter 20: Working with Multiple Pages

---

Divide a report into multiple pages and add navigation elements to make it easier to use.

Working with multiple pages includes

- [adding a page to a report](#)
- [creating page breaks and page sets](#)
- [joining nested page sets](#)
- [adding bookmarks](#)
- [creating a table of contents](#)
- [inserting page numbers](#)
- [controlling page breaks and page numbering](#)
- [creating page layers](#)

**Note:** The pages that you see in the Page Explorer differ from physical, printed pages. For example, if you create page breaks in a long report, the Page Explorer could show only one page, but your report could include several physical pages when you print it.

## Add a Page to a Report

Reports may contain pages with a variety of content. Pages may be added to create a multiple-page report. For example, you are creating a sales report and you want to include general information about the company. You decide to create an introduction page to show this information. When you run the report, page 1 of the report is the introduction page and the following pages contain data.

**Tip:** The TOC Report sample report in the GO Data Warehouse (query) package includes multiple pages. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

You can also [create page sets](#) to control the rendering of pages in a report.

### Steps

1. Pause the pointer over the page explorer button  and click **Report Pages**.

**Tip:** Click **Prompt Pages** to add a prompt page.

2. In the **Insertable Objects** pane, on the **Toolbox** tab , drag **Page** to the work area.
3. To change the order of pages, drag pages above or below other pages.
4. To associate a query to the page, in the **Properties** pane, set the **Query** property to the query.

Associate the page to a query to add data-related objects directly to the page header, body, or footer, such as data items and filters.

5. Double-click the page to open it in page design view.
6. Add objects to the page.

## Create a Page Break or Page Set

You can create simple page breaks, or you can create advanced page sets to associate pages with a query structure to force page breaks. For example, a query contains the data item Product line, which is grouped. Creating a page set that is associated with this query adds page breaks for each product line. When working with dimensional data, you can use the **Page layers** area to show values on a separate page for each member.

**Tip:** The Global Bonus Report sample report in the GO Data Warehouse (analysis) package includes page sets. For more information about The Great Outdoors Company samples, see "["Sample Reports and Packages" \(p. 567\)](#)".

### Steps to Create Simple Page Breaks

1. Click the data item at which the report should break to a new page.
2. From the **Structure** menu, do one of the following:
  - To create a page break without creating a master detail relationship, click **Set Page Break**.  
This option only works if the data item is in a list or repeater. If the data item is in a crosstab or chart, IBM® Cognos® Report Studio creates a page break using a master detail relationship. Setting page breaks without using master detail relationships can improve performance when running the report.
  - To create a page break using a master detail relationship, click **Set Page Break Using Master/Detail**.

Report Studio confirms the page break setting.

**Tip:** To modify simple page break settings, pause the pointer over the page explorer button  and click **Report Pages**, and then select and modify the page set.

### Steps to Create Advanced Page Sets

1. Pause the pointer over the page explorer button  and click **Report Pages**.
2. Create the report pages.
3. In the **Insertable Objects** pane, on the **Toolbox** tab , drag the **Page Set** object to the **Report Pages** pane.
4. In the **Properties** pane, set the **Query** property to the query to associate with the page set.
5. Organize the pages in the report by dragging report pages to page sets.

6. Insert the page containing details in the **Detail Pages** folder.

Detail pages are the pages that repeat based on the items by which you group in the following step.

**Tip:** You can insert multiple detail pages into the same page set and link them using a master detail relationship.

7. Define the grouping structure for the page set:

- Click the page set.
- In the **Properties** pane, double-click the **Grouping & Sorting** property.
- In the **Data Items** pane, drag the data item by which to group data items to the **Groups** folder in the **Groups** pane.
- To sort the data within each group, in the **Data Items** pane, drag the data item by which to sort to the **Detail Sort List** folder, and then click the sort order button  to specify the sort order. For more information about sorting data, see "["Sorting Relational Data" \(p. 328\)](#)" or "["Sorting Dimensional Data" \(p. 375\)](#)".

8. Repeat steps 3 to 7 to create other page sets.

**Tip:** You can nest page sets and join them by defining a master detail relationship. Create nested page sets to have pages occur within other pages. For example, you want pages containing product type information to occur within pages containing product line information.

**Note:** Grouping an item for a page set is not the same as grouping a column in the layout. Grouping a column in the layout visually shows groups in a report. Grouping an item for a page set groups the item in the query. If you want to use an item that is already grouped in the layout, you must still perform step 7.

## Join Nested Page Sets

If you have nested page sets in your report, define a master detail relationship between them to see data in the nested page set that is related to the data in the parent page set. For example, you have a page set that shows pages of product line information. The page set contains a nested page set that shows pages of product type information. For each product line page, you want to see the related product type pages, as shown below:

Product line\_1

- Product type\_1
- Product type\_2
- Product type\_3

Product line\_2

- Product type\_4
- Product type\_5

- Product type\_6

### Steps

1. Pause the pointer over the page explorer button  and click Report Pages.
2. In the Report Pages pane, click the nested page set.
3. In the Properties pane, double-click the Master Detail Relationships property.
4. Click the New Link button.
5. In the Master Query box, click the data item that provides the primary information.
6. Link the master data item to the details by doing one of the following:
  - To link to another data item in the detail query, in the Detail Query box, click the data item that provides the detailed information.
  - To link to a parameter, in the Parameters box, click the parameter that provides the detailed information.
7. Repeat steps 4 to 6 to create other links.

**Tip:** To delete a link, select it and press the Delete key.

For more information about master detail relationships, see "[Create a Master Detail Relationship](#)" (p. 386).

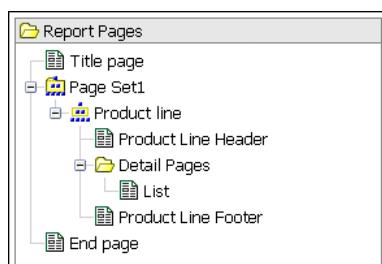
## Example - Preparing a Product List Report

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to create a multiple-page report showing all products sold by the company. You are asked to create title and end pages and to have each product line appear on a new page preceded by a header page and followed by a footer page.

### Steps

1. Open IBM® Cognos® Report Studio with the GO Data Warehouse (query) package.
2. In the Welcome dialog box, click Create a new report or template.
3. In the New dialog box, click List and click OK.
4. In the Insertable Objects pane, on the Source tab , expand Sales and Marketing (query), Sales (query), and Product, and double-click the following:
  - Product line
  - Product type
  - Product name
5. Expand Sales fact and double-click Product cost.

6. Group the **Product line** and **Product type** columns.
7. Pause the pointer over the page explorer button  and click **Report Pages**.
8. In the **Insertable Objects** pane, drag **Page Set** to the work area and associate it to **Query1**.
9. In the **Insertable Objects** pane, drag **Page** to the work area four times to add four new pages.
10. For each page, click the page, and in the **Properties** pane, set the **Name** property as follows:
  - Title page
  - End page
  - Product Line Header
  - Product Line Footer
11. For each page you just created, add objects.  
For example, add a text item to each page to uniquely identify it.
12. Rename the **Page1** page, which contains the list, to **List**.
13. Click the page set and, in the **Properties** pane, double-click the **Grouping & Sorting** property.
14. In the **Data Items** box, drag **Product line** to the **Groups** folder in the **Groups** box and click **OK**.
15. Organize the report pages into the following hierarchy by dragging them to the appropriate location.



When you run the report, the following pages appear:

- Title page
- Product line header page
- A page for Product line\_1
- Product line footer page
- Product line header
- A page for Product line\_2
- Product line footer
- ...
- End page

## Add a Bookmark

Add a bookmark so that users can quickly move from one part of a report to another. For example, a list report contains many rows of data. You add bookmarks so that users can move to specific rows.

You can also drill through to another report.

If you want to use page numbers to move from one part of a report to another, you can also create a table of contents.

Bookmarks work for reports produced in HTML format or PDF. In HTML format, they work best when viewing saved report outputs, as the entire report appears in a single HTML page. When reports are run interactively, more than one HTML page may be generated, and a bookmark works only if the target exists in the page currently being viewed.

If you run a saved report using a URL, and you specify a bookmark within the URL, you always go to the first page of the report. The bookmark defined in the URL is not honored. For more information about using URLs to run reports, see the *Administration and Security Guide*.

**Tip:** You can reduce the number of HTML pages generated when a report is run interactively by specifying a value for the **Rows Per Page** property for a data container in the report.

**Tip:** The Briefing Book sample report in the GO Sales (analysis) package includes bookmarks. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

### Steps

1. In the **Insertable Objects** pane, click the **Toolbox** tab .
2. Drag a **Bookmark**  object to the report.
3. Click the bookmark and, in the **Properties** pane, set the **Source Type** property to a source type.

| Source type              | Description                                                                                                                                                                                                                                                                                                                                       |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Text</b>              | <p>Creates a static bookmark using a text value.</p> <p>For example, a list has sections and you want users to jump from each section to the top of the report.</p>                                                                                                                                                                               |
| <b>Report Expression</b> | <p>Creates a dynamic bookmark whose values are derived from an expression that you define.</p>                                                                                                                                                                                                                                                    |
| <b>Data Item Value</b>   | <p>Creates a dynamic bookmark that has data as possible values. This is useful for creating a context-based bookmark.</p> <p>For example, a list has sections and you want users to jump from the top of the report to a specific section.</p> <p><b>Note:</b> This source type appears only if the bookmark is inserted next to a data item.</p> |

| Source type     | Description                                                                                                                                                                                                             |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Data Item Label | <p>Creates a bookmark that has the label of a data item as its value. Use this source type to jump to the first occurrence of a data item label.</p>                                                                    |
|                 | <p>For example, a list is divided into sections using Product line. You want users to jump to the first product line section that appears in the list rather than to a specific section, such as Camping Equipment.</p> |
|                 | <p><b>Note:</b> This source type appears only if the bookmark is inserted next to a data item.</p>                                                                                                                      |
| Member Caption  | <p>In a crosstab, creates a dynamic bookmark that uses member captions as possible values.</p>                                                                                                                          |
| Cell Value      | <p>In a crosstab, creates a dynamic bookmark that uses cell values as possible values.</p>                                                                                                                              |

4. Set the property that appears under the source type you chose to the bookmark value.  
For example, if the source type is **Data Item Value**, set the **Data Item Value** property to the data item.
5. Right-click the object that will jump to the bookmark and click **Drill-Through Definitions**.  
For example, right-click a data item, image, chart, or text item.  
**Tip:** The object can exist in a different report, so you can jump from one report to another.
6. Click the new drill-through definition button .
7. Click the **Bookmark** tab.
8. Click **Source type** and click one of the source types described in step 3.  
Click the source type to use to produce the value needed to jump to the bookmark. For example, click **Data Item Values** if you want the value to come from a data item such as Product line.
9. If you clicked one of the following source types, specify the value to use to jump to the bookmark.
  - For **Text**, click the ellipsis (...) button next to **Text** and type a text value.
  - For **Data Item Value** or **Data Item Label**, in the **Data item** list, choose a data item.
  - For **Report Expression**, click the ellipsis (...) button next to **Report expression** and define the expression.

## Create a Table of Contents

You can create a table of contents that appears in the rendered output of your report. A table of contents is useful for reports that include sectioned items, grouped items, or multiple pages in the layout. The report output indicates page numbers and allows for easy navigation.

You can include multiple tables of contents in your report, which are useful if your report contains grouped lists. You can also add section numbers in front of entries in the table of contents by modifying the heading level property.

Table of contents entries are logical markers placed anywhere in a report. For example, you can place entries at the top of a page or in a list group header to mark each grouped data value. Although table of contents entries are visible in IBM® Cognos® Report Studio, they cannot be seen when a report is run.

A table of contents works only for reports produced in PDF or non-interactive HTML format. In HTML format, they work best when viewing saved report outputs, as the entire report appears in a single HTML page. When reports are run interactively, more than one HTML page may be generated, and a table of contents works only if the target exists in the page currently being viewed.

**Tip:** You can reduce the number of HTML pages generated when a report is run interactively by specifying a value for the **Rows Per Page** property for a data container in the report.

If you want to quickly move from one part of a report to another without using page numbers, you can add bookmarks.

You must first create a table of contents before adding entries in the report. All entries must be inserted after the table of contents in the report layout.

**Tip:** The Briefing Book sample report in the GO Sales (analysis) package includes a table of contents. For more information about The Great Outdoors Company samples, see "[Sample Reports and Packages](#)" (p. 567).

### Steps

1. In the Insertable Objects pane, click the Toolbox tab.
2. Drag the **Table of Contents** object to the new location, which can be anywhere in the report.  
A table of contents placeholder appears.
3. Drag the **Table of Contents Entry** object to the location of your first table of contents marker.  
**Tip:** You can also click **Insert Table of Contents Entry** from the **Structure** menu.  
The new entry appears in the table of contents.
4. Double-click the **Double click to edit text** box of the new marker.
5. In the **Text** box, type the text to appear in the table of contents and click **OK**.
6. To edit the heading level of a table of contents entry, click the entry and, in the **Properties** pane, set the **Heading Level** property to the level.  
The heading level is used to insert section numbers in front of entries in the table of contents using layout calculations.

When you finish creating the table of contents, run the report. By clicking the arrow to the right of the run report button, you can specify whether to run the report as HTML or PDF.

## Example - Add a Table of Contents to a Report

You are a report author at The Great Outdoors Company, which sells sporting equipment. You are requested to add a table of contents to an existing report so that users can more easily navigate your report.

### Steps to Add a Table of Contents

1. Open Report Studio with the GO Data Warehouse (analysis) package.
2. Open the **Budget vs. Actual** sample report from the **Report Studio Report Samples** folder.
3. Create the report pages:
  - Pause the pointer over the page explorer button  and click **Report Pages**.
  - Click **Page1**, and then, in the **Properties** pane, set the **Name** property to **Budget vs. Actual Sales**.
  - Click the **Budget vs. Actual Sales** page and, from the **Edit** menu, click **Copy**.
  - From the **Edit** menu, click **Paste** to paste the copy of the page in the **Report Pages** pane.
  - Select the new page, and in the **Properties** pane, set the **Name** property to **Table of Contents**.
  - In the **Report Pages** pane, drag the **Table of Contents** page to the top of the list.
4. Pause the pointer over the page explorer button and click **Table of Contents**.
5. Delete the crosstab object:
  - Click somewhere in the report page.
  - In the **Properties** pane, click the select ancestor button  and click **Crosstab**.
  - Click the delete button .
6. Select the Camping Equipment block object and click the delete button.
7. Double-click the report title, type **Table of Contents**, and click **OK**.
8. In the **Insertable Objects** pane, on the **Toolbox** tab , drag a **Table of Contents** object onto the page.
9. Pause the pointer over the page explorer button and click **Budget vs. Actual Sales**.
10. In the **Insertable Objects** pane, on the **Toolbox** tab, drag a **Table of Contents Entry** object to the left of each region.
11. Pause the pointer over the page explorer button and click **Table of Contents**.
12. Double-click the first entry in the table of contents, type **Americas**, and click **OK**.

13. Rename the other table of contents entries as Asia Pacific, Northern Europe, Central Europe, and Southern Europe.
14. Save the report.

## Steps to Format a Table of Contents

1. Number the table of contents entries:
  - In the **Insertable Objects** pane, on the **Toolbox** tab , drag a **Layout Calculation** object to just before the word **Americas**.
  - In the **Report Expression** dialog box, on the **Functions** tab , expand the **Report Functions** folder and double-click the **TOCHeadingCount** expression.
  - At the end of the expression definition, type **1** and click **OK**.
  - Repeat the above three steps to add layout calculations before the other table of contents entries.
  - Ctrl+click only the five table of contents entries and not the five layout calculations.
  - In the **Properties** pane, double-click the **Padding** property and set the left padding to 10 px.
2. Change the color of table of contents entries:
  - Ctrl+click the five layout calculations, the five table of contents entries, and the five page number calculations.
  - In the **Properties** pane, double-click the **Foreground Color** property and set the foreground color to blue.
3. Add links to the table of contents from another page:
  - In the **Insertable Objects** pane, on the **Toolbox** tab, drag a **Bookmark** object to just before the title.
  - Select the bookmark and, in the **Properties** pane, double-click the **Label** property, type **TOC**, and click **OK**.
  - Pause the pointer over the page explorer button  and click **Budget vs. Actual Sales**.
  - In the **Insertable Objects** pane, on the **Toolbox** tab, drag a **Text Item** object to the right of the crosstab object.
  - In the **Text** dialog box, type **Return to the Table of Contents** and click **OK**.
  - Right-click the text object and click **Drill Through Definitions**.
  - Click the add button .
  - On the **Bookmark** tab, in the **Source Type** list, click **Text**.
  - Click the ellipsis (...) button, type **TOC**, and then click **OK** twice.

4. Save the report.
5. Click the arrow to the right of the run report button  and click Run Report - PDF.

On the first page, the table of contents appears. Clicking a region brings you to the corresponding page in the report. You can return to the table of contents by clicking **Return to the Table of Contents** at the end of the last page.

### Table of Contents

|                         |   |
|-------------------------|---|
| 1 Americas .....        | 2 |
| 2 Asia Pacific .....    | 2 |
| 3 Northern Europe ..... | 2 |
| 4 Central Europe .....  | 2 |
| 5 Southern Europe ..... | 4 |

## Insert Page Numbers in a Report

You can insert page numbers in a report and specify the number style to use. You can select a pre-defined page numbering scheme or create a custom scheme. You can easily insert page numbers using the **Page Number** object.

You can also manually create an expression to insert page numbers by inserting a layout calculation and using the different page report functions in the expression editor.

### Steps

1. In the **Insertable Objects** pane, on the **Toolbox** tab , drag **Page Number** to the report.

**Tip:** When you create a new report using one of the existing report layouts, **Page Number** is already inserted in the page footer.

2. Right-click the page number symbol and click **Edit Number Style**.
3. Choose the style to use.

The first three choices apply only to vertical page numbers. The remaining choices specify how vertical and horizontal page values appear.

4. If you want to customize the choice that you made in the previous step, click the edit button , make your changes, and click **OK**.

A custom number style is created. If you later choose a different number style, the custom style is removed from the list.

**Tip:** In the **Custom Number Style** dialog box, when you pause the pointer over a box, a tooltip describes how that box affects page numbers. For example, the **Separator Text** box contains the text, such as a hyphen, that separates page values for both vertical and horizontal pages.

## Control Page Breaks and Page Numbering

You can control page breaks and page numbering in a list, crosstab, table, or report page by choosing any of these options.

| Option                                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Keep with header</b>                     | Keeps all headers on the same page with the number of detail rows specified.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Keep with footer</b>                     | Keeps all footers on the same page with the number of detail rows specified.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Reset page count</b>                     | Resets the page count after a page break to the value specified.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Reset page number</b>                    | Resets the page number after a page break to the value specified.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Repeat every page</b>                    | If the report renders multiple pages, this object is repeated on every page.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Allow contents to break across pages</b> | Allows contents to break across pages. In lists and crosstabs, controls whether a cell is broken across pages, which is useful when there is a lot of text.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Allow horizontal pagination</b>          | <p>In PDF output, allows the columns of a list or crosstab to break across horizontal pages if they do not fit on a single page.</p> <p><b>Tip:</b> In lists, you can select the <b>Repeat every page</b> option for list columns that show on every horizontal page.</p> <p>If the <b>Allow horizontal pagination</b> option is not selected, the size of the list or crosstab is scaled down when necessary so that it fits on a single page.</p> <p><b>Tip:</b> The Horizontal Pagination sample report in the GO Sales (analysis) package includes horizontal pagination. For more information about The Great Outdoors Company samples, see "<a href="#">Sample Reports and Packages</a>" (p. 567).</p> <p>If your report includes nested data frames such as a list within a list, horizontal pagination is supported on either the parent or child frame, but not both. If horizontal pagination is enabled on both the parent and child frame, it will be ignored on the child frame when the report runs. We recommend that you do not enable horizontal pagination on both the parent and child frames.</p> <p>You can also specify page number options that use compound numbering schemes. For example, you can use the numbering scheme 1-1, 1-2, 2-1, 2-2, and so on. For more information, see "<a href="#">Insert Page Numbers in a Report</a>" (p. 545).</p> |

| Option                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Enable horizontal page numbering         | <p>Increments page numbers of horizontal pages separately from the main page numbers when you select a page numbering style that includes horizontal pages. For example, if a page has three page breaks horizontally and you selected the page number style 1a, the horizontal pages are numbered 1a, 1b, and 1c. If you did not select a numbering style that includes horizontal pages, the horizontal pages are all numbered 1 for the first vertical page, 2 for the second vertical page, and so on.</p> |
|                                          | <p>If this option is not selected and there are horizontal pages, all pages are numbered sequentially. For example, if a report has two vertical pages and three horizontal pages, the PDF pages are numbered from 1 to 6. Pages 1 to 3 are the three horizontal pages for the first vertical page and pages 4 to 6 are the three horizontal pages for the second vertical page.</p>                                                                                                                           |
| Allow row contents to break across pages | <p>In tables, allows the contents of a row to break across pages. For example, if a row contains four lines of text, the first two lines from the row appear on the first page, and the last two lines appear on the next page.</p>                                                                                                                                                                                                                                                                            |
| Repeat table rows on page break          | <p>In tables, if a row breaks across pages, repeats the rows that were previously rendered on each page. By default, table rows are repeated.</p>                                                                                                                                                                                                                                                                                                                                                              |
|                                          | <p><b>Note:</b> This option applies to saved reports only. In interactive HTML reports, table rows are always repeated even if this option is not selected.</p>                                                                                                                                                                                                                                                                                                                                                |

You can also specify the style to use for page numbers.

### Steps

1. Click an object.
2. In the **Properties** pane, double-click the **Pagination** property.
3. Specify the page break and numbering options.

## Create Page Layers

When working with dimensional data, you can create page layers in a report to show values for each member on a separate page. For example, your report contains payroll information for the entire company. You want to view values for each department on a separate page.

After you create page layers, a caption appears in the header to indicate the contents of each page. You can navigate between the different pages using links below the report.

Creating page layers is similar to filtering using context. However, with context filters, values are filtered according to the member you add to the **Context filter** area. With page layers, the report is split into a separate page for each child of the member you add to the **Page layers** area.

To create more complex page layers in your reports, such as a report book with title and end pages, create page sets.

### Steps

1. In the source tree, select or search for one or more items on which to filter.
2. Drag the item into the **Page layers** section of the overview area.

The crosstab shows the results for the children of the selected item on separate pages, and a list appears under **Page layers**.

**Tip:** To navigate between pages, click **Page down** and **Page up** below the report.

3. To replace the page breaks with items from the same dimension, select an item from the list. To delete the page breaks, from the list, click **Delete**. To delete all the page breaks, right-click the **Page layers** area and click **Delete All**.

# Chapter 21: Creating Report Templates

A report template is a pattern you use to build reports. Create your own report templates when you frequently produce the same type of report. A template can include the report objects that are described in "["Laying Out a Report" \(p. 437\)](#)".

**Tip:** You can add your own report templates to the New dialog box that appears when you open IBM® Cognos® Report Studio. For more information, see the IBM Cognos Business Intelligence *Administration and Security Guide*.

## Convert a Report to a Template

Convert a new or existing report to a template so it can be reused. You can use sample reports provided with IBM® Cognos® Report Studio.

### Steps

1. Create a new report or open an existing report.
2. Add objects to the work area.
3. From the File menu, click **Convert To Template**.

Any query-related information in the original report, such as data items, calculations, and filters, is removed from the template.

4. From the File menu, click **Save As** to save the template as a new file and keep the original report intact.

## Create a New Template

Create a new template to provide report authors with a layout that they can use to create reports or they can apply to existing reports. IBM® Cognos® Query Studio and IBM Cognos Analysis Studio users can also use these templates. You can also use the template in Query Studio to define a layout for prompt pages.

When you create a new template, the **Source** tab of the **Insertable Object** pane is empty; you can add only report objects.

### Steps

1. From the File menu, click **New**.
2. Click **Report Template** and click **OK**.
3. Pause the pointer over the page explorer button  and click the report page or prompt page to format.

**Tip:** To create a new report page or prompt page, click the **Report Pages** or **Prompt Pages** folder and drag the page to the **Report Pages** or **Prompt Pages** pane.

4. In the **Insertable Objects** pane, click the **Toolbox** tab .

5. Add the objects to the work area.

If you add objects that are not supported by Query Studio, the objects will be ignored when you apply the template.

6. Save the template.

---

# Chapter 22: Managing Existing Reports

---

After you have created a report, you can make changes or enhancements, such as setting up the report for bursting.

Before you modify an existing report, ensure that you have the proper security permissions. For more information, see the *IBM Cognos Business Intelligence Administration and Security Guide*.

You can

- [open and save reports locally](#)
- [copy a report to the clipboard](#)
- [open reports from the clipboard](#)
- [open files from other Studios](#)
- [manage changes in the package](#)

## Open and Save a Report on Your Computer

By default, the reports you create are stored on the IBM® Cognos® BI server. You can, however, open and save reports on your computer. This is useful to send a report to a report author who is working in a different environment or to save the report to a source code controlled directory on a local network or drive.

### Steps

1. Obtain the LFA.dll from your IBM Cognos BI administrator.

The DLL is located in the bin directory where IBM Cognos BI is installed.

2. Open a command prompt window.

3. Register the LFA.dll file by typing the following:

**regsvr32 LFA.dll**

4. In your browser, set your computer and the IBM Cognos BI server as trusted sites.

5. In IBM Cognos Report Studio, from the Tools menu, click Options.

6. Click the Advanced tab.

7. Select the Allow local file access check box and click OK.

The menu items (Local) Open and (Local) Save As appear in the File menu.

8. Close and restart Report Studio.

You can now open and save reports on your computer.

**Note:** The first time you try to open or save a report locally from the Microsoft® Internet Explorer Web browser, Internet Explorer asks you whether you want to allow an ActiveX control on the page to interact with other parts of the page. Click Yes to continue.

## Copy a Report to the Clipboard

You can copy a report specification to the clipboard so you can open it from the clipboard later. This process is different for the Microsoft® Internet Explorer and Mozilla Firefox Web browsers because the clipboard works differently in each Web browser.

### Step in Microsoft Internet Explorer

- From the Tools menu, click Copy To Clipboard.

### Steps in Mozilla Firefox

1. From the Tools menu, click Copy To Clipboard.
2. In the **Copy Report to Clipboard** dialog box, copy the entire report specification.
3. Open a text editor and paste the report specification.
4. Copy all the text from the text editor report specification.

Now the text is saved on your computer's clipboard.

## Open a Report from the Clipboard

You can open a report specification that was previously copied to the clipboard. To copy a report to the clipboard, see "[Copy a Report to the Clipboard](#)" (p. 552). This is useful for importing an XML report specification from outside the IBM® Cognos® BI environment.

Although IBM Cognos Report Studio attempts to validate the report specification, it is your responsibility to ensure that it is correct. For more information, see the IBM Cognos Software Developer Kit *Developer Guide*.

### Step

- From the Tools menu, click Open Report from Clipboard.

## Open a File from Another Studio

You can open reports, templates, or analyses that were created in IBM® Cognos® Query Studio or IBM Cognos Analysis Studio in IBM Cognos Report Studio. All the capabilities of Report Studio are available to you so you can change formatting, layout, calculations, and queries.

You can also open a report, template, or analysis in Report Studio from IBM Cognos Connection with the **Open with Report Studio** action .

### Steps

1. From the **File** menu, click **Open**.
2. Click the report, template, or analysis.
3. Click **Open**.

**Note:** If you make and save changes to a Query Studio report or an Analysis Studio analysis in Report Studio, the report or analysis can no longer be opened in Query Studio and Analysis Studio.

## Analysis Studio Query Specification

Analysis Studio defines each group of rows and columns as a set. When you import an analysis into Report Studio, the report will have one query that processes all the sets found on the crosstab. Each set is defined by 18 data items that segment and summarize the base set definition. To maintain reports converted from Analysis Studio, you must understand what each of these items represents and how they relate to each other.

Before you modify any of the data items in the Analysis Studio set definitions, we recommend that you fully understand each data item and its dependencies. Modifying the data items may cause unpredictable results and may slow the performance of your report.

The data items for a set specify the following:

- the set definition
- set segments, including which members were excluded and hidden individually
- filter rules for defining which members are to be retrieved
- calculations for Subtotal (N items), More & Hidden, Subtotal (included), Subtotal (excluded), and the total

For more information, see the Analysis Studio *User Guide*.

- which subtotals should appear
- any user-defined calculations

Filter rules, filter top bottom

|                            | Revenue               | 2004                  | 2005                  | 2006                   | Years |
|----------------------------|-----------------------|-----------------------|-----------------------|------------------------|-------|
| Tents                      | \$2,140,905.14        | \$2,334,075.18        | \$3,168,354.60        | \$7,643,334.92         |       |
| Packs                      | \$428,427.28          | \$738,480.48          | \$1,033,103.48        | \$2,200,011.24         |       |
| Sleeping Bags              | \$404,513.02          | \$700,438.52          | \$816,478.20          | \$1,921,429.74         |       |
| <i>Subtotal (3 items)</i>  | \$2,973,845.44        | \$3,772,994.18        | \$5,017,936.28        | \$11,764,775.90        |       |
| <i>More &amp; hidden</i>   | \$402,022.06          | \$560,491.46          | \$737,744.04          | \$1,700,257.56         |       |
| <i>Subtotal (included)</i> | \$3,375,867.50        | \$4,333,485.64        | \$5,755,680.32        | \$13,465,033.46        |       |
| <i>Subtotal (excluded)</i> | \$230,330.32          | \$263,247.24          | \$414,318.20          | \$907,895.76           |       |
| <b>Total</b>               | <b>\$3,606,197.82</b> | <b>\$4,596,732.88</b> | <b>\$6,169,998.52</b> | <b>\$14,372,929.22</b> |       |

Excluded list      Hidden list

The default measure identifies which measure is the default for the crosstab or chart. If no default measure is specified, the default measure may be empty.

## Set Definitions

The following table shows the definitions and dependencies for the data items in the set definition.

| Data item              | Definition                                                                                                                                                                                                     | Dependencies |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|
| <set name> (base)      | Defines the set of members to be used for sorting, filtering, and summary operations. This data item provides a generic reference for all other data items and may reference one of the other set definitions. | None         |
| <set name> (level)     | Identifies the level for a level-based set.                                                                                                                                                                    | None         |
| <set name> (list)      | Defines the list of members in a selection-based set.                                                                                                                                                          | None         |
| <set name> (depth N)   | Defines the set of members at N, number of levels down.                                                                                                                                                        | None         |
| <set name> (named set) | References a predefined set.                                                                                                                                                                                   | None         |

## Set Segments Definitions

The following table shows the definitions and dependencies for the data items in the set segment definition.

| Data item                                         | Definition                                                                                                                                                                                                                                                                                      | Dependencies                                 |
|---------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------|
| <i>&lt;set definition&gt;</i> (hidden list)       | Lists the members that are manually hidden using the <b>Hide</b> command in Analysis Studio. This set appears when you try to unhide a member.                                                                                                                                                  | <i>&lt;set definition&gt;</i> (hidden list)  |
| <i>&lt;set definition&gt;</i> (included set)      | Lists the set of members after filters are applied and hidden items are excluded, but before the <b>More</b> limit is applied. Sort or order operations, if any, are defined in this data item.                                                                                                 | <i>&lt;set definition&gt;</i> (hidden list)  |
| <i>&lt;set definition&gt;</i> (visible items set) | Limits the set to show the number of members according to the <b>More</b> limit with a small tolerance. The tolerance allows showing the last two members if they are all that remain in the <b>More</b> portion of the set.<br><br>The choice of visible members can be based on a sort order. | <i>&lt;set definition&gt;</i> (included set) |
| <i>&lt;set definition&gt;</i> (excluded list)     | Identifies members that were manually excluded from the analysis. If no members were manually excluded, the expression defines an <code>emptySet</code> .<br><br>This set appears when you try to remove a member from the <b>Excluded items</b> list in the <b>Properties</b> pane.            | <i>&lt;set definition&gt;</i>                |

## Filters

The following table shows the definitions and dependencies for the data items in the filter.

| Data item                                        | Definition                                                                                                                                                             | Dependencies                                                                                |
|--------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| <i>&lt;set definition&gt;</i> (filter rules)     | Specifies the user-defined filter rules to reduce the set definition using operators such as greater than or less than based on measures, calculations, or attributes. | <i>&lt;set definition&gt;</i>                                                               |
| <i>&lt;set definition&gt;</i> (excluded filters) | Removes members that were manually excluded from the results after applying the user-defined rules.                                                                    | <i>&lt;set definition&gt;</i> (excluded list), <i>&lt;set definition&gt;</i> (filter rules) |

| Data item                                               | Definition                                                                                                                                                                                    | Dependencies                                                                                    |
|---------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| <code>&lt;set definition&gt;</code> (filter top/bottom) | Focuses on the members based on top / bottom / first $n$ where $n$ can be a count or a percentile. For more information about Top/Bottom filters, see the Analysis Studio <i>User Guide</i> . | <code>&lt;set definition&gt;</code> (excluded filters) and totals for sets on the opposite axis |
| <code>&lt;set definition&gt;</code> (filter rules)      | Specifies the user-defined filter rules to reduce the set definition using operators such as greater than or less than based on measures, calculations, or attributes.                        | <code>&lt;set definition&gt;</code>                                                             |

### Subtotals and Related Conditions

The following table shows the definitions and dependencies for the data items in the subtotals and related conditions.

| Data item                                                                | Definition                                                                                                                                                                                                                                                        | Dependencies                                                                                                                                                |
|--------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>&lt;set definition&gt;</code> (subtotal)                           | Aggregates the visible items in the set.                                                                                                                                                                                                                          | <code>&lt;set definition&gt;</code> (visible items set)                                                                                                     |
| <code>&lt;set definition&gt;</code> (subtotal display)                   | Shows the subtotal if the number of included items is greater than the number of visible items.                                                                                                                                                                   | <code>&lt;set definition&gt;</code> (visible items set), <code>&lt;set definition&gt;</code> (included set), <code>&lt;set definition&gt;</code> (subtotal) |
| <code>&lt;set definition&gt;</code> (more and hidden subtotal)           | Calculates the <b>More &amp; hidden</b> subtotal by subtracting the aggregation of the included members from the aggregation of those members that were manually hidden or hidden by exceeding the maximum display limit. Not available for selection-based sets. | <code>&lt;set definition&gt;</code> (included set), <code>&lt;set definition&gt;</code> (visible items set), <code>&lt;set definition&gt;</code>            |
| <code>&lt;set definition&gt;</code> (more and hidden subtotal as set)    | Converts the <b>More &amp; hidden</b> subtotal member to a set for use in set operations.                                                                                                                                                                         | <code>&lt;set definition&gt;</code> (more and hidden subtotal)                                                                                              |
| <code>&lt;set definition&gt;</code> (more and hidden subtotal as member) | Generically references the <b>More &amp; hidden</b> subtotal.                                                                                                                                                                                                     | <code>&lt;set definition&gt;</code> (more and hidden subtotal)                                                                                              |

| Data item                                                        | Definition                                                                                                                                                                                                                                                                                                                                                                                                                                          | Dependencies                                                                                                            |
|------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| <i>&lt;set definition&gt;</i> (more and hidden subtotal display) | Shows the <b>More &amp; hidden</b> subtotal if the number of items hidden or clipped is greater than zero.                                                                                                                                                                                                                                                                                                                                          | <i>&lt;set definition&gt;</i> (more and hidden subtotal)                                                                |
| <i>&lt;set definition&gt;</i> (included subtotal)                | Calculates the <b>Subtotal (included)</b> . For selection-based sets, this data item references the summary data item for the set.                                                                                                                                                                                                                                                                                                                  | <i>&lt;set definition&gt;</i> (included set)                                                                            |
| <i>&lt;set definition&gt;</i> (included subtotal display)        | Shows the <b>Subtotal (included)</b> if any members passed the filter criteria.                                                                                                                                                                                                                                                                                                                                                                     | <i>&lt;set definition&gt;</i> (included set)                                                                            |
| <i>&lt;set definition&gt;</i> (excluded subtotal)                | Calculates the value for <b>Subtotal (excluded)</b> . This data item is not available for selection-based sets.                                                                                                                                                                                                                                                                                                                                     | <i>&lt;set definition&gt;</i> (total), <i>&lt;set definition&gt;</i> (included subtotal), <i>&lt;set definition&gt;</i> |
| <i>&lt;set definition&gt;</i> (excluded subtotal display)        | Shows <b>Subtotal (excluded)</b> if the number of members in the filtered set is less than the base set. This means that the filter rule is filtering out something.                                                                                                                                                                                                                                                                                | <i>&lt;set definition&gt;</i> (excluded subtotal), <i>&lt;set definition&gt;</i>                                        |
| <i>&lt;set definition&gt;</i> (total)                            | <p>Identifies the total for the set. This value is calculated directly from the data store.</p> <p>The expression used to calculate the summary depends on the set type:</p> <ul style="list-style-type: none"> <li>• Details-based set: The selected member.</li> <li>• Level-based and depth-based sets: The root member for the hierarchy.</li> <li>• Selection-based set: The aggregate, typically the sum, of the selected members.</li> </ul> | <i>&lt;set definition&gt;</i>                                                                                           |

## Managing Changes in the Package

If changes were made to the package that was used to create a report, the report must be updated. When you open a report, IBM® Cognos® Report Studio automatically checks to see if the package

has changed. If it has, a message appears indicating that the report will be updated to the latest version of the package. However, you may need to make additional changes to the report if

- the namespace name or query subject and data item names in the package have changed
- the name of the package has changed

## Update a Name Reference

If the namespace name or query subject and data item names in the package have changed, you must update reports that were created with the package to reflect the change.

The names of data items in a report are a concatenation of the namespace name, the query subject name, and the data item name. For example, if you add Order number from the GO Data Warehouse (query) sample package to a filter expression, you will see [Sales (query)].[Sales order].[Order number] in the expression. Similarly, package filter names are a concatenation of the namespace name and the filter name.

### Steps

1. To update a data item name reference:
  - Click the data item.
  - In the Properties pane, double-click the **Expression** property.
  - In the **Expression Definition** box, update the data item name reference.
2. To update a package filter name reference:
  - From the Data menu, click **Filters** .
  - Double-click the package filter.
  - In the **Expression Definition** box, update the namespace name.

## Change the Package

If the name of the package that were used to create a report has changed, you must change the package connection to update the report.

At the same time, you can also change the authoring language for the report.

### Steps

1. From the File menu, click **Report Package**.
2. Click **Another package** and choose the new package.
3. To change the authoring language, click the ellipsis (...) button beside the **Language** box and choose a different report language.

You may need to update the report to reflect the change. For example, any filter expressions in the report must be modified to reflect the syntax rules of the new language.

4. Click **OK**.

The new package is loaded and the report is validated. If errors are found, the **Validation Errors** dialog box appears, showing the elements that must be updated to reflect the package change.

5. Click **Close**.

6. Make any required changes in the report to support the new package.

For example, you may need to link the data items in the report to the new package.

**Tip:** You can use Query Explorer to make the changes.



# Chapter 23: Upgrading Reports

When you open a report that was created in a previous version of IBM® Cognos® BI, it is automatically upgraded. Any problems detected during the upgrade process appear as information messages  and error messages  in the **Upgrade Information** dialog box. You must fix any errors in the report and then validate the report before you can run it. In some cases, the information or error message is linked to the location of the issue in your report. To go to the location of the issue, click the message, and then click **Select**. If only warnings and information appear in the dialog box, these will disappear when you click **OK**.

**Tip:** To view this dialog box again, from the **File** menu, click **Upgrade Information**.

After you upgrade a report to the most recent version of IBM Cognos BI, you can no longer open it with a previous version.

## Lifecycle Manager

You can download IBM Cognos Lifecycle Manager from [www.ibm.com](http://www.ibm.com) to help you test your reports. Lifecycle Manager is a verification tool that checks that your reports run and produce the same results in the new environment.

Lifecycle Manager is a Microsoft® Windows® Operating System-based application for auditing upgrades to the latest version of IBM Cognos BI from IBM Cognos ReportNet® 1.1 MR3 or MR4, and from IBM Cognos 8 versions 8.2, 8.3, or 8.4.

It provides a verification feature that validates, executes, and compares report results from two different IBM Cognos BI releases. This helps to identify upgrade and compatibility issues between releases. User interface design and status reporting functionality provide both a proven practice process and support for upgrade project planning and status reporting. Lifecycle Manager also automates much of the process of bundling the required files, such as reports and models, for the test case. For more information, see the *Lifecycle Manager User Guide*.

## Upgrading Reports from IBM Cognos BI Version 8.4

When you upgrade IBM® Cognos® BI to version 10.1.0, some reports may look or behave differently after the upgrade. This section describes changes that you may encounter in your reports.

## Upgrading Report Styles

IBM Cognos BI includes a new default report style with updated colors and gradients. If your report uses a custom report template, your report will appear the same in this version of IBM Cognos Report Studio as it did in previous versions. By default, new reports and new report objects, such as lists and crosstabs, appear in the new report style.

If you want to continue to work with the previous 8.x report style, set the **Override 10.x styles with 8.x styles on new reports** option (**Tools**, **Options**, **Advanced** tab).

You can update the style of an upgraded report to use the new 10.x style ([File, Report Properties, Report styles, 10.x styles](#)).

For more information about report styles, see ["Create and Modify Report and Object Styles" \(p. 459\)](#).

## Upgrading Legacy Charts

Report Studio version 10.1.0 includes a new default chart technology. You can continue to use and work with the legacy charts or upgrade your legacy charts to the current default charts.

When you open a report that uses the legacy charts, the legacy charts is not upgraded to the current default charts automatically. You can upgrade your legacy charts one at a time to the current default chart. For more information, see ["Convert Charts From One Type to Another" \(p. 145\)](#).

When you add a new chart to a report, Report Studio adds the current default charts. If you want to add new legacy charts or continue to work with existing legacy charts, set the **Use legacy chart authoring** option ([Tools, Options, Advanced tab](#)).

## New Default for Hidden or Deleted Axis Titles

The default for rendering hidden or deleted axis titles has changed. This can change the size or placement of chart elements in your chart report output.

In previous versions of Report Studio, hidden or deleted axis titles were defined and rendered as empty spaces in the report.

In Report Studio, Version 10.1.0, hidden or deleted axis titles are not rendered at all, so the space that was reserved for the title is available for other chart elements. This can affect the placement or size of chart elements, such as labels, the chart body, or the bars in a bar chart.

To make the charts appear the way they did in the previous version, set the x-axis' **Title** property to **Show** and then set the x-axis title's **Default Title** property to **No**. This adds the space for a custom axis title to the report specification but leaves the title blank.

## Upgrading Reports with SAP BW Prompt Variables

When you upgrade reports that use SAP BW data sources and contain variables as prompts from version 8.3 SP2 to version 8.4 or 10.1.0, the upgrade may fail. Prompts that contained SAP BW variables are now empty. This occurs because a default setting in an IBM Cognos configuration file changed in version 8.4.

To successfully upgrade these reports, do one of the following:

### Steps

1. Fully qualify all of the variables in the report by changing `[variable_name]` to `[infoquery].[variable_name]`.
2. Ask your IBM Cognos administrator to change the value for the `DetectSAPVariableUniqueness` parameter in the configuration file, as follows:
  - In the `c10_location\configuration` directory, locate the `qfs_config.xml` file.
  - Find the `DetectSAPVariableUniqueness` parameter and change its value to `false`.

- Save the qfs\_config.xml file.
- Restart the IBM Cognos service.

If more than one variable with the same name exists in the package, this change may cause problems.

## Upgrading Reports from IBM Cognos BI Version 8.1 or 8.2

When you upgrade IBM® Cognos® BI, some features in IBM Cognos BI may behave differently after the upgrade. When you upgrade reports, for example, changes in behavior may cause validation errors. Documentation is available about the behavior changes. This documentation includes examples of the changed behavior and solutions for issues that may occur during the upgrade. For more information, see *Upgrading to Cognos 8 BI 8.3 - Changes in Product Behavior* on [www.ibm.com](http://www.ibm.com).

## Members Containing No Data Are Not Suppressed for SAP BW

If you upgrade a crosstab report that uses a SAP BW data source and includes calculations, filters, or nesting, the suppression of null values may not occur as expected. You may see additional empty rows and columns.

By default, null values are suppressed for list and grouped list reports.

To effectively remove null values in crosstabs that use a SAP BW data source, insert individual members to create the report. You can also ensure that SAP BW members are assigned to proper dimension hierarchies and levels within the BW cube.

## Thousands Separators Missing

You may encounter missing thousands separators in your reports in the following circumstances:

- The report contains an item that does not specify an explicit data format.
- The report item refers to another item in the same report.
- That second report item refers to an item in the IBM Cognos Framework Manager model with Usage property set to **Identifier** or **Attribute**.
- The model item does not specify an explicit format.

To restore the data formats, specify an explicit format either in the report item properties or in the model item properties.

## RQP-DEF-0177 Error When Upgrading Reports

When you upgrade your report from IBM Cognos BI, version 8.2, you encounter the following errors:

RQP-DEF-0177 An error occurred while performing operation 'sqlPrepareWithOptions' status='-120'.

UDA-SQL-0458 PREPARE failed because the query requires local processing of the data. The option to allow local processing has not been enabled.

This error occurs when the query requires local processing because some or all of its constructs are not supported by the database vendor. In IBM Cognos BI, version 8.2, and earlier, IBM Cognos BI proceeded with local processing. In version 8.3 and later version, you must explicitly set the query Processing property to **Limited Local** in order to enable local processing.

### Steps

1. In IBM Cognos Report Studio, pause the pointer over the query explorer button  and click the query.
2. In the Properties pane, set the Processing property to **Limited Local**.

## Upgrading Reports from IBM Cognos ReportNet

When you upgrade from IBM® Cognos® ReportNet®, some reports may look or behave differently after the upgrade. The upgrade does not account for the following issues.

### Undocumented and Unsupported Features

If advanced report authors used undocumented and unsupported features such as JavaScript™ that refer to IBM Cognos HTML objects, they may have to recreate the features to complete the upgrade of the report.

### No Sort Order

Data may appear in a different order after upgrading. If sort order is important, ensure that the report or model specifies a sort order before upgrading.

### Layout Errors

Layout errors are suppressed by default in ReportNet. IBM Cognos BI does not suppress layout errors. Users may have to correct or remove report layout expressions that cause errors.

### Report Format

If you used the default format in ReportNet, upgraded reports will retain that format. However, new reports in IBM Cognos BI will use the IBM Cognos BI format. If you want a consistent style across all reports, you must edit the styles property in each report and select or deselect the **Use 1.x report styles** option.

### Customized Style Sheets

If you edited the ReportNet style sheet (default\_layout.css) or the IBM Cognos BI stylesheet (globalreportstyles.css), reports will lose the formatting after upgrading. You must reapply the changes to the IBM Cognos BI stylesheets and copy the stylesheets to the IBM Cognos BI server and the Web server.

## PDF Reports

PDF rendering in IBM Cognos BI behaves like HTML rendering. After upgrading, PDF reports may have different font size, column wrapping, or word wrapping. You may have to change each affected report or change the default font in the IBM Cognos BI style sheet.

## IF-THEN-ELSE Statements

If you use assignments of different data types after THEN and ELSE in ReportNet, the reports will generate invalid coercion errors after upgrading. You may have to recast the variables or change the assignments in the affected reports.

## Solve Order

In previous versions of IBM Cognos BI, you could specify the solve order for objects such as crosstab node members. Solve order is now specified for data items. When you upgrade a report, solve orders specified in the report are moved to data items. For more information about solve order, see "[Resolve Multiple Calculations for Crosstabs and Charts](#)" (p. 405).

## Chart Behavior

In ReportNet, if a chart is created with a user-specified minimum value and all data values are below the minimum value, the chart starts at the user-specified minimum value and contains no data.

In IBM Cognos BI, the same parameters result in a chart that ignores the user-specified minimum value and uses a range that includes all of the data values for the chart.

## Database Only Processing of Queries

If you specified in ReportNet that the processing for a query should be Database Only, when you upgrade your report to IBM Cognos BI, some processing may now occur locally.

## Changes in the Behavior of Functions Between ReportNet 1.1 and IBM Cognos BI

In ReportNet 1.1, double counting could occur when applying `count` or `count distinct` to a query item. This occurred primarily when querying multiple query subjects that were joined 1-to-N in the model if the counted item existed in the query subject on the 1 side of the join. The explanation is that `count` or `count distinct` was applied after the join operation.

`Count distinct` worked only when the item to which `count distinct` was applied was the column used in the join. In this case, the column to be counted could have identical values for the different values used in the join condition.

In IBM Cognos BI, improvements to the count functionality cannot be handled through an automatic upgrade. The new approach avoids double counting on the 1 side of a 1-to-N join. The explanation is that `count` or `count distinct` is now applied before the join operation.

There is now a lesser need to use `count distinct` and, when used, it will be successful more often. `Count distinct` is no longer required to overcome double counting. Instead, it can be used as intended to select distinct values that exist in a query subject.

When you want to count repeated occurrences of a value, we recommend that you do one of the following:

- Apply a `count` operation on a column based on a query in which an explicit join occurs.  
This applies the count after the join.
- Count rows in a report by using a layout calculation object or by counting the literal value 1.

---

# Chapter 24: Sample Reports and Packages

---

Sample reports are included with IBM® Cognos® Business Intelligence. When installed, you can find them in the **Public Folders** tab in IBM Cognos Connection.

## The Great Outdoors Company Samples

The Great Outdoors Company samples illustrate product features and technical and business best practices. You can also use them for experimenting with and sharing report design techniques and for troubleshooting. As you use the samples, you can connect to features in the product.

For examples related to different kinds of businesses, see the product blueprints at [www.ibm.com](http://www.ibm.com). For information about specific installation choices and environments, see the IBM® Cognos® *Architecture and Deployment Guide*, or the Proven Practices and the IBM Cognos Implementation Roadmaps on [www.ibm.com](http://www.ibm.com). For information about audit samples, see the IBM Cognos *Administration and Security Guide*. For information about Mobile samples, see the IBM Cognos Mobile *Installation and Administration Guide*.

The Great Outdoors Company, or GO Sales, or any variation of the Great Outdoors name, is the name of a fictitious business operation whose sample data is used to develop sample applications for IBM and IBM customers. Its fictitious records include sample data for sales transactions, product distribution, finance, and human resources. Any resemblance to actual names, addresses, contact numbers, or transaction values, is coincidental. Unauthorized duplication is prohibited.

The samples are included with the product and the samples for each studio are described in the related user guide and online help. To use the samples, you must install, set up, and configure them or contact your administrator to find out where they are installed. For instructions on how to install the samples, see the IBM Cognos *Installation and Configuration Guide*. For instructions on how to set up and configure samples, see the IBM Cognos *Administration and Security Guide* or the IBM Cognos *Installation and Configuration Guide*.

The samples consist of the following:

- Two databases that contain all corporate data, and the related sample models for query and analysis
- Five samples cubes and the related models
- A metrics data source including associated metrics and a strategy map for the consolidated company, and a model for Metric extracts.
- Reports, queries, query templates, and dashboards

To run interactive reports, scripts are required. To see all the reports included in the samples packages, copy the files from the samples content installation into deployment folder and then import the deployments into the IBM Cognos Business Intelligence product.

Samples are available to everyone. To implement security, see the *Installation and Configuration Guide*.

## The Great Outdoors Group of Companies

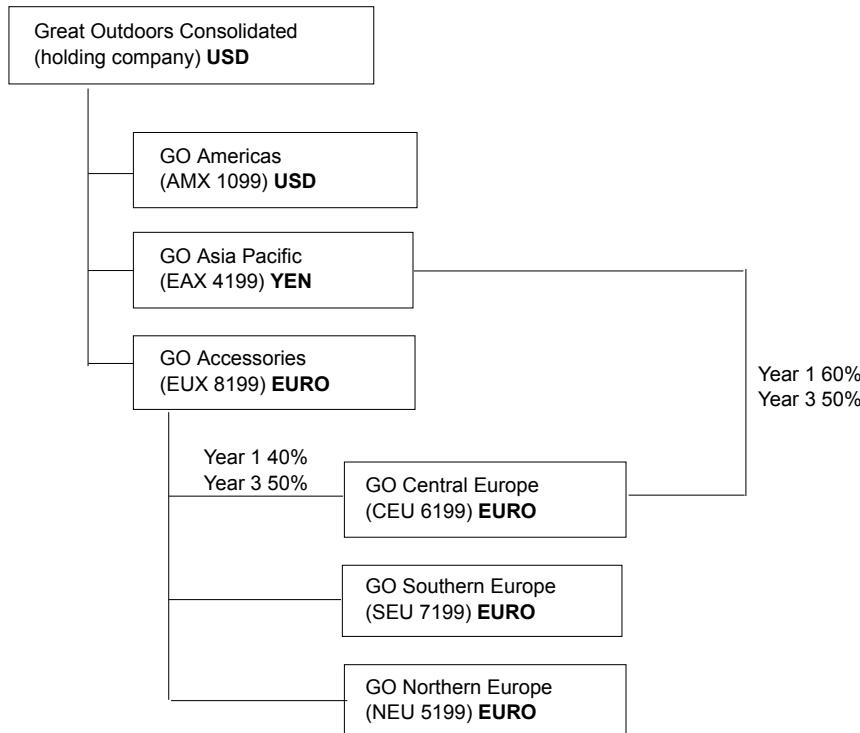
To make designing examples faster, especially financial examples, some general information about The Great Outdoors Company is useful. To look for samples that use particular product features, see the individual sample descriptions in this section.

Revenue for The Great Outdoors Company comes from corporate stores and from franchise operations. The revenues are consolidated from the wholly-owned subsidiaries. There are six distinct organizations, each with its own departments and sales branches. Five of these are regionally-based companies.

The sixth company, GO Accessories:

- Has its own collection of products, differentiated from the other GO companies by brand, name, price, color and size
- Sells from a single branch to all regions and retailers
- Functions both as an operating company based in Geneva, and as a part owner of the three GO subsidiaries in Europe

The diagram below illustrates the consolidated corporate structure, including the percentage changes in ownership for GO Central Europe, and shows the reporting currency and GL prefix for each subsidiary.



Each corporation has the same departmental structure and the same GL structure, shown in the table below. Divisions may not report in the same currencies. For example, the Americas subsidiary

reports in US dollars, but the Corporate division local currency is Canadian dollars, and the Operations division local currency is pesos.

| <b>Division (GL)</b> | <b>Department (GL)</b>             |
|----------------------|------------------------------------|
| Corporate (1700)     | Sales (1720)                       |
|                      | Marketing (1750)                   |
|                      | IS&T (1760)                        |
|                      | Human Resources (1730)             |
|                      | Finance (1740)                     |
|                      | Procurement (1710)                 |
| Operations (1800)    | Production and Distribution (1820) |
|                      | Customer Service (1820)            |

Each corporation has a complete chart of accounts. Most of the accounts, such as those under non-personnel expenses, are at the department level, and contain only summary amounts. For example, although each marketing department has expenses, the cost is unspecified at the transaction level where marketing promotions occur.

## Employees

The Great Outdoors data contains a full list of employees in all divisions, departments, and locations. Data is available for reports about bonuses (Global Bonus report) and sales commissions (Sales Commissions for Central Europe report), training (Employee Training by Year report), and performance reviews and employee satisfaction surveys (Employee Satisfaction 2006). If you use Metric Studio, sample metrics for human resources are also available.

In the GO Data Warehouse (analysis) package, groups of measures and the related dimensions are organized into folders. The employees are organized in hierarchies for region and manager, to make different kinds of aggregation easy to report on. Aggregation has been defined for the Employee Position Summary measures, so that Position count and Planned position count aggregate correctly at each level of time: monthly, quarterly, or yearly. For example, see the Planned Headcount report.

The employees are also listed in a sample LDIF file. This authentication directory is necessary for the Transformer 8 cubes and for IBM® Cognos® Planning samples. No other samples depend on security profiles. For more information, see the IBM Cognos Business Intelligence *Installation and Configuration Guide*.

## Sales and Marketing

Data about sales and marketing is available for all of the companies in the Great Outdoors group. GO Accessories has richer details to support analysis examples. For example, see the Revenue vs % Profit Margin by Product Brand analysis, based on the Sales and Marketing cube. Marketing and sales campaigns are tied to the Great Outdoors regional companies.

Overall, the GO companies have experienced solid growth across most product lines (Sales Growth Year Over Year), in all regions (Revenue by GO Subsidiary 2005), because of factors like an increase in repeat business and new or improved products, such as the high margin sunglasses product line. In the product lines sold by the five regional companies (all but GO Accessories) promotions have had mixed success (Promotion Success by Campaign, Bundle and Quarter). If you use Metric Studio, this can also be seen in the sample metrics.

### **Customer Surveys**

The data also contains information from customer surveys. For example, the product line that includes bug spray, sun screen, and so on has not been successful (Product Satisfaction - Outdoor Protection 2005) and a source of retailer dissatisfaction may be the level of customer service rather than the returns (Customer Returns and Satisfaction). If you use Metric Studio, this information can also be monitored in metrics.

### **Sales Outlets**

Revenue from the corporate outlets is available at the transaction level. Revenue from the franchise outlets is available at the consolidated level only (Sales and Marketing cube). Metrics about retailers show that the number of new retail outlets has dropped over the time period covered by this data.

GO Accessories sells worldwide, and sells only accessories. Transaction data for GO Accessories is the primary source for analysis of product by brand, color and size. The other five subsidiaries in the group of companies are regional and sell all product lines for retailers in their region. For example, the report Top 10 Retailers in 2005 uses sparklines and list data to review revenues at the retailer level.

## **Great Outdoors Database, Models, and Packages**

The Great Outdoors models illustrate modeling techniques and support the samples. The models are based on the GO data warehouse and the GO sales transactional database and are the basis for the sample reports and queries. Each model contains two packages for publishing analysis (dimensional) and query views of the data.

For a description of each sample report or query, see the user guide for the studio that you open the sample in. For more information about modeling techniques, see the *Guidelines for Modeling Metadata*, or the IBM® Cognos® Framework Manager *User Guide*.

You must have access to Framework Manager, the modeling tool in IBM Cognos BI, to look at the sample models. You may also need to set up the sample databases and connections. For instructions, see the IBM Cognos Business Intelligence *Administration and Security Guide* or the IBM Cognos Business Intelligence *Installation and Configuration Guide*.

### **GO Data Warehouse**

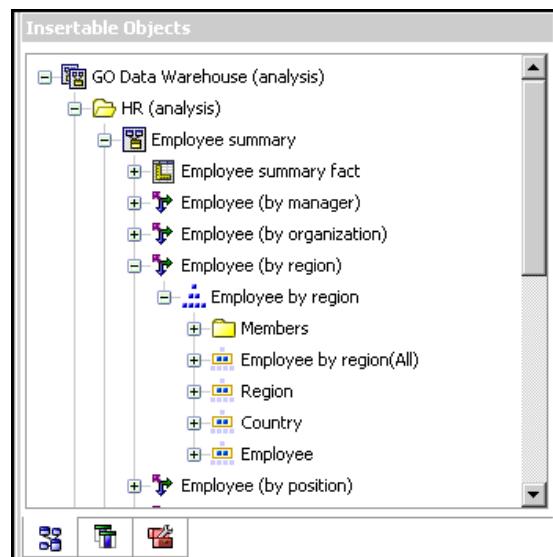
The GO Data Warehouse model, great\_outdoors\_data\_warehouse.cpf, is based on the database GOSALES DW. It contains data about human resources, sales and marketing, and finance, grouped into business areas. In the Database view, the three business areas are grouped into separate namespaces. The Database view contains a fourth namespace (GO Data) for the common information.

The Database view is very similar to the structure of the underlying database. All tables (database query subjects) are unchanged. This enables IBM Cognos BI to retrieve metadata directly from the package in most cases, instead of using a metadata call to the database. The following changes and additions have been made in the Database view:

- Joins have been added as necessary.
- To allow for aggregation at different levels of granularity, some model query subjects have been created. For example, see the relationships between Time and Sales or Sales fact.
- To allow single joins to be made between the lookup tables and each level in a dimension, lookup tables have been copied. For example, see the Products look up tables.

The Business view contains only model query subjects, with no joins. The following changes and additions have been made in the Business view:

- Calculations were added to the model query subjects. For example, the time dimension contains language calculations.
- Where the database has multiple hierarchies, new dimensions have been created to organize each hierarchy. For example, see the employee hierarchies, where employees are organized by manager and region.



### The GO Sales Transactional Database

The GO Sales model, great\_outdoors\_sales.cpf, is based on the GOSALES database, which is structured as a transactional database. It contains principally sales data.

The Database view is very similar to the underlying database structure. The following changes and additions have been made in the Database view:

- To make it possible to join the fact tables to the time dimension, model query subjects and multipart joins have been used.
- Other joins have been added as necessary.

The Business view contains only model query subjects, with no joins. The following changes and additions have been made in the Business view:

- Calculations were added to the model query subjects.
- Model query subjects that were created in the Database view to enable joins on the time dimension have been linked as reference shortcuts.
- Where the database has multiple hierarchies, new dimensions have been created to organize each hierarchy.
- Sales Staff is a subset of the slowly changing Employee dimension. There is no unique Employee key in GO Sales, so a filter retrieves the current record only. This model does not use historical data.

### **The Samples Power Cubes**

The following cubes are delivered with the Great Outdoors samples in English, French, German, Japanese and Chinese:

- sales\_and\_marketing.mdc
- employee\_expenses.mdc
- go\_accessories.mdc
- go\_americas.mdc
- go\_asia\_pacific.mdc
- great\_outdoors\_sales\_en.mdc

### **The Samples Packages**

The Great Outdoors samples include six packages. Below is a brief description of each available package.

Go Data Warehouse (analysis) is a dimensionally modeled view of the GOSALESDW database. This package can be used in all studios, including Analysis Studio. Using this package you can drill up and down.

Go Sales (analysis) is a dimensionally modeled view of the GOSALES database. This package can be used in all studios, including Analysis Studio. Using this package you can drill up and down.

Go Data Warehouse (query) is a non-dimensional view of the GOSALESDW database. This package can be used in all studios except Analysis Studio, and is useful for reporting when there is no need for drilling up and down.

Go Sales (query) is a non-dimension view of the GOSALES database. This package can be used in all studios except Analysis Studio, and is useful for reporting when there is no need for drilling up and down.

Sales and Marketing (cube) is an OLAP package, based on the sales\_and\_marketing.mdc cube.

Great Outdoor Sales (cube) is an OLAP package, based on the great\_outdoors\_sales\_en.mdc cube.

**Note:** The OLAP packages, Great Outdoor Sales (cube) and Sales and Marketing (cube), are not multilingual. The IBM\_Cognos\_PowerCube.zip archive contains five versions of each package; one in English, French, German, Japanese and Chinese.

## Samples in the Sales and Marketing (Cube) Package

The following reports are some of the reports found in the Sales and Marketing (Cube) package.

### Actual vs. Planned Revenue

This report shows the actual revenue versus planned revenue by order method and year. This report is also a target for the measure based scope drill-through from other reports in the same package. This report uses the following features:

- lists
- filters
- multiple prompts
- grouping
- sorting
- summarizing

### Historical Revenue

This prompted report shows a 13-month rolling forecast of monthly and year-to-date revenue. This report uses the following features:

- filters
- cascading prompts
- combination charts
- axis titles

### Revenue by Date Range

This report shows revenue for a date range that is specified on a prompt page. This report uses the following features:

- lists
- crosstabs
- context filters
- custom headers and footers
- multiple prompts
- calculations

## Revenue by Product Brand (2005)

This report shows the revenue and gross profit by product filtered by the product brand. There is always product turnover, so the report conditionally highlights products that are discontinued. This report uses the following features:

- lists
- filters
- prompts
- combination charts
- bar charts
- HTML items
- grouping
- sorting
- axis titles

## Running Total For Promotion

This report shows planned revenue by percentage for product line. This report is optimized for mobile devices. This report uses the following features:

- drill down
- query calculations
- crosstabs
- lists
- pie charts
- tables
- custom headers and footers

## Sales Revenue Expectation

This report shows the goal for the percentage change in sales revenue for retailers. It uses a calculated value for forecasted revenue. This report is optimized for mobile devices. This report uses the following features:

- crosstabs
- bar charts
- line charts
- pie charts

- calculations
- custom headers and footers
- text items

## Same Month Prior Year

This report shows sales volume by product line in one or more months. The report is filtered by a prompt for month. The report generates totals for the selected months and for the same months in the prior year. This report uses the following features:

- crosstabs
- prompts
- custom headers and footers

## Selected Retailer Country

This report uses the revenue from a selected country as a baseline value for a set of countries. A chart shows the difference in revenue for each country as it compares to the base country. The report is filtered by a prompt for country. This report uses the following filters:

- crosstabs
- bar charts
- tables to control where objects appear

## Top Retailers by Country

This report shows the top 10 retailers by country. It is used as source for drill-through to the Total Revenue by Country report. This report uses the following features:

- crosstabs
- prompts
- filters
- line charts
- prompt pages
- query calculations
- calculations
- singletons
- bar charts
- custom headers and footers

## Tree Prompt Retailers Set

This report shows the revenue for the retailers set. This report is optimized for mobile devices. It is a drill-through target for the Sales Revenue Expectation report. This report uses the following features:

- crosstabs
- tree prompts
- combination charts
- prompt pages
- query calculations
- custom headers and footers

## Samples in the GO Data Warehouse (analysis) Package

The following reports are some of the reports found in the GO Data Warehouse (analysis) package.

### Budget vs. Actual

This report shows three years of data by retailer and retailer site for the camping equipment product line. Each year includes budget and actual data. This report uses the following features:

- summarizing
- crosstabs
- context filters

### Customer Returns and Satisfaction

This report shows the customer satisfaction survey results for Asia Pacific in 2007. It highlights the customers who are the least satisfied. It also provides information about customers with the highest number of product returns. This report uses the following features:

- combination charts
- customizing the color and size of a chart
- lists
- formatting a list
- conditional highlighting
- filters
- custom headers and footers
- colors
- lineage

- text items
- grouping
- baselines
- summarizing
- calculations
- drilling through

## Employee Satisfaction 2006

This report shows employee satisfaction survey results by department, compared to targets and industry standards. It also shows employee rankings and terminations. This report uses the following features:

- crosstabs
- conditional highlighting
- combination charts
- lineage
- text items
- calculations

## Employee Training by Year

This report shows employee training data for the selected year and quarter(s). A bar chart shows training costs by region and a crosstab shows data for the selected quarter(s). This report uses the following features:

- context filters
- cascading prompts
- bar charts
- customizing the color of a chart
- crosstabs
- calculations

## Eyewear Revenue by Brand and Size

This report shows a summary of eyewear revenue by brand and compares two prompted retailer sites. The report is filtered by prompts for region, retailer type, and year. This report uses the following features:

- prompts

- bar charts
- lists
- conditional styles
- calculations
- text items
- custom headers and footers
- combination charts
- axis titles
- crosstabs
- grouping

## Global Bonus Report

This list report shows employees who received more than \$2,500 bonus in a year by region. It is grouped by country. It also shows how much the sales target was exceeded for each region. This report uses the following features:

- lists
- page sets (page breaks by country with different sorting and grouping)
- multiple prompts and parameters
- calculations
- filters
- conditional highlighting
- hidden objects
- lineage

## GO Balance Sheet as at Dec 31 2006

This is the Balance sheet report for Americas where current year data is compared to the previous year data. Analysts can see negative trends under Variance where negative percentages are highlighted. This report uses the following features:

- conditional highlighting
- padding
- crosstabs
- text items
- context filters

## Great Outdoors Company Balance Sheet as at Dec 31 2006

This report shows a simple balance sheet with assets, liabilities, and equity for 2006 with a 2005 comparative. It uses Business Insight Advanced. The Getting Started guide provides a step-by-step example of how to create this report. This report uses the following features:

- crosstabs

## Manager Profile

This report shows information about managers, including salary, bonuses, and all compensations grouped by year. This report uses the following features:

- column charts
- lists
- grouping
- summarizing
- custom chart palette
- prompts

## Planned Headcount

This chart report shows headcount variance compared to the plan for each organization for 2004.

This report uses the following features:

- progressive column charts
- templates
- hidden objects
- custom headers and footers
- lists
- baselines

## Positions to Fill

This report shows a list of department names, positions, longest days to fill the positions, and ranking. The report uses a prompt for the year and is a drill-through target for the Recruitment report. This report uses the following features:

- combination charts
- lists
- prompts
- baselines

## Promotion Success

This report shows the financial results of the company's promotions. It shows how much of the company's total revenue is attributable to each promotional campaign. This report uses the following features:

- prompt pages
- HTML items
- summarizing
- axis titles
- bar charts
- lists
- grouping

## Quantity Sold vs. Shipped and Inventory

This report compares the quantity of goods sold and shipped with the opening and closing inventory levels. This report uses the following features:

- filters
- combination charts
- defined y-axes
- custom headers and footers

## Recruitment Report

This report shows a variety of recruitment techniques for certain positions or organizations. This report uses the following features:

- drilling through
- crosstabs
- prompt pages
- colors
- floating object adjustment
- custom headers and footers

## Return Quantity by Order Method

This report shows quantity sold, number of returns, and percentage of returns (with those greater than 5% highlighted) by return reason for each product in the Outdoor Protection product line. This report uses the following features:

- filters
- lists
- conditional highlighting
- grouping

## Returned Items

This report shows the number of returned items by return reason and retailer type. A column chart shows returned items by product line and region for the selected date range. This report uses the following features:

- date and time prompts
- crosstabs
- bar charts
- drilling down
- text items
- sorting

## Returns by Damage, Failed Orders and Complaints in 2006

This report shows quality measures based on product returns. This report uses the following features:

- pie charts
- crosstabs
- indented text
- singletons
- calculations
- drilling through
- text items
- custom headers and footers

## Returns by Failed Orders in 2006

This report shows quality measures based on product returns and focuses on failed orders. This report uses the following features:

- pie charts
- crosstabs
- indented text

- singletons
- drilling through
- calculations, including the `tuple` function

## Returns by Order Method

This report shows product returns and reasons filtered on the order method. The Getting Started guide provides a step-by-step example of how to create this report. This report uses the following features:

- bar charts
- prompts
- crosstabs
- filters
- custom headers and footers

## Revenue by GO Subsidiary 2005

This prompted chart report shows 2005 quarterly revenues for each GO subsidiary. This report uses the following features:

- templates
- colors
- prompts
- hyperlinks
- customizing charts
- singletons
- bar charts
- drilling through
- layout calculations
- pie charts
- calculations
- combination charts
- text items
- blocks
- sorting

## Sales Commissions for Central Europe

This report shows an annual summary of sales commissions, revenues, and gross profit for each branch in Central Europe. It also compares actual commission expenses with planned commission expenses. This report uses the following features:

- prompts
- calculations
- bar charts
- lists
- conditional highlighting
- drilling through
- custom headers and footers
- axis titles

## Sales Growth Year Over Year

This report shows annual sales growth in both percentage and dollar amounts. This report uses the following features:

- bar charts
- lists
- filters
- sorting
- baselines
- axis titles

## Succession Report

This report shows the succession data by department and status for percent ready in a column chart. It also contains a detailed crosstab for the managers associated with the possible successors. This report uses the following features:

- drilling through to the Manager Profile report
- filters
- lists
- grouping

## Top 10 Retailers for 2005

This report shows the top 10 retailers for 2005 by revenue and sales target. This report uses the following features:

- bar charts
- lists
- filters
- multiple queries
- combination charts
- line charts
- notes
- axis titles
- text items
- custom headers and footers

## Samples in the GO Data Warehouse (query) Package

The following reports are some of the reports found in the GO Data Warehouse (query) package.

### Bursted Sales Performance Report

This list report shows how to burst a product sales report to a sales manager for Northern Europe sales staff. To successfully burst this report, IBM® Cognos® BI must be configured to use an email server. This report uses the following features:

- lists
- bursting
- conditional highlighting
- filters
- calculations
- summarizing
- blocks
- custom headers and footers
- sorting
- grouping

## Employee Expenses

This report is used as a data source for the Employee Expenses Power Cube. This report uses the following features:

- lists

## Health Insurance

This report is used as a data source for the Employee Expenses Power Cube. This report uses the following features:

- lists
- filters

## Pension Plan

This report is used as a data source for the Employee Expenses Power Cube. This report uses the following features:

- lists
- filters

## Regular Salary

This report is used as a data source for the Employee Expenses Power Cube. This report uses the following features:

- lists
- filters

## TOC Report

This report takes advantage of the bookmark object to allow a user to navigate through this report easily. This report should be run in PDF or saved HTML format. The report contents show a product order table and an expected volume fact table. This report uses the following features:

- lists
- bookmarks
- background color
- multiple pages
- grouping

## Total Revenue by Country

This report summarizes revenue for Retailer Country and Product Line. It is also a drill-through target for the Top Retailers by Country and Revenue by Order Method reports. This report uses the following features:

- crosstabs
- combination charts
- summarizing
- tables to control where objects appear

## Samples in the GO Sales (analysis) Package

The following reports are some of the reports found in the GO Sales (analysis) package.

### 2005 Quarterly Sales Forecast

This report shows the sales forecast by product line and region for each quarter in 2005. This report uses the following features:

- lists
- summarizing
- grouping
- sorting

### 2005 Sales Summary

This report summarizes revenue and gross profit for 2005 and shows the top sales representatives by revenue and quantity sold. This report uses the following features:

- lists
- filters
- combination charts
- axis titles
- custom headers and footers
- conditions

### Briefing Book

This report shows a Briefing Book style of report. This report uses the following features:

- multiple pages
- crosstabs

- multiple queries
- filters
- pie charts
- singletons
- tables of contents
- bookmarks
- PDF options
- horizontal pagination
- sorting
- custom headers and footers
- text items

## Horizontal Pagination

This report shows crosstabs rendered across several horizontal pages. The first crosstab shows the fit-to-page behavior while the second crosstab shows the horizontal pagination. This report uses the following features:

- multiple pages
- horizontal pagination
- crosstabs
- custom headers and footers

## No Data

Each page of this report presents a different option for dealing with a No Data condition. It also generates invoices of sales for the Order Invoices - Donald Chow, Sales Person report in the GO Sales (query) package. This report uses the following features:

- crosstabs
- custom headers and footers
- no data
- lists

## PDF Page Properties

The two pages of this report appear with different Page Orientation (portrait and landscape) when the report is run in PDF format. This report uses the following features:

- crosstabs

- lists
- page orientation
- PDF options
- custom headers and footers

## Singletons on Page Body

This report uses singleton results to display information with no data relationship in the same layout context. This report uses the following features:

- singletons
- tables
- custom headers and footers

## Table of Contents

This report shows two Tables of Contents: one for the main pages and another for the appendices. This report uses the following features:

- crosstabs
- pie charts
- bookmarks
- tables
- tables of contents
- custom headers and footers
- hyperlinks

## Samples in the GO Sales (query) Package

The following reports are some of the reports found in the GO Sales (query) package.

### Order Invoices - Donald Chow, Sales Person

This report generates invoices for all the sales by Donald Chow. This report uses the following features:

- lists
- adding list row cells
- calculations
- formatting tables

- calculations
- filters
- grouping
- tables to control where objects appear

## Samples in the IBM Cognos Statistics Package

The following reports are some of the sample reports found in IBM® Cognos® Statistics.

IBM Cognos Statistics contains the following data sources:

- GOSALEDW
- BANKLOAN\_CS
- CATALOG\_SALES
- CLOTHING\_DEFECTS
- DISCHARGEDATA
- DVDPPLAYER
- SHAMPOO\_PH

### Advertising Costs on Sales Revenue

This report uses a linear regression statistical object to show the impact of advertising costs on sales revenue.

This report uses the following features:

- Linear regression
- crosstabs

### Catalog Sales

This report uses a correlation statistical object to summarize the relationship between two critical business variables in a retail organization from 1999-2008.

This report uses the following features:

- correlation
- lists

### Clothing Manufacturer Quality

This report uses a control chart statistical object to monitor the clothing manufacturing process to ensure that the proportion of defective clothing is consistent over time and across batches.

This report uses the following features:

- a p,np control chart

## Discharged Patients

This report uses a one-way chi-square statistical object to report analyze whether the number of patients that are discharged varies by day of week.

This report uses the following features:

- a one-way chi-square test
- combination charts

## DVD Score

This report uses a one-way ANOVA statistical object to discover if consumers of various ages rate the design of a DVD player differently.

This report uses the following features:

- one-way ANOVA
- pie charts

## Income Growth

This report uses a boxplot object to understand the key influencers of income.

This report uses the following features:

- a boxplot
- curve estimation
- column charts
- lists

## Shampoo pH Level

This report uses a control chart statistical object to monitor the processing of pH level in shampoo production.

This report uses the following features:

- an X-bar chart and an R chart

## Interactive Samples

The following reports are some of the reports found in the Interactive Samples folder.

## Bursted Sales Performance Report

This list report shows how to burst a product sales report to a sales manager for Northern Europe sales staff. To successfully burst this report, IBM® Cognos® BI must be configured to use an email server. This report uses the following features:

- lists
- bursting
- conditional highlighting
- filters
- calculations
- summarizing
- blocks
- custom headers and footers
- sorting
- grouping

## Percentage Calculation (by year)

This prompted report shows a percentage calculation based on a particular year. This report uses the following features:

- lists
- pie charts

## Recruitment Report

This report shows a variety of recruitment techniques for certain positions or organizations. This report uses the following features:

- drilling through
- crosstabs
- prompt pages
- colors
- floating object adjustment
- custom headers and footers

## Revenue by GO Subsidiary 2005

This prompted chart report shows 2005 quarterly revenues for each GO subsidiary. This report uses the following features:

- templates
- colors
- prompts
- hyperlinks
- customizing charts
- singletons
- bar charts
- drilling through
- layout calculations
- pie charts
- calculations
- combination charts
- text items
- blocks
- sorting

## **Rolling and Moving Averages**

This report shows the rolling and moving average count for the return quantity. A prompt uses a macro to provide static choices within a time dimension. This report uses the following features:

- prompts
- calculations
- crosstabs
- combination charts

---

# **Chapter 25: Limitations When Producing Reports in Microsoft Excel Format**

---

There are limitations when producing reports in Microsoft® Excel format.

## **Unable to Load Images from the IBM Cognos BI Content Store in a Report**

If a report contains an image whose URL points to the IBM® Cognos® Business Intelligence content store, the Microsoft® Excel spreadsheet software generates an access violation error and shuts down.

This problem is a known issue in the Microsoft knowledge base, and Microsoft is currently investigating the problem. This problem occurs only in Excel 2002.

## **Blank Worksheet Appears**

If the Microsoft® Excel spreadsheet software cannot download a worksheet within a timeout period, Excel may instead open a blank worksheet.

## **Warning Message Appears When Excel Opens an IBM Cognos Business Intelligence Report**

Each time the Microsoft® Excel spreadsheet software opens an IBM® Cognos® BI report, the following message appears:

Some of the files in this Web page are not in the expected location. Do you want to download them anyway? If you are sure the Web page is from a trusted source, click Yes.

The Excel workbook in HTML/XML format requires the presence of the file filelist.xml. IBM Cognos BI does not allow the creation of local files on the client side. In addition, a local file that contains URLs introduces a security issue. Consequently, this message will appear whenever you open an IBM Cognos BI report in Excel. If you see this error message, click Yes to open the report.

## **Spreadsheet Content Not Saved for Reports Saved in XLS Format**

If you open a report that was saved in XLS format or run a report in XLS format, and security settings in your Web browser are set so that you are prompted to open or save the report, do not click Save. If you save the report, the spreadsheet content will not be saved. This is because Microsoft® Excel reports in Microsoft Office 2000 HTML format use relative paths to the spreadsheets. The relative URL paths are no longer available when you open a saved XLS report.

Instead, click Open first and then choose to save the report.

## Unable to Load Excel Report in Netscape 7.01

This version of IBM® Cognos® Business Intelligence does not support loading Microsoft® Excel reports in Netscape 7.01.

## Unable to Nest Labels in Charts

Currently, it is not possible to specify nested labels for the category axis via XML.

## Data Series Are Truncated

The Microsoft® Excel spreadsheet software may group data series or categories differently when compared to a chart produced by IBM® Cognos® Business Intelligence.

A 1 KB buffer limit in Excel limits the maximum number of data series per chart to 120. Data series over 120 are truncated.

## Colors Different From Those in HTML or PDF

If an IBM® Cognos® Business Intelligence report contains a chart that uses custom colors, the Microsoft® Excel spreadsheet software may not be able to add the custom color to the Excel color palette. Excel will attempt to match the custom color to one of its available standard colors. As a result, colors in Excel may vary from those seen in HTML or PDF.

We recommend that you use standard colors in charts.

## Repeating Pie Charts

If you have a report that has repeating pie charts and you define a chart title, the Microsoft® Excel spreadsheet software will show each pie with a title that is a concatenation of the chart title and the data series. For example, if the chart title is Quantity Sold by Order Method and Product Line and the data series is Order method, the title of each pie in Excel will be Quantity Sold by Order Method and Product Line, *order method*.

## Unable to Skip Discrete Axis Labels

In IBM® Cognos® Business Intelligence charts, you can control the skipping of discrete axis labels. This feature is not supported in Microsoft® Excel charts.

## Unsupported IBM Cognos Business Intelligence Formatting

About 30% of the formatting functions available in IBM® Cognos® BI are not supported in the Microsoft® Excel spreadsheet software. In particular, Excel does not allow changing locale-dependent formatting attributes, such as the following:

- Decimal Separator
- Exponential Symbol
- Group Separator
- Monetary Decimal Separator
- AM String
- Day Name
- Day Short Name
- Decimal Delimiter Symbol
- Month Name
- Month Short Name
- PM String

In addition, Excel does not support the following:

- Format Width
- International Currency Symbol
- List Separator
- Percent Symbol (Excel does not support percent symbols for charts)
- Multiplier
- Overline Text Format
- PerMill Symbol
- Plus Sign
- Scale (Excel has a different scaling formula than IBM Cognos BI)
- Calendar (Excel does not allow changing the calendar)
- Era Name
- First Day Of Week
- Show Era

## Cells Contain Series of #

Cells in the Microsoft® Excel spreadsheet software have a limit of 255 characters. If your report contains text strings that are longer than 255 characters, they will be formatted as text and appear as #####.

To resolve this problem, use fewer characters.

## Excel Cannot Render Reports with More Than 256 Columns

The Microsoft® Excel spreadsheet software limits the size of a worksheet size to 65,536 rows by 256 columns. If your report contains more than 65,536 rows, it is split into multiple worksheets. The number of worksheets that your report can contain is limited by the physical memory of your computer. If your report contains more than 256 columns, the following error occurs:

*Reports with more than 256 columns cannot be rendered in Excel.*

## Table and Column Widths

The Microsoft® Excel spreadsheet software does not support using percentages to determine the width of tables. If the report contains only one table, the value of the width attribute for the Table element in the report specification determines the width of the table in the Excel worksheet. If the report contains more than one table, Excel determines the width of all the tables in the worksheet. If the tables are nested, the width specified for the outer table is used and, if necessary, the width is adjusted to accommodate data in the nested tables. The columns and rows around the table are merged to preserve the appearance of the nested table. When you save the workbook, only a single table is saved per worksheet.

## Secure Socket Layer (SSL) Is Not Supported in Some Excel Formats and Versions

SSL is supported for only the Microsoft® Excel 2002 format in Microsoft Excel 2002 and Microsoft Excel 2003.

## Number Formats Become Currency Formats in Japanese Excel

A report uses the Number data format and you save it as Microsoft® Excel output. When you open the report in the Japanese version of Microsoft Excel, the data format is listed as Currency rather than Number. This occurs because Japanese Excel interprets the standard Number data format slightly differently than other versions of Excel.

The value appears correctly in Number format. For example, if you specified five digits as your number format, five digits still appear. In Excel, click the **Custom** number format to see the exact format string being used.

## Reports Show Data in Wrong Columns

A report contains a large amount of data that is presented using a large number of nested report objects, such as tables and blocks. When the report is produced in Microsoft® Excel format, some of the data appears in the wrong columns. This occurs because Excel has a 64K limit on how many nested cell objects can appear in a single spreadsheet.

To solve this problem, you can redesign the report to present the data using non-nested structures.

## Unable to Access Reports on Remote Servers

You cannot access a report in Microsoft® Excel format on a remote server.

To resolve this problem, you must change the hostname portion of the gateway URI from localhost to either the IP address of the computer or the computer name. You do this using IBM® Cognos® Configuration.

## Drill-through Reports Are Not Supported in Excel

IBM® Cognos® Business Intelligence does not support drill-through for reports in Microsoft® Excel format.

## Map Reports Are Not Supported in Excel

IBM® Cognos® Business Intelligence does not support map reports in Microsoft® Excel format.

## Unsupported Excel Formatting

IBM® Cognos® Business Intelligence does not support the following formatting functions available in the Microsoft® Excel spreadsheet software:

- background images in table cells
- Excel-specific headers and footers
- text flow and justification
- floating text objects
- white space, normal, and wrap text formatting
- maximum characters

Some layouts do not show exactly in HTML and PDF due to Excel limitations.

## Hyperlink Buttons Are Not Supported in Excel

The Microsoft® Excel spreadsheet software does not support hyperlink buttons.

## Unable to View Reports in Excel Format Sent as Email Attachments

IBM® Cognos® Business Intelligence can send Microsoft® Excel reports in HTML and XML format by email. However, you cannot open them directly from the email message.

Save the Excel email attachments to your computer and view them from there.

## Unsupported Chart Properties in Excel

The following IBM® Cognos® Business Intelligence chart properties are not supported in the Microsoft® Excel spreadsheet software:

- tool tips
- conditional text
- depth
- visual angle
- show values
- marker text location
- show baseline
- new note
- new marker
- truncation text and allow n-degrees rotation category labels
- border
- margin
- box type
- font and font alignment
- footer
- subtitle
- regression line
- baseline
- Include Zero For Auto Scale

In addition, IBM Cognos BI makes sure that Excel reuses the same color palette that is defined in IBM Cognos BI. However, Excel can only use the first 16 colors from the IBM Cognos BI palette. If the number of categories in a chart exceeds 16, the rest of the colors are taken from the default Excel palette.

## Unsupported Chart Types in Excel

About 30% of the chart types available in IBM® Cognos® Business Intelligence are not matched in the Microsoft® Excel spreadsheet software.

The following chart types appear differently or are not supported. Charts that are not supported appear as a default column chart in Excel.

- Bubble charts

Excel does not support regression lines.

- Combination charts

In Excel, combination charts appear as two-dimensional. If a three-dimensional or two-dimensional combination chart includes only an area, column, or line chart, only the one chart appears.

- Gauge charts

- Donut charts

Excel may fill in the donut hole to accommodate extra measures. Excel shows donut three-dimensional charts as donut charts.

- Maps

In Excel, no chart appears.

- Metrics range charts

In Excel, they appear as combination charts with two line charts for the tolerance bars.

- Pareto charts

In Excel, the cumulative line is not displayed.

- Pie charts

Excel shows only one type of pie chart and may show it at a different angle.

- Progressive column charts

- Polar charts

In Excel, a scatter chart appears instead.

- Quadrant charts

- Radar charts

Excel does not support stacked area radar charts. Excel names area radar charts as filled radar charts. Excel shows radar charts as radar with markers.

- Scatter charts

In Excel, three-dimensional scatter charts appear as two-dimensional scatter charts. The z-axis is dropped.

- 100% stacked bar and column charts

In Excel, the percentage of each value in the bar or column is not displayed. Instead, the underlying values are displayed.

## Cell Height and Width Are Incorrect

The width and height of cells that contain data with curly brackets {} or parentheses () may appear incorrectly. This is because the Microsoft® Excel spreadsheet software uses different word wrapping algorithms than IBM® Cognos® Business Intelligence.

---

# Chapter 26: Report Studio Object and Property Reference

---

This appendix contains definitions of the objects and properties found in IBM® Cognos® Report Studio. They are available contextually, by pressing F1 when an object or property is active in the Report Studio authoring environment.

The following objects and properties are referenced:

- [Report Studio Objects](#)

These objects are visible in the Report Studio work area. They can be inserted from the **Toolbox** tab .

- [Report Studio Properties](#)

These properties can be set on Report Studio objects. They are available in the lower-left pane of Report Studio.

- [Data Formatting Properties](#)

These properties can be set on data values by selecting **Layout Data Format** from the **Data** menu, or by editing the **Data Format** property for Report Studio objects.

## Report Studio Objects

The following is a list of objects available in Report Studio.

