SIEMENS

XHQ

Developer's Guide

About This Guide

XHQ Overview

Developing Your Information Model 1

Development Fundamentals 2

XHQ Workbench 3

Building and Managing Components 4

Views and View Elements 5

Working with View Building Tools 6

Building Views 7

Working with Collections and Flow Views 8

Working with Tags 9

The Solution Builder 10

Building and Configuring Solutions 11

Exporting and Importing 12

Appendices A

11/2020 V6.1

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER
Indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ **WARNING**
Indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ **CAUTION**
Indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
Indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage. See the topic, Visual Cues for Online Viewing, for additional XHQ-specific notices.

Qualified Personnel

The product/system described in this documentation may be operated only by personnel qualified for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ **WARNING**
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner. For a complete list, see the Copyright topic.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Copyright © 1998-2020 Siemens AG. All rights reserved. Protected by U.S. Patents Nos. 6,700,590, 7,069,514, 7,478,128, 7,689,579, 7,698,292, 7,814,123, 7,840,607, 8,001,332, 8,078,598, 8,260,783, 8,442,938, 8,566,781, 8,700,671 and 8,700,559; Patents Pending.

Siemens Industry Software Inc.
6 Journey, Suite 200
Aliso Viejo, CA 92656-5318, USA
siemens.com/xhq

XHQ® is a registered trademark of Siemens AG in the United States. This License does not grant LICENSEE any rights to trademarks or service marks of Siemens AG.

All other company, product and service names and logos may be trademarks or service marks of their respective companies. Any rights not expressly granted herein are reserved. LICENSEE may not remove or alter any trademark, logo, copyright or other proprietary notices, legends, symbols or labels from the Licensed Software or the Documentation.

This software is proprietary and confidential. Siemens AG or its suppliers own the title, copyright, and other intellectual property rights in the Software. The Software is licensed, not sold.

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

Microsoft, Active Directory, ActiveX, Authenticode, Developer Studio, DirectX, Microsoft, MS-DOS, Outlook, Excel, PowerPoint, Visual Basic, Visual C++, Visual C#, Visual J#, Visual SourceSafe, Visual Studio, Win32, Windows, Windows Server, WinFX, Windows 7, Windows 10, Windows Server 2012, Windows Server 2016, Windows Server 2019, and the Windows logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries, or both.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Oracle, Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates. Oracle, or its licensor, shall at all times retain all rights, title, interest, including intellectual property rights, in Oracle Programs and media.

SAP, SAP R/3, R/3, R/3 software, mySAP, mySAP.com, xApps, xApp, ABAP, BAPI, and SAP NetWeaver are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

Documentum, OpenText Documentum, OpenText and the Corporate Logo are trademarks or registered trademarks of OpenText in the United States and throughout the world.

IBM, the IBM logo, DB2, and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

InstallShield® is a registered trademark and service mark of Macrovision Corporation and/or Macrovision Europe Ltd. in the United States and/or other countries. DemoShield, InstallFromTheWeb and PackageForTheWeb are service marks and registered trademarks of Macrovision Corporation and/or Macrovision Europe Ltd. in the United States and/or other countries. InstallShield Express, InstallShield for Windows Installer, InstallShield for Windows CE, Express Wizard, InstallShield Objects, WebUpdate, FastReg and NetInstall are trademarks and/or service marks of Macrovision Corporation and/or Macrovision Europe Ltd. InstallShield Software Corporation. InstallShield is a member of Macrovision Corporation.

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions only form one element of such a concept.

Customer is responsible to prevent unauthorized access to its plants, systems, machines and networks. Systems, machines and components should only be connected to the enterprise network or the internet if and to the extent necessary and with appropriate security measures (e.g. use of firewalls and network segmentation) in place.

Additionally, Siemens' guidance on appropriate security measures should be taken into account. For more information about industrial security, please visit https://www.siemens.com/industrialsecurity.

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends to apply product updates as soon as available and to always use the latest product versions. Use of product versions that are no longer supported, and failure to apply latest updates may increase customer's exposure to cyber threats.

For the Siemens Security Advisory, visit https://www.siemens.com/industrialsecurity.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed under https://www.siemens.com/industrialsecurity.

While every effort is made to ensure the accuracy of content, the XHQ product documentation set (which includes online help) could contain inaccuracies or out-dated material (which includes product screenshots and images) due to the large number of product enhancements being added. As such, the documentation set is subject to change at any time without notice. Refer to the README for documentation corrections and addendum. Please note, updates to the documentation set are reflected in the next general availability major release of XHQ.

Table of Contents

About This Guide 24
Conventions Used in This Guide 24
Visual Cues for Online Viewing 25
Related XHQ Product Documentation 26
Contacting Customer Support 28
General Feedback and Comments 29
XHQ Overview 30
6 Steps to Building a Visual Enterprise 31
Step 1 - Develop Your Information Model 32
Step 2 - Develop Reusable Components 32
Step 3 - Build the Solution 32
Step 4 - Configure Components 33
Step 5 - Build Views 34
Step 6 - Deploy and Access the Solution 35
1 | Developing Your Information Model 36
What is an Information Model? 36
Why build an Information Model? 37
Designing Your Information Model 38
Model Examples 38
Things to Consider 40
Your Business Objective 40
The User Community 40
The Backend Systems 41
Organizing Information 41
Practical Guidelines 42
2 | Development Fundamentals 43
Starting XHQ 43
To start and stop the servers 45
To restart the server 46
XHQ Development Start Menu 47
Defining the XHQ Panels 49
The Navigation Panel 50
The Inventory Items Panel 50
The Edit Panel 50
The View Panel 51

Using XHQ Navigation 53
Left- and Right-Click Operations in the View Panel 54
About Common Operations 55
About Peer Views 56
About Links 56
Tree-style Navigation 56
Left- and Right-Click Operations in the Navigation Tree 57
Accessing Help 58
The XHQ Runtime Environment 59
XHQ Solution Viewer at Runtime 59
XHQ Solution Builder at Runtime 60

3 | XHQ Workbench 61
To access the XHQ Workbench 61
XHQ Workbench Menu Commands 63
Common Menu Commands 63
Component Building Menu Commands 66
View Building Menu Commands 67
Right-click Shortcut Menu Commands 72
View Submenu Grouping 74
Customizing Layouts 78
Moving and Docking Panels and Toolbars 78
Managing Layouts 79
To save a customized layout 79
To set a different default layout 80
To switch layouts 80
To revert back to the factory default layout 80
To delete a layout 80
Working with Tabs 81
Tab Tools 84

4 | Building and Managing Components 86
Creating and Editing Components 86
How to name components 86
About Base and Subclass Components 86
To create a [base] component 89
To create a subclass component 92
To edit a component 93
Multiple Levels of Subclassing 93

About Inheritance ... 93
About Export/Import ... 94
Renaming Components ... 95
To rename an existing component globally ... 95
Choosing Your Component Icons ... 96
To assign an icon to a component ... 96
Saving Your Component Changes ... 97
To save component changes in the Component Details tab ... 97
To update your solution to reflect component changes ... 97
Deleting Components from the XHQ Workbench ... 98
To delete components from the XHQ Workbench ... 99
Working with Members ... 100
Defining Primitive Members ... 102
Adding Members ... 103
Assigning a dimension ... 104
Adding Group Members ... 105
To add an existing component member ... 107
To add a new component member ... 107
To add a primitive member ... 108
To add more than one member of the same type ... 108
To rename a component member ... 108
Configuring Members ... 110
Hiding Members and De-cluttering the Tree ... 110
Setting the Tag Attribute ... 114
Deleting Members ... 115
To delete a component member ... 116
Organizing the XHQ Workbench ... 117
To add a page ... 117
To rename a page ... 119
To delete a page ... 120
Assigning Components to Pages ... 121
To assign a component to a page ... 121
To un-assign a component from a page ... 121
To create a new component and add it to a page ... 122
Dealing with Multiple Users ... 123
General Rules for Multiple Users ... 124
Reports ... 125
Generating a Component-Views Summary ... 125

To generate and access a Component-Views summary 125
Generating a Component Summary 126
To generate and access a Component summary 126
Tutorial: Building Components 126
To set-up the repos environment for this tutorial 127
To build components from the XHQ Workbench 127
To add component members 131

5 | Views and View Elements 132
The View Building Environment 132
The Tools Palette 133
The Button Bar 134
The Configuration Properties Tab 135
About the General Properties 136
About the Animation Properties 137
About the Links Properties 139
About the Tooltips Properties 141
About the Tool-specific Properties 141
The Drawing Panel 143
Coordinate System 144
Right-click Menu Commands 144
Performing Basic View Building Tasks 146
Creating and Naming Views 146
Assigning a Unique Name 147
To create a new view 148
To open a view 148
Saving and Closing a View 149
To save a view 149
Reverting to Last Save 149
To revert to a previously saved image 149
Deleting a View 149
To delete a view from an existing component 149
Clearing a View 150
To clear a view 150
Using Undo/Redo 150
Undo Preferences 151
Working with View Elements 152
Pointing and Selecting a View Element 152
To select a view element 152

To select multiple view elements 152
To deselect all view elements 153
To delete a view element 153
Positioning a View Element 153
To move a view element interactively 153
To nudge a view element 153
To move a view element to a precise location 154
Locking View Element(s) Into Position 154
To lock a view element(s) into position 154
Aligning View Elements 155
Flipping an Image 155
Rotating a View Element 156
To rotate a view element 157
Sizing a View Element 158
Resizing Proportionally 158
Determining Position and Size 158
To size a view element using the mouse 159
To resize a view element proportionally 159
To set a precise size for a view element 159
Cutting, Copying, and Pasting 160
To cut and paste a view element selection 160
To copy and paste a view element selection 161
To delete a view element selection 161
To duplicate view elements 161
Setting the Text Style 162
To add fonts to the font list 163
Aligning Text 163
Copying and pasting a View Element style 164
Customizing Tooltips 175
About Custom Tooltips 175
How expressions are evaluated 178
How Tooltips are prioritized in XHQ 179
The Exported XML 179
Arranging Your View Elements 180
Working with Grouped View Elements 180
To group view elements 180
To ungroup view elements 180
To change the membership of a group of view elements 180

Editing Nested View Elements ... 181
Repositioning and Resizing Nested View Elements ... 183
How to Order View Elements ... 184
To order view elements ... 184
How to Align View Elements ... 185
To align a series of two or more view elements ... 185
How to Resize View Elements ... 186
Using the Grid ... 187
To display or hide the grid ... 188
To set grid spacing ... 188
To enable or disable snap-to-grid ... 188
Viewing Expanded Expressions ... 189
Working with Color ... 190
Using the Color Selector ... 191
To change color fill properties for an existing view element ... 192
To change color for text, line, view element fill, or view element outline color ... 193
To save a color ... 193
About Solution-Defined Colors ... 193
Customizing Preferences ... 194
clientsettings.properties ... 194
globaleditingsettings.properties ... 194
Generating a View Details Report ... 195
To generate a View Details report ... 196
To print the View Details report ... 196
Using View Search ... 197
To conduct a view search ... 197

6 | Working With View Building Tools ... 198

Drawing Lines ... 200
To draw a straight line ... 200
To shorten or extend an existing line ... 201
To add arrowheads to a line ... 201
To change the line color ... 201
Creating Irregular and Open Shapes with the Polyline Tool ... 202
To draw irregular open or closed shapes ... 202
To add arrowheads to a polyline ... 203
To modify an existing irregular shape ... 203
To change the fill or line color for an irregular shape ... 203
Using the Rectangle and the Ellipse Tool ... 204

To draw a rectangle or square 204
To draw an ellipse or a circle 204
To change the fill or line color for a rectangle or ellipse 204
To change the line width for a rectangle or ellipse 204
Adding Text 206
To add text into a view 206
To change the font size, font style, and font color of text in a drawing 206
To move text in a drawing 207
To horizontally justify text 207
Inserting Images 208
To insert an image 208
Using the Value Tool 209
To add a Value item into a view 210
To create a background around the value item 211
To change the primitive member for a value item 211
Setting Date, Time, Quality, and Unit 212
About Dates 212
About Time 213
About Quality 214
About Units of Measure 215
To set-up unit conversion for a value expression with multiple references 215
Inserting Views 221
To insert a view into a view 221
Pasting a view within the same view 221
To copy a view and paste within the same view 222
To cut a view and paste within the same view 222
To duplicate a view and paste within the same view 222
Using the XHQ View Instance 223
To configure a view instance 224
Inserting a Keyed View 226
To insert a keyed view 226
Using Templates 227
To map template component members 227
EXAMPLE: Using Templates 227
EXAMPLE: Exporting and Importing Templates Using .view Files 230
Using Containers 232
Configuring a Multi View Container 232
Selecting a Different View Context 235

Configuring a Tab Container 235
Custom Tabs 238
Adding Controls 242
Using Local Variables 242
Using the Edit Box 242
To add and configure the edit box control 243
Configuring the Date/Time Picker 243
The Date/Time Picker at Runtime 245
Customizing the Date/Time Picker 245
To change the default icon 246
To show the week number column 246
To change the picker color scheme 247
Using the Combo Box 249
To configure the combo box control 250
To edit the Value List 251
To set the Collection Data 251
Multiple Selection Combo Box 252
What multiple selection looks like at runtime 254
Using the List Control 255
To configure the list control 259
Using the Button Control 260
To add and configure a button control 260
Using the Text Area Control 261
To add and configure a text area control 261
Configuring the Collection Filter 262
About Operands 262
About Criteria Relations 262
Collection View Layout 263
Executing the Collection Query 264
Configuring an Embedded Trend 265
Trend Properties: Concepts 266
About the Embedded Trend Chart Grid 268
About Start Time, End Time, and Duration/Span 268
About Scale Properties 269
About the Logarithmic Trend Scale 270
Using Dual Scale 270
About the Aggregate Interval 271
Trend Properties: How To Set 272

To set or edit a trend property 276
Pen Attributes: Concepts 277
About Pen Tolerance 278
About Scale Format and Precision 278
About Value Format and Precision 278
About Aggregate Types (History Retrieval) 280
280
About Pen Groups 281
Pen Attributes: In Time Mode 283
To set the Mode trend property to Time 285
To add or edit pen attributes in time mode 286
Pen Attributes: In XY-plot Mode 288
Pen Groups and the XY-plot Mode 291
To set the Mode trend property to XY-plot 291
To add or edit pen attributes in XY-plot mode 292
Editing and Deleting Pen Attributes 294
Using Expressions in the Embedded Trender 294
Valid Pen Expressions 295
About Global Tag References 297
About Late-binding Global Tag References 298
Processing a String Pen Expression as a Numeric Equation 298
Support for Late-binding in an Embedded Trend 299
Examples of Embedded Trends in Runtime 300
Embedding Browser Content 302
To insert a web browser control 303
Invoking the JavaScript Function 303
Removing the Browser Control Borders and Scrollbars 303
Using XHQ Advanced Charts 305
Chart Features 306
The Chart Tab 310
Configuring a Chart 312
To add a series 312
To copy a series 319
To set chart options 319
To define the group settings 326
About the Series Table 328
Series Types 332
Setting Criteria 333

Grouping 334
Group By for Charts 337
To configure group by 337
Drill-down Expressions 337
Renderers 338
Rendering Limits 340
Using Multiple Range Axes 340
Setting the Range Axis Location 341
About Axis Minimum and Maximum Values 342
For the Domain Axis 342
For the Range Axis 342
Annotations 343
Annotation Types 343
Adding Annotations 344
Series Color and Custom Colors 346
Label Font and Color 347
Tooltips and Custom Tooltips 350
Tag Expression 351
Tag Mapping 352
Configuring the Tooltip 353
Embedding Charts in Collection Views 355
Time Offset Types 355
Setting the Tick Unit 358
Setting the Units of Measure 361
Viewing the Chart at Runtime 362
About the Domain Crosshair 364
Using Expressions in Chart Configurations 365
Color Expression 366
Series Mapping Column expression 368
URL Link Address and Action Link Function expressions 368
Date Format 369
Future Date Support for the Time Series Chart 369
Drill-down Series 370
Viewing the Drill-down at Runtime 370
Configuring a Drill-down 371
To configure a drill-down 371
About the Treemap Chart 374
How to use Treemaps 375

Default Treemap Navigation 375
About 3D Charts 376
The Plots 376
Setting Chart Options for 3D Charts 377
Defining Chart Axes 378
Using the Zoom Toolbar at Runtime 381
Inserting the Table Control 385
Configuring the Table Control 385
To configure a table control 391
Setting Table Criteria 401
Using Expressions in Table Control Configurations 402
About the Row Select Function 402
Links Configuration 402
Group by for Table Control 403
To configure group by (design time) 403
Configuring Support Links and Tooltips for Group by Table Controls 404
To configure support links and tooltips for group by table controls 405
Using an ENO Group as the Data Source 408
To configure an ENO table 410
Inserting an Excel Item 413
Configuring the Excel Control 413
To configure the Excel control 417
Progress Indicator for Exporting to Excel 423
Inserting the GISMap Control 424
Provider Pre-requisites 424
About Layers 425
About Markers 426
Mappings 430
Configuring the GISMap Control 431
To use the Google Map provider 431
To use the Bing provider 433
To use the Shapefile provider 433
To use the Image provider 433
To set the GISMap control properties 433
To add and configure layers 435
To add and configure markers 438

7 | Building Views 444
Using Links 444

Defining "Parents," "Children," and "Grandchildren" 444
To attach a link to any view element or group of view elements 444
Using Expressions to Link to a URL or Another View 446
Examples of Relative URLs 446
Setting the View Link Mode 446
Using Action Link Functions 447
Working with Client-side Expressions 449
Client-side Expressions and Secure Data 450
Using Animation 451
Display Characteristics - Animations 452
Animation Types 454
Setting Animation Properties 456
To animate a view element 456
To set a Value Tag animation 457
Animating with Color 458
Animation Examples 458
Precision Animation 460
Using Group By 463
Sort and Filter 463
Truncate and Extract 463
Pasting a Copied View and Re-linking 464
About Substitute References 464
To paste a view from another component and re-link 464
More About Pop-up Views 466
Setting View Link Pop-ups 466
To set a pop-up view using View links 467
Setting Action Link Pop-ups 468
To set a pop-up view using Action link ShowView function 469
Closing Pop-up Views 470
Working with Embedded Views 471
The Views Tab 471
To add view flavors 472
Setting Views Properties 473
About Drillable Views 474
To make a view drillable or non-drillable 474
Setting a non-drillable, embedded view 475
Overriding the Views Properties - Drillable setting 476
To override the Views Properties setting 477

About Views on the Browser Menu 478
To include a view on the shortcut menu list 479
Overriding the Browser Menu Setting 480
Setting the Default View 481
To set the default view through Views Properties 481
Setting Pop-up Views from Views Properties 482
To set a pop-up view through Views Properties 482
Viewing Pop-ups at Runtime 482
Embedding Descendant Views 482
Working with the XHQ Visual Tile Composer 485
Step 1: Building/Importing Libraries 486
To enable the Shapes and Viewlets tabs 486
Building a Library of Shapes 487
To add a library 487
Working with Shapes 488
To create a shape 488
Working with Viewlets 489
To open a viewlet 490
Working with Tiles 490
To resize the tile 491
Step 2: Mapping Shapes 491
About VTC Security 494
About Published Composed Views (CVs) 496
Tutorial: Building Views 499
To create a view 499
To create a process overview for the Hydrocracker unit 499
To create a view for the tag component 500

8 | Working with Collections and Flow Views 504
Creating a Collection 504
To create a collection 505
Working with Collections 506
About Flag Collections 507
To rename a collection 508
To delete a collection 508
Ordering of Collections 509
Using NOT-NULL to Improve Performance 509
To improve performance using NOT-NULL 509
Configuring a Global Collection 510

Working in Detail View 510
To configure a collection 511
Linking a Component to a Global Collection 513
To link a component to a global collection 514
Exposing Collection Records in the XHQ Solution Tree 515
Step 1 - Component Configuration 515
Step 2 - Collection Member Configuration 516
Step 3 - Solution Configuration 516
Using the Query Builder 518
To use the query builder 518
About Flow Views 519
Building Views Against Collections 520
View-Based Collection Filtering 523
Determining Your Collection Filter Criteria 523
Configuring the Collection Filter 523
About Operands 524
About Relations 524
Collection View Layout 525
Executing the Collection Query 526

9 | Working with Tags 527
How Tags Are Configured 527
About Tag Collections 529
To create a tag collection 531
Setting Subscribable Members 532
Tags and Views 533
Dealing with Multiple Data Types 533
About Keyed Views 534
To insert a keyed view 535
Working with Composed Tag Views 536
Using the Tag Pickers 537
The Tag Configuration Viewer 537
To launch the Tag Configuration Viewer from the XHQ Solution Builder 538
About the Tag Configuration Viewer Dialog Box 539
The Search Panel 539
To enter (or edit) search text 540
The Results Section 541
Action Buttons 541
The Configure Tag Dialog Box 542

The Tag Selector ... 543
About Value Tag Animations ... 544
Launching the Tag Selector from the Trend Viewer ... 544
Launching the Tag Selector from the XHQ Workbench ... 545
To launch the Tag Selector from the XHQ Workbench (for Keyed Views) ... 545
To launch the Tag Selector from the XHQ Workbench (for Value Tag Animation) ... 545
About the Tag Selector Dialog Box ... 546
The Search Panel ... 546
To enter (or edit) search text ... 547
The Results Section ... 547

10 | The XHQ Solution Builder ... 548

Touring the XHQ Solution Builder Environment ... 548
To launch the XHQ Solution Builder ... 549
XHQ Solution Builder Menu Commands ... 549
The Solution Menu Commands ... 549
The Configure Menu Commands ... 550
The Help Menu Commands ... 550
Using the Navigation Panel ... 551
To select a node or a component in the navigation panel ... 551
To expand or collapse a node in the navigation panel ... 552
To resize the Navigation panel ... 552
About The Detail View ... 552
Accessing Views ... 553
To work in either navigation or detail view ... 553
About Tooltips ... 554
About the Tag Tooltip ... 554
About the Value Tooltip ... 554
Saving Views ... 556
To save a view from the XHQ Solution Builder ... 556
Accessing Trends ... 557
To verify an item's connection and history ... 557
To select a value item and launch the XHQ Trend Viewer ... 557
Accessing Solution Servers ... 558

11 | Building and Configuring Solutions ... 559

Using Detail View Tabs ... 559
The General Tab ... 564
Using Aliases ... 565

To assign an alias name to a primitive member 565
The Connect Tab 566
Using the Simulator Connection Group 567
Selecting the Query Type for Data Member Configuration 568
About Time-Series 569
About the Quality Column 570
About the Poll Scheduling Option 571
About Retrieval Methods 573
About the Flag (Refresh) Query 574
About Delta Updates 577
About Pass-Through Queries 580
Using the Query Builder 582
To use the query builder 583
The Expression Tab 585
About Expressions 585
Expressions with Primitive Members 586
The Server-side Expression Scheduler 586
How the Scheduler Works 587
Adjusting for Daylight Saving Time 588
To access and set the Expression Scheduler 589
The History Tab 590
About the XHQ Data Recorder 590
Record Marker Mode 591
Storing Data Streams 592
To modify the default value of 13 years 593
Retrieving Data Streams 593
Log Files 593
Configuration Properties 593
Using the XHQ Data Recorder 595
To store real-time values for an XHQ primitive 595
Listing Third-party Historians 596
To display a third-party historian in the Recorder drop-down 596
History Tab vs Connect Tab: Displaying the Trend 597
The Cache Tab 598
About the Unique key 598
About the Index 599
About Multiple Indexes 600
To set the index(es) for a collection 600

The Link Tab 601
Creating a Proxy 602
Creating a Global Collection 605
To create a new global collection from the Link tab 606
The Access Tab 607
The View Access Tab 608
The Collection Health Tab 609
To monitor the collection health 609
Displaying Real-time Values 611
Group Members in the XHQ Solution Builder 612
Inventory Items and Child Objects 614
Historical Viewing and Navigation 615
Historical Viewing and JavaScript Functions 619
URL Parameters Support for Historical Viewing 619

12 | Exporting and Importing 620
Model Export and Import 620
Exporting Components 620
About the Exported ZIP File 621
To export components 621
Importing Components 623
About Component Subclassing and Importing Your Model 623
To import components 624
Copying Repositories 626
XHQ Model Export: List of supported XML Elements 627
Importing the Model Using XHQ Web Services 632
Solution Export and Import 633
Exporting a Solution Configuration 633
To export a solution configuration 633
Importing a Solution Configuration 635
To import a solution configuration 635
XHQ Solution Documents 636
Configuring an XML File for Import 636
XHQ Solution Export: List of supported XML Elements 636
View Export and Import 646
To export a view in XHQ Workbench 646
To import a view into XHQ Workbench 647
Exporting a view containing a GISMap control 647
Exporting and Importing XHQ Animations (XANIs) 647

To export a XANI 648
To import a XANI 649
To re-bind individual elements in an imported XANI 649
Selecting Configurations to Export or Import 650
Role Mapping Configuration Import/Export 650
To export an XHQ security (mapped roles) configuration 650
To import an XHQ security configuration 652
XHQ Role Mapping Export: List of supported XML Elements 652
Solution Definition Import/Export 655
To export a solution definition 655
To import a solution definition 657
XHQ Solution Definition Export: List of supported XML Elements 657
Connector Definition Import/Export 659
XHQ Connector Definition Export: List of supported XML Elements 659
Connector Process Import/Export 661
XHQ Connector Process Export: List of supported XML Elements 661
Connector Group Import/Export 663
XHQ Connector Group Export: List of supported XML Elements 663
Import/Export Delete 665
XML Supported Delete Syntax 665
Import and Export Use of the XML Delete Syntax 669
Using the Command-line to Import/Export 670
Shapes, Viewlets, Mappings Export and Import 671
To export shapes 671
To import shapes to a library 671
To export viewlets 672
To import viewlets 672
To export mappings 673
To import mappings 673

Appendices 674
A - Reserved Words 675
Reserved Symbols 678
B - Supported Characters 679
C - Application Techniques 681
Improving Collection Queries 681
Querying GOOD Criteria 681
Using the Collection Data Combo Box 681
Using a Collection Aggregate Method 682

D - DTD Validation Structure 683
The Solution DTD 683
The Model DTD 685
Component DTD (used in the "model.xml") validator 687
DTD validator for the XHQ Views 689
Role Mapping DTD 695
Solution Definition DTD 696
Connector Definition DTD 696
Connector Process DTD 696
Connector Group DTD 697
Units of Measure DTD 697
DTD for a Tag Sync XML File Import 698
E - XML: Preserving Collection Links 699
F - Chart Mapping and Data Sets 707
For the Bar Chart 709
Categorical Data 709
Numerical Data 710
For the Bubble Chart 711
For the Candlestick Chart 712
For the Gantt Chart 714
For the Histogram Chart 715
For the Pareto Chart 717
For the Pie Chart 718
For the Polar Chart 719
For the Spider Web Chart 720
For the Time Series Chart 722
For the Treemap Chart 724
For the Waterfall Chart 726
For the XY Chart 727
G - Shortcut Keys 730

About This Guide

Conventions Used in This Guide

The following formatting cues are designed to allow you to quickly locate and understand the information provided in this guide.

Formatting Conventions

Convention	Example
Acronyms are spelled out the first time they appear.	Alert Notification System (ANS)
Bold is used for menu names, command options, and dialog box names in primary task procedures.	From the **XHQ Workbench**, go to the **Add** menu and click **New Component**.
Italic is used for glossary terms.	The first step in building this model is to develop reusable software building blocks, called *components*.
A **`monospaced font`** is used for program and code examples.	The subdirectory `\log` is automatically created below the location you choose. All log files are written to this subdirectory. **`C:\XHQ`**
Key combinations appear in uppercase, bold. If joined with a plus sign (+), press and hold the first key while you press the remaining keys.	**CTRL+B**
In See Also notices, sub-chapter headings are in italics, chapter headings are in quotes, and guide titles are in bold.	For more information, go to the *About install.properties* topic located in the "Working with PROPERTIES Files" chapter of the **XHQ Administrator's Guide**.

Visual Cues for Online Viewing

This document uses the following styled paragraphs.

Important notices provide information that are required to completing a given task.

XHQ must run as a domain user.

Warnings tell you that failure to take or avoid a certain action could result in loss of data or application malfunction.

WARNING

Do not modify the `shutdown.dat` template file.

Notes are used to offer information that supplement important points of the main text. Tips suggest certain techniques and procedures that may help you achieve your task quickly.

Depending on your network configuration, include domain information only if the domains are different.

See Also notices provide you with additional references to similar topics and/or concepts within the documentation set. Sub-chapter headings are in italics, chapter headings are in quotes, and guide titles are in bold.

For more information, go to the About the Options Menu topic located in the "Working with PROPERTIES Files" chapter of the **XHQ Administrator's Guide**.

Tips provide additional hints to help you use the product more efficiently.

Use the `NavbarWestVerticalOffset` property to make fine adjustments in pixels. The upper, left-hand corner is the origin. The positive horizontal direction moves to the right and the positive vertical direction moves down.

Web References point you to external web sites that give additional information on the given topic.

Refer to Microsoft support information with regards to the various server settings for application performance and network utilization.

`http://support.microsoft.com`

Related XHQ Product Documentation

The XHQ documentation set includes the following titles.

XHQ Documentation Set

Title	Target Audience
XHQ Administrator's Guide Provides the steps required to begin administering XHQ. It also covers security and access, property settings, redundancy, and localization.	Administrators
XHQ ANS User's Guide Learn how to use and administer the XHQ Alert Notification System (XHQ ANS).	ANS Users, Administrators
XHQ Backup and Recovery Guide Learn how to properly backup XHQ.	Administrators
XHQ Connection Guide Provides information on injecting an XHQ-supported connector type and configuring the connection.	Connector Developers
XHQ Developer's Guide Introduces the XHQ Development Client (Workbench and Solution Builder) user interface and provides information on how to set-up XHQ, develop reusable components, create views, and build a solution hierarchy.	Content and Solution Developers
XHQ Getting Started Gives you step-by-step instruction on how to set up your model and solution.	Content, Connector, and Solution Developers
XHQ Installation Guide Provides the system requirements, installation instructions, and upgrade information for the current release of the XHQ System.	Administrators
XHQ Integrated Data Gateway Guide Includes information on the ADO.NET and the XHQ OPC UA Server.	Application Engineers, Integrators
XHQ Performance Analytics Guide Learn how to use the Engineering Environment to enable the generation of the processes necessary to extract and transform data for source systems, and populate the XHQ Data Store and Data Mart.	Solution Developers/Users, Analysts
XHQ Performance Management Guide Learn how to use Target Management to monitor performance indicators and eLogs to create shift reports.	Administrators, End Users
XHQ Reference Guide Lists the functions and methods used in XHQ, and provides examples,	Content and Solution Developers

Title	Target Audience
usage notes, and parameter descriptions.	
XHQ Reporting Services Guide Introduces the XHQ Reporting Services and provides instruction on how to connect to an XHQ data source.	Application Engineers, End Users
XHQ SDK Reference Guide Provides a set of development tools that allows you to create applications that extend XHQ. Includes information on the Client API and Web Services.	Application Engineers, Integrators
XHQ Solution Design and Architecture Provides best-practice examples for XHQ solution design. Includes information on tag synchronization.	Solution Architects
XHQ Solution Viewer User's Guide Gives you step-by-step instruction on how to access your solution through a browser client and set browser preferences.	All End Users
XHQ System Guide Contains information regarding secure handling of an XHQ implementation.	Administrators, Application Engineers, Integrators
XHQ Trend Viewer User's Guide Learn how to use the XHQ Trend Viewer to view both real-time and historical data.	All End Users

Contacting Customer Support

XHQ Customer Support is a second-level customer support offering, that is, it does not provide XHQ end users with direct support. XHQ end users are to contact their local company help desk or internal application support staff and, in turn, those representatives contact the XHQ Customer Support Team. These representatives are expected to have attended basic product administrative training or possess comparable skills with XHQ, and know and support the specific XHQ customer solution in use.

If the details or response times noted below deviate from those specified in a specific customer contract, the customer contract always takes precedence.

For general XHQ product support or related questions, pre-registered customer or partner support staff with a valid XHQ customer support agreement may contact the XHQ Customer Support Team using any of the following means:

Web Portal

The support portal leverages a system called GTAC (Global Technical Access Center). GTAC provides one common support entry point for many Siemens products. It is available via this URL:

https://www.siemens.com/gtac

Customers must be pre-registered to be able to use the web portal. A log-in can be requested at any time by self-registering in the GTAC portal. Note, the end-user "sold to" identifier is needed in order to register.

Use of the support portal is the preferred means to report incidents to the XHQ Customer Support Team unless immediate interactive telephone assistance is required. The support portal is available twenty four hours per day/seven days per week ("24/7").

E-mail

support.xhq@siemens.com

Phone Support and Hours of Coverage

International: +1 (949) 448-7463

U.S. only: +1 (877) 700-4639

The following paid support levels are available:

Bronze Support: 9/5

9 x 5 hours support. 9 hours per day, 5 days per week, Monday to Friday. Daylight Saving Time is honored.

Choice of one coverage zone out of the following options (the default is Americas):

- Americas (7 am - 6 pm PST; 11 hours coverage due to PST/CST/EST time zone coverage overlap)
- South Central Asia (9:30 am - 6 pm IST; 9 hours coverage)

Excludes national holidays as defined by the following countries for the related coverage zone:

- USA/California (Americas)
- India/Pune (South Central Asia)

Example Americas zone: *Implies coverage from 7:00 AM to 6:00 PM, Pacific Time, Monday to Friday, excluding US national holidays.*

Silver Support

Ability to leverage both support coverage zones **Americas** and **South Central Asia** as defined in Bronze for extended daily coverage hours.

The weekly start/end times of coverage follow the local times of the following coverage zone:

- California/USA (Americas)

This implies weekday coverage from 7 am until 6 pm Pacific Time, Monday to Friday, as in the Americas support coverage zone but with the ability to additionally leverage the South-central Asia coverage zone for additional coverage hours.

Gold Support: 24/7

Silver Support coverage plus 24 hours per day, 7 days per week emergency support for Severity One incidents.

Postal Mail

Siemens Industry Software Inc.

XHQ Operations Intelligence

Attn: XHQ Customer Support Department

6 Journey, Suite 200

Aliso Viejo, CA 92656, USA

General Feedback and Comments

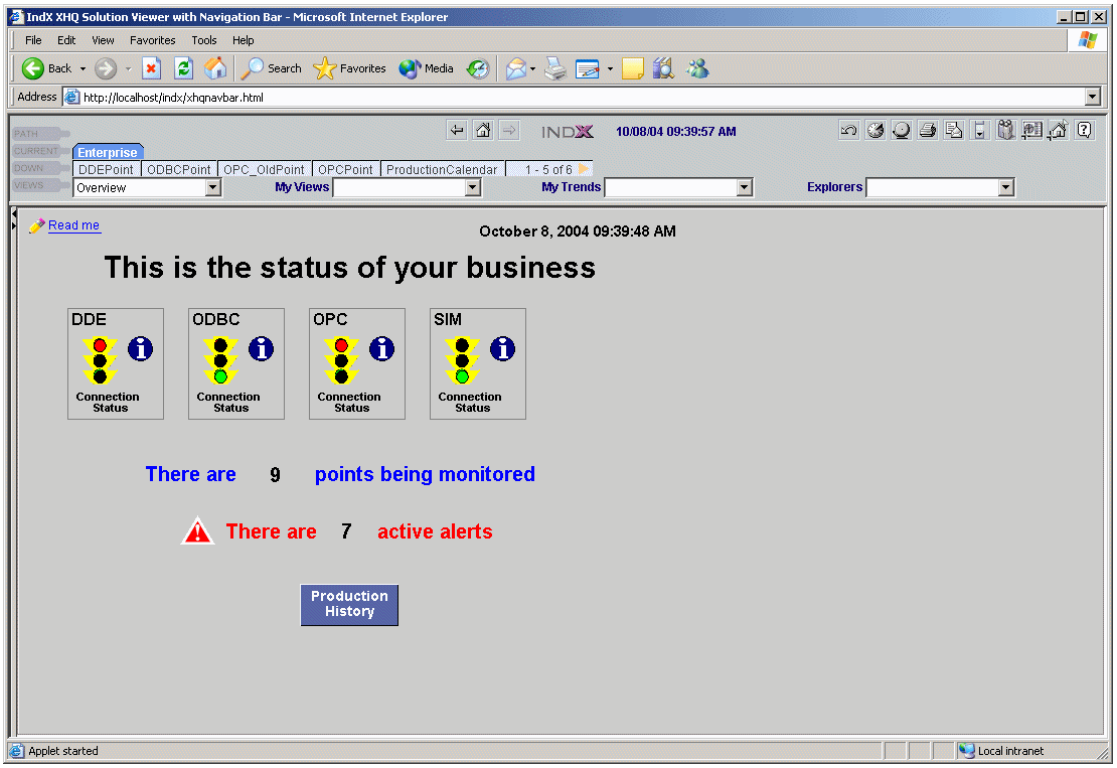
Please send an e-mail to:

info.xhq@siemens.com

Siemens Industry Software Inc. and affiliated Siemens Industry Software companies (collectively referred to as “SISW”) are committed to working with our customers. Your comments, suggestions, and ideas for improvements are very important to us. Thank you for taking the time to send us your feedback.

XHQ Overview

The XHQ System gives you everything you need to connect to backend sources, build visually compelling representations of data, and see the information in a browser. With the XHQ System, data comes to life on your desktop, in the form of animated text, tables, and charts that represent your enterprise. The XHQ System makes it easy to locate and view information from multiple databases and systems, and build specialized views that meet different users' needs. And, unlike other enterprise management systems, the graphical representations in the XHQ System update live as backend data changes.



XHQ Solution Viewer with the XHQ Navigation Bar

The XHQ Operations Intelligence (OI) Platform provides large industrial manufacturers with: a common view of all critical business and operational data from disparate sources throughout the enterprise, enabling plant floor and operations personnel, as well as senior management to monitor real-time performance against business goals and make timely, better informed decisions about plant operations that positively impact the bottom line.

6 Steps to Building a Visual Enterprise

To develop an XHQ application, follow these six steps:

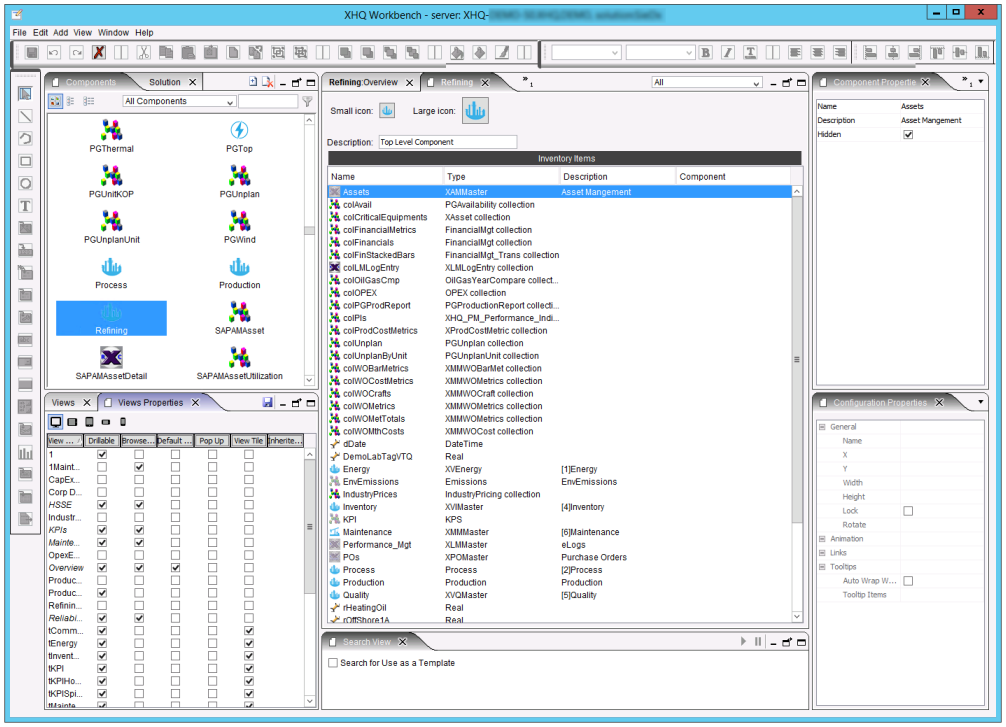
Step 1	Develop an outline of your information model. You need to know what information is available from all the backend systems. You also need to know what information users require, as well as the goals of the business.
Step 2	Develop reusable information components. These are the components that form the building blocks of your final application. They are stored in a catalog for easy access and deployment.
Step 3	Build the model or solution hierarchy. The model holds informational components in a logical relationship. It is easy for casual users to find any information because the model is presented using a navigation tree. Data from the backend systems map into the component instances that show in the navigation tree.
Step 4	Configure the solution, including each instance of the components you used. For each instance of a component, you enter general properties (name, description, etc.), declare connector configuration parameters (or SQL queries, etc.), and (if required) enter a strategy for retrieving and/or storing real-time and time-series values.
Step 5	Develop different views of the information in the components. These views will present the information in different ways, perhaps according to the functional roles of users interacting with the system.
Step 6	Deploy and use a browser client to access the solution. End-users see XHQ primarily as a viewing tool. They are more interested in the information they can see rather than the configuration of XHQ.

Step 1 - Develop Your Information Model

The first step in building an XHQ system is to develop an informational model organized to reflect your business and to present your business information in ways that are meaningful and easy to access.

Step 2 - Develop Reusable Components

Once you have modeled your business, the XHQ development software enables you to construct an intuitive, *object-oriented* information model. The first step in building this model is to develop reusable software building blocks, called *components*. Essentially, a component provides a mechanism to collect and combine data from diverse backend sources. Components logically group the various sets of information that you need to create different views. When you save a component, it is automatically stored in a *catalog* that allows it to be reused many times by many users in many solutions.

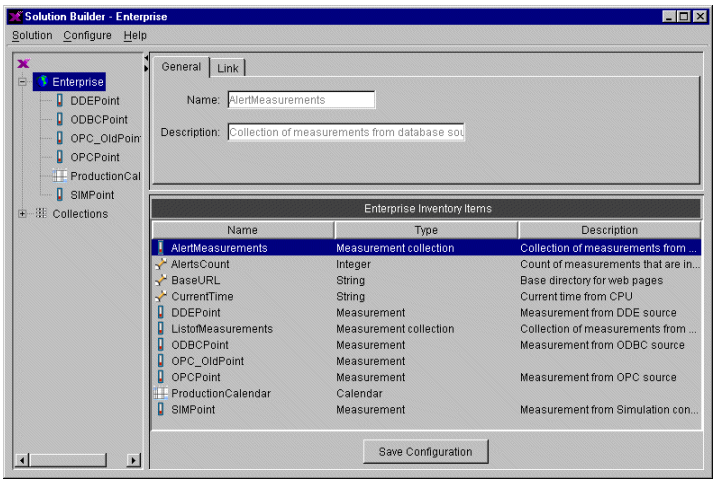


XHQ Workbench

The *catalog* refers to the collection of components and views in a solution. The tool used to develop the components and views is the XHQ Workbench.

Step 3 - Build the Solution

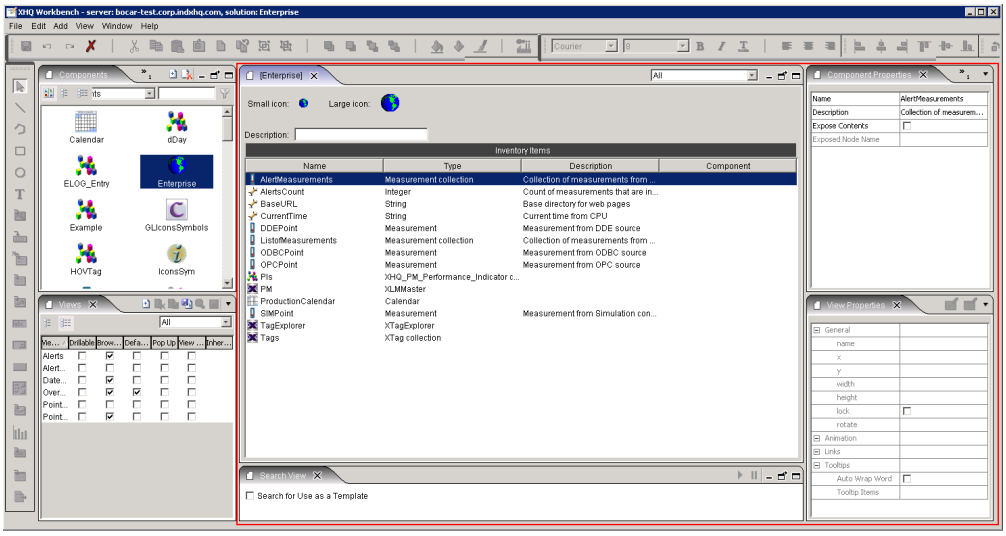
After you have built your library, or catalog, of reusable building blocks, the next step is to take these components and reuse them in order to complete your solution. When you reuse a component, you create a unique *instance* of that component, with a unique name. Different instances of the same component can differ in many ways. For example, they can follow a different set of instructions that specify where, when, how, and how often to retrieve data.



XHQ Solution Builder

Step 4 - Configure Components

After the solution is deployed, XHQ displays your model in a tree-style navigational structure, or graphical directory, that you use to easily locate the information you need.

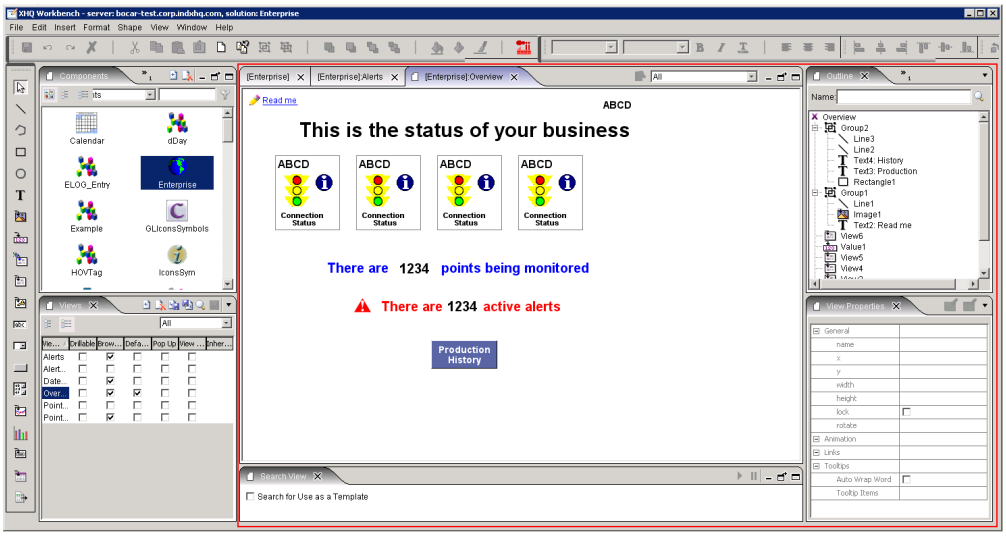


Component Details tab

At this point, all the groundwork users need to build views is done. The information that you want to present is already organized into logical components that are ready to collect data from your backend systems. View builders do not have to be concerned about where the data is coming from or how it is collected.

Step 5 - Build Views

The next step is to build views. Of course, different users want to look at information differently. Some require in-depth detail; some demand summary-style information. Some prefer an animated chart; while others prefer a table.

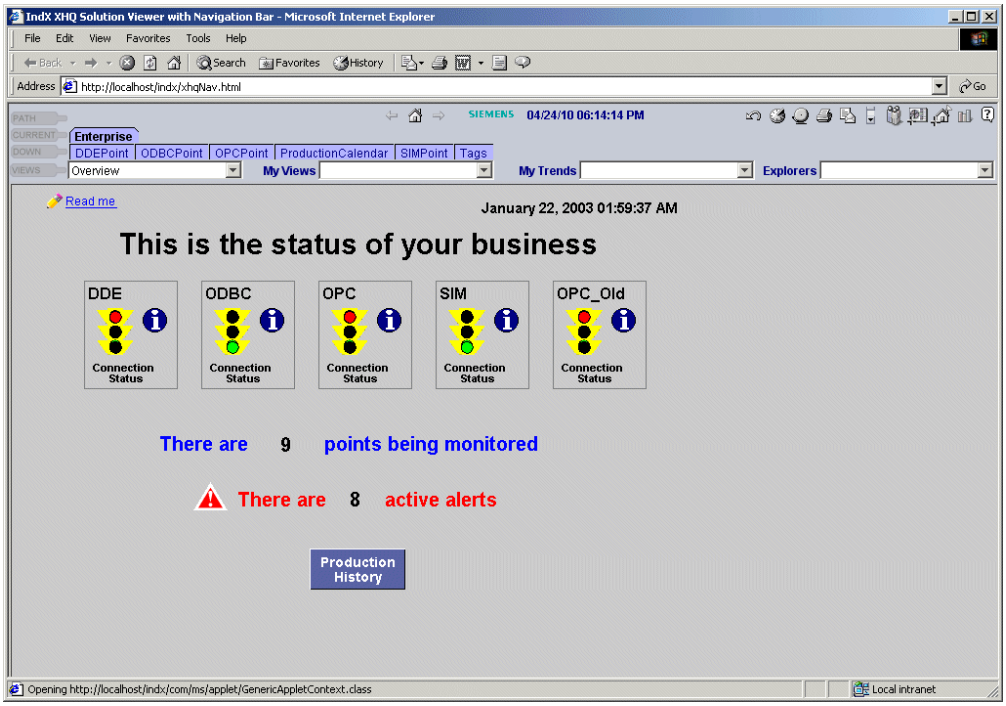


XHQ Workbench

With XHQ, you can create views for the specific needs of every type of user. For example, you could have a planner's view, a maintenance view, and a CEO's view. The XHQ drawing environment lets you build anything from simple tables and reports all the way to complex views with attention-grabbing graphics and animations driven by data from diverse sources.

Step 6 - Deploy and Access the Solution

End-users see XHQ primarily as a viewing tool. They do not see the protocols or queries that support these connections. XHQ thus hides the complexity of backend systems and connections from the end-users of the product, and lets them focus on the use of information rather than the complexities of the configuration.



XHQ Solution Viewer

About Security

In a **new installation** on a **new server** where no existing customer solution (repository) resides, XHQ does not have security enabled for the out-of-the-box default solution. This allows the administrator initial access to XHQ to configure an initial solution and activate an initial security configuration that maps customer-specific active directory groups to default XHQ roles without having to do this at install time.

It is strongly recommended you activate this initial security configuration on the master development server before promoting the XHQ content to the production system (that is, before placing the newly defined XHQ solution on any production network accessible to general end-users) to avoid any user being able to access the solution even if there is no customer data yet available as a general security-related precaution.

After XHQ is initially configured and secured, future updates do not change the security configuration.

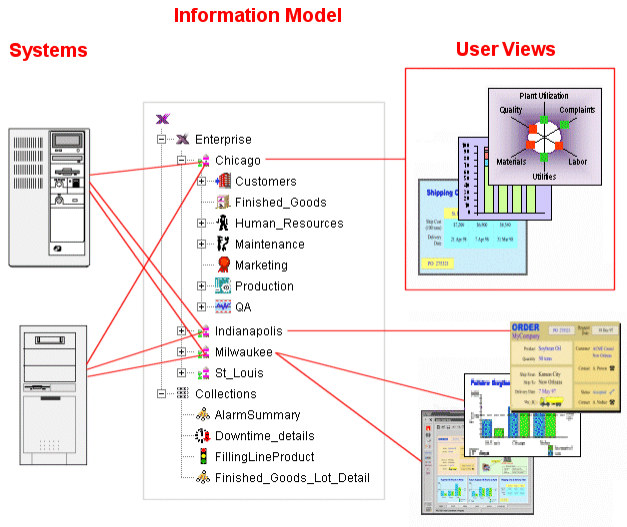
1 | Developing Your Information Model

The first step to building an XHQ System is to develop an informational model that is organized to reflect your business and to present your business information in ways that are meaningful and easy to access.

What is an Information Model?

An informational model is an organization and graphical notation suitable for describing the vocabulary and conceptualization of a problem domain.

A visualization system model allows you to concentrate on the intrinsic purpose of the data rather than its origins and display peculiarities. The model represents a deep understanding of the problem domain and the available information, along with the practical ways in which you can assimilate and use it. In an XHQ application, a model is a hierarchical representation of an organizational structure that holds information where people would logically look for it. The information model with its rich graphics exposes the contained information in very natural, intuitive ways.

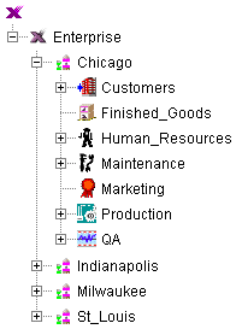


XHQ Information Model Example

An XHQ model delivers information while keeping systems synchronized. Changes made to a model are automatically propagated; everybody sees them immediately. The model assures that information is prepared in well thought out ways to meet the needs of customers. A proper model allows you to extend and renew the information system as backend systems change, or as new features are added.

The XHQ model should not to be confused with the many other types of computer models used by other applications. (For example, UML models for developing computer programs and mathematical models for determining heat and material balance.)

An XHQ information model is simple to represent and construct. It is based on a *containment* metaphor. The top component contains the components underneath, and they, in turn, contain components beneath them. A model should classify components and arrange them logically; components at the same level should be of a similar type or related in terms of business information importance. At any level in the hierarchy, the appropriate data is exposed through the components at that level. Overall, the model should target the casual user and use broad, understandable terms.



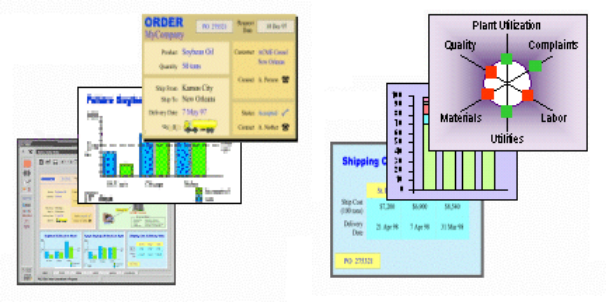
Example of Containment Metaphor

In an information model, components represent real-world business elements. Examples are tangible assets such as plants, equipment, inventory, and production lines, and also less-tangibles such as orders, schedules and work in progress lots. While representing real world elements is useful, it is the incorporation of data (from multiple sources) into the components of that model that facilitates understanding of information.

With an XHQ model, you do not have to be an analyst to interpret information and make decisions. And because components built using XHQ actually contain information from many systems, this model frees users to concentrate on the intrinsic purpose of the data rather than on its origins, access mechanisms, and display peculiarities.

Why build an Information Model?

Managing the business of a corporate enterprise involves combining operational information from many disparate and incompatible systems. Up-to-the-minute information must be prepared for instant use without compelling users to be intensely analytical. Currently, complexity is rampant. Corporations cannot train everyone on every system, and even if they could, people cannot retain the irregularities in finding and displaying pertinent information. More people besides analysts need to be able to interpret information and make decisions.



Organizing Views and Information

There are four main reasons for building an information model for XHQ.

1. An XHQ solution based on a well thought-out information model will make it easy for people to locate and understand information. It will provide a navigation structure that organizes the views and information.

2. An XHQ solution based on a well thought-out information model will make it easy to change and renew the information provided to the users as backend systems change or when new systems come on line.
3. An XHQ solution based on a well thought-out information model will insulate users from the complexity of backend systems and, in turn, protects the backend systems from users.
4. An XHQ solution based on a well thought-out information model will enjoy more reuse of components and views.

In backend systems, data are arranged in functional ways to support the transactions to be applied. Casual users are not familiar with the computer constructs in backend systems that the IT personnel or analytical specialists might understand.

Using a well-designed information model should shield users from technicalities such as queries and database schema. When you build your XHQ solution based on a good information model, the elements you see on the screen should closely resemble assets and activities you see in your enterprise.

Designing Your Information Model

Within an enterprise, similar things are identified and abstracted as informational components. A fundamental purpose is to facilitate re-usability. But, it is the use of the components in a representation of a particular problem domain that is the essence of the informational model. An analysis and modeling exercise is imperative to identify the data available in backend systems and match them with the views needed by end-users. Analysis and modeling is a way to translate those data constructs into tangible real entities that people see and recognize.

For an informational model, you can divide the domain into components that fall into a number of categories:

- Tangible assets such as physical equipment, plants, work centers, vessels, reactors, canning lines, presses, and assembly lines.
- Roles such as CEO, CFO, planner, supervisor, operator, and maintenance.
- Incidents such as work start, emergency interrupt, and batch end.
- Interactions such as order entry, stock level increment, and so forth.

The model you design should be based on the varying needs and perspectives of your users. It will also reflect the application within the particular business or industry.

Model Examples

Models can be made in three extreme varieties:

- **Asset model**
  Here the model is a strict representation of the layout of the physical assets. For example the top level component would be the plant, the next level down would be the four production lines, each production line would contain the various pieces of production equipment that are relevant. Such a model is very useful in the process industry, where equipment fluids in pipes are transformed in various processes to make product.

- Chicago Plant
  - Filling Line
    - Depalletizer
    - Filler
    - Packer
  - Raw Material Storage
    - Container Stock
    - Syrup Tanks
  - Warehouse
  - Water Treatment

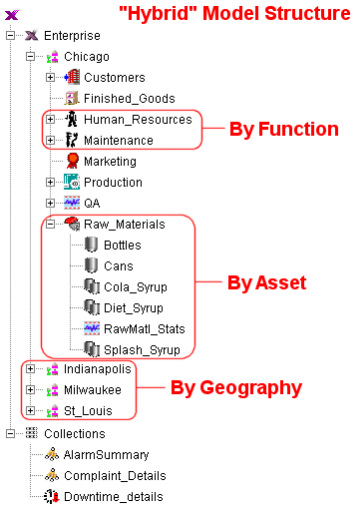
- **Activities model**
  Here the model is a representation if the various tasks that are organized to produce a particular product. Such a model can describe the operations in a motorcar assembly plant and other discrete types of manufacturing.

  - Chicago Plant
    - Finished Goods
    - Procurement
    - Production
    - Shipping

- **Organizational model**
  Here the model represents activities that are very people-intensive. Such a model could be used to describe the activities in an order processing department or the management of project activities.

  - Chicago Plant
    - Maintenance
    - Marketing
    - Production
    - Purchasing
    - Sales
    - Shipping & Receiving

Your finished model will likely be a combination of various model types, but one of the three will represent the essence of the model, and that will enable people in that particular business sector to immediately recognize and understand the information content.



Example of Hybrid Model Structure

Typically, you are concerned only with information that describes certain characteristics of the components. These are the general characteristics such as name, description, and units. However, in any information component, it may contain other information of interest to viewers such as documents that are logically associated with the component. When examining a component, a user may wish to see active maintenance work orders, safety data sheets, plans/schedules, quality results, CAD drawings, design and performance documents.

Things to Consider

The information model you design should allow you to maintain, extend, and renew the XHQ information system as backend systems change, or as you add new features. The model is the mechanism to ensure the information received at many desktops is always accurate and consistent.

Your Business Objective

Consider how the users view their business. The information model should help users at all levels run the business better. Different businesses use different economic, social, environmental, and process drivers to perform business activities. The relative importance may vary across industries, but for the particular business you are modeling these should be evident.

For example, the business of chemical processing is not only driven by concern for product and quality, but a realization of its effect on the environment, the local community, the safety of its personnel, the effective management of the physical assets of the business, and its consumption of energy. As a result, you might expect to see components reflecting these concerns and activities at the highest levels in a hierarchical model.

In addition, a representation of the 'drivers' of the business, such as an organization of the metrics, or measures of how well the business is performing, in key performance indicators, scorecards, and so forth, at the appropriate place in the model will help.

The User Community

Address the different user communities. As a general information system, XHQ must address the needs of several communities within the business. However, different communities tend to reflect their needs to perform their job functions in different ways. Information organization for maintenance workers may be very different than for the process worker, or the document management professional.

Example of User Groups

User	Concerns
Plant Management	What are my current operations costs? What is the operational efficiency of my plant? Where is production relative to plan?
Production Supervisor	What is the status of production? When is the next scheduled maintenance? Where is the overall efficiency of production?

User	Concerns
	How is the product quality trending?
Maintenance Supervisor	When was the last maintenance performed?
	Which maintenance work orders are pending?
	How does PM impact production?

If XHQ is implemented for a specific targeted application, then the needs of a single user group may predominate and provide the appropriate model. If the intent is to provide an easy web-enabled way to get information out of the asset management system, then a model that organizes information specifically according to the assets might work best for that user community.

Consequently, you must identify the types of information and the organization that interests each user community. You must acquire an understanding of how a typical user from each community will access the information. There will be differences in how users perceive their data. Then you must examine the relationships between the user communities. Your task will be to find a way that best provides information access to all the people who will use the system. For example, an informational model that reflects how the assets or equipment is logically organized is one of the easiest models to recognize and use. This might not be the preferred mechanism for the maintenance activities, but it is likely to be the best shared model for the whole user community.

The Backend Systems

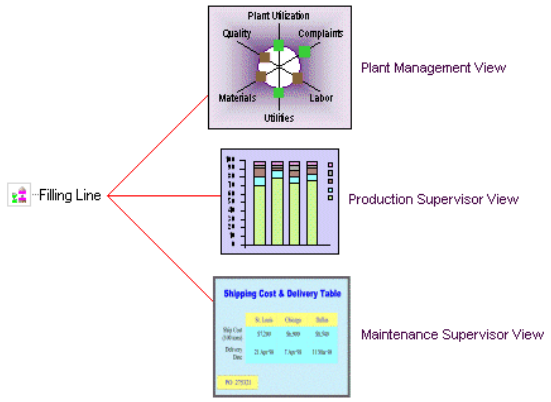
Examine the source locations of the data and how XHQ will receive the data. The entry of information into XHQ imposes some limitations that will influence how you might organize the information. The most significant consideration is whether the source of the data will be *historical* or *collection-based*.

In general, data coming from a relational database will be collection-based. But where there is a fixed number of records (often only one) that map to a particular place in the model, they may be point-based. When there is a form of filtering on records, the data will be collection-based. For example, the list of tanks currently containing feedstock may vary as some tanks move from holding feedstock to holding products. In this case, the number of records varies based on operating conditions, and the data relevant to that aspect of the information model are collection-based.

In cases where there is always exactly one source record or value that maps to one component or member in the solution, a point-based or history-based model should be used. For example, the number of days since the last incident recorded at a site will usually be a single value. If the data change over time and tracking those changes is important, the data are history-based.

Organizing Information

The model should generate consistent information, provide ease of navigation and enable information reuse. As a general principle, XHQ provides everybody's secondary user interface. Each worker such as the process operator, the laboratory technician, the planner, the order entry clerk has a user interface for the computer task that represents their primary job function. XHQ does not set out to replace that user interface. XHQ provides an appropriate shared user interface for each worker to extract information from the other systems that are not primary, and that would otherwise be accessed infrequently to capture information that helps perform the primary function.



One Component, Three Different Views

Maintenance staff would organize according to equipment types with like equipment grouped together. However, the operations staff is likely to use a hierarchical organization starting with the individual production lines, each containing the respective equipment. The document staff would likely organize by document types. There may be inherent conflict between the required models for each user.

Practical Guidelines

Here are some guidelines to help you design your information model.

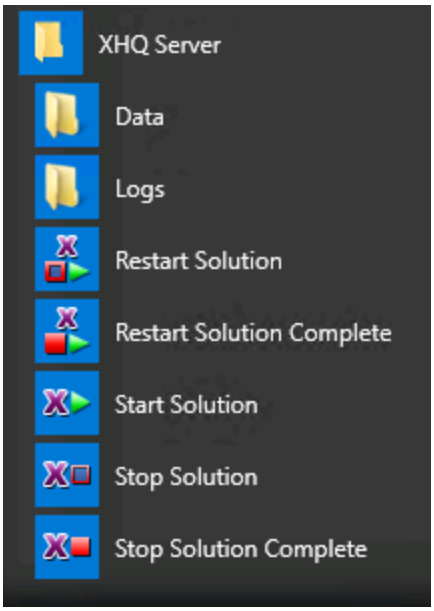
- Be prepared to make compromises to make your model accessible to the widest audience.
- The model you design should keep the users from seeing the disparate, backend systems and allow them to see their business.
- Do not go beyond 4 levels deep below the top component into your model hierarchy. Any more can make the model overly complex. This will mean you may need to reconsider your thinking to provide a simple, organizational model. In some cases, additional levels of components may be used, but the lower levels will be used to improve the reuse within the solution and the lower levels will be hidden from end-users.
- Design from the top; implement from the bottom. Again this process will require some iteration. You will need to shake out the design from the top, but since components are building blocks, you need to start from the bottom.
- Examine the model, looking for organization and differences that are based on the source of the data – normalize these differences so the same model can apply independent of the source.
- Try to think through the ultimate goals for the information management system provided by XHQ. Having done this, you can design components that initially may be very light in providing information, but are the containers into which new information can be populated as it becomes available.
- Get started. XHQ is very agile, extensible, and changeable. You can make many changes on-the-fly without taking the server down, and with little disturbance to the end-user.

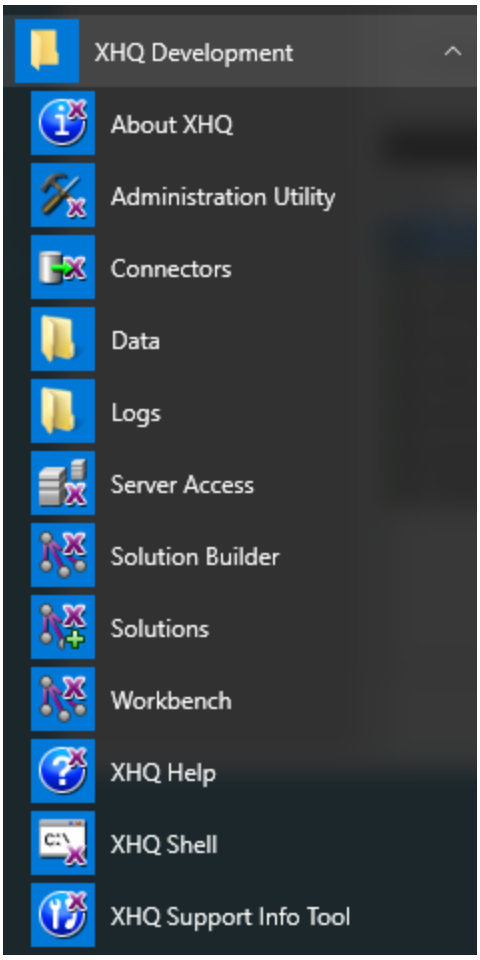
2 | Development Fundamentals

In this section, you will learn how to start XHQ and get an introduction to the XHQ Development Client interface and navigation.

Starting XHQ

All XHQ functions are initiated from the **Start menu**.





Start Menu – XHQ Server and XHQ Development Client Commands

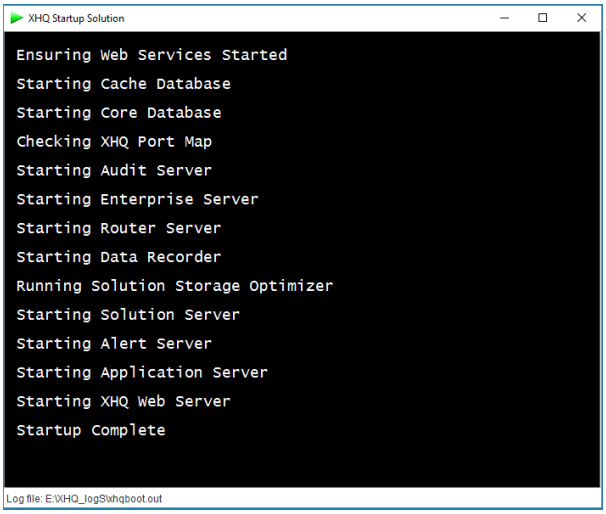
Before you can work in the XHQ Development Client environment, you must first start the XHQ Server.

Icon	Description
	• **Restart Solution Complete** Completely stops all embedded services (for example, database and web service) and IIS (the entire web server, not just XHQ Web Root), and then restarts XHQ.

Icon	Description
	• **Restart Solution** Stops and then starts XHQ.
	• **Start Solution** Starts XHQ.
	• **Stop Solution Complete** Completely stops all embedded services (for example, database and web service) and IIS (the entire web server, not just XHQ Web Root).
	• **Stop Solution** Stops XHQ.

To start and stop the servers

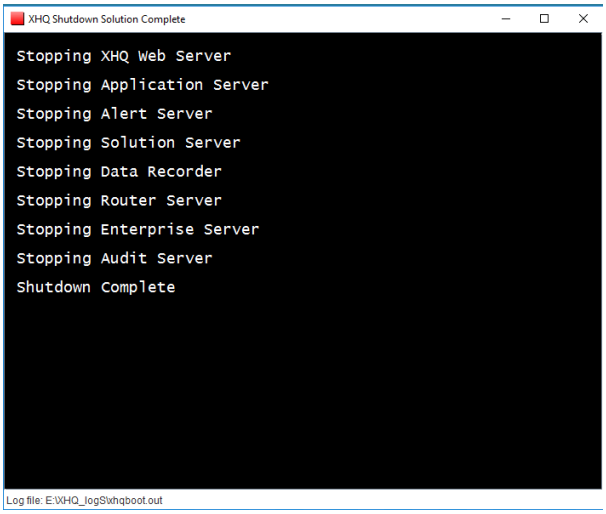
From the Start menu, go to **XHQ Server** then to **Start Solution**.
The XHQ Startup Solution window appears.



Once the servers are started, from the Start menu, go to **XHQ Development** to access the XHQ Development Clients.

To stop servers, from the Start menu, go to **XHQ Server** then to **Stop Solution Complete**.

The XHQ Shutdown Solution Complete window appears.

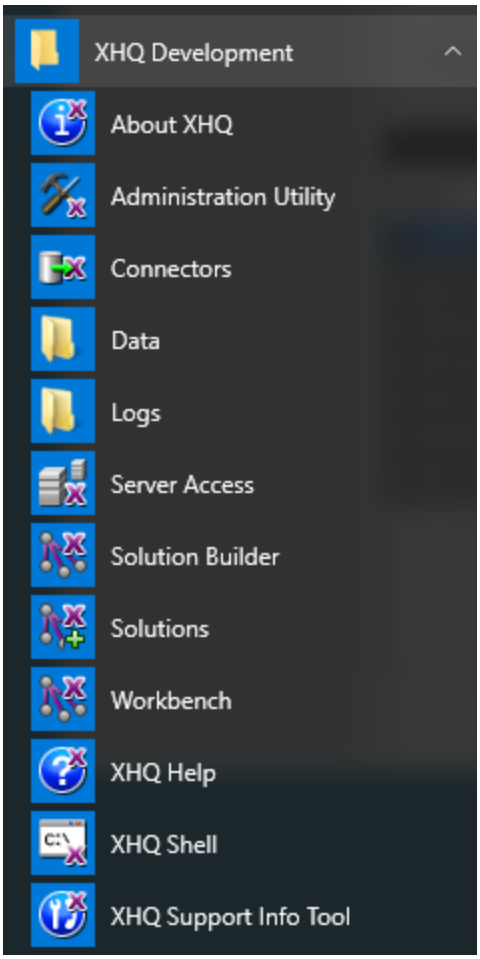


To restart the server

Click **Start**, point to **XHQ Server**, then to **Restart Solution**.
This automatically stops and starts the XHQ Server.

XHQ Development Start Menu

From the XHQ Development Start menu, you can launch the XHQ Development Clients, perform XHQ administrative tasks, access the XHQ Server data and log file directories.



XHQ Development Start Menu Commands

Icon	Description
	Data All development client data and configurable settings can be accessed from this directory, which (by default) is located in `%APPDATA%\XHQ\XHQ devClient\repos`
	Logs All development client log files can be accessed from this directory, which (by default) is located in: `%APPDATA%\XHQ\XHQ devClient\log`

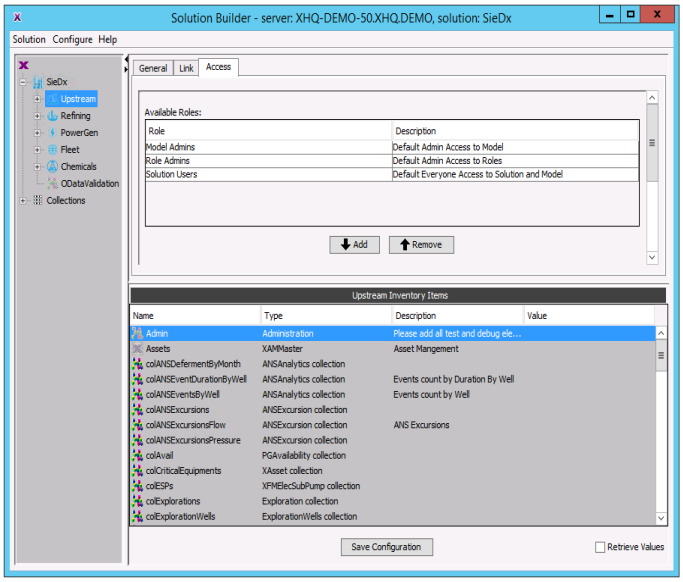
The **Data** and **Logs** menu commands are enabled after an XHQ Development Client (such as the XHQ Workbench) has been launched. The paths that these commands point to are only created after an XHQ Development Client has been started once.

Icon	Description
	About XHQ Launches the XHQ About dialog.
	Administration Utility Launches the XHQ Administration Utility.
	Connectors Launches the Connector Administration dialog, which enables you to add, edit, and remove XHQ connectors, as well as import/export connector definitions.
	Server Access Enables you to identify the XHQ Server host name.
	Solution Builder Launches the XHQ Solution Builder.
	Solutions Launches the Solution Administration, which lists the available solutions and their given top-level component.
	Workbench Launches the XHQ Workbench.
	XHQ Help Launches the XHQ online help files.
	XHQ Shell Launches a command prompt window that automatically references the correct path to run `xhqci`. For details, see the topic, *Installing the XHQ Development Client (Stand-alone)*, located in the **XHQ Installation Guide**.
	XHQ Support Info Tool Launches the XHQ Support Information function of the XHQ Administration Utility.

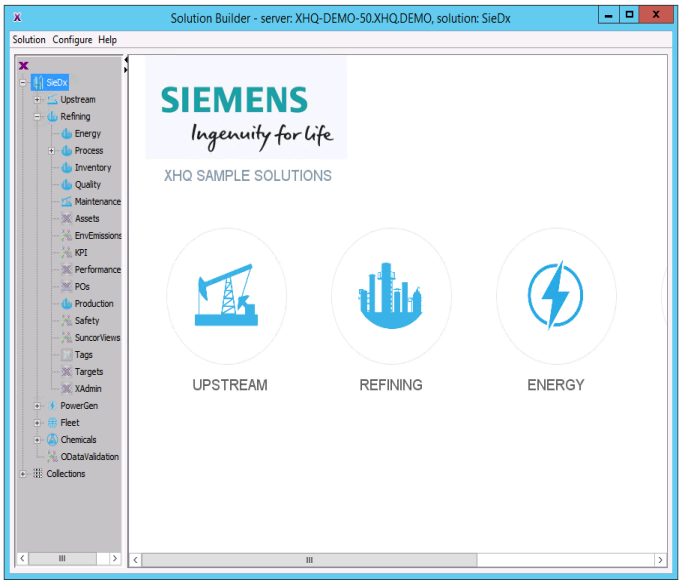
Defining the XHQ Panels

The XHQ Solution Builder interface consists of four primary panels:

- Navigation Panel (both Development and Runtime views)
- Edit Panel (Development view: Detail)
- Inventory Items Panel (Development view: Detail)
- View Panel (Runtime view only)



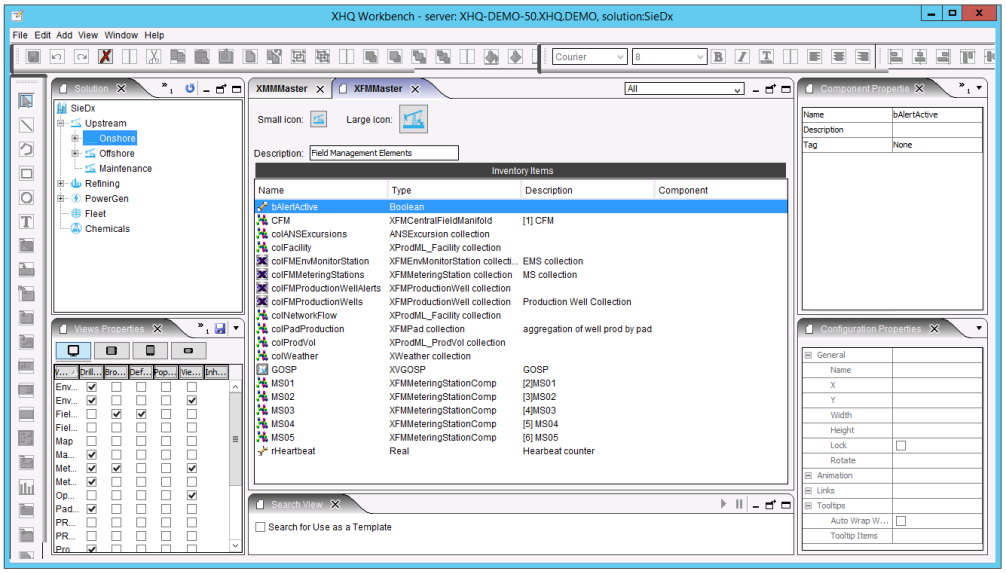
XHQ Solution Builder: Development View



XHQ Solution Builder: Runtime View

The XHQ Workbench default layout consists of:

- A Menu bar
- Several toolbars
- A central Edit panel
- And several tabs



The Navigation Panel

The navigation panel in the XHQ Solution Builder displays a tree-style view of navigation nodes and components within a solution, and provides a mechanism to display the views that are available for any node. In the navigator you can open and close branches of the tree to see the structural content of a solution, and you can select available views for call-up in the view panel.

The Inventory Items Panel

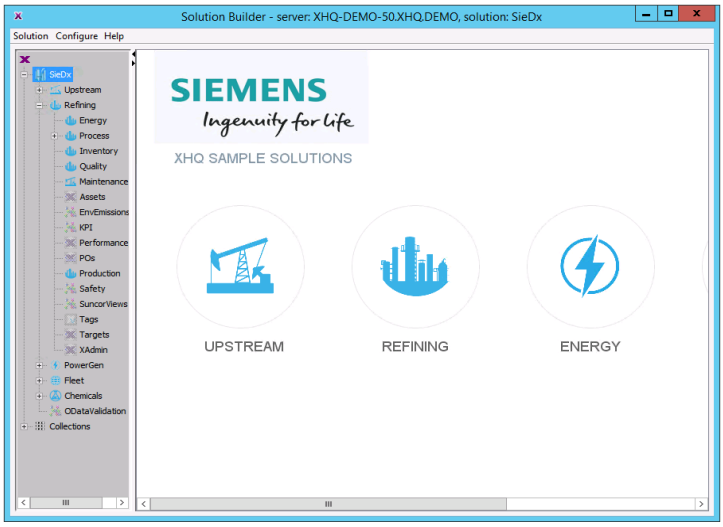
The Inventory Items panel displays a list of all the items contained in a selected instance of a component or solution, including name, type, and description. An Inventory Items panel is displayed in both the XHQ Workbench) and the XHQ Solution Builder.

The Edit Panel

When you select an inventory item, either from the XHQ Workbench or the XHQ Solution Builder, XHQ displays its specific properties in the edit panel, using a tabbed layout (Detail Tabs). By clicking on the available tabs, you can see and enter specific configuration values.

The View Panel

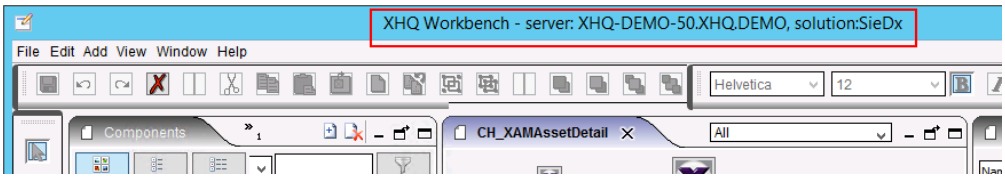
The View panel, which can be displayed while using the XHQ Solution Builder development environment, enables you to observe any available view, updating with live data.



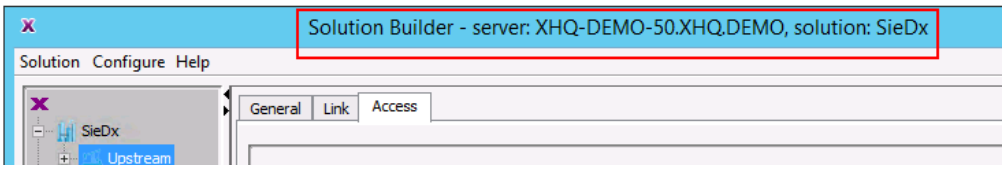
View and Navigation Panels as seen from the XHQ Solution Builder

SIDEBAR: ABOUT THE DEVELOPMENT CLIENT TITLE BAR

The title bars to the XHQ Workbench and the XHQ Solution Builder display the **XHQ Solution Server name** (for example, localhost) as well as the **XHQ Solution name** (for example, Enterprise) to which the XHQ Development Client is connected.



XHQ Workbench Title Bar



XHQ Solution Builder Title Bar

Using XHQ Navigation

XHQ uses Windows-style pull-down menus, left- and right-clicking mouse navigation, and point-and-click selection functionality. There are three ways you can navigate through your solution:

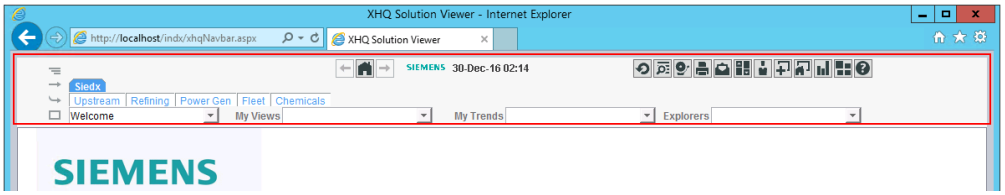
- **XHQ Navigation Bar**
  The XHQ Navigation Bar (XHQ NavBar) offers an alternative to the hierarchical tree typically used to navigate through an XHQ solution.

  For more information, go to the topic, *The XHQ Navigation Bar*, located in the XHQ Solution Viewer User's Guide.

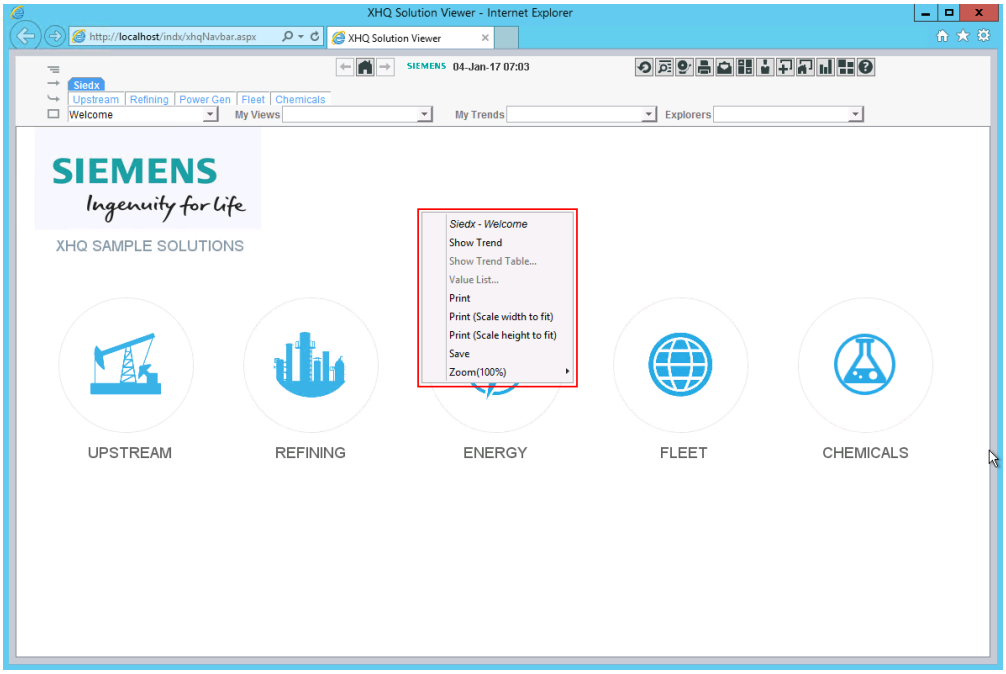
- **Left and Right Click Operations in the View Panel**
  Left-click on a node or component in the tree and the default view for that node appears on the View panel. Right-click on a component and a shortcut menu appears, displaying all the views associated with the given component.

- **Tree style Navigation**
  From the Navigation panel, you can expand or collapse the navigation tree and select a node.

  By default, the navigation panel is hidden; that is, when you launch an XHQ URL, the left panel that displays the solution navigation tree is hidden.



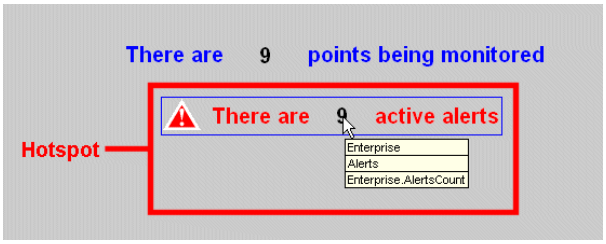
XHQ Navigation Bar



Right-click Shortcut Menu

Left- and Right-Click Operations in the View Panel

When you mouse-over *hotspots* in the View panel, a rectangular blue outline appears, indicating that it is selectable.



Mouse-over Hotspot and Tooltip – View Panel

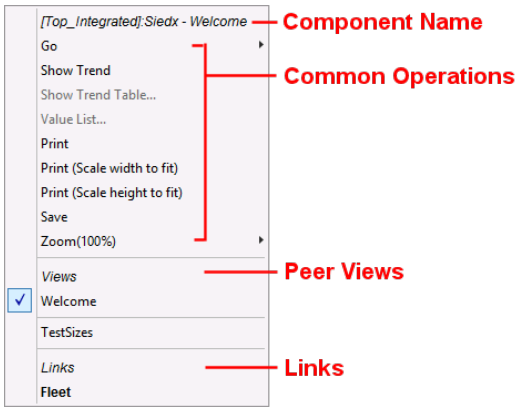
Depending on how the view is configured, left-clicking on a hotspot can open any of the following:

- The default (top) view of the component to which the view belongs.
- A URL link.
- A view link.
- The Trender.

From the View panel, you can right-click on the view to:

- Select a peer view of the component.
- Jump to other views and designated URLs.
- Select variables for the Trend Viewer.
- Print views.
- Go to previously visited views.

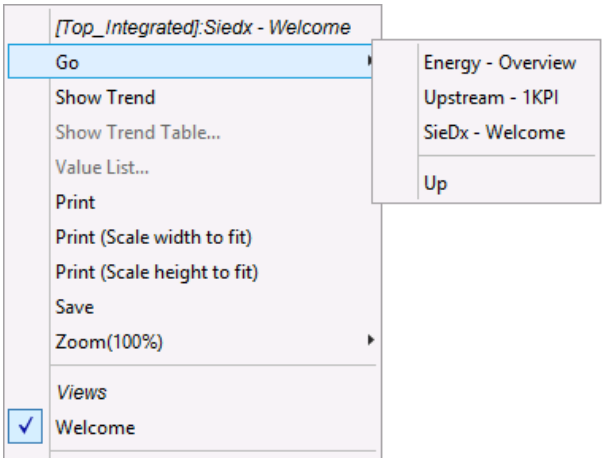
The content of the right-click shortcut menu is dynamically generated to correspond to the context of the view component you select. The first line shows the name, in italics, of the component to which the right-click shortcut menu belongs. Following that are the main parts: the *Common Operations*, the *Peer Views*, and the *Links*.



Right-click Shortcut Menu – View Panel

About Common Operations

In the **Common Operations** section, you will find the ***Go*** command with a right arrow that causes a submenu menu to appear with a list of as many as six previously visited views.



Go Submenu Example

From the ***Go*** submenu, you can also select the ***Up*** command to access the default view of the parent component and the ***Solution Home*** option to access the XHQ solution home view you define.

The next two commands in the Common Operations section are ***Show Trend*** and ***Value List.*** These relate to Trend Chart operations and the list of variables selected for trending. The last item, ***Print***, allows you to print a view from the browser.

About Peer Views

The next major section of the right-click shortcut menu is **Peer Views**. *Peer views* are views against the same component.

In this section, you will find an alphabetically ordered list of the views that are peers of the component to which the current view belongs – the component indicated by the blue outline.

A check mark appears in front of the view that is currently displayed. The bold default item could be in the peer view list or in the link list.

About Links

The bottom portion of the right-click shortcut menu, **Links**, shows each hyperlink item that was configured in the view editor. Those links can jump to either another designated XHQ view from anywhere within the solution, or to a selected URL (web page).

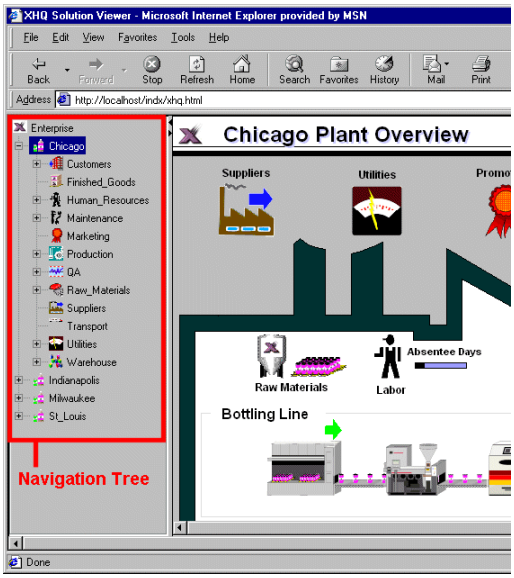
Tree-style Navigation

Tree-style navigation is available from the XHQ Solution Builder and the XHQ Solution Viewer (browser).

By default the navigation panel from the XHQ Solution Viewer is hidden; that is, when you launch an XHQ URL, the left panel that displays the solution navigation tree is hidden. To display the navigation panel, set the `showTree` preference to "yes".

When you click on an icon in the tree, a dark blue background highlight appears behind the icon and the component name and the default view appears in the view panel. From the XHQ Solution Builder, the detail (or structural) view of the component appears. From the XHQ Solution Viewer, the default view is the top view in the view list.

You can expand or collapse nodes in the navigation tree to show or hide additional component levels below the selected node. Nodes that can expand or collapse have either a plus or a minus sign, respectively, in front of the component icon.

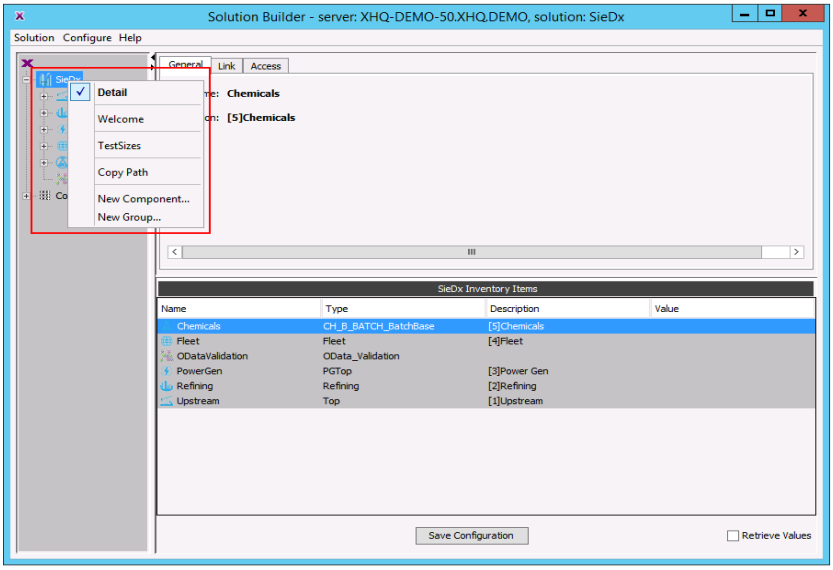


XHQ Solution Viewer – Navigation Tree

Left- and Right-Click Operations in the Navigation Tree

If you *left-click* on an icon in the navigation tree, a blue background appears behind the icon and the component name, and the default view appears in the View panel.

When you *right-click* on an icon in the tree, a list of the accessible component views appears in a shortcut menu. You may select any view from the list to see it in the View panel.



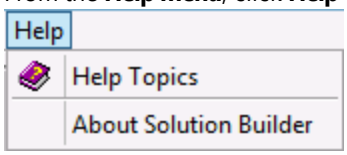
Right-click Shortcut Menu – Navigation Panel

In this shortcut menu, the view in bold is the default view. The checkmark next to the view name indicates that it is currently open in the View panel.

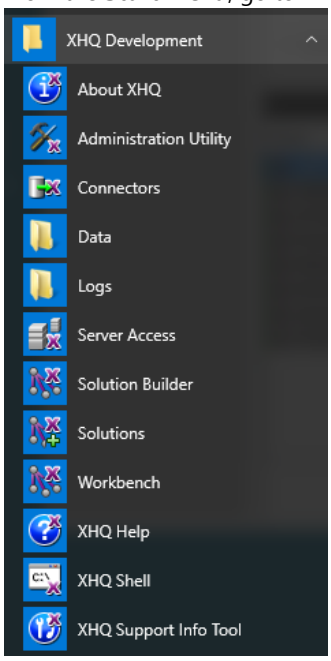
Accessing Help

There are three ways to access the online help system, **XHQ WebHelp**, from any of the XHQ Development Clients:

- From the **Help menu**, click **Help Topics**.



- From the **Start menu**, go to **XHQ Development** > **XHQ Help**.



or

- Press **F1**.

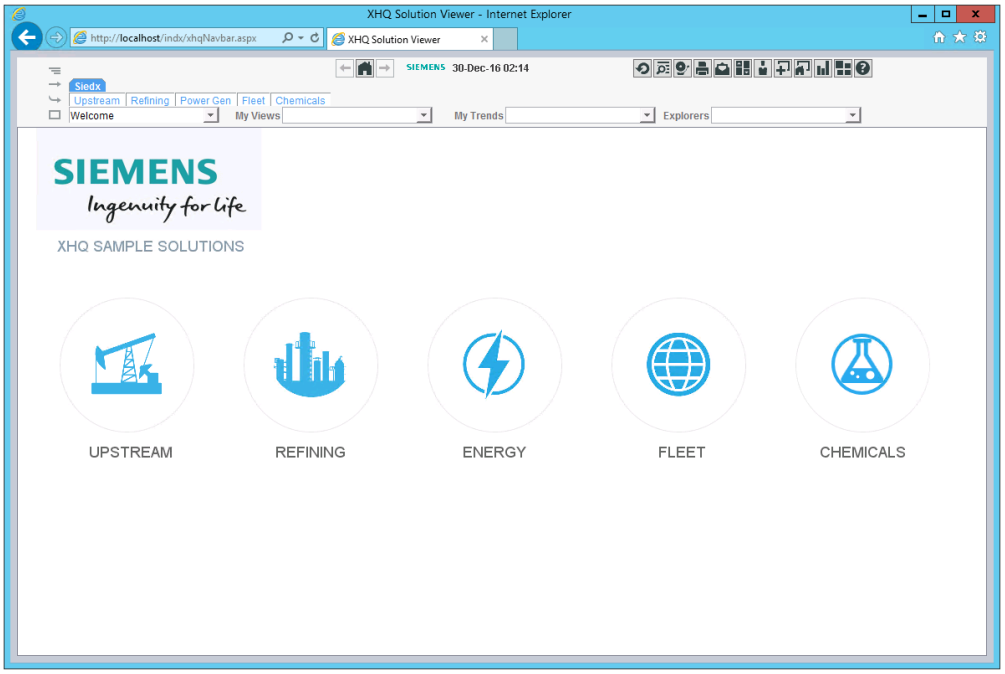
You can access the entire XHQ Documentation Set in PDF format from the XHQ WebHelp homepage quick link.

The XHQ Runtime Environment

A browser is the most common way to access the XHQ runtime environment. You can also access the runtime from within the XHQ Solution Builder.

XHQ Solution Viewer at Runtime

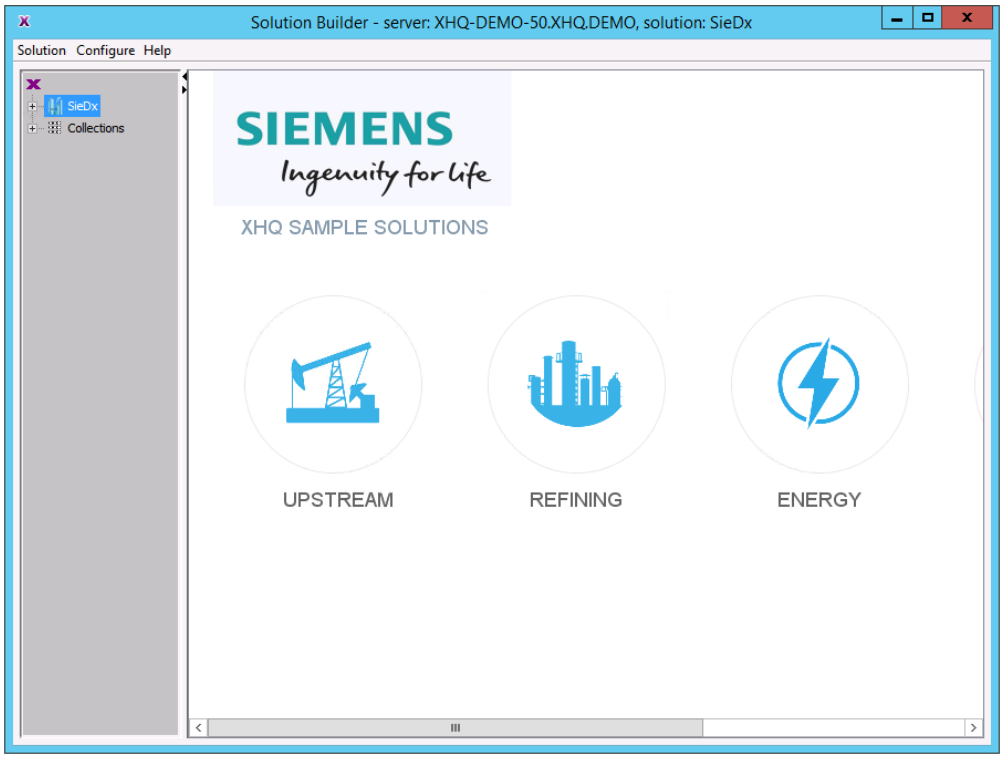
When you access an XHQ solution from a browser, you see the navigation and content panels. You are only permitted to see views. You cannot update components, views, or the solution(s).



Runtime from the XHQ Solution Viewer

XHQ Solution Builder at Runtime

The XHQ Solution Builder shows the same navigation tree and runtime views as the XHQ Solution Viewer. In addition, you can see a detail or structural view of each component and edit many of its runtime properties.



Runtime from the XHQ Solution Builder

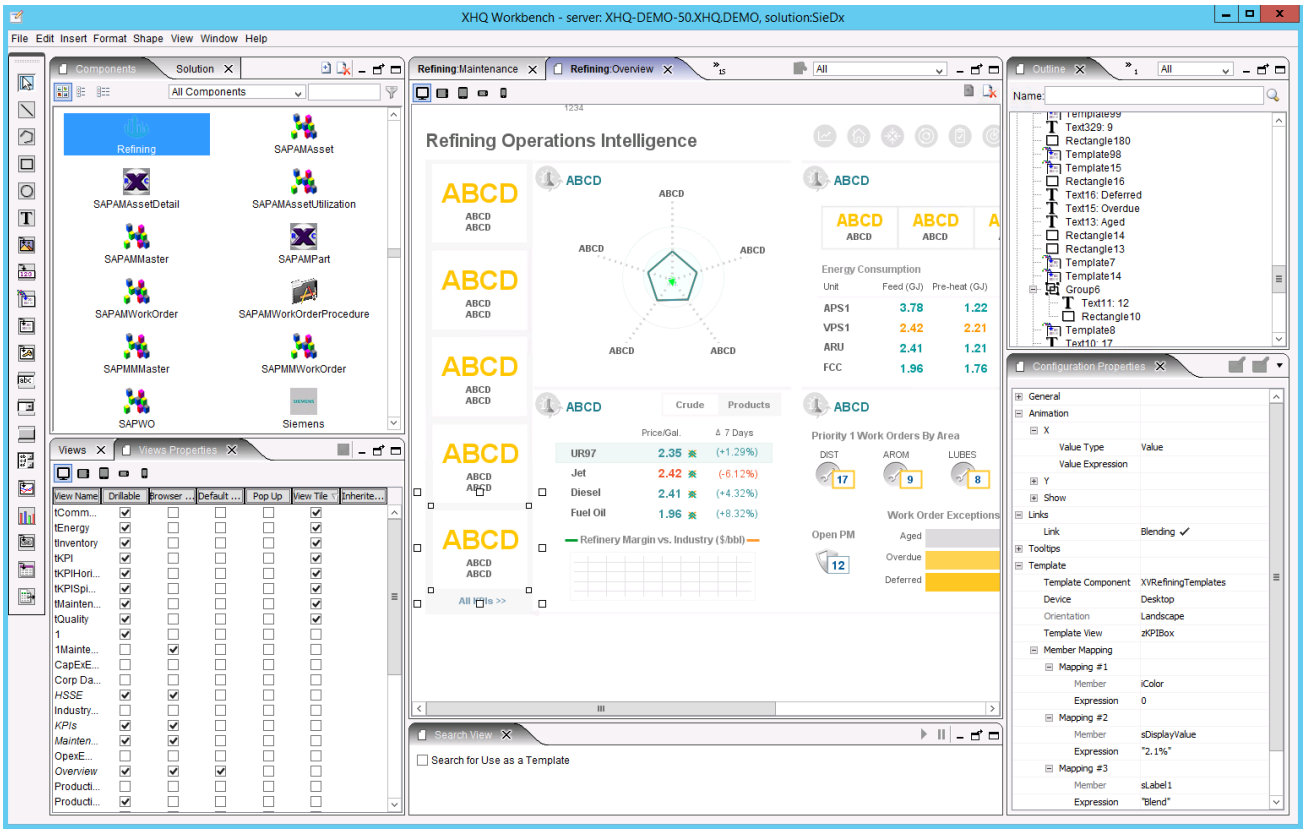
From the XHQ Solution Builder, you can access:

- The detail (or structural view) used by the application developer.
- The views that update with real-time values (identical in build-time and runtime).

When you click on any node in the navigation panel, the selected node highlights, and the content panel displays the top view for the selected node. The top view at build time in the XHQ Solution Builder is always the Detail view. All other available views are sorted alphabetically.

3 | XHQ Workbench

Use the XHQ Workbench for creating views and managing corresponding components.



XHQ Workbench - Default Screen

To access the XHQ Workbench

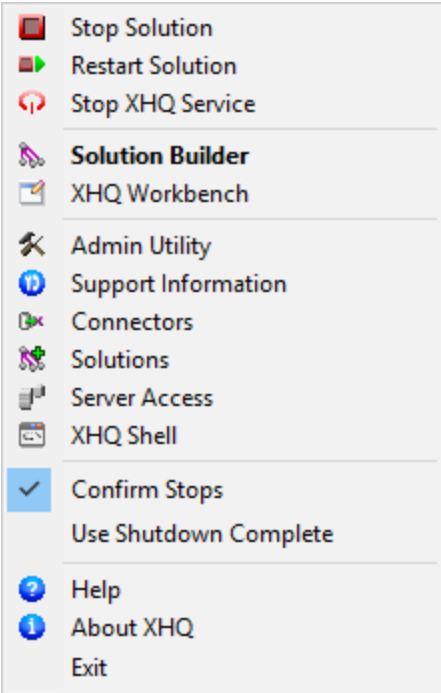
1. Run the **`xhq_tray`** application, which is typically located in the `%XHQ_SERVER_HOME%\bin` directory (which by default is `C:\Program Files (x86)\XHQ\XHQ Server\bin`).

   The XHQ icon is added to the Windows system tray.

2. From the **Windows system tray**, click the XHQ icon and start the XHQ solution.

   Once the solution is started, the XHQ icon shows a green arrow .

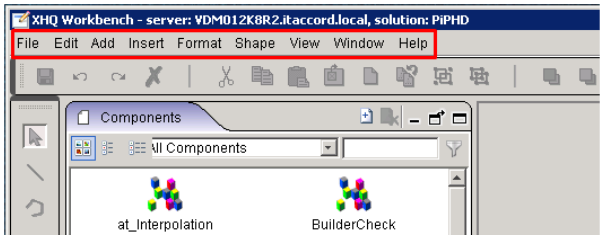
3. Click the **XHQ icon** and, from the shortcut menu, click **XHQ Workbench**.



XHQ System Tray Shortcut Menu

XHQ Workbench Menu Commands

This section describes the menu commands you can perform from the XHQ Workbench. Use the XHQ Workbench to add and edit components and views.

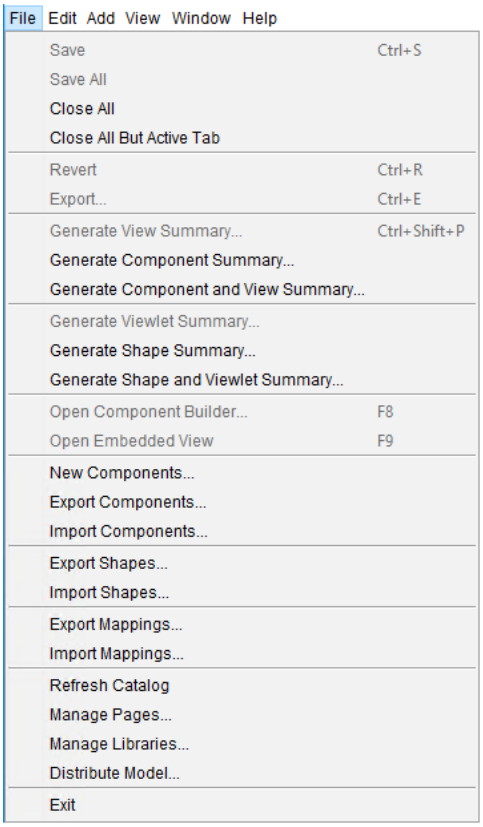


XHQ Workbench Menu Bar

Common Menu Commands

The following menus are **common** to both component and view building.

File Menu

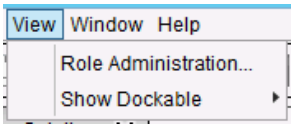


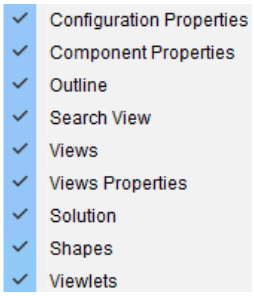
Command	Action	Corresponding Toolbar Icon
Save	Saves current component/view to	

Command	Action	Corresponding Toolbar Icon
	repository.	
Save All	Saves all component(s)/view(s) modified.	
Close All	Closes all open component(s)/view(s) tabs.	
Close All But Active Tab	Closes all open component(s)/view(s) tabs except for one that is currently active.	
Revert	Reverts the view to the last saved state.	
Export	Exports selection to file.	
Generate View Summary	Generates a summary of the view configurations for the current view and displays it in a browser.	
Generate Component Summary	Generates a summary report for the given component and displays it in a browser.	
Generate Component and View Summary	Generates a summary that lists all the components within the model, and all related views and component members.	
Generate Viewlet Summary	Generates a summary report of Viewlets.	
Generate Shape Summary	Generates a summary report of Shapes.	
Generate Shape and Viewlet Summary	Generates a summary report of both Shapes and its Viewlets.	
Open Component Builder	Opens the Component Details tab	
Open Embedded View	Opens the embedded view.	
New Component	Creates a new component. Launches the Component Details tab.	
Export Components	Exports selected component to a ZIP file.	
Import Components	Imports selected components from the exported ZIP file.	
Export Shapes	Exports selected shapes to a ZIP file.	
Import Shapes	Imports selected shapes from the exported ZIP file.	
Export Mappings	Exports the mapping configuration, in XML format, for the selected shape.	