### 3-D Area

A chart in which members of a data series are represented by three-dimensional areas of varying size and color. The three-dimensional area chart is one of three chart types that can be included in a three-dimensional combination chart.

#### Properties of 3-D Area

[Border Color](#), [Borders](#), [Chart Type](#), [Values](#)

### 3-D Bar

A chart in which members of a data series are represented by three-dimensional bars of varying length and color. The three-dimensional bar chart is one of three chart types that can be included in a three-dimensional combination chart.

#### Properties of 3-D Bar

[Border Color](#), [Borders](#), [Chart Type](#), [Values](#)

## 3-D Combination Chart

A chart that includes any number and combination of three-dimensional bar charts, line charts, and area charts. This chart plots any number of data series against one common measure on the vertical numeric axis and one or two common data series on ordinal axes.

### Properties of 3-D Combination Chart

3-D Viewing Angle, Alternate Text, Background Color, Background Effects, Background Image, Baselines, Border, Box Type, Classes, Clickable Regions, Conditional Palette, Conditional Styles, Container Select, Drill-Through Definitions, Floating, Font, Footer, Foreground Color, Margin, Master Detail Relationships, Maximum Hotspots, Name, No Data Contents, Notes, Numerical Axis, Padding, Pagination, Palette, Query, Relative Alignment, Render Page when Empty, Render Variable, Series Color, Set Variable Values, Show Hover, Show Pointer Cursor, Size & Overflow, Style Variable, Subtitle, Suppression, Title, Title, Tooltips, Visible, X Axis, Y Axis

## 3-D Line

A chart in which members of a data series are represented by three-dimensional lines of varying colors. The three-dimensional line chart is one of three chart types that can be included in a three-dimensional combination chart.

### Properties of 3-D Line

Border Color, Borders, Chart Type, Values

## 3-D Scatter Chart

A chart that plots three measures against one or more data series on a three-dimensional X-Y-Z graph.

### Properties of 3-D Scatter Chart

3-D Viewing Angle, Alternate Text, Background Color, Background Effects, Background Image, Baselines, Border, Border Color, Borders, Box Type, Classes, Clickable Regions, Conditional Palette, Conditional Styles, Container Select, Drill-Through Definitions, Floating, Font, Footer, Foreground Color, Legend, Margin, Marker Size (pt), Master Detail Relationships, Maximum Hotspots, Name, No Data Contents, Notes, Padding, Pagination, Palette, Point Shape, Query, Relative Alignment, Render Page when Empty, Render Variable, Set Variable Values, Show Feelers, Show Hover, Show Pointer Cursor, Size & Overflow, Style Variable, Subtitle, Suppression, Title, Title, Tooltips, Values, Visible

## Angular Axis

The angular numeric axis for a polar chart, including labels, titles, range, and scale.

### Properties of Angular Axis

Axis Labels, Axis Line, Axis Title, Axis Title, Classes, Conditional Styles, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

## Angular Measure

The measure that determines the angular position of each data marker on a polar chart.

### Properties of Angular Measure

[Aggregate Function](#), [Conditional Styles](#), [Custom Label](#), [Data Format](#), [Expression](#), [Hierarchy](#), [Hierarchy Unique Name](#), [Label](#), [Level](#), [Level Unique Name](#), [Member](#), [Member Unique Name](#), [Name](#), [Property Unique Name](#), [Rollup Aggregate Function](#), [Root Members Only](#), [Set Sorting](#), [Style Variable](#), [Type](#)

## Area

The data marker used to represent data series in an area chart.

### Properties of Area

[Aggregate Function](#), [Area Shape](#), [Border Color](#), [Borders](#), [Chart Type](#), [Conditional Palette](#), [Expression](#), [Extend Width](#), [Hierarchy](#), [Hierarchy Unique Name](#), [Label](#), [Level](#), [Level Unique Name](#), [Member](#), [Member Unique Name](#), [Name](#), [Palette](#), [Property Unique Name](#), [Rollup Aggregate Function](#), [Root Members Only](#), [Series Type](#), [Set Sorting](#), [Show Values](#), [Type](#)

## Area

The data marker used to represent data series in an area chart. This object applies only to legacy charts.

### Properties of Area

[Axis Assignment](#), [Border Color](#), [Borders](#), [Chart Type](#), [Grouping Type](#), [Value Location](#), [Values](#), [Value Type](#)

## As of Time Expression

An expression that produces a Date-Time value. This expression can be used to show report results for a specific time period that is defined by an expression that you create.

### Properties of As of Time Expression

[Report Expression](#)

## Axis Labels

The labels that appear on an axis.

### Properties of Axis Labels

[Auto Font Size](#), [Classes](#), [Conditional Styles](#), [Data Format](#), [Font](#), [Foreground Color](#), [Nested Label Display](#), [Render Variable](#), [Text Truncation](#)

## Axis Title

The title for an axis of the chart. This object applies only to legacy charts.

## Properties of Axis Title

[Box Type](#), [Classes](#), [Conditional Styles](#), [Default Title](#), [Font](#), [Foreground Color](#), [Master Detail Relationships](#), [Properties](#), [Query](#), [Style Variable](#), [Visible](#)

## Axis Title

The title for an axis of the chart.

## Properties of Axis Title

[Auto Font Size](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Default Title](#), [Font](#), [Foreground Color](#), [Master Detail Relationships](#), [Properties](#), [Query](#), [Render Variable](#)

## Bar

A chart in which members of a data series are represented by bars of varying length and color.

## Properties of Bar

[Aggregate Function](#), [Bar Shape](#), [Bar Width](#), [Bevel](#), [Border Color](#), [Borders](#), [Chart Type](#), [Conditional Palette](#), [Expression](#), [Hierarchy](#), [Hierarchy Unique Name](#), [Label](#), [Level](#), [Level Unique Name](#), [Member](#), [Member Unique Name](#), [Name](#), [Palette](#), [Property Unique Name](#), [Rollup Aggregate Function](#), [Root Members Only](#), [Series Type](#), [Set Sorting](#), [Show Values](#), [Type](#)

## Bar

A chart in which members of a data series are represented by bars of varying length and color. This object applies only to legacy charts.

## Properties of Bar

[Axis Assignment](#), [Border Color](#), [Borders](#), [Chart Type](#), [Connecting Lines](#), [Grouping Type](#), [Value Location](#), [Values](#), [Value Type](#)

## Baseline

A baseline to be rendered on a chart.

## Properties of Baseline

[Aggregate Function](#), [Axis Assignment](#), [Bar](#), [Line](#), or [Area Index](#), [Box Type](#), [Data Item Value](#), [Expression](#), [Hierarchy](#), [Hierarchy Unique Name](#), [Label](#), [Label](#), [Legend Label](#), [Level](#), [Level Unique Name](#), [Line Styles](#), [Master Detail Relationships](#), [Member](#), [Member Offset \(%\)](#), [Member Unique Name](#), [Name](#), [Numeric Value](#), [Percentile](#), [Percent of Axis](#), [Position type](#), [Properties](#), [Property Unique Name](#), [Query](#), [Report Expression](#), [Report Expression](#), [Rollup Aggregate Function](#), [Root Members Only](#), [Set Sorting](#), [Standard Deviations](#), [Type](#)

## Baseline

A baseline for a polar chart, scatter chart, or bubble chart.

## Properties of Baseline

Aggregate Function, Axis Assignment, Data Item Value, Expression, Hierarchy, Hierarchy Unique Name, Label, Label, Legend Label, Level, Level Unique Name, Line Styles, Master Detail Relationships, Member, Member Unique Name, Name, Numeric Value, Percentile, Percent of Axis, Position type, Properties, Property Unique Name, Query, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Standard Deviations, Type

## Baseline

A baseline for a three-dimensional combination chart.

## Properties of Baseline

Aggregate Function, Bar, Line, or Area Index, Box Type, Data Item Value, Expression, Hierarchy, Hierarchy Unique Name, Label, Level, Level Unique Name, Line Styles, Master Detail Relationships, Member, Member Unique Name, Name, Numeric Value, Percentile, Percent of Axis, Position type, Properties, Property Unique Name, Query, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Standard Deviations, Type

## Baseline

A baseline for a three-dimensional scatter chart.

## Properties of Baseline

Aggregate Function, Box Type, Data Item Value, Expression, Hierarchy, Hierarchy Unique Name, Label, Level, Level Unique Name, Line Styles, Master Detail Relationships, Member, Member Unique Name, Name, Numeric Value, Percentile, Percent of Axis, Position type, Properties, Property Unique Name, Query, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Standard Deviations, Type

## Block

A container into which you can insert other objects.

## Properties of Block

Background Color, Background Effects, Background Image, Border, Box Type, Classes, Conditional Styles, Floating, Font, Foreground Color, Horizontal Alignment, Margin, Name, Padding, Render Variable, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Visible, White Space

## Bookmark

A link to another area within the same report. The link can be defined as a static value, a query item, or as the result of a report expression.

## Properties of Bookmark

Data Item Label, Data Item Value, Label, Report Expression, Source Type

## Bubble Chart

A point chart that plots one or more data series against three measures: a Y-value, an X-value, and a bubble whose relative size represents the third measure. Multiple points are plotted for each category. This object applies only to legacy charts.

To help distinguish values, set the Tooltips property of this object to Yes.

### Properties of Bubble Chart

[Alternate Text](#), [Background Color](#), [Background Effects](#), [Background Image](#), [Baselines](#), [Border](#), [Border Color](#), [Borders](#), [Box Type](#), [Bubble Size](#), [Classes](#), [Clickable Regions](#), [Conditional Palette](#), [Conditional Styles](#), [Container Select](#), [Drill-Through Definitions](#), [Floating](#), [Font](#), [Footer](#), [Foreground Color](#), [Legend](#), [Margin](#), [Markers](#), [Marker Text Location](#), [Master Detail Relationships](#), [Maximum Hotspots](#), [Name](#), [No Data Contents](#), [Notes](#), [Padding](#), [Pagination](#), [Palette](#), [Point Shape](#), [Query](#), [Regression Line](#), [Relative Alignment](#), [Render Page when Empty](#), [Render Variable](#), [Set Variable Values](#), [Show Hover](#), [Show Pointer Cursor](#), [Size & Overflow](#), [Style Variable](#), [Subtitle](#), [Suppression](#), [Title](#), [Title](#), [Toolips](#), [Value Location](#), [Values](#), [Visible](#), [X Axis](#), [Y Axis](#)

## Bubble Chart

A point chart that plots one or more data series against three measures: a Y-value, an X-value, and a bubble whose relative size represents the third measure. Multiple points are plotted for each category.

To help distinguish values, set the Tooltips property of this object to Yes.

### Properties of Bubble Chart

[Alternate Text](#), [Background Effects](#), [Border](#), [Box Type](#), [Bubble Size](#), [Classes](#), [Clickable Regions](#), [Color by Value](#), [Colored Regions](#), [Conditional Palette](#), [Conditional Styles](#), [Container Select](#), [Drill-Through Definitions](#), [Drop Shadow](#), [Floating](#), [Font](#), [Footer](#), [Foreground Color](#), [Legend](#), [Margin](#), [Markers](#), [Master Detail Relationships](#), [Material Effects](#), [Matrix rows and columns](#), [Maximum Hotspots](#), [Name](#), [No Data Contents](#), [Notes](#), [Numeric Baselines](#), [Padding](#), [Pagination](#), [Palette](#), [Plot Area Fill](#), [Query](#), [Relative Alignment](#), [Render Page when Empty](#), [Render Variable](#), [Set Variable Values](#), [Show Hover](#), [Show Pointer Cursor](#), [Show Values](#), [Size & Overflow](#), [Subtitle](#), [Suppression](#), [Title](#), [Toolips](#), [Trendlines](#), [Visible](#)

## Bubble Measure

The measure that determines the size of each bubble on a bubble chart.

### Properties of Bubble Measure

[Aggregate Function](#), [Conditional Styles](#), [Custom Label](#), [Data Format](#), [Expression](#), [Hierarchy](#), [Hierarchy Unique Name](#), [Label](#), [Level](#), [Level Unique Name](#), [Member](#), [Member Unique Name](#), [Name](#), [Property Unique Name](#), [Rollup Aggregate Function](#), [Root Members Only](#), [Set Sorting](#), [Type](#)

## Bubble Measure

The measure that determines the size of each bubble on a bubble chart. This object applies only to legacy charts.

## Properties of Bubble Measure

[Aggregate Function](#), [Conditional Styles](#), [Custom Label](#), [Data Format](#), [Expression](#), [Hierarchy](#), [Hierarchy Unique Name](#), [Label](#), [Level](#), [Level Unique Name](#), [Member](#), [Member Unique Name](#), [Name](#), [Property Unique Name](#), [Rollup Aggregate Function](#), [Root Members Only](#), [Set Sorting](#), [Style](#), [Variable](#), [Type](#)

## Bullet Chart

A chart that compares a measure to a target. Use the Colored Regions property to relate the compared measures against additional qualitative measurements, such as colored regions for good, satisfactory, and poor.

## Properties of Bullet Chart

[Alternate Text](#), [Background Effects](#), [Bevel](#), [Border](#), [Box Type](#), [Bullet Indicators](#), [Chart Orientation](#), [Classes](#), [Clickable Regions](#), [Colored Regions](#), [Conditional Styles](#), [Container Select](#), [Drill-Through Definitions](#), [Drop Shadow](#), [Floating](#), [Font](#), [Footer](#), [Foreground Color](#), [Legend](#), [Margin](#), [Master Detail Relationships](#), [Matrix rows and columns](#), [Maximum Hotspots](#), [Name](#), [No Data Contents](#), [Notes](#), [Padding](#), [Pagination](#), [Query](#), [Relative Alignment](#), [Render Page when Empty](#), [Render Variable](#), [Set Variable Values](#), [Show Hover](#), [Show Pointer Cursor](#), [Size & Overflow](#), [Subtitle](#), [Suppression](#), [Title](#), [Toolips](#), [Visible](#)

## Bullet Measure

The measure that determines the size of the bullet for a bullet chart.

## Properties of Bullet Measure

[Aggregate Function](#), [Custom Label](#), [Data Format](#), [Expression](#), [Hierarchy](#), [Hierarchy Unique Name](#), [Label](#), [Level](#), [Level Unique Name](#), [Member](#), [Member Unique Name](#), [Name](#), [Property Unique Name](#), [Rollup Aggregate Function](#), [Root Members Only](#), [Set Sorting](#), [Type](#)

## Button

A button in a static button bar control.

## Properties of Button

[Icon](#), [Label](#)

## Button

A single static button control. The button requires a row in the data table of the control.

## Properties of Button

[Button Definition](#), [Control Enable](#), [Icon](#), [Label](#), [Name](#), [Render Variable](#), [Set Variable Values](#), [Size](#)

## Button Bar

A static button bar control. Each button requires a corresponding row in the data table of the control.

### **Properties of Button Bar**

[Button Gap](#), [Buttons Definition](#), [Button Width](#), [Container Select](#), [Control Enable](#), [Name](#), [Orientation](#), [Render Variable](#), [Set Variable Values](#)

## **Button Icon**

The icon used to identify each button in a data button bar control. The icons are obtained from a data item inserted into the control.

### **Properties of Button Icon**

[Data Item](#)

## **Button Label**

The label used to identify each button in a data button bar control. Each label is determined by the data items inserted into the control.

### **Properties of Button Label**

[Data Item](#)

## **Button Value**

Data items that are used to determine what appears as the button labels in a data button bar control.

### **Properties of Button Value**

[Data Item](#)

## **Calculated Measure**

A data item that is a calculated member.

### **Properties of Calculated Measure**

[Calculation Intersection](#), [Dimension](#), [Expression](#), [Label](#), [Name](#), [Solve Order](#), [Type](#)

## **Calculated Member**

An item, within a dimension, that represents an occurrence of schema data defined as a calculation of two or more members.

### **Properties of Calculated Member**

[Caption](#), [Data Item](#), [Name](#)

## **Calculated Member**

A data item that is a calculated member.

## Properties of Calculated Member

[Calculation Intersection](#), [Detail](#), [Dimension](#), [Expression](#), [Hierarchy](#), [Label](#), [Name](#), [Solve Order](#), [Type](#)

## Caption

The caption on a Field Set object.

## Properties of Caption

[Background Color](#), [Background Image](#), [Border](#), [Classes](#), [Conditional Styles](#), [Font](#), [Foreground Color](#), [Margin](#), [Padding](#), [Style Variable](#)

## Card

A card in a static deck control.

## Properties of Card

[Current Card](#)

## Card Label

The label used to identify a card in a deck control.

## Properties of Card Label

[Data Item](#)

## Card Value

A data item that is used to determine the data that appears in the cards of a data deck control.

## Properties of Card Value

[Data Item](#)

## Category axis

The axis line for an ordinal, or non-numeric, axis.

## Properties of Category axis

[Axis Labels](#), [Axis Line](#), [Axis Title](#), [Axis Title](#), [Drill-Through Definitions](#), [Gridlines](#), [Minor Gridlines](#), [Reverse Category order](#)

## Category Axis

The axis line for an ordinal, or non-numeric, axis. This object applies only to legacy charts.

## Properties of Category Axis

[Allow 45° Rotation](#), [Allow 90° Rotation](#), [Allow Skip](#), [Allow Stagger](#), [Axis Labels](#), [Axis Line](#), [Axis Title](#), [Axis Title](#), [Classes](#), [Conditional Styles](#), [Display Frequency](#), [Drill-Through Definitions](#), [First](#)

[Label Index](#), [Font](#), [Foreground Color](#), [Gridlines](#), [Horizontal Alignment](#), [Label Control](#), [Maximum Truncation Characters](#), [Minor Gridlines](#), [Style Variable](#), [Truncation](#), [Truncation Text](#), [Visible](#)

## Category Baseline

A baseline for the category in a chart.

### Properties of Category Baseline

[Aggregate Function](#), [Expression](#), [Hierarchy](#), [Hierarchy Unique Name](#), [Label](#), [Level](#), [Level Unique Name](#), [Line Style](#), [Master Detail Relationships](#), [Member](#), [Member Offset \(%\)](#), [Member Unique Name](#), [Name](#), [Properties](#), [Property Unique Name](#), [Query](#), [Render Variable](#), [Rollup Aggregate Function](#), [Root Members Only](#), [Set Sorting](#), [Type](#)

## Chart Body

Defines the body style of the chart itself. The style of the body can be set independently of the chart itself. This object applies only to legacy charts.

### Properties of Chart Body

[Background Color](#), [Background Image](#), [Conditional Styles](#), [Fill Effects](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Style Variable](#)

## Chart Body

Defines the body style of the chart itself. The style of the body can be set independently of the chart itself.

### Properties of Chart Body

[Background Color](#), [Background Effects](#), [Background Image](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#)

## Chart Footer

A footer for the chart. This object applies only to legacy charts.

### Properties of Chart Footer

[Background Effects](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Master Detail Relationships](#), [Properties](#), [Query](#), [Style Variable](#), [Visible](#)

## Chart Footer

A footer for the chart.

### Properties of Chart Footer

[Background Effects](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Master Detail Relationships](#), [Properties](#), [Query](#), [Render Variable](#)

## Chart Measure

The measure for the chart.

### Properties of Chart Measure

Aggregate Function, Conditional Styles, Custom Label, Data Format, Expression, Hierarchy, Hierarchy Unique Name, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Property Unique Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Type

## Chart Node Member

A data item, and its accompanying text, to render on the chart.

### Properties of Chart Node Member

Aggregate Function, Custom Label, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Expression, Hierarchy, Hierarchy Unique Name, HTML, HTML Source Variable, Label, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Properties, Property Unique Name, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Sorting, Source Type, Source Type, Source Type, Source Type, Text, Text Source Variable, Type, URL, URL Source Variable

## Chart Subtitle

The subtitle for a chart. This object applies only to legacy charts.

### Properties of Chart Subtitle

Background Effects, Box Type, Classes, Conditional Styles, Font, Foreground Color, Horizontal Alignment, Master Detail Relationships, Properties, Query, Style Variable, Visible

## Chart Subtitle

The subtitle for a chart.

### Properties of Chart Subtitle

Background Effects, Box Type, Classes, Conditional Styles, Font, Foreground Color, Horizontal Alignment, Master Detail Relationships, Properties, Query, Render Variable

## Chart Text Item

The data source and format for a text item, such as a legend item, legend title, axis label, or axis title. This object applies only to legacy charts.

### Properties of Chart Text Item

Aggregate Function, Conditional Styles, Data Format, Data Item Label, Data Item Value, Expression, Hierarchy, Hierarchy Unique Name, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Property Unique Name, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Source Type, Style Variable, Text, Text Source Variable, Type

## Chart Text Item

The data source and format for a text item, such as a legend item, legend title, axis label, or axis title.

### Properties of Chart Text Item

[Aggregate Function](#), [Conditional Styles](#), [Data Format](#), [Data Item Label](#), [Data Item Value](#), [Expression](#), [Hierarchy](#), [Hierarchy Unique Name](#), [Label](#), [Level](#), [Level Unique Name](#), [Member](#), [Member Unique Name](#), [Name](#), [Property Unique Name](#), [Render Variable](#), [Report Expression](#), [Rollup Aggregate Function](#), [Root Members Only](#), [Set Sorting](#), [Source Type](#), [Text](#), [Text Source Variable](#), [Type](#)

## Chart Title

The title text that appears at the top of the chart. This object applies only to legacy charts.

### Properties of Chart Title

[Background Effects](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Master Detail Relationships](#), [Properties](#), [Query](#), [Style Variable](#), [Visible](#)

## Chart Title

The title text that appears at the top of the chart.

### Properties of Chart Title

[Background Effects](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Master Detail Relationships](#), [Properties](#), [Query](#), [Render Variable](#)

## Check Box

A check box in a static check box group control.

### Properties of Check Box

[Icon](#), [Label](#)

## Check Box Group

A static check box group control. Each check box requires a corresponding row in the data table of the control.

### Properties of Check Box Group

[Check Boxes Definition](#), [Container Filter](#), [Container Select](#), [Control Enable](#), [Name](#), [Orientation](#), [Render Variable](#), [Set Variable Values](#), [Text Color](#)

## Check Box Icon

The icon used to identify each check box in a data check box group control. The icons are obtained from a data item inserted into the control.

## Properties of Check Box Icon

[Data Item](#)

## Check Box Label

The label used to identify a check box in a data check box group control. The label is determined by the data items inserted into the control.

## Properties of Check Box Label

[Data Item](#)

## Check Box Value

Data items that are used to determine what appears as the check box labels in a data check box group control.

## Properties of Check Box Value

[Data Item](#)

## Class

The HTML class name for a layout object. Use this attribute to indicate the type of styling to apply to the object when the report is rendered.

## Properties of Class

[Background Color](#), [Background Effects](#), [Background Image](#), [Border](#), [Box Type](#), [Description](#), [Floating](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Label](#), [Label](#), [Margin](#), [Padding](#), [Relative Alignment](#), [Size & Overflow](#), [Spacing & Breaking](#), [Table Properties](#), [Text Flow & Justification](#), [Vertical Alignment](#), [Visible](#), [White Space](#)

## Color by Value Measure

The measure or value used to determine the color of the points in a scatter or bubble chart. The points are colored according to the range of the data item instead of using the palette.

## Properties of Color by Value Measure

[Aggregate Function](#), [Color by Value](#), [Conditional Styles](#), [Custom Label](#), [Data Format](#), [Expression](#), [Hierarchy](#), [Hierarchy Unique Name](#), [Label](#), [Level](#), [Level Unique Name](#), [Member](#), [Member Unique Name](#), [Name](#), [Property Unique Name](#), [Rollup Aggregate Function](#), [Root Members Only](#), [Set Sorting](#), [Type](#)

## Colored Region

Defines a collection of colored regions, which are rectangles drawn in the chart body to highlight parts of the chart. The regions are drawn behind the data markers and in the same order that they appear in this element. The first region is drawn first, on the bottom, and the last region is drawn on top of the other regions.

### Properties of Colored Region

Aggregate Function, End Position, Expression, Hierarchy, Hierarchy Unique Name, Label, Level, Level Unique Name, Master Detail Relationships, Member, Member Unique Name, Name, Properties, Property Unique Name, Query, Render Variable, Rollup Aggregate Function, Root Members Only, Set Sorting, Start Position, Type

## Colored Region

Defines a collection of colored regions, which are rectangles drawn in the chart body to highlight parts of the chart. The regions are drawn behind the data markers and in the same order that they appear in this element. The first region is drawn first, on the bottom, and the last region is drawn on top of the other regions.

### Properties of Colored Region

Aggregate Function, Bottom Position, Expression, Hierarchy, Hierarchy Unique Name, Label, Left Position, Level, Level Unique Name, Master Detail Relationships, Member, Member Unique Name, Name, Properties, Property Unique Name, Query, Render Variable, Right Position, Rollup Aggregate Function, Root Members Only, Set Sorting, Top Position, Type

## Combination Chart

A chart that uses combinations of column charts, area charts, and line charts as data markers to plot multiple data series. This object applies only to legacy charts.

### Properties of Combination Chart

Alternate Text, Background Color, Background Effects, Background Image, Baselines, Border, Box Type, Category Axis, Chart Orientation, Classes, Clickable Regions, Conditional Palette, Conditional Styles, Container Select, Depth, Drill-Through Definitions, Floating, Font, Footer, Foreground Color, Legend, Margin, Markers, Marker Text Location, Master Detail Relationships, Maximum Hotspots, Name, No Data Contents, Notes, Padding, Pagination, Palette, Query, Relative Alignment, Render Page when Empty, Render Variable, Rotate Values, Series Color, Set Variable Values, Show Hover, Show Pointer Cursor, Size & Overflow, Style Variable, Subtitle, Suppression, Title, Title, Tooltips, Value Location, Visible, Visual Angle, Y1 Axis, Y2 Axis, Y2 Axis Position

## Combination Chart

A chart that uses combinations of column charts, area charts, and line charts as data markers to plot multiple data series.

### Properties of Combination Chart

Alternate Text, Background Effects, Border, Box Type, Category Baselines, Chart Orientation, Classes, Clickable Regions, Colored Regions, Combinations, Conditional Styles, Container Select, Depth, Drill-Through Definitions, Drop Shadow, Floating, Font, Footer, Foreground Color, Legend, Margin, Markers, Master Detail Relationships, Material Effects, Matrix rows and columns, Maximum Hotspots, Name, No Data Contents, Notes, Numeric Baselines, Padding, Pagination, Plot Area Fill, Query, Relative Alignment, Render Page when Empty, Render Variable, Series Color, Set

[Variable Values](#), [Show Hover](#), [Show Pointer Cursor](#), [Size & Overflow](#), [Subtitle](#), [Summarize Small Items](#), [Suppression](#), [Title](#), [Tooltips](#), [Trendlines](#), [Visible](#)

## Component Override

Overrides a child object of the Layout Component Reference object.

### Properties of Component Override

[Component Reference](#)

## Conditional Block

A block that can be used for conditional display.

### Properties of Conditional Block

[Background Color](#), [Background Effects](#), [Background Image](#), [Block Variable](#), [Border](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Current Block](#), [Floating](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Margin](#), [Padding](#), [Size & Overflow](#), [Spacing & Breaking](#), [Style Variable](#), [Text Flow & Justification](#), [Visible](#), [White Space](#)

## Conditional Block

Contains the default set of layout objects to render based on a report variable.

### Properties of Conditional Block

[Background Color](#), [Background Effects](#), [Background Image](#), [Block Variable](#), [Border](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Current Block](#), [Floating](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Margin](#), [Padding](#), [Size & Overflow](#), [Spacing & Breaking](#), [Style Variable](#), [Text Flow & Justification](#), [Visible](#), [White Space](#)

## Context Item Text

The text associated with a context area item.

### Properties of Context Item Text

[Background Color](#), [Background Image](#), [Border](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Floating](#), [Font](#), [Foreground Color](#), [Margin](#), [Padding](#), [Relative Alignment](#), [Render Variable](#), [Separator](#), [Size & Overflow](#), [Spacing & Breaking](#), [Style Variable](#), [Text Flow & Justification](#), [Visible](#)

## Crosstab

A layout object used to render the results of a query that aggregates data, and then arranges it in a two-dimensional grid.

### Properties of Crosstab

[Background Color](#), [Background Image](#), [Border](#), [Box Type](#), [Classes](#), [Clickable Regions](#), [Conditional Styles](#), [Container Filter](#), [Container Select](#), [Default Measure](#), [Drop Shadow](#), [Fact Cells Precedence](#), [Floating](#), [Font](#), [Foreground Color](#), [Has Fact Cells](#), [Horizontal Alignment](#), [Margin](#), [Master Detail](#)

[Relationships](#), [Name](#), [No Data Contents](#), [Pagination](#), [Query](#), [Relative Alignment](#), [Render Page when Empty](#), [Render Variable](#), [Rows Per Page](#), [Set Variable Values](#), [Show Hover](#), [Show Pointer Cursor](#), [Size & Overflow](#), [Style Variable](#), [Summary Text](#), [Suppression](#), [Table Properties](#), [Text Flow & Justification](#), [Visible](#)

## Crosstab Columns

Overrides the style for Crosstab Column Member objects that is defined in the GlobalReportStyles.css file.

### Properties of Crosstab Columns

[Background Color](#), [Background Effects](#), [Background Image](#), [Border](#), [Box Type](#), [Conditional Styles](#), [Data Format](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Padding](#), [Size & Overflow](#), [Spacing & Breaking](#), [Style Variable](#), [Text Flow & Justification](#), [Vertical Alignment](#), [White Space](#)

## Crosstab Columns

A list of columns in a crosstab.

### Properties of Crosstab Columns

[Background Color](#), [Background Effects](#), [Background Image](#), [Border](#), [Box Type](#), [Conditional Styles](#), [Data Format](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Padding](#), [Size & Overflow](#), [Spacing & Breaking](#), [Style Variable](#), [Text Flow & Justification](#), [Vertical Alignment](#), [White Space](#)

## Crosstab Corner

The top-left corner of a crosstab, on top of the row labels and to the left of the column labels. It is generally used to represent crosstab members.

### Properties of Crosstab Corner

[Aggregate Function](#), [Background Color](#), [Background Effects](#), [Background Image](#), [Border](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Data Format](#), [Data Item Label](#), [Data Item Label](#), [Data Item Label](#), [Data Item Label](#), [Data Item Value](#), [Expression](#), [Font](#), [Foreground Color](#), [Hierarchy](#), [Hierarchy Unique Name](#), [Horizontal Alignment](#), [HTML](#), [HTML Source Variable](#), [Label](#), [Label](#), [Level](#), [Level Unique Name](#), [Member](#), [Member Unique Name](#), [Name](#), [Padding](#), [Property Unique Name](#), [Report Expression](#), [Report Expression](#), [Report Expression](#), [Report Expression](#), [Rollup Aggregate Function](#), [Root Members Only](#), [Set Sorting](#), [Size & Overflow](#), [Source Type](#), [Source Type](#), [Source Type](#), [Source Type](#), [Spacing & Breaking](#), [Style Variable](#), [Text](#), [Text Flow & Justification](#), [Text Source Variable](#), [Type](#), [URL](#), [URL Source Variable](#), [Vertical Alignment](#), [White Space](#)

## Crosstab Fact Cells

The contents of the fact cells of the crosstab. There is only one fact cell definition for the crosstab, regardless of the number of measures.

## Properties of Crosstab Fact Cells

Aggregate Function, Apply Single Class, Background Color, Background Effects, Background Image, Border, Box Type, Classes, Clickable Region, Conditional Styles, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Drill-Through Definitions, Expression, Font, Foreground Color, Hierarchy, Hierarchy Unique Name, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Padding, Property Unique Name, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Size & Overflow, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, Type, URL, URL Source Variable, Vertical Alignment, White Space

## Crosstab Intersection

An intersection in a crosstab. The cell contents of a specific intersection can be overridden and the style defined.

## Properties of Crosstab Intersection

Aggregate Function, Background Color, Background Effects, Background Image, Border, Box Type, Classes, Clickable Region, Column Coordinate, Conditional Styles, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Define Contents, Drill-Through Definitions, Expression, Font, Foreground Color, Hierarchy, Hierarchy Unique Name, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Padding, Property Unique Name, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Root Members Only, Row Coordinate, Set Sorting, Size & Overflow, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, Type, URL, URL Source Variable, Vertical Alignment, White Space

## Crosstab Member Fact Cells

The contents of the fact cells of a crosstab node member.

## Properties of Crosstab Member Fact Cells

Aggregate Function, Background Color, Background Effects, Background Image, Border, Box Type, Classes, Clickable Region, Conditional Styles, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Define Contents, Drill-Through Definitions, Expression, Font, Foreground Color, Hierarchy, Hierarchy Unique Name, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Padding, Property Unique Name, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Size & Overflow, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, Type, URL, URL Source Variable, Vertical Alignment, White Space

## Crosstab Node Member

A member in the crosstab node.

### Properties of Crosstab Node Member

Aggregate Function, Background Color, Background Effects, Background Image, Border, Box Type, Classes, Clickable Region, Conditional Styles, Data Format, Data Item, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Drill-Through Definitions, Expression, Font, Foreground Color, Hierarchy, Hierarchy Unique Name, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Level, Level Indentation, Level Unique Name, Member, Member Unique Name, Name, Node Coordinate, Padding, Pagination, Properties, Property Unique Name, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Size & Overflow, Sorting, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, Type, URL, URL Source Variable, Vertical Alignment, Visible, White Space

## Crosstab Rows

Overrides the style for Crosstab Row Member objects that is defined in the GlobalReportStyles.css file.

### Properties of Crosstab Rows

Background Color, Background Effects, Background Image, Border, Box Type, Conditional Styles, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

## Crosstab Rows

A list of rows in a crosstab.

### Properties of Crosstab Rows

Background Color, Background Effects, Background Image, Border, Box Type, Conditional Styles, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

## Crosstab Space

Inserts an empty cell on a crosstab edge. Allows for the insertion of non-data cells on an edge.

### Properties of Crosstab Space

Background Color, Background Effects, Background Image, Border, Box Type, Classes, Clickable Region, Conditional Styles, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Drill-Through Definitions, Font, Foreground Color, Horizontal Alignment, HTML, HTML Source Variable, Label, Node Coordinate, Padding, Pagination, Render Fact Cells, Report Expression, Report Expression, Report Expression, Report Expression, Size & Overflow, Source Type, Source Type,

[Source Type](#), [Source Type](#), [Spacing & Breaking](#), [Style Variable](#), [Text](#), [Text Flow & Justification](#), [Text Source Variable](#), [URL](#), [URL Source Variable](#), [Vertical Alignment](#), [Visible](#), [White Space](#)

## Crosstab Summary

The crosstab summary that appears in executed active reports.

### Properties of Crosstab Summary

[Background Color](#), [Background Effects](#), [Background Image](#), [Border](#), [Box Type](#), [Classes](#), [Clickable Region](#), [Conditional Styles](#), [Data Format](#), [Data Item](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Node Coordinate](#), [Output Aggregation Method](#), [Padding](#), [Size & Overflow](#), [Solve Order](#), [Spacing & Breaking](#), [Style Variable](#), [Text Flow & Justification](#), [Vertical Alignment](#), [Visible](#), [White Space](#)

## Cumulation Label

A label that is rendered for the cumulation line in a Pareto chart.

### Properties of Cumulation Label

[Conditional Styles](#), [Data Format](#), [Properties](#), [Query](#)

## Cumulation Line

The cumulation line in a Pareto chart.

### Properties of Cumulation Line

[Axis Labels](#), [Axis Line](#), [Axis Range](#), [Axis Title](#), [Axis Title](#), [Gridlines](#), [Minor Gridlines](#)

## Cumulation Line

A line that shows the cumulative effect of multiple series members on a measure in a Pareto chart. This object applies only to legacy charts.

### Properties of Cumulation Line

[Cumulation Axis](#), [Cumulation Label](#), [Line Styles](#), [Marker Shape](#), [Marker Size \(pt\)](#), [Properties](#), [Value Location](#), [Values](#)

## Cumulation Line

A line that shows the cumulative effect of multiple series members on a measure in a Pareto chart.

### Properties of Cumulation Line

[Cumulation Label](#), [Data Points](#), [Line Style](#), [Show Values](#)

## Cumulation Line Axis

The axis for the cumulation line in a Pareto chart.

### Properties of Cumulation Line Axis

Axis Labels, Axis Line, Axis Title, Classes, Conditional Styles, Data Format, Font, Foreground Color, Gridlines, Minor Gridlines, Scale Interval, Style Variable, Visible

## Cumulation Line Label

A label that is rendered with the cumulation line in a Pareto chart. This object applies only to legacy charts.

### Properties of Cumulation Line Label

Aggregate Function, Data Item Label, Data Item Value, Expression, Hierarchy, Hierarchy Unique Name, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Property Unique Name, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Source Type, Text, Text Source Variable, Type

## Data Button Bar

A data-driven button bar control. Each button label is determined by the data items inserted into the control.

### Properties of Data Button Bar

Button Gap, Button Width, Container Select, Control Enable, Master Detail Relationships, Name, Orientation, Query, Render Variable, Set Variable Values, Sorting, Suppression

## Data Check Box Group

A data-driven check box group control. Each check box label is determined by the data items inserted into the control.

### Properties of Data Check Box Group

Container Filter, Container Select, Control Enable, Master Detail Relationships, Name, Orientation, Query, Render Variable, Set Variable Values, Sorting, Suppression, Text Color

## Data Deck

A data-driven deck control. Each card in the deck is determined by the data items inserted into the control.

### Properties of Data Deck

Container Select, Default Card, Master Detail Relationships, Name, Query, Render Variable, Size, Slide Animation Direction, Sorting, Suppression

## Data Drop-Down List

A data-driven drop-down list control. The items that appear in the list are determined by the data items inserted into the control.

## Properties of Data Drop-Down List

Container Filter, Container Select, Control Enable, Master Detail Relationships, Name, No Value List Item, Query, Render Variable, Set Variable Values, Show Icon, Size, Sorting, Suppression

## Data Item

A set of data values or members.

## Properties of Data Item

Aggregate Function, Calculation Intersection, Detail, Expression, Label, Name, Pre-Sort, Rollup Aggregate Function, Solve Order, Type

## Data List Box

A data-driven list box control. The items that appear in the list are determined by the data items inserted into the control.

## Properties of Data List Box

Container Filter, Container Select, Control Enable, Master Detail Relationships, Multi-Select, Name, No Value List Item, Query, Render Variable, Set Variable Values, Size, Sorting, Suppression

## Data Radio Button Group

A data-driven radio button group control. Each radio button label is determined by the data items inserted into the control.

## Properties of Data Radio Button Group

Container Filter, Container Select, Control Enable, Master Detail Relationships, Name, Orientation, Query, Render Variable, Set Variable Values, Sorting, Suppression, Text Color

## Data Tab Control

A data-driven tab control. The label of each tab is determined by the data items inserted into the control.

## Properties of Data Tab Control

Container Select, Master Detail Relationships, Name, Query, Render Variable, Set Variable Values, Size, Sorting, Suppression, Tab Orientation, Tab Width

## Data Toggle Button Bar

A data-driven toggle button bar control. Each toggle button label is determined by the data items inserted into the control.

## Properties of Data Toggle Button Bar

Button Gap, Button Width, Container Select, Control Enable, Master Detail Relationships, Name, Orientation, Query, Render Variable, Set Variable Values, Sorting, Suppression

## Date

The date when the report runs.

### Properties of Date

Background Color, Classes, Data Format, Font, Foreground Color, Margin, Relative Alignment

## Date & Time Prompt

A prompt control with which you can select a date and time value.

### Properties of Date & Time Prompt

Box Type, Calendar Type, Choices Deselect All Text, Choices Select All Text, Choices Text, Clock Mode, Conditional Styles, Default Selections, Deselect Text, Display Milliseconds, Display Seconds, First Date, Floating, From Text, Hide Adornments, Highest Value Text, Insert Text, Last Date, Lowest Value Text, Multi-Select, Name, Parameter, Range, Remove Text, Render Variable, Required, Select UI, Style Variable, To Text, Visible

## Date Prompt

A prompt control with which you can select a date value.

### Properties of Date Prompt

Box Type, Calendar Type, Choices Deselect All Text, Choices Select All Text, Choices Text, Conditional Styles, Default Selections, Deselect Text, First Date, Floating, From Text, Hide Adornments, Highest Value Text, Insert Text, Last Date, Lowest Value Text, Multi-Select, Name, Parameter, Range, Relative Alignment, Remove Text, Render Variable, Required, Select UI, Style Variable, To Text, Visible

## Deck

A static deck control. Each card in the deck requires a corresponding row in the data table of the control.

### Properties of Deck

Container Select, Current Card, Deck Cards Definition, Default Card, Name, Render Variable, Size, Slide Animation Direction

## Default Measure

A default measure for the chart. If the chart measure cannot be determined by the data series rendered on the chart edges, the default measure is used. This object applies only to legacy charts.

### Properties of Default Measure

Aggregate Function, Conditional Styles, Custom Label, Data Format, Expression, Hierarchy, Hierarchy Unique Name, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Property Unique Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Style Variable, Type

## Default Measure

A default measure for the chart. If the chart measure cannot be determined by the data series rendered on the chart edges, the default measure is used.

### Properties of Default Measure

[Aggregate Function](#), [Conditional Styles](#), [Custom Label](#), [Data Format](#), [Expression](#), [Hierarchy](#), [Hierarchy Unique Name](#), [Label](#), [Level](#), [Level Unique Name](#), [Member](#), [Member Unique Name](#), [Name](#), [Property Unique Name](#), [Rollup Aggregate Function](#), [Root Members Only](#), [Set Sorting](#), [Type](#)

## Detail Filter

A set of conditions in a query that narrow the scope of the data returned. A detail filter is applied before aggregation is complete.

### Properties of Detail Filter

[Application](#), [Definition](#), [Usage](#)

## Dimension

A grouping of descriptive information about an aspect of a business. Dimensions contain levels, whose order defines the hierarchy of organizational structures and data. Dimensions and levels are values by which measures can be viewed, filtered, or aggregated.

### Properties of Dimension

[Name](#)

## Dimensional Edge Summary

A summary that appears on the edge of a crosstab.

### Properties of Dimensional Edge Summary

[Aggregate Data Item](#), [Aggregation Method](#), [Calculation Intersection](#), [Dimension](#), [Hierarchy](#), [Label](#), [Name](#), [Solve Order](#), [Type](#), [Use Set Aggregation](#)

## Display Layer

A map layer that is there for appearance only. Display layers do not correspond to data series or measures.

### Properties of Display Layer

[Border Color](#), [Borders](#), [Conditional Styles](#), [Fill Effects](#), [Labels](#), [Style Variable](#)

## Drop-Down List

A static drop-down list control. Each item in the list requires a corresponding row in the data table of the control.

### Properties of Drop-Down List

Container Filter, Container Select, Control Enable, List Items Definition, Name, No Value List Item, Render Variable, Set Variable Values, Show Icon, Size

## Edge Summary

A summary that appears on the edge of a crosstab.

### Properties of Edge Summary

Aggregate Data Item, Aggregation Method, Calculation Intersection, Label, Name, Solve Order, Type, Use Set Aggregation

## Equation Label

An equation label for a trendline displayed on a chart.

### Properties of Equation Label

Background Effects, Conditional Styles, Font, Foreground Color, Horizontal Alignment, Position

## Explicit Member Set

A set of data items that define an explicit set of members.

### Properties of Explicit Member Set

Detail, Dimension, Hierarchy, Label, Members, Name, Set Sorting, Type

## Fact

The central values that are aggregated and analyzed. Also known as measures, they are special business measurement values, such as sales or inventory levels.

### Properties of Fact

Data Item, Name

## Field Set

A container with a caption, into which you can insert other objects. It is similar to a block object, except that it also has a caption.

### Properties of Field Set

Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Floating, Font, Foreground Color, Horizontal Alignment, Margin, Name, Relative Alignment, Render Variable, Show Caption, Size & Overflow, Style Variable, Visible

## Gauge Axis

The axis on the face of the gauge that contains gauge indicators, such as data ranges, color ranges, and interval markers.

## Properties of Gauge Axis

Aggregate Function, Axis Angles, Axis Justification, Axis Labels, Axis Line, Axis Range, Axis Title, Conditional Palette, Expression, Gauge Axis Colors, Gauge Axis Inner Radius, Gauge Axis Outer Radius, Gauge Needle, Gridlines, Hierarchy, Hierarchy Unique Name, Label, Level, Level Unique Name, Member, Member Unique Name, Minor Gridlines, Name, Palette, Property Unique Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Text Orientation, Type

## Gauge Chart

A chart that plots a data series against a measure using a dial or gauge for the measure, and needles or indicators for the series members.

### Properties of Gauge Chart

Alternate Text, Background Effects, Border, Box Type, Classes, Clickable Regions, Conditional Styles, Container Select, Dial Face Fill, Drill-Through Definitions, Drop Shadow, Floating, Font, Footer, Foreground Color, Gauge Axes, Gauge Border, Gauge Pivot, Legend, Margin, Master Detail Relationships, Matrix rows and columns, Maximum Hotspots, Name, No Data Contents, Notes, Padding, Pagination, Query, Relative Alignment, Render Page when Empty, Render Variable, Set Variable Values, Show Hover, Show Pointer Cursor, Size & Overflow, Subtitle, Suppression, Title, Tooltips, Visible

## Gauge Chart

A chart that plots a data series against a measure using a dial or gauge for the measure, and needles or indicators for the series members. This object applies only to legacy charts.

### Properties of Gauge Chart

Alternate Text, Axis Title, Background Color, Background Effects, Background Image, Border, Border Color, Borders, Box Type, Classes, Clickable Regions, Conditional Palette, Conditional Styles, Container Select, Dial Outline Color, Drill-Through Definitions, Face Color, Floating, Font, Footer, Foreground Color, Gauge Labels, Gauge Palette, Legend, Margin, Master Detail Relationships, Maximum Hotspots, Name, No Data Contents, Notes, Numerical Axis, Padding, Pagination, Palette, Query, Relative Alignment, Render Page when Empty, Render Variable, Set Variable Values, Show Hover, Show Pointer Cursor, Size & Overflow, Style Variable, Subtitle, Suppression, Title, Tooltips, Visible

## Gauge Labels

A label for each gauge in a multiple gauge chart.

### Properties of Gauge Labels

Classes, Conditional Styles, Drill-Through Definitions, Font, Foreground Color, Horizontal Alignment, Maximum Truncation Characters, Style Variable, Truncation, Truncation Text, Visible

## Gauge Numerical Axis

The numeric axis for the gauge chart, including labels, titles, range, and scale.

## Properties of Gauge Numerical Axis

Conditional Styles, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

## Generated Prompt

A control that acts as a placeholder. The report server will replace this control with an appropriate generated prompt control, as if it was on a generated prompt page.

## Properties of Generated Prompt

Hide Adornments, Name, Parameter, Render Variable, Required

## Hierarchy Set

A set of data items that define the members of a hierarchy.

## Properties of Hierarchy Set

Detail, Dimension, Hierarchy, Hierarchy Unique Name, Label, Name, Root Members Only, Set Sorting, Type

## HTML Item

A container into which you can add HTML, such as a link to a multimedia file. HTML items will only appear when you run the report in HTML format.

When you upgrade to the next version of IBM Cognos BI, the report upgrade processes do not account for the use of undocumented and unsupported mechanisms or features such as JavaScript that refers to IBM Cognos HTML items.

## Properties of HTML Item

Aggregate Function, Data Item Label, Data Item Value, Description, Expression, Hierarchy, Hierarchy Unique Name, HTML, HTML Source Variable, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Name, Property Unique Name, Render Variable, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Source Type, Type

## Hyperlink

A hyperlink that can be defined as a static value, a query item, or as the result of a report expression. If a report expression is used, then the other values are ignored.

## Properties of Hyperlink

Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Floating, Font, Foreground Color, Margin, Name, Padding, Relative Alignment, Render Variable, Report Expression, Report Expression, Size & Overflow, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, URL, URL Source Variable, Visible

## Hyperlink Button

A hyperlink that is formatted as a button. The hyperlink can be defined as a static value, a query item, or as the result of a report expression. If a report expression is used, then the other values are ignored.

### Properties of Hyperlink Button

[Background Color](#), [Background Image](#), [Border](#), [Classes](#), [Conditional Styles](#), [Data Item Label](#), [Data Item Label](#), [Data Item Value](#), [Data Item Value](#), [Floating](#), [Foreground Color](#), [Margin](#), [Name](#), [Padding](#), [Relative Alignment](#), [Render Variable](#), [Report Expression](#), [Report Expression](#), [Size & Overflow](#), [Source Type](#), [Source Type](#), [Style Variable](#), [Text](#), [Text Source Variable](#), [URL](#), [URL Source Variable](#)

## Image

A link to an image file. The link can be a static value, or it can come from a report expression or query item. Use the URL source properties of the image object to define the link.

### Properties of Image

[Aggregate Function](#), [Alternate Text](#), [Background Color](#), [Background Image](#), [Border](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Data Item Label](#), [Data Item Value](#), [Drill-Through Definitions](#), [Expression](#), [Floating](#), [Hierarchy](#), [Hierarchy Unique Name](#), [Label](#), [Level](#), [Level Unique Name](#), [Margin](#), [Member](#), [Member Unique Name](#), [Name](#), [Name](#), [Property Unique Name](#), [Relative Alignment](#), [Render Variable](#), [Report Expression](#), [Rollup Aggregate Function](#), [Root Members Only](#), [Set Sorting](#), [Size & Overflow](#), [Source Type](#), [Style Variable](#), [Type](#), [URL](#), [URL Source Variable](#), [Visible](#)

## Intersection (Tuple)

A data item obtained from the combination of two or more members that you specify. An intersection appears as a single, unified member instead of its component members. Each member must be from a different dimension.

### Properties of Intersection (Tuple)

[Dimension](#), [Hierarchy](#), [Label](#), [Members](#), [Name](#), [Type](#)

## Interval Prompt

An advanced prompt control that allows you to enter time duration values.

### Properties of Interval Prompt

[Box Type](#), [Choices Deselect All Text](#), [Choices Select All Text](#), [Choices Text](#), [Conditional Styles](#), [Days Text](#), [Default Selections](#), [Deselect Text](#), [Display Milliseconds](#), [Display Seconds](#), [Floating](#), [From Text](#), [Hide Adornments](#), [Highest Value Text](#), [Hours Text](#), [Insert Text](#), [Lowest Value Text](#), [Milliseconds Text](#), [Minutes Text](#), [Multi-Select](#), [Name](#), [Parameter](#), [Range](#), [Remove Text](#), [Render Variable](#), [Required](#), [Seconds Text](#), [Style Variable](#), [To Text](#), [Visible](#)

## Join

A relationship between a field in one table or query and a field of the same data type in another table or query.

### Properties of Join

[Join Relationships](#)

## Key

An object that uniquely identifies members of a level.

If the unique identifier is a primary or alternate key, you need only one key object. If the unique identifier is a composite key, you need one key object for every data item that participates in making the members of a level unique.

### Properties of Key

[Data Item](#), [Name](#)

## Label

Specifies the text, or label, for the object in a chart.

### Properties of Label

[Conditional Styles](#), [Data Format](#), [Master Detail Relationships](#), [Properties](#), [Query](#), [Render Variable](#)

## Layout Component Reference

A reference to another layout object. Before you can reference an object, its ID property must be set.

### Properties of Layout Component Reference

[Component Reference](#), [Embed](#), [Overrides](#)

## Legend

A key to the patterns or colors assigned to the data series in a chart. This object applies only to legacy charts.

### Properties of Legend

[Absolute Position](#), [Auto Truncation](#), [Background Color](#), [Background Image](#), [Border Color](#), [Borders](#), [Bottom Position \(px\)](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Drill-Through Definitions](#), [Fill Effects](#), [Font](#), [Font Auto-Sizing](#), [Foreground Color](#), [Horizontal Alignment](#), [Left Position \(px\)](#), [Legend Title](#), [Legend Title](#), [Maximum Characters](#), [Position](#), [Right Position \(px\)](#), [Separator](#), [Show Legend Values](#), [Style Variable](#), [Top Position \(px\)](#), [Truncation Text](#), [Visible](#)

## Legend

A key to the patterns or colors assigned to the data series in a chart.

## Properties of Legend

[Auto Font Size](#), [Background Effects](#), [Classes](#), [Conditional Styles](#), [Drill-Through Definitions](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Legend Separator](#), [Legend Title](#), [Render Variable](#), [Show Values](#), [Text Truncation](#)

## Legend Title

The title for the legend, including the title text and text style. If this object is empty, a default title is rendered, if available. This object applies only to legacy charts.

### Properties of Legend Title

[Box Type](#), [Classes](#), [Conditional Styles](#), [Default Title](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Master Detail Relationships](#), [Properties](#), [Query](#), [Style Variable](#), [Visible](#)

## Legend Title

The title for the legend, including the title text and text style. If this object is empty, a default title is rendered, if available.

### Properties of Legend Title

[Auto Font Size](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Default Title](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Master Detail Relationships](#), [Properties](#), [Query](#), [Render Variable](#)

## Level

A set of members with a predefined set of similar characteristics. For example, the members Year 1999 and Year 2000 in the Time dimension form a year level, while the corresponding quarters form a quarter level.

### Properties of Level

[Caption](#), [Name](#), [Sorting](#)

## Level Hierarchy

Specifies how the levels in a dimension are logically ordered.

### Properties of Level Hierarchy

[Name](#)

## Level Set

A set of data items that define the members of a level.

### Properties of Level Set

[Detail](#), [Dimension](#), [Hierarchy](#), [Label](#), [Level](#), [Level Unique Name](#), [Name](#), [Set Sorting](#), [Type](#)

## Line

The data marker used to represent a data series in a line chart.

### Properties of Line

[Aggregate Function](#), [Chart Type](#), [Conditional Palette](#), [Expression](#), [Hierarchy](#), [Hierarchy Unique Name](#), [Label](#), [Level](#), [Level Unique Name](#), [Line and Markers](#), [Line Shape](#), [Member](#), [Member Unique Name](#), [Name](#), [Palette](#), [Property Unique Name](#), [Rollup Aggregate Function](#), [Root Members Only](#), [Series Type](#), [Set Sorting](#), [Show Values](#), [Type](#), [Value Markers](#)

## Line

The data marker used to represent a data series in a line chart. This object applies only to legacy charts.

### Properties of Line

[Axis Assignment](#), [Border Color](#), [Borders](#), [Chart Type](#), [Grouping Type](#), [Line Style](#), [Line Type](#), [Line Weight \(pt\)](#), [Show Data Points](#), [Show line](#), [Value Location](#), [Values](#), [Value Type](#)

## List

A layout object that is used to present query results in a list fashion.

### Properties of List

[Background Color](#), [Background Image](#), [Border](#), [Box Type](#), [Classes](#), [Clickable Regions](#), [Column Titles](#), [Conditional Styles](#), [Container Filter](#), [Container Select](#), [Contents Height](#), [Drop Shadow](#), [Floating](#), [Font](#), [Foreground Color](#), [Grouping & Sorting](#), [Horizontal Alignment](#), [Margin](#), [Master Detail Relationships](#), [Name](#), [No Data Contents](#), [Pagination](#), [Properties](#), [Query](#), [Relative Alignment](#), [Render Page when Empty](#), [Render Variable](#), [Rows Per Page](#), [Set Variable Values](#), [Share Result Set](#), [Show Hover](#), [Show Pointer Cursor](#), [Size & Overflow](#), [Style Variable](#), [Summary Text](#), [Suppression](#), [Table Properties](#), [Text Flow & Justification](#), [Visible](#)

## List Box

A static list box control. Each item in the list requires a corresponding row in the data table of the control.

### Properties of List Box

[Container Filter](#), [Container Select](#), [Control Enable](#), [List Items Definition](#), [Multi-Select](#), [Name](#), [No Value List Item](#), [Render Variable](#), [Set Variable Values](#), [Size](#)

## List Cell

A cell in a row, you can use for a list header or footer.

### Properties of List Cell

[Aggregate Function](#), [Background Color](#), [Background Effects](#), [Background Image](#), [Border](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Data Format](#), [Data Item Label](#), [Data Item Label](#), [Data Item Label](#), [Data](#)

Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Expression, Font, Foreground Color, Hierarchy, Hierarchy Unique Name, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Output Aggregation Method, Padding, Property Unique Name, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Size & Overflow, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, Type, URL, URL Source Variable, Vertical Alignment, White Space

## List Column

A column in a list. Generally, the column will contain a query item, but it may also contain any number of layout objects. For example, the To column in a statement list may present the address in a table format. If the column only contains a query items, then the column will automatically span the group if the query item is grouped.

### Properties of List Column

Allow Sorting, Background Color, Background Effects, Background Image, Border, Box Type, Clickable Region, Column Visibility, Conditional Styles, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Pagination, Render, Render Variable, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

## List Column Body

The contents of a column in a list report.

### Properties of List Column Body

Aggregate Function, Allow Sorting, Background Color, Background Effects, Background Image, Border, Box Type, Classes, Clickable Region, Column Visibility, Conditional Styles, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Data Item Value, Drill-Through Definitions, Expression, Font, Foreground Color, Group Span, Hierarchy, Hierarchy Unique Name, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Padding, Property Unique Name, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Size & Overflow, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, Type, URL, URL Source Variable, Vertical Alignment, White Space

## List Columns

A set of columns in a list.

### Properties of List Columns

Background Color, Background Effects, Background Image, Border, Box Type, Conditional Styles, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

## List Columns

Overrides the style for List Column objects that is defined in the GlobalReportStyles.css file.

### Properties of List Columns

Background Color, Background Effects, Background Image, Border, Box Type, Conditional Styles, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

## List Columns Body Style

Overrides the style for List Column Body objects that is defined in the GlobalReportStyles.css file.

### Properties of List Columns Body Style

Background Color, Background Effects, Background Image, Border, Box Type, Conditional Styles, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

## List Columns Title Style

Overrides the style for List Column Title objects that is defined in the GlobalReportStyles.css file.

### Properties of List Columns Title Style

Background Color, Background Effects, Background Image, Border, Box Type, Conditional Styles, Data Format, Font, Foreground Color, Horizontal Alignment, Padding, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

## List Column Title

The title of a list column.

### Properties of List Column Title

Aggregate Function, Allow Sorting, Background Color, Background Effects, Background Image, Border, Box Type, Classes, Clickable Region, Column Visibility, Conditional Styles, Data Format, Data Item Label, Data Item Label, Data Item Label, Data Item Label, Data Item Value, Data Item Value, Data Item Value, Drill-Through Definitions, Expression, Font, Foreground Color, Hierarchy, Hierarchy Unique Name, Horizontal Alignment, HTML, HTML Source Variable, Label, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Padding, Property Unique Name, Report Expression, Report Expression, Report Expression, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Size & Overflow, Source Type, Source Type, Source Type, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, Type, URL, URL Source Variable, Vertical Alignment, White Space

## List Footer

The footer that appears at the end of a list on each page on which the list is rendered. It is useful for presenting page totals.

## Properties of List Footer

[Background Color](#), [Background Effects](#), [Background Image](#), [Border](#), [Box Type](#), [Conditional Styles](#), [Data Format](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Padding](#), [Properties](#), [Push To Bottom](#), [Size & Overflow](#), [Spacing & Breaking](#), [Style Variable](#), [Text Flow & Justification](#), [Vertical Alignment](#), [White Space](#)

## List Header

The header that appears at the start of a list on each page that the list is rendered. It is useful for presenting carry forward totals.

## Properties of List Header

[Background Color](#), [Background Effects](#), [Background Image](#), [Border](#), [Box Type](#), [Conditional Styles](#), [Data Format](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Padding](#), [Pagination](#), [Properties](#), [Size & Overflow](#), [Spacing & Breaking](#), [Style Variable](#), [Text Flow & Justification](#), [Vertical Alignment](#), [White Space](#)

## List Item

A list item in a static list box control.

## Properties of List Item

[Icon](#), [Label](#)

## List Item Icon

The icon used to identify each item in a data list box control. The icons are obtained from a data item inserted into the control.

## Properties of List Item Icon

[Data Item](#)

## List Item Label

The label used to identify each item in a data list box control. Each label is determined by the data items inserted into the control.

## Properties of List Item Label

[Data Item](#)

## List Item Value

Data items that are used to determine what appears as the item labels in a data list box control.

## Properties of List Item Value

[Data Item](#)

## List Page Footer

The footer that appears at the bottom of every page of a list report. This object is rendered after the list details and other list footers.

### Properties of List Page Footer

[Push To Bottom](#)

## List Page Header

The header in the list that will appear on every page rendered. It occurs after the column titles and before the overall group header of the list.

### Properties of List Page Header

[Display After Overall Header](#)

## List Row

A row in a list.

### Properties of List Row

[Background Color](#), [Background Effects](#), [Background Image](#), [Border](#), [Box Type](#), [Clickable Region](#), [Conditional Styles](#), [Data Format](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Padding](#), [Size & Overflow](#), [Spacing & Breaking](#), [Style Variable](#), [Text Flow & Justification](#), [Vertical Alignment](#), [White Space](#)

## List Row Cells Style

Overrides the style for Row Cells Style objects that is defined in the GlobalReportStyles.css file.

### Properties of List Row Cells Style

[Background Color](#), [Background Effects](#), [Background Image](#), [Border](#), [Box Type](#), [Conditional Styles](#), [Data Format](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Padding](#), [Size & Overflow](#), [Spacing & Breaking](#), [Style Variable](#), [Text Flow & Justification](#), [Vertical Alignment](#), [White Space](#)

## List Summary

A summary that is automatically added as the footer row in a list.

### Properties of List Summary

[Aggregate Data Item](#), [Aggregation Method](#), [Label](#), [Name](#), [Type](#)

## List Summary

The list summary that appears in executed active reports.

## Properties of List Summary

Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Data Format, Data Item, Floating, Font, Foreground Color, Margin, Output Aggregation Method, Padding, Relative Alignment, Render Variable, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Visible

# Map

A chart that uses a map to show data.

## Properties of Map

Alternate Text, Axis Title, Background Color, Background Effects, Background Image, Border, Box Type, Classes, Clickable Regions, Conditional Styles, Container Select, Data Language, Dictionary, Drill-Through Definitions, Expand Features, Floating, Font, Footer, Foreground Color, Ignore Data with No Features, Legend, Map & Layers, Margin, Master Detail Relationships, Maximum Hotspots, Name, No Data Contents, No Data Features Size (pt), Notes, Padding, Pagination, Query, Relative Alignment, Render Page when Empty, Render Variable, Set Variable Values, Show Hover, Show Pointer Cursor, Size & Overflow, Style Variable, Subtitle, Suppression, Title, Title, Tooltips, Visible

# Map Location

Associates a data series with regions on the region layer of the map.

## Properties of Map Location

Aggregate Function, Expression, Hierarchy, Hierarchy Unique Name, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Property Unique Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Type

# Map Location

Associates a data series with points on the point layer of the map.

## Properties of Map Location

Aggregate Function, Expression, Hierarchy, Hierarchy Unique Name, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Property Unique Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Type

# Map Refinement Location

Qualifies the data series members that are associated with regions on the map.

## Properties of Map Refinement Location

Aggregate Function, Expression, Hierarchy, Hierarchy Unique Name, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Property Unique Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Type

## Map Refinement Location

Qualifies the data series members that are associated with points on the map.

### Properties of Map Refinement Location

Aggregate Function, Expression, Hierarchy, Hierarchy Unique Name, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Property Unique Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Type

## Marimekko Chart

A type of bar chart that can show three levels of data.

### Properties of Marimekko Chart

Alternate Text, Background Color, Background Effects, Background Image, Baselines, Border, Border Color, Borders, Box Type, Category Axis, Classes, Clickable Regions, Conditional Palette, Conditional Styles, Container Select, Depth, Drill-Through Definitions, Floating, Font, Footer, Foreground Color, Legend, Margin, Marimekko Totals, Markers, Marker Text Location, Master Detail Relationships, Maximum Hotspots, Name, No Data Contents, Notes, Padding, Pagination, Palette, Query, Relative Alignment, Render Page when Empty, Render Variable, Set Variable Values, Show Hover, Show Pointer Cursor, Size & Overflow, Style Variable, Subtitle, Suppression, Title, Title, Tooltips, Values, Visible, Visual Angle, Y Axis

## Marker

A marker, or symbol, that you can place at a static place in a chart. Markers can designate a point of significance that may help you analyze or understand the data.

### Properties of Marker

Aggregate Function, Data Item Value, Expression, Hierarchy, Hierarchy Unique Name, Label, Label Location, Level, Level Unique Name, Marker Shape, Master Detail Relationships, Member, Member Offset (%), Member Unique Name, Name, Properties, Property Unique Name, Query, Render Variable, Rollup Aggregate Function, Root Members Only, Set Sorting, Type

## Marker

A marker, or symbol, that you can place at a static place in a chart. Markers can designate a point of significance that may help you analyze or understand the data. This object applies only to legacy charts.

### Properties of Marker

Aggregate Function, Axis Assignment, Bar, Line, or Area Index, Box Type, Data Item Value, Expression, Hierarchy, Hierarchy Unique Name, Label, Label, Level, Level Unique Name, Marker Color, Marker Shape, Marker Size (pt), Master Detail Relationships, Member, Member Unique Name, Name, Numeric Value, Percentile, Percent of Axis, Position type, Properties, Property Unique Name, Query, Report Expression, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Standard Deviations, Type

## Marker

A marker on a point chart. Point charts include scatter charts, polar charts, radar charts, and bubble charts. This object applies only to legacy charts.

### Properties of Marker

[Box Type](#), [Data Item Value](#), [Expression](#), [Label](#), [Marker Color](#), [Marker Shape](#), [Marker Size \(pt\)](#), [Master Detail Relationships](#), [Numeric Value](#), [Percentile](#), [Percent of Axis](#), [Position type](#), [Position type](#), [Properties](#), [Query](#), [Report Expression](#), [Standard Deviations](#)

## MDX

A multidimensional expression (MDX) query against an OLAP data source.

### Properties of MDX

[Catalog](#), [Data Source](#), [MDX](#), [Name](#)

## Measure

A data item that defines a measure.

### Properties of Measure

[Dimension](#), [Label](#), [Member](#), [Member Unique Name](#), [Name](#), [Type](#)

## Member

A data item that defines a member.

### Properties of Member

[Detail](#), [Dimension](#), [Hierarchy](#), [Label](#), [Member](#), [Member Unique Name](#), [Name](#), [Type](#)

## Member Children Set

A set of data items that define the children set of a member.

### Properties of Member Children Set

[Detail](#), [Dimension](#), [Hierarchy](#), [Label](#), [Member](#), [Member Unique Name](#), [Name](#), [Set Sorting](#), [Type](#)

## Member Hierarchy

The organization of a dimension's members into a logical tree structure, with each member having one or more "parent" members and an arbitrary number of "child" members.

### Properties of Member Hierarchy

[Name](#)

## Member Property

A property that is associated with members of a level. Attributes can be used to refine a search within level members, or to provide additional information about members.

### Properties of Member Property

[Data Item](#), [Name](#)

## Member Property

A member property from either a level or a hierarchy.

### Properties of Member Property

[Dimension](#), [Hierarchy](#), [Label](#), [Level](#), [Name](#), [Property](#), [Property Unique Name](#), [Type](#)

## Member Set

A data item that represents a named set. Member sets supply the context for expressions that require evaluation for multiple dimensions. For example, they determine the contexts for which you can apply analytical functions such as rank.

### Properties of Member Set

[Data Item](#), [Name](#)

## Metrics Range Chart

A chart that superimposes target value markers, target range markers, and tolerance range markers over any number or combination of bar, line, and area charts.

### Properties of Metrics Range Chart

[Alternate Text](#), [Background Color](#), [Background Effects](#), [Background Image](#), [Baselines](#), [Border](#), [Box Type](#), [Category Axis](#), [Classes](#), [Clickable Regions](#), [Conditional Palette](#), [Conditional Styles](#), [Container Select](#), [Depth](#), [Drill-Through Definitions](#), [Floating](#), [Font](#), [Footer](#), [Foreground Color](#), [Legend](#), [Margin](#), [Marker Color](#), [Marker Label](#), [Markers](#), [Marker Text Location](#), [Master Detail Relationships](#), [Maximum Hotspots](#), [Name](#), [No Data Contents](#), [Notes](#), [Padding](#), [Pagination](#), [Palette](#), [Performance Pattern](#), [Query](#), [Range Label](#), [Relative Alignment](#), [Render Page when Empty](#), [Render Variable](#), [Set Variable Values](#), [Show Hover](#), [Show Pointer Cursor](#), [Size & Overflow](#), [Style Variable](#), [Subtitle](#), [Suppression](#), [Target Color](#), [Target Marker](#), [Target Marker Border Color](#), [Target Marker Position](#), [Target Range \(%\)](#), [Title](#), [Title](#), [Tolerance Color](#), [Tolerance Label](#), [Toolips](#), [Upper Range Skew \(%\)](#), [Value Location](#), [Visible](#), [Visual Angle](#), [Y1 Axis](#)

## Metric Studio Diagram

A Metric Studio diagram rendered as a static image.

### Properties of Metric Studio Diagram

[Alternate Text](#), [Description](#), [Diagram Identifier](#)

## Named Set

A reference to a named set from the model.

### Properties of Named Set

[Detail](#), [Dimension](#), [Hierarchy](#), [Label](#), [Name](#), [Set Sorting](#), [Type](#)

## Note

A note on a chart. The source of the note text can be static text, a query item, or a report expression.

### Properties of Note

[Background Effects](#), [Conditional Styles](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Master Detail Relationships](#), [Note Text](#), [Position](#), [Properties](#), [Render Variable](#), [Size & Overflow](#)

## Note

A note on a chart. The source of the note text can be static text, a query item, or a report expression. This object applies only to legacy charts.

### Properties of Note

[Bottom Position \(px\)](#), [Height \(px\)](#), [Left Position \(px\)](#), [Note Border](#), [Note Text](#), [Width \(px\)](#)

## Note Content

The content and style of a note.

### Properties of Note Content

[Background Color](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Master Detail Relationships](#), [Properties](#), [Query](#), [Style Variable](#), [Visible](#)

## No Value List Item

The default item in a list control. When the default item is selected when the report is run, it sets the value of the data item in the control to null. By default, the no value item is the first item in the list.

### Properties of No Value List Item

[Label](#), [Position in List](#)

## Numerical Axis

The numeric axis for a progressive chart, including labels, titles, range, scale, and gridlines. This object applies only to legacy charts.

### Properties of Numerical Axis

Axis Labels, Axis Line, Axis Title, Axis Title, Classes, Conditional Styles, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

## Numerical Axis

The numeric axis for a Pareto chart, including labels, titles, and gridlines. This object applies only to legacy charts.

### Properties of Numerical Axis

Axis Labels, Axis Line, Axis Title, Axis Title, Classes, Conditional Styles, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

## Numeric Axis

The axis that displays numbers.

### Properties of Numeric Axis

Axis Labels, Axis Line, Axis Range, Axis Title, Axis Title, Gridlines, Minor Gridlines

## Numeric Baseline

The baseline on a numeric axis.

### Properties of Numeric Baseline

Aggregate Function, Data Item Value, Expression, Hierarchy, Hierarchy Unique Name, Label, Level, Level Unique Name, Line Style, Master Detail Relationships, Member, Member Unique Name, Name, Properties, Property Unique Name, Query, Render Variable, Rollup Aggregate Function, Root Members Only, Set Sorting, Type

## Page

A page in a layout.

### Properties of Page

Background Color, Background Image, Border, Classes, Conditional Styles, Font, Foreground Color, Gradient, Horizontal Alignment, Margin, Master Detail Relationships, Name, Pagination, PDF Page Setup, Properties, Query, Render Variable, Size & Overflow, Style Variable, Text Flow & Justification

## Page Body

The main body of a page.

## Properties of Page Body

[Background Color](#), [Background Effects](#), [Background Image](#), [Border](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Padding](#), [Size & Overflow](#), [Spacing & Breaking](#), [Style Variable](#), [Text Flow & Justification](#), [Vertical Alignment](#), [White Space](#)

## Page Break Text

The text associated with a page break.

## Properties of Page Break Text

[Background Color](#), [Background Image](#), [Border](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Floating](#), [Font](#), [Foreground Color](#), [Margin](#), [Padding](#), [Relative Alignment](#), [Render Variable](#), [Size & Overflow](#), [Spacing & Breaking](#), [Style Variable](#), [Text Flow & Justification](#), [Visible](#)

## Page Footer

The footer of a page.

## Properties of Page Footer

[Background Color](#), [Background Effects](#), [Background Image](#), [Border](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Padding](#), [Size & Overflow](#), [Spacing & Breaking](#), [Style Variable](#), [Text Flow & Justification](#), [Vertical Alignment](#), [White Space](#)

## Page Header

The header of a page.

## Properties of Page Header

[Background Color](#), [Background Effects](#), [Background Image](#), [Border](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Padding](#), [Size & Overflow](#), [Spacing & Breaking](#), [Style Variable](#), [Text Flow & Justification](#), [Vertical Alignment](#), [White Space](#)

## Page Number

The number of a page.

## Properties of Page Number

[Background Color](#), [Classes](#), [Font](#), [Foreground Color](#), [Margin](#), [Number Style](#), [Relative Alignment](#)

## Page Set

The set of pages to render according to a grouping structure.

## Properties of Page Set

[Grouping & Sorting](#), [Master Detail Relationships](#), [Name](#), [Pagination](#), [Properties](#), [Query](#)

## Pareto Bars

The bars in a Pareto chart.

### Properties of Pareto Bars

[Axis Labels](#), [Axis Line](#), [Axis Range](#), [Axis Title](#), [Gridlines](#), [Minor Gridlines](#)

## Pareto Chart

A chart in which data series appear as colored sections stacked in columns or bars. The maximum of each column or bar represents the series total as a percentage of the overall total of all data series in the chart.

Negative values are not supported in Pareto charts.

### Properties of Pareto Chart

[Alternate Text](#), [Background Effects](#), [Bar Width](#), [Bevel](#), [Border](#), [Border Color](#), [Borders](#), [Box Type](#), [Category Baselines](#), [Chart Orientation](#), [Classes](#), [Clickable Regions](#), [Colored Regions](#), [Conditional Palette](#), [Conditional Styles](#), [Container Select](#), [Cumulative Line](#), [Depth](#), [Drill-Through Definitions](#), [Drop Shadow](#), [Floating](#), [Font](#), [Footer](#), [Foreground Color](#), [Legend](#), [Margin](#), [Markers](#), [Master Detail Relationships](#), [Material Effects](#), [Matrix rows and columns](#), [Maximum Hotspots](#), [Name](#), [No Data Contents](#), [Notes](#), [Numeric Baselines](#), [Padding](#), [Pagination](#), [Palette](#), [Plot Area Fill](#), [Query](#), [Relative Alignment](#), [Render Page when Empty](#), [Render Variable](#), [Set Variable Values](#), [Show Hover](#), [Show Pointer Cursor](#), [Show Values](#), [Size & Overflow](#), [Subtitle](#), [Summarize Small Items](#), [Suppression](#), [Title](#), [Toolips](#), [Visible](#)

## Pareto Chart

A chart in which data series appear as colored sections stacked in columns or bars. The maximum of each column or bar represents the series total as a percentage of the overall total of all data series in the chart. This object applies only to legacy charts.

Negative values are not supported in Pareto charts.

### Properties of Pareto Chart

[Alternate Text](#), [Background Color](#), [Background Effects](#), [Background Image](#), [Baselines](#), [Border](#), [Border Color](#), [Borders](#), [Box Type](#), [Category Axis](#), [Chart Orientation](#), [Classes](#), [Clickable Regions](#), [Conditional Palette](#), [Conditional Styles](#), [Connecting Lines](#), [Container Select](#), [Cumulative Line](#), [Depth](#), [Drill-Through Definitions](#), [Floating](#), [Font](#), [Footer](#), [Foreground Color](#), [Legend](#), [Margin](#), [Markers](#), [Marker Text Location](#), [Master Detail Relationships](#), [Maximum Hotspots](#), [Name](#), [No Data Contents](#), [Notes](#), [Padding](#), [Pagination](#), [Palette](#), [Pareto Axis](#), [Query](#), [Relative Alignment](#), [Render Page when Empty](#), [Render Variable](#), [Rotate Values](#), [Set Variable Values](#), [Show Hover](#), [Show Pointer Cursor](#), [Size & Overflow](#), [Style Variable](#), [Subtitle](#), [Suppression](#), [Title](#), [Toolips](#), [Value Location](#), [Values](#), [Visible](#), [Visual Angle](#)

## Pie Chart

A chart that uses sections of a circle as data markers to plot one or more data series. The size of each section is proportional to the value of each data series for a given category value. Each pie corresponds to a category value. This object applies only to legacy charts.

### Properties of Pie Chart

[Alternate Text](#), [Avoid Label Collision](#), [Axis Title](#), [Background Color](#), [Background Effects](#), [Background Image](#), [Border](#), [Border Color](#), [Borders](#), [Box Type](#), [Classes](#), [Clickable Regions](#), [Conditional Palette](#), [Conditional Styles](#), [Container Select](#), [Depth](#), [Drill-Through Definitions](#), [Floating](#), [Font](#), [Footer](#), [Foreground Color](#), [Hole Size \(%\)](#), [Labels](#), [Legend](#), [Margin](#), [Master Detail Relationships](#), [Maximum Hotspots](#), [Name](#), [No Data Contents](#), [Notes](#), [Padding](#), [Pagination](#), [Palette](#), [Pie Labels](#), [Query](#), [Relative Alignment](#), [Render Page when Empty](#), [Render Variable](#), [Set Variable Values](#), [Show Hover](#), [Show Pointer Cursor](#), [Size & Overflow](#), [Style Variable](#), [Subtitle](#), [Suppression](#), [Title](#), [Title](#), [Toolips](#), [Value Representation](#), [Values](#), [Visible](#)

## Pie Chart

A chart that uses sections of a circle as data markers to plot one or more data series. The size of each section is proportional to the value of each data series for a given category value. Each pie corresponds to a category value.

### Properties of Pie Chart

[Alternate Text](#), [Axis Title](#), [Background Effects](#), [Bevel](#), [Border](#), [Border Color](#), [Borders](#), [Box Type](#), [Classes](#), [Clickable Regions](#), [Conditional Palette](#), [Conditional Styles](#), [Container Select](#), [Depth](#), [Drill-Through Definitions](#), [Drop Shadow](#), [Exploded Slices](#), [First Slice Angle](#), [Floating](#), [Font](#), [Footer](#), [Foreground Color](#), [Hole Size \(%\)](#), [Legend](#), [Margin](#), [Master Detail Relationships](#), [Material Effects](#), [Matrix rows and columns](#), [Maximum Hotspots](#), [Name](#), [No Data Contents](#), [Notes](#), [Padding](#), [Pagination](#), [Palette](#), [Query](#), [Relative Alignment](#), [Render Page when Empty](#), [Render Variable](#), [Set Variable Values](#), [Show Hover](#), [Show Pointer Cursor](#), [Show Values](#), [Size & Overflow](#), [Slice Direction](#), [Subtitle](#), [Summarize Small Slices](#), [Suppression](#), [Title](#), [Toolips](#), [Visible](#)

## Pie Labels

The labels that will be drawn if multiple pie charts are rendered. If this object does not exist, no labels will be rendered.

### Properties of Pie Labels

[Classes](#), [Conditional Styles](#), [Drill-Through Definitions](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Maximum Truncation Characters](#), [Style Variable](#), [Truncation](#), [Truncation Text](#), [Visible](#)

## Point Color Legend Title

The title for the point color legend. If this object is empty, a default title is rendered, if available.

### Properties of Point Color Legend Title

[Master Detail Relationships](#), [Properties](#), [Query](#)

## Point Layer

A map layer that includes points, such as cities. The color and size of each point is determined by their respective measure.

### Properties of Point Layer

[Border Color](#), [Borders](#), [Color Legend Title](#), [Conditional Palette](#), [Conditional Styles](#), [Fill Effects](#), [Labels](#), [Map Drills](#), [Palette](#), [Show Data Range in Legend](#), [Show Features with No Data](#), [Size Legend Title](#), [Style Variable](#), [Values](#)

## Point Measure

The measure that determines the colors of points on a map chart.

### Properties of Point Measure

[Aggregate Function](#), [Conditional Styles](#), [Custom Label](#), [Data Format](#), [Expression](#), [Hierarchy](#), [Hierarchy Unique Name](#), [Label](#), [Level](#), [Level Unique Name](#), [Member](#), [Member Unique Name](#), [Name](#), [Property Unique Name](#), [Rollup Aggregate Function](#), [Root Members Only](#), [Set Sorting](#), [Style Variable](#), [Type](#)

## Point Size Legend Title

The title for the point size legend. If this object is empty, a default title is rendered, if available.

### Properties of Point Size Legend Title

[Master Detail Relationships](#), [Properties](#), [Query](#)

## Point Size Measure

The measure that determines the size of the points on a map chart.

### Properties of Point Size Measure

[Aggregate Function](#), [Conditional Styles](#), [Custom Label](#), [Data Format](#), [Expression](#), [Hierarchy](#), [Hierarchy Unique Name](#), [Label](#), [Level](#), [Level Unique Name](#), [Maximum Size \(pt\)](#), [Member](#), [Member Unique Name](#), [Minimum Size \(pt\)](#), [Name](#), [Property Unique Name](#), [Rollup Aggregate Function](#), [Root Members Only](#), [Set Sorting](#), [Style Variable](#), [Type](#)

## Polar Chart

A point chart that plots one or more data series against two measures. The radius of a data series is determined by one measure and the arc is determined by the other measure. Multiple points are plotted for each category. They can be distinguished with the help of tool tips if the Tool Tip property is set to Yes.

### Properties of Polar Chart

[Alternate Text](#), [Angular Axis](#), [Background Color](#), [Background Effects](#), [Background Image](#), [Baselines](#), [Border](#), [Border Color](#), [Borders](#), [Box Type](#), [Classes](#), [Clickable Regions](#), [Conditional Palette](#), [Conditional Styles](#), [Container Select](#), [Drill-Through Definitions](#), [Floating](#), [Font](#), [Footer](#), [Foreground Color](#),

[Legend](#), [Margin](#), [Markers](#), [Marker Size \(pt\)](#), [Marker Text Location](#), [Master Detail Relationships](#), [Maximum Hotspots](#), [Name](#), [No Data Contents](#), [Notes](#), [Padding](#), [Pagination](#), [Palette](#), [Point Shape](#), [Query](#), [Radial Axis](#), [Relative Alignment](#), [Render Page when Empty](#), [Render Variable](#), [Set Variable Values](#), [Show Hover](#), [Show Pointer Cursor](#), [Size & Overflow](#), [Spider Effects](#), [Style Variable](#), [Subtitle](#), [Suppression](#), [Title](#), [Title](#), [Tooltips](#), [Value Location](#), [Values](#), [Visible](#)

## Primary Axis

The generic numeric axis for the combination chart. This axis is normally shown as the top-left axis but can be in a different position depending on the chart orientation, as specified in the Chart Orientation property of the chart.

### Properties of Primary Axis

[Axis Labels](#), [Axis Line](#), [Axis Range](#), [Axis Title](#), [Axis Title](#), [Gridlines](#), [Minor Gridlines](#)

## Primary Bottom Axis

The numeric axis for a combination chart. This axis is normally rendered as the bottom-left axis, but can change depending on the orientation of the chart, as specified in the Chart Orientation property of the chart.

### Properties of Primary Bottom Axis

[Axis Labels](#), [Axis Line](#), [Axis Range](#), [Axis Title](#), [Axis Title](#), [Gridlines](#), [Minor Gridlines](#)

## Progressive Chart

A chart that uses columns as data markers to plot one category across a single measure. The top of the first column represents the starting value for the second column. This chart emphasizes the positive or negative contribution of each value to the total.

### Properties of Progressive Chart

[Alternate Text](#), [Background Effects](#), [Bar Width](#), [Bevel](#), [Border](#), [Border Color](#), [Borders](#), [Box Type](#), [Category Baselines](#), [Chart Orientation](#), [Classes](#), [Clickable Regions](#), [Colored Regions](#), [Conditional Palette](#), [Conditional Styles](#), [Connecting Lines](#), [Container Select](#), [Depth](#), [Drill-Through Definitions](#), [Drop Shadow](#), [Floating](#), [Font](#), [Footer](#), [Foreground Color](#), [Legend](#), [Margin](#), [Markers](#), [Master Detail Relationships](#), [Material Effects](#), [Matrix rows and columns](#), [Maximum Hotspots](#), [Name](#), [No Data Contents](#), [Notes](#), [Numeric Baselines](#), [Padding](#), [Pagination](#), [Plot Area Fill](#), [Progressive Palette](#), [Query](#), [Relative Alignment](#), [Render Page when Empty](#), [Render Variable](#), [Set Variable Values](#), [Show Hover](#), [Show Pointer Cursor](#), [Show Values](#), [Size & Overflow](#), [Subtitle](#), [Suppression](#), [Title](#), [Tooltips](#), [Total Column](#), [Visible](#)

## Progressive Chart

A chart that uses columns as data markers to plot one category across a single measure. The top of the first column represents the starting value for the second column. This chart emphasizes the positive or negative contribution of each value to the total. This object applies only to legacy charts.

## Properties of Progressive Chart

Alternate Text, Background Color, Background Effects, Background Image, Baselines, Border, Border Color, Borders, Box Type, Category Axis, Chart Orientation, Classes, Clickable Regions, Conditional Palette, Conditional Styles, Connecting Lines, Container Select, Depth, Drill-Through Definitions, First Column Color, Floating, Font, Footer, Foreground Color, Margin, Markers, Marker Text Location, Master Detail Relationships, Maximum Hotspots, Name, Negative Column Color, No Data Contents, Notes, Padding, Pagination, Palette, Positive Column Color, Progressive Axis, Query, Relative Alignment, Render Page when Empty, Render Variable, Rotate Values, Set Variable Values, Show Hover, Show Pointer Cursor, Size & Overflow, Style Variable, Subtitle, Suppression, Title, Title, Tooltips, Total Column, Value Location, Values, Visible, Visual Angle

## Prompt Button

A predefined button used in prompt pages. Its usage changes according to its Type property, which can be set to Cancel, Back, Next, Finish, or Reprompt.

## Properties of Prompt Button

Background Color, Background Image, Border, Classes, Conditional Styles, Floating, Font, Foreground Color, Margin, Name, Padding, Relative Alignment, Size & Overflow, Style Variable, Type

## Query

The data that is to be retrieved from the database. The query consists of a source, a selection, detailed and summary filters, and dimension information.

## Properties of Query

Auto Group & Summarize, Auto-Sort, Avoid Division by Zero, Cross Product Allowed, Define Member Sets, Execution Method, Execution Optimization, Generated SQL, Maximum Execution Time, Maximum Rows Retrieved, Maximum Tables, Maximum Text Blob Characters, Name, Outer Join Allowed, Override Dimension Info, Processing, Rollup Processing, SQL Join Syntax, Suppress, Use 1.x Behavior, Use Aggregate Cache, Use for Parameter Info, Use Local Cache, User SAP Member Cache, Use SAP MUN as Business Key, Use SQL Parameters, Use SQL With Clause

## Query Operation

Union, Intersect, Except (minus) operations on one or more queries that result in a projection list upon which other queries can be based.

## Properties of Query Operation

Duplicates, Name, Projection List, Set Operation

## Query Reference

A reference to another query defined in the same query set.

## Properties of Query Reference

Cardinality

## Radar Chart

A chart that integrates multiple axes into a single radial figure as lines or stacked areas.

### Properties of Radar Chart

[Alternate Text](#), [Background Color](#), [Background Effects](#), [Background Image](#), [Baselines](#), [Border](#), [Border Color](#), [Borders](#), [Box Type](#), [Category Axis](#), [Classes](#), [Clickable Regions](#), [Conditional Palette](#), [Conditional Styles](#), [Container Select](#), [Drill-Through Definitions](#), [Floating](#), [Font](#), [Footer](#), [Foreground Color](#), [Legend](#), [Margin](#), [Markers](#), [Marker Size \(pt\)](#), [Marker Text Location](#), [Master Detail Relationships](#), [Maximum Hotspots](#), [Name](#), [No Data Contents](#), [Notes](#), [Padding](#), [Pagination](#), [Palette](#), [Point Shape](#), [Query](#), [Radar Type](#), [Radial Axis](#), [Relative Alignment](#), [Render Page when Empty](#), [Render Variable](#), [Set Variable Values](#), [Show Hover](#), [Show Pointer Cursor](#), [Size & Overflow](#), [Spider Effects](#), [Style Variable](#), [Subtitle](#), [Suppression](#), [Title](#), [Title](#), [Toolips](#), [Value Location](#), [Values](#), [Visible](#)

## Radial Axis

The radial numeric axis for a polar chart or radar chart, including labels, titles, range, and scale.

### Properties of Radial Axis

[Axis Labels](#), [Axis Line](#), [Axis Title](#), [Axis Title](#), [Classes](#), [Conditional Styles](#), [Data Format](#), [Font](#), [Foreground Color](#), [Gridlines](#), [Include Zero For Auto Scale](#), [Maximum Value](#), [Minimum Value](#), [Minor Gridlines](#), [Scale](#), [Scale Interval](#), [Style Variable](#), [Use Same Range For All Instances](#), [Visible](#)

## Radial Measure

The measure that determines the distance between the center of the chart and each data marker.

### Properties of Radial Measure

[Aggregate Function](#), [Conditional Styles](#), [Custom Label](#), [Data Format](#), [Expression](#), [Hierarchy](#), [Hierarchy Unique Name](#), [Label](#), [Level](#), [Level Unique Name](#), [Member](#), [Member Unique Name](#), [Name](#), [Property Unique Name](#), [Rollup Aggregate Function](#), [Root Members Only](#), [Set Sorting](#), [Style Variable](#), [Type](#)

## Radio Button

A radio button in a static radio button group control.

### Properties of Radio Button

[Icon](#), [Label](#)

## Radio Button Group

A static radio button group control. Each radio button requires a corresponding row in the data table of the control.

### Properties of Radio Button Group

[Container Filter](#), [Container Select](#), [Control Enable](#), [Name](#), [Orientation](#), [Radio Buttons Definition](#), [Render Variable](#), [Set Variable Values](#), [Text Color](#)

## Radio Button Icon

The icon used to identify each radio button in a data radio button group control. The icons are obtained from a data item inserted into the control.

### Properties of Radio Button Icon

[Data Item](#)

## Radio Button Label

The label used to identify each radio button in a data radio button group control. Each label is determined by the data items inserted into the control.

### Properties of Radio Button Label

[Data Item](#)

## Radio Button Value

Data items that are used to determine what appears as the radio button labels in a data radio button group control.

### Properties of Radio Button Value

[Data Item](#)

## Region Color Legend Title

The title for the region color legend. If this object is empty, a default title is rendered, if available.

### Properties of Region Color Legend Title

[Master Detail Relationships](#), [Properties](#), [Query](#)

## Region Layer

A map layer that includes regions, such as provinces.

### Properties of Region Layer

[Border Color](#), [Borders](#), [Color Legend Title](#), [Conditional Palette](#), [Conditional Styles](#), [Fill Effects](#), [Labels](#), [Map Drills](#), [Palette](#), [Show Data Range in Legend](#), [Show Features with No Data](#), [Style Variable](#), [Values](#)

## Region Measure

The measure that determines the colors of regions on a map chart.

### Properties of Region Measure

[Aggregate Function](#), [Conditional Styles](#), [Custom Label](#), [Data Format](#), [Expression](#), [Hierarchy](#), [Hierarchy Unique Name](#), [Label](#), [Level](#), [Level Unique Name](#), [Member](#), [Member Unique Name](#),

Name, Property Unique Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Style Variable, Type

## Regression Line

A regression line for a bubble chart or scatter chart.

### Properties of Regression Line

Box Type, Line Styles, Number of Regression Lines, Polynomial Exponent, Properties, Regression Type

## Repeater

A table into which you can insert items that will be repeated.

### Properties of Repeater

Grouping & Sorting, Master Detail Relationships, Name, No Data Contents, Pagination, Properties, Query, Render Page when Empty, Render Variable, Rows Per Page, Share Result Set, Suppression

## Repeater Table

Renders query data in a table.

### Properties of Repeater Table

Across, Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Down, Drop Shadow, Floating, Font, Foreground Color, Grouping & Sorting, Horizontal Alignment, Margin, Master Detail Relationships, Name, No Data Contents, Pagination, Properties, Query, Relative Alignment, Render Page when Empty, Render Variable, Repeater Direction, Share Result Set, Size & Overflow, Style Variable, Summary Text, Suppression, Table Properties, Text Flow & Justification, Visible

## Repeater Table Cell

The contents of a repeater table object.

### Properties of Repeater Table Cell

Background Color, Background Effects, Background Image, Border, Box Type, Classes, Conditional Styles, Font, Foreground Color, Horizontal Alignment, Padding, Render Variable, Size & Overflow, Spacing & Breaking, Style Variable, Text Flow & Justification, Vertical Alignment, White Space

## Rich Text Item

Inserts an object that is used to render a subset of HTML in the layout. The HTML may come from either a static or dynamic source, and the object will also render in PDF output. For information about what elements are allowed in rich text items, see Elements Supported in Rich Text Items.

### Properties of Rich Text Item

Aggregate Function, Data Item Label, Data Item Value, Description, Expression, Hierarchy, Hierarchy Unique Name, HTML, HTML Source Variable, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Name, Property Unique Name, Render Variable, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Source Type, Type

## Row Number

Shows the row numbers in a column.

### Properties of Row Number

Background Color, Classes, Data Format, Font, Foreground Color, Margin, Relative Alignment

## Scatter Chart

A point chart that plots one or more data series against two measures. Multiple points are plotted for each category. This object applies only to legacy charts.

### Properties of Scatter Chart

Alternate Text, Background Color, Background Effects, Background Image, Baselines, Border, Border Color, Borders, Box Type, Classes, Clickable Regions, Conditional Palette, Conditional Styles, Container Select, Drill-Through Definitions, Floating, Font, Footer, Foreground Color, Legend, Margin, Markers, Marker Size (pt), Marker Text Location, Master Detail Relationships, Maximum Hotspots, Name, No Data Contents, Notes, Padding, Pagination, Palette, Point Shape, Query, Regression Line, Relative Alignment, Render Page when Empty, Render Variable, Set Variable Values, Show Hover, Show Pointer Cursor, Size & Overflow, Style Variable, Subtitle, Suppression, Title, Title, Tooltips, Value Location, Values, Visible, X Axis, Y Axis

## Scatter Chart

A point chart that plots one or more data series against two measures. Multiple points are plotted for each category.

### Properties of Scatter Chart

Alternate Text, Background Effects, Border, Box Type, Classes, Clickable Regions, Color by Value, Colored Regions, Conditional Palette, Conditional Styles, Container Select, Drill-Through Definitions, Drop Shadow, Floating, Font, Footer, Foreground Color, Legend, Margin, Markers, Master Detail Relationships, Material Effects, Matrix rows and columns, Maximum Hotspots, Name, No Data Contents, Notes, Numeric Baselines, Padding, Pagination, Palette, Plot Area Fill, Query, Relative Alignment, Render Page when Empty, Render Variable, Set Variable Values, Show Hover, Show Pointer Cursor, Show Values, Size & Overflow, Subtitle, Suppression, Title, Tooltips, Trendlines, Visible

## Scatter Marker

A marker, or symbol, that you can place at a static place in a chart. Markers can designate a point of significance that may help you analyze or understand the data.

## Properties of Scatter Marker

Aggregate Function, Expression, Hierarchy, Hierarchy Unique Name, Label, Label Location, Level, Level Unique Name, Marker Shape, Marker Size (pt), Master Detail Relationships, Member, Member Unique Name, Name, Properties, Property Unique Name, Query, Render Variable, Rollup Aggregate Function, Root Members Only, Set Sorting, Type, X-Axis Data Item Value, Y-Axis Data Item Value

## Secondary Axis

The generic numeric axis for the combination chart. This axis is normally shown as the top-right axis but can be in a different position depending on the chart orientation, as specified in the Chart Orientation property of the chart.

### Properties of Secondary Axis

Axis Labels, Axis Line, Axis Range, Axis Title, Gridlines, Minor Gridlines

## Secondary Bottom Axis

The numeric axis for a combination chart. This axis is normally shown as the bottom-right axis, but can change depending on the orientation of the chart, as specified in the Chart Orientation property of the chart.

### Properties of Secondary Bottom Axis

Axis Labels, Axis Line, Axis Range, Axis Title, Gridlines, Minor Gridlines

## Select & Search Prompt

An advanced prompt control that allows you to search for values. You cannot use this prompt control with SAP BW data sources.

### Properties of Select & Search Prompt

Box Type, Cascade Source, Case Insensitive, Choices Deselect All Text, Choices Select All Text, Choices Text, Conditional Styles, Data Format, Deselect Text, Display Value, Floating, Hide Adornments, Insert Text, Keywords Text, Multi-Select, Name, Options Text, Parameter, Properties, Query, Remove Text, Render Variable, Required, Results Deselect All Text, Results Select All Text, Results Text, Rows Per Page, Search Instructions Text, Search Text, Sorting, Static Choices, Style Variable, Use Value, Visible

## Set Expression

A set of members.

### Properties of Set Expression

Detail, Dimension, Expression, Hierarchy, Label, Name, Set Sorting, Type

## Singleton

A query item that you can insert anywhere in the layout of your report where there is no query associated. When the report is run, the singleton object retrieves only the first row value for that query. Singletons are useful for adding boilerplate text to a report, such as a company name or address, to add overall calculations in a report, or to add multidimensional cell calculations in a report.

### Properties of Singleton

[Name](#), [Properties](#), [Query](#), [Render Variable](#)

## Slicer Member Set

A set expression that returns members from a single hierarchy of a single dimension.

### Properties of Slicer Member Set

[Expression](#)

## SQL

An SQL query against a relational data source.

### Properties of SQL

[Data Source](#), [Name](#), [SQL](#), [SQL Syntax](#)

## Summary Filter

A set of conditions in a query that narrow the scope of the data returned. A summary filter is applied after aggregation is complete.

### Properties of Summary Filter

[Definition](#), [Scope](#), [Usage](#)

## Tab

A tab in a static tab control.

### Properties of Tab

[Icon](#), [Label](#)

## Tab Control

A static tab control. Each tab requires a corresponding row in the data table of the control.

### Properties of Tab Control

[Container Select](#), [Current Tab](#), [Name](#), [Render Variable](#), [Set Variable Values](#), [Size](#), [Tab Orientation](#), [Tabs Definition](#), [Tab Width](#)

## Tab Icon

The icon used to identify each tab in a data tab control. The icons are obtained from a data item inserted into the control.

### Properties of Tab Icon

[Data Item](#)

## Tab Label

The label used to identify each tab in a data tab control. Each label is determined by the data items inserted into the control.

### Properties of Tab Label

[Data Item](#)

## Table

A collection of cells in which objects can be organized in a grid fashion.

### Properties of Table

[Background Color](#), [Background Image](#), [Border](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Drop Shadow](#), [Floating](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Margin](#), [Name](#), [Pagination](#), [Relative Alignment](#), [Render Variable](#), [Size & Overflow](#), [Style Variable](#), [Summary Text](#), [Table Properties](#), [Text Flow & Justification](#), [Visible](#)

## Table Cell

The cells within a row.

### Properties of Table Cell

[Background Color](#), [Background Effects](#), [Background Image](#), [Border](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Padding](#), [Size & Overflow](#), [Spacing & Breaking](#), [Style Variable](#), [Table Header](#), [Text Flow & Justification](#), [Vertical Alignment](#), [White Space](#)

## Table of Contents

A collection of Table of Contents Entry objects that refer the same table of contents. The table of contents is generated in the rendered output.

### Properties of Table of Contents

[Background Color](#), [Background Image](#), [Border](#), [Box Type](#), [Classes](#), [Conditional Styles](#), [Floating](#), [Font](#), [Foreground Color](#), [Horizontal Alignment](#), [Margin](#), [No Data Contents](#), [Padding](#), [Render Page when Empty](#), [Render Variable](#), [Size & Overflow](#), [Spacing & Breaking](#), [Style Variable](#), [Table of Contents Name](#), [Text Flow & Justification](#), [Visible](#), [White Space](#)

## Table of Contents Entry

An entry in the table of contents.

### Properties of Table of Contents Entry

Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Floating, Font, Foreground Color, Heading Level, Horizontal Alignment, Margin, Padding, Render Variable, Size & Overflow, Spacing & Breaking, Style Variable, Table of Contents, Text Flow & Justification, Visible, White Space

## Table Row

The rows in a table.

### Properties of Table Row

Background Color, Background Image, Box Type, Classes, Conditional Styles, Font, Foreground Color, Horizontal Alignment, Style Variable, Vertical Alignment

## Tab Value

Data items that are used to determine what appears as the tab labels in a data tab control.

### Properties of Tab Value

Data Item

## Target Measure

The target measure for a bullet chart.

### Properties of Target Measure

Aggregate Function, Custom Label, Data Format, Expression, Hierarchy, Hierarchy Unique Name, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Property Unique Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Type

## Target Measure

The target measure for a metrics range chart.

### Properties of Target Measure

Aggregate Function, Conditional Styles, Custom Label, Data Format, Expression, Hierarchy, Hierarchy Unique Name, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Property Unique Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Style Variable, Type

## Text Box Prompt

A prompt control that allows you to type in a value.

## Properties of Text Box Prompt

Background Color, Border, Box Type, Choices Deselect All Text, Choices Select All Text, Choices Text, Conditional Styles, Default Selections, Deselect Text, Floating, Font, Foreground Color, From Text, Hide Adornments, Hide Text, Highest Value Text, Insert Text, Lowest Value Text, Multi-Line, Multi-Select, Name, Numbers Only, Parameter, Range, Relative Alignment, Remove Text, Render Variable, Required, Size & Overflow, Style Variable, To Text, Use Thousands Separator, Visible

## Text Item

A text item in a report. The content can be static text, or it can come from a query item or report expression.

## Properties of Text Item

Aggregate Function, Background Color, Background Image, Border, Box Type, Classes, Conditional Styles, Data Format, Data Item Label, Data Item Value, Drill-Through Definitions, Expression, Floating, Font, Foreground Color, Hierarchy, Hierarchy Unique Name, Label, Level, Level Unique Name, Margin, Maximum Characters, Member, Member Unique Name, Name, Name, Padding, Property Unique Name, Relative Alignment, Render Variable, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Size & Overflow, Source Type, Spacing & Breaking, Style Variable, Text, Text Flow & Justification, Text Source Variable, Type, Use Detail Value on Page, Visible

## Time

The time when the report runs.

## Properties of Time

Background Color, Classes, Data Format, Font, Foreground Color, Margin, Relative Alignment

## Time Prompt

An advanced prompt control that allows you to select a time value.

## Properties of Time Prompt

Box Type, Choices Deselect All Text, Choices Select All Text, Choices Text, Clock Mode, Conditional Styles, Default Selections, Deselect Text, Display Milliseconds, Display Seconds, Floating, From Text, Hide Adornments, Highest Value Text, Insert Text, Lowest Value Text, Multi-Select, Name, Parameter, Range, Remove Text, Render Variable, Required, Select UI, Style Variable, To Text, Visible

## Toggle Button Bar

A static toggle button bar control. Each toggle button requires a corresponding row in the data table of the control.

### Properties of Toggle Button Bar

Button Gap, Buttons Definition, Button Width, Container Select, Control Enable, Name, Orientation, Render Variable, Set Variable Values

## Tolerance Measure

The tolerance measure for a metrics chart.

### Properties of Tolerance Measure

Aggregate Function, Conditional Styles, Custom Label, Data Format, Expression, Hierarchy, Hierarchy Unique Name, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Property Unique Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Style Variable, Type

## Total Column

A column or bar representing the total cumulative value of all other columns or bars in a progressive chart.

### Properties of Total Column

Aggregate Function, Data Format, Data Item Value, Expression, Hierarchy, Hierarchy Unique Name, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Properties, Property Unique Name, Report Expression, Rollup Aggregate Function, Root Members Only, Set Sorting, Text, Total Column Color, Type

## Total Column Label

The label to be rendered for the total column.

### Properties of Total Column Label

Properties

## Tree Prompt

A data-driven prompt control that shows hierarchical information and allows you to select one or more members.

### Properties of Tree Prompt

Cascade Source, Conditional Styles, Default Selections, Deselect Text, Floating, Hide Adornments, Multi-Select, Name, Parameter, Pre-populate, Pre-populate Levels, Query, Render Variable, Required, Results Deselect All Text, Results Select All Text, Rows Per Page, Size & Overflow, Sorting, Style Variable, Use Value, Visible

## Trendline

A line or curve that indicates the general direction of the data over time.

**Properties of Trendline**[Based on](#), [Line Style](#), [Render Variable](#), [Trendline Label](#), [Type](#)**Trendline Label**

A label that will be shown in the legend for the trend line.

**Properties of Trendline Label**[Conditional Styles](#), [Data Format](#), [Properties](#), [Query](#)**Value Prompt**

A prompt control that allows you to select one or more values from a list.

**Properties of Value Prompt**[Auto-Submit](#), [Background Color](#), [Box Type](#), [Cascade Source](#), [Choices Deselect All Text](#), [Choices Select All Text](#), [Choices Text](#), [Conditional Styles](#), [Data Format](#), [Default Selections](#), [Deselect Text](#), [Display Value](#), [Floating](#), [Font](#), [Foreground Color](#), [From Text](#), [Header Text](#), [Hide Adornments](#), [Highest Value Text](#), [Insert Text](#), [Lowest Value Text](#), [Multi-Select](#), [Name](#), [Parameter](#), [Pre-populate](#), [Properties](#), [Query](#), [Range](#), [Relative Alignment](#), [Remove Text](#), [Render Variable](#), [Required](#), [Results Deselect All Text](#), [Results Select All Text](#), [Rows Per Page](#), [Select UI](#), [Size & Overflow](#), [Sorting](#), [Static Choices](#), [Style Variable](#), [To Text](#), [Use Value](#), [Visible](#)**Variable**

A report variable.

**Properties of Variable**[Name](#), [Report Expression](#), [Type](#)**Variable Text Item**

A text item that renders the value of an active report variable.

**Properties of Variable Text Item**[Active Report Variable](#), [Background Color](#), [Background Image](#), [Border](#), [Box Type](#), [Classes](#), [Floating](#), [Font](#), [Foreground Color](#), [Margin](#), [Padding](#), [Relative Alignment](#), [Render Variable](#), [Size & Overflow](#), [Spacing & Breaking](#), [Text Flow & Justification](#), [Visible](#)**Win-Loss Chart**

A microchart in which the value of each column is either 1 or -1, often denoting a win or a loss.

**Properties of Win-Loss Chart**[Allow Ties](#), [Axis Line](#), [Background Color](#), [Background Effects](#), [Background Image](#), [Border](#), [Border Color](#), [Borders](#), [Box Type](#), [Chart Orientation](#), [Classes](#), [Conditional Styles](#), [Drill-Through Definitions](#), [Floating](#), [Font](#), [Footer](#), [Foreground Color](#), [Loss Color](#), [Margin](#), [Master Detail Relationships](#), [Maximum Hotspots](#), [Name](#), [No Data Contents](#), [Padding](#), [Pagination](#), [Query](#), [Relative Alignment](#),

[Render Page when Empty](#), [Render Variable](#), [Size & Overflow](#), [Style Variable](#), [Subtitle](#), [Suppression](#), [Title](#), [Toolips](#), [Visible](#), [Win Color](#), [Win-Loss Threshold](#)

## Win-Loss Measure

The measure for a win-loss chart.

### Properties of Win-Loss Measure

[Aggregate Function](#), [Custom Label](#), [Data Format](#), [Expression](#), [Hierarchy](#), [Hierarchy Unique Name](#), [Label](#), [Level](#), [Level Unique Name](#), [Member](#), [Member Unique Name](#), [Name](#), [Property Unique Name](#), [Rollup Aggregate Function](#), [Root Members Only](#), [Set Sorting](#), [Type](#)

## X Axis

The horizontal numeric axis for the chart, including labels, titles, range, and scale.

### Properties of X Axis

[Axis Labels](#), [Axis Line](#), [Axis Title](#), [Axis Title](#), [Classes](#), [Conditional Styles](#), [Data Format](#), [Font](#), [Foreground Color](#), [Gridlines](#), [Include Zero For Auto Scale](#), [Maximum Value](#), [Minimum Value](#), [Minor Gridlines](#), [Scale](#), [Scale Interval](#), [Style Variable](#), [Use Same Range For All Instances](#), [Visible](#)

## X Axis

The x-axis on a scatter or bubble chart.

### Properties of X Axis

[Axis Labels](#), [Axis Line](#), [Axis Range](#), [Axis Title](#), [Axis Title](#), [Gridlines](#), [Minor Gridlines](#)

## X Axis

The axis line for an ordinal, or non-numeric, axis.

### Properties of X Axis

[Allow 45° Rotation](#), [Allow 90° Rotation](#), [Allow Skip](#), [Allow Stagger](#), [Axis Labels](#), [Axis Line](#), [Axis Title](#), [Axis Title](#), [Classes](#), [Conditional Styles](#), [Display Frequency](#), [Drill-Through Definitions](#), [First Label Index](#), [Font](#), [Foreground Color](#), [Gridlines](#), [Horizontal Alignment](#), [Label Control](#), [Maximum Truncation Characters](#), [Minor Gridlines](#), [Style Variable](#), [Truncation](#), [Truncation Text](#), [Visible](#)

## X Axis Measure

The measure for the horizontal axis of a scatter chart or bubble chart.

### Properties of X Axis Measure

[Aggregate Function](#), [Conditional Styles](#), [Custom Label](#), [Data Format](#), [Expression](#), [Hierarchy](#), [Hierarchy Unique Name](#), [Label](#), [Level](#), [Level Unique Name](#), [Member](#), [Member Unique Name](#), [Name](#), [Property Unique Name](#), [Rollup Aggregate Function](#), [Root Members Only](#), [Set Sorting](#), [Style Variable](#), [Type](#)

## Y1 Axis

The numeric axis of a metrics chart, or the primary numeric axis of a combination chart.

### Properties of Y1 Axis

[Axis Labels](#), [Axis Line](#), [Axis Title](#), [Axis Title](#), [Classes](#), [Conditional Styles](#), [Data Format](#), [Font](#), [Foreground Color](#), [Gridlines](#), [Include Zero For Auto Scale](#), [Maximum Value](#), [Minimum Value](#), [Minor Gridlines](#), [Scale](#), [Scale Interval](#), [Style Variable](#), [Use Same Range For All Instances](#), [Visible](#)

## Y2 Axis

The secondary numeric axis of a combination chart.

### Properties of Y2 Axis

[Axis Labels](#), [Axis Line](#), [Axis Title](#), [Axis Title](#), [Classes](#), [Conditional Styles](#), [Data Format](#), [Font](#), [Foreground Color](#), [Gridlines](#), [Include Zero For Auto Scale](#), [Maximum Value](#), [Minimum Value](#), [Minor Gridlines](#), [Scale](#), [Scale Interval](#), [Style Variable](#), [Use Same Range For All Instances](#), [Visible](#)

## Y Axis

The vertical numeric axis for the chart, including labels, titles, range, and scale.

### Properties of Y Axis

[Axis Labels](#), [Axis Line](#), [Axis Title](#), [Axis Title](#), [Classes](#), [Conditional Styles](#), [Data Format](#), [Font](#), [Foreground Color](#), [Gridlines](#), [Include Zero For Auto Scale](#), [Maximum Value](#), [Minimum Value](#), [Minor Gridlines](#), [Scale](#), [Scale Interval](#), [Style Variable](#), [Use Same Range For All Instances](#), [Visible](#)

## Y Axis

The y-axis on a scatter or bubble chart.

### Properties of Y Axis

[Axis Labels](#), [Axis Line](#), [Axis Range](#), [Axis Title](#), [Axis Title](#), [Gridlines](#), [Minor Gridlines](#)

## Y Axis

The axis line for an ordinal, or non-numeric, axis.

### Properties of Y Axis

[Allow 45° Rotation](#), [Allow 90° Rotation](#), [Allow Skip](#), [Allow Stagger](#), [Axis Labels](#), [Axis Line](#), [Axis Title](#), [Axis Title](#), [Classes](#), [Conditional Styles](#), [Display Frequency](#), [Drill-Through Definitions](#), [First Label Index](#), [Font](#), [Foreground Color](#), [Gridlines](#), [Horizontal Alignment](#), [Label Control](#), [Maximum Truncation Characters](#), [Minor Gridlines](#), [Style Variable](#), [Truncation Text](#), [Visible](#)

## Y Axis Measure

The measure for the vertical axis of a scatter chart or bubble chart.

### Properties of Y Axis Measure

Aggregate Function, Conditional Styles, Custom Label, Data Format, Expression, Hierarchy, Hierarchy Unique Name, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Property Unique Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Style Variable, Type

## Z Axis

The third numeric axis for a three-dimensional scatter chart, including labels, titles, range, and scale.

### Properties of Z Axis

Axis Labels, Axis Line, Axis Title, Axis Title, Classes, Conditional Styles, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

## Z Axis

The vertical numeric axis for a three-dimensional combination chart, including labels, titles, range, and scale.

### Properties of Z Axis

Axis Labels, Axis Line, Axis Title, Axis Title, Classes, Conditional Styles, Data Format, Font, Foreground Color, Gridlines, Include Zero For Auto Scale, Maximum Value, Minimum Value, Minor Gridlines, Scale, Scale Interval, Style Variable, Use Same Range For All Instances, Visible

## Z Axis Measure

The measure for the third axis of a three-dimensional scatter chart.

### Properties of Z Axis Measure

Aggregate Function, Conditional Styles, Custom Label, Data Format, Expression, Hierarchy, Hierarchy Unique Name, Label, Level, Level Unique Name, Member, Member Unique Name, Name, Property Unique Name, Rollup Aggregate Function, Root Members Only, Set Sorting, Style Variable, Type

## Report Studio Properties

The following is a list of properties available in the lower left pane of Report Studio.

### 3-D Viewing Angle

Specifies the 3-D viewing angle of the chart.

#### Applies to

3-D Combination Chart, 3-D Scatter Chart

## Absolute Position

Specifies that the legend will be positioned absolutely, by setting its pixel position.

### Applies to

[Legend](#)

## Across

Sets the number of times across, or horizontally, that the contents of the object may be rendered.

The default value depends on the setting of the Repeater Direction property. If it is set to Left to right, top to bottom, the default is one. If it is set to Top to bottom, left to right, the default is 20.

### Applies to

[Repeater Table](#)

## Active Report Variable

Specifies the variable to use in the control.

### Applies to

[Variable Text Item](#)

## Aggregate Data Item

Specifies the data item that is used to calculate the summary or aggregation.

### Applies to

[Dimensional Edge Summary](#), [Edge Summary](#), [List Summary](#)

## Aggregate Function

Specifies the type of aggregation to apply. The Automatic setting means that the application groups or summarizes based on the data type. The Summarize setting means that any setting found in the model will be used to determine the type of aggregation. The default setting is Automatic. For more information about each summary function, see the Report Studio Professional Authoring User Guide.

### Applies to

[Angular Measure](#), [Area](#), [Bar](#), [Baseline](#), [Baseline](#), [Baseline](#), [Bubble Measure](#), [Bubble Measure](#), [Bullet Measure](#), [Category Baseline](#), [Chart Measure](#), [Chart Node Member](#), [Chart Text Item](#), [Chart Text Item](#), [Color by Value Measure](#), [Colored Region](#), [Colored Region](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Cumulation Line Label](#), [Data Item](#), [Default Measure](#), [Default Measure](#), [Gauge Axis](#), [HTML Item](#), [Image](#), [Line](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [Map Location](#), [Map Location](#), [Map Refinement Location](#), [Map Refinement Location](#), [Marker](#), [Marker](#), [Numeric Baseline](#), [Point Measure](#), [Point Size Measure](#), [Radial Measure](#), [Region Measure](#), [Rich Text Item](#), [Scatter Marker](#), [Target Measure](#),

[Target Measure](#), [Text Item](#), [Tolerance Measure](#), [Total Column](#), [Win-Loss Measure](#), [X Axis Measure](#), [Y Axis Measure](#), [Z Axis Measure](#)

## Aggregation Method

Specifies the type of summary or aggregation that is applied.

### Applies to

[Dimensional Edge Summary](#), [Edge Summary](#), [List Summary](#)

## Allow 45° Rotation

Specifies whether the labels can be rotated 45 degrees if the labels are long.

### Applies to

[Category Axis](#), [X Axis](#), [Y Axis](#)

## Allow 90° Rotation

Specifies whether the labels can be rotated 90 degrees if the labels are long.

### Applies to

[Category Axis](#), [X Axis](#), [Y Axis](#)

## Allow Skip

Specifies whether some labels can be skipped if they are long.

### Applies to

[Category Axis](#), [X Axis](#), [Y Axis](#)

## Allow Sorting

Specifies whether sorting is allowed for the column.

### Applies to

[List Column](#), [List Column Body](#), [List Column Title](#)

## Allow Stagger

Specifies whether the labels can be staggered if they are long.

### Applies to

[Category Axis](#), [X Axis](#), [Y Axis](#)

## Allow Ties

Specifies whether data values that equal the Win Loss Threshold property are converted to zero and mapped on the zero line.

**Applies to**[Win-Loss Chart](#)

## Alternate Text

Specifies a text alternative for non-text objects such as images and charts. Use to make reports accessible for people who use screen readers.

**Applies to**

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Combination Chart](#), [Combination Chart](#), [Gauge Chart](#), [Gauge Chart](#), [Image](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Metric Studio Diagram](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Radar Chart](#), [Scatter Chart](#), [Scatter Chart](#)

## Angular Axis

Specifies whether the axis is rendered.

**Applies to**[Polar Chart](#)

## Application

Specifies if the condition will be applied before or after aggregation and summarization. When true, the condition will apply to the summarized rows, and a fact, or aggregate, in the expression will be interpreted as the aggregated value of the summarized rows. When false, the condition will apply to the detail database rows from the tabular result set prior to aggregation or summarization, and a fact, or aggregate, in the expression will be interpreted as the individual database value before it has been summarized. This property has no effect on OLAP data sources, on references to non-aggregate items, or when automatic summarization is disabled.

**Applies to**[Detail Filter](#)

## Apply Single Class

Specifies whether to apply all of the classes defined or only one class. When set to Yes, the last class that was applied is used. For example, if you applied a class to a crosstab intersection and another class to the member fact cells of a crosstab row, the class applied to the intersection is the last class applied to the fact cells. For information about the order in which classes are applied, see the Report Studio Professional Authoring User Guide.

This property is set to Yes when you are upgrading a report so that the upgraded report will look the same as the original report. This is because objects in reports created using earlier versions of IBM Cognos BI support only one class.

**Applies to**[Crosstab Fact Cells](#)

## Area Shape

Specifies the shape of an area in an area chart.

### Applies to

[Area](#)

## Auto Font Size

Specifies whether to automatically resize the font to fit the labels.

### Applies to

[Axis Labels](#), [Axis Title](#), [Legend](#), [Legend Title](#)

## Auto Group & Summarize

Specifies whether the application will apply suggested aggregate functions to aggregate data items and group all non-aggregate data items, producing groups and summary rows. If it is set to No, detail rows will be rendered.

### Applies to

[Query](#)

## Auto-Sort

When running the report, specifies whether to automatically sort based on data type.

### Applies to

[Query](#)

## Auto-Submit

Specifies whether the application submits the prompt page automatically, as soon as a value is changed.

### Applies to

[Value Prompt](#)

## Auto Truncation

Specifies whether to allow truncation of text.

### Applies to

[Legend](#)

## Avoid Division by Zero

Specifies whether the application will return a null value when it encounters a division by zero. This property applies only to relational data sources.

**Applies to**[Query](#)

## Avoid Label Collision

Controls how labels are arranged. If set to false, the chart uses the default positions. If set to true, the chart uses a different layout to avoid label collision. To keep existing reports unchanged, set this property to false.

**Applies to**[Pie Chart](#)

## Axis Angles

Specifies the start and end angle and the direction of the gauge axis. All angles are measured in degrees starting from the three o'clock position.

**Applies to**[Gauge Axis](#)

## Axis Assignment

Specifies the numeric axis to use. This property applies only to legacy charts.

**Applies to**[Area, Bar, Baseline, Line, Marker](#)

## Axis Assignment

Specifies which numeric axis to use.

**Applies to**[Baseline](#)

## Axis Direction

## Axis Justification

Specifies whether the data labels are aligned to the inside or outside of the gauge axis.

**Applies to**[Gauge Axis](#)

## Axis Labels

Specifies whether to show or hide axis labels.

### Applies to

[Category axis](#), [Cumulation Line](#), [Gauge Axis](#), [Numeric Axis](#), [Pareto Bars](#), [Primary Axis](#), [Primary Bottom Axis](#), [Secondary Axis](#), [Secondary Bottom Axis](#), [X Axis](#), [Y Axis](#)

## Axis Labels

Specifies whether to show or hide axis labels. This property applies only to legacy charts.

### Applies to

[Angular Axis](#), [Category Axis](#), [Cumulation Line Axis](#), [Numerical Axis](#), [Numerical Axis](#), [Radial Axis](#), [X Axis](#), [X Axis](#), [Y1 Axis](#), [Y2 Axis](#), [Y Axis](#), [Y Axis](#), [Z Axis](#), [Z Axis](#)

## Axis Line

Specifies the properties of the axis line in a chart. This property applies only to legacy charts.

### Applies to

[Angular Axis](#), [Category Axis](#), [Cumulation Line Axis](#), [Numerical Axis](#), [Numerical Axis](#), [Radial Axis](#), [Win-Loss Chart](#), [X Axis](#), [X Axis](#), [Y1 Axis](#), [Y2 Axis](#), [Y Axis](#), [Y Axis](#), [Z Axis](#), [Z Axis](#)

## Axis Line

Specifies the properties of the axis line in a chart.

### Applies to

[Category axis](#), [Cumulation Line](#), [Numeric Axis](#), [Pareto Bars](#), [Primary Axis](#), [Primary Bottom Axis](#), [Secondary Axis](#), [Secondary Bottom Axis](#), [X Axis](#), [Y Axis](#)

## Axis Line

Specifies the properties of the gauge axis line in a gauge chart.

### Applies to

[Gauge Axis](#)

## Axis Range

Specifies the appearance of the range of values on an axis.

### Applies to

[Cumulation Line](#), [Gauge Axis](#), [Numeric Axis](#), [Pareto Bars](#), [Primary Axis](#), [Primary Bottom Axis](#), [Secondary Axis](#), [Secondary Bottom Axis](#), [X Axis](#), [Y Axis](#)

## Axis Title

Specifies whether an axis title is rendered.

## Applies to

Angular Axis, Category axis, Category Axis, Cumulation Line, Numerical Axis, Numerical Axis, Numeric Axis, Pareto Bars, Primary Axis, Primary Bottom Axis, Radial Axis, Secondary Axis, Secondary Bottom Axis, X Axis, X Axis, X Axis, Y1 Axis, Y2 Axis, Y Axis, Y Axis, Y Axis, Z Axis, Z Axis

## Axis Title

Specifies whether an axis title will be rendered. This property applies only to legacy charts.

## Applies to

Angular Axis, Category Axis, Cumulation Line Axis, Gauge Chart, Map, Numerical Axis, Numerical Axis, Pie Chart, Radial Axis, X Axis, X Axis, Y1 Axis, Y2 Axis, Y Axis, Y Axis, Y Axis, Z Axis, Z Axis

## Axis Title

Specifies whether to show the axis titles in a chart.

## Applies to

Category axis, Cumulation Line, Gauge Axis, Numeric Axis, Pareto Bars, Pie Chart, Primary Axis, Primary Bottom Axis, Secondary Axis, Secondary Bottom Axis, X Axis, Y Axis

## Background Color

Specifies the background color for the object.

## Applies to

3-D Combination Chart, 3-D Scatter Chart, Block, Bubble Chart, Caption, Chart Body, Chart Body, Class, Combination Chart, Conditional Block, Conditional Block, Context Item Text, Crosstab, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Crosstab Summary, Date, Field Set, Gauge Chart, Hyperlink, Hyperlink Button, Image, Legend, List, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, List Summary, Map, Marimekko Chart, Metrics Range Chart, Note Content, Page, Page Body, Page Break Text, Page Footer, Page Header, Page Number, Pareto Chart, Pie Chart, Polar Chart, Progressive Chart, Prompt Button, Radar Chart, Repeater Table, Repeater Table Cell, Row Number, Scatter Chart, Table, Table Cell, Table of Contents, Table of Contents Entry, Table Row, Text Box Prompt, Text Item, Time, Value Prompt, Variable Text Item, Win-Loss Chart

## Background Effects

Specifies the characteristics of a background. You can add background effects only to objects that have a fixed height and width; if a percentage size is given, the effects are ignored.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Block](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Chart Body](#), [Chart Footer](#), [Chart Footer](#), [Chart Subtitle](#), [Chart Subtitle](#), [Chart Title](#), [Chart Title](#), [Class](#), [Combination Chart](#), [Combination Chart](#), [Conditional Block](#), [Conditional Block](#), [Crosstab Columns](#), [Crosstab Columns](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Rows](#), [Crosstab Rows](#), [Crosstab Space](#), [Crosstab Summary](#), [Equation Label](#), [Gauge Chart](#), [Gauge Chart](#), [Legend](#), [List Cell](#), [List Column](#), [List Column Body](#), [List Columns](#), [List Columns Body Style](#), [List Columns Title Style](#), [List Column Title](#), [List Footer](#), [List Header](#), [List Row](#), [List Row Cells Style](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Note](#), [Page Body](#), [Page Footer](#), [Page Header](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Radar Chart](#), [Repeater Table Cell](#), [Scatter Chart](#), [Scatter Chart](#), [Table Cell](#), [Win-Loss Chart](#)

## Background Image

Specifies an image to be used as the background for the object.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Block](#), [Bubble Chart](#), [Caption](#), [Chart Body](#), [Chart Body](#), [Class](#), [Combination Chart](#), [Conditional Block](#), [Conditional Block](#), [Context Item Text](#), [Crosstab](#), [Crosstab Columns](#), [Crosstab Columns](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Rows](#), [Crosstab Rows](#), [Crosstab Space](#), [Crosstab Summary](#), [Field Set](#), [Gauge Chart](#), [Hyperlink](#), [Hyperlink Button](#), [Image](#), [Legend](#), [List](#), [List Cell](#), [List Column](#), [List Column Body](#), [List Columns](#), [List Columns](#), [List Columns Body Style](#), [List Columns Title Style](#), [List Column Title](#), [List Footer](#), [List Header](#), [List Row](#), [List Row Cells Style](#), [List Summary](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Page](#), [Page Body](#), [Page Break Text](#), [Page Footer](#), [Page Header](#), [Pareto Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Prompt Button](#), [Radar Chart](#), [Repeater Table](#), [Repeater Table Cell](#), [Scatter Chart](#), [Table](#), [Table Cell](#), [Table of Contents](#), [Table of Contents Entry](#), [Table Row](#), [Text Item](#), [Variable Text Item](#), [Win-Loss Chart](#)

## Bar, Line, or Area Index

Specifies which combination object to use when calculating the position. When multiple series exist, the index of 0 represents the topmost bar, line, or area in the Series drop zone, the index of 1 represents the second one, and so on.

### Applies to

[Baseline](#), [Baseline](#), [Marker](#)

## Bar Shape

Specifies the shape of the bars in a bar chart.

### Applies to

[Bar](#)

## Bar Width

Specifies the width of the bars as a percentage of the space available. For example, if you specify 50 percent, the bar takes up half of the space available and the bars are separated from each other. If you specify 100 percent, there is no space between the bars. The default value is 80.

### Applies to

[Bar](#), [Pareto Chart](#), [Progressive Chart](#)

## Based on

Specifies the series data item on which the trendline is based.

### Applies to

[Trendline](#)

## Baselines

Adds reference lines to a chart based on numeric or statistical values, calculations, or layout calculations. This property applies only to legacy charts.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Bubble Chart](#), [Combination Chart](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Polar Chart](#), [Progressive Chart](#), [Radar Chart](#), [Scatter Chart](#)

## Bevel

Specifies whether the chart appears with a beveled edge.

### Applies to

[Bar](#), [Bullet Chart](#), [Pareto Chart](#), [Pie Chart](#), [Progressive Chart](#)

## Block Variable

Specifies a variable based on which the block can be conditionally rendered.

### Applies to

[Conditional Block](#), [Conditional Block](#)

## Border

Specifies the width, style, and color for the border of the object.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Block](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Caption](#), [Class](#), [Combination Chart](#), [Combination Chart](#), [Conditional Block](#), [Conditional Block](#), [Context Item Text](#), [Crosstab](#), [Crosstab Columns](#), [Crosstab Columns](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab](#)

[Rows](#), [Crosstab Rows](#), [Crosstab Space](#), [Crosstab Summary](#), [Field Set](#), [Gauge Chart](#), [Gauge Chart](#), [Hyperlink](#), [Hyperlink Button](#), [Image](#), [List](#), [List Cell](#), [List Column](#), [List Column Body](#), [List Columns](#), [List Columns](#), [List Columns Body Style](#), [List Columns Title Style](#), [List Column Title](#), [List Footer](#), [List Header](#), [List Row](#), [List Row Cells Style](#), [List Summary](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Page](#), [Page Body](#), [Page Break Text](#), [Page Footer](#), [Page Header](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Prompt Button](#), [Radar Chart](#), [Repeater Table](#), [Repeater Table Cell](#), [Scatter Chart](#), [Scatter Chart](#), [Table](#), [Table Cell](#), [Table of Contents](#), [Table of Contents Entry](#), [Text Box Prompt](#), [Text Item](#), [Variable Text Item](#), [Win-Loss Chart](#)

## Border Color

Specifies the color of the border rendered around specific chart elements, such as bars, stacks, areas, points, or pie slices.

### Applies to

[3-D Area](#), [3-D Bar](#), [3-D Line](#), [3-D Scatter Chart](#), [Area](#), [Bar](#), [Bubble Chart](#), [Display Layer](#), [Gauge Chart](#), [Legend](#), [Line](#), [Marimekko Chart](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Point Layer](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Radar Chart](#), [Region Layer](#), [Scatter Chart](#), [Win-Loss Chart](#)

## Border Color

Specifies the color of the borders around the chart body in a pie chart.

### Applies to

[Pie Chart](#)

## Borders

Specifies whether borders are rendered around specific chart elements, such as bars, stacks, areas, points, or pie slices.

### Applies to

[3-D Area](#), [3-D Bar](#), [3-D Line](#), [3-D Scatter Chart](#), [Area](#), [Bar](#), [Bubble Chart](#), [Display Layer](#), [Gauge Chart](#), [Legend](#), [Line](#), [Marimekko Chart](#), [Pareto Chart](#), [Pie Chart](#), [Point Layer](#), [Polar Chart](#), [Progressive Chart](#), [Radar Chart](#), [Region Layer](#), [Scatter Chart](#), [Win-Loss Chart](#)

## Borders

Specifies whether borders appear around the chart body.

### Applies to

[Area](#), [Bar](#), [Pareto Chart](#), [Progressive Chart](#)

## Borders

Specifies whether borders appear around the chart body in a pie chart.

**Applies to**[Pie Chart](#)**Bottom Position**

Specifies the position of the bottom edge of the colored region.

**Applies to**[Colored Region](#)**Bottom Position (px)**

Specifies the pixel position of the bottom edge of the note measured from the bottom of the chart.

**Applies to**[Note](#)**Bottom Position (px)**

Specifies the pixel position of the bottom edge of the legend measured from the bottom of the chart.

**Applies to**[Legend](#)**Box Type**

Specifies whether to override the default box type for the object. When Box Type is set to None, the object is not rendered and its space is not reserved in the report. You could use this property to remove an object completely from a report when a specific condition is met. When Box Type is set to Inline, you can insert other objects on the same line as the object. When Box Type is set to Block, you can insert other objects only on the lines above and below the object.

**Applies to**

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Block](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Class](#), [Combination Chart](#), [Combination Chart](#), [Conditional Block](#), [Conditional Block](#), [Context Item](#), [Text](#), [Crosstab](#), [Field Set](#), [Gauge Chart](#), [Gauge Chart](#), [Hyperlink](#), [Image](#), [List](#), [List Summary](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Page Break Text](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Radar Chart](#), [Repeater Table](#), [Scatter Chart](#), [Scatter Chart](#), [Table](#), [Table of Contents](#), [Table of Contents Entry](#), [Text Item](#), [Variable Text Item](#), [Win-Loss Chart](#)

**Box Type**

Specifies whether to override the default box type for the object. When Box Type is set to None, the object is not rendered and its space is not reserved in the report.

### Applies to

Axis Title, Axis Title, Baseline, Baseline, Chart Footer, Chart Footer, Chart Subtitle, Chart Subtitle, Chart Title, Chart Title, Crosstab Columns, Crosstab Columns, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Rows, Crosstab Rows, Crosstab Space, Crosstab Summary, Date & Time Prompt, Date Prompt, Interval Prompt, Legend, Legend Title, Legend Title, List Cell, List Column, List Column Body, List Columns, List Columns, List Columns Body Style, List Columns Title Style, List Column Title, List Footer, List Header, List Row, List Row Cells Style, Marker, Marker, Note Content, Page Body, Page Footer, Page Header, Regression Line, Repeater Table Cell, Select & Search Prompt, Table Cell, Table Row, Text Box Prompt, Time Prompt, Value Prompt

## Bubble Size

Specifies how the bubble size is computed. Minimum-Based assigns the smallest bubble to the minimum data value. Zero-Based computes the bubble size relative to 0. This option is compatible with Excel 2002. Zero-Based with Negatives shows negative bubbles as hollow, and the bubbles get larger as they get further from 0. This option is compatible with Excel 2007.

### Applies to

Bubble Chart

## Bubble Size

## Bubble Size

Specifies how bubble sizes are calculated on a bubble chart.

### Applies to

Bubble Chart

## Bullet Indicators

Specifies the size and shape of the bullet chart data marker that represents the actual value.

### Applies to

Bullet Chart

## Button Definition

Defines the button in the control.

### Applies to

Button

## Button Gap

Specifies the gap between buttons in the button bar.

### Applies to

[Button Bar](#), [Data Button Bar](#), [Data Toggle Button Bar](#), [Toggle Button Bar](#)

## Buttons Definition

Defines the buttons in the button bar.

### Applies to

[Button Bar](#), [Toggle Button Bar](#)

## Button Width

Specifies the width of buttons in the button bar.

### Applies to

[Button Bar](#), [Data Button Bar](#), [Data Toggle Button Bar](#), [Toggle Button Bar](#)

## Calculation Intersection

Specifies whether to suppress calculated values that occur at the intersection of a data source calculated member and a query-defined calculated member. When values are suppressed, the characters specified for the Not Applicable characters data format appear in the cells.

### Applies to

[Calculated Measure](#), [Calculated Member](#), [Data Item](#), [Dimensional Edge Summary](#), [Edge Summary](#)

## Calendar Type

Specifies the type of calendar to show. The date values are mapped to the selected calendar before being formatted. The default value is inherited from the user's content language.

### Applies to

[Date & Time Prompt](#), [Date Prompt](#)

## Caption

Specifies the caption for the level.

### Applies to

[Level](#)

## Caption

Specifies the caption.

**Applies to**

Calculated Member

## Cardinality

Specifies the cardinality for this join operand.

**Applies to**

Query Reference

## Cascade Source

Specifies the parameter whose value is used to filter the values displayed in this control.

**Applies to**

Select & Search Prompt, Tree Prompt, Value Prompt

## Case Insensitive

Specifies whether to perform a case insensitive search by default.

**Applies to**

Select & Search Prompt

## Catalog

Specifies the OLAP catalog.

**Applies to**

MDX

## Category Axis

Specifies whether the axis is rendered.

**Applies to**

Combination Chart, Marimekko Chart, Metrics Range Chart, Pareto Chart, Progressive Chart, Radar Chart

## Category Baselines

Adds reference lines on the category axis of a chart based on numeric or statistical values, calculations, or layout calculations.

**Applies to**

Combination Chart, Pareto Chart, Progressive Chart

## Chart Orientation

Specifies whether the chart is rendered vertically or horizontally.

### Applies to

[Bullet Chart](#), [Combination Chart](#), [Combination Chart](#), [Pareto Chart](#), [Pareto Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Win-Loss Chart](#)

## Chart Type

Specifies whether the data may be rendered as either a bar, line, or area. This property applies only to legacy charts.

### Applies to

[Area](#), [Bar](#), [Line](#)

## Chart Type

Specifies whether the data may be rendered as either a bar, line, or area.

### Applies to

[Area](#), [Bar](#), [Line](#)

## Chart Type

Specifies whether the data may be rendered as either a bar, line or area.

### Applies to

[3-D Area](#), [3-D Bar](#), [3-D Line](#)

## Check Boxes Definition

Defines the check boxes in the check box group.

### Applies to

[Check Box Group](#)

## Choices Deselect All Text

Specifies the text for the link below the choices box that deselects all the items in the box. This property applies to all prompts with either multiple selections and ranges or multiple selections and search. The default link text is Deselect All.

### Applies to

[Date & Time Prompt](#), [Date Prompt](#), [Interval Prompt](#), [Select & Search Prompt](#), [Text Box Prompt](#), [Time Prompt](#), [Value Prompt](#)

## Choices Select All Text

Specifies the text for the link below the choices box that selects all the items in the box. This property applies to all prompts with either multiple selections and ranges or multiple selections and search. The default link text is Select All.

### Applies to

[Date & Time Prompt](#), [Date Prompt](#), [Interval Prompt](#), [Select & Search Prompt](#), [Text Box Prompt](#), [Time Prompt](#), [Value Prompt](#)

## Choices Text

Specifies the title that appears above the choices box when multiple selections are enabled. This property applies to the following prompt types: value, text box, date, date & time, time, interval, and select & search. The default title text is Choices.

### Applies to

[Date & Time Prompt](#), [Date Prompt](#), [Interval Prompt](#), [Select & Search Prompt](#), [Text Box Prompt](#), [Time Prompt](#), [Value Prompt](#)

## Classes

Specifies a class to apply to the object. The class provides a default style. If you apply more than one class, the style properties from all classes are merged together when they are applied. However, if the classes have style properties in common, the style properties from the last class applied override those from previous classes.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Angular Axis](#), [Axis Labels](#), [Axis Title](#), [Axis Title](#), [Block](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Caption](#), [Category Axis](#), [Chart Footer](#), [Chart Footer](#), [Chart Subtitle](#), [Chart Subtitle](#), [Chart Title](#), [Chart Title](#), [Combination Chart](#), [Combination Chart](#), [Conditional Block](#), [Conditional Block](#), [Context Item Text](#), [Crosstab](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Space](#), [Crosstab Summary](#), [Cumulation Line Axis](#), [Date](#), [Field Set](#), [Gauge Chart](#), [Gauge Chart](#), [Gauge Labels](#), [Hyperlink](#), [Hyperlink Button](#), [Image](#), [Legend](#), [Legend](#), [Legend Title](#), [Legend Title](#), [List](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [List Summary](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Note Content](#), [Numerical Axis](#), [Numerical Axis](#), [Page](#), [Page Body](#), [Page Break Text](#), [Page Footer](#), [Page Header](#), [Page Number](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Pie Labels](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Prompt Button](#), [Radar Chart](#), [Radial Axis](#), [Repeater Table](#), [Repeater Table Cell](#), [Row Number](#), [Scatter Chart](#), [Scatter Chart](#), [Table](#), [Table Cell](#), [Table of Contents](#), [Table of Contents Entry](#), [Table Row](#), [Text Item](#), [Time](#), [Variable Text Item](#), [Win-Loss Chart](#), [X Axis](#), [X Axis](#), [Y1 Axis](#), [Y2 Axis](#), [Y Axis](#), [Z Axis](#), [Z Axis](#)

## Clickable Region

### Applies to

[Crosstab Fact Cells](#), [List Column](#), [List Column Body](#), [List Column Title](#), [List Row](#)

## Clickable Region

### Applies to

[Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Space](#), [Crosstab Summary](#)

## Clickable Regions

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Combination Chart](#), [Combination Chart](#), [Crosstab](#), [Gauge Chart](#), [Gauge Chart](#), [List](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Radar Chart](#), [Scatter Chart](#), [Scatter Chart](#)

## Clock Mode

Specifies whether the arms of the clock move.

### Applies to

[Date & Time Prompt](#), [Time Prompt](#)

## Color by Value

Specifies how color by value is rendered in a chart.

### Applies to

[Bubble Chart](#), [Color by Value Measure](#), [Scatter Chart](#)

## Colored Regions

Specifies rectangular colored regions drawn in the chart body to highlight parts of the chart. The regions are rendered behind the data markers. The regions are drawn in the same order that they appear in this property. The first region is drawn first, on the bottom, and the last region is drawn on top of the other regions.

### Applies to

[Bubble Chart](#), [Bullet Chart](#), [Combination Chart](#), [Pareto Chart](#), [Progressive Chart](#), [Scatter Chart](#)

## Color Legend Title

Specifies a title within the legend above the palette for the region color. If this object is not defined, no additional title is drawn. If no legend is drawn, this object is ignored. Styling for this object is inherited from the legend title.

### Applies to

[Region Layer](#)

## Color Legend Title

Specifies a title within the legend above the palette for the point color. If this object is not defined, no additional title is drawn. If no legend is drawn, this object is ignored. Styling for this object is inherited from the legend title.

### Applies to

[Point Layer](#)

## Column Coordinate

Uniquely identifies the column of a node member or spacer on an edge of the crosstab. You cannot modify this value.

### Applies to

[Crosstab Intersection](#)

## Column Titles

Specifies where or whether column titles may be rendered.

### Applies to

[List](#)

## Column Visibility

Specifies whether a column is visible when the report is executed. You can set column visibility based on a condition.

### Applies to

[List Column](#), [List Column Body](#), [List Column Title](#)

## Combinations

Specifies which axes to show and the chart types to use for the series.

### Applies to

[Combination Chart](#)

## Component Reference

Specifies the layout object that is referenced. An object is a reusable component only if it has a name.

### Applies to

[Layout Component Reference](#)

## Component Reference

Specifies the layout object that is referenced. An object is a reusable component only if it has a name. You cannot modify this value.

### Applies to

[Component Override](#)

## Conditional Palette

Specifies a conditional palette for the chart.

### Applies to

[Area](#), [Bar](#), [Bubble Chart](#), [Gauge Axis](#), [Line](#), [Pareto Chart](#), [Pie Chart](#), [Progressive Chart](#), [Scatter Chart](#)

## Conditional Palette

Specifies a conditional palette for the chart. This property applies only to legacy charts.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Bubble Chart](#), [Combination Chart](#), [Gauge Chart](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Pie Chart](#), [Point Layer](#), [Polar Chart](#), [Progressive Chart](#), [Radar Chart](#), [Region Layer](#), [Scatter Chart](#)

## Conditional Styles

Specifies the conditions and styles used to style the object.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Angular Axis](#), [Angular Measure](#), [Axis Labels](#), [Axis Title](#), [Axis Title](#), [Block](#), [Bubble Chart](#), [Bubble Chart](#), [Bubble Measure](#), [Bubble Measure](#), [Bullet Chart](#), [Caption](#), [Category Axis](#), [Chart Body](#), [Chart Footer](#), [Chart Footer](#), [Chart Measure](#), [Chart Subtitle](#), [Chart Subtitle](#), [Chart Text Item](#), [Chart Text Item](#), [Chart Title](#), [Chart Title](#), [Color by Value Measure](#), [Combination Chart](#), [Combination Chart](#), [Conditional Block](#), [Conditional Block](#), [Context Item Text](#), [Crosstab](#), [Crosstab Columns](#), [Crosstab Columns](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Rows](#), [Crosstab Rows](#), [Crosstab Space](#), [Crosstab Summary](#), [Cumulation Label](#), [Cumulation Line Axis](#), [Date & Time Prompt](#), [Date Prompt](#), [Default Measure](#), [Default Measure](#), [Display Layer](#), [Equation Label](#), [Field Set](#), [Gauge Chart](#), [Gauge Chart](#), [Gauge Labels](#), [Gauge Numerical Axis](#), [Hyperlink](#), [Hyperlink Button](#), [Image](#), [Interval Prompt](#), [Label](#), [Legend](#), [Legend](#), [Legend Title](#), [Legend Title](#), [List](#), [List Cell](#), [List Column](#), [List Column Body](#), [List Columns](#), [List Columns](#), [List Columns Body Style](#), [List Columns Title Style](#), [List Column Title](#), [List Footer](#), [List Header](#), [List Row](#), [List Row Cells](#), [Style](#), [List Summary](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Note](#), [Note Content](#), [Numerical Axis](#), [Numerical Axis](#), [Page](#), [Page Body](#), [Page Break Text](#), [Page Footer](#), [Page Header](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Pie Labels](#), [Point Layer](#), [Point Measure](#), [Point Size Measure](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Prompt Button](#), [Radar Chart](#), [Radial Axis](#), [Radial](#)

[Measure](#), [Region Layer](#), [Region Measure](#), [Repeater Table](#), [Repeater Table Cell](#), [Scatter Chart](#), [Scatter Chart](#), [Select & Search Prompt](#), [Table](#), [Table Cell](#), [Table of Contents](#), [Table of Contents Entry](#), [Table Row](#), [Target Measure](#), [Text Box Prompt](#), [Text Item](#), [Time Prompt](#), [Tolerance Measure](#), [Tree Prompt](#), [Trendline Label](#), [Value Prompt](#), [Win-Loss Chart](#), [X Axis](#), [X Axis](#), [X Axis Measure](#), [Y1 Axis](#), [Y2 Axis](#), [Y Axis](#), [Y Axis Measure](#), [Z Axis](#), [Z Axis](#), [Z Axis Measure](#)

## Connecting Lines

Specifies the properties of the lines that connect the segments of a stacked bar. This property is ignored for clustered bars. This property applies only to legacy charts.

### Applies to

[Bar](#), [Pareto Chart](#), [Progressive Chart](#)

## Connecting Lines

Specifies the properties of the lines that connect the segments of a stacked bar in a progressive chart. This property is ignored for clustered bars.

### Applies to

[Progressive Chart](#)

## Connections

### Container Filter

Filters items in the control or container based on the value of a variable.

### Applies to

[Check Box Group](#), [Crosstab](#), [Data Check Box Group](#), [Data Drop-Down List](#), [Data List Box](#), [Data Radio Button Group](#), [Drop-Down List](#), [List](#), [List Box](#), [Radio Button Group](#)

### Container Select

Selects an item in the control or container based on the value of a variable. For example, selects a row in a list, selects a named list item from a drop down list, or selects a named tab from a tab control.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Button Bar](#), [Check Box Group](#), [Combination Chart](#), [Combination Chart](#), [Crosstab](#), [Data Button Bar](#), [Data Check Box Group](#), [Data Deck](#), [Data Drop-Down List](#), [Data List Box](#), [Data Radio Button Group](#), [Data Tab Control](#), [Data Toggle Button Bar](#), [Deck](#), [Drop-Down List](#), [Gauge Chart](#), [Gauge Chart](#), [List](#), [List Box](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Radar Chart](#), [Radio Button Group](#), [Scatter Chart](#), [Scatter Chart](#), [Tab Control](#), [Toggle Button Bar](#)

## Contents Height

Specifies the relative height of list rows. This property is used only when a list has a height defined in the Size and Overflow property.

Stretched means that the rows will be evenly sized to fit in the list's height. This is default HTML behavior.

Minimal means that rows will take up only as much space as they need, and be compressed at the top of the list. You can position a footer at the bottom of the list by setting the Push To Bottom property to Yes on a footer object inside the list.

### Applies to

[List](#)

## Control Enable

Enables the control or container based on the value of a variable.

### Applies to

[Button](#), [Button Bar](#), [Check Box Group](#), [Data Button Bar](#), [Data Check Box Group](#), [Data Drop-Down List](#), [Data List Box](#), [Data Radio Button Group](#), [Data Toggle Button Bar](#), [Drop-Down List](#), [List Box](#), [Radio Button Group](#), [Toggle Button Bar](#)

## Cross Product Allowed

Specifies whether the query will be allowed to run if there is a cross join between database tables. This type of query generates a result set that includes all possible unique combinations of values from the first and second table. The default value is Deny.

### Applies to

[Query](#)

## Cumulation Axis

Specifies whether the axis for the cumulation line is rendered.

### Applies to

[Cumulation Line](#)

## Cumulation Axis

## Cumulation Label

Specifies whether a label for the cumulation line is rendered in the legend.

### Applies to

[Cumulation Line](#)

## Cumulation Label

Specifies the label that is shown with the cumulation line on a Pareto chart

### Applies to

[Cumulation Line](#)

## Cumulative Line

Specifies whether the cumulation line is rendered.

### Applies to

[Pareto Chart](#)

## Cumulative Line

Specifies whether the cumulation line is rendered in a Pareto chart.

### Applies to

[Pareto Chart](#)

## Current Block

Specifies which block is currently being authored.

### Applies to

[Conditional Block](#), [Conditional Block](#)

## Current Card

Specifies which card to show in the report.

### Applies to

[Card](#), [Deck](#)

## Current Tab

Specifies which tab to show in the report.

### Applies to

[Tab Control](#)

## Custom Label

Overrides the default label for the data item.

**Applies to**

[Angular Measure](#), [Bubble Measure](#), [Chart Node Member](#), [Default Measure](#), [Point Measure](#), [Point Size Measure](#), [Radial Measure](#), [Region Measure](#), [Target Measure](#), [Tolerance Measure](#), [Win-Loss Measure](#), [X Axis Measure](#), [Y Axis Measure](#), [Z Axis Measure](#)

**Custom Label**

Specifies the data source and format for a text item, such as a legend item, legend title, axis label, or axis title.

**Applies to**

[Bubble Measure](#), [Bullet Measure](#), [Chart Measure](#), [Color by Value Measure](#), [Default Measure](#), [Target Measure](#)

**Data Format**

Specifies the data format of the object.

**Applies to**

[Angular Axis](#), [Angular Measure](#), [Axis Labels](#), [Bubble Measure](#), [Bubble Measure](#), [Bullet Measure](#), [Chart Measure](#), [Chart Node Member](#), [Chart Text Item](#), [Chart Text Item](#), [Color by Value Measure](#), [Crosstab Columns](#), [Crosstab Columns](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Rows](#), [Crosstab Rows](#), [Crosstab Space](#), [Crosstab Summary](#), [Cumulation Label](#), [Cumulation Line Axis](#), [Date](#), [Default Measure](#), [Default Measure](#), [Gauge Numerical Axis](#), [Label](#), [List Cell](#), [List Column](#), [List Column Body](#), [List Columns](#), [List Columns](#), [List Columns Body Style](#), [List Columns Title Style](#), [List Column Title](#), [List Footer](#), [List Header](#), [List Row](#), [List Row Cells Style](#), [List Summary](#), [Numerical Axis](#), [Numerical Axis](#), [Point Measure](#), [Point Size Measure](#), [Radial Axis](#), [Radial Measure](#), [Region Measure](#), [Row Number](#), [Select & Search Prompt](#), [Target Measure](#), [Target Measure](#), [Text Item](#), [Time](#), [Tolerance Measure](#), [Total Column](#), [Trendline Label](#), [Value Prompt](#), [Win-Loss Measure](#), [X Axis](#), [X Axis Measure](#), [Y1 Axis](#), [Y2 Axis](#), [Y Axis](#), [Y Axis Measure](#), [Z Axis](#), [Z Axis Measure](#)

**Data Item**

Specifies a reference to a data item. You cannot modify this value.

**Applies to**

[Button Icon](#), [Button Label](#), [Button Value](#), [Card Label](#), [Card Value](#), [Check Box Icon](#), [Check Box Label](#), [Check Box Value](#), [Crosstab Node Member](#), [Crosstab Summary](#), [List Item Icon](#), [List Item Label](#), [List Item Value](#), [List Summary](#), [Member Set](#), [Radio Button Icon](#), [Radio Button Label](#), [Radio Button Value](#), [Tab Icon](#), [Tab Label](#), [Tab Value](#)

**Data Item**

Specifies a reference to a data item.

### Applies to

Calculated Member, Fact, Key, Member Property

## Data Item Label

Specifies the data item label that defines the text to render.

### Applies to

Chart Node Member, Chart Text Item, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Cumulation Line Label, Hyperlink, Hyperlink Button, List Cell, List Column Body, List Column Title, Text Item

## Data Item Label

Specifies the data item label that defines the HTML to render.

### Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, HTML Item, List Cell, List Column Body, List Column Title, Rich Text Item

## Data Item Label

Specifies the data item label that defines the URL.

### Applies to

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Hyperlink, Hyperlink Button, Image, List Cell, List Column Body, List Column Title

## Data Item Label

Specifies the data item label that defines the bookmark. The value used as the bookmark reference must match this value.

### Applies to

Bookmark, Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, List Cell, List Column Body, List Column Title

## Data Item Value

Specifies the data item value that defines the text to render.

### Applies to

Chart Node Member, Chart Text Item, Chart Text Item, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space,

[Cumulation Line Label](#), [Hyperlink](#), [Hyperlink Button](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [Text Item](#), [Total Column](#)

## Data Item Value

Specifies the data item value that defines the HTML to render.

### Applies to

[Chart Node Member](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Space](#), [HTML Item](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [Rich Text Item](#)

## Data Item Value

Specifies the data item value that defines the URL.

### Applies to

[Chart Node Member](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Space](#), [Hyperlink](#), [Hyperlink Button](#), [Image](#), [List Cell](#), [List Column Body](#), [List Column Title](#)

## Data Item Value

Specifies the data item value that defines the bookmark. The value used as the bookmark reference must match this value.

### Applies to

[Bookmark](#), [Chart Node Member](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Space](#), [List Cell](#), [List Column Body](#), [List Column Title](#)

## Data Item Value

Specifies the numeric position by using a data item value.

### Applies to

[Baseline](#), [Baseline](#), [Baseline](#), [Baseline](#), [Marker](#), [Marker](#), [Numeric Baseline](#)

## Data Item Value

Specifies the numeric position of a data item.

### Applies to

[Marker](#)

## Data Language

Specifies the language of the data.

**Applies to**

[Map](#)

## Data Points

Specifies whether to show data points on the chart and how they are formatted.

**Applies to**

[Cumulation Line](#)

## Data Source

Specifies the query data source.

**Applies to**

[MDX, SQL](#)

## Days Text

Specifies the title that appears above the days box in interval prompts. The default title text is Days.

**Applies to**

[Interval Prompt](#)

## Deck Cards Definition

The cards in the deck. Each card must have a corresponding row in the static data table.

**Applies to**

[Deck](#)

## Default Card

Card to display when no other card matches the current variable state.

**Applies to**

[Data Deck, Deck](#)

## Default Measure

Specifies the default measure to use for a crosstab or chart. If the measures of the crosstab or chart cannot be determined by what is being rendered on the edges, then the default measure will be rendered.

**Applies to**

[Crosstab](#)

## Default Selections

Specifies the collection of default selections for a prompt control.

### Applies to

[Date & Time Prompt](#), [Date Prompt](#), [Interval Prompt](#), [Text Box Prompt](#), [Time Prompt](#), [Tree Prompt](#), [Value Prompt](#)

## Default Title

Specifies whether the default title is generated.

### Applies to

[Axis Title](#), [Legend Title](#)

## Default Title

Specifies whether the default title may be generated.

### Applies to

[Axis Title](#), [Legend Title](#)

## Define Contents

Overrides the content of the selected crosstab intersection. Use this property to hide measure values for individual cells or to define custom content.

### Applies to

[Crosstab Intersection](#), [Crosstab Member Fact Cells](#)

## Define Member Sets

Specifies the set structure of a query. If it is not defined, it is assumed that each data item defines an unrelated set.

### Applies to

[Query](#)

## Definition

Specifies the expression to evaluate when filtering the data.

### Applies to

[Detail Filter](#), [Summary Filter](#)

## Depth

Specifies the three-dimensional depth effect of the chart. A value of zero indicates a flat chart.

### Applies to

[Combination Chart](#), [Combination Chart](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Progressive Chart](#), [Progressive Chart](#)

## Description

Specifies a description for the object, that is used to assist authoring.

### Applies to

[Class](#), [HTML Item](#), [Metric Studio Diagram](#), [Rich Text Item](#)

## Deselect Text

Specifies the text for the link that deselects the items when the selection is optional. This property applies to the following prompt types: text box, date, date & time, time, interval, value, select & search, and tree. The default link text is Deselect.

### Applies to

[Date & Time Prompt](#), [Date Prompt](#), [Interval Prompt](#), [Select & Search Prompt](#), [Text Box Prompt](#), [Time Prompt](#), [Tree Prompt](#), [Value Prompt](#)

## Detail

Specifies whether the data item is to be used for calculating aggregates or not. When set to Yes, the data item is used to aggregate the lowest level details.

### Applies to

[Calculated Member](#), [Data Item](#), [Explicit Member Set](#), [Hierarchy Set](#), [Level Set](#), [Member](#), [Member Children Set](#), [Named Set](#), [Set Expression](#)

## Diagram Identifier

Identifies an impact or custom diagram in Metric Studio. Copy the identifier from Metric Studio (Diagrams tab, View the Diagram Identifier button in the Actions column) and paste it in this property. Report Studio decodes the identifier into an image URL. The diagram is imported as a static image.

### Applies to

[Metric Studio Diagram](#)

## Dial Face Fill

Specifies the fill color and effects for the dial face of a gauge chart.

### Applies to

[Gauge Chart](#)

## Dial Outline Color

Specifies the dial outline color in a gauge chart.

### Applies to

[Gauge Chart](#)

## Dictionary

Specifies the aliases to use when matching data values to feature names in the map.

### Applies to

[Map](#)

## Dimension

Specifies a reference to a dimension. You cannot modify this value.

### Applies to

[Calculated Measure](#), [Calculated Member](#), [Dimensional Edge Summary](#), [Explicit Member Set](#), [Hierarchy Set](#), [Intersection \(Tuple\)](#), [Level Set](#), [Measure](#), [Member](#), [Member Children Set](#), [Member Property](#), [Named Set](#), [Set Expression](#)

## Display After Overall Header

Specifies whether the list page header is to be rendered after the overall header.

### Applies to

[List Page Header](#)

## Display Frequency

Specifies the frequency for which chart labels are to be rendered. If set to 3, for example, every third label will be rendered. This property applies only to legacy charts.

### Applies to

[Category Axis](#), [X Axis](#), [Y Axis](#)

## Display Milliseconds

Specifies whether to show the milliseconds. The format of the milliseconds can be controlled by selecting a specific format. This property is ignored if seconds are not rendered. The default value is inherited from the user's content language.

### Applies to

[Date & Time Prompt](#), [Interval Prompt](#), [Time Prompt](#)

## Display Seconds

Specifies whether to show the seconds. The format of the seconds can be controlled by selecting a specific format. The default value is inherited from the user's content language.

### Applies to

Date & Time Prompt, Interval Prompt, Time Prompt

## Display Value

Specifies the values rendered to the report user when the prompt is used. These values can be different than the ones that are actually used by the report.

### Applies to

Select & Search Prompt, Value Prompt

## Down

Specifies the number of times down, or rows, that the frame contents may be rendered.

The default value depends on the setting of the Repeater Direction property. If it is set to Left to right, top to bottom, the default is 20. If it is set to Top to bottom, left to right, the default is one.

### Applies to

Repeater Table

## Drill-Through Definitions

Specifies report-to-report drill-through definitions, bookmarks, and drill-through text for the object.

### Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Bubble Chart, Bullet Chart, Category axis, Category Axis, Combination Chart, Combination Chart, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Gauge Chart, Gauge Chart, Gauge Labels, Image, Legend, Legend, List Column Body, List Column Title, Map, Marimekko Chart, Metrics Range Chart, Pareto Chart, Pareto Chart, Pie Chart, Pie Chart, Pie Labels, Polar Chart, Progressive Chart, Progressive Chart, Radar Chart, Scatter Chart, Scatter Chart, Text Item, Win-Loss Chart, X Axis, Y Axis

## Drop Shadow

Defines a drop shadow that is rendered around a container.

### Applies to

Crosstab, List, Repeater Table, Table

## Drop Shadow

Specifies whether a drop shadow appears on a chart and how it appears.

**Applies to**

[Bubble Chart](#), [Bullet Chart](#), [Combination Chart](#), [Gauge Chart](#), [Pareto Chart](#), [Pie Chart](#), [Progressive Chart](#), [Scatter Chart](#)

**Duplicates**

Specifies whether duplicate rows will be preserved or removed.

**Applies to**

[Query Operation](#)

**Embed**

Specifies how to embed the reference object. A reference, or pointer, to the object is stored, by default. Alternatively, a copy of the external object can be stored in the report.

**Applies to**

[Layout Component Reference](#)

**End Position**

Specifies the position of one edge of the colored region along the numeric axis. The region extends from the position defined in the Start Position to the position defined in this property. The value specified in this property must be greater than the value specified in the Start Position property.

**Applies to**

[Colored Region](#)

**Execution Method**

Specifies whether the query is a candidate to run concurrently. If set to Concurrent, the query may still execute sequentially based on other factors. If not explicitly set, the query will execute sequentially. Concurrent query execution may improve performance in some cases.

**Applies to**

[Query](#)

**Execution Optimization**

Specifies how much of the query processing is performed by the client and how much is performed by the database server. If the database server can perform all the query processing, it does.

If All Rows is selected, the optimizer adopts a plan that retrieves all rows of the result set in the least amount of time. This value is generally used in a batch environment.

If First Rows is selected, the optimizer adopts a plan that retrieves the first row as quickly as possible. This value is generally used in an interactive environment.

If Incremental is selected, the optimizer retrieves the first N rows, and then retrieves the next N rows.

#### Applies to

[Query](#)

### Expand Features

Specifies whether to center and expand the feature in the map chart. When set to Yes, the map feature is centered and expanded to take up all available space in the chart. When set to No, the map feature is not expanded.

#### Applies to

[Map](#)

### Exploded Slices

Specifies the slices that appear pulled out of a pie chart and their appearance.

#### Applies to

[Pie Chart](#)

### Expression

Specifies the expression that defines the slicer member set.

#### Applies to

[Slicer Member Set](#)

### Expression

Specifies the numeric position for a data marker in a scatter chart.

#### Applies to

[Marker](#)

### Expression

Specifies the expression used to populate the data item.

#### Applies to

[Angular Measure](#), [Area](#), [Bar](#), [Baseline](#), [Baseline](#), [Baseline](#), [Baseline](#), [Bubble Measure](#), [Bubble Measure](#), [Bullet Measure](#), [Calculated Measure](#), [Calculated Member](#), [Category Baseline](#), [Chart Measure](#), [Chart Node Member](#), [Chart Text Item](#), [Chart Text Item](#), [Color by Value Measure](#), [Colored Region](#), [Colored Region](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Cumulation Line Label](#), [Data Item](#), [Default Measure](#), [Default Measure](#), [Gauge Axis](#), [HTML Item](#), [Image](#), [Line](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [Map Location](#), [Map Location](#), [Map Refinement Location](#), [Map Refinement Location](#), [Marker](#), [Marker](#), [Numeric Baseline](#), [Point Measure](#), [Point Size Measure](#), [Radial Measure](#), [Region Measure](#), [Rich Text Item](#), [Scatter Marker](#), [Set Expression](#), [Target Measure](#), [Target Measure](#), [Text Item](#), [Tolerance Measure](#), [Total Column](#), [Win-Loss Measure](#), [X Axis Measure](#), [Y Axis Measure](#), [Z Axis Measure](#)

## Extend Width

Specifies whether the area extends to the width of the chart. This property applies only to defined areas of area charts. When the area chart is converted to a matrix of charts, this property is not supported.

### Applies to

[Area](#)

## Face Color

Specifies the color to show on the face of each gauge in a gauge chart.

### Applies to

[Gauge Chart](#)

## Fact Cells Precedence

Specifies which style property will override the other style property for intersecting cells in a crosstab, the row's properties, or the column's properties.

This property only applies to style properties that are both set, but to different values. For example, if the row's background color is set to yellow and the column's background color is set to red, you can select which of these properties will override the other. If only the row or the column has a set background color, then that color will be used on the intersecting cell, regardless of this setting.

### Applies to

[Crosstab](#)

## Fill Effects

Specifies the fill effects for the object.

### Applies to

[Chart Body](#), [Display Layer](#), [Legend](#), [Point Layer](#), [Region Layer](#)

## First Column Color

Specifies the color, gradient, or pattern to be used for the first column in the progressive chart.

### Applies to

[Progressive Chart](#)

## First Date

Specifies the earliest date to render in the control, and the earliest date that can be selected.

### Applies to

[Date & Time Prompt](#), [Date Prompt](#)

## First Label Index

Specifies which label will be rendered first. If set to 5, for example, the fifth label will be the first label rendered. Subsequent labels will be rendered as defined in the Display Frequency property. This property applies only to legacy charts.

### Applies to

[Category Axis](#), [X Axis](#), [Y Axis](#)

## First Slice Angle

Specifies the angle at which the first pie slice begins in a pie chart.

### Applies to

[Pie Chart](#)

## Floating

Specifies how objects flow around an object.

Float controls the way the content that follows the selected object will flow around or below it.

Clear controls where the selected object is positioned, relative to other floating objects.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Block](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Class](#), [Combination Chart](#), [Combination Chart](#), [Conditional Block](#), [Conditional Block](#), [Context Item Text](#), [Crosstab](#), [Date & Time Prompt](#), [Date Prompt](#), [Field Set](#), [Gauge Chart](#), [Gauge Chart](#), [Hyperlink](#), [Hyperlink Button](#), [Image](#), [Interval Prompt](#), [List](#), [List Summary](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Page Break Text](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Prompt Button](#), [Radar Chart](#), [Repeater Table](#), [Scatter Chart](#), [Scatter Chart](#), [Select & Search Prompt](#), [Table](#), [Table of Contents](#), [Table of Contents Entry](#), [Text Box Prompt](#), [Text Item](#), [Time Prompt](#), [Tree Prompt](#), [Value Prompt](#), [Variable Text Item](#), [Win-Loss Chart](#)

## Font

Specifies the font family, size, weight, style, and effects used to display the object's text.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Angular Axis](#), [Axis Labels](#), [Axis Title](#), [Axis Title](#), [Block](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Caption](#), [Category Axis](#), [Chart Body](#), [Chart Body](#), [Chart Footer](#), [Chart Footer](#), [Chart Subtitle](#), [Chart Subtitle](#), [Chart Title](#), [Chart Title](#), [Class](#), [Combination Chart](#), [Combination Chart](#), [Conditional Block](#), [Conditional Block](#), [Context Item Text](#), [Crosstab](#), [Crosstab Columns](#), [Crosstab Columns](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Rows](#), [Crosstab Rows](#), [Crosstab Space](#), [Crosstab Summary](#), [Cumulation Line Axis](#), [Date](#), [Equation Label](#), [Field Set](#), [Gauge Chart](#), [Gauge Chart](#), [Gauge Labels](#), [Gauge Numerical Axis](#), [Hyperlink](#), [Legend](#), [Legend](#), [Legend Title](#), [Legend Title](#), [List](#), [List Cell](#), [List Column](#), [List Column Body](#), [List Columns](#), [List](#)

[Columns Body Style](#), [List Columns Title Style](#), [List Column Title](#), [List Footer](#), [List Header](#), [List Row](#), [List Row Cells Style](#), [List Summary](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Note](#), [Note Content](#), [Numerical Axis](#), [Numerical Axis](#), [Page](#), [Page Body](#), [Page Break Text](#), [Page Footer](#), [Page Header](#), [Page Number](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Pie Labels](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Prompt Button](#), [Radar Chart](#), [Radial Axis](#), [Repeater Table](#), [Repeater Table Cell](#), [Row Number](#), [Scatter Chart](#), [Scatter Chart](#), [Table](#), [Table Cell](#), [Table of Contents](#), [Table of Contents Entry](#), [Table Row](#), [Text Box Prompt](#), [Text Item](#), [Time](#), [Value Prompt](#), [Variable Text Item](#), [Win-Loss Chart](#), [X Axis](#), [X Axis](#), [Y1 Axis](#), [Y2 Axis](#), [Y Axis](#), [Y Axis](#), [Z Axis](#), [Z Axis](#)

## Font Auto-Sizing

Specifies whether to allow automatic resizing of the font.

### Applies to

[Legend](#)

## Footer

Specifies whether a chart footer is rendered. This property applies only to legacy charts.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Bubble Chart](#), [Combination Chart](#), [Gauge Chart](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Radar Chart](#), [Scatter Chart](#), [Win-Loss Chart](#)

## Footer

Specifies whether a chart footer is rendered.

### Applies to

[Bubble Chart](#), [Bullet Chart](#), [Combination Chart](#), [Gauge Chart](#), [Pareto Chart](#), [Pie Chart](#), [Progressive Chart](#), [Scatter Chart](#)

## Foreground Color

Specifies the color of the object's text.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Angular Axis](#), [Axis Labels](#), [Axis Title](#), [Axis Title](#), [Block](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Caption](#), [Category Axis](#), [Chart Body](#), [Chart Body](#), [Chart Footer](#), [Chart Footer](#), [Chart Subtitle](#), [Chart Subtitle](#), [Chart Title](#), [Chart Title](#), [Class](#), [Combination Chart](#), [Combination Chart](#), [Conditional Block](#), [Conditional Block](#), [Context Item Text](#), [Crosstab](#), [Crosstab Columns](#), [Crosstab Columns](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Rows](#), [Crosstab Rows](#), [Crosstab Space](#), [Crosstab Summary](#), [Cumulation Line Axis](#), [Date](#), [Equation Label](#), [Field Set](#), [Gauge Chart](#), [Gauge Chart](#), [Gauge Labels](#), [Gauge Numerical Axis](#), [Hyperlink](#), [Hyperlink Button](#), [Legend](#), [Legend](#)

[Legend Title](#), [Legend Title](#), [List](#), [List Cell](#), [List Column](#), [List Column Body](#), [List Columns](#), [List Columns](#), [List Columns Body Style](#), [List Columns Title Style](#), [List Column Title](#), [List Footer](#), [List Header](#), [List Row](#), [List Row Cells Style](#), [List Summary](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Note](#), [Note Content](#), [Numerical Axis](#), [Numerical Axis](#), [Page](#), [Page Body](#), [Page Break Text](#), [Page Footer](#), [Page Header](#), [Page Number](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Pie Labels](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Prompt Button](#), [Radar Chart](#), [Radial Axis](#), [Repeater Table](#), [Repeater Table Cell](#), [Row Number](#), [Scatter Chart](#), [Scatter Chart](#), [Table](#), [Table Cell](#), [Table of Contents](#), [Table of Contents Entry](#), [Table Row](#), [Text Box Prompt](#), [Text Item](#), [Time](#), [Value Prompt](#), [Variable Text Item](#), [Win-Loss Chart](#), [X Axis](#), [X Axis](#), [Y1 Axis](#), [Y2 Axis](#), [Y Axis](#), [Y Axis](#), [Z Axis](#), [Z Axis](#)

## From Text

Specifies the label that appears beside the beginning of a range. This property applies to the following prompt types: date, date & time, time, and interval. The default label text is From.

### Applies to

[Date & Time Prompt](#), [Date Prompt](#), [Interval Prompt](#), [Text Box Prompt](#), [Time Prompt](#), [Value Prompt](#)

## Gauge Axes

Specifies the start and end angles for the gauge axes.

### Applies to

[Gauge Chart](#)

## Gauge Axis Colors

Specifies the colors of the gauge axis.

### Applies to

[Gauge Axis](#)

## Gauge Axis Inner Radius

Specifies the inner radius of the gauge axis as a percentage of the maximum allowed.

### Applies to

[Gauge Axis](#)

## Gauge Axis Outer Radius

Specifies the outer radius of the gauge axis as a percentage of the maximum allowed.

### Applies to

[Gauge Axis](#)

## Gauge Border

Specifies whether borders appear around the chart body in a gauge chart.

### Applies to

[Gauge Chart](#)

## Gauge Labels

Specifies whether gauge labels are rendered.

### Applies to

[Gauge Chart](#)

## Gauge Needle

Specifies the size and style of the gauge needle. The chart contains one needle for each row in the data series. All needles on a gauge axis use the same specified styles.

### Applies to

[Gauge Axis](#)

## Gauge Palette

Specifies the palette that controls the look of the dial portion of a gauge.

### Applies to

[Gauge Chart](#)

## Gauge Pivot

Specifies whether to show the center pivot point and its color, size, and style.

### Applies to

[Gauge Chart](#)

## Generated SQL

Shows the generated SQL or MDX.

### Applies to

[Query](#)

## Gradient

Specifies a color gradient for the page.

### Applies to

[Page](#)

## Gridlines

Specifies the properties of the major gridlines in the gauge axis of a gauge chart.

### Applies to

Gauge Axis

## Gridlines

Specifies the properties of the gridlines in a chart.

### Applies to

Angular Axis, Category Axis, Cumulation Line Axis, Gauge Numerical Axis, Numerical Axis, Numerical Axis, Radial Axis, X Axis, X Axis, Y1 Axis, Y2 Axis, Y Axis, Y Axis, Z Axis, Z Axis

## Gridlines

Specifies the properties of the major gridlines in a chart.

### Applies to

Category axis, Cumulation Line, Numeric Axis, Pareto Bars, Primary Axis, Primary Bottom Axis, Secondary Axis, Secondary Bottom Axis, X Axis, Y Axis

## Grouping & Sorting

Specifies the grouping and sorting structure.

### Applies to

List, Page Set, Repeater, Repeater Table

## Grouping Type

Specifies whether the absolute, stacked, or 100 percent stacked data may be drawn.

### Applies to

Area, Bar, Line

## Group Span

Specifies the group that this cell should visually span.

### Applies to

List Column Body

## Has Fact Cells

Specifies the contents of the fact cells of the crosstab. There is only one fact cell definition for the crosstab, regardless of the number of measures.

**Applies to**[Crosstab](#)

## Header Text

Specifies the title that appears above the list of choices in a value prompt. The default title text is the name of the level above the data items that are listed as choices; for example, Regions.

**Applies to**[Value Prompt](#)

## Heading Level

Specifies the heading level of the table of contents entry.

**Applies to**[Table of Contents Entry](#)

## Height (px)

Specifies the height of the note, in pixels.

**Applies to**[Note](#)

## Hide Adornments

Specifies whether to hide the asterisk (\*) on required prompts and arrow (->) on type-in prompts that are in an error state.

**Applies to**

[Date & Time Prompt](#), [Date Prompt](#), [Generated Prompt](#), [Interval Prompt](#), [Select & Search Prompt](#), [Text Box Prompt](#), [Time Prompt](#), [Tree Prompt](#), [Value Prompt](#)

## Hide Text

Specifies whether to replace characters entered in the prompt control with asterisk (\*) characters.

**Applies to**[Text Box Prompt](#)

## Hierarchy

Specifies a reference to a hierarchy. You cannot modify this value.

**Applies to**

[Angular Measure](#), [Area](#), [Bar](#), [Baseline](#), [Baseline](#), [Baseline](#), [Bubble Measure](#), [Bubble Measure](#), [Bullet Measure](#), [Calculated Member](#), [Category Baseline](#), [Chart Measure](#), [Chart Node Member](#),

[Chart Text Item](#), [Chart Text Item](#), [Color by Value Measure](#), [Colored Region](#), [Colored Region](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Cumulation Line Label](#), [Default Measure](#), [Default Measure](#), [Dimensional Edge Summary](#), [Explicit Member Set](#), [Gauge Axis](#), [Hierarchy Set](#), [HTML Item](#), [Image](#), [Intersection \(Tuple\)](#), [Level Set](#), [Line](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [Map Location](#), [Map Location](#), [Map Refinement Location](#), [Map Refinement Location](#), [Marker](#), [Marker](#), [Member](#), [Member Children Set](#), [Member Property](#), [Named Set](#), [Numeric Baseline](#), [Point Measure](#), [Point Size Measure](#), [Radial Measure](#), [Region Measure](#), [Rich Text Item](#), [Scatter Marker](#), [Set Expression](#), [Target Measure](#), [Target Measure](#), [Text Item](#), [Tolerance Measure](#), [Total Column](#), [Win-Loss Measure](#), [X Axis Measure](#), [Y Axis Measure](#), [Z Axis Measure](#)

## Hierarchy Unique Name

Specifies the Hierarchy Unique Name (HUN) of the hierarchy to which the data item belongs.