Command	Action	Corresponding Toolbar Icon
Import Mappings	Imports the mapping configuration for the selected shape.	
Refresh Catalog	Applies any pending changes to the current model. Clears the XHQ Workbench cache (if it exists) and refreshes the XHQ Workbench. **Note**: The time it takes to update depends on the number of component instances changed by the component class update.	
Manage Pages	Allows you to create pages to organize components.	
Manage Libraries	Allows you to add/rename/delete libraries. **Note**: To delete a library, it must not contain any shapes.	
Distribute Model	Syncs the changes between distributed servers included in a realm.	
Exit	Closes the XHQ Workbench.	

View Menu



Command	Action
Role Administration	Launches the *Role Administration utility*, which allows you to enable security within XHQ.
Show Dockable	Allows you to show/hide each of the dockable tabbed panels. 

Window Menu

Command	Action
Save Layout As	Saves currently opened tabs/views.
Open Layout	Opens previously saved tabs/views.
Delete Layout	Deletes previously saved tabs/views.
Reload Default Layout	Loads the layout used in the last session.
Reset Default Layout	Loads the default XHQ built-in layout.

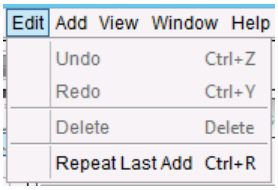
Component Building Menu Commands

The following menus are specific to adding and/or editing.components.

XHQ Workbench Component Menu Bar

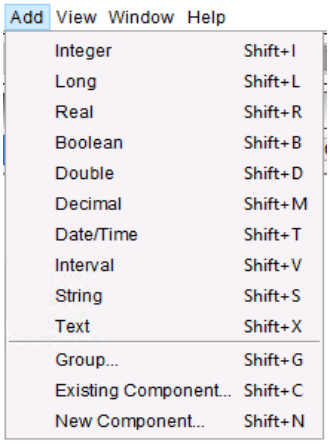
The File, View, Window, and Help menus are the same for both component and view building.

Edit Menu



Command	Action
Undo	Reverses the last command.
Redo	Reverses the action of the Undo command.
Delete	Removes the selected Inventory item.
Repeat last add	Allows you to add multiple members of the same type. The default member name increments by one for each repeat.

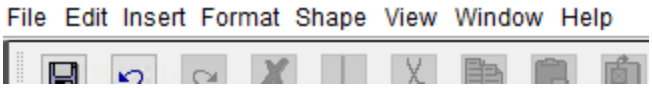
Add Menu



Command	Action
Integer	Adds the given type member.
Long	
Real	
Boolean	
Double	
Decimal	
Date/Time	
Interval	
String	
Text	
Group	Adds a group member.
Existing Component	Adds an existing component as a member.
New Component	Adds a new component as a member.

View Building Menu Commands

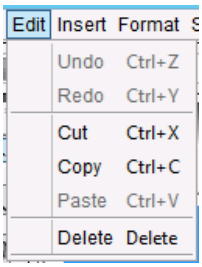
The following menus are specific to adding and/or editing views.



XHQ Workbench View Menu Bar

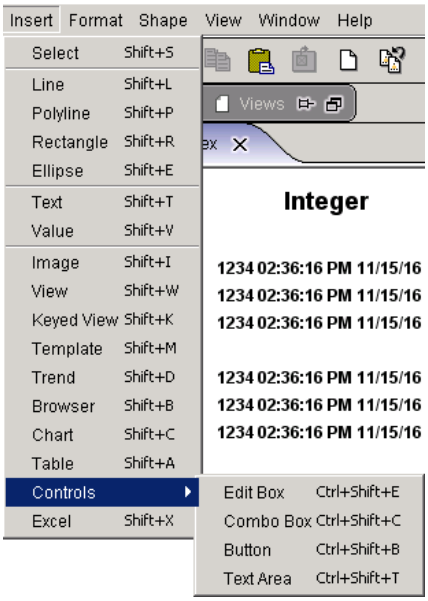
The File, View, Window, and Help menus are the same for both component and view building.

Edit Menu



Command	Action	Corresponding Toolbar Icon
Undo	Reverses the last command.	
Redo	Reverses the action of the Undo command.	
Cut	Removes the selection and copies it to the clipboard. **Cut vs. Delete:** In addition to removing it from the view, the CUT command copies the selected view element to the clipboard so that you have the option of pasting it elsewhere. On the other hand, the DELETE command *completely* removes the selected view element.	
Copy	Copies the selection to the clipboard.	
Paste	Inserts the cut or copied view element or view from the buffer into the same view or a different one.	
Delete	Removes the selected view element.	

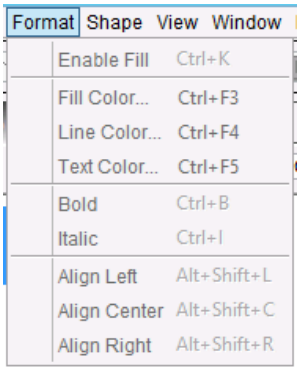
Insert Menu



Command	Action	Corresponding Toolbar Icon
Select	Selects, moves, and resizes view elements.	
Line	Draws straight lines in any directions.	
Polyline	Draws connected line segments and closed shapes.	
Rectangle	Draws squares and rectangles.	
Ellipse	Draws circles and ellipses.	
Text	Enters, selects, and edits text.	
Value	Inserts live updating values from a data source.	
Image	Imports images.	
View	Inserts an existing view into the current view.	
Keyed View	Inserts a keyed view, which is an embedded tag view.	
Template	Inserts a template (a reusable view based on a component model).	

Command	Action	Corresponding Toolbar Icon
Trend	Inserts either a time plot or an XY-plot into the current view.	
Browser	Inserts a web control.	
Chart	Inserts an XHQ chart.	
Table	Inserts a table control.	
Controls	Inserts an Edit box, Combo box, Button, or Text Area tool.	
Excel	Inserts an Excel item.	

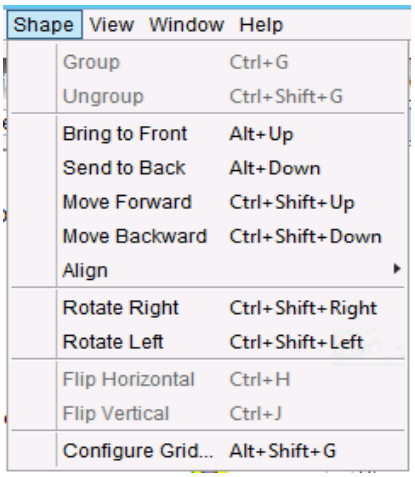
Format Menu



Command	Action	Corresponding Toolbar Icon
Enable Fill	Toggles between fill and no fill for a selected view element.	
Fill Color	Selects and applies fill color to a selected view element.	
Line Color	Selects line or outline color for a selected view element.	
Text Color	Applies color to the text you select.	

Command	Action	Corresponding Toolbar Icon
Bold	Applies the **bold** style.	
Italic	Applies the *italicized* style.	
Align Left	Horizontally aligns the left edge of each **text** line.	
Align Center	Horizontally aligns the center of each **text** line.	
Align Right	Horizontally aligns the right edge of each **text** line.	

Shape Menu



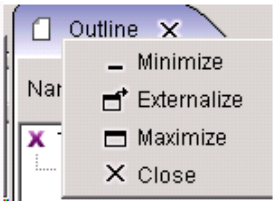
Command	Action	Corresponding Toolbar Icon
Group	Groups multiple selected view elements into one unit.	
Ungroup	Reverts group back into individual view elements.	
Bring to Front	Brings the view element to the top of the stack.	
Send to Back	Sends the view element to the bottom of the stack.	
Move Forward	Advances the view element by one position towards the top of the stack.	

Command	Action	Corresponding Toolbar Icon
Move Backward	Moves the view element back one position towards the bottom of the stack.	
Align Lefts	Aligns the objects you select by their left edges.	
Align Centers (Vertically)	Aligns the objects you select vertically by their centers.	
Align Rights	Aligns the objects you select by their right edges.	
Align Tops	Aligns the objects you select by their top edges.	
Align Middles (Horizontally)	Aligns the objects you select horizontally by their middles.	
Align Bottoms	Aligns the objects you select by their bottom edges.	
Rotate Right	Rotates a view element 90 degrees clockwise (towards the right).	
Rotate Left	Rotates a view element 90 degrees counterclockwise (towards the left).	
Flip Horizontal	Flips an image horizontally along its vertical axis.	
Flip Vertical	Flips an image vertically along its horizontal axis.	
Configure Grid	Opens the Grid Configuration dialog box.	

Right-click Shortcut Menu Commands

Right click menu has different functions if applied in different areas of the workbench

On a Tab



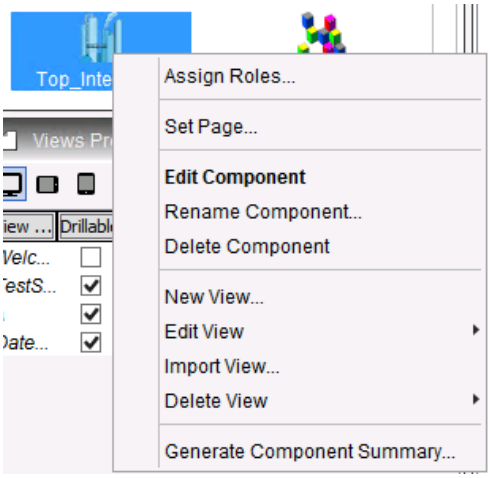
Command	Description
Minimize	Minimizes the tab in upper bar.

Command	Description
Externalize	Detaches the tab from main Workbench window.
Maximize	Makes the tab full screen.
Close	Closes the tab.

You can restore a minimized panel by clicking Restore .

On a Component

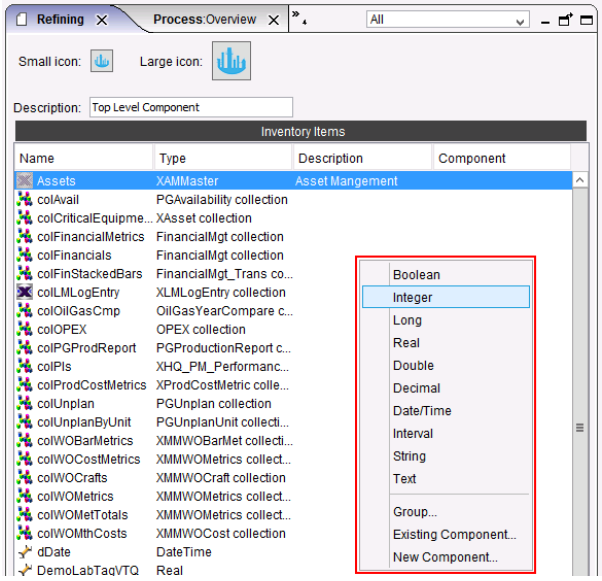
Right-clicking on a component produces the following shortcut menu.



Command	Description
Assign Roles	Assign different permissions/roles for component
Set Page	Assign component to a specific page
Edit Component	Opens Component Details tab of the selected component
Rename Component	Renames the component
Delete Component	Deletes component
New View	Creates a new view
Edit View	Edits a previously created view
Import View	Imports previously exported view
Delete View	Deletes View
Generate Component Summary	Generates a summary of the given component and all its views.

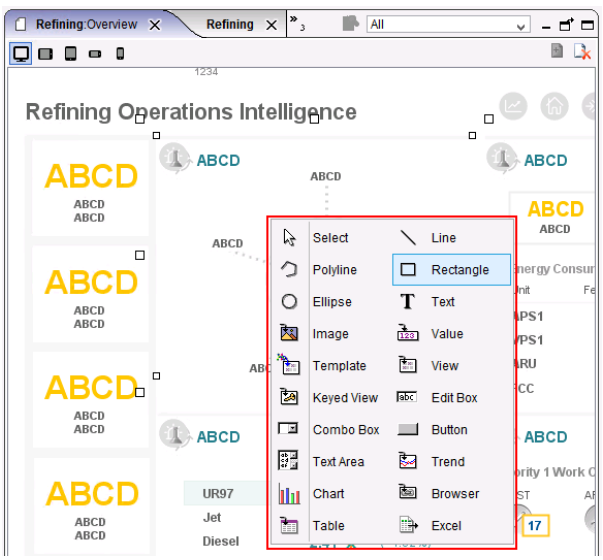
Inside the Component Details tab

Right-clicking anywhere inside the Component Details tab and a shortcut menu appears with options similar to the Add menu options.



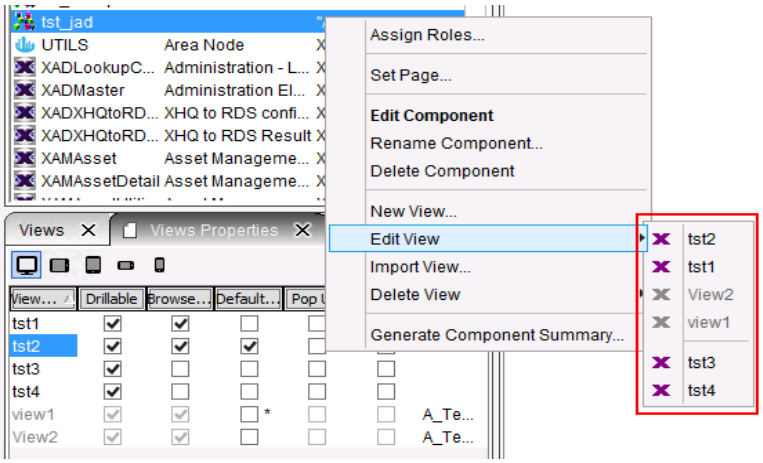
On a View

Right-clicking on a view produces a shortcut menu that is similar to the View Tools Palette or the Insert menu.



View Submenu Grouping

From the **Right-click Shortcut Menu**, point to **Edit View** and a submenu appears.



Example of Submenu with Partitions

To make finding views on this submenu easier, they are separated into the following partitions:

- **Browsable**
  The component "owns" these views. These views are both browsable and editable. A line separates the browsable views from the non-browsable.
- **Non-browsable**
  The component owns these views, but browsing has been disabled (through *Views Properties*). These views are still editable.
- **Default**
  The default view is the first view on the list, depending on whether it is browsable or non-browsable.

Special considerations:

- **Inherited**
  Inherited views are **grayed-out**. Browsing has either been enabled or disabled, depending on the Views Properties configuration. To edit, double click on the grayed-out view name. This takes you to the component from which the view was inherited and opens the related Views Properties.
- **Asterisk**
  This indicates that the view was inherited from the base component.
- **Virtual**
  Virtual views are in **bold**.

In each group, views are listed **alphabetically**, with the exception of the default view.

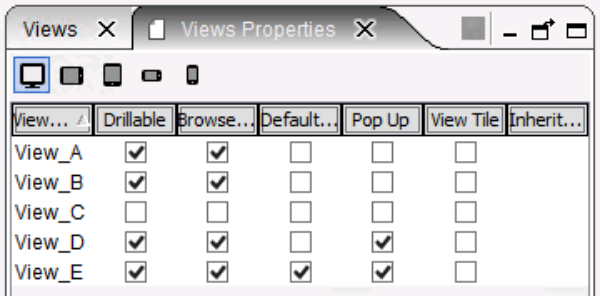
In the XHQ Solution Builder, inheritance is irrelevant.

Consider the following examples.

EXAMPLE 1

Given: The following **Views Properties** configuration.

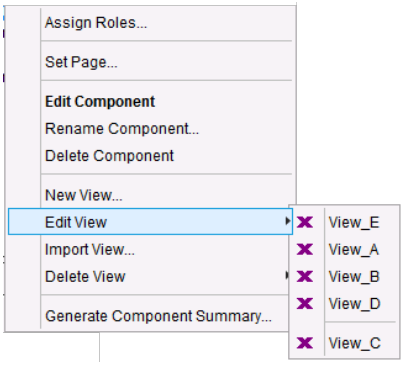
For more information, see the topic, *Setting Views Properties*.



Things to Note

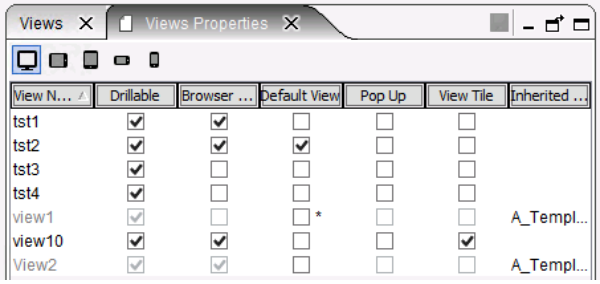
- Browsing is disabled for view `View_C`.
- `View_E` is the default view.

This properties configuration results in the following **Edit View** submenu:



EXAMPLE 2

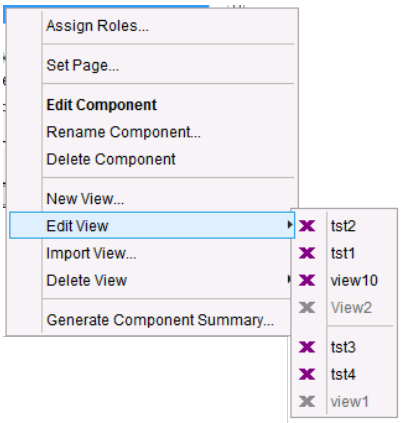
Given: The following **Views Properties** configuration.



Things to Note

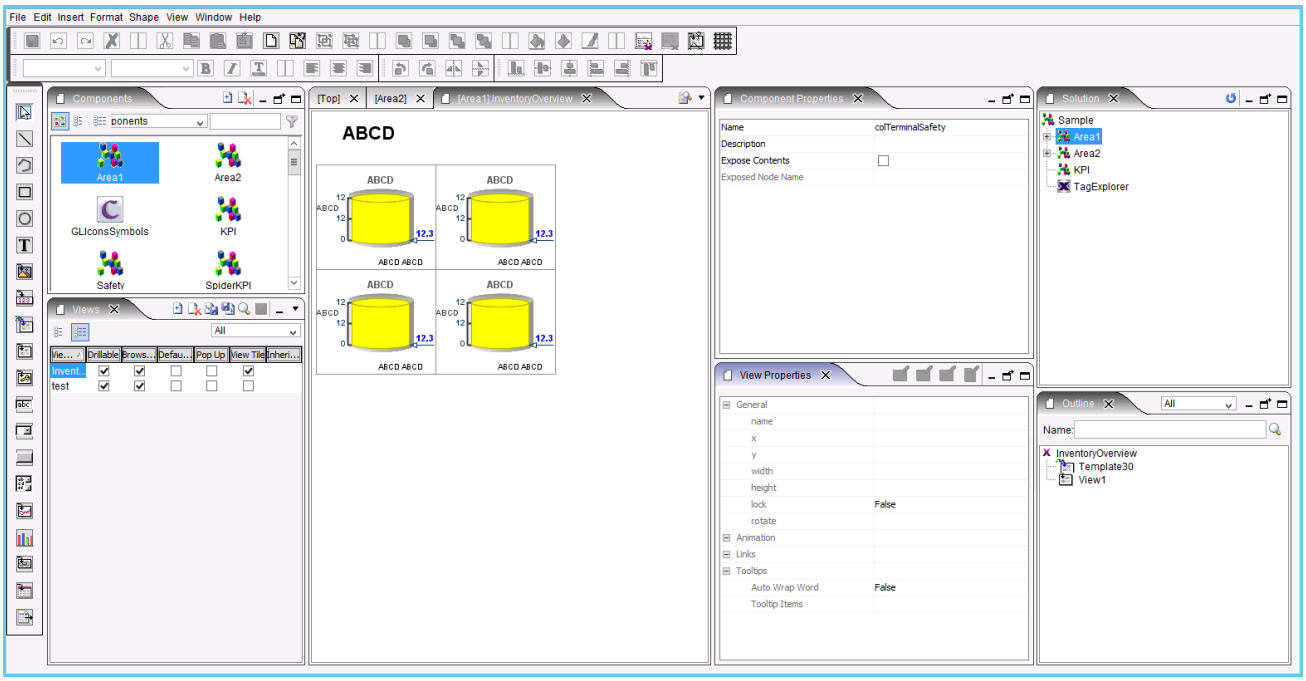
- Browsing is disabled for `tst3`, `tst4`, and `view1`.
- The views, `view1` and `View2`, are inherited, and therefore are grayed-out.
- The asterisk indicates that `view1` is inherited from a base component.

This results in the following **Edit View** submenu:


Assign Roles...
Set Page...
Edit Component
Rename Component...
Delete Component
New View...
Edit View
Import View...
Delete View
Generate Component Summary...
tst2
tst1
view10
View2
tst3
tst4
view1

Customizing Layouts

The XHQ Workbench interface consists of moveable panels and toolbars that may be docked in different configurations (or layouts), allowing you to customize your workspace.

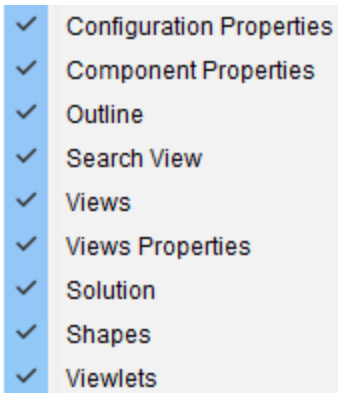


XHQ Workbench - Customized Layout

Moving and Docking Panels and Toolbars

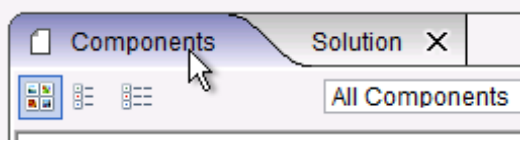
The XHQ Workbench interface consists of moveable panels and toolbars.

To enable dockable panels, go to the **View menu** and select **Show Dockable**. Check which of the following dockable panels to enable.



Dockable Panels

To move a panel, click the **tab**.

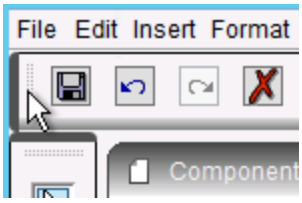


Cursor on Panel Tab

Hold and drag it to a new location. Notice the **blue** outlined rectangle that appears as you drag it around the workspace. This indicates a place where you can dock the element. You simply drag the panel to the new position and drop it.

You may move a dockable panel outside of the XHQ Workbench.

Moving a toolbar works the same way. Click the toolbar **title bar** (on a horizontal toolbar, it's the **two dotted lines** located at the left edge; on the vertical toolbar, it's located at the top edge).



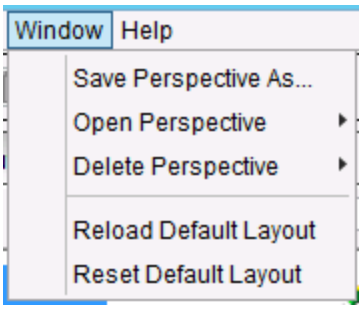
Cursor on Toolbar Title Bar

Hold and drag the toolbar. Notice the **red** bar that appears over allowable positions. Drop the toolbar into the new position and it snaps into place.

Managing Layouts

You can save different layouts (workspace configurations) and use them for different tasks. You can set the default layout so it opens automatically when XHQ Workbench launches. You can also reload the default layout back to factory settings (the original layout when you first launched the XHQ Workbench.

Go to the **Window menu** to manage your layouts.



XHQ Workbench Window Menu

To save a customized layout

1. Configure your workspace.
2. From the **Window menu**, click **Save Layout As**.

To set a different default layout

This layout opens automatically when XHQ Workbench launches.

From the **Window menu**, click **Reload Default Layout**.

To switch layouts

From the **Window menu**, click **Open Layout** and select another layout.

To revert back to the factory default layout

From the **Window menu**, click **Reset Default Layout**.

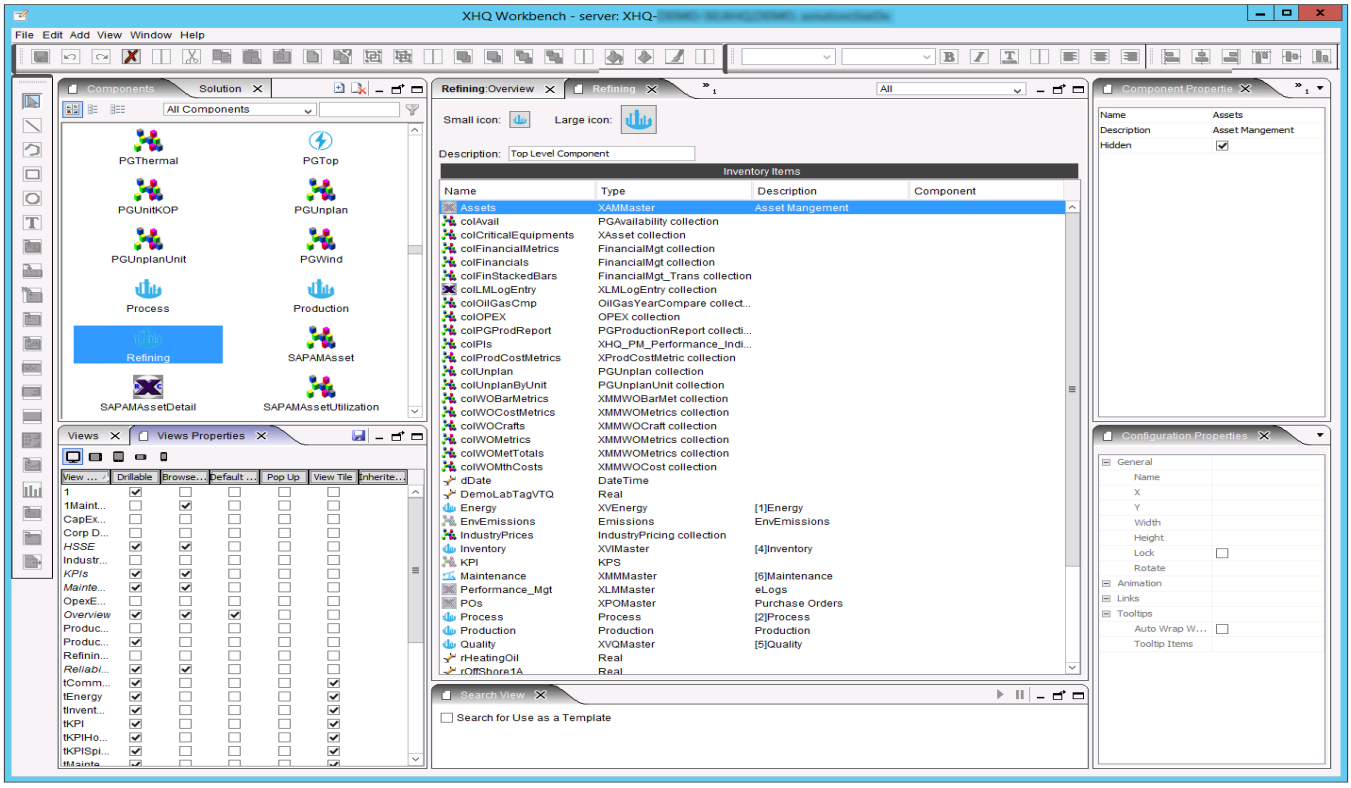
This returns the workspace configuration to the original layout when you first launched the XHQ Workbench.

To delete a layout

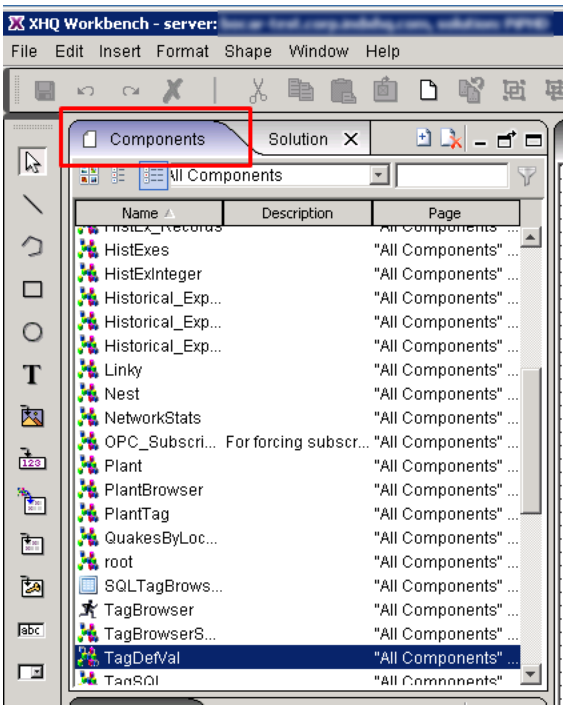
1. From the **Window menu**, click **Delete Layout** and select the layout you want to delete.
2. Click **OK** to confirm deletion.

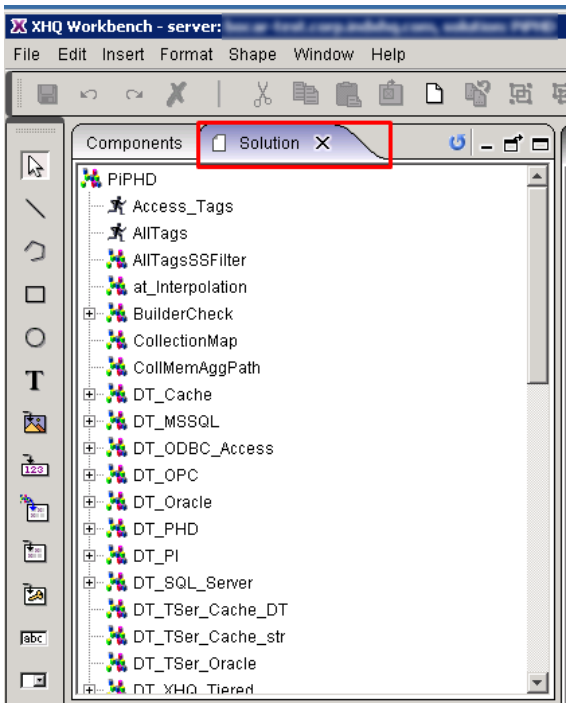
Working with Tabs

The XHQ Workbench consists of the several interface items, the primary being the tabs.



When the XHQ Workbench starts up, the **Components** and **Solution** tabs appears on the left.





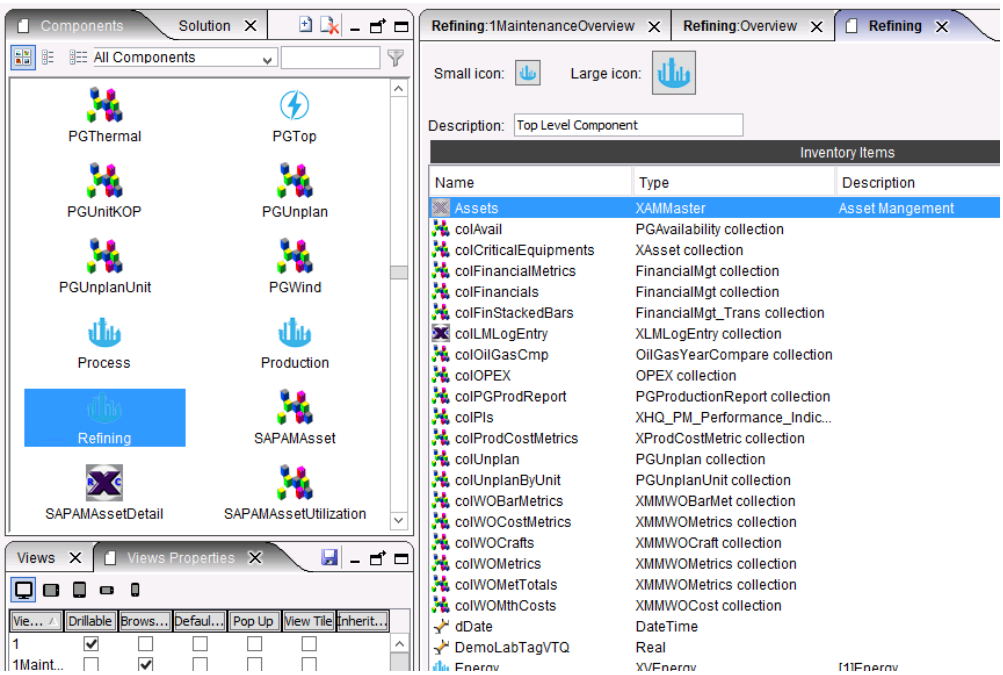
Currently, the Solution tab only displays the model hierarchy and is not yet fully functional.

You may execute commands such as building components from the XHQ Workbench through the Menu bar or the right-click shortcut menu.

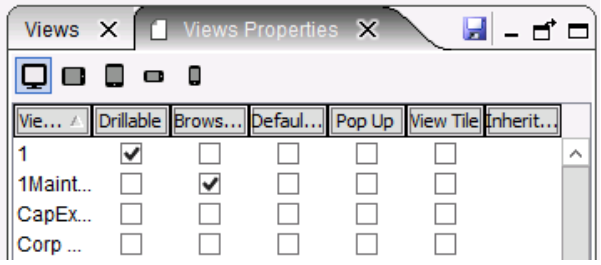
The Components tab and the Solution tab are integrated. So if you select a component in the Components tab, the first instance of that component is also selected in the Solution tab.

Use the search filter in the Components tab to help you find a component.

When you double-click a component in the Components tab, the corresponding component member tab appears in the main panel, which is to the right of the Components tab.

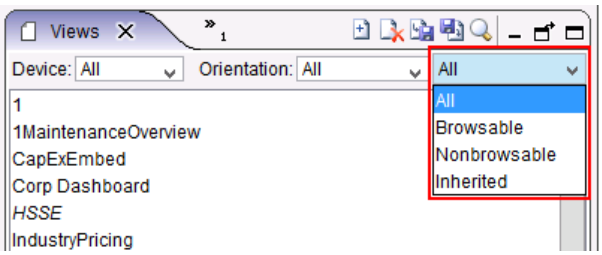


You can click on any component in the Components tab and all the views related with that component appears in the **Views tab** , which is docked below the Components tab.

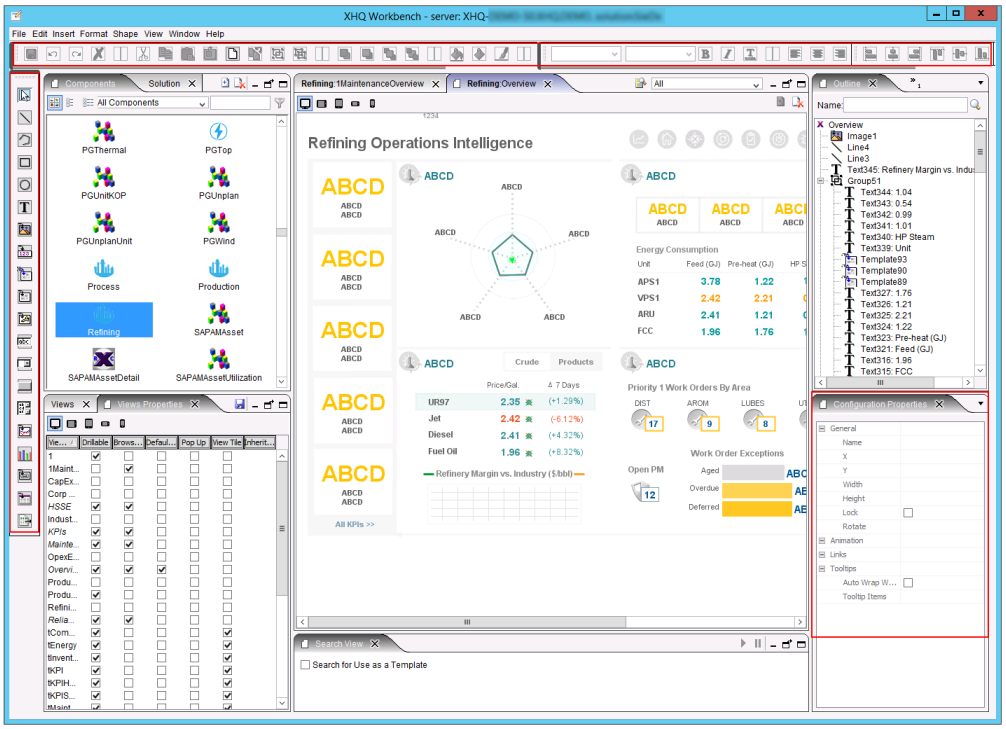


When you double-click a view in the Views tab, the view appears as a tabbed item in the main panel.

Views can be filtered by Browsable, Nonbrowsable and Inherited

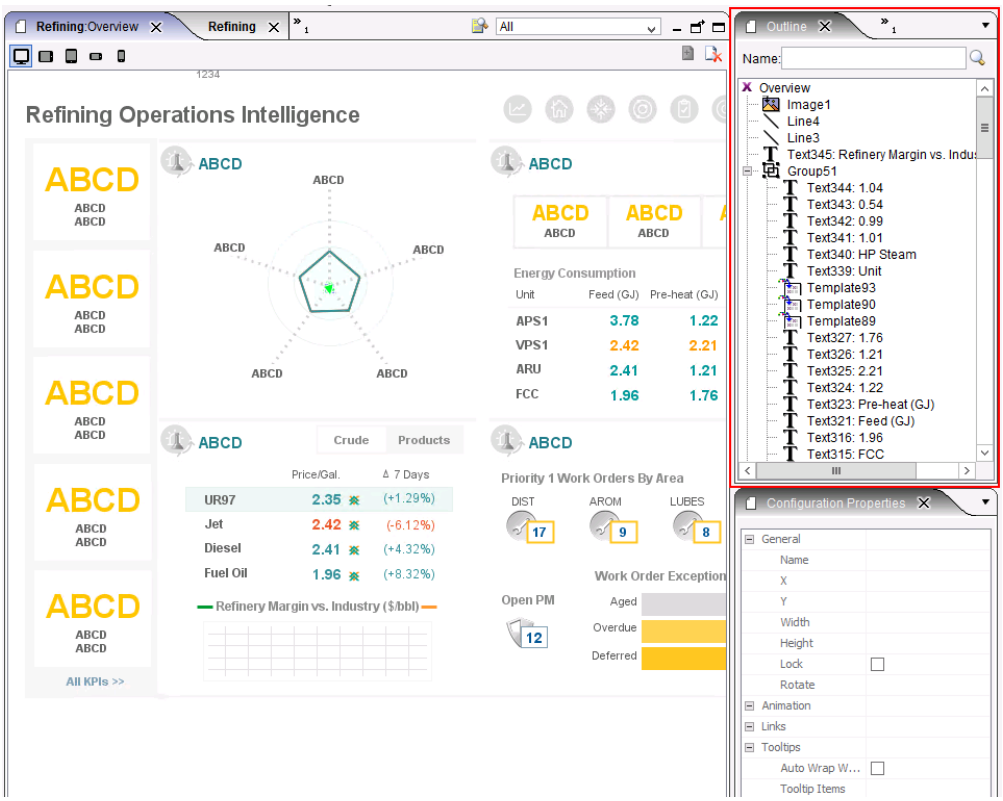


You can edit the active view by using the main and right-click menus and the toolbar and tools palette. The **Configuration Properties** tab, located in the lower right-hand corner, enables you to set properties for the given view.



For more information on the **Shapes tab** and **Viewlets tab**, go to the topic, *Working with the XHQ Visual Tile Composer*.

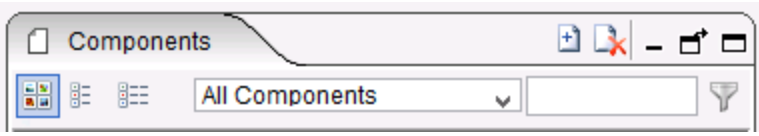
The **Outline** tab, which is docked above the Configuration Properties tab, lists the view elements. Viewing the elements in this hierarchical manner is helpful in complex views. You can change the order in which the view elements appear by simply dragging an element and dropping it in the desired location within the hierarchy.



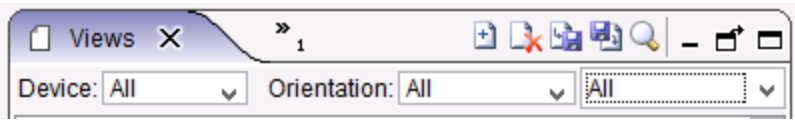
Tab Tools

Tab Tools, or *Panel Editors*, typically appear on the upper left corner of each panel.

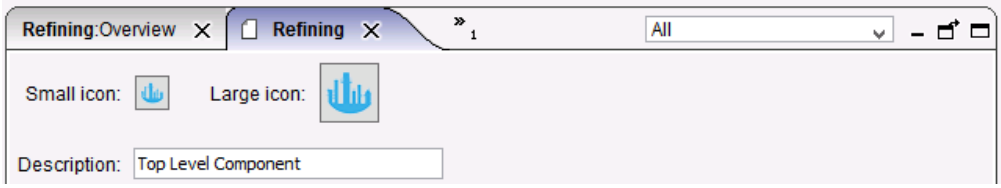
Each tabbed item has a corresponding set of tools



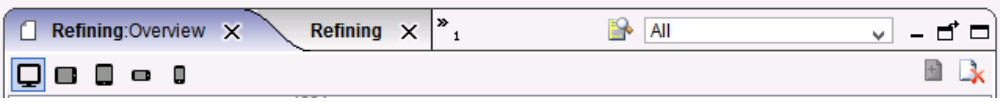
Components Tab



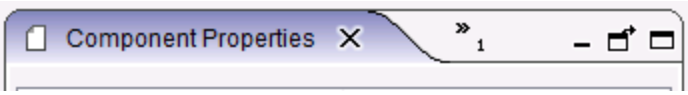
Views Tab



Component Details Tab



View Details Tab



Component Properties Tab

Outline Tab

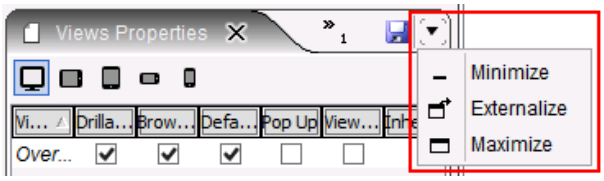
Views Properties Tab

Search View Tab

Each tab has the following tools in common.

–	Collapses the panel. To expand back to the default position, click .
	Undocks the panel, causing it to float over the main interface. To dock the panel back to the default position, click .
	Maximizes the panel across the entire interface. To return to the default position and size, click .

If the panel sizes are too narrow, these tools are collapsed and can be accessed by clicking the down arrow.



4 | Building and Managing Components

Creating and Editing Components

You use the XHQ Workbench to create new components. This section focuses on simple create and edit tasks, as well as more sophisticated editing functions, such as adding or deleting different types of members.

You must have Component Editing rights to create/edit/delete components.

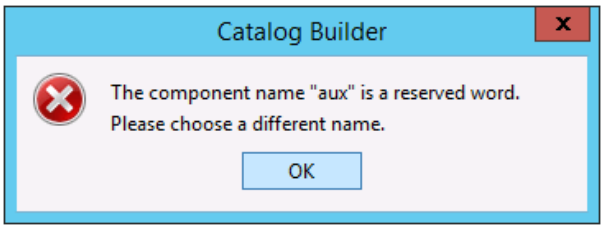
For more information, refer to the topic, *Security, Access, and Privileges*, located in the XHQ Administrator's Guide.

How to name components

You can use letters, numbers, underscores, and most UTF-8 characters in your component name. The component name can begin with either a letter or a number.

Component names (as well as component descriptions) are limited to 256 characters.

Avoid using *reserved words* as component, component member and view names. These are words that have a specific meaning to the XHQ System. If you use a reserved word, you will receive a notification error asking you to choose a different name.



Reserved Word Error Message

For a list of reserved words, see the topic, *Reserved Words*.

About Base and Subclass Components

XHQ allows one (or more) level(s) of inheritance, whereby you can define a new component based on an existing component. This technique of extending an existing component to define a new component is called *component subclassing*. The existing component is called the **base component**. The new component is called the **subclass component**. This subclassing technique is powerful in that it allows the reuse of the base component across solutions and thus, in effect, makes the base component portable or 'constant' across solutions. The subclass component takes on (or inherits) the members and views of the base component and adds to it its own members and views. The subclass component *extends* the base component.

You may subclass up to 8 levels. For details, see the topic, *Multiple Levels of Subclassing*.

Another way to look at component subclassing is that it divides an actual component into a "fixed" part and a "variable" part. The "fixed" part remains unchanged across solutions, but the "variable" part changes from solution to solution. A subclass component implemented in a solution will, therefore, consist of:

- The base component (the "fixed" part);
- Plus the additional members and views that only apply to the local solution (the "variable" part).

From the local solution, the base component, including its members and associated views, is not editable. However, the members and views that are added locally (the "variable" part) are editable.

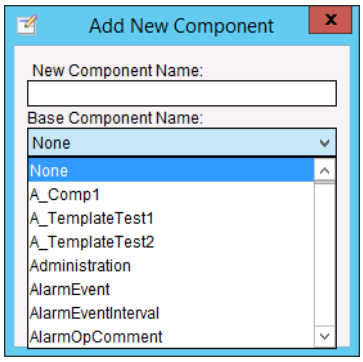
For example:

> All sites use the **`Name`**, **`Description`**, and **`CurrentValue`** members in a core tag component. Only a few of the sites use the **`Units`**, **`HighLimit`**, and **`LowLimit`** members. In this case, the base or core tag component will have the **`Name`**, **`Description`**, and **`CurrentValue`** members and site tag components can then extend the core tag to include some or all of the other members.

Things to note about Component Subclasses and XHQ

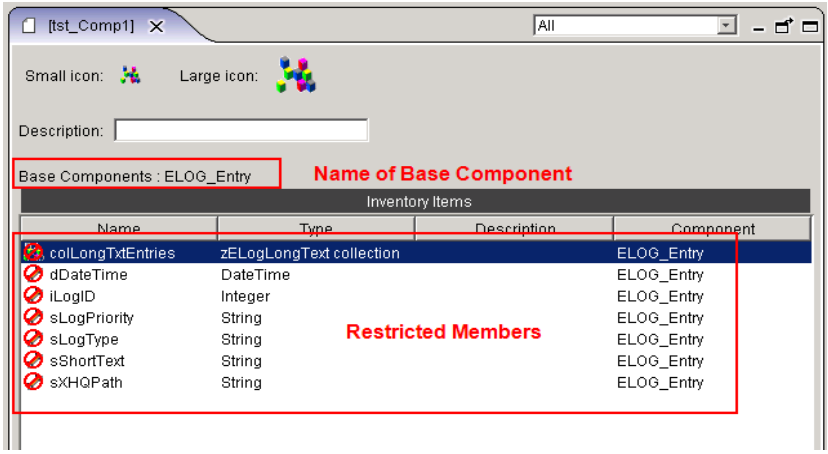
Here are some rules to consider when using Component Subclassing within XHQ.

- You can only define a subclass component during component creation when the component does not have any members. Therefore, when creating a subclass component, only components that are not subclasses themselves will appear on the list of base components that you can choose from.



Example: List of available base components

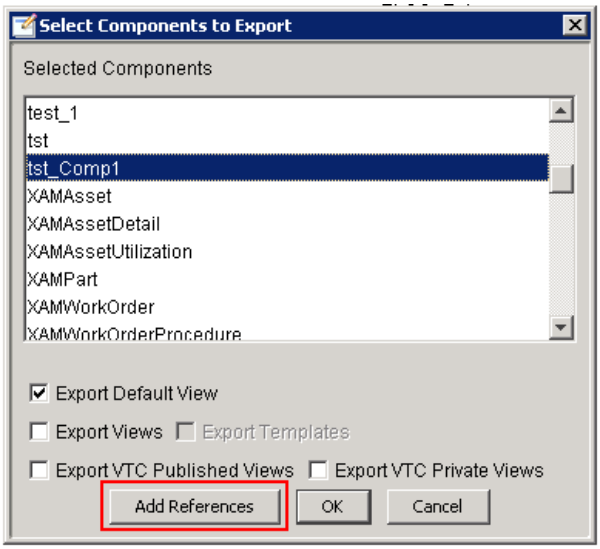
- Once you define a subclass, you cannot reassign or delete its association (link) to the base component.
- Any changes to the members or views of the base component are automatically applied to all of its subclass components.
- You cannot delete a base component until all of its subclass components have been deleted.
- Renaming either the base component or the subclass component is allowed and does not break the "subclass to base" association (link).
- In the XHQ Workbench > Component Detail tab, a red circle icon next to a member (Inventory Item) name means that it cannot be edited or deleted. It is a *restricted* member. These members have been inherited from a given base component.



- The configuration of a restricted member will not be displayed on the Component Properties tab.
- It is important to note that views and members inherited from the base component cannot be modified or deleted. Therefore, in a subclass component, you can only create, delete, and modify its members and views if they have not been inherited from the base component.

Specific rules also apply when you are **exporting** or **importing** a component.

- If you are exporting a subclass component, clicking "Add References" adds the base component. However, if you are exporting a base component, "Add References" does not add the subclass component(s).



- A component defined as a subclass cannot be changed to extend a different component or changed to a base component.
- Subclasses may be defined before base classes.
- Subclasses may extend exactly ONE component.
- Subclass member and view name must be different from their base class member and view names.

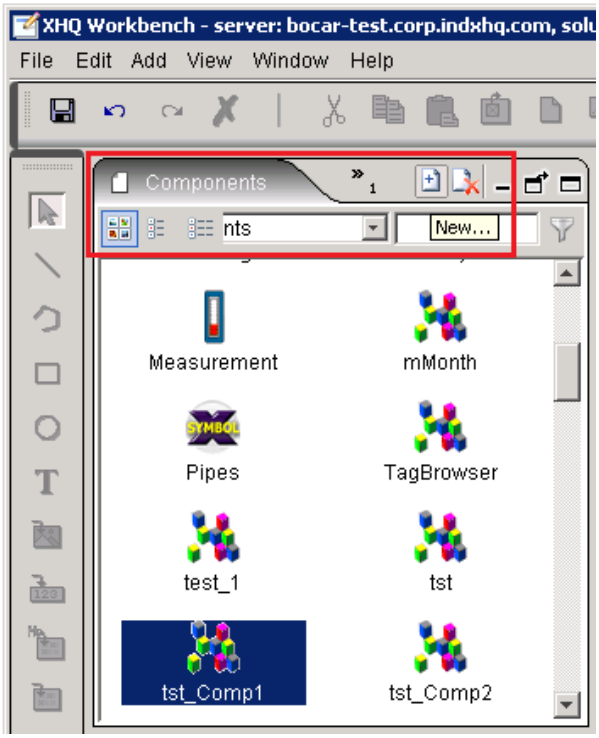
For more information on this topic, go to the section, *Exporting and Importing*.

To create a [base] component

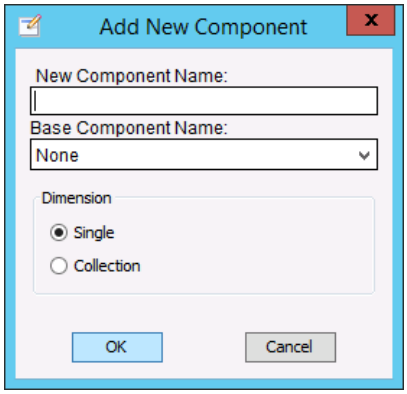
You must have Component Editing rights to create/edit/delete components.

For more information, refer to the topic, *Security, Access, and Privileges*, located in the XHQ Administrator's Guide.

1. Launch the **XHQ Workbench**.
2. Next to the **Components** tab, click the **New** icon.



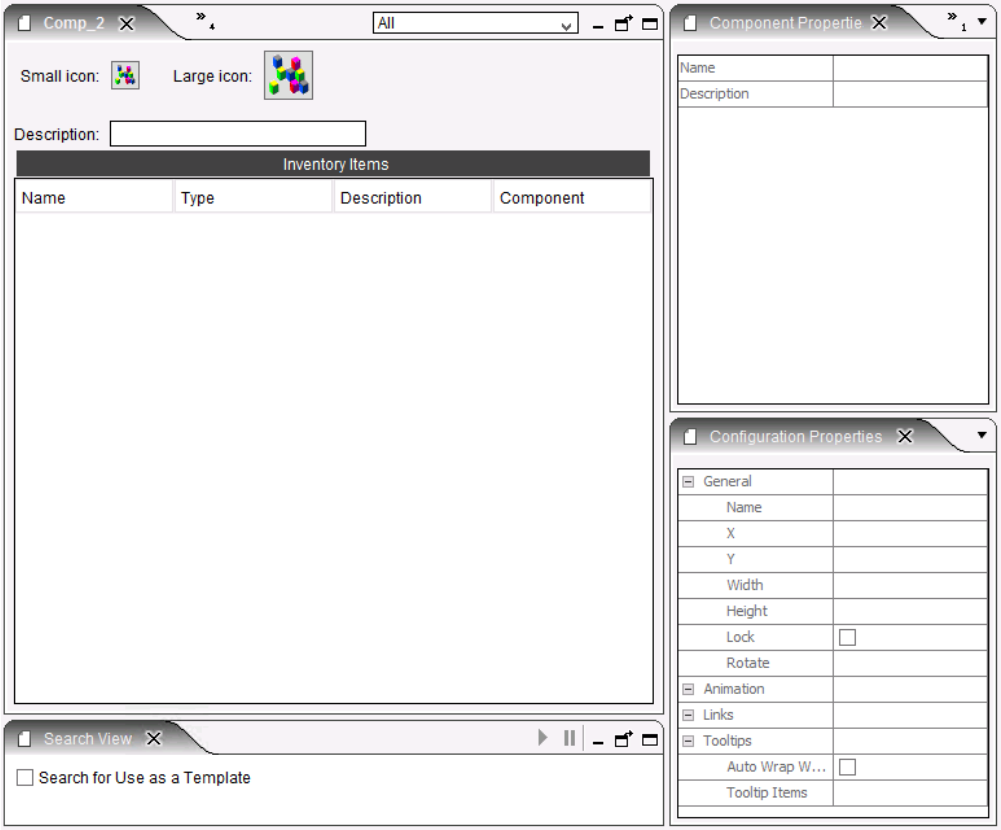
The "Add New Component" dialog appears.



3. In the **New Component Name** box, enter your unique component name.

   For more information on this task, go to the topic, *How to name components*.

4. In the **Base Component Name** box, select **None**.
5. Click **OK**.
   The Component Detail tab appears.



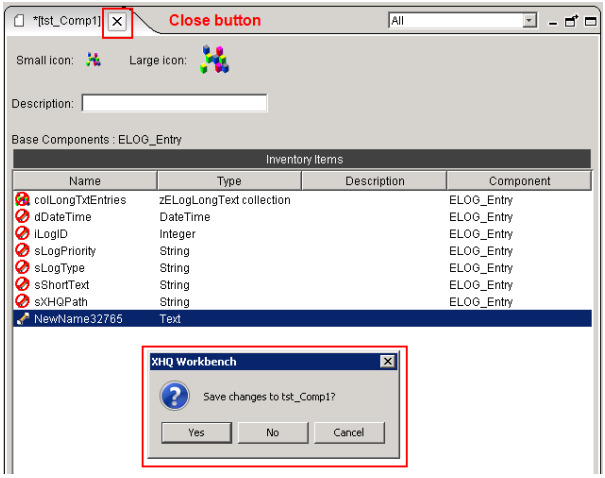
6. In the **Description** field, enter a description for the component.
7. Next, add component members by right-clicking on the **Add** menu.

   For more information on component members, go to the topic, *Working with Members*.

8. To save this component record, do one of the following:
   - Click **Save**, and then close the window.
     The component is saved to the repository.

   or

   - Click the **Close** [X] button at the upper right corner of the window. A message appears asking you to save your changes. Click **Yes**.


*[tst_Comp1]
Close button
All
Small icon:
Large icon:
Description:
Base Components : ELOG_Entry
Inventory Items
Name
Type
Description
Component
colLongTxtEntries
zELogLongText collection
ELOG_Entry
dDateTime
DateTime
ELOG_Entry
iLogID
Integer
ELOG_Entry
sLogPriority
String
ELOG_Entry
sLogType
String
ELOG_Entry
sShortText
String
ELOG_Entry
sXHQPath
String
ELOG_Entry
NewName32765
Text
XHQ Workbench
Save changes to tst_Comp1?
Yes
No
Cancel

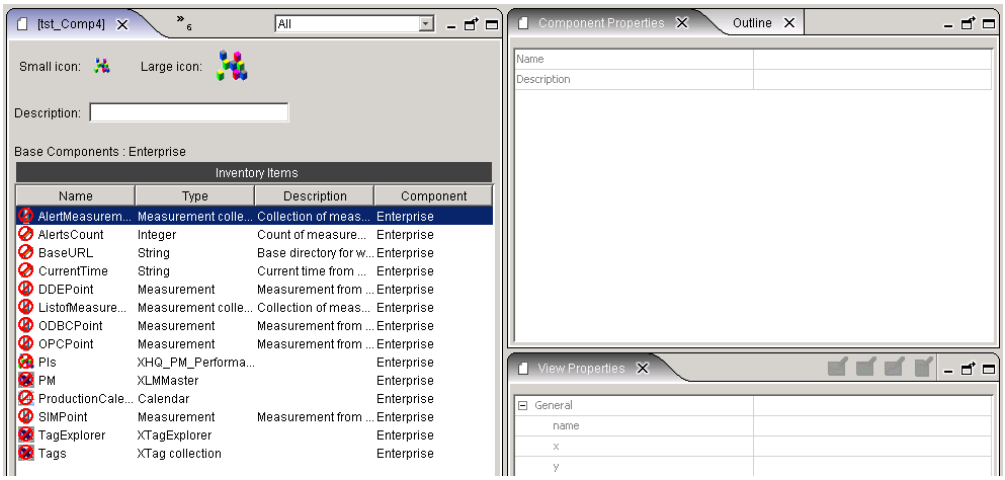
To create a subclass component

1. Launch the **XHQ Workbench**.
2. From the **File menu**, click **New Component**.
   The "Add New Component" dialog appears.
3. In the **New Component Name** box, enter your unique component name.
4. In the **Base Component Name** box, browse and select the base component.

   For more information on base and subclass components, go to the topic, *About Base and Subclass Components*.

5. Click **OK**.
   The Component Details tab appears.

   The Component Details tab lists the members that have been inherited from the base component. The red circle icon next to each member means that it cannot be edited.



6. In the **Component Details tab**, enter a **Description** for the component.
7. Next, add component members by right-clicking anywhere within the Component Details tab and select a member type to add.

   For more information on component members, go to the topic, *Working with Members*.

8. To save this component record, do one of the following:
   - Click the **Close** [X] button located next to the component name on the Component Details tab. A message appears asking you to save your changes. Click **Yes**.

or

   - From the **File** menu, click **Save**.
     The component is saved to the repository.

To edit a component

You must have Component Editing rights to create/edit/delete components.

For more information, refer to the topic, *Security, Access, and Privileges*, located in the XHQ Administrator's Guide.

1. From the **XHQ Workbench** > **Components tab**, right-click the component icon, and then, on the shortcut menu, click **Edit Component**.
   The Component Details tab appears, displaying the structural information for the component you selected in the XHQ Workbench.
2. **Edit** the component.

   If this is a subclass component, you cannot modify any members that have been inherited from the base component.

3. To save your changes, click the **Close** [X] button at the upper right corner of the window.
   A message appears asking you to save your changes.
4. Click **Yes**.

Multiple Levels of Subclassing

You may subclass **up to 8 levels**. Here are some things to note when using multiple levels of subclassing.

- By default, the **Allow Subclassing** checkbox is not enabled in the "Add New Component" dialog box. To enable, set the `ShowAllowSubclassingFlag` global setting property to true. If Allow Subclassing is not enabled, only inheritable components are created. In this case, the only way to create a non-inheritable component is through an import.
- From the "Add New Component" dialog box, if the **Base Component Name** is set "None", then the new component you create does not inherit members of another component. If you select a Base Component, then the component you are creating uses this base component as its parent.
- When **Allow Subclassing** is checked, then you are allowing the component you are creating to be used as a parent. In other words, it can be used as a base component for new components.
- A component cannot be deleted if it is referenced as a base component by another component.

About Inheritance

Because you can create a new component based on an existing component, the new component inherits all the members and the views of the existing base component, as well as all other member and views of component(s) that are levels above this existing base component. Note, however, if a component already has 8 levels of inheritance, it cannot be used as a base component and will not be included in the "Base Component Name" list of the "Add New Component" dialog box.

The "Base Component Name" list displays the inheritance chain, from the most immediate to the most remote component. And in the Component Details tab, the "Component" column of the "Inventory Items" table displays the name of the component from which the member is inherited.

If a component is inherited from a base component that has no views (and, therefore, has no Default View set), then a Default View for this inherited component must be explicitly set. By default, an inherited component of any level has no explicit Default View. Only a base level component automatically sets the Default View flag on its first non-flow view. For inherited components, the Default View "trickles" down the inheritance chain until either of the following occurs:

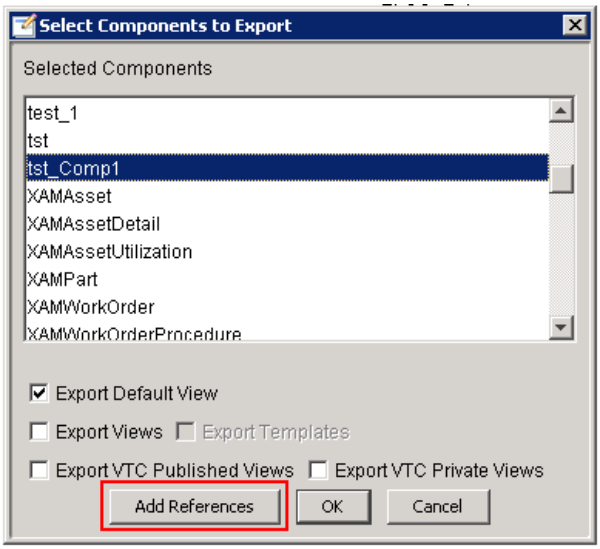
- A component in the chain has the Default View explicitly set
- The base component is reached.

However, if the base component has no views (or no non-flow views), then the components immediately inherited from the base component have no Default View. In this case, the Default View must be explicitly set.

Only non-flow views may be a Default View (flow views cannot be a Default View).

About Export/Import

When you export a component that has several levels of inheritance, the export contains all the properties, members, and views created for the component. In addition, it contains a list of components involved in the inheritance chain but does not include inherited members and views. To include inherited members and view, you must click "Add References."



When you import a component with an inheritance chain, a check is made to see if all base components exist in the solution. Should a base component not exist, then the appropriate inheritance level is created, but it will have no direct members/views. To import components, in the inheritance chain, with associated members and views, you must click "Add References."

Renaming Components

When you rename a component, XHQ automatically updates every instance of that component with the new name. In addition, renaming either the base component or the subclass component is allowed and does not break the "subclass to base" association (link).

For more information, go to the topic, *About Base and Subclass Components*.

To rename an existing component globally

1. From the **XHQ Workbench** > **Components tab**, right-click on the component.
2. On the shortcut menu, click **Rename Component**.
   The "Rename Component" dialog box appears.
3. In the **New Name** box, enter a name.
4. Click **OK**.
   Every instance of the component is renamed to the new name.

Choosing Your Component Icons

There are two types of icons used to represent the component: small and large. These icons are used on the inventory panel of the XHQ Workbench, and the navigator of the XHQ Solution Builder and the XHQ Solution Viewer (browser client).

These are **not** Microsoft Windows ICO icon files. Animated GIFs are currently not supported.

Small icon: For a small icon, use a 16x16 pixels `.gif`, `.jpg/.jpeg`, or `.png` image. This icon will be used to represent the component (node) in the navigator tree.

Large icon: For a large icon, use a 32x32 pixels `.gif`, `.jpg/.jpeg`, or `.png` image. This icon will be used to represent the component in the XHQ Workbench.

You can use any standard icon-editing tool that will help you prepare explicit 16x16 and 32x32 `.gif`, `.jpg/.jpeg`, or `.png` icons.

You should store all your icons as individual files in a common directory. Use an icon file naming convention that indicates the name of the icon and its size.

Examples: `Pump16.gif` is a 16x16 pixels `.gif` icon.

`32_gear.jpg` is a 32x32 pixels `.jpg` icon.

After you select the first icon, the Select image window defaults to the same location when you add the second icon. The system maintains this default location for the duration of your working session. However, once you exit XHQ, you need to re-establish your default location.

Once you select the icons for a component, you can then add and configure other items to the component.

To assign an icon to a component

1. From the **XHQ Workbench** > **Components tab**, double-click the component icon.
   The Component Details tab appears.
2. From the **Component Details tab**, click either the **small icon** or the **large icon**.
   The "Select icon image" dialog box appears.
3. Browse to a `.gif`, `.jpg/.jpeg`, or `.png` file.
4. Select the image.
   A preview of the image is displayed.
5. Click **OK**.

Saving Your Component Changes

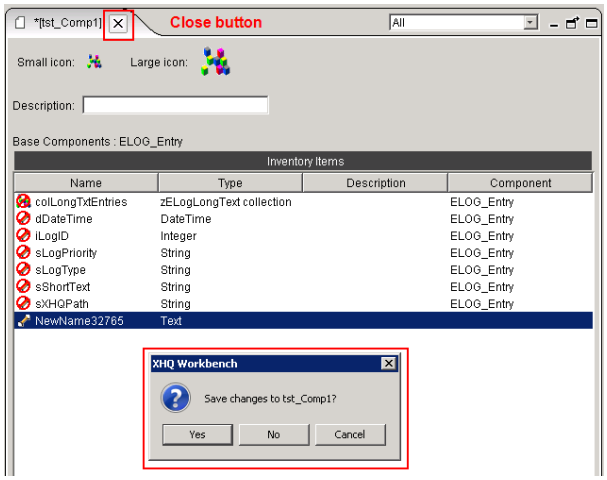
After you create or edit your components, you will need to save your component changes in the Component Details tab and then update your solution in the XHQ Solution Builder before any changes you make to components are reflected in the deployed solution. You simply click the **Save button** on the main toolbar.

To save component changes in the Component Details tab

1. From the **Component Details tab**, create or edit your component.
2. Do one of the following:
   - On the main toolbar, click **Save**.

   or

   - Click the **Close** button [X] located on the Component Details tab. A message appears, asking you to save your changes. Click **Yes**.



To update your solution to reflect component changes

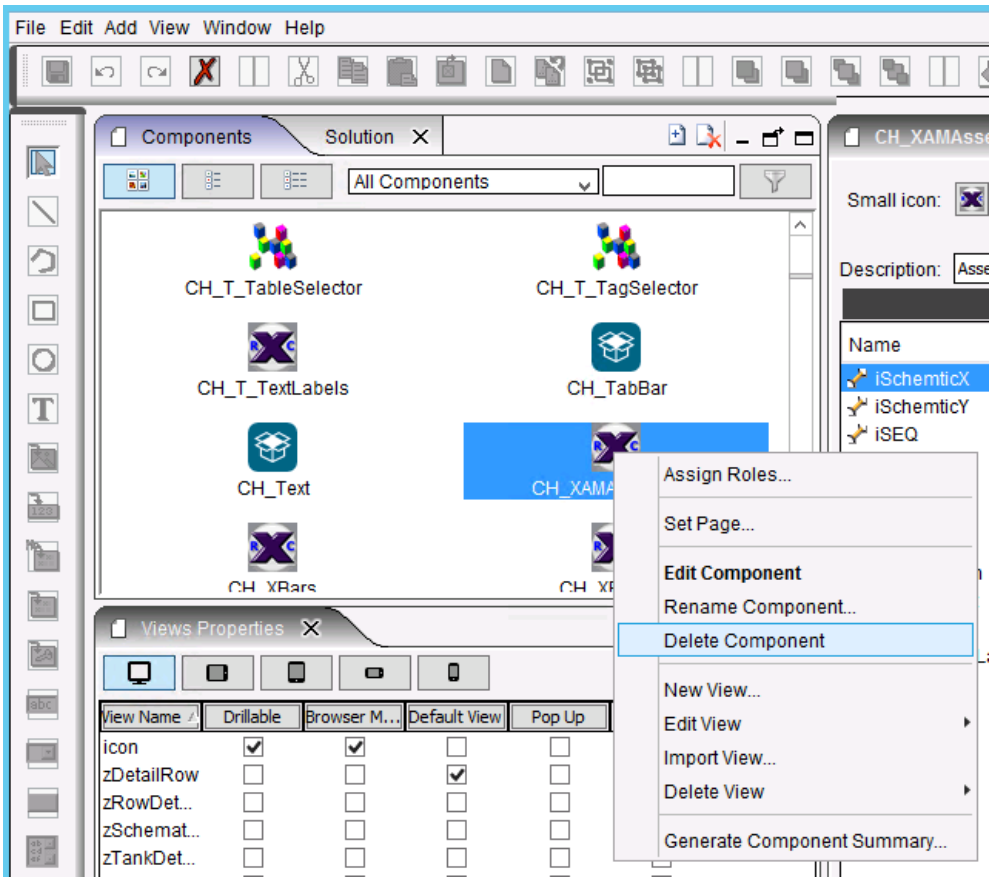
From the **XHQ Workbench**, click the **File menu**, and then click **Refresh Catalog**.
Any changes made to the components should now be reflected in your solution.

When you refresh the XHQ Workbench, you apply any pending changes from your current model and clear the XHQ Workbench cache (if it exists).

Deleting Components from the XHQ Workbench

This command, located on a component's right-click shortcut menu, allows you to permanently delete components from the XHQ Workbench.

You cannot delete a base component until all of its subclass components have been deleted.



Delete Component – Right-click Shortcut Menu Command

WARNING – Use this feature with caution.
Because of the potentially damaging effects of this command, only Solution Administrators with Component Editing rights are allowed to delete components from the XHQ Workbench. Therefore, security will need to be enabled.

In addition, make sure that you do **not** remove a top-level (root) component. This will make your solution unstable.

For more information, refer to the topic, *Security, Access, and Privileges*, located in the XHQ Administrator's Guide.

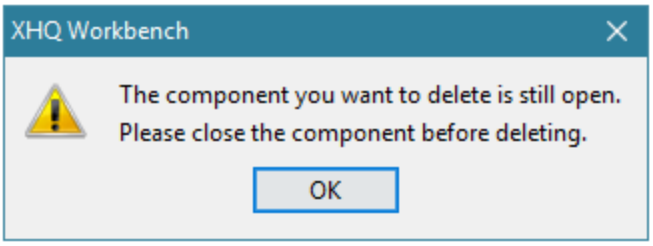
XHQ automatically removes all references to a deleted component. Therefore, you can delete a component that is embedded in other components.

To delete components from the XHQ Workbench

You must have Component Editing rights to create/edit/delete components.

For more information, refer to the topic, *Security, Access, and Privileges*, located in the XHQ Administrator's Guide.

1. Make sure the component you want to delete is not open, or the following prompt appears.

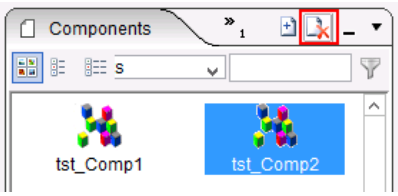


2. In the **Components tab**, do one of the following:

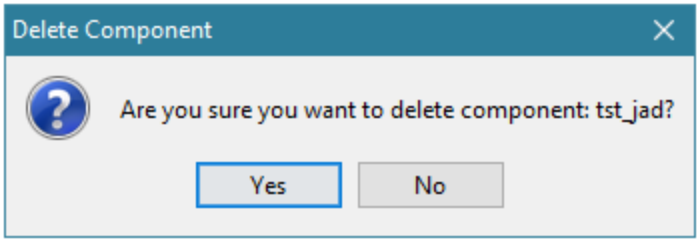
- Right-click on the component you want to delete and, from the shortcut menu, click **Delete Component**.

or

- From the **Component tab**, click the **Delete** icon.



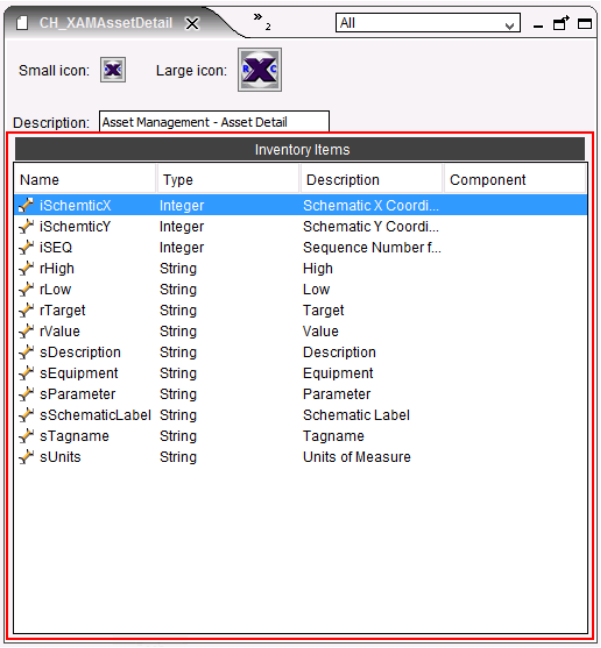
A message appears, asking you to confirm component removal.



3. Click **Yes**.
   The component is removed from the XHQ Workbench.

Working with Members

In XHQ, the items that make up a component are called *members*. A component's members – and associate name, type, and description – are displayed in the Inventory Items panel of the Component Details tab.



Component Details tab – Inventory Items Panel

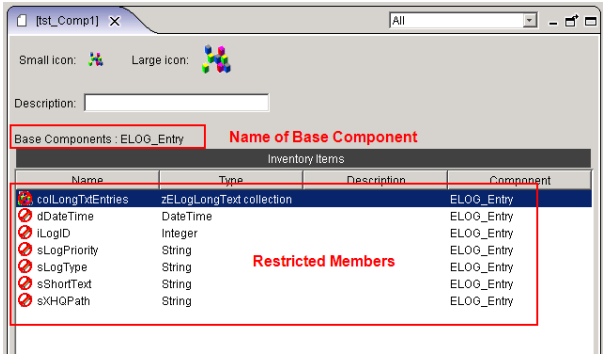
A member can either be a *primitive member* or another component. Every member that is added to a component must have a **unique name**. You can use letters, numbers, underscores, and most UTF-8 characters in the member name. The member name can begin with either a letter or a number.

Components with member names that begin with a number cannot be used as a collection. In addition, collection member names are limited to 30 characters or less.

Also, avoid using *reserved words* as member names. These are words that have a specific meaning to the XHQ system. If you use a reserved word, you will receive a notification error asking you to choose a different name.

For a list of reserved words, refer to the topic, *Reserved Words*.

In the Component Details tab, a red circle icon next to a member (inventory item) name means that it cannot be edited or deleted (cleared). It is a restricted member. These members have been inherited from a given base component. The configuration of a restricted member will not be displayed on the "Component Properties" tab.



Any changes to the members or views of the base component are automatically applied to all of its subclass components.

Defining Primitive Members

Primitive members hold values of variables and can be one of the following data types:

Data Type	Description
Boolean	An expression that results in a value of either TRUE or FALSE.
Integer	A 32-bit integer value.
Long	A signed, 64-bit (8-byte) stored integer.
Real	A 32-bit real value.
Double	A signed IEEE, 64-bit (8-byte), double-precision, floating-point stored number. The minimum stored value is limited to (-)1E-130. Any number less than this limit is stored as zero (0). The maximum stored value is limited to (-)9E125. Any number greater than this limit is stored as 9E125.
Decimal	A signed, 128-bit (16-byte) stored integer scaled by a variable power of 10. The scaling factor, which ranges from 0 to 28, specifies the number of digits to the right of the decimal point.
Date/Time	A IEEE, 64-bit (8-byte) stored integer, representing the following: Dates ranging from January 1 of the year 1 through December 31 of the year 9999; Times from 0:00:00 (midnight) through 11:59:59 PM.
Interval	A period of time, such as "3 years" or "90 days" or "5 minutes and 45 seconds." "YEAR-MONTH" contains one or both of the fields YEAR and MONTH. "DAY-TIME" contains one or more consecutive fields from the set DAY, HOUR, MINUTE, and SECOND.
String	A character field capable of storing up to 2000 characters.
Text	A character field capable of storing up to 2 billion characters of data.

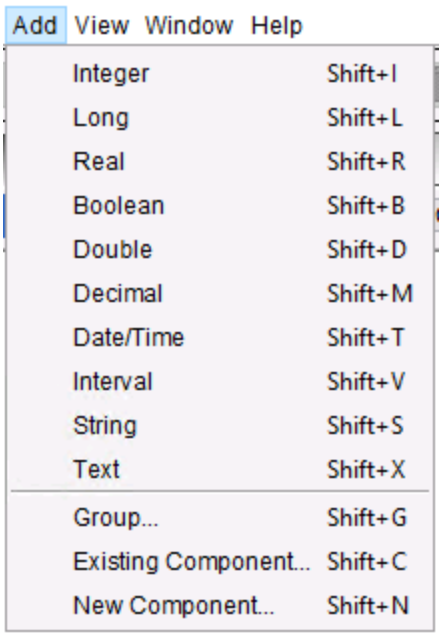
Adding Members

These are the types of members you can add to a component:

- **A Primitive member**
  Contains values of variables and can be any of the data types listed in the previous topic.
- **A Group**

  For details, see the topic, *Adding Group Members*.

- **An Existing component**
  Includes existing members, components, and collection components.
- **A New Component**
  Allows you to create a new component.

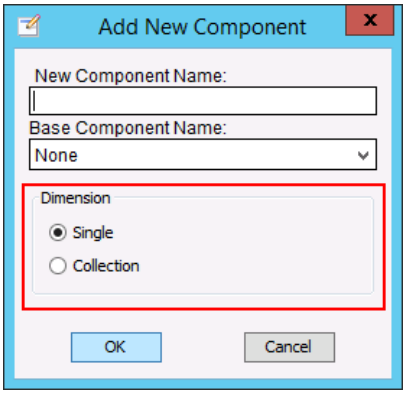


Add Menu Commands – Component Details tab

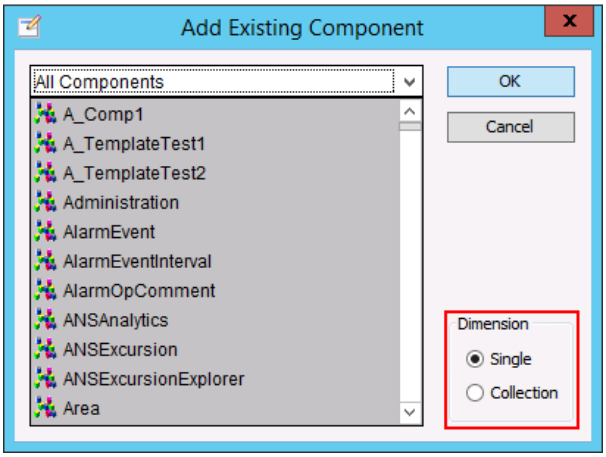
Assigning a dimension

If you want to add another component to the component you are working on, you must indicate its *dimension*. The dimension shows the relationship to the external data. A *single dimension* delivers one member from an external data source. A *collection dimension* delivers a varying number, depending on the contents of the external source at the time the collection is gathered. For example, in real time, a collection can map all current orders from a relational database into XHQ. The collection (member) size will change as the number of orders changes.

Dimensions are set for component members only and do not apply to primitive members. As such, you define the dimension type when you are either adding a new component or an existing one.

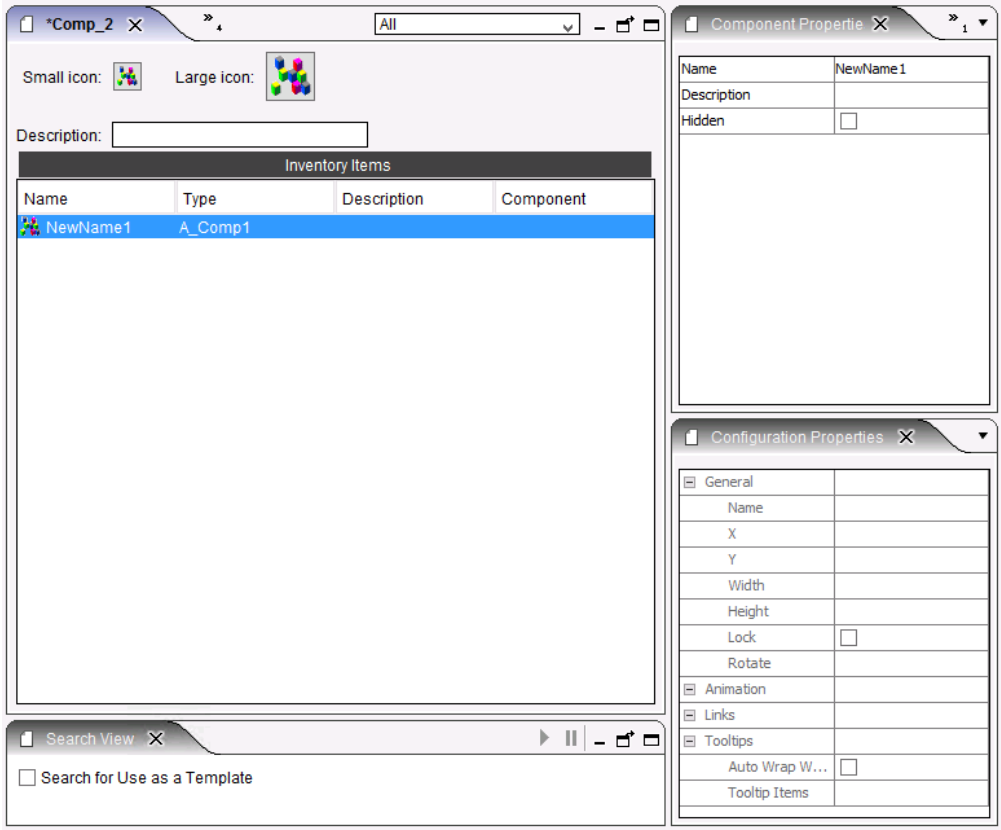


Dimension box for Add New Component dialog



Dimension box for Add Existing Component dialog

After adding an existing or new component, it appears in the inventory list with a default name. And because you have specified an instance of a component within another component, the name of the added component is presented in the Type field of the inventory items panel.



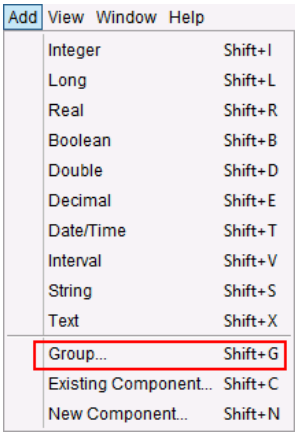
Adding an Existing Component

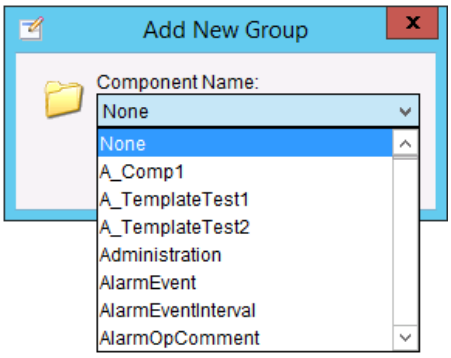
Adding Group Members

A Group member is a node in the XHQ Solution that can have a type (a component class). It may act as the parent node of other objects or groups and may be associated to other items. Groups of a defined type may only have component objects of the same type (or subclass) or other group children. That is, when a group has a type, it may only contain objects or groups of that type.

Groups with no type may have children of any type.

Adding a Group member is similar to adding a primitive member.





Component Details tab: Adding Group Member

To enable the **Group** option, you must set the `DisableExtendedNamespaceUI` global property to **false**.

For more information on this global property, go to the topic, *Additional Global Settings*, located in the XHQ Administrator's Guide.

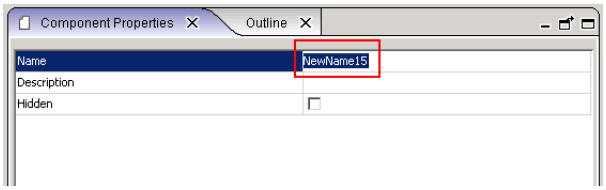
For information on how **security** works with group members, go to the topic, *About XHQ Security*, located in the XHQ Administrator's Guide.

To add an existing component member

1. From the **Component Details tab**, on the **Add** menu, click **Existing Component**.
   The "Add Existing Component" dialog box appears.
2. **Select** an existing component from the list box.
3. Under Dimension, click **Single** or **Collection**.

   For more information on this feature, go to the topic, *Assigning a dimension*.

4. Click **OK**.
   This returns you to the Component Details tab.
5. Under the **Component Properties** tab, in the **Name** box, enter your unique member name.



   Collection member names are limited to 30 characters or less.

6. In the **Description** box, enter a description for this member.
7. OPTIONAL
   Click (check) **Hidden** to set the visibility of the component member.

   For more information on this feature, go to the topic, *Hiding Members and De cluttering the Tree*.

8. On the **File** menu, click **Save**.

To add a new component member

1. Select the **Component Details tab** and do one of the following:
   - From the **Add** menu, click **New Component**.

   or

   - Right-click anywhere inside the tab and, from the shortcut menu, click **New Component**.

   The "Add New Component" dialog box appears.
2. In the **New Component Name** box, enter a name for the component.
3. OPTIONAL
   If this is a subclass component, select the **Base** component from the given list of available components.

   For more information on this feature, go to the topic, *About Base and Subclass Components*.

4. Under Dimension, click **Single** or **Collection**.

   For more information on this feature, go to the topic, *Assigning a dimension*.

5. Click **OK**.
   This returns you to the Component Details tab.

6. Under the **Component Properties** tab, in the **Name** box, enter your unique member name.
7. In the **Description** box, enter a description for this member.
8. OPTIONAL
   Click (check) **Hidden** to set the visibility of the component member.

   For more information on this feature, go to the topic, *Hiding Members and De cluttering the Tree*.

9. Skip the Tag setting, because tag attributes can only be set for primitive members.

   For more information on the tag attributes, go to the topic, *Setting the Tag Attribute*.

10. From the **File** menu, click **Save**.

To add a primitive member

Lựa chọn loại thành viên thích hợp,phù hợp: BOOLEAN, INTERGER, real, string

1. Select the **Component Details tab**.
2. Then, from the **Add** menu, select the appropriate member type: Boolean, Integer, Real, or String.
   The new member appears in the inventory list with a default name.

   Thành viên mới xuất hiện trong danh sách hàng tồn kho với tên mặc định.

2. Under the **Component Properties** tab, in the **Name** box, enter your unique member name.
3. In the **Description** box, enter a description for this member.
4. OPTIONAL
   Click (check) **Hidden** to set the visibility of the component member.

   For more information on this feature, go to the topic, *Hiding Members and De cluttering the Tree*.

5. OPTIONAL
   Select a **Tag** for the component member.

   For more information on this feature, go to the topic, *Setting the Tag Attribute*.

6. From the **File** menu, click **Save**.

To add more than one member of the same type

1. Select the **Component Details tab** and add a member.
2. From the **Edit** menu, click **Repeat last add**.
   This repeats your last add command, allowing you to add multiple members of the same type. The default member name increments by one for each repeated add.

To rename a component member

There are a couple of ways to rename a component member.

1. From the **Component Details tab**, locate the component member.
2. Do either of the following:
   - In the **Inventory Items list**, click inside the **Name** cell and edit.

   or

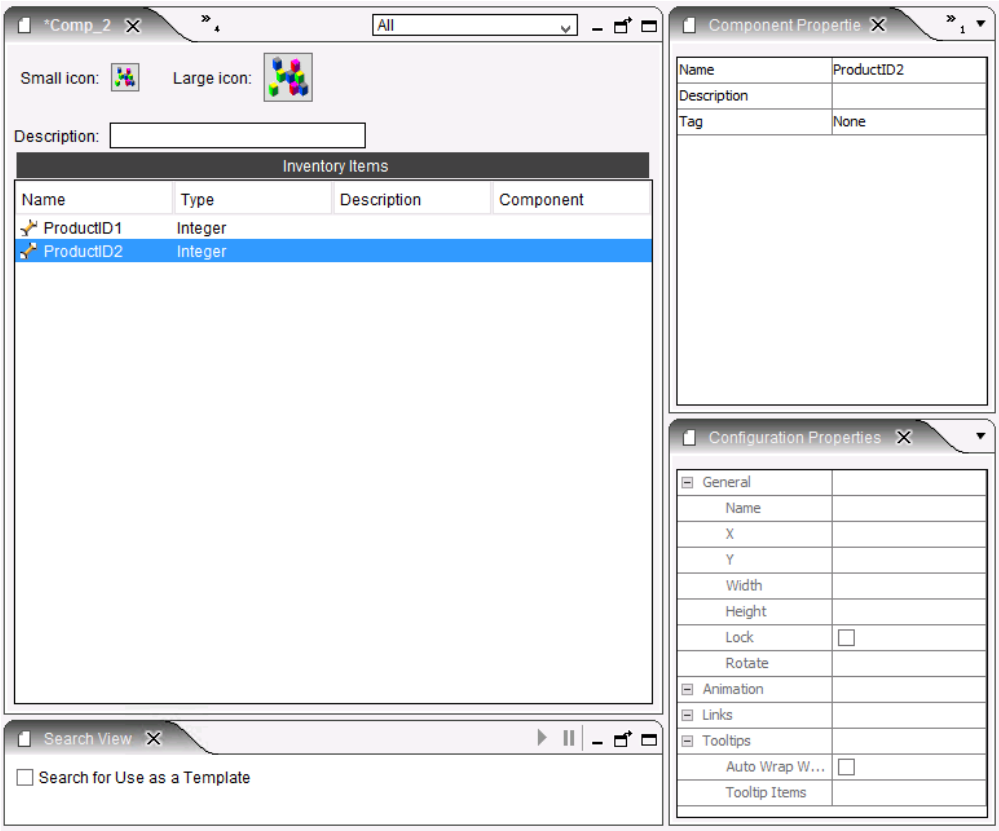
- Go to the **Component Properties tab**, click inside the **Name** cell and edit.

3. Press **Enter**.
4. *Optional*
   Edit the **Description**.

EXAMPLE: USING THE "REPEAT LAST ADD" COMMAND

From an existing list of components, we want to add two members of the Integer type.

1. Select the **Component Details tab** and from the **Add** menu, click **Integer**.
   The default member name generated is "NewName1."
2. In the **General** tab, enter the member name *ProductID1*.
3. On the **Edit** menu, click **Repeat last add**.
   Notice the default name generated is an incremented version of the member previously added. In this case, "ProductID2."

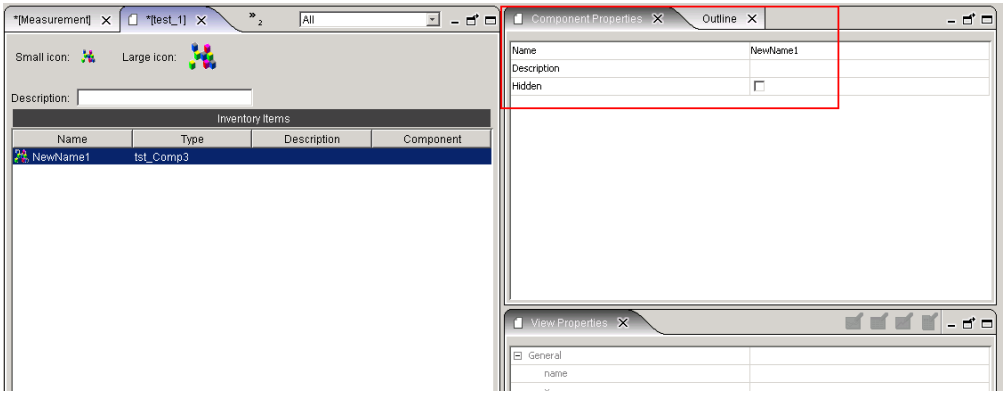


This command is helpful when adding members of the same type. You can also use this command to do multiple additions of existing or new members.

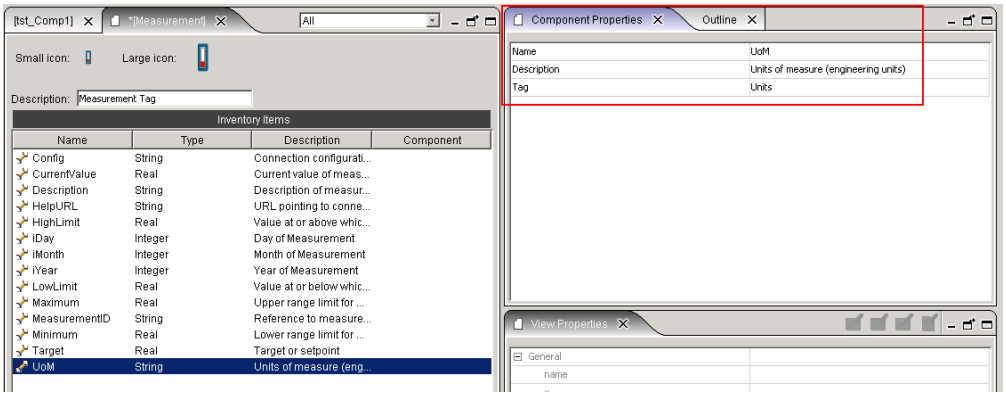
Configuring Members

Through the **Component Properties tab**, you can set the following attributes for each component member.

- Component Member Name
- Description
- Hide or Show Component
- Tag



Component Properties with Hidden Option

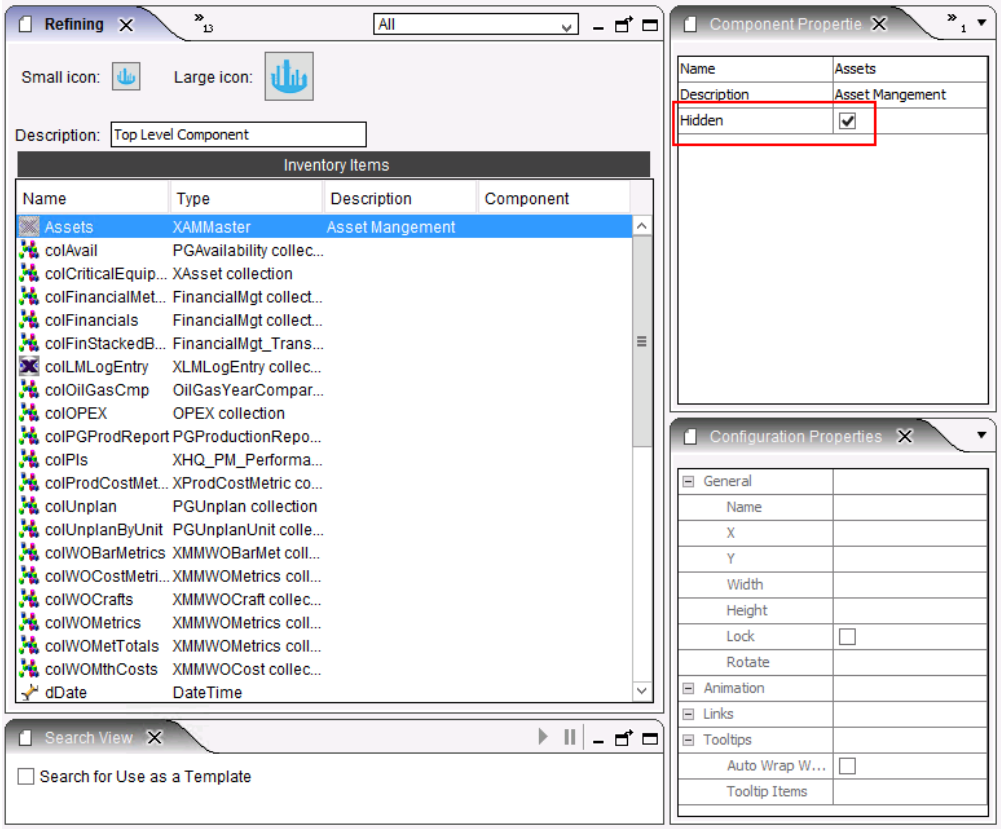


Component Properties with Tag Option

In this section, you will learn how to hide component members in the solution tree and how to set a tag attribute for a primary component member.

Hiding Members and De-cluttering the Tree

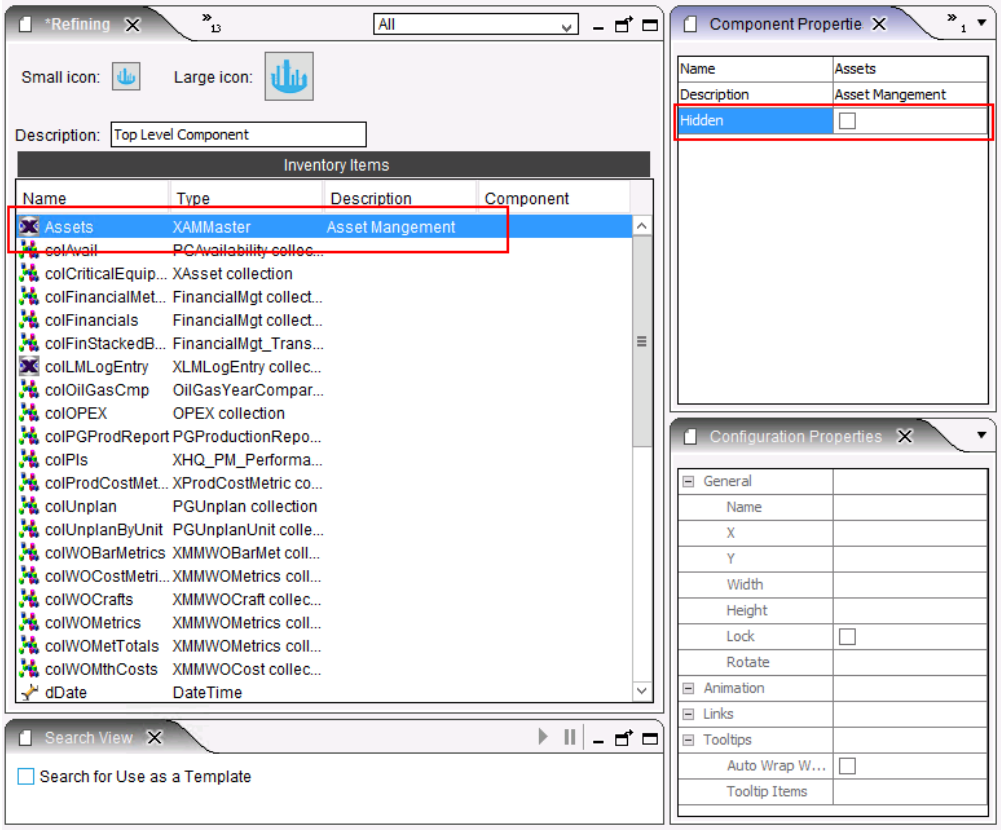
To reduce "clutter" in you solution tree, you can set the visibility of component members from the Component Properties tab.



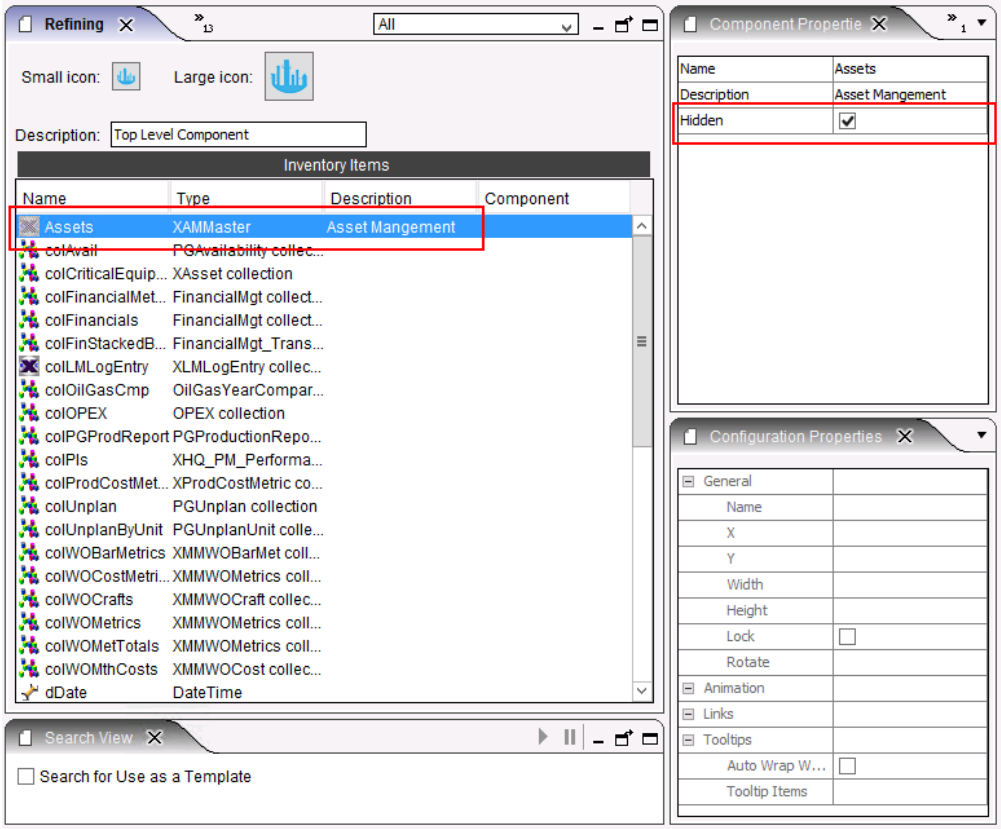
Component Details tab – Hidden Checkbox

Primitive members cannot be hidden.

When editing in the **Component Details tab**, this **Hidden** checkbox is available for all Inventory Item (component) members. If selected, the corresponding icon for the given component member is dimmed.



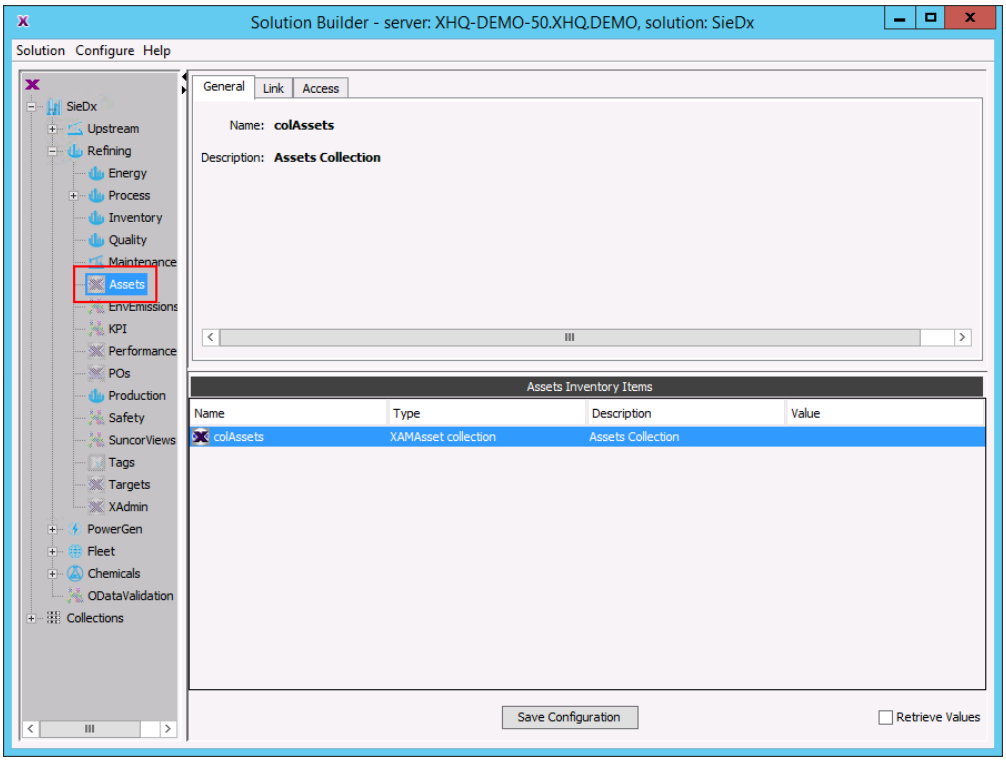
Non-Hidden Component Icon



Hidden Component Icon Dimmed

The **Repeat last add** command will copy the previous item added, including the Hidden state.

During runtime in the **XHQ Solution Builder**, a component member with **Hidden** set true is also displayed in the navigation tree with its icon dimmed. The "children" of the component member, however, still display their standard icons.



Hidden Component in XHQ Solution Builderr Navigation Tree

In a **global collection**, a hidden component member is displayed in the Inventory Items listing with its icon dimmed.

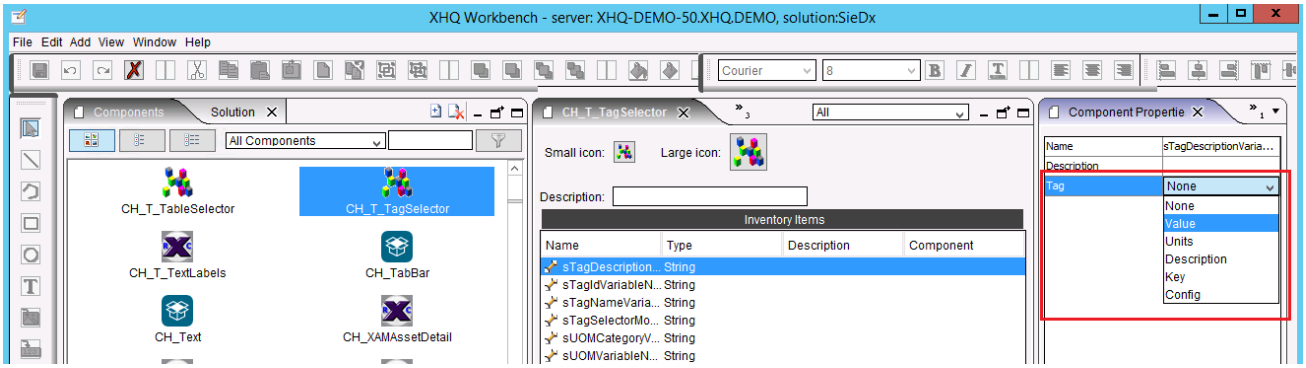
During runtime from the **XHQ Solution Viewer** (browser client), all component members (and their children) with **Hidden** set true are completely concealed in the navigation tree.

Setting the Tag Attribute

Tag configuration is done from the component level. Each **primitive** component member can be configured to be one of the following tag attributes:

- **None**
  This is the default. No tag attribute is applied to the component member.
- **Value**
  If Value is selected for a member, then that member is used as the default pen for trending.
- **Unit**
  If Unit is selected for a member, then that member is used to hold the tag unit.
- **Description**
  If Description is selected for a member, then that member is used to hold the tag's description string.

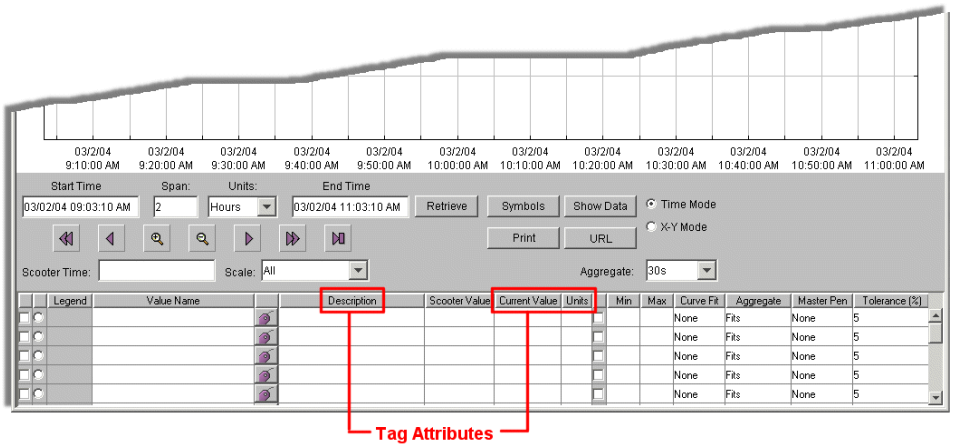
You configure the tag from the Component Properties tab.