### Applies to

[Angular Measure](#), [Area](#), [Bar](#), [Baseline](#), [Baseline](#), [Baseline](#), [Bubble Measure](#), [Bubble Measure](#), [Bullet Measure](#), [Category Baseline](#), [Chart Measure](#), [Chart Node Member](#), [Chart Text Item](#), [Chart Text Item](#), [Color by Value Measure](#), [Colored Region](#), [Colored Region](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Cumulation Line Label](#), [Default Measure](#), [Default Measure](#), [Gauge Axis](#), [Hierarchy Set](#), [HTML Item](#), [Image](#), [Line](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [Map Location](#), [Map Location](#), [Map Refinement Location](#), [Map Refinement Location](#), [Marker](#), [Marker](#), [Numeric Baseline](#), [Point Measure](#), [Point Size Measure](#), [Radial Measure](#), [Region Measure](#), [Rich Text Item](#), [Scatter Marker](#), [Target Measure](#), [Target Measure](#), [Text Item](#), [Tolerance Measure](#), [Total Column](#), [Win-Loss Measure](#), [X Axis Measure](#), [Y Axis Measure](#), [Z Axis Measure](#)

## Highest Value Text

Specifies the label that appears beside the highest value option when ranges are enabled. This property applies to the following prompt types: date, date & time, time, value, text box, and interval. The default label text is Latest date, Latest time, or Highest interval.

### Applies to

[Date & Time Prompt](#), [Date Prompt](#), [Interval Prompt](#), [Text Box Prompt](#), [Time Prompt](#), [Value Prompt](#)

## Hole Size (%)

Specifies the size of the hole in a donut chart. A value of zero indicates a pie chart.

### Applies to

[Pie Chart](#), [Pie Chart](#)

## Horizontal Alignment

Specifies how the contents of the selected object is aligned horizontally.

**Applies to**

[Block](#), [Category Axis](#), [Chart Body](#), [Chart Body](#), [Chart Footer](#), [Chart Footer](#), [Chart Subtitle](#), [Chart Subtitle](#), [Chart Title](#), [Chart Title](#), [Class](#), [Conditional Block](#), [Conditional Block](#), [Crosstab Columns](#), [Crosstab Columns](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Rows](#), [Crosstab Rows](#), [Crosstab Space](#), [Crosstab Summary](#), [Equation Label](#), [Field Set](#), [Gauge Labels](#), [Legend](#), [Legend](#), [Legend Title](#), [Legend Title](#), [List](#), [List Cell](#), [List Column](#), [List Column Body](#), [List Columns](#), [List Columns](#), [List Columns](#), [List Columns Body Style](#), [List Columns Title Style](#), [List Column Title](#), [List Footer](#), [List Header](#), [List Row](#), [List Row Cells Style](#), [Note](#), [Note Content](#), [Page](#), [Page Body](#), [Page Footer](#), [Page Header](#), [Pie Labels](#), [Repeater Table](#), [Repeater Table Cell](#), [Table](#), [Table Cell](#), [Table of Contents](#), [Table of Contents Entry](#), [Table Row](#), [X Axis](#), [Y Axis](#)

## Hours Text

Specifies the title that appears above the hours box in interval prompts. The default title text is Hrs.

**Applies to**

[Interval Prompt](#)

## HTML

Specifies the static text used as HTML.

**Applies to**

[Chart Node Member](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Space](#), [HTML Item](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [Rich Text Item](#)

## HTML Source Variable

Specifies a variable based on which the HTML source is chosen.

**Applies to**

[Chart Node Member](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Space](#), [HTML Item](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [Rich Text Item](#)

## Icon

Specifies the image used to identify the object in the control.

**Applies to**

[Button](#), [Button](#), [Check Box](#), [List Item](#), [Radio Button](#), [Tab](#)

## Ignore Data with No Features

Specifies whether to allow data that has no corresponding features. When set to Yes, data with no corresponding features will be ignored. When set to No, the map will not run if it contains data with no corresponding features.

### Applies to

[Map](#)

## Include Zero For Auto Scale

Specifies whether the value zero is included in the automatic calculation of the numeric scale. This property applies only to legacy charts.

### Applies to

[Angular Axis](#), [Gauge Numerical Axis](#), [Numerical Axis](#), [Numerical Axis](#), [Radial Axis](#), [X Axis](#), [Y1 Axis](#), [Y2 Axis](#), [Y Axis](#), [Z Axis](#), [Z Axis](#)

## Insert Text

Specifies the label that appears on the button that is used to add items to the selected items box in all multiple selection prompts. The default label text is Insert.

### Applies to

[Date & Time Prompt](#), [Date Prompt](#), [Interval Prompt](#), [Select & Search Prompt](#), [Text Box Prompt](#), [Time Prompt](#), [Value Prompt](#)

## Join Relationships

Specifies how to join the two queries.

### Applies to

[Join](#)

## Keywords Text

Specifies the title that appears above the keyword search box in select & search prompts. The default title text is Keywords.

### Applies to

[Select & Search Prompt](#)

## Label

Specifies the static text that defines the bookmark. The value used as the bookmark reference must match this value.

**Applies to**

[Bookmark](#), [Chart Node Member](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Space](#), [List Cell](#), [List Column Body](#), [List Column Title](#)

**Label**

Specifies the class label for a local class.

**Applies to**

[Class](#)

**Label**

Specifies the class label for a global class. You cannot modify this label.

**Applies to**

[Class](#)

**Label**

Specifies the name of the no value list item.

**Applies to**

[No Value List Item](#)

**Label**

Specifies the name of the object in the control.

**Applies to**

[Button](#), [Button](#), [Check Box](#), [List Item](#), [Radio Button](#), [Tab](#)

**Label**

Specifies whether a label is rendered for the baseline.

**Applies to**

[Baseline](#), [Baseline](#)

**Label**

Specifies whether a label is rendered for the marker.

**Applies to**

[Marker](#), [Marker](#)

## Label

Specifies the label of the object.

### Applies to

Angular Measure, Area, Bar, Baseline, Baseline, Baseline, Bubble Measure, Bubble Measure, Bullet Measure, Calculated Measure, Calculated Member, Category Baseline, Chart Measure, Chart Node Member, Chart Text Item, Chart Text Item, Color by Value Measure, Colored Region, Colored Region, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Cumulation Line Label, Data Item, Default Measure, Default Measure, Dimensional Edge Summary, Edge Summary, Explicit Member Set, Gauge Axis, Hierarchy Set, HTML Item, Image, Intersection (Tuple), Level Set, Line, List Cell, List Column Body, List Column Title, List Summary, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Marker, Marker, Measure, Member, Member Children Set, Member Property, Named Set, Numeric Baseline, Point Measure, Point Size Measure, Radial Measure, Region Measure, Rich Text Item, Scatter Marker, Set Expression, Target Measure, Target Measure, Text Item, Tolerance Measure, Total Column, Win-Loss Measure, X Axis Measure, Y Axis Measure, Z Axis Measure

## Label Control

Controls how the labels in a chart are rendered.

### Applies to

Category Axis, X Axis, Y Axis

## Label Location

Specifies whether to show the label in the legend or in the chart.

### Applies to

Marker, Scatter Marker

## Labels

Specifies whether labels are rendered in the chart.

### Applies to

Display Layer, Point Layer, Region Layer

## Labels

Specifies whether labels are rendered.

### Applies to

Pie Chart

## Last Date

Specifies the latest date rendered in the control, and the last date that can be selected.

### Applies to

[Date & Time Prompt](#), [Date Prompt](#)

## Left Position

Specifies the position of the left edge of the colored region.

### Applies to

[Colored Region](#)

## Left Position (px)

Specifies the pixel position of the bottom edge of the note measured from the left edge of the chart.

The position is measured from the left edge of the note.

### Applies to

[Note](#)

## Left Position (px)

Specifies the pixel position of the left edge of the legend measured from the left edge of the chart.

### Applies to

[Legend](#)

## Legend

Specifies whether the legend is rendered.

### Applies to

[3-D Scatter Chart](#), [Bubble Chart](#), [Combination Chart](#), [Gauge Chart](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Pie Chart](#), [Polar Chart](#), [Radar Chart](#), [Scatter Chart](#)

## Legend

Specifies whether a legend is rendered and where the legend is positioned in a chart.

### Applies to

[Bubble Chart](#), [Bullet Chart](#), [Combination Chart](#), [Gauge Chart](#), [Pareto Chart](#), [Pie Chart](#), [Progressive Chart](#), [Scatter Chart](#)

## Legend Label

Specifies whether to render the baseline in the legend.

### Applies to

Baseline, Baseline

## Legend Separator

Specifies the separator to use between the legend entry and the value when you show values in the legend. The default is a comma (,) followed by a space.

### Applies to

Legend

## Legend Title

Specifies whether a legend title is rendered. This property applies only to legacy charts.

### Applies to

Legend

## Legend Title

Specifies whether a legend title is rendered in a chart.

### Applies to

Legend, Legend

## Level

Specifies a reference to a level. You cannot modify this value.

### Applies to

Angular Measure, Area, Bar, Baseline, Baseline, Baseline, Baseline, Bubble Measure, Bubble Measure, Bullet Measure, Category Baseline, Chart Measure, Chart Node Member, Chart Text Item, Chart Text Item, Color by Value Measure, Colored Region, Colored Region, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Cumulation Line Label, Default Measure, Default Measure, Gauge Axis, HTML Item, Image, Level Set, Line, List Cell, List Column Body, List Column Title, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Marker, Marker, Member Property, Numeric Baseline, Point Measure, Point Size Measure, Radial Measure, Region Measure, Rich Text Item, Scatter Marker, Target Measure, Target Measure, Text Item, Tolerance Measure, Total Column, Win-Loss Measure, X Axis Measure, Y Axis Measure, Z Axis Measure

## Level Indentation

Controls the indentation of the contents of node members in a crosstab.

### Applies to

Crosstab Node Member

## Level Unique Name

Specifies the Level Unique Name (LUN) of the level to which the data item belongs.

### Applies to

[Angular Measure](#), [Area](#), [Bar](#), [Baseline](#), [Baseline](#), [Baseline](#), [Bubble Measure](#), [Bubble Measure](#), [Bullet Measure](#), [Category Baseline](#), [Chart Measure](#), [Chart Node Member](#), [Chart Text Item](#), [Chart Text Item](#), [Color by Value Measure](#), [Colored Region](#), [Colored Region](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Cumulation Line Label](#), [Default Measure](#), [Default Measure](#), [Gauge Axis](#), [HTML Item](#), [Image](#), [Level Set](#), [Line](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [Map Location](#), [Map Location](#), [Map Refinement Location](#), [Map Refinement Location](#), [Marker](#), [Marker](#), [Numeric Baseline](#), [Point Measure](#), [Point Size Measure](#), [Radial Measure](#), [Region Measure](#), [Rich Text Item](#), [Scatter Marker](#), [Target Measure](#), [Target Measure](#), [Text Item](#), [Tolerance Measure](#), [Total Column](#), [Win-Loss Measure](#), [X Axis Measure](#), [Y Axis Measure](#), [Z Axis Measure](#)

## Line and Markers

Specifies whether to display a line and whether to display markers.

### Applies to

[Line](#)

## Line Shape

Specifies the shape of a line in a line chart.

### Applies to

[Line](#)

## Line Style

Specifies the style, color, and weight of the line in a Pareto chart.

### Applies to

[Cumulation Line](#)

## Line Style

Specifies the style, color, and weight of the line.

### Applies to

[Category Baseline](#), [Numeric Baseline](#), [Trendline](#)

## Line Style

Specifies the style, color, and weight of the line. This property applies only to legacy charts.

**Applies to**

Line

## Line Styles

Specifies the style, color, and weight of the line.

**Applies to**

Baseline, Baseline, Baseline, Baseline, Cumulation Line, Regression Line

## Line Type

Specifies the type of line used to connect data values.

**Applies to**

Line

## Line Weight (pt)

Specifies the line thickness in points. A value of zero indicates the thinnest possible line.

**Applies to**

Line

## List Items Definition

Defines the list items in the control.

**Applies to**

Drop-Down List, List Box

## Loss Color

Specifies a color, color gradient, or pattern to apply to the loss values.

**Applies to**

Win-Loss Chart

## Lowest Value Text

Specifies the label that appears beside the lowest value option when ranges are enabled. This property applies to the following prompt types: date, date & time, time, value, text box, and interval. The default label text is Earliest date, Earliest time, or Lowest interval.

**Applies to**

Date & Time Prompt, Date Prompt, Interval Prompt, Text Box Prompt, Time Prompt, Value Prompt

## Map & Layers

Sets the map and layers for a map chart.

### Applies to

[Map](#)

## Map Drills

Controls the drill definitions in a map. It is possible to define different drill targets for each region or point.

### Applies to

[Point Layer](#), [Region Layer](#)

## Margin

Specifies the margin properties for the object.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Block](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Caption](#), [Class](#), [Combination Chart](#), [Combination Chart](#), [Conditional Block](#), [Conditional Block](#), [Context Item Text](#), [Crosstab](#), [Date](#), [Field Set](#), [Gauge Chart](#), [Gauge Chart](#), [Hyperlink](#), [Hyperlink Button](#), [Image](#), [List](#), [List Summary](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Page](#), [Page Break Text](#), [Page Number](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Prompt Button](#), [Radar Chart](#), [Repeater Table](#), [Row Number](#), [Scatter Chart](#), [Scatter Chart](#), [Table](#), [Table of Contents](#), [Table of Contents Entry](#), [Text Item](#), [Time](#), [Variable Text Item](#), [Win-Loss Chart](#)

## Marimekko Totals

Specifies whether to show totals for each column at the top of the chart.

### Applies to

[Marimekko Chart](#)

## Marker Color

Specifies a color for the target value markers in a metrics chart.

### Applies to

[Metrics Range Chart](#)

## Marker Color

Specifies the color of the markers.

#### Applies to

[Marker](#), [Marker](#)

### Marker Label

Specifies whether the label for the Target Marker will appear in the legend.

#### Applies to

[Metrics Range Chart](#)

### Markers

Adds reference points to a chart based on numeric or statistical values, calculations, or layout calculations. This property applies only to legacy charts.

#### Applies to

[Bubble Chart](#), [Combination Chart](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Polar Chart](#), [Progressive Chart](#), [Radar Chart](#), [Scatter Chart](#)

### Markers

Adds reference points to a chart based on numeric or statistical values, calculations, or layout calculations.

#### Applies to

[Bubble Chart](#), [Combination Chart](#), [Pareto Chart](#), [Progressive Chart](#), [Scatter Chart](#)

### Marker Shape

Specifies the shape of the markers.

#### Applies to

[Cumulation Line](#), [Marker](#), [Marker](#)

### Marker Shape

Specifies the shape of the marker symbol.

#### Applies to

[Marker](#), [Scatter Marker](#)

### Marker Size (pt)

Specifies the size of markers in points. A value of zero means do not show markers.

#### Applies to

[3-D Scatter Chart](#), [Cumulation Line](#), [Marker](#), [Marker](#), [Polar Chart](#), [Radar Chart](#), [Scatter Chart](#), [Scatter Marker](#)

## Marker Text Location

Specifies where the text of the marker is rendered.

### Applies to

[Bubble Chart](#), [Combination Chart](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Polar Chart](#), [Progressive Chart](#), [Radar Chart](#), [Scatter Chart](#)

## Master Detail Relationships

Specifies relationships between the master data container and the detail data container. Specifically, specifies how query items in the master query are linked to query items or parameters in the detail query.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Axis Title](#), [Axis Title](#), [Baseline](#), [Baseline](#), [Baseline](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Category Baseline](#), [Chart Footer](#), [Chart Footer](#), [Chart Subtitle](#), [Chart Subtitle](#), [Chart Title](#), [Chart Title](#), [Colored Region](#), [Colored Region](#), [Combination Chart](#), [Combination Chart](#), [Crosstab](#), [Data Button Bar](#), [Data Check Box Group](#), [Data Deck](#), [Data Drop-Down List](#), [Data List Box](#), [Data Radio Button Group](#), [Data Tab Control](#), [Data Toggle Button Bar](#), [Gauge Chart](#), [Gauge Chart](#), [Label](#), [Legend Title](#), [Legend Title](#), [List](#), [Map](#), [Marimekko Chart](#), [Marker](#), [Marker](#), [Metrics Range Chart](#), [Note](#), [Note Content](#), [Numeric Baseline](#), [Page](#), [Page Set](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Point Color Legend Title](#), [Point Size Legend Title](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Radar Chart](#), [Region Color Legend Title](#), [Repeater](#), [Repeater Table](#), [Scatter Chart](#), [Scatter Chart](#), [Scatter Marker](#), [Win-Loss Chart](#)

## Material Effects

Specifies a material effect, such as plastic or metallic.

### Applies to

[Bubble Chart](#), [Combination Chart](#), [Pareto Chart](#), [Progressive Chart](#), [Scatter Chart](#)

## Material Effects

Specifies a material effect, such as plastic or metallic, in a pie chart.

### Applies to

[Pie Chart](#)

## Matrix rows and columns

Specifies whether to render a matrix of charts in rows and columns.

### Applies to

[Bubble Chart](#), [Bullet Chart](#), [Combination Chart](#), [Gauge Chart](#), [Pareto Chart](#), [Pie Chart](#), [Progressive Chart](#), [Scatter Chart](#)

## Maximum Characters

Specifies the maximum number of characters to show before the text is truncated.

### Applies to

[Legend](#), [Text Item](#)

## Maximum Execution Time

Specifies the maximum period, in seconds, that the query can spend to open the database cursor and to retrieve the first row of data. An error is returned if the specified time is exceeded. Note that this property is not for the total time required to execute the query. If no value is specified, no error is returned and the query runs until complete.

### Applies to

[Query](#)

## Maximum Hotspots

Specifies the maximum number of hotspots generated in a chart. If specified, this value overrides the hotspot configuration settings in IBM® Cognos® Administration.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Combination Chart](#), [Combination Chart](#), [Gauge Chart](#), [Gauge Chart](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Radar Chart](#), [Scatter Chart](#), [Scatter Chart](#), [Win-Loss Chart](#)

## Maximum Rows Retrieved

Specifies the maximum number of database rows that the query can retrieve. An error is returned if the number of database rows returned exceeds the specified value. If no value is specified, no error is returned and the query returns all rows.

### Applies to

[Query](#)

## Maximum Size (pt)

Specifies the maximum size used for map point features that have corresponding data. For example, if the minimum size is 2pt and the maximum size is 12pt, the size of each point is calculated using linear interpolation that is based on its measure value.

### Applies to

[Point Size Measure](#)

## Maximum Tables

Specifies the maximum number of tables that the query can retrieve. An error is returned if the number of tables in the generated Cognos SQL exceeds the specified value. If no value is specified, no error is returned and there is no restriction on the number of tables that can be queried.

### Applies to

[Query](#)

## Maximum Text Blob Characters

Specifies the maximum number of characters that the query is allowed to retrieve for each text BLOB. An error is returned if the number of characters retrieved exceeds the specified value. If no value is specified, no error is returned and text BLOBs can be of any size.

### Applies to

[Query](#)

## Maximum Truncation Characters

Specifies the maximum number of characters to show before the label is truncated.

If the Allow Truncation property is set to Yes and no value is specified in this property, the application will automatically determine the optimum number of characters after which to truncate. Use this property only if you want explicit control over the truncation level. Note that regardless of this property's setting, no truncation will occur if there is sufficient space.

### Applies to

[Category Axis](#), [Gauge Labels](#), [Pie Labels](#), [X Axis](#), [Y Axis](#)

## Maximum Value

Specifies the maximum value for the numeric scale. If no value is specified, one will be calculated based on the data.

### Applies to

[Angular Axis](#), [Gauge Numerical Axis](#), [Numerical Axis](#), [Numerical Axis](#), [Radial Axis](#), [X Axis](#), [Y1 Axis](#), [Y2 Axis](#), [Y Axis](#), [Z Axis](#), [Z Axis](#)

## MDX

The text of the typed-in MDX. It is assumed to be appropriate for the type and data source. If it is not the query may fail or produce unexpected results.

### Applies to

[MDX](#)

## Member

Specifies a reference to a member. You cannot modify this value.

### Applies to

[Angular Measure](#), [Area](#), [Bar](#), [Baseline](#), [Baseline](#), [Baseline](#), [Bubble Measure](#), [Bubble Measure](#), [Bullet Measure](#), [Category Baseline](#), [Chart Measure](#), [Chart Node Member](#), [Chart Text Item](#), [Chart Text Item](#), [Color by Value Measure](#), [Colored Region](#), [Colored Region](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Cumulation Line Label](#), [Default Measure](#), [Default Measure](#), [Gauge Axis](#), [HTML Item](#), [Image](#), [Line](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [Map Location](#), [Map Location](#), [Map Refinement Location](#), [Map Refinement Location](#), [Marker](#), [Marker](#), [Measure](#), [Member](#), [Member Children Set](#), [Numeric Baseline](#), [Point Measure](#), [Point Size Measure](#), [Radial Measure](#), [Region Measure](#), [Rich Text Item](#), [Scatter Marker](#), [Target Measure](#), [Target Measure](#), [Text Item](#), [Tolerance Measure](#), [Total Column](#), [Win-Loss Measure](#), [X Axis Measure](#), [Y Axis Measure](#), [Z Axis Measure](#)

## Member Offset (%)

Specifies a position relative to the next item in the chart. This is a percentage value. Negative values indicate before the member and positive values indicate after the member.

### Applies to

[Baseline](#), [Category Baseline](#), [Marker](#)

## Members

Specifies the members of the intersection (tuple).

### Applies to

[Intersection \(Tuple\)](#)

## Members

Specifies the members in the member set.

### Applies to

[Explicit Member Set](#)

## Member Unique Name

Specifies the Member Unique Name (MUN) of the member.

### Applies to

[Angular Measure](#), [Area](#), [Bar](#), [Baseline](#), [Baseline](#), [Baseline](#), [Baseline](#), [Bubble Measure](#), [Bubble Measure](#), [Bullet Measure](#), [Category Baseline](#), [Chart Measure](#), [Chart Node Member](#), [Chart Text Item](#), [Chart Text Item](#), [Color by Value Measure](#), [Colored Region](#), [Colored Region](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Cumulation Line Label](#), [Default Measure](#), [Default Measure](#), [Gauge Axis](#), [HTML Item](#), [Image](#), [Line](#), [List Cell](#),

[List Column Body](#), [List Column Title](#), [Map Location](#), [Map Location](#), [Map Refinement Location](#), [Map Refinement Location](#), [Marker](#), [Marker](#), [Measure](#), [Member](#), [Member Children Set](#), [Numeric Baseline](#), [Point Measure](#), [Point Size Measure](#), [Radial Measure](#), [Region Measure](#), [Rich Text Item](#), [Scatter Marker](#), [Target Measure](#), [Target Measure](#), [Text Item](#), [Tolerance Measure](#), [Total Column](#), [Win-Loss Measure](#), [X Axis Measure](#), [Y Axis Measure](#), [Z Axis Measure](#)

## Milliseconds Text

Specifies the title that appears above the milliseconds box in interval prompts. The default title text is ms.

### Applies to

[Interval Prompt](#)

## Minimum Size (pt)

Specifies the minimum size used for map point features that have corresponding data. For example, if the minimum size is 2pt and the maximum size is 12pt, the size of each point is calculated using linear interpolation that is based on its measure value.

### Applies to

[Point Size Measure](#)

## Minimum Value

Specifies the minimum value for the numeric scale. If no value is specified, one will be calculated based on the data.

### Applies to

[Angular Axis](#), [Gauge Numerical Axis](#), [Numerical Axis](#), [Numerical Axis](#), [Radial Axis](#), [X Axis](#), [Y1 Axis](#), [Y2 Axis](#), [Y Axis](#), [Z Axis](#), [Z Axis](#)

## Minor Gridlines

Specifies the properties of the minor gridlines in the gauge axis of a gauge chart.

### Applies to

[Gauge Axis](#)

## Minor Gridlines

Specifies the properties of the minor gridlines in a chart. This property applies only to legacy charts.

### Applies to

[Angular Axis](#), [Category Axis](#), [Cumulation Line Axis](#), [Gauge Numerical Axis](#), [Numerical Axis](#), [Numerical Axis](#), [Radial Axis](#), [X Axis](#), [X Axis](#), [Y1 Axis](#), [Y2 Axis](#), [Y Axis](#), [Y Axis](#), [Z Axis](#), [Z Axis](#)

## Minor Gridlines

Specifies the properties of the minor gridlines in a chart.

### Applies to

[Category axis](#), [Cumulation Line](#), [Numeric Axis](#), [Pareto Bars](#), [Primary Axis](#), [Primary Bottom Axis](#), [Secondary Axis](#), [Secondary Bottom Axis](#), [X Axis](#), [Y Axis](#)

## Minutes Text

Specifies the title that appears above the minutes box in interval prompts. The default title text is Mins.

### Applies to

[Interval Prompt](#)

## Multi-Line

Specifies whether to allow multi-line editing in the text control

### Applies to

[Text Box Prompt](#)

## Multi-Select

Specifies whether the control allows the selection of multiple values. Note that an associated parameterized filter expression will override this object's setting. If you edit this property but do not get the expected results, check the associated expression for the presence of an operator that specifies how many items can be selected. Examples of multiple selection operators are "in" and "not in"; examples of single selection operators are equal (=), less than (<) and greater than (>).

For example, if you used the prompt wizard to create a parameterized filter expression and selected one of the operators that specify selection rules, changing the value of this property is not sufficient to change this restriction. You must also edit the filter's expression to remove the offending operator.

To edit a filter expression, you must select the filter, which is accessible from the Query view, using the Explorer bar.

### Applies to

[Data List Box](#), [Date & Time Prompt](#), [Date Prompt](#), [Interval Prompt](#), [List Box](#), [Select & Search Prompt](#), [Text Box Prompt](#), [Time Prompt](#), [Tree Prompt](#), [Value Prompt](#)

## Name

Specifies the name of the object.

### Applies to

[Angular Measure](#), [Area](#), [Bar](#), [Baseline](#), [Baseline](#), [Baseline](#), [Bubble Measure](#), [Bubble Measure](#), [Bullet Measure](#), [Calculated Measure](#), [Calculated Member](#), [Category Baseline](#), [Chart Measure](#), [Chart](#)

[Node Member](#), [Chart Text Item](#), [Chart Text Item](#), [Color by Value Measure](#), [Colored Region](#), [Colored Region](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Cumulation Line Label](#), [Data Item](#), [Default Measure](#), [Default Measure](#), [Dimension](#), [Dimensional Edge Summary](#), [Edge Summary](#), [Explicit Member Set](#), [Fact](#), [Gauge Axis](#), [Hierarchy Set](#), [HTML Item](#), [Image](#), [Intersection \(Tuple\)](#), [Level](#), [Level Set](#), [Line](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [List Summary](#), [Map Location](#), [Map Location](#), [Map Refinement Location](#), [Map Refinement Location](#), [Marker](#), [Marker](#), [Measure](#), [Member](#), [Member Children Set](#), [Member Property](#), [Member Set](#), [Named Set](#), [Numeric Baseline](#), [Page](#), [Page Set](#), [Point Measure](#), [Point Size Measure](#), [Query](#), [Radial Measure](#), [Region Measure](#), [Rich Text Item](#), [Scatter Marker](#), [Set Expression](#), [Target Measure](#), [Target Measure](#), [Text Item](#), [Tolerance Measure](#), [Total Column](#), [Variable](#), [Win-Loss Measure](#), [X Axis Measure](#), [Y Axis Measure](#), [Z Axis Measure](#)

## Name

Specifies a unique name that allows layout objects to be reused, usually to take advantage of any applied styling.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Block](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Button](#), [Button Bar](#), [Check Box Group](#), [Combination Chart](#), [Combination Chart](#), [Crosstab](#), [Data Button Bar](#), [Data Check Box Group](#), [Data Deck](#), [Data Drop-Down List](#), [Data List Box](#), [Data Radio Button Group](#), [Data Tab Control](#), [Data Toggle Button Bar](#), [Date & Time Prompt](#), [Date Prompt](#), [Deck](#), [Drop-Down List](#), [Field Set](#), [Gauge Chart](#), [Gauge Chart](#), [Generated Prompt](#), [HTML Item](#), [Hyperlink](#), [Hyperlink Button](#), [Image](#), [Interval Prompt](#), [List](#), [List Box](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Prompt Button](#), [Radar Chart](#), [Radio Button Group](#), [Repeater](#), [Repeater Table](#), [Rich Text Item](#), [Scatter Chart](#), [Scatter Chart](#), [Select & Search Prompt](#), [Singleton](#), [Tab Control](#), [Table](#), [Text Box Prompt](#), [Text Item](#), [Time Prompt](#), [Toggle Button Bar](#), [Tree Prompt](#), [Value Prompt](#), [Win-Loss Chart](#)

## Name

Specifies the unique name identifier for a query operation.

### Applies to

[Query Operation](#)

## Name

Specifies the unique name identifier for an SQL object.

### Applies to

[SQL](#)

## Name

Specifies the unique name identifier for an MDX object.

**Applies to**

MDX

## Name

Specifies a unique name that allows layout objects to be reused, usually to take advantage of any applied formatting.

**Applies to**

Calculated Member, Key, Level Hierarchy, Member Hierarchy, Member Property

## Negative Column Color

Specifies the color, gradient, or pattern to be used for columns in the progressive chart that represent negative values.

**Applies to**

Progressive Chart

## Nested Label Display

Specifies how to display the axis labels for nested categories. Concatenated separates the labels with a comma.

**Applies to**

Axis Labels

## No Data Contents

Specifies whether to show the no data contents tab for the selected query frame. When set to Yes, you can specify on this tab what to show when there is no data. When set to No, the tab is hidden and the query frame reverts to the default behavior.

**Applies to**

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Bubble Chart, Bullet Chart, Combination Chart, Combination Chart, Crosstab, Gauge Chart, Gauge Chart, List, Map, Marimekko Chart, Metrics Range Chart, Pareto Chart, Pareto Chart, Pie Chart, Pie Chart, Polar Chart, Progressive Chart, Progressive Chart, Radar Chart, Repeater, Repeater Table, Scatter Chart, Scatter Chart, Table of Contents, Win-Loss Chart

## No Data Features Size (pt)

Specifies the point size used for map point features that do not have corresponding data.

**Applies to**

Map

## Node Coordinate

Uniquely identifies a node member or spacer on an edge of a crosstab. Used by the Crosstab Intersection object to uniquely identify the intersection of elements from each edge. You cannot modify this value.

### Applies to

[Crosstab Node Member](#), [Crosstab Space](#), [Crosstab Summary](#)

## Note Border

Specifies the properties for the border of a note.

### Applies to

[Note](#)

## Notes

Specifies a block of text that you can position on a chart.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Bubble Chart](#), [Combination Chart](#), [Gauge Chart](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Radar Chart](#), [Scatter Chart](#)

## Notes

Specifies whether a note is rendered in a chart.

### Applies to

[Bubble Chart](#), [Bullet Chart](#), [Combination Chart](#), [Gauge Chart](#), [Pareto Chart](#), [Pie Chart](#), [Progressive Chart](#), [Scatter Chart](#)

## Note Text

Specifies the text that appears in a note within a chart. To edit the note text, you must switch to Page Design view (View > Page Design)

### Applies to

[Note](#), [Note](#)

## No Value List Item

Specifies whether to define a default list item. When the default item is selected, sets the value of the data item in the control to null. By default, the no value item is the first item in the list.

### Applies to

[Data Drop-Down List](#), [Data List Box](#), [Drop-Down List](#), [List Box](#)

## Number of Regression Lines

Specifies whether there will be one regression line for all the data or one for each series.

### Applies to

[Regression Line](#)

## Numbers Only

Specifies whether the Text Box Prompt allows numbers only.

### Applies to

[Text Box Prompt](#)

## Number Style

Specifies the style to use for page numbers.

### Applies to

[Page Number](#)

## Numerical Axis

Specifies whether the axis is rendered.

### Applies to

[3-D Combination Chart](#)

## Numerical Axis

Specifies whether the numeric axis of a gauge chart is rendered.

### Applies to

[Gauge Chart](#)

## Numeric Baselines

Adds reference lines on the numeric axis of a chart based on numeric or statistical values, calculations, or layout calculations.

### Applies to

[Bubble Chart](#), [Combination Chart](#), [Pareto Chart](#), [Progressive Chart](#), [Scatter Chart](#)

## Numeric Value

Specifies the numeric position by using a number.

### Applies to

[Baseline](#), [Baseline](#), [Baseline](#), [Baseline](#), [Marker](#)

## Numeric Value

Specifies the value of the numeric position.

### Applies to

[Marker](#)

## Options Text

Specifies the text for the additional prompt options link in select & search prompts. The default label text is Options.

### Applies to

[Select & Search Prompt](#)

## Orientation

Specifies the orientation of the control.

### Applies to

[Button Bar](#), [Check Box Group](#), [Data Button Bar](#), [Data Check Box Group](#), [Data Radio Button Group](#), [Data Toggle Button Bar](#), [Radio Button Group](#), [Toggle Button Bar](#)

## Outer Join Allowed

Specifies whether outer joins are allowed on the object. This property applies to a single query and overrides the setting in Framework Manager, the modeling tool.

### Applies to

[Query](#)

## Output Aggregation Method

Specifies the aggregation method.

### Applies to

[Crosstab Summary](#), [List Cell](#), [List Summary](#)

## Override Dimension Info

Specifies dimension information for a query.

### Applies to

[Query](#)

## Overrides

Specifies whether to override child objects. Before you can override child objects, they must have a name.

### Applies to

Layout Component Reference

## Padding

Specifies the space between the object and the margin. If there is a border then it specifies the space between the object and the border.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Block](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Caption](#), [Class](#), [Combination Chart](#), [Combination Chart](#), [Conditional Block](#), [Conditional Block](#), [Context Item Text](#), [Crosstab Columns](#), [Crosstab Columns](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Rows](#), [Crosstab Rows](#), [Crosstab Space](#), [Crosstab Summary](#), [Gauge Chart](#), [Gauge Chart](#), [Hyperlink](#), [Hyperlink Button](#), [List Cell](#), [List Column](#), [List Column Body](#), [List Columns](#), [List Columns](#), [List Columns Body Style](#), [List Columns Title Style](#), [List Column Title](#), [List Footer](#), [List Header](#), [List Row](#), [List Row Cells Style](#), [List Summary](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Page Body](#), [Page Break Text](#), [Page Footer](#), [Page Header](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Prompt Button](#), [Radar Chart](#), [Repeater Table Cell](#), [Scatter Chart](#), [Scatter Chart](#), [Table Cell](#), [Table of Contents](#), [Table of Contents Entry](#), [Text Item](#), [Variable Text Item](#), [Win-Loss Chart](#)

## Pagination

Specifies pagination rules, such as page breaks, keep-with properties, page counts, and numbering.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Combination Chart](#), [Combination Chart](#), [Crosstab](#), [Crosstab Node Member](#), [Crosstab Space](#), [Gauge Chart](#), [Gauge Chart](#), [List](#), [List Column](#), [List Header](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Page](#), [Page Set](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Radar Chart](#), [Repeater](#), [Repeater Table](#), [Scatter Chart](#), [Scatter Chart](#), [Table](#), [Win-Loss Chart](#)

## Palette

Specifies the palette to use for the chart. This property applies only to legacy charts.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Bubble Chart](#), [Combination Chart](#), [Gauge Chart](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Radar Chart](#), [Scatter Chart](#)

## Palette

Specifies the color palette for a chart.

**Applies to**

[Area](#), [Bar](#), [Bubble Chart](#), [Gauge Axis](#), [Line](#), [Pareto Chart](#), [Pie Chart](#), [Scatter Chart](#)

## Palette

Map layers use the numeric palette, in which the color of a region or point is based on its numeric value.

**Applies to**

[Point Layer](#), [Region Layer](#)

## Parameter

Specifies the parameter that is satisfied by values chosen in the prompt control.

**Applies to**

[Date & Time Prompt](#), [Date Prompt](#), [Generated Prompt](#), [Interval Prompt](#), [Select & Search Prompt](#), [Text Box Prompt](#), [Time Prompt](#), [Tree Prompt](#), [Value Prompt](#)

## Pareto Axis

Specifies whether the axis is rendered.

**Applies to**

[Pareto Chart](#)

## PDF Page Setup

Specifies the orientation and paper size for PDF documents.

**Applies to**

[Page](#)

## Percentile

Specifies a position based on a data percentile value. This value must be greater than zero.

**Applies to**

[Baseline](#), [Baseline](#), [Baseline](#), [Baseline](#), [Marker](#), [Marker](#)

## Percent of Axis

Specifies a position based on a percentage along the numeric axis. This value must be greater than zero.

**Applies to**

[Baseline](#), [Baseline](#), [Baseline](#), [Baseline](#), [Marker](#), [Marker](#)

## Performance Pattern

Controls what portions of the range markers for tolerance and target ranges are rendered on a metrics chart.

### Applies to

[Metrics Range Chart](#)

## Pie Labels

Specifies whether pie labels are rendered.

### Applies to

[Pie Chart](#)

## Plot Area Fill

Specifies the fill color and effects for the plot area of a chart.

### Applies to

[Bubble Chart](#), [Combination Chart](#), [Pareto Chart](#), [Progressive Chart](#), [Scatter Chart](#)

## Point Shape

Specifies the shape of the markers. If you choose a value of series or category, the marker shape varies accordingly.

### Applies to

[3-D Scatter Chart](#), [Bubble Chart](#), [Polar Chart](#), [Radar Chart](#), [Scatter Chart](#)

## Polynomial Exponent

Specifies the highest exponential value to use in the regression calculation.

### Applies to

[Regression Line](#)

## Position

Specifies where to position the legend.

### Applies to

[Legend](#)

## Position

Specifies the position of a note in a chart.

**Applies to**[Equation Label](#), [Note](#)

## Position in List

Specifies the position of the no list value item in the list.

**Applies to**[No Value List Item](#)

## Position type

Specifies the source type for the numeric position.

**Applies to**[Baseline](#), [Baseline](#), [Baseline](#), [Baseline](#), [Marker](#)

## Position type

Specifies the type of numeric position on the X-axis. This property applies only to legacy charts.

**Applies to**[Marker](#)

## Position type

Specifies a type of numeric position.

**Applies to**[Marker](#)

## Positive Column Color

Specifies the color, gradient, or pattern to be used for columns in the progressive chart that represent positive values.

**Applies to**[Progressive Chart](#)

## Pre-populate

Specifies whether to pre-populate the control with values, but only if the parent of this prompt control is optional. This only applies to prompt controls that have a parent in a cascade.

**Applies to**[Tree Prompt](#), [Value Prompt](#)

## Pre-populate Levels

Specifies the number of levels to pre-populate the prompt with. The default value is 1, which will pre-populate the prompt with only the root members.

### Applies to

[Tree Prompt](#)

## Pre-Sort

Sorts the data that is used by the query to produce a temporary cube when needed, such as for a crosstab against a relational data source. This property affects the default order of members of a level populated from the data item.

### Applies to

[Data Item](#)

## Processing

Specifies whether the query engine will pick up a minimal amount of processing. Local processing only occurs if the database cannot handle the load. This property applies only to dimensional, or OLAP, data sources.

### Applies to

[Query](#)

## Progressive Axis

Specifies whether the axis is rendered.

### Applies to

[Progressive Chart](#)

## Progressive Palette

Specifies the color palette in a progressive chart.

### Applies to

[Progressive Chart](#)

## Projection List

Shows the list of projected data items for the set operation. You can automatically generate the list or manually add data items.

### Applies to

[Query Operation](#)

## Properties

Specifies a list of data items from the query in scope to associate to the selected object. This is necessary when you want to reference a data item that is in the associated query, but not used in the layout. For more information, see the Report Studio Professional Authoring User Guide.

### Applies to

[Axis Title](#), [Axis Title](#), [Baseline](#), [Baseline](#), [Baseline](#), [Baseline](#), [Category Baseline](#), [Chart Footer](#), [Chart Footer](#), [Chart Node Member](#), [Chart Subtitle](#), [Chart Subtitle](#), [Chart Title](#), [Chart Title](#), [Colored Region](#), [Colored Region](#), [Crosstab Node Member](#), [Cumulation Label](#), [Cumulation Line](#), [Label](#), [Legend Title](#), [Legend Title](#), [List](#), [List Footer](#), [List Header](#), [Marker](#), [Marker](#), [Marker](#), [Note](#), [Note Content](#), [Numeric Baseline](#), [Page](#), [Page Set](#), [Point Color Legend Title](#), [Point Size Legend Title](#), [Region Color Legend Title](#), [Regression Line](#), [Repeater](#), [Repeater Table](#), [Scatter Marker](#), [Select & Search Prompt](#), [Singleton](#), [Total Column](#), [Total Column Label](#), [Trendline Label](#), [Value Prompt](#)

## Property

Specifies a reference to a member property. You cannot modify this value.

### Applies to

[Member Property](#)

## Property Unique Name

Specifies the Member Property Unique Name (MPUN) of the member property.

### Applies to

[Angular Measure](#), [Area](#), [Bar](#), [Baseline](#), [Baseline](#), [Baseline](#), [Bubble Measure](#), [Bubble Measure](#), [Bullet Measure](#), [Category Baseline](#), [Chart Measure](#), [Chart Node Member](#), [Chart Text Item](#), [Chart Text Item](#), [Color by Value Measure](#), [Colored Region](#), [Colored Region](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Cumulation Line Label](#), [Default Measure](#), [Default Measure](#), [Gauge Axis](#), [HTML Item](#), [Image](#), [Line](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [Map Location](#), [Map Location](#), [Map Refinement Location](#), [Map Refinement Location](#), [Marker](#), [Marker](#), [Member Property](#), [Numeric Baseline](#), [Point Measure](#), [Point Size Measure](#), [Radial Measure](#), [Region Measure](#), [Rich Text Item](#), [Scatter Marker](#), [Target Measure](#), [Target Measure](#), [Text Item](#), [Tolerance Measure](#), [Total Column](#), [Win-Loss Measure](#), [X Axis Measure](#), [Y Axis Measure](#), [Z Axis Measure](#)

## Push To Bottom

Specifies whether to position the footer as low as possible inside the parent object.

### Applies to

[List Footer](#), [List Page Footer](#)

## Query

Specifies a reference to a query.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Axis Title](#), [Axis Title](#), [Baseline](#), [Baseline](#), [Baseline](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Category Baseline](#), [Chart Footer](#), [Chart Footer](#), [Chart Subtitle](#), [Chart Subtitle](#), [Chart Title](#), [Chart Title](#), [Colored Region](#), [Colored Region](#), [Combination Chart](#), [Combination Chart](#), [Crosstab](#), [Cumulation Label](#), [Data Button Bar](#), [Data Check Box Group](#), [Data Deck](#), [Data Drop-Down List](#), [Data List Box](#), [Data Radio Button Group](#), [Data Tab Control](#), [Data Toggle Button Bar](#), [Gauge Chart](#), [Gauge Chart](#), [Label](#), [Legend Title](#), [Legend Title](#), [List](#), [Map](#), [Marimekko Chart](#), [Marker](#), [Marker](#), [Metrics Range Chart](#), [Note Content](#), [Numeric Baseline](#), [Page](#), [Page Set](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Point Color Legend Title](#), [Point Size Legend Title](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Radar Chart](#), [Region Color Legend Title](#), [Repeater](#), [Repeater Table](#), [Scatter Chart](#), [Scatter Chart](#), [Scatter Marker](#), [Select & Search Prompt](#), [Singleton](#), [Tree Prompt](#), [Trendline Label](#), [Value Prompt](#), [Win-Loss Chart](#)

## Radar Type

Specifies how the radar chart is rendered.

### Applies to

[Radar Chart](#)

## Radial Axis

Specifies whether the axis is rendered.

### Applies to

[Polar Chart](#), [Radar Chart](#)

## Radio Buttons Definition

Defines the buttons in the radio button group.

### Applies to

[Radio Button Group](#)

## Range

Specifies whether this control accepts ranges. The setting of the associated parameterized expression for this property will override the setting of this object. If you edit this property but do not get the expected results, check the associated expression for the presence or absence of an `in_range` operator.

For example, if you created this prompt control with the prompt wizard and set up an associated parameterized filter that accepts ranges, changing the value of this property is not sufficient to change this restriction. You must also edit the filter's expression and remove the `in_range` operator.

To edit a filter expression, you must select the filter, which is accessible from the Query view, using the Explorer bar.

**Applies to**[Date & Time Prompt](#), [Date Prompt](#), [Interval Prompt](#), [Text Box Prompt](#), [Time Prompt](#), [Value Prompt](#)

## Range Label

Specifies whether the label for the Target Range will appear in the legend.

**Applies to**[Metrics Range Chart](#)

## Regression Line

Specifies whether a regression line is rendered. A regression line is a straight or curved line that best approximates the data points in the series.

**Applies to**[Bubble Chart](#), [Scatter Chart](#)

## Regression Type

Specifies the type of regression used.

**Applies to**[Regression Line](#)

## Relative Alignment

Specifies how to vertically align this object, relative to its siblings.

**Applies to**[3-D Combination Chart](#), [3-D Scatter Chart](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Class](#), [Combination Chart](#), [Combination Chart](#), [Context Item Text](#), [Crosstab](#), [Date](#), [Date Prompt](#), [Field Set](#), [Gauge Chart](#), [Gauge Chart](#), [Hyperlink](#), [Hyperlink Button](#), [Image](#), [List](#), [List Summary](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Page Break Text](#), [Page Number](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Prompt Button](#), [Radar Chart](#), [Repeater Table](#), [Row Number](#), [Scatter Chart](#), [Scatter Chart](#), [Table](#), [Text Box Prompt](#), [Text Item](#), [Time](#), [Value Prompt](#), [Variable Text Item](#), [Win-Loss Chart](#)

## Remove Text

Specifies the label that appears on the button that is used to remove items from the selected items box in all multiple selection prompts. The default label text is Remove.

**Applies to**[Date & Time Prompt](#), [Date Prompt](#), [Interval Prompt](#), [Select & Search Prompt](#), [Text Box Prompt](#), [Time Prompt](#), [Value Prompt](#)

## Render

For list reports, specifies whether to render the column in the report output. Whether set to Yes or No, the query for the column is always executed.

### Applies to

List Column

## Render Fact Cells

Specifies whether to render values in the fact cells of the spacer on an edge of the crosstab.

### Applies to

Crosstab Space

## Render Page when Empty

Specifies whether to render a page when data containers on a page contain no data. If all data containers on a page have this property set to No and do not have any data to render, the page is not rendered. If any data container on a page has data or has this property set to Yes, the page is rendered.

### Applies to

3-D Combination Chart, 3-D Scatter Chart, Bubble Chart, Bubble Chart, Bullet Chart, Combination Chart, Combination Chart, Crosstab, Gauge Chart, Gauge Chart, List, Map, Marimekko Chart, Metrics Range Chart, Pareto Chart, Pareto Chart, Pie Chart, Pie Chart, Polar Chart, Progressive Chart, Progressive Chart, Radar Chart, Repeater, Repeater Table, Scatter Chart, Scatter Chart, Table of Contents, Win-Loss Chart

## Render Variable

Specifies a variable based on which the object can be conditionally rendered.

### Applies to

3-D Combination Chart, 3-D Scatter Chart, Axis Labels, Axis Title, Block, Bubble Chart, Bubble Chart, Bullet Chart, Button, Button Bar, Category Baseline, Chart Footer, Chart Subtitle, Chart Text Item, Chart Title, Check Box Group, Colored Region, Colored Region, Combination Chart, Combination Chart, Context Item Text, Crosstab, Data Button Bar, Data Check Box Group, Data Deck, Data Drop-Down List, Data List Box, Data Radio Button Group, Data Tab Control, Data Toggle Button Bar, Date & Time Prompt, Date Prompt, Deck, Drop-Down List, Field Set, Gauge Chart, Gauge Chart, Generated Prompt, HTML Item, Hyperlink, Hyperlink Button, Image, Interval Prompt, Label, Legend, Legend Title, List, List Box, List Column, List Summary, Map, Marimekko Chart, Marker, Metrics Range Chart, Note, Numeric Baseline, Page, Page Break Text, Pareto Chart, Pareto Chart, Pie Chart, Pie Chart, Polar Chart, Progressive Chart, Progressive Chart, Radar Chart, Radio Button Group, Repeater, Repeater Table, Repeater Table Cell, Rich Text Item, Scatter Chart, Scatter Chart, Scatter Marker, Select & Search Prompt, Singleton, Tab Control, Table, Table of

[Contents](#), [Table of Contents Entry](#), [Text Box Prompt](#), [Text Item](#), [Time Prompt](#), [Toggle Button Bar](#), [Tree Prompt](#), [Trendline](#), [Value Prompt](#), [Variable Text Item](#), [Win-Loss Chart](#)

## Repeater Direction

Specifies the direction in which to populate the rendered repeater cells.

### Applies to

[Repeater Table](#)

## Report Expression

Specifies the report expression that defines the text to render.

### Applies to

[Chart Node Member](#), [Chart Text Item](#), [Chart Text Item](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Space](#), [Cumulation Line Label](#), [Hyperlink](#), [Hyperlink Button](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [Text Item](#), [Total Column](#)

## Report Expression

Specifies the report expression used to define the HTML to render.

### Applies to

[Chart Node Member](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Space](#), [HTML Item](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [Rich Text Item](#)

## Report Expression

Specifies the report expression that defines the URL.

### Applies to

[Chart Node Member](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Space](#), [Hyperlink](#), [Hyperlink Button](#), [Image](#), [List Cell](#), [List Column Body](#), [List Column Title](#)

## Report Expression

Specifies the report expression that defines the bookmark. The value used as the bookmark reference must match this value.

### Applies to

[Bookmark](#), [Chart Node Member](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Space](#), [List Cell](#), [List Column Body](#), [List Column Title](#)

## Report Expression

Specifies the numeric position by using a report expression.

### Applies to

[Baseline](#), [Baseline](#), [Baseline](#), [Baseline](#), [Marker](#)

## Report Expression

Specifies the report expression for the member position.

### Applies to

[Baseline](#), [Marker](#)

## Report Expression

Specifies a report expression.

### Applies to

[As of Time Expression](#)

## Report Expression

Specifies the expression to evaluate when determining the value for this variable.

### Applies to

[Variable](#)

## Report Expression

Specifies the numeric position from a report expression.

### Applies to

[Marker](#)

## Required

Specifies whether the prompt is required or optional. If this property is set to required, the prompt must have a value entered before the report can be run. The Usage setting of the associated parameterized filter for this property will override the setting of this object. If you edit this property, but do not get the expected results, verify the Usage setting of the associated filter.

For example, if you created this prompt control with the prompt wizard and set the associated parameterized filter to be optional, changing the value of this property is not sufficient to change this setting. You must also edit the filter's Required property to match the setting for this object's Required property.

To edit a filter expression, you must select the filter, which is accessible from the Query view, using the Explorer bar.

**Applies to**

[Date & Time Prompt](#), [Date Prompt](#), [Generated Prompt](#), [Interval Prompt](#), [Select & Search Prompt](#), [Text Box Prompt](#), [Time Prompt](#), [Tree Prompt](#), [Value Prompt](#)

**Results Deselect All Text**

Specifies the text for the link below the results box that deselects all the items in the box. This property applies to all prompts with multiple selections and search, tree prompts, and value prompts. The default link text is Deselect All.

**Applies to**

[Select & Search Prompt](#), [Tree Prompt](#), [Value Prompt](#)

**Results Select All Text**

Specifies the text for the link below the results box that selects all the items in the box. This property applies to all prompts with multiple selections and search, tree prompts, and value prompts. The default link text is Select All.

**Applies to**

[Select & Search Prompt](#), [Tree Prompt](#), [Value Prompt](#)

**Results Text**

Specifies the title that appears above the results box in select & search prompts. The default title text is Results.

**Applies to**

[Select & Search Prompt](#)

**Reverse Category order**

Specifies whether to change the order of the categories, such as the bars in a bar chart. The default is No, which means that for a horizontal bar chart, bars start from the bottom to the top.

**Applies to**

[Category axis](#)

**Right Position**

Specifies the position of the right edge of the colored region.

**Applies to**

[Colored Region](#)

**Right Position (px)**

Specifies the pixel position of the right edge of the legend measured from the left edge of the chart.

**Applies to**[Legend](#)

## Rollup Aggregate Function

Specifies the type of aggregation to apply to summarized values. These values appear at the higher levels of lists and crosstabs. For OLAP data sources, a rollup aggregate function of Count Distinct is supported for only levels and member sets.

**Applies to**

[Angular Measure](#), [Area](#), [Bar](#), [Baseline](#), [Baseline](#), [Baseline](#), [Bubble Measure](#), [Bubble Measure](#), [Bullet Measure](#), [Category Baseline](#), [Chart Measure](#), [Chart Node Member](#), [Chart Text Item](#), [Chart Text Item](#), [Color by Value Measure](#), [Colored Region](#), [Colored Region](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Cumulation Line Label](#), [Data Item](#), [Default Measure](#), [Default Measure](#), [Gauge Axis](#), [HTML Item](#), [Image](#), [Line](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [Map Location](#), [Map Location](#), [Map Refinement Location](#), [Map Refinement Location](#), [Marker](#), [Marker](#), [Numeric Baseline](#), [Point Measure](#), [Point Size Measure](#), [Radial Measure](#), [Region Measure](#), [Rich Text Item](#), [Scatter Marker](#), [Target Measure](#), [Target Measure](#), [Text Item](#), [Tolerance Measure](#), [Total Column](#), [Win-Loss Measure](#), [X Axis Measure](#), [Y Axis Measure](#), [Z Axis Measure](#)

## Rollup Processing

Specifies where to compute aggregates. The Extended setting means that aggregates are computed using an extended aggregate operation. The Database setting means that aggregates are computed by the database software. The Local setting means that aggregates are computed by the data retrieval software in the report server, using a running aggregate.

**Applies to**[Query](#)

## Root Members Only

Specifies whether the set contains the root members or all of the members of the hierarchy.

**Applies to**

[Angular Measure](#), [Area](#), [Bar](#), [Baseline](#), [Baseline](#), [Baseline](#), [Baseline](#), [Bubble Measure](#), [Bubble Measure](#), [Bullet Measure](#), [Category Baseline](#), [Chart Measure](#), [Chart Node Member](#), [Chart Text Item](#), [Chart Text Item](#), [Color by Value Measure](#), [Colored Region](#), [Colored Region](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Cumulation Line Label](#), [Default Measure](#), [Default Measure](#), [Gauge Axis](#), [Hierarchy Set](#), [HTML Item](#), [Image](#), [Line](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [Map Location](#), [Map Location](#), [Map Refinement Location](#), [Map Refinement Location](#), [Marker](#), [Marker](#), [Numeric Baseline](#), [Point Measure](#), [Point Size Measure](#), [Radial Measure](#), [Region Measure](#), [Rich Text Item](#), [Scatter Marker](#), [Target Measure](#), [Target Measure](#), [Text Item](#), [Tolerance Measure](#), [Total Column](#), [Win-Loss Measure](#), [X Axis Measure](#), [Y Axis Measure](#), [Z Axis Measure](#)

## Rotate Values

Controls whether or not the values displayed on the chart are rotated when the Chart Orientation property is set to Horizontal. May help make values easier to read on horizontal charts. This property applies only to legacy charts.

### Applies to

[Combination Chart](#), [Pareto Chart](#), [Progressive Chart](#)

## Row Coordinate

Uniquely identifies the row of a node member or spacer on an edge of the crosstab. You cannot modify this value.

### Applies to

[Crosstab Intersection](#)

## Rows Per Page

Specifies the maximum number of rows to show at one time. For Value and Select & Search prompts, allows you to extend or reduce the maximum number of rows beyond the default of 5000.

### Applies to

[Crosstab](#), [List](#), [Repeater](#), [Select & Search Prompt](#), [Tree Prompt](#), [Value Prompt](#)

## Scale

Specifies whether the numeric scale is logarithmic or linear.

### Applies to

[Angular Axis](#), [Gauge Numerical Axis](#), [Numerical Axis](#), [Numerical Axis](#), [Radial Axis](#), [X Axis](#), [Y1 Axis](#), [Y2 Axis](#), [Y Axis](#), [Z Axis](#), [Z Axis](#)

## Scale Interval

Specifies the interval between ticks on the numeric scale. If no value is specified, one will be calculated based on the data.

### Applies to

[Angular Axis](#), [Cumulation Line Axis](#), [Gauge Numerical Axis](#), [Numerical Axis](#), [Numerical Axis](#), [Radial Axis](#), [X Axis](#), [Y1 Axis](#), [Y2 Axis](#), [Y Axis](#), [Z Axis](#), [Z Axis](#)

## Scope

Specifies the scope of the filter in terms of the number of levels.

### Applies to

[Summary Filter](#)

## Search Instructions Text

Specifies the instructions that appear above the keyword search box in select & search prompts. The default text is as follows: Type one or more keywords separated by spaces.

### Applies to

[Select & Search Prompt](#)

## Search Text

Specifies the label that appears on the Search button in select & search prompts. The default label text is Search.

### Applies to

[Select & Search Prompt](#)

## Seconds Text

Specifies the title that appears above the seconds box in interval prompts. The default title text is s.

### Applies to

[Interval Prompt](#)

## Select UI

Specifies which interface the prompt control renders.

### Applies to

[Date & Time Prompt](#), [Date Prompt](#), [Time Prompt](#), [Value Prompt](#)

## Separator

Specifies the separator to use when showing multiple values.

### Applies to

[Context Item Text](#), [Legend](#)

## Series Color

Specifies whether data marker colors are synchronized between charts when you use the same data series for multiple charts within a combination chart. For example, if you create a line chart and a bar chart using the countries data series, and you set this property to Yes, the data marker for Sweden is the same color in both charts. Use this property to eliminate redundant legend entries and help you interpret the data.

### Applies to

[3-D Combination Chart](#), [Combination Chart](#), [Combination Chart](#)

## Series Type

Specifies how the series will be rendered; for example, absolute, stacked, or stacked 100%.

### Applies to

[Area](#), [Bar](#), [Line](#)

## Set Operation

Specifies the set operation to apply to one or more queries, that results in a projection list on which other queries can be based.

### Applies to

[Query Operation](#)

## Set Sorting

Specifies how the set is sorted. By default, the set is not sorted.

### Applies to

[Angular Measure](#), [Area](#), [Bar](#), [Baseline](#), [Baseline](#), [Baseline](#), [Bubble Measure](#), [Bubble Measure](#), [Bullet Measure](#), [Category Baseline](#), [Chart Measure](#), [Chart Node Member](#), [Chart Text Item](#), [Chart Text Item](#), [Color by Value Measure](#), [Colored Region](#), [Colored Region](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Cumulation Line Label](#), [Default Measure](#), [Default Measure](#), [Explicit Member Set](#), [Gauge Axis](#), [Hierarchy Set](#), [HTML Item](#), [Image](#), [Level Set](#), [Line](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [Map Location](#), [Map Location](#), [Map Refinement Location](#), [Map Refinement Location](#), [Marker](#), [Marker](#), [Member Children Set](#), [Named Set](#), [Numeric Baseline](#), [Point Measure](#), [Point Size Measure](#), [Radial Measure](#), [Region Measure](#), [Rich Text Item](#), [Scatter Marker](#), [Set Expression](#), [Target Measure](#), [Target Measure](#), [Text Item](#), [Tolerance Measure](#), [Total Column](#), [Win-Loss Measure](#), [X Axis Measure](#), [Y Axis Measure](#), [Z Axis Measure](#)

## Set Variable Values

Sets the value of variables when the control or an item in the control is selected.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Button](#), [Button Bar](#), [Check Box Group](#), [Combination Chart](#), [Combination Chart](#), [Crosstab](#), [Data Button Bar](#), [Data Check Box Group](#), [Data Drop-Down List](#), [Data List Box](#), [Data Radio Button Group](#), [Data Tab Control](#), [Data Toggle Button Bar](#), [Drop-Down List](#), [Gauge Chart](#), [Gauge Chart](#), [List](#), [List Box](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Radar Chart](#), [Radio Button Group](#), [Scatter Chart](#), [Scatter Chart](#), [Tab Control](#), [Toggle Button Bar](#)

## Share Result Set

Specifies whether to share an identical query between data containers that use it. To share a query, data containers must be lists, repeaters, or repeater tables and must use the same grouping structure and list of properties. The data containers cannot be part of a master detail relationship. When set to Yes, the query sends only one request to the database and shares the result. When set to No, the query is not shared.

### Applies to

[List](#), [Repeater](#), [Repeater Table](#)

## Show Caption

Specifies whether, or where, to show the caption.

### Applies to

[Field Set](#)

## Show Data Points

Specifies whether to show value markers and how they are formatted.

### Applies to

[Line](#)

## Show Data Range in Legend

Specifies whether to show the full range of data in the legend. If set to No, only the values from the palette will be shown.

### Applies to

[Point Layer](#), [Region Layer](#)

## Show Features with No Data

Indicates whether to show the features of a map that do not have corresponding data.

### Applies to

[Point Layer](#), [Region Layer](#)

## Show Feelers

Specifies whether feeler lines are rendered for each marker.

### Applies to

[3-D Scatter Chart](#)

## Show Hover

Specifies whether to highlight areas in the chart that are selectable.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Combination Chart](#), [Combination Chart](#), [Crosstab](#), [Gauge Chart](#), [Gauge Chart](#), [List](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Radar Chart](#), [Scatter Chart](#), [Scatter Chart](#)

## Show Icon

When defined, specifies whether to show the icon for each list item in the drop down list.

### Applies to

[Data Drop-Down List](#), [Drop-Down List](#)

## Show Legend Values

Specifies whether and how to show legend values.

### Applies to

[Legend](#)

## Show line

Specifies whether a line may be rendered. This allows you to show markers without lines.

### Applies to

[Line](#)

## Show Pointer Cursor

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Combination Chart](#), [Combination Chart](#), [Crosstab](#), [Gauge Chart](#), [Gauge Chart](#), [List](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Radar Chart](#), [Scatter Chart](#), [Scatter Chart](#)

## Show Values

Specifies which values to show in a chart legend.

### Applies to

[Legend](#)

## Show Values

Specifies the appearance of data labels in area, bar, line, Pareto, and progressive charts.

### Applies to

[Area](#), [Bar](#), [Line](#), [Pareto Chart](#), [Progressive Chart](#)

## Show Values

Specifies whether to show the labels for the data markers on the cumulation line in a Pareto chart.

### Applies to

[Cumulation Line](#)

## Show Values

Specifies the appearance of data labels in a pie chart.

### Applies to

[Pie Chart](#)

## Show Values

Specifies the appearance of data labels in a scatter chart.

### Applies to

[Scatter Chart](#)

## Show Values

Specifies the appearance of data labels in a bubble chart.

### Applies to

[Bubble Chart](#)

## Size

Specifies the height and width of the control.

### Applies to

[Button](#), [Data Deck](#), [Data Drop-Down List](#), [Data List Box](#), [Data Tab Control](#), [Deck](#), [Drop-Down List](#), [List Box](#), [Tab Control](#)

## Size & Overflow

Specifies the height and width of the object, as well as how overflow content should be treated, using scroll bars and clipping.

### Applies to

[Block](#), [Class](#), [Conditional Block](#), [Conditional Block](#), [Context Item Text](#), [Crosstab](#), [Crosstab Columns](#), [Crosstab Columns](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Rows](#), [Crosstab Rows](#), [Crosstab Space](#), [Crosstab Summary](#), [Field Set](#), [Hyperlink](#), [Hyperlink Button](#), [Image](#), [List](#), [List Cell](#), [List Column](#), [List Column Body](#), [List Columns](#), [List Columns Body Style](#), [List Columns Title Style](#), [List Column Title](#), [List Footer](#), [List Header](#), [List Row](#), [List Row Cells Style](#), [List Summary](#), [Note](#), [Page](#), [Page Body](#), [Page Break Text](#), [Page Footer](#), [Page Header](#), [Prompt Button](#), [Repeater Table](#), [Repeater Table Cell](#), [Table](#), [Table Cell](#), [Table of Contents](#), [Table of Contents Entry](#), [Text Box Prompt](#), [Text Item](#), [Tree Prompt](#), [Value Prompt](#), [Variable Text Item](#), [Win-Loss Chart](#)

## Size & Overflow

Specifies the absolute height and width of the object, as well as how overflow content should be treated, using scroll bars and clipping.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Combination Chart](#), [Combination Chart](#), [Gauge Chart](#), [Gauge Chart](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Radar Chart](#), [Scatter Chart](#), [Scatter Chart](#)

## Size Legend Title

Specifies a title within the legend above the palette for the point size. If this object is not defined, no additional title is drawn. If no legend is drawn, this object is ignored. Styling for this object is inherited from the legend title.

### Applies to

[Point Layer](#)

## Slice Direction

Specifies the direction in which slices appear in a pie chart.

### Applies to

[Pie Chart](#)

## Slide Animation Direction

Specifies the direction from which a new card appears. Select Auto Horizontal or Auto Vertical to automatically set the direction based on the order of cards in the deck. For example, when Auto Horizontal is specified, if you view the first card in a deck and you select to view the fourth card, the fourth card appears from the right. If you then select to view the first card, the first card appears from the left.

### Applies to

[Data Deck](#), [Deck](#)

## Solve Order

Specifies the solve order in the crosstab. The item with the lowest solve order value is calculated first, followed by the next lowest value, and so on. For identical values, column items are calculated first, then row items, and then the measure.

### Applies to

[Crosstab Summary](#)

## Solve Order

Specifies the solve order in crosstabs and charts. The item with the lowest solve order value is calculated first, followed by the next lowest value, and so on. For identical values, in crosstabs, column items are calculated first, then row items, and then the measure. In charts, x-axis items are calculated first and then legend items.

### Applies to

[Calculated Measure](#), [Calculated Member](#), [Data Item](#), [Dimensional Edge Summary](#), [Edge Summary](#)

## Sorting

Specifies the desired sort sequence.

### Applies to

[Chart Node Member](#), [Crosstab Node Member](#), [Data Button Bar](#), [Data Check Box Group](#), [Data Deck](#), [Data Drop-Down List](#), [Data List Box](#), [Data Radio Button Group](#), [Data Tab Control](#), [Data Toggle Button Bar](#), [Level](#), [Select & Search Prompt](#), [Tree Prompt](#), [Value Prompt](#)

## Source Type

Specifies the source type of the text.

### Applies to

[Chart Node Member](#), [Chart Text Item](#), [Chart Text Item](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Space](#), [Cumulation Line Label](#), [Hyperlink](#), [Hyperlink Button](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [Text Item](#)

## Source Type

Specifies the source type of the HTML text.

### Applies to

[Chart Node Member](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Space](#), [HTML Item](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [Rich Text Item](#)

## Source Type

Specifies the source type of the URL.

### Applies to

[Chart Node Member](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Space](#), [Hyperlink](#), [Hyperlink Button](#), [Image](#), [List Cell](#), [List Column Body](#), [List Column Title](#)

## Source Type

Specifies the source type of the bookmark.

### Applies to

[Bookmark](#), [Chart Node Member](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Space](#), [List Cell](#), [List Column Body](#), [List Column Title](#)

## Spacing & Breaking

Specifies text properties such as line height, letter spacing, and word breaking.

### Applies to

[Block](#), [Class](#), [Conditional Block](#), [Context Item Text](#), [Crosstab Columns](#), [Crosstab Columns](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Rows](#), [Crosstab Rows](#), [Crosstab Space](#), [Crosstab Summary](#), [Hyperlink](#), [List Cell](#), [List Column](#), [List Column Body](#), [List Columns](#), [List Columns](#), [List Columns Body Style](#), [List Columns Title Style](#), [List Column Title](#), [List Footer](#), [List Header](#), [List Row](#), [List Row Cells Style](#), [List Summary](#), [Page Body](#), [Page Break Text](#), [Page Footer](#), [Page Header](#), [Repeater Table Cell](#), [Table Cell](#), [Table of Contents](#), [Table of Contents Entry](#), [Text Item](#), [Variable Text Item](#)

## Spider Effects

Specifies whether the chart is rendered with web-like flat concentric circles. The Radial Axis property must be set to Show for this property to take effect.

### Applies to

[Polar Chart](#), [Radar Chart](#)

## SQL

The text of the typed-in SQL. It is assumed to be appropriate for the type and data source. If it is not the query may fail, or produce unexpected results.

**Applies to**

[SQL](#)

## SQL Join Syntax

Controls the syntax to generate for joins. Click Implicit to generate joins in the WHERE clause.

Click Explicit to generate INNER JOIN syntax. If unspecified, the value of the corresponding governor in the model is used.

**Applies to**

[Query](#)

## SQL Syntax

Specifies the syntax of the SQL in the query. A value of Cognos SQL indicates that IBM® Cognos® extended SQL-92 syntax is used. A value of Native SQL indicates that native database SQL is used. You cannot use Cognos SQL if the Processing property for the query is set to Database Only; it must have a value of Limited Local. A value of Pass-Through indicates that a standalone query text syntax is used. The default value is Native SQL.

**Applies to**

[SQL](#)

## Standard Deviations

Specifies a distance from the mean in standard deviations. This value can be positive or negative. A value of zero indicates the mean value.

**Applies to**

[Baseline](#), [Baseline](#), [Baseline](#), [Baseline](#), [Marker](#), [Marker](#)

## Start Position

Specifies the position of one edge of the colored region along the numeric axis. The region extends from the position defined in this property to the position defined in the End Position property. The value that you specify in this property must be less than the value that is specified in the End Position property.

**Applies to**

[Colored Region](#)

## Static Choices

Represents a collection of static choices used by the prompt object.

**Applies to**

[Select & Search Prompt](#), [Value Prompt](#)

## Style Variable

Specifies a variable based on which the object can be conditionally styled.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Angular Axis](#), [Angular Measure](#), [Axis Title](#), [Block](#), [Bubble Chart](#), [Bubble Measure](#), [Caption](#), [Category Axis](#), [Chart Body](#), [Chart Footer](#), [Chart Subtitle](#), [Chart Text Item](#), [Chart Title](#), [Combination Chart](#), [Conditional Block](#), [Conditional Block](#), [Context Item Text](#), [Crosstab](#), [Crosstab Columns](#), [Crosstab Columns](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Rows](#), [Crosstab Rows](#), [Crosstab Space](#), [Crosstab Summary](#), [Cumulation Line Axis](#), [Date & Time Prompt](#), [Date Prompt](#), [Default Measure](#), [Display Layer](#), [Field Set](#), [Gauge Chart](#), [Gauge Labels](#), [Gauge Numerical Axis](#), [Hyperlink](#), [Hyperlink Button](#), [Image](#), [Interval Prompt](#), [Legend](#), [Legend Title](#), [List](#), [List Cell](#), [List Column](#), [List Column Body](#), [List Columns](#), [List Columns](#), [List Columns Body Style](#), [List Columns Title Style](#), [List Column Title](#), [List Footer](#), [List Header](#), [List Row](#), [List Row Cells Style](#), [List Summary](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Note Content](#), [Numerical Axis](#), [Numerical Axis](#), [Page](#), [Page Body](#), [Page Break Text](#), [Page Footer](#), [Page Header](#), [Pareto Chart](#), [Pie Chart](#), [Pie Labels](#), [Point Layer](#), [Point Measure](#), [Point Size Measure](#), [Polar Chart](#), [Progressive Chart](#), [Prompt Button](#), [Radar Chart](#), [Radial Axis](#), [Radial Measure](#), [Region Layer](#), [Region Measure](#), [Repeater Table](#), [Repeater Table Cell](#), [Scatter Chart](#), [Select & Search Prompt](#), [Table](#), [Table Cell](#), [Table of Contents](#), [Table of Contents Entry](#), [Table Row](#), [Target Measure](#), [Text Box Prompt](#), [Text Item](#), [Time Prompt](#), [Tolerance Measure](#), [Tree Prompt](#), [Value Prompt](#), [Win-Loss Chart](#), [X Axis](#), [X Axis](#), [X Axis Measure](#), [Y1 Axis](#), [Y2 Axis](#), [Y Axis](#), [Y Axis](#), [Y Axis Measure](#), [Z Axis](#), [Z Axis](#), [Z Axis](#), [Measure](#)

## Subtitle

Specifies whether a chart subtitle is rendered. This property applies only to legacy charts.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Bubble Chart](#), [Combination Chart](#), [Gauge Chart](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Radar Chart](#), [Scatter Chart](#), [Win-Loss Chart](#)

## Subtitle

Specifies whether a chart subtitle is rendered.

### Applies to

[Bubble Chart](#), [Bullet Chart](#), [Combination Chart](#), [Gauge Chart](#), [Pareto Chart](#), [Pie Chart](#), [Progressive Chart](#), [Scatter Chart](#)

## Summarize Small Items

Specifies whether to summarize small items, such as slices, lines, areas, bars, or columns, in the chart and how they are summarized. You cannot summarize small items in charts that have matrix edges or in charts that have multiple numeric axes.

### Applies to

[Combination Chart](#), [Pareto Chart](#)

## Summarize Small Slices

Specifies whether to summarize small items, such as slices, lines, areas, bars, or columns, in the chart and how they are summarized. You cannot summarize small items in charts that have matrix edges or in charts that have multiple numeric axes.

### Applies to

[Pie Chart](#)

## Summary Text

Specifies summary text for table-like objects. Use to make your reports accessible for people who use screen readers. The summary text is never displayed in visual Web browsers. Summary text is used only for screen readers and speech browsers.

### Applies to

[Crosstab](#), [List](#), [Repeater Table](#), [Table](#)

## Suppress

For SAP BW and DB2 OLAP data sources, specifies the type of suppression to apply to the query results. This property overrides the corresponding governor in the model. If unspecified, the value of the governor in the model is used.

### Applies to

[Query](#)

## Suppression

Specifies zero suppression options for the object.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Combination Chart](#), [Combination Chart](#), [Crosstab](#), [Data Button Bar](#), [Data Check Box Group](#), [Data Deck](#), [Data Drop-Down List](#), [Data List Box](#), [Data Radio Button Group](#), [Data Tab Control](#), [Data Toggle Button Bar](#), [Gauge Chart](#), [Gauge Chart](#), [List](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Radar Chart](#), [Repeater](#), [Repeater Table](#), [Scatter Chart](#), [Scatter Chart](#), [Win-Loss Chart](#)

## Table Header

Specifies whether the cell is a table header. Use to make reports accessible for people who use screen readers. When set to Yes, screen readers and speech browsers programmatically create relationships between the table header and table cells.

**Applies to**[Table Cell](#)

## Table of Contents

Specifies the name of the table of contents to which the entry belongs.

**Applies to**[Table of Contents Entry](#)

## Table of Contents Name

Specifies the name that uniquely identifies the table of contents.

**Applies to**[Table of Contents](#)

## Table Properties

Specifies the properties for the table object.

**Applies to**[Class](#), [Crosstab](#), [List](#), [Repeater Table](#), [Table](#)

## Tab Orientation

Specifies the orientation of the tabs in the tab control.

**Applies to**[Data Tab Control](#), [Tab Control](#)

## Tabs Definition

Defines the tabs in the tab control.

**Applies to**[Tab Control](#)

## Tab Width

Specifies the width of tabs in the tab control.

**Applies to**[Data Tab Control](#), [Tab Control](#)

## Target Color

Specifies a color for the vertical lines that mark the target ranges for target measure values in a metrics chart.

### Applies to

[Metrics Range Chart](#)

## Target Marker

Specifies whether the status indicators will appear in the legend.

### Applies to

[Metrics Range Chart](#)

## Target Marker Border Color

Specifies a color for the borders around target value markers in a metrics chart.

### Applies to

[Metrics Range Chart](#)

## Target Marker Position

Specifies whether the status indicators will be rendered over the first bar in the cluster or the middle of the cluster. Does not apply to stacked charts.

### Applies to

[Metrics Range Chart](#)

## Target Range (%)

Specifies target ranges centered around target measure values.

### Applies to

[Metrics Range Chart](#)

## Text

Specifies the static text to render.

### Applies to

[Chart Node Member](#), [Chart Text Item](#), [Chart Text Item](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Space](#), [Cumulation Line Label](#), [Hyperlink](#), [Hyperlink Button](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [Text Item](#), [Total Column](#)

## Text Color

Specifies the color of the text of each label.

### Applies to

[Check Box Group](#), [Data Check Box Group](#), [Data Radio Button Group](#), [Radio Button Group](#)

## Text Flow & Justification

Specifies text flow properties, such as direction, writing mode, and justification.

### Applies to

[Block](#), [Class](#), [Conditional Block](#), [Conditional Block](#), [Context Item Text](#), [Crosstab](#), [Crosstab Columns](#), [Crosstab Columns](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Rows](#), [Crosstab Rows](#), [Crosstab Space](#), [Crosstab Summary](#), [Hyperlink](#), [List](#), [List Cell](#), [List Column](#), [List Column Body](#), [List Columns](#), [List Columns](#), [List Columns Body Style](#), [List Columns Title Style](#), [List Column Title](#), [List Footer](#), [List Header](#), [List Row](#), [List Row Cells Style](#), [List Summary](#), [Page](#), [Page Body](#), [Page Break Text](#), [Page Footer](#), [Page Header](#), [Repeater Table](#), [Repeater Table Cell](#), [Table](#), [Table Cell](#), [Table of Contents](#), [Table of Contents Entry](#), [Text Item](#), [Variable Text Item](#)

## Text Orientation

Specifies the orientation of the text on the gauge axis.

### Applies to

[Gauge Axis](#)

## Text Source Variable

Specifies a variable based on which the text source can be chosen.

### Applies to

[Chart Node Member](#), [Chart Text Item](#), [Chart Text Item](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Space](#), [Cumulation Line Label](#), [Hyperlink](#), [Hyperlink Button](#), [List Cell](#), [List Column Body](#), [List Column Title](#), [Text Item](#)

## Text Truncation

Specifies whether and how legend items are truncated.

### Applies to

[Axis Labels](#), [Legend](#)

## Title

Specifies whether a chart title is rendered. This property applies only to legacy charts.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Bubble Chart](#), [Combination Chart](#), [Gauge Chart](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Radar Chart](#), [Scatter Chart](#), [Win-Loss Chart](#)

## Title

Specifies whether a chart title is rendered.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Combination Chart](#), [Combination Chart](#), [Gauge Chart](#), [Gauge Chart](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Radar Chart](#), [Scatter Chart](#), [Scatter Chart](#)

## Tolerance Color

Specifies a color for the vertical lines that mark the tolerance ranges for target measure values in a metrics chart.

### Applies to

[Metrics Range Chart](#)

## Tolerance Label

Specifies whether the label for the Target Tolerance will appear in the legend.

### Applies to

[Metrics Range Chart](#)

## Tooltips

Specifies whether tooltips are shown in a chart when you hover over data elements.

### Applies to

[Bubble Chart](#), [Bullet Chart](#), [Combination Chart](#), [Gauge Chart](#), [Pareto Chart](#), [Progressive Chart](#), [Scatter Chart](#)

## Tooltips

Specifies whether tooltips are shown in a pie chart when you hover over data elements.

### Applies to

[Pie Chart](#)

## Tooltips

Specifies whether tooltips are shown in the chart when you hover over data elements. Tooltips are not supported in PDF documents. This property applies only to legacy charts.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Bubble Chart](#), [Combination Chart](#), [Gauge Chart](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Pie Chart](#), [Polar Chart](#), [Progressive Chart](#), [Radar Chart](#), [Scatter Chart](#), [Win-Loss Chart](#)

## Tooltips

### Top Position

Specifies the position of the top edge of the colored region.

### Applies to

[Colored Region](#)

### Top Position (px)

Specifies the pixel position of the top edge of the legend measured from the bottom of the chart.

### Applies to

[Legend](#)

## Total Column

Specifies whether a total column is rendered. This property applies only to legacy charts.

### Applies to

[Progressive Chart](#)

## Total Column

Specifies whether a bar that shows the total cumulative value is rendered.

### Applies to

[Progressive Chart](#)

## Total Column Color

Specifies the color, gradient, or pattern of the total column on the progressive chart.

### Applies to

[Total Column](#)

## To Text

Specifies the label that appears beside the end of a range. This property applies to the following prompt types: date, date & time, time, and interval. The default label text is To.

### Applies to

[Date & Time Prompt](#), [Date Prompt](#), [Interval Prompt](#), [Text Box Prompt](#), [Time Prompt](#), [Value Prompt](#)

## Trendline Label

Specifies whether to show the default label for the trend line. When set to No, you can type your own label text.

### Applies to

[Trendline](#)

## Trendlines

Adds a trend line or curve that indicates the general direction of the data over time.

### Applies to

[Bubble Chart](#), [Combination Chart](#), [Scatter Chart](#)

## Truncation

Specifies whether labels can be truncated.

### Applies to

[Category Axis](#), [Gauge Labels](#), [Pie Labels](#), [X Axis](#), [Y Axis](#)

## Truncation Text

Specifies the text to append when a label is truncated.

### Applies to

[Category Axis](#), [Gauge Labels](#), [Legend](#), [Pie Labels](#), [X Axis](#), [Y Axis](#)

## Type

Specifies the type of variable.

### Applies to

[Variable](#)

## Type

Specifies the type of trend line.

**Applies to**

Trendline

**Type**

Specifies the behavior of the prompt button.

**Applies to**

Prompt Button

**Type**

Specifies the type of object.

**Applies to**

Angular Measure, Area, Bar, Baseline, Baseline, Baseline, Baseline, Bubble Measure, Bubble Measure, Bullet Measure, Calculated Measure, Calculated Member, Category Baseline, Chart Measure, Chart Node Member, Chart Text Item, Chart Text Item, Color by Value Measure, Colored Region, Colored Region, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Cumulation Line Label, Data Item, Default Measure, Default Measure, Dimensional Edge Summary, Edge Summary, Explicit Member Set, Gauge Axis, Hierarchy Set, HTML Item, Image, Intersection (Tuple), Level Set, Line, List Cell, List Column Body, List Column Title, List Summary, Map Location, Map Location, Map Refinement Location, Map Refinement Location, Marker, Marker, Measure, Member, Member Children Set, Member Property, Named Set, Numeric Baseline, Point Measure, Point Size Measure, Radial Measure, Region Measure, Rich Text Item, Scatter Marker, Set Expression, Target Measure, Target Measure, Text Item, Tolerance Measure, Total Column, Win-Loss Measure, X Axis Measure, Y Axis Measure, Z Axis Measure

**Upper Range Skew (%)**

Specifies a percentage that affects the positioning of tolerance bar and range with respect to the target value.

**Applies to**

Metrics Range Chart

**URL**

Specifies the URL, using static text.

**Applies to**

Chart Node Member, Crosstab Corner, Crosstab Fact Cells, Crosstab Intersection, Crosstab Member Fact Cells, Crosstab Node Member, Crosstab Space, Hyperlink, Hyperlink Button, Image, List Cell, List Column Body, List Column Title

**URL Source Variable**

Specifies a variable based on which the URL source can be chosen.

### Applies to

[Chart Node Member](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Space](#), [Hyperlink](#), [Hyperlink Button](#), [Image](#), [List Cell](#), [List Column Body](#), [List Column Title](#)

## Usage

Specifies whether the usage of this object is Optional, Required, or Disabled. When Optional, this condition applies if all parameters referenced by the expression are provided with values. If the expression does not refer to any parameters, then this condition is always applied. When Disabled, this condition is never applied, which is useful for authoring and testing purposes.

### Applies to

[Detail Filter](#), [Summary Filter](#)

## Use 1.x Behavior

Specifies that IBM® Cognos® ReportNet® query semantics are used if they differ from IBM Cognos BI query rules.

### Applies to

[Query](#)

## Use Aggregate Cache

For SAP BW data sources, specifies whether to use the aggregation cache. We recommend that you use the default value.

### Applies to

[Query](#)

## Use Detail Value on Page

Specifies whether to render a detail value or an aggregate value for a text item that uses a data item as its source. Use this property only when you want to render the value that appears in the first or last detail row of a list, repeater or repeater table on the same page as the text item.

### Applies to

[Text Item](#)

## Use for Parameter Info

Specifies whether the query should be given priority when determining parameter information. Queries with this property set to Yes are checked for parameter information first, followed by queries with this property set to Default. Queries with this property set to No will not be checked for parameter information unless it is referenced in a query that will be checked. Setting this property to Yes on parameterized queries can improve performance in displaying prompt pages.

**Applies to**[Query](#)

## Use Local Cache

Specifies whether a query is a candidate for query reuse. If set to Yes, the query engine can reuse an existing SQL result. If set to No, the query is executed rather than using cached results. This property applies only to relational and dimensionally-modeled relational (DMR) data sources.

**Applies to**[Query](#)

## User SAP Member Cache

Signals to the SAP BW provider whether the query associated with this property is cached to the IBM® Cognos® BI member cache. When set to Yes, the member cache is populated with the dimensions in the query, encrypted, and saved for later use.

**Applies to**[Query](#)

## Use Same Range For All Instances

Specifies that all instances of the chart use the same maximum value. When set to No, the axis maximum value is recalculated for each chart instance. It is only relevant if the chart is involved in a master detail relationship.

**Applies to**[Angular Axis](#), [Gauge Numerical Axis](#), [Numerical Axis](#), [Numerical Axis](#), [Radial Axis](#), [X Axis](#), [Y1 Axis](#), [Y2 Axis](#), [Y Axis](#), [Z Axis](#), [Z Axis](#)

## Use SAP MUN as Business Key

Specifies whether to return the full SAP MUN as the value for the business key. Use this query hint to allow a drill-down on a business key for a non-leaf member of an external hierarchy. When set to Yes, the full MUN appears in the report output.

**Applies to**[Query](#)

## Use Set Aggregation

Specifies whether set aggregation is used.

**Applies to**[Dimensional Edge Summary](#), [Edge Summary](#)

## Use SQL Parameters

Specifies whether the generated SQL uses parameter markers or literal values. When set to Marker, specifies that the generated SQL uses markers to denote that the value will be provided later. When set to Literal, uses literal values in the generated SQL. If not specified, the server determines the behavior.

### Applies to

[Query](#)

## Use SQL With Clause

Specifies whether to send a request to the database using an SQL WITH clause. When set to Yes, and if the database supports WITH clauses, a WITH clause request is generated. When set to No, or if the database does not support WITH clauses, a request using derived tables is generated.

### Applies to

[Query](#)

## Use Thousands Separator

Specifies whether to delimit digit groups with the thousands separator.

### Applies to

[Text Box Prompt](#)

## Use Value

Specifies the values used by the prompt object.

### Applies to

[Tree Prompt](#)

## Use Value

Specifies the values used by the prompt object. These values can be different than the ones that are rendered to the user.

### Applies to

[Select & Search Prompt](#), [Value Prompt](#)

## Value Location

Specifies where values and labels are to be rendered in the chart. This property applies only to legacy charts.

**Applies to**

[Area](#), [Bar](#), [Bubble Chart](#), [Combination Chart](#), [Cumulation Line](#), [Line](#), [Metrics Range Chart](#), [Pareto Chart](#), [Polar Chart](#), [Progressive Chart](#), [Radar Chart](#), [Scatter Chart](#)

## Value Markers

Specifies whether to show special value markers and how they are formatted.

**Applies to**

[Line](#)

## Value Representation

Specifies whether values are rendered as percentages.

**Applies to**

[Pie Chart](#)

## Values

Specifies what values to show in the chart and whether to show the corresponding measure, series, or category label.

**Applies to**

[3-D Scatter Chart](#), [Bubble Chart](#), [Polar Chart](#), [Scatter Chart](#)

## Values

Specifies whether values are rendered in the chart.

**Applies to**

[3-D Area](#), [3-D Bar](#), [3-D Line](#), [Area](#), [Bar](#), [Cumulation Line](#), [Line](#), [Marimekko Chart](#), [Pareto Chart](#), [Point Layer](#), [Progressive Chart](#), [Radar Chart](#), [Region Layer](#)

## Values

Specifies whether values are rendered.

**Applies to**

[Pie Chart](#)

## Value Type

Specifies whether absolute values are rendered rather than cumulative values.

**Applies to**

[Area](#), [Bar](#), [Line](#)

## Vertical Alignment

Specifies how objects contained in this object are vertically aligned.

### Applies to

[Class](#), [Crosstab Columns](#), [Crosstab Columns](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Rows](#), [Crosstab Rows](#), [Crosstab Space](#), [Crosstab Summary](#), [List Cell](#), [List Column](#), [List Column Body](#), [List Columns](#), [List Columns](#), [List Columns Body Style](#), [List Columns Title Style](#), [List Column Title](#), [List Footer](#), [List Header](#), [List Row](#), [List Row Cells Style](#), [Page Body](#), [Page Footer](#), [Page Header](#), [Repeater Table Cell](#), [Table Cell](#), [Table Row](#)

## Visible

Specifies whether a column or row is visible when the report is executed.

### Applies to

[Crosstab Node Member](#), [Crosstab Space](#), [Crosstab Summary](#)

## Visible

Specifies whether to display the object. When set to No, the object is hidden but a fixed space is reserved in the report.

### Applies to

[3-D Combination Chart](#), [3-D Scatter Chart](#), [Angular Axis](#), [Axis Title](#), [Block](#), [Bubble Chart](#), [Bubble Chart](#), [Bullet Chart](#), [Category Axis](#), [Chart Footer](#), [Chart Subtitle](#), [Chart Title](#), [Class](#), [Combination Chart](#), [Combination Chart](#), [Conditional Block](#), [Conditional Block](#), [Context Item Text](#), [Crosstab](#), [Cumulation Line Axis](#), [Date & Time Prompt](#), [Date Prompt](#), [Field Set](#), [Gauge Chart](#), [Gauge Chart](#), [Gauge Labels](#), [Gauge Numerical Axis](#), [Hyperlink](#), [Image](#), [Interval Prompt](#), [Legend](#), [Legend Title](#), [List](#), [List Summary](#), [Map](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Note Content](#), [Numerical Axis](#), [Numerical Axis](#), [Page Break Text](#), [Pareto Chart](#), [Pareto Chart](#), [Pie Chart](#), [Pie Chart](#), [Pie Labels](#), [Polar Chart](#), [Progressive Chart](#), [Progressive Chart](#), [Radar Chart](#), [Radial Axis](#), [Repeater Table](#), [Scatter Chart](#), [Scatter Chart](#), [Select & Search Prompt](#), [Table](#), [Table of Contents](#), [Table of Contents Entry](#), [Text Box Prompt](#), [Text Item](#), [Time Prompt](#), [Tree Prompt](#), [Value Prompt](#), [Variable Text Item](#), [Win-Loss Chart](#), [X Axis](#), [X Axis](#), [Y1 Axis](#), [Y2 Axis](#), [Y Axis](#), [Z Axis](#), [Z Axis](#)

## Visual Angle

Specifies the angle, in degrees, in which the chart objects will be displayed when the chart has 3-D effects. This property applies only to legacy charts.

### Applies to

[Combination Chart](#), [Marimekko Chart](#), [Metrics Range Chart](#), [Pareto Chart](#), [Progressive Chart](#)

## White Space

Specifies whether the text in the object is wrapped or appears all on one line. The term white space refers to the characters at which text can be wrapped, the spaces between words. The default for this property is to wrap text.

### Applies to

[Block](#), [Class](#), [Conditional Block](#), [Conditional Block](#), [Crosstab Columns](#), [Crosstab Columns](#), [Crosstab Corner](#), [Crosstab Fact Cells](#), [Crosstab Intersection](#), [Crosstab Member Fact Cells](#), [Crosstab Node Member](#), [Crosstab Rows](#), [Crosstab Rows](#), [Crosstab Space](#), [Crosstab Summary](#), [List Cell](#), [List Column](#), [List Column Body](#), [List Columns](#), [List Columns](#), [List Columns Body Style](#), [List Columns Title Style](#), [List Column Title](#), [List Footer](#), [List Header](#), [List Row](#), [List Row Cells Style](#), [Page Body](#), [Page Footer](#), [Page Header](#), [Repeater Table Cell](#), [Table Cell](#), [Table of Contents](#), [Table of Contents Entry](#)

## Width (px)

Specifies the width of the note, in pixels.

### Applies to

[Note](#)

## Win Color

Specifies a color, color gradient, or pattern to apply to the win values.

### Applies to

[Win-Loss Chart](#)

## Win-Loss Threshold

Specifies the win-loss value in a win-loss chart. It represents values that are ties, which are mapped on the zero line.

### Applies to

[Win-Loss Chart](#)

## X Axis

Specifies whether the axis is rendered.

### Applies to

[3-D Combination Chart](#), [Bubble Chart](#), [Scatter Chart](#)

## X-Axis Data Item Value

Specifies the scatter marker position on the X-axis.

**Applies to**

Scatter Marker

## Y1 Axis

Specifies whether the axis is rendered.

**Applies to**

Combination Chart, Metrics Range Chart

## Y2 Axis

Specifies whether the axis is rendered.

**Applies to**

Combination Chart

## Y2 Axis Position

Specifies how the second Y axis is rendered. When Y2 Axis Position is set to Dual, the Y2 axis appears across from the Y1 numeric axis. When Y2 Axis Position is set to Bipolar, the Y2 axis appears below the Y1 axis. For example, in a combination chart showing Revenue and Quantity sold by Retailer type, with the Y2 Axis Position set to Dual, the Revenue columns and the Quantity sold line overlap because the Revenue axis (Y1) and the Quantity sold axis (Y2) are across from each other. However, with the Y2 Axis Position set to Bipolar, the Revenue columns appear above the Quantity sold line, and the data does not overlap. This property applies only to legacy charts.

**Applies to**

Combination Chart

## Y Axis

Specifies whether the axis is rendered.

**Applies to**

3-D Combination Chart, Bubble Chart, Marimekko Chart, Scatter Chart

## Y-Axis Data Item Value

Specifies the scatter marker position on the Y-axis.

**Applies to**

Scatter Marker

## Data Formatting Properties

The following is a list of properties available in the data formatting dialog.

## "Not Applicable" Characters

Specifies the characters to be displayed when the value to be formatted was not applicable. The default value is two dashes (--). Note that the format will be applied only if the data source supports this error condition.

## Any Error Characters

Specifies the characters to be displayed when the value to be formatted was not available because of an error. This property is overridden by the more specific formatting error conditions, such as Security Error Characters. The default value is two dashes (--). Note that the format will be applied only if the data source supports this error condition.

## Calendar Type

Specifies the type of calendar to be displayed. The date values will be mapped to the selected calendar before being formatted. The default value is inherited from the user's content language. Note that the Japanese Imperial setting is only applicable for Japanese languages.

## Clock

Specifies whether to display the time in 12-hour or 24-hour format. The default value is inherited from the user's content language.

## Currency

Specifies the currency to be used. The default currency symbol will be displayed unless the values of the Currency Display and Currency Symbol properties are changed. The default value is inherited from the model.

## Currency Display

Specifies whether to display the international or local currency symbol. By default, the local currency symbol is displayed.

## Currency Symbol

Specifies a character or characters to use as the symbol to identify the local currency. This symbol will precede the number and any sign, even if it is a leading sign. A space between the symbol and the numeric value can be specified by entering it in this property, after the symbol. The default value is inherited from the user's content language.

## Currency Symbol Position

Specifies where the currency symbol will appear. If End is selected, any spaces that follow the character or characters in the Currency Symbol or International Currency Symbol properties will be rendered between the number and the symbol. The default value is inherited from the user's content language.

## Date Ordering

Specifies the order in which to display the day, month, and year. The default value is inherited from the user's content language.

## Date Separator

Specifies the character to be displayed between the year, month, and day. The default value is inherited from the user's content language.

## Date Style

Specifies the date style. The results rendered are determined by the language. Generally, Short uses only numbers, Medium uses some abbreviated words, Long uses complete words, and Full includes all available details.

## Decimal Separator

Specifies the character that will separate non-decimal numbers from decimals. This property is ignored if no decimals are displayed. The default value is inherited from the user's content language.

## Display AM / PM Symbols

Specifies whether to display the AM or PM symbols. The default value is inherited from the user's content language.

## Display As Exponent

Specifies whether to render values in scientific notation, using exponents. If this property is set to No, scientific notation will not be used. If this property is not specified, scientific notation will be used only when values exceed the maximum number of digits. The default value is inherited from the user's content language.

## Display Days

Specifies whether to display the day. The format of the day can be controlled by selecting one of the specific formats. Selecting Julian means that the 3-digit day of the year will be displayed. The default value is inherited from the user's content language.

## Display Eras

Specifies whether to display the era. The default value is inherited from the user's content language.

## Display Hours

Specifies whether to display the hours. The default value is inherited from the user's content language.

## Display Milliseconds

Specifies whether to display the milliseconds. The format of the milliseconds can be controlled by selecting one of the specific formats. This property is ignored if seconds are not displayed. The default value is inherited from the user's content language.

## Display Minutes

Specifies whether to display the minutes. The format of the minutes can be controlled by selecting one of the specific formats. The default value is inherited from the user's content language.

## Display Months

Specifies whether to display the month. The format of the month can be controlled by selecting one of the specific formats. The default value is inherited from the user's content language.

## Display Months

Specifies whether to display the month.

## Display Seconds

Specifies whether to display the seconds. The format of the seconds can be controlled by selecting one of the specific formats. The default value is inherited from the user's content language.

## Display Time Zone

Specifies whether to display the time zone. The default value is inherited from the user's content language.

## Display Weekdays

Specifies whether to display the weekday. The format of the weekday can be controlled by selecting one of the specific formats. The default value is inherited from the user's content language.

## Display Years

Specifies whether to display the year. The first two digits of the year, which indicate the century, can be controlled by selecting one of the associated property values. The default value is inherited from the user's content language.

## Display Years

Specifies whether to display the year.

## Divide By Zero Characters

Specifies the characters to be displayed when a numeric value is the result of a division by zero. The default value is /0. Note that the format will be applied only if the data source supports this error condition.

## Exponent Symbol

Specifies the character to be displayed to identify exponents if the scientific notation is used. The symbol will be rendered after the number, separated by a space. The default value is inherited from the user's content language.

## Group Size (digits)

Specifies the primary grouping size. If a value is specified it represents the number of digits to the left of the decimal point to be grouped together and separated by the thousands separator. The default value is inherited from the user's content language.

## International Currency Symbol

Specifies a character or characters to use as a symbol to identify the international currency. This symbol will replace the currency symbol. A space between the symbol and the numeric value can be specified by entering it in this property, after the symbol. The default value is inherited from the user's content language.

## Mantissa (digits)

Specifies the number of digits to be displayed following the exponent symbol if the scientific notation is used.

## Maximum No. of Digits

Specifies the maximum number of digits that can be displayed. If the maximum number of digits is not sufficient to display the value, a scientific notation will be used. The default value is inherited from the user's content language.

## Minimum No. of Digits

Specifies the minimum number of digits that can be displayed. If the minimum number of digits is too high to display a value, the padding character will be used. The default value is inherited from the user's content language.

## Missing Value Characters

Specifies the character or characters to be displayed when the value is missing. If no value is entered for this property, an empty string will be displayed.

## Negative Pattern

Specifies a presentation format, based on patterns, for negative numbers. Some restrictions exist. The numerical part of the negative pattern is ignored. Only the suffix and the prefix are used. For example, in the pattern ABC#,##0.#EFG, ABC is the prefix, EFG is the suffix and #,##0.# is the numerical part of the pattern.

## Negative Sign Position

Specifies where the negative sign will appear. The default value is inherited from the user's content language.

## Negative Sign Symbol

Specifies how to display negative numbers. The default value is inherited from the user's content language.

## No. of Decimal Places

Specifies the number of digits to be displayed to the right of the decimal point. If this property is not set, the number of decimal places will vary depending on the number rendered.

## Numeric Overflow Characters

Specifies the characters to be displayed when a numeric value is the result of a numeric overflow. The default value is two dashes (--). Note that the format will be applied only if the data source supports this error condition.

## Padding Character

Specifies the character that will be used to pad values that have fewer digits than the minimum number of digits. The default value is inherited from the user's content language.

## Pattern

Specifies a presentation format that is based on patterns. The pattern format overrides formats specified in other properties. For example, to format the date as 2009/12/31 23:59:59 PM, use the pattern yyyy/mm/dd hh:mm:ss aa. For example, to format thousands using the letter K, set the Format Type to Number, set the Scale to -3 (to remove 000), and then use the pattern to #####K.

## Percentage Symbol

Specifies whether to display the values per hundred (percent) or per thousand. The symbol will be appended to the number and any trailing sign. A space between the numeric value and the symbol can be specified by entering it in this property, after the symbol. The default value is inherited from the user's content language.

## Percent Scale (integer)

Scale to be applied to value after formatting. If omitted, no percent scale will be applied and the value will be formatted according to the normal decimal positioning associated with the percent (or per mille) symbol.

## Scale

Specifies how many digits to move the decimal delimiter for formatting purposes. For example, move the decimal three spaces to present values in thousands. The default value is inherited from the database field.

## Secondary Group Size (digits)

Specifies the secondary grouping size. If a value is specified it represents the number of digits to the left of the primary group that will be grouped together and separated by the thousands separator. If this property is left blank, the secondary grouping of digits is the same number as the primary group size, as specified by the Group Size (digits) property. The default value is inherited from the user's content language.

## Security Error Characters

Specifies the characters to be displayed when the value to be formatted was not available for security reasons. The default value is #!Security. Note that the format will be applied only if the data source supports this error condition.

## Thousands Separator

Specifies how to delimit digit groups, such as thousands. This property is only used if the Use Thousands Separator property is set to Yes. The default value is inherited from the user's content language.

## Time Separator

Specifies the character to be displayed between the hour, minute, and second. The default value is inherited from the user's content language.

## Time Style

Specifies the time style to be displayed. The exact results that will be rendered are determined by the language. Generally, Short means that the minimum details will be displayed, Long adds seconds, and Full means that all details are displayed, including the time zone. The default value is inherited from the user's content language.

## Time Unit

Specifies the unit of measure of the value. This property will be ignored if any day or time components are shown. The default value is inherited from the user's content language.

## Use Thousands Separator

Specifies whether the grouping delimiter will be applied as defined by the Group Size property. The default value is inherited from the user's content language.

## Zero Value Characters

Specifies the character or characters to be displayed when the value is zero (0). If no value is entered for this property, the Maximum No. of Digits property determines how many zero digits are displayed.



---

# Appendix A: Calculation Components

---

You build calculations, or expressions, in the expression editor using the components that are defined in this section.

## Operators

Operators specify what happens to the values on either side of the operator. Operators are similar to functions, in that they manipulate data items and return a result.

(

Identifies the beginning of an expression.

### Syntax

( expression )

)

Identifies the end of an expression.

### Syntax

( expression )

\*

Multiplies two numeric values.

### Syntax

value1 \* value2

,

Separates expression components.

### Syntax

expression ( parameter1, parameter2 )

/

Divides two numeric values.

### Syntax

value1 / value2

## Appendix A: Calculation Components

||

Concatenates, or joins, strings.

### Syntax

```
string1 || string2
```

+

Adds two numeric values.

### Syntax

```
value1 + value2
```

-

Subtracts two numeric values or negates a numeric value.

### Syntax

```
value1 - value2
or
- value
```

<

Compares the values that are represented by "value1" against "value2" and retrieves the values that are less than "value2".

### Syntax

```
value1 < value2
```

<=

Compares the values that are represented by "value1" against "value2" and retrieves the values that are less than or equal to "value2".

### Syntax

```
value1 <= value2
```

<>

Compares the values that are represented by "value1" against "value2" and retrieves the values that are not equal to "value2".

### Syntax

```
value1 <> value2
```

=

Compares the values that are represented by "value1" against "value2" and retrieves the values that are equal to "value2".

**Syntax**

```
value1 = value2
```

&gt;

Compares the values that are represented by "value1" against "value2" and retrieves the values that are greater than "value2".

**Syntax**

```
value1 > value2
```

-&gt;

Separates the components in a literal member expression.

**Syntax**

```
[namespace].[dimension].[hierarchy].[level]->[L1]
```

&gt;=

Compares the values that are represented by "value1" against "value2" and retrieves the values that are greater than or equal to "value2".

**Syntax**

```
value1 >= value2
```

**and**

Returns "true" if the conditions on both sides of the expression are true.

**Syntax**

```
argument1 and argument2
```

**auto**

Works with summary expressions to define the scope to be adjusted based on the grouping columns in the query. The scope is context-dependent.

**Syntax**

```
aggregate_function (expression AUTO)
```

**between**

Determines if a value falls in a given range.

**Syntax**

```
expression between value1 and value2
```

**Example**

```
[Revenue] between 200,000 and 300,000
```

## Appendix A: Calculation Components

**Result:** Returns the number of results with revenues between 200,000 and 300,000.

| Revenue          | Between |
|------------------|---------|
| -----            | -----   |
| \$332,986,338.06 | false   |
| \$230,110,270.55 | true    |
| \$107,099,659.94 | false   |

## case

Works with when, then, else, and end. Case identifies the beginning of a specific situation, in which when, then, and else actions are defined.

### Syntax

```
case expression { when expression then expression } [else expression] end
```

## contains

Determines if "string1" contains "string2".

### Syntax

```
string1 contains string2
```

## currentMeasure

Keyword that can be used as the first argument of member summary functions. This function appears in the Total Revenue by Country sample report in the GO Data Warehouse (query) package.

### Syntax

```
aggregate_function (currentMeasure within set expression)
```

## default

Works with the lookup construct.

### Syntax

```
lookup (....) in (....) default (....)
```

## distinct

A keyword used in an aggregate expression to include only distinct occurrences of values. See also the function unique.

### Syntax

```
distinct dataItem
```

### Example

```
count (distinct [OrderDetailQuantity])
```

Result: 1704

## else

Works with the if or case constructs. If the if condition or the case expression are not true, then the else expression is used. This function appears in the Top 10 Retailers for 2005 sample report in the GO Data Warehouse (analysis) package.

### Syntax

```
if (condition) then else (expression) , or case else (expression)
end
```

## end

Indicates the end of a case or when construct.

### Syntax

```
case end
```

## ends with

Determines if "string1" ends with "string2".

### Syntax

```
string1 ends with string2
```

## for

Works with summary expressions to define the scope of the aggregation in the query.

### Syntax

```
aggregate_function (expression for expression { , expression })
```

## for all

Works with summary expressions to define the scope to be all the specified grouping columns in the query. See also the for clause.

### Syntax

```
aggregate_function (expression for ALL expression { , expression })
```

## for any

Works with summary expressions to define the scope to be adjusted based on a subset of the grouping columns in the query. Equivalent to the for clause.

### Syntax

```
aggregate_function (expression for ANY expression { , expression })
```

## for report

Works with summary expressions to set the scope to be the whole query. See also the for clause. This function appears in the Customer Returns and Satisfaction sample report in the GO Data Warehouse (analysis) package.

### Syntax

```
aggregate_function (expression for report)
```

## if

Works with the then and else constructs. If defines a condition; when the if condition is true, the then expression is used. When the if condition is not true, the else expression is used. This function appears in the Top 10 Retailers for 2005 sample report in the GO Data Warehouse (analysis) package.

### Syntax

```
if (condition) then (expression) else (expression)
```

## in

Determines if "expression1" exists in a given list of expressions.

### Syntax

```
expression1 in (expression_list)
```

## in\_range

Determines if "expression1" exists in a given list of constant values or ranges.

### Syntax

```
expression1 in_range { constant : constant [, constant : constant] }
```

### Example 1

```
[code] in_range { 5 }
```

Result: This is equivalent to [code] = 5.

### Example 2

```
[code] in_range { 5: }
```

Result: This is equivalent to [code] >= 5.

### Example 3

```
[code] in_range { :5 }
```

Result: This is equivalent to [code] <= 5.

### Example 4

```
[code] in_range { 5:10 }
```

Result: This is equivalent to ( [code] >= 5 and [code] <= 10 ).

**Example 5**

```
[code] in_range { :5,10,20: }
```

Result: This is equivalent to ( [code] <= 5 or [code] = 10 or [code] >= 20 ).

**is missing**

Determines if "value" is undefined in the data.

**Syntax**

```
value is missing
```

**is null**

Determines if "value" is undefined in the data.

**Syntax**

```
value is null
```

**is not missing**

Determines if "value" is defined in the data.

**Syntax**

```
value is not missing
```

**is not null**

Determines if "value" is defined in the data.

**Syntax**

```
value is not null
```

**like**

Determines if "string1" matches the pattern of "string2".

**Syntax**

```
string1 LIKE string2
```

**lookup**

Finds and replaces data with a value you specify. It is preferable to use the case construct.

**Syntax**

```
lookup (name) in (value1 --> value2) default (expression)
```

**Example**

```
lookup ([Country]) in ('Canada'--> ([List Price] * 0.60), 'Australia'-->
([List Price] * 0.80)) default ([List Price])
```

## Appendix A: Calculation Components

### not

Returns TRUE if "argument" is false or returns FALSE if "argument" is true.

#### Syntax

```
NOT argument
```

### or

Returns TRUE if either of "argument1" or "argument2" are true.

#### Syntax

```
argument1 or argument2
```

### prefilter

Performs a summary calculation before applying the summary filter.

#### Syntax

```
summary ([expression] prefilter)
```

### rows

Counts the number of rows output by the query. Use with Count () .

#### Syntax

```
count (ROWS)
```

### starts with

Determines if "string1" starts with "string2".

#### Syntax

```
string1 starts with string2
```

### then

Works with the if or case constructs. When the if condition or the when expression are true, the then expression is used. This function appears in the Top 10 Retailers for 2005 sample report in the GO Data Warehouse (analysis) package.

#### Syntax

```
if (condition) then ..., or case expression when expression then end
```

### when

Works with the case construct. You can define conditions to occur when the when expression is true.

#### Syntax

```
case [expression] when ... end
```

## Summaries

This list contains predefined functions that return either a single summary value for a group of related values or a different summary value for each instance of a group of related values.

### aggregate

Returns a calculated value using the appropriate aggregation function, based on the aggregation type of the expression. This function appears in the Budget vs. Actual sample report in the GO Data Warehouse (analysis) package.

#### Syntax

```
aggregate (expression [auto])
aggregate (expression for [all|any] expression { , expression })
aggregate (expression for report)
```

### average

Returns the average value of selected data items. Distinct is an alternative expression that is compatible with earlier versions of the product.

#### Syntax

```
average ([distinct] expression [auto])
average ([distinct] expression for [all|any] expression { , expression })
average ([distinct] expression for report)
```

#### Example

```
average (Sales)
```

**Result:** Returns the average of all Sales values.

### count

Returns the number of selected data items excluding null values. Distinct is an alternative expression that is compatible with earlier versions of the product.

#### Syntax

```
count ([distinct] expression [auto])
count ([distinct] expression for [all|any] expression { , expression })
count ([distinct] expression for report)
```

#### Example

```
count (Sales)
```

**Result:** Returns the total number of entries under Sales.

### maximum

Returns the maximum value of selected data items. Distinct is an alternative expression that is compatible with earlier versions of the product.

## Appendix A: Calculation Components

### Syntax

```
maximum ([distinct] expression [auto])
maximum ([distinct] expression for [all|any] expression { , expression })
maximum ([distinct] expression for report)
```

### Example

```
maximum (Sales)
```

Result: Returns the maximum value out of all Sales values.

## median

Returns the median value of selected data items.

### Syntax

```
median (expression [auto])
median (expression for [all|any] expression { , expression })
median (expression for report)
```

## minimum

Returns the minimum value of selected data items. Distinct is an alternative expression that is compatible with earlier versions of the product.

### Syntax

```
minimum ([distinct] expression [auto])
minimum ([distinct] expression for [all|any] expression { , expression })
minimum ([distinct] expression for report)
```

### Example

```
minimum (Sales)
```

Result: Returns the minimum value out of all Sales values.

## moving-average

Returns a moving average by row for a specified set of values over a specified number of rows. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

### Syntax

```
moving-average (numeric_expression , numeric_expression [at expression
{ , expression }] [<for-option>] [prefilter])
moving-average (numeric_expression , numeric_expression [<for-option>]
[prefilter])
<for-option> ::= for expression { , expression } | for report | auto
```

### Example

```
moving-average (Qty , 3)
```

Result: For each row, returns the quantity and a moving average of the current row and the preceding two rows.

| Qty | Moving-Average (Qty, 3) |
|-----|-------------------------|
| 200 | 200                     |
| 700 | 450                     |
| 400 | 433.3333                |
| 200 | 433.3333                |
| 200 | 266.6667                |
| 500 | 300.0000                |

## moving-total

Returns a moving total by row for a specified set of values over a specified number of rows. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

### Syntax

```
moving-total (numeric_expression , numeric_expression [at expression
{ , expression }] [<for-option>] [prefilter])
moving-total (numeric_expression , numeric_expression [<for-option>]
[prefilter])
<for-option> ::= for expression { , expression } |for report|auto
```

### Example

```
moving-total (Qty , 3)
```

Result: For each row, returns the quantity and a moving total of the current row and the preceding two rows.

| Qty | Moving-Total (Qty, 3) |
|-----|-----------------------|
| 200 | 200                   |
| 700 | 900                   |
| 400 | 1300                  |
| 200 | 1300                  |
| 200 | 800                   |
| 500 | 900                   |

## percentage

Returns the percent of the total value for selected data items. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources. This function appears in the Percentage Calculation (by year) interactive sample report.

## Appendix A: Calculation Components

### Syntax

```
percentage (numeric_expression [at expression { , expression }]
[<for-option>] [prefilter])
percentage (numeric_expression [<for-option>] [prefilter])
<for-option> ::= for expression { , expression } |for report|auto
```

### Example

```
percentage (Sales 98)
```

**Result:** Returns the percentage of the total sales for 1998 that is attributed to each sales representative.

| Sales Rep | Sales 98 | Percentage |
|-----------|----------|------------|
| Gibbons   | 60646    | 7.11%      |
| Flertjan  | 62523    | 7.35%      |
| Cornel    | 22396    | 2.63%      |

## percentile

Returns a value, on a scale of one hundred, that indicates the percent of a distribution that is equal to or below the selected data items. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

### Syntax

```
percentile (numeric_expression [at expression { , expression }]
[<for-option>] [prefilter])
percentile (numeric_expression [<for-option>] [prefilter])
<for-option> ::= for expression { , expression } |for report|auto
```

### Example

```
percentile (Sales 98)
```

**Result:** For each row, returns the percentage of rows that are equal to or less than the quantity value of that row.

| Qty | Percentile (Qty) |
|-----|------------------|
| 800 | 1                |
| 700 | 0.875            |
| 600 | 0.75             |
| 500 | 0.625            |
| 400 | 0.5              |
| 400 | 0.5              |
| 200 | 0.25             |
| 200 | 0.25             |

## quantile

Returns the rank of a value within a range that you specify. It returns integers to represent any range of ranks, such as 1 (highest) to 100 (lowest). The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

### Syntax

```
quantile (numeric_expression , numeric_expression [at expression
{ , expression }] [<for-option>] [prefilter])
quantile (numeric_expression , numeric_expression [<for-option>
[prefilter])
<for-option> ::= for expression { , expression } |for report|auto
```

### Example

```
quantile (Qty , 4)
```

**Result:** Returns the quantity, the rank of the quantity value, and the quantity values broken down into 4 quantile groups (quartiles).

| Qty | Rank (Qty) | Quantile (Qty, 4) |
|-----|------------|-------------------|
| 800 | 1          | 1                 |
| 700 | 2          | 1                 |
| 600 | 3          | 2                 |
| 500 | 4          | 2                 |
| 400 | 5          | 3                 |
| 400 | 5          | 3                 |
| 200 | 7          | 4                 |
| 200 | 7          | 4                 |

## quartile

Returns the rank of a value, represented as integers from 1 (highest) to 4 (lowest), relative to a group of values. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

### Syntax

```
quartile (numeric_expression [at expression { , expression }] [<for-option>
[prefilter])
quartile (numeric_expression [<for-option>] [prefilter])
<for-option> ::= for expression { , expression } |for report|auto
```

### Example

```
quartile (Qty)
```

**Result:** Returns the quantity and the quartile of the quantity value represented as integers from 1 (highest) to 4 (lowest).

## Appendix A: Calculation Components

| Qty | Quartile (Qty) |
|-----|----------------|
| 450 | 1              |
| 400 | 1              |
| 350 | 2              |
| 300 | 2              |
| 250 | 3              |
| 200 | 3              |
| 150 | 4              |
| 100 | 4              |

## rank

Returns the rank value of selected data items. The sort order is optional; descending order (DESC) is assumed by default. If two or more rows tie, then there is a gap in the sequence of ranked values (also known as Olympic ranking). The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources. Distinct is an alternative expression that is compatible with earlier versions of the product. Null values are ranked last. This function appears in the Top 10 Retailers for 2005 sample report in the GO Data Warehouse (analysis) package.

### Syntax

```
rank (expression [ASC|DESC] { , expression [ASC|DESC] } [at expression
{ , expression }] [<for-option>] [prefilter])
rank ([distinct] expression [ASC|DESC] { , expression [ASC|DESC] }
[<for-option>] [prefilter])
<for-option> ::= for expression { , expression } |for report|auto
```

### Example

```
rank (Sales 98)
```

Result: For each row, returns the rank value of sales for 1998 that is attributed to each sales representative. Some numbers are skipped when a tie between rows occurs.

| Sales    | Rep | Sales 98 | Rank |
|----------|-----|----------|------|
| Gibbons  |     | 60000    | 1    |
| Flertjan |     | 50000    | 2    |
| Cornel   |     | 50000    | 2    |
| Smith    |     | 48000    | 4    |

## running-average

Returns the running average by row (including the current row) for a set of values. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

### Syntax

```
running-average (numeric_expression [at expression { , expression }]
[<for-option>] [prefilter])
running-average (numeric_expression [<for-option>] [prefilter])
<for-option> ::= for expression { , expression } | for report | auto
```

### Example

```
running-average (Qty)
```

**Result:** For each row, returns the quantity and a running average of the current and the previous rows.

| Name  | Qty | Avg | Running-Average for name |
|-------|-----|-----|--------------------------|
| Smith | 7   | 5   | 7                        |
| Smith | 3   | 5   | 5                        |
| Smith | 6   | 5   | 5.33                     |
| Smith | 4   | 5   | 5                        |
| Wong  | 3   | 4   | 3                        |
| Wong  | 5   | 4   | 4                        |

## running-count

Returns the running count by row (including the current row) for a set of values. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

### Syntax

```
running-count (numeric_expression [at expression { , expression }]
[<for-option>] [prefilter])
running-count (numeric_expression [<for-option>] [prefilter])
<for-option> ::= for expression { , expression } | for report | auto
```

### Example

```
running-count (Qty)
```

**Result:** For each row, returns the quantity and a running count of the position of the current row.

## Appendix A: Calculation Components

| Name  | Qty | Count | Running-Count for name |
|-------|-----|-------|------------------------|
| Smith | 7   | 4     | 1                      |
| Smith | 3   | 4     | 2                      |
| Smith | 6   | 4     | 3                      |
| Smith | 4   | 4     | 4                      |
| Wong  | 3   | 3     | 1                      |
| Wong  | 5   | 3     | 2                      |

## running-difference

Returns a running difference by row, calculated as the difference between the value for the current row and the preceding row, (including the current row) for a set of values. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

### Syntax

```
running-difference (numeric_expression [at expression { , expression }]
[<for-option>] [prefilter])
running-difference (numeric_expression [<for-option>] [prefilter])
<for-option> ::= for expression { , expression } | for report | auto
```

### Example

```
running-difference (Qty)
```

Result: For each row, returns the quantity and a running difference between the value for the current row and the preceding row.

| Name  | Qty | Running-Difference for name |
|-------|-----|-----------------------------|
| Smith | 7   | NULL                        |
| Smith | 3   | -4                          |
| Smith | 6   | 3                           |
| Smith | 4   | -2                          |
| Wong  | 3   | -1                          |
| Wong  | 5   | 2                           |

## running-maximum

Returns the running maximum by row (including the current row) for a set of values. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

## Syntax

```
running-maximum (numeric_expression [at expression { , expression }]
[<for-option>] [prefilter])
running-maximum (numeric_expression [<for-option>] [prefilter])
<for-option> ::= for expression { , expression } |for report|auto
```

## Example

```
running-maximum (Qty)
```

**Result:** For each row, returns the quantity and a running maximum of the current and previous rows.

| Name  | Qty | Max | Running-Maximum (Qty) for name |
|-------|-----|-----|--------------------------------|
| Smith | 2   | 7   | 2                              |
| Smith | 3   | 7   | 3                              |
| Smith | 6   | 7   | 6                              |
| Smith | 7   | 7   | 7                              |
| Wong  | 3   | 5   | 3                              |
| Wong  | 5   | 5   | 5                              |

## running-minimum

Returns the running minimum by row (including the current row) for a set of values. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

## Syntax

```
running-minimum (numeric_expression [at expression { , expression }]
[<for-option>] [prefilter])
running-minimum (numeric_expression [<for-option>] [prefilter])
<for-option> ::= for expression { , expression } |for report|auto
```

## Example

```
running-minimum (Qty)
```

**Result:** For each row, returns the quantity and a running minimum of the current and previous rows.

## Appendix A: Calculation Components

| Name  | Qty | Min | Running-Minimum (Qty) for name |
|-------|-----|-----|--------------------------------|
| Smith | 7   | 2   | 7                              |
| Smith | 3   | 2   | 3                              |
| Smith | 6   | 2   | 3                              |
| Smith | 2   | 2   | 2                              |
| Wong  | 4   | 3   | 4                              |
| Wong  | 5   | 3   | 4                              |

## running-total

Returns a running total by row (including the current row) for a set of values. The "<for-option>" defines the scope of the function. The "at" option defines the level of aggregation and can be used only in the context of relational datasources.

### Syntax

```
running-total (numeric_expression [at expression { , expression }]
[<for-option>] [prefilter])
running-total (numeric_expression [<for-option>] [prefilter])
<for-option> ::= for expression { , expression } | for report | auto
```

### Example

```
running-total (Qty)
```

**Result:** For each row, returns the quantity and a running total of the current and previous rows.

| Name  | Qty | Total | Running-Total (Qty) for name |
|-------|-----|-------|------------------------------|
| Smith | 2   | 18    | 2                            |
| Smith | 3   | 18    | 5                            |
| Smith | 6   | 18    | 11                           |
| Smith | 7   | 18    | 18                           |
| Wong  | 3   | 12    | 3                            |
| Wong  | 5   | 12    | 8                            |

## standard-deviation

Returns the standard deviation of selected data items.

### Syntax

```
standard-deviation (expression [auto])
standard-deviation (expression for [all|any] expression { , expression })
standard-deviation (expression for report)
```

**Example**

```
standard-deviation (ProductCost)
```

**Result:** Returns a value indicating the deviation between product costs and the average product cost.

**standard-deviation-pop**

Computes the population standard deviation and returns the square root of the population variance.

**Syntax**

```
standard-deviation-pop (expression [auto])
standard-deviation-pop (expression for [all|any] expression
{ , expression })
standard-deviation-pop (expression for report)
```

**Example**

```
standard-deviation-pop (ProductCost)
```

**Result:** Returns a value of the square root of the population variance.

**total**

Returns the total value of selected data items. Distinct is an alternative expression that is compatible with earlier versions of the product. This function appears in the Budget vs. Actual sample report in the GO Data Warehouse (analysis) package.

**Syntax**

```
total ([distinct] expression [auto])
total ([distinct] expression for [all|any] expression { , expression })
total ([distinct] expression for report)
```

**Example**

```
total (Sales)
```

**Result:** Returns the total value of all Sales values.

**variance**

Returns the variance of selected data items.

**Syntax**

```
variance (expression [auto])
variance (expression for [all|any] expression { , expression })
variance (expression for report)
```

**Example**

```
variance (Product Cost)
```

**Result:** Returns a value indicating how widely product costs vary from the average product cost.

**variance-pop**

Returns the population variance of a set of numbers after discarding the nulls in this set.

## Appendix A: Calculation Components

### Syntax

```
variance-pop (expression [auto])
variance-pop (expression for [all|any] expression { , expression })
variance-pop (expression for report)
```

### Example

```
variance-pop (Qty)
```

**Result:** For each row, returns the population variance of a set of numbers after discarding the nulls in this set.

## Member Summaries

This list contains predefined functions that return either a single summary value for a set of members or a different summary value for each member of a set of members.

### aggregate

Returns a calculated value using the appropriate aggregation function based on the aggregation type of the expression.

### Syntax

```
aggregate (< currentMeasure|numeric_expression > within set set_expression)
aggregate (< currentMeasure|numeric_expression > within < detail|aggregate >
expression)
```

### average

Returns the average value of the selected data items.

### Syntax

```
average (< currentMeasure|numeric_expression > within set set_expression)
average (< currentMeasure|numeric_expression > within < detail|aggregate >
expression)
```

### Example

```
average (Sales)
```

**Result:** Returns the average of all Sales values.

### count

Returns the number of selected data items excluding null values.

### Syntax

```
count (< currentMeasure|numeric_expression > within set set_expression)
count (< currentMeasure|numeric_expression > within < detail|aggregate >
expression)
```

### Example

```
count (Sales)
```

**Result:** Returns the total number of entries under Sales.

## maximum

Returns the maximum value of selected data items.

### Syntax

```
maximum (< currentMeasure|numeric_expression > within set set_expression)
maximum (< currentMeasure|numeric_expression > within < detail|aggregate >
expression)
```

### Example

```
maximum (Sales)
```

**Result:** Returns the maximum value out of all Sales values.

## median

Returns the median value of selected data items.

### Syntax

```
median (< currentMeasure|numeric_expression > within set set_expression)
median (< currentMeasure|numeric_expression > within < detail|aggregate >
expression)
```

## minimum

Returns the minimum value of selected data items.

### Syntax

```
minimum (< currentMeasure|numeric_expression > within set set_expression)
minimum (< currentMeasure|numeric_expression > within < detail|aggregate >
expression)
```

### Example

```
minimum (Sales)
```

**Result:** Returns the minimum value out of all Sales values.

## percentage

Returns the percent of the total value for the selected data items.

### Syntax

```
percentage (numeric_expression [tuple member_expression { , member_
expression }] within set set_expression)
```

### Example

```
percentage ([gosales].[sales measures].[quantity] tuple [gosales].[Staff].[].
[department] -> [West] within set children ([gosales].[Staff].[].[Staff]))
```

## percentile

Returns a value, on a scale from 0 to 100, that indicates the percent of a distribution that is equal to or below the selected data items.

### Syntax

```
percentile (numeric_expression [tuple member_expression { , member_
expression }] within set set_expression)
```

## quantile

Returns the rank of a value for the specified range. It returns integers to represent any range of ranks, such as 1 (highest) to 100 (lowest).

### Syntax

```
quantile (numeric_expression , numeric_expression [tuple member_expression
{ , member_expression }] within set set_expression)
```

## quartile

Returns the rank of a value, represented as integers from 1 (highest) to 4 (lowest), relative to a group of values.

### Syntax

```
quartile (numeric_expression [tuple member_expression { , member_expression }]
within set set_expression)
```

## rank

Returns the rank value of the selected data items. The type of ranking returned (Olympic, dense, or serial) is data source dependent. The sort order is optional; DESC is assumed by default.

### Syntax

```
rank (numeric_expression [ASC|DESC] [tuple member_expression { , member_
expression }] within set set_expression)
```

### Example

```
rank ([gosales].[sales measures].[quantity] tuple [gosales].[Staff].[] .
[department] -> [West] within set children ([gosales].[Staff].[].[Staff]))
```

## standard-deviation

Returns the standard deviation of the selected data items.

### Syntax

```
standard-deviation (< currentMeasure|numeric_expression > within set
set_expression)
standard-deviation (< currentMeasure|numeric_expression > within <
detail|aggregate > expression)
```

## standard-deviation-pop

Returns the standard deviation population of the selected data items.

### Syntax

```
standard-deviation-pop (< currentMeasure|numeric_expression > within set
set_expression)
```

```
standard-deviation-pop (< currentMeasure|numeric_expression > within < detail|aggregate > expression)
```

## total

Returns the total value of the selected data items.

### Syntax

```
total (< currentMeasure|numeric_expression > within set set_expression)
total (< currentMeasure|numeric_expression > within < detail|aggregate > expression)
```

## variance

Returns the variance of the selected data items.

### Syntax

```
variance (< currentMeasure|numeric_expression > within set set_expression)
variance (< currentMeasure|numeric_expression > within < detail|aggregate > expression)
```

## variance-pop

Returns the variance population of the selected data items.

### Syntax

```
variance-pop (< currentMeasure|numeric_expression > within set set_expression)
variance-pop (< currentMeasure|numeric_expression > within < detail|aggregate > expression)
```

## Constants

A constant is a fixed value that you can use in an expression.

## date

Inserts the current system date.

## date-time

Inserts the current system date and time.

## time with time zone

Inserts a zero time with time zone.

## timestamp with time zone

Inserts an example of a timestamp with time zone.

## **interval**

Inserts a zero interval: 000 00:00:00.000.

## **interval year**

Inserts a zero year interval: 0 year.

## **interval month**

Inserts a zero month interval: 0 month.

## **interval year to month**

Inserts a zero year to month interval: 0000-00 year to month.

## **interval day**

Inserts a zero day interval: 0 day.

## **interval hour**

Inserts a zero hour interval: 0 hour.

## **interval minute**

Inserts a zero minute interval: 0 minute.

## **interval second**

Inserts a zero second interval: 0 second.

## **interval day to hour**

Inserts a zero day to hour interval: 0 00 day to hour.

## **interval day to minute**

Inserts a zero day to minute interval: 0 00:00 day to minute.

## **interval day to second**

Inserts a zero day to second interval: 0 00:00:00.000000000 day to second.

## **interval hour to minute**

Inserts a zero hour to minute interval: 00:00 hour to minute.

## **interval hour to second**

Inserts a zero hour to second interval: 00:00:00.000000000 hour to second.

## interval minute to second

Inserts a zero minute to second interval: 00:00.000000000 minute to second.

## null

Inserts "null" if the expression conditions are not met.

## number

Inserts the number 0, which can be replaced with a new numeric value.

## string

Inserts an empty string as two single quotation marks between which you can type a string.

## time

Inserts the current system time.

# Constructs

This list contains constructs and templates that can be used to create an expression. Templates combine multiple functions into a group. For example, the search case template includes the case, when, else, and end functions.

## if then else

This construct is the template for an if...then...else statement. This construct appears in the Top 10 Retailers for 2005 sample report in the GO Data Warehouse (analysis) package.

### Syntax

```
IF ([Country] = 'Canada') THEN ([List Price] * 0.60) ELSE ([List Price])
```

## in\_range

This is the template for an in\_range expression.

### Syntax

```
[code] IN_RANGE { :30 , 40, 50, 999: }
```

### Example 1

```
[code] IN_RANGE { 5 }
```

Result: This is equivalent to [code] = 5.

### Example 2

```
[code] IN_RANGE { 5: }
```

Result: This is equivalent to [code] >= 5.

## Appendix A: Calculation Components

### Example 3

```
[code] IN_RANGE { :5 }
```

Result: This is equivalent to [code] <= 5.

### Example 4

```
[code] IN_RANGE { 5:10 }
```

Result: This is equivalent to ( [code] >= 5 and [code] <= 10 ).

### Example 5

```
[code] IN_RANGE { :5,10,20: }
```

Result: This is equivalent to ( [code] <= 5 or [code] = 10 or [code] >= 20 ).

## search case

This construct is the template for a search case, including the case, when, else, and end functions.

### Syntax

```
CASE WHEN [Country] = 'Canada' THEN ([List Price] * 0.60) WHEN [CountryCode] > 100 THEN [List Price] * 0.80 ELSE [List Price] END
```

## simple case

This construct is the template for a simple case, including the case, when, else, and end functions.

### Syntax

```
CASE [Country] WHEN 'Canada' THEN ([List Price] * 0.60) WHEN 'Australia' THEN [List Price] * 0.80 ELSE [List Price] END
```

## Business Date/Time Functions

This list contains business functions for performing date and time calculations.

### \_add\_days

Returns the date or datetime, depending on the format of "date\_expression", that results from adding "integer\_expression" days to "date\_expression".

### Syntax

```
_add_days (date_expression, integer_expression)
```

### Example 1

```
_add_days (2002-04-30 , 1)
```

Result: 2002-05-01

### Example 2

```
_add_days (2002-04-30 12:10:10.000, 1)
```

Result: 2002-05-01 12:10:10.000

### **Example 3**

```
_add_days (2002-04-30 00:00:00.000, 1/24)
```

Note that the second argument is not a whole number. This is supported by some database technologies and increments the time portion.

Result: 2002-04-30 01:00:00.000

## **\_add\_months**

Returns the date or datetime, depending on the format of "date\_expression", that results from the addition of "integer\_expression" months to "date\_expression".

### **Syntax**

```
_add_months (date_expression, integer_expression)
```

### **Example 1**

```
_add_months (2002-04-30 , 1)
```

Result: 2002-05-30

### **Example 2**

```
_add_months (2002-04-30 12:10:10.000, 1)
```

Result: 2002-05-30 12:10:10.000

## **\_add\_years**

Returns the date or datetime, depending on the format of "date\_expression", that results from the addition of "integer\_expression" years to "date\_expression".

### **Syntax**

```
_add_years (date_expression, integer_expression)
```

### **Example 1**

```
_add_years (2002-04-30 , 1)
```

Result: 2003-04-30

### **Example 2**

```
_add_years (2002-04-30 12:10:10.000 , 1)
```

Result: 2003-04-30 12:10:10.000

## **\_age**

Returns a number that is obtained from subtracting "date\_expression" from today's date. The returned value has the form YYYYMMDD, where YYYY represents the number of years, MM represents the number of months, and DD represents the number of days.

## Appendix A: Calculation Components

### **Syntax**

```
_age (date_expression)
```

### **Example**

```
_age (1990-04-30) (if today's date is 2003-02-05)
```

Result: 120906, meaning 12 years, 9 months, and 6 days.

## **\_day\_of\_week**

Returns the day of week (1 to 7), where 1 is the first day of the week as indicated by the second parameter (1 to 7, 1 being Monday and 7 being Sunday). Note that in ISO 8601 standard, a week begins with Monday being day 1.

### **Syntax**

```
_day_of_week (date_expression, integer)
```

### **Example**

```
_day_of_week (2003-01-01 , 1)
```

Result: 3

## **\_day\_of\_year**

Returns the day of year (1 to 366) in "date\_expression". Also known as Julian day.

### **Syntax**

```
_day_of_year (date_expression)
```

### **Example**

```
_day_of_year (2003-03-01)
```

Result: 61

## **\_days\_between**

Returns a positive or negative number representing the number of days between "date\_expression1" and "date\_expression2". If "date\_expression1" < "date\_expression2", then the result will be a negative number.

### **Syntax**

```
_days_between (date_expression1 , date_expression2)
```

### **Example**

```
_days_between (2002-04-30 , 2002-06-21)
```

Result: -52

## **\_days\_to\_end\_of\_month**

Returns a number representing the number of days remaining in the month represented by "date\_expression".

**Syntax**

```
_days_to_end_of_month (date_expression)
```

**Example**

```
_days_to_end_of_month (2002-04-20 14:30:22.123)
```

Result: 10

**\_first\_of\_month**

Returns a date or datetime, depending on the argument, by converting "date\_expression" to a date with the same year and month but with the day set to 1.

**Syntax**

```
_first_of_month (date_expression)
```

**Example 1**

```
_first_of_month (2002-04-20)
```

Result: 2002-04-01

**Example 2**

```
_first_of_month (2002-04-20 12:10:10.000)
```

Result: 2002-04-01 12:10:10.000

**\_last\_of\_month**

Returns a date or datetime, depending on the argument, that is the last day of the month represented by "date\_expression".

**Syntax**

```
_last_of_month (date_expression)
```

**Example 1**

```
_last_of_month (2002-01-14)
```

Result: 2002-01-31

**Example 2**

```
_last_of_month (2002-01-14 12:10:10.000)
```

Result: 2002-01-31 12:10:10.000

**\_make\_timestamp**

Returns a timestamp constructed from "integer\_expression1" (the year), "integer\_expression2" (the month), and "integer\_expression3" (the day). The time portion defaults to 00:00:00.000 .

**Syntax**

```
_make_timestamp (integer_expression1, integer_expression2, integer_
expression3)
```

### **Example**

```
_make_timestamp (2002 , 01 , 14)
```

Result: 2002-01-14 00:00:00.000

## **\_months\_between**

Returns a positive or negative integer number representing the number of months between "date\_expression1" and "date\_expression2". If "date\_expression1" is earlier than "date\_expression2", then a negative number is returned.

### **Syntax**

```
_months_between (date_expression1, date_expression2)
```

### **Example**

```
_months_between (2002-04-03 , 2002-01-30)
```

Result: 2

## **\_week\_of\_year**

Returns the number of the week of the year of "date\_expression" according to the ISO 8601 standard. Week 1 of the year is the first week of the year to contain a Thursday, which is equivalent to the first week containing January 4th. A week starts on Monday (day 1) and ends on Sunday (day 7).

### **Syntax**

```
_week_of_year (date_expression)
```

### **Example**

```
_week_of_year (2003-01-01)
```

Result: 1

## **\_years\_between**

Returns a positive or negative integer number representing the number of years between "date\_expression1" and "date\_expression2". If "date\_expression1" < "date\_expression2" then a negative value is returned.

### **Syntax**

```
_years_between (date_expression1, date_expression2)
```

### **Example**

```
_years_between (2003-01-30 , 2001-04-03)
```

Result: 1

## ymdint\_between

Returns a number representing the difference between "date\_expression1" and "date\_expression2". The returned value has the form YYYYMMDD, where YYYY represents the number of years, MM represents the number of months, and DD represents the number of days.

### Syntax

```
_ymdint_between (date_expression1 , date_expression2)
```

### Example

```
_ymdint_between (1990-04-30 , 2003-02-05)
```

Result: 120906, meaning 12 years, 9 months and 6 days.

## Block Functions

This list contains functions used to access members of a set, usually in the context of Analysis Studio.

## firstFromSet

Returns the first members found in the set up to "numeric\_expression\_maximum" + "numeric\_expression\_overflow". If "numeric\_expression\_maximum" + "numeric\_expression\_overflow" is exceeded, then only the maximum number of members are returned. For a set that has only a few members more than the specified numeric\_expression\_maximum, the numeric\_expression\_overflow allows the small set of extra members to be included. If the set has more members than the overflow allows, then only the numeric\_expression\_maximum members will be returned.

### Syntax

```
_firstFromSet (set_expression , numeric_expression_maximum , numeric_expression_overflow)
```

### Example 1

```
_firstFromSet ([great_outdoors_company].[Products].[Products].[Product line] , 2 , 8)
```

Result: Returns the five members in the Product line set. The first two members are returned within the maximum and the following three members are returned as the overflow.

```
Camping Equipment
Golf Equipment
Mountaineering Equipment
Outdoor Protection
Personal Accessories
```

### Example 2

```
_firstFromSet ([great_outdoors_company].[Products].[Products].[Product line] , 2 , 2)
```

Result: Camping Equipment, Golf Equipment

## \_remainderSet

Returns the set containing "member\_expression" when the size of "set\_expression" is greater than "numeric\_expression"; i.e., a new member will be generated if the number of members in "set\_expression" is larger than the specified "numeric\_expression".

### Syntax

```
_remainderSet (member_expression, set_expression , numeric_expression)
```

### Example

```
_remainderSet (member (aggregate (currentMeasure within set [great_outdoors_company].[Products].[Products].[Product line]) , 'Product Aggregate' , 'Product Aggregate' , [great_outdoors_company].[Products].[Products]) , [great_outdoors_company].[Products].[Products].[Product line] , 1)
```

Result: Quantity sold for Product Aggregate

## Macro Functions

This list contains functions that can be used within a macro. A macro may contain one or more macro functions. A macro is delimited by a number sign (#) at the beginning and at the end. Everything between the number signs is treated as a macro expression and is executed at run time. For macro functions that accept expressions of datatype timestamp with time zone as arguments, the accepted format is 'yyyy-mm-dd hh:mm:ss[.ff]+hh:mm' where fractional seconds are optional and can be represented by 1 to 9 digits. In lieu of a space separating the date portion to the time portion, the character 'T' is also accepted. Also, in lieu of the time zone '+hh:mm', the character 'Z' is accepted and will be processed internally as '+00:00'. The macro functions that return expressions of datatype timestamp with time zone return 9 digits by default for their fractional seconds. The macro function timestampMask () can be used to trim the output if required.

+

Concatenates two strings.

### Syntax

```
value1 + value2
```

### Example

```
'{ ' + $runLocale + ' }'
```

Result: {en-us}

## \_add\_days

Returns the timestamp with time zone (as a string) that results from adding "integer\_expression" number of days to "string\_expression", where "string\_expression" represents a timestamp with time zone.

### Syntax

```
_add_days (string_expression , integer_expression)
```

**Example 1**

```
_add_days ('2005-11-01 12:00:00.000-05:00' , -1)
```

Result: 2005-10-31 12:00:00.000000000-05:00

**Example 2**

```
_add_days ($current_timestamp , 1)
```

Result: 2005-11-02 12:00:00.000000000-05:00

**Example 3**

```
timestampMask (_add_days ($current_timestamp , 1) , 'yyyy-mm-dd')
```

Result: 2005-11-02

**\_add\_months**

Returns the timestamp with time zone (as a string) that results from adding "integer\_expression" number of months to "string\_expression", where "string\_expression" represents a timestamp with time zone.

**Syntax**

```
_add_months (string_expression , integer_expression)
```

**Example 1**

```
_add_months ('2005-11-01 12:00:00.000-05:00' , -1)
```

Result: 2005-10-01 12:00:00.000000000-05:00

**Example 2**

```
_add_months ($current_timestamp , 1)
```

Result: 2005-12-01 12:00:00.000000000-05:00

**Example 3**

```
timestampMask (_add_months ($current_timestamp , 1) , 'yyyy-mm-dd')
```

Result: 2005-12-01

**\_add\_years**

Returns the timestamp with time zone (as a string) that results from adding "integer\_expression" number of years to "string\_expression", where "string\_expression" represents a timestamp with time zone.

**Syntax**

```
_add_years (string_expression , integer_expression)
```

**Example 1**

```
_add_years ('2005-11-01 12:00:00.000-05:00' , -1)
```

Result: 2004-11-01 12:00:00.000000000-05:00

## Appendix A: Calculation Components

### Example 2

```
_add_years ($current_timestamp , 1)
```

Result: 2006-11-01 12:00:00.000000000-05:00

### Example 3

```
timestampMask (_add_years ($current_timestamp , 1) , 'yyyy-mm-dd')
```

Result: 2006-11-01

## array

Constructs an array out of the list of parameters.

### Syntax

```
array (string_expression|array_expression { , string_expression|array_expression })
```

### Example

```
csv (array ('a1' , array ('x1' , 'x2') , 'a2'))
```

Result: 'a1','x1','x2','a2'

## csv

Constructs a string from the elements of the array where the values are separated by commas.

Optionally, the separator and quote strings can be specified. The default separator is a comma ( , ) and the default quote character is a single quote ( ' ).

### Syntax

```
csv (array_expression [, separator_string [, quote_string]])
```

### Example

```
csv (array ('a1' , 'a2'))
```

Result: 'a1','a2'

## dq

Surrounds "string\_expression" with double quotes.

### Syntax

```
dq (string_expression)
```

### Example

```
dq ('zero')
```

Result: "zero"

## getConfigurationEntry

Get an entry from the IBM® Cognos® configuration file. The force\_decode\_flag is optional and must be one of: 'true' , '1' , 1 , 'false' , '0' , 0. The default is 'true'. When true, the value of the configuration entry will be decrypted into plain text if it is encrypted.

### Syntax

```
getConfigurationEntry (entry_string , force_decode_flag)
```

### Example

```
getConfigurationEntry ('serverLocale')
```

Result: en

## grep

Searches for and returns elements of an array that match the pattern specified in "pattern\_string".

### Syntax

```
grep (pattern_string , array_expression)
```

### Example

```
csv (grep ('s' , array ('as' , 'an' , 'arts')))
```

Result: 'as', 'arts'

## \_first\_of\_month

Returns a timestamp with time zone (as a string) by converting the day value in "string\_expression" to 1, where "string\_expression" is a timestamp with time zone.

### Syntax

```
_first_of_month (string_expression)
```

### Example 1

```
_first_of_month ('2005-11-11 12:00:00.000-05:00')
```

Result: 2005-11-01 12:00:00.000000000-05:00

### Example 2

```
timestampMask (_first_of_month ('2005-11-11 12:00:00.000-05:00') , 'yyyymmdd')
```

Result: 20051101

## \_last\_of\_month

Returns a timestamp with time zone (as a string) that is the last day of the month represented by "string\_expression", where "string\_expression" is a timestamp with time zone.

### Syntax

```
_last_of_month (string_expression)
```

### Example 1

```
_last_of_month ('2005-11-11 12:00:00.000-05:00')
```

Result: 2005-11-30 12:00:00.000000000-05:00

### Example 2

```
timestampMask (_last_of_month ('2005-11-11 12:00:00.000-05:00') , 'yyyy-mm-dd')
```

Result: 2005-11-30

## join

Joins the elements of an array using "separator\_string".

### Syntax

```
join (separator_string , array_expression)
```

### Example

```
sq (join (' || ' , array ('as' , 'an' , 'arts')))
```

Result: 'as || an || arts'

## prompt

Prompts the user for a single value. Only "prompt\_name" is required. The datatype defaults to "string" when it is not specified. The prompt is optional when "defaultText" is specified. "Text", when specified, will precede the value. "QueryItem" can be specified to take advantage of the prompt information properties of "queryItem". "Trailing\_text", when specified, will be appended to the value.

### Syntax

```
prompt (prompt_name , datatype , defaultText , text , queryItem , trailing_text)
```

### Example 1

```
select . . . where COUNTRY_MULTILINGUAL.COUNTRY_CODE > #prompt ('Starting CountryCode' , 'integer' , '10') #
```

Result: select . . . where COUNTRY\_MULTILINGUAL.COUNTRY\_CODE > 10

### Example 2

```
[gosales].[COUNTRY].[COUNTRY] = #prompt ('countryPrompt' , 'string' , '''Canada''') #
```

Result: [gosales].[COUNTRY].[COUNTRY] = 'Canada'

### Notes

- The "defaultText" parameter must be specified such that it is literally valid in the context of the macro since no formatting takes place on this value. The default string ""Canada"" in Example 2 is specified as a string using single quotes, in which the embedded single quotes are doubled up, thus 3 quotes. This results in the string being properly displayed within single

quotes in the expression. As a general rule for the string datatype, "defaultText" should always be specified like this, except in the context of a stored procedure parameter. For "defaultText" of types 'date' or 'datetime', a special format should be used in the context of SQL. Examples of these formats are 'DATE "2001-12-25"' and 'DATETIME "2001-12-25 12:00:00"'. In all other contexts, use the date/datetime without the keyword and escaped single quotes (e.g., '2001-12-25').

## **promptmany**

Prompts the user for one or more values. Only "prompt\_name" is required. The datatype defaults to string when it is not specified. The prompt is optional when "defaultText" is specified. "Text", when specified, will precede the list of values. "QueryItem" can be specified to take advantage of the prompt information properties of "queryItem". "Trailing\_text", when specified, will be appended to the list of values.

### **Syntax**

```
promptmany (prompt_name , datatype , defaultText , text , queryItem , trailing_text)
```

#### **Example 1**

```
select . . . where COUNTRY_MULTILINGUAL.COUNTRY in (# promptmany
('CountryName') #)
```

Result: select . . . where COUNTRY\_MULTILINGUAL.COUNTRY\_CODE in ( 'Canada' , 'The Netherlands' , 'Russia' )

#### **Example 2**

```
select . . . from gosales.gosales.dbo.COUNTRY_MULTILINGUAL COUNTRY_
MULTILINGUAL , gosales.gosales.dbo.COUNTRY XX where COUNTRY_MULTILINGUAL.COUNTRY_
CODE = XX.COUNTRY_CODE # promptmany ('Selected CountryCodes' , 'integer' , '
', ' and COUNTRY_MULTILINGUAL.COUNTRY_CODE in (' , ' , ') ') #
```

Result: select . . . from gosales.gosales.dbo.COUNTRY\_MULTILINGUAL COUNTRY\_MULTILINGUAL , gosales.gosales.dbo.COUNTRY XX where COUNTRY\_MULTILINGUAL.COUNTRY\_CODE = XX.COUNTRY\_CODE and COUNTRY\_MULTILINGUAL.COUNTRY\_CODE in ( 'Canada' , 'The Netherlands' , 'Russia' )

## **sb**

Surrounds "string\_expression" with square brackets.

### **Syntax**

```
sb (string_expression)
```

#### **Example**

```
sb ('abc')
```

Result: [abc]

## Appendix A: Calculation Components

### sq

Surrounds "string\_expression" with single quotes.

#### Syntax

```
sq (string_expression)
```

#### Example

```
sq ('zero')
```

Result: 'zero'

### sort

Sorts the elements of the array in alphabetical order. Duplicates are retained.

#### Syntax

```
sort (array_expression)
```

#### Example

```
csv (sort (array ('s3', 'a', 'x')))
```

Result: 'a', 's3', 'x'

### split

Splits a string or string elements of the array into separate elements.

#### Syntax

```
split (pattern_string, string_expression|array_expression)
```

#### Example 1

```
csv (split ('::', 'ab=c::de=f::gh=i'))
```

Result: 'ab=c', 'de=f', 'gh=i'

#### Example 2

```
csv (split ('=' , split ('::', 'ab=c::de=f::gh=i')))
```

Result: 'ab', 'c', 'de', 'f', 'gh', 'i'

### substitute

Searches for a pattern in a string or in the string elements of an array and substitutes the first occurrence of "pattern\_string" with "replacement\_string".

#### Syntax

```
substitute (pattern_string, replacement_string, string_expression|array_expression)
```

#### Example 1

```
#sq (substitute ('^cn=', '***', 'cn=help')) #
```

Result: '\*\*\*help'

### Example 2

```
csv (substitute ('^cn=' , '***' , array ('cn=help' , 'acn=5'))) #
Result: '***help', 'acn=5'
```

### Example 3

```
csv (substitute ('cn=' , '' , array ('cn=help' , 'acn=5'))) #
Result: 'help', 'a5'
```

## timestampMask

Returns "string\_expression1", representing a timestamp with time zone, trimmed to the format specified in "string\_expression2". The format in "string\_expression2" must be one of the following: 'yyyy', 'mm', 'dd', 'yyyy-mm', 'yyyymm', 'yyyy-mm-dd', 'yyyymmdd', 'yyyy-mm-dd hh:mm:ss', 'yyyy-mm-dd hh:mm:ss+hh:mm', 'yyyy-mm-dd hh:mm:ss.ff3', 'yyyy-mm-dd hh:mm:ss.ff3+hh:mm', 'yyyy-mm-ddThh:mm:ss', 'yyyy-mm-ddThh:mm:ss+hh:mm', 'yyyy-mm-ddThh:mm:ss.ss', 'yyyy-mm-ddThh:mm:ss.ss+hh:mm', or 'yyyy-mm-ddThh:mm:ss.ss.ff3+hh:mm'. The macro functions that return a string representation of a timestamp with time zone show a precision of 9 digits for the fractional part of the seconds by default. The format options allow this to be trimmed down to a precision of 3 or 0.

### Syntax

```
timestampMask (string_expression1 , string_expression2)
```

### Example 1

```
timestampMask ($current_timestamp , 'yyyy-dd-mm') #
Result: 2005-11-01
```

### Example 2

```
timestampMask ('2005-11-01 12:00:00.000-05:00' , 'yyyy-mm-dd
hh:mm:ss+hh:mm') #
Result: 2005-11-01 12:00:00-05:00
```

### Example 3

```
timestampMask ('2005-11-01 12:00:00.123456789-05:00' , 'yyyy-mm-ddThh:mm:
ss+hh:mm.ff3+hh:mm') #
Result: 2005-11-01T12:00:00.123-05:00
```

## toLocal

Returns the string representing a timestamp with time zone resulting from adjusting "string\_expression" to the time zone of the operating system. Note that the macro function timestampMask () can be used to trim the output.

### Syntax

```
toLocal (string_expression)
```

## Appendix A: Calculation Components

### Example 1

```
toLocal ('2005-11-01 17:00:00.000-00:00') # where OS local time zone is -05:00
```

Result: 2005-11-01 12:00:00.000000000-05:00

### Example 2

```
timestampMask (toLocal ('2005-11-01 17:00:00.000-00:00') , 'yyyy-mm-dd hh:mm:ss+hh:mm') # where OS local time zone is -05:00
```

Result: 2005-11-01 12:00:00-05:00

### Example 3

```
toLocal ('2005-11-01 13:30:00.000-03:30') # where OS local time zone is -05:00
```

Result: 2005-11-01 12:00:00.000000000-05:00

## toUTC

Returns the string representing a timestamp with time zone resulting from adjusting "string\_expression" to the zero-point reference UTC time zone, also known as GMT time. Note that the macro function timestampMask () can be used to trim the output.

### Syntax

```
toUTC (string_expression)
```

### Example 1

```
toUTC ('2005-11-01 12:00:00.000-05:00')
```

Result: 2005-11-01 17:00:00.000000000-00:00

### Example 2

```
timestampMask(toUTC ('2005-11-01 12:00:00.000-05:00') , 'yyyy-mm-dd hh:mm:ss.ff3+hh:mm')
```

Result: 2005-11-01 17:00:00.000-00:00

### Example 3

```
toUTC ($current_timestamp)
```

Result: 2005-11-01 17:00:00.000000000-00:00

## unique

Removes duplicate entries from the array. The order of the elements is retained.

### Syntax

```
unique (array_expression)
```

### Example

```
csv (unique (array ('s3', 'a', 's3', 'x')))
```

Result: 's3', 'a', 'x'

## urlencode

URL-encodes the passed argument. This function is useful when specifying XML connection strings.

### Syntax

```
urlencode (prompt ('userValue'))
```

### Example

```
urlencode (prompt ('some_val'))
```

Result: %27testValue%27

## CSVIdentityName

Uses the identity information of the current authenticated user to look up values in the specified parameter map. Each individual piece of the user's identity (account name, group names, role names) is used as a key into the map. The unique list of values that is retrieved from the parameter map is then returned as a string, where each value is surrounded by single quotes and where multiple values are separated by commas.

### Syntax

```
CSVIdentityName (%parameter_map_name [, separator_string])
```

### Example

```
CSVIdentityName (%security_clearance_level_map)
```

Result: 'level\_500', 'level\_501', 'level\_700'

## CSVIdentityNameList

Returns the pieces of the user's identity (account name, group names, role names) as a list of strings. The unique list of values is returned as a string, where each value is surrounded by single quotes and where multiple values are separated by commas.

### Syntax

```
CSVIdentityNameList ([separator_string])
```

### Example

```
CSVIdentityNameList ()
```

Result: 'Everyone', 'Report Administrators', 'Query User'

## CAMPassport

Returns the Cognos® Access Manager passport.

### Syntax

```
CAMPassport ()
```

### Example

```
CAMPassport ()
```

## Appendix A: Calculation Components

Result: 111:98812d62-4fd4-037b-4354-26414cf7ebef:3677162321

### CAMIDList

Returns the pieces of the user's Cognos® Access Manager ID (CAMID), such as account name, group names, or role names, as a list of values separated by commas.

#### Syntax

```
CAMIDList ([separator_string])
```

#### Example

```
#CAMIDList () #
```

Result: CAMID ( "::Everyone" ), CAMID ( ":Authors" ), CAMID ( ":Query Users" ), CAMID ( ":Consumers" ), CAMID ( ":Metrics Authors" )

### CAMIDListForType

Returns an array of the user's Cognos® Access Manager IDs (CAMIDs) based on the identity type (account, group, or role). CAMIDListForType can be used with the macro functions csv or join.

#### Syntax

```
CAMIDListForType (identity_type)
```

#### Example

```
[qs].[userRole] in (# csv (CAMIDListForType ('role')) #)
```

Result: [qs].[userRole] in ( 'CAMID ( "::System Administrators" )' , 'CAMID ( ":Authors" )' )

## Common Functions

### abs

Returns the absolute value of "numeric\_expression". Negative values are returned as positive values.

#### Syntax

```
abs (numeric_expression)
```

#### Example 1

```
abs (15)
```

Result: 15

#### Example 2

```
abs (-15)
```

Result: 15

## cast

Converts "expression" to a specified data type. Some data types allow for a length and precision to be specified. Make sure that the target is of the appropriate type and size. The following can be used for "datatype\_specification": character, varchar, char, numeric, decimal, integer, smallint, real, float, date, time, timestamp, time with time zone, timestamp with time zone, and interval. When type casting to an interval type, one of the following interval qualifiers must be specified: year, month, or year to month for the year-to-month interval datatype; day, hour, minute, second, day to hour, day to minute, day to second, hour to minute, hour to second, or minute to second for the day-to-second interval datatype. Notes: When you convert a value of type timestamp to type date, the time portion of the timestamp value is ignored. When you convert a value of type timestamp to type time, the date portion of the timestamp is ignored. When you convert a value of type date to type timestamp, the time components of the timestamp are set to zero. When you convert a value of type time to type timestamp, the date component is set to the current system date. It is invalid to convert one interval datatype to the other (for instance because the number of days in a month is variable). Note that you can specify the number of digits for the leading qualifier only, i.e. YEAR(4) TO MONTH, DAY(5). Errors will be reported if the target type and size are not compatible with the source type and size.

### Syntax

```
cast (expression , datatype_specification)
```

### Example 1

```
cast ('123' , integer)
```

Result: 123

### Example 2

```
cast (12345 , varchar (10))
```

Result: a string containing 12345

## ceil

Returns the smallest integer that is greater than or equal to "numeric\_expression".

### Syntax

```
ceil (numeric_expression)
```

## ceiling

Returns the smallest integer that is greater than or equal to "numeric\_expression".

### Syntax

```
ceiling (numeric_expression)
```

### Example 1

```
ceiling (4.22)
```

Result: 5

### **Example 2**

```
ceiling (-1.23)
```

Result: -1

## **char\_length**

Returns the number of logical characters in "string\_expression". The number of logical characters can be distinct from the number of bytes in some East Asian locales.

### **Syntax**

```
char_length (string_expression)
```

### **Example**

```
char_length ('Canada')
```

Result: 6

## **character\_length**

Returns the number of characters in "string\_expression".

### **Syntax**

```
character_length (string_expression)
```

### **Example**

```
character_length ('Canada')
```

Result: 6

## **coalesce**

Returns the first non-null argument (or null if all arguments are null). Requires two or more arguments in "expression\_list".

### **Syntax**

```
coalesce (expression_list)
```

### **Example**

```
coalesce ([Unit price], [Unit sale price])
```

Result: Returns the unit price, or the unit sale price if the unit price is null.

## **current\_date**

Returns a date value representing the current date of the computer that the database software runs on.

### **Syntax**

```
current_date
```

**Example**

```
current_date
```

Result: 2003-03-04

**current\_time**

Returns a time with time zone value, representing the current time of the computer that runs the database software if the database supports this function. Otherwise, it represents the current time of the computer that runs IBM® Cognos® BI software.

**Syntax**

```
current_time
```

**Example**

```
current_time
```

Result: 16:33:11+05:00

**current\_timestamp**

Returns a datetime with time zone value, representing the current time of the computer that runs the database software if the database supports this function. Otherwise, it represents the current time of the computer that runs IBM® Cognos® BI software.

**Syntax**

```
current_timestamp
```

**Example**

```
current_timestamp
```

Result: 2003-03-03 16:40:15.535000+05:00

**exp**

Returns 'e' raised to the power of "numeric\_expression". The constant 'e' is the base of the natural logarithm.

**Syntax**

```
exp (numeric_expression)
```

**Example**

```
exp (2)
```

Result: 7.389056

**extract**

Returns an integer representing the value of datepart (year, month, day, hour, minute, second) in "datetime\_expression".

## Appendix A: Calculation Components

### Syntax

```
extract (datepart , datetime_expression)
```

#### Example 1

```
extract (year , 2003-03-03 16:40:15.535)
```

Result: 2003

#### Example 2

```
extract (hour , 2003-03-03 16:40:15.535)
```

Result: 16

## floor

Returns the largest integer that is less than or equal to "numeric\_expression".

### Syntax

```
floor (numeric_expression)
```

#### Example 1

```
floor (3.22)
```

Result: 3

#### Example 2

```
floor (-1.23)
```

Result: -2

## ln

Returns the natural logarithm of "numeric\_expression".

### Syntax

```
ln (numeric_expression)
```

#### Example

```
ln (4)
```

Result: 1.38629

## localtime

Returns a time value, representing the current time of the computer that runs the database software.

### Syntax

```
localtime
```

#### Example

```
localtime
```

Result: 16:33:11

## localtimestamp

Returns a datetime value, representing the current timestamp of the computer that runs the database software.

### Syntax

```
localtimestamp
```

### Example

```
localtimestamp
```

Result: 2003-03-03 16:40:15.535000

## lower

Returns "string\_expression" with all uppercase characters shifted to lowercase.

### Syntax

```
lower (string_expression)
```

### Example

```
lower ('ABCDEF')
```

Result: abcdef

## mod

Returns the remainder (modulus) of "integer\_expression1" divided by "integer\_expression2". "Integer\_expression2" must not be zero or an exception condition is raised.

### Syntax

```
mod (integer_expression1, integer_expression2)
```

### Example

```
mod (20 , 3)
```

Result: 2

## nullif

Returns null if "expression1" equals "expression2", otherwise returns "expression1".

### Syntax

```
nullif (expression1, expression2)
```

## octet\_length

Returns the number of bytes in "string\_expression".

### Syntax

```
octet_length (string_expression)
```

## Appendix A: Calculation Components

### **Example 1**

```
octet_length ('ABCDEF')
```

Result: 6

### **Example 2**

```
octet_length ('')
```

Result: 0

## **position**

Returns the integer value representing the starting position of "string\_expression1" in "string\_expression2" or 0 when the "string\_expression1" is not found.

### **Syntax**

```
position (string_expression1 , string_expression2)
```

### **Example 1**

```
position ('C' , 'ABCDEF')
```

Result: 3

### **Example 2**

```
position ('H' , 'ABCDEF')
```

Result: 0

## **power**

Returns "numeric\_expression1" raised to the power "numeric\_expression2". If "numeric\_expression1" is negative, then "numeric\_expression2" must result in an integer value.

### **Syntax**

```
power (numeric_expression1 , numeric_expression2)
```

### **Example**

```
power (3 , 2)
```

Result: 9

## **\_round**

Returns "numeric\_expression" rounded to "integer\_expression" places to the right of the decimal point. Notes: "integer\_expression" must be a non-negative integer. Rounding takes place before data formatting is applied.

### **Syntax**

```
_round (numeric_expression , integer_expression)
```

### **Example**

```
_round (1220.42369, 2)
```

Result: 1220.42

## sqrt

Returns the square root of "numeric\_expression". "Numeric\_expression" must be non-negative.

### Syntax

```
sqrt (numeric_expression)
```

### Example

```
sqrt (9)
```

Result: 3

## substring

Returns the substring of "string\_expression" that starts at position "integer\_expression1" for "integer\_expression2" characters or to the end of "string\_expression" if "integer\_expression2" is omitted. The first character in "string\_expression" is at position 1.

### Syntax

```
substring (string_expression , integer_expression1 [, integer_expression2])
```

### Example

```
substring ('abcdefg' , 3 , 2)
```

Result: cd

## trim

Returns "string\_expression" trimmed of leading and trailing blanks or trimmed of a certain character specified in "match\_character\_expression". "Both" is implicit when the first argument is not stated and blank is implicit when the second argument is not stated.

### Syntax

```
trim ([[trailing|leading|both] [match_character_expression] ,] string_expression)
```

### Example 1

```
trim (trailing 'A' , 'ABCDEFA')
```

Result: ABCDEF

### Example 2

```
trim (both , ' ABCDEF ')
```

Result: ABCDEF

## upper

Returns "string\_expression" with all lowercase characters converted to uppercase.

## Appendix A: Calculation Components

### Syntax

```
upper (string_expression)
```

### Example

```
upper ('abcdef')
```

Result: ABCDEF

## Dimensional Functions

### ancestor

Returns the ancestor of "member" at "level" or at "integer" number of levels above "member".  
Note: The result is not guaranteed to be consistent when there is more than one such ancestor.

### Syntax

```
ancestor (member, level|integer)
```

### Example 1

```
ancestor ([TrailChef Water Bag] , 1)
```

Result: Cooking Gear

### Example 2

```
ancestor ([TrailChef Water Bag] , 2)
```

Result: Camping Equipment

### Example 3

```
ancestor ([TrailChef Water Bag] , [great_outdoors_company].[Products].
[Products].[Product type])
```

Result: Cooking Gear

### ancestors

Returns all the ancestors of "member" at "level" or "index" distance above the member. (Most data sources support only one ancestor at a specified level. If the data source supports more than one ancestor, the result is a member set.)

### Syntax

```
ancestors (member , level|index)
```

### Example 1

```
ancestors ([TrailChef Water Bag] , 1)
```

Result: Cooking Gear

### Example 2

```
ancestors ([TrailChef Water Bag] , 2)
```

Result: Camping Equipment

**Example 3**

```
ancestors ([TrailChef Water Bag] , [great_outdoors_company].[Products].[Products].[Product type])
```

Result: Cooking Gear

**bottomCount**

Sorts a set according to the value of "numeric\_expression" evaluated at each of the members of "set\_expression" and returns the bottom "index\_expression" members.

**Syntax**

```
bottomCount (set_expression , index_expression , numeric_expression)
```

**Example**

```
bottomCount ([great_outdoors_company].[Products].[Products].[Product line] , 2 , [Revenue])
```

Result: Returns the bottom two members of the set sorted by revenue.

| Product line             | Revenue         |
|--------------------------|-----------------|
| -----                    | -----           |
| Outdoor Protection       | \$3,171,114.92  |
| Mountaineering Equipment | \$20,891,350.60 |

**bottomPercent**

Sorts "numeric\_expression2", evaluated at the corresponding members of "set\_expression", and picks up the bottommost elements whose cumulative total is equal to or less than "numeric\_expression1" percent of the total.

**Syntax**

```
bottomPercent (set_expression , numeric_expression1 , numeric_expression2)
```

**Example**

```
bottomPercent (set ([Camping Equipment] , [Golf Equipment] , [Mountaineering Equipment]) , 40 , [2006])
```

Result: For the set of Camping Equipment, Golf Equipment, and Mountaineering Equipment, returns the members whose percentage total are less than or equal to 40% for 2006.

**bottomSum**

Sorts "numeric\_expression2", evaluated at the corresponding member of "set\_expression", and picks up the bottommost elements whose cumulative total is equal to or less than "numeric\_expression1".

**Syntax**

```
bottomSum (set_expression , numeric_expression1 , numeric_expression2)
```

## Appendix A: Calculation Components

### Example

```
bottomSum (members ([great_outdoors_company].[Products].[Products].[Product line]) , 6000000 , tuple ([2006] , [great_outdoors_company].[Measures].[Gross profit]))
```

## caption

Returns the caption values of "level", "member", or "set\_expression". The caption is the string display name for an element and does not necessarily match the unique identifier used to generate the business key or member unique name (MUN) for the element. The caption is not necessarily unique; for example, the caption for a month may return the month name without further year details to make the value unique.

### Syntax

```
caption (level|member|set_expression)
```

### Example 1

```
caption ([TrailChef Water Bag])
```

Result: TrailChef Water Bag

### Example 2

```
caption ([great_outdoors_company].[Products].[Products].[Product line])
```

Result: Returns the caption values of the Product line set.

```
Camping Equipment
Mountaineering Equipment
Personal Accessories
Outdoor Protection
Golf Equipment
```

## children

Returns the set of children of a specified member.

### Syntax

```
children (member)
```

### Example

```
children ([Camping Equipment])
```

Result: Returns the set of children for Camping Equipment.

```
Cooking Gear
Tents
Sleeping Bags
Packs
Lanterns
```

## closingPeriod

Returns the last sibling member among the descendants of a member at "level". This function is typically used with a time dimension.

## Syntax

```
closingPeriod (level [, member])
```

### Example 1

```
closingPeriod ([great_outdoors_company].[Years].[Years].[Month])
```

Result: 2006/Dec

### Example 2

```
closingPeriod ([great_outdoors_company].[Years].[Years].[Year])
```

Result: 2006

### Example 3

```
closingPeriod ([great_outdoors_company].[Years].[Years].[Month] , [2006 Q 4])
```

Result: 2006/Dec

## cousin

Returns the child member of "member2" with the same relative position as "member1" to its parent. This function appears in the Revenue by GO Subsidiary 2005 sample report in the GO Data Warehouse (analysis) package.

## Syntax

```
cousin (member1 , member2)
```

### Example 1

```
cousin ([Irons] , [Camping Equipment])
```

Result: Cooking Gear

### Example 2

```
cousin ([Putters] , [Camping Equipment])
```

Result: Sleeping Bags

## completeTuple

Identifies a cell location (intersection) based on the specified members, each of which must be from a different dimension. However, completeTuple () implicitly includes the default member from all dimensions not otherwise specified in the arguments, rather than the current member. CompleteTuple will use the default measure rather than the currentMeasure in the query if the measure is not defined in the completetuple function. This function appears in the Planned Headcount sample report in the GO Data Warehouse (analysis) package.

## Syntax

```
completeTuple (member { , member })
```

### Example 1

```
completeTuple ([Mountaineering Equipment] , [Fax])
```

## Appendix A: Calculation Components

**Result:** The completeTuple does not pick up the currentMember by default as the tuple function does. The values in the first column are identical across each year because the default member of the Years dimension, the root member, is used rather than the current member. Likewise, the first column displays Revenue rather than Quantity Sold because the Revenue measure is the default from the Measures dimension. CompleteTuple will use the default measure rather than the currentMeasure in the query if the measure is not defined in the completetuple function.

| Quantity Sold | Mountaineering Sales by Fax |
|---------------|-----------------------------|
| -----         | -----                       |
| 2004          | \$1,220,329.38              |
| 2005          | \$1,220,329.38              |
| 2006          | \$1,220,329.38              |

### Example 2

```
completeTuple ([Mountaineering Equipment] , [Fax] , [Quantity sold] , currentMember ([great_outdoors_company].[Years].[Years]))
```

**Result:** The completeTuple function uses the currentMember of the Years dimension and the Quantity sold measure.

| Quantity Sold | Mountaineering Sales by Fax |
|---------------|-----------------------------|
| -----         | -----                       |
| 2004          | 0                           |
| 2005          | 8,746                       |
| 2006          | 7,860                       |

## currentMember

Returns the current member of the hierarchy during an iteration. If "hierarchy" is not present in the context in which the expression is being evaluated, its default member is assumed. This function appears in the Rolling and Moving Averages interactive sample report.

### Syntax

```
currentMember (hierarchy)
```

## defaultMember

Returns the default member of "hierarchy".

### Syntax

```
defaultMember (hierarchy)
```

### Example 1

```
defaultMember ([great_outdoors_company].[Products].[Products])
```

**Result:** Products

**Example 2**

```
defaultMember ([great_outdoors_company].[Years].[Years])
```

Result: Year

**Example 3**

```
defaultMember (hierarchy ([great_outdoors_company].[Measures].[Quantity sold]))
```

Result: Revenue

**descendants**

Returns the set of descendants of "member" or "set\_expression" at "level" (qualified name) or "distance" (integer 0..n) from the root. Multiple options may be specified (separated by a space) to determine which members are returned. self: Only the members at the specified level are included in the final set (this is the default behaviour in the absence of any options). before: If there are any intermediate levels between the member's level and the one specified, members from those levels are included. If the level specified is the same as the member upon which the function is applied, the member is included in the final set. beforewithmember: If there are any intermediate levels between the member's level and the one specified, members from those levels are included. The member upon which the function is applied is also included in the final set. after: If other levels exist after the specified level, members from those levels are included in the final set. This function appears in the Sales Commissions for Central Europe sample report in the GO Data Warehouse (analysis) package.

**Syntax**

```
descendants (member|set_expression , level|distance [, { self|before|beforewithmember|after }])
```

**Example 1**

```
descendants ([great_outdoors_company].[Products].[Products].[Products] , [great_outdoors_company].[Products].[Products].[Product type])
```

Result: Returns the set of descendants of the Products set at the Product type level. Note: [great\_outdoors\_company].[Products].[Products].[Products] is the root member of the Products hierarchy.

```
Cooking Gear
Sleeping Bags
Packs
Tents
...
Eyewear
Knives
Watches
```

**Example 2**

```
descendants ([great_outdoors_company].[Products].[Products].[Products] , 1)
```

Result: Returns the set of descendants of the Products set at the first level.

```
Camping Equipment
Golf Equipment
Mountaineering Equipment
Outdoor Protection
Personal Accessories
```

## Appendix A: Calculation Components

### Example 3

```
descendants ([great_outdoors_company].[Products].[Products].
[Products] , 3 , before)
```

**Result:** Returns the descendants of the Products set before the third level.

```
Camping Equipment
Cooking Gear
Sleeping Bags
Packs
Tents
...
Eyewear
Knives
Watches
```

### Example 4

```
descendants ([great_outdoors_company].[Products].[Products].
[Products] , 2 , self before)
```

**Result:** Returns the set of descendants of the Products set before and including the second level.

```
Camping Equipment
Cooking Gear
Sleeping Bags
Packs
Tents
...
Eyewear
Knives
Watches
```

## except

Returns the members of "set\_expression1" that are not also in "set\_expression2". Duplicates are retained only if the optional keyword all is supplied as the third argument.

### Syntax

```
except (set_expression1 , set_expression2 [, all])
```

### Example

```
except (set ([Camping Equipment] , [Mountaineering Equipment]) , set
([Camping Equipment] , [Golf Equipment]))
```

**Result:** Mountaineering Equipment

## filter

Returns the set resulting from filtering a specified set based on the Boolean condition. Each member is included in the result if and only if the corresponding value of "Boolean\_expression" is true.

### Syntax

```
filter (set_expression , Boolean_expression)
```

### Example

```
filter ([Product line] , [Gross margin] > .30)
```

**Result:** Mountaineering Equipment

## firstChild

Returns the first child of "member".

### Syntax

```
firstChild (member)
```

### Example 1

```
firstChild ([By Product Lines])
```

Result: Camping Equipment

### Example 2

```
firstChild ([Camping Equipment])
```

Result: Cooking Gear

## firstSibling

Returns the first child of the parent of "member".

### Syntax

```
firstSibling (member)
```

### Example 1

```
firstSibling ([Outdoor Protection])
```

Result: Camping Equipment

### Example 2

```
firstSibling ([Camping Equipment])
```

Result: Camping Equipment

## \_format

Associates a format with the expression. The format\_keyword can be PERCENTAGE\_0, PERCENTAGE\_1, or PERCENTAGE\_2. PERCENTAGE\_1 returns a percentage with one digit to the right of the decimal point, PERCENTAGE\_2 returns a percentage with two digits to the right of the decimal point, and PERCENTAGE\_3 returns a percentage value out of one with three digits to the right of the decimal point (for example, 0.965).

### Syntax

```
_format (expression , format_keyword)
```

### Example

```
_format ([Unit Sale Price] / [Unit Price] , PERCENTAGE_2)
```

Result: 75.12%

## emptySet

Returns an empty member set for "hierarchy". This is most often used as a placeholder during development or with dynamic report design (either with the IBM® Cognos® Software Development Kit or via report design). By creating a data item that contains the emptyset function, it is possible to build complex expressions that can later be revised by redefining the emptyset data item.

### Syntax

```
emptySet (hierarchy)
```

### Example

```
except ([great_outdoors_company].[Products].[Products].[Product line] , emptyset
 ([great_outdoors_company].[Products].[Products]))
```

**Result:** Returns the Product line set and an empty set for the Products set.

```
Camping Equipment
Golf Equipment
Mountaineering Equipment
Outdoor Protection
Personal Accessories
```

## generate

Evaluates "set\_expression2" for each member of "set\_expression1" and joins the resulting sets by union. The result retains duplicates only when the optional keyword "all" is supplied as the third argument.

### Syntax

```
generate (set_expression1 , set_expression2 [, all])
```

### Example

```
generate ([Product line] , topCount (descendants (currentMember ([great_
outdoors_company].[Products].[Products]) , [great_outdoors_company].[Products].
[Products].[Product name]) , 2 , [Revenue]))
```

**Result:** Returns the top two products by revenue for each product line.

## head

Returns the first "index\_expression" elements of "set\_expression". The default for "index\_expression" is 1.

### Syntax

```
head (set_expression [, index_expression])
```

### Example 1

```
head (members ([great_outdoors_company].[Products].[Products].[Product
line]))
```

**Result:** Camping Equipment

**Example 2**

```
head (members ([great_outdoors_company].[Products].[Products].[Product line]) , 2)
```

Result: Returns the top two members of the Product line set.

```
Camping Equipment
Mountaineering Equipment
```

**hierarchize**

Orders the members of "set\_expression" in a hierarchy. Members in a level are sorted in their natural order. This is the default ordering of the members along a dimension when no other sort conditions are specified.

**Syntax**

```
hierarchize (set_expression)
```

**Example**

```
hierarchize (set ([Golf Equipment] , [Mountaineering Equipment] , [Camping Equipment]))
```

Result: Returns Camping Equipment, Golf Equipment, Mountaineering Equipment.

**hierarchy**

Returns the hierarchy that contains "level", "member", or "set\_expression".