Component Details tab – Tag Configuration

With the exception of the "Value" attribute, each tag attribute can be assigned to only one primitive member. For example, if the "Description" tag attribute is already assigned to member A and a user tries to reassign "Description" to member B, then member A's tag attribute resets to "None" and member B assumes the "Description" attribute. Again, only the "Value" attribute can be assigned to multiple members.

Once set from the XHQ Workbench, the given tag attribute is displayed on the XHQ Trend Viewer's (XHQ Interactive Trender) Legend Panel, as well as in the Tooltip that appears when you mouse over a tag value in the XHQ Solution Viewer.



Tag Attributes – XHQ Trend Viewer

Deleting Members

From the Inventory Items panel, you can delete component members. Deleting a component member removes its references in that particular component.

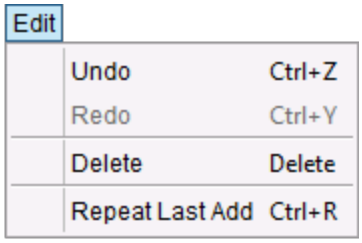
A red circle icon next to a member (inventory item) name means that it cannot be deleted. It is a *restricted* member. These members have been inherited from a given base component. The configuration of a restricted member will not be displayed on the Component Properties tab.

To delete a component member

1. From the **Component Details tab**, go to the list in the **Inventory Items** panel and click the member you want to delete.
2. Do one of the following:
   - From the main toolbar, click the Delete [X] button.

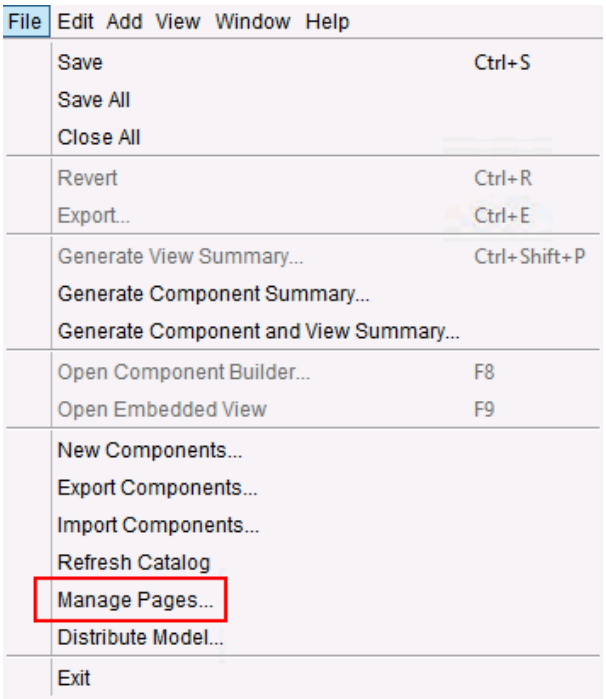
   or

   - From the **Edit** menu, click **Delete**.



Organizing the XHQ Workbench

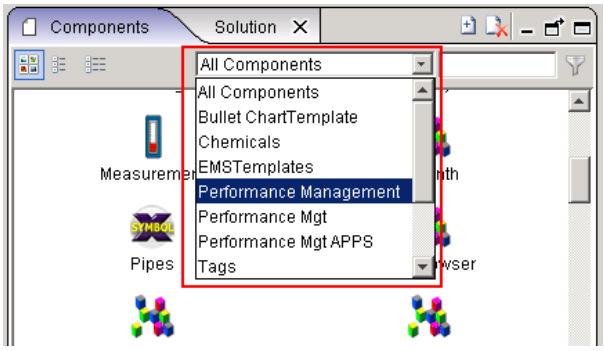
Through the XHQ Workbench, you can manage and organize large quantities of components by grouping related components into *pages*.



XHQ Workbench Menu Command to Manage Pages

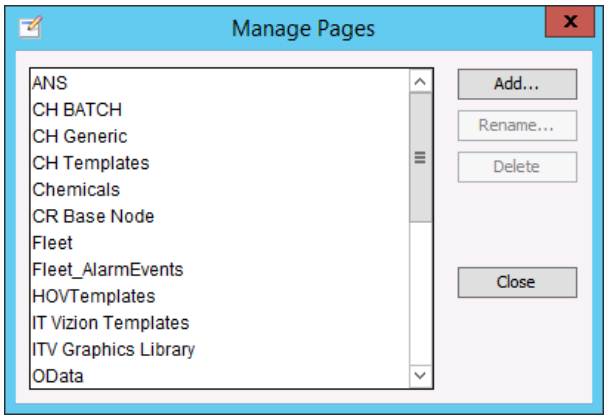
Pages are not hierarchical. You cannot contain pages within pages.

From the Components tab, you can select which page to display. The **All Components** page displays the entire catalog of components.

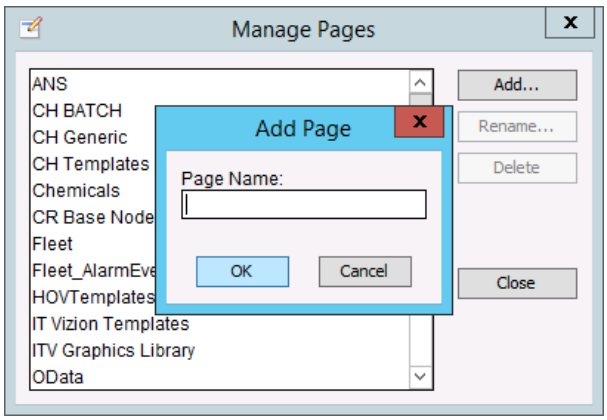


To add a page

1. On the **File menu**, click **Manage Pages**.
   The "Manage Pages" dialog box appears.



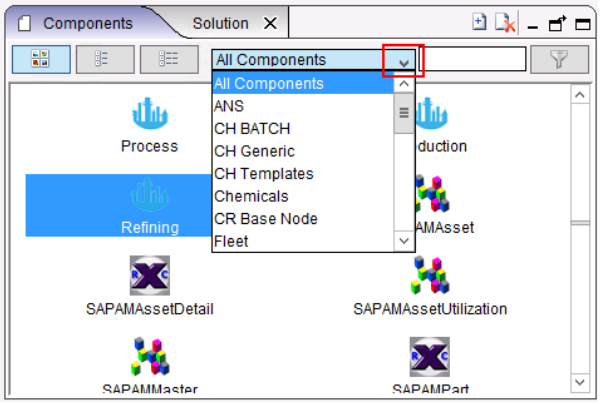
2. Click **Add**.
   The "Add Page" dialog box appears.



3. **Enter** a page name.
   Valid page names consist of letters, numbers, underscores [_], and spaces.

   Page names are not case-sensitive. Therefore, if you try to add a page with the name "Page2" and "page2" already exists, a message appears asking you to enter a different name.

4. Click **OK** and **Close** the "Manage Pages" dialog box.
5. From a list box in the Components tab, click the drop-down arrow to view a list of available pages.

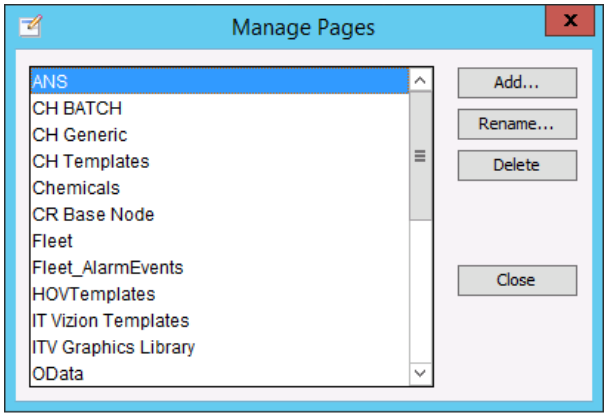


   Page names are listed in alphabetical order.

6. **Select** a page to display.

To rename a page

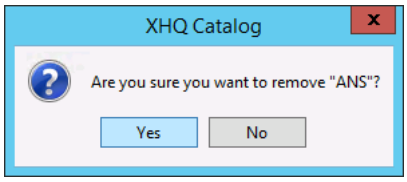
1. On the **File menu**, click **Manage Pages**.
   The "Manage Pages" dialog box appears.



2. From the list, **select** the page to rename.
3. Click **Rename**.
   The "Rename Page" dialog box appears.
4. **Enter** a new page name.
   Valid page names consist of letters, numbers, underscores [_], and spaces.
5. Click **OK** and **Close** the "Manage Pages" dialog box.

To delete a page

1. On the **File menu**, click **Manage Pages**.
   The "Manage Pages" dialog box appears.
2. From the list, **select** the page(s) to delete and click **Delete**.
   A message appears, asking you to verify the removal.



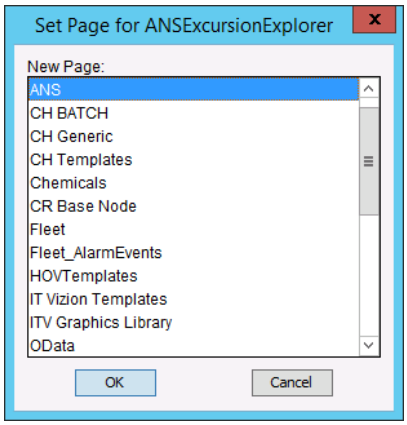
3. Click **Yes** and **Close** the "Manage Pages" dialog box.

Assigning Components to Pages

Assigning a component is one way to populate a page. The other way is to add a component directly to that page. The component, then, appears on both the "All Components" page and the page to which it is assigned.

To assign a component to a page

1. From the **Components tab**, view the **All Components** page.
2. **Select** a component and **right-click**.
   The component shortcut menu appears.
3. Click **Set Page**.
   The "Set Page" dialog box appears.



4. **Select** a page.
   The component you selected in step 2 will then be assigned to this page.

To un-assign a component from a page

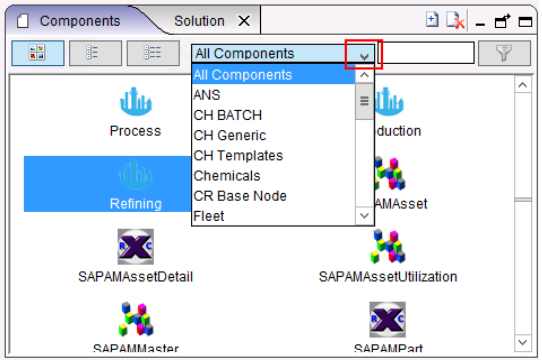
1. From the **Component tab**, view the page from which you want to un-assign the component.
2. **Right-click** on the component and, from the menu, click **Set Page**.
   The "Set Page" dialog box appears.
3. Select **All Components only** and click **OK**.
   The component no longer appears on this page.

To create a new component and add it to a page

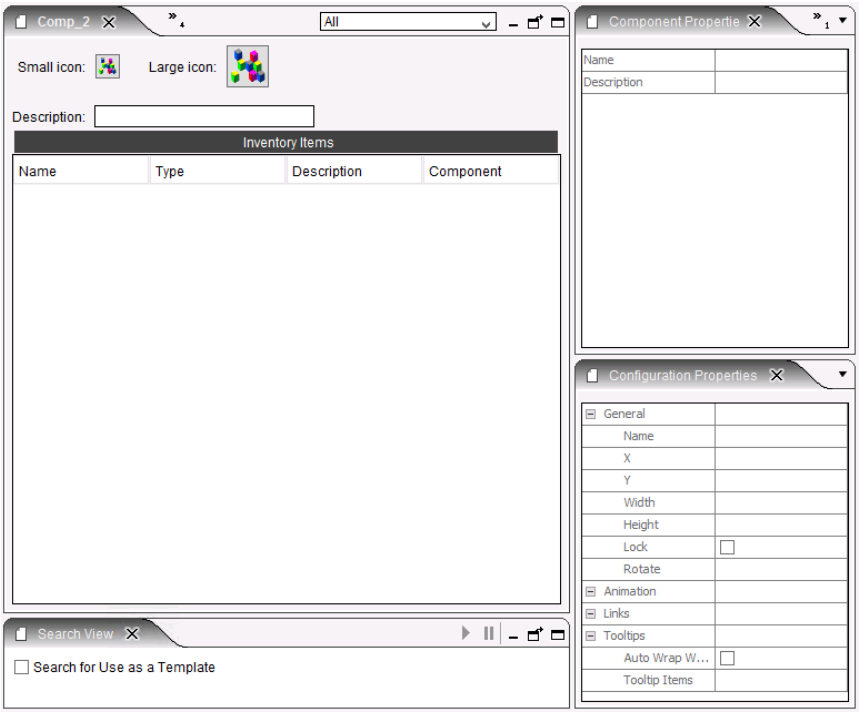
You must have Component Editing rights to create/edit/delete components.

For more information, refer to the topic, *Security, Access, and Privileges*, located in the XHQ Administrator's Guide.

1. From the **Component tab**, **select a page** to open and view.



2. On the **File menu**, click **New Component**.
   The "Add New Component" dialog box appears.
3. Enter a **Component name** and click **OK**.
   The Component Details tab appears.



4. Configure the component and save.

   Any components created during this step will be assigned to this page.

5. **Close** the Component Details tab.
   The component has been added to the page.

Dealing with Multiple Users

If you have instances when more than one user has access to the same page, then it is important to know the page refresh criteria and how they affect editing priorities.

For example, when multiple users are making changes to pages, those changes may not be reflected immediately on the other users' catalog of components.

Also refer to the topic, *General Rules for Multiple Users*.

Page List Refresh Criteria

The page list updates for any of the following reasons:

- A page rename/addition/deletion is performed.

  A component rename/addition/deletion does not affect the page list.

- An attempt to assign a component to a non-existent page.
- Any attempt to access a non-existent page.

  If a user is currently on a page that is determined to be non-existent, the user will automatically get moved to the "All Components" default page

Mulitple Users Scenarios

User1 Action	User2 Action	Expected Results
Is viewing Page1	Removes Page1	User1 will continue to see Page1 until some action causes the page to update (refer to the Page List Refresh Criteria). Once that occurs, User1 automatically gets moved to the "All Components" default page.
Is viewing Page1	Adds Page2	User1 will not see Page2 until a page list refresh occurs.
Is viewing Page1	Renames Page1 to P1	User1 is still be able to assign components to Page1 and does not see a page name change until a page refresh occurs.
Is viewing Page1	Removes Page1 and renames Page2 to Page1	Once Page1 is removed, User1 can no longer assign components to Page1, and receives a message regarding the status of Page1. This failed assignment causes a page refresh and User1 automatically gets moved to the "All Components" default page. Following User2's rename, the "new" Page1 then contains Page2's contents and not the "old" Page1 contents. If User1 tries to access Page2, a page refresh occurs, Page2 disappears from the list, and User1 is automatically moved to the "All Components" default page.
Adds Page3	Subsequently, attempts to add Page3	User1 successfully adds Page3. User2 is unable to add Page3; a "page already exists" message displays.
Assigns Comp1 to Page1	Subsequently, assigns Comp1 to Page2	For User1, Page1 continues to show Comp1 until a page refresh occurs, at which point Comp1 then appears on Page2 and no longer on Page1. For User2, Comp1 no longer appears on Page1.

User1 Action	User2 Action	Expected Results
Assigns Comp1 to Page1	Subsequently, deletes Comp1	For User1, Page1 continues to show Comp1 until a page refresh occurs. At this point, if User1 attempts to reassign Comp1, a page refresh occurs and Comp1 no longer appears.
Assigns Comp1 to Page1	Subsequently, renames Comp1 to Comp2	For User1, Page1 continues to show Comp1. If User1 attempts to edit Comp1, it will appear in the Component Builder as Comp2. Once User1 saves the component in the Component Builder, the XHQ Workbenchupdates and then displays Comp2 instead of Comp1. For User2, the rename is immediately applied.

General Rules for Multiple Users

Here are some rules to consider when there are multiple users developing components.

- If a developer changes the XHQ Workbench configuration by adding, or deleting, a component or component member, then all other developers will have to refresh their XHQ Workbenchs to see these changes. (There are two ways to refresh the XHQ Workbench. You can close and reopen the XHQ Workbench, or, you can switch from one view to another by going to the View menu and selecting a different view.)

From the XHQ Solution Builder, once the "Update to Latest Component" command is executed, all other developers will just have to refresh their XHQ Solution Builders (using the "Refresh" command from the Solution menu).

- If a developer makes a change to the XHQ Workbench and another developer clicks the "Update to Latest Component" command from the XHQ Solution Builder, then other developers will have to "Refresh" their XHQ Solution Builders in order to see the changes in effect.
- Users should not be editing the same view or component at the same time. If, for some reason, two users are editing the same view or component concurrently, then the user who clicks "Save" (from the File menu of the XHQ Workbench) first will get their changes saved.
- Consider this situation: a user has a component open and has made changes to it without saving; then the server crashes or the network connection is lost. Once they try to save, they are told that there is no connection to the server. All modifications are then saved to a recovery file on the local machine.

  For more information on the file recovery mechanism, go to the topic, *XHQ Workbench File Recovery*, located in the XHQ Administrator's Guide.

- Notify all users before you restart the server. If users have unsaved changes to either the XHQ Workbench or the XHQ Solution Builder and the server is restarted, unsaved changes are either saved in a recovery file (as is the case for the XHQ Workbench) or are lost (XHQ Solution Builder).
- When an "Update to Latest Component" is executed, the Solution Server denies all requests until the update is complete. Therefore, this should be a scheduled event.
- Consider this situation: you make some changes to the XHQ Workbench (for example, add a new member to a component) and you do a save. The server is then restarted before you can "Update to Latest Component". Once the server is back up and running, you still must do an "Update to Latest Component" in order to see your changes in the XHQ Solution Builder.

Also refer to the topic, *Dealing with Multiple Users*.

Reports

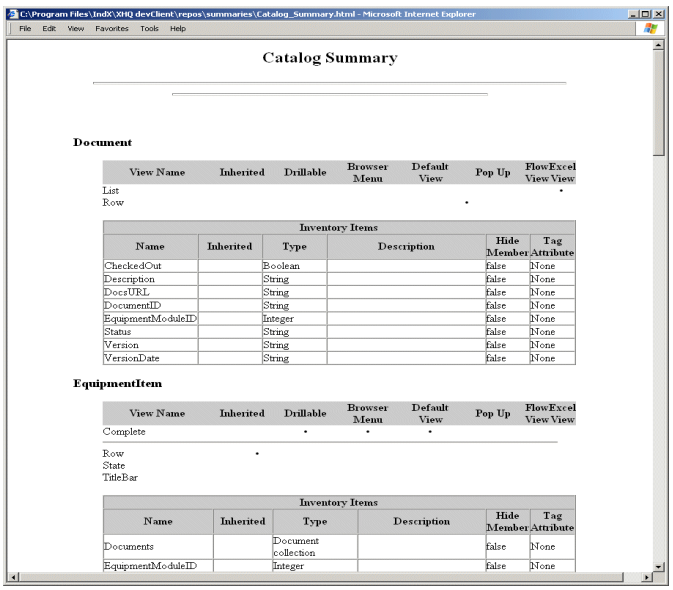
To help you design and verify your solution, XHQ includes a reporting feature that allows you to generate Component and Component/Views summaries. The summary is saved in `.html` format and can be accessed through your browser.

Generating a Component-Views Summary

The Component-Views Summary lists all the components within the XHQ Workbench, and each view and component member (inventory item). The default filename given to the XHQ Workbench summary is **`Catalog_Summary.html`**. If this file already exist in the given directory, a dialog box appears, prompting you to overwrite the existing file.

The default location for this file is:

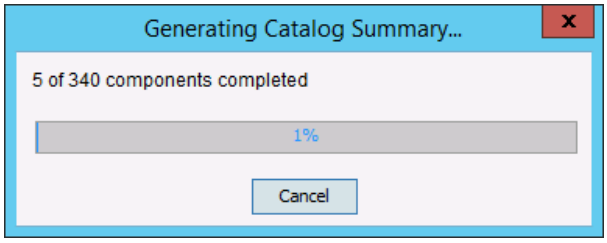
`%APPDATA%\XHQ\XHQ devClient\repos\summaries\Catalog_Summary.html.`



Component-Views Summary as viewed from a Browser

To generate and access a Component-Views summary

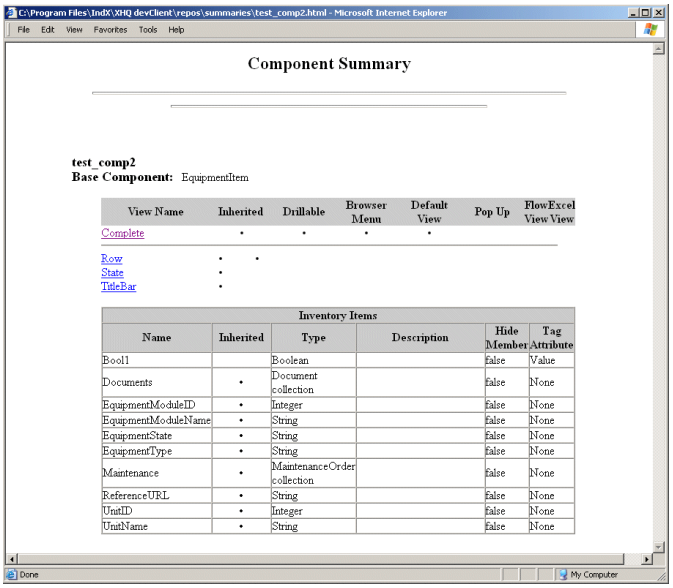
1. Launch **XHQ Workbench** and from the **File** menu, click **Generate Component and Views Summary**.
   The Summary starts generating.



2. **Navigate** to **`%APPDATA%\XHQ\XHQ devClient\repos\summaries`** and click **`Catalog_Summary.html`**.
   The file opens in a browser.

Generating a Component Summary

Unlike the Component-Views Summary, the Component Summary provides information for only one component and all its views. A link is available for each view listed. As such, when the Component Summary is generated, these linked secondary pages are also created.



Component Summary as viewed from a Browser

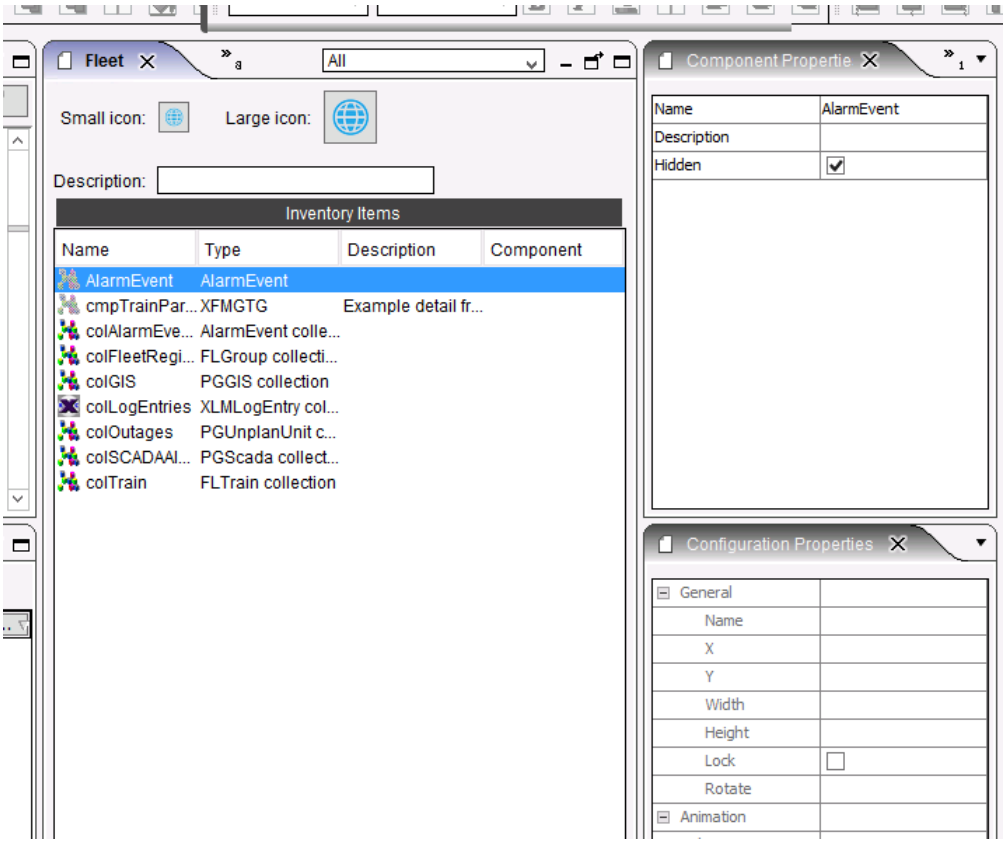
The default filename given to the main page is **`<componentname>.html`**, where `<componentname>` is the name of the component for which you are generating the summary. If a file with this filename already exist in the given directory, a dialog box appears, prompting you to overwrite the existing file.

To generate and access a Component summary

1. Launch XHQ Workbench and from the **Components tab**, right-click on the component.
   The component shortcut menu appears.
2. On the shortcut menu, click **Generate Component Summary**.
   The Summary starts generating.
3. **Navigate** to **`%APPDATA%\XHQ\XHQ devClient\repos\summaries`** and click **`<componentname>.html`**, where `<componentname>` is the name of the component for which you are generating the summary.
   The file opens in a browser.

Tutorial: Building Components

From the XHQ Workbench, you can create and edit components. When you select an existing component from the **Components tab**, a tab appears in the central **Edit panel**. The component name is displayed on the tab. The top section shows the component description and the two icons that represent the component. Members are displayed in the **Inventory Items panel**. To the right are the tabbed panels that presents the fields for the **Component and Configuration properties** for the selected member.



Edit and Properties Panels

To set-up the repos environment for this tutorial

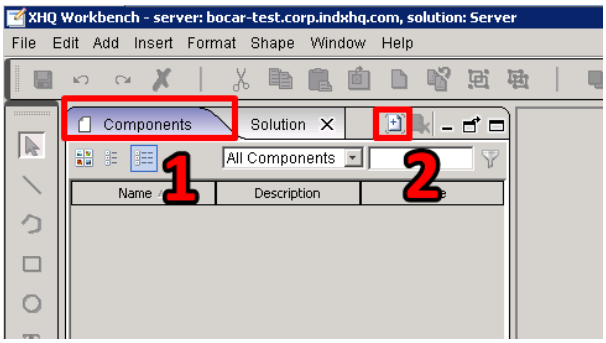
This task shows you how to set-up a standard XHQ environment from scratch using an empty repos.

1. Go to the location specified by the environment variable `%XHQ_SERVER_REPOS%` on the server (which is typically `<systemdrive>:\XHQ\data\repos`).
2. Rename the existing `repos` folder to **`repos_original`**.
3. Then rename `repos_empty` to **`repos`**.
4. From the XHQ installation media, copy the **Getting Started** root folder onto the server (for example, `C:\Getting Started`).

To build components from the XHQ Workbench

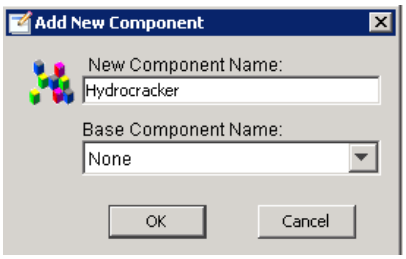
This task assumes that you have copied the complete **Getting Started** directory to your local disk drive, as instructed in step 4 of the previous task. All references are relative to this location, unless otherwise specified.

1. Launch the XHQ Workbench.
2. From the **Components tab**, click the **New** button.



The "Add New Component" dialog appears.

3. In the **New Component Name** box, enter the unique component name, `Hydrocracker`.

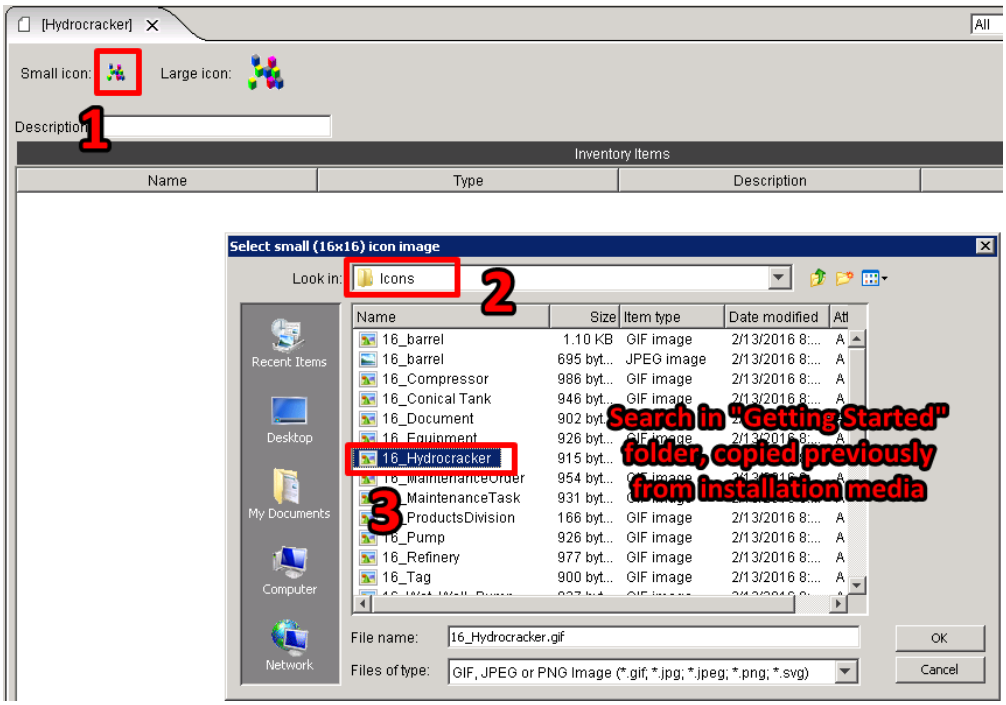


For more information on this task, go to the topic, *How to name components*.

4. In the **Base Component Name** box, select **None**
5. Click **OK**.
   The component members list tab appears.
6. Click the **small icon** to select a 16x16 icon to associate with the component.

   These icons are not Windows icon files. They are .GIF files of a specific size: 16 x 16 pixels for the small icons and 32 x 32 pixels for the large icons. The files have been named in such a way to identify the size and purpose of the icon. This naming convention or a similar one is recommended.

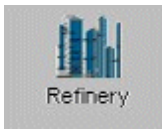
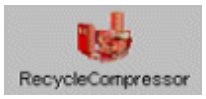
   The "Select icon image" dialog appears.
7. Navigate to the **Getting Started** directory.
8. Open the **Icons** folder, locate and select the **16_Hydrocracker** .GIF image.



9. Click **OK**.
10. Next, click the **large icon** to select a 32x32 icon to associate with the component.
11. Again, navigate to the **Getting Started/icons** folder.
12. Locate and select the **32_Hydrocracker** .GIF image.
13. Click **OK**.
    This returns you back to the Component Details tab.
14. Enter a **Description**.
15. **Save** the component record.

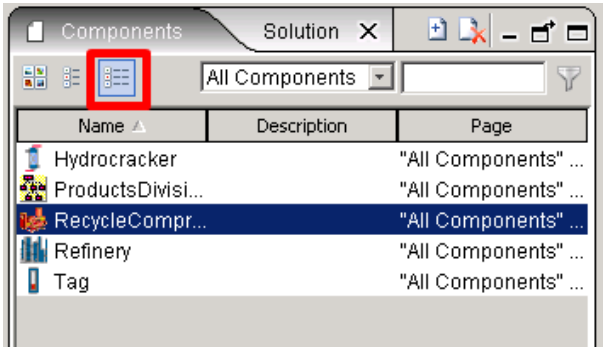
    Unsaved records have an asterisk (*) next to the component name.

16. **Repeat Steps 2 to 15** to create and assign icons to the following components:

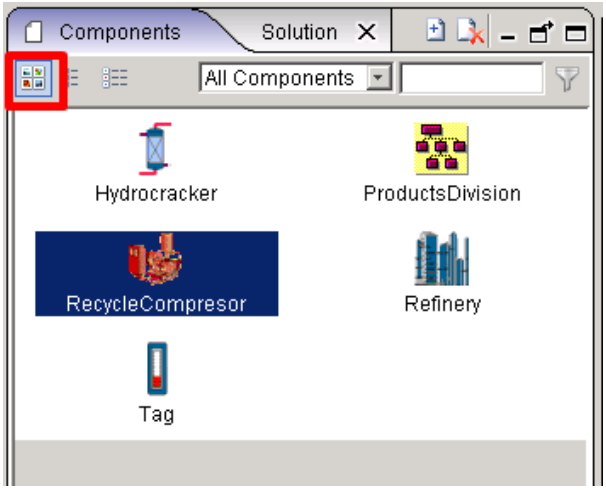
Create this component...	And use this icon
ProductsDivision	
Tag	
Refinery	
RecycleCompressor	

When complete, the **Components tab** displays the list of components you added.

(if **Details** view is selected)



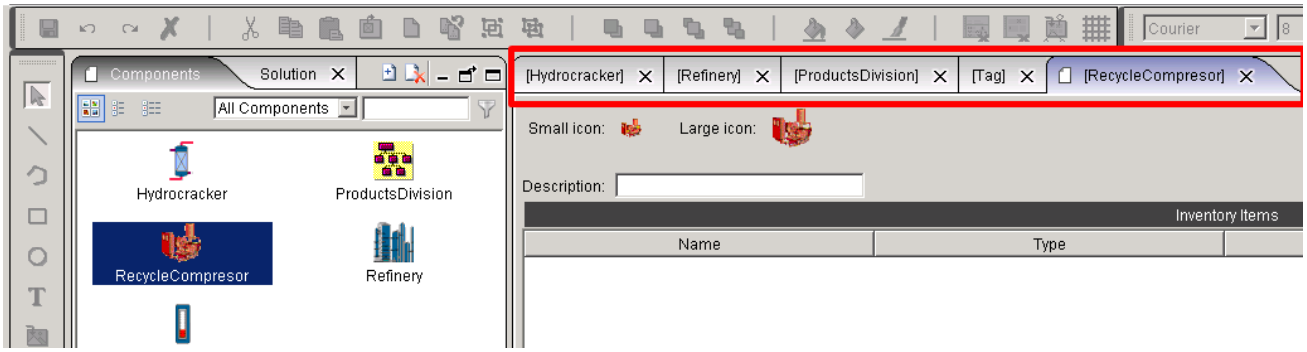
Components Tab - Details View



Components Tab - Large Icons View

To add component members

1. From the **Components tab**, select a component.
   This activates a tab for this component in the main editing panel.



2. From the **Add menu**, you can select either a primitive member of a specific data type, or an existing component, or create a new component.

For more information on adding component members, go to the topic, *Working with Members*.

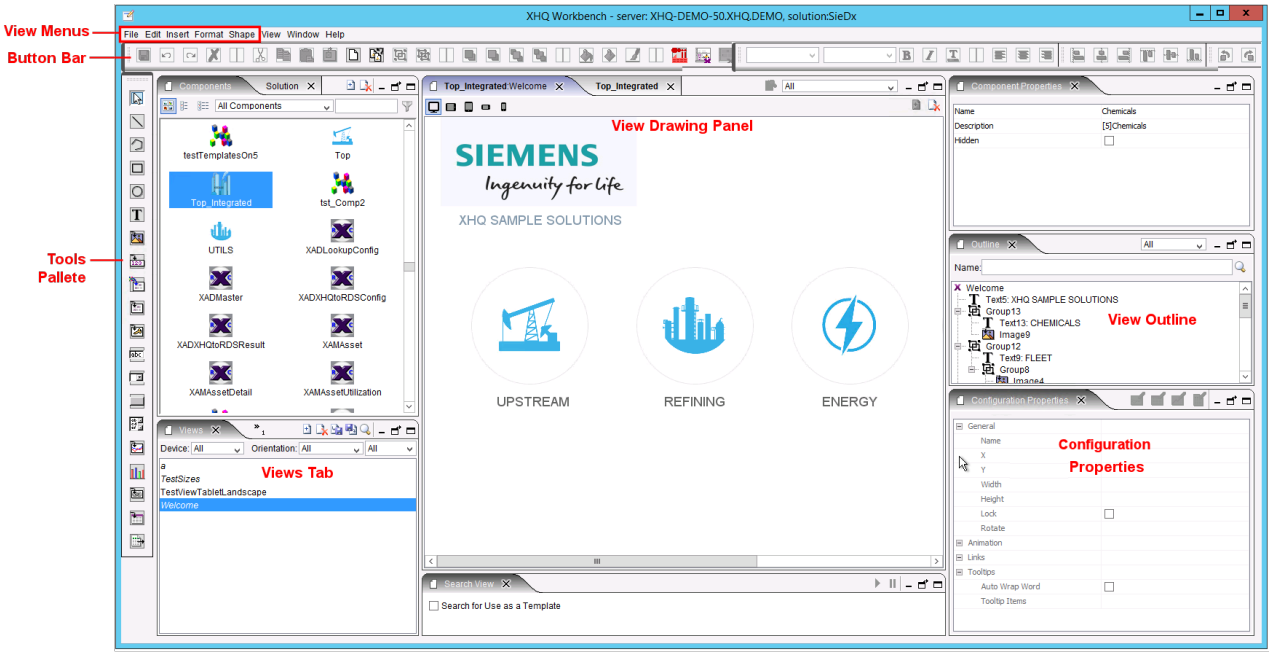
5 | Views and View Elements

With the XHQ Workbench, you can build *views* that attach to the components you have built. A view combines backgrounds, bitmaps, and previously prepared animations to present information contained within a solution. A view may also contain additional graphic elements that enhance the appearance of the information, such as static text for headers and labels, and rectangles for borders.

Views are pictorial representations of the data your backend systems generate about your enterprise. A view displays information contained within the component instance or from any of its member components or submember components below it in the same path hierarchy.

The View Building Environment

The XHQ Workbench contains the following view-related sections.



XHQ Workbench: View Building

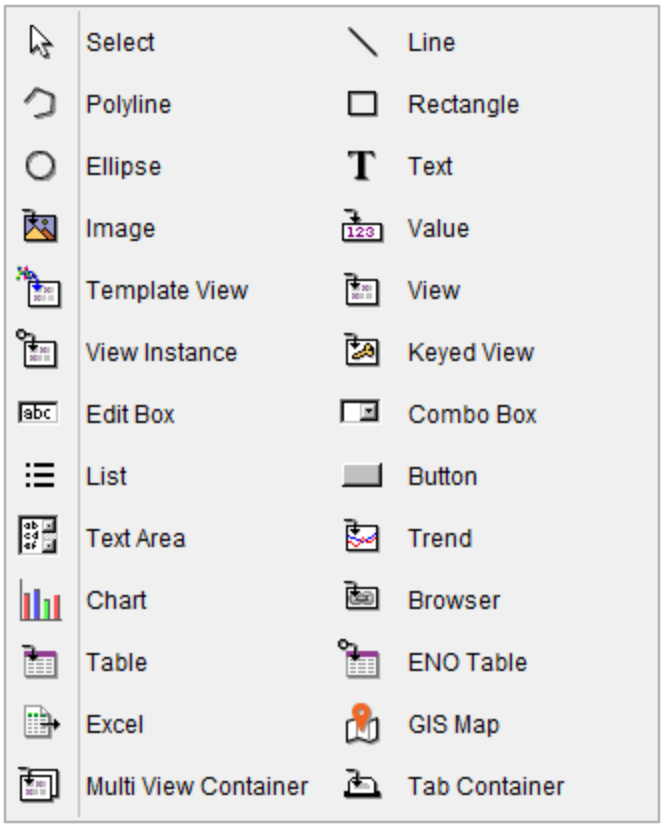
The **Drawing Panel** is where you work with view elements. You add to the drawing panel by simple point-click operations.

To the left is the **Tools Palette**, which provides basic drawing and insertion tools. The **Button Bar** provides general operations functions as well as tools for working with color and multiple view elements. The **Configuration Properties tab** provides additional tools for linking, animating, and modifying the appearance of view elements in a view.

The topic, *Building Views*, covers technical aspects of view building, such as creating views that include live data and animation effects.

The Tools Palette

The tools you need to build views are presented in the Tools Palette.



Tools Palette

For detailed information on each tool, go to the topic, *Working with View Building Tools*.

The Button Bar

The Button Bar is located above the edit panel. It contains icons to activate commonly used functions that span multiple modes of the model-editing environment, and also general tools that affect the operation of the environment. The following table summarizes the function of each button.

Button Bar Options

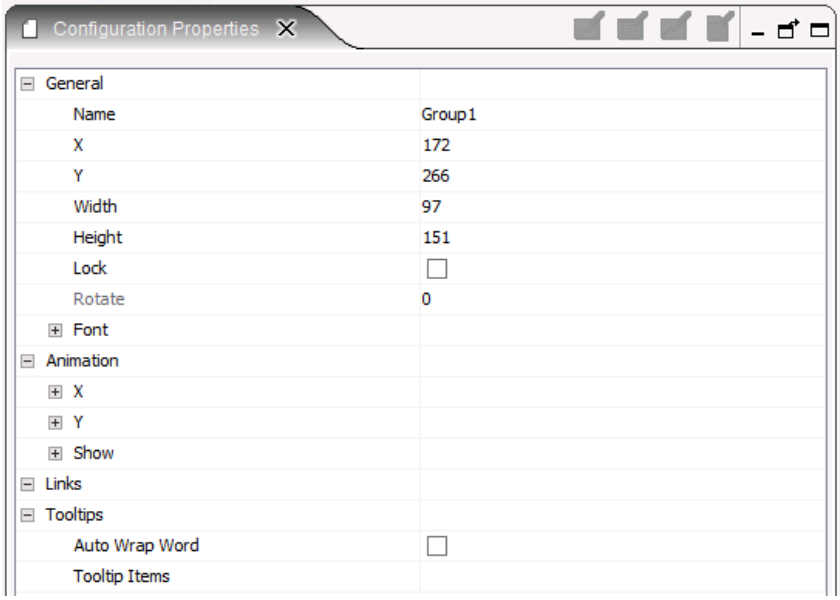
Icon	Button Name	Description
	Save to repository	Saves a new view; saves changes made to a view.
	Undo	Use this command to undo your very last action.
	Redo	Use this command to reapply an action that you have undone.
	Delete selection	Deletes selected view element(s).
	Cut selection	Removes the selection and copies it to the clipboard. **Cut vs. Delete:** In addition to removing it from the view, the CUT command copies the selected view element to the clipboard so that you have the option of pasting it elsewhere. On the other hand, the DELETE command *completely* removes the selected view element.
	Copy selection	Copies the selection to the buffer (clipboard).
	Paste from clipboard	Inserts the cut or copied view element or view from the buffer into the same view or a different one.
	Export selection to file	Saves the selection as XANI file.
	Clear contents	Clears the view and reverts to a blank view.
	Revert to saved	Reverts to the previously saved version of the view.
	Group selection	Groups multiple view elements.
	Ungroup selection	Ungroups a previously grouped set of view elements.
	Move backward	Moves the selected view element one step backward.

Icon	Button Name	Description
	Move forward	Moves the selected view element one step forward.
	Bring to front	Moves the selected view element to the top.
	Send to back	Moves the selected view element to the bottom.
	Set fill color	Selects and applies fill color to a selected view element.
	Enable fill	Toggles between fill and no fill for a selected view element.
	Set line color	Selects line or outline color for a selected view element.
	Open Symbol Factory	Opens the Symbol Factory.
	Launch Component Builder	Opens the Component Details tab for the component associated with the active view.
	Launch View Builder	Opens the View Drawing tab to allow you to edit an embedded view.
	View Details	Opens a browser; lists view configurations of the current view.
	Configure Grid	Opens the Grid Configuration dialog box. Allows you to display/hide grid, enable/disable snap-to-grid, set grid spacing.

Each of the commands in the table above has a shortcut key assigned to it. For a list of these key combinations, go to the topic, *Shortcut Keys*.

The Configuration Properties Tab

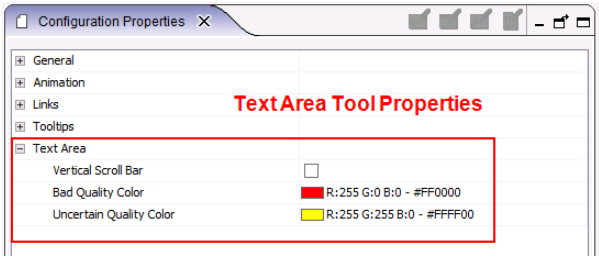
The Configuration Properties tab enables you to configure properties that affect the view elements in your views and animations.



XHQ Workbench: Configuration Properties Tab

The properties are grouped into the following:

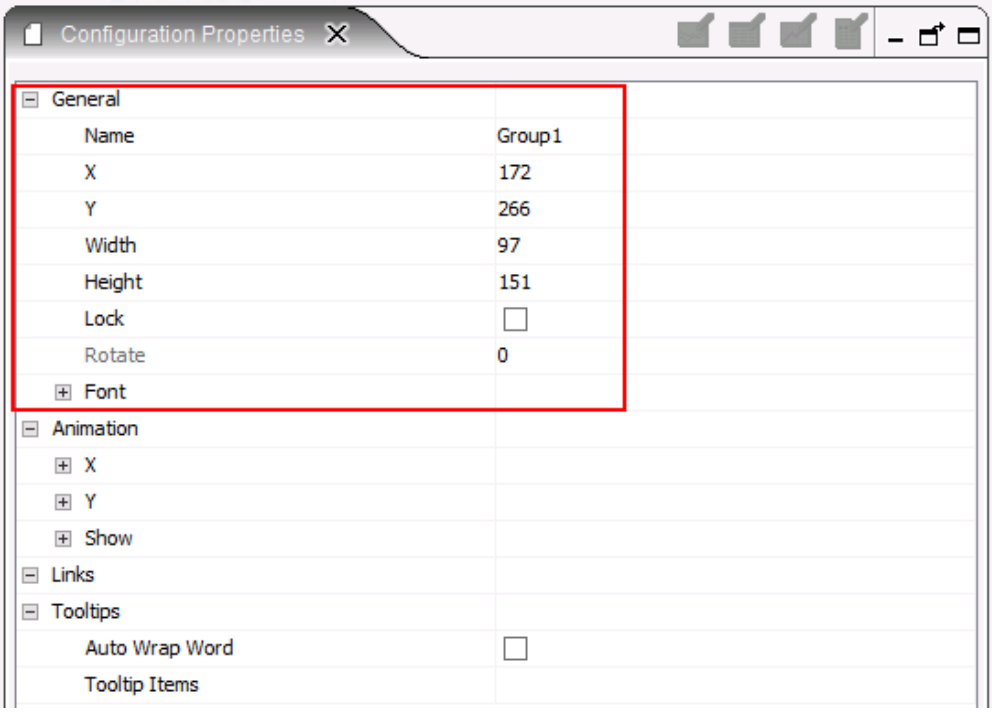
- General
- Animation
- Links
- Tooltips
- Tool-specific Properties



XHQ Workbench: Tool-specific Properties Group

About the General Properties

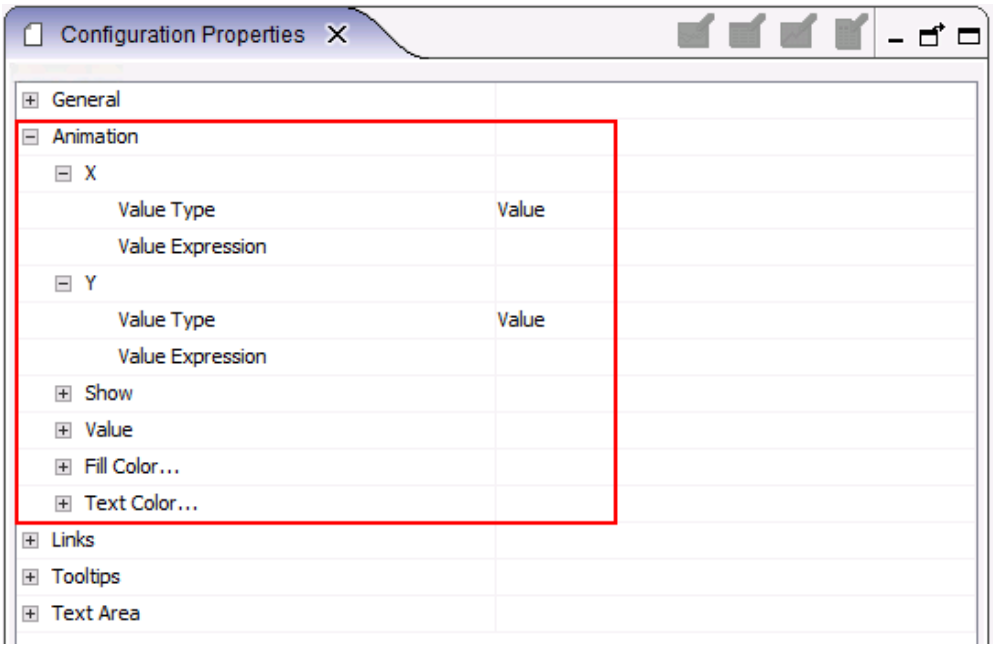
The General group contains tools for positioning, sizing, and aligning view elements, as well as selecting font attributes.



General Properties

About the Animation Properties

For the Animation group, enter values that affect the presentation of a view element at runtime.



Animation Properties

For more information, refer to the topic, *Using Animation*.

Animations must be configured to link through a solution model to collect data from backend data sources. By modifying display properties based on changes in values in backend systems, animations help draw a user's attention to specific information.

Animation Effects

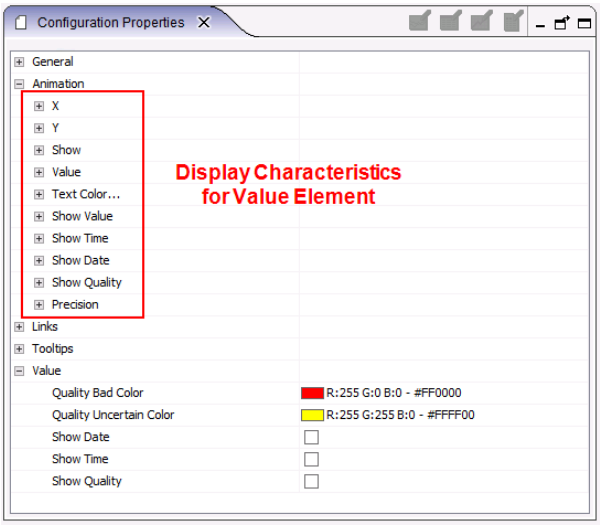
Display Characteristic (Animation)	Animation Type	Variables
• X location • Y location • Text color • Show/hide • Width • Height • Fill color • Line color	• Value range • Value breakpoint • Value • Value tag	• Real • Integer • Boolean • String

The Value breakpoint type is not available for non-color-related animation, such as Width, Height, Show, and Value.

The list of available display characteristics varies, depending on the type of view element you are animating.

For example, for a **value** element, you can set animation for the following display characteristics:

- X location
- Y location
- Show
- Value
- Text Color
- Show Value
- Show Time
- Show Date
- Show Quality
- Precision

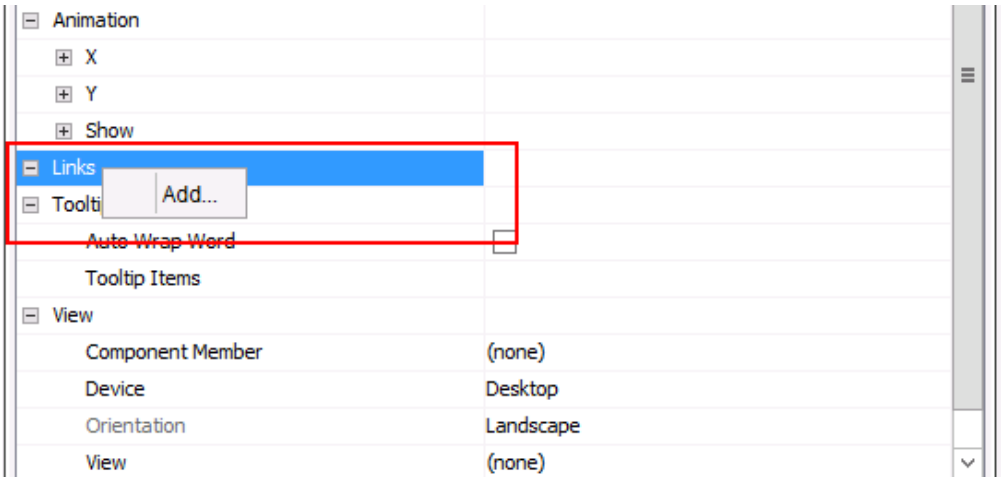


About the Links Properties

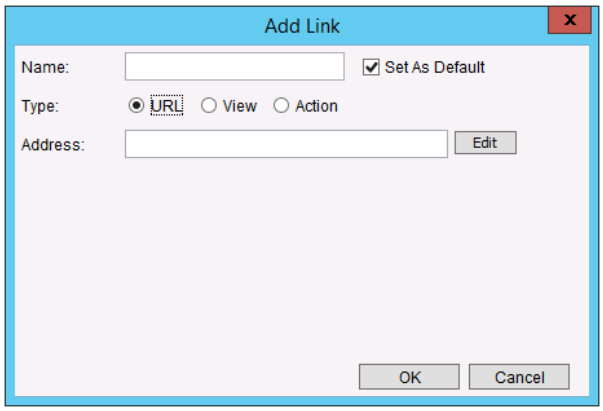
Under the Links group, you set hyperlinks to other content, including other views and URLs. You can also set the Action type, which allows you to perform any action specified by a function expression.

For more information, refer to the topics, *Using Links* and *Action Link Functions* (which is located in the XHQ Reference Guide).

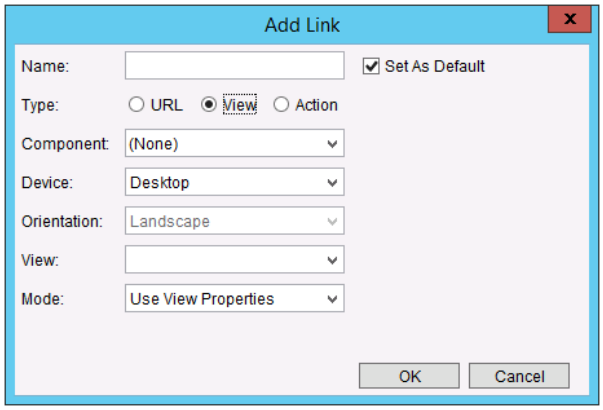
To add a link to a view, from the **Configuration Properties tab**, right-click inside the Links cell. A shortcut menu appears with the Add option.



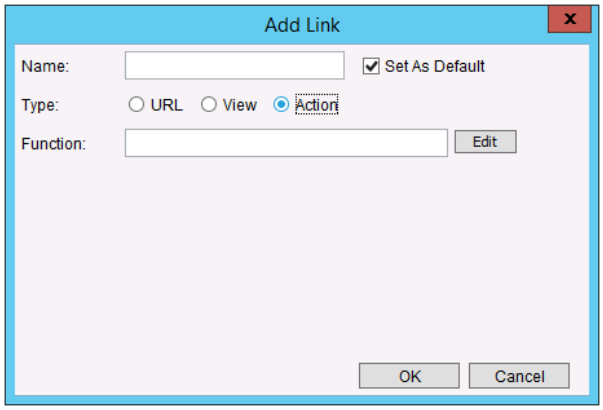
Click **Add**. The "Add Link" dialog appears. Select a link **Type**.



Add Link Dialog: URL Type



Add Link Dialog: View Type

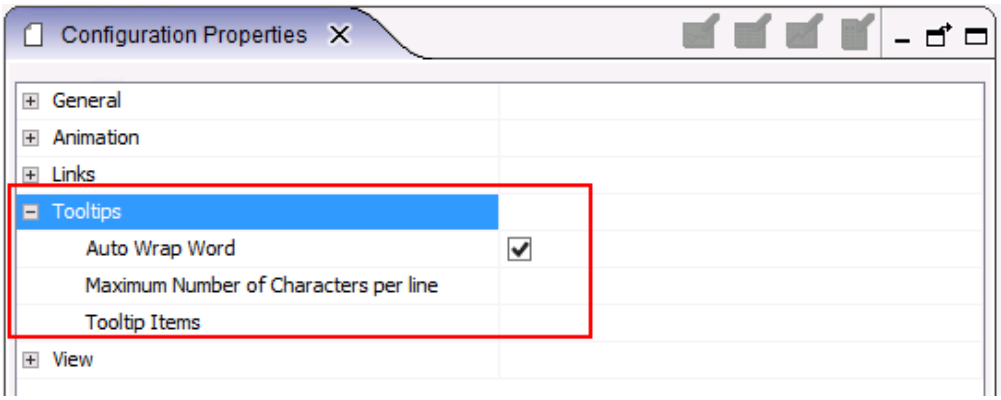


Add Link Dialog: Action Type

About the Tooltips Properties

This grouping enables you to customize the Tooltip for any view element.

For more information, refer to the topic, *Customizing Tooltips*.



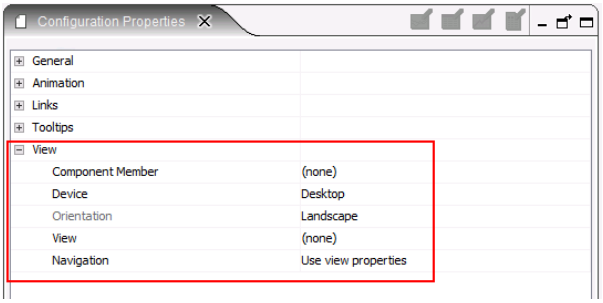
Tooltip Properties

About the Tool-specific Properties

This properties group is based on the tool you select. The tools involved are: line, polyline, text, value, view, template, keyed view, edit box control, combo box control, button control, text area control, trend, embedded browser, and chart.

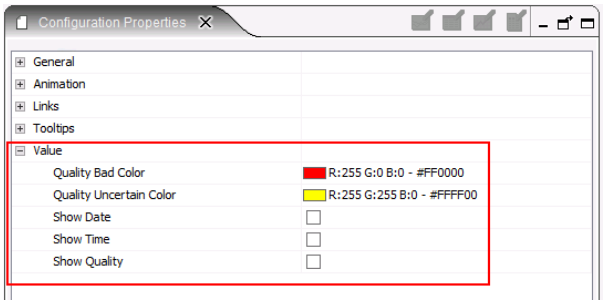
Note the following examples.

When you add a view to the drawing panel, the **View** properties group appears.



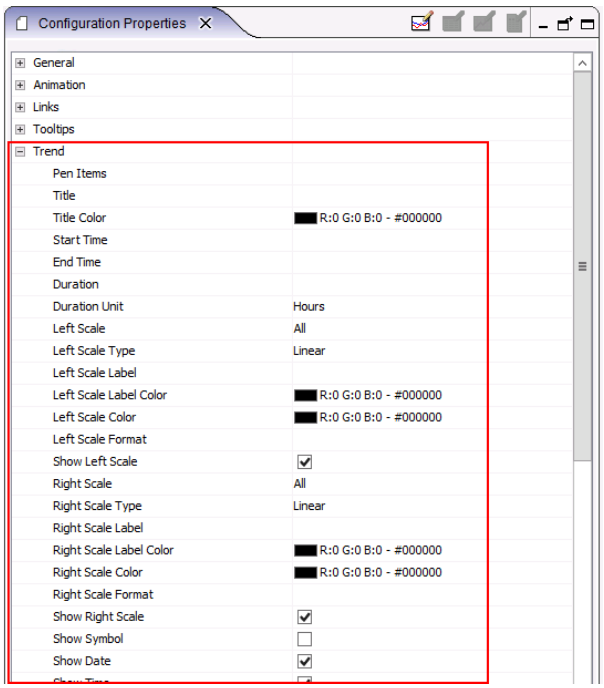
Tool-specific Properties Group: View

When you insert a Value item, the **Value** properties group appears.



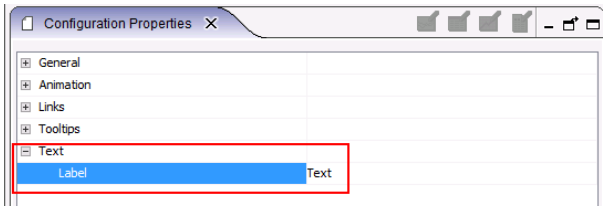
Tool-specific Properties Group: Value

When you insert a trend within the view, the **Trend** properties group appears.



Tool-specific Properties Group: Trend (image above shows a partial list of properties)

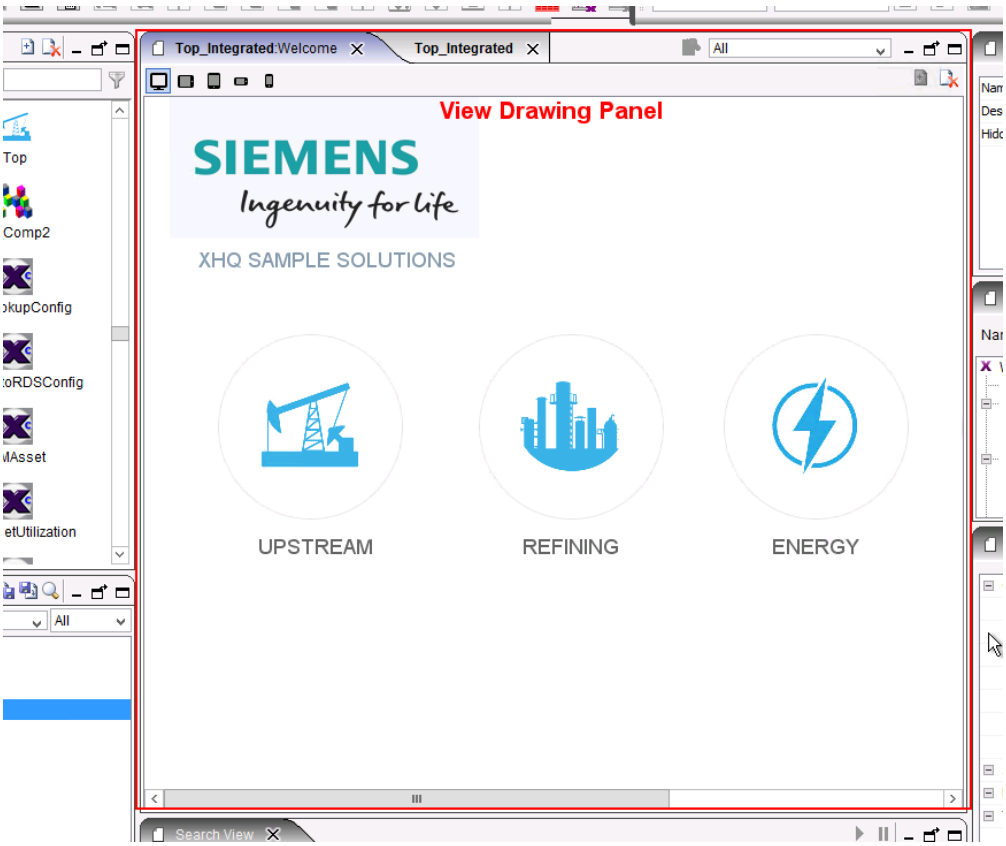
And when you insert a text item, the **Text** properties group appears.



Tool-specific Properties Group: Text

The Drawing Panel

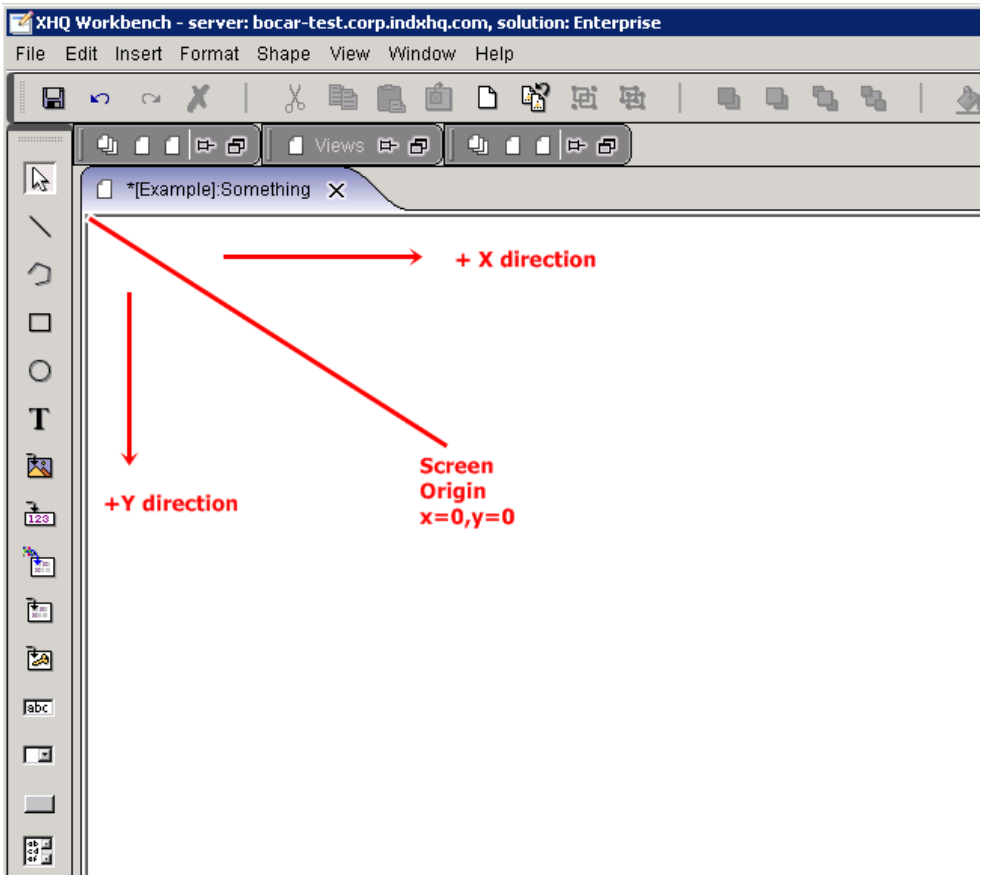
This area of the XHQ Workbench is where you create and edit view elements. To change the size of the drawing panel, drag a border or corner of the View Drawing tab.



XHQ Workbench: View Drawing Panel

Coordinate System

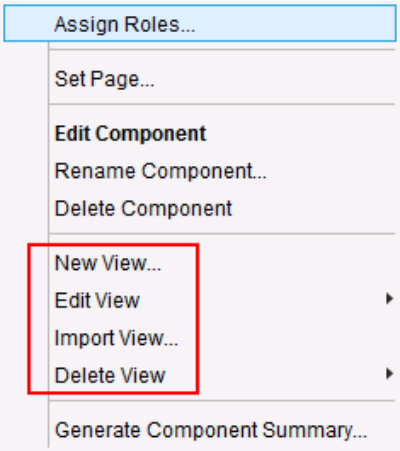
The screen origin (X=0, Y=0) is located at the upper left corner of the drawing panel.



Coordinate System

Right-click Menu Commands

Right-click a component in the XHQ Workbench and, from the shortcut menu, select any of the view-related commands.



XHQ Workbench Right-Click Shortcut Menu

View-related Commands

Command	Description
New View	Creates a new view for the given component.
Edit View	Edits a given view for the given component. **Note:** Views that are listed but are grayed-out (disabled) belong to a base component and therefore cannot be edited.
Export View	Exports the view file for the given component.
Import View	Loads a view file on to the given component.
Delete View	Removes the view from the given component. **Note:** Views that are listed but are grayed-out (disabled) belong to a base component and therefore cannot be deleted.
Views Properties	Enables you to identify drillable views, designate a default view, set pop-up views, or define the browser menu view selection list. **Note:** Views that are listed but have a gray background (View Name column) are read-only. These views belong to a base component and therefore their view property settings cannot be edited. For more information, refer to the section, *Building Views*.
View Search	Searches through the XHQ Workbench to find where the chosen component and view appear as an embedded view or as a collection. **Note:** Search is available on all views, including inherited views from the base component.

Performing Basic View Building Tasks

In this section, you will learn simple tasks such as creating, saving, and deleting views.

Creating and Naming Views

When you create a new view, you must use a name that is unique among views within the component where it is attached.

Rules for View Names

- View names must be at least one character long;
- You can use letters, numbers and underscores;
- You cannot use spaces or other special characters, such as dashes and periods;
- View names can start with a letter, a number, or an underscore;
- You cannot use *reserved words* as view names.
  These are words that have a specific meaning to the XHQ system. If you use a reserved word, you will receive a notification error asking you to choose a different name.

For a list of reserved words, refer to the topic, *Reserved Words*.

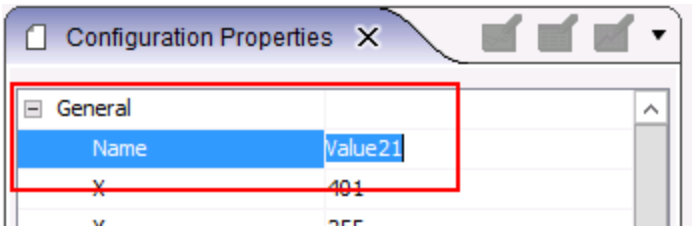
Also, when you name views, consider that all of the views available for a component will be sorted alphabetically in the view list that displays in the XHQ Solution Builder.

About the NavBar View Tab

At runtime, the NavBar tab displays only the first 64 characters of the view name. If the view name is longer than this, then it is appended with "...".

Assigning a Unique Name

Through the **General property**, you can assign a unique name for a given view element. This allows you to reference the tool from elsewhere within your solution. For example, you may use the edit box name to refer to the control from an expression within a view.



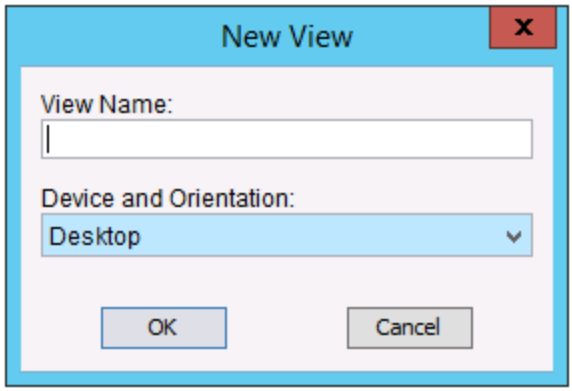
General Property– For Nameable View Elements

Nameable View Elements

Icon	Tool
	Value
	View
	Keyed View
	Template
	Controls – Edit box, Combo box, Button, Text Control
	Embedded Browser Control

To create a new view

1. From the **Component tab**, **right-click** on the component.
   The shortcut menu appears.
2. Click **New View**.
   The "New View" dialog box appears.



3. In the **View Name** box, enter a name.
4. Select a **Device and Orientation**.
5. Click **OK**.
   A View tab launches with a blank page.

To open a view

1. From the **XHQ Workbench**, **right-click** on a component.
2. On the shortcut menu, point to **Edit View** and then click on a view from the list displayed.
   The View tab launches with the view you select.

Saving and Closing a View

Save the views you are working on before exiting the XHQ Workbench. When you save a view, the next time you reload or request that view (or any other view that contains it), the system loads the newly saved version of the view.

If you attempt to exit the XHQ Workbench before saving, the system will ask you if you want to save your work.

To save a view

Do one of the following:

- Click the **Save** icon in the button bar.

or

- Click the **close** icon to exit the XHQ Workbench. You will be prompted to save your changes. Click **Yes**.

Reverting to Last Save

Use the **Revert to last** button, , to remove all changes made to a view since it was last saved.

This command is not available until you alter the previously saved view.

To revert to a previously saved image

1. Make changes to a saved view.
2. Click the **Revert to last** button.
   This deletes all the changes you made in step 1 and reverts back to the previously saved version of the view.

Deleting a View

Deleting a view is an irreversible operation.

To delete a view from an existing component

1. From the XHQ Workbench, **right-click** on the component.
   The component submenu appears.
2. On the submenu, click **Delete View**.
   The "Delete View" dialog appears.
3. Select the view you want to delete.
4. Click **OK**.

Clearing a View

With the **Clear contents** command, you can clear your view and revert to a blank view.

To clear a view

1. Select the view you want to clear.
2. Click the **Clear contents** button .

   A message appears, asking you to verify this action.
3. Click **OK**.

Using Undo/Redo

With the **Undo** command, you can undo any of the following actions performed on a view element: Reposition, Resize, Delete, Add, Group, and Ungroup.

The Undo command is only enabled when you perform any of these actions. The **Redo** command allows you to reapply an action that you have just undone.

Actions that you can Undo

Action	When you Undo
Reposition	An **Undo Reposition** moves the view element back to it's last known location. **Note:** This action can be performed on a single view element or a set of (grouped) elements.
Resize	An **Undo Resize** restores the view element's size. **Note:** This action can be performed on a single view element or a set of (grouped) elements.
Delete	An **Undo Delete** replaces the deleted view element(s) back to the view. **Note:** This action can be performed on a single view element or a set of (grouped) elements.
Add	An **Undo Add** removes the added view element(s), which is done by *inserting* the view element using the Tools palette, from the view. **Note:** This action can be performed on a single view element or a set of (grouped) elements.
Cut	An **Undo Cut** replaces the cut view element(s) back to the view. **Cut vs. Delete:** In addition to removing it from the view, the CUT command *copies* the selected view element to the clipboard so that you have the option of pasting it elsewhere. On the other hand, the DELETE command *completely* removes the selected view element.
Paste	An **Undo Paste** removes the previously pasted view element(s).
Duplicate	An **Undo Duplicate** removes duplicated draw items (which is done by selecting one or more draw items and then pressing CTRL+D).
Group	An **Undo Group** removes the grouping, making each selected view element editable.
Ungroup	An **Undo Ungroup** restores the grouping of selected view elements.

If the Undo or Redo button is enabled and you mouse-over either button, a Tooltip appears. The Tooltip helps you determine which action will be undone and/or redone.

If the action cannot be undone, the Undo button and menu command are inactive. Therefore, to undo use the **Revert to saved** command to revert to the last saved version of the view. (Save your work often to prevent you from losing your work should your computer shut down unexpectedly.)

Undo Preferences

By default, you can undo up to 25 actions. To change this limit, set the global property **`MaxUndoStackSize`** to a value between 1 and 100.

For more information, go to the topic, *Additional Global Settings*, located in the XHQ Administrator's Guide.

The following options will remove all the "undo" steps retained during a view building session:

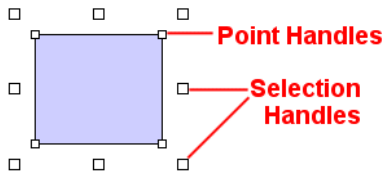
- Revert to saved
- Clear contents
- Close
- Exit

Working with View Elements

This section provides step-by-step instructions on how to manipulate, size and position view elements in the XHQ Workbench.

Pointing and Selecting a View Element

To work on a view element, you must first select it. The Select tool enables you to select single or multiple view elements and move and scale them. View elements always remain selected until you make a new selection, or deselect by clicking a blank space on the background (clicking out of the view element). When you select a view element, eight outer *selection handles* appear at the selection boundary, and four inner *point handles* appear at the corners of the view element (for lines, only two point handles appear).



Selection and Point Handles

You can select multiple view elements to perform common operations on all of them (for example, to nudge the position or to set a common text color). You can also tab to alternately select each of the view elements in a view. You simply press the **TAB** key to select another element, and then repeat tab to step through each view element in the view until the desired view element is selected.

To select a view element

1. Click on the **Select tool**.
   The cursor changes to a pointer.
2. Click anywhere on the view element.
   You can click inside most view elements to select them, whether they are filled or not. However, to select an unfilled rectangle, you must click on its outline.

To select multiple view elements

1. Click on the **Select tool**.
   The cursor changes to a pointer.
2. Do one of the following:
   - Point at a desired view element and click to select it, then hold down the **Shift** key and select one or more additional view elements in the same manner.

   or

   - Click the mouse and drag the **Select tool** diagonally across the screen, fully surrounding the view elements you want to select with the rubber-band cursor (also called a marquee) that appears. When you release the mouse button, the surrounded view elements are selected.

To deselect all view elements

1. Click a blank area on the background.
2. Deselect one view element from a group of selected view elements.
3. Hold down the **Shift** key and click anywhere on the view element you wish to deselect.

To delete a view element

Do one of the following:

- Select the view element to be deleted and press **DELETE** on your keyboard.

or

- Select the view element to be deleted and click the **Delete Selection** icon.

Positioning a View Element

The XHQ Workbench provides several ways to position view elements. Dragging is the fastest way to reposition a view element. You pick a view element using the Select tool, and then interactively drag it to any location in the view element area. You can also nudge a view element vertically or horizontally in increments of 1, 5, or 10 pixels, depending on which keystroke combination you use. In addition, you can enter the exact x- and y-coordinates for the view element under the General property.

For more information on x- and y-coordinate positions, refer to the topic, *Coordinate System*.

To move a view element interactively

1. Select the view element with the **Select tool**.
2. Click and **hold** the left mouse button and **drag** the view element to the desired location.
3. **Release** the mouse button when the object is positioned as desired.

To nudge a view element

1. Select the view element with the **Select tool**.
2. Do one of the following:
   - Use the **arrow keys** on your keyboard to nudge the view element in increments of **1 pixel**.

   or

   - Hold the **CONTROL** key and use the arrow keys to nudge the view element in increments of **5 pixels**.

   or

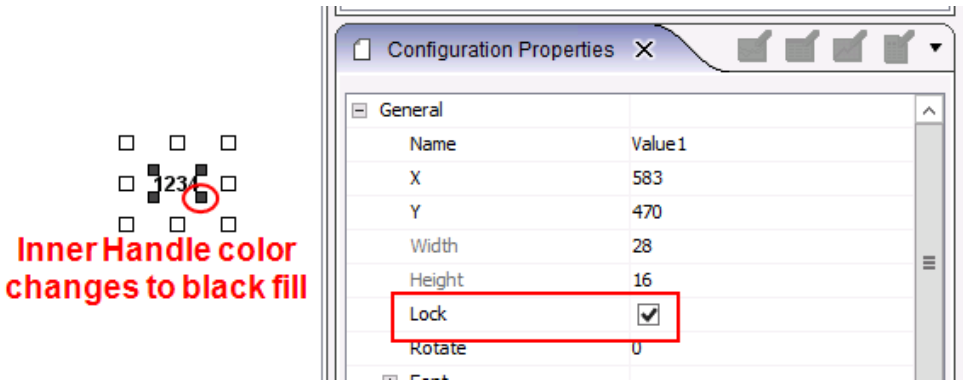
   - Hold the **SHIFT** key and use the arrow keys to nudge the view element in increments of **10 pixels**.

To move a view element to a precise location

1. Select the view element with the **Select tool**.
2. Click the **General** property.
3. In the **X** and **Y** boxes, enter the desired positions.

Locking View Element(s) Into Position

You can lock the position of an individual view element or a set of view elements by selecting the **Lock** option (located on the General property). When a view element is locked, the inner (point) handles are black filled.



Lock Option

When you include a locked view element as part of a group, the Lock attribute is removed. The view element will move or resize with the group.

To lock a view element(s) into position

1. Select a view element (or a set of view elements).
2. From the **General** property, enable (check) the **Lock** option.
   The point handles around the selected view element(s) become dark gray in color.

Aligning View Elements

From the main toolbar, you can align a set of selected view elements by their edges or their horizontal/vertical axis.

View Elements Alignment Buttons

Icon	Button Name	Function
	Lefts	Aligns the objects you select by their **left edges**.
	Centers (Vertically)	Aligns the objects you select **vertically** by their **centers**.
	Rights	Aligns the objects you select by their **right edges**.
	Tops	Aligns the objects you select by their **top edges**.
	Middles (Horizontally)	Aligns the objects you select **horizontally** by their **middles**.
	Bottoms	Aligns the objects you select by their **bottom edges**.

For more information, go to the topic, *How to Align View Elements*

Flipping an Image

You can flip an image along its horizontal axis (where the top becomes the bottom, and vice versa) or along its vertical axis (where the left side becomes the right side, and vice versa)

Flip Options

Icon	Button Name	Function
	Flip Horizontal	Flips an image along its **vertical axis**.
	Flip Vertical	Flips an image along its **horizontal axis**.

Rotating a View Element

Rotating a view element turns it around its center point. You can rotate a view element 90 degrees clockwise (towards the right) or 90 degrees counter-clockwise (towards the left).

The Rotate option is only available for the following view elements:

- Line
- Polyline
- Rectangle
- Ellipse
- Text
- Value
- Image

You cannot rotate a set of selected view elements or a group of view elements.

Icon	Command	Function
	Rotate Left	Rotates a view element 90 degrees **counter-clockwise** (towards the left).
	Rotate Right	Rotates a view element 90 degrees **clockwise** (towards the right).

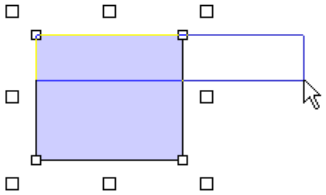
To rotate a view element

1. **Select** the view element and selection handles appear around the view element.
2. Do one of the following:
   - Click the **Rotate Left** option to rotate the view element 90 degrees counter-clockwise to the left.
   - Click the **Rotate Right** option to rotate the view element 90 degrees clockwise to the right.

A **positive** rotation angle moves the view element clockwise. A **negative** rotation angle moves the view element counter-clockwise.

Sizing a View Element

You can size view elements using the mouse by interacting with the **inner** point handles of a view element. While you drag using the mouse, an outline of the view element shows as it transforms and the handles remain in their original positions. When you release the mouse, the shape appears at its new size with appropriate new handles.



Sizing the View Element

You can also size a view element by specifying the desired width and height in the General property.

Resizing Proportionally

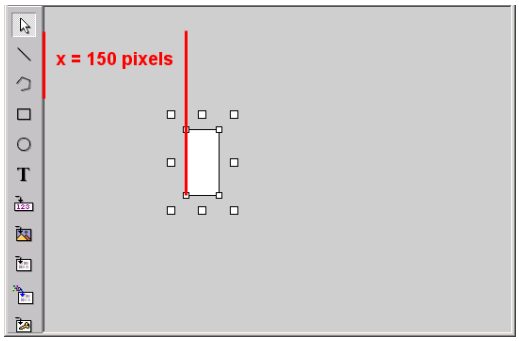
In the XHQ Workbench, to resize view elements proportionally, hold the SHIFT key down, click on any corner of the resizable element, and drag to the desired size. Both the width and height of the element are adjusted accordingly.

Proportional resize does not apply to the following view elements: lines, polylines, buttons, edit boxes, and combo boxes.

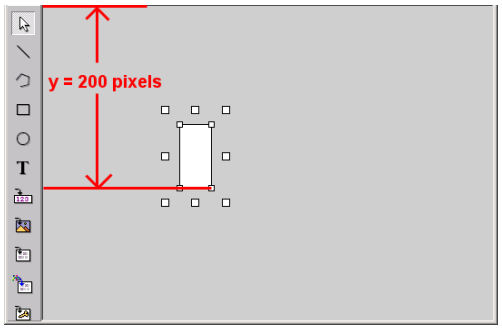
As a solution best practice, resizing: 1) grouped view elements, 2) templates, or 3) embedded views (which include key views) is not recommended. Resizing affects every element in the view proportionally, which may result in undesired line widths or font sizes. Consider using a different view or an animation to get the desired behavior.

Determining Position and Size

On the Drawing panel, the screen origin (X=0, Y=0) is located at the upper left corner. Therefore, the positive **x** direction moves to the right along a horizontal plane. While the positive **y** direction moves down along a vertical plane.

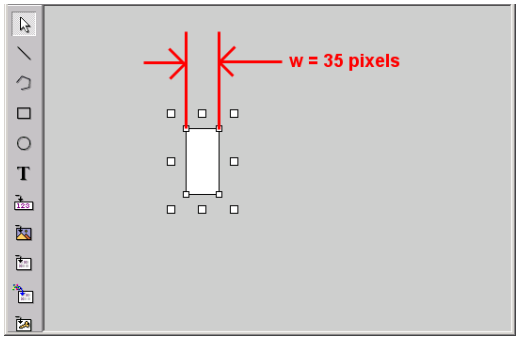


x-position

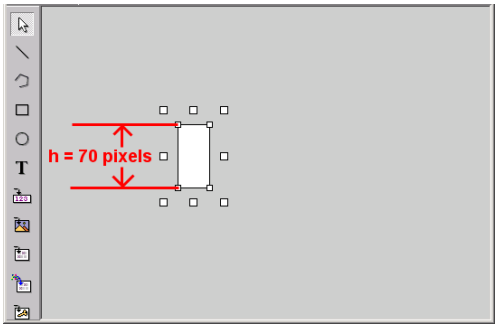


y-position

The **w** represents the width of the selected view element, and the **h** the height.



Width Dimension



Height Dimension

To size a view element using the mouse

1. Select the view element with the **Select tool**.
   The selection boundary handles appear.
2. **Drag one of the inner selection handles** in any direction to modify the size and/or shape of the view element.
   An outline of the modified view element appears.
3. **Release the mouse** when the view element is sized as desired.

To resize a view element proportionally

1. Select the view element with the **Select tool**.
   The selection boundary handles appear.
2. Hold down the **SHIFT** key, click on any corner, and drag to the desired size.

   Proportional resize does not apply to the following view elements: lines, polylines, buttons, edit boxes, and combo boxes.

To set a precise size for a view element

1. Select the view element with the **Select tool**.
2. Click the **General** property.
3. Enter the desired **w** (width) and/or **h** (height) for the view element.
   Width and height are specified in pixels.

Cutting, Copying, and Pasting

Cutting, Copying, and Pasting

Icon	Button Name	Function
	Delete	Deletes selection image(s).
	Cut	Removes the selection and copies it to the buffer (clipboard).
	Copy	Copies the selection to the buffer (clipboard). **Tip:** You *copy and paste styles among similar view elements*.
	Paste	Inserts the cut or copied view element from the buffer into the same view or a different one.

To cut and paste a view element selection

1. Select the view element(s) with the **Select tool**.
2. Click the **Cut** button.

   This removes the view element(s) you selected from the view and copies it to the buffer (clipboard).
3. Do one of the following, using the **Paste** button:
   - Insert the view element into another area in the same view.

   or
   - Insert the view element into another view.

To copy and paste a view element selection

1. Select the view element(s) with the **Select tool**.
2. Click the **Copy** button.

   This copies the view element(s) you selected to the buffer (clipboard).
3. Do one of the following, using the **Paste** button:
   - Insert the view element into another area in the same view.

   or
   - Insert the view element into another view.

To delete a view element selection

1. Select the view element(s) with the **Select tool**.
2. Click the **Delete** button.

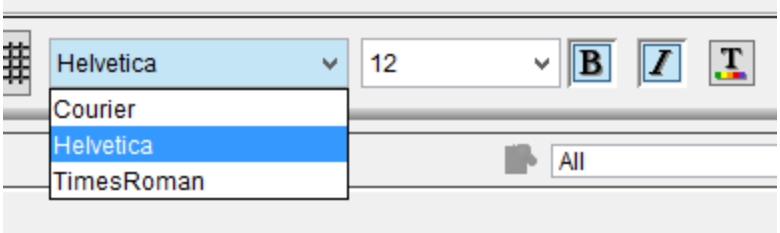
   This deletes the view element(s) you selected.

To duplicate view elements

1. Select the view element(s) with the **Select tool**.
2. Press **CTRL+D**.
   This automatically duplicates the view element(s) you selected.

Setting the Text Style

The following options set the style of the selected text element and are available from the **Button Bar**.

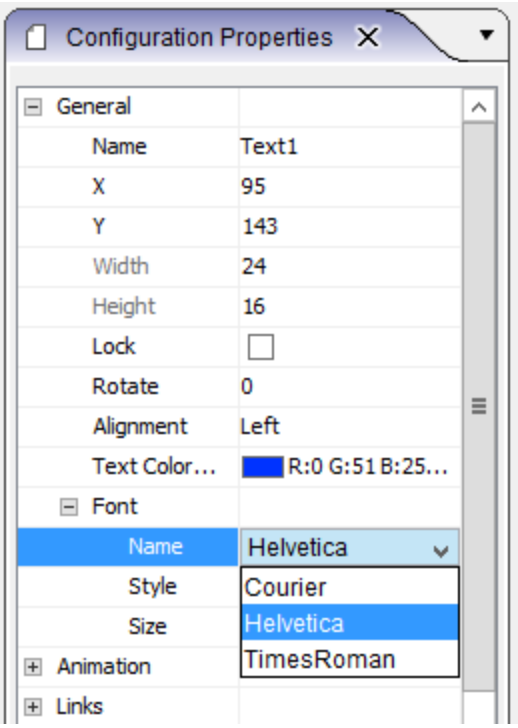


Button Bar Font-related Options

Font Style Buttons

Icon	Button Name	Function
	Text color	Applies color to the text you select. **See Also:** For more information on how to set the text color, go to the topic, *Working with Color*.
	Bold	Makes text **bold**.
	Italic	Makes text *italic*.

You can also set font styles from the **Configuration Properties tab**.



Font-related Configuration Properties

By default, the fonts available from the XHQ Workbench are Courier, Helvetica, and Times Roman.

To add fonts to the font list

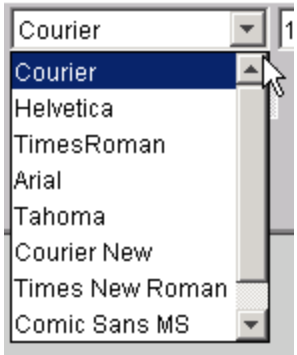
For more information, go to the topic, *Using globaleditingsettings.properties*, located in the XHQ Administrator's Guide.

1. Close all open XHQ Workbench sessions.
2. Edit the `globaleditingsettings.properties` file, which is typically located in the `<install directory>:\XHQ\data\repos` folder on the server.
3. Add the `ViewBuilder_fonts` property and set its value to a comma-separated font list.

   Example:

   ```
   ViewBuilder_fonts=Arial,Tahoma,Courier New,Times New Roman,Comic Sans MS
   ```

4. Save the `globaleditingsettings.properties` file.
5. Open a view.
   The added fonts now appear in the drop-down fonts list after the default fonts and in the order they were listed in the property value.



Aligning Text

To set the horizontal alignment for text, click on the following buttons.

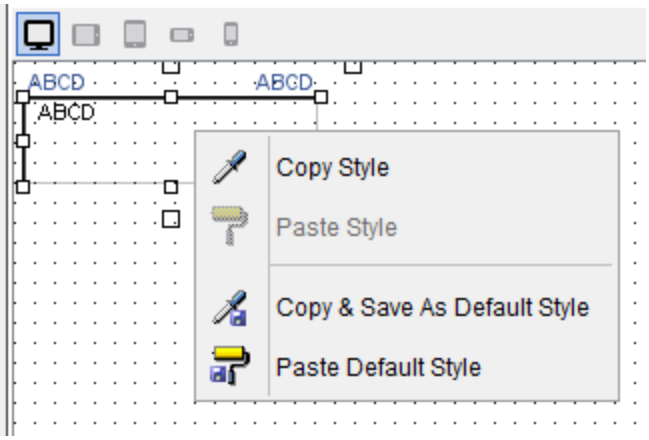
Text Alignment Buttons

Icon	Button Name	Function
	Align Left	Horizontally aligns the **left** edge of each text line.
	Align Center	Horizontally aligns the **center** of each text line.
	Align Right	Horizontally aligns the **right** edge of each text line.

The text alignment options are only enabled when inserting Text- or Value-type view elements.

Copying and pasting a View Element style

You can copy and paste styles among *similar* view elements. For example, you can copy the style of an existing value item and paste it to a new value item. Simply right-click on an existing view element from which you want to copy the style. A shortcut menu appears with the following options.



Right-click Shortcut Menu - Copy and Paste Style

Icon	Option	Function
	Copy Style	Copies the style attributes defined for the selected view element. If this option is disabled, then no style has been applied to the given view element.
	Paste Style	Pastes the copied view element styles to the selected view element. If this option is disabled, then no style as been copied.
	Copy & Save As Default Style	Copies the style attributes defined for the selected view element and saves the collective style as the default.
	Paste Default Style	Pastes the default style to the selected view element.

The following view element types support the copy/paste style feature. Each table lists the various styles that can be copied and pasted.

Text View Element Styles

Text Styles	Default Value
Text Color (RGB)	0,0,0 (Black)
Font Name	Helvetica
Font Size	12

Text Styles	Default Value
Font Style	Bold
Text Alignment	Left align

Value View Element Styles

Value Styles	Default Value
Text Color	0,0,0 (Black)
Font Name	Helvetica
Font Size	12
Font Style	Bold
Text Alignment	Left
Quality Bad Color	255,0,0 (Red)
Quality Uncertain Color	255,255,0 (Yellow)
Decimal Place	0
Number Format	General
Group Digits	Disabled
Time Format	hh:mm:ss
Date Format	MM/dd/yyyy
Unit Color	0,0,0
Unit Style	Bold

Table View Element Styles

Table Styles	Default Value
Row Height	20 (pixels)
Auto Fit	Disabled
Wrap Text	Disabled
Background Color	192,192,192 (Gray)
Header Background Color	192,192,192
Footer Background Color	192,192,192
Selected Row Background Color	0,0,112 (Blue)
Selected Row Text Color	255,255,255 (White)
Even Row Background Color	255,255,255
Odd Row Background Color	255,255,255
Table Border Line Color	255,255,255

Table Styles	Default Value
Grid Line Color	192,192,192
Show Horizontal Grid Line	Enabled
Show Vertical Grid Line	Enabled

Default Column View Element Styles

Default Column Styles	Default Value
Header Alignment	Left
Header Font Name	Helvetica
Header Font Style	Bold
Header Font Size	12
Header Text Color	0,0,0 (Black)
Data Alignment	Left
Data Font Name	Helvetica
Data Font Style	Bold
Data Font Size	12
Data Text Color	0,0,0
Fix Width	False
Column Width	100 (pixels)
Format Pattern	-

Default Column View Element Styles - Number

Number Column Styles	Default Value
Header Alignment	Left
Header Font Name	Helvetica
Header Font Style	Bold
Header Font Size	12
Header Text Color	0,0,0 (Black)
Data Alignment	Left
Data Font Name	Helvetica
Data Font Style	Bold
Data Font Size (FORMAT_GENERAL)	12
Data Text Color	0,0,0

Number Column Styles	Default Value
Fix Width	False
Column Width	100 (pixels)
Format Pattern	-

Default Column View Element Styles - Decimal

Decimal Column Styles	Default Value
Header Alignment	Left
Header Font Name	Helvetica
Header Font Style	Bold
Header Font Size	12
Header Text Color	0,0,0 (Black)
Data Alignment	Left
Data Font Name	Helvetica
Data Font Style	Bold
Data Font Size (FORMAT_GENERAL)	12
Data Text Color	0,0,0
Fix Width	False
Column Width	100 (pixels)
Format Pattern	-

Default Column View Element Styles - Date/Time

Date/Time Column Styles	Default Value
Header Alignment	Left
Header Font Name	Helvetica
Header Font Style	Bold
Header Font Size	12
Header Text Color	0,0,0 (Black)
Data Alignment	Right
Data Font Name	Helvetica
Data Font Style	Bold
Data Font Size (FORMAT_GENERAL)	12
Data Text Color	0,0,0

Date/Time Column Styles	Default Value
Fix Width	False
Column Width	100 (pixels)
Format Pattern	-

Line View Element Styles

Line Styles	Default Value
Color (RGB)	0,0,0 (Black)
Width	3 (pixels)
Style	Solid line
Arrowhead	No arrowhead
Arrowhead Style	Open arrow

Polyline View Element Styles

Polyline Styles	Default Value
Line Color	0,0,0 (Black)
Line Width	1 (pixel)
Line Style	Solid line
Arrowhead	No arrowhead
Arrowhead Style	Open arrow
Filled	False
Fill Color	255,255,255 (White)

Rectangle View Element Styles

Rectangle Styles	Default Value
Line Color	0,0,0 (Black)
Line Width	1 (pixel)
Line Style	Solid line
Filled	False
Fill Color	255,255,255 (White)

Ellipse View Element Styles

Ellipse Styles	Default Value
Line Color	0,0,0 (Black)
Line Width	1 (pixel)

Ellipse Styles	Default Value
Line Style	Solid line
Filled	False
Fill Color	255,255,255 (White)

Edit Box View Element Styles

Edit Box Styles	Default Value
Text Color	0,0,0 (Black)
Font Name	Helvetica
Font Size	12
Font Style	Bold
Text Alignment	Left align
Filled	True
Fill Color	255,255,255 (White)

Combo Box View Element Styles

Combo Box Styles	Default Value
Text Color	0,0,0 (Black)
Font Name	Helvetica
Font Size	12
Font Style	Bold
Text Alignment	Left align
Filled	True
Fill Color	255,255,255 (White)

Button View Element Styles

Button Styles	Default Value
Text Color	0,0,0 (Black)
Font Name	Helvetica
Font Size	12
Font Style	Bold
Text Alignment	Left align
Filled	True
Fill Color	192,192,192 (Gray)

Text Area View Element Styles

Text Area Styles	Default Value
Text Color	0,0,0 (Black)
Font Name	Helvetica
Font Size	12
Font Style	Bold
Text Alignment	Left align
Filled	True
Fill Color	192,192,192 (Gray)
Vertical Scroll Bar	Disabled
Bad Quality Color	255,0,0 (Red)
Uncertain Quality Color	255,255,0 (Yellow)

List View Element - Checkbox Styles

Checkbox Styles	Default Value
Font Name	Helvetica
Font Size	12
Font Style	Bold
Text Alignment	Left align
Text Color	0,0,0 (Black)
Filled	True
Fill Color for each Checkbox	255,255,255 (White)
Control Size	12 (points)
Control Color	255,255,255
Control Position	Left
Enable Select All	False
Inner Margin	14 (pixels)
Outer Margin	14 (pixels)
Orientation	Vertical
Mode	Auto Expand
Gap	12 (pixels)
Vertical Align ("Scrollable" Mode)	Bottom

Checkbox Styles	Default Value
Mark Color	0,0,0
Fill Color for entire Control	255,255,255

List View Element - Radio Styles

Radio Styles	Default Value
Font Name	Helvetica
Font Size	12
Font Style	Bold
Text Alignment	Left align
Text Color	0,0,0 (Black)
Filled	True
Fill Color for each Radio button	255,255,255 (White)
Control Size	12 (points)
Control Color	255,255,255
Control Position	Left
Inner Margin	14 (pixels)
Outer Margin	14 (pixels)
Orientation	Vertical
Mode	Auto Expand
Gap	12 (pixels)
Vertical Align ("Scrollable" Mode)	Bottom
Mark Color	0,0,0
Fill Color for entire Control	255,255,255
Select First Radio	False

Trend Element Styles

Trend Styles	Default Value
Text Color	0,0,0
Font Name	Helvetica
Font Size	12
Date Format	MM/dd/yy
Time Format	hh:mm:ss

Trend Styles	Default Value
Filled	True
Fill Color	255,255,255
Left Scale Label Color	0,0,0
Left Scale Color	0,0,0
Right Scale Label Color	0,0,0
Right Scale Color	0,0,0
X-Label Color	0,0,0
Border Color	0,0,0
Background Color	255,255,255
Left Margin Min	0
Right Margin Min	0
Hide Horizontal Grid Line	False
Hide Vertical Grid Line	False
Pen Color 0	0,0,0
Pen Color 1	0,0,255
Pen Color 2	232,0,232
Pen Color 3	0,160,0
Pen Color 4	176,176,0
Pen Color 5	128,128,128
Pen Color 6	0,176,176
Pen Color 7	172,0,0
Pen Color 8	63,8,91
Pen Color 9	0,100,100
Pen Color 10	255,0,0
Pen Color 11	255,0,255
Pen Color 12	211,211,211
Pen Color 13	0,255,0
Pen Color 14	255,165,0
Pen Color 15	255,105,180

Chart Element Styles

Chart Styles	Default Value
Chart Background Color	255,255,255
Plot Background Color	211,211,211
Grid Line Color	0,0,255
Minor Grid Line Color	255,255,255
Domain Text Color	0,0,0
Domain Font Name	Helvetica
Domain Font Size	12 (points)
Domain Font Style	Normal
Range Font Color	0,0,0
Range Font Name	Helvetica
Range Font Size	12 (points)
Range Font Style	Normal
Show Plot Border	True
Series Color 0	0,0,255
Series Color 1	255,0,0
Series Color 2	0,255,0
Series Color 3	255,165,0
Series Color 4	255,182,193
Series Color 5	0,255,255
Series Color 6	255,0,255
Series Color 7	255,255,0
Series Color 8	169,169,169
Series Color 9	0,0,0
Series Color 10	211,211,211

Tab Container View Element - Horizontal Layout Styles

Horizontal Tab Styles	Default Value
Tab Width	Determined by the length of the tab name and font size.
Tab Gap	4 (pixels)
Tab Background Color	192,192,192
Tab Text Color	0,0,0

Horizontal Tab Styles	Default Value
Selected Tab Background Color	114,160,193
Selected Tab Font Color	255,255,255
Tab Font Name	Helvetica
Tab Font Size	12 (points)
Tab Font Style	Bold

Tab Container View Element - Vertical Layout Styles

Vertical Tab Styles	Default Value
Tab Width	Determined by the height of selected tab font.
Tab Gap	4 (pixels)
Tab Background Color	192,192,192
Tab Text Color	0,0,0
Selected Tab Background Color	114,160,193
Selected Tab Font Color	255,255,255
Tab Font Name	Helvetica
Tab Font Size	12 (points)
Tab Font Style	Bold

Customizing Tooltips

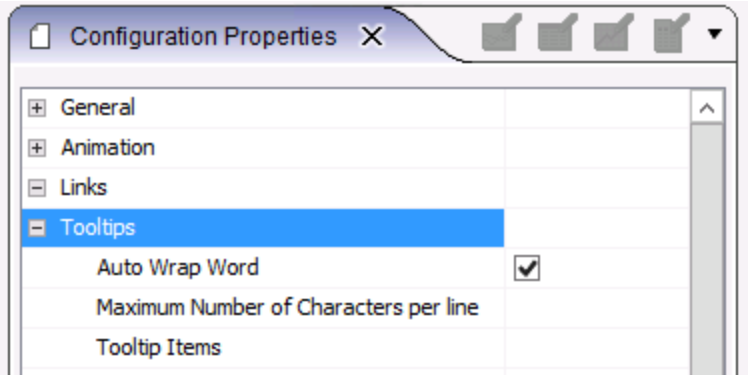
Customized Tooltips allow you to maximize the use of limited view space. For example, you can show Targets, Limits, descriptions, or units of measure for any view element in a Tooltip (an alternative to permanently placing these values onto the view itself).

Another advantage of customized Tooltips is the ability to provide details of underlying elements that go into a KPI calculation without forcing the end-user to navigate through numerous pop-up windows. Custom Tooltips can also alleviate the frustration of losing filter selections that occur by drilling down into a view. You simply provide these details in a custom Tooltip that appears when the user hovers the pointer over the given view element.

The behavior of Tooltips is the same in the browser runtime environment as it is in the XHQ Solution Builder.

About Custom Tooltips

From the XHQ Workbench, selecting a view element enables the **Tooltips property**.

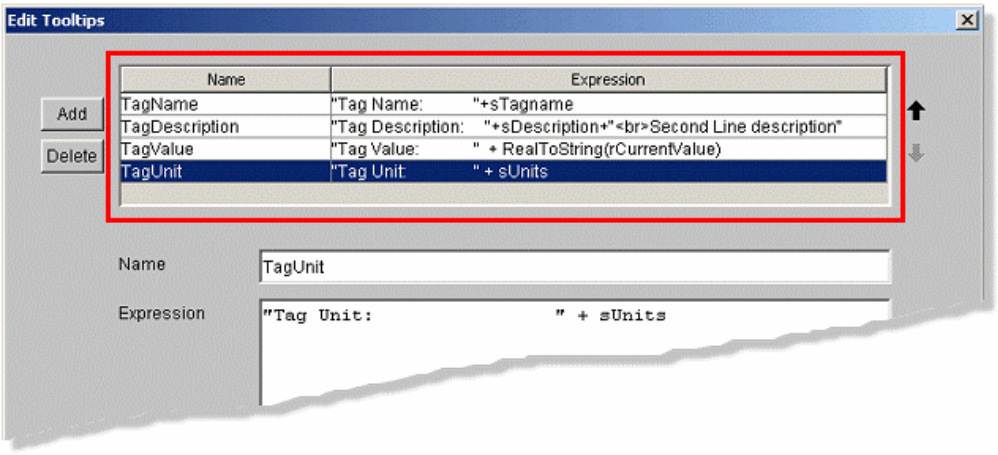


Customized Tooltips are composed of multiple lines. Each Tooltip line item has a **Name** and an **Expression** associated with it. The Name is a label that describes the Expression. The Expression is an XHQ expression that is evaluated when the mouse hovers over the given view element at runtime.

The Name label is not used during runtime.

The order in which the customized Tooltip displays each line during runtime is determined by the order in which they are added during design time.

For example, the following list is configured from the **Edit Tooltips** dialog box:\

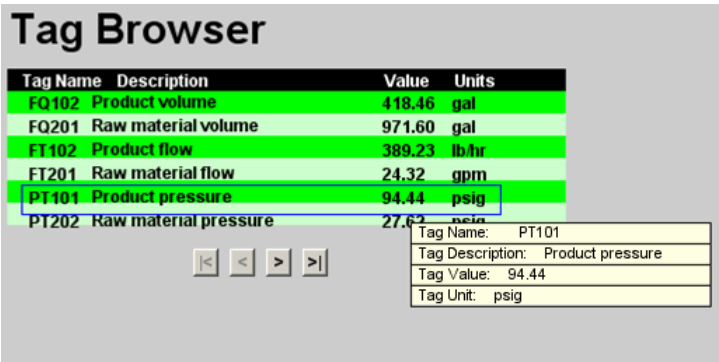


Edit Tooltips: Items List

According to this example list, the item order is as follows:

1. TagName
2. TagDescription
3. TagValue
4. TagUnit

As a result, the following customized Tooltip appears when the pointer hovers over the given view element during runtime:



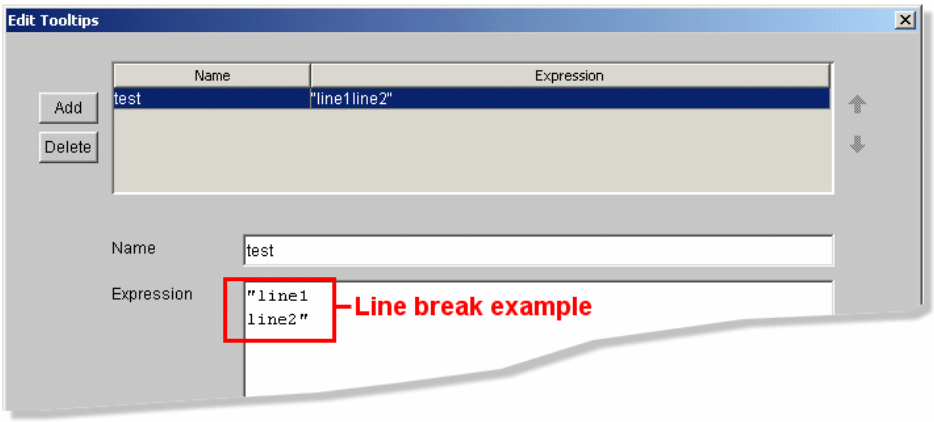
Example: Customized Tooltip

To manage long lines of text, use either of these methods:

- **Auto word wrap**
  From the XHQ Workbench, in the Tooltips property, select this checkbox to enable automatic word wrapping. Once selected, a secondary option appears, allowing you to set the maximum number of character number per line.

  For more information, see the topic, *More about auto word wrapping*.

- **Line break**
  From the Edit Tooltips dialog box, in the Tooltip item expression, you can set a line break by pressing ENTER.



More about automatic word wrapping

If the "Auto word wrap" option is selected, you must set the maximum number of characters per line.

> The **Maximum character number per line option** is only visible if the **Auto word wrap** option is selected (checked).

During runtime, a line break is forced based on this maximum number, without breaking a whole word. The only case when a word is broken is if the length of the whole word is longer than the maximum number. If the maximum number is set to 0 (zero), or if the text box is left blank, then word wrapping is disabled (not performed).

Both the "Auto word wrap" and the "Maximum character number per line" preferences are persisted (continued) to the next Tooltip you configure (even to the next session of the XHQ Workbench). For example, you set auto word wrap with a maximum number of 25 characters per line for the Tooltip of a text element. Then, you configure the Tooltip for a view element. The "Auto word wrap" option is automatically checked and the maximum number of 25 is repeated from your previous Tooltip configuration for the text element.

How expressions are evaluated

An important thing to note about Tooltip expressions is that they do not need to be of the type String.

Example: The following is a **valid** Tooltip item expression:

```
"Tag Value:   " + format(rCurrentValue, "0.###")
```

Tooltip expressions are initialized when the view is initialized at runtime. In addition, expressions are only evaluated at the time the Tooltip is displayed, which is when the pointer hovers over the view element to which the expressions are configured.

If the Tooltip expression does not include a late-bound method reference, then the Tooltip expression is evaluated right before it is displayed. However, if the Tooltip expression does include a late-bound method reference, then the expression is evaluated each time the reference (or references, if multiple are included) changes.

Should the quality of the expression value be BAD, the value that appears for the Tooltip item are red asterisks (****).

For more information, go to the topic, *About Quality*.

USING FORMATTING FUNCTIONS FOR EXPRESSIONS

Consider the following example:

```
"Tag Value:   " + RealToString(rCurrentValue)
```

It is important to note that functions such as **`RealToString`** and **`DoubleToString`**, for example, return **raw data** in string format. These functions are useful for passing values from the applet to JavaScript, but are not useful in a Tooltip.

The problem with these functions is that raw data values can potentially have several digits trailing a decimal point. So, a raw data value of "21.2767888999999923" displayed in a Tooltip at runtime may be excessive and unnecessary for the end-user.

To address this issue, use the **`format()`** function, as seen in the following examples:

- `String format(float number, string formattingPattern)`
- `String format(double number, string formattingPattern)`
- `String format(decimal number, string formattingPattern)`

Where:

Parameter	Description
`number`	Is the number to be formatted.
`formattingPattern`	Is the formatting pattern that supports the standard number formatting characters such as the pound (#) sign. **Web Reference:** For a list of supported standard number formatting characters, go to: `http://docs.oracle.com/javase/7/docs/api/java/text/DecimalFormat.html`
Returns:	A formatted number that is rounded up.
Example:	Given: `format(21.2767888999999923, "0.###")` Returns: `"21.277"`

These number formatting functions are supported in both client- and server-side expressions.

So, the previous expression (*incorrect* version):

```
"Tag Value:   " + RealToString(rCurrentValue)
```

Now becomes (*correct* version):

```
"Tag Value:   " + format(rCurrentValue, "0.###")
```

How Tooltips are prioritized in XHQ

In general, during a mouse-over event, XHQ Tooltips (configured or not) are checked in this order:

- First, XHQ checks for and uses **customized** Tooltips;
- If none exist and the given view element is a value item, then XHQ defaults to the **VTQ** (value, time, quality) Tooltip for the value item.

  For more information on VTQ (or value) Tooltips, go to the topic, *About Tooltips*.

- Lastly, if neither of these previous cases pertain, then XHQ uses the Tooltip for the **parent container** of the given view element. If the parent container does not have a Tooltip and the previous two cases pertain, then no Tooltip displays for the view element.

The Exported XML

The following is an example of the exported XML for a customized Tooltip.

```
<XHQ:TooltipData
auto_word_wrap="true"
max_char_num_per_line="21">
<XHQ:TooltipItem
name="line1">
<XHQ:TooltipItemValueExpr>
<XHQ:Expression
data="&quot;aaaaa bbbbb ccccc ddddd&quot;"
/>
</XHQ:TooltipItemValueExpr>
</XHQ:TooltipItem>
<XHQ:TooltipItem
name="line2">
<XHQ:TooltipItemValueExpr>
<XHQ:Expression
data="&quot;aaaaabbbbbcccccdddddeeeee&quot;"
/>
</XHQ:TooltipItemValueExpr>
</XHQ:TooltipItem>
</XHQ:TooltipData>
```

Arranging Your View Elements

Several features of the draw environment enable you to work more easily with groups of view elements. These features, collectively called arranging, include grouping, ordering, and aligning. You can find the grouping and ordering tools on the Button Bar. The aligning tools are located in the General property.

Working with Grouped View Elements

Grouping is a mechanism for locking multiple view elements together. You can use a group to nest multiple groups for organization, for retaining a collection of complex elements and their spatial layout, and to prevent accidental operations on parts of a group. Alternatively, grouping allows you to manipulate many view elements as a single unit.

As a solution best practice, resizing grouped view elements is not recommended since it may result in an undesired sizing of the view. Resizing the group affects every element in the view (including text and lines) proportionally. If a group of view elements needs to have a dynamic resizing behavior, build it with animation to accommodate this behavior.

You cannot group a view within a flow view.

To group view elements

1. Perform a multiple selection of view elements by doing one of the following:
   - Hold down the **Shift** key while you click on each of the objects you want to include in the group.

   or

   - Click the mouse and drag the **Select** tool diagonally across the screen, fully surrounding the view elements you want to select with the "rubber-band" cursor. When you release the mouse button, the surrounded view elements will be selected.
2. Click the **Group** icon on the button bar.

To ungroup view elements

1. Click the **Select tool**.
2. **Select the group** by clicking anywhere on any member of the group.
   The selection boundary appears with selection handles around the extremities of the group.
3. Click the **Ungroup** icon on the button bar.

To change the membership of a group of view elements

1. Click the **Select tool**.
2. **Select the group** by clicking anywhere on any member of the group.
   The selection boundary appears with selection handles around the extremities of the group.
3. Click the **Ungroup** icon.

4. Hold down **Shift** and do one of the following:
   - Click on new members one at a time to add them.

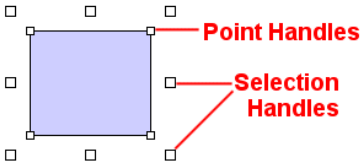
   or

   - Click on members to remove them.
5. Click the **Group** icon on the button bar.

Editing Nested View Elements

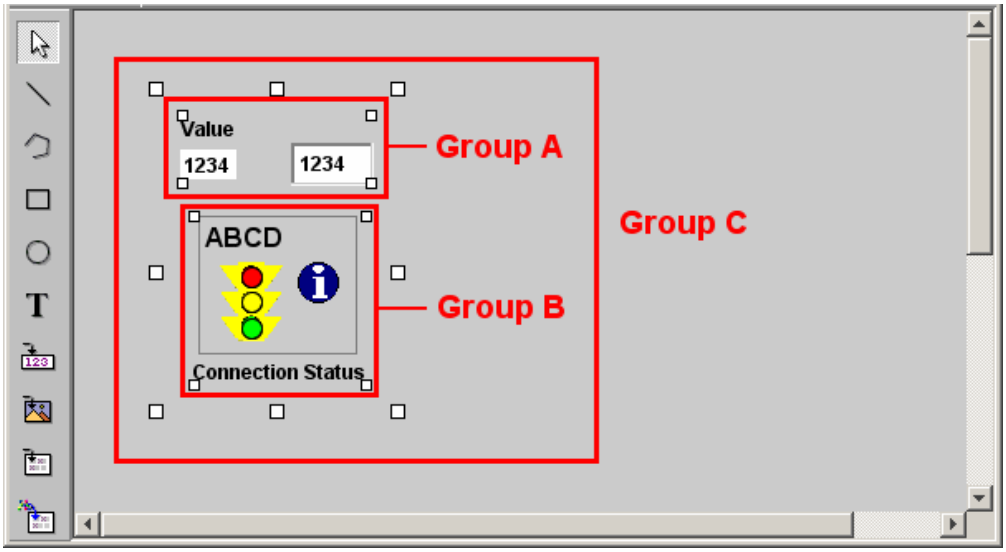
In this topic, you will learn how to edit view elements within nested groups WITHOUT having to ungroup the set of view elements.

About Handles
When you select a group of view elements, eight **outer selection handles** appear outside of the group's boundary, and four **inner point handles** appear at the corners of the group's boundary.



To select (to edit) a single element within a nested group, you simply place the cursor on top of the view element you want to edit. Then, you click through the view elements until the view element you want to edit is highlighted (or selected).

To illustrate this process, consider the following example.



Group C consists of these nested groups:

- ***Group A***, which contains:
  a. A nested group that includes the text element "Value" and a value element;
  b. An edit box control.
- ***Group B***, which contains:
  a. An image element;
  b. The locked text element "Connection Status".

In this example, you want to edit the text element in Group A to display "Rate" instead of "Value".

1. **Place the cursor** over the "**Value**" text element in Group A.
2. **Click** until the "**Value**" text element is **selected**.
   White [inner] point handles indicate which element is selected (active).

The "Click" sequence is as follows.

On this click . . .	This group or view element is selected . . .
First	Group C
Second	Group A
Third	The nested group that includes the text element "Value" and a value element
Fourth	The text element "Value"

More about Handles
Notice the placement and the color of the selection and point handles during each "Click" action above. When clicking through the nested view elements, the outer selection handles outline the parent group's boundary. After the first click, the selection handles turn gray. Alternatively, the inner point handles define the border of the actual group or view element that is selected and are white to show that the chosen element is in active (selected) state.
If the position or size of the selected view element is altered, the bounding handles of all parent groups are recalculated accordingly.

3. Click the **Text tab**.
4. In the text box, replace "Value" with **Rate**.
5. **Save** to repository.

Additional Things to Note when Editing Nested View Elements

- Certain commands are disabled during the editing process: delete, cut, copy, ungroup, and rotate. In addition, you will not be able to change the stack order by moving view elements forwards or backwards.
- To reposition a nested view element that is in active state, do any of the following.
  - From the General property, edit the position and/or size (x, y, w, h).
  - Drag the group or the nested view element to a new position, which changes the x and y values.
  - Press the up, down, left, or right arrow keys on your keyboard to move the group or nested view element to a new position.
  - For nested images, drag the inner point handles to modify the image size, which changes the w and h values.
- When you include a locked view element as part of a group, the Lock attribute is removed. The view element will move or resize with the group.

Repositioning and Resizing Nested View Elements

There are four ways to reposition and/or resize a selected (active) view element that is nested within a group:

- Change the x, y, w, and h values from the General property;
- Change the x- and y-coordinates by dragging the view element;
- Change the x- and y-coordinates by using the up, down, left, or right arrow keys;
- Change the w and h values by dragging the inner point handles.

As a solution best practice, resizing: 1) grouped view elements, 2) templates, or 3) embedded views (which include key views) is not recommended. Resizing affects every element in the view proportionally, which may result in undesired line widths or font sizes. Consider using a different view or an animation to get the desired behavior.

Physical vs. Scaled Coordinates

When editing an active nested view element, notice that there are two sets of x, y, w, and h values.

The first set of values (to the left) are the **physical coordinates**. The second set (to the right) are the **scaled coordinates**, which are the physical coordinates with a scale factor applied. The scale factor is determined by the outer group's coordinates. The scaled coordinates are the same as the physical coordinates when the scale factor is 1.0.

To reposition, edit the physical coordinates. The view element's scaled coordinates will change accordingly to reflect the scale factors of the outer groups.

If you resize the outer group by doubling the width (w), then the scale factor is 2.0. Note the differences between the physical and the scaled coordinates in the following image. The physical coordinates to not change, but because of the scale factor of 2.0, the scaled coordinates are doubled.

Repositioning

There are two ways to reposition an active grouped element:

- By using the **arrow keys**
- By placing the cursor within the boundary of the inner handles and **dragging** the object

Use the arrow keys on your keyboard to nudge the view element in increments of 1 pixel. Hold down the CONTROL key and use the arrow keys to nudge the view element in increments of 5 pixels. Hold down the SHIFT key and use the arrow keys to nudge the view element in increments of 10 pixels.

When you reposition the active grouped element by "n" number of pixels, the physical coordinates are recalculated. If the scale factor is not equal to 1.0, then the physical coordinate values are rounded to the nearest whole number. It is important to note that in this case, the grouped element may move more or less than "n" pixels.

Resizing by dragging

When you resize the view element by dragging the inner point handles from (x_1,y_1) to (x_2,y_2), the physical coordinates are recalculated accordingly. Again, if the scale factor is not equal to 1.0, then the physical coordinate values are rounded to the nearest whole number. Therefore, the grouped element may not be positioned exactly at (x_2,y_2) when you release the cursor.

How to Order View Elements

View elements are nominally stacked from back to front in the order that you create them. When you adjust the order of overlapping view elements, a change in visual effects may result. The order functions let you change the stacking order of view elements using simple commands such as move backwards, or move forwards.

Grouped view elements assume the relative position of the top view element.

Ordering Icons

Icon	Button Name	Function
	Move Backward	Moves the view element towards the bottom of the stack by one position
	Move Forward	Advances the view element towards the top of the stack by one position
	Bring to Front	Brings the view element to the top of the stack
	Send to Back	Brings the view element to the bottom of the stack

To order view elements

1. Click the **Select tool**.
2. Select a view element or group.
3. Do one of the following to change the stacking order:
   - Click the **Move Backward** icon to move the view element towards the bottom of the stack by one position.

   or

   - Click the **Move Forward** icon to advance the view element towards the top of the stack by one position.

   or

   - Click the **Bring to Front** icon to bring the view element to the top of the stack.

   or

   - Click the **Send to Back** icon to bring the view element to the bottom of the stack.

How to Align View Elements

The drawing environment provides alignment options that allow you to easily arrange selections of view elements. When you select any view element, a selection boundary appears around it. The align option uses the rectangles contained by the selection handles. For example, **Tops** will arrange all selected view elements so that the tops of the selection boundaries match up precisely. Wherever you perform an alignment, it is applied against the most extreme view element. For example, in the **Tops** example, all the view elements will align with the topmost view element.

View Elements Alignment Buttons

Icon	Button Name	Function
	Lefts	Aligns the objects you select by their **left edges**.
	Centers (Vertically)	Aligns the objects you select **vertically** by their **centers**.
	Rights	Aligns the objects you select by their **right edges**.
	Tops	Aligns the objects you select by their **top edges**.
	Middles (Horizontally)	Aligns the objects you select **horizontally** by their **middles**.
	Bottoms	Aligns the objects you select by their **bottom edges**.

To align a series of two or more view elements

1. Click the **Select tool**.
2. **Select two or more view elements**.
   The align icons appear in the main toolbar.
3. Click the appropriate **icon** to align the view elements.

How to Resize View Elements

You can resize view elements by dragging the **inner** point handles of a view element. When you release the mouse, the shape appears at its new size.

As a solution best practice, resizing: 1) grouped view elements, 2) templates, or 3) embedded views (which include key views) is not recommended. Resizing affects every element in the view proportionally, which may result in undesired line widths or font sizes. Consider using a different view or an animation to get the desired behavior.

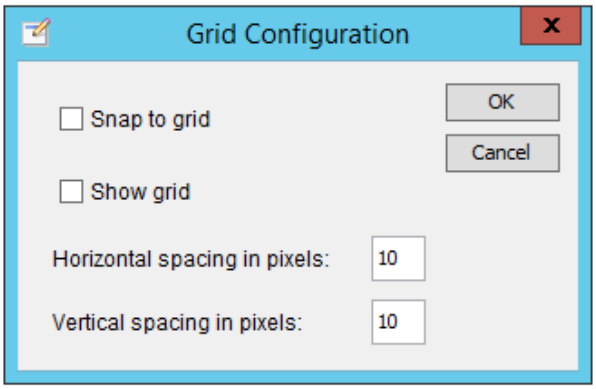
Currently, you cannot resize text or value items.

When resizing, the view elements automatically scale to fit the new dimension. Consider the following rules when resizing embedded views (again, as stated above, resizing embedded views is not recommended).

- Groups, templates, and embedded views cannot be resized smaller than the minimum size constraints of different classes of draw items.
- Size constraints are also "inherited" from embedded elements. For example, a group has a button with a minimum size of 20. This group is then grouped. The "outer" group then also has a minimum size constraint of 20, since the "inner" group had the constraint from the button.
- Each item may have a minimum height or width constraint, such as a button.
- Some items have a minimum size constraint dependent on the data entered at runtime (for example, a text item).

Using the Grid

Use the grid to align elements evenly on your view. The grid is displayed as equally spaced dots. You can enter horizontal and vertical values to set the distance between grid dots. This spacing is measured in pixels.

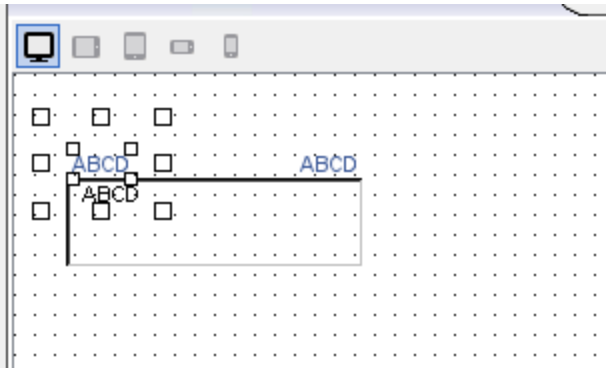


Grid Configuration dialog box

The **Snap-to-grid** option automatically aligns images to the grid. When you release a view element, its edges are aligned with the nearest grid "line".

The grid does not have to be visible in order to use the Snap-to-grid feature.

Grid preferences are persisted to the next session of the XHQ Workbench.

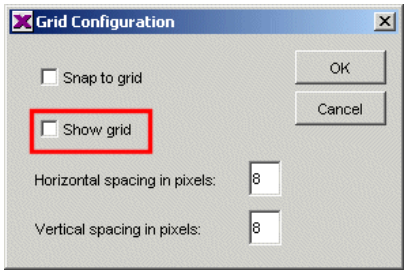


Example: Grid in View Drawing panel

To display or hide the grid

1. From the button bar, click the **Grid** button.

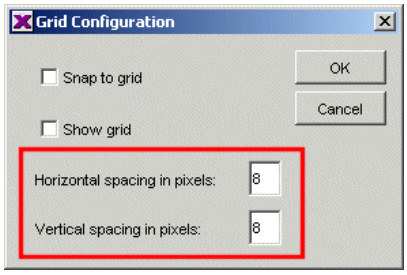
   The "Grid Configuration" dialog box appears.



2. Select to display, or deselect to hide, **Show Grid**.

To set grid spacing

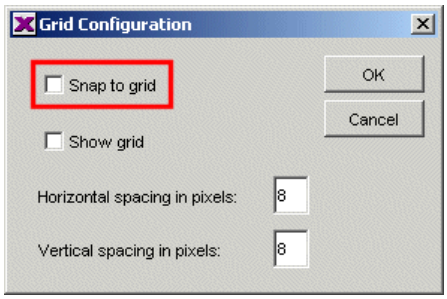
1. Click the **Grid** button.
   The "Grid Configuration" dialog box appears.



2. Under **Horizontal**, enter the horizontal measurement, in pixels, between grid dots.
   The default is 8 pixels.
3. Under **Vertical**, enter the vertical measurement, in pixels, between grid dots.
   The default is 8 pixels.

To enable or disable snap-to-grid

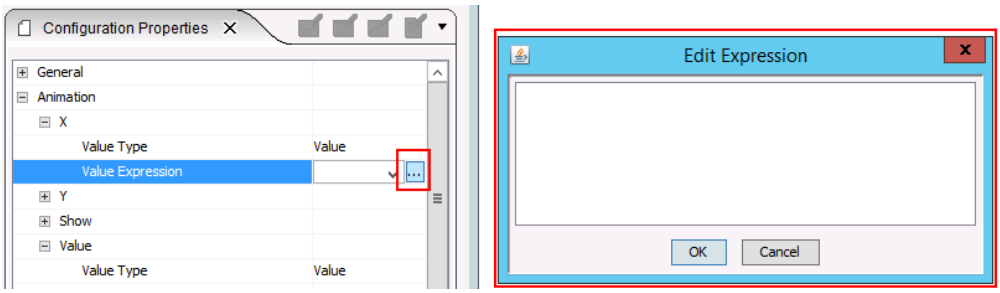
1. Click the **Grid** button.
   The "Grid Configuration" dialog box appears.



2. Select to enable, or deselect to disable, **Snap-to-grid**.

Viewing Expanded Expressions

When working with long expressions in the XHQ Workbench, use the Expanded Expressions feature to view and edit your expressions.



Example: The Edit Expression Button and Dialog box from the Animation Property

Clicking on the Edit button launches the "Edit Expression" dialog box. This dialog allows you to view, and edit, the entire expression.

This feature is enabled for the following view configuration properties.

Configuration Property	Description
Animation	**Expression** combo box
Links	**URL** type, **Address** text box **Action** link type, **Function** text box
View	With added **Filter**, **Expressions** combo box With added **Filter** and **Between** Relation, second **Expressions** combo box
Template	**Expression** combo box

Expression boxes that only allow constants or variable references do **not** support this feature.

Working with Color

You can apply colors to view elements that you have drawn, independently filling regular and irregular shapes and coloring text. When you draw a view element such as an ellipse or rectangle, the view element first appears with a black outline and a white fill color. The default color for text is black. Once you have drawn a view element or placed text, you can apply different color properties.

You can switch the fill properties of an existing view element between filled and unfilled using the fill color on/off icon.

An invalid color property value defaults to black.

From the Button bar, you can select fill and line color using the color icons.

View Element Color

Icon	Button Name	Function
	Set fill color	Selects and applies fill color to a selected view element.
	Enable fill	Toggles between fill and no fill for a selected view element.
	Set line color	Selects line or outline color for a selected view element.

The following options set the color of the selected text element and is available from the main toolbar.

Text Color

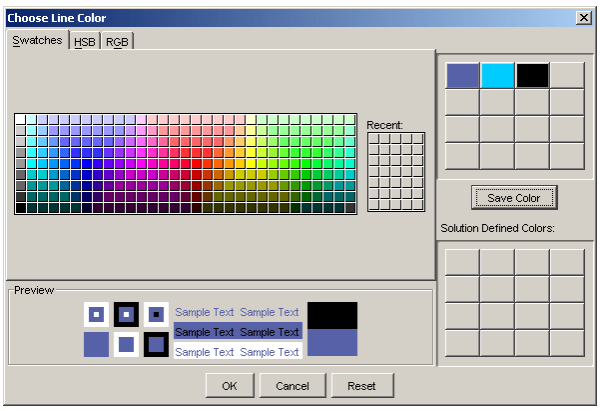
Icon	Button Name	Function
	Text color	Applies color to the text you select.

For both text and view elements, you select colors in the same way using the color selector.

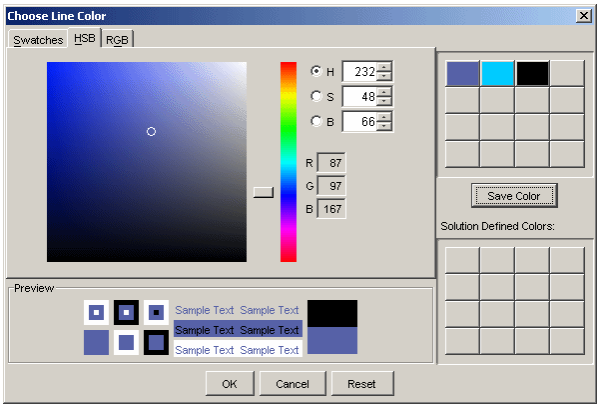
You cannot perform color operations on imported bitmaps.

Using the Color Selector

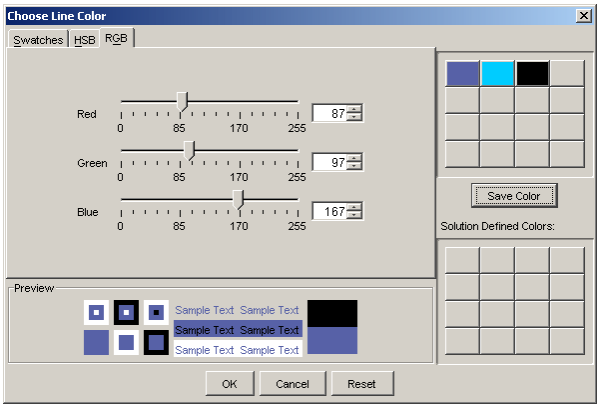
The same color selector appears whether you click on the text color, fill color, or outline color icon. The color selector provides three tabs: Swatches, HSB (Hue, Saturation, Brightness), and RGB (Red, Green, Blue).



Color Palette, Swatches Tab

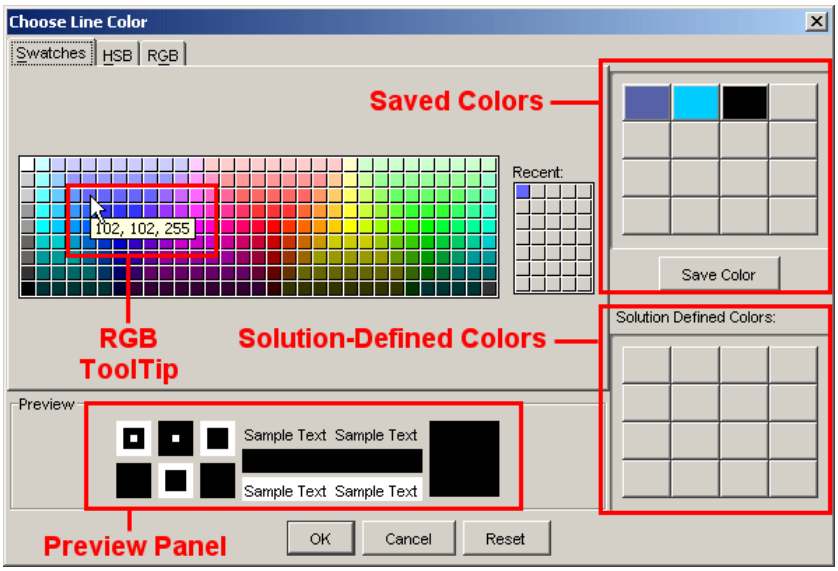


Color Palette, HSB Tab



Color Palette, RGB tab

The Preview panel at the bottom of all three palettes shows various applications of the current color selection. The Save Color palette to the right of the dialog box, allows you to create your own color palette, displaying up to sixteen of your "most used" colors.



Color Palette Panels

For more information on solution-defined colors, go to the topic, *About Solution Defined Colors*.

On the Swatches palette, a Tooltip appears when your mouse hovers over a color square, showing the RGB values for that particular color.

Make sure you test your color combinations for readability before you deploy your solution. For example, you may not want to use white for uncertain quality values if your background is white.

To change color fill properties for an existing view element

1. Select an existing square or rectangle view element with the **Select tool**.
2. Do one (or both) of the following:
   - Click the **Color fill on/off** tool.

   and/or

   - Click the **Color fill** tool to select a new color.

To change color for text, line, view element fill, or view element outline color

1. Select the view element or text that you want to color.
2. Do one of the following:
   - For text, select the text and then click the **Text Color** icon.
   - For an ellipse, rectangle, or irregular shape, click the **Select Fill** color icon.
   - For the outline color of an ellipse or rectangle, click the **Line Color** icon.

   The Swatches palette automatically displays. You can use the Swatches palette, the HSB palette, and/or the RGB palette to select the desired color. Each of these palettes is located under the corresponding tab.
3. Do one of the following:
   - To select a color using the swatches palette, click the Swatches tab, then click a color in the palette. When you point at any color swatch with the Select tool, the corresponding RGB values for that color display in the format (R, G, B).
   - To select a color using the HSB palette, click the HSB tab. You can select H (hue), S (saturation), or B (Brightness) and click anywhere on the color palette, or adjust the slider, to provide a wide variety of colors.

     Alternately, you can enter the desired Hue, Saturation, and Brightness values in the text boxes on the right of the palette.
     The RGB values for the selected color also display.
   - To select a color using the RGB palette, click the RGB tab, then adjust the red, green, and blue sliders to achieve the desired color.

   You can also click Cancel to cancel the operation, or Reset to restore the original color.
4. Click **OK**.

To save a color

1. Select or enter a color from any of the three palettes.
   You can use the Swatches palette, the HSB palette, and/or the RGB palette to select the desired color or colors you use often.
2. Click **Save Color**.
   The color appears on the Saved Colors palette. As you add more colors, the most recently saved color will always appear on the first square. You can save up to sixteen colors.

About Solution-Defined Colors

In addition to the 16 *user*-defined colors, there are 16 *solution*-defined colors that are set through the **`globaleditingsettings.properties`** file.

For more information, go to the topic, *Using globaleditingsettings.properties*, located in the XHQ Administrator's Guide.

Customizing Preferences

When you launch the XHQ Workbench for the first time, you will note that certain properties, such as the fill color, are pre-determined. These pre-determined, or default, values are referred to as the *XHQ defaults*.

There are two ways to customize these properties:

- By editing the **`clientsettings.properties`** file located in the folder specified by the environment variable `%APPDATA%XHQ\XHQ devClient\repos` on the user's local machine. These are the *User defaults*.

or

- By editing the **`globaleditingsettings.properties`** file located in the folder specified by the environment variable `%XHQ_SERVER_REPOS%` on the server. These are the *Solution defaults*.

Solution defaults override XHQ defaults; User defaults override Solution defaults.

Take for example, the fill color (background color) for the rectangle view element. When you first launch the XHQ Workbench after a brand new XHQ install, the fill color for the rectangle is set to white (with an RGB value of 255,255,255). This is the **XHQ default** value.

To set the **Solution default** value for the rectangle fill color, which overrides the XHQ default value, edit the **`globaleditingsettings.properties`** file by setting the `RectangleItem fillColor` property to an RGB value consistent with the look-and-feel of the particular solution.

Likewise, to set the **User default** value for the rectangle fill color, which overrides the Solution default, edit the **`clientsettings.properties`** file by setting the `RectangleItem fillColor` property to a different RGB value.

`globalsettings.properties` vs. `globaleditingsettings.properties`

The difference between these two files is that `globalsettings.properties` is used during runtime (XHQ Solution Viewer) and `globaleditingsettings.properties` is used during design time (XHQ Workbench).

clientsettings.properties

The **`clientsettings.properties`** file records your client property settings (such as font or color settings) and archives your most recent settings. It is not deleted during the new installation of the development client.

The development client creates the `clientsettings.properties` file "on-the-fly" when the product is used.

For more information, go to the topic, *About clientsettings.properties*, located in the XHQ Administrator's Guide.

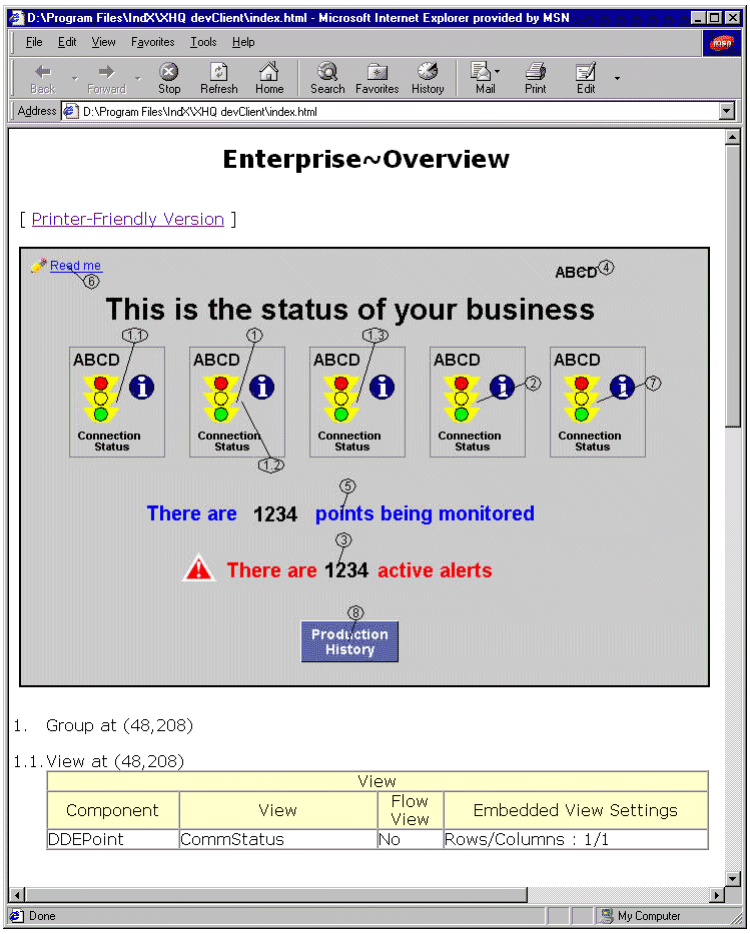
globaleditingsettings.properties

The **`globaleditingsettings.properties`** file allows you to customize properties (such as color palette, text style, and fill scheme) for standard view elements, namely graphics, used in developing content views for a solution.

For a list of properties, go to the topic, *Using globaleditingsettings.properties*, located in the XHQ Administrator's Guide.

Generating a View Details Report

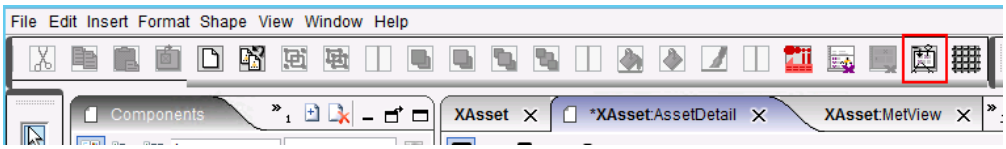
With XHQ, you can produce a report to help you keep track of all your animations, links, and views. Accessed through the XHQ Workbench, the View Details tool generates a detailed report of the view's configuration. The report is created as an `html` page that includes the view with *callouts* for animated elements and associated settings in a table.



View Details Report Example

To generate a View Details report

From the **XHQ Workbench**, on the Button toolbar, click the **View Details** icon.



The report is created as an `.HTML` page and opens in your browser.

The default location is:

`%APPDATA%\XHQ\XHQ devClient\repos\summaries\<viewname>.html`

To print the View Details report

1. With the report displayed in your browser, go to the **File** menu and click **Print**.
   The "Print" dialog appears. The default printer is automatically selected.
2. Click **Print**.

Using View Search

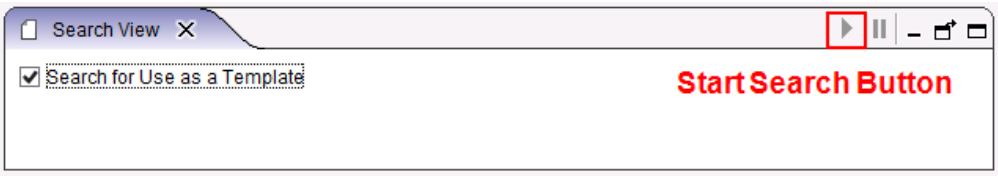
With the Search View tab, you can search through the XHQ Workbench to find all the places where a particular component view is embedded, or even used as a template. Search is available on all views, including inherited views from the base component.

For more information about base components, go to the topic, *About Base and Subclass Components*.

When a reference to that component view is found, the component that it belongs to and its view name are listed on the results table. Selecting an item from this table opens the view in the XHQ Workbench.

To conduct a view search

1. From the **XHQ Workbench**, go to the Search View tab.



2. OPTIONAL
   Select (check) **Search for Use as a Template** to search for places where the particular component view is also used as a template.

For more information on templates, refer to the topic, *Using Templates*.

3. Click the **Start** button.
4. To open a resulting view in the XHQ Workbench, do one of the following:
   - Double-click on the view.

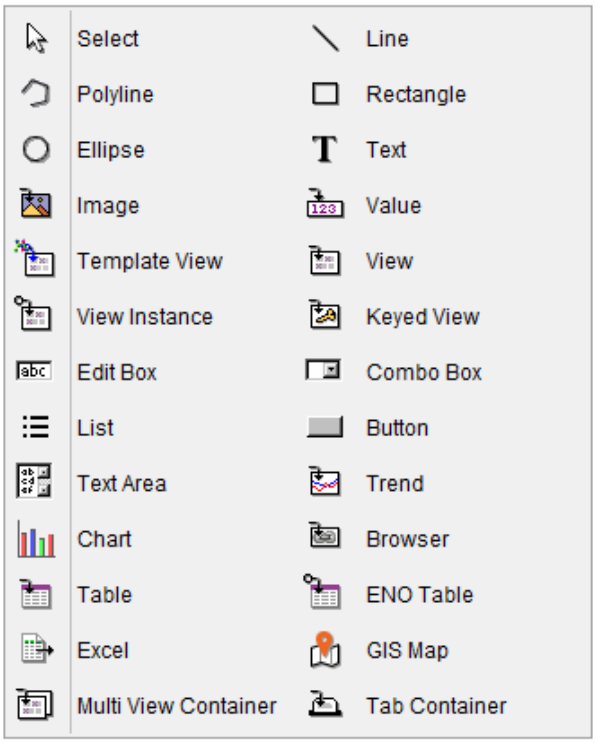
   or

   - Select the view and click **Edit**.

6 | Working With View Building Tools

XHQ provides a comprehensive drawing environment with a rich set of tools that help you produce high-quality views and animations.

The following table summarizes the tools available from the XHQ Workbench.



Tools Palette

Tool Description

Icon	Tool	Description
	Select	Selects, moves, and resizes view elements.
	Line	Draws straight lines in any directions.
	Polyline	Draws connected line segments and closed shapes.
	Rectangle	Draws squares and rectangles.
	Ellipse	Draws circles and ellipses.
	Text	Enters, selects, and edits text.

Icon	Tool	Description
	Image	Imports images.
	Value	Inserts live updating values from a data source.
	Template	Inserts a template (a reusable view based on a component model).
	View	Inserts an existing view into the current view.
	View Instance	Adds a view instance to a view. In this case, the embedded view configuration is based on the XHQ solution instance.
	Keyed View	Inserts a keyed view, which is an embedded tag view.
	Edit Box	Inserts an Edit box.
	Combo Box	Inserts a Combo box.
	List	Inserts a List control (checkbox or radio button).
	Button	Inserts a Button tool.
	Text Area	Inserts a Text Area tool.
	Trend	Inserts either a time plot or an XY-plot into the current view.
	Chart	Inserts an XHQ chart.
	Browser	Inserts a web control.
	Table	Inserts a table control.
	ENO Table	Adds an ENO (Extended Namespace Object) table to a view. Essentially, this allows the XHQ table control to use an ENO group as its data source.
	Excel	Inserts an Excel control.
	GIS Map Control	Inserts a GIS Map control.
	Multi View Container	Inserts a Multi View container.
	Tab Container	Inserts a Tab container.

Drawing Lines

The line tool draws lines and line segments. By default, the line tool draws thin, black lines. To draw straight lines and segments, you select the line tool, click anywhere, and drag. When you release the mouse button, the end point handle of the line appears.

To draw a straight line

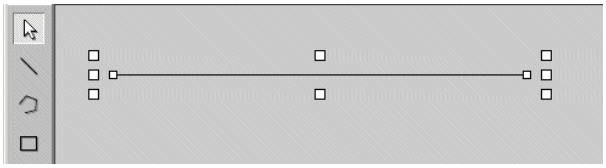
1. Select the **Line Tool** .

   The cursor changes to the insertion cursor.

2. Click where you want the line to **begin** and then drag.

   To draw the line at fixed angles in increments of 15°, press and hold the <SHIFT> key. The line will "snap" to the closest 15° angle multiple as you drag the cursor.

3. Release the mouse button where you want the line to **end**.
   When you release the mouse button, the line's boundary handles appear and the Line tab appears in the Edit panel.



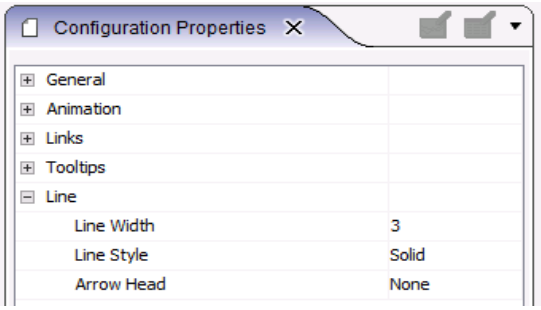
4. OPTIONAL
   To **thicken the width of the line** you have drawn, click the **Line** tab.
   The Line Width field appears with the default line width 1 (pixel) displayed.
5. Highlight the default line width and **enter the new line width** in pixels that you want to assign to the line.
6. OPTIONAL
   To **change the line style**, click the **Line** tab.
7. From the **Line Style** list box, select **Solid**, **Dotted**, or **Dashed**.
8. Press **Enter** to accept your change(s).

To shorten or extend an existing line

1. Select an existing line with the **Select tool**.
2. Place the **cursor** at either **endpoint**.
3. **Click** and **hold**.
   The line changes color, indicating that it is selected.
4. Press and hold the **<SHIFT> key**.
   The line will now "snap" to the angle of the existing line.
5. **Shorten** or **extend** the line by sliding the cursor in or out, along the existing line.

To add arrowheads to a line

1. Create a line and click the **Line** property.
   The Arrowhead field appears.



2. Select the arrowhead **location** (Origin, End, or Both).
3. Select the arrowhead **style** (Open or Solid).

To change the line color

1. Select an existing line with the **Select tool**.
2. Click the **Set line color** button to select a new color.

Creating Irregular and Open Shapes with the Polyline Tool

The polyline tool draws irregular open and closed shapes from straight-line segments. By default, the polyline tool draws thin black lines. To draw with the polyline tool, click anywhere on the drawing area and drag to draw the first line segment. The first line segment appears with point handles at the start and end. You draw a new line segment by clicking and dragging to a new location. Each time, the new line segment connects to the last point handle and terminates with a new point handle. If you click in close proximity to the start point, a join occurs, and a white filled shape is formed. You change the color of the shape using the color fill tool. In this way, you can create precise polylines and irregular shapes out of straight-line segments. You can also modify a polyline that you have drawn by dragging to adjust the position of any of its point handles.

To draw irregular open or closed shapes

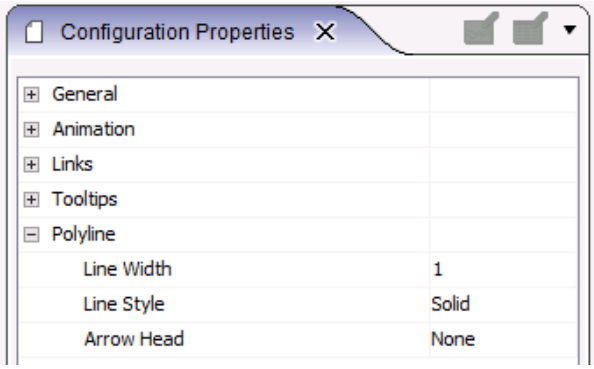
1. Select the **Polyline tool** .

   The cursor changes to the insertion cursor.
2. Click where you want the first line segment to **begin** and drag.
3. Click and drag to where you want the second segment to **end**.
4. **Continue adding segments** in the same manner, as desired.
5. To **stop adding segments** and complete the shape, do one of the following:

   - If the last segment ends at the beginning point handle of the first segment, click on that point handle to complete the object, creating a polygon.

   or

   - Click another tool in the toolbox to create an open shape.
6. OPTIONAL
   To **thicken the width of the polyline shape** you have drawn, click on the **Polyline** property.
   The Polyline field appears with the default polyline width 1 (pixel) displayed.

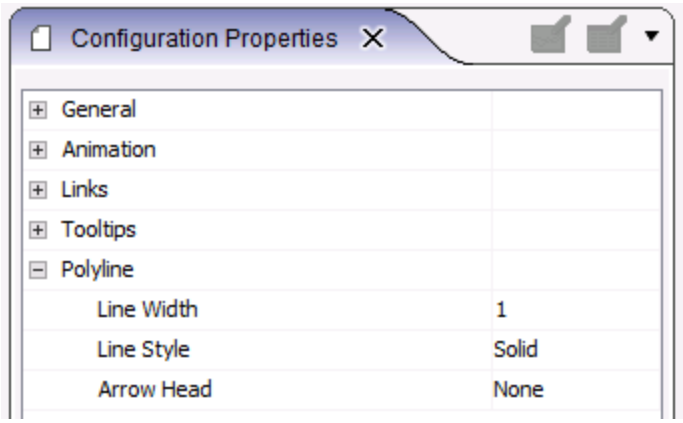


7. Highlight the default line width and **enter the new line width** you want to assign to the polyline shape.
8. OPTIONAL
   To **change the line style**, click the **Polyline** property.

   From the **Line Style** list box, select **Solid**, **Dotted**, or **Dashed**.
9. Press **Enter** to accept your change(s).

To add arrowheads to a polyline

1. Create a line and click the **Polyline** property.
   The Arrowhead field appears.



2. Select the arrowhead location (Origin, End, or Both).
3. Select the arrowhead style (Open or Solid).

To modify an existing irregular shape

1. Select a polyline with the **Select tool**.
2. Click on any point handle and drag to reposition it.

To change the fill or line color for an irregular shape

1. Select an existing polyline with the **Select tool**.
2. Do one or both of the following:
   - Click the **Set fill color** button.

   and/or

   - Click the **Set line color** button to select a new color.

Using the Rectangle and the Ellipse Tool

In addition to lines and irregular shapes, the drawing environment provides tools to draw regular shapes such as rectangles, squares, and ellipses. Images that you draw with these tools appear with the default line color (black), and fill color (white). You operate each tool using the same mouse interactions. To draw a shape, select the appropriate tool, then click and drag diagonally in any direction to set the size of the shape that you want. The outline of the shape will display as you drag the mouse. When you release the mouse, the shape appears with the point handles showing. By dragging with the mouse on a point handle, you can make immediate adjustments to the size of a regular shape.

To draw a rectangle or square

1. Click the Rectangle tool .
2. Click in the drawing area at the position where you want to **begin** drawing.
3. **Drag diagonally** to draw the rectangle or square.
   An outline of the image appears as you drag.
4. **Release the mouse** button when the image reaches the desired shape and size.

To draw an ellipse or a circle

1. Click the **Ellipse tool** .
2. Click in the drawing area at the position where you want to **begin** drawing.
3. **Drag diagonally** to draw the ellipse or circle.
   An outline of the image appears as you drag.
4. **Release the mouse** button when the image reaches the desired shape and size.

To change the fill or line color for a rectangle or ellipse

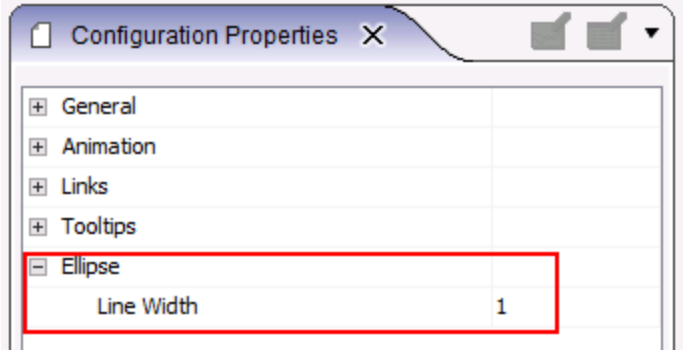
1. Select an existing shape with the **Select tool**.
2. Do one or both of the following:
   - Click the **Set fill color** button.

   and/or

   - Click the **Set line color** tool to select a new color.

To change the line width for a rectangle or ellipse

1. Select an existing shape with the **Select tool**.
2. Click the **Rectangle property** or **Ellipse property**.



3. Enter the new line width you want to assign to the shape.

Adding Text

The XHQ drawing environment offers text manipulation functions that enable you to place static text into drawings to create items such as notations, labels, and descriptions. You enter and change text by selecting the text tool, placing an insertion point, and typing.

To add text into a view

1. Click the **Text tool** .
2. Click on the drawing area at the desired location for the text.
   A text box appears, with the word Text as a temporary space holder.
3. Click the **Text** property.
4. In the text entry field, replace the word Text with the desired text.
5. Press **Enter**, or click another icon or tool.
   The text you entered appears in the view.

To change the font size, font style, and font color of text in a drawing

1. Click the **Select tool** .
2. Click anywhere in the text image.
   The selection handles appear around the text.
3. Click the **General** property.
4. Do one or more of the following:
   - To change font size, type the desired point size in the font size field and press Enter, or click another tool, icon, or field.

   or

   - To change font, select the desired font from the drop-down list.

   or

   - To add or remove bold or italic attributes, click the **B** or **I** icons.

The text style of the selected text field immediately changes.

To move text in a drawing

1. Click the **Select tool** .
2. Click anywhere in the text image.
   Selection handles appear around the text.
3. Drag the text image to the desired location.

To horizontally justify text

1. Add (insert) a Text or a Value view element.
3. From the main toolbar select from the following:

Icon	Button Name	Function
	Left Justifies	Horizontally aligns the text you select to the **left**.
	Center Justifies	Horizontally aligns the text you select to the **center**.
	Right Justifies	Horizontally aligns the text you select to the **right**.

Text justification buttons are only enabled when inserting Text- or Value-type view elements.

Inserting Images

Although the XHQ Workbench provides a very efficient vector graphics environment, you may still insert external bitmap images in `.gif`, `.jpg`, `.png`, or `.svg` format.

Animated `GIF`s are currently not supported.

In general, `GIF` images are much smaller in size than equivalent `JPEG` images. `GIF` files should be used for logos, line drawings, and icons. Typically, `JPEG`s are most often used for photographic images. The `PNG` format is an alternative to the `GIF` format. With `PNG`, however, you can avoid the patent license restrictions of the `GIF` format and have support up to 24-bit in color (as opposed to the 256 colors of the `GIF` format).

Bitmap images are pixel-based graphics that are inefficient in terms of storage and download size in comparison to vector-based graphics. Bitmaps may also seriously affect the display load time of your finished application. Therefore, minimize your use of bitmap images, and, wherever possible, use bitmap compression and optimization tools to reduce the file size.

Bitmap images may be resized. Other manipulations, particularly color transformations, cannot be carried out with bitmap images. However, the resolution of the bitmap image may be compromised if you stretch and make the bitmap image too large. Use the standard size, move, and delete tools to manipulate bitmap images.

After you select a bitmap image to insert, the system defaults to the same location where you picked your first bitmap image. The system maintains this default location for the duration of your working "session." However, once you exit XHQ, you need to re-establish your default location.

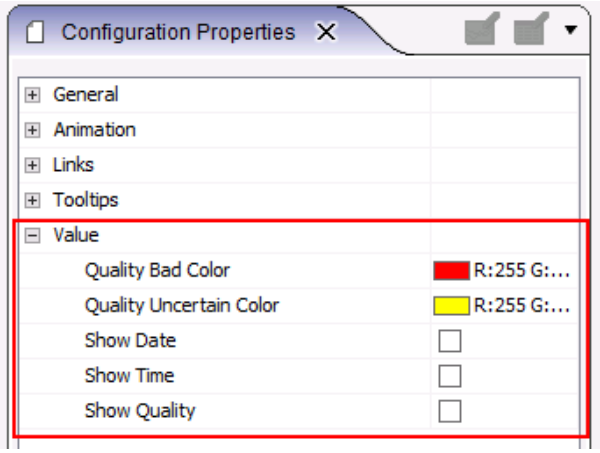
To insert an image

1. Click the **Insert Image tool** .

   The cursor changes to the insertion cursor.
2. **Place the cursor** at the approximate upper left-hand position where you want to insert the bitmap, then click.
   The Select image window appears.
3. **Locate** and **select the bitmap image** you want to insert.
   The system displays in the Select image window a small image of each bitmap in the directory.
4. Click **OK**.
   The bitmap image appears at the insertion point, with selection handles. You can adjust the location and size of the bitmap using the Select tools.

Using the Value Tool

With the **Value tool** , you can easily add live fields that update automatically as data values change. Values that update live can be Integer, Real, Boolean, or String. The value item placeholder format varies depending on the type of primitive.



Value Properties

Value Tool Placeholder Format

Primitive Member Type	Default Placeholder Format
Integer	1234
Real	12.34
Boolean	True
String	ABCD
Unknown	####

The values you display in Value Item are derived from specific primitive members that you configured from the XHQ Workbench.

If you insert a Real value item, and you make changes to the type (general, scientific, fixed) and the number of digits, then the next time you insert another Real value item, the system uses a placeholder that has the same type and number of digits as the Real value item you previously set.

From the **Value tab**, you can set the following properties.

Value Property Settings

Value Property	Settings
Quality color	[Choose Bad quality color.] [Choose Uncertain quality color.]

Value Property	Settings
Numeric format	The top drop-down selection box lets you choose to display the value item as a **General**, **Number-Scientific**, or **Number-Fixed**. XHQ only lets you choose formats that have logical meaning to the value item. Therefore, you cannot display a string item as a scientific number. In that case, XHQ will default to the general mode and display the string as a general.
Digits	This is the number of digits to display to the right of the decimal point. Enter the number of digits to apply. If you enter "-1" in the Digits field, XHQ displays all digits in the value item. If you enter "0" in the Digits field, XHQ displays 0 (zero) places.
Group digits	Checking this box automatically arranges digits to the **left** of the decimal point in groups of threes, divided by a separator (like a comma). *Example:* 1,234,567 **Note:** This feature is only enabled for Integer or Real values and applies to single-point or collection values. In addition, the grouping separator is automatically localized (for example, comma in the US, decimal in Germany).
Time format	Allows you to select hour/minute (if you have selected to associate time with the value item). **See Also:** For more information, refer to the topic, *Setting Date Time Quality and Unit*.
Date format	Allows you to month/date/year formats (if you have selected to associate date with the value item). **See Also:** For more information, refer to the topic, *Setting Date Time Quality and Unit*.
Units of Measure	Allows you to set the units of measure (UoM) for a value item, as well as a unit conversion. **See Also:** For more information, refer to the topic, *About Units of Measure*.

To add a Value item into a view

1. Click the **Value tool** .

   The cursor changes to the insertion cursor, , once you move the mouse into the View Panel.

2. **Click on the drawing area** at the desired location for the value item.
   The component's primitive members appear in a list at the insertion point.
3. **Select the primitive member** whose value you want to appear on the screen. A value item placeholder appears.

4. Click on the **Value** property.
   By default the Date, Time, and Quality selection boxes, which appear under the Show/Hide label, are not selected.
5. If you want to add date, time, or quality values alongside the value item, click in the appropriate Date, Time, and Quality boxes.
6. Set the general appearance of your value item display by clicking on the down arrow on any of the drop-down selection boxes or clicking on the attribute icons.
7. Set the color of the "good" quality by clicking on the Text color icon. Set the color for "uncertain" and "bad" quality by clicking on the Uncertain and Bad color icons.
8. Go to the main toolbar to set the desired attributes for your value item display such as font, font size, font color, bold and italic.
9. **Save** your changes.

To create a background around the value item

1. After adding the value item, click the **Rectangle tool**.
   The cursor changes to the insertion cursor.
2. **Draw** a rectangle around the value item.
3. Click the **Enable fill** button.
4. Click the **Set fill color** button.
   The Choose Fill Color dialog box appears.
5. Click on a **background color** and click **OK**.
6. Click the **Set line color** button.
   The Choose Line Color dialog box appears.
7. Click on a **border color** and click **OK**.
8. Click the **Send to back** button.
   The rectangle moves behind the value item.

To change the primitive member for a value item

1. From the drawing panel, click on the **value item**.
2. Click the Animation property.
3. Do one of the following:
   - From the evaluating drop-down list, select the primitive member whose value you wish to display. The members listed are those contained in the component for which you are building the current view.

or

   - If you want to display the value of a primitive member that belongs to a component further down in the solution hierarchy, such as a grandchild, you can type the name of the component in the evaluating field using the form:

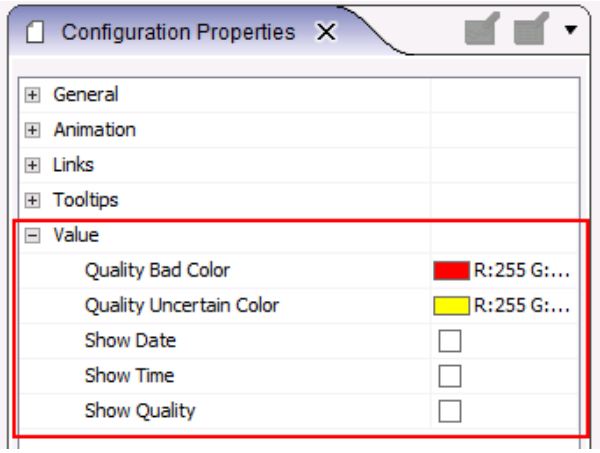
     `component2.component3.member`

     Where `component2` is a child of the current component (which would be `component1`), `component3` is a grandchild of `component1`, and member is a primitive member of `component 3`.

     You can include any number of nested component members, but the last member in the string must always be a primitive member.

Setting Date, Time, Quality, and Unit

XHQ lets you enable date, time, and quality indicators to the value you create. These optional indicators display next to the Value Item.



Value Tab: Date, Time Stamp, and Quality

You can position the date and time indicators to the left, right, above, or beneath the value item by selecting the value item placeholder and clicking on the left, right, upper, or lower calendar/clock icons that appear in the drawing panel.

You can position the quality indicator (black square) to the left or right of the value item by selecting the value item placeholder and clicking on the left or right quality icons (the Q icons) that appear to the left or right of the # sign.

About Dates

If you select just the **Date** option, you can position the date to the left, right, above, or beneath the value item by clicking on the left, right, upper, or lower calendar/clock icons that appear in the edit panel. In your view, the current date then appears next to the value item placeholder, according to your placement.

Next, select a date format from the drop-down selection box. The following table lists the date formats available.

Date Formats

Date Format	Example
MM/DD	03/12
MM/DD/YY	03/12/02
MM/DD/YYYY	03/12/2002
MM/DD (short text)	Tue, Mar/12
MM/DD/YYYY (short text)	Tue, Mar/12/2002
DD/MM/YY	12/03/02
MMM DD, YY	Mar 12, 02
MMM DD, YYYY	Mar 12, 2002

Date Format	Example
DD-MMM-YY	12-Mar-02
DD-MMM-YYYY	12-Mar-2002

About Time

If you select the **Time** option, the current time appears next to the value item placeholder (to the left of the date if you have selected the date option) and the time format drop-down selection box appears.

Time Format	Example
HH:MM	03:24
HH:MM:SS	03:24:47
HH:MM (24)	15:24
HH:MM:SS (24)	15:24:47

About Quality

If you select the **Quality** option, the quality placeholder (a square) appears to the right of the value item placeholder and the right-most Quality icon (Q) is selected.

Bad Quality

BAD quality has the following indicators:

Quality	Symbol	Example
BAD	red asterisks	*****
BAD, Access denied	red question marks	?????
BAD, Calculation error	red asterisks	*****
BAD, Comm failure	red asterisks	*****
BAD, Config Error	red asterisks	*****
BAD, Connector not running	red asterisks	*****
BAD, Device failure	red asterisks	*****
BAD, Last known value	value	**1234.56**
BAD, Non-existent	red pound signs	#####
BAD, Not configured	red pound signs	#####
BAD, Not connected	red asterisks	*****
BAD, Not initialized	black pound signs	#####
BAD, Not supported	red asterisks	*****
BAD, Out of service	red asterisks	*****
BAD, Sensor failure	red asterisks	*****
BAD, Units conversion failure	red asterisks	*****

When using expressions, please note that there is a distinct order when displaying BAD quality: "BAD, Access denied" supersedes "Bad, Non-existent"; "BAD, Non-existent" supersedes "BAD, Not connected".

Consider the following example.

You have an expression containing four integers added together, "a + b + c", where the integers have the following quality:

a = "BAD, Access denied"

b = "BAD, Non-existent"

c = "BAD, Not connected"

Because the quality "BAD, Access denied" supersedes all the others, the overall quality of the expression will then be "BAD, Access denied".

Bad Quality Value Markers

A BAD quality value **as a marker** is recorded for points at the start of XHQ (the point's activation time) as well as the shutdown of XHQ (the point's deactivation time). A BAD marker is also recorded when history is changed to "None" and back to "DR", and when the "Record data" checkbox is cleared (unchecked) and when it is set (checked).

In summary, BAD quality values as **markers** are recorded during the following conditions:

- XHQ is started up.
- XHQ is shutdown.
- History is configured for the Data Recorder.
- History is changed from the Data Recorder (DR) to None or to another historian.
- Recording is stopped (Record data is unchecked).
- Recording is resumed (Record data is checked).

About Units of Measure

You can set the units of measure (UoM) for a value item, as well as a unit conversion (by defining a "From Unit" – "Convert to Unit**"** combination). The Unit option enables you to display the unit next to the value item.

Things to Note

- If the value expression has only a single member or tag reference, the "From Unit" selection is disabled and is set to *Default* (as seen in the image above).
- If the value expression has multiple references, then the "From Unit" and the "Convert to Unit" selections are set to *None* by default. For unit conversion, you must explicitly specify the "From Unit" and the "Convert to Unit" selections.

To set-up unit conversion for a value expression with multiple references

1. Place a **value** onto the drawing panel and select a member.
   From the Value tab, note the initial default values for the Units of Measurement options. The "From Unit" option is disabled.
2. Click the **Animation property**.
3. Enter a value expression that contains multiple references.

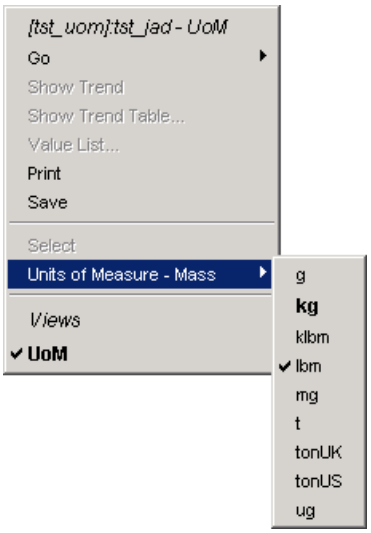
   Example: `xhqreal01+xhqreal02`

4. Click the **Value property**.
   The "From Unit" option is now enabled.
5. Select a **Unit Category**, the **From Unit**, and the **Convert to Unit**.
6. OPTIONAL
   Select **Unit** to show the unit with the value.

   Example: 27.21 (lbm)

7. **Save** your changes.

At **runtime**, right-click on this value item and the following shortcut and sub-menus appear:

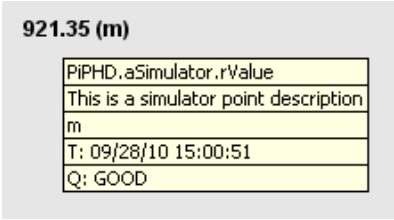


Runtime Right-click Shortcut Menu with Units of Measurement

For detailed information on the runtime behavior of UoM, see the topic, *Units of Measure at Runtime*.

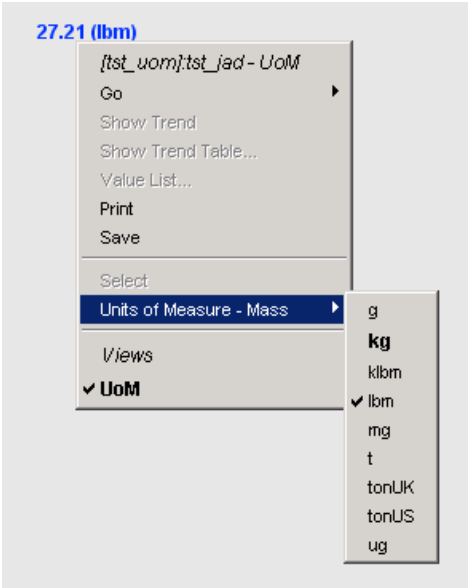
Units of Measure at Runtime

At runtime, the unit is displayed with the value (if enabled), as well as in the value item's Tooltip.



Tooltip with Unit of Measure

When you right-click on a **Value Item** with the Unit option enabled, the " Units of Measure" command appears on the shortcut menu. The submenu lists units that belong to the given category.



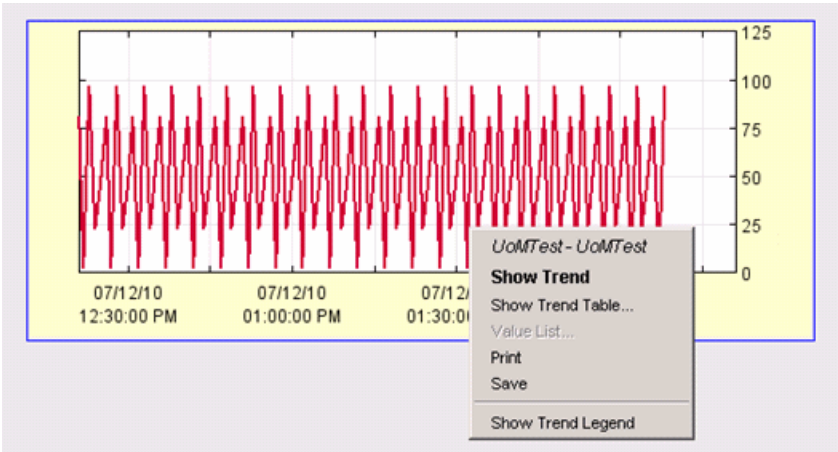
Category: Mass

View Default Unit (in bold): kg

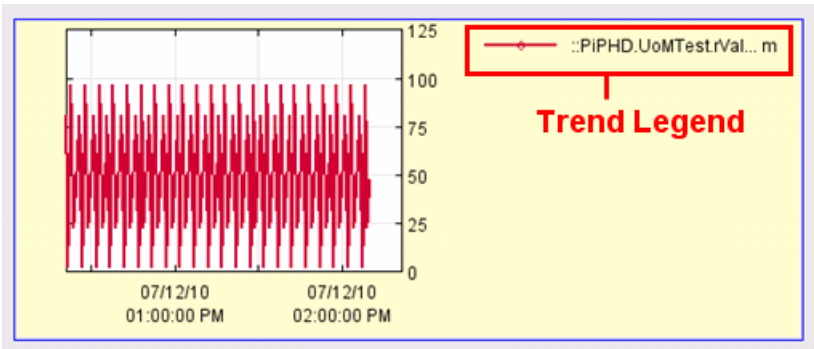
Selected Unit (which check mark): lbm

The units displayed in the submenu are sorted in alphabetic order. The maximum number of units displayed is 20 (which is the default). Use the global property **`MaxUnitOfMeasure`** to set the maximum number. If there are more than 20 units in the category, a `<More...>` command appears at the end of the list.

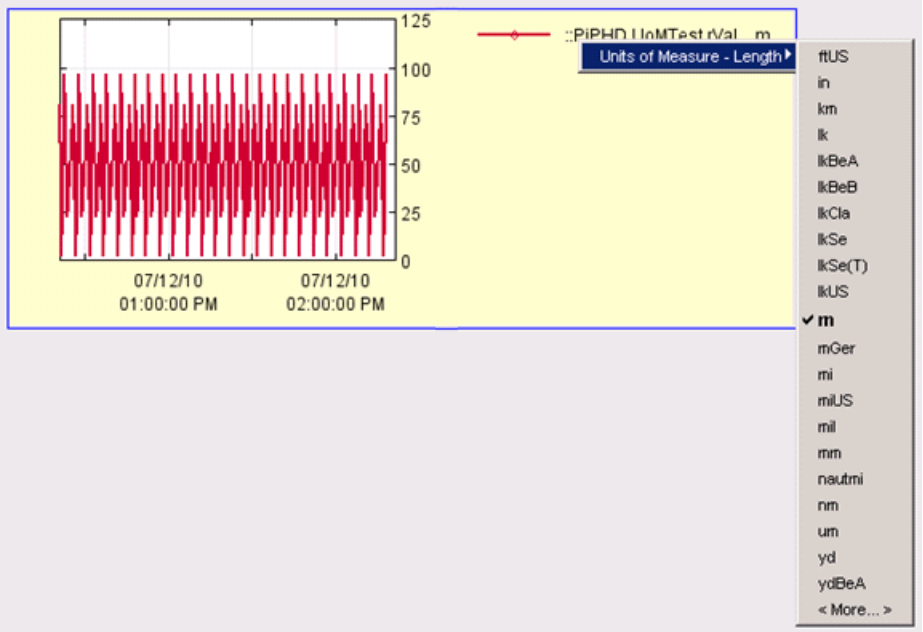
To show the units of measure in an **Embedded Trend**, from the right-click shortcut menu, click the **Show Trend Legend** command.



The Trend Legend displays the trend color (which, in the example below, is red), name (`::PiPHD.UoMTest.rVal...`), and assigned unit (m).

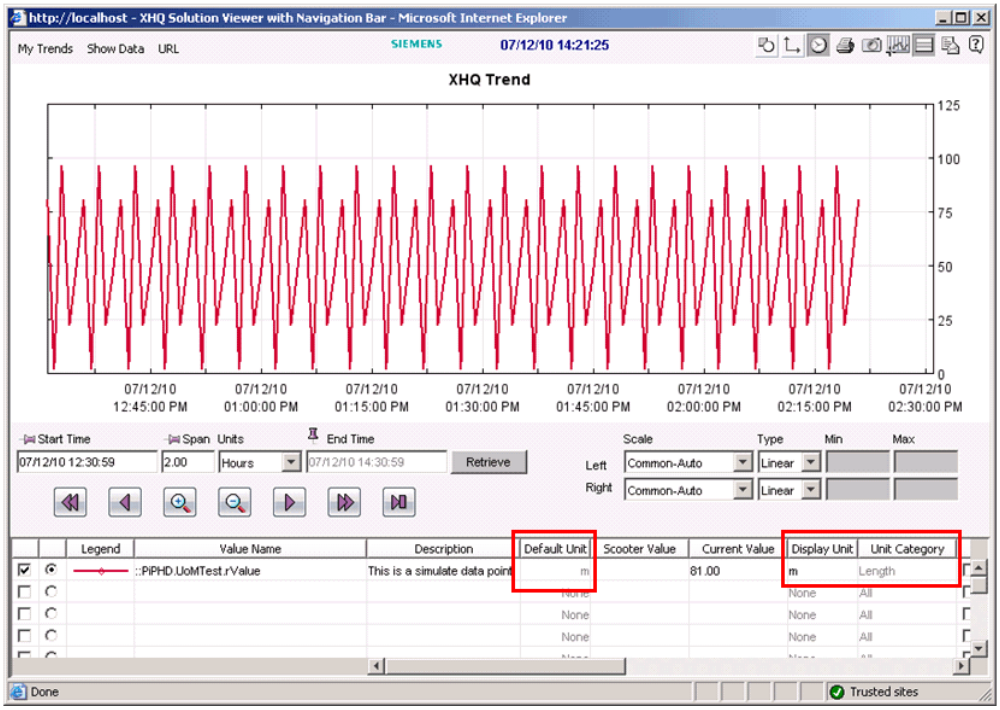


Right-click on the legend name to view the Units of Measure shortcut menu, which lists the compatible units for the given category (in this case, Length).

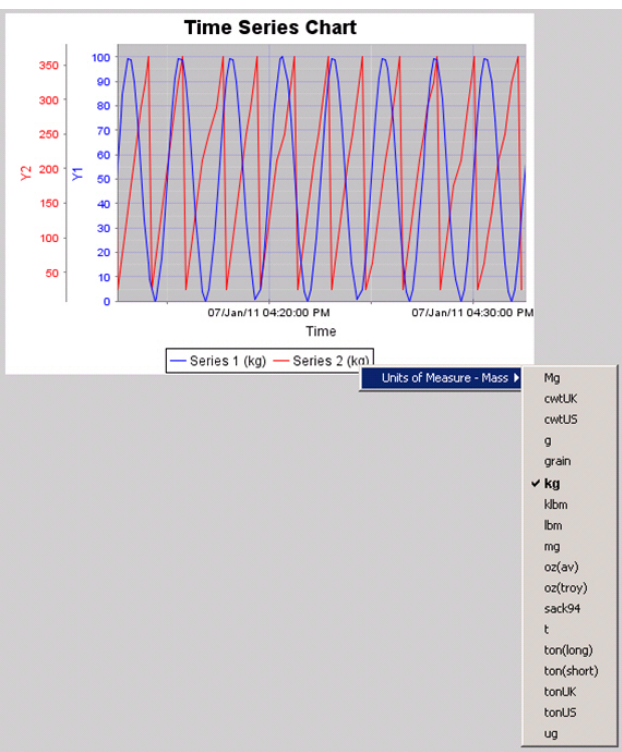


The unit conversion for an Embedded Trend works the same way as the unit conversion of a value item.

For an **Interactive Trender**, the "Default Unit" column shows the "From Unit" option, the "Display Unit" column shows the current unit (which is the "Convert To Unit" option), and the "Unit Category" column shows the "Unit Category" option.

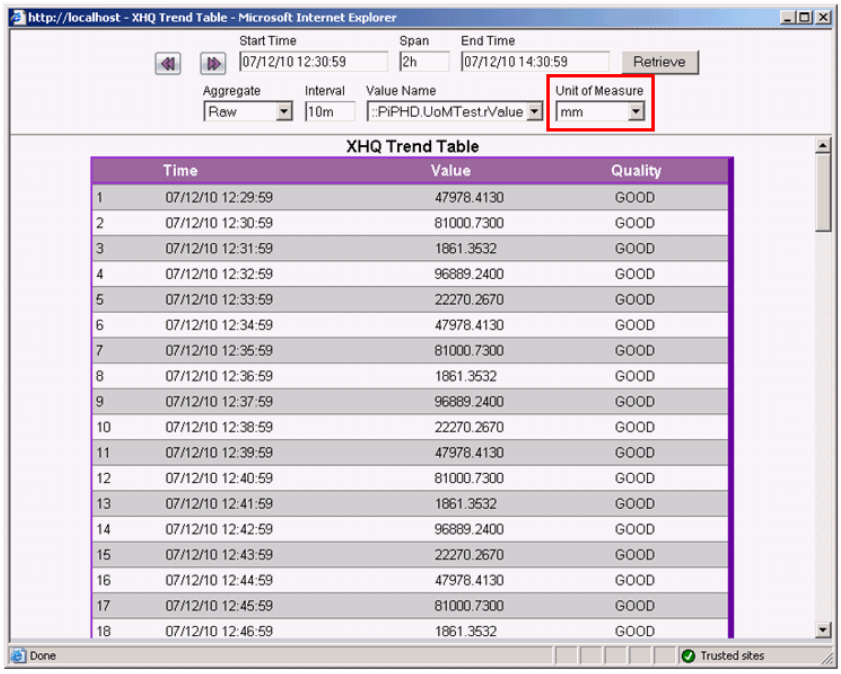


Units of Measure in the Interactive Trender



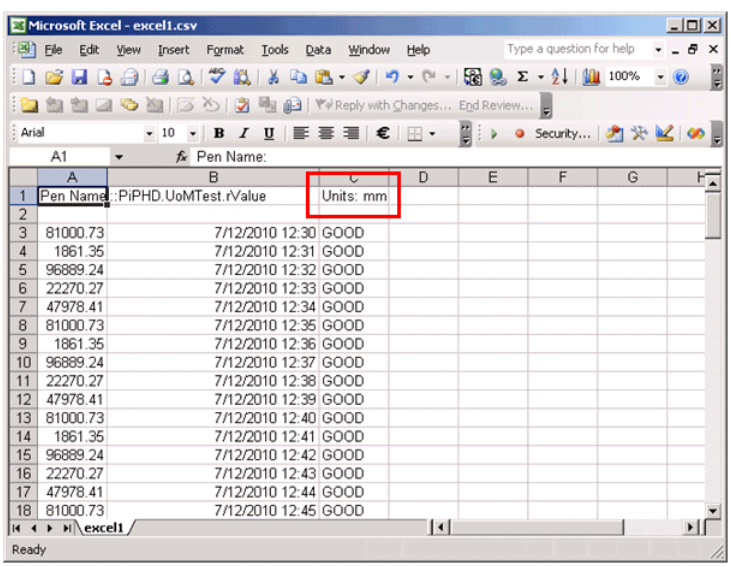
If the Units of Measure is configured for a **Chart** (Interactive or embedded in a view), the option appears by going to the chart **Legend**, mousing over the series name, and right-clicking. The Units of Measure option appears with a submenu listing the units that belong to the given category.

When showing data in the **Trend Table**, the "Unit of Measure" drop-down option shows the current unit of measure. The unit (in this case, mm) is carried over from the "Display Unit" column of the Interactive Trender.



Units of Measure in the Trend Table

And like the Trend Table, when showing data in an **Excel spreadsheet**, the unit (mm) is also carried over from the "Display Unit" column of the Interactive Trender.



Units of Measure in Excel

Inserting Views

While you are editing a view, you can reuse other views of components in the XHQ Workbench. You can insert an existing view into another view, complete with animations and links to a data source. Views attach directly to components and component instances. Consequently, the insert view operation limits your choice of views to those attached to the member components contained in the within the current component. Following an insert view, the animation works immediately without additional configuration.

As a solution best practice, resizing: 1) grouped view elements, 2) templates, or 3) embedded views (which include key views) is not recommended. Resizing affects every element in the view proportionally, which may result in undesired line widths or font sizes. Consider using a different view or an animation to get the desired behavior.

Refer to the topic, *Building Views*, for step-by-step instructions on how to copy an embedded element from one view to another view in another part of the solution.

To insert a view into a view

1. Click the **Insert View** tool .
2. Place the cursor at the approximate upper left-hand position where you want to insert the view, and then click. A small square appears.
3. Select the **View** property.
4. Select a component from the Component drop-down list.
5. Select a view from the View drop-down list.
   The view appears. You can move the inserted view within the drawing panel using the Select tool, but you cannot resize it.

You can repeat the above procedure to add multiple views.

Pasting a view within the same view

XHQ makes it easy to copy or cut a view element, or an entire view, and paste it within the same view or within a different view anywhere else in the solution model. In this section, instructions on how to cut and copy within views is provided.

You can click on icons to perform cut-and-paste operations or you can use keystroke equivalents to perform the same operations.

Cut, Copy and Paste Icons

Icon	Function	Keystroke Equivalent
	Cut Selection	CTRL+X
	Copy Selection	CTRL+C
	Paste Selection	CTRL+V

To copy a view and paste within the same view

1. With the view displayed, use the **Select tool** to select the view or view element you want to copy.
2. If you are just copying a part of the view, click once and drag a cursor rectangle around the part of the view you want to copy.
3. Release the mouse button when you have selected the portion of the view you want to copy.
4. Click the **Copy** icon.
5. Click the **Paste** icon.
   The copied view or view element is pasted in the view window directly on top of the image you copied. You can paste the view or view image as many times as you want. You can nudge each cut view or view element vertically or horizontally and keep the image "in line" horizontally or vertically with the original view element you copied.
6. **Close** the window to **save** the changes to your view.
   The Save dialog box appears.
7. Click **Yes**.

To cut a view and paste within the same view

1. With a view displayed, use the **Select tool** to select the view or view element you want to cut and paste.
2. Click on the **Cut** icon.
   The view or view element you have selected is "cut" or deleted from the view.
3. Click on the **Paste** icon.
   The cut view or view element is pasted in the view window. You can paste the view or view image as many times as you want. You can nudge each cut view or view element vertically or horizontally and keep the image "in line" horizontally or vertically with the pasted view element.
4. **Close** the window to **save** the changes to your view.
   The Save changes to dialog appears.
5. Click **Yes**.

To duplicate a view and paste within the same view

1. With a view displayed, use the **Select tool** to select the view or view element you want to duplicate.
2. Hold down the **CTRL** key and press **D**.
   The view or view element you have duplicated is displayed at the default duplication offset point.
3. Hold down the **CTRL** key and press **D** for each duplication you want to make.
   XHQ maintains the default duplication offset point for each duplication.
4. You can **change the offset** for the duplication by selecting the duplicated object and moving it to another position anywhere in the View window or by nudging it to another position in the view window.
5. After you have established the offset reference point, hold down the **CTRL** key and press **D** for each duplication you want to make.
   The system will duplicate at the offset reference point you have established.
6. **Close** the window to **save** the changes to your view, and click **Yes**.

Using the XHQ View Instance

Unlike the embedded view element configuration (which is based on component members of a modeled class), the **XHQ view instance** allows the XHQ view developer to embed a view using the instance context from the XHQ solution.

⊟ View Instance	
Path	::Sample.PI_Group.KK1
Component	XHQ_Top
Device	Tablet
Orientation	Landscape
View	KPI
Navigation	Use view properties

View Instance Configuration Properties - For an ENO instance

⊟ View Instance	
Path	::Sample.PI_Group
Component	XHQ_Top
Device	Tablet
Orientation	Landscape
View	KPI
Navigation	Use view properties
Row Count	1
Column Count	1
Use Navigation Button	☑
No Layout	☐

View Instance Configuration Properties - For a typed ENO Group instance

Property Descriptions

Property Name	Description
Path	The path to the XHQ solution node. The node can be: • A **modeled** instance. • An **ENO** instance. • A **typed ENO Group** instance.
Component	*This property is READ ONLY.* The class of the instance. This property value is determined by the specified instance.
Device	The device of the embedded view.

Property Name	Description
Orientation	The orientation of the embedded view.
View	The name of the embedded view.
Navigation	The view navigation behavior.
Row Count	The number of rows used to display items in a **typed ENO Group**. So, this property is only valid when the node is a typed ENO Group instance.
Column Count	The number of columns used to display items in a **typed ENO Group**. So, this property is only valid when the node is a typed ENO Group instance.
Use Navigation Button	This property is only valid when the node is a **typed ENO Group** instance.
No Layout	This property is only valid when the node is a **typed ENO Group** instance.

See also *Using an ENO Group as the Data Source*.

Global Properties

For more information, see the topic, *View and ENO Group Instances*, located in the XHQ Administrator's Guide.

View Variable Method

View instances support the view variable method, **`count()`**.

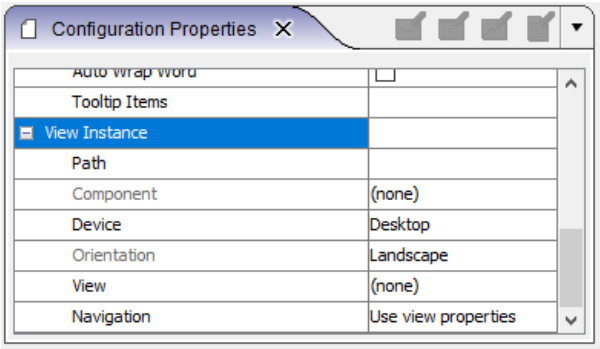
Example: `$ViewInstance1.count()`

Return: The total number of items in the ENO group.

To configure a view instance

1. From the **Tool palette**, click the **View Instance** tool and place the tool on the Drawing panel.

2. In the **Configuration properties** tab, expand **View Instance**.



3. Enter the **Path** to the XHQ solution node.

The View Instance properties that appear depends on the node instance. The node can be:

- A **modeled** instance.
- An **ENO** instance.
- A **typed ENO Group** instance.

4. Configure the given properties.

Property Name	Description
Component	*This property is READ ONLY.* The class of the instance. This property value is determined by the specified instance.
Device	The device of the embedded view.
Orientation	The orientation of the embedded view.
View	The name of the embedded view.
Navigation	The view navigation behavior.
Row Count	The number of rows used to display items in a **typed ENO Group**. So, this property is only valid when the node is a typed ENO Group instance.
Column Count	The number of columns used to display items in a **typed ENO Group**. So, this property is only valid when the node is a typed ENO Group instance.
Use Navigation Button	This property is only valid when the node is a **typed ENO Group** instance.
No Layout	This property is only valid when the node is a **typed ENO Group** instance.

5. **Save** the view instance property configurations.

Inserting a Keyed View

Because tags are not modeled (they are not members of other components in the solution), tag views cannot be embedded in the usual way. To embed a tag view, you will need to use the **Keyed View tool**.

For more information on the Keyed View tool, go to the topic, *About Keyed Views*

To insert a keyed view

1. From the **Tools palette**, click the **Insert Keyed View** tool and place the tool on the Drawing panel.

   A keyed view element placeholder appears.
2. Click the **Keyed View property**.
3. To select a **Tag Component**, click on the browse button next to the text box.
   The "Select Component" dialog box appears.
4. From the list, select a tag component, or a component that extends a tag component, and then click **OK**.
5. Under **View**, select the view to embed.
6. In the **Keyed Value** text box, enter an expression that results in the name of the desired tag.
7. Select a **Navigation** setting.

Select …	If you want to …
Use views properties	Use the settings defined in the Views Properties tab.
Always drillable	Set the view as always drillable, regardless of whether the view is, by default, drillable or not.
Never drillable	Set the view as never drillable, regardless of whether the view is, by default, drillable or not.

8. **Save** the view.

EXAMPLE: Working with Composed Tag Views

Some solutions have a component which is composed of more than one tag (in typical situations, it is composed of three tags). This component is called a **Controller**. All tags in the Controller have the same name, but are distinguished by a give suffix (for example, .OP, .SP, and .IP). The Controller consists of views which are compositions of views of its tags. Therefore, a Controller can be built without having to model it in the solution.

The Controller is a component with only member, Name, which references the base part of embedded tags. The Controller's view consists of "keyed views" of its tags, where the Name member is used in the expression to properly resolve the name of each embedded tag.

The Controller's view is then embedded in the appropriate view as a template. The expression for its Name member results in the base of the desired tag composition. This name is then passed to the expressions of keyed views embedded in the Controller's view with an appropriate suffix, so all necessary linking is complete.

Using Templates

Templates allow you to create "reusable" views. These "reusable" views can be in any view (regardless of the structure of the component in which the view is contained in) as well as used across multiple components.

Advantages to using templates:

- Changes made to a single instance of the view are automatically applied to all other instances of the same view.
- From a modeling standpoint, the Business Model is not "polluted" by components of a typical UI Model.

As a solution best practice, resizing templates is not recommended since it may result in an undesired sizing of the view. Resizing the template affects every element in the view (including text and lines) proportionally. If a template needs to have a dynamic resizing behavior, build it with animation to accommodate this behavior.

To map template component members

1. Add a template to your view using the Template tool.
2. In the **Configuration Properties tab**, expand **Template**.
3. Click the **Browse** button to select the **Template Component**.
   This displays the component views and members.
4. For **Template View**, click inside the cell and select a view.
5. Next, under **Member Mapping**, map the members of the component.

EXAMPLE: Using Templates

To further illustrate the usefulness of templates, consider the following example.

Step 1 – Create the components

Component Member	Primitive Member Name	Type
Meter	value_m	Real
	max_m	Real
	min_m	Real
Pump1	speed_p1	Int
	temp_p1	Real
	vibration_p1	Real
Pump2	speed_p2	Int
	temp_p2	Real
	vibration_p2	Real

Step 2 – Create the views for Meter

There are three views for Meter: Analog1, Analog2, and Digital.

The Analog1 view displays the attributes of the Meter component in the following manner.

View Elements for Analog1	Description
Value	Displays "value_m"
Value	Displays "min_m"
Value	Displays "max_m"
Lines	For tick marks
Lines	Animated to swing through the min-max range, based on value

The **Analog2 view** displays the attributes of the Meter component in the following manner.

View Elements for Analog2	Description
Value	Displays "value_m"
Value	Displays "min_m"
Value	Displays "max_m"
Lines	For tick marks
Lines	Animated to swing through the min-max range, based on value

The **Digital view** displays the attributes of the Meter component in the following manner.

View Elements for Digital	Description
Value	Displays "value_m"
Lines	Animated to show and hide as value increases and decreases.

Step 3 – Create the Summary_p1 view for Pump1 using the Template Tool

The **Summary_p1** view has embedded views for displaying **speed** and **temperature** for **Pump1**. Through the Template tool, you will be reusing the **Analog1** and **Analog2** views for the **Meter** component to display these fields.

1. **Create a new view** for Pump1 named **Summary_p1**.
   The View tab appears.
2. From the **Tools palette**, click the **Insert Template** tool and place the tool on the Drawing panel.
3. Click the **Template property**.
4. Select a **Template Component**.
   The "Select Component" dialog box appears.
5. Click **Meter** and click **OK**.
   The Template tab displays Meter views and members.
6. Under **View**, select **Analog1**.

Next, map the members of the Analog1 view with the following "expressions":

Summary_p1 View for Speed

Member	Expression
max_m	500
min_m	0
value_m	speed_p1

7. Under **Member**, select **max_m**.
8. In the **Expression** box, type **500** and press Enter.
9. Repeat for **min_m** equal to **0**.
10. Under **Member**, select **value_m**.
11. Under **Expression**, select **speed_p1**.
12. **Save** the view.
13. Next, you will use the Template tool again, this time using the Analog2 view for temperature.
14. From the **Tools palette**, click the **Insert Template** tool and place the tool on the Drawing panel.
15. Click the **Template property**.
16. Select a **Template Component**.
    The Select Component dialog box appears.
17. Click **Meter** and click **OK**.
    The Template tab displays Meter views and members.
18. Under **View**, select **Analog2**.

    Next, map the members of the Analog2 view with the following "expressions":

Summary_p1 View for Temperature

Member	Expression
max_m	500
min_m	10 + speed_p1
value_m	temp_p1

19. Under **Member**, select **max_m**.
20. In the **Expression** box, type **500** and press **Enter**.
21. Repeat for **min_m** equal to **10+speed_p1**.
22. Under **Member**, select **value_m**.
23. Under Expression, select temp_p1.
24. **Save** the view.

Step 4 – Create the Summary_p2 view for Pump2 using the Template Tool

The Summary_p2 view has an embedded view for displaying vibration for Pump2. Through the Template tool, you will be reusing the Digital view for the Meter component to display this field.

1. **Create a new view** for Pump2 named **Summary_p2**.
   The XHQ Workbench appears.
2. From the **Tools palette**, click the **Insert Template** tool and place the tool on the Drawing panel.
3. Click the **Template property**.
4. Select a **Template Component**.
   The Select Component dialog box appears.
5. Click **Meter** and click **OK**.
   The Template tab displays Meter views and members.
6. Under **View**, select **Digital**.

Next, map the members of the Digital view with the following "expressions":

Summary_2 View for Vibration

Member	Expression
max_m	100
min_m	-100
value_m	vibration_p2

7. Under **Member**, select **max_m**.
8. In the **Expression** box, type **100** and press **Enter**.
9. Repeat for **min_m** equal to **-100**.
10. Under **Member**, select **value_m**.
11. Under **Expression**, select **vibration_p2**.
12. **Save** the view.

EXAMPLE: Exporting and Importing Templates Using .view Files

In this example, you have two sets of repos:

Repos A	Repos B
Contains Comp1 Comp1 has a View1 View1 includes Template1	(Initially, this is an empty repos.)

Your goal is to export Comp1 from Repos A and import it into Repos B.

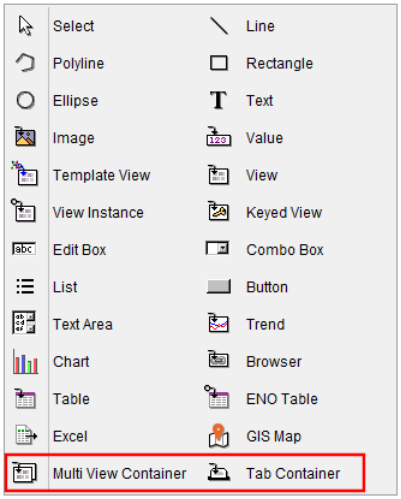
1. Export **the component only** from Repos A and import it to Repos B.
2. Then, from the XHQ Workbench of **Repos A**, right-click on **Comp1**.
3. From the shortcut menu, select **Export View** and **export View1** using the **.view format**.

4. From **Repos B**, open the **XHQ Workbench**.
5. Right-click on the imported **Comp1**, and from the shortcut menu, select **Import View**.
6. Navigate to the **.view** file that was exported in step 3.

Using Containers

There are two container tools:

- **Multi View**
  The Multi View Container can hold multiple views and, based on the configuration, can load/show one view at a time.
- **Tab**
  The Tab Container can hold multiple tabs, with each tab linked to a view defined in a Multi View Container.



Tools Palette

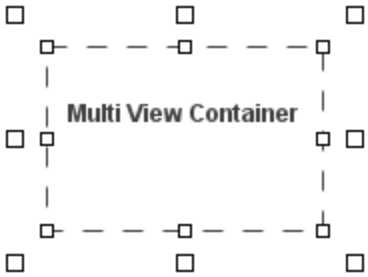
Use of containers replaces the view stacking technique (laying multiple views in the same viewing area).

Configuring a Multi View Container

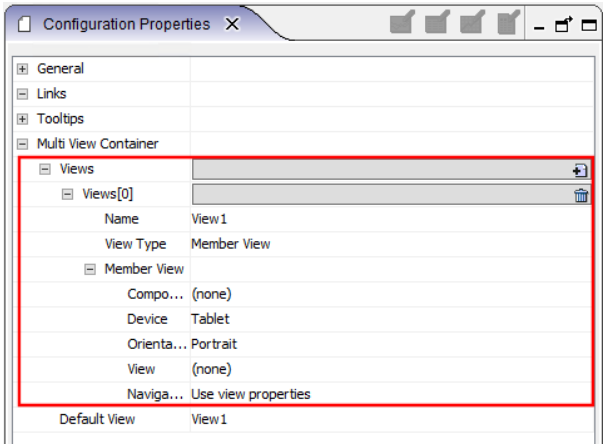
The Multi View Container supports all XHQ view types for the views it contains: Member Views (which includes Collection Member Views), Template Views, Keyed Views, and Instance Views.

Multi View Container Tool Icon

Place a Multi View Container on the view drawing panel.

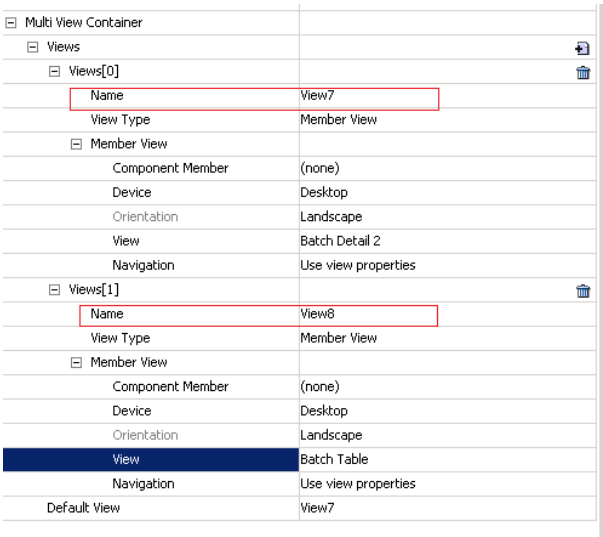


Point and selection handles appear around the view element. Select the container and go to the **Configuration Properties tab**. Click inside the Views property cell to add the first view. Additional properties for this first view appear.



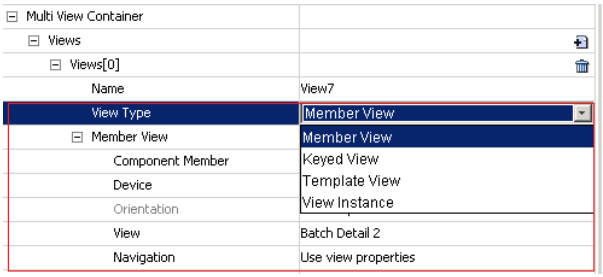
Multi View Container Properties

To **add more views**, simply continue to click inside the Views property cell.

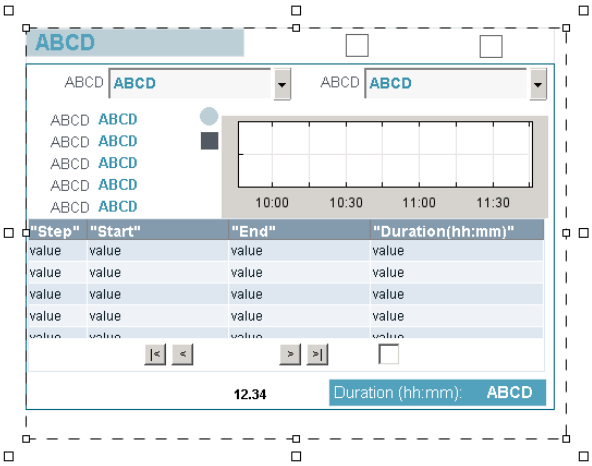


To **delete a view**, click on the Trash icon next to the name of the view you want to delete.

The contained view configuration is the same as configuring a view item outside of the view container for Member Views, Template Views, Keyed Views, and Instance Views.



This container can hold an **array of views**. The first view created automatically becomes the Default view, but you can designate any of the views in the container as the Default view that initially appears at edit time as well as runtime. The default view is displayed in the View Edit panel.



Drag-n-drop to add an existing view item to itself from the View Edit panel or the Outline window.

At edit time, the Multi View Container clips the view it contains if the bounding rectangle of the view is *larger* than the bounding rectangle of the Multi View Container itself.

Multi View Container Property Settings

Multi View Container Property	Settings
Views	Click inside this cell to add a view.
Views[0]	View array element ID.
Name	View variable name associated with the view. The Multi View Container supports the view variable method, `selectview("viewVarName")`. Upon being called, the view, `<viewVarName>`, is the view selected.
View Type	*Options:* • Member View • Keyed View • Template View • View Instance
*for **Member View** type*	*Options:* • Component Member/Collection • Device • Orientation • View • Navigation
*for **Keyed View** type*	*Options:* • Tag Component • Device • Orientation

Multi View Container Property	Settings
	• View • Navigation • Key Value
*for **Template View** type*	*Option:* • Template Component
*for **View Instance** type*	*Options:* • Path • Component • Device • Orientation • View • Navigation
Default View	The view that is displayed by default during view edit time as well as when the view is initially launched at runtime. This is the first view in the view array.

Selecting a Different View Context

The Multi View Container **view variable method**, **`selectView`**, is designed to be used in the same view context. On the other hand, the **system function**, **`selectView`**, is designed to be used in the different view context.

The syntax of the system function is as follows:

```
Boolean selectView(String multiViewContainerVarName, String viewVarName)
```

Where:

- **`multiViewContainerVarName`** is the Multi View Container view variable name.
- **`viewVarName`** is the view variable name in the Multi View Container.

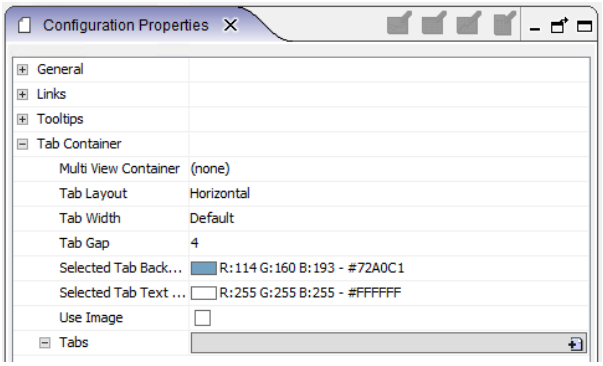
Returns true if successful; else, false.

Configuring a Tab Container

Tab Container Tool Icon

Place a Tab Container on the view drawing panel.

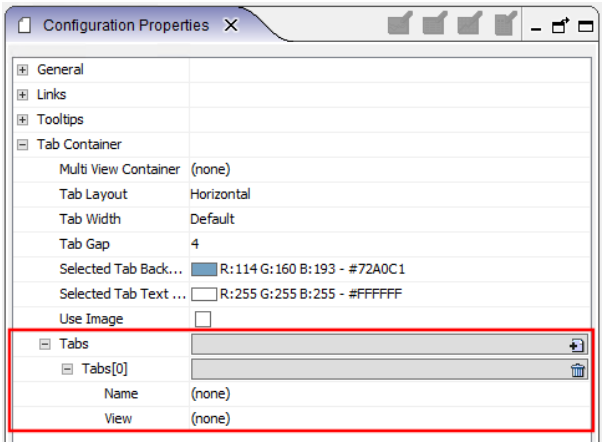
Point and selection handles appear around the view element. Select the container and go to the **Configuration Properties tab**.



Tab Container Properties

To **add more tabs**, simply continue to click inside the Tabs property cell. To **delete a tab**, click on the Trash icon next to the name of the tab you want to delete.

The Tab Container is associated with one Multi View Container. Each tab has a Name and View property.

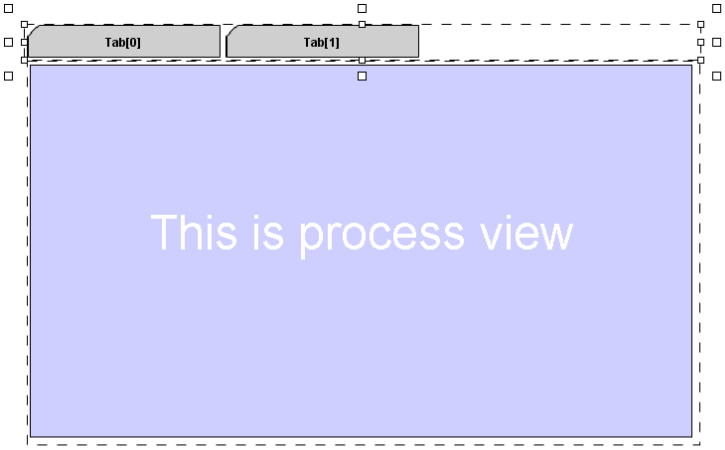


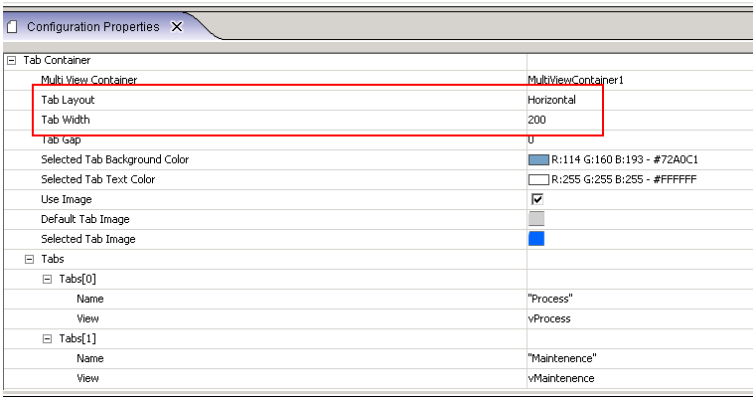
The **Name property** is a String XHQ expression. The value of this expression is used as the label for the tab. The **View property** is a view variable name defined in the associated Multi View Container.

This container can hold an **array of tabs**.

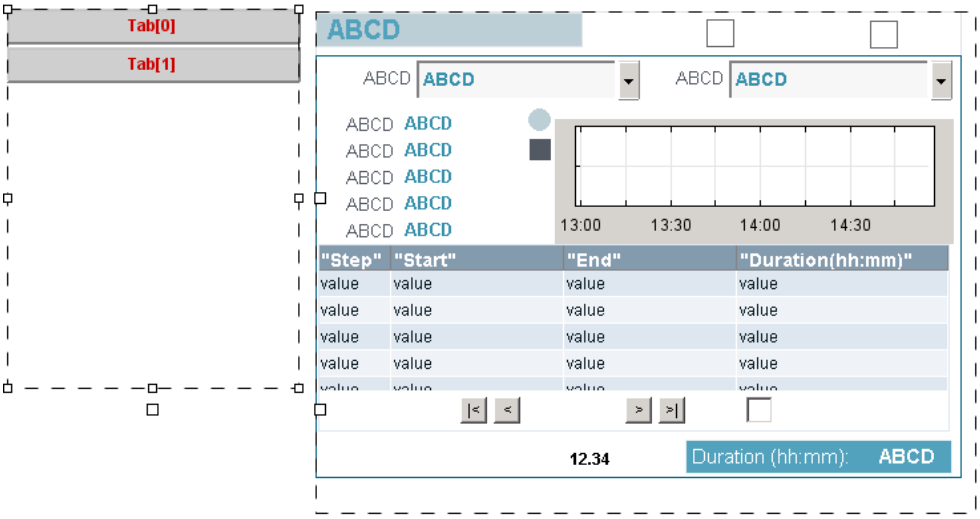
The tab layout may be configured horizontally or vertically.

Horizontal Tab Layout Example:





Vertical Tab Layout Example:



Tab Container	
Multi View Container	MultiViewContainer1
Tab Layout	Vertical
Tab Height	Default
Tab Gap	4
Selected Tab Background Color	R:114 G:160 B:193 - #72A0C1
Selected Tab Text Color	R:255 G:255 B:255 - #FFFFFF
Tabs	
Tabs[0]	
Name	"Batch Detail"
View	View12
Tabs[1]	
Name	"Batch Table"
View	View13

The default value of the tab layout is determined by the Tab Container bounding rectangle dimension: horizontal layout if the width is greater than the height; else, it is vertical.

At runtime, the default selected tab is determined by the default view defined for the associated Multi View Container. If there is no default view defined, then there is no tab is selected by default.

Tab Container Property Settings

List Property	Settings
Multi View Container	The name of the associated Multi View Container.
Tab Layout	Tabs can be horizontal or vertical.
Tab Width *or* Tab Height	For a **horizontal** Tab Container (Tab Width), the tab height is the same as the Tab Container height; the width is configurable. For a **vertical** Tab Container (Tab Height), the tab height is configurable and the width is the same as the Tab Container width.
Tab Gap	The space (in pixels) between tabs. The default value is 4 pixels.
Selected Tab Background Color	Click the color box to choose the background color of the selected tab. The default is 114,160,193 (RGB).
Selected Tab Text Color	Click the color box to choose the text color of the selected tab. The default is 255,255,255 (RGB).
Use Image	Enable (check) to use a tab image. Designate both a default and a selected tab image.

Custom Tabs

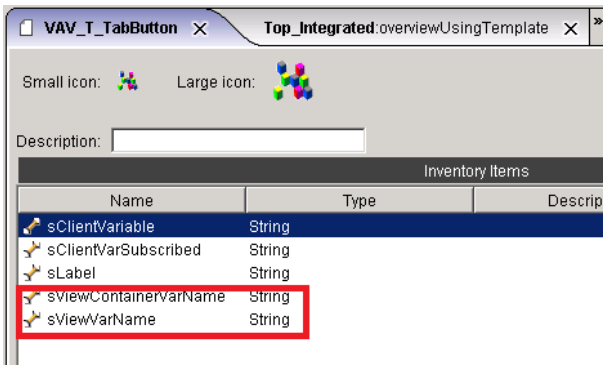
Custom tabs use either:

- The **system function**, **`selectView`**, (which is preferred);
- Or the **view variable method**, also named **`selectView`**.

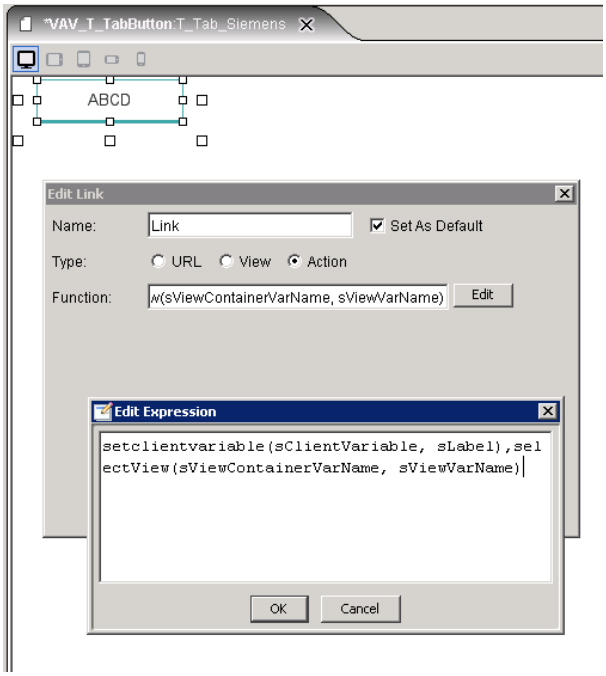
Using the System Function

Add an action link inside the tab view template. This action link uses the system function, **`showView`**, to associate the tab template with the view in the view container.