**Syntax**

```
hierarchy (level|member|set_expression)
```

**Example 1**

```
hierarchy ([Cooking Gear])
```

Result: Returns every member in the hierarchy that contains Cooking Gear.

```
Products
Camping Equipment
Cooking Gear
TrailChef Water Bag
TrailChef Canteen
...
Mountain Man Extreme
Mountain Man Deluxe
```

**Example 2**

```
hierarchy ([great_outdoors_company].[Products].[Products].[Product line])
```

Result: Returns every member in the hierarchy that contains the Product line.

```
Products
Camping Equipment
Cooking Gear
TrailChef Water Bag
TrailChef Canteen
...
Mountain Man Extreme
Mountain Man Deluxe
```

## item

Returns a member from the "index" location within "set\_expression". The index into the set is zero based.

### Syntax

```
item (set_expression , index)
```

### Example

```
item (children ([Camping Equipment]) , 2)
```

Result: Sleeping Bags

## intersect

Returns the intersection of "set\_expression1" and "set\_expression2". The result retains duplicates only when the optional keyword "all" is supplied as the third argument.

### Syntax

```
intersect (set_expression1 , set_expression2 [, all])
```

### Example

```
intersect (set ([Camping Equipment] , [Mountaineering Equipment]) , set ([Camping Equipment] , [Outdoor Protection] ,) , all)
```

Result: Camping Equipment

## lag

Returns the sibling member that is "index\_expression" number of positions prior to "member".

### Syntax

```
lag (member , index_expression)
```

### Example 1

```
lag ([Tents] , 1)
```

Result: Cooking Gear

### Example 2

```
lag ([Tents] , -2)
```

Result: Packs

## lastChild

Returns the last child of a specified member.

### Syntax

```
lastChild (member)
```

### Example 1

```
lastChild (Cooking Gear)
```

Result: TrailChef Utensils

### **Example 2**

```
lastChild ([By Product Line])
```

Result: Golf Equipment

## **lastPeriods**

Returns the set of members from the same level that ends with "member". The number of members returned is the absolute value of "integer\_expression". If "integer\_expression" is negative, members following and including the specified member are returned. Typically used with a time dimension. This function appears in the Rolling and Moving Averages interactive sample report.

### **Syntax**

```
lastPeriods (integer_expression , member)
```

### **Example 1**

```
lastPeriods (2 , [2006 Q 4])
```

Result: Returns the last two members from the level that ends with 2006 Q 4.

```
2006 Q 3
2006 Q 4
```

### **Example 2**

```
lastPeriods (-3 , [2006 Q 4])
```

Result: Returns the last three members from the level that starts with 2006 Q 4.

```
2006 Q 4
2007 Q 1
2007 Q 2
```

## **lastSibling**

Returns the last child of the parent of a specified member.

### **Syntax**

```
lastSibling (member)
```

### **Example**

```
lastSibling ([Camping Equipment])
```

Result: Golf Equipment

## **lead**

Returns the sibling member that is "index\_expression" number of positions after "member". If "index\_expression" is negative, returns the sibling member that is "index\_expression" number of positions before "member".

### **Syntax**

```
lead (member , index_expression)
```

## Appendix A: Calculation Components

### **Example 1**

```
lead ([Outdoor Protection] , 1)
```

Result: Personal Accessories

### **Example 2**

```
lead ([Outdoor Protection] , -2)
```

Result: Golf Equipment

## level

Returns the level of "member".

### **Syntax**

```
level (member)
```

### **Example**

```
level ([Golf Equipment])
```

Result: Returns the members on the Golf Equipment level.

```
Camping Equipment
Mountaineering Equipment
Personal Accessories
Outdoor Protection
Golf Equipment
```

## levels

Returns the level in "hierarchy" whose distance from the root is specified by "index".

### **Syntax**

```
levels (hierarchy , index)
```

### **Example 1**

```
levels ([great_outdoors_company].[Products].[Products] , 2)
```

Result: Returns the members two levels from the root Products hierarchy.

```
Cooking Gear
Sleeping Bags
Packs
Tents
...
Irons
Putters
Woods
Golf Accessories
```

### **Example 2**

```
levels ([great_outdoors_company].[Products].[Products] , 1)
```

Result: Returns the members one level from the root Products hierarchy.

```
Camping Equipment
Mountaineering Equipment
Personal Accessories
```

```
Outdoor Protection
Golf Equipment
```

## linkMember

Returns the corresponding member in "level" or "hierarchy" (of the same dimension). For level-based hierarchies, a level must be specified as the second argument, and for parent-child hierarchies, a hierarchy must be specified. An exception is thrown when the second parameter does not resolve to a hierarchy of the member's dimension. Note that calculated members are not supported as the first argument.

### Syntax

```
linkMember (member , level|hierarchy)
```

## members

Returns the set of members in "hierarchy" or "level". In the case of a hierarchy, the order of the members in the result is not guaranteed. If a predictable order is required, an explicit ordering function (such as hierarchize) must be used.

### Syntax

```
members (hierarchy|level)
```

### Example 1

```
members ([great_outdoors_company].[Years].[Years])
```

Result: Returns the members in Years.

### Example 2

```
members ([great_outdoors_company].[Products].[Products].[Product line])
```

Result: Returns the members in Product line.

## nextMember

Returns the next member in the "member" level.

### Syntax

```
nextMember (member)
```

### Example

```
nextMember ([Outdoor Protection])
```

Result: Golf Equipment

## openingPeriod

Returns the first sibling member among the descendants of a member at "level". This function is typically used with a time dimension.

### Syntax

```
openingPeriod (level [, member])
```

## Appendix A: Calculation Components

### Example 1

```
openingPeriod ([great_outdoors_company].[Years].[Years].[Month])
```

Result: 2004/Jan

### Example 2

```
openingPeriod ([great_outdoors_company].[Years].[Years].[Year])
```

Result: 2004

### Example 3

```
openingPeriod ([great_outdoors_company].[Years].[Years].[Month] , [2006 Q 4])
```

Result: 2006/Oct

## order

Arranges the members of "set\_expression" according to their "value\_expression" and the third parameter. ASC and DESC arrange members in ascending or descending order, respectively, according to their position in the set hierarchy. Then the children of each member are arranged according to "value\_expression". BASC and BDESC arrange members in the set without regard to the hierarchy. In the absence of an explicit specification, ASC is the default.

### Syntax

```
order (set_expression , value_expression [, ASC|DESC|BASC|BDESC])
```

### Example 1

```
order (members ([Great Outdoors Company].[Product].[Product].[Product type]) , [Quantity sold] , BASC)
```

Result: Returns the quantity sold for each product type in no particular order.

| Product Line      | Quantity |
|-------------------|----------|
| Woods             | 13,924   |
| Irons             | 14,244   |
| Safety            | 22,332   |
| ...               | ...      |
| Sunscreen         | 215,432  |
| Insect Repellents | 270,074  |
| Lanterns          | 345,096  |

### Example 2

```
order (members ([Great Outdoors Company].[Product].[Product].[Product type]) , [Quantity sold] , ASC)
```

Result: Returns the quantity sold for each product type in ascending order.

| Product Line | Quantity |
|--------------|----------|
| Woods        | 13,924   |
| Irons        | 14,244   |
| Putters      | 23,244   |
| ...          | ...      |
| Tents        | 130,664  |
| Cooking Gear | 198,676  |
| Lanterns     | 345,096  |

## ordinal

Returns the zero-based ordinal value (distance from the root level) of "level".

### Syntax

```
ordinal (level)
```

#### Example 1

```
ordinal ([great_outdoors_company].[Products].[Products].[Product line])
```

Result: 1

#### Example 2

```
ordinal ([great_outdoors_company].[Products].[Products].[Product type])
```

Result: 2

## parallelPeriod

Returns a member from a prior period in the same relative position as "member". This function is similar to the cousin function, but is more closely related to time series. It takes the ancestor of "member" at "level" (called "ancestor") and the sibling of "ancestor" that lags by "integer\_expression" positions, and returns the parallel period of "member" among the descendants of that sibling. When unspecified, "integer\_expression" defaults to 1 and "member" defaults to the current member.

### Syntax

```
parallelPeriod (level [, integer_expression [, member]])
```

#### Example 1

```
parallelPeriod ([great_outdoors_company].[Years].[Years].[Quarter] , -1 , [2006/Aug])
```

Result: 2006/Nov

## Appendix A: Calculation Components

### Example 2

```
parallelPeriod ([great_outdoors_company].[Years].[Years].[Quarter] , 1 , [2006/Aug])
```

Result: 2006/May

### Example 3

```
parallelPeriod ([great_outdoors_company].[Years].[Years].[Year] , 2 , [2006/Aug])
```

Result: 2004/Aug

## parent

Returns the member that is the parent of "member" or "measure".

### Syntax

```
parent (member|measure)
```

### Example

```
parent ([Cooking Gear])
```

Result: Camping Equipment

## periodsToDate

Returns a set of sibling members from the same level as "member", as constrained by "level". It locates the ancestor of "member" at "level" and returns that ancestor's descendants at the same level as "member" (up to and including "member"). Typically used with a time dimension. This function appears in the Rolling and Moving Averages interactive sample report.

### Syntax

```
periodsToDate (level , member)
```

### Example

```
periodsToDate ([great_outdoors_company].[Years].[Years].[Year] , [2004/Mar])
```

Result: Returns values for [2004/Jan], [2004/Feb], [2004/Mar]

## prevMember

Returns the member that immediately precedes "member" in the same level. This function appears in the Sales Growth Year Over Year sample report in the GO Data Warehouse (analysis) package.

### Syntax

```
prevMember (member)
```

### Example 1

```
prevMember ([Outdoor Protection])
```

Result: Personal Accessories

**Example 2**

```
prevMember ([2005])
```

Result: 2004

**member**

Defines a member based on "value\_expression" in "hierarchy". "String1" identifies the member created by this function. It must be unique in the query and different from any other member in the same hierarchy. "String2" is the caption of the member; if it is absent, the caption is empty. To ensure predictable results, it is recommended that you supply the "hierarchy". Note: All calculations used as grouping items whose sibling items are other calculations or member sets should be explicitly assigned to a hierarchy using this function. The results are not predictable otherwise. The only exception is where the calculation involves only members of the same hierarchy as the siblings. In this case, the calculation is assumed to belong to that hierarchy.

**Syntax**

```
member (value_expression [, string1 [, string2 [, hierarchy]]])
```

**Example**

```
member (total (currentMeasure within set filter ([great_outdoors_company].[Products].[Products].[Product name] , caption ([great_outdoors_company].[Products].[Products].[Product name]) starts with 'B')) , 'BProducts' , 'B Products' , [great_outdoors_company].[Products].[Products])
```

Result: Returns the quantity sold and revenue for all products that start with the letter B.

**nestedSet**

Returns the set of members of "set\_expression2" evaluated in the context of the current member of "set\_expression1".

**Syntax**

```
nestedSet (set_expression1 , set_expression2)
```

**Example**

```
nestedSet (members ([Product line]) , topCount (descendants (currentMember ([great_outdoors_company].[Products].[Products].[Product name]) , [great_outdoors_company].[Products].[Products].[Product name]) , 2 , [Revenue]))
```

Result: Returns the top two products by revenue for each product line.

**set**

Returns the list of members defined in the expression. The members must belong to the same hierarchy.

**Syntax**

```
set (member { , member })
```

**Example**

```
set ([Golf Equipment] , [Irons] , [TrailChef Cup])
```

## Appendix A: Calculation Components

Result: Returns Golf Equipment, Irons, and TrailChef Cup.

### siblings

Returns the children of the parent of the specified member.

#### Syntax

```
siblings (member)
```

#### Example

```
siblings ([Golf Equipment])
```

Result: Returns the siblings of Golf Equipment.

```
Camping Equipment
Golf Equipment
Mountaineering Equipment
Outdoor Protection
Personal Accessories
```

### tail

Returns the last "index\_expression" elements of "set expression". The default for "index\_expression" is 1.

#### Syntax

```
tail (set_expression [, index_expression])
```

#### Example 1

```
tail (members ([great_outdoors_company].[Products].[Products].[Product line]))
```

Result: Returns the last member of the Product line set.

```
Personal Accessories
```

#### Example 2

```
tail (members ([great_outdoors_company].[Products].[Products].[Product line]) , 2)
```

Result: Returns the last two members of the Product line set.

```
Outdoor Protection
Personal Accessories
```

### topCount

Sorts a set according to the values of "numeric\_expression" evaluated at each of the members of "set\_expression" and returns the top "index\_expression" members.

#### Syntax

```
topCount (set_expression , index_expression , numeric_expression)
```

#### Example

```
topCount ([great_outdoors_company].[Products].[Products].[Product line] , 2 , [Revenue])
```

**Result:** Returns the top two revenues for the Product line set.

| Product line         | Revenue         |
|----------------------|-----------------|
| Camping Equipment    | \$89,713,990.92 |
| Personal Accessories | \$31,894,465.86 |

## topPercent

**Sorts** "numeric\_expression2", evaluated at the corresponding members of "set\_expression", and picks up the topmost elements whose cumulative total is at least "numeric\_expression1" percent of the total.

### Syntax

```
topPercent (set_expression , numeric_expression1 , numeric_expression2)
```

### Example

```
topPercent (set ([Camping Equipment] , [Golf Equipment] , [Mountaineering Equipment]) , 40 , [2006])
```

**Result:** For the set of Camping Equipment, Golf Equipment, and Mountaineering Equipment, returns the members whose percentage totals are greater than or equal to 40% for 2006.

## topSum

**Sorts** "numeric\_expression2", evaluated at the corresponding members of "set\_expression", and picks up the topmost elements whose cumulative total is at least "numeric\_expression1".

### Syntax

```
topSum (set_expression , numeric_expression1 , numeric_expression2)
```

### Example

```
topSum (children ([Products]) , 16000000 , tuple ([2006] , [great_outdoors_company].[Measures].[Gross profit]))
```

## tuple

Identifies a cell location (intersection) based on the specified members, each of which must be from a different dimension. This function implicitly includes the current member from all dimensions that are not otherwise specified in the arguments. The current member of any dimension not specified in the evaluating context is assumed to be the default member of that dimension. The value of this cell can be obtained with the "value" function.

### Syntax

```
tuple (member { , member })
```

### Example

```
tuple ([Mountaineering Equipment] , [Fax])
```

## Appendix A: Calculation Components

Result: Returns the Mountaineering Equipment sales by fax.

### union

Returns data for "set\_expression1" and "set\_expression2". The result retains duplicates only when the optional keyword "all" is supplied as the third argument.

#### Syntax

```
union (set_expression1 , set_expression2 [, all])
```

#### Example 1

```
union (set ([Camping Equipment] , [Golf Equipment]) , set ([Golf Equipment] , [Mountaineering Equipment]))
```

Result: Returns data for both sets as one new set, showing the Golf Equipment column only once.

#### Example 2

```
union (set ([Camping Equipment] , [Golf Equipment]) , set ([Golf Equipment] , [Mountaineering Equipment]) , all)
```

Result: Returns data for both sets as one new set, showing the Golf Equipment column twice.

### roleValue

Returns the value of the attribute that is associated with the role whose name is specified by "string" within the specified context. "Member" or "set\_expression" is optional only in a number of limited circumstances, where it can be derived from another context. Applications can be made portable across different data sources and models by accessing attributes by role rather than by query item ID. For dimensionally-modeled relational (DMR) data sources, assignment of roles is the modeler's responsibility. Intrinsic roles that are defined for members of all data source types include: '\_businessKey', '\_memberCaption', '\_memberDescription', '\_memberUniqueName'. Additional roles can be defined in Framework Manager for each level in a hierarchy. For example, a Product type level may have an attribute column called "Type Shipping Container", and the Product level may have a "Product Shipping Container" attribute. Each of these could be assigned a custom role in Framework Manager called "Container". The property could then be referenced independently of the actual column name by using the roleValue function.

#### Syntax

```
roleValue (string [, member|set_expression])
```

#### Example 1

```
roleValue ('_memberCaption' , [Sales].[Product].[Product].[Product line] -> [all].[1])
```

Result: Camping Equipment

#### Example 2

```
roleValue ('_businessKey' , [great_outdoors_company].[Years].[Years].[Year])
```

Result: Returns the value of the attribute that is associated with the business key role.

```
("2004-01-01","2004-12-31")
("2005-01-01","2005-12-31")
("2006-01-01","2006-12-31")
```

### **Example 3**

```
roleValue ('_memberUniqueName' , [great_outdoors_company].[Years].[Years].[Year])
```

**Result:** Returns the value of the attribute that is associated with the MUN role.

```
[great_outdoors_company].[Years].[Years].[Year] ->:[PC].[Years (Root)].[20040101-20041231]
[great_outdoors_company].[Years].[Years].[Year] ->:[PC].[Years (Root)].[20050101-20051231]
[great_outdoors_company].[Years].[Years].[Year] ->:[PC].[Years (Root)].[20060101-20061231]
```

## **rootMember**

Returns the root member of a single-root hierarchy. This function appears in the Promotion Success sample report in the GO Data Warehouse (analysis) package.

### **Syntax**

```
rootMember (hierarchy)
```

## **rootMembers**

Returns the root members of a hierarchy.

### **Syntax**

```
rootMembers (hierarchy)
```

### **Example**

```
rootMembers ([great_outdoors_company].[Years].[Years])
```

**Result:** By Time

## **subset**

Returns a subset of members in "set\_expression" starting at "index\_expression1" from the beginning. If the count "index\_expression2" is specified, that many members are returned (if available). Otherwise, all remaining members are returned.

### **Syntax**

```
subset (set_expression, index_expression1 [, index_expression2])
```

### **Example 1**

```
subset (members ([great_outdoors_company].[Products].[Products].[Product line]) , 2)
```

**Result:** Returns the members of the Product line set starting at the second member.

```
Mountaineering Equipment
Outdoor Protection
Personal Accessories
```

## Appendix A: Calculation Components

### Example 2

```
subset (members ([great_outdoors_company].[Products].[Products].[Product
line]) , 2 , 2)
```

Result: Returns two members of the Product line set starting at the second member.

```
Mountaineering Equipment
Outdoor Protection
```

## unique

Removes all duplicates from "set\_expression". The remaining members retain their original order.

### Syntax

```
unique (set_expression)
```

## value

Returns the value of the cell identified by "tuple". Note that the default member of the Measures dimension is the Default Measure.

### Syntax

```
value (tuple)
```

### Example 1

```
value (tuple ([great_outdoors_company].[Years].[Years].[Year] ->:[PC].[Years
(Root)].[20040101-20041231] , [great_outdoors_company].[Measures].[Revenue]))
```

Result: \$34,750,563.50

### Example 2

```
value (tuple ([2004] , [Camping Equipment] , [Revenue]))
```

Result: \$20,471,328.88

## DB2

## ascii

Returns the ASCII code value of the leftmost character of the argument as an integer.

### Syntax

```
ascii (string_expression)
```

### Example

```
ascii (a)
```

Result: Returns 65, the ASCII code value of "a".

## ceiling

Returns the smallest integer greater than or equal to "numeric\_expression".

**Syntax**

```
ceiling (numeric_expression)
```

**Example**

```
ceiling (0.75)
```

Result: Returns 0.8.

**char**

Returns a string representation of a date/time value or a decimal number.

**Syntax**

```
char (expression)
```

**chr**

Returns the character that has the ASCII code value specified by "integer\_expression". "Integer\_expression" should be between 0 and 255.

**Syntax**

```
chr (integer_expression)
```

**Example**

```
chr (65)
```

Result: Returns a, the character for the ASCII code value of 65.

**concat**

Returns a string that is the result of concatenating "string\_expression1" with "string\_expression2".

**Syntax**

```
concat (string_expression1, string_expression2)
```

**Example**

```
concat ([Sales target (query)].[Sales staff].[First name], [Sales target (query)].[Sales staff].[Last name])
```

Result: Returns the first name and last name; e.g., Bob Smith.

**date**

Returns a date from a single input value. "Expression" can be a string or integer representation of a date.

**Syntax**

```
date (expression)
```

**Example**

```
date ('1998-01-08')
```

## Appendix A: Calculation Components

Result: Returns 8 January 1998.

### day

Returns the day of the month (1-31) from "date\_expression". "Date\_expression" can be a date value or a string representation of a date.

#### Syntax

```
day (date_expression)
```

#### Example

```
day ('1998-01-08')
```

Result: Returns 8.

### dayname

Returns a character string containing the data source-specific name of the day (for example, Sunday through Saturday or Sun. through Sat. for a data source that uses English, or Sonntag through Samstag for a data source that uses German) for the day portion of "date\_expression". "Date\_expression" can be a date value or a string representation of a date.

#### Syntax

```
dayname (date_expression)
```

#### Example

```
dayname ('1998-01-08')
```

Result: Returns Thursday.

### dayofweek

Returns the day of the week in "date\_expression" as an integer in the range 1 to 7, where 1 represents Sunday. "date\_expression" can be a date value or a string representation of a date.

#### Syntax

```
dayofweek (date_expression)
```

#### Example

```
dayofweek ('1998-01-08')
```

Result: Returns 5.

### dayofweek\_iso

Returns the day of the week in "date\_expression" as an integer in the range 1 to 7, where 1 represents Monday. "date\_expression" can be a date value or a string representation of a date.

#### Syntax

```
dayofweek_iso (date_expression)
```

**Example**

```
dayofweek_iso ('1998-01-08')
```

Result: Returns 4.

**dayofyear**

Returns the day of the year in "date\_expression" as an integer in the range 1 to 366.

"Date\_expression" can be a date value or a string representation of a date.

**Syntax**

```
dayofyear (date_expression)
```

**Example**

```
dayofyear (current_date)
```

Result: Returns the day of the year for the current date; e.g., if it was January 28, the expression would return 28.

**days**

Returns an integer representation of a date. "Expression" can be a date value or a string representation of a date.

**Syntax**

```
days (expression)
```

**dec**

Returns the decimal representation of "string\_expression1" with precision "numeric\_expression1", scale "numeric\_expression2", and decimal character "string\_expression2". "String\_expression1" must be formatted as an SQL Integer or Decimal constant.

**Syntax**

```
dec (string_expression1 [, numeric_expression1 [, numeric_expression2
[, string_expression2]]])
```

**decimal**

Returns the decimal representation of "string\_expression1" with precision "numeric\_expression1", scale "numeric\_expression2" and decimal character "string\_expression2". "String\_expression1" must be formatted as an SQL Integer or Decimal constant.

**Syntax**

```
decimal (string_expression1 [, numeric_expression1 [, numeric_expression2
[, string_expression2]]])
```

**difference**

Returns an integer value representing the difference between the values returned by the data source-specific soundex function for "string\_expression1" and "string\_expression2". The value returned

## Appendix A: Calculation Components

ranges from 0 to 4, with 4 indicating the best match. Note that 4 does not mean that the strings are equal.

### Syntax

```
difference (string_expression1 , string_expression2)
```

### Example 1

```
difference ([Sales target (query)].[Sales staff].[First name],[Sales (query)].[Retailers].[Contact first name])
```

Result: 0

### Example 2

```
difference ([Sales target (query)].[Sales staff].[First name],[Sales target (query)].[Sales staff].[First name])
```

Result: 4

## digits

Returns the character string representation of a non-floating point number.

### Syntax

```
digits (numeric_expression)
```

## double

Returns the floating-point representation of an expression. "Expression" can either be a numeric or string expression.

### Syntax

```
double (expression)
```

## event\_mon\_state

Returns the operational state of a particular state monitor.

### Syntax

```
event_mon_state (string_expression)
```

## float

Returns the floating-point representation of a number.

### Syntax

```
float (numeric_expression)
```

## hex

Returns the hexadecimal representation of a value.

**Syntax**

```
hex (expression)
```

**hour**

Returns the hour, an integer from 0 (midnight) to 23 (11:00 pm), from "time\_expression". "Time\_expression" can be a time value or a string representation of a time.

**Syntax**

```
hour (time_expression)
```

**Example**

```
hour (01:22:45)
```

**Result:** Returns 1.

**insert**

Returns a string where "integer\_expression2" characters have been deleted from "string\_expression1" beginning at "integer\_expression1" and where "string\_expression2" has been inserted into "string\_expression1" at its start. The first character in the string is at position 1.

**Syntax**

```
insert (string_expression1, integer_expression1, integer_expression2, string_expression2)
```

**integer**

Returns the integer representation of an expression. "Expression" can be a numeric value or a string representation of a number.

**Syntax**

```
integer (expression)
```

**Example**

```
integer (84.95)
```

**Result:** 85

**int**

Returns the integer representation of an expression. "Expression" can be a numeric value or a string representation of a number.

**Syntax**

```
int (expression)
```

**Example**

```
int (84.95)
```

**Result:** 85

## julian\_day

Returns an integer value representing the number of days from January 1, 4712 BC (the start of the Julian date calendar) to the date value specified in "expression". "Expression" can be a date value or a string representation of a date.

### Syntax

```
julian_day (expression)
```

### Example

```
julian_day ('2009-06-29')
```

Result: 2455012.22130739595741034

## lcase

Returns "string\_expression" with all uppercase characters shifted to lowercase.

### Syntax

```
lcase (string_expression)
```

### Example

```
lcase ([Sales (query)].[Sales staff].[Last name])
```

Result: Returns last names with no uppercase letters.

## left

Returns the leftmost "integer\_expression" characters of "string\_expression".

### Syntax

```
left (string_expression, integer_expression)
```

### Example

```
left ([Sales (query)].[Sales staff].[Last name] , 3)
```

Result: Returns the first three characters of each last name.

## length

Returns the length of the operand in bytes. Exception: double byte string types return the length in characters.

### Syntax

```
length (expression)
```

### Example

```
length ([Sales (query)].[Sales staff].[Record start date])
```

Result: Returns 4; dates always return a value of 4.

## locate

Returns the starting position of the first occurrence of "string\_expression1" within "string\_expression2". The search starts at position start "integer\_expression" of "string\_expression2". The first character in a string is at position 1. If "string\_expression1" is not found, zero is returned.

### Syntax

```
locate (string_expression1, string_expression2 [, integer_expression])
```

### Example

```
locate (A, [Sales (query)].[Sales staff].[Last name] , 2)
```

**Result:** Returns the position of the character A in the last names starting at the second character of the last name.

## long\_varchar

Returns a long string.

### Syntax

```
long_varchar (string_expression)
```

## ltrim

Returns "string\_expression" with leading spaces removed.

### Syntax

```
ltrim (string_expression)
```

### Example

```
ltrim ([Sales (query)].[Sales staff].[Last name])
```

**Result:** Returns last names with any leading spaces removed.

## microsecond

Returns the microsecond (time-unit) part of a value. "Expression" can be a timestamp or a string representation of a timestamp.

### Syntax

```
microsecond (expression)
```

### Example

```
microsecond (01:45:34.056)
```

**Result:** Returns 056.

## midnight\_seconds

Returns an integer value in the range 0 to 86400 representing the number of seconds between midnight and time value specified in the argument. "Expression" can be a time value, a timestamp or a string representation of a time.

### Syntax

```
midnight_seconds (expression)
```

### Example

```
midnight_seconds (01:45:34.056)
```

Result: Returns 6334.

## minute

Returns the minute (an integer from 0-59) from "time\_expression". "Time\_expression" can be a time value, a timestamp, or a string representation of a time.

### Syntax

```
minute (time_expression)
```

### Example

```
minute (01:45:34.056)
```

Result: Returns 45.

## month

Returns the month (an integer from 1-12) from "date\_expression".

### Syntax

```
month (date_expression)
```

### Example

```
month (2005-11-01)
```

Result: Returns 11.

## monthname

Returns a character string containing the data source-specific name of the month (for example, January through December or Jan. through Dec. for an English data source, or Januar through Dezember for a German data source) for the month portion of "date\_expression".

### Syntax

```
monthname (date_expression)
```

### Example

```
monthname (2005-11-01)
```

Result: November

## quarter

Returns the quarter in "date\_expression" as a number in the range 1 to 4, where 1 represents January 1 through March 31.

**Syntax**

```
quarter (date_expression)
```

**Example**

```
quarter (2005-11-01)
```

Result: Returns 4.

**radians**

Returns the number of radians converted from "numeric\_expression" degrees.

**Syntax**

```
radians (numeric_expression)
```

**repeat**

Returns a string consisting of "string\_expression" repeated "integer\_expression" times.

**Syntax**

```
repeat (string_expression, integer_expression)
```

**Example**

```
repeat (XYZ, 3)
```

Result: Returns XYZXYZXYZ.

**replace**

Replaces all occurrences of "string\_expression2" in "string\_expression1" with "string\_expression3".

**Syntax**

```
replace (string_expression1, string_expression2, string_expression3)
```

**Example**

```
replace ([Sales (query)].[Sales staff].[Position code], A, a)
```

Result: Returns position codes with all occurrences of "A" replaced by "a".

**right**

Returns the rightmost "integer\_expression" characters of "string\_expression".

**Syntax**

```
right (string_expression, integer_expression)
```

**Example**

```
right ([Sales (query)].[Sales staff].[Position code], 3)
```

Result: Returns the rightmost 3 characters of each position code.

## round

Returns "numeric\_expression" rounded to "integer\_expression" places to the right of the decimal point. If "integer\_expression" is negative, "numeric\_expression" is rounded to the nearest absolute value "integer\_expression" places to the left of the decimal point. Rounding takes place before data formatting is applied.

### Syntax

```
round (numeric_expression, integer_expression)
```

### Example

```
round (3.14159265, 3)
```

Result: Returns 3.142.

## rtrim

Returns "string\_expression" with trailing spaces removed.

### Syntax

```
rtrim (string_expression)
```

### Example

```
rtrim ([Sales (query)].[Sales staff].[Last name])
```

Result: Returns last names with any spaces at the end of the name removed.

## second

Returns the second (an integer from 0-59) from "time\_expression".

### Syntax

```
second (time_expression)
```

### Example

```
second (01:45:34.056)
```

Result: Returns 34.

## sign

Returns an indicator of the sign of "numeric\_expression": +1 if "numeric\_expression" is positive, 0 if zero, or -1 if negative.

### Syntax

```
sign (numeric_expression)
```

### Example

```
sign ([Revenue])
```

Result: Returns + for positive values and - for negative values.

## smallint

Returns the small integer representation of a number.

### Syntax

```
smallint (expression)
```

## soundex

Returns a 4 character string code obtained by systematically abbreviating words and names in "string\_expression" according to phonetics. Can be used to determine if two strings sound the same. For example, does sound-of ('SMITH') = sound-of ('SMYTH').

### Syntax

```
soundex (string_expression)
```

## space

Returns a string consisting of "integer\_expression" spaces.

### Syntax

```
space (integer_expression)
```

### Example

```
space (5)
```

Result: Returns 5 spaces.

## substr

Returns the substring of "string\_expression" that starts at position "integer\_expression1" for "integer\_expression2" characters. The first character in "string\_expression" is at position 1.

### Syntax

```
substr (string_expression , integer_expression1 [, integer_expression2])
```

### Example

```
substr ([Sales (query)].[Sales staff].[Position code] , 3 , 5)
```

Result: Returns characters 3 to 7 of the position codes.

## table\_name

Returns an unqualified name of a table or view based on the object name in "string\_expression1" and the schema name given in "string\_expression2". It is used to resolve aliases.

### Syntax

```
table_name (string_expression1 [, string_expression2])
```

## table\_schema

Returns the schema name portion of the two-part table or view name based on the object name in "string\_expression1" and the schema name in "string\_expression2". It is used to resolve aliases.

### Syntax

```
table_schema (string_expression1 [, string_expression2])
```

## time

Returns a time from a value.

### Syntax

```
time (expression)
```

## timestamp

Returns a timestamp from a value or a pair of values. "Expression1" must represent a date value, and "expression2" must represent a time value.

### Syntax

```
timestamp (expression1 [, expression2])
```

### Example

```
timestamp (11 November 2005 , 12:00:00.000000)
```

**Result:** Returns 2005-11-11 12:00:00.000000.

## timestamp\_iso

Returns a datetime in the ISO format (yyyy-mm-dd hh:mm:ss.nnnnnn) converted from the IBM format (yyyy-mm-dd-hh.mm.ss.nnnnnn). If "expression" is a time, it inserts the value of the CURRENT DATE for the date elements and zero for the fractional time element.

### Syntax

```
timestamp_iso (expression)
```

### Example

```
timestamp_iso (11 November 2005 , 12:00:00.000000)
```

**Result:** Returns 2005-11-11 12:00:00.000000.

## timestampdiff

Returns an estimated number of intervals of type "expression1" based on the difference between two timestamps. "Expression2" is the result of subtracting two timestamp types and converting the result to CHAR. Valid values of "expression1" are: 1 Fractions of a second; 2 Seconds; 4 Minutes; 8 Hours; 16 Days; 32 Weeks; 64 Months; 128 Quarters; 256 Years.

### Syntax

```
timestampdiff (expression1, expression2)
```

## to\_char

Returns the string representation of a timestamp with the format of "string\_expression".

### Syntax

```
to_char (timestamp_expression , string_expression)
```

## translate

Returns "string\_expression1" in which characters from "string\_expression3" are translated to the equivalent characters in "string\_expression2". "String\_expression4" is a single character that is used to pad "string\_expression2" if it is shorter than "string\_expression3". If only "string\_expression1" is present, then this function translates it to uppercase characters.

### Syntax

```
translate (string_expression1 [, string_expression2, string_expression3
[, string_expression4]])
```

### Example 1

```
translate ('abcdefg')
```

Result: Returns ABCDEFG.

### Example 2

```
translate ('mnlop' , n, m , -)
```

Result: Returns n-nlop.

## trunc

Returns "numeric\_expression1" truncated to "numeric\_expression2" places to the right of the decimal point. If "numeric\_expression2" is negative, "numeric\_expression1" is truncated to the absolute value of "numeric\_expression2" places to the left of the decimal point.

### Syntax

```
trunc (numeric_expression1, numeric_expression2)
```

### Example

```
trunc (3.14159265, 3)
```

Result: Returns 3.141.

## truncate

Returns "numeric\_expression1" truncated to "numeric\_expression2" places to the right of the decimal point. If "numeric\_expression2" is negative, "numeric\_expression1" is truncated to the absolute value of "numeric\_expression2" places to the left of the decimal point.

### Syntax

```
truncate (numeric_expression1, numeric_expression2)
```

## Appendix A: Calculation Components

### **Example**

```
truncate (3141.59265, -3)
```

Result: Returns 3.

## ucase

Returns "string\_expression" with all lowercase characters shifted to uppercase.

### **Syntax**

```
ucase (string_expression)
```

### **Example**

```
ucase (XY896Zbcd789)
```

Result: Returns XY896ZBCED789.

## value

Returns the first non-null argument (or null if all arguments are null). The Value function takes two or more arguments.

### **Syntax**

```
value (expression_list)
```

### **Example**

```
value ([Unit cost], [Unit price], [Unit sale price])
```

Result: Returns the first non-null value.

## varchar

Returns a VARCHAR representation of expression, with length numeric\_expression.

### **Syntax**

```
varchar (expression [, numeric_expression])
```

## week

Returns the week of the year in "date\_expression" as an integer value in the range 1 to 53.

### **Syntax**

```
week (date_expression)
```

### **Example**

```
week (11 November 2005)
```

Result: Returns 45.

## year

Returns the year from "date\_expression".

**Syntax**

```
year (date_expression)
```

**Example**

```
year (11 November 2005)
```

Result: Returns 2005.

**DB2 Math****log**

Returns the natural logarithm of "numeric\_expression".

**Syntax**

```
log (numeric_expression)
```

**log10**

Returns the base ten logarithm of "numeric\_expression".

**Syntax**

```
log10 (numeric_expression)
```

**rand**

Generates a random number using "integer\_expression" as a seed value.

**Syntax**

```
rand (integer_expression)
```

**DB2 Trigonometry****acos**

Returns the arccosine of "numeric\_expression" in radians. The arccosine is the angle whose cosine is "numeric\_expression".

**Syntax**

```
acos (numeric_expression)
```

**asin**

Returns the arcsine of "numeric\_expression" in radians. The arcsine is the angle whose sine is "numeric\_expression".

**Syntax**

```
asin (numeric_expression)
```

**atan**

Returns the arctangent of "numeric\_expression" in radians. The arctangent is the angle whose tangent is "numeric\_expression".

## Appendix A: Calculation Components

### Syntax

```
atan (numeric_expression)
```

## atanh

Returns the hyperbolic arctangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

### Syntax

```
atanh (numeric_expression)
```

## atan2

Returns the arctangent of the x and y coordinates specified by "numeric\_expression1" and "numeric\_expression2", respectively, in radians. The arctangent is the angle whose tangent is "numeric\_expression2" / "numeric\_expression1".

### Syntax

```
atan2 (numeric_expression1 , numeric_expression2)
```

## cos

Returns the cosine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

### Syntax

```
cos (numeric_expression)
```

## cosh

Returns the hyperbolic cosine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

### Syntax

```
cosh (numeric_expression)
```

## cot

Returns the cotangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

### Syntax

```
cot (numeric_expression)
```

## degrees

Returns "numeric\_expression" radians converted to degrees.

### Syntax

```
degrees (numeric_expression)
```

**sin**

Returns the sine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
sin (numeric_expression)
```

**sinh**

Returns the hyperbolic sine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
sinh (numeric_expression)
```

**tan**

Returns the tangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
tan (numeric_expression)
```

**tanh**

Returns the hyperbolic tangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
tanh (numeric_expression)
```

# Informix

**cardinality**

Returns the number of elements in a collection column (SET, MULTISET, LIST).

**Syntax**

```
cardinality (string_expression)
```

**char\_length**

Returns the number of logical characters in "string\_expression". The number of logical characters can be distinct from the number of bytes in some East Asian locales.

**Syntax**

```
char_length (string_expression)
```

## concat

Returns a string that is the result of concatenating, or joining, "string\_expression1" to "string\_expression2".

### Syntax

```
concat (string_expression1 , string_expression2)
```

### Example

```
concat ([Sales (query)].[Sales staff].[First name], [Sales (query)].[Sales staff].[Last name])
```

Result: Returns the first name and last name; e.g., Bob Smith.

## date

Returns the date value of "string\_expression", "date\_expression", or "integer\_expression".

### Syntax

```
date (string_expression|date_expression|integer_expression)
```

## day

Returns an integer that represents the day of the month (1-31).

### Syntax

```
day (date_expression)
```

## extend

Adjusts the precision of a datetime or date expression. The expression cannot be a quoted string representation of a date value. If you do not specify first and last qualifiers, the default qualifiers are year to fraction (3). If the expression contains fields that are not specified by the qualifiers, the unwanted fields are discarded. If the first qualifier specifies a larger (more significant) field than what exists in the expression, the new fields are filled in with values returned by the current function. If the last qualifier specifies a smaller (less significant) field than what exists in the expression, the new fields are filled in with constant values. A missing month or day field is filled in with 1, and missing hour to fraction fields are filled in with 0.

### Syntax

```
extend (date_expression , ' { ' year to second ' } ')
```

### Example

```
extend (some_date_column , { year to second })
```

## hex

Returns the hexadecimal encoding of "integer\_expression".

**Syntax**

```
hex (integer_expression)
```

**initcap**

Returns "string\_expression" with the first letter of each word in uppercase and all other letters in lowercase. A word begins after any character other than a letter. Thus, in addition to a blank space, symbols such as commas, periods, and colons can introduce a new word.

**Syntax**

```
initcap (string_expression)
```

**length**

Returns the number of bytes in "string\_expression", not including any trailing blank spaces. For byte or text "string\_expression", length returns the full number of bytes, including any trailing blank spaces.

**Syntax**

```
length (string_expression)
```

**lpad**

Returns "string\_expression1" left-padded by "string\_expression2" to the total number of characters specified by "integer\_expression". The sequence of "string\_expression2" occurs as many times as necessary to make the return string the length specified by "integer\_expression".

**Syntax**

```
lpad (string_expression1 , integer_expression , string_expression2)
```

**mdy**

Returns a type date value with three expressions that evaluate to integers that represent the month (integer\_expression1), day (integer\_expression2), and year (integer\_expression3).

**Syntax**

```
mdy (integer_expression1 , integer_expression2 , integer_expression3)
```

**month**

Returns an integer corresponding to the month portion of "date\_expression".

**Syntax**

```
month (date_expression)
```

**nvl**

Returns the value of "expression1" if "expression1" is not NULL. If "expression1" is NULL, then returns the value of "expression2".

## Appendix A: Calculation Components

### Syntax

```
nvl (expression1 , expression2)
```

### Example

```
nvl ([Unit sale price] , [Unit price])
```

Result: Returns the unit sale price, or returns the unit price if the unit sale price is NULL.

## octet\_length

Returns the number of bytes in "string\_expression", including any trailing spaces.

### Syntax

```
octet_length (string_expression)
```

## replace

Returns "string\_expression1" in which every occurrence of "string\_expression2" is replaced by "string\_expression3". If you omit the "string\_expression3" option, every occurrence of "string\_expression2" is omitted from the return string.

### Syntax

```
replace (string_expression1 , string_expression2 [, string_expression3])
```

### Example

```
replace ([Sales (query)].[Products].[Product line code] , -)
```

Result: Returns all product line codes without the character "-"

## round

Returns the rounded value of "numeric\_expression". If you omit "integer\_expression", the value is rounded to zero digits or to the units place. The digit range of 32 (+ and -) refers to the entire decimal value. Rounding takes place before data formatting is applied.

### Syntax

```
round (numeric_expression [, integer_expression])
```

### Example

```
round (125, -1)
```

Result: 130

## rpad

Returns "string\_expression1" right-padded by "string\_expression2" to the total number of characters specified by "integer\_expression". The sequence of "string\_expression2" occurs as many times as necessary to make the return string the length specified by "integer\_expression".

### Syntax

```
rpad (string_expression1 , integer_expression , string_expression2)
```

## substr

Returns the substring of "string\_expression" that starts at position "integer\_expression1" for "integer\_expression2" characters. The first character in "string\_expression" is at position 1. If you omit "integer\_expression2", returns the substring of "string\_expression" that starts at position "integer\_expression1" and ends at the end of "string\_expression".

### Syntax

```
substr (string_expression , integer_expression1 [, integer_expression2])
```

### Example

```
substr ([Sales (query)].[Sales staff].[Position code], 3 , 5)
```

Result: Returns characters 3 to 7 of the position codes.

## to\_char

Returns the character string "date\_expression" with the specified "string\_expression" formatting. You can use this function only with built-in data types.

### Syntax

```
to_char (date_expression , string_expression)
```

## to\_date

Returns "string\_expression1" as a date according to the date format you specify in "string\_expression2". If "string\_expression1" is NULL, then a NULL value is returned.

### Syntax

```
to_date (string_expression1 , string_expression2)
```

## trunc

Returns the truncated value of "numeric\_expression". If you omit "integer\_expression", then "numeric\_expression" is truncated to zero digits or to the unit's place. The digit limitation of 32 (+ and -) refers to the entire decimal value.

### Syntax

```
trunc (numeric_expression [, integer_expression])
```

## weekday

Returns an integer that represents the day of the week of "date\_expression". Zero (0) represents Sunday, one (1) represents Monday, and so on.

### Syntax

```
weekday (date_expression)
```

## Appendix A: Calculation Components

### year

Returns a four-digit integer that represents the year of "date\_expression".

#### Syntax

```
year (date_expression)
```

## Informix Math

### log10

Returns the logarithm of "numeric\_expression" to base 10.

#### Syntax

```
log10 (numeric_expression)
```

### logn

Returns the natural logarithm of "numeric\_expression".

#### Syntax

```
logn (numeric_expression)
```

### root

Returns the root value of "numeric\_expression1". Requires at least one numeric argument (the radians argument). If only "numeric\_expression1" is supplied, 2 is used as a default value for "numeric\_expression2". Zero cannot be used as the value of "numeric\_expression2".

#### Syntax

```
root (numeric_expression1 [, numeric_expression2])
```

## Informix Trigonometry

### acos

Returns the arccosine of "numeric\_expression" in radians. The arccosine is the angle whose cosine is "numeric\_expression".

#### Syntax

```
acos (numeric_expression)
```

### asin

Returns the arcsine of "numeric\_expression" in radians. The arcsine is the angle whose sine is "numeric\_expression".

#### Syntax

```
asin (numeric_expression)
```

**atan**

Returns the arctangent of "numeric\_expression" in radians. The arctangent is the angle whose tangent is "numeric\_expression".

**Syntax**

```
atan (numeric_expression)
```

**atan2**

Returns the arctangent of the x and y coordinates specified by "numeric\_expression1" and "numeric\_expression2", respectively, in radians. The arctangent is the angle whose tangent is "numeric\_expression1".

**Syntax**

```
atan2 (numeric_expression1 , numeric_expression2)
```

**cos**

Returns the cosine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
cos (numeric_expression)
```

**sin**

Returns the sine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
sin (numeric_expression)
```

**tan**

Returns the tangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
tan (numeric_expression)
```

**MS Access****ascii**

Returns the ascii code value of the leftmost character of "string\_expression".

**Syntax**

```
ascii (string_expression)
```

## ceiling

Returns the smallest integer greater than or equal to "numeric\_expression".

### Syntax

```
ceiling (numeric_expression)
```

## chr

Returns the character that has the ASCII code value specified by "integer\_expression". "Integer\_expression" should be between 0 and 255.

### Syntax

```
chr (integer_expression)
```

## concat

Returns a string that is the result of concatenating, or joining, "string\_expression1" to "string\_expression2".

### Syntax

```
concat (string_expression1 , string_expression2)
```

### Example

```
concat ([Sales (query)].[Sales staff].[First name], [Sales (query)].[Sales staff].[Last name])
```

Result: Returns the first name and last name; e.g., Bob Smith.

## curdate

Returns a date value representing the current date of the computer that the database software runs on.

### Syntax

```
curdate ()
```

## curtime

Returns a time value representing the current time of the computer that the database software runs on.

### Syntax

```
curtime ()
```

## dayname

Returns a character string containing the data source-specific name of the day (for example, Sunday through Saturday or Sun. through Sat. for an English data source, or Sonntag through Samstag for a German data source) for the day portion of "date\_expression".

**Syntax**

```
dayname (date_expression)
```

**dayofmonth**

Returns the day of the month (1-31) from "date\_expression". Returns the days field (a signed integer) from "interval\_expression".

**Syntax**

```
dayofmonth (date_expression|interval_expression)
```

**dayofweek**

Returns the day of the week in "date\_expression" as an integer (1-7), where 1 represents Monday.

**Syntax**

```
dayofweek (date_expression)
```

**dayofyear**

Returns the day of the year in "date\_expression" as an integer (1-366).

**Syntax**

```
dayofyear (date_expression)
```

**hour**

Returns the hour from "time\_expression" as an integer from 0 (midnight) to 23 (11:00 pm).

**Syntax**

```
hour (time_expression)
```

**instr**

Searches "string\_expression1" for the first occurrence of "string\_expression2" and returns an integer specifying the position of "string\_expression2". "Integer\_expression1" sets the starting position for the search. If "integer\_expression1" is omitted, the search begins at the first character position of "string\_expression1". "Integer\_expression2" specifies the type of string comparison. "Integer\_expression1" is required if "integer\_expression2" is specified.

**Syntax**

```
instr ([integer_expression1 ,] string_expression1 , string_expression2
[, integer_expression2])
```

**lcase**

Returns "string\_expression" with all uppercase characters converted to lowercase.

**Syntax**

```
lcase (string_expression)
```

## Appendix A: Calculation Components

### left

Returns the leftmost "integer\_expression" characters of "string\_expression".

#### Syntax

```
left (string_expression , integer_expression)
```

#### Example

```
left ([Sales (query)].[Sales staff].[Last name] , 3)
```

**Result:** Returns the first three characters of each last name.

### length

Returns the number of characters in "string\_expression", excluding trailing blanks and the string termination character.

#### Syntax

```
length (string_expression)
```

### locate

Returns the starting position of the first occurrence of "string\_expression1" within "string\_expression2". The search starts at position "integer\_expression" of "string\_expression2". The first character in a string is at position 1. If "string\_expression1" is not found, then zero is returned.

#### Syntax

```
locate (string_expression1 , string_expression2 [, integer_expression])
```

### ltrim

Returns "string\_expression" with leading spaces removed.

#### Syntax

```
ltrim (string_expression)
```

### minute

Returns the minute (an integer from 0-59) from "time\_expression".

#### Syntax

```
minute (time_expression)
```

### month

Returns the month (an integer from 1-12) from "date\_expression".

#### Syntax

```
month (date_expression)
```

## monthname

Returns a character string containing the data source-specific name of the month (for example, January through December or Jan. through Dec. for an English data source, or Januar through Dezember for a German data source) for the month portion of "date\_expression".

### Syntax

```
monthname (date_expression)
```

### Example

```
monthname (2005-11-01)
```

Result: November

## now

Returns a datetime value representing the current date and time of the computer that the database software runs on.

### Syntax

```
now ()
```

## position

Returns the starting position of "string\_expression1" in "string\_expression2". The first character in a string is at position 1.

### Syntax

```
position (string_expression1 , string_expression2)
```

## quarter

Returns the quarter in "date\_expression" as a number (1-4), where 1 represents January 1 through March 31.

### Syntax

```
quarter (date_expression)
```

## right

Returns the rightmost "integer\_expression" characters of "string\_expression".

### Syntax

```
right (string_expression , integer_expression)
```

## round

Returns "numeric\_expression" rounded to the nearest value "integer\_expression" places right of the decimal point. If "integer\_expression" is negative, "numeric\_expression" is rounded to the

## Appendix A: Calculation Components

nearest absolute value "integer\_expression" places to the left of the decimal point. Rounding takes place before data formatting is applied.

### Syntax

```
round (numeric_expression , integer_expression)
```

### Example

```
round (125, -1)
```

Result: 130

## rtrim

Returns "string\_expression" with trailing spaces removed.

### Syntax

```
rtrim (string_expression)
```

### Example

```
rtrim ([Sales (query)].[Sales staff].[Last name])
```

Result: Returns last names with any spaces at the end of the name removed.

## sign

Returns an indicator of the sign of "numeric\_expression", +1 if positive, 0 if zero, or -1 if negative.

### Syntax

```
sign (numeric_expression)
```

## space

Returns a string consisting of "integer\_expression" spaces.

### Syntax

```
space (integer_expression)
```

## substr

Returns the substring of "string\_expression" that starts at position "integer\_expression1" for "integer\_expression2" characters. The first character in "string\_expression" is at position 1.

### Syntax

```
substr (string_expression , integer_expression1 , integer_expression2)
```

### Example

```
substr ([Sales (query)].[Sales staff].[Position code], 3 , 5)
```

Result: Returns characters 3 to 7 of the position codes.

## substring

Returns the substring of "string\_expression" that starts at position "integer\_expression1" for "integer\_expression2" characters. The first character in "string\_expression" is at position 1.

### Syntax

```
substring (string_expression , integer_expression1 , integer_expression2)
```

### Example

```
substring ([Sales (query)].[Sales staff].[Position code], 3 , 5)
```

Result: Returns characters 3 to 7 of the position codes.

## truncate

Returns "string\_expression" with trailing spaces removed.

### Syntax

```
truncate (string_expression)
```

## ucase

Returns "string\_expression" with all lowercase characters converted to uppercase.

### Syntax

```
ucase (string_expression)
```

## week

Returns the week of the year in "date\_expression" as an integer value (1-53), where 1 represents the first week of the year.

### Syntax

```
week (date_expression)
```

## year

Returns the year from "date\_expression".

### Syntax

```
year (date_expression)
```

## MS Access Cast

### cast\_decimal

Returns the value of "expression" cast as a decimal.

### Syntax

```
cast_decimal (expression)
```

## Appendix A: Calculation Components

### cast\_float

Returns the value of "expression" cast as a float.

#### Syntax

```
cast_float (expression)
```

### cast\_integer

Returns the value of "expression" cast as an integer.

#### Syntax

```
cast_integer (expression)
```

#### Example

```
cast_integer (84.95)
```

Result: 84

### cast\_numeric

Returns "string\_expression" cast as a numeric value.

#### Syntax

```
cast_numeric (string_expression)
```

### cast\_real

Returns the value of "expression" cast as a real value.

#### Syntax

```
cast_real (expression)
```

### cast\_smallint

Returns "expression" cast as a small integer.

#### Syntax

```
cast_smallint (expression)
```

### cast\_varchar

Returns the value of "expression" cast as a variable character field.

#### Syntax

```
cast_varchar (expression)
```

## MS Access Math

### log

Returns the natural logarithm of "numeric\_expression".

**Syntax**

```
log (numeric_expression)
```

**rand**

Generates a random number using "integer\_expression" as a seed value.

**Syntax**

```
rand (integer_expression)
```

**MS Access Trigonometry****atan**

Returns the arctangent of "numeric\_expression" in radians. The arctangent is the angle whose tangent is "numeric\_expression".

**Syntax**

```
atan (numeric_expression)
```

**cos**

Returns the cosine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
cos (numeric_expression)
```

**sin**

Returns the sine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
sin (numeric_expression)
```

**tan**

Returns the tangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
tan (numeric_expression)
```

**Netezza****ascii**

Returns a number representing the ASCII code value of the leftmost character of "string\_expression"; for example, ascii ( 'A' ) is 65.

## Appendix A: Calculation Components

### Syntax

```
ascii (string_expression)
```

## chr

Returns the character that has the ASCII code value specified by "integer\_expression". "Integer\_expression" should be between 0 and 255.

### Syntax

```
chr (integer_expression)
```

## decode

Compares "expr" to each search value one by one. If "expr" is equal to a search, then it returns the corresponding result. If no match is found, it returns "default". If "default" is omitted, it returns null.

### Syntax

```
decode (expr , search , result [, search , result]... [, default])
```

## initcap

Returns "string\_expression", with the first letter of each word in uppercase, all other letters in lowercase. Words are delimited by white space or characters that are not alphanumeric.

### Syntax

```
initcap (string_expression)
```

## instr

Searches "string\_expression1" starting at position "integer\_expression1" for the "integer\_expression2" occurrence of "string\_expression2". If "integer\_expression1" is negative then the search is backwards from the end of "string\_expression1". Returns an integer indicating the position of "string\_expression2".

### Syntax

```
instr (string_expression1 , string_expression2 [, integer_expression1 [, integer_expression2]])
```

## lpad

Returns "string\_expression1" padded to length "integer\_expression" with occurrences of "string\_expression2". If "string\_expression1" is longer than "integer\_expression", the appropriate portion of "string\_expression1" is returned.

### Syntax

```
lpad (string_expression1 , integer_expression [, string_expression2])
```

**ltrim**

Returns "string\_expression1", with leading characters removed up to the first character not in "string\_expression2"; for example, ltrim ( 'xyxXxyAB' , 'xy' ) returns XxyAB.

**Syntax**

```
ltrim (string_expression1 [, string_expression2])
```

**months\_between**

Returns the number of months from "date\_expression1" to "date\_expression2". If "date\_expression1" is later than "date\_expression2" then the result will be a positive number. The days and time portions of the difference are ignored, i.e., the months are not rounded, except if "date\_expression1" and "date\_expression2" are the last days of a month.

**Syntax**

```
months_between (date_expression1 , date_expression2)
```

**next\_day**

Returns the datetime of the first weekday named by "string\_expression" that is later than "datetime\_expression". The return value has the same hours, minutes, and seconds as "datetime\_expression".

**Syntax**

```
next_day (datetime_expression , string_expression)
```

**nvl**

Returns "expression" if not null, otherwise returns "constant". Valid for "numeric\_expression", "string\_expression", "date\_expression", and "time\_expression".

**Syntax**

```
nvl (expression , constant)
```

**round**

Returns "numeric\_expression" rounded to the nearest value "integer\_expression" places right of the decimal point. If "integer\_expression" is negative, "numeric\_expression" is rounded to the nearest absolute value "integer\_expression" places to the left of the decimal point; for example, round (125, -1) rounds to 130.

**Syntax**

```
round (numeric_expression [, integer_expression])
```

**rpad**

Returns "string\_expression1" right-padded to length "integer\_expression" with occurrences of "string\_expression2". If "string\_expression1" is longer than "integer\_expression", the appropriate portion of "string\_expression1" is returned. If "string\_expression2" is not specified, then spaces are used.

## Appendix A: Calculation Components

### Syntax

```
rpad (string_expression1 , integer_expression [, string_expression2])
```

## rtrim

Returns "string\_expression1", with final characters removed after the last character not in "string\_expression2"; for example, rtrim ( 'ABxXxyx' , 'xy' ) returns ABxX. If "string\_expression2" is not specified, the final space characters are removed.

### Syntax

```
rtrim (string_expression1 [, string_expression2])
```

## substr

Returns the substring of "string\_expression" that starts at position "integer\_expression1". The first character in "string\_expression" is at position 1. "Integer\_expression2" can be used to select fewer characters; by default it selects characters to the end of the string.

### Syntax

```
substr (string_expression , integer_expression1 [, integer_expression2])
```

## {current\_db}

### Syntax

```
{current_db}
```

## {current\_user}

### Syntax

```
{current_user}
```

## {session\_user}

### Syntax

```
{session_user}
```

## to\_char

Returns the string representation of "expression" with the format of "string\_expression". "Expression" can be either a date value or a numeric value.

### Syntax

```
to_char (expression [, string_expression])
```

## to\_date

Converts "string\_expression1" to a datetime value as specified by the format "string\_expression2".

**Syntax**

```
to_date (string_expression1 , string_expression2)
```

**to\_number**

Converts "string\_expression1" to a numeric value as specified by the format "string\_expression2".

**Syntax**

```
to_number (string_expression1 , string_expression2)
```

**translate**

Returns "string\_expression1", with all occurrences of each character in "string\_expression2" replaced by its corresponding character in "string\_expression3".

**Syntax**

```
translate (string_expression1 , string_expression2 , string_expression3)
```

**date\_trunc**

Truncates "date\_expression1" to a value as specified by the format "string\_expression1".

**Syntax**

```
date_trunc (string_expression1 , date_expression1)
```

**trunc**

Truncates digits from "numeric\_expression1" using "numeric\_expression2" as the precision.

**Syntax**

```
trunc (numeric_expression1 [, numeric_expression2])
```

**version**

Returns the "string\_expression1" value of the database version.

**Syntax**

```
version ()
```

**Netezza Math****log**

Returns the logarithm of "numeric\_expression2" to the base "numeric\_expression1".

**Syntax**

```
log (numeric_expression1 , numeric_expression2)
```

## Netezza Trigonometry

### acos

Returns the arccosine of "numeric\_expression" in radians. The arccosine is the angle whose cosine is "numeric\_expression".

#### Syntax

```
acos (numeric_expression)
```

### asin

Returns the arcsine of "numeric\_expression" in radians. The arcsine is the angle whose sine is "numeric\_expression".

#### Syntax

```
asin (numeric_expression)
```

### atan

Returns the arctangent of "numeric\_expression" in radians. The arctangent is the angle whose tangent is "numeric\_expression".

#### Syntax

```
atan (numeric_expression)
```

### atan2

Returns the arctangent of the x and y coordinates specified by "numeric\_expression1" and "numeric\_expression2", respectively, in radians. The arctangent is the angle whose tangent is "numeric\_expression2" / "numeric\_expression1".

#### Syntax

```
atan2 (numeric_expression1 , numeric_expression2)
```

### cos

Returns the cosine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

#### Syntax

```
cos (numeric_expression)
```

### degrees

Returns the degrees where "numeric\_expression" is an angle expressed in radians.

#### Syntax

```
degrees (numeric_expression)
```

### radians

Returns the radians where "numeric\_expression" is an angle expressed in degrees.

**Syntax**

```
radians (numeric_expression)
```

**sin**

Returns the sine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
sin (numeric_expression)
```

**tan**

Returns the tangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
tan (numeric_expression)
```

**Netezza Fuzzy****le\_dst**

Returns a value indicating how different the two input strings are, calculated according to the Levenshtein edit distance algorithm.

**Syntax**

```
le_dst (string_expression1 , string_expression2)
```

**dle\_dst**

Returns a value indicating how different the two input strings are, calculated according to the Damerau-Levenshtein distance algorithm

**Syntax**

```
dle_dst (string_expression1 , string_expression2)
```

**Netezza Phonetic****nysiis**

Returns a Soundex representation of "string\_expression" using the New York State Identification and Intelligence System (NYSIIS) variation of Soundex.

**Syntax**

```
nysiis (string_expression)
```

**dbl\_mp**

Returns a composite 32-bit value of "string\_expression".

## Appendix A: Calculation Components

### Syntax

```
dbl_mp (string_expression)
```

## pri\_mp

Returns the 4 character primary metaphone string from "numeric\_expression" returned by dbl\_mp.

### Syntax

```
pri_mp (numeric_expression)
```

## sec\_mp

Returns the 4 character secondary metaphone string from "numeric\_expression" returned by dbl\_mp.

### Syntax

```
sec_mp (numeric_expression)
```

## score\_mp

Returns a score for how closely "numeric\_expression" and "numeric\_expression2" match.

### Syntax

```
score_mp (numeric_expression , numeric_expression2 , numeric_expression3 ,
numeric_expression4 , numeric_expression5 , numeric_expression6)
```

## Oracle

## add\_months

Returns the datetime resulting from adding "integer\_expression" months to "date\_expression".

### Syntax

```
add_months (date_expression , integer_expression)
```

## ascii

Returns a number representing the ASCII code value of the leftmost character of "string\_expression".

### Syntax

```
ascii (string_expression)
```

### Example

```
ascii ('A')
```

Result: Returns '65'

## ceil

Returns the smallest integer greater than or equal to "numeric\_expression".

**Syntax**

```
ceil (numeric_expression)
```

**char\_length**

Returns the number of logical characters in "string\_expression". The number of logical characters can be distinct from the number of bytes in some East Asian locales.

**Syntax**

```
char_length (string_expression)
```

**chr**

Returns the character that has the ASCII code value specified by "integer\_expression". "Integer\_expression" should be between 0 and 255.

**Syntax**

```
chr (integer_expression)
```

**concat**

Returns a string that is the result of concatenating, or joining, "string\_expression1" to "string\_expression2".

**Syntax**

```
concat (string_expression1 , string_expression2)
```

**Example**

```
concat ([Sales (query)].[Sales staff].[First name], [Sales (query)].[Sales staff].[Last name])
```

**Result:** Returns the first name and last name; e.g., Bob Smith.

**decode**

Compares "expression" to each search value one by one. If "expression" is equal to a search, then it returns the corresponding result. If no match is found, it returns "default", or if "default" is omitted, it returns null.

**Syntax**

```
decode (expression , search , result [, search , result]... [, default])
```

**dump**

Returns internal representation of "expression" with the format of "numeric\_expression1" starting from position "numeric\_expression2" for "numeric\_expression3" characters.

**Syntax**

```
dump (expression [, numeric_expression1 [, numeric_expression2 [, numeric_expression3]]])
```

## greatest

Returns the greatest value in "expression\_list".

### Syntax

```
greatest (expression_list)
```

## initcap

Returns "string\_expression" with the first letter of each word in uppercase and all other letters in lowercase. Words are delimited by white space or characters that are not alphanumeric.

### Syntax

```
initcap (string_expression)
```

## instr

Searches "string\_expression1" starting at position "integer\_expression1" for the "integer\_expression2" occurrence of "string\_expression2". If "integer\_expression1" is negative, then the search occurs backwards from the end of "string\_expression1". Returns an integer indicating the position of "string\_expression2".

### Syntax

```
instr (string_expression1, string_expression2 [, integer_expression1
[, integer_expression2]])
```

## instrb

Searches "string\_expression1" starting at position "integer\_expression1" for the "integer\_expression2" occurrence of "string\_expression2". If "integer\_expression1" is negative, then the search occurs backwards from the end of "string\_expression1". Returns the position (byte number) where "string\_expression2" was found.

### Syntax

```
instrb (string_expression1, string_expression2 [, integer_expression1
[, integer_expression2]])
```

## least

Returns the least value in "expression\_list".

### Syntax

```
least (expression_list)
```

## length

Returns the number of characters in "string\_expression".

### Syntax

```
length (string_expression)
```

## **lengthb**

Returns the number of bytes in "string\_expression".

### **Syntax**

```
lengthb (string_expression)
```

## **lpad**

Returns "string\_expression1" left-padded to the length defined by "integer\_expression" with occurrences of "string\_expression2". If "string\_expression1" is longer than "integer\_expression", the appropriate portion of "string\_expression1" is returned.

### **Syntax**

```
lpad (string_expression1, integer_expression [, string_expression2])
```

## **ltrim**

Returns "string\_expression1" with leading characters removed up to the first character not in "string\_expression2".

### **Syntax**

```
ltrim (string_expression1 [, string_expression2])
```

### **Example**

```
ltrim ('xyxXxyAB' , 'xy')
```

Result: XxyAB

## **months\_between**

Returns the number of months from "date\_expression1" to "date\_expression2". If "date\_expression1" is later than "date\_expression2" then the result will be a positive number. The days and time portion of the difference are ignored, so the months are not rounded unless "date\_expression1" and "date\_expression2" are the last days of a month.

### **Syntax**

```
months_between (date_expression1 , date_expression2)
```

## **new\_time**

Returns the datetime in "new\_timezone" for "datetime\_expression" in "old\_timezone".

"Old\_timezone" and "new\_timezone" can be one of 'AST', 'ADT', 'BST', 'BDT', 'CST', 'CDT', 'EST', 'EDT', 'HST', 'HDT', 'MST', 'MDT', 'NST', 'PST', 'PDT', 'YST', or 'YDT'.

### **Syntax**

```
new_time (datetime_expression , old_timezone , new_timezone)
```

## next\_day

Returns the datetime of the first weekday named by "string\_expression" that is later than "datetime\_expression". The return value has the same format as "datetime\_expression".

### Syntax

```
next_day (datetime_expression , string_expression)
```

## nls\_initcap

Returns "string\_expression1" with the first letter of each word in uppercase and all other letters in lowercase. A word begins after any character other than a letter. Thus, in addition to a blank space, symbols such as commas, periods, and colons can introduce a new word. "String\_expression2" specifies the sorting sequence.

### Syntax

```
nls_initcap (string_expression1 [, string_expression2])
```

## nls\_lower

Returns "string\_expression1" with all letters in lowercase. "String\_expression2" specifies the sorting sequence.

### Syntax

```
nls_lower (string_expression1 [, string_expression2])
```

## nls\_upper

Returns "string\_expression1" with all letters in uppercase. "String\_expression2" specifies the sorting sequence.

### Syntax

```
nls_upper (string_expression1 [, string_expression2])
```

## nvl

Returns "expression" unless it is null. If "expression" is null, returns "constant". Valid for "numeric\_expression", "string\_expression", "date\_expression", and "time\_expression".

### Syntax

```
nvl (expression , constant)
```

### Example

```
nvl ([Unit sale price] , [Unit price])
```

**Result:** Returns the unit sale price, or returns the unit price if the unit sale price is NULL.

## replace

Replaces all occurrences of "string\_expression2" in "string\_expression1" with "string\_expression3". If "string\_expression3" is not specified, then it removes all occurrences of "string\_expression2".

### Syntax

```
replace (string_expression1 , string_expression2 [, string_expression3])
```

## round

Returns "numeric\_expression" rounded to the nearest value "integer\_expression" places right of the decimal point. If "integer\_expression" is negative, "numeric\_expression" is rounded to the nearest absolute value "integer\_expression" places to the left of the decimal point. Rounding takes place before data formatting is applied.

### Syntax

```
round (numeric_expression [, integer_expression])
```

### Example

```
round (125 , -1)
```

Result: Returns 130

## rpad

Returns "string\_expression1" right-padded to length "integer\_expression" with occurrences of "string\_expression2". If "string\_expression1" is longer than "integer\_expression", the appropriate portion of "string\_expression1" is returned. If "string\_expression2" is not specified, then occurrences of "string\_expression2" are replaced with spaces.

### Syntax

```
rpad (string_expression1 , integer_expression [, string_expression2])
```

## rtrim

Returns "string\_expression1" with the final characters removed after the last character not in "string\_expression2". If "string\_expression2" is not specified, the final space characters are removed.

### Syntax

```
rtrim (string_expression1 [, string_expression2])
```

### Example

```
rtrim ('ABxXxyx' , 'xy')
```

Result: Returns 'ABxX'

## sign

Returns an indicator of the sign of "numeric\_expression", +1 if positive, 0 if zero, or -1 if negative.

### Syntax

```
sign (numeric_expression)
```

## soundex

Returns a character string containing the phonetic representation of "string\_expression".

### Syntax

```
soundex (string_expression)
```

## substr

Returns the substring of "string\_expression" that starts at position "integer\_expression1" for "integer\_expression2" characters or to the end of "string\_expression" if "integer\_expression2" is omitted. The first character in "string\_expression" is at position 1.

### Syntax

```
substr (string_expression , integer_expression1 [, integer_expression2])
```

### Example

```
substr ([Sales (query)].[Sales staff].[Position code], 3 , 5)
```

Result: Returns characters 3 to 7 of the position codes.

## substrb

Returns the substring of "string\_expression" that starts at position "numeric\_expression1" and ends after "numeric\_expression2" bytes. The first byte in "string\_expression" is at position 1. If you omit "numeric\_expression2", returns the substring of "string\_expression" that starts at position "numeric\_expression1" and ends at the end of "string\_expression".

### Syntax

```
substrb (string_expression , numeric_expression1 [, numeric_expression2])
```

### Example

```
substrb ([Sales (query)].[Sales staff].[Position code], 3 , 5)
```

Result: Returns characters 3 to 7 of the position codes.

## {sysdate}

Returns a datetime value representing the current date and time of the computer that the database software runs on.

### Syntax

```
{ sysdate }
```

## to\_char

Returns the string representation of "expression" with the format of "string\_expression". "Expression" can be either a date value or a numeric value.

**Syntax**

```
to_char (expression [, string_expression])
```

**to\_date**

Converts "string\_expression1" to a datetime value as specified by the format "string\_expression2". "String\_expression3" specifies the format elements, such as language.

**Syntax**

```
to_date (string_expression1 [, string_expression2 [, string_expression3]])
```

**to\_number**

Converts "string\_expression1" to a numeric value as specified by the format "string\_expression2". "String\_expression3" specifies the format elements, such as currency information.

**Syntax**

```
to_number (string_expression1 , string_expression2 , string_expression3)
```

**translate**

Returns "string\_expression1" with all occurrences of each character in "string\_expression2" replaced by the corresponding character in "string\_expression3".

**Syntax**

```
translate (string_expression1 , string_expression2 , string_expression3)
```

**trunc**

Truncates "date\_expression" using the format specified by "string\_expression". For example, if "string\_expression" is 'year', then "date\_expression" is truncated to the first day of the year.

**Syntax**

```
trunc (date_expression , string_expression)
```

**Example**

```
trunc (2003-08-22 , 'year')
```

Result: Returns 2003-01-01.

**trunc**

Truncates digits from "numeric\_expression1" using "numeric\_expression2" as the precision.

**Syntax**

```
trunc (numeric_expression1 , numeric_expression2)
```

**{user}**

Returns the username of the current Oracle user.

## Appendix A: Calculation Components

### Syntax

```
{ user }
```

## vsize

Returns the number of bytes in the internal representation of "expression". "Expression" must be a string expression.

### Syntax

```
vsize (expression)
```

## Oracle Math

### log

Returns the logarithm of "numeric\_expression2" to the base "numeric\_expression1".

### Syntax

```
log (numeric_expression1 , numeric_expression2)
```

## Oracle Trigonometry

### acos

Returns the arccosine of "numeric\_expression" in radians. The arccosine is the angle whose cosine is "numeric\_expression".

### Syntax

```
acos (numeric_expression)
```

### asin

Returns the arcsine of "numeric\_expression" in radians. The arcsine is the angle whose sine is "numeric\_expression".

### Syntax

```
asin (numeric_expression)
```

### atan

Returns the arctangent of "numeric\_expression" in radians. The arctangent is the angle whose tangent is "numeric\_expression".

### Syntax

```
atan (numeric_expression)
```

### atan2

Returns the arctangent of the x and y coordinates specified by "numeric\_expression1" and "numeric\_expression2", respectively, in radians. The arctangent is the angle whose tangent is "numeric\_expression2" / "numeric\_expression1".

**Syntax**

```
atan2 (numeric_expression1 , numeric_expression2)
```

**cos**

Returns the cosine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
cos (numeric_expression)
```

**cosh**

Returns the hyperbolic cosine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
cosh (numeric_expression)
```

**sin**

Returns the sine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
sin (numeric_expression)
```

**sinh**

Returns the hyperbolic sine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
sinh (numeric_expression)
```

**tan**

Returns the tangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
tan (numeric_expression)
```

**tanh**

Returns the hyperbolic tangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
tanh (numeric_expression)
```

## Red Brick

### ceil

Returns the smallest integer greater than or equal to "numeric\_expression" or "string\_expression". Note that "string\_expression" must represent a valid numeric value.

#### Syntax

```
ceil (numeric_expression|string_expression)
```

### concat

Returns a string that is the result of concatenating, or joining, "string\_expression1" to "string\_expression2".

#### Syntax

```
concat (string_expression1 , string_expression2)
```

#### Example

```
concat ([Sales (query)].[Sales staff].[First name], [Sales (query)].[Sales staff].[Last name])
```

Result: Returns the first name and last name; e.g., Bob Smith.

### {current\_user}

Returns the database username (authorization ID) of the current user.

#### Syntax

```
{ current_user }
```

### date

Returns a date value. "Expression" can be either characters or a timestamp.

#### Syntax

```
date (expression)
```

### dateadd

Adds "interval" to "datetime\_expression" and returns a result that is the same datetime data type as "datetime\_expression". "Datepart" refers to the year, month, day, hour, minute, second.

"Interval" must be an integer and "datetime\_expression" can be a date, time, or timestamp.

#### Syntax

```
dateadd ({ datepart } , interval, datetime_expression)
```

## **datediff**

Determines the difference between two datetime expressions and returns an integer result in "datepart" units. "Datepart" refers to a year, month, day, hour, minute, or second. "Datetime\_expression1" and "datetime\_expression2" can be dates, times, or timestamps.

### **Syntax**

```
datediff ({ datepart } , datetime_expression1, datetime_expression2)
```

## **datename**

Extracts "datepart" of "datetime\_expression" and returns its value as a character string. "Datepart" refers to a year, month, day, hour, minute, or second. "Datetime\_expression" can be a date, a time, or a timestamp.

### **Syntax**

```
datename ({ datepart } , datetime_expression)
```

## **dec**

Converts "expression" to a decimal value with the data type decimal (precision, scale). The default value of precision is 9. The default value of scale is 0.

### **Syntax**

```
dec (expression , [precision , scale])
```

## **decimal**

Converts "expression" to a decimal value with the data type decimal (precision, scale). The default value of precision is 9. The default value of scale is 0.

### **Syntax**

```
decimal (expression , [precision , scale])
```

## **decode**

Compares and converts "expression" to another value. If "expression" matches "target", it is replaced, otherwise it is replaced by "default" or null if no default is specified. The expressions can be any data type as long as they are all the same data type.

### **Syntax**

```
decode (expression , target , replacement [,default])
```

## **float**

Converts "numeric\_expression" into a double-precision floating-point value.

### **Syntax**

```
float (numeric_expression)
```

## ifnull

Tests "expression" for missing values and replaces each one with "substitute". If "expression" is null, "substitute" is returned, otherwise it returns the value of "expression". The expressions can be any data type as long as they are all the same data type.

### Syntax

```
ifnull (expression, substitute)
```

## int

Converts "numeric\_expression" into an integer value and returns an integer value. If "numeric\_expression" is null, it returns null.

### Syntax

```
int (numeric_expression)
```

## integer

Converts "numeric\_expression" into an integer value and returns an integer value. If "numeric\_expression" is null, it returns null.

### Syntax

```
integer (numeric_expression)
```

### Example

```
integer (84.95)
```

Result: 85

## length

Returns an integer result specifying the number of characters in "string\_expression". If "string\_expression" is null, it returns null.

### Syntax

```
length (string_expression)
```

## lengthb

Returns an integer result specifying the number of bytes in "string\_expression". If "string\_expression" is null, it returns null.

### Syntax

```
lengthb (string_expression)
```

## ltrim

Removes leading blanks from "string\_expression". If "string\_expression" is null, it returns null.

**Syntax**

```
ltrim (string_expression)
```

**nullif**

Returns null if both "expression1" and "expression2" have the same value. If they have different values, the value of "expression1" is returned. "Expression1" and "expression2" can be any data type as long as they are the same data type.

**Syntax**

```
nullif (expression1 , expression2)
```

**positionb**

Returns an integer that is relative to the beginning byte position of "string\_expression1" in "string\_expression2". If "string\_expression1" is not located, the result is 0. If "string\_expression1" is of zero length, the result is 1. If "string\_expression1" is null, an error message is returned. If "string\_expression2" is null, the result is 0.

**Syntax**

```
positionb (string_expression1, string_expression2)
```

**real**

Returns a real value. If "numeric\_expression" is null, it returns null.

**Syntax**

```
real (numeric_expression)
```

**round**

Returns "numeric\_expression" rounded to the nearest value "integer\_expression" places to the right of the decimal point. If "integer\_expression" is negative, "numeric\_expression" is rounded to the nearest absolute value "integer\_expression" places to the left of the decimal point. Rounding takes place before data formatting is applied.

**Syntax**

```
round (numeric_expression , integer_expression)
```

**Example**

```
round (125, -1)
```

Result: 130

**rtrim**

Removes trailing blanks from "string\_expression". If "string\_expression" is null, it returns null.

**Syntax**

```
rtrim (string_expression)
```

## Appendix A: Calculation Components

### **Example**

```
rtrim ([Sales (query)].[Sales staff].[Last name])
```

Result: Returns last names with any spaces at the end of the name removed.

## sign

Determines the sign of "numeric\_expression", and returns 1 for a positive value, -1 for a negative value, and 0 for zero.

### **Syntax**

```
sign (numeric_expression)
```

## string

Converts "expression" to a character string. "Expression" can be either numeric or datetime.

### **Syntax**

```
string (expression [, length [, scale]])
```

## substr

Returns a substring of "string\_expression" that begins at position "start\_integer" and continues for "length\_integer" characters. If "length\_integer" is not specified, a substring from "start\_integer" to the end of "string\_expression" is returned.

### **Syntax**

```
substr (string_expression , start_integer , length_integer)
```

### **Example**

```
substr ([Sales (query)].[Sales staff].[Position code], 3 , 5)
```

Result: Returns characters 3 to 7 of the position codes.

## substrb

Returns a substring of "string\_expression" that begins at position "start\_integer" and continues for "length\_integer" bytes. If "length\_integer" is not specified, a substring from "start\_integer" to the end of "string\_expression" is returned.

### **Syntax**

```
substrb (string_expression , start_integer , length_integer)
```

## time

Creates a time value from "expression", which can be a character string or a time-stamp data type expression.

### **Syntax**

```
time (expression)
```

## timestamp

Creates a time-stamp value from "timestamp\_expression", which is a character string.

### Syntax

```
timestamp (timestamp_expression)
```

## timestamp

Creates a time-stamp value from "time\_expression" and "date\_expression". If either "time\_expression" or "date\_expression" is null, the resulting time-stamp expression is also null.

### Syntax

```
timestamp (date_expression , time_expression)
```

## to\_char

Converts "source\_date" to the character string specified by "format\_string". "Source\_date" can be a date, time, or timestamp data type.

### Syntax

```
to_char (source_date, format_string)
```

## SQL Server

### ascii

Returns a number representing the ascii code value of the leftmost character of "string\_expression".

### Syntax

```
ascii (string_expression)
```

### Example

```
ascii ('A')
```

Result: 65

### char

Returns the character that has the ASCII code value specified by "integer\_expression". "Integer\_expression" should be between 0 and 255.

### Syntax

```
char (integer_expression)
```

### Example

```
char (65)
```

Result: A

## charindex

Searches "string\_expression2" for the first occurrence of "string\_expression1" and returns an integer. "Start\_location" is the character position to start searching for "string\_expression1" in "string\_expression2". If "start\_location" is not specified, is a negative number, or is zero, the search starts at the beginning of "string\_expression2".

### Syntax

```
charindex (string_expression1 , string_expression2 [, start_location])
```

## {current\_user}

Returns the name of the current user.

### Syntax

```
{ current_user }
```

## datalength

Returns the length in bytes of "string\_expression".

### Syntax

```
datalength (string_expression)
```

## dateadd

Returns the date resulting from the addition of "integer\_expression" units (indicated by "datepart" (day, month, year)) to "date\_expression".

### Syntax

```
dateadd ({ datepart } , integer_expression , date_expression)
```

## datediff

Returns the number of "datepart" (day, month, year) units between "date\_expression1" and "date\_expression2".

### Syntax

```
datediff ({datepart} , date_expression1 , date_expression2)
```

### Example

```
datediff ({yy} , 1984-01-01 , 1997-01-01)
```

Result: 13

## datename

Returns "datepart" from "date\_expression", which can be a datetime, smalldatetime, date, or time value as an ASCII string. Note that "datepart" must be a keyword representing a datepart or its abbreviation recognized by Microsoft® SQL Server and must be enclosed in curly brackets.

**Syntax**

```
datename (' { ' datepart ' } ' , date_expression)
```

**Example**

```
datename ({mm} , 2000-01-01)
```

Result: January

**datepart**

Returns part of "date\_expression" (for example, the month) as an integer. "date\_expression" can be a datetime, smalldatetime, date, or time value. Note that "datepart" must be a keyword representing a datepart or its abbreviation recognized by Microsoft® SQL Server and must be enclosed in curly brackets.

**Syntax**

```
datepart (' { ' datepart ' } ' , date_expression)
```

**Example**

```
datepart ({wk} , 2000-01-01)
```

Result: 1 (first week of the year)

**day**

Returns the day portion of "date\_expression". Same as extract (day from date\_expression).

**Syntax**

```
day (date_expression)
```

**difference**

Returns an integer value representing the difference between the values returned by the data source-specific soundex function for "string\_expression1" and "string\_expression2". The value returned ranges from 0 to 4, with 4 indicating the best match. Note that 4 does not mean that the strings are equal.

**Syntax**

```
difference (string_expression1 , string_expression2)
```

**Example 1**

```
difference ([Sales target (query)].[Sales Staff].[First name],[Sales (query)].[Retailers].[Contact first name])
```

Result: 0

**Example 2**

```
difference ([Sales target (query)].[Sales Staff].[First name],[Sales target (query)].[Sales Staff].[First name])
```

Result: 4

## getdate

Returns a datetime value representing the current date and time of the computer that the database software runs on.

### Syntax

```
getdate ()
```

## left

Returns the leftmost "integer\_expression" characters of "string\_expression".

### Syntax

```
left (string_expression , integer_expression)
```

### Example

```
left ([Sales (query)].[Sales staff].[Last name] , 3)
```

Result: Returns the first three characters of each last name.

## ltrim

Returns "string\_expression" with leading spaces removed.

### Syntax

```
ltrim (string_expression)
```

## month

Returns the month portion of "date\_expression". Same as extract (month from date\_expression).

### Syntax

```
month (date_expression)
```

## patindex

Returns an integer that represents the starting position of the first occurrence of "string\_expression1" in the "string\_expression2". Returns 0 if "string\_expression1" is not found. The % wildcard character must precede and follow "string\_expression1", except when searching for first or last characters.

### Syntax

```
patindex (string_expression1 , string_expression2)
```

### Example

```
patindex ('%po%', 'Report')
```

Result: 3

**replace**

Replaces all occurrences of "string\_expression2" in "string\_expression1" with "string\_expression3".

**Syntax**

```
replace (string_expression1 , string_expression2 , string_expression3)
```

**replicate**

Returns a string consisting of "string\_expression" repeated "integer\_expression" times.

**Syntax**

```
replicate (string_expression , integer_expression)
```

**reverse**

Returns "string\_expression" in reverse order.

**Syntax**

```
reverse (string_expression)
```

**right**

Returns the rightmost "integer\_expression" characters of "string\_expression".

**Syntax**

```
right (string_expression , integer_expression)
```

**round**

Returns "numeric\_expression" rounded to the nearest value "integer\_expression" places to the right of the decimal point. Rounding takes place before data formatting is applied.

**Syntax**

```
round (numeric_expression , integer_expression)
```

**Example**

```
round (125, -1)
```

Result: 130

**rtrim**

Returns "string\_expression" with trailing spaces removed.

**Syntax**

```
rtrim (string_expression)
```

**Example**

```
rtrim ([Sales (query)].[Sales staff].[Last name])
```

## Appendix A: Calculation Components

**Result:** Returns last names with any spaces at the end of the name removed.

### sign

Returns an indicator of the sign "numeric\_expression": +1 if "numeric\_expression" is positive, 0 if zero or -1 if negative.

#### Syntax

```
sign (numeric_expression)
```

### soundex

Returns a four character string representing the sound of the words in "string\_expression".

#### Syntax

```
soundex (string_expression)
```

### space

Returns a string consisting of "integer\_expression" spaces.

#### Syntax

```
space (integer_expression)
```

### str

Returns a string representation of "numeric\_expression" where "integer\_expression1" is the length of the string returned and "integer\_expression2" is the number of decimal digits.

#### Syntax

```
str (numeric_expression [, integer_expression1 [, integer_expression2]])
```

### stuff

Returns a string where "integer\_expression2" characters have been deleted from "string\_expression1" beginning at "integer\_expression1", and where "string\_expression2" has been inserted into "string\_expression1" at its start. The first character in a string is at position 1.

#### Syntax

```
stuff (string_expression1 , integer_expression1 , integer_expression2 , string_expression2)
```

### year

Returns the year portion of "date\_expression". Same as extract (year from date\_expression).

#### Syntax

```
year (date_expression)
```

## SQL Server Math

### log

Returns the natural logarithm of "numeric\_expression".

#### Syntax

```
log (numeric_expression)
```

### log10

Returns the base ten logarithm of "numeric\_expression".

#### Syntax

```
log10 (numeric_expression)
```

### pi

Returns the constant value of pi as a floating point value.

#### Syntax

```
pi ()
```

### rand

Generates a random number using "integer\_expression" as the seed value.

#### Syntax

```
rand (integer_expression)
```

## SQL Server Trigonometry

### acos

Returns the arccosine of "numeric\_expression" in radians. The arccosine is the angle whose cosine is "numeric\_expression".

#### Syntax

```
acos (numeric_expression)
```

### asin

Returns the arcsine of "numeric\_expression" in radians. The arcsine is the angle whose sine is "numeric\_expression".

#### Syntax

```
asin (numeric_expression)
```

### atan

Returns the arctangent of "numeric\_expression" in radians. The arctangent is the angle whose tangent is "numeric\_expression".

## Appendix A: Calculation Components

### Syntax

```
atan (numeric_expression)
```

## atn2

Returns the arctangent of the x and y coordinates specified by "numeric\_expression1" and "numeric\_expression2", respectively, in radians. The arctangent is the angle whose tangent is "numeric\_expression1".

### Syntax

```
atn2 (numeric_expression1, numeric_expression2)
```

## cos

Returns the cosine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

### Syntax

```
cos (numeric_expression)
```

## cot

Returns the cotangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

### Syntax

```
cot (numeric_expression)
```

## degrees

Returns "numeric\_expression" radians converted to degrees.

### Syntax

```
degrees (numeric_expression)
```

## radians

Returns the number of radians converted from "numeric\_expression" degrees.