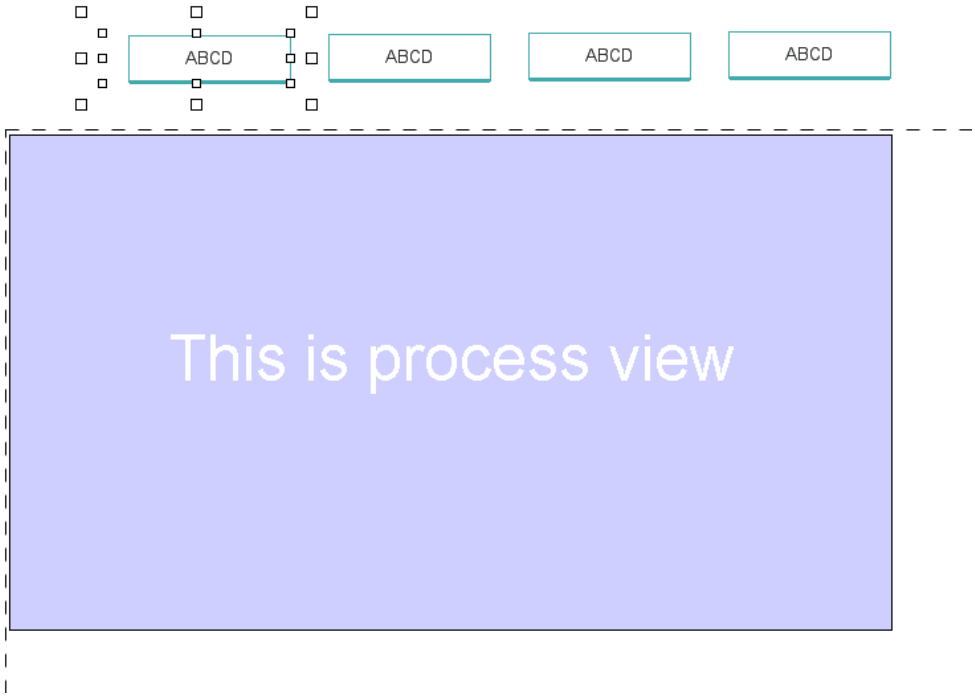
Given the following view variable names:



Add the action link, using the **`selectView`** function in the expression.



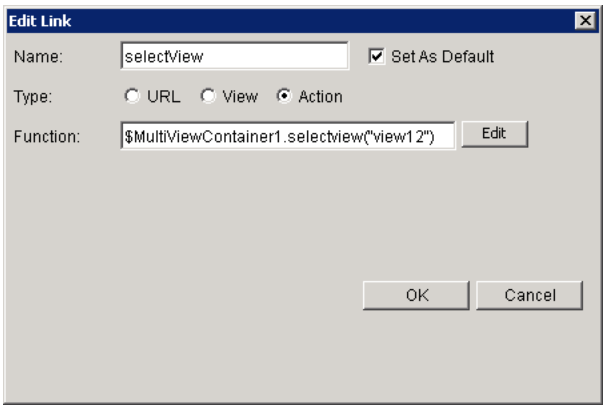
This associates the tab template with the view in the view container.



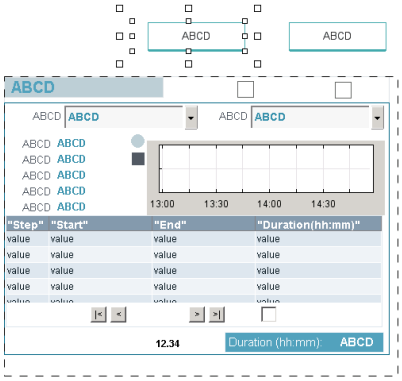
Template	
Template Component	VAV_T_TabButton
Device	Desktop
Orientation	Landscape
Template View	T_Tab_Siemens
Member Mapping	
sClientVariable	"PumpDetailTab"
sClientVarSubscribed	
sLabel	"Process"
sViewContainerVarName	"MultiViewContainer1"
sViewVarName	"vProcess"

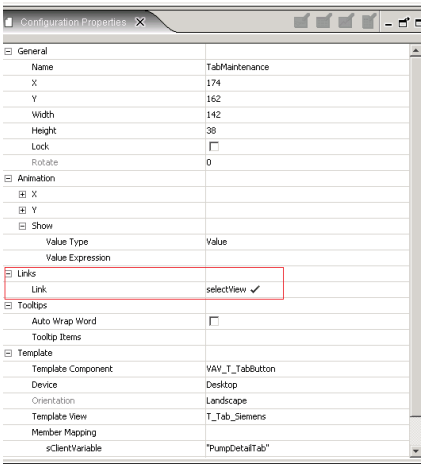
Using the View Variable Method

Add the action link to the custom tab view template.



The action link then invokes showView defined by the Multi View Container.





Adding Controls

The use of controls in a view allows the end-user to do two things:

- Input values that affect view content.
- Initiate (trigger) actions from within the view.

To prevent a collection query from being executed every time the end-user changes a value in a control, a specific control can be selected to act as a trigger.

The following control types are available:

- **Edit Box**
  The edit box allows the user to enter text.
- **Combo Box**
  The combo box allows the user to either enter or select a value from a given list.
- **List**
  The list control allows the end user to select from either a list of checkboxes or radio buttons.
- **Text Area**
  The text area allows you to add notes and logs with wrapping and scrolling text.
- **Button**
  The button allows the user to execute an expression when pressed or update the data in a collection view.

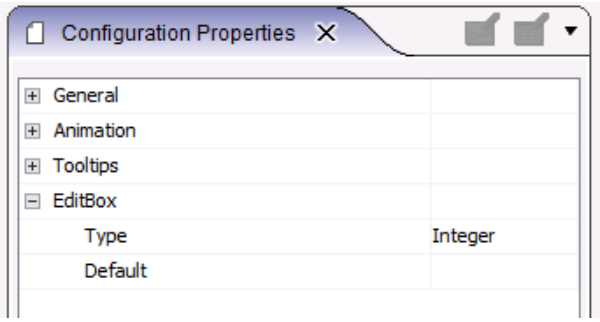
Using Local Variables

Controls are defined using *local variables*. Local variables allow contents to be referenced in expressions and used in a filter.

Using the Edit Box

Edit Box Control

The edit box control allows the end-user to enter data (String, Real, Integer, Boolean) during runtime. In addition, you can also configure the edit box control to act as a trigger to initiate a collection query just by entering (or changing) a value.



Edit Box Properties

For the edit box control, you typically configure the following data:

- **Type**

  You select what type of data can be entered in the control. This does not affect the behavior of the control, only its use in an expression.

  Once used in an expression, changing a type may invalidate the expression.

- **Default**

  The expression that generates the initial value of the edit box when it is first loaded during runtime. This value is re-evaluated each time the view is refreshed (for example, when the user navigates away from and then back to the view).

  Use the `setViewVariableValue` function to update this control after the view is loaded.

  For more information, refer to the topic, *setViewVariableValue*, located in the XHQ Reference Guide.

However, if you select the **DateTime** value for the **Type**, then the Edit Box configuration is augmented to include settings for the **Date/Time Picker**.

To add and configure the edit box control

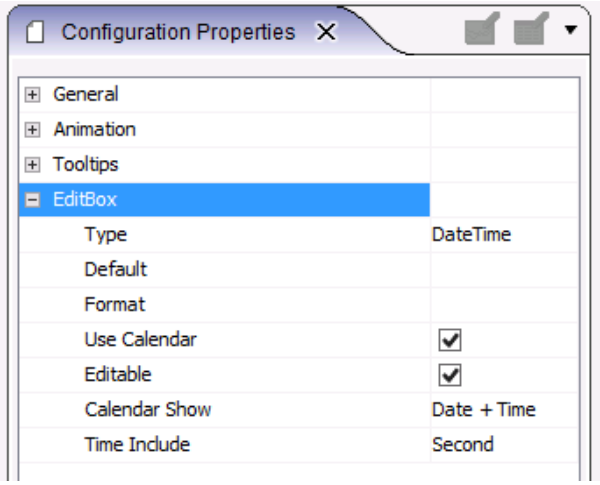
1. Insert an edit box control.
2. Click the **EditBox property**.
3. In the **Type** box, select the data type the end-user can enter in the control during runtime.

   If you select DateTime for the Type, then go to the topic, *Configuring the DateTime Picker* for additional information.

4. In the **Default Value** box, enter the expression that will be used to generate the default value for this control.
5. **Save** the view.

Configuring the Date/Time Picker

In the **EditBox property**, for **Type**, select **DateTime**. The configuration settings for the Date/Time Picker appear on the tab.



Edit Box Control - Date/Time Picker Settings

On the Drawing Panel, notice the calendar icon to the right of the Edit Box control. To enable this icon, check the **Use Calendar** box. By default, it is not checked.

Checking the **Use Calendar** box enables the following options:

- Editable box
- Calendar Show group
- Time Include group

You may set the *runtime* edit box as Editable or Read Only. By checking the **Editable** box, you are allowing the end-user to enter the Date/Time value manually during runtime.

About the Use Calendar and Editable Options

This combination...	Results in this during Runtime
☐ Use Calendar	(This is the default.) The user enters the Date/Time in the edit box. The Calendar icon is disabled.
☑ Use Calendar ☑ Editable	The user either enters the Date/Time in the box or clicks on the Calendar icon to pick the Date/Time.
☑ Use Calendar ☐ Editable	The user may only click on the Calendar icon to pick the Date/Time. The box is Read Only.

If the Picker is editable, a String typed **expression** may be used to format the Date/Time value. However, if the Format expression is not configured, or if it is configured but the format is not a valid date/time format, then the default timestamp format is used.

For **Calendar Show**, you may configure the Date/Time Picker to show Date only, Time only, or both Date and Time. The default setting is to show both.

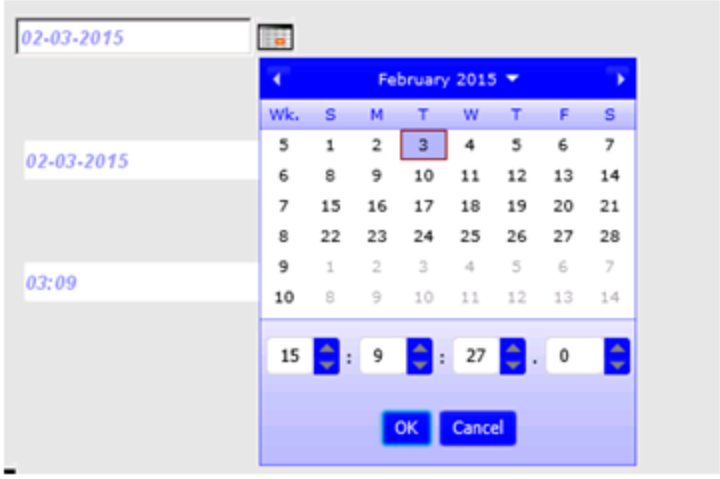
For **Time Include**, you may configure the picker to show Minute, Second, or Millisecond. The default is Second. If you selected to show Date only, then the Time Include options are disabled.

Localization

The Date/Time Picker supports localization for the month and day names. It also supports localization for the first day of the week. (For example, in the US, the first day of the week is Sunday, but in France, it is Monday.)

The Date/Time Picker at Runtime

From the XHQ Solution Viewer, if the picker is configured, a calendar icon appears next to the edit box. Click this icon to launch the Date/Time Picker.



Date/Time Picker at Runtime

Customizing the Date/Time Picker

You can customize the following Date/Time Picker features by setting the given global property.

Customizable Feature	Description	Global Property
Picker Icon	Change the default icon.	CalendarImg1
Week Column	Show the week number (first column).	CalendarShowWeekNumber
Color	Change the calendar color scheme, using RGB values.	CalendarThemeColor

To change the default icon

Changing the default icon for the Date/Time Picker is similar to replacing the icons on the XHQ NavBar.

For more information, see the topic, *Property Settings for the Navigation Bar*, located in the XHQ Administrator's Guide.

1. Select a custom image that is in **24x24 GIF** format and record its filename.
2. Place this image file in the **`<repos>/images`** directory.
3. Open the **`globalsettings.properties`** file in a text editor.
4. Add (or find and edit) the **`CalendarImg1`** property and set its value equal to the image filename.

   Property names are **case-sensitive**.

   Example:

   `CalendarImg1=NewPickerIcon`

5. **Save** and close the text editor.

To show the week number column

The week numbers appear in the **first column** of the Date/Time Picker. By default, the property is set to false, so the week number column is hidden.

1. Open the **`globalsettings.properties`** file in a text editor.
2. Add (or find and edit) the **`CalendarShowWeekNumber`** property and set its value to **true**.

   Property names are **case-sensitive**.

   Example:

   `CalendarShowWeekNumber=true`

3. **Save** and close the text editor.

To change the picker color scheme

Use the **RGB** value for the color. By default, the calendar color scheme is set to black (RGB `0,0,0`).

1. Open the **`globalsettings.properties`** file in a text editor.
2. Add (or find and edit) the **`CalendarThemeColor`** property and set its value to an RGB value.

Property names are **case-sensitive**.

Example 1:

```
CalendarThemeColor=0,0,255
```

This results in a blue theme.

Example 2:

```
CalendarThemeColor=255,0,0
```

This results in a red theme.

3. **Save** and close the text editor.

Using the Combo Box

Combo Box Control

With the combo box control, the end-user (during runtime) can either enter data or select from a list of predefined values. Like the edit box control, you can also configure the combo box to act as a trigger to initiate a collection query just by entering (or selecting) a value.

For the combo box control, you configure the following data:

- **Type**
  You select what type of data can be entered in the control.
- **Default**
  The expression that generates the initial value of the combo box when it is first loaded during runtime. This value is re-evaluated each time the view is refreshed (for example, when the user navigates away from and then back to the view).

  > Use the `setViewVariableValue` function to update this control after the view is loaded. Fore more information on this function, refer to the topic, *setViewVariableValue*, located in the XHQ Reference Guide.
  > In the XHQ HTML5 Solution Viewer (at runtime), the default value is added to the combo box list when the default value does not exist in the combo box list. For example, if the combo box list does not contain the default string value "%", then "%" will be added to the list.

- **Editable**
  By checking the *Editable* box, you are allowing the end-user to enter data into the control during runtime. This enables the user to enter a value that is not on the combo box list.
- **Allow Multiple Selection**
  By checking this box, the *Allow Multiple Selection* option is enabled.
- **Data Source**
  The data source type used to populate the combo box. The options are *Value List* or *Collection* type.
- **Value List** or **Collection**
  When you select *Value List*, you define the set of enumerated values that your end-user is to choose from. When selecting *Collection Data*, the combo box is populated (filled) with values from the given Value/Name column pair in a collection.

  > For the Collection Data, the query returned is actually a SELECT DISTINCT Name, Value from Table. The DISTINCT option is on the Value/Name pair and not just the Name. See the sidebar topic, *About the Value/Name Column Pair*.

If the combo box is not editable, then during runtime, the combo box appears with a blank (empty) entry. Selecting this blank entry will clear out the collection filter.

> The **maximum number of records** that can be returned for selection in the combo box control is **500**.

SIDEBAR: ABOUT THE VALUE/NAME COLUMN PAIR

The **Name column** is used to populate the combo box. The **Value column** is the value returned when a name is selected from the combo box.

Example: A combo box with the following columns:

Name	Value (with sort)
Jan	1
Feb	2
Mar	3

If "Jan" is selected from the combo box, then the returned value is "1".

To configure the combo box control

1. Insert a combo box control and go to the **Configuration Properties tab**.
2. Expand the **ComboBox property**.
3. In the **Type** box, select the data type.
   For example, if the Editable box is checked, this is the data type the end-user can enter in the control during runtime.
4. In the **Default Value** box, enter the expression that will be used to generate the default value for this control.
5. OPTIONAL
   Check **Editable** if you want to allow the end-user to enter the combo box data during runtime.
   This enables the user to enter a value that is not on the combo box list.

   If the combo box is not editable, then during runtime, the combo box appears with a blank (empty) entry.

6. OPTIONAL
   Check **Allow Multiple Selection**.

   For more information, go to the topic, *Multiple Selection Combo Box*.

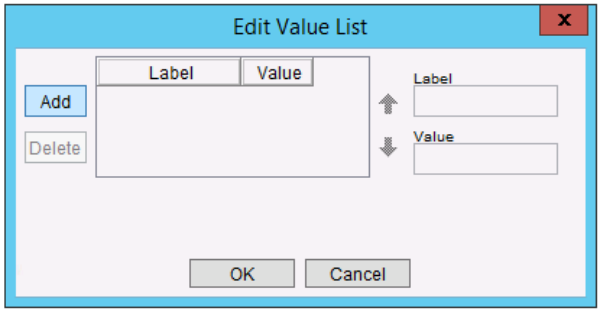
7. Select a **Data Source** (either Value List or Collection).
8. Do one of the following:
   - *Edit the* **Value List**.

   or

   - *Set the* **Collection Data**.
9. Click **OK** to save your edits and close the pop-up dialog.
10. **Save** the view.

To edit the Value List

The "Edit the Value List" dialog box appears.



1. In the **Label** box, enter the label for this list item.
2. In the **Value** box, enter the value for this list item.
3. Click **Add**.
   This populates the box with the label and value data you entered.
4. Repeat these steps to add more list items.
5. OPTIONAL
   To rearrange the list items, click either the **Up** arrow or the **Down** arrow, located next to the value list item group box.
   This moves the list item accordingly.
6. Click **OK** to save your edits and close the pop-up dialog.

To set the Collection Data

1. In the **Collection Data** cell, do one of the following:
   - Select the collection member from the available list of collection members.

   or

   - Enter the relative name of the collection member.

     Collection member names are limited to 30 characters or less.

   or

   - Click Browse to enter an expression.
2. Select the **Value**, (a corresponding) **Name**, and **Sort** columns.
3. Set the **Sort** order.

Multiple Selection Combo Box

Go to the **Configuration Properties** tab, expand the **ComboBox** properties, and check **Allow Multiple Selection**.

When Allow Multiple Selection is enabled, the combo box is not editable (the "Editable" option is grayed-out).

The given **Default** value expression must match the combo box **Type** or be a String type.

⊟ ComboBox	
Type	Integer
Default	2
Editable	☐
Allow Multiple Selection	☑
Data Source	Value List
Value List	1,2,3

Example: Single Default value, Integer type

⊟ ComboBox	
Type	String
Default	"bb"
Editable	☐
Allow Multiple Selection	☑
Data Source	Value List
Value List	aa,bb,cc

Example: Single Default value, String type

The Default for a multiple selection combo box can either be a **single** value or **multiple** values.

Use a String type to specify multiple default selected values, separating each value with a comma.

⊟ ComboBox	
Type	String
Default	"bb,cc"
Editable	☐
Allow Multiple Selection	☑
Data Source	Value List
Value List	aa,bb,cc

Example: Multiple Default values, String type

To enable multiple selection on all combo boxes for every view, set the global property, `AllowComboBoxMultipleSelection` to true (by default, it is set to false).

For more information on this setting, go to the topic, *Additional Global Settings*, located in the XHQ Administrator's Guide.

Example: Using a multiple selection combo box view variable as a collection filter

In the following example configuration, the combo box view variable, `$ComboBox1`, allows multiple selections.

⊟ Table	
Collection	ColPop
Is Group By	☐
Enable Group By	☐
Columns	"Country","State","City",
⊟ Collection Criterias	
⊟ Filter #1	
Member	sCountry
Relation	==
Expression	$ComboBox1

This results in a table that shows all records that matches any of the selected values of `$ComboBox1`.

Example: Passing selected values from one view to another

This example shows you how to pass selected values from one view to another view, which in turn uses the selected values as a collection filter.

Given:

ViewA Includes the following controls:

- **`$ComboBox1`**
  Allows multiple selection; displays a list of countries.
- **`$Button1`**
  Has expression like `setClientVariable("selected_country", $ComboBox1), showView (<ViewB>).`

ViewB Includes the following controls:

- **`$ComboBox1`**
  Allows multiple selection; displays a list of countries. The default selection is set to `getClientVariable("selected_country").`
- **`$Table1`**
  Shows collection filtered by `$ComboBox1`.

At Runtime:

The user opens ViewA, selects countries from the multiple selection combo box (`$ComboBox1`), and then clicks the button (`$Button1`) to navigate to ViewB.

ViewB opens; the selected countries are checked in the multiple selection combo box (`$ComboBox1`), and the data in the table control (`$Table1`) are filtered accordingly.

What multiple selection looks like at runtime

Each item in a multiple selection combo box is associated with a checkbox.

Example: Multiple Selection Combo Box at Runtime

The editor field of the combo box displays the total number of items selected. Check **`(All)`** to select (check) all items listed in the combo box. To clear (deselect) all boxes, first check **`(All)`**, and then uncheck **`(All)`**.

Things to Note

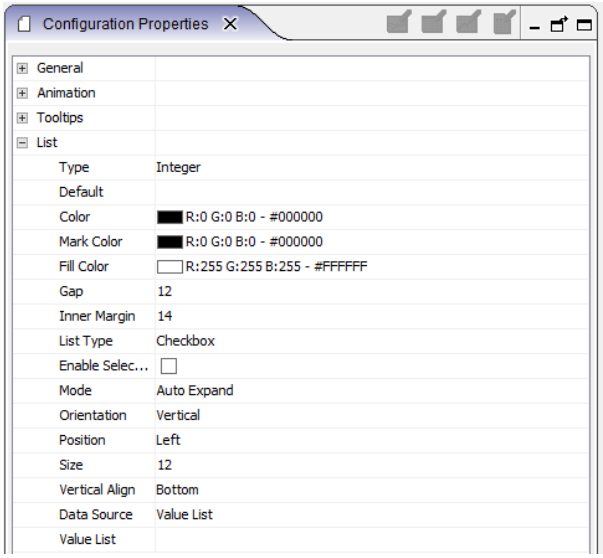
When multiple items are selected in the combo box:

- And the combo box view variable is associated with a collection query filter, then all selections are OR'd together to form query criteria.
- And the combo box view variable is used elsewhere, the value is only valid if the data type of the combo box is String or Text. If the data type is other than String (or Text), BAD quality is displayed.
- And the user saves this view (with multi-selections) to "My Views", the filter is retained as part of the saved views.

Using the List Control

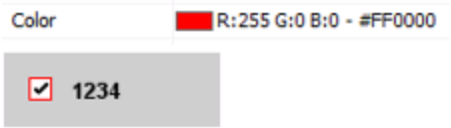
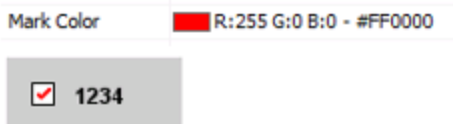
List Control

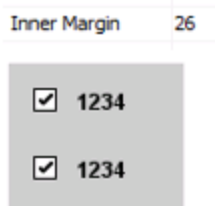
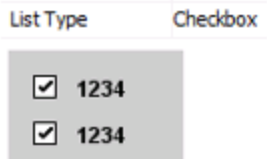
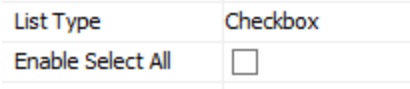
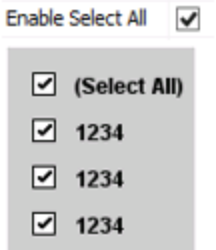
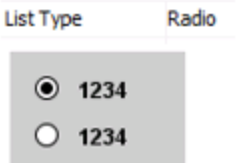
With this tool, the end-user (during runtime) can make selection(s) from either a list of checkboxes or radio buttons. Like the combo box control, you can use either a Value List or Collection data source.



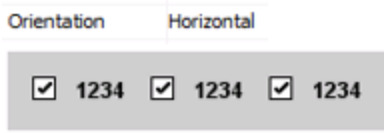
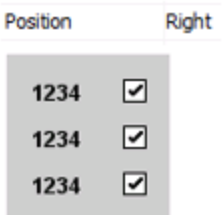
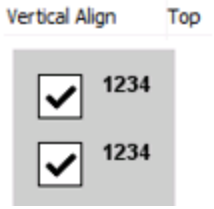
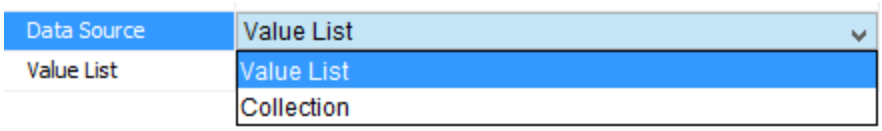
List Properties

List Property Settings

List Property	Settings
Type	Select what type of data can be entered for the control.
Default	The expression that generates the initial list values when the control is first loaded during runtime. This value is re-evaluated each time the view is refreshed.
Color	The color of the checkbox/radio button outline. 
Mark Color	The color of the check mark (checkbox)/selection mark (radio button). 
Fill Color	The color of the fill inside the box/button.

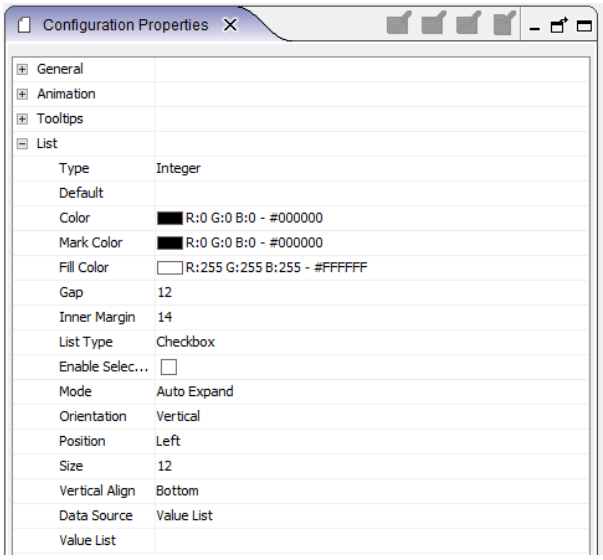
List Property	Settings
Gap	The horizontal space (in pixels) between the box/button and the label.
Inner Margin	The vertical space (in pixels) between each list item. 
List Type > Checkbox	Create a checkbox list. Enables user to select multiple items.  
Enable Select All	Added "Select All" option that enables the user to automatically select all list items. 
List Type > Radio	Create a radio button list. User is limited to just one option from the list provided. 

List Property	Settings
	List Type: Radio Select First: ☐
Select First	The first list item is automatically selected. Select First ☑ ◉ 1234 ○ 1234 ○ 1234
Mode	These modes control how long lists are shown. *Options*: • **Auto Expand** Automatically shows entire list. Mode: Auto Expand ☑ 1234 ☑ 1234 ☑ 1234 ☑ 1234 ☑ 1234 • **Scrollable** Enables the user to scroll through list items. Depending on the Orientation of the list, the scroll bar will either be vertical or horizontal. Mode: Scrollable ☑ 1234 ☑ 1234 ☑ 1234 ☑ 1234 ☑ 1234
Orientation	Layout list items vertically or horizontally.

List Property	Settings
	
Position	The checkboxex/radio buttons can be positioned to the left or right of the corresponding label. 
Size	The size of the checkbox/radio button.
Vertical Align	Align the label to the Top, Center, or Bottom of the checkbox/radio button image. 
Data Source	The data source type used to populate (fill) the list items. The options are **Value List** or **Collection** type. 
Value List	When you select *Value List*, you define the set of enumerated values that your end-user is to choose from.
Collection	When selecting *Collection Data*, the combo box is populated with values from the given Value/Name column pair in a collection.

To configure the list control

1. Insert a list control and go to the **Configuration Properties tab**.



2. Expand **List**.
3. In the **Type** box, select what type of data can be entered for the control.
4. In the **Default Value** box, enter the expression that will be used to generate the default value for this control.
5. Set the *various properties* that control the "look and feel" of the list items.
6. Select a **Data Source** (either Value List or Collection).
7. Do one of the following:
   - *Edit the* **Value List**.

   or

   - *Set the* **Collection Data**.
9. Click **OK** to save your edits.
10. **Save** the view.

Using the Button Control

You use the button control to initiate a collection query once it is associated with a view-in-view's collection filter. You can also use the button to start a given action when it is pressed. In fact, a button can initiate both a collection query AND a given action simultaneously. It is activated when it is pressed and **fires only once for each "press and release" cycle**.

For the button control, you configure the following data:

- **Label** (optional)
  The button label allows you to visually distinguish it from other buttons in the same view. The default label is set to "Apply".

  When a button is added, it is given the default label of "Apply". You can modify the name but **you cannot leave it blank**. If you leave the control name blank, it will result in the last known name being restored.

- **Expression** (optional)
  The expression allows the button to take a specific action when it is pressed.

  When pressed, the expression associated with the button is evaluated. By using an `eval()` function, you can execute JavaScript to take an action.

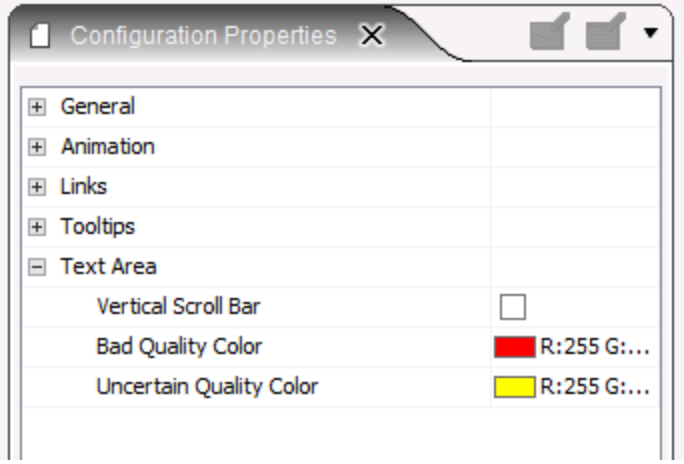
To add and configure a button control

1. Insert a button control.
2. Click the **Button property**.
3. In the **Label** box, enter the control label.
4. OPTIONAL
   In the Expression box, enter an expression.
5. **Save** the view.

Using the Text Area Control

Text Area Control

The text area allows you to add notes and logs with wrapping and scrolling text.



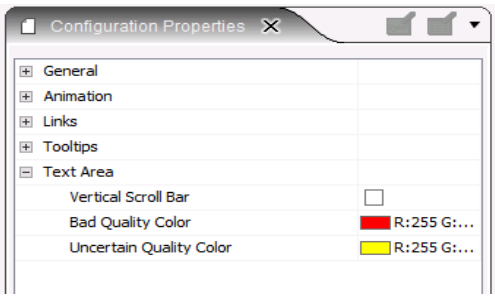
Text Area Control

For the text area control, you configure the following data:

- **Vertical Scroll Bar**
  Check to include a vertical scroll bar on the text area.
- **Value Display**
  Select color displays for Bad and Uncertain values.

To add and configure a text area control

1. Insert a text area control.
   The component's primitive members appear in a list at the insertion point.
2. Select the primitive member and draw the text area control.
3. Click the **Text Area** property.



4. OPTIONAL
   Click (check) **Vertical Scroll Bar** to add a vertical scroll bar to the text area.

5. OPTIONAL
   Set **Bad** and **Uncertain** value colors by clicking on the color square.
   This launches the Property Color selector.

Configuring the Collection Filter

After creating your controls, the next step is to configure the collection filter. This is done by defining the filter *criteria* through the **View property**.

A collection filter is made up of one or more criterion. An empty criterion is added to the filter and its various elements are modified afterwards. Therefore, you can:

- Add criterion to a collection filter.
  You can add one or more.
- Modify the criterion once it is added to the collection filter.
  This is done by specifying and modifying either an operand or a relation.
- Delete a criterion from a collection filter.

Criteria are composed of up to four elements: one to three *operands* and a *relation*. The operands are entered as expressions and the relation is chosen from a given list of supported relations.

About Operands

Operands provide the values that are used to compare against the values in the collection for the purpose of filtering or ordering the collection records. Operands are defined using expressions that may contain data references that must be within the context of the component containing the collection member (not the collection itself) or variable references that are defined on the view itself.

About Criteria Relations

Criteria relations define the logical relationship between operands in the criteria structure.

Criteria Descriptions

Symbol	Meaning	Description
>	greater than	Placed between two operands to determine if the first value is greater than the second value.
<	less than	Placed between two operands to determine if the first value is less than the second value.
>=	greater than or equal to	Placed between two operands to determine if the first value is greater than or equal to the second value.
<=	less than or equal to	Placed between two operands to determine if the first value is less than or equal to the second value.
==	equal to	Placed between two operands to determine if the first value is equal to the second value.
!=	not equal to	Placed between two operands to determine if the first value is not equal to the second value.

Symbol	Meaning	Description
Between	between	Placed between two operands to determine if the value is between the two values. This is inclusive. For example, "3 between 10" is seen as "3≤x≤10" and includes both "3" and "10".
Null	is null	Sets Member operand as Null.
Not Null	is not null	Sets Member operand as Not Null.

Collection View Layout

You can set the number of rows and columns as well as insert navigation buttons to the collection view.

Collection View Placement

Control Name	Description
Rows	Enter the number of rows.
Cols	Enter the number of columns.
No Layout	Select (check) to apply no layout. Overlays the view of each record on top of one another (each record's view is placed in the same location). Typically, animation configurations are used to set the location at runtime.
Nav. Btns	Select (check) to insert navigation buttons that allow you to scroll through the set of collection views.
	|< Scrolls to the **First** view.
	< Scrolls to the **Previous** view.
	> Scrolls to the **Next** view.
	>| Scrolls to the **Last** view.

Executing the Collection Query

With view-based collection filtering, you can execute a collection query without requiring a change to the collection. For example, a control can be linked to a view-in-view's collection filter and used to execute the collection query from within a view.

In the case of a button control, you can explicitly associate it with a view-in-view's collection filter. When the end-user presses the button control at runtime, it will cause the view-in-view to reevaluate its criteria and then execute the collection query.

The combo box and edit controls may also be used when a button control is not associated with a view-in-view's collection filter. In general, any control variable used in an expression that is part of a criteria operand will cause a collection query to be executed. So, an end-user can change the value of a control linked with a collection filter and execute the collection query.

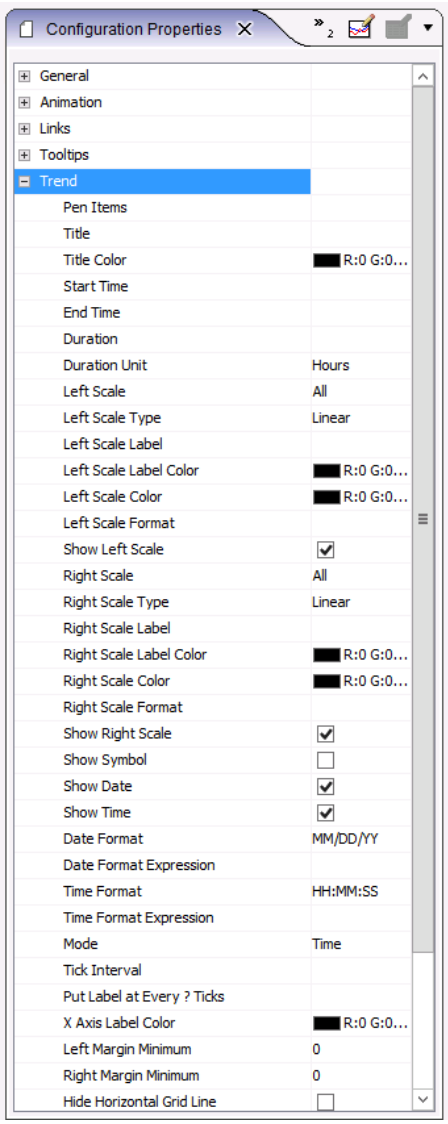
Configuring an Embedded Trend

Use the Trend tool to insert a trend plot into a view and configure it.

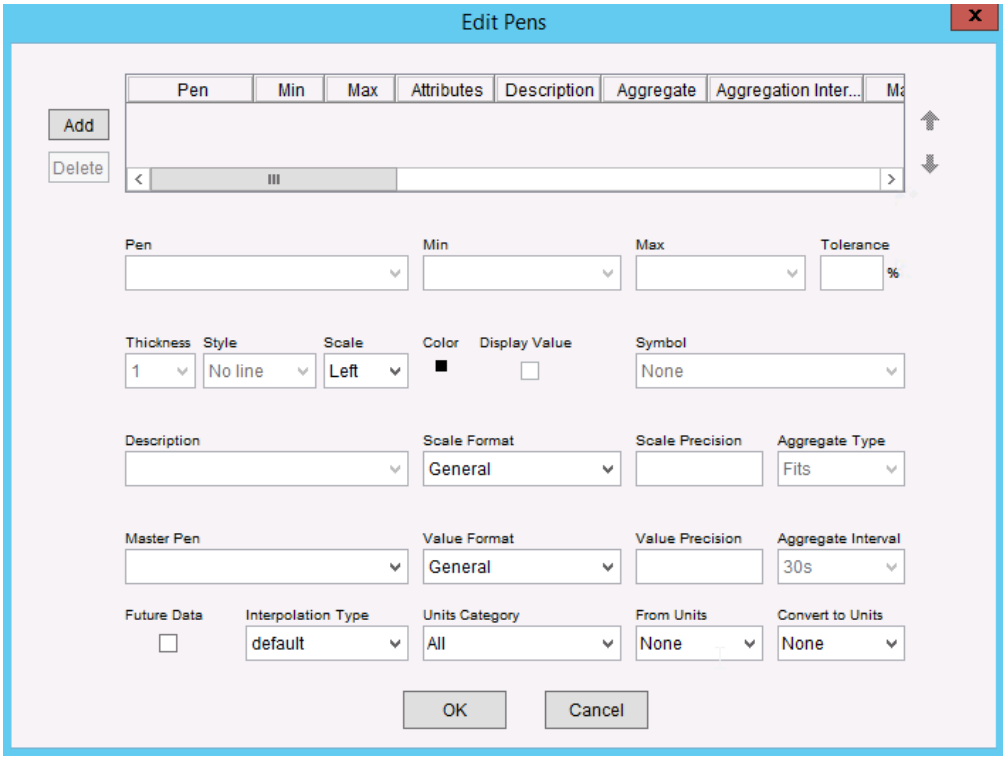
To place a trend, you simply click the **Insert Trend** tool . Then, on the **Drawing Panel**, draw a rectangle at the position where you want to insert the trend. From the **Configuration Properties** tab, expand **Trend**.

There are two major items to consider when configuring trend plots:

- Trend Property Values
- Pen Configuration



Trend Properties



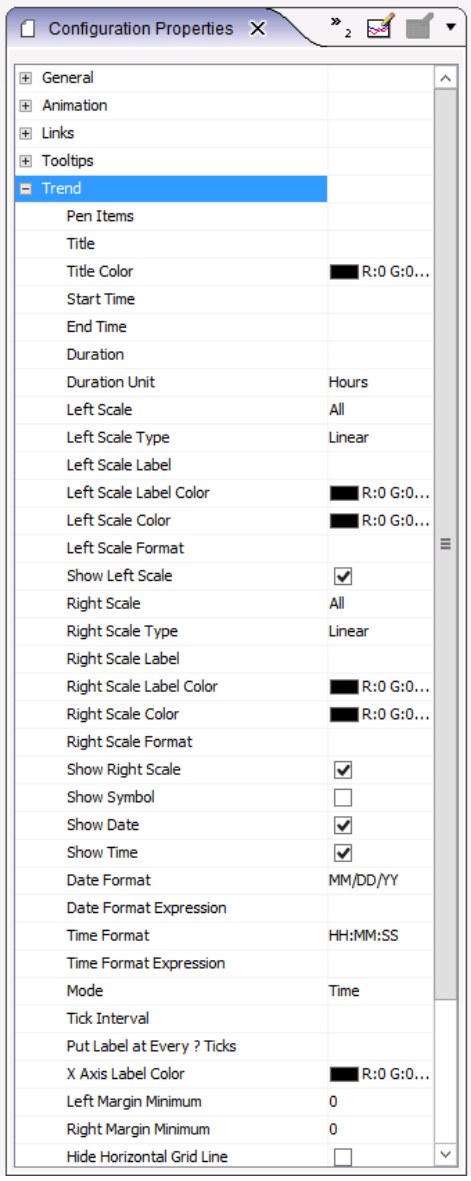
Pen Properties

To edit trend property values or pen configuration, click the "Edit" button next to the respective value list table.

The trend properties and pen attributes you set for the embedded trend are automatically applied to the related XHQ Trend Viewer (XHQ Interactive Trender) during runtime.

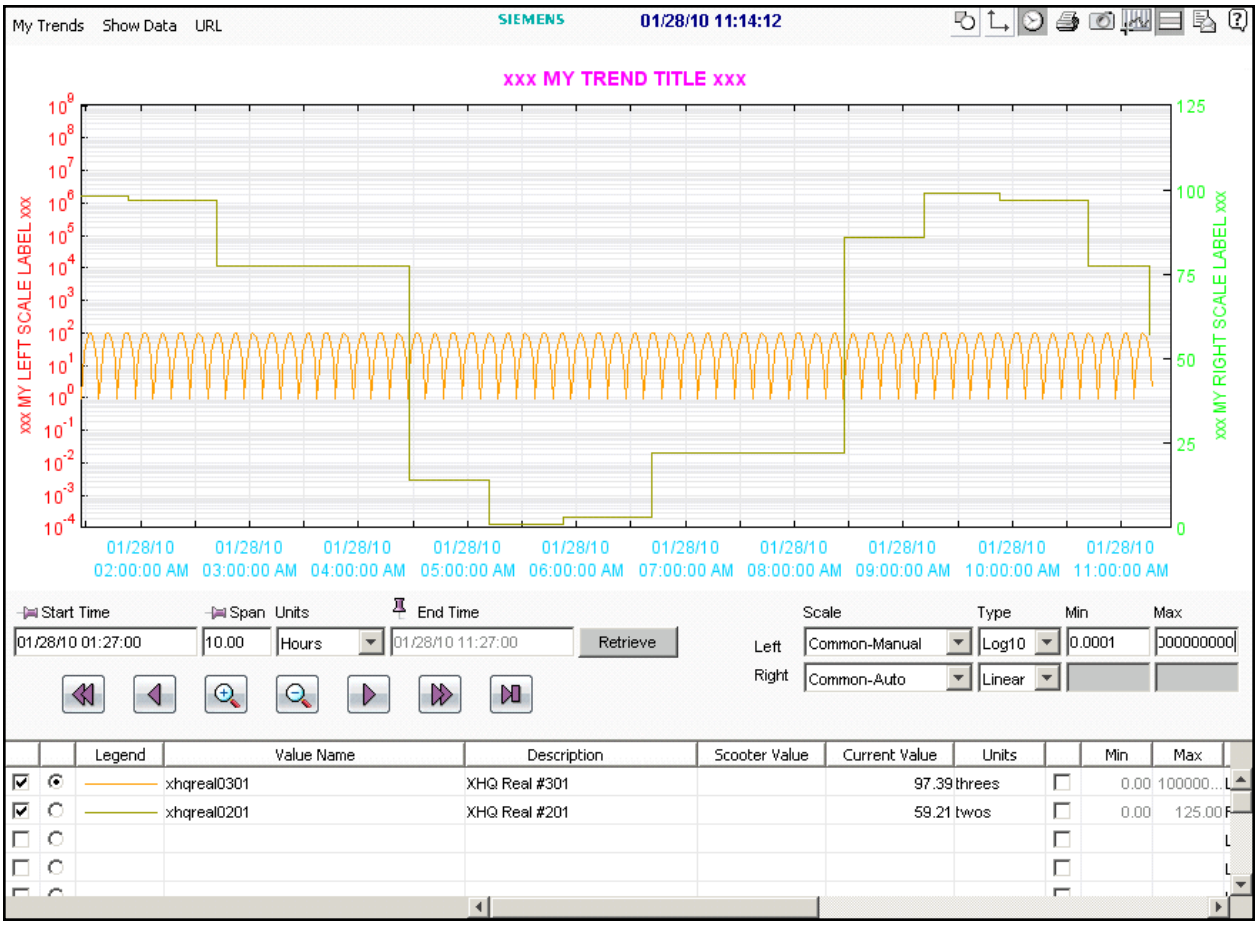
Trend Properties: Concepts

This section is the first of two parts that deals with the Trend Properties dialog box. This part discusses some concepts behind the different properties you can set.



Trend Properties

Grayed-out properties cannot be edited. The default display mode uses All for both Left and Right scales. In this case, the scale color is the pen (or legend) color.



XHQ Trend Viewer Example

About the Embedded Trend Chart Grid

The embedded trend chart displays up to sixteen *line traces* (pens). Line traces are curvilinear representations of how a variable changes over time.

In the vertical, y-axis direction (also known as the **Scale** axis), the grid spacing is always set to 5 intervals. In the horizontal, x-axis direction (the **Time** axis), the number of grid intervals is automatically determined, depending on the following factors:

- Trend size
- Date/time format (which affects the label length)
- Trend duration

About Start Time, End Time, and Duration/Span

For the trend's Start Time, End Time, and Duration (or Span) properties, you can enter a string expression or a string constant, or choose from the list of given primitive members. If you select "None" or leave the list box empty, the default property value is used. The default of End Time is the current time; the default of Duration is two (2) hours; and the default Start Time is calculated using the default End Time minus the default Duration.

String expressions are evaluated during runtime. If the string expression is not in a valid date/time format, the default property value is then used.

It is important to note that you only need to configure **only one or two** of these three properties. If you inadvertently configure all three, an error message appears.

About Scale Properties

The Scale (y-axis), which shows the relative minimum and maximum trend values, has four display modes:

- **All**
  Which displays the Scale for all pens (traces).
- **None**
  Which hides the Scale.
- **Common-Auto**
  Which automatically sets a common max and min value for all pens displayed.
- **Common-Manual**
  Which enables you to set a common max and min value for all pens displayed. When you set the "Scale" to "Common-Manual", the "Manual Min" and "Manual Max" fields are enabled, allowing you to set these values.

About Single Scale

In **Single** Scale mode, the Y-scale displays the trend values for the trace you selected. To select a trace, you simply click the corresponding radio button in the second column of the Legend panel.

About All Scales

When the Trend Viewer is first invoked, the default mode is **All** Scales.

This may not be the case for a Trend Viewer that is launched from an embedded trend.

For embedded trends, trend properties and pen attributes can be pre-configured during design time. In runtime, when you launch the Trend Viewer from the embedded trend, the properties you set are automatically applied. For example, if you set the Scale property to Common-Manual during design time, then when you launch the Trend Viewer from the embedded trend, the Common-Manual scale displays.

All Scales mode displays the relative minimum and maximum values for all your traces. A pair of labels appear for each trace, matching the given trace color. The maximum values are at the top of the Y-scale, and the minimum values at the bottom. The maximum values are in descending order, according to the order of their appearance in the legend panel. The minimum values are in ascending order.

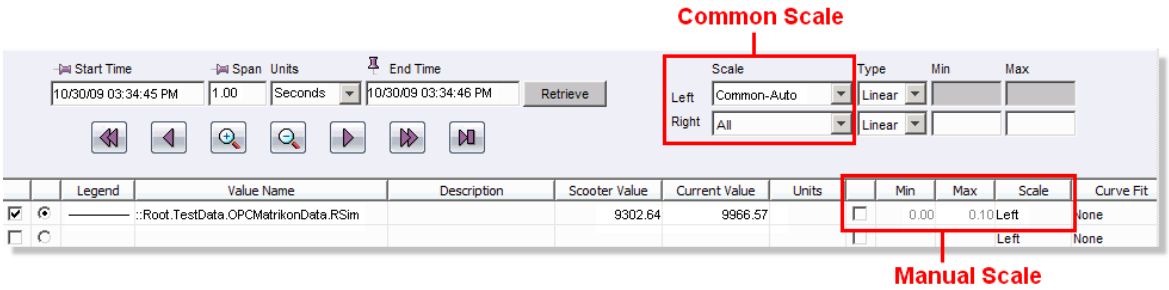
Things to note when working with Pen Groups

When the Scale is set to **All** for a pen group, the attribute of the master pen (such as tolerance, precision, or color) are applied. The high/low limit of the pen group is determined by the min/max and the tolerance of the master pen.

Setting Min and Max Values

From the Trend Viewer, there are two ways of setting Min and Max Values. The first is by *common scale*.

From the Scale drop-down box, located on the Control Panel, select **Common-Auto** to automatically set a common max and min value for all pens displayed; or select **Common-Manual**, to manually enter a common max and min value the given pen.



Setting Min and Max Values by Common Scale or by Manual Scale Input

The second way to set Min and Max values deals with enabling *manual input*. The checkboxes to the left of the Trend Scale Min and Max Values allow you to disable or enable manual input of min and max values.

About the Logarithmic Trend Scale

In addition to Linear, you can also set the scale type to:

- **Log10**
  Which is the *common* logarithm of base-10. To use Log10 based format, set **`UseLog10BaseFormat=true`** in globalsetting.properties file.
- **Log**
  Which is the *natural* logarithm of base-e.

The *logarithm* of a given number is the power to which the base is raised to yield the number.

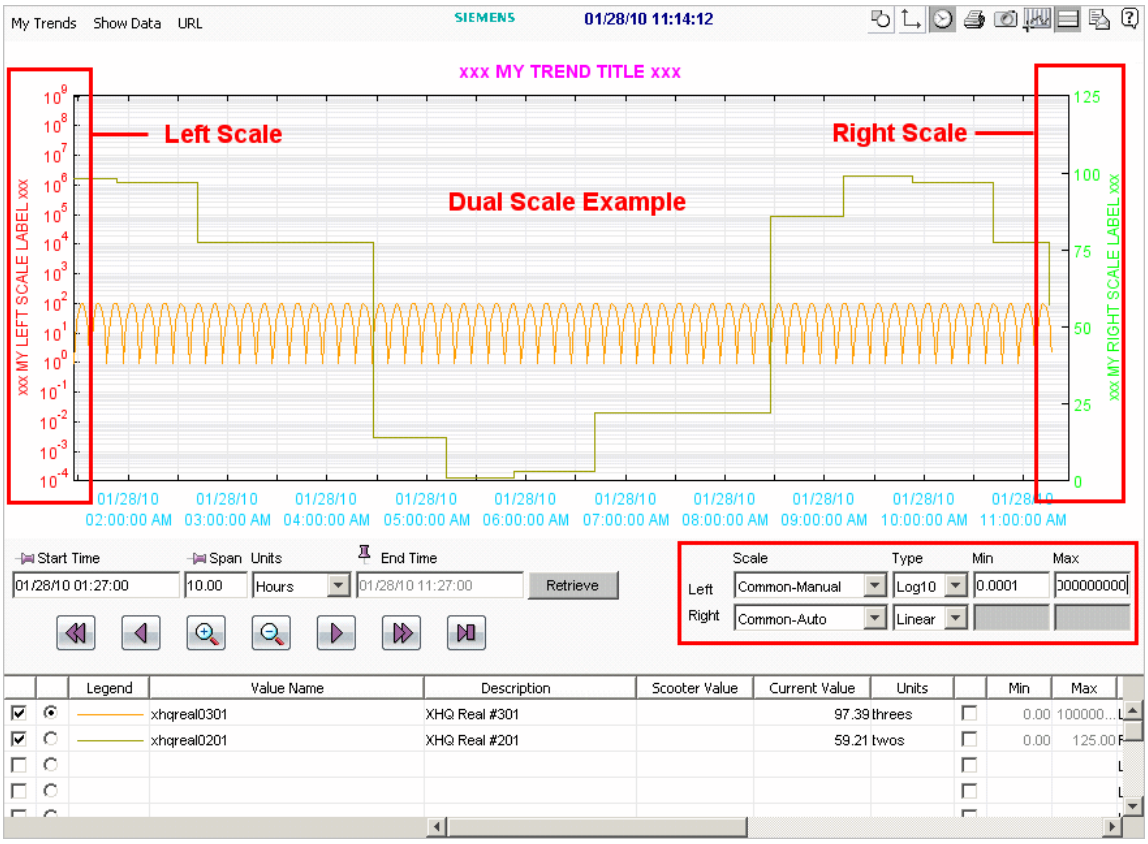
So, `logarithm (base b) of n=x if b^x=n.`

Where `n` is the given number; `b` is the base; and `x` is the power raised.

Also, `(base)^log(n)=n.`

Using Dual Scale

Pens can be assigned to display in either the Left- and/or Right-hand Y-scale.



Dual Scale Example

In the example above, the **Left scale** is set to Common-Manual, so a Min value of 0.0001 and a Max value of 1000000000 is manually entered. Because the Scale Type of Log10 is selected, the scale tick marks are displayed in the standard logarithm scale of base 10, with a Min value of 10^{-4} and a Max value of 10^{9}.

The **Right scale** is set to Common-Auto, and so a common Min and Max value is automatically set for all pens displayed. In this case, it is a Min value of 0.00 and a Max value of 125.00. The Scale Type is Linear.

About the Aggregate Interval

The *aggregation interval* is the interval over which the aggregate is computed. These intervals are defined in seconds, minutes, hours, days, or weeks.

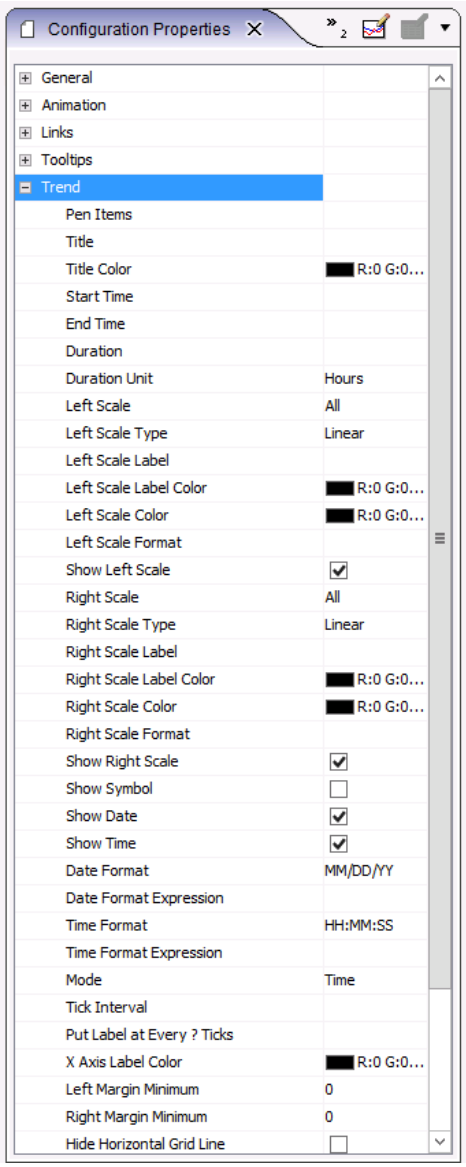
The aggregation interval is meaningful only for the Average, Min, Max, and Interpolate retrieved types. It has no affect on retrieval of Raw or Fits data.

The aggregate interval values listed in the Trend Viewer can be set by the administrator, by editing the `globalsettings.properties` file. You may also enter any desired aggregation interval by typing it directly in the interval field.

For more information on setting the aggregation intervals, refer to the topic, *Using globalsettings.properties*, located in the XHQ Administrator's Guide.

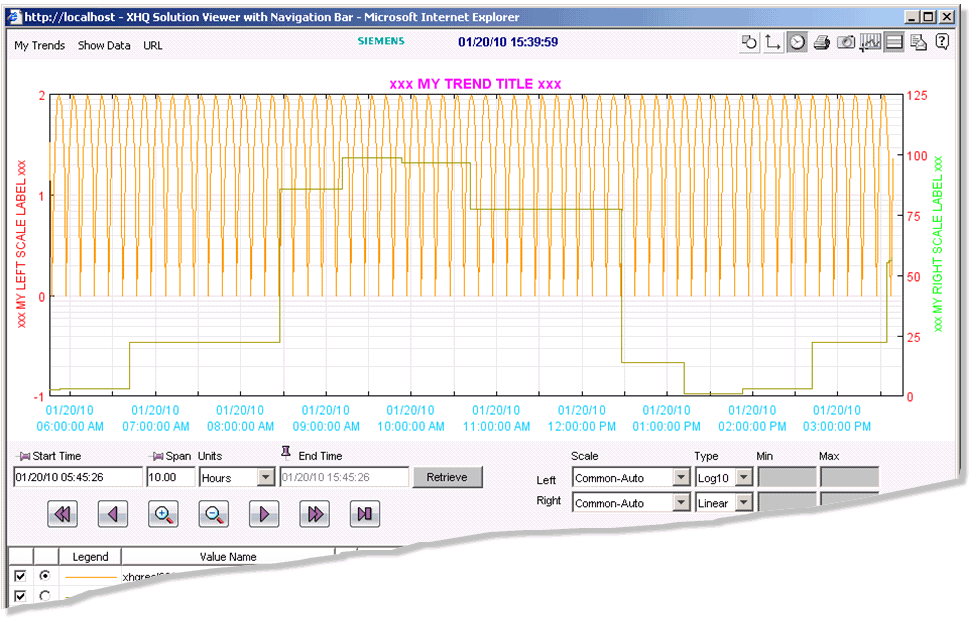
Trend Properties: How To Set

To access the Trend properties, insert a trend and go to the Configuration Properties tab.



Trend Properties

The property values you set for the embedded trend are automatically applied to the related XHQ Trend Viewer (XHQ Interactive Trender) during runtime.



Edit Trend Properties

Property	Description
Title	Displays the trend title. This field accepts string expressions, which allow you to have some dynamic determination of the title at runtime. You can also choose from the list of given primitive members. The title must be enclosed in **double quotation marks**. *Example:* "Acme Trends"
Title Color	Sets the trend chart title color (in RGB format). *Example:* 255,0,255
Start Time	Sets the trend start time. Enter a string expression or a string constant, or choose from the list of given primitive members. The default start time is calculated as the default end time minus the default duration (span) time. **Important:** In regards to the Start Time, the End Time, and the Duration, you will only need to configure **one or two** of these three properties.
End Time	Sets the trend end time. Enter a string expression or a string constant, or choose from the list of given primitive members. The default end time is the current time plus the `MaxTrendFreeTime` value. The display end time on the trender is the end time specified by the user plus the `MaxTrendFreeTime`. See the topic, *Additional Global Settings*, located in the XHQ Administrator's Guide, for details on the **`MaxTrendFreeTime`** property.
Duration	Sets the time span to view the trend. Enter a string expression or a string constant, or choose from the list of given primitive members. The default duration (span) is 2 hours.

Property	Description
Duration Unit	Sets the duration in weeks, days, hours, minutes, or seconds. The default duration unit is hours.
Scale *(Left and Right)*	Enables you to view the Y-scale values for the following: • **All** Which displays the Scale for all pens (traces). This is the default Scale value. • **None** Which hides the Scale labels. • **Common-Auto** Which automatically sets a common max and min value for all pens displayed. • **Common-Manual** Which enables you to set a common max and min value for all pens displayed.
Scale Type *(Left and Right)*	Displays the Y-scale in: • **Linear** This is the default. • **Log10** Common logarithm of base-10. • **Log** Natural logarithm of base-e.
Manual Min *(Left and Right)*	Allows you to set the minimum limit for the scale. Enter a string expression or a string constant, or choose from the list of given primitive members. **Note:** This property is **dependent on the Scale**. It is only enabled when the Scale property is set to **Common-Manual**.
Manual Max *(Left and Right)*	Allows you to set the maximum limit for the scale. Enter a string expression or a string constant, or choose from the list of given primitive members. **Note:** This property is **dependent on the Scale**. It is only enabled when the Scale property is set to **Common-Manual**.
Scale Label *(Left and Right)*	Displays the Y-axis label. This field accepts string expressions, which allow you to have some dynamic determination of the label at runtime. You can also choose from the list of given primitive members. The title must be enclosed in double quotation marks. *Example:* "Units Exported"
Scale Label Color *(Left and Right)*	Sets the Y-axis label color (in RGB format). *Example:* 255,0,0
Scale Color *(Left and Right)*	Sets the color for the Y-axis tickmarks (in RGB format). *Example:* 255,0,0
Scale Format *(Left and Right)*	Sets the number formatting for the left/right scale when the left/right scale (respectively) is in Single Scale mode (Common-Auto, Common-Manual).

Property	Description
	The scale format uses the following: • "0" (zero) defines a digit, zero included. • "#" (pound sign) defines a digit, but zero is not included. • "E" is scientific notation. *Example:* "0.###E0" formats the number 1234 as "1.234E3"
Show Scale *(Left and Right)*	Set to either True (to display the scale) or False (to hide).
Show Symbol	If set to True, displays the assigned pen symbols on the trend. If False, the symbols are hidden. **Note:** This property is **dependent on the Mode**. It is only enabled when the plot Mode property is set to **Time**.
Show Date	If **TRUE**, the Date values are displayed on the X-axis label of the Trend Chart. If **FALSE**, the Date values are not displayed on the X-axis label of the Trend Chart.
Show Time	If **TRUE**, the Time values are displayed on the X-axis label of the Trend Chart. If **FALSE**, the Time values are not displayed on the X-axis label of the Trend Chart.
Date Format	This sets the date format for the X-axis label of the Trend Chart. This must be a valid Java date format. Select from the given set of formats available.
Time Format	This sets the time format for the X-axis label of the Trend Chart. This must be a valid Java time format. Select from the given set of formats available.
Mode	Select either the Time or XY-plot mode. **Note:** In XY-plot, also known as scatter plot, mode, only symbols are displayed. The set of values plotted is determined by the time span, or duration, you specify. These values represent the interpolated values at the given aggregate interval selected.
Tick Interval	Sets the number of tick marks displayed on the X-axis.
Put Label at Every ? Ticks	Specify a number that is used to place the date time label of the time line ticks. For example, if you enter the value 3, the trend places the time label at every three ticks.
X-Axis Label Color	Sets the X-axis label color (in RGB format). *Example:* 0,200,255
Margin Minimum *(Left and Right)*	Sets the margin distance (in pixels) of the left and/or right edge of the trend chart from the Trend Viewer window.

Property	Description
Hide Horizontal Grid Line	Set to True to hide the horizontal grid, or False to display.
Hide Vertical Grid Line	Set to True to hide the vertical grid, or False to display.
Hide X-Axis Ticks	Set to True to hide the tick marks on the X-axis, or False to display.
Hide Y-Axis Ticks	Set to True to hide the tick marks on the Y-axis, or False to display.
Show Legend	Set to True to display the legend, or False to hide.
Border Color	Sets the color of the trend chart border in RGB format. **Tip:** This can be used to hide the trend chart border. The default is black. *Example:* 0,0,0
Background Color	Sets the color of the trend chart background in RGB format. By default, the background color is white (RGB 255,255,255). *Example:* 255,255,0

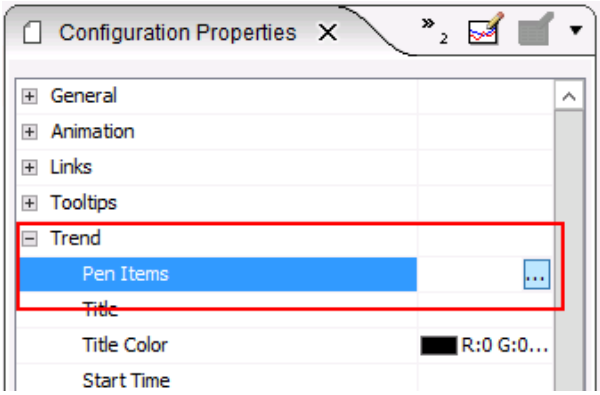
To set or edit a trend property

1. From the **Configuration Properties tab** Trend properties.
   The "Edit Trend" dialog box appears.
2. Click inside the cell next to the property you want to set.
3. Depending on the property type, do one of the following:
   - Click the down arrow and select a value from the given list.
   - Enter a value using the proper format.

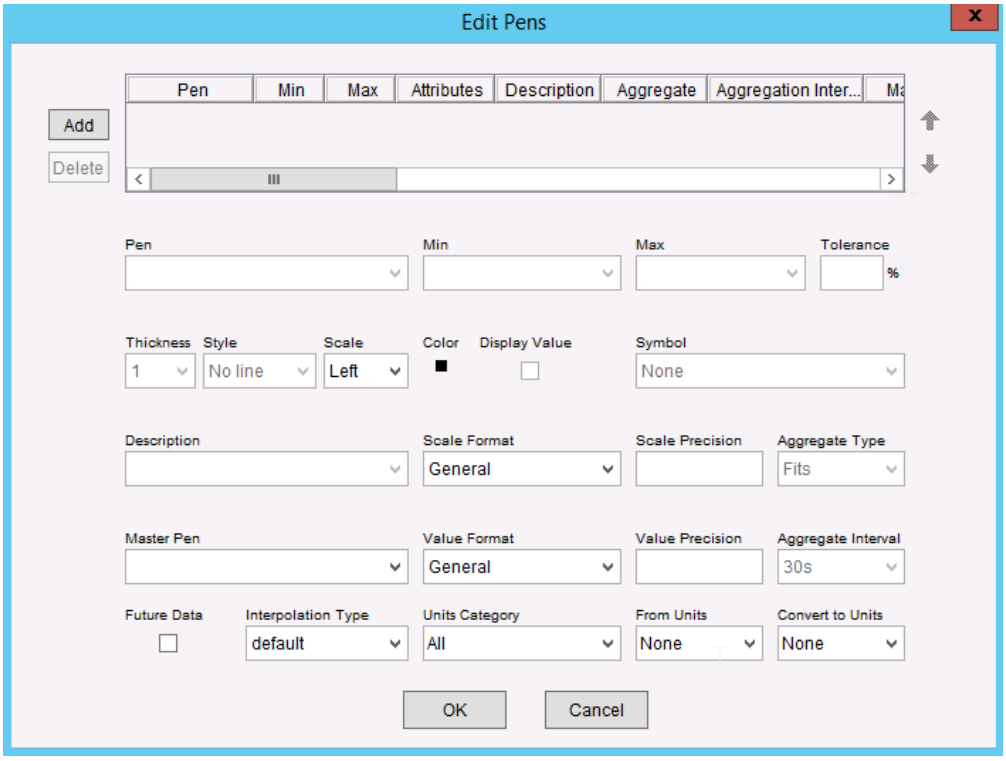
Pen Attributes: Concepts

This section is the first of three parts that deals with the Pen Attributes dialog box. This part discusses some concepts behind the different properties you can set.

The images below show you how to access the dialog box, and what it actually looks like.



Edit or Add Pen Attributes



Pen Attributes

About Pen Tolerance

The *pen tolerance* is the percentage value applied to the maximum and/or minimum when calculating the high and/or low limit of each trend, respectively. The default tolerance value is 5%.

How Tolerance Factors Into High/Low Limit Calculations

To determine the trend high/low limit, you can apply (multiply) the tolerance (percentage) to the range of the Min and Max values or to the absolute Min or absolute Max value.

To apply the tolerance to the range of Max and Min values, the formulas are:

```
high_limit = max + (max - min) * tolerance
low_limit = min - (max - min) * tolerance
```

To apply the tolerance to the absolute Min (for the low limit) or absolute Max (for the high limit) value, the formulas are:

```
high_limit = max + abs(max) * tolerance
low_limit = min - abs(min) * tolerance
```

About Scale Format and Precision

Scale Format enables you to select the format for the y-axis scale labels. There are three pen scale formats to choose from: "General," "Number-Scientific," or "Number-Fixed". If either the "Number-Scientific" or the "Number-Fixed" format is selected, the Scale Precision text box is enabled, allowing you to enter the number of digits to display to the right of the decimal point. The default scale precision is zero (0).

About Value Format and Precision

If you choose to display pen values on the trend chart, you can set how these values are to appear. Like the Scale, you have three pen value formats to choose from: "General," "Number-Scientific," or "Number-Fixed". If either the "Number-Scientific" or the "Number-Fixed" format is selected, the Value Precision text box is enabled, allowing you to enter the number of digits to display to the right of the decimal point.

A benefit in setting the Value Format and the Value Precision is that if you export the pen values to Excel, the values will retain the same format you set.

How "General", "Number-Scientific", or "Number-Fixed" Formats are Displayed

The *General* format displays the number as it was entered (or received). This is the default format. It does not impose any user-configured precision as the other two format types. (Therefore, when the format is set to General, the Precision field is disabled, or grayed-out.)

The *Number-Fixed* format forces the Precision you indicate, no matter what the intrinsic precision of the number is. (*Precision* is the maximum number of significant digits after the decimal point.) For example, if the intrinsic number has more digits after the decimal point than the indicated Precision, then it is rounded to the appropriate position after the decimal. On the other hand, if the intrinsic number has fewer digits than the indicated Precision, then extra zeros are added.

Example: Number-Fixed Format

If the number is . . .	And the Precision is . . .	Then it will be displayed as . . .
3.82	4 (decimal places)	3.8200
3.8273	2	3.83

The *Number-Scientific* format is useful when expressing very large or very small numbers. The number is displayed in scientific format with an exponential.

Example: Number-Scientific Format

If the number is . . .	And the Precision is . . .	Then it will be displayed as . . .
3.824	2 (decimal places)	3.82 E0
392.65432	3	3.927 E2 The decimal is placed just past the first non-zero number.

More about Numeric Value Formats

When displayed in the General format (which is the default), scooter values, real-time values, and relative minimum and maximum values use the following rules.

There is a maximum of six significant digits and *bias* is placed to the left of the decimal point. This means that as the number increases, fewer digits are shown to the right of the decimal point.

Example: 99,999.783 is displayed as **99999.8**

In addition, values greater than or equal to 10^6, or values that are less than or equal to 10^{-6} are expressed in scientific notation.

Example: 10,988,673.22 is displayed as **1.09887E7**

About Aggregate Types (History Retrieval)

The following table describes the available aggregate (history retrieval) types.

Aggregate Types

Type	Description
Fits	Up to 1,000 raw data values selected to reproduce the appearance of plotting raw data. The aggregation interval is not used in this mode.
Average	The average value over each aggregation interval.
Min	The minimum value during each aggregation interval.
Max	The maximum value during each aggregation interval.
Interpolate	An interpolated value at the beginning of each aggregation interval.
Raw	Data straight from the historian. The aggregation interval is not used in this mode.

The timestamp of each aggregate value is the time at the beginning of the aggregation interval. For example, a one-hour average value with a time of 1:00 pm is the average over the interval from 1:00 pm through 2:00 pm.

About Pen Groups

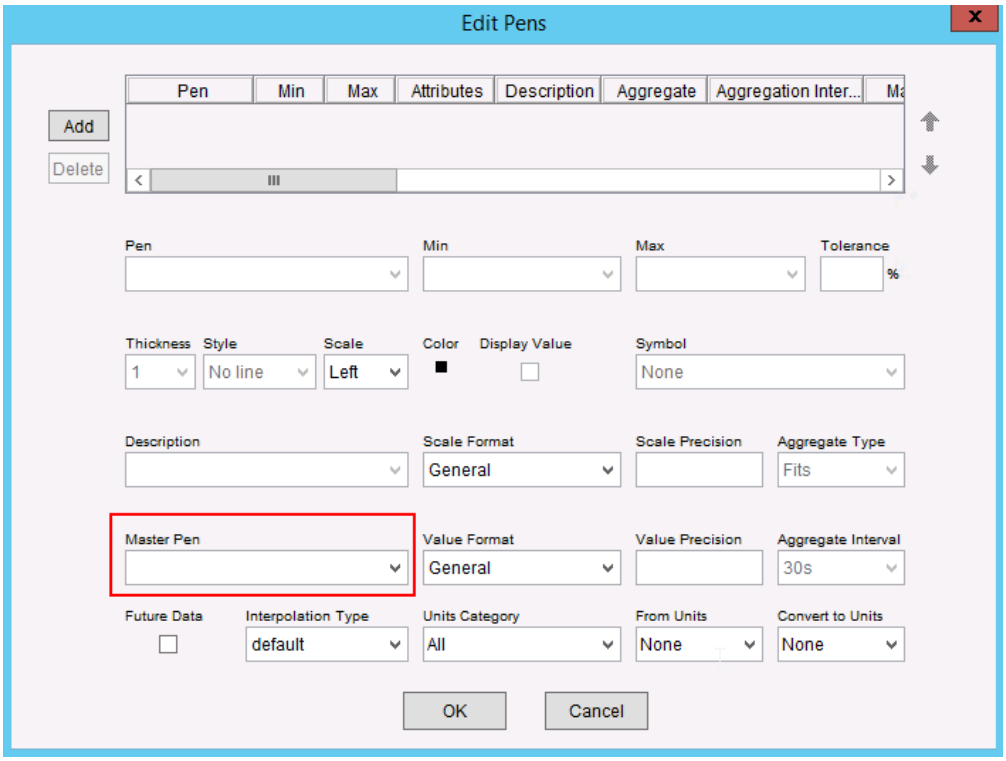
You can group pens that share common attributes. This saves you from having to display multiple pens that have the same configurations. Instead, you can display a single pen that represents the entire group.

In the XY-plot mode, the first pen on the pen value list is the X-value pen. The subsequent pen(s) are the Y-value(s). An important thing to note is that the X-value pen cannot be part of a pen group. It is invalid for the X-value pen to be either a master pen or a slave pen. The Master Pen selection list will not include the X-value pen when in the XY-plot mode.

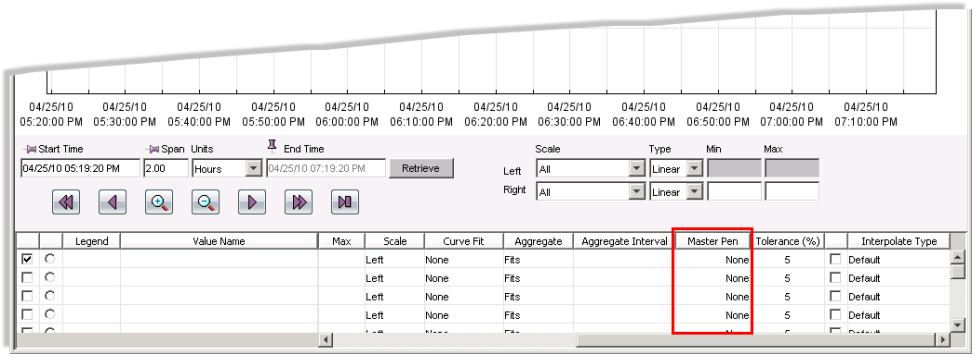
A pen can be a "master" pen, a "slave" pen, or a "standalone" pen. When you designate a pen as the master, its scale format, scale precision, value format, value precision, and tolerance is used for the entire pen group. A slave is a pen within a designated pen group that is not the master. And a standalone pen is a pen that does not belong to any group.

The master pen of one group cannot be a slave pen of another group. Likewise, a slave pen in one group cannot be a master pen in another group. In other words, a pen can join only one group, in which it is either the master pen or a slave pen.

There are two ways you can configure a pen group by assigning the master pen: during design-time through the XHQ Workbench Trend property, and during run-time through the XHQ Trend Viewer Legend panel. You can add or remove a pen from a group in either instance.



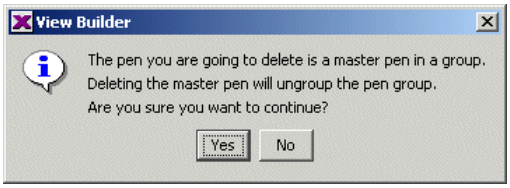
Pen Group Configuration



Pen Group Configuration via XHQ Trend Viewer Legend Panel

The Master Pen selection lists designated master pens, standalone pens, and the predefined item of "None". To assign a master pen, select a standalone pen from the list. To add a pen to a group (a slave pen), select a master pen from the list. To remove a pen from a group, select "None" as the Master Pen.

You delete a master pen the same way you would a standalone pen. In design-time, this is done in the "Edit Pens" dialog box. You simply select the pen from the table listing all available pens and then click the "Delete" button. A message dialog appears, asking you to confirm the delete action.



Message: Confirm Master Pen Deletion

Pen Groups and the Scale Property

If the Scale property for the trend is set to "All", then only the high and low limit of the pen group is displayed; the high and low limit of each pen within the group is not displayed. The high/low limit of the pen group is determined by the min and max values of the group, as well as the tolerance of the group.

Pen Groups and the Min/Max Properties

If the "Manual Min" and the "Manual Max" properties are configured for the master pen, then these values automatically become the min and max values for the entire pen group. If the "Manual Min" and "Manual Max" properties have not been defined for the master pen, then the min and max values are automatically calculated using the following formulas:

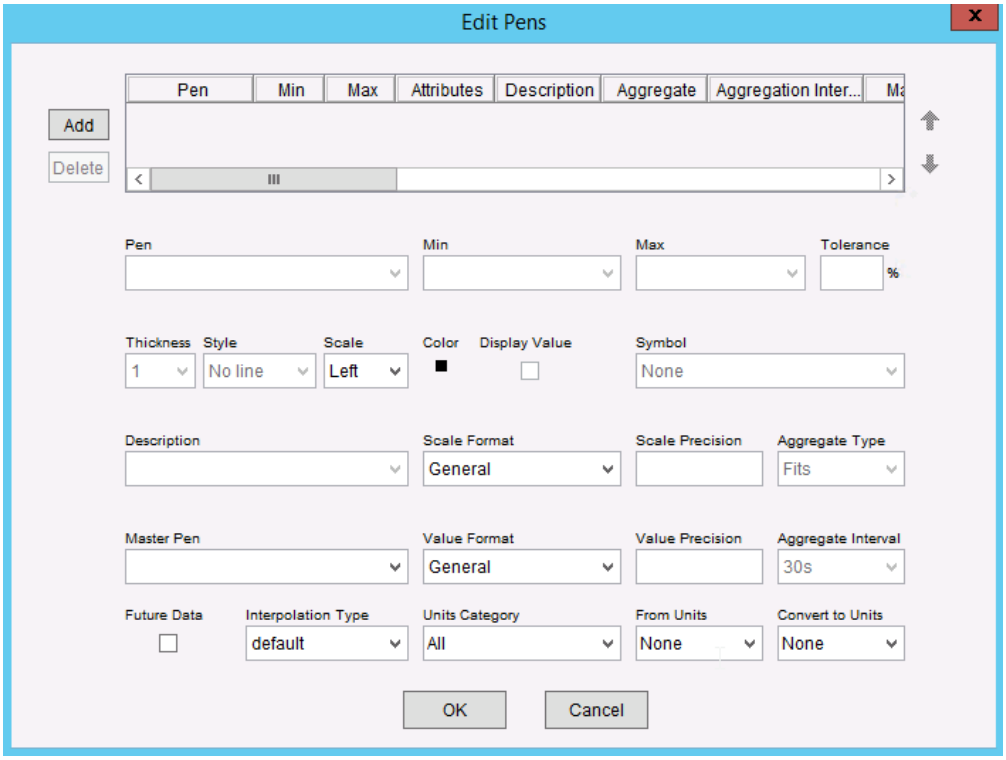
```
group_max = max(pen_1.max, pen_2.max, . . . , pen_n.max)
group_min = min(pen_1. min, pen_2. min, . . . , pen_n. min)
```

Where "pen_1, pen_2, . . . , pen_n" are the pens within the group.

To calculate the high and low limits for the pen group, the tolerance for the master pen is applied to the group_max and group_min values.

Pen Attributes: In Time Mode

To access the Pen Attributes dialog box, insert a trend, and, from the Trend properties, click inside the cell next to the Pen property.



The pen attributes you set for the embedded trend are automatically applied to the related Trend Viewer (Interactive Trender) during runtime.

Time Mode – Edit Pens Controls

Control	Description
Add	Click to add pen to list.
Delete	Click to delete pen from list.
<Value List>	Displays each pen configuration. **Important**: The first pen always represents the X-value, and the subsequent pen(s), the Y-value(s).
<Up Arrow, Down Arrow>	Allows you to set priority for the selected pen. **Tip**: To set the X-value, use the arrows to move a selected pen to the top-most pen location.
Pen	Select a pen name from the given drop-down list. You can also enter a constant or a string variable.
Min	Sets the minimum trend scale value. Select from the given drop-down list, or enter an expression (or constant) of the type Integer or Real.

Control	Description
Max	Sets the maximum trend scale value. Select from the given drop-down list, or enter an expression (or constant) of the type Integer or Real.
Tolerance	The *pen tolerance* is the percentage value applied to the maximum and/or minimum when calculating the high and/or low limit of each trend. The default tolerance value is 5%.
Thickness	Select a line thickness (in pixels).
Style	Select a line style. Choose from: No Line, Solid, Dashed, Dotted.
Scale	Display the pen in either the Left or the Right Y-axis.
Color	Specifies the line and symbol color. Launches the Color Selector.
<checkbox> **Display Value**	Enable (check) this option to render the trace visible on the trend chart.
Symbol	Displays a set of predefined symbols to associate with the given Y value.
Description	Enter a string expression, or select from the given list.
Scale Format	This displays the pen scale in either "General," "Number-Scientific," or "Number-Fixed" format. The default scale format is "General". **See Also:** For more information on these formats, go to the topic, *How "General," "Number-Scientific," and "Number-Fixed" Formats are Displayed*.
Scale Precision	If either "Number-Scientific" or "Number-Fixed" format is selected, the Scale Precision text box is enabled, allowing you to enter the number of digits to display to the right of the decimal point. The default scale precision is zero (0). **See Also:** For more information on these formats, go to the topic, *How "General," "Number-Scientific," and "Number-Fixed" Formats are Displayed*.
Aggregate Type	The aggregation types available are Average, Min, Max, Interpolate, Fits, and Raw. For time plot mode, the default is Fits. **See Also:** For more information on aggregate types, go to the topic, *About Aggregate Types (History Retrieval)*.
Master Pen	When you designate a pen as the master, its scale format, scale precision, value format, value precision, and tolerance is used for the entire pen group.
Value Format	This displays the pen value in either "General," "Number-Scientific," or "Number-Fixed" format. The default scale format is "General". **See Also:** For more information on these formats, go to the topic, *How "General," "Number-Scientific," and "Number-Fixed" Formats are Displayed*.
Value Precision	If either "Number-Scientific" or "Number-Fixed" format is selected, the Value Precision text box is enabled, allowing you to enter the number of

Control	Description
	digits to display to the right of the decimal point. The default value precision is zero (0). **See Also:** For more information on these formats, go to the topic, *How "General," "Number-Scientific," and "Number-Fixed" Formats are Displayed*.
Aggregation Interval	The *aggregation interval* is the interval over which the aggregate is computed. These intervals are defined in seconds, minutes, hours, days, or weeks. The default options are: 30s; 5m; 10m; 2h; 5d; 5w. **Note:** The aggregation interval is meaningful only for the Average, Min, Max, and Interpolate retrieval types. It has no affect on retrieval of Raw or Fits data.
<checkbox> **Future Data**	If checked, future data points are retrieved. If unchecked, only historical data points are retrieved.
Interpolation Type	• **Default** The interpolate type is determined by the backend data configuration. • **Linear** Uses a linear line to connect two data points. • **Stepped** Uses a stepped line to connect two data points. **Note:** The client trend configuration for an interpolation type has only visual impact on the data points connection. It does not change the interpolation type of the backend data source.
Unit Category	The Units of Measure category. **See Also:** Go to the topic, *About Units of Measure*, for details.
From Unit	The view default unit. **See Also:** Go to the topic, *About Units of Measure*, for details.
Convert to Unit	The selected unit. **See Also:** Go to the topic, *About Units of Measure*, for details.

To set the Mode trend property to Time

1. From the **Configuration Properties** tab, click **Trend** properties.
2. Scroll to find the **Mode** property and click inside the adjacent cell.
3. From the drop-down list box, select **Time**.
4. Click **OK**.

To add or edit pen attributes in time mode

1. From the **Configuration Properties** tab, click **Trend properties** and then click inside the cell next to the **Pen** property.
   The "Edit Pens" dialog box appears.

   Once the trend is added to the view, to access the "Edit Pens" dialog again, you simply double-click the trend control in the Edit panel. This directly launches the Edit dialog; you do not have to click the "Pen Items" property in the Configuration panel.

2. Click **Add**.
   This enables the fields (controls), except for the Scale and Value Precision fields (which are enabled when a Format of either "Number-Scientific" or "Number-Fixed" is selected).

   Once sixteen pens have been added (the maximum), the Add button becomes disabled. To re-enable the button, you must delete a pen.

3. From the **Pen** box, do one of the following:
   - Select a member from the drop-down list.

   or

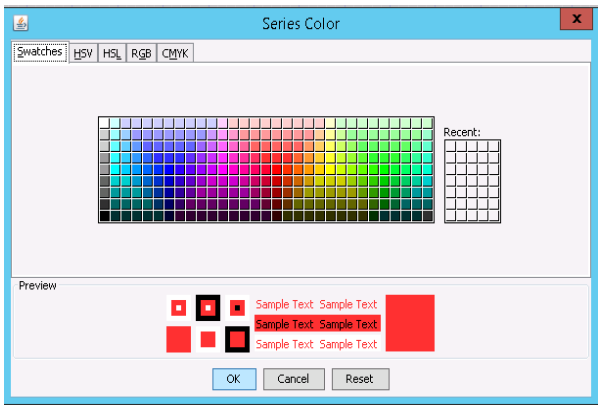
   - Enter a string variable or a constant.
4. From the **Min** box, do one of the following:
   - Select a minimum trend scale value from the drop-down list.

   or

   - Enter an expression (or a constant) of the type Integer or Real.
5. From the **Max** box, do one of the following:
   - Select a maximum trend scale value from the drop-down list.

   or

   - Enter an expression (or a constant) of the type Integer or Real.
6. Enter a pen **Tolerance** (a percentage).
   The *pen tolerance* is the percentage value applied to the maximum and/or minimum when calculating the high and/or low limit of each trend. The default tolerance value is 5%.
7. Select a line **Thickness**.
8. OPTIONAL
   Select a line **Style**.
9. To select a line **Color**, click on the color square.
   The "Color Selector" appears.



For more information on the color selector, refer to the topic, *Using the Color Selector.*

10. OPTIONAL
    Enable (check) **Display Value** to render the trace visible on the Trend Chart.
11. OPTIONAL
    Assign a **Symbol** to the trace.
12. For **Description**, do one of the following:
    - Enter a string expression.

    or

    - Select from the list.
13. Select a **Scale Format**.
    This displays the pen scale in either "General," "Number-Scientific," or "Number-Fixed" format. The default scale format is "General".
14. If either "Number-Scientific" or "Number-Fixed" format is selected, the **Scale Precision** text box is enabled, allowing you to enter the number of digits to display to the right of the decimal point. The default scale precision is zero (0).
15. Next, select an **Aggregate Type**.
    The default type is "Fits".

    For more information on aggregate types, go to the topic, *About Aggregate Types History Retrieval.*

16. OPTIONAL
    Select a **Master Pen**.

    For more information on master pens, go to the topic, *About Pen Groups*.

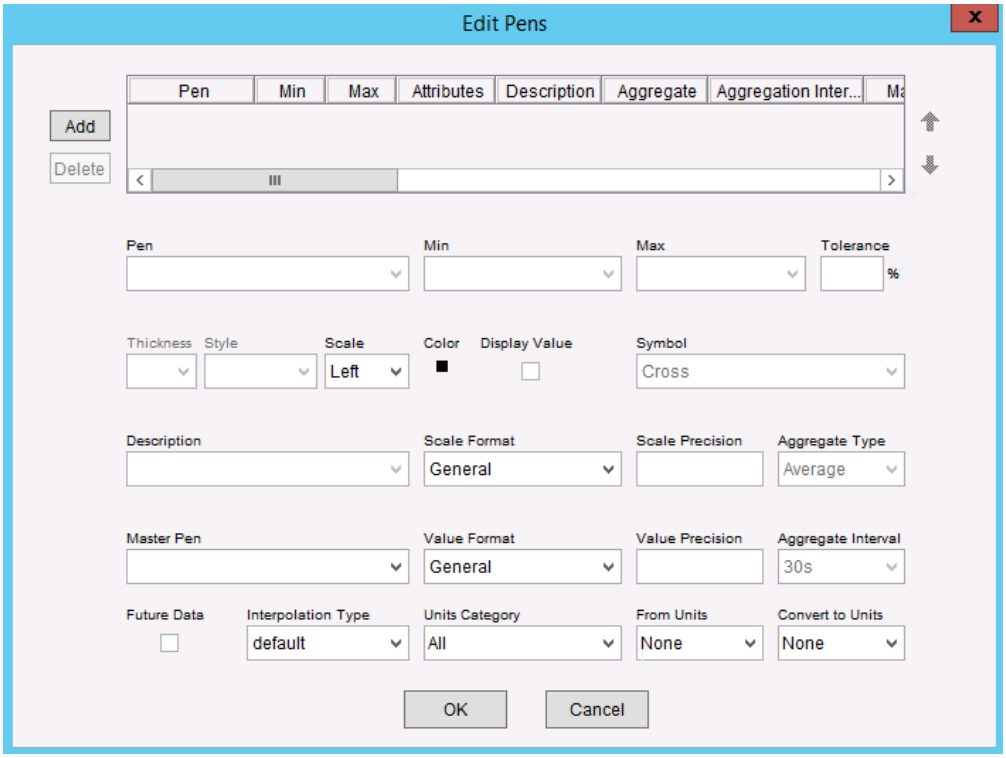
17. Select a **Value Format**.
    This displays the pen value in either "General," "Number-Scientific," or "Number-Fixed" format. The default scale format is "General".
18. If either "Number-Scientific" or "Number-Fixed" format is selected, the **Value Precision** text box is enabled, allowing you to enter the number of digits to display to the right of the decimal point. The default value precision is zero (0).
19. Click **OK**.
    The pen is added to the pen value table.

Pen Attributes: In XY-plot Mode

In XY mode, only symbols are displayed. The set of values plotted is determined by the time span you specify. They represent the interpolated values at the given aggregate interval you select.

The pen attributes you can set in XY mode are different to the ones you can set in Time mode.

To access the Pen Attributes dialog box, insert a trend, and, from the Trend tab, click the Edit button next to the pen values table.



XY-Plot Mode – Edit Pens dialog box

The pen attributes you set for the embedded trend are automatically applied to the related XHQ Trend Viewer (XHQ Interactive Trender) during runtime.

XY-Plot Mode – Edit Pens Controls

Control	Description
Add	Click to add pen to list.
Delete	Click to delete pen from list.
<Value List>	Displays each pen configuration. **Important**: The first pen always represents the X-value, and the subsequent pen(s), the Y-value(s).
<Up Arrow, Down Arrow>	Allows you to set priority for the selected pen.

Control	Description
	Tip: To set the X-value, use the arrows to move a selected pen to the top-most pen location.
Pen	Select a pen name from the given drop-down list. You can also enter a constant or a string variable.
Min	Sets the minimum Y value. Select from the list of integer or real primitive members and variables of the current XHQ view. Or, enter a valid numeric expression. **Note**: If this value is not set, then it defaults to the auto-scaling for the Y-axis. Otherwise, manual scaling is enabled.
Max	Sets the maximum Y value. Select from the list of integer or real primitive members and variables of the current XHQ view. Or, enter a valid numeric expression. **Note**: If this value is not set, then it defaults to the auto-scaling for the Y-axis. Otherwise, manual scaling is enabled.
Tolerance	The *pen tolerance* is the percentage value applied to the maximum and/or minimum when calculating the high and/or low limit of each trend. The default tolerance value is 5%.
<Thickness>	Disabled for XY-plot.
<Style>	Disabled for XY-plot.
Scale	Display the pen in either the Left or the Right Y-axis.
Color	Specifies the symbol color. Launches the Color Selector.
<checkbox> **Display Value**	Enable (check) this option to render the trace visible on the trend chart.
Symbol	Displays a set of predefined symbols to associate with the given Y value.
Description	Enter a string expression, or select from the given list.
Scale Format	This displays the pen scale in either "General," "Number-Scientific," or "Number-Fixed" format. The default scale format is "General".
Scale Precision	If either "Number-Scientific" or "Number-Fixed" format is selected, the Scale Precision text box is enabled, allowing you to enter the number of digits to display to the right of the decimal point. The default scale precision is zero (0).
Aggregate Type	The aggregation types available are Average, Min, Max, and Interpolate. For time plot mode, the default is Interpolate. **See Also:** For more information on aggregate types, go to the topic, *About Aggregate Types (History Retrieval)*.
<Master Pen>	In the XY-plot mode, the first pen on the pen value list is the X-value pen. The subsequent pen(s) are the Y-value(s). An important thing to note is that the X-value pen cannot be part of a pen group. It is invalid for the X-value pen to be either a master pen or a slave pen. The Master

Control	Description
	Pen selection list will not include the X-value pen when in the XY-plot mode. **See Also:** For more information on why this pen attribute is disabled, go to the topic, *Pen Groups and the XY plot Mode*.
Value Format	This displays the pen value in either "General," "Number-Scientific," or "Number-Fixed" format. The default scale format is "General".
Value Precision	If either "Number-Scientific" or "Number-Fixed" format is selected, the Value Precision text box is enabled, allowing you to enter the number of digits to display to the right of the decimal point. The default value precision is zero (0).
Aggregate Interval	The *aggregation interval* is the interval over which the aggregate is computed. These intervals are defined in seconds, minutes, hours, days, or weeks. The default options are: 30s; 5m; 10m; 2h; 5d; 5w. **Note:** The aggregation interval is meaningful only for the Average, Min, Max, and Interpolate retrieval types. It has no affect on retrieval of Raw or Fits data.
<Future Data>	Disabled for XY-plot.
Interpolate Type	• **Default** The interpolate type is determined by the backend data configuration. • **Linear** Uses a linear line to connect two data points. • **Stepped** Uses a stepped line to connect two data points. **Note:** The client trend configuration for an interpolation type has only visual impact on the data points connection. It does not change the interpolation type of the backend data source.
Unit Category	The Units of Measure category. **See Also:** Go to the topic, *About Units of Measure*, for details.
From Unit	The view default unit. **See Also:** Go to the topic, *About Units of Measure*, for details.
Convert to Unit	The selected unit. **See Also:** Go to the topic, *About Units of Measure*, for details.

Pen Groups and the XY-plot Mode

In the XY-plot mode, the first pen on the pen value list is the X-value pen. The subsequent pen(s) are the Y-value(s). An important thing to note is that the X-value pen cannot be part of a pen group. It is invalid for the X-value pen to be either a master pen or a slave pen. The Master Pen selection list will not include the X-value pen when in the XY-plot mode.

There is an instance when switching from Time mode to XY-plot mode that a master pen or a slave pen arbitrarily becomes the X-value pen. As a result, the master or slave pen then reverts back to standalone status and will remain so even if you switch back to Time mode. In the master pen situation, all slave pens within the group also become standalone pens.

To set the Mode trend property to XY-plot

1. From the **Configuration Properties** tab, click **Trend** properties.
2. Scroll to find the **Mode** property and click inside the adjacent cell.
3. From the drop-down list box, select **X-Y**.
4. Click **OK**.

To add or edit pen attributes in XY-plot mode

1. From the **Configuration Properties** tab, click **Trend properties** and then click inside the cell next to the **Pen** property.
   The "Edit Pens" dialog box appears.

   Once the trend is added to the view, to access the "Edit Pens" dialog again, you simply double-click the trend control in the Edit panel. This directly launches the Edit dialog; you do not have to click the "Pen Items" property in the Configuration panel.

2. Click **Add**.
   This enables the fields (controls), except for the Thickness, Style, Master Pen, and Scale and Value Precision fields (which are enabled when a Format of either "Number-Scientific" or "Number-Fixed" is selected.)

   Once sixteen pens have been added (the maximum), the Add button becomes disabled. To re-enable the button, you must delete a pen.

   The first pen listed always represents the X-value, and the subsequent pen(s), the Y-value(s). To set the X-value, use the arrows to move a selected pen to the top-most pen location.

3. From the **Pen** box, do one of the following:
   - Select a member from the drop-down list.

   or

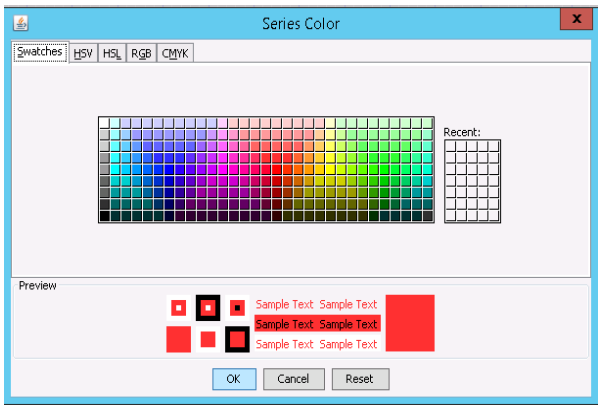
   - Enter a string variable or a constant.
4. For the **Min** box, do one of the following:
   - Enter a valid numeric expression.

   or

   - Select from the list of integer or real primitive members and variables of the current XHQ view.
5. For the **Max** box, do one of the following:
   - Enter a valid numeric expression.

   or

   - Select from the list of integer or real primitive members and variables of the current XHQ view.
6. Enter a pen **Tolerance** (a percentage).
   The *pen tolerance* is the percentage value applied to the maximum and/or minimum when calculating the high and/or low limit of each trend. The default tolerance value is 5%
7. OPTIONAL
   To select a symbol **Color**, click on the color square.
   The "Color Selector" appears.



For more information on the color selector, refer to the topic, *Using the Color Selector.*

8. Enable (check) **Display Value** to render the trace visible on the Trend Chart.
9. Next, select a **Symbol** to represent the Y-values.
10. For **Description**, do one of the following:
    - Enter a string expression.

    or

    - Select from the given list.
11. Select a **Scale Format**.
    This displays the pen scale in either "General," "Number-Scientific," or "Number-Fixed" format. The default scale format is "General".
12. If either "Number-Scientific" or "Number-Fixed" format is selected, the **Scale Precision** text box is enabled, allowing you to enter the number of digits to display to the right of the decimal point. The default scale precision is zero (0).
13. Select an **Aggregation Type**.
    For time-sliced retrieval, the aggregation types available are Average, Min, Max, and Interpolate. The default type is "Interpolate".

    For more information on aggregate types, go to the topic, *About Aggregate Types History Retrieval.*

14. Select a **Value Format**.
    This displays the pen value in either "General," "Number-Scientific," or "Number-Fixed" format. The default scale format is "General".
15. If either "Number-Scientific" or "Number-Fixed" format is selected, the **Value Precision** text box is enabled, allowing you to enter the number of digits to display to the right of the decimal point. The default value precision is zero (0).
16. Click **OK**.
    The pen is added to the pen value table.

Editing and Deleting Pen Attributes

Once the pen attributes have been set, they are then displayed in the pen value table of the Trend tab.

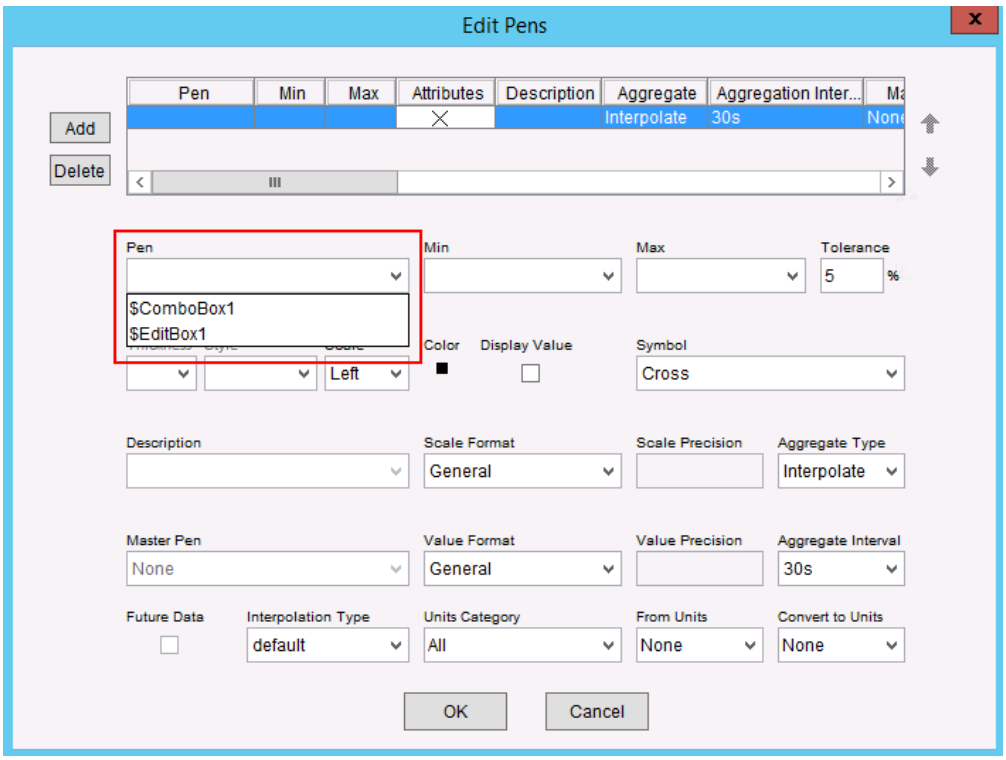
To edit a pen attribute, click the "Edit" button next to the pen value table. In the "Edit Pens" dialog box, select (highlight) the pen from the table and make your changes.

To delete a pen attribute, again, open the "Edit Pens" dialog box. From the pen value table, select the pen and click "Delete".

Using Expressions in the Embedded Trender

Pens for the embedded Trender support the following:

- Expressions with single or multiple references;
- Global tag references.



You can use any function and any operator supported in an expression.

For more information on supported functions and operators, go to the *XHQ Reference Guide*.

For more information expressions in the XHQ Interactive Trender, go to the *XHQ Trend Viewer User's Guide*.

EXAMPLES: WHEN TO USE EXPRESSIONS IN AN EMBEDDED TRENDER

Example 1:

You have a global view that includes process graphics and global tag references. You embed a trend within this global view and use the global tag references in a pen expression.

Example 2:

You have a reusable view with embedded trends; the view is used against multiple object instances. In this case, you would use pen expressions with component member references.

Valid Pen Expressions

The following are examples of valid pen expressions for the **Embedded Trender**.

Valid Pen Expressions

Pen Expressions can have . . .	Example
Multiple string member and string variable references	`$TagnamePrefix + TagName + $TagnamePostfix` Where: `TagName` is a String-type component member; `$TagnamePrefix` and `$TagnamePostfix` are String-type view variables.
A single numeric member reference	`rValue*2` Where `rValue` is a Float-type component member.
Multiple numeric member references	`(rValue1 + rValue2) / 2` Where `rValue1` and `rValue2` are Float-type component members. **Important**: Only the following time-sliced aggregate types are valid for the pen expression: Interpolate, Average, Min, and Max. If the aggregation type is not valid, then the Interpolate type is used by default.
One or more global tag references **See Also**: *About Global Tag References*	`(r:xhqreal0301 + r:'11FC0123') / 2` Where `xhqreal0301` and `11FC0123` are global tags that have Float-type values. **Note**: Global tag references are simply alias-type identifiers. Therefore, the same rules for aliases apply.
One or more late-binding global tag references **See Also**: *About Late binding Global Tag References*	`referenceToReal($EditBox1) + referenceToReal($EditBox2)` Where $EditBox1 and $EditBox2 are String-type control variables.
Both numeric member references and global tag references	`(rValue + r:xhqreal0301) / 2` Where `rValue` is a Float-type member, and `xhqreal0301` is a global tag. **Important**: Only the following time-sliced aggregate types are valid for the pen expression: Interpolate, Average, Min, and Max. If the aggregation type is not

Pen Expressions can have . . .	Example
	valid, then the Interpolate type is used by default.
View variable references	`rValue*$editBox` Where rValue is a Float-type member and $editBox is a Float-type view variable.

About the ReferenceTo Function Limitation

The `ReferenceTo` function (used in the embedded trend) does not handle member variables. Consider the following examples with member variables:

- referenceToReal(sTagname1)
- referenceToReal(sTagname1) + referenceToReal(sTagname2)

In both cases, the trend does not display.

The workaround for this issues is to use **single quote** references instead. In fact, you can refrain from using the ReferenceTo function in the embedded trend altogether.

Instead of this:	Use this:
`referenceToReal("13AR334") + referenceToReal("13AR334")`	`"'13AR334'+'13AR334'"`
`referenceToReal($Value1) + referenceToReal ($Value2)`	`"'" + $Value1 + "'" +"+'"+ $Value2 + "'"`
`referenceToReal(sTagname1)`	`sTagname1`
`referenceToReal(sTagname1) + referenceToReal(sTagname2)`	`"'" + sTagname1+ "'" +"+'"+ sTagname2+ "'"`

About Global Tag References

A pen expression can contain one or more global tag references. Global tag references are simply alias-type identifiers, and so, the same rules for aliases apply. The global tag reference(s) must use the following format:

```
GlobalTagReference ::= Cast:SimpleIdentifier | Cast:SingleQuotedSimpleIdentifier
```

For example, a **`Cast`** prefix (see format example above) must be used when entering the tag reference. (The cast defines the data type of the reference.) The Cast prefix is followed by a colon, which in turn is followed by the **`SimpleIdentifier`** (or tag name). The following casts are allowed:

- `b (Boolean)`
- `d (Double)`
- `i (Integer)`
- `l (Long)`
- `m (Decimal)`
- `r (Real)`

The **`SimpleIdentifier`**, or tag name, must be enclosed in single (or back) quotes if ANY of the following is **true**:

- Does not begin with a letter followed by either a zero, letters, or numbers
- Includes spaces, localized strings, and/or special characters (such as '#', '@', '%', '$', '&', '-', and so forth).

In general, best practice is to always enclose a tag name within single quotes.

Examples	Comments
`r:xhqreal0301`	This is a **VALID** global tag reference. A Cast prefix is defined, followed by a colon and then the SimpleIdentifier, which begins with a letter.
`r:11FC0123`	This is an **INVALID** global tag reference. The SimpleIdentifier begins with a number and not a letter.
`r:'11FC0123'`	This is a **VALID** global tag reference. A Cast prefix is defined, followed by a colon and then the SimpleIdentifier, which is enclosed in single quotes because it does not begin with a letter.
`r:'$FC + TagName + 001'`	This is **VALID** global tag reference. A Cast prefix is defined, followed by a colon and then the SimpleIdentifier, which is enclosed in single quotes because it contains spaces and special characters.

About Late-binding Global Tag References

To use late-binding global tag references in a pen expression, functions are applied. In design time, these functions behave like the cast used by a static tag reference to indicate the data type of the reference. At runtime, the argument, which is a String-type expression, is resolved to an alias (or tag name).

- `referenceToBoolean`
- `referenceToDouble`
- `referenceToInteger`
- `referenceToLong`
- `referenceToDecimal`
- `referenceToReal`

These functions are to be used only in pen expressions. Although these functions are similar to the `subscibeTo` functions, they do not subscribe to the references.

For more information, go to the topic, *Support for Late binding in an Embedded Trend*.

Processing a String Pen Expression as a Numeric Equation

XHQ processes a pen expression of the type String one of two ways:

- First, if the pen expression value is a **valid tag name**, it processes the value as a **global tag reference**.
- However, if the value is an **invalid tag name**, then it processes the pen expression as a **numeric equation**.

When a String-typed pen expression is processed as a numeric equation, the same processing rules for the Interactive Trender apply.

Example: The value entered for a String-typed pen expression is **`xhqreal30 + 1`**.

First, XHQ verifies if this value is a valid tag name. If it is, then the value is processed as a global tag reference. If not, then it is processed as a numeric equation.

Support for Late-binding in an Embedded Trend

For late-binding, a pen is defined with either a string variable or a member that is of the type String. At runtime, the "value" of the string expression is evaluated.

Fore more information, refer to the topic, *Late-binding Expressions*, located in the XHQ Reference Guide.