### Syntax

```
radians (numeric_expression)
```

## sin

Returns the sine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

### Syntax

```
sin (numeric_expression)
```

**tan**

Returns the tangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
tan (numeric_expression)
```

**Teradata****account**

Returns the account string for the current user.

**Syntax**

```
{account}
```

**add\_months**

Returns the date or the datetime resulting from adding "integer\_expression" months to "date\_expression" or "datetime\_expression".

**Syntax**

```
add_months (date_expression|datetime_expression , integer_expression)
```

**bytes**

Returns the number of bytes contained in "byte\_expression". "Byte\_expression" is restricted to BYTE or VARBYTE.

**Syntax**

```
bytes (byte_expression)
```

**case\_n**

Evaluates "condition\_expression\_list" and returns the position of the first true condition, provided that no prior condition in the list evaluates to unknown. The keywords must be enclosed in curly brackets. No case is an optional condition that evaluates to true if every expression in the list evaluates to false. No case or unknown is an optional condition that evaluates to true if every expression in the list evaluates to false, or if an expression evaluates to unknown and all prior conditions in the list evaluate to false. Unknown is an optional condition that evaluates to true if an expression evaluates to unknown and all prior conditions in the list evaluate to false.

**Syntax**

```
case_n (condition_expression_list [, NO CASE|UNKNOWN|NO CASE OR UNKNOWN
[, UNKNOWN]])
```

## char2hexint

Returns the hexadecimal representation for "string\_expression".

### Syntax

```
char2hexint (string_expression)
```

## characters

Returns an integer value representing the number of logical characters or bytes contained in "string\_expression".

### Syntax

```
characters (string_expression)
```

## database

Returns the name of the default database for the current user.

### Syntax

```
{database}
```

## date

Returns the current date.

### Syntax

```
{date}
```

## format

Returns the declared format for "expression" as a variable character string of up to 30 characters.

### Syntax

```
format (expression)
```

## index

Returns the starting position of "string\_expression2" in "string\_expression1".

### Syntax

```
index (string_expression1 , string_expression2)
```

## log

Computes the base 10 logarithm of "numeric\_expression". "Numeric\_expression" must be a non-zero, positive, numeric expression.

### Syntax

```
log (numeric_expression)
```

**nullif**

Returns null if "scalar\_expression1" and "scalar\_expression2" are equal. Otherwise, it returns "scalar\_expression1". "Scalar\_expression1" and "scalar\_expression2" can be any data type.

**Syntax**

```
nullif (scalar_expression1 , scalar_expression2)
```

**nullifzero**

If "numeric\_expression" is zero, converts it to null to avoid division by zero.

**Syntax**

```
nullifzero (numeric_expression)
```

**profile**

Returns the current profile for the session or null if none.

**Syntax**

```
{profile}
```

**random**

Returns a random integer number for each row of the results table. "Lower\_bound" and "upper\_bound" are integer constants. The limits for "lower\_bound" and "upper\_bound" range from -2147483648 to 2147483647 inclusive. "Upper\_bound" must be greater than or equal to "lower\_bound".

**Syntax**

```
random (lower_bound , upper_bound)
```

**role**

Returns the current role for the session or null if none.

**Syntax**

```
{role}
```

**session**

Returns the number of the session for the current user.

**Syntax**

```
{session}
```

**soundex**

Returns a character string that represents the Soundex code for "string\_expression".

## Appendix A: Calculation Components

### Syntax

```
soundex (string_expression)
```

## substr

Returns the substring of "string\_expression" that starts at position "integer\_expression1" for "integer\_expression2" characters. The first character in "string\_expression" is at position 1. If "integer\_expression2" is omitted, returns the substring of "string\_expression" that starts at position "integer\_expression1" and ends at the end of "string\_expression".

### Syntax

```
substr (string_expression , integer_expression1 [, integer_expression2])
```

### Example

```
substr ([Sales (query)].[Sales staff].[Position code], 3 , 5)
```

Result: Returns characters 3 to 7 of the position codes.

## time

Returns the current time based on a 24-hour day.

### Syntax

```
{time}
```

## type

Returns the data type defined for "expression".

### Syntax

```
type (expression)
```

## user

Returns the user name of the current user.

### Syntax

```
{user}
```

## vargraphic

Returns a character string that represents the vargraphic code for "string\_expression".

### Syntax

```
vargraphic (string_expression)
```

## zeroifnull

Converts data from null to 0 to avoid errors created by a null value. If "numeric\_expression" is not null, returns the value of "numeric\_expression". If "numeric\_expression" is a character string,

it is converted to a numeric value of float data type. If "numeric\_expression" is null or zero, it returns zero.

### Syntax

```
zeroifnull (numeric_expression)
```

## Teradata Trigonometry

### acos

Returns the arccosine of "numeric\_expression" in radians. The arccosine is the angle whose cosine is "numeric\_expression". "Numeric\_expression" must be between -1 and 1, inclusive.

### Syntax

```
acos (numeric_expression)
```

### acosh

Returns the inverse hyperbolic cosine of "numeric\_expression" where "numeric\_expression" can be any real number equal to or greater than 1.

### Syntax

```
acosh (numeric_expression)
```

### asin

Returns the arcsine of "numeric\_expression" in radians. The arcsine is the angle whose sine is "numeric\_expression". "Numeric\_expression" must be between -1 and 1, inclusive.

### Syntax

```
asin (numeric_expression)
```

### asinh

Returns the inverse hyperbolic sine of "numeric\_expression" where "numeric\_expression" can be any real number.

### Syntax

```
asinh (numeric_expression)
```

### atan

Returns the arctangent of "numeric\_expression" in radians where the arctangent is the angle whose tangent is "numeric\_expression".

### Syntax

```
atan (numeric_expression)
```

## Appendix A: Calculation Components

### atan2

Returns the arctangent of the x and y coordinates specified by "numeric\_expression1" and "numeric\_expression2", respectively, in radians. The returned angle will be between - and  $\pi$  radians, excluding  $\pi$ .

#### Syntax

```
atan2 (numeric_expression1, numeric_expression2)
```

### atanh

Returns the inverse hyperbolic tangent of "numeric\_expression" where "numeric\_expression" can be any real number between 1 and -1, excluding 1 and -1.

#### Syntax

```
atanh (numeric_expression)
```

### cos

Returns the cosine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

#### Syntax

```
cos (numeric_expression)
```

### cosh

Returns the hyperbolic cosine of "numeric\_expression" where "numeric\_expression" can be any real number.

#### Syntax

```
cosh (numeric_expression)
```

### sin

Returns the sine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

#### Syntax

```
sin (numeric_expression)
```

### sinh

Returns the hyperbolic sine of "numeric\_expression" where "numeric\_expression" can be any real number.

#### Syntax

```
sinh (numeric_expression)
```

### tan

Returns the tangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
tan (numeric_expression)
```

**tanh**

Returns the hyperbolic tangent of "numeric\_expression" where "numeric\_expression" can be any real number.

**Syntax**

```
tanh (numeric_expression)
```

**SAP BW****SAP BW Trigonometry****arccos**

Returns the arccosine of "numeric\_expression" in radians. The arccosine is the angle whose cosine is "numeric\_expression".

**Syntax**

```
arccos (numeric_expression)
```

**arcsin**

Returns the arcsine of "numeric\_expression" in radians. The arcsine is the angle whose sine is "numeric\_expression".

**Syntax**

```
arcsin (numeric_expression)
```

**arctan**

Returns the arctangent of "numeric\_expression" in radians. The arctangent is the angle whose tangent is "numeric\_expression".

**Syntax**

```
arctan (numeric_expression)
```

**cos**

Returns the cosine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
cos (numeric_expression)
```

**sin**

Returns the sine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

## Appendix A: Calculation Components

### Syntax

```
sin (numeric_expression)
```

## tan

Returns the tangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

### Syntax

```
tan (numeric_expression)
```

## coshyp

Returns the hyperbolic cosine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

### Syntax

```
coshyp (numeric_expression)
```

## sinhyp

Returns the hyperbolic sine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

### Syntax

```
sinhyp (numeric_expression)
```

## tanhyp

Returns the hyperbolic tangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

### Syntax

```
tanhyp (numeric_expression)
```

## SAP BW Math

### log10

Returns the base ten logarithm of "numeric\_expression".

### Syntax

```
log10 (numeric_expression)
```

## Sybase

### ascii

Returns a number representing the ascii code value of the leftmost character of "string\_expression".

**Syntax**

```
ascii (string_expression)
```

**Example**

```
ascii ('A')
```

Result: 65

**char**

Converts "integer\_expression" to a character value. Char is usually used as the inverse of ascii where "integer\_expression" must be between 0 and 255. If the resulting value is the first byte of a multibyte character, the character may be undefined.

**Syntax**

```
char (integer_expression)
```

**charindex**

Returns an integer that represents the starting position of "string\_expression1" within "string\_expression2". If "string\_expression1" is not found, zero is returned. If "string\_expression1" contains wildcard characters, charindex treats them as literals.

**Syntax**

```
charindex (string_expression1 , string_expression2)
```

**datalength**

Returns the length in bytes of "string\_expression".

**Syntax**

```
datalength (string_expression)
```

**dateadd**

Returns the date resulting from adding "integer\_expression" units indicated by datepart (day, month, year) to "date\_expression". Note that "datepart" must be enclosed in curly brackets.

**Syntax**

```
dateadd (' { ' datepart ' } ' , integer_expression , date_expression)
```

**Example**

```
dateadd ({dd} , 16 , 1997-06-16)
```

Result: Jul 2, 1997

**datediff**

Returns the number of units indicated by "datepart" (day, month, year) between "date\_expression1" and "date\_expression2". Note that "datepart" must be enclosed in curly brackets.

## Appendix A: Calculation Components

### Syntax

```
datediff (' { ' datepart ' } ' , date_expression1 , date_expression2)
```

### Example

```
datediff ({yy} , 1984-01-01 , 1997-01-01)
```

Result: 13

## datename

Returns "datepart" of "date\_expression" as an ASCII string. "Date\_expression" can be a datetime, smalldatetime, date, or time value. Note that "datepart" must be enclosed in curly brackets.

### Syntax

```
datename (' { ' datepart ' } ' , date_expression)
```

### Example

```
datename ({mm} , 1999-05-01)
```

Result: May

## datepart

Returns "datepart" of "date\_expression" as an integer. "Date\_expression" can be a datetime, smalldatetime, date, or time value. Note that the datepart argument must be enclosed in curly brackets.

### Syntax

```
datepart (' { ' datepart ' } ' , date_expression)
```

### Example

```
datepart ({mm} , 1999-05-01)
```

Result: 5

## day

Returns the day of the month (1-31) from "date\_expression".

### Syntax

```
day (date_expression)
```

## difference

Returns an integer value representing the difference between the values returned by the data source-specific soundex function for "string\_expression1" and "string\_expression2". The value that is returned ranges from 0 to 4, with 4 indicating the best match. Note that 4 does not mean that the strings are equal.

### Syntax

```
difference (string_expression1 , string_expression2)
```

**Example 1**

```
difference ([Sales target (query)].[Sales staff].[First name],[Sales (query)].[Retailers].[Contact first name])
```

Result: 0

**Example 2**

```
difference ([Sales target (query)].[Sales staff].[First name],[Sales target (query)].[Sales staff].[First name])
```

Result: 4

**getdate**

Returns current system date and time.

**Syntax**

```
getdate ()
```

**left**

Returns the leftmost "integer\_expression" characters of "string\_expression".

**Syntax**

```
left (string_expression , integer_expression)
```

**Example**

```
left ([Sales (query)].[Sales staff].[Last name] , 3)
```

Result: Returns the first three characters of each last name.

**ltrim**

Returns "string\_expression" with any leading spaces removed.

**Syntax**

```
ltrim (string_expression)
```

**month**

Returns the month number (1-12) from "date\_expression".

**Syntax**

```
month (date_expression)
```

**patindex**

Returns an integer representing the starting position of the first occurrence of "string\_expression1" in "string\_expression2" or returns 0 if "string\_expression1" is not found. By default, patindex returns the offset in characters. The offset can be returned in bytes by setting the return type to bytes. The % wildcard character must precede and follow the pattern in "string\_expression1", except when searching for first or last characters.

## Appendix A: Calculation Components

### Syntax

```
patindex (string_expression1 , string_expression2 [using {bytes|chars|characters}])
```

## rand

Returns a random float value between 0 and 1, using the optional "integer\_expression" as a seed value.

### Syntax

```
rand (integer_expression)
```

## replicate

Returns a string with the same datatype as "string\_expression", containing the same expression repeated "integer\_expression" times or as many times as will fit into a 225-byte space, whichever is less.

### Syntax

```
replicate (string_expression , integer_expression)
```

## reverse

Returns the reverse of "string\_expression".

### Syntax

```
reverse (string_expression)
```

## right

Returns the rightmost "integer\_expression" characters of "string\_expression".

### Syntax

```
right (string_expression , integer_expression)
```

## round

Returns "numeric\_expression" rounded to the nearest value "integer\_expression" places to the right of the decimal point. Rounding takes place before data formatting is applied.

### Syntax

```
round (numeric_expression, integer_expression)
```

### Example

```
round (125, -1)
```

Result: 130

## rtrim

Returns "string\_expression" with trailing spaces removed.

**Syntax**

```
rtrim (string_expression)
```

**Example**

```
rtrim ([Sales (query)].[Sales staff].[Last name])
```

Result: Returns last names with any spaces at the end of the name removed.

**soundex**

Returns a four-character soundex code for character strings that are composed of a contiguous sequence of valid single- or double-byte Roman letter.

**Syntax**

```
soundex (string_expression)
```

**space**

Returns a string with "integer\_expression" single-byte spacing.

**Syntax**

```
space (integer_expression)
```

**str**

Returns a string representation of "numeric\_expression". "Integer\_expression1" is the length of the returned string and has a default setting of 10. "Integer\_expression2" is the number of decimal digits and has a default setting of 0. Both are optional values.

**Syntax**

```
str (numeric_expression [, integer_expression1 [, integer_expression2]])
```

**stuff**

Deletes "integer\_expression2" characters from "string\_expression1" starting at "integer\_expression1", and inserts "string\_expression2" into "string\_expression1" at that position. To delete characters without inserting other characters, "string\_expression2" should be null and not " ", which indicates a single space.

**Syntax**

```
stuff (string_expression1 , integer_expression1 , integer_expression2 , string_expression2)
```

**substring**

Returns the substring of "string\_expression" that starts at position "integer\_expression1". "Integer\_expression2" specifies the number of characters in the substring.

**Syntax**

```
substring (string_expression , integer_expression1 , integer_expression2)
```

### Example

```
substring ([Sales (query)].[Sales staff].[Position code], 3 , 5)
```

Result: Returns characters 3 to 7 of the position codes.

## to\_unichar

Returns a unichar expression with the value "integer\_expression". If "integer\_expression" is in the range 0xD800..0xDFFF, the operation is aborted. If the "integer\_expression" is in the range 0..0xFFFF, a single Unicode value is returned. If "integer\_expression" is in the range 0x10000..0x10FFFF, a surrogate pair is returned.

### Syntax

```
to_unichar (integer_expression)
```

## uhighsurr

Returns 1 if the Unicode value at "integer\_expression" is the high half of a surrogate pair (which should appear first in the pair). Otherwise, it returns 0. This function allows you to write explicit code for surrogate handling. Particularly, if a substring starts on a Unicode character where uhightsurr () is true, extract a substring of at least 2 Unicode values, as substr() does not extract just 1. Substr () does not extract half of a surrogate pair.

### Syntax

```
uhightsurr (string_expression , integer_expression)
```

## ulowsurr

Returns 1 if the Unicode value at "integer\_expression" is the low half of a surrogate pair (which should appear second in the pair). Otherwise, it returns 0. This function allows you to explicitly code around the adjustments performed by substr (), stuff (), and right (). Particularly, if a substring ends on a Unicode value where ulowsurr () is true, extract a substring of 1 less characters (or 1 more), since substr () does not extract a string that contains an unmatched surrogate pair.

### Syntax

```
ulowsurr (string_expression , integer_expression)
```

## uscalar

Returns the Unicode scalar value for the first Unicode character in "string\_expression". If the first character is not the high-order half of a surrogate pair, then the value is in the range 0..0xFFFF. If the first character is the high-order half of a surrogate pair, a second value must be a low-order half, and the return value is in the range 0x10000..0x10FFFF. If this function is called on a Unicode character expression containing an unmatched surrogate half, the operation is aborted.

### Syntax

```
uscalar (string_expression)
```

**year**

Returns the year from date\_expression.

**Syntax**

```
year (date_expression)
```

**Sybase Math****log**

Returns the natural logarithm of "numeric\_expression".

**Syntax**

```
log (numeric_expression)
```

**log10**

Returns the base ten logarithm of "numeric\_expression".

**Syntax**

```
log10 (numeric_expression)
```

**pi**

Returns the constant value of pi as a floating point value.

**Syntax**

```
pi ()
```

**sign**

Returns an indicator denoting the sign of "numeric\_expression": +1 if "numeric\_expression" is positive, 0 if "numeric\_expression" is zero, or -1 if "numeric\_expression" is negative.

**Syntax**

```
sign (numeric_expression)
```

**Sybase Trigonometry****acos**

Returns the arccosine of "numeric\_expression" in radians. The arccosine is the angle whose cosine is "numeric\_expression".

**Syntax**

```
acos (numeric_expression)
```

**asin**

Returns the arcsine of "numeric\_expression" in radians. The arcsine is the angle whose sine is "numeric\_expression".

## Appendix A: Calculation Components

### Syntax

```
asin (numeric_expression)
```

## atan

Returns the arctangent of "numeric\_expression" in radians. The arctangent is the angle whose tangent is "numeric\_expression".

### Syntax

```
atan (numeric_expression)
```

## tan

Returns the tangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

### Syntax

```
tan (numeric_expression)
```

## atn2

Returns the angle, in radians, whose tangent is "numeric\_expression1" / "numeric\_expression2".

### Syntax

```
atn2 (numeric_expression1, numeric_expression2)
```

## cos

Returns the cosine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

### Syntax

```
cos (numeric_expression)
```

## cot

Returns the cotangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

### Syntax

```
cot (numeric_expression)
```

## degrees

Returns "numeric\_expression" radians converted to degrees.

### Syntax

```
degrees (numeric_expression)
```

## radians

Returns the degree equivalent of "numeric\_expression". Results are of the same type as "numeric\_expression". For numeric or decimal expressions, the results have an internal precision of 77 and

a scale equal to that of "numeric\_expression". When the money datatype is used, an internal conversion to float may cause some loss of precision.

### Syntax

```
radians (numeric_expression)
```

## sin

Returns the sine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

### Syntax

```
sin (numeric_expression)
```

# Postgres

## Postgres String

### overlay

Returns the "string\_expression1" replacing "string\_expression2" from character position numeric\_expression.

### Syntax

```
overlay (string_expression1 , string_expression2 , numeric_expression1 [, numeric_expression2])
```

### btrim

Returns string\_expression1 after removing the longest string of characters in "string\_expression2".

### Syntax

```
btrim (string_expression1 [, string_expression2])
```

### initcap

Returns "string\_expression", with the first letter of each word in uppercase and all other letters in lowercase. Words are delimited by white space or characters that are not alphanumeric.

### Syntax

```
initcap (string_expression)
```

### lpad

Returns "string\_expression1" padded to length "integer\_expression" with occurrences of "string\_expression2". If "string\_expression1" is longer than "integer\_expression", the appropriate portion of "string\_expression1" is returned.

### Syntax

```
lpad (string_expression1 , integer_expression [, string_expression2])
```

## Appendix A: Calculation Components

### ltrim

Returns "string\_expression1", with leading characters removed up to the first character not in "string\_expression2"; for example, ltrim ( 'xyxXxyAB' , 'xy' ) returns XxyAB.

#### Syntax

```
ltrim (string_expression1 [, string_expression2])
```

### md5

Returns the MD5 hash of "string\_expression1".

#### Syntax

```
md5 (string_expression1)
```

### to\_hex

Returns the hexadecimal string representation of "numeric\_expression1".

#### Syntax

```
to_hex (numeric_expression1)
```

### repeat

Returns the "string\_expression" repeated "numeric\_expression1" times.

#### Syntax

```
repeat (string_expression , numeric_expression1)
```

### replace

Returns "string\_expression" with "string\_expression2" replaced with "string\_expression3".

#### Syntax

```
replace (string_expression , string_expression2 , string_expression3)
```

### rpad

Returns "string\_expression1" right-padded to length "integer\_expression" with occurrences of "string\_expression2". If "string\_expression1" is longer than "integer\_expression", the appropriate portion of "string\_expression1" is returned. If "string\_expression2" is not specified, then spaces are used.

#### Syntax

```
rpad (string_expression1 , integer_expression [, string_expression2])
```

### rtrim

Returns "string\_expression1", with final characters removed after the last character not in "string\_expression2"; for example, rtrim ( 'ABxXxyx' , 'xy' ) returns ABxX. If "string\_expression2" is not specified, the final space characters are removed.

**Syntax**

```
rtrim (string_expression1 [, string_expression2])
```

**split\_part**

Returns "numeric\_expression" field having split "string\_expression1" on "string\_expression2".

**Syntax**

```
split_part (string_expression1 , string_expression2 , numeric_expression)
```

**ascii**

Returns a number representing the ASCII code value of the leftmost character of "string\_expression"; for example, ascii ( 'A' ) is 65.

**Syntax**

```
ascii (string_expression)
```

**chr**

Returns the character that has the ASCII code value specified by "integer\_expression". "Integer\_expression" should be between 0 and 255.

**Syntax**

```
chr (integer_expression)
```

**{current\_catalog}****Syntax**

```
{current_catalog}
```

**{current\_db}****Syntax**

```
{current_db}
```

**{current\_schema}****Syntax**

```
{current_schema}
```

**{current\_user}****Syntax**

```
{current_user}
```

## {session\_user}

### Syntax

```
{session_user}
```

## Postgres Data type formatting

### to\_char

Returns the string representation of "expression" with the format "string\_expression".  
"Expression" can be either a date value or a numeric value.

### Syntax

```
to_char (expression , string_expression)
```

### to\_date

Converts "string\_expression1" to a date value as specified by the format "string\_expression2".

### Syntax

```
to_date (string_expression1 , string_expression2)
```

### to\_number

Converts "string\_expression1" to a numeric value as specified by the format "string\_expression2".

### Syntax

```
to_number (string_expression1 , string_expression2)
```

### to\_timestamp

Converts "string\_expression1" to a timestamp value as specified by the format "string\_expression2".

### Syntax

```
to_timestamp (string_expression1 , string_expression2)
```

### translate

Returns "string\_expression1", with each occurrence of each character in "string\_expression2" replaced by its corresponding character in "string\_expression3".

### Syntax

```
translate (string_expression1 , string_expression2 , string_expression3)
```

### date\_trunc

Returns the timestamp to the specified precision.

### Syntax

```
date_trunc (string_expression , timestamp_expression)
```

## version

Returns the string value of the database version.

### Syntax

```
version ()
```

## Postgres Math

### log

Returns the base 10 logarithm of "numeric\_expression1" or logarithm to the base "numeric\_expression2".

### Syntax

```
log (numeric_expression1 [, numeric_expression2])
```

### ln

Returns the natural logarithm of "numeric\_expression1".

### Syntax

```
ln (numeric_expression)
```

### cbrt

Returns the cube root of "numeric\_expression1".

### Syntax

```
cbrt (numeric_expression)
```

### div

Returns the integer quotient of "numeric\_expression1" divided by "numeric\_expression2".

### Syntax

```
div (numeric_expression1 , numeric_expression2)
```

### pi

Returns the constant of pi.

### Syntax

```
pi ()
```

## Postgres Trigonometry

### acos

Returns the arccosine of "numeric\_expression" in radians. The arccosine is the angle whose cosine is "numeric\_expression".

## Appendix A: Calculation Components

### Syntax

```
acos (numeric_expression)
```

## asin

Returns the arcsine of "numeric\_expression" in radians. The arcsine is the angle whose sine is "numeric\_expression".

### Syntax

```
asin (numeric_expression)
```

## atan

Returns the arctangent of "numeric\_expression" in radians. The arctangent is the angle whose tangent is "numeric\_expression".

### Syntax

```
atan (numeric_expression)
```

## atan2

Returns the arctangent of the x and y coordinates specified by "numeric\_expression1" and "numeric\_expression2", respectively, in radians. The arctangent is the angle whose tangent is "numeric\_expression2" / "numeric\_expression1".

### Syntax

```
atan2 (numeric_expression1 , numeric_expression2)
```

## cos

Returns the cosine of "numeric\_expression", where "numeric\_expression" is an angle expressed in radians.

### Syntax

```
cos (numeric_expression)
```

## cot

Returns the cotangent of "numeric\_expression", where "numeric\_expression" is an angle expressed in radians.

### Syntax

```
cot (numeric_expression)
```

## degrees

Returns the degrees where "numeric\_expression" is an angle expressed in radians.

### Syntax

```
degrees (numeric_expression)
```

**radians**

Returns the radians where "numeric\_expression" is an angle expressed in degrees.

**Syntax**

```
radians (numeric_expression)
```

**sin**

Returns the sine of "numeric\_expression", where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
sin (numeric_expression)
```

**tan**

Returns the tangent of "numeric\_expression", where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
tan (numeric_expression)
```

**Vertica****Vertica String****overlay**

Returns the "string\_expression1", replacing "string\_expression2" from character position numeric\_expression.

**Syntax**

```
overlay (string_expression1 , string_expression2 , numeric_expression1 [, numeric_expression2])
```

**btrim**

Returns string\_expression1 after removing the longest string of characters in string\_expression2.

**Syntax**

```
btrim (string_expression1 [, string_expression2])
```

**initcap**

Returns "string\_expression", with the first letter of each word in uppercase and all other letters in lowercase. Words are delimited by white space or characters that are not alphanumeric.

**Syntax**

```
initcap (string_expression)
```

## Appendix A: Calculation Components

### lpad

Returns "string\_expression1" padded to length "integer\_expression" with occurrences of "string\_expression2". If "string\_expression1" is longer than "integer\_expression", the appropriate portion of "string\_expression1" is returned.

#### Syntax

```
lpad (string_expression1 , integer_expression [, string_expression2])
```

### ltrim

Returns "string\_expression1", with leading characters removed up to the first character not in "string\_expression2"; for example, ltrim ( 'xyxXxyAB' , 'xy' ) returns XxyAB.

#### Syntax

```
ltrim (string_expression1 [, string_expression2])
```

### to\_hex

Returns the hexadecimal string representation of "numeric\_exp1".

#### Syntax

```
to_hex (numeric_expression1)
```

### repeat

Returns the "string\_expression" repeated "numeric\_expression1" times.

#### Syntax

```
repeat (string_expression , numeric_expression1)
```

### replace

Returns "string\_expression" having replaced "string\_expression2" with "string\_expression3".

#### Syntax

```
replace (string_expression , string_expression2 , string_expression3)
```

### rpad

Returns "string\_expression1" right-padded to length "integer\_expression" with occurrences of "string\_expression2". If "string\_expression1" is longer than "integer\_expression", the appropriate portion of "string\_expression1" is returned. If "string\_expression2" is not specified, then spaces are used.

#### Syntax

```
rpad (string_expression1 , integer_expression [, string_expression2])
```

### rtrim

Returns "string\_expression1", with final characters removed after the last character not in "string\_expression2"; for example, rtrim ( 'ABxXxyx' , 'xy' ) returns ABxX. If "string\_expression2" is not specified, the final space characters are removed.

**Syntax**

```
rtrim (string_expression1 [, string_expression2])
```

**ascii**

Returns a number representing the ASCII code value of the leftmost character of "string\_expression"; for example, ascii ( 'A' ) is 65.

**Syntax**

```
ascii (string_expression)
```

**chr**

Returns the character that has the ASCII code value specified by "integer\_expression". "Integer\_expression" should be between 0 and 255.

**Syntax**

```
chr (integer_expression)
```

**current\_database**

Returns the name of the current database.

**Syntax**

```
current_database ()
```

**current\_schema**

Returns the name of the current schema

**Syntax**

```
current_schema ()
```

**{current\_user}****Syntax**

```
{current_user}
```

**{session\_user}****Syntax**

```
{session_user}
```

**Vertica Data type formatting****to\_char**

Returns the string representation of "expression" with the format of "string\_expression". "Expression" can be either a date value or a numeric value.

## Appendix A: Calculation Components

### Syntax

```
to_char (expression , string_expression)
```

## to\_date

Converts "string\_expression1" to a date value as specified by the format "string\_expression2".

### Syntax

```
to_date (string_expression1 , string_expression2)
```

## to\_number

Converts "string\_expression1" to a numeric value as specified by the format "string\_expression2".

### Syntax

```
to_number (string_expression1 , string_expression2)
```

## to\_timestamp

Converts "string\_expression1" to a timestamp value as specified by the format "string\_expression2".

### Syntax

```
to_timestamp (string_expression1 , string_expression2)
```

## translate

Returns "string\_expression1", with each occurrence of each character in "string\_expression2" replaced by its corresponding character in "string\_expression3".

### Syntax

```
translate (string_expression1 , string_expression2 , string_expression3)
```

## date\_trunc

Returns the timestamp to the specified precision.

### Syntax

```
date_trunc (string_expression , timestamp_expression)
```

## version

Returns the string value of the database version.

### Syntax

```
version ()
```

## Vertica Math

### log

Returns the base 10 logarithm of "numeric\_expression1" or logarithm to the base "numeric\_expression2".

**Syntax**

```
log (numeric_expression1 [, numeric_expression2])
```

**In**

Returns the natural logarithm of "numeric\_expression1".

**Syntax**

```
ln (numeric_expression)
```

**cbrt**

Returns the cube root of "numeric\_expression1".

**Syntax**

```
cbrt (numeric_expression)
```

**pi**

Returns the constant of pi.

**Syntax**

```
pi ()
```

**Vertica Trigonometry****acos**

Returns the arccosine of "numeric\_expression" in radians. The arccosine is the angle whose cosine is "numeric\_expression".

**Syntax**

```
acos (numeric_expression)
```

**asin**

Returns the arcsine of "numeric\_expression" in radians. The arcsine is the angle whose sine is "numeric\_expression".

**Syntax**

```
asin (numeric_expression)
```

**atan**

Returns the arctangent of "numeric\_expression" in radians. The arctangent is the angle whose tangent is "numeric\_expression".

**Syntax**

```
atan (numeric_expression)
```

## Appendix A: Calculation Components

### atan2

Returns the arctangent of the x and y coordinates specified by "numeric\_expression1" and "numeric\_expression2", respectively, in radians. The arctangent is the angle whose tangent is "numeric\_expression2" / "numeric\_expression1".

#### Syntax

```
atan2 (numeric_expression1 , numeric_expression2)
```

### cos

Returns the cosine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

#### Syntax

```
cos (numeric_expression)
```

### cot

Returns the cotangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

#### Syntax

```
cot (numeric_expression)
```

### degrees

Returns the degrees where "numeric\_expression" is an angle expressed in radians.

#### Syntax

```
degrees (numeric_expression)
```

### radians

Returns the radians where "numeric\_expression" is an angle expressed in degrees.

#### Syntax

```
radians (numeric_expression)
```

### sin

Returns the sine of "numeric\_exp" where "numeric\_expression" is an angle expressed in radians.

#### Syntax

```
sin (numeric_expression)
```

### tan

Returns the tangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

#### Syntax

```
tan (numeric_expression)
```

# Paracel

## Paracel String

### **overlay**

Returns the "string\_expression1", replacing "string\_expression2" from character position numeric\_expression.

#### **Syntax**

```
overlay (string_expression1 , string_expression2 , numeric_expression1 [, numeric_expression2])
```

### **ltrim**

Returns "string\_expression1", with leading characters removed up to the first character not in "string\_expression2"; for example, ltrim ( 'xyxXxyAB' , 'xy' ) returns XxyAB.

#### **Syntax**

```
ltrim (string_expression1 [, string_expression2])
```

### **replace**

Returns "string\_expression", having replaced "string\_expression2" with "string\_expression3".

#### **Syntax**

```
replace (string_expression , string_expression2 , string_expression3)
```

### **rtrim**

Returns "string\_expression1", with final characters removed after the last character not in "string\_expression2"; for example, rtrim ( 'ABxXxyx' , 'xy' ) returns ABxX. If "string\_expression2" is not specified, the final space characters are removed.

#### **Syntax**

```
rtrim (string_expression1 [, string_expression2])
```

## **current\_database**

Returns the name of the current database.

#### **Syntax**

```
current_database ()
```

## **current\_schema**

Returns the name of the current schema

#### **Syntax**

```
current_schema ()
```

## {current\_user}

### Syntax

```
{current_user}
```

## {session\_user}

### Syntax

```
{session_user}
```

## Paraccel Data type formatting

### to\_char

Returns the string representation of "expression" with the format of "string\_expression". "Expression" can be either a date value or a numeric value.

### Syntax

```
to_char (expression , string_expression)
```

### to\_date

Converts "string\_expression1" to a date value as specified by the format "string\_expression2".

### Syntax

```
to_date (string_expression1 , string_expression2)
```

### to\_number

Converts "string\_expression1" to a numeric value as specified by the format "string\_expression2".

### Syntax

```
to_number (string_expression1 , string_expression2)
```

### translate

Returns "string\_expression1", with each occurrence of each character in "string\_expression2" replaced by its corresponding character in "string\_expression3".

### Syntax

```
translate (string_expression1 , string_expression2 , string_expression3)
```

### version

Returns the string value of the database version.

### Syntax

```
version ()
```

## Paracel Math

### **cbrt**

Returns the cube root of "numeric\_expression1".

#### **Syntax**

```
cbrt (numeric_expression)
```

### **pi**

Returns the constant of pi.

#### **Syntax**

```
pi ()
```

## MySQL

### MySQL String

#### **lpad**

Returns "string\_expression1" padded to length "integer\_expression" with occurrences of "string\_expression2". If "string\_expression1" is longer than "integer\_expression", the appropriate portion of "string\_expression1" is returned.

#### **Syntax**

```
lpad (string_expression1 , integer_expression [, string_expression2])
```

#### **ltrim**

Returns "string\_expression1", with leading characters removed up to the first character not in "string\_expression2"; for example, ltrim ( 'xyxXxyAB' , 'xy' ) returns XxyAB.

#### **Syntax**

```
ltrim (string_expression1 [, string_expression2])
```

#### **hex**

Returns the hexadecimal string representation of "numeric\_expression1".

#### **Syntax**

```
hex (numeric_expression1)
```

#### **repeat**

Returns the "string\_expression" repeated "numeric\_expression1" times.

#### **Syntax**

```
repeat (string_expression , numeric_expression1)
```

## Appendix A: Calculation Components

### replace

Returns "string\_expression" having replaced "string\_expression2" with "string\_expression3".

#### Syntax

```
replace (string_expression , string_expression2 , string_expression3)
```

### reverse

Returns "string\_expression" reversed.

#### Syntax

```
reverse (string_expression)
```

### right

Returns the rightmost "numeric\_expression" characters from "string\_expression1".

#### Syntax

```
right (string_expression1 , numeric_expression)
```

### rpad

Returns "string\_expression1" right-padded to length "integer\_expression" with occurrences of "string\_expression2". If "string\_expression1" is longer than "integer\_expression", the appropriate portion of "string\_expression1" is returned. If "string\_expression2" is not specified, then spaces are used.

#### Syntax

```
rpad (string_expression1 , integer_expression [, string_expression2])
```

### rtrim

Returns "string\_expression1", with final characters removed after the last character not in "string\_expression2"; for example, rtrim ( 'ABxXxyx' , 'xy' ) returns ABxX. If "string\_expression2" is not specified, the final space characters are removed.

#### Syntax

```
rtrim (string_expression1 [, string_expression2])
```

### soundex

Returns a soundex string of "string\_expression1".

#### Syntax

```
soundex (string_expression1)
```

### ascii

Returns a number representing the ASCII code value of the leftmost character of "string\_expression"; for example, ascii ( 'A' ) is 65.

**Syntax**

```
ascii (string_expression)
```

**database**

Returns the current database name

**Syntax**

```
database ()
```

**schema**

Returns the current schema name

**Syntax**

```
schema ()
```

**session\_user**

Return the user name returned by the client

**Syntax**

```
session_user ()
```

**system\_user**

Return the user name returned by the client

**Syntax**

```
system_user ()
```

**version**

Returns the string value of the database version.

**Syntax**

```
version ()
```

**MySQL Math****log**

Returns the base 10 logarithm of "numeric\_expression1" or logarithm to the base "numeric\_expression2".

**Syntax**

```
log (numeric_expression)
```

**ln**

Returns the natural logarithm of "numeric\_expression1".

## Appendix A: Calculation Components

### Syntax

```
ln (numeric_expression)
```

### pi

Returns the constant of pi.

### Syntax

```
pi ()
```

## MySQL Trigonometry

### acos

Returns the arccosine of "numeric\_expression" in radians. The arccosine is the angle whose cosine is "numeric\_expression".

### Syntax

```
acos (numeric_expression)
```

### asin

Returns the arcsine of "numeric\_expression" in radians. The arcsine is the angle whose sine is "numeric\_expression".

### Syntax

```
asin (numeric_expression)
```

### atan

Returns the arctangent of "numeric\_expression" in radians. The arctangent is the angle whose tangent is "numeric\_expression".

### Syntax

```
atan (numeric_expression)
```

### atan2

Returns the arctangent of the x and y coordinates specified by "numeric\_expression1" and "numeric\_expression2", respectively, in radians. The arctangent is the angle whose tangent is "numeric\_expression2" / "numeric\_expression1".

### Syntax

```
atan2 (numeric_expression1 , numeric_expression2)
```

### cos

Returns the cosine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

### Syntax

```
cos (numeric_expression)
```

**cot**

Returns the cotangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
cot (numeric_expression)
```

**degrees**

Returns the degrees where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
degrees (numeric_expression)
```

**radians**

Returns the radians where "numeric\_expression" is an angle expressed in degrees.

**Syntax**

```
radians (numeric_expression)
```

**sin**

Returns the sine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
sin (numeric_expression)
```

**tan**

Returns the tangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
tan (numeric_expression)
```

# Greenplum

## Greenplum String

**overlay**

Returns the "string\_expression1" replacing "string\_expression2" from character position "numeric\_expression".

**Syntax**

```
overlay (string_expression1 , string_expression2 , numeric_expression1 [,
numeric_expression2])
```

## Appendix A: Calculation Components

### btrim

Returns "string\_expression1" after removing the longest string of characters in "string\_expression2".

#### Syntax

```
btrim (string_expression1 [, string_expression2])
```

### initcap

Returns "string\_expression" with the first letter of each word in uppercase and all other letters in lowercase. Words are delimited by white space or characters that are not alphanumeric.

#### Syntax

```
initcap (string_expression)
```

### lpad

Returns "string\_expression1" padded to length "integer\_expression" with occurrences of "string\_expression2". If "string\_expression1" is longer than "integer\_expression", the appropriate portion of "string\_expression1" is returned.

#### Syntax

```
lpad (string_expression1 , integer_expression [, string_expression2])
```

### ltrim

Returns "string\_expression1", with leading characters removed up to the first character not in "string\_expression2"; for example, ltrim ( 'xyXxyAB' , 'xy' ) returns XxyAB.

#### Syntax

```
ltrim (string_expression1 [, string_expression2])
```

### md5

Returns the MD5 hash of "string\_expression1".

#### Syntax

```
md5 (string_expression1)
```

### to\_hex

Returns the hexadecimal string representation of "numeric\_expression1".

#### Syntax

```
to_hex (numeric_expression1)
```

### repeat

Returns the "string\_expression" repeated "numeric\_expression1" times.

#### Syntax

```
repeat (string_expression , numeric_expression1)
```

**replace**

Returns "string\_expression" having replaced "string\_expression2" with "string\_expression3".

**Syntax**

```
replace (string_expression , string_expression2 , string_expression3)
```

**rpad**

Returns "string\_expression1" right-padded to length "integer\_expression" with occurrences of "string\_expression2". If "string\_expression1" is longer than "integer\_expression", the appropriate portion of "string\_expression1" is returned. If "string\_expression2" is not specified, then spaces are used.

**Syntax**

```
rpad (string_expression1 , integer_expression [, string_expression2])
```

**rtrim**

Returns "string\_expression1", with final characters removed after the last character not in "string\_expression2"; for example, rtrim ( 'ABxXxyx' , 'xy' ) returns ABxX. If "string\_expression2" is not specified, the final space characters are removed.

**Syntax**

```
rtrim (string_expression1 [, string_expression2])
```

**split\_part**

Returns "numeric\_expression" field having split "string\_expression1" on "string\_expression2".

**Syntax**

```
split_part (string_expression1 , string_expression2 , numeric_expression)
```

**ascii**

Returns a number representing the ascii code value of the leftmost character of "string\_expression"; for example, ascii ( 'A' ) is 65.

**Syntax**

```
ascii (string_expression)
```

**chr**

Returns the character that has the ASCII code value specified by "integer\_expression". "Integer\_expression" should be between 0 and 255.

**Syntax**

```
chr (integer_expression)
```

**current\_database**

Returns the name of the current database.

## Appendix A: Calculation Components

### Syntax

```
current_database ()
```

## current\_schema

Returns the name of the current schema.

### Syntax

```
current_schema ()
```

## {current\_user}

### Syntax

```
{current_user}
```

## {session\_user}

### Syntax

```
{session_user}
```

## Greenplum Data type formatting

### to\_char

Returns the string representation of "expression" with the format of "string\_expression". "Expression" can either be a date value or a numeric value.

### Syntax

```
to_char (expression , string_expression)
```

### to\_date

Converts "string\_expression1" to a date value as specified by the format "string\_expression2".

### Syntax

```
to_date (string_expression1 , string_expression2)
```

### to\_number

Converts "string\_expression1" to a numeric value as specified by the format "string\_expression2".

### Syntax

```
to_number (string_expression1 , string_expression2)
```

### to\_timestamp

Converts "string\_expression1" to a timestamp value as specified by the format "string\_expression2".

### Syntax

```
to_timestamp (string_expression1 , string_expression2)
```

**translate**

Returns "string\_expression1" with each occurrence of each character in "string\_expression2" replaced by its corresponding character in "string\_expression3".

**Syntax**

```
translate (string_expression1 , string_expression2 , string_expression3)
```

**date\_trunc**

Returns the timestamp to the specified precision.

**Syntax**

```
date_trunc (string_expression , timestamp_expression)
```

**version**

Returns the string value of the database version.

**Syntax**

```
version ()
```

**Greenplum Math****log**

Returns the base 10 logarithm of "numeric\_expression1" or logarithm to the base "numeric\_expression2".

**Syntax**

```
log (numeric_expression1 [, numeric_expression2])
```

**ln**

Returns the natural logarithm of "numeric\_expression1".

**Syntax**

```
ln (numeric_expression)
```

**cbrt**

Returns the cube root of "numeric\_expression1".

**Syntax**

```
cbrt (numeric_expression)
```

**pi**

Returns the constant of pi.

**Syntax**

```
pi ()
```

## Greenplum Trigonometry

### acos

Returns the arccosine of "numeric\_expression" in radians. The arccosine is the angle whose cosine is "numeric\_expression".

#### Syntax

```
acos (numeric_expression)
```

### asin

Returns the arcsine of "numeric\_expression" in radians. The arcsine is the angle whose sine is "numeric\_expression".

#### Syntax

```
asin (numeric_expression)
```

### atan

Returns the arctangent of "numeric\_expression" in radians. The arctangent is the angle whose tangent is "numeric\_expression".

#### Syntax

```
atan (numeric_expression)
```

### atan2

Returns the arctangent of the x and y coordinates specified by "numeric\_expression1" and "numeric\_expression2", respectively, in radians. The arctangent is the angle whose tangent is "numeric\_expression2" / "numeric\_expression1".

#### Syntax

```
atan2 (numeric_expression1 , numeric_expression2)
```

### cos

Returns the cosine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

#### Syntax

```
cos (numeric_expression)
```

### cot

Returns the cotangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

#### Syntax

```
cot (numeric_expression)
```

**degrees**

Returns the degrees where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
degrees (numeric_expression)
```

**radians**

Returns the radians where "numeric\_expression" is an angle expressed in degrees.

**Syntax**

```
radians (numeric_expression)
```

**sin**

Returns the sine of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
sin (numeric_expression)
```

**tan**

Returns the tangent of "numeric\_expression" where "numeric\_expression" is an angle expressed in radians.

**Syntax**

```
tan (numeric_expression)
```

## Report Functions

**\_add\_days**

Returns the datetime resulting from adding "integer\_expression" days to "timestamp\_expression".

**Syntax**

```
_add_days (timestamp_expression , integer_expression)
```

**Example**

```
_add_days (2007-01-14 00:00:00.000 , 3)
```

Result: 2007-01-17 00:00:00.000

**\_add\_months**

Returns the datetime resulting from adding "integer\_expression" months to "timestamp\_expression".

**Syntax**

```
_add_months (timestamp_expression , integer_expression)
```

## \_add\_years

Returns the datetime resulting from adding "integer\_expression" years to "timestamp\_expression".

### Syntax

```
_add_years (timestamp_expression , integer_expression)
```

## \_age

Returns a number by subtracting "timestamp\_expression" from today's date.

### Syntax

```
_age (timestamp_expression)
```

### Example

`_age ([Query1].[Date]), where [Query1].[Date] is March 2, 2004, and today is July 8, 2009`

Result: 50,406, where 5 is the number of years, 04 is the number of months, and 06 is the number of days.

## \_day\_of\_week

Returns the day of the week (between 1 and 7) for "timestamp\_expression" where "integer\_expression" indicates which day of that week is day 1. To determine "integer\_expression", choose the day of the week and count from Monday; for example, if you choose Wednesday, "integer\_expression" would be 3 because Wednesday is the third day from Monday.

### Syntax

```
_day_of_week (timestamp_expression , integer_expression)
```

### Example

`_day_of_week ( 2009-01-01 , 7 ), where 7 means that Sunday is the first day of the week.`

Result: 5

## \_day\_of\_year

Returns the ordinal for the day of the year in "timestamp\_expression" (1 to 366). Also known as Julian day.

### Syntax

```
_day_of_year (timestamp_expression)
```

## \_days\_between

Returns a positive or negative number representing the number of days between "timestamp\_expression1" and "timestamp\_expression2". If "timestamp\_expression1" < "timestamp\_expression2", the result will be a negative number.

**Syntax**

```
_days_between (timestamp_expression1 , timestamp_expression2)
```

**\_days\_to\_end\_of\_month**

Returns a number representing the number of days remaining in the month represented by "timestamp\_expression".

**Syntax**

```
_days_to_end_of_month (timestamp_expression)
```

**\_first\_of\_month**

Returns a datetime that is the first day of the month represented by "timestamp\_expression".

**Syntax**

```
_first_of_month (timestamp_expression)
```

**Example 1**

```
_first_of_month (2009-05-04 00:00:00.000)
```

Result: Returns 2009-05-01 00:00:00.000

**Example 2**

```
_first_of_month (current_date)
```

Result: Returns Jul 1, 2009 if the current date is July 30, 2009.

**\_last\_of\_month**

Returns a datetime that is the last day of the month represented by "timestamp\_expression".

**Syntax**

```
_last_of_month (timestamp_expression)
```

**\_make\_timestamp**

Returns a timestamp constructed from "integer\_expression1" (the year), "integer\_expression2" (the month), and "integer\_expression3" (the day). The time portion defaults to 00:00:00.000 .

**Syntax**

```
_make_timestamp (integer_expression1 , integer_expression2 , integer_expression3)
```

**\_months\_between**

Returns a positive or negative number representing the number of months between "timestamp\_expression1" and "timestamp\_expression2". If "timestamp\_expression1" < "timestamp\_expression2", the result will be a negative number.

## Appendix A: Calculation Components

### Syntax

```
_months_between (timestamp_expression1 , timestamp_expression2)
```

## week\_of\_year

Returns the week number (1-53) of the year for "timestamp\_expression". According to the ISO 8601, week 1 of the year is the first week to contain a Thursday, which is equivalent to the first week containing January 4th. A week starts on a Monday (day 1) and ends on a Sunday (day 7).

### Syntax

```
_week_of_year (timestamp_expression)
```

## years\_between

Returns a positive or negative integer representing the number of years between "timestamp\_expression1" and "timestamp\_expression2". If "timestamp\_expression1" < "timestamp\_expression2", a negative value is returned.

### Syntax

```
_years_between (timestamp_expression1 , timestamp_expression2)
```

## ymdint\_between

Returns a number representing the difference between "timestamp\_expression1" and "timestamp\_expression2". This value has the form YYMMDD, where YY represents the number of years, MM represents the number of months, and DD represents the number of days.

### Syntax

```
_ymdint_between (timestamp_expression1 , timestamp_expression2)
```

### Example

```
_ymdint_between ([Query1].[Date (close date)] , [Query1].[Date (ship date)]),
where [Query1].[Date (close date)] is February 20, 2004, and [Query1].[Date
(ship date)] is January 19, 2004.
```

Result: 101, where 1 is the number of months and 01 is the number of days.

## abs

Returns the absolute value of "numeric\_expression". If "numeric\_expression" is negative, a positive value is returned.

### Syntax

```
abs (numeric_expression)
```

## AsOfDate

Returns the date value of the AsOfDate expression, if it is defined. Otherwise, AsOfDate returns the report execution date.

**Syntax**`AsOfDate ()`**AsOfTime**

Returns the time value of the AsOfTime expression, if it is defined. Otherwise, AsOfTime returns the report execution time.

**Syntax**`AsOfTime ()`**BurstKey**

Returns the burst key.

**Syntax**`BurstKey ()`**BurstRecipients**

Returns the distribution list of burst recipients.

**Syntax**`BurstRecipients ()`**ceiling**

Returns the smallest integer that is greater than or equal to "numeric\_expression".

**Syntax**`ceiling ( numeric_expression )`**CellValue**

Returns the value of the current crosstab cell.

**Syntax**`CellValue ()`**character\_length**

Returns the number of characters in "string\_expression".

**Syntax**`character_length ( string_expression )`**ColumnNumber**

Returns the current column number.

### Syntax

```
ColumnNumber ()
```

## CubeCreatedOn

Returns the date and time when the cube was created. "Dimension" specifies from which cube to retrieve the metadata. If the dimension source is an IBM® Cognos® PowerCube (.mdc), the function returns a blank string because the initial creation date of a PowerCube is not maintained.

### Syntax

```
CubeCreatedOn (dimension)
```

## CubeCurrentPeriod

Returns the current period for the cube. "Dimension" specifies from which cube to retrieve the metadata.

### Syntax

```
CubeCurrentPeriod (dimension)
```

## CubeDataUpdatedOn

Returns the date time that data in the cube was last updated. "Dimension" specifies from which cube to retrieve the metadata.

### Syntax

```
CubeDataUpdatedOn (dimension)
```

## CubeDefaultMeasure

Returns the name of the default measure for the cube. "Dimension" specifies from which cube to retrieve the metadata.

### Syntax

```
CubeDefaultMeasure (dimension)
```

## CubeDescription

Returns the description of the cube. "Dimension" specifies from which cube to retrieve the metadata.

### Syntax

```
CubeDescription (dimension)
```

## CubeIsOptimized

Returns "true" if the cube is optimized. "Dimension" specifies from which cube to retrieve the metadata.

### Syntax

```
CubeIsOptimized (dimension)
```

## CubeName

Returns the name of the cube. "Dimension" specifies from which cube to retrieve the metadata.

### Syntax

```
CubeName (dimension)
```

## CubeSchemaUpdatedOn

Returns the date time that the cube schema was last updated. "Dimension" specifies from which cube to retrieve the metadata.

### Syntax

```
CubeSchemaUpdatedOn (dimension)
```

## exp

Returns the constant 'e' raised to the power of "numeric\_expression". The constant 'e' is the base of the natural logarithm.

### Syntax

```
exp (numeric_expression)
```

### Example

```
exp (2)
```

Result: 7.389056

## extract

Returns an integer representing the value of "date\_part\_expression" in "datetime\_expression". "Date\_part\_expression" could be the year, month, day, hour, minute, or second.

### Syntax

```
extract (date_part_expression , datetime_expression)
```

### Example 1

```
extract (year , 2003-03-03 16:40:15.535)
```

Result: 2003

### Example 2

```
extract (hour , 2003-03-03 16:40:15.535)
```

Result: 16

## floor

Returns the largest integer that is less than or equal to "numeric\_expression".

### Syntax

```
floor (numeric_expression)
```

## GetLocale

Returns the run locale (deprecated).

### Syntax

```
GetLocale ()
```

## HorizontalPageCount

Returns the current horizontal page count.

### Syntax

```
HorizontalPageCount ()
```

## HorizontalPageNumber

Returns the current horizontal page number.

### Syntax

```
HorizontalPageNumber ()
```

## InScope

Returns Boolean 1 (true) when the cell is in the scope of the data items and MUNs; otherwise, returns Boolean 0 (false).

### Syntax

```
InScope (dataItem , MUN, ...)
```

## IsAccessible

Returns Boolean 1 (true) if the report is run with the accessibility features enabled. Use this function as a variable expression with a conditional block to make your reports accessible. For example, you can add a list or crosstab equivalent to a chart in reports that are run with accessibility features enabled.

### Syntax

```
IsAccessible()
```

## IsBursting

Returns Boolean 1 (true) when the report will be distributed to the recipient; otherwise, returns Boolean 0 (false).

### Syntax

```
IsBursting ('recipientName')
```

## IsCrosstabColumnNameMember

Returns Boolean 1 (true) if the current node is a crosstab column node member.

**Syntax**

```
IsCrosstabColumnNodeMember ()
```

**IsCrosstabRowNodeMember**

Returns Boolean 1 (true) if the current node is a crosstab row node member.

**Syntax**

```
IsCrosstabRowNodeMember ()
```

**IsFirstColumn**

Returns Boolean 1 (true) if the current column is the first column.

**Syntax**

```
IsFirstColumn ()
```

**IsInnerMostCrosstabColumnNodeMember**

Returns Boolean 1 (true) if the current node is an innermost crosstab column node member.

**Syntax**

```
IsInnerMostCrosstabColumnNodeMember ()
```

**IsInnerMostCrosstabRowNodeMember**

Returns Boolean 1 (true) if the current node is an innermost crosstab row node member.

**Syntax**

```
IsInnerMostCrosstabRowNodeMember ()
```

**IsLastColumn**

Returns Boolean 1 (true) if the current column is the last column.

**Syntax**

```
IsLastColumn ()
```

**IsLastInnerMostCrosstabColumnNodeMember**

Returns Boolean 1 (true) if the current node is the last innermost crosstab column node member.

**Syntax**

```
IsLastInnerMostCrosstabColumnNodeMember ()
```

**IsLastInnerMostCrosstabRowNodeMember**

Returns Boolean 1 (true) if the current node is the last innermost crosstab row node member.

**Syntax**

```
IsLastInnerMostCrosstabRowNodeMember ()
```

## IsOuterMostCrosstabColumnNodeMember

Returns Boolean 1 (true) if the current node is an outermost crosstab column node member.

### Syntax

```
IsOuterMostCrosstabColumnNodeMember ()
```

## IsOuterMostCrosstabRowNodeMember

Returns Boolean 1 (true) if the current node is an outermost crosstab row node member.

### Syntax

```
IsOuterMostCrosstabRowNodeMember ()
```

## IsPageCountAvailable

Returns Boolean 1 (true) if the page count is available for the current execution of the report; otherwise, returns Boolean 0 (false).

### Syntax

```
IsPageCountAvailable ()
```

## In

Returns the natural logarithm of "numeric\_expression".

### Syntax

```
ln (numeric_expression)
```

## Locale

Returns the run locale.

### Syntax

```
Locale ()
```

## lower

Returns "string\_expression" with all uppercase characters converted to lowercase. This function appears in the Bursted Sales Performance Report sample report in the GO Data Warehouse (query) package.

### Syntax

```
lower (string_expression)
```

## mapNumberToLetter

Adds "integer\_expression" to "string\_expression".

### Syntax

```
mapNumberToLetter (string_expression , integer_expression)
```

**Example**

```
mapNumberToLetter ('a' , 1)
```

Result: b

**mod**

Returns an integer value representing the remainder (modulo) of "integer\_expression1" / "integer\_expression2".

**Syntax**

```
mod (integer_expression1 , integer_expression2)
```

**ModelPath**

Returns the model path.

**Syntax**

```
ModelPath ()
```

**Now**

Returns the current system time.

**Syntax**

```
Now ()
```

**nullif**

Returns null if "string\_expression1" equals "string\_expression2" (case-insensitive), otherwise returns "string\_expression1".

**Syntax**

```
nullif (string_expression1 , string_expression2)
```

**octet\_length**

Returns the number of bytes in "string\_expression".

**Syntax**

```
octet_length (string_expression)
```

**PageCount**

Returns the current page count. This function works only when the report output is Adobe® PDF or Microsoft® Excel. If you save the report output, this function works for all formats.

**Syntax**

```
PageCount ()
```

## PageName

Returns the current page name.

### Syntax

```
PageName ()
```

## PageNumber

Returns the current page number.

### Syntax

```
PageNumber ()
```

## ParamCount

Returns the parameter count of "parameterName".

### Syntax

```
ParamCount ('parameterName')
```

## ParamDisplayValue

Returns a string that is the parameter display value of "parameterName". This function appears in the Recruitment Report sample report in the GO Data Warehouse (analysis) package.

### Syntax

```
ParamDisplayValue ('parameterName')
```

## ParamName

Returns the parameter name of "parameterName".

### Syntax

```
ParamName ('parameterName')
```

## ParamNames

Returns all parameter names.

### Syntax

```
ParamNames ()
```

## ParamValue

Returns the parameter value of "parameterName".

### Syntax

```
ParamValue ('parameterName')
```

## position

Returns the integer value representing the starting position of "string\_expression1" in "string\_expression2". Returns 0 if "string\_expression1" is not found.

### Syntax

```
position (string_expression1 , string_expression2)
```

## power

Returns "numeric\_expression1" raised to the power of "numeric\_expression2".

### Syntax

```
power (numeric_expression1 , numeric_expression2)
```

### Example

```
power (3 , 2)
```

Result: 9

## ReportAuthorLocale

Returns the author locale.

### Syntax

```
ReportAuthorLocale ()
```

## ReportCreateDate

Returns the date when the report was created.

### Syntax

```
ReportCreateDate ()
```

## ReportDate

Returns the report execution date and time.

### Syntax

```
ReportDate ()
```

## ReportDescription

Returns the report description. This function works only when the report is run from IBM® Cognos® Connection.

### Syntax

```
ReportDescription ()
```

## ReportID

Returns the report ID.

### Syntax

```
ReportID ()
```

## ReportLocale

Returns the run locale.

### Syntax

```
ReportLocale ()
```

## ReportName

Returns the report name. This function works only when the report is run from IBM® Cognos® Connection.

### Syntax

```
ReportName ()
```

## ReportOption

Returns the value of the run option variable identified by "optionName", such as attachmentEncoding, burst, cssURL, email, emailAsAttachment, emailAsURL, emailBody, emailSubject, emailTo, emailToAddress, history, metadataModel, outputEncapsulation, outputFormat, outputLocale, outputPageDefinition, outputPageOrientation, primaryWaitThreshold, print, printer, printerAddress, prompt, promptFormat, saveAs, saveOutput, secondaryWaitThreshold, verticalElements, or xsIURL.

### Syntax

```
ReportOption ('optionName')
```

## ReportOutput

Returns the name of the output format, such as CSV, HTML, layoutDataXML, MHT, PDF, rawXML, singleXLS, spreadsheetML, XLS, XML, or XLWA.

### Syntax

```
ReportOutput ()
```

## ReportPath

Returns the report path. This function works only when the report is run from IBM® Cognos® Connection.

### Syntax

```
ReportPath ()
```

## ReportProductLocale

Returns the product locale.

### Syntax

```
ReportProductLocale ()
```

## ReportSaveDate

Returns the date when the report was last saved.

### Syntax

```
ReportSaveDate ()
```

## round

Returns "numeric\_expression" rounded to the nearest value with "integer\_expression" significant digits to the right of the decimal point. If "integer\_expression" is negative, "numeric\_expression" is rounded to the nearest absolute value with "integer\_expression" significant digits to the left of the decimal point. Rounding takes place before data formatting is applied.

### Syntax

```
round (numeric_expression , integer_expression)
```

### Example

```
round (125, -1)
```

Result: 130

## RowNumber

Returns the current row.

### Syntax

```
RowNumber ()
```

## ServerLocale

Returns the locale of the server that runs the report.

### Syntax

```
ServerLocale ()
```

## ServerName

Returns the name of the web server where the run request originated from. The value may be empty if the request is executed from the scheduler.

### Syntax

```
ServerName ()
```

## Appendix A: Calculation Components

### sqrt

Returns the square root of "numeric\_expression". "Numeric\_expression" must not be a negative value.

#### Syntax

```
sqrt (numeric_expression)
```

### substring

Returns the substring of "string\_expression" that starts at position "integer\_expression1" for "integer\_expression2" characters or to the end of "string\_expression" if "integer\_expression2" is -1. The first character in "string\_expression" is at position 1.

#### Syntax

```
substring (string_expression , integer_expression1 , integer_expression2)
```

#### Example

```
substring ([Sales (analysis)].[Sales staff].[Sales staff].[Sales staff].[Position code], 3 , 5)
```

Result: Returns characters 3 to 7 of the position codes.

### TOCHeadingCount

Returns the table of contents heading count for a specified heading level.

#### Syntax

```
TOCHeadingCount (headingLevel)
```

### Today

Returns the current system date.

#### Syntax

```
Today ()
```

### trim

Returns "string\_expression" trimmed of any leading and trailing blanks or trimmed of the character specified by "match\_character\_expression". "Trim\_what\_expression" may be "leading", "trailing", or "both" (default). "Match\_character\_expression" can be an empty string to trim blanks or can specify a character to be trimmed.

#### Syntax

```
trim (trim_what_expression , match_character_expression , string_expression)
```

### upper

Returns "string\_expression" with all lowercase characters converted to uppercase.

**Syntax**

```
upper (string_expression)
```

**URLEncode**

Returns the URL encoded value of the input text.

**Syntax**

```
URLEncode ('text')
```

**Data Type Casting Functions****date2string**

Returns a date as a string in YYYY-MM-DD format.

**Syntax**

```
date2string (date_expression)
```

**date2timestamp**

Converts "date\_expression" to a timestamp. The time part of the timestamp will equal zero.

**Syntax**

```
date2timestamp (date_expression)
```

**date2timestampTZ**

Converts "date\_expression" to a timestamp with a time zone. The time and time zone parts of the timestamp will equal zero.

**Syntax**

```
date2timestampTZ (date_expression)
```

**DTinterval2string**

Returns a date time interval as a string in DDDD HH:MM:SS.FFFFFFF or -DDDD HH:MM:SS.FFF format.

**Syntax**

```
DTinterval2string (date_time_interval_expression)
```

**DTinterval2stringAsTime**

Returns a date time interval as a string in HHHH:MM:SS.FFFFFFF or HH:MM:SS.FFF format.

Days are converted to hours.

**Syntax**

```
DTinterval2stringAsTime (date_time_interval_expression)
```

## int2DTinterval

Converts an integer to a date time interval. "String\_expression" specifies what "integer\_expression" represents: "ns" = nanoseconds, "s" = seconds (default), "m" = minutes, "h" = hours, "d" = days.

### Syntax

```
int2DTinterval (integer_expression , string_expression)
```

#### Example 1

```
int2DTinterval (1020, "h")
```

Result: 42 days 12 hours

#### Example 2

```
int2DTinterval (1020, "s")
```

Result: 17 minutes

## int2YMininterval

Converts "integer\_expression" to a year month interval. "String\_expression" specifies what "integer\_expression" represents: "y" = years, "m" = months (default).

### Syntax

```
int2YMininterval (integer_expression , string_expression)
```

## number2string

Converts "numeric\_expression" to a string, using the %g format specifier (C/C++ syntax).

### Syntax

```
number2string (numeric_expression)
```

## string2date

Returns "string\_expression" as a date in YYYY-MM-DD format.

### Syntax

```
string2date (string_expression)
```

## string2double

Returns a floating point number. "String\_expression" has the following form: "[whitespace] [sign] [digits] [digits] [ {d|D|e|E }[sign]digits]"

### Syntax

```
string2double (string_expression)
```

## string2DTinterval

Returns "string\_expression" as a date time interval in [-]DD HH:MM[:SS[.FFF]] format.

### Syntax

```
string2DTinterval (string_expression)
```

**string2int32**

Returns an integer. "String\_expression" has the following form: "[whitespace] [{+|-}] [digits]"

**Syntax**

```
string2int32 (string_expression)
```

**string2int64**

Returns a long integer. "String\_expression" has the following form: "[whitespace] [{+|-}] [digits]"

**Syntax**

```
string2int64 (string_expression)
```

**string2time**

Returns "string\_expression" as a time in HH:MM:SS.FFFFFFFF format.

**Syntax**

```
string2time (string_expression)
```

**string2timestamp**

Returns "string\_expression" as a timestamp in YYYY-MM-DD [T|t|[white space]+]  
HH:MM:SS.FFFFFFFF format.

**Syntax**

```
string2timestamp (string_expression)
```

**string2timestampTZ**

Returns "string\_expression" in YYYY-MM-DD HH:MM:SS.FFFFFFFF +HHMM or YYYY-MM-  
DD [T|t] HH:MM:SS.FFF -HHMM format.

**Syntax**

```
string2timestampTZ (string_expression)
```

**string2YMininterval**

Returns "string\_expression" as a Year Month Interval in [-]YY MM format.

**Syntax**

```
string2YMininterval (string_expression)
```

**time2string**

Returns a time as a string in HH:MM:SS.FFF format.

**Syntax**

```
time2string (time_expression)
```

**timestamp2date**

Converts "timestamp\_expression" to a date. The time part of the timestamp will be ignored.

## Appendix A: Calculation Components

### Syntax

```
timestamp2date (timestamp_expression)
```

## timestamp2string

Returns a timestamp as a string in YYYY-MM-DD HH:MM:SS.FFFFFFFF format.

### Syntax

```
timestamp2string (timestamp_expression)
```

## timestamp2timestampTZ

Converts "timestamp\_expression" to a timestamp with a time zone. The displacement part of the timestamp with the time zone will be zero.

### Syntax

```
timestamp2timestampTZ (timestamp_expression)
```

## timestampTZ2date

Converts "timestamp\_time\_zone\_expression" to a date. The time and time zone parts of the timestamp will be ignored.

### Syntax

```
timestampTZ2date (timestamp_time_zone_expression)
```

## timestampTZ2string

Returns a timestamp with the time zone as a string in YYYY-MM-DD HH:MM:SS.FFFFFFFF +HHMM or YYYY-MM-DD HH:MM:SS.FFF -HHMM format.

### Syntax

```
timestampTZ2string (timestamp_time_zone_expression)
```

## timestampTZ2timestamp

Converts "timestamp\_time\_zone\_expression" to a timestamp. The displacement part of the timestamp with the time zone will be ignored.

### Syntax

```
timestampTZ2timestamp (timestamp_time_zone_expression)
```

## timeTZ2string

Returns a time with the time zone as a string in HH:MM:SS.FFF +HHMM or HH:MM:SS.FFFFFFFF -HHMM format. For example, -05:30 means a timezone of GMT minus 5 hours and 30 minutes

### Syntax

```
timeTZ2string (timeTZ_expression)
```

## YMinterval2string

Returns "year\_month\_interval\_expression" as a string in (YY MM) or -(YY MM) format.

## Syntax

```
YMinterval2string (year_month_interval_expression)
```

## Appendix A: Calculation Components

---

# Appendix B: Troubleshooting

---

This chapter describes some common problems you may encounter. For more troubleshooting problems, see the Troubleshooting section of the IBM Cognos Business Intelligence *Administration and Security Guide*.

## Problems Creating Reports

The topics in this section document problems you may encounter when creating reports.

### Division by Zero Operation Appears Differently in Lists and Crosstabs

If you have a list that accesses a relational data source, a calculation containing a division by zero operation appears as a null value, such as an empty cell. In a crosstab, the division by zero operation appears as /0. This happens when the **Avoid Division by Zero** property is set to **Yes**, which is the default.

To have a consistent display of null values in lists and crosstabs, define an if-then-else statement in the expression in the crosstab cell that changes the value /0 to the value null.

### Application Error Appears When Upgrading a Report

When upgrading a report, the following error appears if the report contains data items in the page layout that are not in a data container:

*RSV-SRV-0040 An application error has occurred. Please contact your Administrator.*

This error occurs when IBM® Cognos® BI cannot determine the query reference for a data item.

Such data items are identified by a small red circle with a white x icon that appears in the lower left corner.

To correct the error, drag the data items into a container. If the container is a list, we recommend that you drag the data items into the list page header or footer, or the overall header or footer. If you want to see the first row of the item on each page or in the overall report, drag the item to the list page header or overall header. If you want to see the item's last row on each page or in the overall report, drag the item to the list page footer or overall footer.

**Tip:** If a header or footer does not exist, create it.

### Nested List Report Containing a Data Item That is Grouped More Than Once Does Not Run After Upgrade

When you upgrade a nested list report that contains a data item that is grouped in both lists, the report does not run and an error such as the following appears:

*OP-ERR-0199: The query is not supported. The dimensions on the edge are inconsistent. The dataItems from dimension= "[Product line]" must be adjacent.*

This error occurs when the report is run against a dimensional data source and both lists are using the same query. This error does not occur if the report is run against a relational data source.

For example, you have a list that contains the grouped items Product line and Product type and a nested list that contains the data items Year, Quarter, and Unit sale price. Year, Quarter, and Product line are grouped items in the nested list.

To resolve the issue, delete the data item that is grouped in both lists from the inner list.

### Steps to Delete a Grouped Data Item From an Inner List

1. Click anywhere in the report.
2. In the Properties pane, click the select ancestor button and click the List link that represents the inner list.
3. Double-click the **Grouping & Sorting** property.
4. In the Groups pane, select the data item that you want and click the delete button.

## Background Color in Template Does not Appear

When creating a Query Studio template in Report Studio, if you add a list object and change its background color, the color change does not appear when you apply the template to a Query Studio report.

To work around this issue, do one of the following:

- Edit the style sheet (CSS) classes for lists in Report Studio.
- Do not add any objects to the page body when you are creating a Query Studio template.
- Leave the page body blank.

## Subtotals in Grouped Lists

When using an IBM Cognos PowerCube that contains a ragged hierarchy, if you group on the first level in the hierarchy, subtotals may appear in the wrong place or show wrong values.

To resolve the issue, group on the second level.

## Chart Labels Overwrite One Another

In Report Studio and Query Studio, if you define a chart and render it in HTML or PDF format using the default sizes, the axis labels of the chart may overwrite each other.

To avoid this problem, make the chart wider or taller by modifying the height and width properties of the chart or enable the **Allow Skip** property.

## Chart Shows Only Every Second Label

You create a report that includes a chart. The **Allow Skip** option is set to false, but when you run the report, labels are skipped.

This can occur if there is not enough room for all labels and the options **Allow 45 Degree Rotation**, **Allow 90 Degree Rotation**, and **Allow Stagger** are also set to false. IBM Cognos BI has no options for making the labels fit, so it skips every second label.

The solution is to select either **Allow 45 Degree Rotation**, **Allow 90 Degree Rotation**, or **Allow Stagger**.

## Chart Gradient Backgrounds Appear Gray in Internet Explorer

In Report Studio, you can define a custom palette for a chart that includes a gradient. When the chart is rendered in HTML format in Microsoft® Internet Explorer, the chart background appears gray. This is an Internet Explorer issue.

To avoid this problem, select the chart and define the color white as the chart background.

For more information, see the Microsoft Knowledge Base article # 294714 at <http://support.microsoft.com>.

## Metadata Change in Oracle Essbase Not Reflected in Reports and in the Studios

When there is a metadata change on the Oracle Essbase server, the change is not immediately reflected in the metadata tree in the studios. In addition, when a report is run, the report does not pick up the republished changes.

To view the new structure, you must restart the IBM® Cognos® Content Manager server.

## Relationships Not Maintained in a Report With Overlapping Set Levels

In a report, the relationship between nested or parallel member sets at overlapping levels in the same dimension may not always be maintained.

For example, a named set in the data source that contains members from both a Year and Month member is nested under Year, but is not properly grouped by year.

In another example, an error message such as this appears:

*OP-ERR-0201 Values cannot be computed correctly in the presence of multiple hierarchies ([Product].[B1], [Product].[Product]) that each have a level based on the same attribute (Product).*

This problem occurs in the following scenarios involving non-measure data items X and Y, which overlap in the same dimension:

- X and Y together as ungrouped report details
- Y nested under X
- Y appended as an attribute of a group based on X

When using named sets, or sets that cover more than one level of a hierarchy, do not use sets from the same dimension in more than one place in the same report. They should appear on only one level of one edge.

## Summaries in Query Calculations Include Nulls with SAP BW Data Sources

When using an SAP BW data source in IBM® Cognos® Report Studio, null values in the database are returned in the result set and the `count` summary function includes the empty cells in the following scenarios:

- A query calculation includes an arithmetic calculation where one or more NULL operands and an aggregation is performed on the calculation.
- The result of a query calculation is a constant, such as `current_time` and `current_date`.

The `count` summary function should normally exclude null values.

To avoid this problem, for the first scenario, ensure that both operands do not return null values. For example, the original expression is `[num1]+[num2]`. Instead, use the following expression:

```
if ([num1] is null) then (0) else ([num1])
```

```
if ([num2] is null) then (0) else ([num2])
```

There is no workaround for the second scenario.

## Creating Sections on Reports That Access SAP BW Data Sources

SAP BW data sources may have problems with sections in reports under different circumstances:

If a section in a report uses the lowest-level query item in a ragged hierarchy, such as the children of the not assigned node, the following BAPI error may appear:

*BAPI error occurred in function module BAPI\_MDDATASET\_SELECT\_DATA. Value <value-Name> for characteristic <cubeName> unknown*

For more information about working with ragged or unbalanced hierarchies, see the Report Studio *User Guide*.

### Lowest-level Query Item in a Ragged Hierarchy

The solution is to remove the section from the lowest-level query item.

### Several Multicubes with SAP Variables

The solution is to use one SAP multicube when creating sections in reports.

## Error Characters (--) Appear in Reports

When you run a report, you see two dash (--) characters in your report instead of values.

These characters may appear if you use an OLAP data sources other than PowerCube and Microsoft® SQL Server 2005 Analysis Services (SSAS), and you apply aggregation to calculations and measures that use rollups other than Sum (Total), Maximum, Minimum, First, Last, and Count.

All other types of rollup either fail or return error cells, which typically display as two dash characters (--).

This problem occurs in, but is not limited to, the following:

- footers

- aggregate function
- summary filters and detail filters that use a summary
- detail, summary, and context filters that select more than one member of a hierarchy that is used elsewhere on the report

If you are working with a SSAS 2005 data source, these characters may also appear in summary cells if you use an OR filter in the summary. To avoid this problem, do not use OR filters in summaries.

## Columns, Rows, or Data Disappear With SSAS 2005 Cubes

Microsoft® SQL Server 2005 Analysis Services (SSAS) has a feature called AutoExists that removes tuples that have no facts at the intersection of two hierarchies of the same dimension.

Columns, rows, or data can disappear if you set the default member of a hierarchy to a member that does not exist with every other member in the dimension. To avoid this problem, change the default member that caused the disappearance to a member that exists with all other members in the dimension.

Columns, rows, or data can also disappear if members are specified that result in one or more non-existent tuples. There is currently no workaround for this scenario. For more information, see Microsoft Knowledge Base article #944527 at <http://support.microsoft.com>.

You may also encounter unexpected results if the default member of a hierarchy is a member that doesn't also exist in all other hierarchies in the dimension, and if you query members from different hierarchies in the same dimension.

For example a crosstab includes the following (using the Adventure Works cube):

- Rows: Generate([Adventure\_Works].[Account].[Accounts],set([Balance Sheet],[Units])) nested with  
children([Adventure\_Works].[Department].[Departments]->:[YK].[Department].[Departments]).&[1]))
- Column: [Adventure\_Works].[Account].[Account Number].[Account Number]
- Measure: [Adventure\_Works].[Measures].[Amount]

You run the report and notice that the query renders with some blank cells. You then apply the simple detail filter [Amount]>1 and run the report. Only row labels are displayed and all data and columns are missing.

In the Adventure Works cube, the [Account].[Accounts] attribute has a default member set to [Net Income]. When evaluating the GENERATE set expression, SSAS looks in the entire cube space and looks at all coordinates for the [Account] dimension. These coordinates include both [Account][Account Type].&[] and [Account].[Accounts].[Net Income]. Because these two coordinates don't exist within the same hierarchy, SSAS returns an empty set.

To avoid this problem the SSAS administrator must set the default member in the cube to a member that exists in all other hierarchies.

## Function Unreliable with Sets

If you create an expression that uses the `descendants` function with sets, you may encounter unpredictable results. Some expected members may be missing or may have blank captions or labels.

This problem occurs if the `descendants` function uses a set as its first parameter instead of a single member and if the `descendants` function is nested under another data item from the same hierarchy.

To avoid this problem, replace the first parameter in the `descendants` function with the function `currentmember(H)`, where `H` is the hierarchy of the desired set and under which the expression is nested. For example, use `descendants(currentmember(H))`.

## Report Differences Between TM1 Executive Viewer and IBM Cognos BI with TM1 Data Sources

When using an IBM® Cognos® TM1® data source, comparable reports created in an IBM Cognos BI studio and in TM1 Executive Viewer may contain different cell values. This occurs because the TM1 Executive Viewer product uses an algorithm for selecting default members for non-projected dimensions that differs slightly from traditional OLAP clients.

To avoid this problem, when filtering your reports in the IBM Cognos BI studios, use context filters that match the default selections shown in the Executive Viewer user interface. This ensures that the cell values in IBM Cognos BI match the values in Executive Viewer.

## Order of Metadata Tree Differs for TM1 Data Sources

When using a an IBM® Cognos® TM1® data source, the order of members in the metadata tree of the **Insertable Objects** pane of an IBM Cognos BI studio may differ from the order shown in TM1 Architect.

By default, TM1 Architect renders members of hierarchies using a slightly different algorithm than does IBM Cognos BI. IBM Cognos BI automatically renders member metadata from TM1 data sources in hierarchical order.

From within TM1 Architect, if you want to see how an IBM Cognos BI studio will render a hierarchy, click the Hierarchy Sort button.

## Problems Calculating Data

The topics in this section document problems you may encounter when calculating or summarizing data.

### Unexpected Summary Values in Nested Sets

If a report contains nested sets, summaries other than the inner set summaries may contain unexpected values. For example, you insert a summary in a crosstab that contains a set with years in the rows.

| <b>Revenue</b> |                  |
|----------------|------------------|
| 2004           | 914,352,803.72   |
| 2005           | 1,159,195,590.16 |
| Total          | 2,073,548,393.88 |

You then nest a product line set within years.

| <b>Revenue</b> |                                  |
|----------------|----------------------------------|
| 2004           | Camping Equipment 332,986,338.06 |
|                | Golf Equipment 153,553,850.98    |
| 2005           | Camping Equipment 402,757,573.17 |
|                | Golf Equipment 168,006,427.07    |
| Total          | 2,073,548,393.88                 |

Notice that the summary value does not change to represent the total of the new values. This occurs because the within set aggregation used with dimensional packages does not take into account sets that are nested below the set that is summarized.

To show the correct summary values, if the inner and outer sets do not belong to the same dimension, you can nest a copy of the inner summary item under the outer summary item, as follows.

|       |                   | <b>Revenue</b>   |
|-------|-------------------|------------------|
| 2004  | Camping Equipment | 332,986,338.06   |
|       | Golf Equipment    | 153,553,850.98   |
|       | Total             | 486,540,189.04   |
| 2005  | Camping Equipment | 402,757,573.17   |
|       | Golf Equipment    | 168,006,427.07   |
|       | Total             | 570,764,000.24   |
| Total | Total             | 1,057,304,189.28 |

## Null Results for Calculations Using SAP BW Data Sources

When using a SAP BW data source, the expression you use in your calculation is evaluated as a null value if your expression contains a null item. For example, in the calculation `some_expression = result`, the result is null if a row or column that the expression references includes a null value.

To avoid obtaining null values as the result of your calculations, suppress null values before you create the calculation.

## Incorrect Results with IBM Cognos PowerCubes and Time Measures

If a report uses an IBM® Cognos® PowerCube data source and the following combination of data items, you will encounter incorrect results:

- a measure with a Time State Rollup set to Average or Weighted Average
- an aggregate (*members from time dimension*) expression
- an intersection with a member in a relative time hierarchy

To avoid incorrect results, do not use this combination in your reports.

## Incorrect Results in Summaries When Using OLAP Data Sources

When using an OLAP data source, summaries that use `for` clauses give incorrect results.

This occurs because `for` clauses require access to the detail rows of the fact table. OLAP data sources do not have detail rows.

For example, this report uses a dimensionally-modeled relational (DMR) data source and contains the following summaries:

- `mx: maximum ([Revenue] for [Year (ship date)])`
- `mx2: maximum (Aggregate([Revenue]) for [Year (ship date)])`

| Year                     | Quarter | Revenue                 | mx         | mx2            |
|--------------------------|---------|-------------------------|------------|----------------|
| 2004                     | Q1 2004 | 221,704,705.31          | 252,408.9  | 235,750,316.25 |
|                          | Q2 2004 | 222,143,384.57          | 252,408.9  | 235,750,316.25 |
|                          | Q3 2004 | 235,750,316.25          | 252,408.9  | 235,750,316.25 |
|                          | Q4 2004 | 234,754,397.59          | 252,408.9  | 235,750,316.25 |
| <b>2004 - Summary</b>    |         | <b>914,352,803.72</b>   |            |                |
| 2005                     | Q1 2005 | 293,228,460.53          | 292,402.7  | 306,706,702.72 |
|                          | Q2 2005 | 278,180,759.96          | 292,402.7  | 306,706,702.72 |
|                          | Q3 2005 | 281,079,666.95          | 292,402.7  | 306,706,702.72 |
|                          | Q4 2005 | 306,706,702.72          | 292,402.7  | 306,706,702.72 |
| <b>2005 - Summary</b>    |         | <b>1,159,195,590.16</b> |            |                |
| 2006                     | Q1 2006 | 344,124,267.07          | 363,575.08 | 391,874,462.51 |
|                          | Q2 2006 | 391,874,462.51          | 363,575.08 | 391,874,462.51 |
|                          | Q3 2006 | 378,118,012.54          | 363,575.08 | 391,874,462.51 |
|                          | Q4 2006 | 381,774,358.78          | 363,575.08 | 391,874,462.51 |
| <b>2006 - Summary</b>    |         | <b>1,495,891,100.9</b>  |            |                |
| 2007                     | Q1 2007 | 471,624,367.69          | 349,132.3  | 479,269,923.82 |
|                          | Q2 2007 | 479,269,923.82          | 349,132.3  | 479,269,923.82 |
|                          | Q3 2007 | 166,441,982.56          | 349,132.3  | 479,269,923.82 |
| <b>2007 - Summary</b>    |         | <b>1,117,336,274.07</b> |            |                |
| <b>Overall - Summary</b> |         | <b>4,686,775,768.85</b> |            |                |

Notice that the `mx` and `mx 2` values are different, where `mx2` is based on visible data, but `mx` is not. This result is correct.

The following report uses an OLAP data source and contains the same summaries.

| Year                     | Quarter  | Revenue                 | mx             | mx2            |
|--------------------------|----------|-------------------------|----------------|----------------|
| 2004                     | 2004 Q 1 | 221,704,705.31          | 235,750,316.25 | 235,750,316.25 |
|                          | 2004 Q 2 | 222,143,384.57          | 235,750,316.25 | 235,750,316.25 |
|                          | 2004 Q 3 | 235,750,316.25          | 235,750,316.25 | 235,750,316.25 |
|                          | 2004 Q 4 | 234,754,397.59          | 235,750,316.25 | 235,750,316.25 |
| <b>2004 - Summary</b>    |          | <b>914,352,803.72</b>   |                |                |
| 2005                     | 2005 Q 1 | 293,228,460.53          | 306,706,702.72 | 306,706,702.72 |
|                          | 2005 Q 2 | 278,180,759.96          | 306,706,702.72 | 306,706,702.72 |
|                          | 2005 Q 3 | 281,079,666.95          | 306,706,702.72 | 306,706,702.72 |
|                          | 2005 Q 4 | 306,706,702.72          | 306,706,702.72 | 306,706,702.72 |
| <b>2005 - Summary</b>    |          | <b>1,159,195,590.16</b> |                |                |
| 2006                     | 2006 Q 1 | 344,124,267.07          | 391,874,462.51 | 391,874,462.51 |
|                          | 2006 Q 2 | 391,874,462.51          | 391,874,462.51 | 391,874,462.51 |
|                          | 2006 Q 3 | 378,118,012.54          | 391,874,462.51 | 391,874,462.51 |
|                          | 2006 Q 4 | 381,774,358.78          | 391,874,462.51 | 391,874,462.51 |
| <b>2006 - Summary</b>    |          | <b>1,495,891,100.90</b> |                |                |
| 2007                     | 2007 Q 1 | 471,624,367.69          | 479,269,923.82 | 479,269,923.82 |
|                          | 2007 Q 2 | 479,269,923.82          | 479,269,923.82 | 479,269,923.82 |
|                          | 2007 Q 3 | 166,441,982.56          | 479,269,923.82 | 479,269,923.82 |
| <b>2007 - Summary</b>    |          | <b>1,117,336,274.07</b> |                |                |
| <b>Overall - Summary</b> |          | <b>4,686,775,768.85</b> |                |                |

Notice that mx and mx2 values are now the same. Both summaries are based on visible data. The mx value is incorrect.

Incorrect results also appear for footer summaries.

To avoid this problem, when using OLAP data sources, ensure that the parameter that precedes the `for` clause is an aggregate function.

## Problems Filtering Data

The topics in this section document problems you may encounter when filtering data.

### Unexplained Discrepancies in Number Calculations

You might find unexplained discrepancies in number calculations due to round-off errors. For example:

- You run regression tests and find differences in numbers. They are different only because of the rounding off of decimal places.
- You choose not display zeros in reports, but the zeros are displayed anyway because there are decimal places (0.0000000000000426, for example) that are rounded off to zero in reports.

Round-off problems are not specific to IBM® Cognos® software. They can occur in any environment where rounding off occurs.

## Binary Round-Off Errors

Discrepancies in calculations might occur due to binary round-off errors. For example, if the number 1.1 is represented as a binary floating point number and your report format includes a large number of decimal places, the number 1.1 might actually be something like 1.09999999999997.

If your report is formatted to use only one decimal point, decimal round-off takes place, compensating for the binary round-off. So the number appears to be 1.1 when it is really 1.09999999999997. When the number is used in calculations, you might get round-off errors. For example, Microsoft® Excel calculations use binary numbers (without rounding off decimal places) but formatting in reports shows rounded off decimal places, which can create small discrepancies.

## Division Round-Off Errors

Calculations that involve division typically incur round-off errors, regardless of how the numbers are represented. Examples of such calculations are Average and Percent of Base.

## Design Guidelines to Minimize Round-Off Effect

The best solution is to change the underlying database schema or cube model but that may not always be possible. Another solution is to minimize the round-off effect by following these guidelines when authoring reports and creating models in FrameWork Manager and external OLAP cubes:

- Avoid storing data in floating point format whenever possible. This is especially true for currency values, which should be stored as either fixed-point decimals or as integers with a scale value such as 2.

For example, in a cube, the Revenue for Camping Equipment in 2004 is \$20,471,328.88. If revenue details are stored as floating point numbers, round-off errors might occur when revenue is calculated.

The round up errors might have slight differences, depending on the order of calculation. If revenue for Products is calculated first and revenue for Time is calculated second, you might get a different round-off error than if Time is calculated first and Products is calculated second.

Total revenue might be calculated as the number above. Or there might be slight discrepancies, for example, \$20,471,328.8800001 as opposed to \$20,471,328.88. The internal number might be slightly different than what is displayed. The number might even be for different runs of the same report, depending on the order that the OLAP engine uses for calculation.

- In reports, avoid division whenever possible. When division is unavoidable, try to do it as late as possible in the calculation process. For example, instead of Total([Revenue]/1000), use Total ([Revenue])/1000.
- When doing comparisons, add a margin to allow for round-off. For example, you may want [Profit %] to be a fractional value formatted as a percentage with no decimals. However, the filter [Profit %]<>0 (or [Profit %] NOT BETWEEN 0 and 0) rejects zero values and may still return values that appear to be 0% after formatting.

To avoid this, filter in one of these two ways:

- [Profit %] NOT BETWEEN -0.005 and 0.005
- ([Profit %] < -0.005) OR ([Profit %] > 0.005)

Note that 0.005 is equivalent to 0.5%, which displays as either 0% or 1%, depending on floating point precision losses.

In some cases, you may prefer control round-off errors by rounding values explicitly. For example, instead of [Profit %], use round([Profit %],2).

- Recalculate numbers every time instead of reusing calculations that might contain rounded off decimals.

There might be additional considerations for Microsoft® Analysis Services 2005/2008, especially when comparing report results from different runs (as happens in Lifecycle Manager). Refer to Microsoft documentation for more information.

## **HRESULT= DB\_E\_CANTCONVERTVALUE Error When Filtering on a \_make\_timestamp Column**

You cannot filter on a \_make\_timestamp column, and the following error messages appear:

*UDA-SQL-0114 The cursor supplied to the operation "sqlOpenResult" is inactive*

*UDA-SQL-0206 The OLEDB driver returned the following value: HRESULT= DB\_E\_CANTCON-  
VERTVALUE*

*RSV-SRV-0025 Unable to execute this request*

The solution is to apply the filter after aggregation and not before.

## **Problems Running Reports**

The topics in this section document problems you may encounter when viewing or running reports.

### **Report Runs Slowly**

The following is a list of questions that will help you to troubleshoot a slow report.

- Does your IBM Cognos environment conform with the supported environments?  
Supported environments can be found at [www.ibm.com](http://www.ibm.com).
- Has the report always been slow or did it recently become slow?

If it recently became slow, can you identify an event that occurred just before the report began to run slowly? Events could include changes to configuration settings, changes to tuning settings, a recent upgrade where your previous settings have not been applied, an introduction of firewalls or proxies, changes to existing firewalls or proxies, changes to virus scans on temp directories, or temporary table space restrictions on the database. This event could have caused the change in report performance.

- Is the performance slow for all reports or just one report?

If all reports are slow, the issue may be due to your environment or database. If all reports from a specific package are slow, the issue may be due to the model design. If just one report is slow, the issue may be due to a specific report element.

- How many queries does your report contain?

The number of queries on the report will proportionally affect the report execution time.

- Does the report run slowly for everyone, or just for one user?

If the report runs slowly for just one user, the issue may be due to something in that user's environment, such as virus scanning, page file size or location settings, or their location on the network.

- Is the report burst or run often by many people?

If many people are running the same report at the same time, you may need to scale your environment or consider using dispatcher routing rules to direct all requests for a specific package or group of users to a specific server or server group. For more information, see the *IBM Cognos Business Intelligence Administration and Security Guide*.

- Do your queries require local processing?

The following report elements require local processing: crosstabs and charts, master relationships, unions or joins, multiple fact queries, bursting, and non-vendor specific functions. Local processing requires the IBM Cognos server to compute operations on the result set returned by the database, which can impact the SQL execution time.

- Does your environment use a Custom Authentication Provider?

Using a Custom Authentication Provider could cause a memory leak if the code is not destroying objects correctly.

- Have you reviewed the logs in the *c10\_location/logs* directory and the audit logs?

They may help you identify the source of the problem. Monitoring your processes, such as the Java™ and Business Intelligence bus processes could also identify excessive memory use.

- Is your environment tuned correctly?

For more information, see the Performance Tuning Settings for IBM Cognos 8 Business Intelligence and the IBM Cognos 8 Business Intelligence Performance Tuning Cheat Sheet documents available online at [www.ibm.com](http://www.ibm.com).

- Have you recently upgraded?

Ensure that any tuning settings that were applied to your previous installation are applied to the new environment. Ensure that your models have been verified, upgraded, and republished. Verify that the IBM Cognos Framework Manager governor that allows enhanced model portability at runtime is not enabled. Depending on your upgrade method, you may also need to open and save the reports again after upgrading.

The following tips from the Report Studio *User Guide* may help you improve report performance.

- Change the order in which items are queried from the database. For more information, see "[Working with Relational Queries](#)" (p. 329) or "[Working with Dimensional Queries](#)" (p. 377).
- Create sections without creating master detail relationships. For more information, see "[Divide Data into Sections](#)" (p. 313).
- Share queries between lists and repeaters. For more information, see "[Shared Queries Between Lists and Repeaters](#)" (p. 64).

- Convert queries to SQL. For more information, see "[Convert a Query to SQL](#)" (p. 341) for relational reporting or "[Convert a Query to SQL](#)" (p. 391) for dimensional reporting.
- Avoid using functions with limited support. For more information, see "[Quality of Service Indicators](#)" (p. 344) for relational reporting or "[Quality of Service Indicators](#)" (p. 402) for dimensional reporting.
- Be aware of the limitations of creating expressions with SAP BW data sources. For more information, see "[Using Microsoft Excel Functions with SSAS 2005 Data Sources](#)" (p. 404).
- Use Select & Search prompts instead of value prompts if your list of prompts is long. For more information, see "[Build Your Own Prompt and Prompt Page](#)" (p. 423).
- Provide your own prompt values in a prompt to avoid accessing the database. For more information, see "[Specify Prompt Values](#)" (p. 433).
- Suppress null cells using filters. For more information, see "[Suppress Null Cells Using Filters](#)" (p. 498).
- Set page breaks without creating master detail relationships. For more information, see "[Create a Page Break or Page Set](#)" (p. 536).
- Do not modify IBM Cognos Analysis Studio set definitions. For more information, see "[Analysis Studio Query Specification](#)" (p. 553).
- Do not put filters on non-identifiers.
- Avoid combining large hierarchy levels and sets in a drill-through report in a way that creates large queries. For more information, see "[Recommendation - Drilling Down in Very Large Data Sources](#)" (p. 417).
- Use database functions when possible. For more information, see "[Calculation Components](#)" (p. 769).
- Use fixed width objects in PDF reports by setting their **Size & Overflow** properties. For more information, see "[Set List Properties](#)" (p. 111), "[Set Crosstab Properties](#)" (p. 118), "[Customizing Chart Properties](#)" (p. 147), or "[Set Map Properties](#)" (p. 205).

The following are Proven Practices documents on [www.ibm.com](http://www.ibm.com) that may help you improve your report performance.

- Performance Tuning Settings for IBM Cognos 8 Business Intelligence
- IBM Cognos 8 Business Intelligence Performance Tuning Cheat Sheet
- Writing Efficient OLAP Queries
- Cognos 8 Business Intelligence (Business Intelligence) on IBM AIX best practices
- IBM Cognos ReportNet® and Java Heap

The *Installation and Configuration Guide* also includes a section on performance maintenance.

## Summaries in Report Do not Correspond to the Visible Members

If a crosstab or chart created in IBM® Cognos® Report Studio using a dimensional data source has a context-dependent set function such as `filter` or `topCount` on an edge, summaries do not correspond to the visible members. This occurs when the summaries use the `within set` aggregation mode.

This problem occurs because a summary that uses the `within set` aggregation mode uses a set that is dependent on the members that it intersects with on the opposite edge. For example, the following crosstab has the top three products returned as columns. The expression used to generate the columns is

`topCount ([Product],3,[Return quantity])`

where `[Product]` is the level.

| Return quantity        | BugShield L 89110 | BugShield 90110 | Sun 30 94110  | Total(RetProducts) | Min(RetProducts) |
|------------------------|-------------------|-----------------|---------------|--------------------|------------------|
| Americas               | 25,219            | 19,870          | 13,814        | 62,392             | 17,303           |
| Asia Pacific           | 22,822            | 19,171          | 6,389         | 54,758             | 12,765           |
| Northern Europe        | 8,325             | 14,634          | 4,065         | 32,936             | 8,325            |
| Central Europe         | 17,627            | 13,854          | 14,089        | 45,570             | 13,854           |
| Southern Europe        | 7,196             | 4,726           | 5,401         | 20,220             | 5,790            |
| <b>Total(Region)</b>   | <b>81,189</b>     | <b>72,255</b>   | <b>43,758</b> | <b>215,876</b>     | <b>58,037</b>    |
| <b>Minimum(Region)</b> | <b>7,196</b>      | <b>4,726</b>    | <b>4,065</b>  | <b>20,220</b>      | <b>5,790</b>     |

The summary values for `Total(ReturnedProducts)` and `Minimum(ReturnedProducts)` for all rows except `Central Europe` do not correspond to the member values in the crosstab. This means that the top three products returned in all regions except for Central Europe are not Bug Shield Lotion 89110, Bug Shield Extreme 90110, and Sun Shelter 30 94110. Note that the summary values for `Total(Region)` and `Minimum(Region)` do correspond to the visible member values. That is because those summary values represent the total and minimum quantities returned for those three products in each region.

You can see what the top three products returned in each region are by dragging the columns to the right of the rows, creating the following single-edge crosstab.

|                 |                                  |               |
|-----------------|----------------------------------|---------------|
| Americas        | <i>BugShield Lotion 89110</i>    | 25,219        |
|                 | <i>BugShield Extreme 90110</i>   | 19,870        |
|                 | <i>TrailChef Water Bag 1110</i>  | 17,303        |
|                 | <b>Total(ReturnedProducts)</b>   | <b>62,392</b> |
|                 | <b>Minimum(ReturnedProducts)</b> | <b>17,303</b> |
| Asia Pacific    | <i>BugShield Lotion 89110</i>    | 22,822        |
|                 | <i>BugShield Extreme 90110</i>   | 19,171        |
|                 | <i>Single Edge 73110</i>         | 12,765        |
|                 | <b>Total(ReturnedProducts)</b>   | <b>54,758</b> |
|                 | <b>Minimum(ReturnedProducts)</b> | <b>12,765</b> |
| Northern Europe | <i>BugShield Extreme 90110</i>   | 14,634        |
|                 | <i>Star Peg 16110</i>            | 9,977         |
|                 | <i>BugShield Lotion 89110</i>    | 8,325         |
|                 | <b>Total(ReturnedProducts)</b>   | <b>32,936</b> |
|                 | <b>Minimum(ReturnedProducts)</b> | <b>8,325</b>  |
| Central Europe  | <i>BugShield Lotion 89110</i>    | 17,627        |
|                 | <i>Sun Shelter 30 94110</i>      | 14,089        |
|                 | <i>BugShield Extreme 90110</i>   | 13,854        |
|                 | <b>Total(ReturnedProducts)</b>   | <b>45,570</b> |
|                 | <b>Minimum(ReturnedProducts)</b> | <b>13,854</b> |
| Southern Europe | <i>Single Edge 73110</i>         | 7,234         |
|                 | <i>BugShield Lotion 89110</i>    | 7,196         |
|                 | <i>TrailChef Cup 4110</i>        | 5,790         |
|                 | <b>Total(ReturnedProducts)</b>   | <b>20,220</b> |
|                 | <b>Minimum(ReturnedProducts)</b> | <b>5,790</b>  |

To obtain summary values that reflect the visible members, modify the expression of the data item containing the context-dependent set function so that it includes a tuple that is locked to the default member of every hierarchy that appears on the opposite edge. For this example, modify the expression to the following:

```
topCount ([Product],3,tuple([Return quantity], defaultMember([Retailer site])))
```

where [Product] is the level and [Retailer site] is the hierarchy.

When you run the report, all summary values reflect the visible members in the crosstab.

| Return quantity        | <i>BugShield Lotion 89110</i> | <i>BugShield Extreme 90110</i> | <i>Sun Shelter 30 94110</i> | <b>Total(ReturnedProducts)</b> | <b>Minimum(ReturnedProducts)</b> |
|------------------------|-------------------------------|--------------------------------|-----------------------------|--------------------------------|----------------------------------|
| Americas               | 25,219                        | 19,870                         | 13,814                      | 58,903                         | 13,814                           |
| Asia Pacific           | 22,822                        | 19,171                         | 6,389                       | 48,382                         | 6,389                            |
| Northern Europe        | 8,325                         | 14,634                         | 4,065                       | 27,024                         | 4,065                            |
| Central Europe         | 17,627                        | 13,854                         | 14,089                      | 45,570                         | 13,854                           |
| Southern Europe        | 7,196                         | 4,726                          | 5,401                       | 17,323                         | 4,726                            |
| <b>Total(Region)</b>   | <b>81,189</b>                 | <b>72,255</b>                  | <b>43,758</b>               | <b>197,202</b>                 | <b>42,848</b>                    |
| <b>Minimum(Region)</b> | <b>7,196</b>                  | <b>4,726</b>                   | <b>4,065</b>                | <b>17,323</b>                  | <b>4,065</b>                     |

## Cannot Find the Database in the Content Store (Error QE-DEF-0288)

You cannot retrieve data from the selected database when running a report from IBM Cognos Query Studio, IBM Cognos Connection, or Report Studio.

The following error message appears:

*QE-DEF-0288 Unable to find the database...*

If this error does not occur when you are logged on as an administrator, then to solve the problem, ensure that the user has permissions to the signon embedded. If this error always occurs, the data source has not been created. Create the data source with the name mentioned in the error message.

## Parse Errors When Opening or Running an Upgraded Report

Earlier versions of ReportNet® and IBM Cognos Business Intelligence included the `cast_Date` function for reports that run on an Oracle database. This function does not exist for Oracle in IBM Cognos 8.1.2 MR1 and later versions. If a report that uses an Oracle database includes the `cast_Date` function, parse errors will be received when you try to open or run the report.

## Overflow Error Occurs When a Value in a Crosstab Is More Than 19 Characters

In a crosstab report, values support a maximum of 19 characters, including the decimal point. If a value exceeds 19 digits, an overflow error occurs. By default, the decimal precision is set to 7 digits, which restricts the number of integers to 11 digits.

To use more than 11 integers, you must edit the `qfs_config.xml` file in the `c10_location\configuration` directory. For more information, see the section about reducing decimal precision in the IBM Cognos Business Intelligence *Administration and Security Guide*.

## The ORA-00907 Error Appears When Running a Report

When using an Oracle 9.2 data source, under certain circumstances, multiple or nested join operations may fail and produce the following error:

*ORA-00907: missing right parenthesis*

A query that uses both a left outer join and an ON clause condition returns zero values instead of null values.

## A Report or Analysis Does Not Run Because of Missing Items

You attempt to run a report or analysis and a message indicates that one or more items are missing or changed. Each missing item is listed by its MUN (member unique name). The MUN includes the complete path within the hierarchy for the item. When you place your cursor on an item in the **Insertable Objects** pane, the MUN for that item is displayed in a tooltip. This situation may occur if members have been removed from or changed in the data source. It may also occur when you attempt to run a report that uses items to which you do not have access. For example, an administrator may create an analysis that includes items that you do not have the correct permission to access.

The solution is to find a suitable replacement in the **Insertable Objects** pane, and drag it to the **New Item** column. The report or analysis will then run.

## Cannot View Burst Report

When you burst a report, each burst output is sent to the associated list of recipients. If a list of recipients contains invalid entries, the following occurs:

- The burst output is not saved to IBM Cognos Content Manager.

Consequently, you cannot view the burst output in IBM Cognos Connection.

- If you choose to send the output by email, only valid recipients will receive an email. Although the output is sent as an attachment if you select the **Attach the report** check box, no link is generated if you select the **Include a link to the report** check box.
- The following error message appears in the run history for the report, where parameter 1 is the burst key, parameter 2 is the list of recipients, and parameter 3 contains the error messages returned by Content Manager:

*An error occurred while saving the output for the burst instance <param type="string" index="1" /> with the recipients (<param type="string" index="2" />). Here are the details: <param type="string" index="3" />*

**Note:** The list of recipients includes both the valid and invalid recipients.

For example, a report is set up to burst on Country, and the recipients are managers. Running the report produces the following countries and recipients:

- Canada: John, Mary
- US: Peter, Frank
- France: Danielle, Maryse

Frank is an invalid recipient. The burst outputs for Canada and France are saved to Content Manager, but not the U.S. output. If you choose to send an email to each recipient and you selected the **Include a link to the report** check box, the email to Peter will not contain a link to the output for US. The error message that is generated will contain Peter and Frank as values for parameter 2 with no indication as to which is invalid.

### Steps to Correct or Remove Burst Recipients

1. View the error message in the run history for the report.

2. From the list of recipients, determine which recipients are invalid.

You may need to consult with your administrator to find out which recipients are invalid.

3. Correct or remove the invalid recipients.

Correcting or removing invalid recipients will depend on how the list of recipients was defined, such as through a calculated field or a burst table.

4. Run the report again.

## A report upgraded from reportNet does not retain its original look

When you upgrade a report to IBM® Cognos® Business Intelligence, a new style sheet is applied that changes the look of the report.

To preserve the formatting that was used in the original report, you can select a different style sheet. This retains the original look of the report and specifies that any new items added to the report, such as list columns or crosstab levels, have the original formatting applied to them.

### Steps

1. In IBM Cognos Report Studio, from the File menu, click **Report Properties**.
2. Click **Report styles** and select **1.x styles**.

## Measure Format Disappears in SSAS 2005

Microsoft® SQL Server 2005 Analysis Services (SSAS) does not propagate formatting through calculations. IBM® Cognos® compensates for this whenever possible, but cannot guarantee to do so in all cases. As a result, if you are working with a Microsoft SSAS cube, any calculation (other than a non-count summary) that is based on or intersects with a formatted measure, such as a currency, may lose the measure format. This may also happen if you use a detail filter or context filter (slicer).

For example, a crosstab includes members on one edge and a measure with formatting, such as a currency symbol and decimal places, applied on the other edge. When you run the report, you see the formatting for each cell. However, if you add a detail filter, such as **measure > 1** and run the report, all the formatting disappears.

Additionally, because the SSAS behavior depends on the fine details of the MDX generated by Cognos Business Intelligence, whether the format is lost in a report can change from release to release.

To avoid this problem, specify explicit formatting for the affected row, column, or cell.

## Drill-through Links are Not Active in the Safari Browser

When viewing a PDF report in the Macintosh Safari browser, you cannot open hyperlinks. This is because the Macintosh Safari browser does not have the necessary Adobe® Acrobat plug-in.

To avoid this problem, use the HTML format when creating drill-through reports that may be viewed in Safari.

## Data Does Not Appear in a Target Report or the Wrong Data Appears

If no data appears when you drill through to a target report or if the wrong data appears, the problem might be data source conformance. The business keys might be different or might be mismatched.

For example, the business key for Camping Equipment might be 100 in the data source for the source report and 1 in the data source for the target report, in which case no data appears in the target report. Another example might be that the business key for Camping Equipment is 100 in the data source for the source report but, in the data source for the target report, 100 is the business key for Golf Equipment, in which case the wrong data appears in the target report.

To solve the problem, ensure that business keys have the same value in both data sources. If there are cases where data does not appear to match, contact your database administrator or data modeler.

For more information about data source conformance, search for "conformed dimensions" and "business keys" in the Transformer *User Guide* and the Report Studio *User Guide*.

You might also want to see "[Unexpected or Empty Results When Drilling Through](#)" (p. 983).

## A Running Total in Grouped Reports Gives Unexpected Results

You have a running total calculation in a grouped report that returns unexpected values.

Because tabulation of the running total calculation depends on the order in which the grouping is executed, you must ensure that the grouped totals are tabulated before applying the running total.

To ensure that the grouping is executed in correct order, define a running total calculation as a freestanding calculation outside the query subject in IBM Cognos Framework Manager, and ensure that the Regular Aggregate property is set to Automatic.

This may also be an issue with other running, moving, and ranking aggregations.

## PCA-ERR-0057 Recursive Evaluation Error

You run a report and encounter the following error:

*PCA-ERR-0057 Recursive evaluation has exceeded limit. Calculated member trace: COG\_OQP\_USR\_Aggregate(Retailer Type): COG\_OQP\_INT\_m2: COG\_OQP\_INT\_m1: COG\_OQP\_USR\_Aggregate(Retailer Type): COG\_OQP\_INT\_m2: COG\_OQP\_INT\_m1: COG\_OQP\_USR\_Aggregate(Retailer Type): COG\_OQP\_INT\_m2: COG\_OQP\_INT\_m1: COG\_OQP\_USR\_Aggregate(Retailer Type): COG\_OQP\_INT\_m2: COG\_OQP\_INT\_m1*

You may encounter this error when two or more data items form a recursive evaluation. For example, in the above error, the calculation of Aggregate(Retailer Type) is dependent on a column expression while at the same time the column expression is dependent on Aggregate(Retailer Type). Therefore, the cyclic relationship cannot be resolved.

To avoid this problem, ensure that calculations do not have cyclic relationships.

## Arithmetic Overflow Error When Running a Report in PDF Format

If you use a Microsoft SQL Server 2005 data source and your report includes aggregations, you may encounter the following error when you run your report in PDF format:

*RQP-DEF-0177 An error occurred while performing operation 'sqlOpenResult' status=-28'. UDA-SQL-0114 The cursor supplied to the operation "sqlOpenResult" is inactive. UDA-SQL-0564 [Microsoft OLE DB Provider for SQL Server] Arithmetic overflow error converting expression to data type int. (SQLSTATE=22003, SQLERRORCODE=8115)*

This error occurs because the action is performed in the database, and the database data type is too small.

This error did not occur in IBM Cognos Business Intelligence version 8.3 or earlier because aggregation was processed locally, by the Business Intelligence server. In version 8.4 or later, aggregation is processed at the database level.

To avoid this problem, increase the size of the the database data type.

## RQP-DEF-0177 An error occurred while performing operation 'sqlPrepareWithOptions' status='-69' UDA-SQL-0043 Error

You cannot run a report in Report Studio or Query Studio, and the following error messages appear:

*RQP-DEF-0177 An error occurred while performing operation 'sqlPrepareWithOptions' status='-69' UDA-SQL-0043 The underlying database detected an error during processing the SQL request.[NCR][ODBC Teradata Driver][Teradata Database] Partial string matching requires character operands*

These error messages do not indicate an IBM Cognos Application Firewall problem.

There is a problem with your data source not converting numeric data items. Ask your administrator to consult the topic *Enable Conversion of Numeric Search Keys to Strings in Queries* in the IBM Cognos Business Intelligence *Administration and Security Guide*.

## Problems When Drilling Through

The topics in this section document problems you may encounter when drilling through reports.

### Cannot Drill Through Between PowerCubes Because MUNs Do Not Match

We recommend that business keys be unique throughout the dimension for PowerCubes. These keys are used as the source value for levels in a hierarchy of a dimension. If the values are not unique throughout the dimension, the corresponding Category Code values may be generated with tildes.

For example, if a category for the Product Line level has a source value of 101 and a category in the Product Type level has a source value of 101, the Category Code value for the Product Type level is automatically generated with a unique value such as 101~245. The Category Code values are used in the Member Unique Name (MUN) for each member, for example, [Sales and Marketing].[Products].[Products].[Product type]->:[PC].[@MEMBER].[101~245].

Because these values are generated automatically, they cannot be guaranteed from one cube build to the next or in a build for another cube with the same dimension structure using the same source values. Therefore, drilling from one PowerCube to another on what appears to be the same member might not work since the MUNs might not match.

If the MUNs do not match, consult the cube modellers to see if the business keys can be made unique throughout the dimension. If this is not likely, or might take some time to resolve, you can use calculations to pass the source value from one PowerCube to another for drill-through.

For more information about drill-through access, see the Drill-Through Access chapter in the IBM Cognos Business Intelligence *Administration and Security Guide*.

#### Steps to pass the business key (source value) from the source report to the target report

1. In the target report, create a filter with the following syntax:

`filter([Hierarchy or Level], roleValue('_businessKey', [Hierarchy or Level]) = ?Parameter?)`

For example:

`filter([Sales Cube].[Products].[Products].[Product type], roleValue('_businessKey',[Sales Cube].[Products].[Products].[Product type]) = ?Prod Type?)`

2. In the source report, create a Query Calculation which is used to pass the business key (source value) to the target report by mapping it to the target parameter in the drill-through definition. Use the following syntax:

`roleValue('_businessKey', [Hierarchy or Level])`

For example:

`roleValue('_businessKey', [sales_and_marketing].[Products].[Products].[Product type])`

## Unexpected or Empty Results When Drilling Through

When you drill from a source report to a target report, there might be no data returned. This might be the correct result if there is no data that corresponds to the drill-through selections or if you do not have permission to view the data.

In other cases, if no data or the wrong data appears, the source item might not be mapped to the target correctly or the values in the data sources might not be conformed (the values do not match in both data sources).

If you have the necessary permissions, you can debug drill-through definitions by using the drill-through assistant from the **Go To** page (right-click the selection in the source report and select **Go To**). You can view the passed source values and the mapping to the target report parameters. You can use this tool for both authored and package drill-through definitions.

You might be able to correct the problem by modifying the parameter mapping in the drill-through definition. For example, when you drill from a cube to a relational data source, sometimes no data is returned or the wrong data is returned because the business key values in the two data sources do not match. You can change the drill-through definition to pass the caption of the IBM Cognos PowerCube member instead of the business key, but you must also change the target report to filter on the corresponding string value and not the business key value.

However, it is best to ensure the data sources are conformed. In this example, the business keys in the cube should match the business keys in the relational source. Filtering on a key is more efficient than filtering on a larger string that may or may not be indexed in the database. For more information on data source conformance, search for "conformed dimensions" and "business keys" in the IBM Cognos Transformer *User Guide* and the Report Studio *User Guide*.

### Steps to Pass the Caption from a PowerCube to a Relational Source

1. Ensure that the target report filters on a string value that matches the caption being passed from the PowerCube.
2. Edit the drill-through definition as follows:
  - If the drill-through definition was created in Report Studio, open the report, and go to the drill-through definition associated with the drill-through source object. On the parameter mapping page, select **Member Caption** in the **Property to pass** column.
  - If the drill-through definition was created in the source package, go to IBM Cognos Connection, **Drill-Through Definitions**, and open the package drill-through definition. On the **Target** tab of the drill-through definition, select **Member Caption** in the **Property to pass** column for the appropriate parameter.

When you drill through, instead of the business key, the caption is passed to the target.

You might also want to see "[Data Does Not Appear in a Target Report or the Wrong Data Appears](#)" (p. 980). For more information about drill-through access, see the Drill-Through Access chapter in the IBM Cognos Business Intelligence *Administration and Security Guide*.

## Cannot Drill Through From a Relational Source to a Cube

By default, you cannot drill through from a relational data source to a cube. This is because a cube expects a Member Unique Name (MUN) as a parameter value and relational sources do not use MUNs.

Members have properties which include a business key and a caption. If either of these match data items within the relational source, drilling through can be performed as long as the cube target report is authored in Report Studio.

If the source data source has a query item, for example display name, that corresponds to a member property in the target cube, for example caption, you must create the parameter on the caption in the target report.

To pass the data item to the cube target, do the following:

- In the cube target report, create a parameter that accepts the caption of the member. This parameter should be created in a Query Calculation object from the Toolbox tab with the following syntax. Type the following

*filter([Hierarchy or Level], caption([Hierarchy or Level]) = ?Parameter?)*

For example:

*filter([sales\_and\_marketing].[Products].[Products].[Product line], caption([sales\_and\_marketing].[Products].[Products].[Product line]) = ?Product Line?)*

For more information about drill-through access, see the Drill-Through Access chapter in the IBM Cognos Business Intelligence *Administration and Security Guide*.

# Appendix C: Accessibility Features

Accessibility features help users who have a physical disability, such as restricted mobility or limited vision, to use information technology products.

## Keyboard Shortcuts

This product uses the following standard Microsoft® Windows® navigation keys.

| Description                    | Shortcut Keys |
|--------------------------------|---------------|
| Opens the online help.         | F1            |
| Closes the Web browser window. | Alt+F4        |
| Copies objects.                | Ctrl+C        |
| Pastes objects.                | Ctrl+V        |

## Considerations to Improve Report Accessibility

Creating accessible reports ensures access of information to all users, with all levels of ability.

For example, people with a visual impairment may use screen reading technology to access the information in a report.

The following are some design considerations for creating accessible reports:

- Avoid using visual cues, such as bold text or color, to convey important information.
- Avoid using pictures and OLE Objects in PDF documents, as these items are tagged as artifacts and ignored by the screen reader.
- Avoid using conditional formatting to convey important information.
- When selecting color palettes for report objects, choose patterns or shades of gray.
- Ensure that there is a table corresponding to chart types that are rendered as images because the screen reader ignores this information.
- Deliver reports in HTML format, which is the most supported output format for most screen readers.
- Ensure that the report has a title.
- Gain an understanding for screen reading technology.

- Avoid spelling and grammatical errors, as they cause the screen reading software to misinterpret the information.
- Avoid using features like calendar boxes and up and down selections on time controls. Instead use prompts such as check boxes, radio buttons, combo boxes, and multi-select boxes.
- Ensure that the target application is accessible when using embedded Web applications or drill-through paths.
- Avoid using large, complex list or crosstab reports.

Displaying the information in multiple simple lists or crosstab reports is more manageable for assistive technology users.
- Add alternate text to images, charts, and other visual objects so that screen readers can provide context for them.
- When using tables, add summary text to provide context for the table content. If the top cells in a table behave as headers, designate these cells as headers so that screen readers can identify the relationships.

## Enable Accessible Report Outputs

If you want to include accessibility features, such as alternate text, summary text, designated cell headers in tables and accessible conditional layouts, you must enable these accessibility features in the report output.

You can enable accessible report outputs in one of the following ways:

- in the IBM® Cognos® Report Studio run options, so that the report has accessibility features enabled when you run the report from within Report Studio.
- in IBM Cognos Connection as a run option (**Run with options**, **Enable accessibility support**), so that report consumers can run the report once with accessibility features.
- in IBM Cognos Connection as a property (**Set properties**, **Report tab**), so that report consumers can always run the report with accessibility features.
- in IBM Cognos Connection, as a user preference (**My area**, **My preferences**), so that report consumers can enable accessibility features for all of their reports. Accessibility settings in the report properties overwrite this setting.
- in IBM Cognos Administration, as a server-wide option, so that all reports for all IBM Cognos users have accessibility features enabled. Accessibility settings in the user preferences and report properties overwrite this setting.

Administrators can also change a system-level setting that forces the accessibility features on or off regardless of any other settings.

For information about the last four options, see the IBM Cognos Connection *User Guide* or the IBM Cognos *Administration and Security Guide*.

### Step for Report Studio

- In Report Studio, from the **Run** menu, click **Run Options** and select the **Include accessibility features** check box.

## Add Alternate Text to Images and Charts

You can add alternate text for images, maps, and charts to make your reports accessible. When a screen reader encounters one of these objects, it reads the alternate text that you added to the object.

You can add translations for the text to support users in multiple languages. When users run a report, IBM® Cognos® BI uses the alternate text in the appropriate language.

If you use images only for visual spacing in your report, leave the **Alternate Text** property empty.

For the report output to contain the alternate text, you must enable the accessibility features for the report. For more information, see "["Enable Accessible Report Outputs"](#) (p. 986).

### Steps

1. Select the image or chart object.
2. In the **Properties** pane, click the select ancestor button  and select the **Image**, **Map**, or chart object.
3. Double-click the **Alternate Text** property.
4. Select **Specified text** and click the ellipsis (...) button.
5. In the **Default text** box, type a description for the object, and click **Add**.
6. In the **Languages** dialog box, select the languages that apply to your text.
7. Double-click a language and type the translation of the text for each language and click **OK**.

## Add Summary Text to Tables

You can provide summary text for crosstabs, lists, repeater tables, and table objects. This text provides context for the entire object to make your reports accessible. When a screen reader encounters one of these objects in HTML report outputs, it reads the description that you added to the object.

The table summary is not displayed in visual Web browsers. Only screen readers and speech browsers use the summary text. The summary text is usually read immediately before the table caption.

You can add translations for the text to support users in multiple languages. When users run a report, IBM® Cognos® BI uses the summary text in the appropriate language.

If you use tables for report layout, leave the summary empty to indicate to screen readers that the table is used exclusively for visual layout and not for presenting tabular data.

For the report output to contain the summary text, you must enable the accessibility features for the report. For more information, see "[Enable Accessible Report Outputs](#)" (p. 986).

### Steps

1. Select the crosstab, list, or table.
2. In the Properties pane, click the select ancestor button  and select the Crosstab, List, Repeater Table, or Table object.
3. Double-click the **Summary Text** property.
4. Select **Specified text** and click the ellipsis (...) button.
5. In the **Default text** box, type a description for the object, and click **Add**.
6. In the **Languages** dialog box, select the languages that apply to your text.
7. Double-click a language and type the translation of the text for each language.

## Designate Cells Headers in Tables

You can specify whether specific table cells are table headers. This allows screen readers and speech browsers to identify the relationships between the cells in your tables.

For the report output to contain the cell headers, you must enable the accessibility features for the report. For more information, see "[Enable Accessible Report Outputs](#)" (p. 986).

### Steps

1. Select the table cells.
2. In the Properties pane, set the **Table Header** property to **Yes**.

## Example - Conditionally Show a List Below a Chart for an Accessible Report

Charts are rendered as images in report outputs, such as HTML and PDF. As a result, they are difficult to navigate for visually impaired users and screen readers cannot convey the information shown in charts. To make your reports accessible, you can add a conditional layout that shows list or crosstab equivalents of the chart when the accessibility features are enabled for the report output.

To set up this conditional layout, use a conditional block that contains the list or crosstab and the report function **IsAccessible**. The **IsAccessible** function is a Boolean function that returns **Yes** (or true) when the accessibility features are enabled for the report output and **No** (or false) when the accessibility features are not enabled.

Therefore, the list or crosstab becomes conditional and appears only when the accessible report output option is enabled. In addition, you should add alternate text for the chart object so that screen readers can let users know that an accessible list or crosstab follows.

### Steps

1. Open IBM® Cognos® Report Studio with the sample GO Sales (query) package.

2. Create a new report with a pie chart.
3. Add data items to the chart. In the **Insertable Objects** pane, on the **Source** tab , do the following:
  - Expand **Sales (query)** and **Sales** and then insert **Quantity** in the **Default measure** area.
  - Expand **Products** and insert **Product line** in the **Series (Pie Slices)** area.
4. Run the report to see your chart.
5. Add alternate text for the pie chart:
  - Select the chart and, in the **Properties** pane, double-click the **Alternate Text** property.
  - Select **Specified text** and click the ellipsis (...) button.
  - Type default text for the chart, such as  
**A pie chart shows the quantity of sales by product line as a percentage of total sales. An equivalent list is below.**

If you want to add the alternate text in additional languages, click the add button .
6. From the **Toolbox** tab , insert a **Conditional Blocks** object below the chart.
7. Select the conditional block and, from the **Properties** pane, double-click the **Block Variable** property.
8. Under **Variable**, select **New boolean variable** and type a name for the variable, such as **accessibility\_condition**.
9. In the **Report Expression** dialog box, specify the expression that defines the variable:
  - From the **Functions** tab , expand **Report Functions** and insert the **IsAccessible** function to the **Expression Definition** box.
  - Type a closing bracket at the end of the function **IsAccessible ()** and click **OK**.
10. Select the conditional block and in the **Properties** pane, change the **Current Block** property to **Yes**.
11. From the **Toolbox** tab , add a list inside the conditional block.
12. Add the same data items from the chart to the list. In the **Insertable Objects** pane, on the **Source** tab , do the following:
  - Expand **Sales (query)** and **Products** and insert **Product line** in the list.
  - Expand **Sales** and insert **Quantity** as the second column in the list.
13. Add summary text for the list
  - Select the list and in the **Properties** pane, double-click the **Summary Text** property.
  - Select **Specified text** and click the ellipsis (...) button.

## Appendix C: Accessibility Features

- Type default text for the list, such as

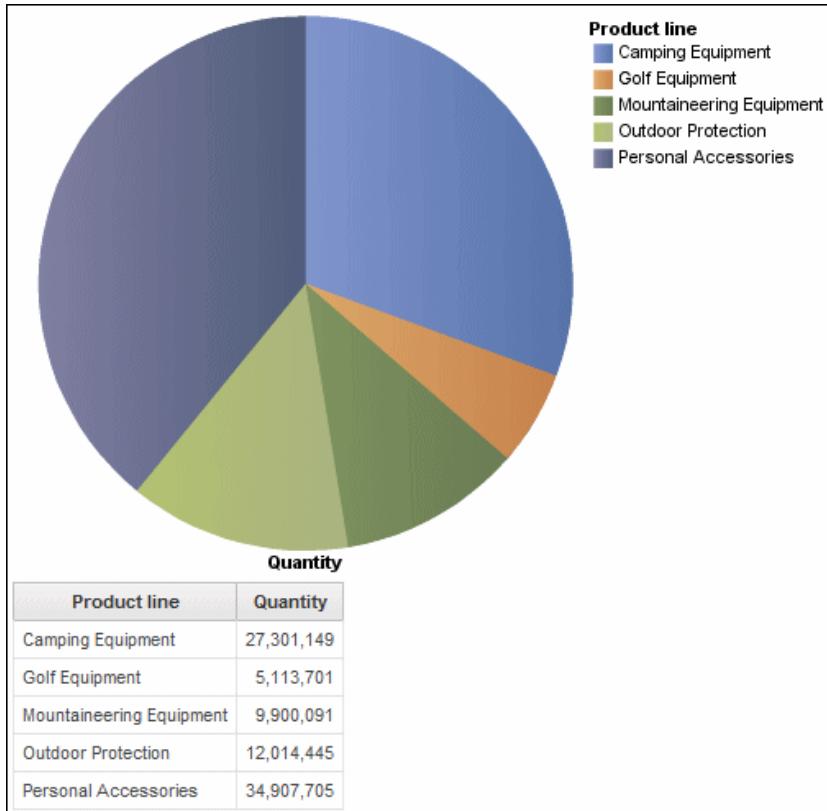
The first column lists all product lines and the second column lists sales quantities.

If you want to add the alternate text in additional languages, click the add button .

14. Select the conditional block and in the Properties pane, change the Current Block property to No.
15. From the Run menu, click Run Options and select the Include accessibility features check box.
16. Run the report.

Because we just chose to include the accessibility features, you see both the chart and the list. Also, when a screen reader encounters the image for the chart, it reads the alternate text that you added.

If you clear the Include accessibility features check box and run the report again, you see only the chart.



## IBM and accessibility

See the IBM® Accessibility Center (<http://www.ibm.com/able>) for more information about the commitment that IBM has to accessibility.

---

# Glossary

---

## **attribute**

In dimensional models, a property that provides qualitative information about members of a level in a dimension. For example, the Store level within the Retailer dimension might have properties such as address or retail space. In general, dimensional attributes do not have measure values or rollups associated with them, but are used to locate or filter members.

In relational models, a query item that is not a measure or identifier. When a query item is an attribute, it is not intended to be aggregated, or used for grouping or generating prompt pick lists.

In BI modeling, a characteristic of an entity which is descriptive rather than a unique identifier or an aggregative measure.

In TM1, a property that provides qualitative information about dimensions.

## **burst**

To create several report results by running a single report once. For example, the user can create a report that shows sales for each employee, and run it once, sending different results to regional managers by bursting on region.

## **calculated member**

A member of a dimension whose measure values are not stored but are calculated at run time using an expression.

## **cardinality**

For OLAP data sources, the number of members in a hierarchy. The cardinality property for a hierarchy is used to assign solve orders to expressions.

For relational data sources, a numerical indication of the relationship between two query subjects, query items, or other model objects.

## **cascading prompt**

A prompt that uses values from a previous prompt to filter the values in the current prompt or pick list.

## **condition**

An expression that yields a Boolean value. Conditions are used in query expressions, query filters, and Boolean report variables that can be used for conditional formatting, styles, data sources, layouts, and blocks.

## **Content Manager**

The service that retrieves information from the content store, and saves information to the content store.

### **cube**

A multidimensional representation of data needed for online analytical processing, multidimensional reporting, or multidimensional planning applications.

### **data source**

The source of data itself, such as a database or XML file, and the connection information necessary for accessing the data.

In TM1®, the file or data used as the source for the TurboIntegrator import process.

### **data tree**

Within a studio, a structure that contains objects such as query subjects, query items, dimensions, levels, and members. A data tree is used as a palette of the available data that can be inserted into calculations, filters, display areas, and other authoring gestures.

### **dimension**

In Cognos Planning, a list of related items such as Profit and Loss items, months, products, customers, and cost centers, including calculations. The rows, columns, and pages of a cube are created from dimensions.

In Cognos BI, TM1, and Express, a broad grouping of descriptive data about a major aspect of a business, such as products, dates, or locations. Each dimension includes different levels of members in one or more hierarchies and an optional set of calculated members or special categories.

### **dimensional data source**

A data source containing data modeled using OLAP concepts, including dimensions, hierarchies, and measures.

### **drill down**

In a multidimensional representation of data, to access information by starting with a general category and moving downwards through the hierarchy of information. For example from Years to Quarters to Months.

In TM1, to access information by starting with a general category and moving through the hierarchy of information. For example, in a database, to move from field to file to record.

### **drill up**

To navigate from one level of data to a less detailed level. The levels are set by the structure of the data.

### **group**

A collection of users who can share access authorities for protected resources.

### **hierarchy**

The organization of a set of entities into a tree structure, with each entity (except the root) having one or more parent entities and an arbitrary number of child entities.

In Data Manager, a particular view of a business dimension. A hierarchy contains the definition of related reference data that is organized into a tree structure of members related as parents and children.

### **layout**

The arrangement of printed matter on a screen or page, including margins, line spacing, type specification, header and footer information, indents, and more.

### **level**

A set of entities or members that form one section of a hierarchy in a dimension and represent the same type of object. For example, a geographical dimension might contain levels for country, region, and city.

### **locale**

A setting that identifies language or geography and determines formatting conventions such as collation, case conversion, character classification, the language of messages, date and time representation, and numeric representation.

### **measure**

A performance indicator that is quantifiable and used to determine how well a business is operating. For example, measures can be Revenue, Revenue/Employee, and Profit Margin percent.

### **member**

In Data Manager, a node in a reference structure.

### **model**

In Data Manager, a system, consisting of fact data and metadata, that represents the aspects of a business.

### **Multidimensional Expression Language**

The multidimensional equivalent of Structured Query Language (SQL).

### **namespace**

For authentication and access control, a configured instance of an authentication provider that allows access to user and group information. In Framework Manager, namespaces uniquely identify query items, query subjects, and so on. You import different databases into separate namespaces to avoid duplicate names.

In XML and XQuery, a uniform resource identifier (URI) that provides a unique name to associate with the element, attribute, and type definitions in an XML schema or with the names of elements, attributes, types, functions, and errors in XQuery expressions.

### **package**

A subset of a model, which can be the whole model, to be made available to the Cognos server.

### **page set**

In Report Studio, a set of one or more designed pages which repeat in the report output for each instance of a chosen query item.

### **prompt**

A report element that asks for parameter values before the report is run.

### **properties pane**

Within a studio, a pane that provides an overview of the properties for selected data. The properties pane can also be used to make several changes and apply them at the same time, instead of repeating several different commands.

### **query**

A request for information from a data source based on specific conditions: for example, a request for a list of all customers in a customer table whose balances are greater than \$1000.

### **query item**

A representation of a column of data in a data source. Query items may appear in a model or in a report and contain a reference to a database column, a reference to another query item, or a calculation.

### **query subject**

A named collection of query items that are closely functionally related. Query subjects are defined using Framework Manager to represent relational data and form the set of available data for authoring reports in Query Studio and Report Studio. A query subject is similar to a relational view in that it can be treated as a table but does not necessarily reflect the data storage.

### **repeater**

In Report Studio, a cell container that repeats values within itself with no predefined internal structure.

### **repeater table**

In Report Studio, a table-like container that repeats cells across and down the page or row in the associated query.

### **report**

A set of data deliberately laid out to communicate business information.

### **report output**

The output produced as a result of executing a report specification against a data set.

### **report specification**

An executable definition of a report, including query and layout rules, which can be combined with data to produce a report output.

**style sheet**

A specification of formatting instructions that, when applied to structured information, provides a particular rendering of that information (for example, online or printed). Different style sheets can be applied to the same piece of structured information to produce different presentations of the information.

**summary**

In reporting and analysis, an aggregate value that is calculated for all the values of a particular level or dimension. Examples of summaries include total, minimum, maximum, average, and count.

**template**

In report authoring, a reusable report layout or style that can be used to set the presentation of a query or report.

In Data Manager, a component that can be used to define reference structure attributes or dimension table columns together with their semantics.

**work area**

The area within a studio that contains the report, analysis, query, or agent currently being used.

## Glossary

# Index

## Symbols

- ! characters, [357](#)
- \* characters, [495](#)
- % calculations
  - in dimensional reporting, [397](#)
  - in relational reporting, [343](#)
- 100% stacked charts, [144](#)
- 3-D charts, [144](#)
- characters
  - errors when filtering, [374](#)
  - marking special cells, [495](#)
  - troubleshooting in reports, [966](#)

## A

- absolute charts, [142](#)
- accessibility features, [985](#)
  - keyboard shortcuts, [985](#)
- accessible reports, [985](#)
  - adding alternate text, [987](#)
  - adding summary text, [987](#)
  - conditional blocks, [988](#)
  - report output options, [986](#)
  - specifying cell headers in tables, [988](#)
- active reports, [215](#)
  - adding controls, [218](#)
  - adding interactivity, [217](#)
  - controls, [215](#)
  - converting existing reports, [216](#)
  - creating and managing active report variables, [217](#)
  - defining connections between controls, [221](#)
  - example, [227, 231](#)
  - running, [226](#)
  - saving in MHT format, [226](#)
  - setting properties, [216](#)
  - specifying reaction behavior of controls, [223](#)
  - specifying selection behavior of controls, [222](#)
  - summarizing data, [226](#)
  - variables, [216](#)
- aggregating
  - limitations with measures, [363](#)

- mapping functions from data sources, [323](#)
- member sets, [363](#)
- modes, [83](#)
- samples and time dimensions, [569](#)
- specifying aggregation properties in models, [317](#)
- troubleshooting running totals, [981](#)
- using a simple summary in dimensional reporting, [360](#)
- using simple summaries in relational reporting, [315](#)
- values in crosstabs and charts, [361](#)

## aliases

- in maps, [208](#)
- aligning objects, [448](#)

## alternate text

- accessibility, [987](#)

## alternative hypothesis

## Analysis Studio

- creating templates in Report Studio, [549](#)

## annotations

- chart baselines, [170](#)
- notes in charts, [169](#)

## ANOVA

- post hoc tests, [262](#)

## area charts

## arithmetic overflow errors

## attributes

- definition, [991](#)

## Auto Group & Summarize

- automating IBM Cognos BI using the IBM Cognos Software Development Kit, [88](#)

## averages

- moving and rolling, [364](#)

## axes

- in charts, [127](#)
- titles in maps, [207](#)

## B

## backgrounds

- adding background effects to charts, [443](#)
- adding colors to objects, [461](#)
- changing in current default charts, [157](#)
- changing in legacy charts, [157](#)

- colors not appearing in templates, [964](#)
  - inserting background images, [442](#)
  - troubleshooting in charts, [965](#)
  - bar charts, [133](#)
  - baselines
    - adding to charts, [170](#)
  - bevels
    - adding to legacy charts, [158](#)
  - binary round-off errors, [971](#)
  - bivariate correlation, [277](#)
  - BlackBerry devices
    - creating mobile reports for, [107](#)
  - blocks
    - inserting, [445](#)
    - resizing, [456](#)
  - bookmarks
    - adding, [540](#)
    - creating for drill-through access, [518](#)
  - borders
    - adding to charts, [443](#)
    - adding to legacy charts, [158](#)
    - adding to objects, [440](#)
  - bottom filtering, [354](#)
  - boxplots, [248](#)
  - bubble charts, [136](#)
    - specifying bubble size in current default charts, [190](#)
  - Build Prompt Page tool, [422](#)
  - bullet charts, [137](#)
    - customizing current default charts, [194](#)
  - bursting
    - definition, [991](#)
  - burst reports, [487](#)
    - creating against cubes, [492](#)
    - creating against dimensional data sources, [492](#)
    - creating burst tables, [488](#)
    - creating calculated fields to define recipients, [487](#)
    - defining groups, [490](#)
    - defining recipients, [487](#)
    - denial of service errors in dimensional reporting, [388](#)
    - denial of service errors in relational reporting, [339](#)
    - enabling in IBM Cognos Connection, [492](#)
    - Microsoft Excel format, [598](#)
    - setting options, [490](#)
    - troubleshooting, [978](#)
  - Business Insight dashboards
    - filtering, [105](#)
  - business keys
    - using for drill-through, [521](#)
  - button bars, [219](#)
- C**
- calculated fields in burst reports, [487](#)
  - calculated members
    - definition, [991](#)
  - Calculation Intersection, [406](#)
  - calculations
    - adding in dimensional reporting, [396](#)
    - adding in relational reporting, [342](#)
    - building in dimensional reporting, [398](#)
    - building query calculations in relational reporting, [344](#)
    - creating simple in dimensional reporting, [397](#)
    - creating simple in relational reporting, [343](#)
    - function quality of service indicators in dimensional reporting, [402](#)
    - function quality of service indicators in relational reporting, [344](#)
    - intersecting in crosstabs and charts, [404](#)
    - limitations in dimensional reporting, [403](#)
    - limitations in relational reporting, [343](#)
    - limitations with units of measure in dimensional reporting, [403](#)
    - limitations with units of measure in relational reporting, [342](#)
    - null values, [407](#)
    - position, [69](#)
    - resolving in dimensional data sources, [406](#)
    - solve order in dimensional reporting, [402](#)
    - solve order in relational reporting, [343](#)
    - unexplained number discrepancies, [971](#)
    - using quotation marks in dimensional reporting, [403](#)
    - using quotation marks in relational reporting, [345](#)
  - CAMID, *See* [IBM Cognos Access Manager ID](#)
  - cardinality
    - definition, [991](#)
  - cascading prompts
    - adding, [435](#)
    - definition, [991](#)
  - cascading style sheets, [459](#)
  - cases
    - statistics, [240](#)

- cast\_Date functions
  - troubleshooting, [978](#)
- categorical data, [236](#)
- changed features
  - data formats for OLAP, [69](#)
  - data trees for SAP BW, [72](#)
  - equality filters, [71](#)
  - for aggregate, [71](#)
  - in version 8.4, [64](#)
  - slicers, [69](#)
  - summaries on non-additive measures for SAP BW, [70](#)
- chart configurations
  - 100% stacked, [144](#)
  - 3-D, [144](#)
  - stacked, [143](#)
  - standard, [142](#)
- charts, [127](#)
  - accessible reports, [988](#)
  - adding alternate text, [987](#)
  - aggregating values, [361](#)
  - background effects, [443](#)
  - background effects in legacy charts, [158](#)
  - backgrounds in current default charts, [157](#)
  - backgrounds in legacy charts, [157](#)
  - bands of color in the background, [165](#)
  - baselines, [170](#)
  - borders, [443](#)
  - changing backgrounds in current default charts, [157](#)
  - changing backgrounds in legacy charts, [157](#)
  - changing colors, [151](#)
  - changing lines, [161](#)
  - changing scales, [161](#)
  - colored regions in current default charts, [159](#)
  - configurations, [130](#)
  - converting current default charts to a matrix, [184](#)
  - converting types, [145](#)
  - creating, [127](#)
  - cumulation line in current default Pareto charts, [192](#)
  - cumulation line in legacy Pareto charts, [193](#)
  - current default, [127](#)
  - customizing current default combination charts, [183](#)
  - customizing data labels in current default charts, [179](#)
  - customizing labels, [161](#)
  - customizing legend items in current default charts, [167](#)
  - customizing legend items in legacy charts, [168](#)
  - customizing palettes, [151](#)
  - defining colors by values in current default charts, [188](#)
  - defining query contexts, [180](#)
  - displaying every second label, [964](#)
  - drilling up and down, [194](#)
  - drop shadows, [443](#)
  - elements, [127](#)
  - fills, [443](#)
  - gridlines, [165](#)
  - inserting microcharts into crosstabs, [193](#)
  - legacy, [127](#)
  - legacy and current default, [562](#)
  - markers, [173](#)
  - notes, [169](#)
  - positioning legends in current default charts, [166](#)
  - properties not supported in Microsoft Excel, [598](#)
  - regression lines in current default charts, [176](#)
  - regression lines in legacy charts, [177](#)
  - resizing, [456](#)
  - resizing current default, [150](#)
  - resizing legacy, [150](#)
  - resolving multiple calculations, [404](#)
  - series colors in current default combination charts, [183](#)
  - setting properties, [147](#)
  - showing data points, [181](#)
  - showing value markers, [181](#)
  - specifying bubble size in current default bubble charts, [190](#)
  - specifying default measures, [122](#)
  - summarizing small slices or items in current default charts, [181](#)
  - tooltips, [147](#)
  - troubleshooting axis labels, [964](#)
  - troubleshooting gray gradient backgrounds, [965](#)
  - types, [130](#)
  - types not supported in Microsoft Excel, [599](#)
  - chart types, [130](#)
    - area charts, [133](#)
    - bar charts, [133](#)
    - bubble charts, [136](#)
    - bullet charts, [137](#)
    - column charts, [131](#)
    - combination charts, [134](#)

- gauge charts, 138
- line charts, 131
- Marimekko charts, 140
- metrics range charts, 142
- microcharts, 139
- Pareto charts, 138
- pie charts, 132
- point charts, 134
- polar charts, 141
- progressive column charts, 139
- quadrant charts, 136
- radar charts, 141
- scatter charts, 135
- win-loss charts, 139
- checkbox groups, 219
- classes
  - changing the default for a report, 459
- clipboards
  - copying reports, 552
  - opening reports, 552
- coefficients, 282
- Cognos SQL
  - in dimensional reporting, 389
  - in relational reporting, 340
- collinearity diagnostics, 282
- colored regions
  - adding to current default charts, 159
- colors
  - adding to current default charts, 188
  - adding to objects, 461
  - changing in charts, 151
  - not appearing in templates, 964
- column charts, 131
- columns, 91
  - adding multiple items in a single column, 451
  - calculated in dimensional reporting, 396
  - calculated in relational reporting, 342
  - grouping in relational reporting, 308
  - hiding or showing in list controls, 225
  - setting the group span, 310
  - sorting in dimensional reporting, 377
  - sorting in relational reporting, 329
  - swapping with rows, 122
  - width limitations in Microsoft Excel, 596
- combination charts, 134
- concatenating strings, 404
- conditional block lists
  - inserting, 445
- conditional blocks
  - accessible reports, 988
- conditional palettes
  - creating, 153
- conditional rendering, 478
  - adding multiple layouts, 484
  - adding variables, 478
  - example, 481
  - hiding and showing objects, 480
- conditional styles
  - creating, 473
  - using variables, 477
- Condition Explorer, 77
- conditions
  - definition, 991
- confidence intervals, 236
- Content Manager
  - definition, 991
- context filters
  - creating, 371
  - prompting with, 373
- control charts, 292
- controls
  - active reports, 215
  - adding data, 220
  - adding to an active report, 218
  - button bars, 219
  - buttons, 219
  - checkbox groups, 219
  - data button bars, 219
  - data checkbox groups, 219
  - data decks, 218
  - data drop down lists, 219
  - data list boxes, 219
  - data radio button groups, 219
  - data tab controls, 219
  - data toggle button bars, 219
  - decks, 218
  - defining connections, 221
  - drop down lists, 219
  - list boxes, 219
  - radio button groups, 219
  - specifying reaction behavior, 223
  - specifying selection behavior, 222

tab, 219  
 toggle button bars, 219  
 variable text items, 218  
 converting  
   lists to crosstabs, 123  
   lists to repeaters, 113  
   reports to templates, 549  
   to MDX in dimensional reporting, 391  
   to SQL in dimensional reporting, 391  
   to SQL in relational reporting, 341  
 copying reports, 552  
 correlations, 282  
 Count  
   changes between IBM Cognos ReportNet and IBM Cognos BI, 565  
 Count Distinct, 565  
 covariance matrix, 282  
 crosstab node members, 117  
 crosstab nodes, 117  
 crosstabs, 117  
   aggregating values, 361  
   aggregation modes, 83  
   allow sorting, 226  
   applying table styles, 449  
   creating single-edge, 120  
   crosstab node members, 117  
   crosstab nodes, 117  
   default measures, 122  
   discontinuous, 125  
   formatting, 118  
   indenting data, 123  
   intersections, 118  
   nesting data, 121  
   order of styles, 118  
   overflow errors, 978  
   resolving multiple calculations, 404  
 CRX-API-0018 errors, 402  
 CSV format  
   importing your own files, 463  
   producing reports in, 98  
 cubes  
   definition, 991  
   troubleshooting drilling through to relational data, 983  
 cumulation lines  
   in current default charts, 192  
   in legacy charts, 193  
 currencies  
   data format limitations in Microsoft Excel output, 596  
   formatting, 495  
   formatting disappearing in SSAS 2005, 980  
 current default charts, 127, 562  
 curve estimation, 287

**D**

dashboards  
   creating, 417  
   using bullet charts, 137  
   using gauge charts, 138  
   using global filters, 421  
 data button bars, 219  
 data checkbox groups, 219  
 data decks, 218  
 data does not appear in target report  
   troubleshooting drilling through, 980  
 data drop down lists, 219  
 data formats, 495  
   changed features for OLAP, 69  
   date and time symbols, 501  
   decimal format symbols, 509  
   locale-sensitive, 497  
   Microsoft Excel limitations, 596  
   specifying default, 495  
   specifying for objects, 496  
   using patterns, 500  
 data handling, 66  
 data list boxes, 219  
 data packages  
   refreshing, 90  
   specifying, 89  
 data points  
   showing in charts, 181  
 data radio button groups, 219  
 data series in charts, 127  
 data sources, 347  
   definition, 992  
 data stores  
   conformed dimensions, 520  
 data tab controls, 219  
 data toggle button bars, 219

data trees  
  changes for SAP BW, 72  
  customizing, 349  
  definition, 992  
date & time prompts, 424  
date prompts, 424  
dates  
  filtering, 327  
  using for drill-through access, 523  
decimals  
  rounding, 496  
decks, 218  
degrees of freedom, 236  
denial of service errors  
  in dimensional reporting, 388  
  in relational reporting, 339  
depth  
  charts, 147  
derived columns, 327  
descriptive statistics, 241  
detail filters, 324  
diagrams  
  Metric Studio, 200  
dial charts, 138  
dictionary properties in maps, 208  
dimensional coercion rules, 408  
dimensional data sources  
  creating burst reports, 492  
  creating master detail relationships, 388  
  definition, 992  
  resolving calculations, 406  
  running reports against, 97  
  using with queries, 379  
dimensionally-modeled relational data sources  
  adding dimensional data, 347  
  limitations when aggregating measures, 363  
dimensional reporting, 347  
  adding data, 347  
  best practices, 74  
  drilling up and down, 413  
  filtering data, 371  
  sorting data, 375  
  summarizing data, 358  
  using calculations, 396  
  working with queries, 377  
dimensions, 91, 347  
  conformed, 380  
  conformed for drill-through access, 520  
  definition, 992  
  non-conformed, 380  
  overlapping named set levels, 965  
  searching, 350  
discontinuous crosstabs  
  creating, 125  
distributing reports, *See* burst reports  
dividing by zero  
  troubleshooting, 963  
Don't Print class, 460  
donut charts  
  customizing data labels in current default charts, 178  
double counting, 565  
drill down  
  definition, 992  
drilling down, 413  
  creating drill-up/drill-down reports, 414  
  in charts, 194  
  using member sets, 414  
drilling through, 515  
  business keys, 521  
  concepts, 516  
  creating drill-through reports, 523  
  cubes to relational data, 983  
  data does not appear in target report, 980  
  IBM Cognos Visualizer reports, 533  
  in maps, 211  
  links not active in Safari Web browsers, 980  
  members and values, 519  
  Microsoft Excel limitations, 597  
  model-based access, 516  
  multiple values, 524  
  packages, 518  
  paths, 516  
  PowerCubes and packages, 523  
  PowerPlay reports, 533  
  relational data to cubes, 984  
  report-based access, 516  
  report formats, 517  
  selection contexts, 517  
  specifying drill-through text, 526  
  troubleshooting PowerCubes, 982  
  using bookmarks, 518

- using dates, [523](#)
  - using member unique names, [519](#)
  - using scope in model-based reporting, [521](#)
  - using URLs, [518](#)
  - with conformed dimensions, [520](#)
  - wrong data appears in target report, [980](#)
  - drilling up**, [413](#)
    - creating drill-up/drill-down reports, [414](#)
    - in charts, [194](#)
    - using member sets, [414](#)
  - drill up**
    - definition, [992](#)
  - drop down lists**, [219](#)
    - defining no list value item, [225](#)
  - drop shadows**
    - adding to legacy chart elements, [158](#)
  - drop shadows in charts**, [443](#)
- E**
- emailing reports**, *See* [burst reports](#)
  - empty cells**
    - specifying what appears for empty data containers, [500](#)
    - suppressing, [498](#)
  - equality filters**
    - changed features, [71](#)
  - error messages**
    - CRX-API-0018, [402](#)
    - denial of service in dimensional reporting, [388](#)
    - denial of service in relational reporting, [339](#)
    - HRESULT, [973](#)
    - missing items, [978](#)
    - OP-ERR-0199, [308, 963](#)
    - OP-ERR-0201, [965](#)
    - OP-ERR-0209, [374](#)
    - OP-ERR-0210, [374](#)
    - OP-ERR-0212, [374](#)
    - OP-ERR-0213, [374](#)
    - ORA-00907, [978](#)
    - overflow errors in crosstabs, [978](#)
    - PCA-ERR-0057, [981](#)
    - QE-DEF-0288, [977](#)
    - QE-DEF-0478, [409](#)
    - RQP-DEF-0177, [981](#)
    - RSV-SRV-0025, [973](#)
    - RSV-SRV-0040, [963](#)
  - UDA-SQL-0114**, [973, 981](#)
  - UDA-SQL-0206**, [973](#)
  - UDA-SQL-0564**, [981](#)
  - errors**
    - application errors upgrading a report, [963](#)
    - filtering results in error characters, [374](#)
    - null values in count summaries, [966](#)
    - out of memory, [417](#)
    - parse errors with upgraded reports, [978](#)
    - recursive evaluation, [981](#)
    - troubleshooting error cells in reports, [363](#)
  - escape characters**
    - in calculations in dimensional reporting, [403](#)
    - in calculations in relational reporting, [345](#)
  - example**
    - linear regression, [279](#)
    - one-sample T test, [257](#)
    - one-way ANOVA, [260](#)
    - one-way chi-square, [267](#)
    - two-way chi-square, [271](#)
  - Excel 2000 format**
    - deprecation notices, [60](#)
  - excluded variable statistics**, [282](#)
  - Explorer Bar**, [77](#)
  - expression editor**
    - browsing data, [401](#)
    - creating filters in dimensional reporting, [371](#)
    - creating filters in relational reporting, [324](#)
    - creating parameters to create prompts, [428](#)
    - dimensional coercion rules, [408](#)
    - searching for values, [401](#)
  - expressions**
    - length, [402](#)
    - using summary functions in relational reporting, [317](#)
  - extended data items**, [356](#)
    - enabling for new reports, [458](#)
    - report option, [85](#)
- F**
- facts**, [91](#)
  - field sets**
    - inserting, [445](#)
  - fills**
    - adding to charts, [443](#)
    - adding to legacy charts, [158](#)

## filtering

- Business Insight dashboards, 105
- creating detail filters, 324
- creating global prompts in dashboards, 421
- creating summary filters, 324
- data from SAP BW data sources, 375
- data in dimensional reporting, 371
- data in relational reporting, 324
- date columns, 327
- drilling through from IBM Cognos Series 7 to IBM Cognos BI, 534
- error characters, 374
- limitations when specifying the scope, 374
- limitations with dimensional data sources, 373
- members in a set, 355
- multiple-fact queries, 380
- suppressing null values, 498
- to top or bottom values, 354
- troubleshooting \_make\_timestamp columns, 973
- using context filters, 371
- using slicers, 371

## filters

- in\_range, 436

## first-rows optimization

- in dimensional reporting, 391
- in relational reporting, 342

## floating objects, 457

## fonts

- setting in maps, 206
- specifying colors, 461
- specifying for reports, 441

## footers

- adding to maps, 207
- adding to reports and lists, 439

## for clauses, 317

## FOR clauses

- incorrect results for OLAP, 970
- limitations when using in summary functions, 319

## foregrounds

- adding colors to objects, 461

## formatting

- disappearing in SSAS 2005, 980

## functions

- quality of service indicators in dimensional reporting, 402

quality of service indicators in relational reporting, 344

**G**

- gauge charts, 138
- generated prompts, 425
- global filters in dashboards, 421
- glossaries
  - accessing the IBM WebSphere Business Glossary, 103
- Great Outdoors Company, 568
  - databases, models, and packages, 570
  - samples, 567
- Great Outdoors Company samples, 567
- gridlines
  - adding to charts, 165
  - in charts, 127
  - major and minor, 161
- grouping
  - data in relational reporting, 308
  - setting group spans, 310
- troubleshooting incorrect running totals, 981
- troubleshooting nested list reports, 963
- troubleshooting subtotals in grouped lists, 964

## groups

- definition, 992

**H**

- headers
  - adding to reports and lists, 439
- heights
  - specifying for objects, 456
- hiding columns
  - list reports, 112
- hiding objects, 480
- hierarchies, 91
  - definition, 992
  - inserting, 351
- highlighting
  - adding colors to objects, 461
  - using conditional styles, 473
- histograms, 246
- horizontal pagination, 545
- HRESULT errors, 973
- HTML format
  - producing reports in, 94

- HTML items
  - inserting, [446](#)
- hyperlinks
  - buttons are not supported for Microsoft Excel, [597](#)
  - inserting, [446](#)
- I**
  - IBM Cognos Access Manager ID, [488](#)
  - IBM Cognos for Microsoft Office
    - creating reports for, [106](#)
  - IBM Cognos Software Development Kit, [88](#)
  - IBM Cognos Visualizer reports
    - setting up drill-through access to IBM Cognos BI, [533](#)
  - IBM WebSphere Business Glossary, [103](#)
  - icons, [91](#)
  - images
    - adding alternate text, [987](#)
    - adding to chart backgrounds, [443](#)
    - inserting in backgrounds, [442](#)
    - inserting in reports, [441](#)
    - Microsoft Excel limitations, [593](#)
  - in\_range filters, [436](#)
  - indenting
    - data in crosstabs, [123](#)
    - objects, [450](#)
  - Insertable Objects pane, [77](#)
  - intersections of members
    - in dimensional reporting, [400](#)
  - interval data, [236](#)
  - interval prompts, [424](#)
  - iPhone devices
    - creating mobile reports for, [107](#)
- J**
  - Japanese
    - Microsoft Excel limitations, [596](#)
- join relationships
  - creating in relational reporting, [336](#)
- justification
  - setting for objects, [455](#)
- L**
  - labels
    - customizing in charts, [161](#)
    - specifying in current default gauge charts, [186](#)
  - specifying in legacy gauge charts, [187](#)
  - layout component caches, [455](#)
  - layout component references
    - inserting, [446](#)
  - layouts, [80](#)
    - adding, [484](#)
    - choosing, [90](#)
    - definition, [993](#)
    - pages, [80](#)
    - relating to queries in dimensional reporting, [378](#)
    - relating to queries in relational reporting, [330](#)
    - report objects, [81](#)
    - report recommendations, [437](#)
    - using the page structure view, [437](#)
  - legacy charts, [127, 562](#)
  - legends
    - adding titles in maps, [210](#)
    - hiding or showing in maps, [206](#)
    - in charts, [127](#)
    - in maps, [210](#)
    - positioning in current default charts, [166](#)
  - level hierarchies, [347](#)
  - levels, [91, 347](#)
    - definition, [993](#)
    - member unique names, [519](#)
  - levels of measurement, [236](#)
  - LFA.dll file, [551](#)
  - limitations
    - aggregating measures in dimensionally-modeled relational or relational data sources, [363](#)
  - lineage information, [101](#)
  - linear regression, [282](#)
    - example, [279](#)
  - linear scales
    - showing, [161](#)
  - line breaks, [456](#)
  - line charts, [131](#)
  - lines
    - adding regression lines to charts, [176](#)
    - adding regression lines to legacy charts, [177](#)
    - changing in charts, [161](#)
  - line spacing, [456](#)
  - linking data items for drilling, [414](#)
  - list boxes, [219](#)
  - list reports
    - hiding columns, [112](#)

lists, 111

    adding headers and footers, 439

    allow sorting, 226

    applying table styles, 449

    converting to crosstabs, 123

    converting to repeaters, 113

    formatting, 111

    grouping data, 308

    hiding or showing columns, 225

    limitations, 112

    single-edge crosstabs, 120

    subtotals in grouped lists, 964

literal strings

    in calculations in dimensional reporting, 403

    in calculations in relational reporting, 345

locales

    definition, 993

    locale-sensitive properties, 497

location-aware reports

    creating, 107

locking objects, 82

logarithmic scales, 163

    showing, 161

## M

mailing labels

    creating, 114

maps, 203

    aliases, 208

    creating, 204

    customizing, 205

    dictionary property, 208

    drilling through, 211

    ignoring data with no features, 206

    legends, 210

    matching data values, 208

    Microsoft Excel limitations, 597

    notes, 211

    parts of a map report, 204

    point layers, 209

    region layers, 207

    resizing, 456

    showing or hiding properties, 205

margins

    setting for objects, 450

Marimekko charts, 140

markers

    adding to charts, 173

master detail relationships

    creating in dimensional reporting, 386

    creating in relational reporting, 337

    denial of service errors in dimensional reporting, 388

    denial of service errors in relational reporting, 339

    linking members from two dimensional data

    sources, 388

    using to join page sets, 537

matrix

    charts, 184

matrix reports, 117

MDX

    adding your own in dimensional reporting, 390

    converting to in dimensional reporting, 391

    working with in dimensional reporting, 389

measures, 91

    definition, 993

    specifying defaults, 122

member properties, 347

    inserting, 352

members, 91

    definition, 993

    drill-through access, 519

    excluding, 353

    expanding and collapsing, 356

    folder, 347

    inserting properties, 352

    intersections in dimensional reporting, 400

    moving, 354

member sets

    aggregating, 363

    creating, 352, 414

member unique names, 412

    relational metadata, 519

    troubleshooting running reports with, 413

metrics range charts, 142

Metric Studio diagrams, 200

MHT

    saving active reports, 226

microcharts, 139

    inserting, 193

Microsoft Excel

    limitations of nested report objects, 597

    producing reports in, 99

- report limitations, [593](#)
  - missing items
    - troubleshooting, [978](#)
  - missing values, [407](#)
    - suppressing, [498](#)
  - missing Values
    - treatment, [236](#)
  - mobile devices
    - creating reports for, [107](#)
    - unsupported features, [108](#)
  - models, [91](#)
    - aggregation properties, [317](#)
    - definition, [993](#)
    - sample models and packages, [570](#)
    - using design filters, [94](#)
  - model selection statistics, [282](#)
  - moving averages, [364](#)
  - Multidimensional Expression Language
    - definition, [993](#)
  - multilingual reports, [484](#)
  - MUNs, *See member unique names*
- N**
- named sets
    - limitations when summarizing, [360](#)
    - nested or parallel sets overlapping, [965](#)
  - names
    - updating references, [558](#)
  - namespaces
    - definition, [993](#)
  - native SQL
    - in dimensional reporting, [389](#)
    - in relational reporting, [340](#)
  - nested list reports
    - not running after upgrading, [963](#)
  - nested report objects
    - Microsoft Excel limitations, [597](#)
  - nested sets
    - unexpected summary values, [968](#)
  - nesting
    - data in crosstabs, [121](#)
  - new features
    - version 10.1.0, [53](#)
    - version 8.4, [61](#)
  - no data
    - specifying what appears, [500](#)
  - specify not to render pages, [98](#)
  - No Data Contents, [66, 500](#)
  - nominal data, [236](#)
  - nonparametric tests, [265](#)
  - normal distribution, [236](#)
  - notes
    - adding to charts, [169](#)
    - adding to maps, [211](#)
  - null, [236](#)
  - null values, [407](#)
    - calculations with SAP BW data sources, [969](#)
    - suppressing, [498](#)
    - suppressing with filters, [498](#)
    - troubleshooting in count summaries, [966](#)
  - number data formats
    - Microsoft Excel limitations, [596](#)
- O**
- objects, [82](#)
    - adding, [81](#)
    - aligning, [448](#)
    - as containers, [82](#)
    - finding in reports, [83](#)
    - hiding and showing, [480](#)
    - hierarchies, [83](#)
    - indenting, [450](#)
    - inheriting formatting from parents, [83](#)
    - inline or block, [82](#)
    - inserting formatting objects, [444](#)
    - locking and unlocking, [82](#)
    - reusing for layouts, [453](#)
    - specifying data format, [496](#)
    - updating reused, [455](#)
  - OLAP data sources
    - inserting data, [347](#)
    - limitations with relational functions, [357](#)
  - one-sample T test, [256](#)
    - example, [257](#)
  - one-sample t-tests, [255](#)
  - one-tailed test of significance, [236](#)
  - one-way ANOVA
    - example, [260](#)
  - one-way chi-square, [266](#)
    - example, [267](#)
  - opening reports
    - from other studios, [552](#)

- opening reports from your computer, [551](#)
  - OP-ERR-0049 errors, [308](#)
  - OP-ERR-0199 errors, [308, 963](#)
  - OP-ERR-0201 errors, [965](#)
  - OP-ERR-0209 errors, [374](#)
  - OP-ERR-0210 errors, [374](#)
  - OP-ERR-0212 errors, [374](#)
  - OP-ERR-0213 errors, [374](#)
  - ORA-00907 errors, [978](#)
  - Oracle data sources
    - troubleshooting, [978](#)
  - Oracle Essbase
    - changes, [965](#)
  - ordinal data, [236](#)
  - out of memory errors, [417](#)
  - overflow errors in crosstabs, [978](#)
- P**
- packages, [91, 347](#)
    - changing, [558](#)
    - definition, [993](#)
    - drilling through, [518](#)
    - referencing items in child queries, [337](#)
    - refreshing, [90](#)
    - selecting for your report, [89](#)
  - padding
    - applying to objects, [450](#)
  - page breaks, [545](#)
    - controlling, [458](#)
    - creating, [536](#)
    - creating with page layers, [547](#)
  - Page Explorer, [77](#)
  - page layers, [547](#)
  - page numbers
    - adding, [545](#)
    - modifying with options, [545](#)
  - pages, [80](#)
    - adding, [535](#)
    - associating queries with, [535](#)
    - not rendering when there is no data, [98](#)
  - page sets
    - creating, [536](#)
    - definition, [993](#)
    - joining nested page sets, [537](#)
  - pagination
    - horizontal in PDF output, [545](#)
  - specifying for HTML output, [458](#)
  - palettes
    - conditional, [153](#)
    - customizing in charts, [151](#)
  - parameters
    - for prompts, [428](#)
    - using in master detail relationships in dimensional reporting, [386](#)
    - using in master detail relationships in relational reporting, [337](#)
  - parameter values
    - clearing, [94](#)
  - Pareto charts, [138](#)
    - cumulation lines in current default charts, [192](#)
    - cumulation lines in legacy charts, [193](#)
  - part and partial correlations, [282](#)
  - partial regression plots, [282](#)
  - PCA-ERR-0057 errors, [981](#)
  - PDF format
    - producing reports, [94](#)
    - setting page options, [100](#)
  - percent calculations
    - in dimensional reporting, [397](#)
    - in relational reporting, [343](#)
  - personal data
    - importing your own files, [463](#)
  - pie charts, [132](#)
    - converting to current default donut charts, [191](#)
    - customizing data labels in current default charts, [178](#)
    - exploding slices in current default charts, [191](#)
    - limitations for Microsoft Excel outputs, [594](#)
    - setting the first slice angle in current default charts, [190](#)
  - point charts, [134](#)
  - point layers, [209](#)
  - polar charts, [141](#)
  - populations, [236](#)
  - position
    - calculations, [69](#)
  - post hoc tests
    - ANOVA, [262](#)
  - PowerCubes
    - incorrect results, [969](#)
    - troubleshooting drilling through, [982](#)

PowerPlay reports  
 setting up drill-through access to IBM Cognos BI, 533

printing  
 not printing objects, 460

progressive column charts, 139

prompt pages  
 creating your own, 423

prompts, 421  
 adding prompt buttons, 434  
 changing prompt control interfaces, 430  
 creating cascading, 435  
 creating in a report page, 429  
 creating using query macros, 511

date, 424  
 date & time, 424  
 definition, 994  
 generated, 425  
 in\_range, 436  
 interval, 424  
 modifying, 430  
 pages, 422  
 parameters, 428  
 requiring user input, 431  
 select & search, 424  
 selecting multiple values, 431  
 showing and hiding, 432  
 specifying default selections, 432  
 specifying default values, 433  
 text box, 423  
 time, 424  
 tree, 435  
 upgrading SAP BW variables, 562  
 using context filters, 373  
 using slicer filters, 373  
 using the Build Prompt Page tool, 422  
 using the expression editor, 428  
 value, 423

Properties pane, 77

properties panes  
 definition, 994

**Q**

QE-DEF-0288 errors, 977

QE-DEF-0478 errors, 409

Q-Q plots, 254

quadrant charts, 136

quadrants  
 adding to current default charts, 159

quantitative data, 236

queries, 81  
 associating with pages, 535  
 connecting between data sources in dimensional reporting, 379  
 connecting between data sources in relational reporting, 330  
 definition, 994  
 in dimensional reporting, 377  
 in relational reporting, 329  
 multiple-fact, 380  
 referencing data items in relational reporting, 330, 378  
 referencing package items in child queries, 337  
 relating to layouts in dimensional reporting, 378  
 relating to layouts in relational reporting, 330  
 using with dimensional data sources, 379  
 working with MDX in dimensional reporting, 389  
 working with SQL in dimensional reporting, 389  
 working with SQL in relational reporting, 340

query calculations  
 creating in dimensional reporting, 398  
 creating in relational reporting, 344  
 null values in count summaries, 966

query context  
 defining in charts, 180

Query Explorer, 77  
 using in dimensional reporting, 377  
 using in relational reporting, 329

query items  
 definition, 994

query macros, 511  
 adding, 511  
 using to create prompts, 511

Query Studio  
 creating templates in Report Studio, 549

query subjects  
 definition, 994

quick tours, 51

quotation marks  
 in calculations in dimensional reporting, 403  
 in calculations in relational reporting, 345

**R**

- radar charts, 141
- radio button groups, 219
- ragged hierarchies, 357
- ratio data, 236
- referenced objects
  - changing, 454
  - updating, 455
- refreshing packages, 90
- region layers, 207
- regression lines, 161
  - in current default charts, 176
  - in legacy charts, 177
- regression methods
  - backward, 282
  - forward, 282
  - stepwise, 282
- relational functions
  - limitations when used with OLAP data sources, 357
- relational reporting, 307
  - adding data, 307
  - best practices, 74
  - filtering data, 324
  - grouping data, 308
  - sorting data, 328
  - summarizing data, 314
  - troubleshooting drilling through to cubes, 984
  - using calculations, 342
  - working with queries, 329
- Render Page when Empty, 98
- repeaters
  - converting from lists, 113
  - definition, 994
  - formatting, 449
- repeater tables
  - definition, 994
- report formats
  - CSV, 98
  - Excel, 99
  - HTML, 94
  - PDF, 94
  - specifying, 94
  - XML, 100
- ReportNet reports
  - formatting changes when upgrading, 979
- report outputs
  - accessibility options, 986
  - definition, 994
- report properties, 458
- reports
  - active, 215
  - adding interactivity, 217
  - converting to active reports, 216
  - definition, 994
  - multilingual, 484
- report samples, 567
- report specifications, 88
  - definition, 994
- residuals statistics, 282
- resizing
  - current default charts, 150
  - legacy charts, 150
  - objects, 456
- reusing
  - styles, 438
- rich text items
  - inserting, 446
  - supported elements, 447
- rolling averages, 364
- rollup calculations
  - in dimensional reporting, 397
  - in relational reporting, 343
- rounding numbers, 496
- round-off errors, 971
- row numbers
  - inserting, 446
- rows
  - swapping with columns, 122
- rows per page
  - controlling for multiple containers in HTML and PDF, 98
  - setting, 94
- RQP-DEF-0177 errors, 563, 981, 982
- R square change, 282
- RSV-SRV-0025 errors, 973
- RSV-SRV-0040 errors, 963
- running and moving summaries
  - limitations with dimensional functions, 403
- running reports, 94
  - against dimensional data sources, 97
  - against SAP BW, 97

- cannot find database errors, [977](#)
- ORA-00907 errors, [978](#)
- Report Studio is slow, [973](#)
- running totals
  - dimensional functions, [396](#)
  - troubleshooting, [981](#)
- run options
  - accessibility options, [986](#)
- S**
  - Safari Web browsers
    - drill-through links not active, [980](#)
  - samples, [236, 567](#)
    - cubes, [572](#)
    - database, models, and packages, [570](#)
    - employees, [569](#)
    - GO data warehouse, [570](#)
    - GO Data Warehouse (analysis) package, [576](#)
    - GO Data Warehouse (query) package, [584](#)
    - GO Sales (analysis) package, [586](#)
    - GO Sales (query) package, [588](#)
    - GO Sales transactional database, [571](#)
    - interactive, [590](#)
    - packages, [572](#)
    - Sales and Marketing (Cube) package, [573](#)
    - sales and marketing data, [569](#)
    - statistical analysis, [589](#)
    - The Great Outdoors Company, [568](#)
  - SAP BW data sources
    - creating burst reports, [492](#)
    - creating expression, [404](#)
    - large queries, [379](#)
    - limitations when calculating data, [342](#)
    - limitations when sorting data, [376](#)
    - null results, [969](#)
    - query calculations, [966](#)
    - running reports against, [97](#)
    - sectioning reports, [966](#)
    - units of measure notation, [97](#)
  - SAP variable properties
    - unsupported, [97](#)
  - saving reports to your computer, [551](#)
  - scales
    - changing intervals, [161](#)
    - linear, [161](#)
    - logarithmic, [161](#)
  - scatter charts, [135](#)
  - scatterplot of residuals versus dependent variables, [282](#)
  - scope
    - setting in summary filters, [324](#)
    - using in drill-through access, [521](#)
  - screen resolution requirements, [76](#)
  - scrollbars
    - adding to block objects, [456](#)
  - searching
    - dimensions, [350](#)
  - section headings, [313](#)
  - sections
    - creating for dimensional data, [547](#)
    - creating on reports with SAP BW data sources, [966](#)
  - Secure Socket Layer
    - Microsoft Excel limitations, [596](#)
  - select & search prompts, [424](#)
  - set definitions, [353](#)
  - set expressions
    - building in dimensional reporting, [398](#)
    - limitations in list reports, [112](#)
  - sets
    - editing, [353](#)
    - filtering, [355](#)
  - sets of members
    - creating, [352](#)
    - nested or parallel sets overlapping, [965](#)
  - showing objects, [480](#)
  - single-edge crosstabs, [120](#)
  - singletons, [92](#)
  - sizes
    - specifying for objects, [456](#)
  - slicer filters
    - prompting with, [373](#)
  - slicers
    - changed features, [69](#)
    - creating, [371](#)
  - solve order, [404](#)
    - combined with calculated rollup aggregate function, [405](#)
    - dimensional data sources, [406](#)
  - sorting
    - crosstabs, [226](#)
    - data in dimensional reporting, [375](#)
    - data in relational reporting, [328](#)
    - limitations with SAP BW data sources, [376](#)

- lists, 226
- multiple columns in dimensional reporting, 377
- multiple columns in relational reporting, 329
- source trees
  - customizing, 349
- spaces
  - adding around objects, 450
- special values, 69
- spider charts, 141
- SQL
  - adding your own in dimensional reporting, 390
  - adding your own in relational reporting, 340
  - Cognos in dimensional reporting, 389
  - Cognos in relational reporting, 340
  - converting to in dimensional reporting, 391
  - converting to in relational reporting, 341
  - first-rows optimization in dimensional reporting, 391
  - first-rows optimization in relational reporting, 342
  - native in dimensional reporting, 389
  - native in relational reporting, 340
  - working with in dimensional reporting, 389
  - working with in relational reporting, 340
- SSAS 2005 cubes
  - troubleshooting, 967
- SSAS 2005 data sources
  - creating expression, 404
  - disappearing data formats, 980
- SSAS cubes
  - problems when very large, 417
- stacked charts, 143
- standard charts, 142
- standard deviation, 236
- standard error, 236
- standardized residuals histograms, 282
- standardized residuals PP plots, 282
- star charts, 141
- statistical analysis
  - ANOVA, 262
  - bivariate correlation, 277
  - boxplots, 248
  - control charts, 292
  - curve estimation, 287
  - descriptive statistics, 241
  - histograms, 246
  - linear regression, 282
- one-sample T test, 256
- one-way chi-square, 266
- Q-Q plots, 254
- statistical process control, 291
- terms, 236
  - two-way chi-square, 270
- statistical power, 236
- statistical process control, 291
- statistical significance, 236
- statistics
  - cases, 240
- strings
  - concatenating, 404
- studios
  - Oracle Essbase changes, 965
- styles
  - conditional, 473
  - reusing, 438
- style sheets
  - definition, 994
- subtotals
  - troubleshooting in grouped lists, 964
- summaries
  - adding simple in dimensional reporting, 360
  - adding simple in relational reporting, 315
  - adding to active reports, 226
  - aggregate in dimensional reporting, 366
  - aggregate in relational reporting, 320
  - automatic in dimensional reporting, 366
  - automatic in relational reporting, 320
  - average in dimensional reporting, 366
  - average in relational reporting, 320
  - calculated, 405
  - calculated in dimensional reporting, 366
  - calculated in relational reporting, 320
  - count distinct in dimensional reporting, 367
  - count distinct in relational reporting, 321
  - count in dimensional reporting, 367
  - count in relational reporting, 321
  - custom in dimensional reporting, 368
  - custom in relational reporting, 321
  - definition, 995
  - FOR clauses, 970
  - maximum in dimensional reporting, 368
  - maximum in relational reporting, 321
  - median in dimensional reporting, 368

median in relational reporting, 321  
minimum in dimensional reporting, 368  
minimum in relational reporting, 321  
none in dimensional reporting, 368  
none in relational reporting, 321  
not applicable in dimensional reporting, 368  
not applicable in relational reporting, 322  
null values, 407  
rolling and moving averages, 364  
standard deviation in dimensional reporting, 368  
standard deviation in relational reporting, 322  
summarize in dimensional reporting, 369  
summarize in relational reporting, 322  
total in dimensional reporting, 369  
total in relational reporting, 322  
troubleshooting, 976  
variance in dimensional reporting, 369  
variance in relational reporting, 323  
summarizing data  
  in dimensional reporting, 358  
  in relational reporting, 314  
summary filters, 324  
summary functions  
  in dimensional reporting, 365  
  in relational reporting, 320  
  limitations when using FOR clauses, 319  
  using in expressions in relational reporting, 317  
summary text  
  adding to tables, 987  
summary values  
  unexpected when using nested sets, 968  
suppressing  
  data, 405  
Symbian devices  
  creating mobile reports for, 107

**T**

tab controls, 219  
table of contents, 66  
tables  
  adding summary text, 987  
  applying styles, 449  
  inserting, 448  
  Microsoft Excel width limitations, 596  
  specifying cell headers, 988

tables of contents  
  creating, 542  
tabular data  
  viewing, 94  
target reports  
  creating for drilling through from IBM Cognos Series 7, 534  
templates  
  background colors not appearing, 964  
  converting from reports, 549  
  creating, 549  
  definition, 995  
text  
  adding to reports, 441  
  specifying flow in objects, 455  
text box prompts, 423  
textures  
  adding to legacy charts, 158  
time periods  
  showing data only for specific, 365  
time prompts, 424  
TM1 data sources  
  order of metadata tree, 968  
  report differences, 968  
toggle button bars, 219  
tooltips  
  adding to charts, 147  
  hiding or showing in maps, 206  
top filtering, 354  
totals  
  troubleshooting running totals, 981  
tree prompts  
  adding, 425  
  controlling the data that appears, 435  
trend lines, 161  
  in current default charts, 176  
  in legacy charts, 177  
tuples  
  in dimensional reporting, 400  
two-tailed test of significance, 236  
two-way chi-square, 270  
  example, 271  
TXT data  
  importing your own files, 463  
types of error, 236

## U

UDA-SQL-0043 errors, 982  
UDA-SQL-0114 errors, 973, 981  
UDA-SQL-0206 errors, 973  
UDA-SQL-0458 errors, 563  
UDA-SQL-0564 errors, 981  
unbalanced hierarchies, 357  
unexplained number calculations, 971  
union query  
    creating, 332  
units of measure, 495  
    notation for SAP BW data sources, 97  
unlocking objects, 82  
upgrading  
    SAP BW prompt variables, 562  
upgrading reports  
    applications errors, 963  
    errors, 563  
    formatting not retained, 979  
    members not suppressed for SAP BW, 563  
    removing upgrade messages, 561  
    reports will not open, 978  
    thousands separators missing, 563  
user interface, 76

## V

validating reports, 93  
value markers  
    showing in charts, 181  
value prompts, 423  
values  
    special, 69  
    specifying defaults in prompts, 433  
variable processing summary, 282  
variables  
    active reports, 216, 217  
    adding, 478  
    conditional styles, 477  
variable text items, 218  
viewing reports  
    troubleshooting, 978  
views, 79  
    page design view, 80  
    page structure view, 80  
visual aids, 78

## W

waterfall charts, 139  
Web browser settings, 87  
Web-safe color palettes, 461  
white spaces  
    adding padding to objects, 450  
    margins, 450  
widths  
    Microsoft Excel limitations, 596  
    specifying for objects, 456  
Windows Mobile devices  
    creating mobile reports for, 107  
win-loss charts, 139  
word breaks, 456  
work areas  
    definition, 995  
wrong data appears in target report  
    troubleshooting drilling through, 980

## X

XLS format  
    importing your own files, 463  
    limitations, 593  
XML  
    importing your own files, 463  
    producing a report, 100  
    report specifications, 88  
XQE-CON-0007 errors, 379

## Z

zeros  
    suppressing, 498