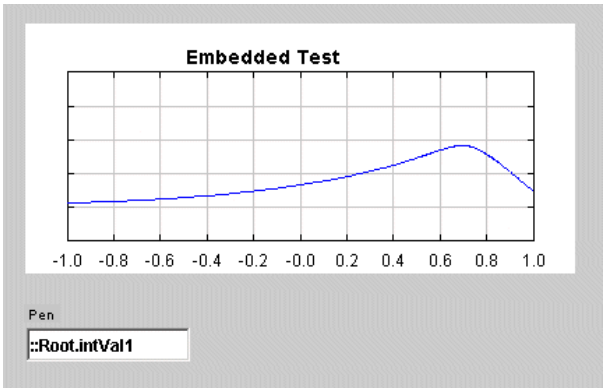
Consider the following example.

From the **Edit Pens** dialog box, you define a pen using a string variable:

At runtime, you have the embedded trend and a related text box. In the text box, you enter:

```
::Root.intVal1
```

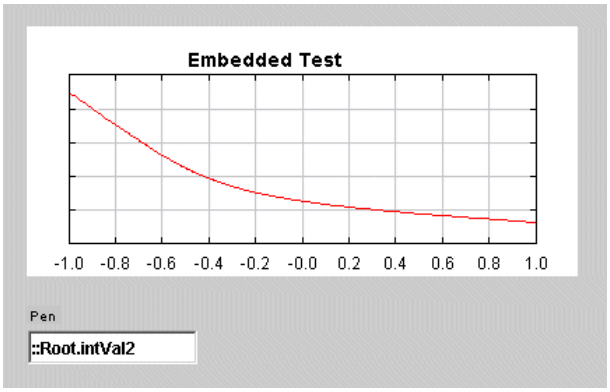
and press **Enter**. The embedded trend dynamically resolves the reference and plots the data.



In the same text box, you enter:

```
::Root.intVal2
```

and press **Enter**. The trend plot changes, using the newly evaluated string data as the new reference.



Examples of Embedded Trends in Runtime

Example 1:

Figures 1 and 2: Trend and Pen configurations during design time.

Figure 3: Corresponding embedded trend during runtime.

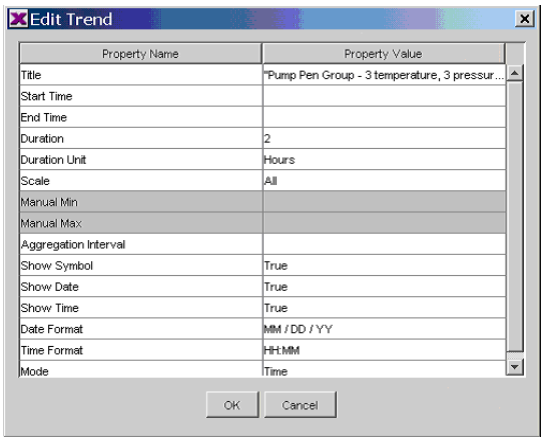


Figure 1

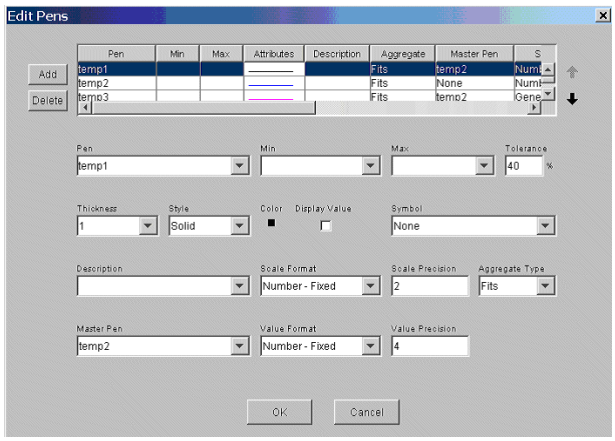


Figure 2

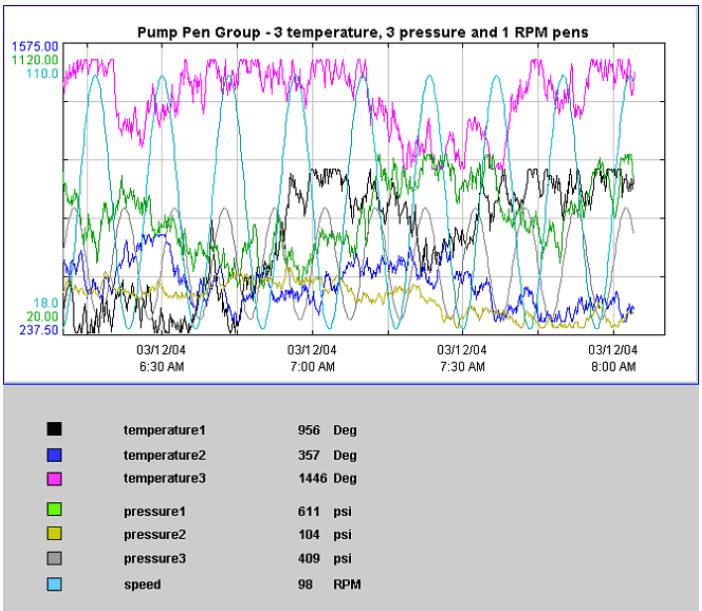


Figure 3

Example 2

Figure 1: Trend configuration during design time.

Figure 2: Corresponding embedded trend during runtime.

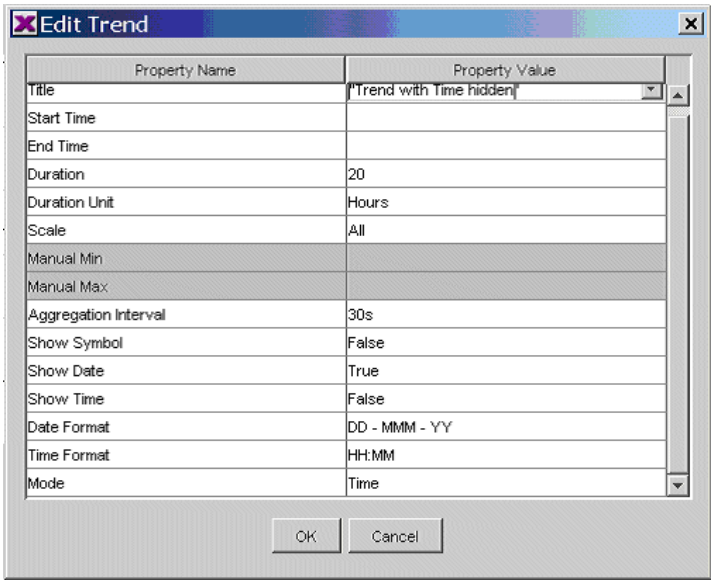


Figure 1 – Edit Trend dialog box

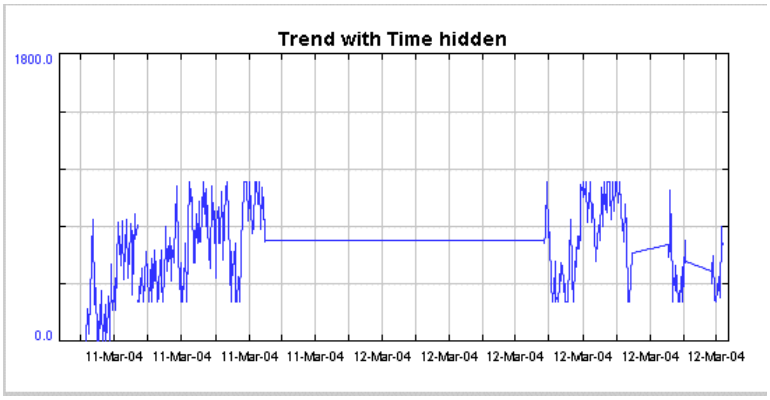
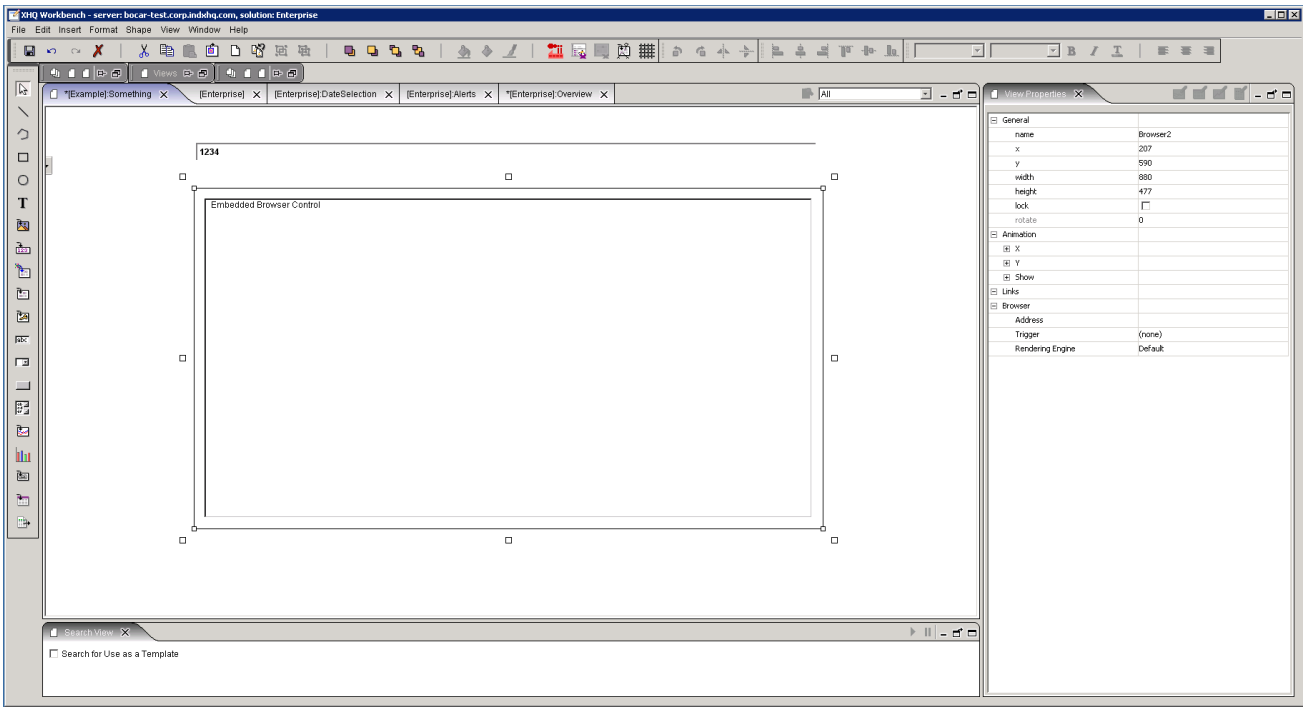


Figure 2 – Embedded Trend during runtime

Because embedded trends deal with only point data, expressions are not supported.

Embedding Browser Content

With the Browser tool, you can embed a web browser control that displays any web page directly into a view.



Browser Tool

Each embedded browser control launches a new instance of the browser. Therefore, multiple browser controls in a single view may have an adverse effect on performance. The recommended maximum number of controls per view is 3.

Embedding the XHQ Solution Viewer

The web control for HTML5 requires the embedded page to have embedding enabled where the X-Frame-Options is not set to SAMEORIGIN and the Content-Security-Policy header is set on `customHeaders.js`, accordingly.

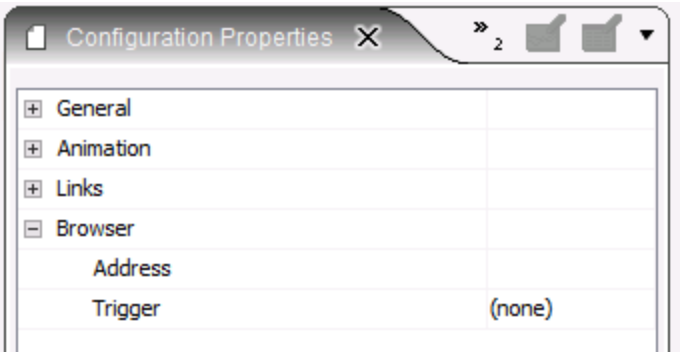
In the following example, the Content-Security-Policy header is updated in order to embed an image from the `api.weather.gov` site:

```
"Content-Security-Policy" : "default-src 'self'; script-src
https://maps.googleapis.com https://dev.virtualearth.net 'self'
'unsafe-inline' 'unsafe-eval'; img-src * 'unsafe-inline' data:;
style-src https://fonts.googleapis.com 'self' 'unsafe-inline';
font-src https://fonts.googleapis.com https://fonts.gstatic.com
'self' 'unsafe-inline' data:; worker-src blob: 'self' 'unsafe-
inline' 'unsafe-eval'; child-src https://api.weather.gov;"
```

For more information, see the topic, *About customHeaders.json*, located in the XHQ Administrator's Guide.

To insert a web browser control

1. From the **Tool palette**, click the **Browser** tool .
2. Click in the drawing area and **drag diagonally** to draw the control, and release the mouse when the image reaches the desired size.
3. Place the browser control anywhere on the drawing area.
   The "Browser" property appears on the Configuration Properties tab.



4. For the Browser property, do one of the following:
   - In the **Address** box, enter a URL.
   - Or, to configure a URL as an XHQ expression, click **Edit** to enter an expression.
5. OPTIONAL
   In the **Trigger** box, enter a control button ID.

   Example: For Trigger, you enter `$button1`. So, at runtime, the browser control loads the content only when Button1 is clicked.

   If the Trigger option is not enabled, then the browser view updates automatically when the view is displayed (that is, when the URL expression changes).
6. **Save** the view.

Invoking the JavaScript Function

The browser control `eval` function can be used to invoke the JavaScript function in the page contained within the browser control from an XHQ view.

For details, go to the topic, *JavaScript Call Function*, located in the XHQ Reference Guide.

Removing the Browser Control Borders and Scrollbars

To remove the borders and scrollbars, you must have edit rights to the HTML page you want to display in the browser control. Set the `style` attribute of the `<html>` and `<body>` tags to **`style="overflow: hidden; border-sytle: none;"`**.

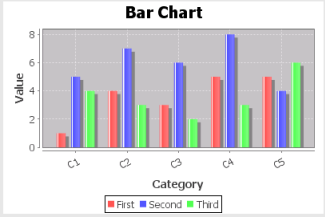
Consider the following example:

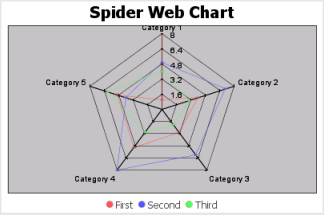
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html style="overflow: hidden; border-sytle: none;">
<head>
```

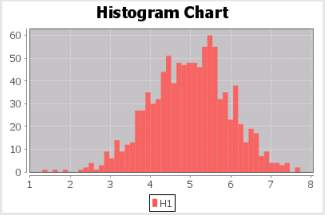
```
    <title>test</title>
</head>
<body style="overflow: hidden; border-style: none;">
Hello World!
</body>
</html>
```

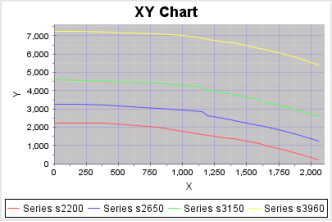
Using XHQ Advanced Charts

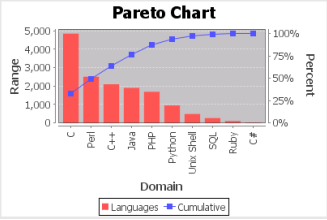
XHQ Advanced Charting enables you to build dynamic charts using data from tags, aliases and members, and XHQ collections. Currently, the following charts are supported:

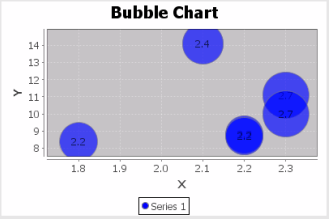


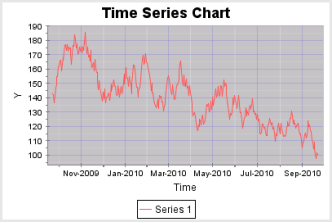


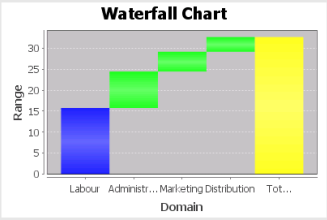


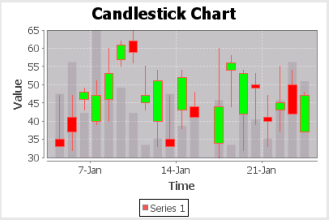


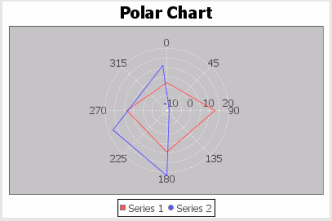


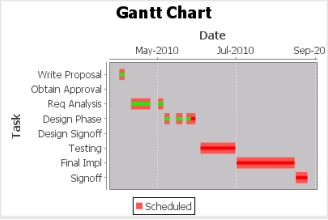


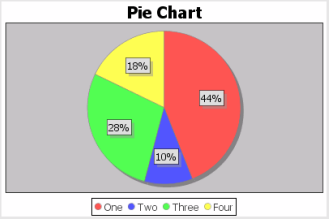












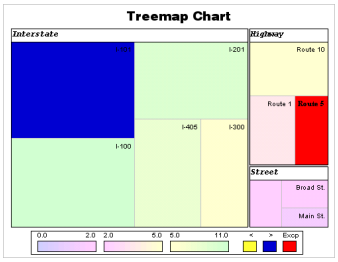


Chart Features

The following table lists which features are configurable for each chart.

Chart Features

	Bar	Bubble	Candlestick	Gantt	Histogram	Pareto	Pie	Polar	Spider web	Time Series	Treemap	Waterfall	XY
Alignment	X												X
Allow Interaction	X	X	X	X	X	X	X	X	X	X	X	X	X
Annotation	X	X	X	X		X				X		X	X
Chart Background Color	X	X	X	X	X	X	X	X	X	X	X	X	X
Collection	X	X	X	X	X	X	X	X	X	X	X	X	X
Combine Domain										X			X
Domain Axis Label	X	X	X	X	X	X				X		X	X
Domain Axis Value Format	X*	X	X	X	X					X			X
Domain Crosshair										X			X
Domain Min/Max Value	X*	X			X								X
Drill-down	X	X	X	X		X	X	X	X	X	X	X	X
Grid Lines	X	X	X	X	X	X		X		X		X	X
Plot Background Color	X	X	X	X	X	X	X	X	X	X	X	X	X
Range Axis	X	X	X	X	X	X				X		X	X

	Bar	Bubble	Candlestick	Gantt	Histogram	Pareto	Pie	Polar	Spider web	Time Series	Treemap	Waterfall	XY
Label													
Range Axis Location	X	X	X	X	X					X		X	X
Range Axis Value Format	X	X	X		X	X				X		X	X
Show Item Label	X	X		X		X	X					X	
Show Legend	X	X	X	X	X	X	X	X	X	X	X	X	X
Show Plot Border	X	X	X	X	X	X	X	X	X	X	X	X	X
Subtitle	X	X	X	X	X	X	X	X	X	X		X	X
Tag										X			
Tag Mapping													X
Time Offset										X			
Title	X	X	X	X	X	X	X	X	X	X		X	X
Units of Measure										X			

** Only for domain/range mapping type.*

For additional information on charting data sets and mapping, go to the topic, *Chart Mapping and Data Sets*.

The 3D Charts support the following features.

3D Chart Features

Feature	Line 3D	Bar 3D	Area 3D	Pie 3D	Stacked Bar 3D	Scatter Plot 3D
Allow Interaction	X	X	X	X	X	X
Axis Label (Value, Column, Row)	X	X	X		X	
Axis Label (X,Y,Z)						X
Axis Value Format (Value)	X	X	X		X	
Axis Value Format (X,Y,Z)						X
Chart Background Color	X	X	X	X	X	X
Collection	X	X	X	X	X	X
Grid Lines (Value, Column, Row, and X,Y,Z)	X	X	X		X	X
Min/Max (Value)	X	X	X		X	
Min/Max (X,Y,Z)						X
Plot Background Color	X	X	X	X	X	X
Show Legend	X	X	X	X	X	X
Show Zoom Toolbar	X	X	X	X	X	X
Subtitle	X	X	X	X	X	X

Feature	Line 3D	Bar 3D	Area 3D	Pie 3D	Stacked Bar 3D	Scatter Plot 3D
Title	X	X	X	X	X	X
View Point	X	X	X	X	X	X
View Distance	X	X	X	X	X	X

The Chart Tab

Using the Chart tool , click in the drawing area and **drag diagonally** to draw the chart. Release the mouse when the image reaches the desired size. Click the **Chart properties**.

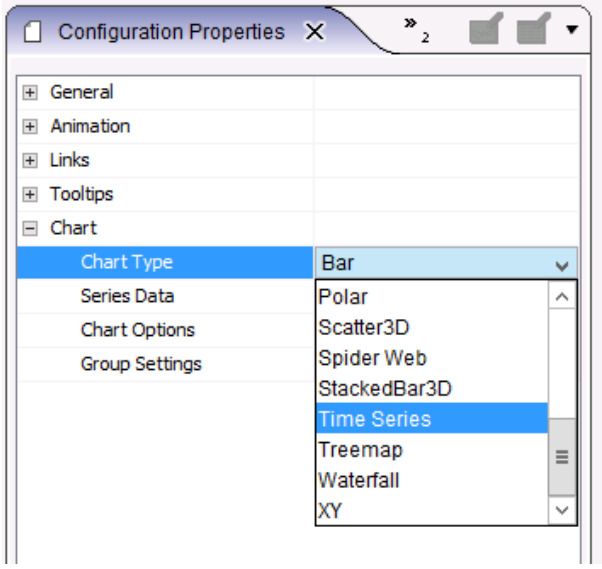


Chart Properties

Select a chart type, then add a chart series. Depending on the chart type you select, your next step is to configure chart-specific options:

- **Chart Options**
  Define the chart title, background color, Domain axis values, and plot information.
- **Group Settings**
  Set the renderers and Range axis values for a group of series.

The table below lists the option(s) available for each chart type.

Configurable Options for each Chart Type

Chart Type	Chart Options	Group Settings
Bar	X	X
Bubble	X	X
Candlestick	X	X
Gantt	X	X
Histogram	X	X
Pareto	X	X
Pie	X	
Polar	X	

Chart Type	Chart Options	Group Settings
Spider Web	X	
Time Series	X	X
Treemap	X	
Waterfall	X	X
XY	X	X

Configuring a Chart

Configuration consists of several steps:

- Selecting a Chart Type
- **Adding** a chart series.
- Setting **Chart Options**.
- Defining the **Group Settings**.

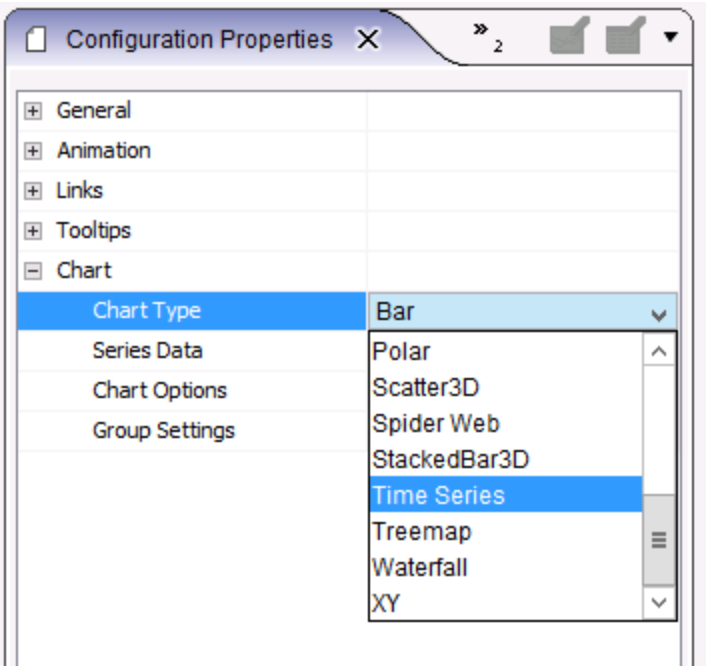
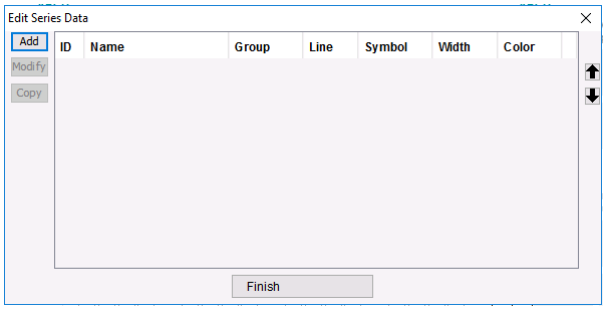


Chart Properties

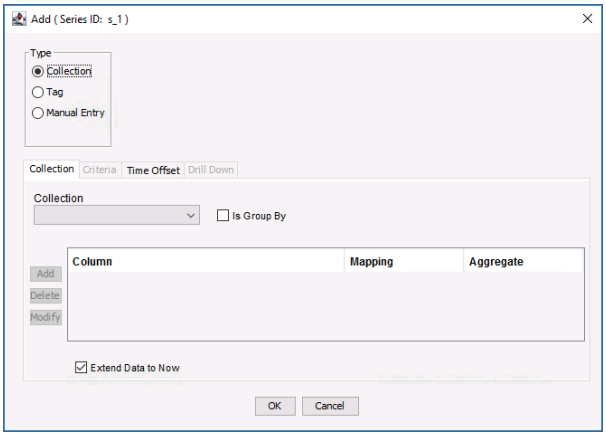
To add a series

1. From the **Configuration Properties panel**, expand the **Chart properties**, select a chart type and then click inside the cell next to **Series Data**.
   The "Edit Series Data" dialog box for that chart type appears.



Once the chart data is set, to access the "Edit Series Data" dialog again, you simply double-click the chart control in the Edit panel. This directly launches the Edit dialog; you do not have to click the "Series Data" property in the Configuration panel.

2. Click **Add**.
   The "Add" dialog box for that chart type appears.



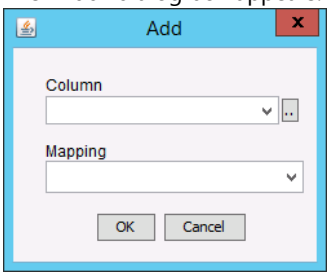
3. Depending on the chart type, select one of the following **Types**:
   - Collection
   - Manual Entry
   - Tag (for Time Series chart only)
   - Tag Mapping (for XY chart only)

Collection type

a. For **Type**, click **Collection**.

b. In the **Collection tab**, select a **Collection** from the drop-down list.
   The list of available columns appears in the "Columns" box.

c. Click **Add**.
   The "Add" dialog box appears.



d. For **Column**, click the **down arrow** and select a column *OR* click the **Browse button** to enter an **expression**.

   Your next step is to map the columns based on the chart type you selected (for example, Time Series).

   For a list of data mapping rules for each chart type, go to the topic, *Chart Mapping and Data Sets*.

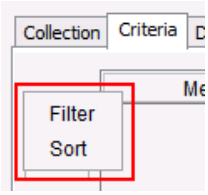
e. For **Mapping**, click the down arrow and select an option for the given chart type.

Example: For the Time Series chart, the available mapping options are Time, Value, Annotation, Tooltip, and Color.

f. Click **OK**.

g. OPTIONAL
Repeat the preceding steps to add more columns.

h. OPTIONAL
To apply a constraint to filter rows before the data set is retrieved, click the **Criteria tab** and select a criteria type: **Filter** or **Sort**.



For more information, go to the topic *Setting Criteria*.

i. OPTIONAL
Click the **Time Offset tab** and select one of the following:

- None
- Golden Run
- Compared Run

For more information, go to the topic, *Time Offset Types*.

j. OPTIONAL
Click the Drill Down tab and select one of the following:

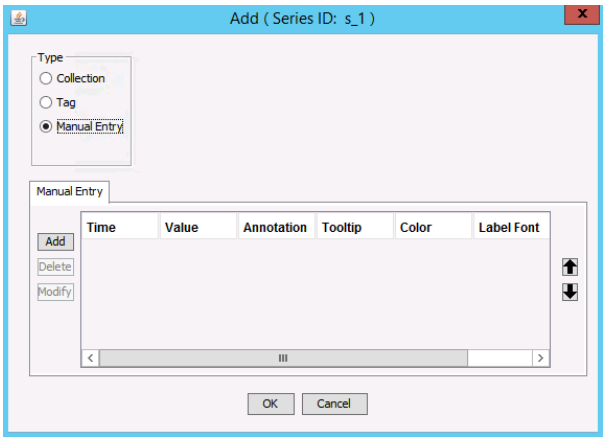
- Do Nothing
- Update Dependent Items
- Links

k. Click **OK** and continue to step 4.

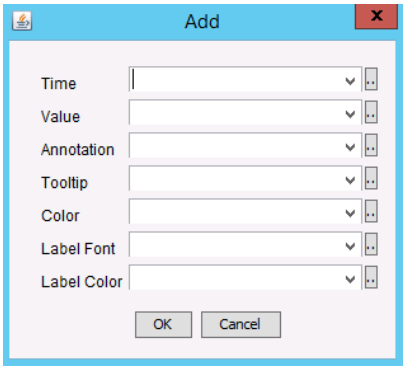
Manual Entry type

Manual Entry allows you to directly enter data when configuring a chart instead of retrieving it from a collection or a tag. As such, it is meant to be used when a *small amount of data* is needed to be *quickly entered* without having anything configured in the backend. Manual entry of data for a series is available for **all** charts, including 3D charts. Historical Navigation works with manual entry mappings that involve a time component. And all mapping fields support expression entries.

a. For **Type**, click **Manual Entry**.



b. From the **Manual Entry tab**, click **Add** to add a row of data.
The "Add" dialog box appears.



This dialog lists all the **possible mapping columns** available for the given **chart type**. The example dialog image above is for a Spider Web chart.

Mapping fields support **expression** entries. The context is the view itself.

The **data types** that are compatible for each mapping are as follows.

Mapping Type	Numeric	String	Time
Category		X	
Tooltip		X	
Color	X	X	
Key		X	
Annotation		X	
Item		X	
Task Name		X	
Sub Task		X	
Value	X		
Domain	X		

Mapping Type	Numeric	String	Time
Range	X		
X	X		
Y	X		
Z	X		
Box Size	X		
Shade	X		
Exception	X		
Percent Complete	X		
High	X		
Low	X		
Open	X		
Close	X		
Volume	X		
Parent Category		X	X
Time			X
Start Date			X
End Date			X
Label Font		X	
Label Color	X	X	

c. For each **mapping type**, do one of the following:

- Click the Down arrow to select a value.
- Or, click the **Browse button** to enter an **expression**.

These are some **examples of expressions** you can use.

For mappings that support the **Numeric** data type, such as *Value* or *Domain*:

r:'PT001'

r:PT001

r:PT001 * 2

r:PT001 + r:PT002

$EditBox1

$Value

400 * 2

For mappings that support the **String** data type, such as *Category* or *Tooltip*:

"Category1"

$EditBox2

For mappings that support the **Time** data type, such as *Time*:

DateValue("01/Apr/15 12:00:00 AM")

As stated earlier, **Historical Navigation** works with manual entry mappings that involve a time component. For example, tag "r:PT001" is configured as part of a mapping and historical navigation is enabled. So, when navigating to time period t1, the value at time t1 for tag "r:PT001" is retrieved and automatically updated in the chart.

For more information on expressions, go to the topic, *Using Expressions in Chart Configurations*.

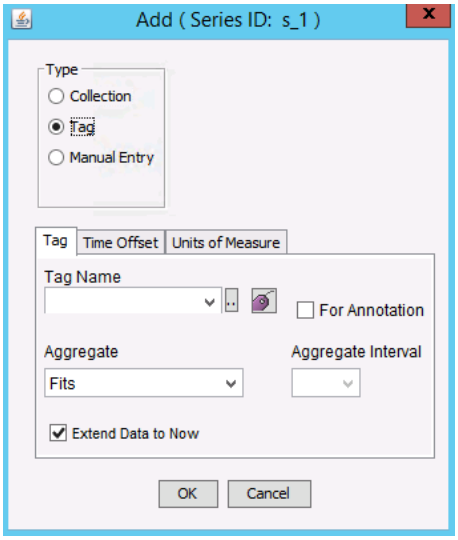
d. Click **OK**.
The mapping columns you added appear in the Manual Entry table.

e. OPTIONAL
Click the Up/Down arrows to rearrange the rows.

f. Click **OK** and continue to step 4.

Tag type (for Time Series Chart only)

a. For **Type**, click **Tag**.



b. For the **Tag Name**, do one of the following:

- **Enter** a valid Tag name.
- Click the **Browse button** to enter an **expression**.

For information, go to the topic, *Using Expressions in Chart Configurations*.

- Click the **Tag Picker button** to select a tag from the **Tag Selector**.

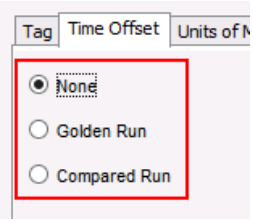
c. OPTIONAL
Check **Annotation** to use the tag for annotation.

d. Select the **Aggregate** type.

e. For time-sliced aggregates (for example, the Average aggregate), select an **Aggregate Interval**.

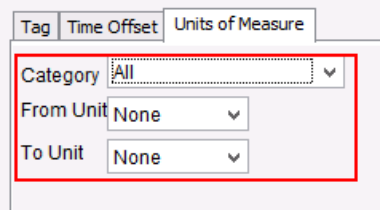
f. OPTIONAL
Check **Extend Data to Now**.

g. Click the **Time Offset tab** and select the offset option.



For more information, go to the topic, *Time Offset Types*.

h. Click the **Unit of Measure tab**.



i. Select a **Category**, the **From Unit**, and the **To Unit**.

For more information, go to the topic, *Setting the Units of Measure*.

j. Click **OK** and continue to step 4.

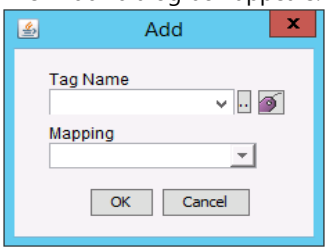
Tag Mapping type (for XY Chart only)

Tag Mapping allows you to plot time-correlated values (retrieved from two different tags) onto a single XY plot.

For more information, go to the topic, *Tag Mapping*.

a. For **Type**, click **Tag Mapping**.

b. From the **Tag Mapping tab**, click **Add**.
The "Add" dialog box appears.



c. For the **Tag Name**, do one of the following:

- **Enter** a valid Tag name.
- Click the **Browse button** to enter an **expression**.

For information, go to the topic, *Using Expressions in Chart Configurations*.

- Click the **Tag Picker button** to select a tag from the **Tag Selector**.

For the Tag Name, click the **Browse** button to enter an expression.
For information, go to the topic, *Using Expressions in Chart Configurations*.

d. For **Mapping**, click the down arrow and select an option for the XY Chart.

e. Click **OK**.

f. OPTIONAL
Repeat the preceding steps to add more tags.

g. Click (to highlight) a tag and select an **Aggregate** type.

h. For time-sliced aggregates (for example, the Average aggregate), select an **Aggregate Interval**.

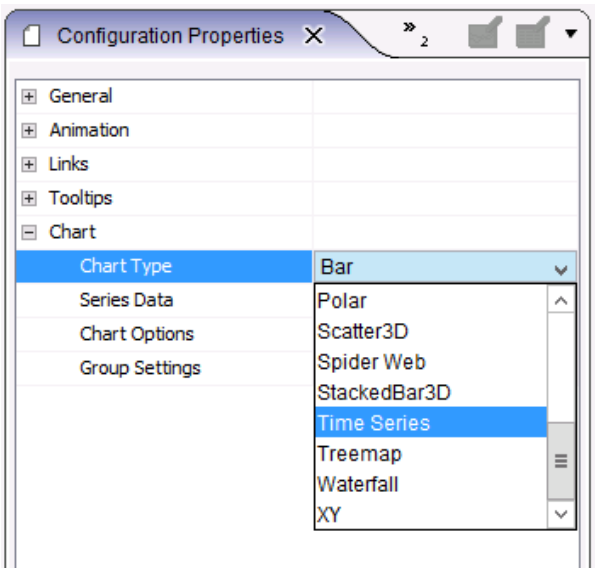
i. OPTIONAL
Repeat steps g and h.

j. Click **OK** and continue to step 4.

4. OPTIONAL
Repeat for each series you want to add.

To copy a series

1. From the **Chart tab**, select a chart.



2. From the table, **select the existing series** you want to copy.
This enables the "Copy" button.

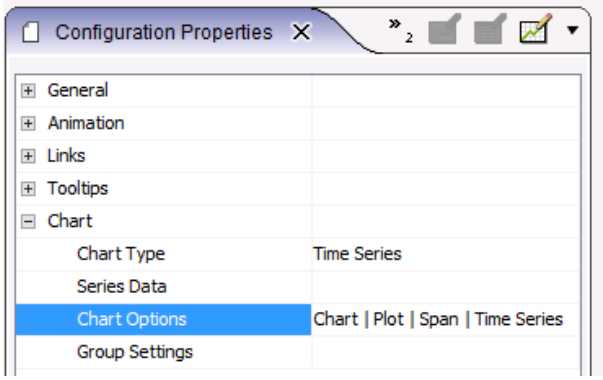
3. Click **Copy**.

The new series appears at the end of the table. It has automatically been given a new, unique series name and has the same configurations as the series from which it was copied.

To set chart options

In this step, you will configure properties that have to do with how your chart and plot will look, such as the title, axis labels, background colors, and so forth. Chart options are dependent on the chart type you select. For example, the Histogram Chart does not have a Span tab, but has a Histogram tab that allows you to set the bins and bounds for the chart.

1. From the **Chart properties**, for **Chart Options**, click **Chart**.



2. Enter a chart **Title** and **Subtitle** name, enclosed by quotations, *OR* click the **Browse button** to enter an expression.

   For more information, go to the topic, *Using Expressions in Chart Configurations*.

   For the **Subtitle**, multiple lines can be entered, separated by a `<br>`.

   Examples:

   `SubtitleLine1<br>SubtitleLine2`

   `"This is line 1"<br>"This is line 2"`

3. OPTIONAL
   In the **Domain group**, do the following:

   The **Domain group** options are not available for the Polar, Pie, and Spider Web charts.

   a. Enter the **Axis Label**.
      This is the label for the domain axis.

   b. Enter the **Value Format** for the tick labels.

      For a Numeric Axis
      The value format must be a valid `DecimalFormat`.

      For a Date Axis
      The value format must be a valid `SimpleDateFormat`.

   c. Select a **Tick Label Angle**, which tilts the tick label at a given angle.

      For charts with categories on the domain, the available options are from 0° (degrees) to 90°, incrementing by 10°. For all other charts, the options are limited to 0° and 90°.

   d. Enter a **Minimum Value** for the domain axis (enabled for the XY chart only).

   e. Enter a **Maximum Value** for the domain axis (enabled for the XY chart only).

4. OPTIONAL
   Click the **color block** (square) to select a **Background Color** for the chart area.
   The Color Selector appears.

   For more information on color, go to the topic *Using the Color Selector*.

5. OPTIONAL
   Check **Show Legend** to display the legend at runtime, and select the legend location to **Bottom**, **Left**, **Right**, or **Top**.

6. OPTIONAL (enabled for the Time Series and XY charts only)
   Check **Split Chart Groups** to separate charts by groups that share a common domain axis.

   For more information, go to the topic, *Grouping*.

7. OPTIONAL
   Check **Allow Interaction with Chart** to enable the end-user to interact with the chart within the view.
8. OPTIONAL (enabled for the Time Series, XY, and Bar charts only)
   - Check **Y-axis Color Based on Series** to set the Y-axis color to the series color. This is the default behavior.
   - Or, leave unchecked to set the Y-axis color to black, which is the same color as the X-axis.
9. OPTIONAL
   For **Tick Mark Label Font & Color**, check **Domain** and/or **Range**.

   The following charts support the Tick Mark Label Font & Color feature:

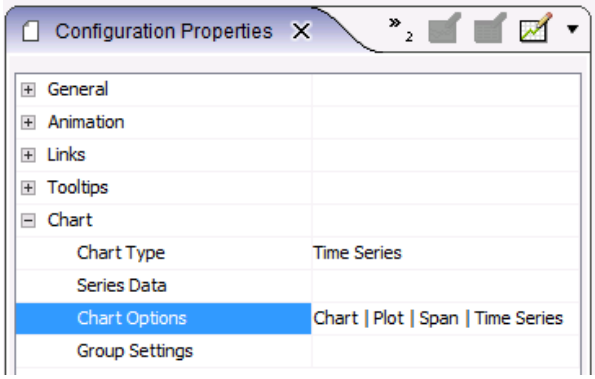
   - XY
   - Time-Series
   - Bar
   - Bubble
   - Candlestick
   - Gantt
   - Histogram
   - Pareto
   - Waterfall
   - Polar
     (the Domain is for the angle labels, and Range is for the value range labels)
   - Spiderweb
     (only the Range is available to set all tick labels in the chart)

   The Pie and Treemap charts do not support this feature because they do not have a domain or a range.

   a. Click **Font** and select a font type and size.
   b. Click the color block to select a **Color** for the Domain and/or Range.
      If the Range portion of Tick Mark Label Font & Color is enabled, the color that is set overrides any default color from the series.

      See the topic, *Label Font and Color* for information on how to use the label font and label color mappings.

10. Click **Plot**.



11. Select an **Orientation** (enabled for the Bar and XY charts only).
12. Configure the **Annotation** (enabled for the Time Series and XY charts only).

- Select **Pointer** to include an arrow pointing to a specific (x,y) location. Determine an arrow **Angle** in degrees.
- Select **Text** to show the annotation as a text label near a specific (x,y) location.
- Set the **Text Anchor** to define a point on the annotation's framing rectangle that is aligned with the specific (x,y) location.

For more information, go to the topic, *Annotations*.

13. OPTIONAL (enabled for the Time Series and XY charts only)
    Check **Domain Crosshair**. This displays a domain crosshair at the center of the plot.

    For more information, go to the topic, *About the Domain Crosshair*.

14. Click the color block to select a **Background Color** for the plot area.
15. Configure the major **Grid Lines** (for the Time Series and XY charts only)

Option	Description
Horizontal Only	Display only horizontal grid lines.
Vertical Only	Display only vertical grid lines.
Both	Display both horizontal and vertical grid lines.
None	Hide the grid

16. Configure the **Minor Grid Lines** (for the Time Series and XY charts only).
17. Select the major **Grid Lines Color**.
18. Select the **Minor Grid Lines Color**.
19. Check the box to **Show Item Label** at runtime.

    For more information, go to the topic, *Label Font and Color*.

20. Check the box to **Show Plot Border** at runtime.
21. Depending on your chart type, do one of the following:
    - Go directly to step 20.
    - Configure the following **Span** options (for Time Series and XY charts only):

      Click the **Browse button** to enter an expression.

      For more information, go to the topic, *Using Expressions in Chart Configurations*.

Option	Description
Start Time	Enter the start time. **Note:** This supports the Date/Time expression.
End Time	Enter the end time. **Note:** This supports the Date/Time expression.
Duration	The duration, together with the Unit option, specifies the amount of time to plot. If both Start Time and End Time are not specified, but the Duration is, then the span for the plot is controlled by the Duration; in which case, the end time is the current time. **Note:** The types supported by this expression are: integer, long, float, double, decimal, or time interval.
Unit	Defines the units in weeks, days, hours (default), minutes, or seconds.
Tag Mapping Show History	*For the XY chart only.* Check to select the Historical mode. For more information, go to the topic, *Tag Mapping*.

- Configure the following **Histogram tab** options:

  Click the **Browse button** to enter an expression.

  For more information, go to the topic, *Using Expressions in Chart Configurations*.

Option	Description
Bins	The number of bins to use for the Histogram chart.
Lower Bound	The lower bound to use for the bins setting.
Upper Bound	The upper bound to use for the bins setting.

- Configure the following **Bar tab** options:

  Click the **Browse button** to enter an expression.

  For more information, go to the topic, *Using Expressions in Chart Configurations*.

Option	Description
Interval Width	Is used to calculate the start and end x-values. This allows the bars to be separated for a better visual display. The default is 2. This option is only enabled when the Domain and Range mappings are used.
Item Margin	Sets the spacing between the bars within each category. The value is expressed as a percentage of the available bar width. At runtime, this margin is distributed evenly between each bar. The larger the percentage, the larger the space is in between each bar. The default is “0.10” (or 10%). To remove the space between the bars (in the same category), set the Item Margin to 0.0.

- For a Spider Web chart, configure the following **Spider Web tab** options:

Option	Description
Show Web Line Labels	Check to display the number labels on each web line.
Fill Web	Check to fill each web series.
Number of Web Lines	Set the number of web lines for the Spider Web chart. The default is 5 lines.
Axis Label Gap	The gap between the axis and the label. Specify this value as a **percentage** of the available drawing space. The default value is 0.05 (which is 5 percent).
Interior Gap	The space between the edge of the plot and the plot itself. This is the region where the axis label appears. Specify this value as a **percentage** of the available drawing space. The default value is 0.10 (which is 10 percent).
Start Angle	The starting angle, in **degrees**, of the plot. The default is 90 degrees.

- For a Gantt chart, configure the following **Gantt tab** options:

For each of the options below, you can click the **Browse button** to enter an **expression** or click the **color block** to use the **Color Selector**.

Click the **Browse button** to enter an expression. For more information, go to the topic, *Using Expressions in Chart Configurations*.

Click the **color block** to use the **Color Selector**

Option	Description
Task/Subtask Color	Set the color of all the tasks and subtasks.
% Complete Color	Set the color of the section representing completed tasks.
% Incomplete Color	Set the color of the section representing incomplete tasks.

- For a Waterfall chart, configure the following **Waterfall tab** options:

If you set these options, then the last value inputed is ignored. Instead, the sum of all previous values are displayed. By default, the last value inputed is used.

Option	Description
Calculate Total	When checked, the last bar is calculated automatically. There's no need to include the total in the data set when this option is checked.
Total Label	The label used for the total bar. The default label is **Total**. This field is enabled only when the **Calculate Total** box is checked.

- For a Treemap chart, configure the following **Treemap tab** options:

If available for the option, click the **Browse button** to enter an expression. For more information, go to the topic, *Using Expressions in Chart Configurations*.

Click the **color block** to use the **Color Selector**

Option	Description
Item Color Logic	Select whether to use the color table for the item color. *Options:* • **None** - The color of the item is specified by the Background Color for the Item group. • **Color Table** - You select the item color using the color table.
Item Group:	*These options configure the items that represent the **leaves**.*
Background Color	The background color of the item if the color table is not used.
Label Color	The item label color.
Label Alignment	The item label alignment. *Options:* Bottom Center, Bottom Left, Bottom Right, Center, Center Left, Center Right, Top Center, Top Left, Top Right
Label Font	The item label font.
Exception Group:	*These options configure the exception items. Exception items are specifically marked by the Exception mapping column.*
Background Color	The exception item background color.
Label Color	The exception item label color.
Label Alignment	The exception item label alignment. *Options:* Bottom Center, Bottom Left, Bottom Right, Center, Center Left, Center Right, Top Center, Top Left, Top Right
Label Font	The exception item label font.
Category Group:	*These options configure the category items.*
Background Color	The background color of the category panel.
Border	The color of the category border.
Label Color	The category label color.
Label Alignment	The category label alignment. *Options:* Top Left, Top Center, Top Right
Label Font	The category label font.
Color Table Exception Group:	*These options configure the items that fall below the smallest value or above the largest value in the color table.*
Below Min Value Color	The background color of the items that fall **below the minimum value** of the color table.
Below Max Value Color	The background color of the items that fall **above the maximum value** of the color table.
Value Number Format	Specifies the format of all values that appear in the treemap.

22. OPTIONAL (for Time Series chart only)
    Click the **Time Series tab** and configure the following options.

If available for the option, click the **Browse button** to enter an expression. For more information, go to the topic, *Using Expressions in Chart Configurations*.

Click the **color block** to use the **Color Selector**

Option	Description
Time Offset	A Time Offset chart is a **Time Series chart** that has one series serving as the **Golden Run**, with the other remaining series as Compared Run that are offset by the Compared Run Time. For more information, go to the topic, *Time Offset Types*.
Use Domain Tick Unit	Select this checkbox to show the tick labels of actual point values on the domain axis (x-axis) only. For more information, go to the topic, *Setting the Tick Unit.*
Item Margin	Sets the spacing between the bars. The value is expressed as a percentage of the available bar width. At runtime, this margin is distributed evenly between each bar. The larger the percentage, the larger the space is in between each bar. The default is "0.10" (or 10%). To remove the spaces between the bars, set the Item Margin to 0.0.

23. Click **OK**.

To set options for **3D Charts**, go to the topic, *Setting Chart Options for 3D Charts*.

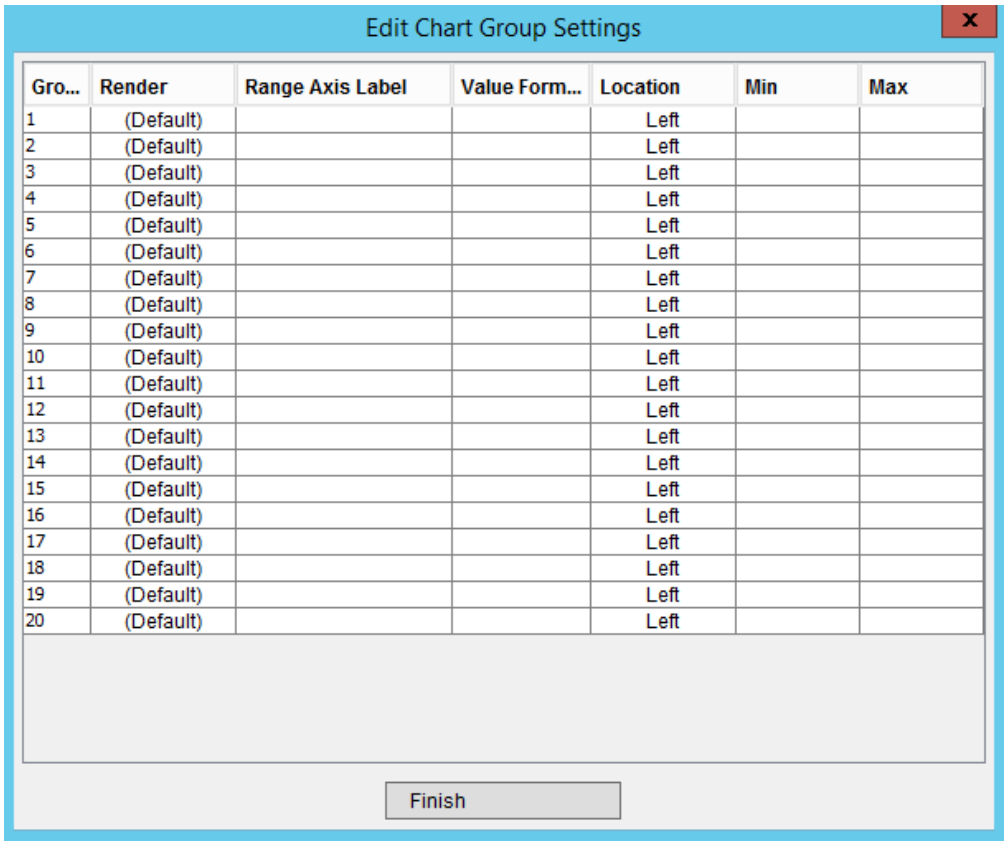
To define the group settings

The "Group Settings" dialog allows you to set renderers and range axis labels for a group.

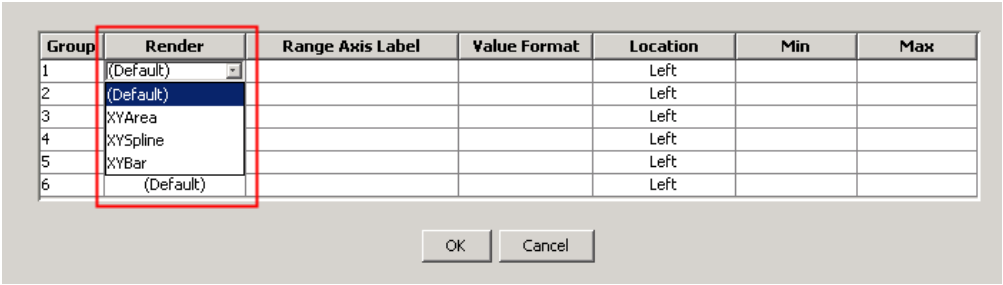
For additional information, go to the topics:

- *Grouping*
- *Renderers*
- *Using Multiple Range Axes*

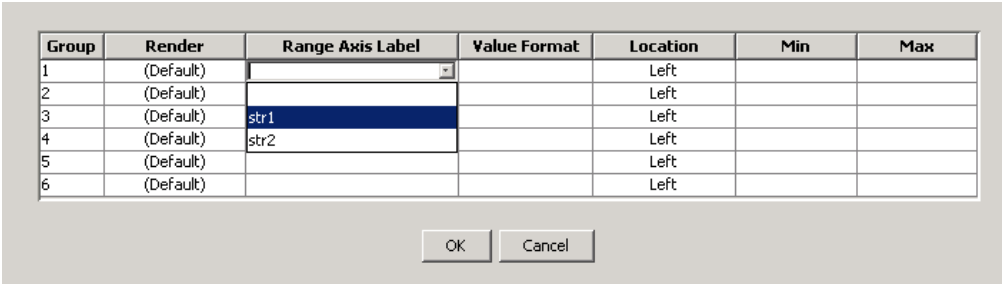
1. From the **Chart tab** in the XHQ Workbench, click **Group Settings**.
   The "Group Settings" dialog box appears.



2. Do the following for **Group 1**:
   - Click inside the **Render cell** and select the renderer.



   - Click inside the **Range Axis Label cell** and select the label for Group 1.



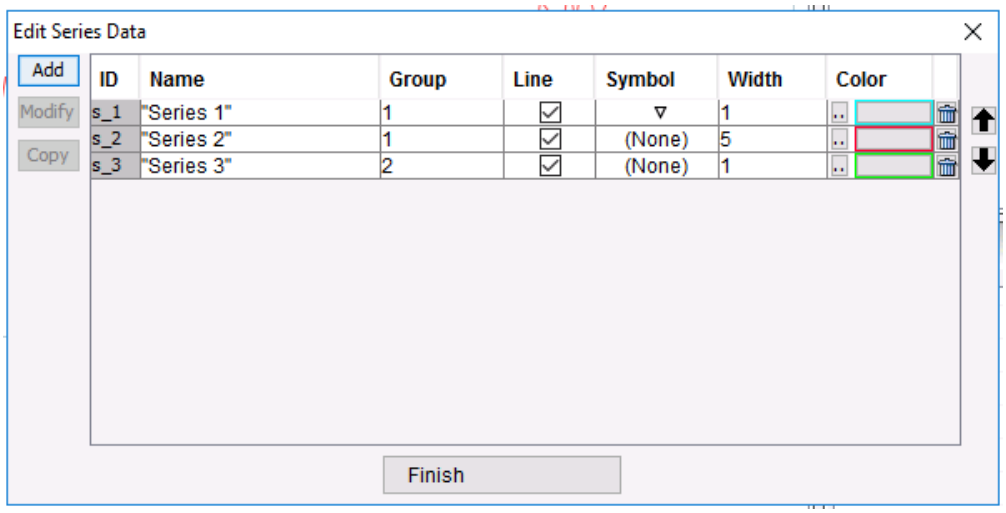
   - Click inside the **Value Format cell** and select a value.
   - Select the **Location** of the Range axis.
   - Enter a numeric value for **Min** (which is the minimum value for the Range axis).

- Enter a numeric value for **Max** (which is the maximum value for the Range axis).

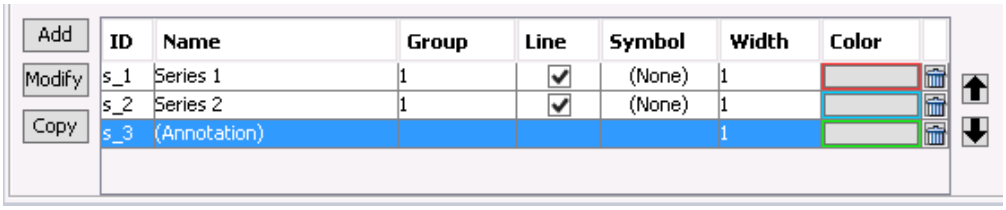
3. **Repeat step 2** for additional groups.
4. Click **OK**.

About the Series Table

The Series Data dialog displays the chart configuration.



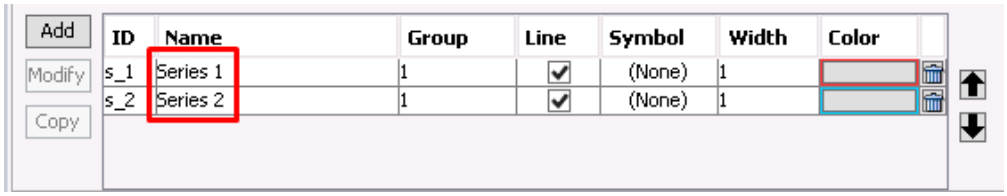
Example: Series Table



Example: Series Table with Annotation

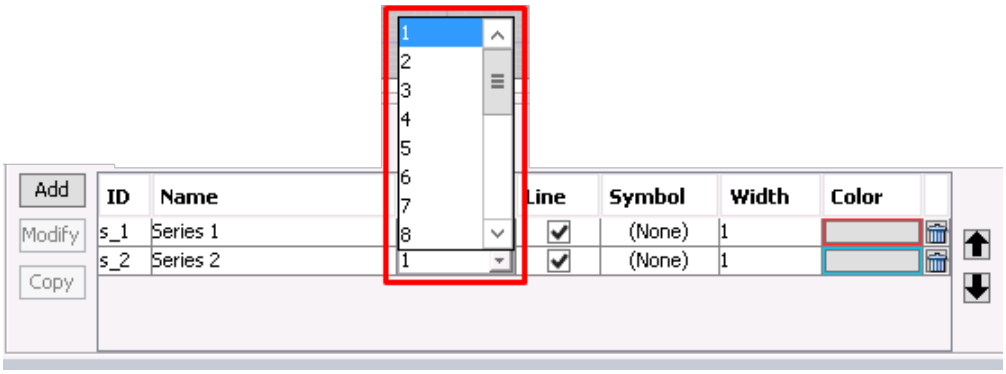
The Series Table enables you to set (depending on the chart type) the following options.

- Modify the series **Name**.
  Double-clicking on the series name makes the cell editable.

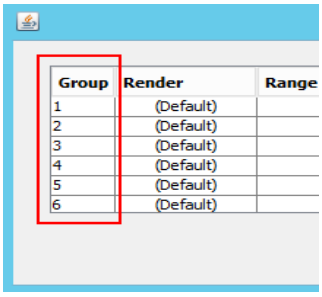


  After renaming the series, press **Enter** to save the change.

- Combine like series into **Groups**.
  Click inside the Group cell to display the drop-down list.

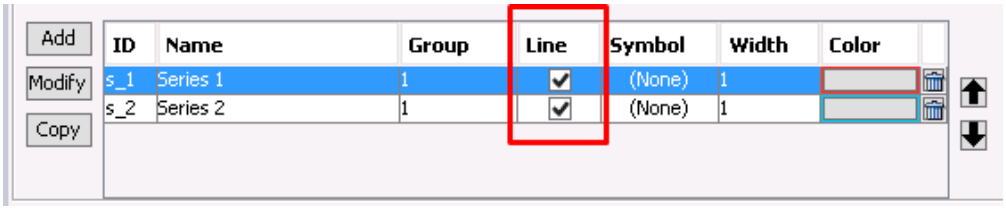


By default, there are six groups available. Each group corresponds to a group in the Group Settings panel.



See the topic *Grouping* for details.

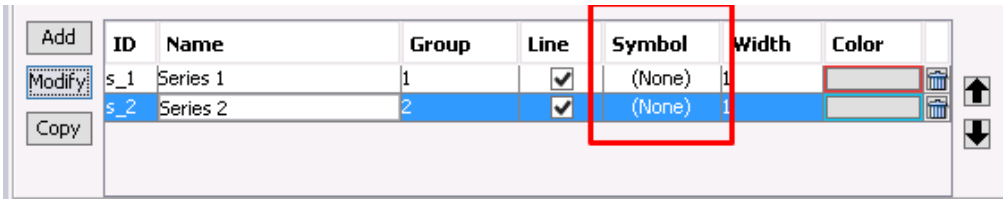
- Show or hide plot **Lines**.
  Check this option to display plot lines; deselect to hide.



The **Line option** is only available for the **Time Series**, **XY**, and **Bar** charts.

By default, this column is *disabled* for **Bar** charts. To enable, you must select a line-based *renderer* (such as the LineAndShape renderer) for the Bar chart **Group**.

- Assign **Symbols**.
  This option allows you to assign a symbol that marks each data point on a line chart. Click inside the Symbol cell to display the drop-down list.



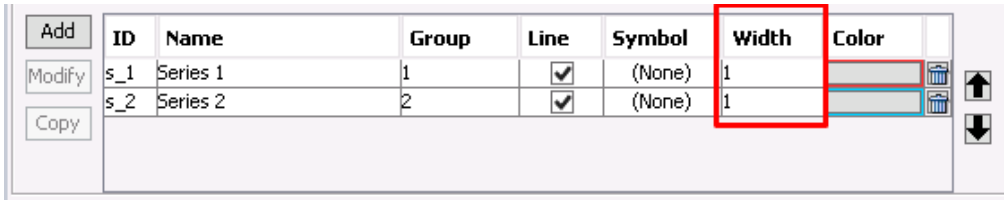
The following symbols are available: square, circle, up-pointing triangle, diamond, horizontal rectangle, down-pointing triangle, horizontal ellipse, right-pointing triangle, vertical rectangle, and left-pointing triangle.

When **(None)** is selected, the symbol is hidden.

The **Symbol option** is only available for the **Time Series**, **XY**, and **Bar** charts.

By default, this column is *disabled* for **Bar** charts. To enable, you must select a line-based *renderer* (such as the LineAndShape renderer) for the Bar chart **Group**.

- Select plot line **Widths** (in pixels).
  Click inside the Width cell to display the drop-down list.

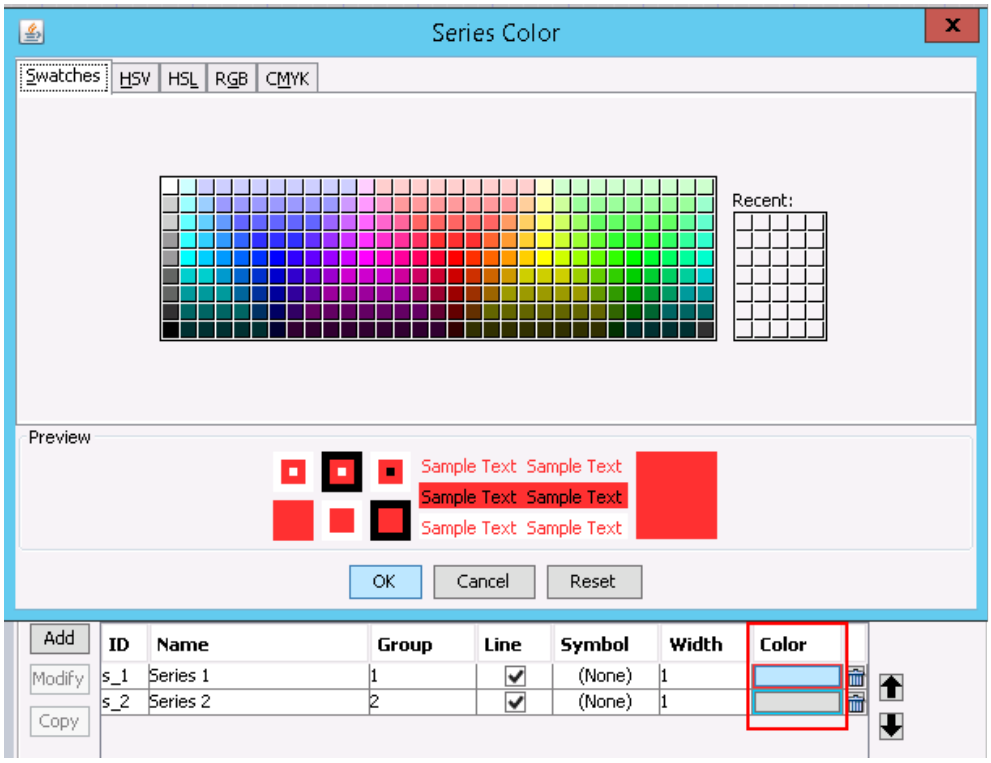


You can select from line widths of 1 pixel to 9 pixels.

The **Width option** is only available for the **Time Series**, **XY**, and **Bar** charts.

By default, this column is *disabled* for **Bar** charts. To enable, you must select a line-based *renderer* (such as the LineAndShape renderer) for the Bar chart **Group**.

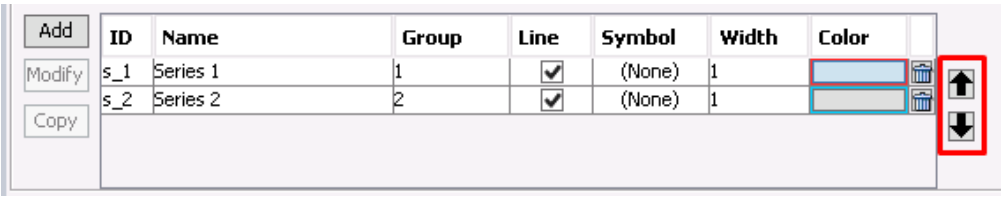
- Select a plot **Color**.
  Click the color swatch to launch the **Color Selector**.
  The "Series Color" dialog appears.



For more information on color, go to the topic, *Using the Color Selector*.

- **Delete** a series from the table.
  Click the garbage can icon next to the series you want to delete.

- Use the **Up and Down Arrows** to order the series.
  From the table, select the series, then click either the up or down arrow to re-order the series within the table.



Series Types

When adding a series to a chart, you will need to select the type of series. The type(s) available depend(s) on the chart type you select. Below, is a summary of the series type(s) that is/are available for each chart.

Series Types for each Chart Type

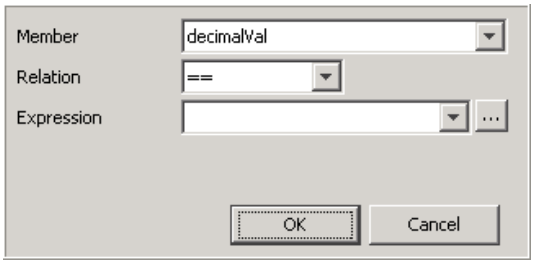
Chart Type	Collection	Tag	Tag Mapping
Bar	X		
Bubble	X		
Candlestick	X		
Gantt	X		
Histogram	X		
Pareto	X		
Pie	X		
Polar	X		
Spider Web	X		
Time Series	X	X	
Treemap	X		
Waterfall	X		
XY	X		X

Setting Criteria

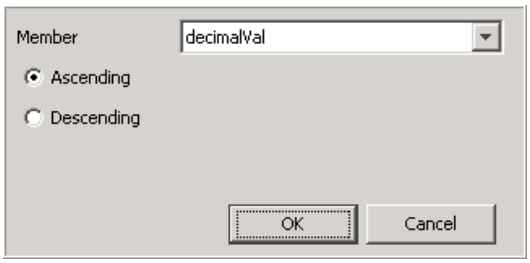
When adding a collection-type series to the chart, you can set criteria to filter the collection.

Specify a second expression for the "between" criteria relation.

You can **Filter** against a collection member using an expression.



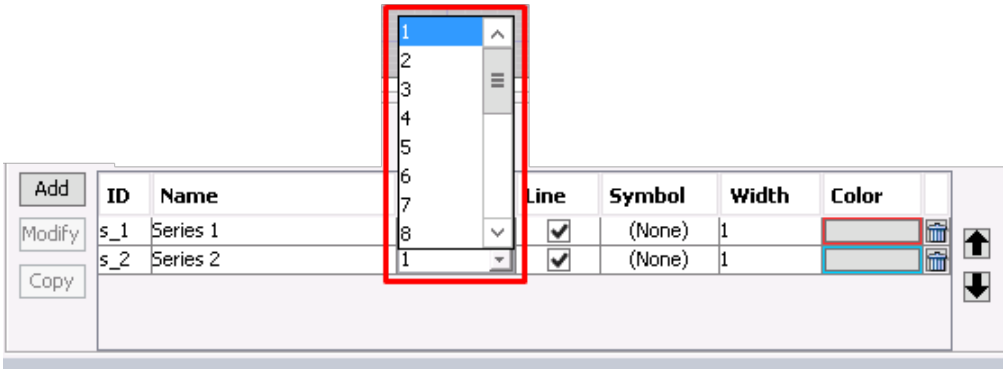
You can **Sort** against a collection member in ascending or descending order.



Refer to the topic, *Using Expressions in Chart Configurations*, for more information on expressions.

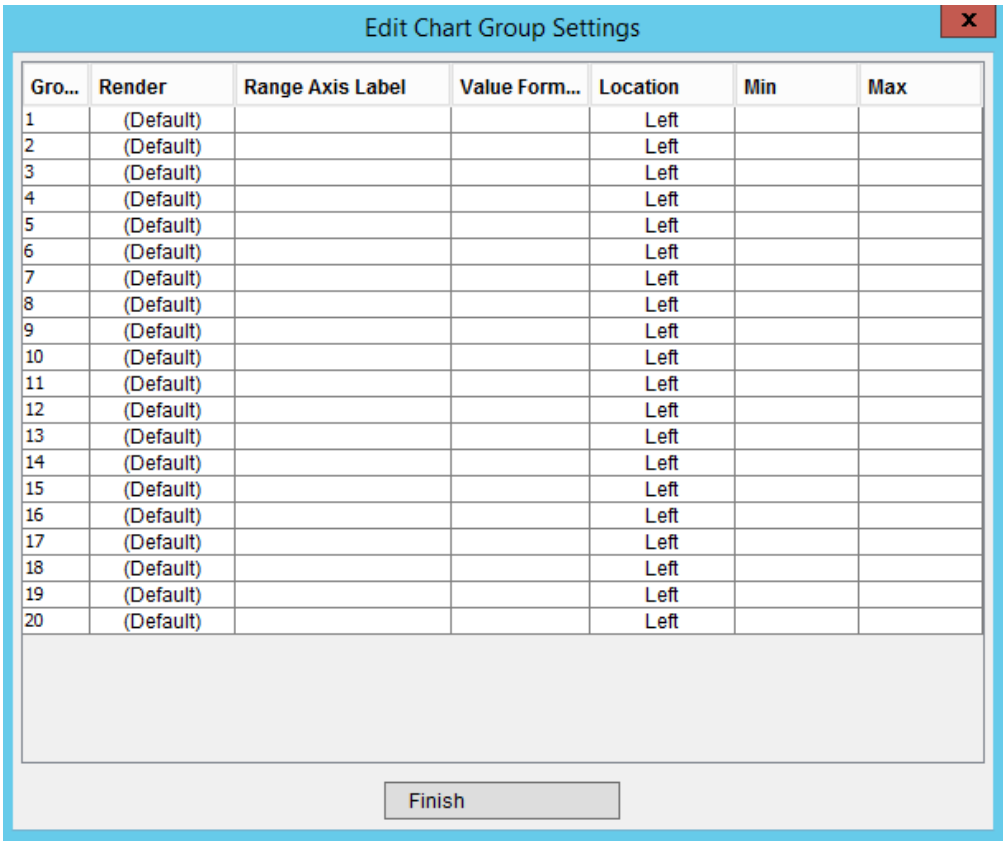
Grouping

You can group multiple series together as a unit using the Group column in the Series Table.



Series Table: Group Column

Grouped series share the same settings: Renderer, Range Axis Label, Location, and Min/Max values. For example, if a Renderer is defined for a group, then all the series that belong to that group will have the Renderer applied to them. The **Group Settings** dialog is used to define these settings.



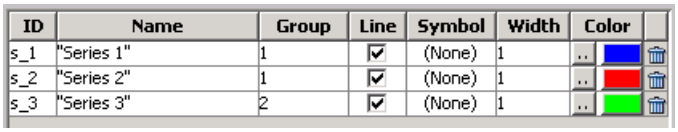
Gro...	Render	Range Axis Label	Value Form...	Location	Min	Max
1	(Default)			Left		
2	(Default)			Left		
3	(Default)			Left		
4	(Default)			Left		
5	(Default)			Left		
6	(Default)			Left		
7	(Default)			Left		
8	(Default)			Left		
9	(Default)			Left		
10	(Default)			Left		
11	(Default)			Left		
12	(Default)			Left		
13	(Default)			Left		
14	(Default)			Left		
15	(Default)			Left		
16	(Default)			Left		
17	(Default)			Left		
18	(Default)			Left		
19	(Default)			Left		
20	(Default)			Left		

Group Settings Dialog

All series in a group can be combined into a **single** chart when the **Split Chart Groups** option (which is located in the **Chart Options** dialog) is checked.

EXAMPLE: SPLITTING CHART GROUPS

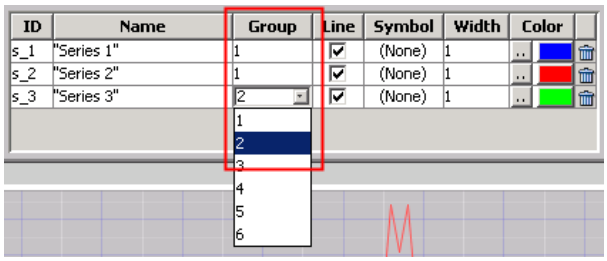
The following three series (for a Time Series chart) are configured as shown in the following figure:



ID	Name	Group	Line	Symbol	Width	Color
s_1	"Series 1"	1	☑	(None)	1	
s_2	"Series 2"	1	☑	(None)	1	
s_3	"Series 3"	2	☑	(None)	1	

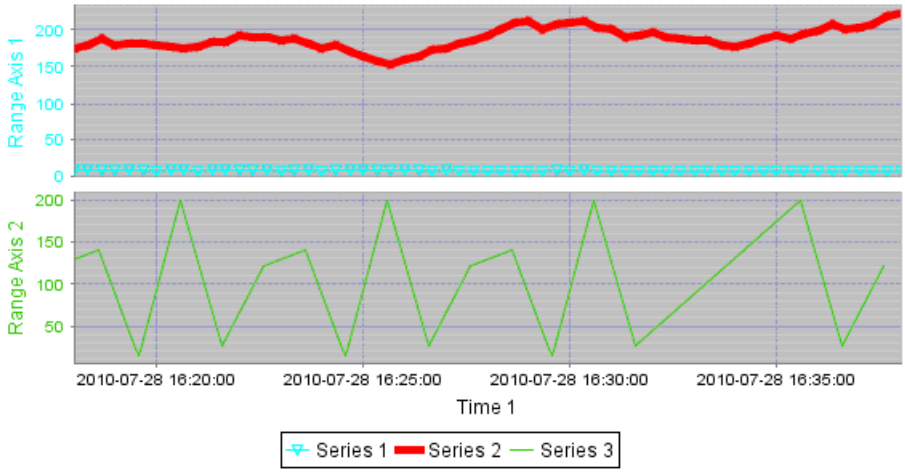
- Series 1 and 2 belong to **Group 1**.
- Series 3 belongs to **Group 2**.

1. To change the group, click inside the **Group cell**.
   A list of numbers from 1 to 6 appears.

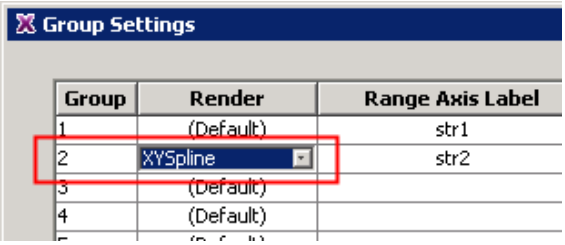


2. Select a group number.
3. Next, click **Group Settings**.
   The "Group Settings" panel appears.
4. In the **Group Settings** panel, set the **Range Axis Label** for Group 1 and 2.
5. Click **OK** to close the dialog.
6. Click **Chart Options**.
   The "Chart Options" panel appears.
7. On the **Chart tab**, check the **Spit Chart Groups** option.
8. Click **OK** and the **Update**.

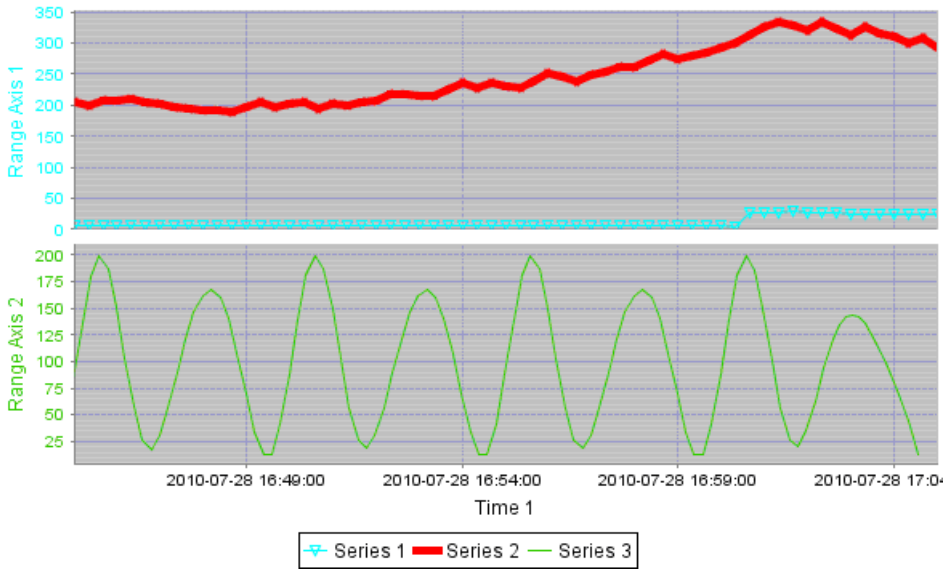
At runtime, the resulting plot contains Group 1 (top) and Group 2 (bottom):



To smooth out the line, open the Group Settings and apply the **XYSpline Renderer** to **Group 2**.



Note the plot difference in Group 2:



See the topic, *Renderers*, for more information.

Group By for Charts

Group by is available for all collection-based series, including 3D charts. When group by is enabled, the data is grouped and aggregated before it is rendered in the chart.

To configure group by

1. Add a **collection-based series**.
2. Select the **Enable Group By** option.
3. Add all the **required mappings** for the chart.
4. OPTIONAL
   During mapping, select one of the following aggregates:
   - Average
   - Count
   - Max
   - Min
   - Sum

   The default is None.

Important Things to Note

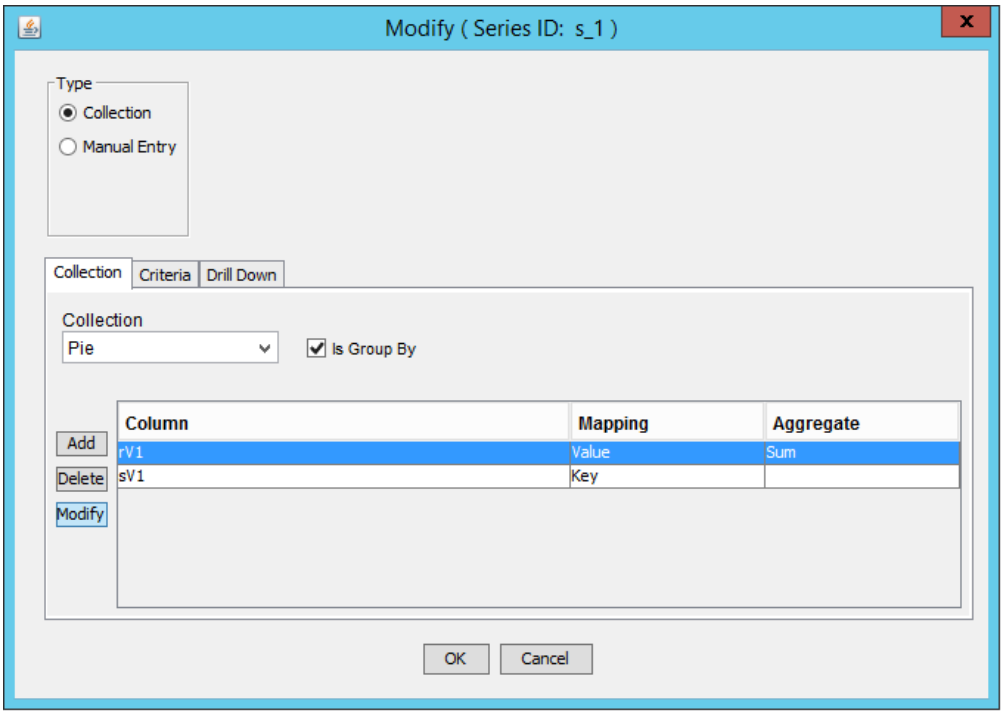
- During mapping, if a column is a String type, only the Count aggregate is available for selection.
- If there are any non-simple expressions when Group By is enabled, a message will indicate that those mappings will not be involved in the Group By query.
- Any mapped column that doesn't have an aggregate set will automatically be a Group By column.
- If the Group By option is currently checked but then becomes unchecked, all mappings will be cleared of their aggregate settings.

Drill-down Expressions

With Group By enabled, an expression can be generated to access a mapped column of the selected entity.

For example, given a Pie chart with the following configurations:

- **Name** of the chart is "Chart1".
- **Allow Interaction with Chart** is enabled.
- Enable **Group By** option is checked.
- Contains two mappings:
  - Value: aggregate Sum
  - Key
- Drill-down option **Update Dependent Items** is selected.



To retrieve the mapped "Key" for the clicked pie slice, set the value animation for a **ValueItem** control that is formatted as follows:

```
$Chart1.[SeriesID]._Mapping.[Mapping].selected()
```

Where:

- `SeriesID` is a series ID for `$Chart1`.
  For example, `s_1`
- `Mapping` is one of the mapped mappings for the `SeriesID`.
  For example, Value or Key.
- `_Mapping` is a special keyword that follows directly after the series ID to indicate what follows in the expression is a mapping, instead of the normal collection column.

Renderers

Applying a Renderer causes the plot to display differently without making any changes to the actual data set. The Render option is accessed from the **Group Settings** dialog. Renderers, however, are only available for the following chart types.

Charts with Renderers

Chart Type(s)	Renderers Available
Time Series and XY Charts	• **XYLineAndShape** Draws a line between each data point and overlays a shape at each point. This is the **Default** renderer. • **XYArea** Draws each item in the data set using a polygon that fills the area

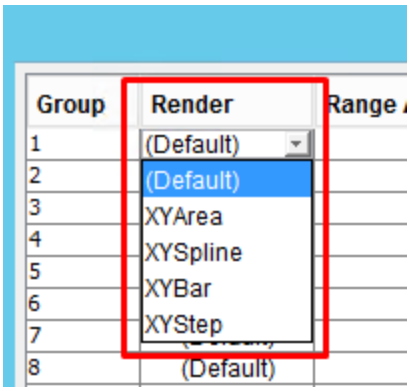
Chart Type(s)	Renderers Available
	between the X-axis and the data point. • **XYSpline** Connects data point using spline curves, resulting in a smooth line passing though all the data points. • **XYBar** Renders points as bars. This renderer is useful to do a composite chart containing bars and lines. • **XYStackedArea** Draws each item in the data set using a polygon that fills the area between the x-axis and the data point. This is different from the XYArea renderer in that the area is stacked on top of one another for the series. **Note**: Currently, this renderer is only available for the Time Series chart. The Time component for each series should be the same for the best stacked result. • **XYStackedArea100%** Draws each item in the data set using a polygon that fills the area between the x-axis and the data point. This is different from the XYArea renderer in that the area is stacked on top of one another for the series, with values normalized to 100%. **Note**: Currently, this renderer is only available for the Time Series chart. The Time component for each series should be the same for the best stacked result. • **XYStackedBar** Renders points as bars. This is different from the XYBar renderer in that the bars are stacked on top of one another for the series. Note: Currently, this renderer is only available for the Time Series chart. The Time component for each series should be the same for the best stacked result. • **XYStackedBar100%** Renders points as bars. This is different from the XYBar renderer in that the bars are stacked on top of one another for the series, with values normalized to 100%. Note: Currently, this renderer is only available for the Time Series chart. The Time component for each series should be the same for the best stacked result.
Bar **(Category Data only)**	• **Bar** Renders the Category data items as bars. This is the **Default** renderer. • **Bar3D** Renders the Category data items as 3-dimensional bars. • **BarStacked** Renders the Category data items as stacked bars, normalizing the values to 100%. • **AreaStacked** Renders the Category data items as stacked area.

Chart Type(s)	Renderers Available
	• **AreaStacked100%** Renders the Category data items as stacked area, normalizing the values to 100%. • **LineAndShape** Displays data items by drawing a shape at each data point and/or connecting data point with straight lines.

Renderer Support for Runtime Features

Renderer	Color	Annotation	Tooltip	Drill-down	Golden Run Support*
Bar	X	X	X	X	
Bar3D	X	X	X	X	
BarStacked	X	X	X	X	
BarStacked100%	X	X	X	X	
AreaStacked	X	X	X		
AreaStacked100%	X	X	X		
LineAndShape	X	X	X	X	
XYLineAndShape	X	X	X	X	X
XYArea		X	X		X
XYSpline	X	X	X	X	X
XYBar	X	X	X	X	X
XYStackedArea		X	X		
XYStackedArea100%		X	X		
XYStackedBar		X	X		
XYStackedBar100%		X	X		

* Golden Run feature is for **Time Series Charts** only.

Rendering Limits

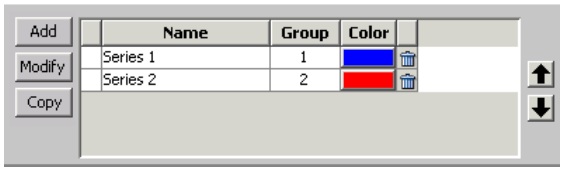
There are limits to the number of items that are rendered. These limits ease the plot times and produces charts that are more readable. The number of items are limited to 1000 for all charts.

Using Multiple Range Axes

Multiple Range axes are available for the **Time Series**, **XY**, and **Bar** charts.

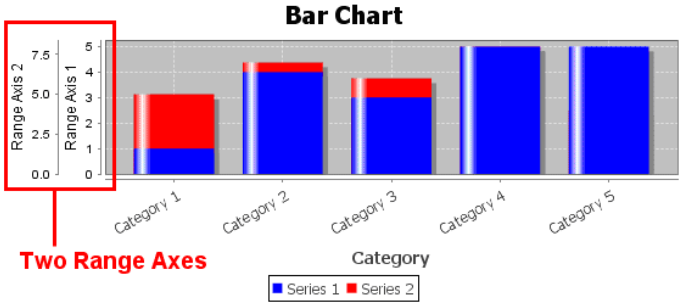
EXAMPLE: USING MULTIPLE RANGE AXES

1. Create a **Bar chart**.
2. **Add two series**, setting Series 1 to **Group 1** and Series 2 to **Group 2**.



3. Click **Group Settings**.
   The "Group Settings" panel appears.
4. For **Group 1**, click inside the **Range Axis Label cell** and select a value.
5. For **Group 2**, click inside the **Range Axis Label cell** and select a value.
6. Click anywhere outside the panel to save and close.
7. Click **OK** and **Update**.

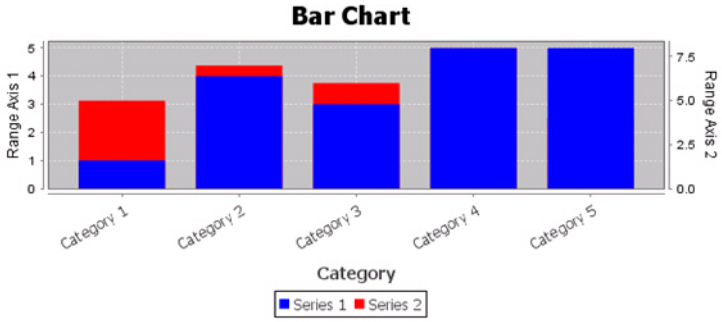
At runtime, the resulting plot displays two Range axes:



Setting the Range Axis Location

You can place the Range axis on either side of the plot, left or right. Or, you can select None to hide the Y-axis at runtime.

In the previous Bar Chart example for multiple Range axes, if the **Location** for **Group 2** is set to the **Right**, then the resulting plot is:



About Axis Minimum and Maximum Values

By default, plots are automatically ranged for both the Domain and Range axes. This enables all series to fit within the plot area without requiring you to set axis parameters. You may, depending on the chart type, manually set the minimum and/or maximum axis values for both the Domain and Range axes.

Important Things to Note

- Minimum and maximum values must be numeric values.
- The maximum value must be larger than the minimum value. If not, the plot will not update.

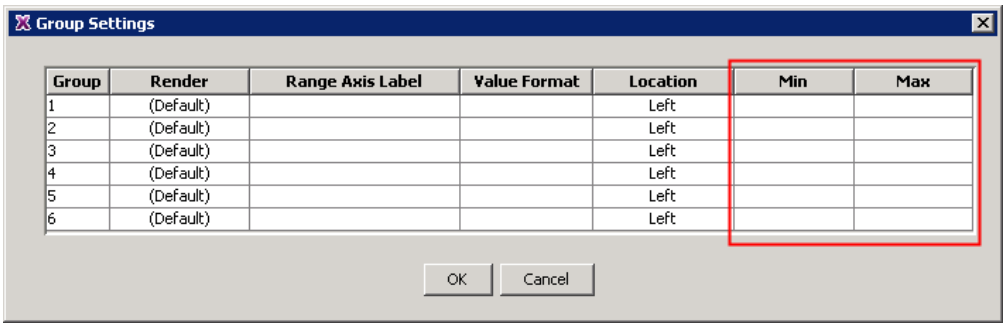
For the Domain Axis

Only the **XY chart** allows you to manually set the minimum/maximum Domain axis values from the **Chart Options** dialog.

You may set the Domain axis minimum/maximum values for the **Time Series chart** using the **Span** Start and End Time.

For the Range Axis

You can manually set the minimum/maximum Range axis values from the **Group Settings** dialog. This, however, is only available for the **Time Series** and **XY charts**.



Group Settings: Range Min/Max Value

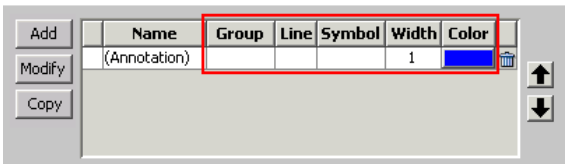
Annotations

An annotation is text that is displayed on the plot at a specific (x,y) location.

To see which charts support annotations, go to the topic, *Chart Features*.

Additional Things to Note

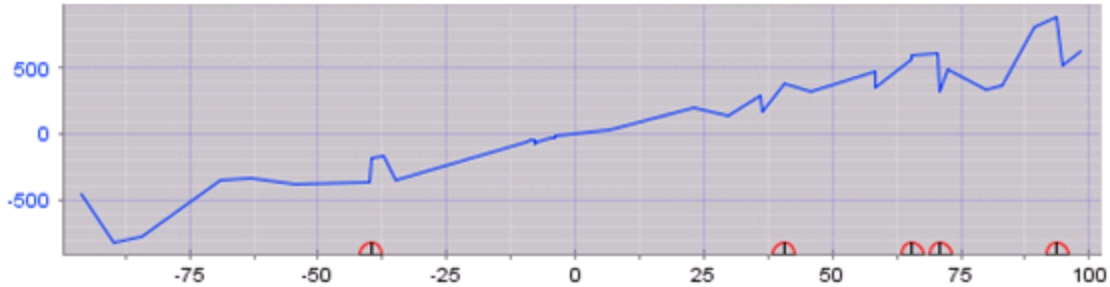
- The Annotation series must exist with the Chart series.
- Only one Annotation series is allowed per chart.
- In an Annotation series, the Group, Line, and Symbol columns are disabled. Only the **Width** and **Color** can be edited.



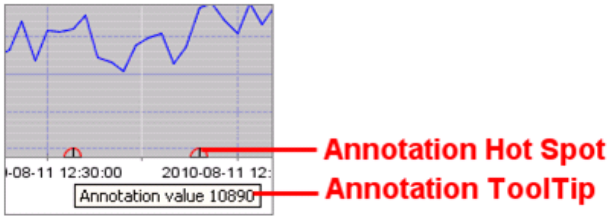
Annotation Types

There are two types of annotation that can be configured:

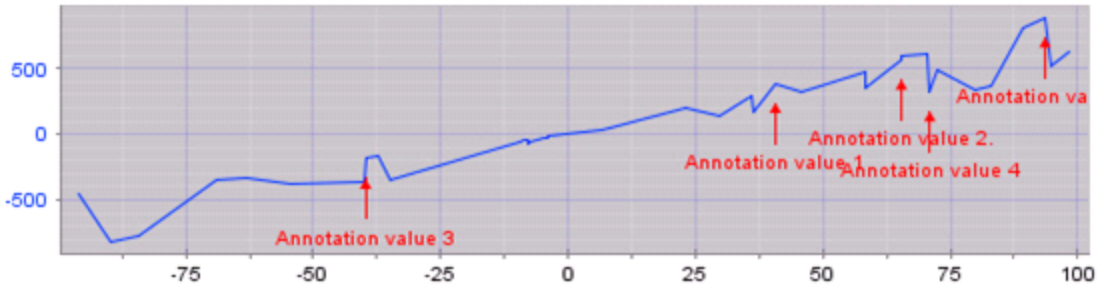
Type 1 This annotation type results in annotations that appear along the X-axis as semi-circles.



Mouse-over these annotations to display the Tooltip.



Type 2 This annotation type results in annotations that appear at the location specified by the annotation mapping.



The following table shows which chart(s) support these annotation types as well as the mapping requirements.

Annotation Mapping Requirements

Chart Type	Type 1	Type 2
Time Series	Time, Annotation	Time, Value, Annotation
XY	Domain, Annotation	Domain, Range, Annotation
Gantt	(not supported)	Task Name, End Date, Annotation
Pareto	(not supported)	Value, Category, Annotation
Waterfall	(not supported)	Value, Category, Annotation **Note:** Annotation series must have the same filter as the main series.
Bar	Domain, Annotation (XY-based)	Value, Category, Annotation (category-based) Domain, Range, Annotation (XY-based)
Candlestick	Time, Annotation	Time, High, Annotation
Bubble	X, Annotation	X, Y, Z, Annotation

Adding Annotations

To add an annotation to a **Time Series chart**, you must map the Time column to a Date/Time data type, the Value column to a Numeric data type, and the **Annotation** column to a **String**.

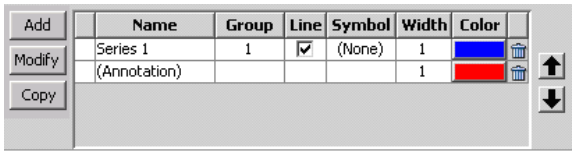
For a list of data mapping rules for each chart type, go to the topic, *Chart Mapping and Data Sets*.

To add an annotation to an **XY chart**, you must map the Domain and Range columns to Numeric data types, and the **Annotation** column to a **String**.

EXAMPLE: Adding a Type 1 annotation to a Time Series chart

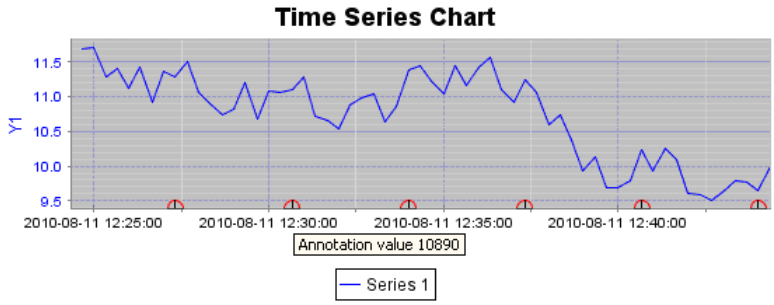
1. Create a Time Series chart.
2. Click **Add** to configure a **series**.
3. **Map** the **Time** and **Value** columns and click **Add**.
   This returns you to the series table.

4. Click **Add** to configure an **annotation**.
5. **Map** the **Time** and **Annotation** columns.
6. Click **Add**.
   The series table lists a series and an annotation.



7. **Save**.

At runtime, the resulting plot displays the annotations along the X-axis.



Series Color and Custom Colors

The color associated with a series is normally set by the Color column of the series table, with the following exceptions:

- **Treemap** Chart
  Colors of items are controlled by Shade mapping and "Treemap" Chart Options.
- **Gantt** Chart
  Colors are controlled by "Gantt" Chart Options.
- **Pie** Chart
  Pie slice colors are automatically assigned.
- **Waterfall** Chart
  Bar colors are automatically assigned.

In order to control the colors for the individual entity of certain charts, the Color mapping can be added to a series.

For additional information, go to the topic, *Chart Mapping and Data Sets*.

The table below describes this mapping for each supported charts in detail.

Chart Color Mapping

Chart Type	How Color Mapping Affects Resulting Plot	Requirements
Bar	Bars are colored according to the Color mapping. Bars without a color (namely, no color was assigned in the data) default to the series color.	The Domain column must be sorted "ASC".
Bubble	Bubbles are colored according to the Color mapping. Bubbles without a color (namely, no color was assigned in the data) default to the series color.	N/A
Pareto	Bars are colored according to the Color mapping. Bars without a color (namely, no color was assigned in the data) default to the series color. **Note**: The accumulated percentage line is not effected by the custom color.	N/A
Pie	Pie slices are colored according to the Color mapping. Slices without a color (namely, no color was assigned in the data) are automatically assigned a color.	N/A
Polar	The symbol at each point is colored	Must enable symbol to see custom

Chart Type	How Color Mapping Affects Resulting Plot	Requirements
	according to the Color mapping. Symbols without a color (namely, no color was assigned in the data) are assigned the default color, which is white.	colors. And the Domain column must be sorted "ASC".
Spider web	The symbol at each point is colored according to the Color mapping. Symbols without a color (namely, no color was assigned in the data) are assigned the default color, which is white.	Must enable symbol to see custom colors.
Time Series	The symbol at each point is colored according to the Color mapping. Symbols without a color (namely, no color was assigned in the data) are assigned the default color, which is white.	Must enable symbol to see custom colors. Time column must be sorted "ASC".
XY	The symbol at each point is colored according to the Color mapping. Symbols without a color (namely, no color was assigned in the data) are assigned the default color, which is white.	Must enable symbol to see custom colors. Domain column must be sorted "ASC".

Label Font and Color

To control the label font and colors, add the following mappings.

Map	Description
Label Color	Color formats are: • **Hex** *Examples*: `#0000FF` (for blue), `#FF0000` (for red) • **Numeric** (Microsoft Access Code) *Examples*: 16711680 (for blue), 255 (for red)
Label Font	The Font format is: **`Name;Style;Size`** *Where* • Name Select from Helvetica, Courier, or TimesRoman • Style Select from the following integer values:

- 0=Plain
- 1=Bold
- 2=Italic
- 3=Bold Italic

- Size
  Enter an integer value for the size of the font.

Examples:

Courier;1;14

TimesRoman;3;12

To display the labels at runtime, make sure to enable the **Show Item Label** option in the Chart Options > *Plot tab*.

The following charts support the **Label Color** and **Label Font** mappings:

- XY
- Time-Series
- Bar
- Bubble
- Gantt
- Pareto

The mappings can also be used on **Annotation** series for the following charts:

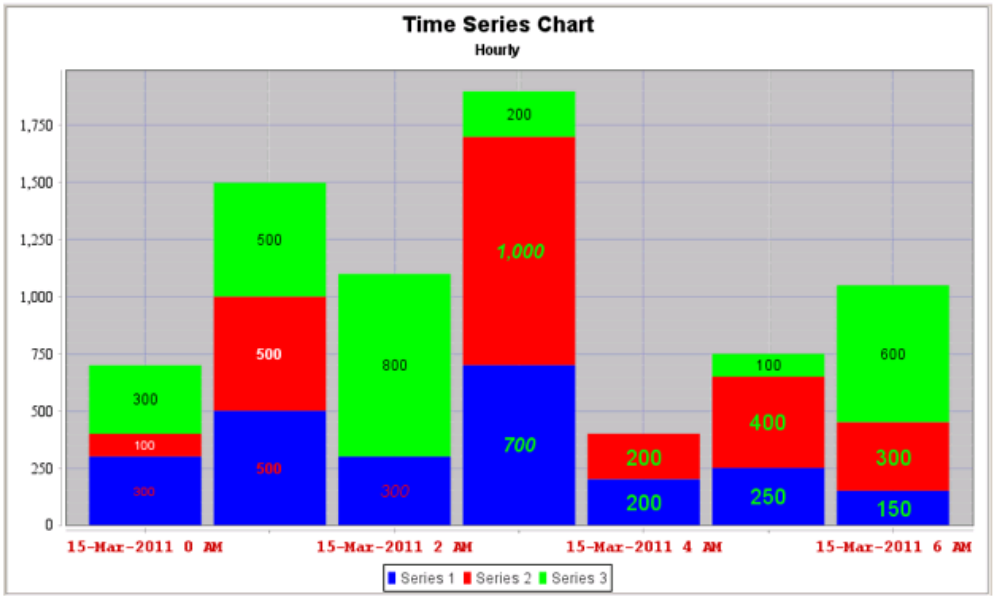
- Bar
- Waterfall
- Pareto
- XY
- Time-Series

The Pie chart does not support this feature. In addition, the following **Renderers** for XY and Time-Series charts are not supported for these new mappings:

- XY Area
- XY Stacked Area
- XY Stacked100%

For additional information, go to the topic, *Chart Mapping and Data Sets*.

The following is the runtime example of an XY Stacked Bar chart with the item labels configured to have various fonts and colors.


Time Series Chart
Hourly
1,750
1,500
1,250
1,000
750
500
250
0
300
100
500
500
500
800
300
200
1,000
700
200
200
100
400
250
600
300
150
15-Mar-2011 0 AM
15-Mar-2011 2 AM
15-Mar-2011 4 AM
15-Mar-2011 6 AM
Series 1
Series 2
Series 3

Tooltips and Custom Tooltips

By default, each chart type has a built-in Tooltip which appears when hovering over an entity at runtime. If a custom Tooltip mapping is used, the custom Tooltip appears. The table below describes how the Tooltip appears.

Chart Default Tooltip Format

Chart Type	Default Tooltip Format**	Custom Tooltip
Bar	(**Series Name, Category**) = **Value**	Displays custom Tooltip, if available. Otherwise, use Default Tooltip. Supports HTML formatting of Tooltip.
Bubble	**Series Name**: (**X, Y, Z**)	Displays custom Tooltip, if available. Otherwise, use Default Tooltip. Supports HTML formatting of Tooltip.
Candlestick	**Series Name** -> Date=**Time** High=**High** Low=**Low** Open=**Open** Close=**Close**	Displays custom Tooltip, if available. Otherwise, use Default Tooltip. Supports HTML formatting of Tooltip.
Gantt	**Start Date – End Date**	Displays custom Tooltip, if available. Otherwise, use Default Tooltip. Supports HTML formatting of Tooltip.
Histogram	**Series Name**: (Bin Value, Bin Length)	Not supported.
Pareto	(**Series Name, Category**) = **Value**	Displays custom Tooltip, if available. Otherwise, use Default Tooltip. Supports HTML formatting of Tooltip.
Pie	**Key**: (**Value**, Percentage)	Displays custom Tooltip, if available. Otherwise, use Default Tooltip. Supports HTML formatting of Tooltip.
Polar	**Series Name**: (**Domain,Range**)	Displays custom Tooltip, if available. Otherwise, use Default Tooltip. Supports HTML formatting of Tooltip.
Spider web	(**Series Name, Category**) = **Value**	Displays custom Tooltip, if available. Otherwise, use Default Tooltip. Supports HTML formatting of Tooltip.
Time Series	**Time, Value**	Displays custom Tooltip, if available. Otherwise, use Default Tooltip. Supports HTML formatting of Tooltip.
Treemap	**Item** **Box Size**	Displays custom Tooltip, if available. Otherwise, use Default Tooltip.
Waterfall	(**Series Name, Category**) = **Value**	Displays custom Tooltip, if available. Otherwise, use Default Tooltip. Supports HTML formatting of Tooltip.

Chart Type	Default Tooltip Format**	Custom Tooltip
XY	**Series Name**: (**Domain, Range**)	Displays custom Tooltip, if available. Otherwise, use Default Tooltip. Domain column must be sorted "ASC". Supports HTML formatting of Tooltip.

*** Bold text in the "Default Tooltip Format" column are mappings or chart entities.*

Tooltips for charts can be **turned off** by setting the property **`DisableToolTipInView`** to **TRUE** in the `globalsettings.properties` file.

For more information, go to the topic, *Using globalsettings.properties*, located in the XHQ Administrator's Guide.

Tag Expression

Tags are currently supported by the **Time Series chart**. When entering a tag expression in the XHQ Workbench, the following are supported:

- A simple single reference tag expression as a string.
  Example: `"PT002"`
- A simple single reference as a non-string.
  Example: `r:PT002`
- A multiple reference tag expression as a string.
  Examples: `"PT002 + PT003"` or `"r:PT002 + r:PT003"`
- A multiple reference tag expression as a non-string.
  Example: `r:PT002 + r:PT003/2`

Only a tag expression that is a **simple, single reference** allows an interactive chart to be launched from a chart in a view.

Valid example: `"PT002"`

The following expression examples do not allow the interactive chart to launch properly.

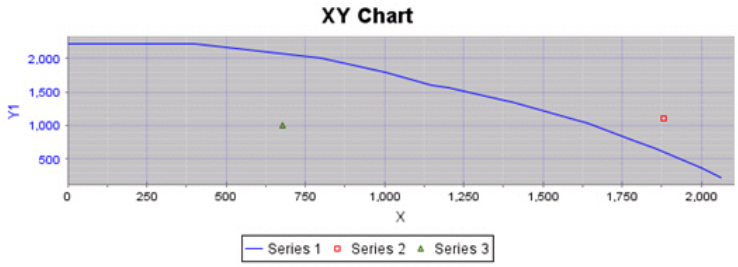
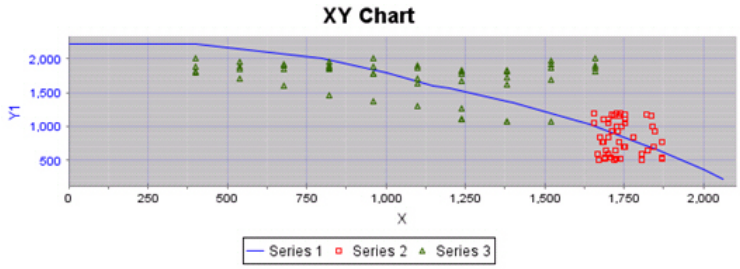
Invalid examples: `"r:PT002"` *or* `"PT002 + PT003"` or `r:PT002`

Tag Mapping

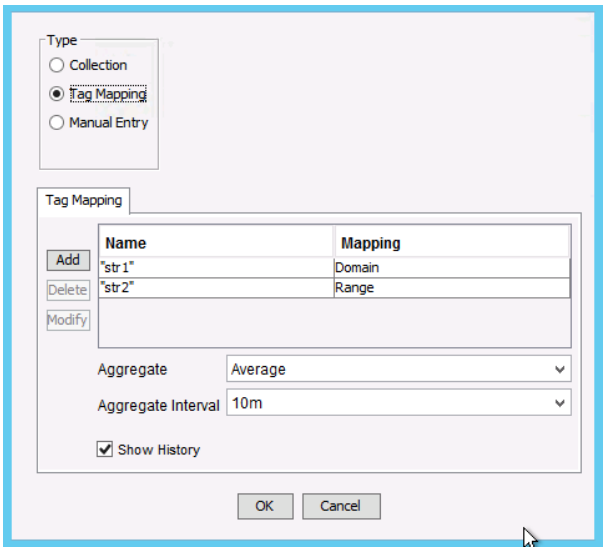
Tag Mapping allows you to plot a pair of time-correlated values (retrieved from two different tags) onto a single **XY** plot.

Real-time Charting is currently supported by the XY chart when the Tag Mapping type is used. These charts automatically update when viewing in real-time or whenever a collection, or tag, updates.

Modes of Tag Mapping

Actual Mode	 Only the latest point is shown on the plot. When a new point is received, it replaces the existing point.
Historical Mode	 All the points between a given time span are shown on the plot. The points are plotted using the selected symbol only; connecting lines are not used. The values of the tag pair can be configured to aggregate over a period of time. The aggregates available are: FITS, Average, Min, Max, Interpolate, and Raw. For non-time-sliced aggregate types (FITS and Raw), new points are added to the plot once they are received.

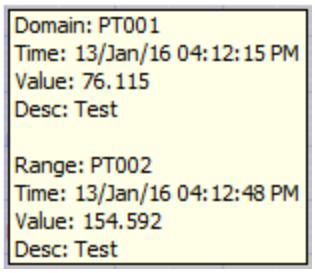
For steps on how to configure a Tag Mapping type, see the topic, *To add a series*.



Tag Mapping Configuration

Configuring the Tooltip

You can customize the Tag Mapping Tooltip that appears when a user hovers over a point at runtime.



The values displayed on the Tooltip can be customized by setting the global property, **`TagMappingTooltipFormat`**. The default value for the `TagMappingTooltipFormat` property is:

```
TagMappingTooltipFormat="Domain: %1$s <br> Time: %2$s <br> Value: %3$.3f <br> Desc:
%4$s<br><br>Range: %5$s <br> Time: %6$s <br> Value: %7$.3f <br> Desc: %8$s"
```

Note, the Tooltip values are displayed in a *specific order*:

1. Domain tag name (as String)
2. Domain time (as String)
   The time is formatted according to the globalsettings property `DefaultTimestampFormat`.
3. Domain value (as Double)
4. Domain description (as String)
5. Range tag name (as String)
6. Range time (as String)
   The time is formatted according to the globalsettings property `DefaultTimestampFormat`.
7. Range value (as Double)
8. Range description (as String)

For example, if you wanted to display only the tag name and the value in the Tooltip, you would set the `TagMappingTooltipFormat` property in the **`globalsettings.properties`** file to this:

```
TagMappingTooltipFormat="Domain: %1$s <br> Value: %3$.3f <br><br>Range: %5$s <br>
Value: %7$.3f"
```

For more information, go to the topic, *Using globalsettings.properties*, located in the XHQ Administrator's Guide.

Embedding Charts in Collection Views

At runtime, when launching a pop-up embedded chart within a collection view, the following **UI restrictions** occur:

- The **Path field** in the **Collection tab** for a series is ***disabled***.
  The actual path to the nested collection index that is being referenced cannot be described by a path. Regardless, the global collection that is being referenced is still visible.
- The series table **Add button** is ***disabled***.
  The valid path to a nested collection cannot be configured.
- The **Email button** is ***disabled***.
  This prevents an interactive chart from a nested collection from being e-mailed.
- The **Save Chart menu** is ***disabled*** to prevent an invalid configuration from being saved.

If a new chart type is selected from the **Chart menu**, then all the restrictions described above are set back to the default.

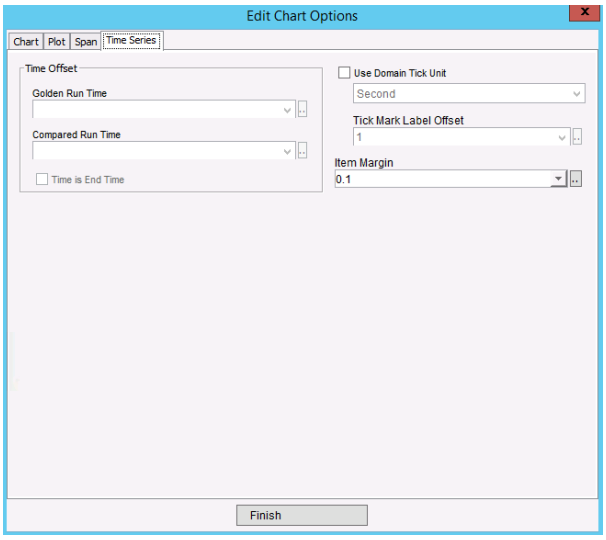
Time Offset Types

A Time Offset chart is a **Time Series chart** that has one series serving as the **Golden Run**, with the other remaining series as Compared Run that are offset by the Compared Run Time.

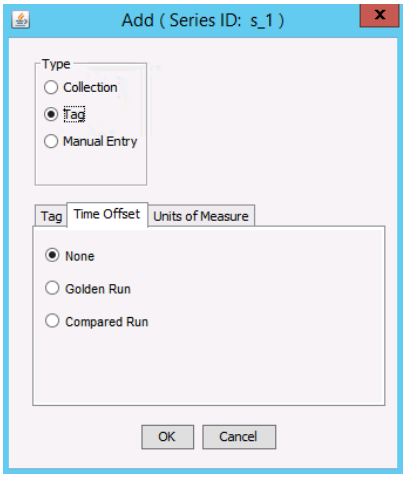
The minimum steps to configure a Time Offset chart are as follows:

1. Select a Time Series chart.
2. Add one or more series and select "Golden Run" for each of them in the Time Offset tab.
3. Add one or more series and select "Compared Run" for each of them in the Time Offset tab.
4. Open up Chart Options and go to the Time Series tab. Enter an expression for each of the Golden Run Time and Compared Run Time field. Select the "Time is End Time" if the time expressions are supposed to represent the "end time" instead of the default "start time".
5. Go to the Span tab (also under Chart Options) and enter a duration. This will be the duration used for the whole chart.

This chart only displays a single Domain axis for all Compared Run series. There is a hidden Domain axis that is used by the Golden Run series.



Time Configuration for Time Offset series (located in Chart Options)



Time Offset Configuration (located in Series configuration)

When configuring the Time Series chart, the Time Offset options are:

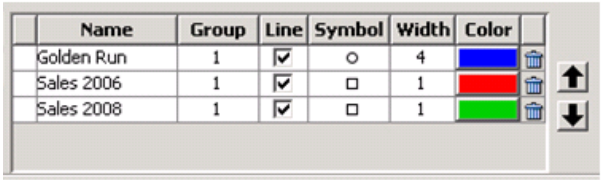
Option	Description
None	No time offset. This is the default.
Golden Run	Select this to specify the base series from which the other series are offset. This series must be the first series in the Series Table.
Compared Run	Select this to specify a series that is offset by a Compared Run amount of time from the Golden Run. The Compared Run Time is configured under the Time Series tab of Chart Options. A positive integer offsets by that amount after the Golden Run period. A negative integer offsets by that amount before to the Golden Run period.

EXAMPLE: SETTING THE TIME OFFSET

Given: The following series configurations.

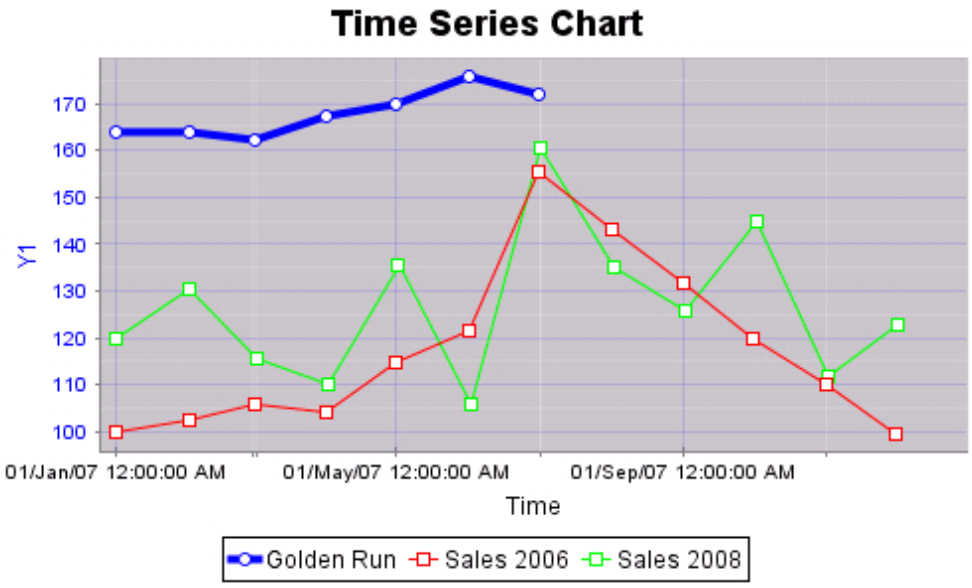
Name	Color	Symbol	Width	Time Offset Type
Golden Run	Blue	Circle	4	Golden Run
Compared Run	Red	Square	1	Compared Run

Results: Series Table:



Name	Group	Line	Symbol	Width	Color
Golden Run	1	☑	○	4	
Sales 2006	1	☑	□	1	
Sales 2008	1	☑	□	1	

Runtime Plot:



Setting the Tick Unit

The tick unit labels are actual point values on the domain axis (x-axis).

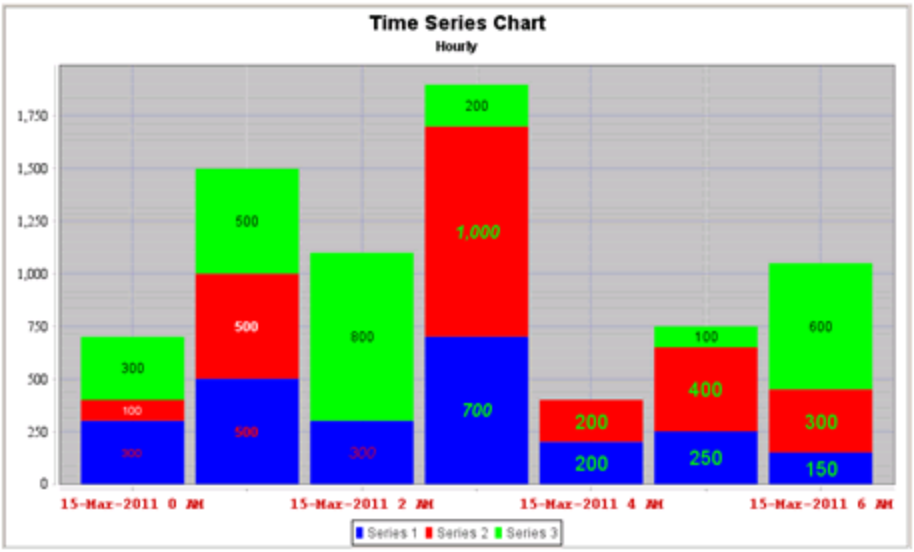
EXAMPLE: SETTING THE TICK UNIT FOR THE DOMAIN AXIS

Minor and other tick labels are not shown.

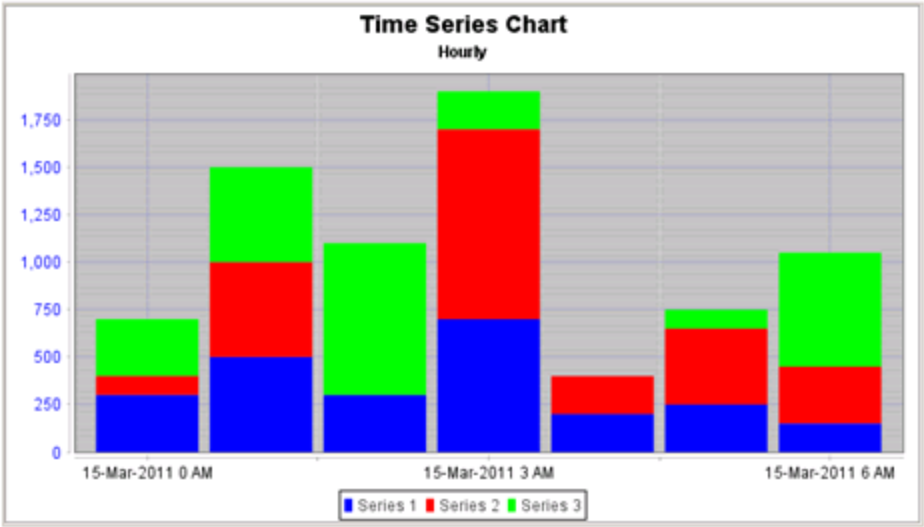
Procedure: To set the tick unit for the domain axis in a Time Series chart, do the following:

1. Create a **Times Series** chart.
2. Select the **XYStackedBar renderer**.
   (Note: Any of the time-series renderers will work.)
3. From **Chart Options**, click **Time Series** and check **Use Domain Tick Unit**.

   The tick unit drop-down list is enabled.
4. Select the tick unit appropriate for the dataset.
   This automatically assigns the default format for the selected tick unit.
5. *OPTIONAL*
   Select the **Tick Mark Label Offset** to space the labels accordingly for better display of the domain tick mark label and avoid overlap. The default value is 1, which means all labels are displayed. When set to greater than 1, the labels are offset by that value. For example, an offset of 2 means that every other label is displayed. The runtime chart looks like this, notice the domain label spacing:



Likewise, an offset value of 3 means every third label is displayed.



6. *OPTIONAL*
   To view the default Value format, click the **Chart tab**.

   The default Value format for each available tick unit is as follows:

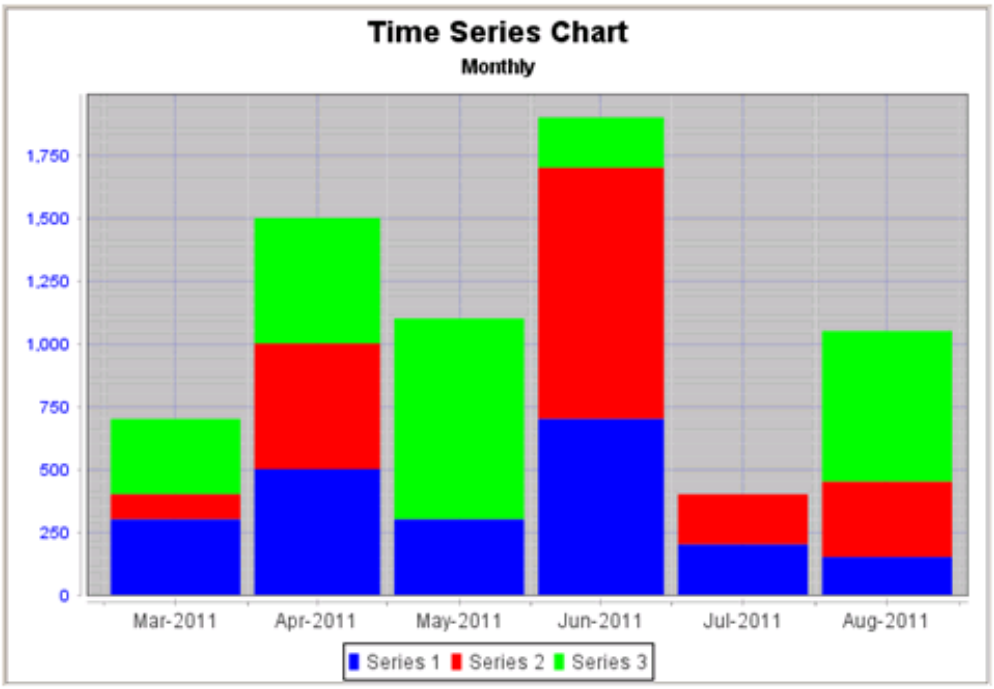
Unit	Format
Second	d-MMM-yyyy H:mm:ss a
Minute	d-MMM-yyyy H:mm a
Hour	d-MMM-yyyy H a
Day	d-MMM-yyyy
Month	MMM-yyyy
Year	yyyy

   If **Use Domain Tick Unit** is selected, then at runtime, the date portion of the tooltip is formatted using the this Value format.

7. *OPTIONAL*

   To edit the Value format, **enter a valid expression** in the **Value Format** box.
8. Click **OK**.
9. Set the **Span** for the chart.

Example: Runtime plot of a Monthly Time Series chart:


Time Series Chart
Monthly
1,750
1,500
1,250
1,000
750
500
250
0
Mar-2011
Apr-2011
May-2011
Jun-2011
Jul-2011
Aug-2011
Series 1
Series 2
Series 3

Setting the Units of Measure

In charting, the Units of Measure allows the current unit of a **Time Series chart** to be **converted** to another compatible unit that is within the same category.

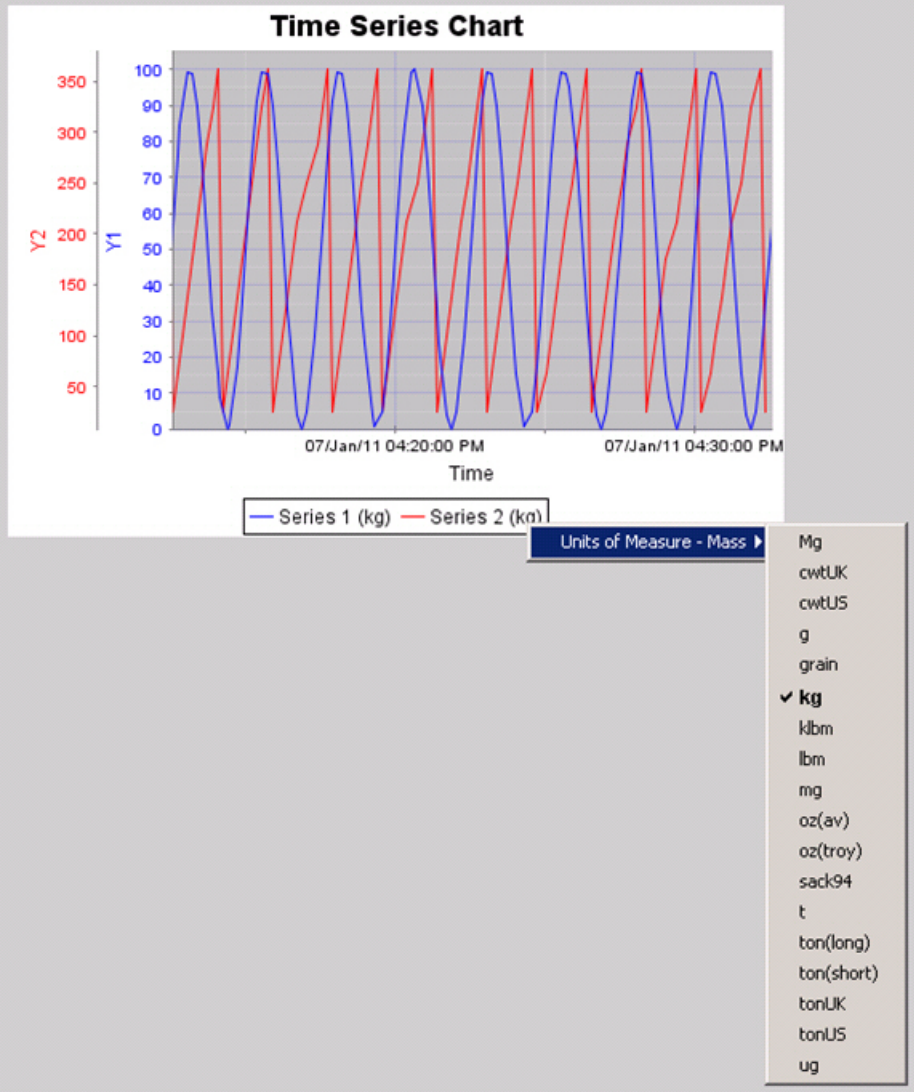
The Units of Measure for charting is similar to the Units of Measure for trending.

Additional Things to Note

- The Units of Measure feature is enabled only for the **Tag type** series.
- This feature requires you to interact with the Chart legend. Therefore, the **Show Legend** (and if the chart is embedded in a view, the **Allow Interaction with Chart**) option must be enabled (checked).

Viewing the Chart at Runtime

During runtime, you can access the **Units of Measure** option (if configured) by going to the chart **Legend**, mouse over the series name, and right-clicking. The Units of Measure option appears with a submenu listing the units that belong to the given category.



Example: Units of Measure

In the example above, the **Category** is Mass. The **Default Unit** is in bold, which in this case is kg (kilograms). The **Selected Unit** appears with a check mark.

EXAMPLE: CONFIGURING THE UNITS OF MEASURE

1. From a **Time Series** chart, **add** a new series.
   The "Add" dialog box appears.
2. For **Type**, click **Tag** and, from the Tag tab, enter a tag.

3. Click the **Units of Measure tab**.
4. Select a **Category**, the **From Unit**, and the **To Unit**.

   Selecting a non-default **From Unit** or **To Unit** value allows the initial chart to display that unit instead of the default. If you leave these options blank, then the chart initially displays the default unit from the backend.

5. Click **OK**.
6. OPTIONAL
   Repeat steps 1 to 5 for additional series.

   Tip
   For multiple series, set each series in its own separate group. This results in each series having its own Range axis, making it easier to see the effect of unit change at runtime.

7. Click **Chart Options**.
   The "Chart Options" dialog appears.
8. In the **Chart tab**, check **Show Legend**.

   The **Allow Interaction with Chart** option is enabled only for a chart that is embedded in a view.

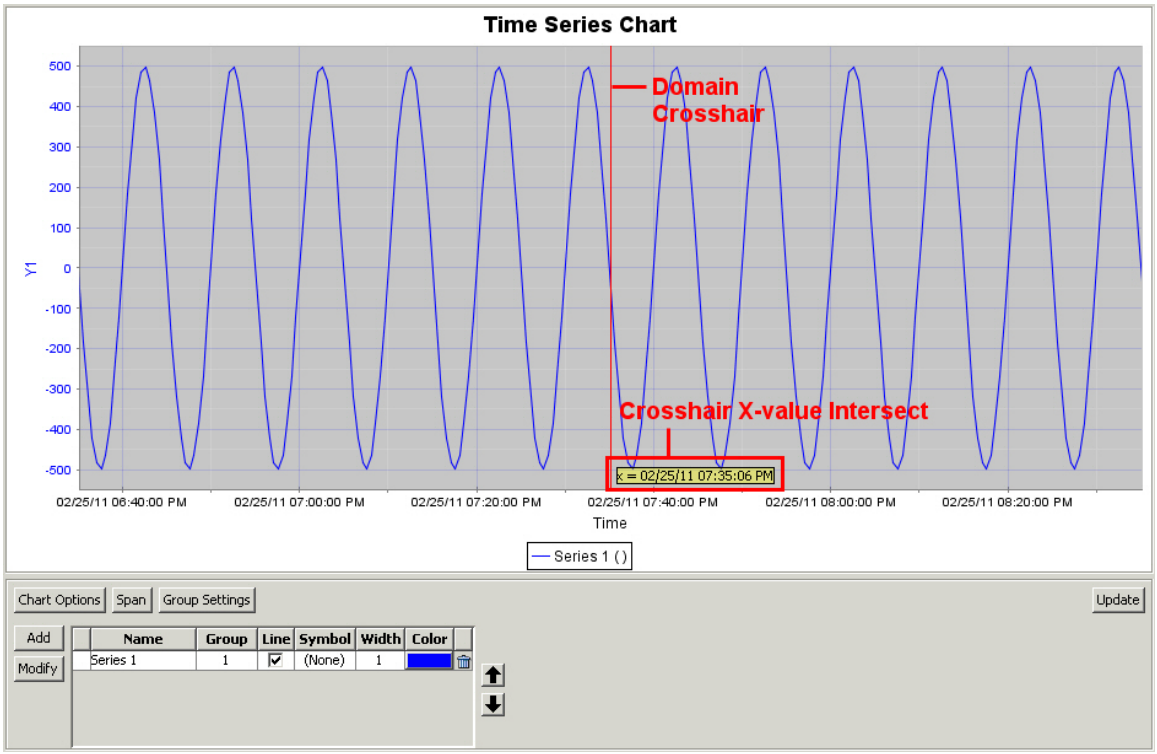
9. Click **OK**.
10. From the **Charts menu**, click **Save Chart**.
11. Next, open the chart view (runtime).
12. In the **Legend**, mouse-over the series name and right-click.
    The "Units of Measure" option appears with a submenu listing the units that belong to the given category.
13. Select a **new unit** to which you want to convert.
    The Legend shows the new unit for the series. The values on the series' Range axis is also converted to the new unit.

About the Domain Crosshair

The crosshair feature is available for the **Time Series** and **XY** charts in Interactive Chart mode. At runtime, the crosshair appears at the center of the plot, if the chart has valid data.

Additional Things to Note

- If the data is not valid, the crosshair does not appear.
- The X-value of where the crosshair intersects the plot appears on the chart area.
- This option is only used in the Interactive Chart mode. The crosshair is not saved when the chart is saved.



Example: Domain Crosshair

EXAMPLE: ENABLING THE DOMAIN CROSSHAIR

1. Create a Time Series chart and launch the Interactive Chart.
2. Add a series and map the necessary columns.
3. Click **Chart Options**.
   The "Chart Options" panel appears.
4. Click the **Plot tab**.
5. Check **Domain Crosshair**.
6. Click **OK**.

Using Expressions in Chart Configurations

Options that support expressions have a **Browse** button that launches the "Edit Expression" box.

You can also type a custom expression directly into the option box.

Literal string expressions must be enclosed in double quotation marks ("MyTitle").

Click the down arrow to display a list containing compatible members and variables for the supported expression type.

Color Expression

For color typed expression, the interface is different. In addition to the normal color button to bring up the color palette dialog, there is also an extra expression button to allow an expression to be set. The format for a color expression must either be a string like `#FFFFFF` or `#FFF` for hex format, or an integer.

If the color expression cannot be parsed to a valid color (for example, an expression such as `$EditBox1` versus a valid color like `#FFFFFF`), then the color button will change to show a dollar sign ($) to indicate that it is an expression and the actual color is evaluated at runtime.

The following options support expressions.

Option (Location)	Supported Expression Type
Title (Chart Options > Chart tab)	String
Subtitle (Chart Options > Chart tab)	String
Domain Axis Label (Chart Options > Chart tab)	String
Domain Value Format (Chart Options > Chart tab)	String
Chart Background Color (Chart Options > Chart tab)	String
Plot Background Color (Chart Options > Plot tab)	String
Grid Lines Color (Chart Options > Plot tab)	For Time Series, XY, Bar, Pareto, Polar, Histogram, Candlestick String
Minor Grid Lines Color (Chart Options > Plot tab)	For Time Series, XY, Bar, Pareto, Polar, Histogram, Candlestick String
Domain Min and Max Values (Chart Options > Chart tab)	Integer, Long, Float, Double, Decimal, DateTime, or Time Interval
Range Min and Max Values (Group Settings dialog box)	Integer, Long, Float, Double, Decimal, DateTime, or Time Interval
Range Axis Label (Group Settings dialog box)	String
Range Value Format	String

Option (Location)	Supported Expression Type
(Group Settings dialog box)	
Start and End Date (Chart Options > Span tab)	*Time Series chart only* DateTime
Duration (Chart Options > Span tab)	*Time Series chart only* DateTime
Tag Name (Add series dialog box > Tag Type)	*Time Series chart only* String
Tag Name (Add series dialog box > Tag Mapping Type)	*XY chart only* String
Series Name (Series table)	*String*
Bins (Chart Options > Histogram tab)	Non-zero positive Integer
Lower Bound, Upper Bound (Chart Options > Histogram tab)	Integer, Long, Float, Double, Decimal
The Treemap . . . **Item Background Color** **Item Label Color** **Category Background Color** **Category Border** **Category Label Color** **Exception Background Color** **Exception Label Color** **Color Table Exception Below Min Value Color** **Color Table Exception Above Max Value Color** (Chart Options > Treemap tab)	String
Series Mapping Column (Add series dialog box > Collection)	*Various types depending on mapping. See below for examples and restrictions.*
URL Link Address (Add series dialog box > Drill Down)	String *See below for examples and restrictions*
Action Link Function (Add series dialog box > Drill Down)	String *See below for example and restrictions*

Series Mapping Column expression

1. Constant mapping of Value.
   Examples: 9, pow(3,2), sqrt(81). All three examples of these examples result in a value of "9" for the value mapping. For Time Series chart, this will be a horizontal line located at 9.

   A **constant string expression** results in the given string for all values.

   Example 1: If you map the string "My tooltip" to the Tooltip mapping, at runtime, "My tooltip" appears as the tooltip for all entities.

   Example 2: If you map the String control variable `$EditBox1` to the Tooltip mapping, at runtime, whatever value you enter in that edit box will be the tooltip for all entities.
2. Other expression examples:
   a. **rV1** – simple expression mapping a collection column to Value for Time Series chart.
   b. **IFF($ComboBox1==1, rV1, rV2)** – a more involved expression like this can be mapped to Value for Time Series chart.
   c. **"Tooltip: " + sV3 + "<br>" + "Name: " + sV2** – this expression can be mapped to Tooltip for Treemap.
   d. **ReferenceToReal(tagName).at(dtT)** – this expression can be mapped to Value of the Time Series chart.

URL Link Address and Action Link Function expressions

1. Example of URL Link: "http://www.google.com/search?as_q=" + $Chart1.s_1.sV2.selected()
2. To reference the current selected item, regardless of chart type, use the function "selected()" as follows: $Chart1.s_1.column1.selected(), where "$Chart1" refers to the current chart, "s_1" is the series ID of the current series, and "column1" is the collection column to use. NOTE: Only the series ID of the current series being configured is allowed to be used in these type of expressions.

Date Format

Whenever a date, or time, is displayed in a chart, its format is determined by the global setting properties: **`DefaultDateOnlyFormat`** or **`DefaultTimeOnlyformat`**, respectively.

For more information on global properties, refer to the section, *Working with .PROPERTIES Files*, located in the XHQ Administrator's Guide.

Future Date Support for the Time Series Chart

The Charting object's Time Series chart supports future dates.

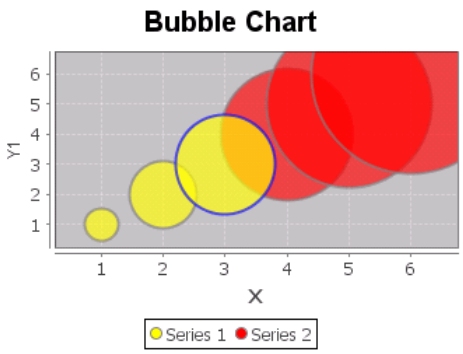
Things to Note

- If the chart detects that T2 (the end time) of the time range is **greater than** the current time (that is, a time in the future), then it retrieves all available data within that range, including future data. This applies to both collection-based and tag-based series.
- During "future data" mode, no series are being subscribed. Therefore, the chart does not dynamically update.

Drill-down Series

Drill-down is available for all the charts except the Histogram chart. For charts with multiple series, you can configure a different drill-down for each series you add.

Viewing the Drill-down at Runtime



Drill-down Example (with default link)

Depending on the chart, when you hover over a series entity at runtime (with drill-down configured and the default link set), either a **blue outline** appears around it or the given entity **expands** to a larger size. These two visual cues indicate a drill-down hotspot. Click on this hotspot to drill-down. Right-click on the item and a shortcut menu appears, listing all the drill-down links for the series entity.

Chart Hotspot Action

Chart Type	Hotspot Action	Requirements
Bar	Blue outline around entity (both Category- and XY-based)	The Domain column must be sorted "ASC".
Bubble	Blue outline around entity	N/A
Candlestick	Blue outline around entity	Time column must sort "ASC".
Gantt	Blue outline around entity	N/A
Pareto	Blue outline around entity	N/A
Pie	Pie slice expands	N/A
Polar	Symbol expands	Must enable symbol. Domain column must sort "ASC".
Spider web	Symbol expands	Must enable symbol.
Time Series	Symbol expands	Must enable symbol and Time column must sort "ASC".
Treemap	Blue outline around entity	N/A
Waterfall	Blue outline around entity	N/A
XY	Symbol expands	Must enable symbol and Domain column must sort "ASC".

Configuring a Drill-down

When adding a series to a chart (with the exception of the Histogram chart), you can set the drill-down option.

Option	Description
Do Nothing	No drill-down. This is the default.
Update Dependent Items	Updates items within the view that are dependent on the selected series entity. This is done by adding a Value to the same view, and configuring its Value animation to the name that is assigned to the chart (for example, `$Chart1`). **Tip:** The assigned chart name can be found in the General tab.
Links	Configures links for the series. **Note:** The UI to configure a link is exactly the same as for the Links tab in the XHQ Workbench.

Once the drill-down option is selected, open **Chart Options** and, in the Chart tab, enable (check) **Allow Interaction with Chart**.

To configure a drill-down

1. Select a chart (with the exception of the Histogram chart).
2. Click **Add** to add a new series.
   The "Add" dialog box appears.
3. In the **Collection tab**, select a **Collection** and map the columns.
4. Click the **Drill Down tab**.
5. Do one of the following.
   - For a **Do Nothing** series:
     a. Click **OK** and **Save** the view.

   or

   - For an **Update Dependent Items** series:
     a. Click **Update Dependent Items**.
     b. Click **OK**.
     c. Within the same view, add a **Value item**.
     d. Click the **Animation tab**.
     e. Click **Add** and select **Value** from the drop-down list.
     f. In the Expressions box, select the expression that corresponds to the assigned chart name.

     Tip
     To determine the assigned chart name, select the chart and click the General tab. Note the name given to the chart.

     Example: To add a dependent item, in the same view, add a Value item and configure its Value animation to an expression similar to **`$Chart1.SeriesID.member.selected()`**.

Updating Dependent Expressions
The **`$Chartx.SeriesID`** expression prefix is resolved to a collection that is configured for the series (as specified by `SeriesID`). Any aggregate function, such as `count( )`, that can be applied against a collection can be used.

For additional information on entering expressions, go to the topic, *Using Expressions in Chart Configurations*.

g. **Save** the view.

or

- To add **Links** to the series:
  a. Click on **Links.**
  b. Enter a name for link in the Name field.
  c. Select a link type: URL, View, or Action.
  d. Enter the required field(s) for the selected link type: URL – Address, View – View and Mode, Action – Function.
  e. Select **Set as Default** if the link is to be the default link. Selecting this option for one of the links will allow the blue highlight around the series item to be shown when hovering over it.
  f. Click **Add** to add the link to the Links table.
  g. Repeat steps b) through e) to add additional links.
  h. Click **OK** and **Save** the view.

EXAMPLE: CONFIGURING A DRILL-DOWN FOR A BUBBLE CHART

Given: The following series configurations.

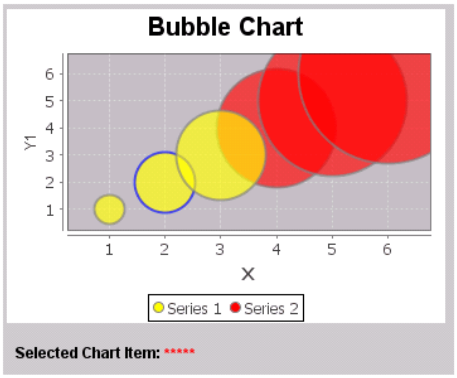
Name	Color	Drill Down Type	View
Series 1	Yellow	Links/View type Set as Default	BubbleRow (In this case, this view is configured to pop-up.)
Series 2	Red	Update Dependent Items	N/A

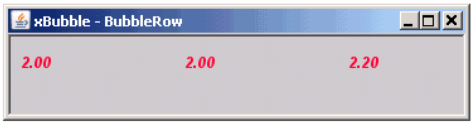
Data:

Series 1			Series 2		
X	Y	Z	X	Y	Z
1	1	1.1	4	4	4.4
2	2	2.2	5	5	5.5
3	3	3.3	6	6	6.6

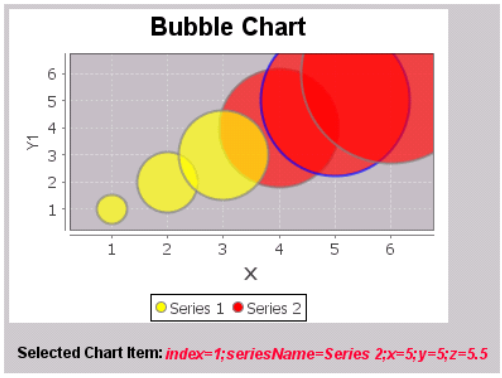
Results:

At runtime, drilling down on a **yellow** bubble results in the view, **BubbleRow**, popping up with the correct bubble context passed to it.



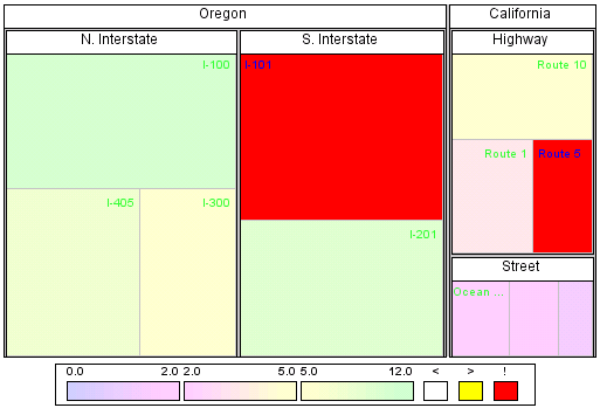


Drilling down on a **red** bubble updates the value item that is within the view. This value item is a text string describing the selected bubble.



About the Treemap Chart

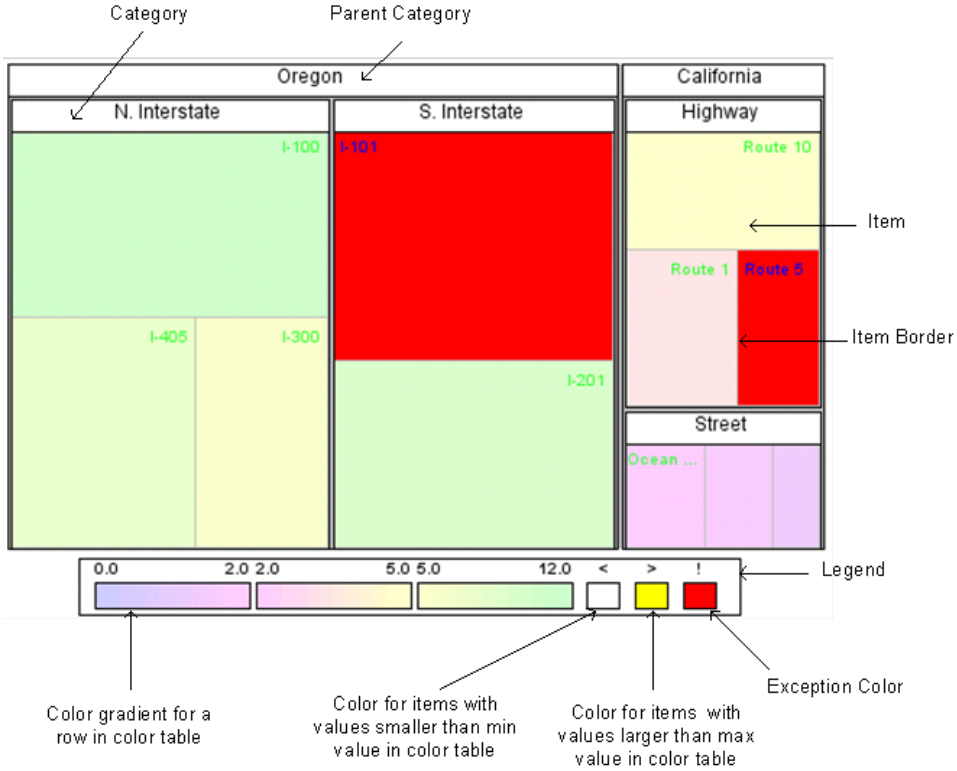
Treemap charts are used to display large hierarchical collections of quantitative (tree-structured) data using nested rectangles.



Treemap Chart Example

A treemap is generated by dividing the display area into a nested sequence of rectangles, whose areas correspond to an attribute of the data set. The area of the nested rectangles represents the magnitude of an attribute. It can be used to visualize a quantity, such as the production or flow rates. The area of the enclosing rectangle represents the total, or sum of, all the values of the nested (smallest) rectangles. Color is added to represent a certain condition or state.

The treemap chart uses a *split-squarified* layout approach, which draws the rectangles using the best aspect ratio.



Note the color gradients in the legend of the treemap example above. The spectrum of colors add a second dimension to be measured. In this case, the first dimension is determined by the size of the nested rectangles. The second dimension specifies the spectrum of colors to be displayed. Common practices involve darker colors for larger values and lighter colors for smaller values. Also, the darker-to-lighter (or vice versa) color representation can represent degrees of "good" or "bad".

How to use Treemaps

In general, treemaps are used to pick or spot a particular condition or status from a large amount of data items.

A treemap should not be used to make fine quantitative comparison between items.

The following are some real-world examples of how treemaps can be used.

EXAMPLE 1

A treemap could represent a number of compressors grouped in a hierarchy of fields. Each rectangle corresponds to a compressor. The size of the box is proportional to the flow rate, and the color represents any abnormal condition.

EXAMPLE 2

A treemap could be used in monitoring the daily production of a certain number of wells, group cluster, or field. Each box in the treemap represents a well. The size of the nested (smallest) rectangles is proportional to the number of barrels of oil produced each day. The color representation indicates if a well is below, equal to, or above the target.

Default Treemap Navigation

The treemap has built-in navigation to either drill down into or up onto the categories. For example:

- Clicking on a category panel that is currently the only top-level category drills up one level.
- Clicking on a category panel that is currently not the only top-level category drills down to that category.

About 3D Charts

The following 3D charts are available:

- Line
- Bar
- Area
- Pie
- Stacked Bar
- Scatter Plot

The configuration and use of these charts are similar to the existing 2D charts, like the XY and Bar charts. For 2D charts, however, only two data components are required (for example, Value and Category). For 3D charts, there is a third component. The exception to this rule is the 3D Pie chart, which still only needs two mapping components.

The Plots

3D charts are divided into three plot groups, depending on the type of data that is needed to populate them:

- The Category Plot
- The XYZ Plot
- The Pie Plot

The Category Plot

This includes Line, Bar, Area, and Stacked Bar charts.

Required Data	Source	Chart Axis Name
Value	Series mapping	Value
Category	Series mapping	Column
Series Name	Name of the series	Row

The Series Name also appears in the legend.

The XYZ Plot

This is in the form of a Scatter Plot.

Required Data	Source	Chart Axis Name
X	Series mapping	X
Y	Series mapping	Y
Z	Series mapping	Z

The Series Name for each series does not appear on the plot itself, but does appear in the legend.

The Pie Plot

This includes the Pie chart.

Required Data	Source
Key	Series mapping
Value	Series mapping

Since there are no axes for the Pie chart, the Series Name is not visible anywhere on this chart.

Setting Chart Options for 3D Charts

From the **Chart** properties > **Chart Type**, select a 3D chart and click **Chart Options**.

The following general settings are available for all 3D charts.

Option	Description
Title	The chart title.
Subtitle	The chart subtitle, which appears below the title.
Default View Point	These options allow you to set how the chart appears when it is first loaded. *View Angle Options*: • **Above** A view point looking at the chart from the front and above. • **Above Left** A view point looking at the chart from the front and above to the left. • **Above Right** A view point looking at the chart from the front and above to the right. *View Distance Options*: • **Zoom to Fit** Adjusts the viewing distance so that the chart fits the current panel size. To use this option, the Zoom Toolbar must be enabled. • **Distance** A positive number that represents the distance between the viewing point and the origin.
Chart Background Color	The background color of the chart.
Plot Background Color	The background color of the plot.
Show Legend	Check the box to show the Legend. Then specify where to place it: Bottom, Left, Right, or Top.
Show Zoom Toolbar	Check the box to enable the Zoom Toolbar, which will appear on the right side of the chart. This toolbar is used to control zooming, panning, and so forth.
Allow Interaction with Chart	Check the box to allow the user, at runtime, to interact with the chart (such as zooming, panning, and so forth).

Defining Chart Axes

The 3D Chart Axes tab is different for the Category and Scatter plot groups.

For Category Plot charts

Option	Description
Axes	Displays the axes configuration. *Options*: • **Show Value Axis** Check the box to show the Value axis. • **Show Column Axis** Check the box to show the Column axis. • **Show Row Axis** Check the box to show the Row axis.

Value	Configuration for the Value axis.
	Options:
	• **Axis Label**
	The label for the Value axis.
	• **Value Format**
	The format of the values that appear on the tick marks.
	• **Minimum Value**
	The minimum value on the Value axis.
	• **Maximum Value**
	The maximum value on the Value axis
	• **Grid Lines**
	To show grid lines, check the box to enable the following options:
	  ◦ **Grid Width**
	  Enter the grid line width. The default value is 0.2.
	  ◦ **Grid Color**
	  Click the color block and set the grid line color. The default color is blue.

Column	Configuration for the Column axis.
	Options:
	• **Axis Label**
	The label for the Column axis.
	• **Grid Lines**
	To show grid lines, check the box to enable the following options:
	  ◦ **Grid Width**
	  Enter the grid line width. The default value is 0.2.
	  ◦ **Grid Color**
	  Click the color block and set the grid line color. The default color is blue.

Row	Configuration for the Row axis.

Option	Description

	Options:
	• **Axis Label**
	The label for the Row axis.
	• **Grid Lines**
	To show grid lines, check the box to enable the following options:
	  ◦ **Grid Width**
	  Enter the grid line width. The default value is 0.2.
	  ◦ **Grid Color**
	  Click the color block and set the grid line color. The default color is blue.

For Scatter Plot

The chart axes configuration screen is similar to the Category Plot screen, except the axes labeling has been changed to “X”, “Y”, and “Z”. Since all three axes are numeric-based, settings such as Value Format, Minimum Value, and Maximum Value can apply to all three axes.

Option	Description
Axes	Displays the axes configuration. *Options*: • **Show X Axis** Check the box to show the X axis. • **Show Y Axis** Check the box to show the Y axis. • **Show Z Axis** Check the box to show the Z axis.

X	Configuration for the X axis.
	Options:
	• **Axis Label**
	The label for the X axis.
	• **Value Format**
	The format of the values that appear on the tick marks.
	• **Minimum Value**
	The minimum value on the X axis.
	• **Maximum Value**
	The maximum value on the X axis
	• **Grid Lines**
	To show grid lines, check the box to enable the following options:
	  ◦ **Grid Width**
	  Enter the grid line width. The default value is 0.2.
	  ◦ **Grid Color**
	  Click the color block and set the grid line color. The default color is blue.

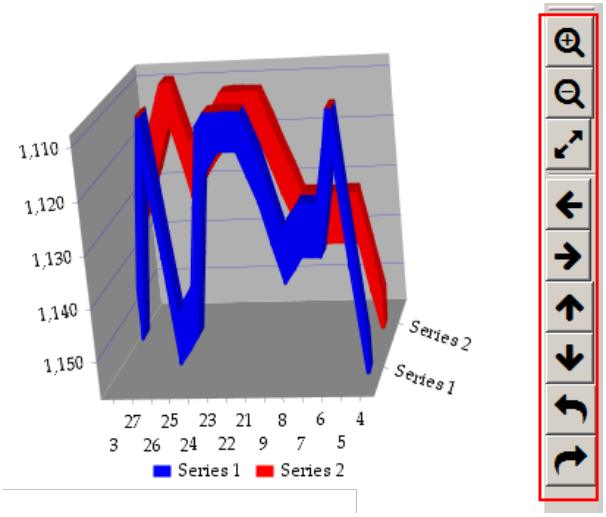
Option	Description

Y	Configuration for the Y axis.
	Options:
	• **Axis Label**
	The label for the Y axis.
	• **Value Format**
	The format of the values that appear on the tick marks.
	• **Minimum Value**
	The minimum value on the Y axis.
	• **Maximum Value**
	The maximum value on the Y axis
	• **Grid Lines**
	To show grid lines, check the box to enable the following options:
	  ◦ **Grid Width**
	  Enter the grid line width. The default value is 0.2.
	  ◦ **Grid Color**
	  Click the color block and set the grid line color. The default color is blue.

Z	Configuration for the Z axis.
	Options:
	• **Axis Label**
	The label for the Z axis.
	• **Value Format**
	The format of the values that appear on the tick marks.
	• **Minimum Value**
	The minimum value on the Z axis.
	• **Maximum Value**
	The maximum value on the Z axis
	• **Grid Lines**
	To show grid lines, check the box to enable the following options:
	  ◦ **Grid Width**
	  Enter the grid line width. The default value is 0.2.
	  ◦ **Grid Color**
	  Click the color block and set the grid line color. The default color is blue.

Using the Zoom Toolbar at Runtime

If enabled, the **Zoom Toolbar** appears (by default) on the right side of the chart.



For information on how to enable the Zoom Toolbar, go to the topic, *Setting Chart Options for 3D Charts*.

At runtime, the user can move this tool bar to any of the other edges by dragging the top of the tool bar and placing it to the preferred edge. The user may also move the toolbar to its own window by dragging it outside of the chart area.

Hovering of a toolbar button displays a default tooltip.

Zooming In or Out

From the Zoom Toolbar, click .

If the toolbar is not enabled, you may also:

- Click the image to select it and then use the mouse wheel to zoom in or out;
- Or right-click the image to show the pop-up menu and select **Zoom In** or **Zoom Out**.

Zoom To Fit

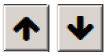
From the Zoom Toolbar, click .

If the toolbar is not enabled, you may also right-click the image to show the pop-up menu and select **Zoom To Fit**.

Rotate Left or Right

From the Zoom Toolbar, click to rotate left or right (about a vertical axis). If the toolbar is disabled, the same effect may be achieved by clicking on the image, holding down on the mouse, and dragging to the left or the right.

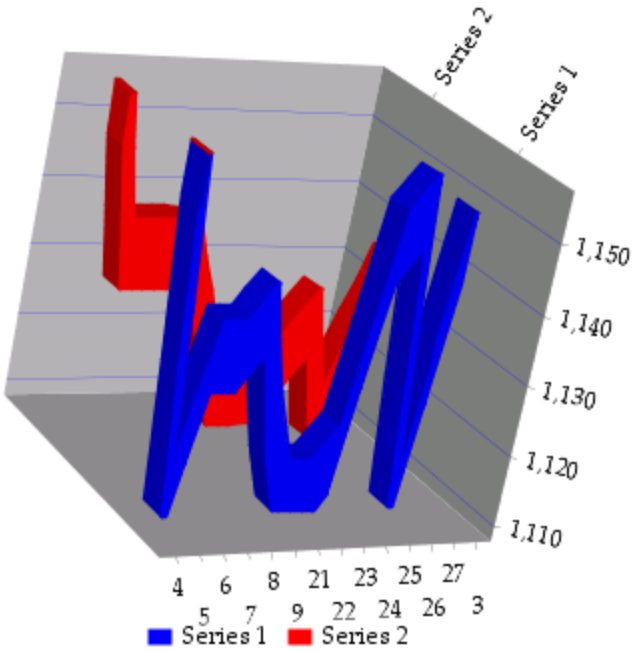
Rotate Up or Down

From the Zoom Toolbar, click  to rotate left or right (about a horizontal axis). If the toolbar is disabled, the same effect may be achieved by clicking on the image, holding down on the mouse, and dragging upward or downward.

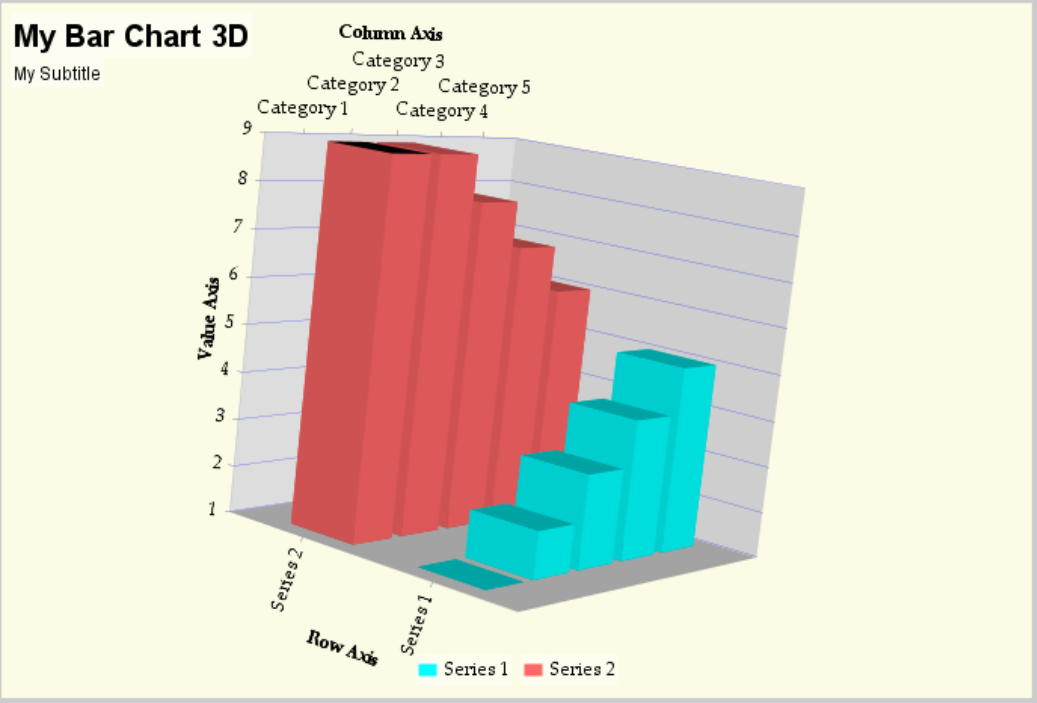
Roll Clockwise or Counter-clockwise

This action is only possible with the Zoom Toolbar by clicking . The image spins around the center point clockwise or counter-clockwise.

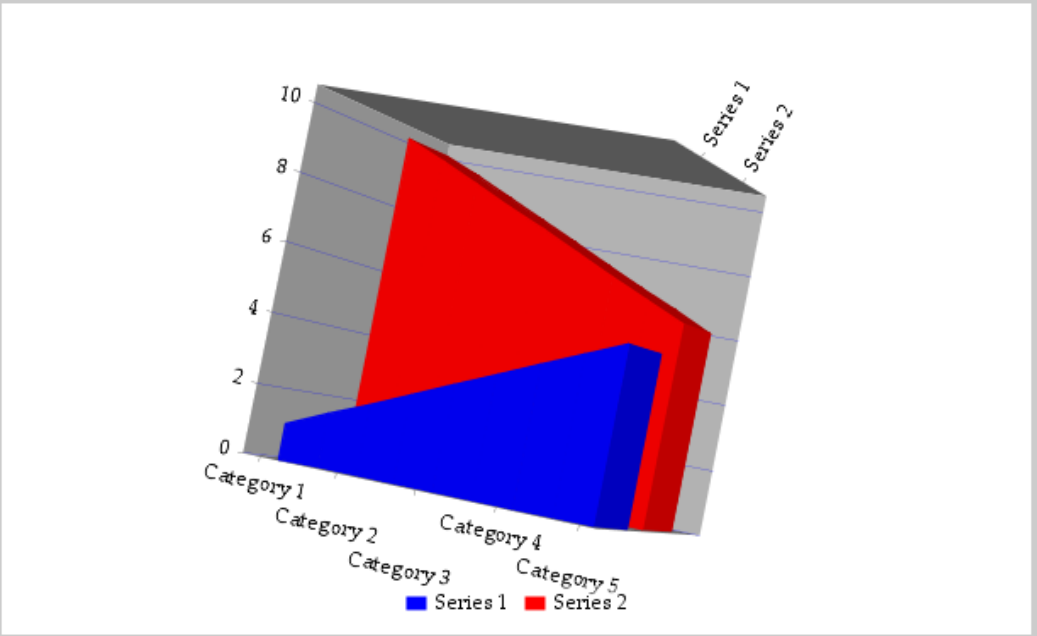
Examples of 3D Charts



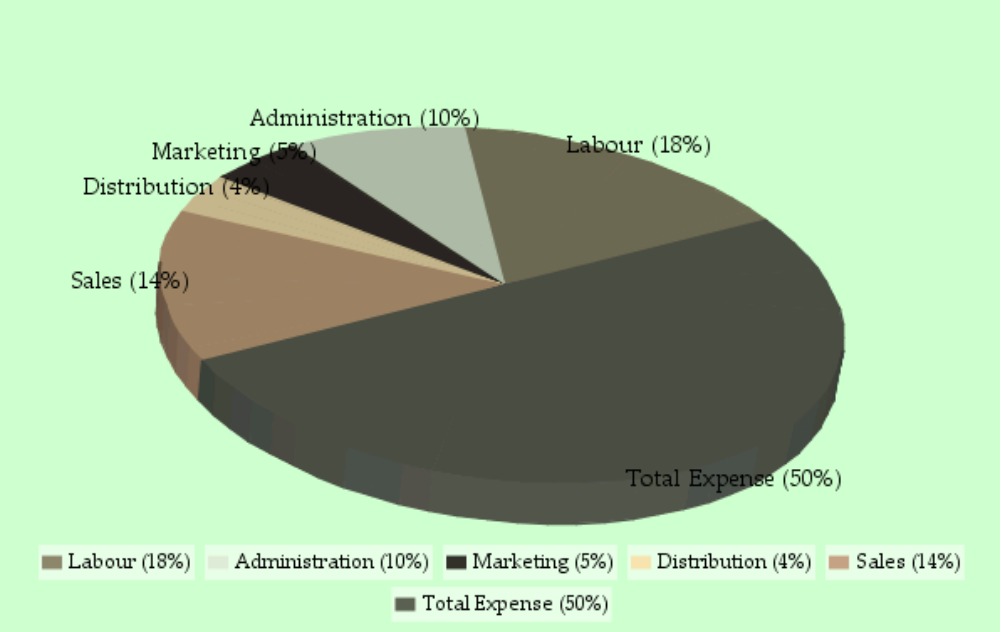
Line chart



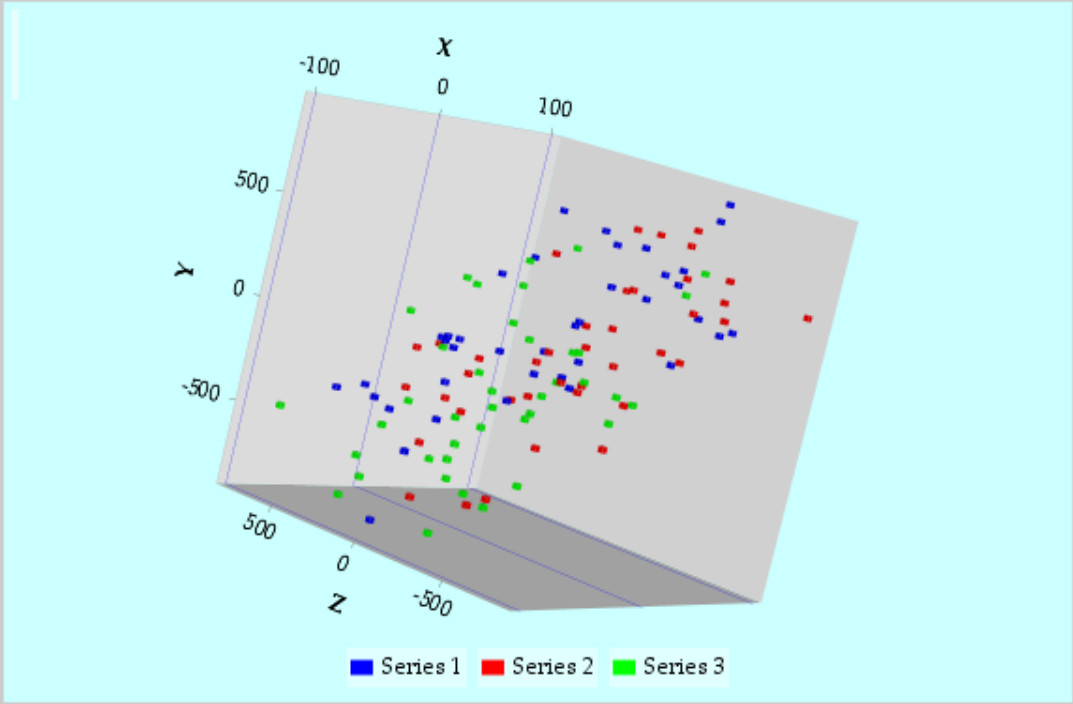
Bar Chart



Area Chart



Pie Chart



Scatter Plot Chart

Inserting the Table Control

The Table control is used to present XHQ collection member data in a generic, tabular form. It fully supports XHQ expressions to configure the data source and filtering criteria. It can pass selected row context to another draw item by using table control view variables.

Using the Table control has several advantages over using the embedded view template to build collection views. For example:

- The Table control is more flexible. It does not need to build a view template to simply show record data in text.
- The Table has a better look-and-feel and supports common table control user interactions.
- The Table makes it simple to construct record browser and master-slave table representations.

See also *Using an ENO Group as the Data Source*.

Configuring the Table Control

The Table Control properties consist of the following:

- A **Data Source** property
- A list of **Column** properties

  There are three types of table columns: **Member Mapping** , **Calculated Column**, and **Template View** .

- A list of **Data Criteria** properties

  There are three types of data criteria: **Table Filtering**, **Table Sort**, and **Table Trigger**.

- A list of **Table UI** and **Color** properties

Data Source

Data Source Properties

Property	Description	Type
CollectionMember	The collection member reference that is used to populate the table.	MemberReference

Member Mapping Table Column

Member Mapping Column Properties

Property	Description	Type
Member Mapping	This property sets the primitive data member in the collection member used for the column. This is only valid when the column type is "Member Mapping".	MemberReference

Property	Description	Type
Display Name	The column title. This defaults to the primitive data member name.	XHQ Expression
Column Width	The column width in pixels (px). If "AutoFit" is set to true, then the column width is automatically adjusted to fit the longest text in the column. The default column width is 100 pixels.	Integer
Header Alignment	This property overrides the default alignment for the data in the column. The default header alignment is left.	Enumeration: • CenterAlignment • LeftAlignment • RightAlignment
Header Font Name	This property allows the user to select a font name (from a list of fonts) for the column header. By default, the available fonts are Courier, Helvetica, and Times Roman.	String
Header Font Style	This property allows the user to select a font style (from a list of font styles) for the column header. The available font styles are Plain, Bold, and Italic.	Enumeration
Header Font Size	This property allows the user to select a font size for the column header.	Integer
Header Text Color	This sets the text color for the column header.	RGB color
Data Alignment	This property overrides the default data alignment in the column. By default, numerical values are right aligned; all else, are left aligned.	Enumeration: • CenterAlignment • LeftAlignment • RightAlignment
Data Font Name	This property allows the user to select a font name (from a list of fonts) for the column data. By default, the available fonts are Courier, Helvetica, and Times Roman.	String
Data Font Style	This property allows the user to select a font style (from a list of font styles) for the column data. The available font styles are Plain, Bold, and Italic.	Enumeration
Data Font Size	This property allows the user to select a font size for the column data.	Integer
Data Text Color	This sets the text color for the column data.	RGB color
Formatting Pattern	When specified, this property defines the format for numerical and data/time values. For example, "#0.000" can be used for float values.	String

Calculated Table Column

This table column uses an XHQ expression to configure the Value Expression property. The expression uses the collection row value as context and supports aggregate functions, such as `rRef(TagName).avg(DateAdd("hour",-12),Now())`.

Calculated columns cannot be sorted.

Calculated Column Properties

Property	Description	Type
Value Expression	The calculated value.	XHQ Expression
Display Name	The column title. This defaults to the primitive data member name.	XHQ Expression
Width	The column width in pixels (px). If "AutoFit" is set to true, then the column width is automatically adjusted to fit the longest text in the column. The default column width is 100 pixels.	Integer
Header Alignment	This property overrides the default alignment for the data in the column. The default header alignment is left.	Enumeration: • CenterAlignment • LeftAlignment • RightAlignment
Header Font Name	This property allows the user to select a font name (from a list of fonts) for the column header. By default, the available fonts are Courier, Helvetica, and Times Roman.	String
Header Font Style	This property allows the user to select a font style (from a list of font styles) for the column header. The available font styles are Plain, Bold, and Italic.	Enumeration
Header Font Size	This property allows the user to select a font size for the column header.	Integer
Header Text Color	This sets the text color for the column header.	RGB color
Data Alignment	This property overrides the default data alignment in the column. By default, numerical values are right aligned; all else, are left aligned.	Enumeration: • CenterAlignment • LeftAlignment • RightAlignment
Data Font Name	This property allows the user to select a font name (from a list of fonts) for the column data. By default, the available fonts are Courier, Helvetica, and Times Roman.	String
Data Font Style	This property allows the user to select a font style (from a	Enumeration

Property	Description	Type
	list of font styles) for the column data. The available font styles are Plain, Bold, and Italic.	
Data Font Size	This property allows the user to select a font size for the column data.	Integer
Data Text Color	This sets the text color for the column data.	RGB color
Formatting Pattern	When specified, this property defines the format for numerical and data/time values. For example, "#0.000" can be used for float values.	String

Template View Table Column

A template view column uses a component view to render the cells in the column. If the selected component is the same as the component collection for the table, the collection row context is used to populate the view. If the selected component is not the same as the component collection, the template view member mapping is used to drive the expressions used in the template view.

These properties are the same as those used in the embedded collection view.

Template View Column Properties

Property	Description	Type
Sort By	The **collection member** is used to sort the template view column.	N/A
Display Name	The column title. This defaults to the primitive data member name.	XHQ Expression
Column Width	The column width in pixels (px). If "AutoFit" is set to true, then the column width is automatically adjusted to fit the longest text in the column. The default column width is 100 pixels.	Integer
Header Alignment	This property overrides the default alignment for the data in the column. The default header alignment is left.	Enumeration: • CenterAlignment • LeftAlignment • RightAlignment
Header Font Name	This property allows the user to select a font name (from a list of fonts) for the column header. By default, the available fonts are Courier, Helvetica, and Times Roman.	String
Header Font Style	This property allows the user to select a font style (from a list of font styles) for the column header.	Enumeration

Property	Description	Type
	The available font styles are Plain, Bold, and Italic.	
Header Font Size	This property allows the user to select a font size for the column header.	Integer
Header Text Color	This sets the text color for the column header.	RGB color
Component Name	The name of the component for the template view.	String
View Name	The template view name.	String
Template View Member Mapping List	This property allows the user to configure the template view member mapping.	Enumeration

Data Criteria

Table Filtering Criteria Properties

Property	Description	Type
Data Member	This property sets the primitive data member used in the filtering criteria.	MemberReference
Relation	This property sets the op code in the filtering criteria.	Enumeration
Expression	If this applies, this property sets value in the filtering criteria.	XHQ Expression

These properties are the same as those used in the embedded collection view.

Table Sort Criteria Properties

Property	Description	Type
Data Member	This property sets the primitive data member used in the filtering criteria.	MemberReference
Relation	This property sets the sort.	Enumeration
Expression	This property sets the sort direction.	Enumeration: • Ascending • Descending

In this case, sorting is **case-sensitive**. By default, all filtering criteria are case-sensitive. This applies to the filtering criteria of collection views, the table control, and the Excel item. To change the behavior, set the global setting property `FilterCaseSensitive` to false.

These properties are the same as those used in the embedded collection view.

Table Trigger Criteria Properties

Property	Description	Type
Expression	This property sets the control that triggers the table to retrieve explicitly.	View Variable

Table UI Properties

Table User Interface and Miscellaneous Properties

Property	Description	Type
Navigation Type	This property sets the navigation type that is used for the grid. The default is Scrolling. • **Paging** Navigation buttons appear at the bottom of the table as well as information about the current page and total number of pages. • **Scrolling** The vertical scroll bar appears when it is necessary to scroll through the whole data set.	Enumeration
Page Size	This property is enabled when the Navigation Type is set to Paging. If it is set to the default value of 0 (zero), the number of rows per page is calculated at runtime to be the number of rows visible in the table. In Paging mode, this property allows the number of rows per page to be explicitly set. The range of valid values is from 0 to 500.	Integer
Enable Row Selection	If true, the rows in the table can be selected during runtime. During runtime, when you mouse-over a row, a blue border appears. All dependent items are updated when a row is selected.	Boolean
Auto Fit	If true, each column width is automatically adjusted to fit the longest text in the column. The default is false. **When Excel uses a Table control as the data source** If the table control **Auto Fit** is enabled (checked), then all columns are resized to fit the data. If the table control **Auto Fit** is disabled (not checked), then all columns are sized to the column width configuration. Furthermore, if the table control **Auto Fit** is disabled AND **Wrap Text** is enabled (checked), then all columns wrap	Boolean

Property	Description	Type
	text as needed.	
Row Height	This configures the height in pixels. The default is 20 pixels.	Integer
Wrap Text	This is enabled when Auto Fit is set to false. If true, the row height is automatically adjusted to fit the longest text in the row. The default is false. **About Wrapping Text** Setting the Wrap Text table property to true enables the text to automatically wrap (or appear in multiple lines) within the table cell. Note, however, the cell content data type must be of the type String. For all other data types (such as Double, Integer, and so forth) wrapping is ignored.	Boolean

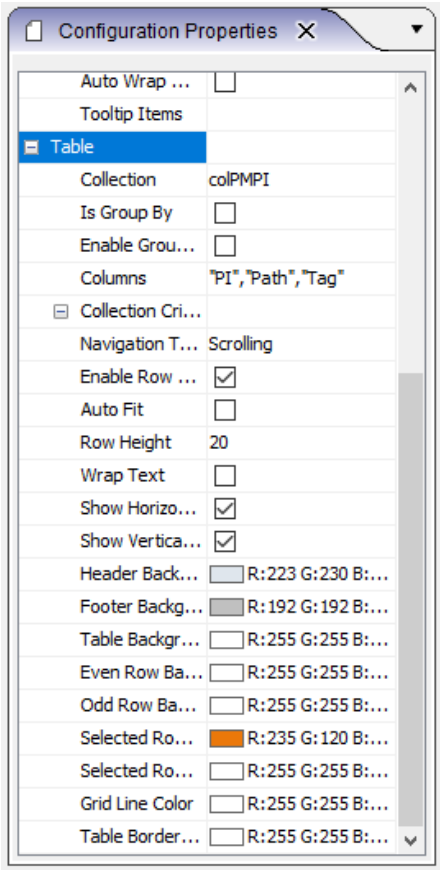
Table Color and Miscellaneous Properties

Property	Description	Type
Table Background Color	The default is SystemColor.control.	Color
Header Background Color	The default is SystemColor.control.	Color
Footer Background Color	The default is SystemColor.control.	Color
Selected Row Background Color	The default is SystemColor.textHighlight.	Color
Selected Row Text Color	The default is SystemColor.textHighlightText.	Color
Table Border Line Color	The default is white.	Color
Grid Line Color	The default is gray.	Color
Show Horizontal Grid Line	The default is true.	Boolean
Show Vertical Grid Line	The default is true.	Boolean
Even Row Background Color	The default is SystemColor.white.	Color
Odd Row Background Color	The default is SystemColor.white.	Color

To configure a table control

1. From the **Tool palette**, click the **Table** tool .
2. Click in the drawing area and **drag diagonally** to draw the control, releasing the mouse when the image reaches the desired size.

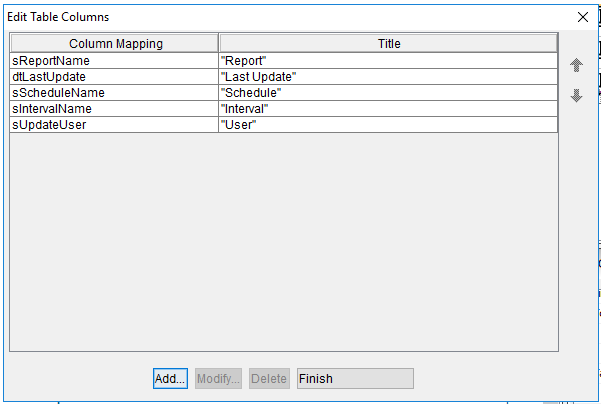
3. In the **Configuration properties** tab, click inside the **Collection** cell and from the drop-down list, select the collection member reference that will be used to populate the table.



4. For information on **Group By**, see the topic, *Configuring Support Links and Tooltips for Group by Table Controls*.

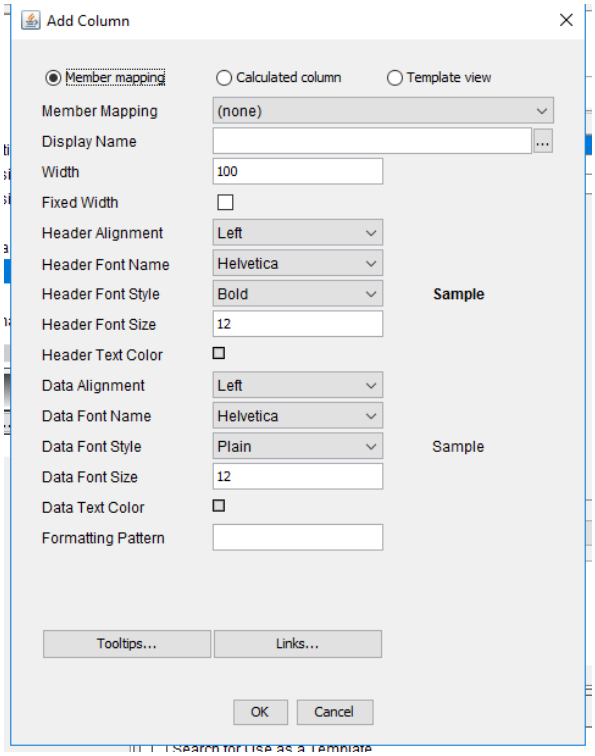
The next step is to configure the column properties.

5. Click inside the **Columns** cell and click **Browse**.
   The "Edit Table Columns" dialog appears.



Once table columns have been mapped, to access the "Edit Table Columns" dialog again, you simply double-click the table control in the Edit panel. This directly launches the Edit dialog; you do not have to click the "Columns" property in the Configuration panel.

6. Select a column and click **Modify** to edit the column, or click **Add**.
   The "Add Column" dialog appears.
7. Do one of the following:
   - Click **Member mapping**.



Define the following **Member mapping column** properties:

Member Mapping Column Properties

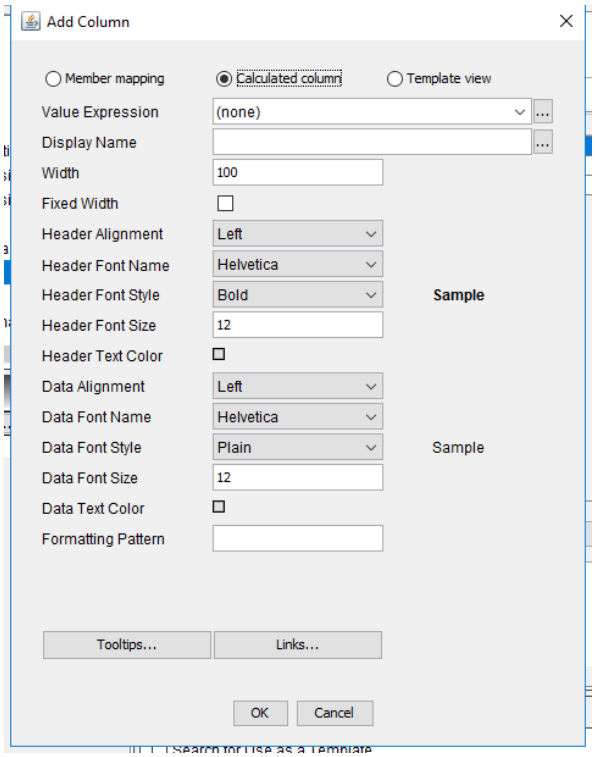
Property	Description	Type
Member Mapping	This property sets the primitive data member in the collection member used for the column. This is only valid when the column type is "Member Mapping".	MemberReference
Display Name	The column title. This defaults to the primitive data member name.	XHQ Expression
Column Width	The column width in pixels (px). If "AutoFit" is set to true, then the column width is automatically adjusted to fit the longest text in the column. The default column width is 100 pixels.	Integer
Header Alignment	This property overrides the default alignment for the data in the column. The default header alignment is left.	Enumeration: • CenterAlignment • LeftAlignment • RightAlignment
Header Font Name	This property allows the user to select a font name	String

Property	Description	Type
	(from a list of fonts) for the column header. By default, the available fonts are Courier, Helvetica, and Times Roman.	
Header Font Style	This property allows the user to select a font style (from a list of font styles) for the column header. The available font styles are Plain, Bold, and Italic.	Enumeration
Header Font Size	This property allows the user to select a font size for the column header.	Integer
Header Text Color	This sets the text color for the column header.	RGB color
Data Alignment	This property overrides the default data alignment in the column. By default, numerical values are right aligned; all else, are left aligned.	Enumeration: • CenterAlignment • LeftAlignment • RightAlignment
Data Font Name	This property allows the user to select a font name (from a list of fonts) for the column data. By default, the available fonts are Courier, Helvetica, and Times Roman.	String
Data Font Style	This property allows the user to select a font style (from a list of font styles) for the column data. The available font styles are Plain, Bold, and Italic.	Enumeration
Data Font Size	This property allows the user to select a font size for the column data.	Integer
Data Text Color	This sets the text color for the column data.	RGB color
Formatting Pattern	When specified, this property defines the format for numerical and data/time values. For example, "#0.000" can be used for float values.	String

After the above **Member mapping column** properties have been configured, continue to *Step 8* to configure column **Tooltips**.

or

- Click **Calculated column**.



Define the following **Calculated column** properties:

Calculated Column Properties

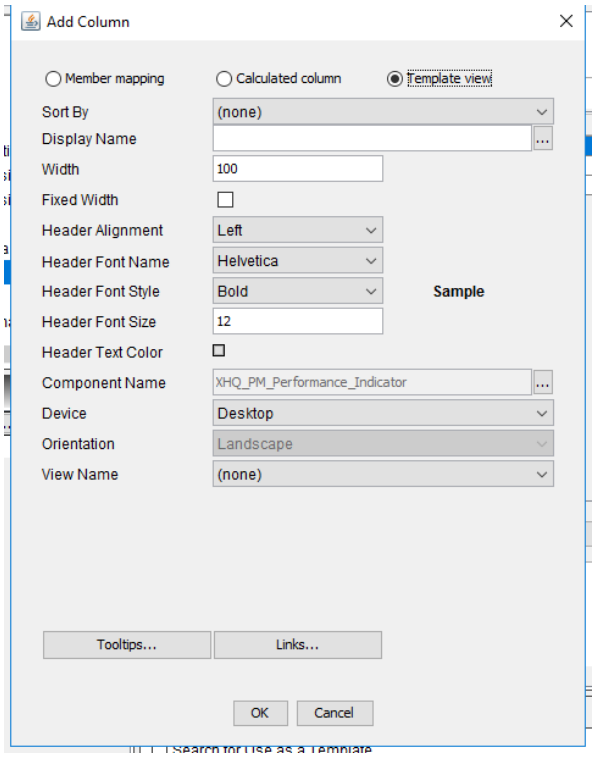
Property	Description	Type
Value Expression	The calculated value.	XHQ Expression
Display Name	The column title. This defaults to the primitive data member name.	XHQ Expression
Width	The column width in pixels (px). If "AutoFit" is set to true, then the column width is automatically adjusted to fit the longest text in the column. The default column width is 100 pixels.	Integer
Header Alignment	This property overrides the default alignment for the data in the column. The default header alignment is left.	Enumeration: • CenterAlignment • LeftAlignment • RightAlignment
Header Font Name	This property allows the user to select a font name (from a list of fonts) for the column header. By default, the available fonts are Courier, Helvetica, and Times Roman.	String
Header Font Style	This property allows the user to select a font style (from a list of font styles) for the column header. The available font styles are Plain, Bold, and Italic.	Enumeration

Property	Description	Type
Header Font Size	This property allows the user to select a font size for the column header.	Integer
Header Text Color	This sets the text color for the column header.	RGB color
Data Alignment	This property overrides the default data alignment in the column. By default, numerical values are right aligned; all else, are left aligned.	Enumeration: • CenterAlignment • LeftAlignment • RightAlignment
Data Font Name	This property allows the user to select a font name (from a list of fonts) for the column data. By default, the available fonts are Courier, Helvetica, and Times Roman.	String
Data Font Style	This property allows the user to select a font style (from a list of font styles) for the column data. The available font styles are Plain, Bold, and Italic.	Enumeration
Data Font Size	This property allows the user to select a font size for the column data.	Integer
Data Text Color	This sets the text color for the column data.	RGB color
Formatting Pattern	When specified, this property defines the format for numerical and data/time values. For example, "#0.000" can be used for float values.	String

After the above **Calculated column** properties have been configured, continue to *Step 8* to configure column **Tooltips**.

or

- Click **Template view**.



Define the following **Template View column** properties:

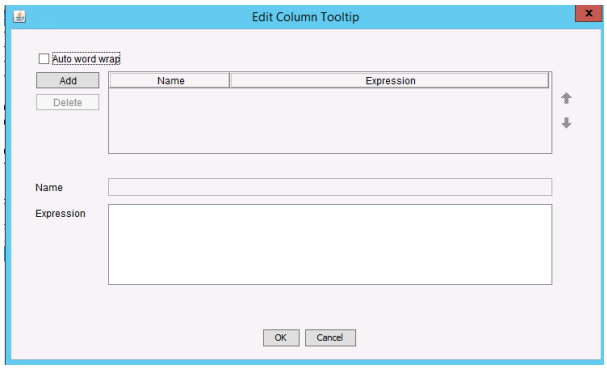
Template View Column Properties

Property	Description	Type
Display Name	The column title. This defaults to the primitive data member name.	XHQ Expression
Column Width	The column width in pixels (px). If "AutoFit" is set to true, then the column width is automatically adjusted to fit the longest text in the column. The default column width is 100 pixels.	Integer
Header Alignment	This property overrides the default alignment for the data in the column. The default header alignment is left.	Enumeration: • CenterAlignment • LeftAlignment • RightAlignment
Header Font Name	This property allows the user to select a font name (from a list of fonts) for the column header. By default, the available fonts are Courier, Helvetica, and Times Roman.	String
Header Font Style	This property allows the user to select a font style (from a list of font styles) for the column header. The available font styles are Plain, Bold, and Italic.	Enumeration
Header Font Size	This property allows the user to select a font size for	Integer

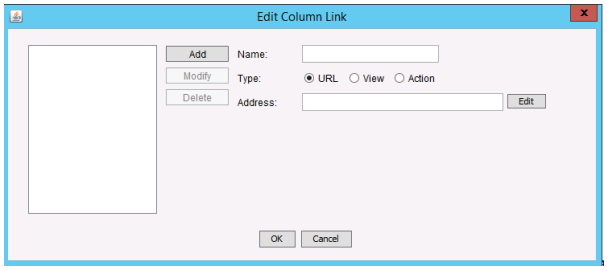
Property	Description	Type
	the column header.	
Header Text Color	This sets the text color for the column header.	RGB color
Component Name	The name of the component for the template view.	String
View Name	The template view name.	String
Template View Member Mapping List	This property allows the user to configure the template view member mapping.	Enumeration

After the above **Template View column** properties have been configured, the next step is to configure the column **Tooltips**.

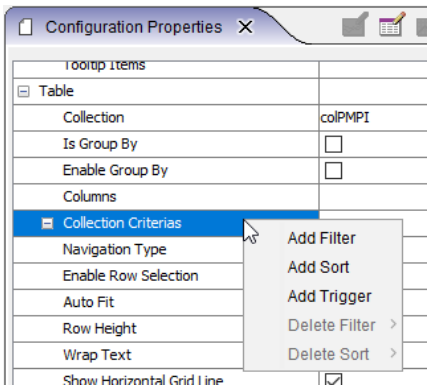
8. From the same "Add (or Modify) Column" dialog, click **Tooltips**.
   The "Edit Column Tooltip" dialog appears.



9. Add an **expression** for the Tooltip and click **OK**.
   This returns you to the "Add (or Modify) Column" dialog.
10. Click **Links**.
    The "Edit Column Link" dialog appears.



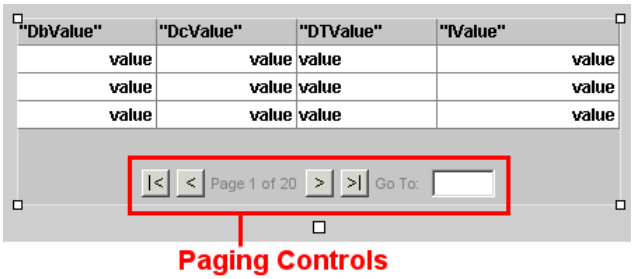
11. Configure the **column link** and click **OK**.
    This returns you to the "Add (or Modify) Column" dialog.
12. Click **OK**.
    This returns you to the "Edit Table Columns" dialog.
13. Click **Finish**.
    Next, configure the Collection Criteria.
14. From the **Configuration Properties tab**, right-click inside the **Collection Criteria** cell.
    A pop-up menu appears with the available criteria settings.



15. Select a criteria type: **Filter**, **Sort**, or **Trigger** and configure accordingly.

For more information, go to the topic, *Setting Table Criteria*.

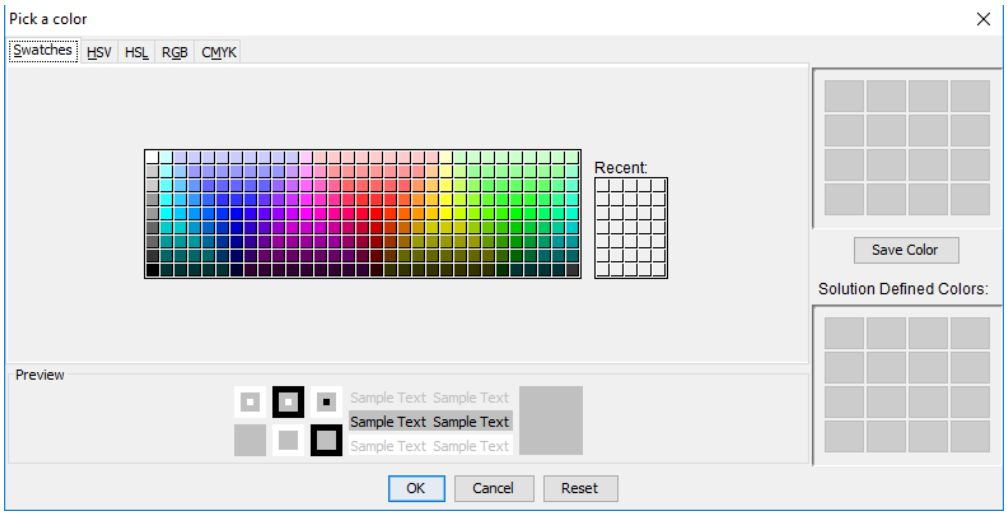
16. Next, define the following table properties:

Property	Description
Navigation Type	• **Paging** Navigation buttons are displayed at the bottom of the table.  • **Scrolling** Vertical scrollbars are displayed when it is necessary to scroll through the entire data set. This is the default.
Page Size	*For the* **Paging** *Navigation Type only.* Enter the number of rows per page. The range of valid values is from 0 (zero) to 500. The default is 0 (zero) pages. Therefore, the number of rows per page is calculated at runtime.
Enable Row Selection	If checked, table rows can be selected at runtime. During runtime, a blue outline appears around the selected row and dependent items are updated.
Auto Fit	If set to true, each column width is automatically adjusted to fit the longest text in the column. The default is false.
Row Height	Enter the row height (in **pixels**). The default is 20 pixels.

Property	Description
Wrap Text	If set to true, the text automatically wraps (or appears in multiple lines) within the table cell. This property is enabled when Auto Fit is set to false. The default is false. **About Wrapping Text** Setting the Wrap Text table property to true enables the text to automatically wrap (or appear in multiple lines) within the table cell. Note, however, the cell content data type must be of the type **String** or the type **Text**. For all other data types (such as Double, Integer, and so forth) wrapping is ignored.
Show Horizontal Grid Line	Check to show horizontal grid line; deselect to hide. The default is checked (enabled).
Show Vertical Grid Line	Check to show vertical grid line; deselect to hide. The default is checked (enabled).

The remaining table control properties enable you to change background, text, and line color.

17. To change the color of a background/text/line property, click the **Browse** button next to the given property. The "Pick a color" dialog appears.



18. Select a color for the given property and click **OK**
19. **Save** the table control properties.

Setting Table Criteria

You can **Filter** against a collection member using an expression.

Collection Criterias	
Filter #1	
Member	(none)
Relation	==
Expression	

Filter Criteria

For more information on expressions, go to the topic, *Using Expressions in Table Control Configurations*.

You can **Sort** by collection member in ascending or descending order.

Collection Criterias	
Sort #1	
Member	(none)
Sort Direction	Ascending

Sort Criteria

In this case, sorting is **case-sensitive**.

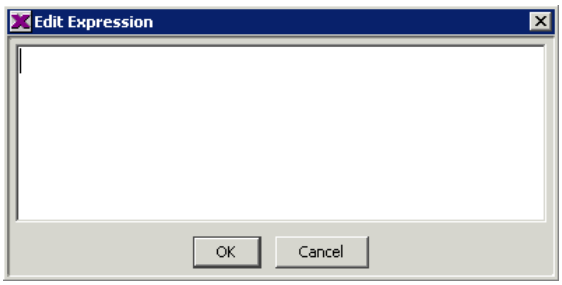
You can set a **Trigger**, to trigger the table to retrieve explicitly.

Collection Criterias	
Trigger	
Button ID	(none)
Apply To Initial Population	☐

Trigger Criteria

Using Expressions in Table Control Configurations

Options that support expressions have a **Browse** button that launches the "Edit Expression" box.



Edit Expression Box

You can also type a custom expression directly into the option box, or click the down arrow to display a list containing compatible members and variables for the supported expression type.

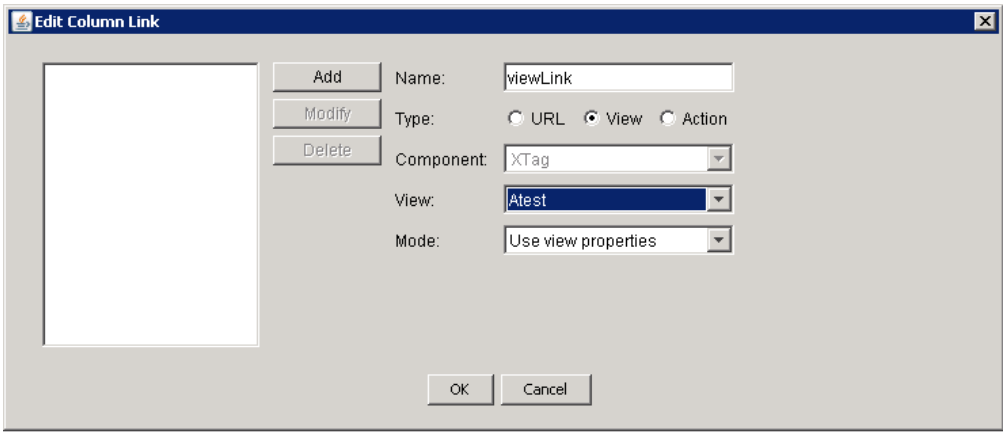
About the Row Select Function

Using this function, the expression is evaluated when a row in the table (which is referenced by the view variable, $Table1) is selected.

Syntax:	`$Table1.<column>.selected()`
Example:	`$Table1.TagName.selected()`
Returns:	This function returns the value object in the selected row of the column specified by the collection member. The type of the return value is the same as that of the collection member. The function returns null with bad quality if there is no row selected or the value object in the selected cell is null.

Links Configuration

When you select a **Collection Member** and check **Enable Row Selected** (in the Properties tab) for a Table control, then the Links tab is enabled.



URL and **Action links** behave the same as in other draw items. So do the **View** and **Mode** properties of **View links**. The exception is that the **Component** property of the View link and **Expressions** are determined by the collection used by the Table Control.

Example for using Expressions in the Table Link:

A collection named "colTable" has "sColumn1" and is configured to be used by the Table Control. The following expression is configured to set the client variable "A":

```
SetClientVariable("A", sColumn1)
```

Group by for Table Control

When Group by is enabled, data is grouped before it is rendered in the table control. Table controls support both runtime (XHQ HTML5 Solution Viewer) and design time (XHQ Workbench) options.

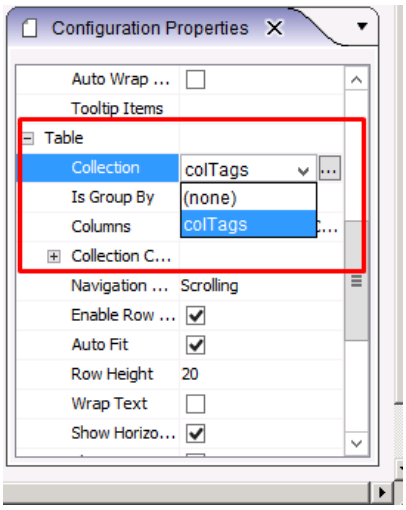
For more information, go to the topic, *Using Group By*.

Setting Group by

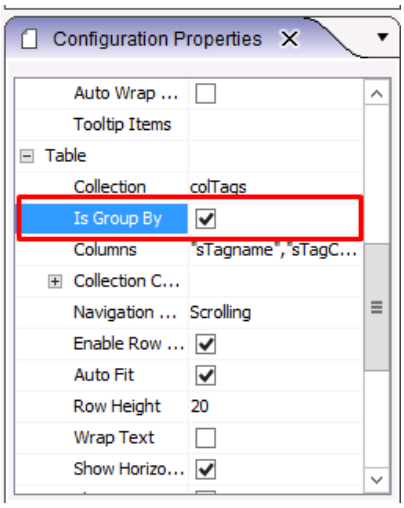
For the XHQ application . . .	Set the Group by property . . .
XHQ Solution Viewer at **runtime**	Enable Group by Table Collection: Tags Is Group By: ☐ Enable Group By: ☑ Columns Collection Criterias *This option is only available for the XHQ HTML5 Solution Viewer.*
XHQ Workbench at **design time**	Is Group by Table Collection: Tags Is Group By: ☑ Columns Collection Criterias

To configure group by (design time)

1. From the XHQ Workbench**Tool palette**, click the **Table** tool.
2. In the **Drawing Panel**, drag diagonally to draw the control, releasing the mouse when the image reaches the desired size.
3. In the **Configuration Properties tab**, expand **Table**, click inside the **Collection** cell and from the drop-down list, select the collection member reference that will be used to populate the table.



4. Check **Is Group By**.



Configuring Support Links and Tooltips for Group by Table Controls

The links and tooltips configuration for group by/aggregation columns is similar to regular table columns.

However, there is a major difference in constructing the XHQ expression. The top view context is used to build the XHQ expressions in the links and tooltips for group by/aggregation table columns, while the collection row context is used to build the XHQ expressions in the links and tooltips for regular table columns.

For example, the following expression can be used by a regular table to configure an action link:

```
setClentVariable("tagkey", TagName)
```

Where `TagName` is a member of the collection component.

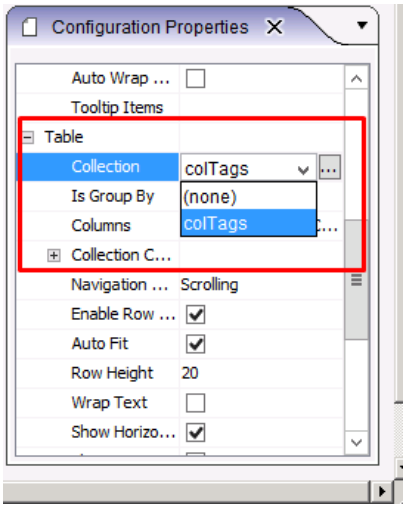
To configure the same link for a group by/aggregation table, the expression uses the top view context as follows:

```
setClientVariable("tagkey", $table1._column1.current())
```

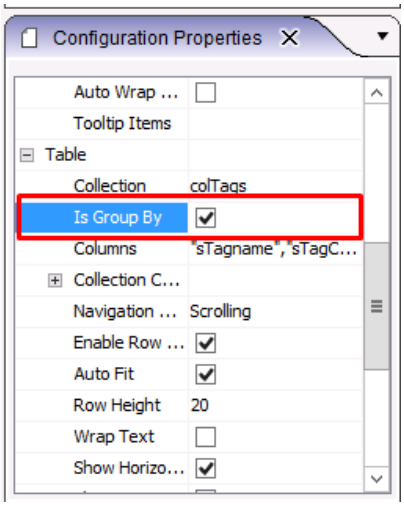
For more information, go to the topic, *Using Group By*.

To configure support links and tooltips for group by table controls

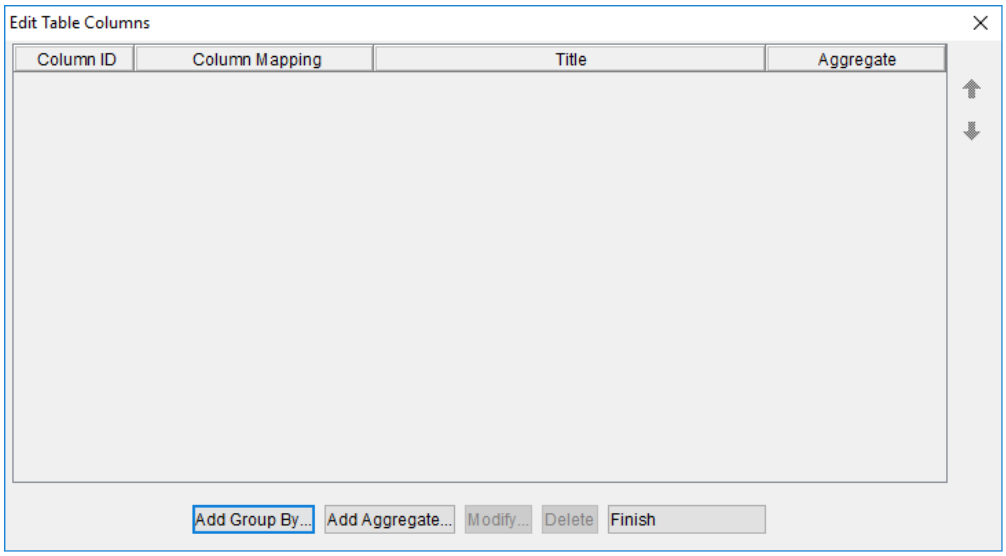
1. In the **Drawing Panel**, drag diagonally to draw the control, releasing the mouse when the image reaches the desired size.
2. In the **Configuration Properties tab**, expand **Table**, click inside the **Collection** cell and from the drop-down list, select the collection member reference that will be used to populate the table.



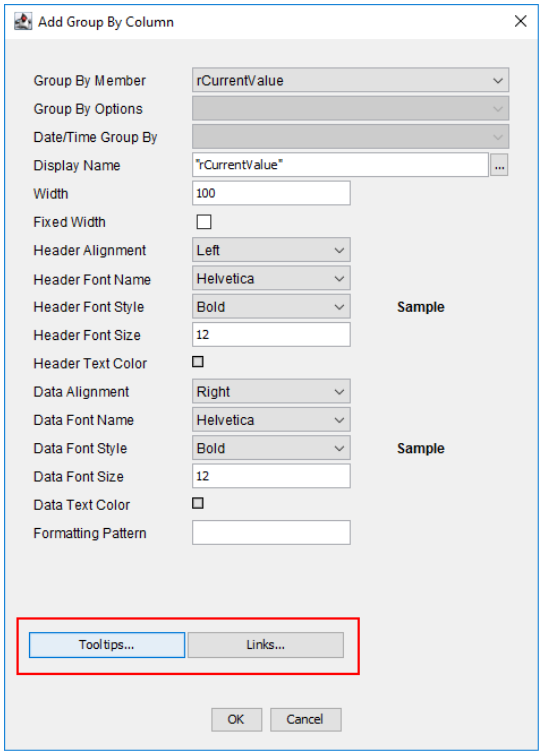
3. Check **Is Group By**.



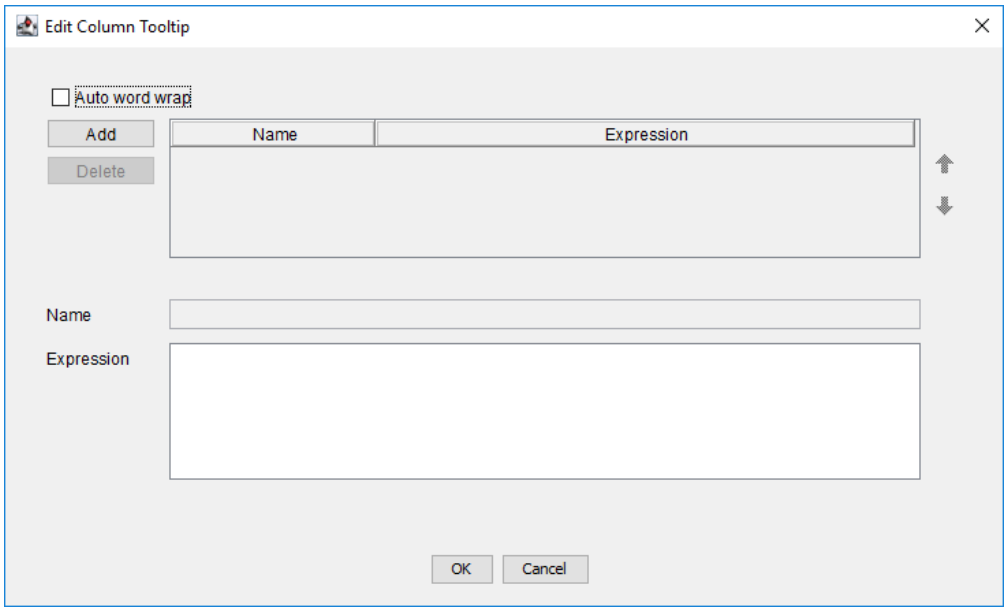
4. Next, select **Columns**, click **Browse**.
   The "Edit Columns" dialog appears.



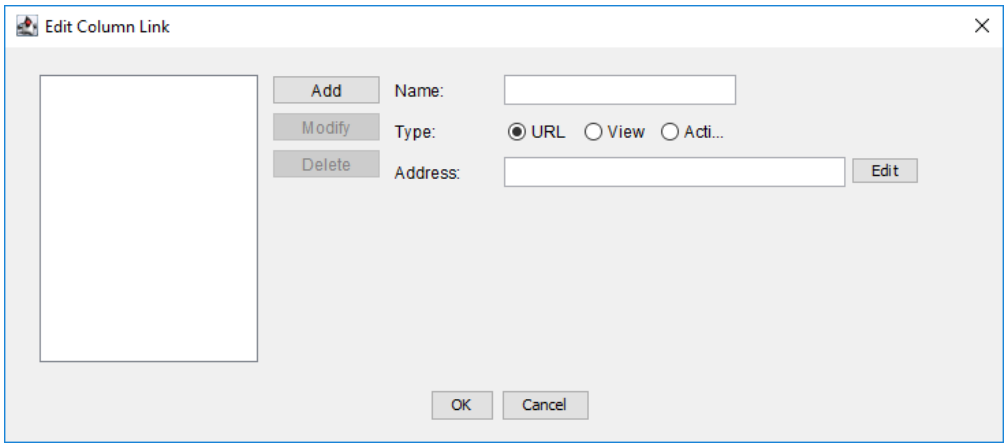
5. From the bottom button bar, click **Add Group By**.
   The "Add Group By Column" dialog appears.
6. Click **Tooltips**.



The "Edit Column Tooltip" dialog appears.



7. Add an **expression** for the tooltip and click **OK**.
   This returns you to the "Add Group By Column" dialog.
8. Click **Links**.
   The "Edit Column Link" dialog appears.



9. Add the **column link** and click **OK**.
   This returns you to the "Add Group By Column" dialog.
10. Click **OK**.
    This returns you to the "Edit Columns" dialog.
11. Click **Finish**.

Using an ENO Group as the Data Source

With this tool, you can set the XHQ table control to use an ENO (Extended Namespace Object) Group as the data source.

For more information on the table control, see the topic, *Inserting the Table Control*.

ENO Table	
Path	::Sample.PI_Group
Component	XHQ_PM_Performance_Indicator
Columns	"Owner","Metric","Name"
Navigation Type	Scrolling
Enable Row Selection	☑
Auto Fit	☐
Row Height	22
Wrap Text	☑
Show Horizontal Grid Line	☑
Show Vertical Grid Line	☑
Header Background Color	R:192 G:192 B:192 - #C0C0C0
Footer Background Color	R:192 G:192 B:192 - #C0C0C0
Table Background Color	R:192 G:192 B:192 - #C0C0C0
Even Row Background Color	R:255 G:255 B:255 - #FFFFFF
Odd Row Background Color	R:255 G:255 B:255 - #FFFFFF
Selected Row Background Color	R:0 G:0 B:112 - #000070
Selected Row Text Color	R:255 G:255 B:255 - #FFFFFF
Grid Line Color	R:128 G:128 B:128 - #808080
Table Border Line Color	R:255 G:255 B:255 - #FFFFFF

ENO Table Properties

Property Descriptions

Property Name	Description
Path	The path to a typed ENO Group instance. This path can either be **absolute** (for example, `::Sample.PI_Group`) or **relative** (`PI_Group`).
Component	*This property is READ ONLY.* The class of the instance. This property value is determined by the specified instance.
Columns	The column titles. Click the Browse button to configure the column properties.
Navigation Type	This property sets the navigation type that is used for the grid. The default is Scrolling.

Property Name	Description
	• **Paging** Navigation buttons appear at the bottom of the table as well as information about the current page and total number of pages. • **Scrolling** The vertical scroll bar appears when it is necessary to scroll through the whole data set.
Enable Row Selection	If true, the rows in the table can be selected during runtime. During runtime, when you mouse-over a row, a blue border appears. All dependent items are updated when a row is selected.
Auto Fit	If true, each column width is automatically adjusted to fit the longest text in the column. The default is false. **When Excel uses a Table control as the data source** If the table control **Auto Fit** is enabled (checked), then all columns are resized to fit the data. If the table control **Auto Fit** is disabled (not checked), then all columns are sized to the column width configuration. Furthermore, if the table control **Auto Fit** is disabled AND **Wrap Text** is enabled (checked), then all columns wrap text as needed.
Row Height	This configures the height in pixels. The default is 20 pixels.
Wrap Text	This is enabled when Auto Fit is set to false. If true, the row height is automatically adjusted to fit the longest test in the row. The default is false.
Show Horizontal Grid Line	The default is true.
Show Vertical Grid Line	The default is true.
Header Background Color	The default is SystemColor.control.
Footer Background Color	The default is SystemColor.control.
Table Background Color	The default is SystemColor.control.
Even Row Background Color	The default is SystemColor.white.
Odd Row Background Color	The default is SystemColor.white.
Selected Row Background Color	The default is SystemColor.textHighlight.

Property Name	Description
Selected Row Text Color	The default is SystemColor.textHighlightText.
Grid Line Color	The default is gray.
Table Border Line Color	The default is white.

Global Properties

For more information, see the topic, *View and ENO Group Instances*, located in the XHQ Administrator's Guide.

ENO Table Variable Method

Typed ENO Group instances support the following variable methods.

`count()`.

Example: `$TableInstance1.count()`

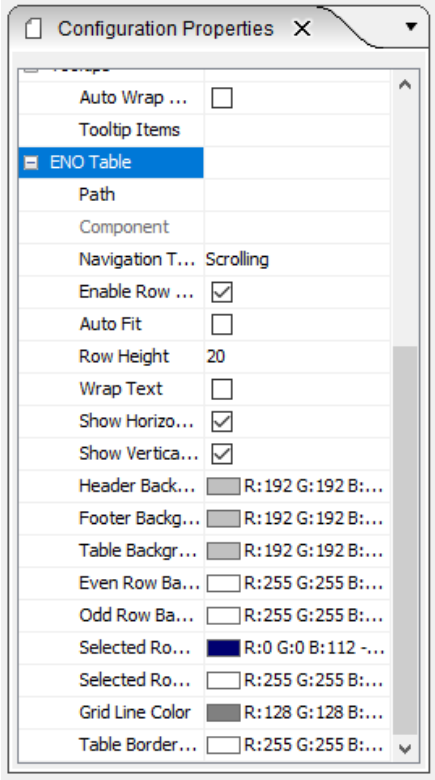
Return: The total number of items in the ENO group.

Example: `$TableInstance1._column1.selected()`

Return: The value of a specific column in the selected row.

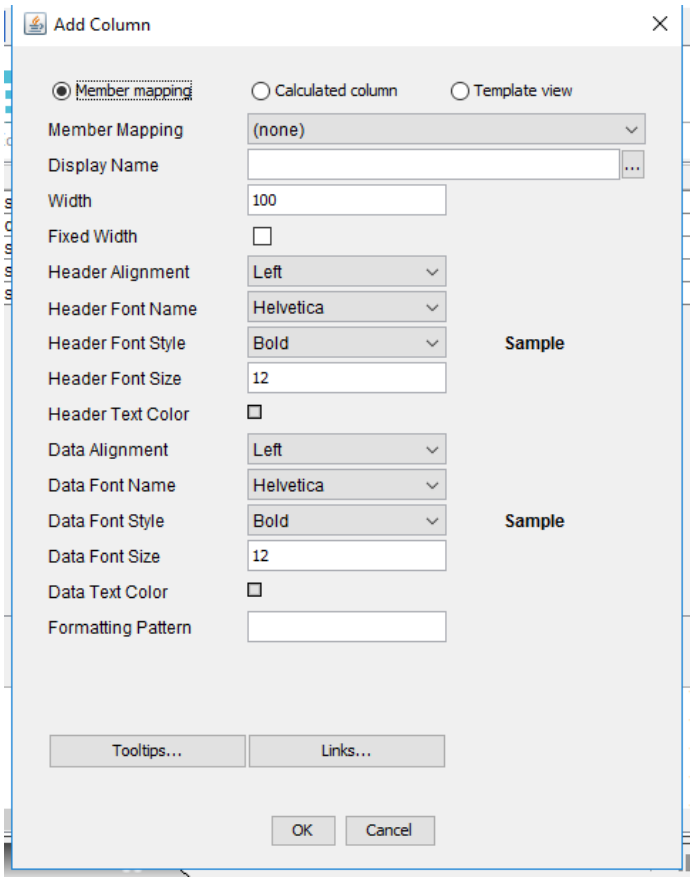
To configure an ENO table

1. From the **Tool palette**, click the **ENO Table** tool .
2. Click in the drawing area and **drag diagonally** to draw the control, releasing the mouse when the image reaches the desired size.
3. In the **Configuration properties** tab and enter the **Path** to the typed ENO Group instance.



The next step is to configure the column properties.

4. Click inside the **Columns** cell and click **Browse**.
   The "Edit Table Columns" dialog appears.
5. Select a column and click **Modify** to edit a column or click **Add** to add a column.



At this point, the ENO table configuration is similar to the table control configuration, with the exception of the collection criteria, which is not needed for the ENO table.

6. To configure the column properties, see *Steps 7 to 13* of the topic, *To configure a table control*.
7. From the same topic, *To configure a table control*, continue with *Steps 16 to 19*.

Inserting an Excel Item

The Excel tool enables you to export collection data from the XHQ view to an Excel spreadsheet.

Configuring the Excel Control

The Excel Control properties consist of the following:

- A **Data Source** property
- A list of **Column** properties

  There are two types of table columns: **Member Mapping** and **Calculated Column**.

- A list of **Data Criteria** properties

  There are two types of data criteria: **Collection Filtering** and **Collection Sort**.

- A list of **Miscellaneous** properties

Data Source

If the data source is a **collection member**, then you must specify the columns to export, as well as the collection criteria.

If the data source is a **collection view variable** ($View1), then specify the columns to export. (Because the criteria defined on the collection view is reused, the Criteria tab is disabled.)

If the data source is a **table control variable** ($Table1), then the column and criteria set on the table item are used.

Data Source Properties

Property	Description	Type
Data Source	The data source for the Excel export, which can be a collection member, a collection view variable, or a table control variable.	MemberReference VariableReference

Member Mapping Column

Member Mapping Column Properties

Property	Description	Type
Member Mapping	This property sets the primitive data member in the collection member used for the column. This is only valid when the column type is "Member Mapping".	MemberReference
Display Name	The column title. This defaults to the primitive data member name.	XHQ Expression
Header Alignment	This property overrides the default alignment for the data in the column.	Enumeration:

Property	Description	Type
	The default header alignment is left.	• CenterAlignment • LeftAlignment • RightAlignment
Header Font Name	This property allows the user to select a font name (from a list of fonts) for the column header. The font list can be configured using the XHQ global property, UseFontAlias. By default, the available fonts are Courier, Helvetica, and Times Roman.	String
Header Font Style	This property allows the user to select a font style (from a list of font styles) for the column header. The available font styles are Plain, Bold, and Italic.	Enumeration
Header Font Size	This property allows the user to select a font size for the column header.	Integer
Header Text Color	This sets the text color for the column header.	RGB color
Data Alignment	This property overrides the default data alignment in the column. By default, numerical values are right aligned; all else, are left aligned.	Enumeration: • CenterAlignment • LeftAlignment • RightAlignment
Data Font Name	This property allows the user to select a font name (from a list of fonts) for the column data. The font list can be configured using the XHQ global property, UseFontAlias. By default, the available fonts are Courier, Helvetica, and Times Roman.	String
Data Font Style	This property allows the user to select a font style (from a list of font styles) for the column data. The available font styles are Plain, Bold, and Italic.	Enumeration
Data Font Size	This property allows the user to select a font size for the column data.	Integer
Data Text Color	This sets the text color for the column data.	RGB color
Formatting Pattern	When specified, this property defines the format for numerical and data/time values. For example, "#0.000" can be used for float values.	String

Calculated Column

This table column uses an XHQ expression to configure the Value Expression property. The expression uses the collection row value as context and supports aggregate functions, such as `rRef(TagName).avg(DateAdd("hour",-12),Now()).`

Calculated columns cannot be sorted.

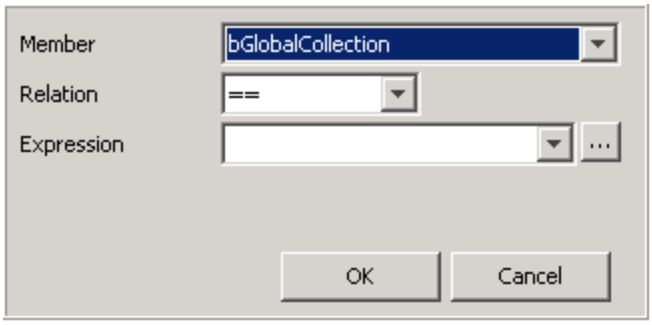
Calculated Column Properties

Property	Description	Type
Value Expression	The XHQ expression for calculated values. The expression uses the collection row value as the context.	XHQ Expression
Display Name	The column title. This defaults to the primitive data member name.	XHQ Expression
Header Alignment	This property overrides the default alignment for the data in the column. The default header alignment is left.	Enumeration: • CenterAlignment • LeftAlignment • RightAlignment
Header Font Name	This property allows the user to select a font name (from a list of fonts) for the column header. The font list can be configured using the XHQ global property, UseFontAlias. By default, the available fonts are Courier, Helvetica, and Times Roman.	String
Header Font Style	This property allows the user to select a font style (from a list of font styles) for the column header. The available font styles are Plain, Bold, and Italic.	Enumeration
Header Font Size	This property allows the user to select a font size for the column header.	Integer
Header Text Color	This sets the text color for the column header.	RGB color
Data Alignment	This property overrides the default data alignment in the column. By default, numerical values are right aligned; all else, are left aligned.	Enumeration: • CenterAlignment • LeftAlignment • RightAlignment
Data Font Name	This property allows the user to select a font name (from a list of fonts) for the column data. The font list can be configured using the XHQ global property, UseFontAlias. By default, the available fonts are Courier, Helvetica, and Times Roman.	String
Data Font Style	This property allows the user to select a font style (from a list of font styles) for the column data. The available font styles are Plain, Bold, and Italic.	Enumeration
Data Font Size	This property allows the user to select a font size for the column data.	Integer
Data Text Color	This sets the text color for the column data.	RGB color

Property	Description	Type
Formatting Pattern	When specified, this property defines the format for numerical and data/time values. For example, "#0.000" can be used for float values.	String

Data Criteria

The Collection **Filtering** criteria is determined by three properties: the Data Member, the Relation, and the Expression.

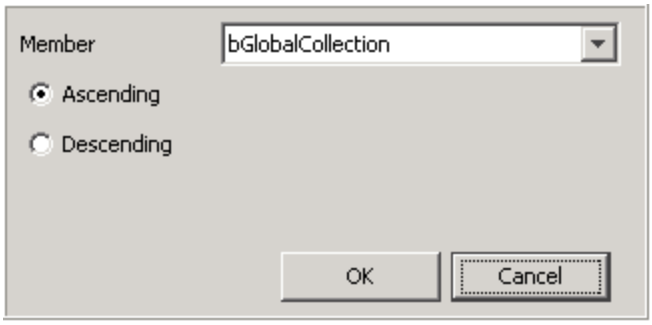


Collection Filtering Criteria Properties

Property	Description	Type
Data Member	This property sets the primitive data member used in the filtering criteria.	MemberReference
Relation	This property sets the op code in the filtering criteria.	Enumeration
Expression	If this applies, this property sets value in the filtering criteria.	XHQ Expression

These filtering properties are the same as those used in the embedded collection view.

The Collection **Sorting** criteria is determined by two properties: the Data Member and the Expression.



Collection Sorting Criteria Properties

Property	Description	Type
Data Member	This property sets the primitive data member used in the filtering criteria.	MemberReference
Expression	This property sets the sort direction.	Enumeration: • Ascending • Descending

In this case, sorting is **case-sensitive**. By default, all filtering criteria are case-sensitive. This applies to the filtering criteria of collection views, the table control, and the Excel item. To change the behavior, set the global setting property `FilterCaseSensitive` to false.

These sorting properties are the same as those used in the embedded collection view.

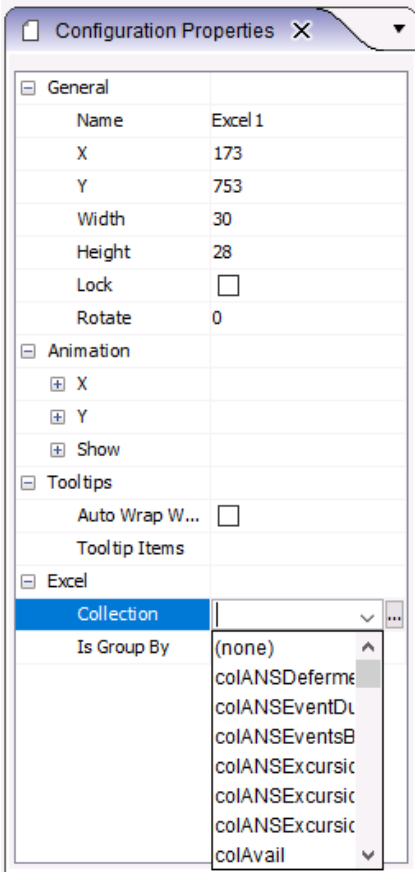
Miscellaneous Properties

Miscellaneous Properties

Property	Description	Type
Max Collection Size	This property sets the maximum number of rows that are retrieved and exported to Excel. The default size is set to 5000.	Integer
Change Excel Image	Enables you to set a custom image for the Excel tool. This image can be resized.	Image

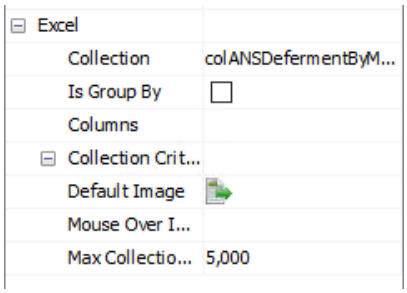
To configure the Excel control

1. From the **Tool palette**, click the **Excel** tool .
2. Click in the drawing area to insert the control.
3. In the **Configuration properties** tab, click inside the **Collection** cell and from the drop-down list, select the collection member reference that will be used to populate the table.

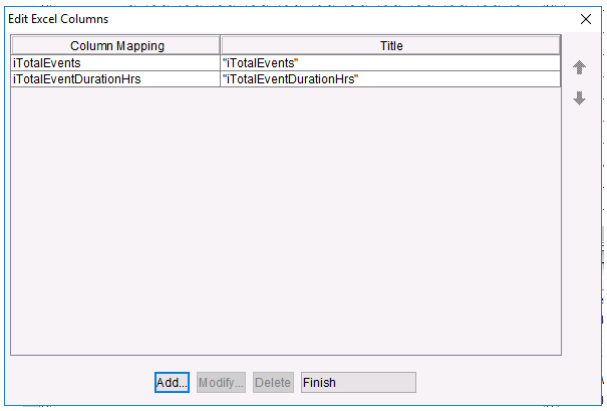


For information on **Group By**, go to the topic, *Using Group By*.

When a collection is selected, additional Excel properties appear.

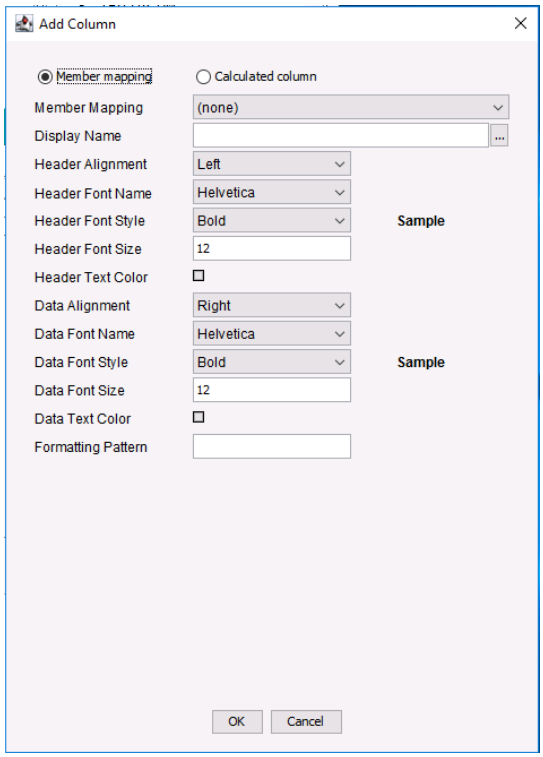


4. Click inside the **Columns** cell and click **Browse**.
   The "Edit Excel Columns" dialog appears.



Once the collection member reference is set, to access the "Edit Excel Columns" dialog again, you simply double-click the Excel control in the Edit panel. This directly launches the Edit dialog; you do not have to click the "Columns" property in the Configuration panel.

5. Select a column and click **Modify** to edit the column, or click **Add**.
   The "Add Column" dialog appears.
6. Do one of the following:
   - Click **Member mapping**.



Define the following **Member mapping column** properties:

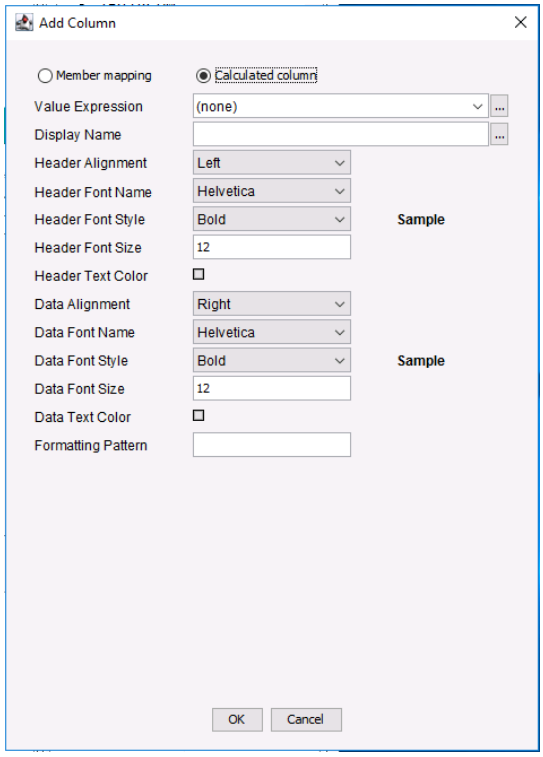
Member Mapping Column Properties

Property	Description	Type
Member Mapping	This property sets the primitive data member in the collection member used for the column. This is only valid when the column type is "Member Mapping".	MemberReference
Display Name	The column title. This defaults to the primitive data member name.	XHQ Expression
Header Alignment	This property overrides the default alignment for the data in the column. The default header alignment is left.	Enumeration: • CenterAlignment • LeftAlignment • RightAlignment
Header Font Name	This property allows the user to select a font name (from a list of fonts) for the column header. By default, the available fonts are Courier, Helvetica, and Times Roman.	String
Header Font Style	This property allows the user to select a font style (from a list of font styles) for the column header. The available font styles are Plain, Bold, and Italic.	Enumeration
Header Font Size	This property allows the user to select a font size for the column header.	Integer
Header Text Color	This sets the text color for the column header.	RGB color
Data Alignment	This property overrides the default data alignment in the column. By default, numerical values are right aligned; all else, are left aligned.	Enumeration: • CenterAlignment • LeftAlignment • RightAlignment
Data Font Name	This property allows the user to select a font name (from a list of fonts) for the column data. By default, the available fonts are Courier, Helvetica, and Times Roman.	String
Data Font Style	This property allows the user to select a font style (from a list of font styles) for the column data. The available font styles are Plain, Bold, and Italic.	Enumeration
Data Font Size	This property allows the user to select a font size for the column data.	Integer
Data Text Color	This sets the text color for the column data.	RGB color
Formatting Pattern	When specified, this property defines the format for numerical and data/time values. For example, "#0.000" can be used for float values.	String

After the above **Member mapping column** properties have been configured, continue to *Step 8* to configure Collection Criteria.

or

- Click **Calculated column**.



Define the following **Calculated column** properties:

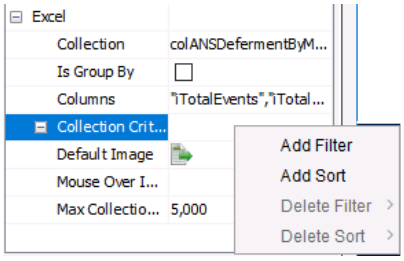
Calculated Column Properties

Property	Description	Type
Value Expression	The calculated value.	XHQ Expression
Display Name	The column title. This defaults to the primitive data member name.	XHQ Expression
Header Alignment	This property overrides the default alignment for the data in the column. The default header alignment is left.	Enumeration: • CenterAlignment • LeftAlignment • RightAlignment
Header Font Name	This property allows the user to select a font name (from a list of fonts) for the column header. By default, the available fonts are Courier, Helvetica, and Times Roman.	String
Header Font Style	This property allows the user to select a font style (from a list of font styles) for the column header. The available font styles are Plain, Bold, and Italic.	Enumeration

Property	Description	Type
Header Font Size	This property allows the user to select a font size for the column header.	Integer
Header Text Color	This sets the text color for the column header.	RGB color
Data Alignment	This property overrides the default data alignment in the column. By default, numerical values are right aligned; all else, are left aligned.	Enumeration: • CenterAlignment • LeftAlignment • RightAlignment
Data Font Name	This property allows the user to select a font name (from a list of fonts) for the column data. By default, the available fonts are Courier, Helvetica, and Times Roman.	String
Data Font Style	This property allows the user to select a font style (from a list of font styles) for the column data. The available font styles are Plain, Bold, and Italic.	Enumeration
Data Font Size	This property allows the user to select a font size for the column data.	Integer
Data Text Color	This sets the text color for the column data.	RGB color
Formatting Pattern	When specified, this property defines the format for numerical and data/time values. For example, "#0.000" can be used for float values.	String

After the above **Calculated column** properties have been configured, the next step is to configure the Collection Criteria

8. Right-click inside the **Collection Criteria** cell.
   A pop-up menu appears with the available criteria settings.



9. Select a criteria type, **Filter** or **Sort**, and configure accordingly.
10. Select the Default and/or Mouse Over images.
11. Set the **Max Collection Size**.
12. **Save** the Excel control properties.

Progress Indicator for Exporting to Excel

If the process of exporting to Excel is taking too long, a status message appears. To set (delay) the period of time until the message appears, configure the **`ProgressBarDelayTimer`** global property. The default is set to 4 seconds.

For more information, go to the topic, *Using globalsettings.properties*, located in the XHQ Administrator's Guide.

Inserting the GISMap Control

The **GISMap control** allows maps from various *providers* to be displayed as *layers*. *Markers* can be added as a top layer to highlight special landmarks. A static representation of this tool is configured in the XHQ Workbench and is activated and displayed at runtime in the XHQ HTML5 Solution Viewer.

The following **libraries** can be used with the XHQ HTML5 Solution Viewer:

- **OL3-GoogleMaps**
  The standard Open Layers version 3 does not have Google Maps built-in. So, this third-party library integrates Google Map on top of Open Layers 3. Since OpenLayers 3 was built with HTML5 support in mind, it supports mobile directly out-of-the-box.
- **OL3-LayerSwitcher**
  Layer switcher similar to what was available in Open Layers 2.
- **OL3-Popup**
  Basic pop-up for Open Layers 3.
- **Shapefile-js**
  Library to convert Shapefiles to GeoJSON so that it can be displayed in Open Layers 3.

Provider Pre-requisites

The available **providers** are:

- Google Map
- Open Street Map
- ArcGISRest
- XYZ

This provider is for any service where there is an/a {X}/{Y}/{Z} in the URL. This includes ArcGISCache.

- Shapefile
- Bing Map
- Image

While XHQ integrates third-party map providers, any costs and commercial arrangements associated with using these map providers are the responsibilities of the customer, not Siemens.

The following are tasks you are required to do **prior to** using the given provider.

Provider Pre-requisites

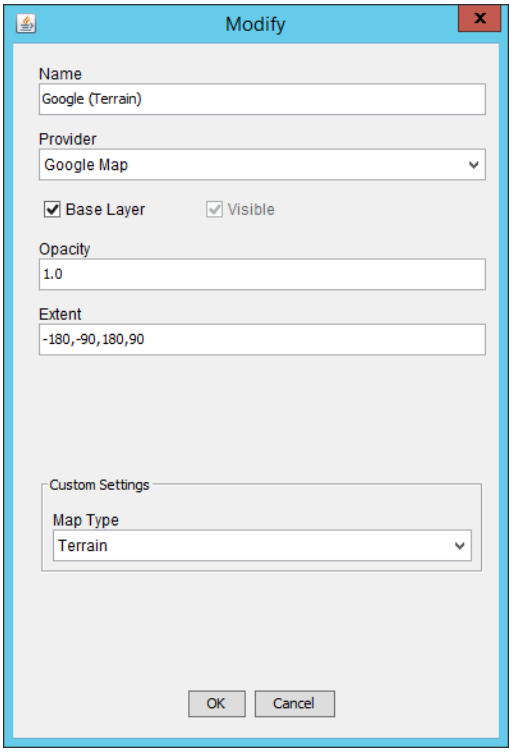
Provider	Pre-requisite
Shapefile	Upload shapefiles to the repos at **`%XHQ_WEB_DATA%\repos\data\webapps\sv\GISMap\shapefiles`**. Each shapefile must be packaged as a ZIP file containing files with DBF, SHP, or SHX extensions.
Image	Upload images to the repos at **`%XHQ_WEB_DATA%\repos\data\webapps\sv\GISMap\`**Images. Images may be PNG, BMP, JPG or JPEG formats.

Provider	Pre-requisite
Google Map	Requires an API **key** and **version** number. • The Google Map API **key** can be set in the `globalsettings.properties` file with the property, **`GISMapGoogleMapKey`**, set to the key value. • The Google Map API **version** can be set in the `globalsettings.properties` file with the property, **`GISMapGoogleMapVersion`**, set to the version value. If this property is not set, the default value, 3.27, is used.
Bing	Requires a key. The Bing Map key can be set in the `globalsettings.properties` file with the property, **`GISMapBingMapKey`**, set to the key value.

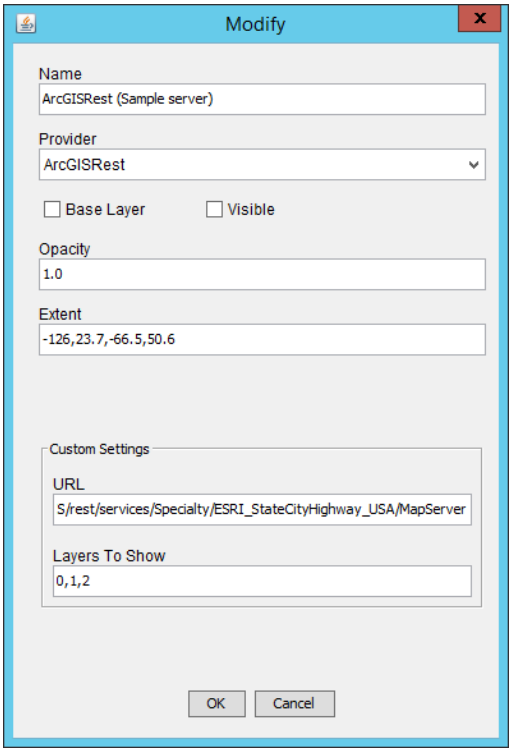
About Layers

You can add as many layers as needed as long as the layer **Name is unique**.

When adding or editing a layer, a provider-specific dialog box appears, allowing you to configure items required by the given provider.



Example: Google Map



Example: ArcGISRest

The configurable layer options for each provider are:

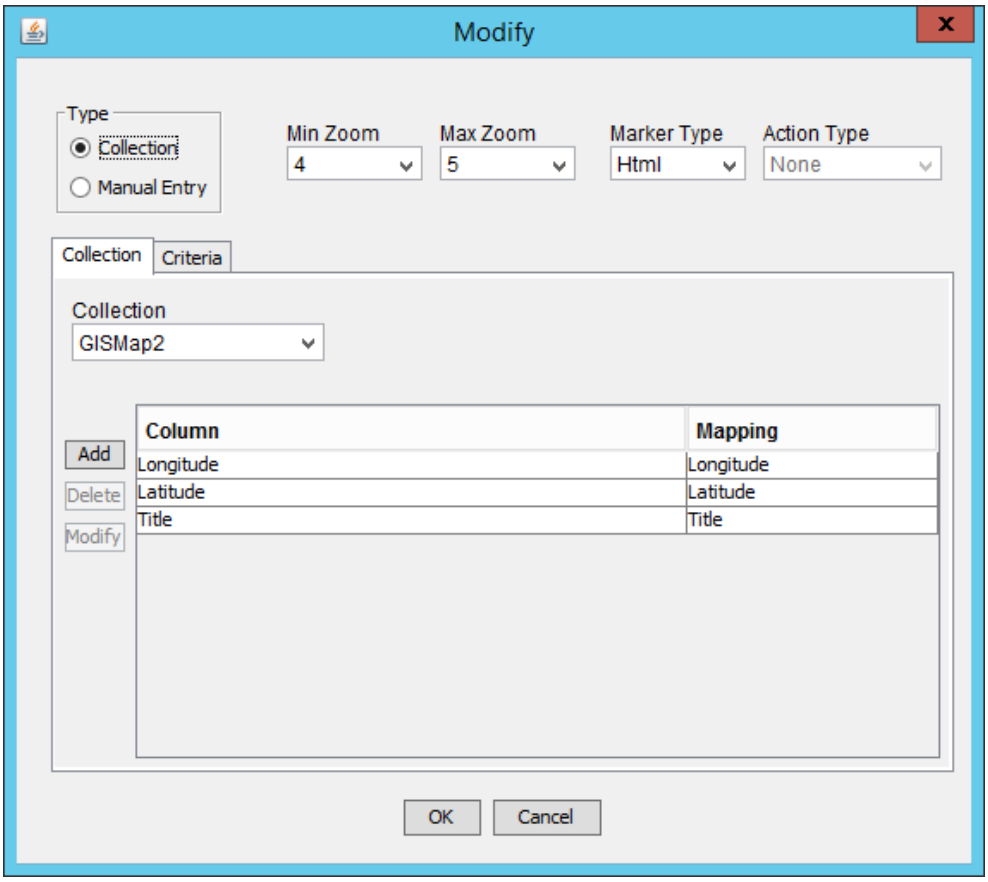
Provider	Custom Setting(s)
Google Map	**Map Type** *Options*: • Road

Provider	Custom Setting(s)	
	• Hybrid • Satellite • Terrain	
ArcGISRest	**URL** Enter the URL to the ArcGIS server.	**Layers To Show** Select which layer to show. *Format*: <layer1>, <layer2>, etc.
XYZ	**URL** This provider is for any service where there is an/a {X}/{Y}/{Z} in the URL. This includes ArcGISCache.	
Shapefile	**Shapefile Name** Select a name from a list of files stored in the repos.	**Line Width** The width of the lines separating borders.
	Fill Color The fill color of the area represented by the Shapefile.	**Line Color** The color of the lines separating borders.
	Show Feature Properties Check (enable) to pop-up a properties dialog when the user left clicks on a Shapefile feature.	**Radius** For circular shapes, the radius of the circle. The default is 5.0.
Bing Map	**Map Type** *Options*: • Road • Aerial • Aerial with Labels	
Image	**Image Name** The name of the image to use for this provider. Select from a list of images that is stored in the repos, or enter a full URL to an image on the internet.	

About Markers

Markers can be added to specify locations on the map and, optionally, can be configured to show only certain zoom levels.

You can configure two types of markers: a **Collection** type or a **Manual Entry** type.



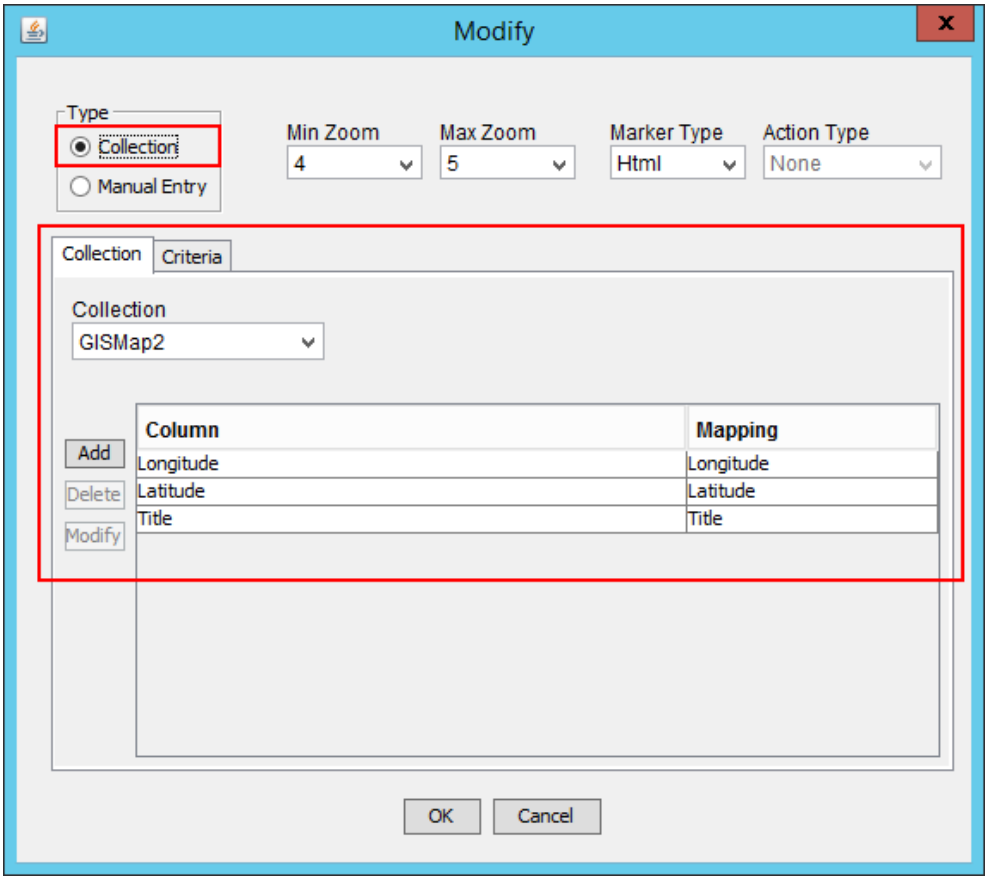
Example: Modify Marker Dialog Box

Option	Description
Type: **Collection**	Select marker mappings for a collection.
Type: **Manual Entry**	Select mappings for markers by entering all the values manually.
Min Zoom	Minimum zoom level for markers.
Max Zoom	Maximum zoom level for markers.
Marker Type	*Options*: • **Icon** Marker displays an icon and/or text. In this case, all mappings are available. • **HTML** Marker displays as HTML. Mappings are limited to Longitude, Latitude, and Title
Action Type	This is the action to take when a marker is **left-clicked**. It is enabled only for the **Icon** maker type. *Options*: • **None** No action taken. • **Drilldown** Evaluates the JavaScript in the **Action** mapping.

Option	Description
	• **Popup** Pop ups the content in the **Popup Content** mapping.

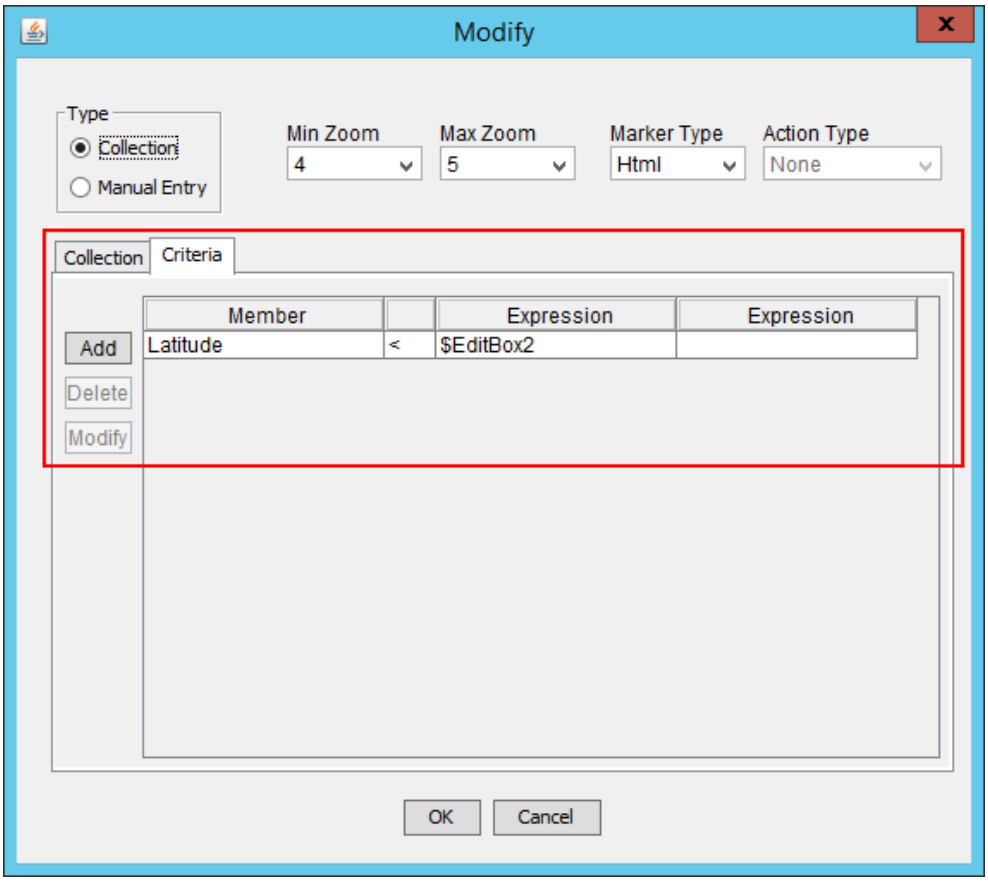
Collection Type Marker

For this marker type, relevant data (such as longitude and latitude, and so forth) for each marker must be retrieved from a collection and then mapped. These mappings support expressions within the view.



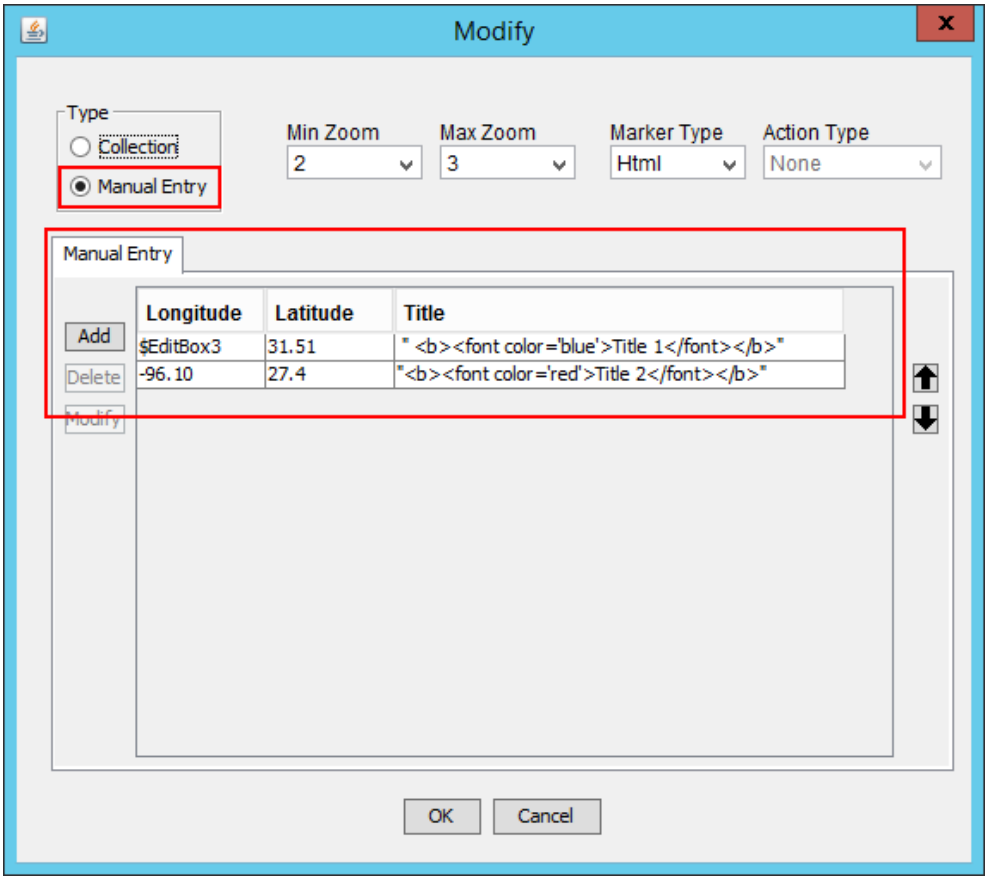
Collection Type Marker Dialog Box - Collection Tab

For the Collection type, click the **Criteria tab** to filter the markers that appear at a zoom level.



Collection Type Marker Dialog Box - Criteria Tab

Manual Entry Type Marker



Manual Entry Type Marker Dialog Box

Mappings

The minimum mappings to get data for each marker are:

- Longitude
- Latitude
- Title or Icon URL

The following mappings all support expressions within the view.

Mapping Option	Description	Data Type	Required?
Longitude	The Longitude of the marker.	Numeric	Yes
Latitude	The Latitude of the marker.	Numeric	Yes
Title	For **Marker Type** > **Icon** The title to be displayed along with the icon. Leave this field blank if a text is not needed for the marker.	String	Title and/or Icon URL

Mapping Option	Description	Data Type	Required?
	For **Marker Type** > **HTML** HTML content may be used for the Title.		Yes
Icon URL	For Marker Type > Icon only The marker icon, or marker image. Leave this field blank if an image is not needed for the marker. There are two (2) different formats to use: • The **image name**. This image name is automatically prefixed with the correct path to the "Images" folder in the repos. Make sure the image exists in the repos. • The **full URL** to an image on the internet.	String	Title and/or Icon
Action	This is the JavaScript action that gets evaluated when left-clicking on the marker. This mapping is used when the left-click field, **Action Type**, is set to **Drilldown**.	String	No
Tooltip	The Tooltip that appears when you hover over the marker. This field has full HTML support.	String	No
Popup Content	The content that is displayed when you left-click on a marker. This mapping is used when the left-click field, **Action Type**, is set to **Popup**.	String	No

Configuring the GISMap Control

Provider Pre-requisites

In order to use the GISMap control, you must address the pre-requisite(s) required for your given provider.

To use the Google Map provider

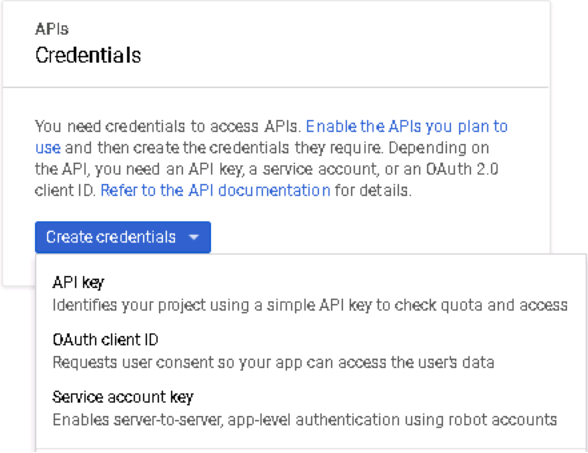
1. Acquire the Google Map API **key** and **version number**.

Generating a Google Map Key

a. Go to https://console.developers.google.com/apis/credentials.

b. From the "Credentials" page, click **Create** and follow the prompts to create a new project.

c. Click the **Create credentials** drop-down and select **API key**.



A pop-up appears with the API key.

d. Depending on your needs, select either a basic key or a restricted one.

2. Open the **`globalsettings.properties`** file.

For more information, refer to the topic, *Using globalsettings.properties*, located in the XHQ Administrator's Guide.

3. **Add** the property, **`GISMapGoogleMapKey`**, and set the value to your Google Map API key.

4. **Add** the property, **`GISMapGoogleMapVersion`**, and set the value to your Google Map version number.

Examples:

```
GISMapGoogleMapKey=ACmeSyBM8T1wen6-coigMbCpdFjazZSlq-mT4w0
GISMapGoogleMapVersion=3.28
```

If the version property is not set, the **default value of 3.27** is used.

5. **Save** the properties file.

Troubleshooting Google Maps

At runtime, if the Google map appears blank, add the following URL links to the browser's Trusted Sites list:

- http://maps.google.com
- http://maps.gstatic.com
- http://mt0.googleapis.com
- http://maps.googleapis.com

To use the Bing provider

1. Acquire the Bing Map **key**.

   Generating a Bing Map Key

   a. Go to https://www.bingmapsportal.com/ .
   b. Under **Help**, click **API & Controls**.
   c. Follow the prompts to get a maps key.
   d. Depending on your needs, select either a Basic or an Enterprise key.

2. Open the **`globalsettings.properties`** file.

   For more information, refer to the topic, *Using globalsettings.properties*, located in the XHQ Administrator's Guide.

3. **Add** the property, **`GISMapBingMapKey`**, and set the value to your Bing Map key.

   Example:

   `GISMapBingMapKey=AclMe06FBIqlHv-zZNz4ntyUou2RS7EnosQX4XuS0BFC-7Z-RJadGXDV4go9t__W`

4. **Save** the properties file.

To use the Shapefile provider

1. Before using this provider, make sure the Shapefile is packaged as a **ZIP** file and contains files with DBF, SHP, and/or SHX extensions.
2. Copy this ZIP file to the repos at **`%XHQ_WEB_DATA%\repos\data\webapps\sv\GISMap\shapefiles`**.

To use the Image provider

1. Before using this provider, make sure the images are in PNG, GIF, JPG/JPEG, or BMP format.
2. Copy these images to the **`%XHQ_WEB_DATA%\repos\data\webapps\sv\GISMap\Images`** folder.

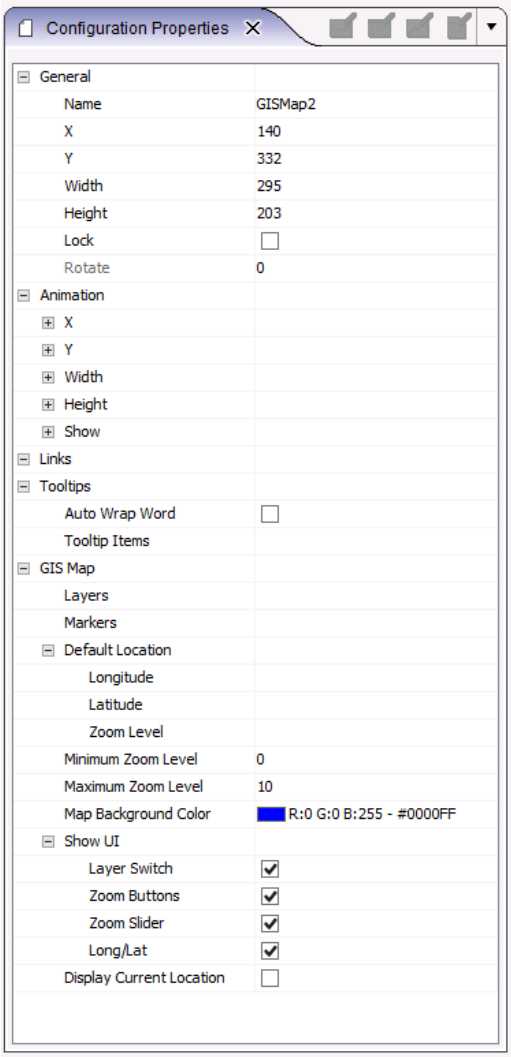
Configuring the control

To set the GISMap control properties

1. From the **Tools palette**, click the **GISMap** tool .
2. Click in the drawing area and drag diagonally to draw a rectangle.

   This rectangle is a *static* representation of the GISMap control. When properly configured, the map will appear at runtime in the XHQ HTML5 Solution Viewer

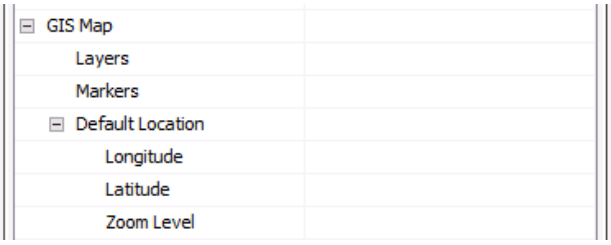
3. In the **Configuration properties** tab, **expand GIS Map**.



Property	Description
Layers	See the topic, *To add and configure layers*
Markers	See the topic, *To add and configure markers*
Default Location > Longitude*	The **longitude** applied on the map when it is initially loaded. An expression may be used. *Default*: Longitude=0
Default Location > Latitude*	The **latitude** applied on the map when it is initially loaded. An expression may be used. *Default*: Latitude=0
Default Location > Zoom Level*	The **zoom level** when the map is initially loaded. An expression may be used. *Default*: Zoom Level=0
Maximum Zoom Level	The **maximum** zoom level.
Minimum Zoom Level	The **minimum** zoom level.

Property	Description
Map Background Color	This sets the background color for the map. If you do not set a color, then the background defaults to blue.
Show UI > Layer Switcher	Displays the layer switcher control.
Show UI > Zoom Buttons	Displays the zoom buttons.
Show UI > Zoom Slider	Displays the zoom slider.
Show UI > Long/Lat	Displays the longitude and latitude.
Display Current Location	*This feature is only for secure connections, namely https.* Displays a circular marker showing the user's current location, and moves as the user changes location.

*** For Longitude, Latitude, and Zoom Level**

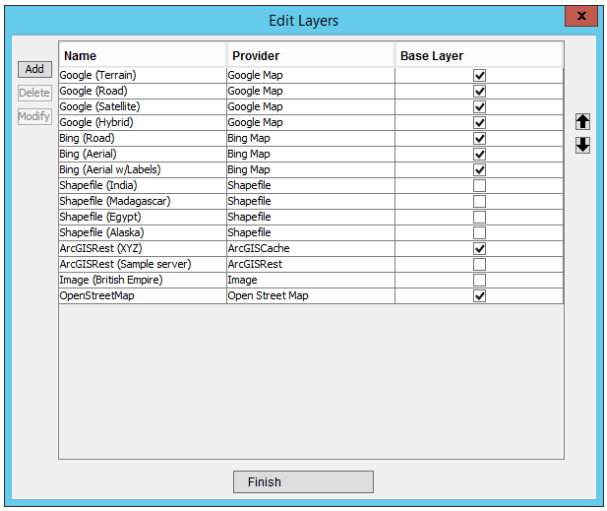


If these "Default Location" properties are not configured, for the purpose of rendering the map and markers, their **default value** is zero (0).

4. Configure the GISMap control properties.

To add and configure layers

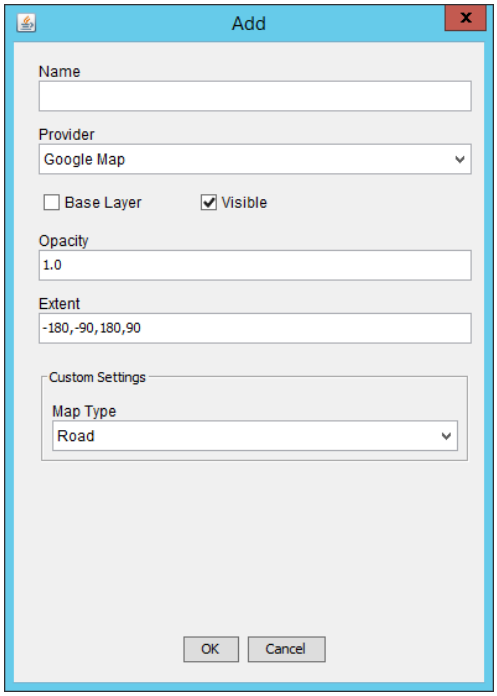
1. Click inside the **Layers** cell and then click the **Browse** button.
   The "Edit Layers" dialog appears.



You may add as many layers as needed, as long as each **layer name is unique**. Layer names may include spaces, but spaces at the beginning or at the end of names are ignored when comparing for uniqueness.

See the topic, *About Layers*, for more information.

2. Click **Add**.
   The "Add" dialog appears.



3. Enter a unique **Name**.
4. Select a **Provider**.
   The dialog then displays provider-specific custom options.
5. Determine if this is a **Base Layer**.

6. If not a base layer, then check **Visible**.
   This determines whether a non-base layer is initially visible when the map is loaded.
7. Enter an **Opacity**.
   Opacity is a real value from 0 to 1.0.
8. Enter an **Extent**, using the comma-separated format:
   `BottomLeftLongititude,BottomLeftLatitude,TopRightLongitude,TopRightLatitude`

   The extent is the visible area of the map.

   Example:

   `-180,-90,180,90`
9. Set the provider-specific custom options.

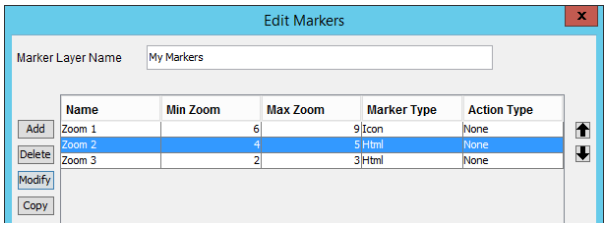
Provider	Custom Setting(s)	
Google Map	**Map Type** *Options*: • Road • Hybrid • Satellite • Terrain	
ArcGISRest	**URL** Enter the URL to the ArcGIS server.	**Layers To Show** Select which layer to show. *Format*: <layer1>, <layer2>, etc.
XYZ	**URL** This provider is for any service where there is an/a {X}/{Y}/{Z} in the URL. This includes ArcGISCache.	
Shapefile	**Shapefile Name** Select a name from a list of files stored in the repos. **Fill Color** The fill color of the area represented by the Shapefile. **Show Feature Properties** Check (enable) to pop-up a properties dialog when the user left clicks on a Shapefile feature.	**Line Width** The width of the lines separating borders. **Line Color** The color of the lines separating borders. **Radius** For circular shapes, the radius of the circle. The default is 5.0.
Bing Map	**Map Type** *Options*: • Road • Aerial • Aerial with Labels	
Image	**Image Name** The name of the image to use for this provider. Select from a list of images that is	

Provider	Custom Setting(s)
	stored in the repos, or enter a full URL to an image on the internet.

10. Click **OK**.
    This returns you to the "Edit Layers" dialog.
11. Click **Finish**.

To add and configure markers

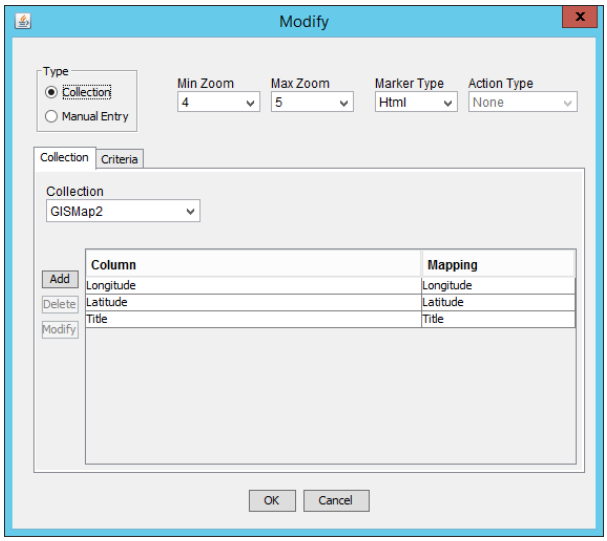
1. Click inside the **Markers** cell and then click the **Browse** button.
   The "Edit Markers" dialog appears.



You may add as many marker mapping configurations as needed. Minimum and maximum zoom levels among the mapping configurations are allowed to overlap. This means that there could be markers from multiple mapping configurations existing on the same map.

Once the GIS map is added to the view, to access the "Edit Markers" dialog again, you simply double-click the GIS map control in the Edit panel. This directly launches the Edit dialog; you do not have to click the "Markers" property in the Configuration panel.

2. Enter the **Marker Layer Name**.
   This is the name that appears in the Layer Switcher control for the marker layer.
3. Click **Add** to select a collection for mapping.
   The "Modify" dialog appears.



4. For **Type**, select either **Collection** or **Manual Entry** for mapping.

   See the topic, *About Markers*, for more information.

5. Select the **Min Zoom** level.
6. Select the **Max Zoom** level.
7. For **Marker Type**, do one of the following:
   - Select **Icon**. *This is the default*.
     All the GISMap mapping options are available. In this case, you can map the Title option or the Icon URL option, or map both.

   or

   - Select **HTML**.
     Only the Longitude, Latitude, and Title mapping options are available. The Title option may contain HTML content.

     For the HTML Marker Type, the following left-click actions (events) are disabled:
     - MarkersLassoList
     - MarkersVisibleAreaList
     - SelectedMarkerLongitude
     - SelectedMarkerLatitude
     - SelectedMarkerTitle

8. For the Icon marker type only, select an **Action Type** to take when the marker is left-clicked:
   - **None**
     No action taken.
   - **Drill down**
     Evaluates the JavaScript in the **Action** mapping.
   - **Popup**
     Displays the content of the **Popup Content** mapping in a pop-up.
9. Depending on the marker type you selected, do the following:

   For Collection type

   a. Click the **Collection tab** and select a **Collection** from the drop-down.
   b. Click **Add** to map the collection column to the GISMap option.
   c. Click the **Criteria tab** and **Add** a filter.

   For Manual Entry type

   - From the **Manual Entry tab**, click **Add** and manually enter mapping values.

10. Click **OK**.
    This returns you to the "Edit Markers" dialog.
11. Click **Finish**.

Sending events to the GISMap

Sending events to the GISMap is done through *variable method reference*. For example, to set the visible layers, you would use the expression, `$GISMap1.setvisiblelayers("layer1|layer2").`

The following table lists the methods available for the GISMap control.

Method names are in **lowercase**.

Method	Description
`gotodefaultlocation()`	Goes to the default location specified by the `DefaultLongitude`, `DefaultLatitude`, and `DefaultZoomLevel` properties.

`setlocation()`	Sets the current location of the map. There are two types:
	• The **current zoom level** mode,
	• And the **zoom to a specific level** mode.
	
	For "current zoom level" mode
	
	Syntax:
	`setlocation(long,lat,zoomtolocationtype)`
	
	Parameters:
	• `long` (Float)
	The longitude
	• `lat` (Float)
	The latitude
	• `zoomtolocationtype` (String)
	This determines how the move to this location is performed. There are three options: normal, pan*, or fly*.
	
	* Currently, Google Map does not work with Pan or Fly.
	
	Example:
	`setlocation(-95.35,29.72,"normal")`
	
	For "zoom to a specific level" mode
	
	Syntax:
	`setlocation(long,lat,zoomlevel,zoomtolocationtype)`
	
	Parameters:
	• `long` (Float)
	The longitude
	• `lat` (Float)
	The latitude
	• `zoomlevel` (Integer)
	The specific zoom level.
	• `zoomtolocationtype` (String)
	This determines how the move to this location is performed. There are three options: normal, pan*, or fly*.
	
	* Currently, Google Map does not work with pan or fly.
	
	Example:
	`setlocation(-95.35,29.72,5,"normal")`

Method	Description

`setvisiblelayers()`	Sets the layers that are visible; all other layers are hidden.
	Important: This call only affects **non-base layers**. Base layers included in the parameter are ignored.
	The parameter is:
	• `layersstring` (String)
	Use the following format for this string: "`layer1|layer2|...|layerN`"
	Syntax:
	`setvisiblelayers("layer1|layer2|layer3")`
	Example:
	There are 3 layers: OSM, My Markers, and a non-base Google Map layer.
	`setvisiblelayers("OSM|My Markers")`
	At runtime, the "OSM" and "My Markers" layers are visible; the "Google Map" layer is invisible.

EXAMPLE: USING SETVISIBLELAYERS() METHOD

Given: A GISMap control in the view, GISMap1. GISMap1 has the following layers, "Google Map", "OSM", and "My Markers".

1. Add a **text control**, that is a **String** type and named **`Editbox1`**.
2. Next, add a **button control** and set its **Expression** property to **`$GISMap1.setvisiblelayers($EditBox1)`**.
3. At runtime, in the EditBox1 textbox, enter the name of the layer you want visible.
4. Then, click the button to trigger the method call to perform the layer visibility.

Getting events from the GISMap

To receive events from the map, use an expression like `$GISMap1.zoomlevel` (which returns the current zoom level of the map).

Once you type the variable to the map control, `$GISMap1`, the auto-completion feature pops up a list of events to choose from.

The following table lists the available events.

Event names do not have the same case restriction as methods.

Dynamic Values

Events	Description
`MarkersItemsCount`	The total number of markers retrieved. *Return Type*: Integer
`MarkersLassoList`	The list of markers selected using the lasso. *Usage*: `$GISMap1.MarkersLassoList`

Events	Description
	Return Type: String The list of markers are delimited by the pipe ("|") symbol with the following format: `marker1|marker2|...|markerN`

`MarkersVisibleAreaList`	The list of markers in the current visible area.
	Usage:
	`$GISMap1.MarkersVisibleAreaList`
	Return Type: String
	The list of markers are delimited by the pipe ("|") symbol with the following format:
	`marker1|marker2|...|markerN`

`SelectedMarkerLatitude`	The latitude of the selected (clicked on) marker. *Return Type*: Real
`SelectedMarkerLongitude`	The longitude of the selected marker. *Return Type*: Real
`SelectedMarkerTitle`	The title of the selected marker. *Return Type*: String
`ZoomLevel`	The current zoom level. *Return Type*: Integer

`CurrentExtent`	The current extent.
	Usage:
	`$GISMap1.CurrentExtent`
	Return Type: String
	Extent locations are delimited by the pipe ("|") symbol with the following format:
	`longBottomLeft|latBottomLeft|longTopRight|latTopRight`

`VisibleLayers`	The layers currently visible.
	Usage:
	`$GISMap1.VisibleLayers`
	Return Type: String
	The list of layers are delimited by the pipe ("|") symbol with the following format:
	`layer1|layer2|...layerN`

Static Values

Events	Description
`MaxZoomLevel`	Maximum zoom level. *Return Type*: Integer
`MinZoomLevel`	Minimum zoom level.

Events	Description
	Return Type: Integer
`DefaultLongitude`	Default longitude. *Return Type*: Real
`DefaultLatitude`	Default latitude. *Return Type*: Real
`DefaultZoomLevel`	Default zoom level. *Return Type*: Integer

7 | Building Views

This section describes how to construct complex views that contain hyperlinks, value items, and animations.

Using Links

To increase the flexibility of navigation between the various views you have prepared and to provide access to external links, XHQ enables you to insert hyperlinks to other views, URLs, and actions (which are specified by function expressions).

One or more links can be attached to any view element or group of view elements in a view. If you link multiple view elements and then group them, each individual view element retains its own links. Or, you can group view elements and then create one or more links for the entire group.

Any view element or group of view elements that you link becomes a hot spot at runtime, and a bounding rectangle will appear when you mouse over the view element to indicate that it has one or more links.

At runtime, clicking a hyperlink to another view causes the default hyperlinked view to display and replace the current display. You can also right click on any view element and select any of its other views, animations, trends, or external links.

When a hyperlinked view has been selected, either by clicking on a view element or by right-clicking and selecting from the list, the hyperlinked view displays. If the hyperlinked view is an XHQ view, it replaces the current view. If the hyperlinked view is to a URL, another instance of the browser launches to display the linked page. If the hyperlinked view is to an action, the given function is performed.

Defining "Parents," "Children," and "Grandchildren"

The concept of *parent, child,* and *grandchild* pertains to the nesting of components and views. For example, when you animate a view element, you may want to drive the animation from a data value that is contained in another component further down in the solution hierarchy. Assume Component 1 contains Component 2, which contains Component 3. Component 1 is the *parent*. Component 2 is therefore a *child* of Component 1, while Component 3 is a *grandchild* of Component 1.

A similar concept applies to hyperlinks. You can provide hyperlinks to views that belong to other components further down in the solution hierarchy. If the parent has a link and the child does not, then the child inherits the parent link. If both parent and child have links, then the child will launch its own link.

To attach a link to any view element or group of view elements

1. **Select the view element** for which you want to provide a hyperlink.
2. Click on the **Links** tab.
3. Enter a descriptive name for the link in the **Name** field.
   This is the name that will appear in the runtime hyperlink menu.
4. Do one of the following:
   - Select **URL** to attach a link to a URL and proceed to step 5.

   or

- Select **View** to attach a link to another XHQ view and proceed to step 8.

or

- Select **Action** to perform a given function and proceed to step 12.

The right-hand area of the Links tab changes to provide content appropriate for the type of link you are adding.

5. If you selected **URL** in step 4, enter the full URL address for the hyperlink in the **Address** text box, enclosed by double quotation marks.

   Example: "http://www.acme.com/xhq"

6. OPTIONAL: Click **Set as Default** to specify this view as the default view.
7. Click **Add**.
   The name of the hyperlink appears in the Links list; the hyperlink is attached.
8. If you selected **View** in step 4, continue by doing one of the following:

- In the **Component** drop-down list, **select a component** that contains the view for which you want to provide a link. The components listed are the children of the component for which you are building the current view.

or

- If you want to select a component further down in the solution hierarchy, such as a grandchild, you can type in the name of the component using the form:

```
component2.component3.component4
```

Where:
`component2` is a child of the current component (which would be `component1`), `component3` is a grandchild of `component1`, and so forth.

If you do not select a component, the views displayed in step 9 below are peer views of the current component.

9. In the **View** drop-down list, select the view to which you want to create a hyperlink.
   The views you see in the list belong to the component you selected in step 8.
10. Under **Mode** you can set how the view will display when the view link is executed. Select from the following:

    - **Use views properties**
      You can *indirectly* set view links to pop-up through this mode if the Pop Up property is selected from the Views Properties tab.
    - **Replace**
      This mode is the *traditional* behavior of the view link, which causes the existing view in the view panel to be replaced.
    - **Pop Up**
      You can *directly* set view links to pop-up through this mode. It causes a new pop-up window to display the view.

      For more information, go to the topic, *Setting View Link Pop ups*.

11. If desired, click the **Set as Default** checkbox to specify this view as the default view.
12. Click **Add**.
    The name of the hyperlink appears in the Links list; the hyperlink is attached.
13. If you selected **Action** in step 4, enter a function expression in the **Function** text box.

    Example: `ShowView("::MySolution.comp1", "defaultView")`

For more information on Action links, refer to the topic, *Using Action Link Functions*.

14. OPTIONAL
    Click **Set as Default** to specify this view as the default view.
15. Click **Add**.
    The name of the function appears in the Links list.
16. From the **File** menu, click **Save to repository**.

Using Expressions to Link to a URL or Another View

When you create a link to another URL or view, you can use an expression that can be evaluated and return a URL string, based on the operators and operands you include in the expression. For example, suppose you have ten views of ten separate pump components and you want to link those views to the URLs where the pumps' maintenance history records reside. Instead of specifying a separate URL link for each pump, you could enter an expression in the Address field that automatically links to the appropriate URL, depending on the pump view displayed.

```
"http://www.mycom.com/machineno"+IntToString(machineno)+".pdf"
```

When entering a URL, the entire address must be enclosed in **double** quotes.

The above example assumes that the member "`machineno`" is a member of the component against which you building the view and URL link.

Expressions are a very powerful tool that you can use in a variety of ways to gain flexibility.

Examples of Relative URLs

Here are some examples of using relative URL for links in **Internet Explorer**. These depend on the value in the "Material" field. "Nicaragua" is the host name where the Enterprise Server is running.

Relative URL Examples (for Internet Explorer users only)

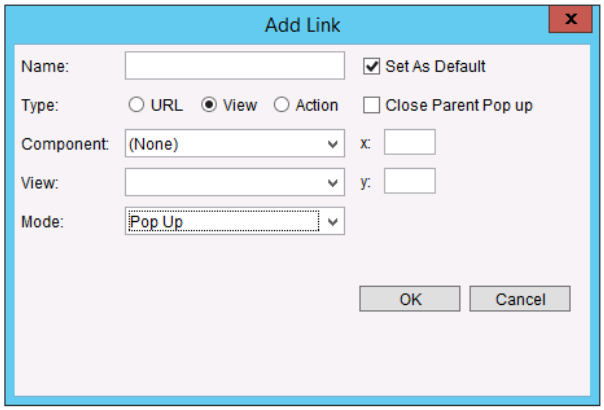
Expression In URL Link Value	Resolves To URL In IE*
"http:/XHQ/"+Material+".htm"	http://acme/XHQ/IX%20Unleaded%20Regular.htm
"http:/XHQDemo/XHQ/"+Material+ ".htm"	http://acme/XHQDemo/XHQ/IX%20Unleaded %20Regular.htm
"http:XHQ/"+Material+".htm"	http://acme/XHQ/IX%20Unleaded%20Regular.htm
"http:XHQDemo/XHQ/"+Material+".htm"	http://acme/XHQDemo/XHQ/IX%20Unleaded %20Regular.htm

* Assuming "Material" is "IX Unleaded Regular"

Setting the View Link Mode

From the Links tab of the XHQ Workbench, you can set how the view will display when the view link is executed. Under **Mode**, you can select from the following:

- **Use views properties**
  You can *indirectly* set view links to pop-up through this mode if the Pop Up property is selected from the Views Properties tab.
- **Replace**
  This mode is the *traditional* behavior of the view link, which causes the existing view in the view panel to be replaced.
- **Pop Up**
  You can *directly* set view links to pop-up through this mode. It causes a new pop-up window to display the view.

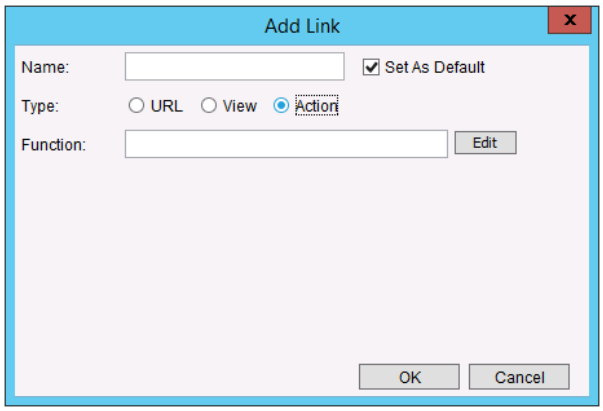


View Link - Pop Up Mode

For more information on how to set an action link pop-up, go to the topic, *Setting Action Link Pop ups*.

Using Action Link Functions

In addition to the URL and View link types on the Links tab, the Action type allows you to perform any action specified by a function expression. The function expression is entered in the Function box.



Action Link Pop Up

The function expression(s) you use should have no type associated with it. Separate multiple expressions with a comma.

For more information, go to the topic, *Action Link Functions*, located in the XHQ Reference Guide.

For more information on the use of commas in multiple expressions, refer to the topic, *Comma Operator*, located in the XHQ Reference Guide.

For more information on how to set an action link pop-up, go to the topic, *Setting Action Link Pop ups*.

Example: Using the showView function in an Action Link to show a mobile view

1. To add a link to a view, from the **Configuration Properties tab**, right-click inside the **Links** cell.
   A shortcut menu appears with the Add option.
2. Click **Add**.
   The "Add Link" dialog appears.
3. Enter a **Name**.
4. For **Type**, click **Action**.
5. For **Function**, click **Edit**.
   The "Edit Expression" dialog appears.
6. Enter an expression in the following format:

   ```
   showView('<objPath>','<viewName>@<deviceType_firstLetter><orientation_firstLetter>'
   [,mode,filter,x,y])
   ```

   Where:

   `[@<deviceType_firstLetter><orientation_firstLetter>]` is OPTIONAL.

   Values for `<deviceType_firstLetter>` are: D (for Desktop), T (for Tablet), and P (for Phone).

   Values for `<orientation_firstLetter>` are: L (for Landscape view) and P (for Portrait view).

   @TL is for a Tablet Landscape view.

   @TP is for a Tablet Portrait view.

   @PL is for a Phone Landscape view.

   @PP is for a Phone Portrait view.

   If `<orientation_firstLetter>` is not given, then by default, the Desktop view is shown.

   Example:

   ```
   showView('::Acme.WesternDivision.Sales','Forecasts@TL','1')
   ```

7. Click **OK** through the open dialogs.

Working with Client-side Expressions

Basically, expressions that execute within a view and are configured in the XHQ Workbench are client-side expressions. Expressions that execute within the XHQ Solution Server and are configured in the XHQ Solution Builder are server-side expressions.

For more information on XHQ expressions, see the topic, *Expressions*, located in the XHQ Reference Guide.

Examples of Client-side Expressions:

```
"Mytext1"+"\\"+"Mytest2"
replace(sMember,"\\","/")
indexOf(sMember,"\\")
lastIndexOf(sMember,"\\")
findIndexOf(sMember,"\\")
"This is a 10\" diameter pipe"
```

As seen in the last example above, strings with embedded double-quotes are allowed.

When working with client-side expressions, local variables are set up to be displayed within the view. These variables are local to the given view. Therefore, there is no way to refer to these variables from another view within a view. As such, if you want to calculate, for example, a variable within View A that uses a variable from View B, you either need to calculate the value server-side (from the XHQ Solution Builder), or pass the variable into the JavaScript environment.

In the following example, the values are all accessible within the same view. A local variable is set up by dropping a value onto the view and naming the variable.

EXAMPLE: USING CLIENT-SIDE EXPRESSIONS

*Given:*The view is open in the XHQ Workbench.

1. Select the **Value tool**.
2. **Click on the drawing area** at the desired location for the value item.
   Note the string of pound signs, ####. At this point, an *un-typed* value (variable) is created. Before you can name it or use it, a value or an expression must be assigned to it.
3. With the value item still selected, click the **Animation tab**.
4. Click **Add** and, from the shortcut menu, select **Value**.
5. In the **Evaluating** box, enter either a constant or an expression.
   Once a value (in this case, an expression) is entered, the variable becomes *typed* and the pound signs are replaced by the string, 1234. Now the value can be named.
6. Click the **General tab**.
   Note the default value name, `ValueN` (where N is some consecutive number).
7. Replace the default name with a new, meaningful name.

To use this variable in another expression, refer to it by name and prefixing it with the dollar sign ($). For example, `$Value1+$Value2`.

Like the Value item, a Control can be added to the view and named, as a result, creating another local variable that can be used in expressions.

Client-side Expressions and Secure Data

Client-side expressions may return GOOD quality results when accessing secure data, even though the value of the secure data is not actually retrieved. This may cause existing visualization expressions to behave differently and possibly break applications.

Using Animation

An *animation* is a self-contained view element that dynamically changes characteristics such as fill color and size based on fluctuations in backend system data. Animations must be configured to link through a solution model to collect data from backend data sources. By modifying display properties based on changes in values in backend systems, animations help draw attention to specific information.

The combination of display characteristic (or animation), animation type, and evaluation variable influences the animation effects that show at runtime. You can apply several animations against the different display characteristics of a view element. For example, dynamic changes in text color, background color, size, and location can produce the complex effect of a sphere moving around the screen while blinking to annunciate alert conditions, and changing in size to indicate the severity of the problem.

Animation Effects

Display Characteristic (Animation)	Animation Type	Variables
• X location • Y location • Text color • Show/hide • Width • Height • Fill color • Line color	• Value range • Value breakpoint • Value • Value tag	• Real • Integer • Boolean • String

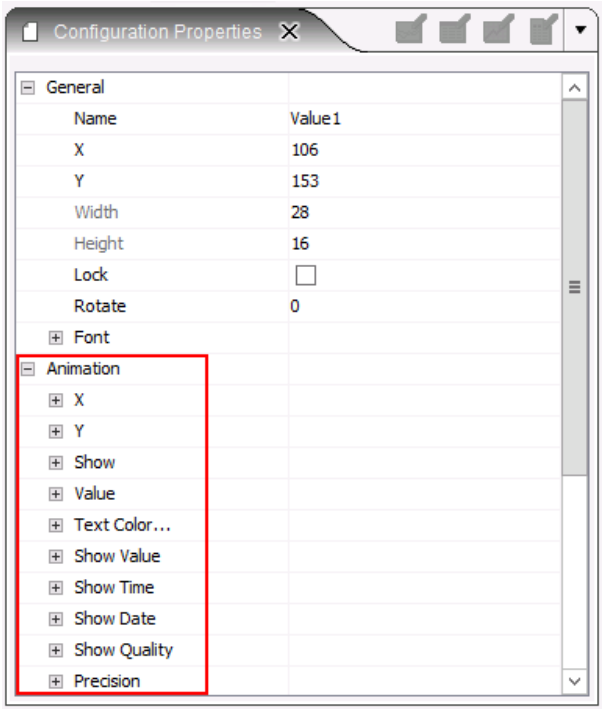
The Value breakpoint type is not available for non-color-related animation, such as Width, Height, Show, and Value.

Some combinations of display characteristics and animation types are unnatural. The system guides you by presenting choices that are relevant to your previous selection. For example, if you choose to show the fill color of a rectangle changing based on value breakpoints in a (real) value, the system will not present the option for you to choose a Boolean.

The values that you use to control animations are derived from specific component members that you configure in the XHQ Solution Builder.

Display Characteristics - Animations

The list of available display characteristics (or animations) varies, depending on the type of view element you are animating.



Example of Available Characteristics

When you add an animation, it appears in the Animation table and is removed from the list of available options you can add. By adding an animation characteristic, you can, for example, make a line change color based on variations in a data value. Similarly, you can set a view element size as well as its position on the display to change dynamically.

You can only apply one animation of a particular type to a particular view element. For example, you cannot select Fill Color twice for one view element.

Animation Characteristics

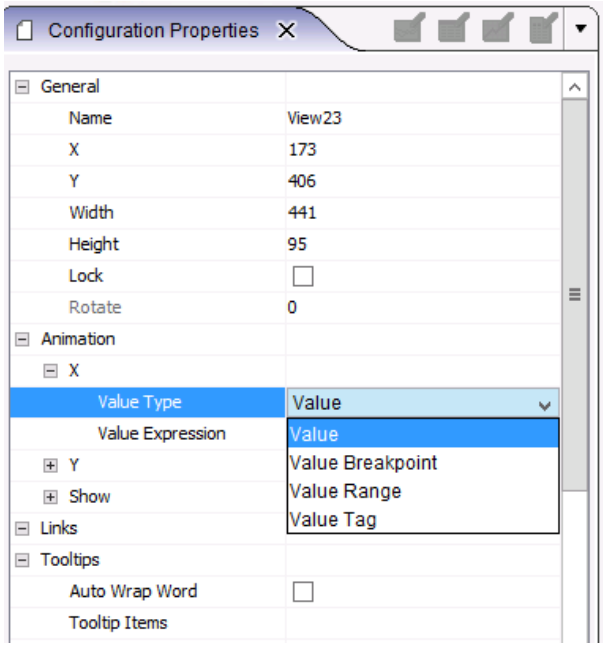
Display Characteristic (Animation)	Description
X and Y location	Control the horizontal/vertical location of a view element based on data values.
Text Color	Controls the color of the view element based on data values.
Text	Presents the value of a text string.
Value	Control how values are displayed.
Width and Height	Control the size of an object based on data values.
Fill and Line Color	Control the color of an object based on data

Display Characteristic (Animation)	Description
	values.
Show or Hide	Controls whether or not an object is displayed based on data values.

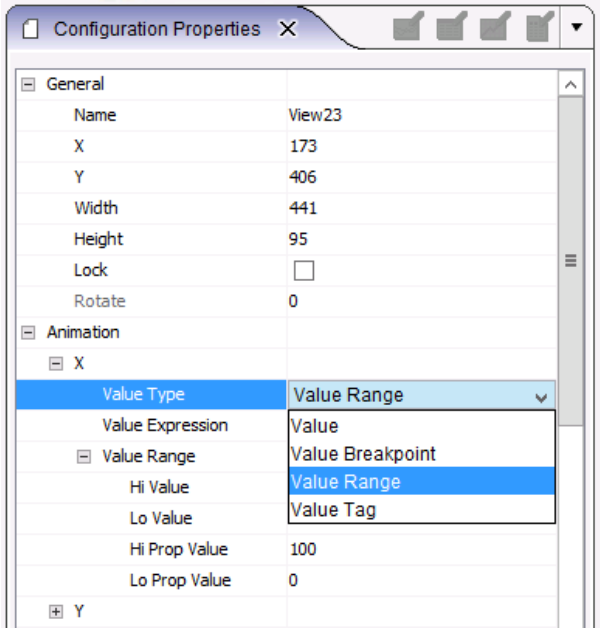
Animation Types

The XHQ Workbench permits four types of animations for creating complex, dynamic view elements.

- **Value**
  This animation type pertains to the display of text values only.



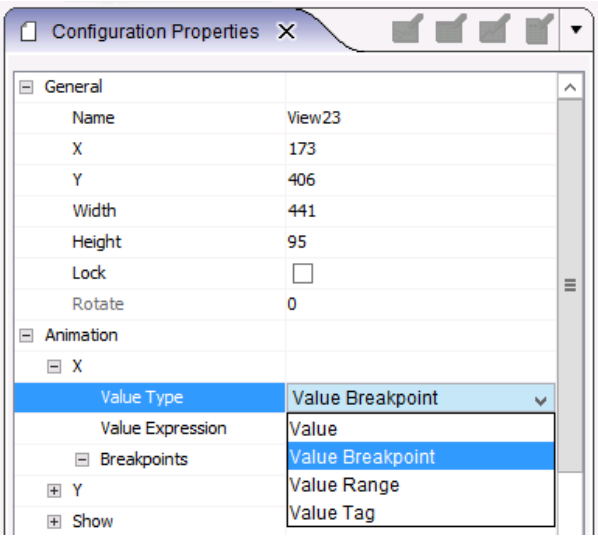
- **Value Range**
  Used to limit the activity of an animation between two range limits of a variable. You might use a range animation to control your display of a view element of a vessel, where you want to show animation activity by varying the height of a view element between 10 and 90 pixels, based on the incoming value.



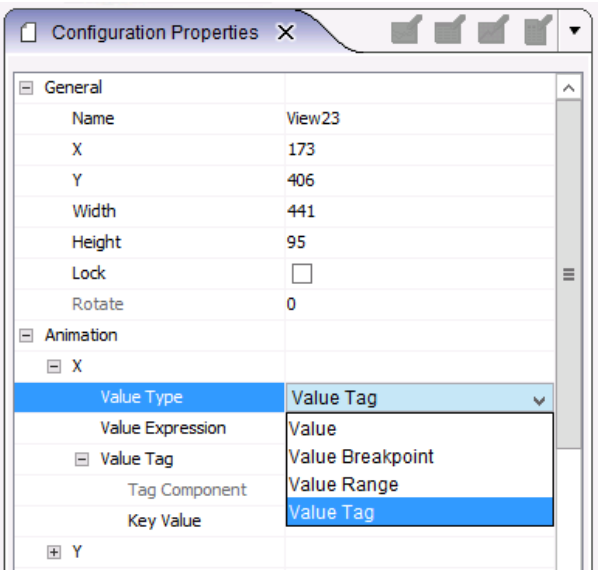
- **Value Breakpoint**
  Used to change the display in specific steps as an input value varies. Value Breakpoints can be set for any color

animation. For example, you might display the number of orders in process with an icon that shifts from white to green to yellow to red in steps at 50, 100, and 200 orders.

The Value breakpoint type is not available for non-color-related animation, such as Width, Height, Show, and Value.



- **Value Tag**
  Used to display the value of a member of a tag component, which is referenced by its key in a tag collection.



Setting Animation Properties

In general, you can animate various, display view elements with a very simple, interactive technique. The view element is visible during the configuration process, and you enter all configuration parameters through the Edit panel. The following procedure describes the basic steps required to animate a view element.

To animate a view element

1. Select the view element.
2. Select the **Animation** tab to view the animation dialog.
3. Click **Add** and, from the shortcut menu, select an animation from the list of available options.
   The animation you select is displayed in the "Set" column of the Animation table.

   The options listed on this drop-down shortcut menu depends on the type of view element you are animating.

4. From the **Using** drop-down list, select the type of animation.
   The animation type you select is displayed in the "Using" column of the Animation table.
5. From the **Evaluating** drop-down list, select the primitive member to drive the animation.
   The members listed are those contained in the component for which you are building the current view.
6. OPTIONAL
   If you want to drive the animation from a primitive member that belongs to a component further down in the solution hierarchy, such as a grandchild, you can **enter in the name of the component** in the evaluating field using the form:

   ```
   component2.component3.member
   ```

   Where:
   `component2` is a child of the current component (which would be `component1`), `component3` is a grandchild of `component1`, and member is a primitive member of `component 3`.

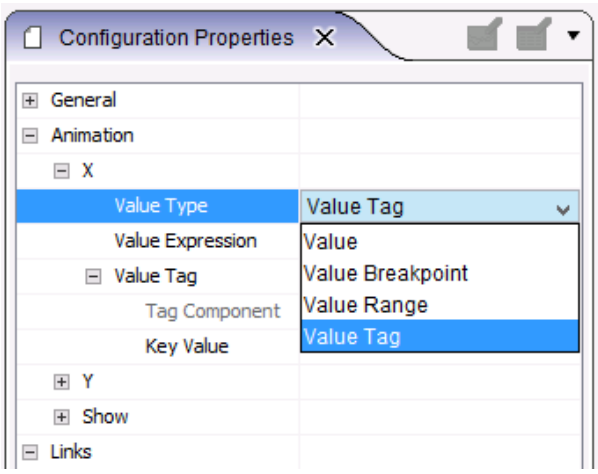
   The evaluating primitive member you select is displayed in the "Evaluating" column of the Animation table.

   You can include any number of nested component members, but the last member in the string must always be a primitive member.

7. Depending upon your previous selections, you may be required to enter particular configuration parameters that match your choice of display characteristic (or animation) and animation type.
8. Click the **Save** icon on the button bar.

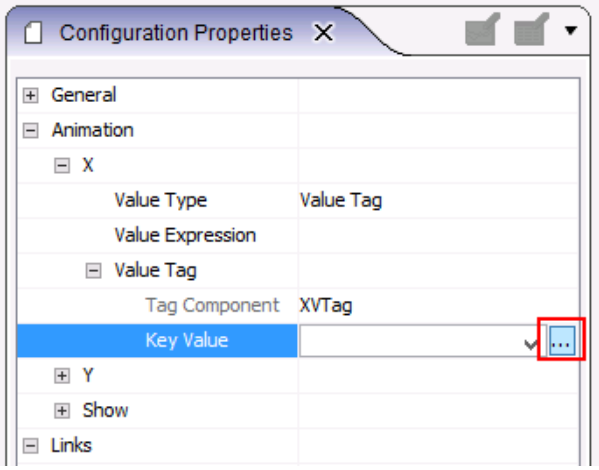
To set a Value Tag animation

1. From the **Tools palette** of the XHQ Workbench, click the **Value tool** and place the value on the Drawing panel.
   A value element placeholder appears.
2. From the **Edit panel**, click the **Configuration tab** and expand **Animation**.
3. For **Value Type**, select the **Value Tag** animation type.



The animation type you select is displayed in the "Using" column of the Animation table.

4. Next to the **Tag Component** text box, click the browse button.
   The "Select Tag Component" dialog box appears.
5. Select from the list of tag components available and then click **OK**.
6. From the **Key Value** field, do one of the following:
   - Click the **browse button**. The "Tag Selector" dialog box appears; **select a unique key value** and click **OK**. The "Key Value" field is then populated with the unique key value you selected.



or

   - **Enter a string** expression that results in the name of the desired tag.
7. **Save** the view.

Animating with Color

The color of your view elements can change in many ways based on changes in data being provided by the backend systems. You can dynamically change color of:

- The fill of any regular or irregular shape.
- The fill color of any text string.

You can change the color based on value, value range, and value breakpoint.

Test your color combinations for readability before you deploy your solution. For example, you may not want to use white for uncertain quality values especially if your background is white.

The evaluation variable used to show color changes can either be real or integer primitive member. The configuration entries, and the resulting dialogs presented to you based on your choice, will be different based on the animation type.

Animation Examples

The following examples demonstrate some common applications of animation.

Perhaps you want to show a rectangle that will change color based on the inventory level of a particular item. You may choose to show a white square to indicate normal, a green square at 50, and a yellow square at 100. To implement such a scheme, you will use a color fill animation, with value breakpoint animation, against the number of orders from the database.

EXAMPLE 1: ANIMATING WITH COLOR USING VALUE BREAKPOINT

In this example, assume you want to show the inventory for an item in an updating text field, and then change the color of the displayed background gold to red when the inventory level drops below 50 (the value of a break-point). In this way the viewer is alerted to an abnormal condition.

1. Configure the solution to extract the value from the backend data source and place it in the solution.
2. From the **XHQ Workbench**, click anywhere in the view element of the rectangle that you want to animate.
3. In the **Edit Panel**, click the **Animation** tab.
4. Click **Add**.
   A drop-down list is displayed that contains supported set options.
5. Click **Fill Color**.
6. From the **Using** drop-down list, click **Value Breakpoint**.
   The Value Breakpoint options appear to the right.
7. In the text box located below the Add (Breakpoint value) button (directly to the left of the Breakpoints text box), **enter** the breakpoint value of **50** for the variable.
8. Click **Add**.
   The breakpoint value you entered appears in the Breakpoints text box.
   Two small square icons appear under Color column.
9. Click on the bottom **Color** icon to activate the color picker and select **red**.
   This is the color you desire the square to appear when the order number is below the breakpoint value of 50.

10. Click the top **Color** icon to activate the color picker and select **gold**.
    This is the color you desire the square to appear when the order number is above or equal to the breakpoint value.
11. From the **Evaluating** drop-down list, select the primitive member to drive the animation.
12. Click the **Save** icon on the button bar.
13. From the XHQ Solution Builder, configure the solution and view the animation in runtime mode.

EXAMPLE 2: ANIMATING WITH COLOR USING VALUE RANGE

In this example, assume you would like to create a bar chart that shows the average standard cost of a unit. You can animate the height of a rectangle to simulate the varying height of the bar.

1. Configure the solution to extract the average standard unit cost from the backend database and place it in the solution.
2. Click anywhere in the view element of the rectangle that you want to animate.
3. Click **Add**.
   A drop-down list is displayed that contains supported set options.
4. Select **Height** from the list.
5. From the **Using** drop-down list, click **Value Range**.
   The Value Range options appear to the right.
6. In the field that corresponds to **HI Value**, enter the input value at the maximum end of the range.
7. In the field that corresponds to **HI Prop Value**, enter the desired maximum height of the view element in pixels.
8. In the field that corresponds to **LO Value**, enter the input value at the minimum end of the range.
9. In the field that corresponds to **LO Prop Value**, enter the desired minimum height of the view element in pixels.
10. From the **Evaluating** drop-down list, select the primitive member to drive the animation.
11. Click the **Save** icon on the button bar.
12. From the XHQ Solution Builder, configure the solution and view the animation in runtime mode.

EXAMPLE 3: ANIMATING WITH EXPRESSIONS

In this example, you use an expression to animate a graphical view element. Given, a component called **`PumpStation`** that has three members assigned to it: **`Pump1`**, **`Pump2`**, and **`Pump3`**. Each member has a primitive assigned to it called **`FlowRate`**. You build a view against **`PumpStation`** called **`detail`**, in which you need to see the average flow rate for all three pumps.

1. Open the **`detail`** view.
2. Using the **Value** tool, insert a value.
3. Click the **Animation** tab.
4. Click on **Add**.
   A drop-down list is displayed that contains supported *set* options.
5. Select **Height** from the list.
6. Select **Value Range** from the **Using** drop-down list.
   The Animation table displays the values you have selected.
7. **Enter the expression** you want to drive the animation.
   For the purpose of this instruction, type:

```
(Pump1.FlowRate+Pump2.FlowRate+Pump3.FlowRate)/3
```

The expression instructs XHQ to take the flow rate of Pump1, add to it the flow rate of Pump2, add to it the flow rate of Pump3, and take that total value and divide it by three to return the average flow rate for all three pumps.

8. Press **Enter**.
   The evaluating primitive member you selected is displayed in the evaluating column of the Animation table.
9. In the field that corresponds to **HI Value**, enter the input value at the maximum end of the range.
10. In the field that corresponds to **HI Prop Value**, enter the desired maximum height of the view element in pixels.
11. In the field that corresponds to **LO Value**, enter the input value at the minimum end of the range.
12. In the field that corresponds to **LO Prop Value**, enter the desired minimum height of the view element in pixels.
13. Click the **Save** icon on the button bar.
14. From the XHQ Solution Builder, configure and save the solution, then view **`detail`** in runtime mode.

Precision Animation

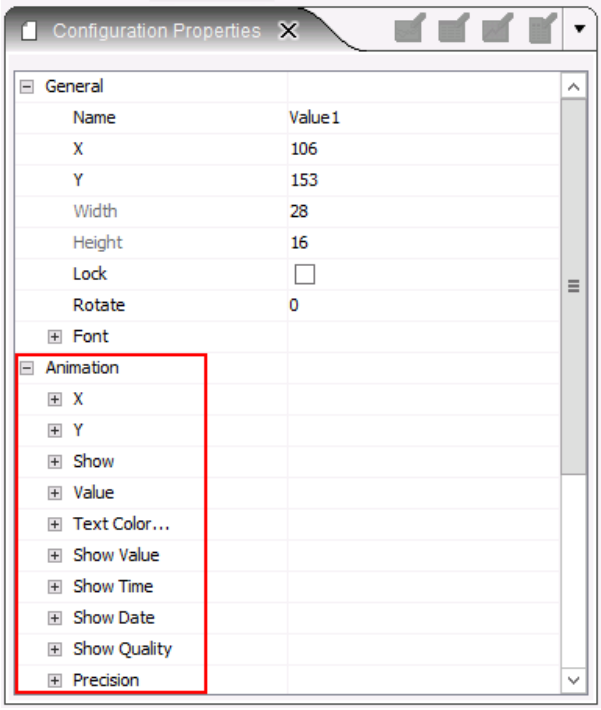
Precision animation provides the ability to dynamically change the number of digits shown after the decimal point of a given value item. This type of animation uses an expression that is based on any number of these data sources: client variables, view variables, points or collection data.

Important Things to Note

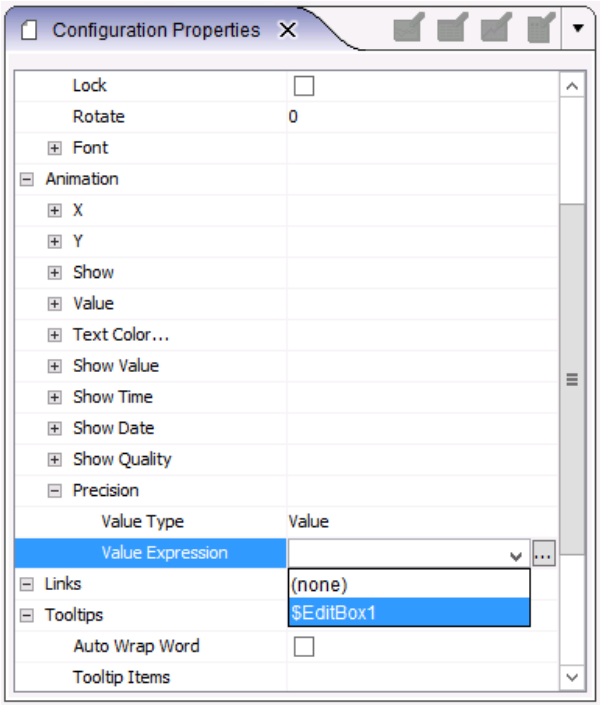
- You must set the "Number-Fixed" field, in the Value tab, to zero (0) digits in order for dynamic precision animations to work in a view.
- Do not use constant value expressions for precision animations.
- The "General" format forces the value item to two decimal places and prevents the precision animation from being applied.
- The type returned by the expression does not have to be an integer; it could be of another type that the animation will attempt to convert into an integer.

EXAMPLE: Creating a precision animation

1. From the XHQ Workbench, **insert a Value item** that is of type **REAL**.
2. **Select** the value item and go to the **Configuration Properties tab**.
3. Expand **Value** and for **Number Format**, select the **Number-Fixed**.
4. For **Digits**, enter 0.
5. Expand **Animation**.



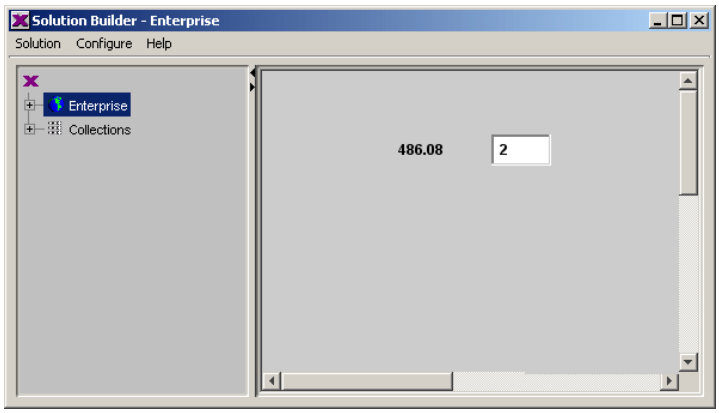
6. Then expand **Precision**.
7. Select the **Value type** and **enter** the appropriate **Value expression**.



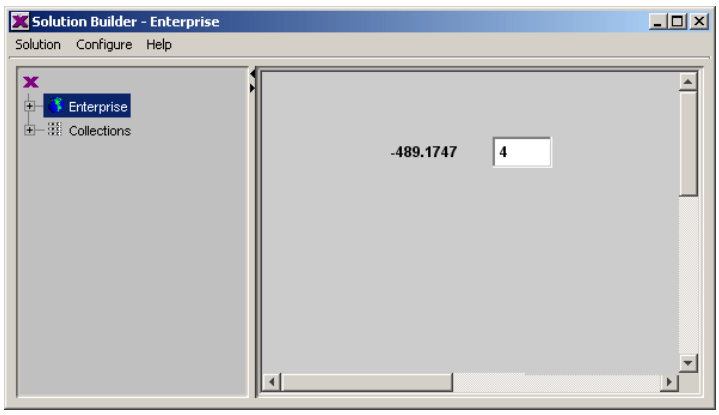
In this example, the precision animation is driven by an edit box control that handles integers.

8. **Save** the view.

During runtime, the edit box control shows the default value of "2"; therefore, the value item is displayed to two decimal places.



Changing the edit box value to "4" displays the value item to four decimal places.



The **maximum** number of decimal places allowed is **eight**. Any value greater is automatically rounded down to eight.

Using Group By

The XHQ Workbench provides an option to apply Group By and Aggregate functions to collections. The results of applying group by is available to all charts and tables, more specifically:

- Group By is supported for all collections except non-cached non-relational data (for example, no support for PassThru for XML and SAP3).
- Grouping options are configurable to support controls, with no limit on the number of times grouping can be applied to a single collection.
- The XHQ application developer can specify one or more fields as grouping fields.
- The XHQ application developer can apply an aggregate function to any field that isn't part of the grouping fields. These are called "grouped" fields. The following aggregate functions are supported:
  - Count
  - Average (mean) – ignores non-numeric values
  - Sum – ignores non-numeric values
  - Minimum
  - Maximum
- The XHQ application developer can supply a name to use when referred to grouped fields, subject to the rule that names must be unique within the query.
- A field that isn't part of the grouping fields can be used as many times as desired for grouping.

A combo box having the source referencing a group by table can use only the items in the table group by.

For the runtime option in the XHQ Solution Viewer, go to the topic, *Grouping Records in a Table*, located in the XHQ Solution Viewer User's Guide.

Sort and Filter

From the XHQ Workbench, you can configure **sorting** for *Table controls*, *Excel controls* and *Charts* based on *Group By* or Aggregate columns. Then at runtime, when a view with sorting configured is loaded, the user is able to interact with the control to sort any displayed column.

In addition to collection members, you can set **filtering** on Group by and Aggregate columns for *Table controls*, *Excel controls* and *Charts*.

Truncate and Extract

From the XHQ Workbench, *Group By* for Date/Time data types has two options: Extract and Truncate. The **Extract** option returns the part of the Date/Time you select. For example, extracting the "Month" from the given Date/Time example of 2018-05-04 13:55:55:555 returns a numeric value of 5.

The **Truncate** option rounds down the Date/Time to the nearest unit selected. For example, truncating "Month" from the same Date/Time example returns the Date/Time value of 2018-05-01 00:00:00:.000. Again, this is only for Date/Time data types. The Extract and Truncate parameters are: Minute, Hour, Day, Week, Month, Quarter, and Year.

The LEGACY embedded database does not support the Week and Quarter parameters for the Extract option.

Pasting a Copied View and Re-linking

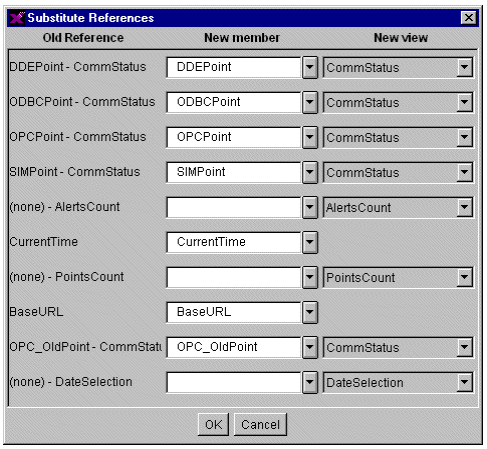
You can cut a selected portion of a view (and its associated animation associations) and paste it into a view that was created for a different component.

To animate correctly, a view needs data sources (component primitive members). Consequently, it makes most sense to copy a view or view element and paste it in a view that shares a similar data structure. If you paste a view to a component with identically named members, the animations are automatically re-established. If, however, the data members are different, the Substitute References dialog appears, allowing you to re-link the imported view's animations to the alternative primitive members of the target component.

When you copy and paste a **Chart** or **Table** (and its related value) to a different component, you must map ***substitute references***. In this case the collection member must be mapped before its value reference.

About Substitute References

The Substitute References dialog box allows you to accept automatic links or assign new data references.



Substitute References Example

The Substitute References dialog box has three columns.

- **Old Reference**
  Lists the component and view (either embedded or link), from which you are copying.
- **New member**
  Lists available components. Select the component that includes the view to which you are mapping to.
- **New view**
  Lists the available views of the component (new member) selected. Choose the view you are mapping to. For single components, flow views are not listed. For collection components, all views, including flows views, are listed.

To paste a view from another component and re-link

1. With a view displayed, use the **Select** tool to select the view or view element you want to copy.
   If you are just copying a part of the view, left click once and draw a square around the part of the view you want

to copy. Unclick when you have drawn a square around the portion of the view you want to copy.

2. Click the **Copy** icon.
3. From the **XHQ Workbench**, **right-click** on the component with the view to which you want to paste the copied view.
4. Point to **Edit View**, and from the submenu, **select** the view you want to paste to.
5. Click the **Paste** icon.
   The "Substitute References" window appears.
6. If the primitive members in the **Old Reference** field match the primitive members displayed in the **New Member** field, click **OK** to confirm the automatic linkages.
7. If some data member names are not found in the component where the view is being imported, you must re-link the view animation to alternative primitive members. Click on the **New member** drop-down list to re-link the view animation to an alternative primitive member.
8. Click **OK**.
   The view is pasted into the current view.

More About Pop-up Views

There are three ways to set the "Pop Up" mode in the XHQ system:

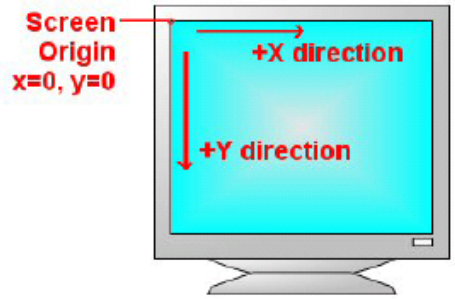
- A View link
- An Action link with the `ShowView` function
- A JavaScript `ShowView` function

By default, the pop-up view window appears at the center of the screen.

Once the "Pop Up" mode is applied, you can establish where on the screen the pop-up window will appear by setting the x- and y-coordinates of the top, left corner of the pop-up window.

> The coordinates you specify must place the top, left corner of the pop-up window *inside* the physical screen.

The follow image shows how the x,y position (of the top, left corner) is determined when a pop-up window is launched:



For example, if the x- and y-coordinates are set to (300,200), then the position of the top, left corner of the pop-up window will be 300 pixels to the right of the screen origin (0,0) and 200 pixels down.

> The pop-up window is automatically sized such that no scroll bars are needed. However, if the view size is greater than the screen size, then the pop-up window size is set to the screen size.

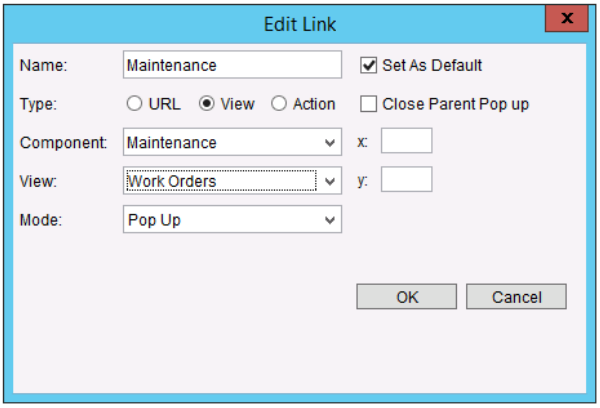
In the following sections, you will learn how to set the "Pop Up" mode for the three different methods mentioned above, and how to define the x- and y-coordinates.

Setting View Link Pop-ups

From the Links tab of the XHQ Workbench, you can set how the view will display when the view link is executed. Under **Mode**, you can select from the following:

- **Use views properties**
  You can *indirectly* set view links to pop-up through this mode if the Pop Up property is selected from the Views Properties tab.
- **Replace**
  This mode is the *traditional* behavior of the view link, which causes the existing view in the view panel to be replaced.

- **Pop Up**
  You can *directly* set view links to pop-up through this mode. It causes a new pop-up window to display the view.



View Link Pop Up

To set a pop-up view using View links

1. **Select the view element** (for which you want to add a view link), and click the **Links tab**.
2. Enter a descriptive name for the link in the **Name** field.
3. Under Link **Type**, select **View**.
   This creates link to another XHQ view.
4. Do one of the following:
   - In the **Component** drop-down list, **select a component** that contains the view to which you want to create a link.
     The components listed are the children of the component for which you are building the current view.

   or

   - If you want to select a component further down in the solution hierarchy, such as a grandchild, you can type in the name of the component using the form:

```
component2.component3.component4
```

   Where:
   `component2` is a child of the current component (which would be `component1`), and `component3` is the grandchild of `component1`, and so forth.

   For more information on parent-children relationships, go to the topic, *Defining Parents Children and Grandchildren*.

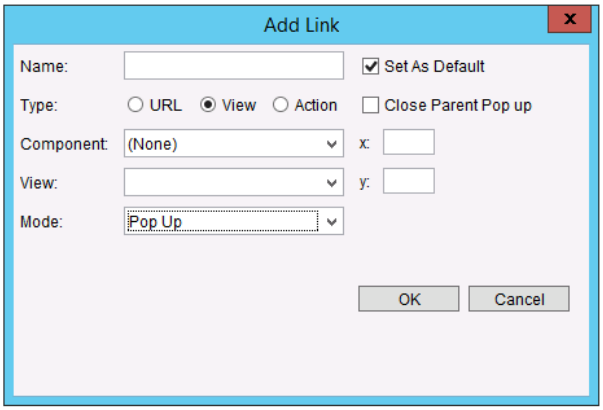
5. In the **View** drop-down list, select the view to which you want to create a link.
6. Under **Mode**, do one of the following:
   - Select **Use views properties** to apply pop-up settings defined in the "Views Properties" tab and proceed to step 9.

     For more information, go to the topic, *Setting Pop up Views from Views Properties*.

   or

   - Select **Pop Up** to pop-up a new window that displays the view you selected in step 5.
     When this option is selected, an "X" and a "Y" edit box appear to the right. To customize the placement of the

pop-up window, enter the X and Y values (in pixels) that define the coordinates of the top, left corner of the pop-up window.



The *Replace* mode loads the view link in the same window, replacing the previous view. It does not launch a new pop-up window.

7. Enter the **X** and **Y** coordinates of the top, left corner of the pop-up window.
   The X and Y values are in *pixels*.

   The coordinates you specify must place the top, left corner of the pop-up window *inside* the physical screen.

8. OPTIONAL
   If the parent window (the window where this view will be launched) is a pop-up, select **Close parent Popup** to close the parent window once this view is loaded.
9. OPTIONAL
   If you have multiple links set for a single view element, select **Set as Default** to specify this view as the default view for the pop-up window.
10. Click **Add**.
    The name of the view link appears in the Links list.
11. From the **File** menu, click **Save to repository**.

Setting Action Link Pop-ups

The Action link type allows you to perform actions specified by a function expression.

For more information on these functions and their parameters, refer to the topic, *Action Link Functions*, located in the XHQ Reference Guide.

The `ShowView` function, in particular, displays a view given the object path, view name, and mode. The syntax for this function is as follows:

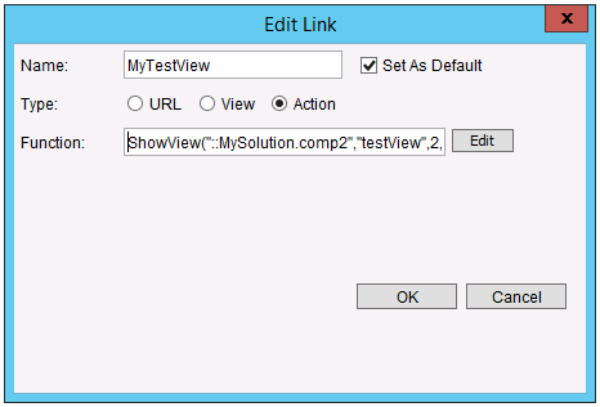
```
ShowView (objPath, viewName[,mode,autoClose,x,y])
```

Brackets [ ] indicate that the parameter is optional.

To launch an Action link pop-up, set the optional **`mode`** parameter to the value of "2". To define where on the screen the pop-up view window appears, also set the **`XY`** **coordinates** of the top, left corner of the pop-up window.

To use the XY parameters, you must set the **mode** to a value of 2-"popup" and specify an **autoClose** value.

```
ShowView("::MySolution.comp2","testView",2,0,300,300)
```



Action Link Pop Up

To set a pop-up view using Action link ShowView function

1. **Select the view element** (for which you want to add a view link), and click the **Links tab**.
2. Enter a descriptive name for the link in the **Name** field.
3. Under Link **Type**, select **Action**.
4. Enter a function expression in the **Function** text box, using the following example syntax.

   Example: `ShowView (objPath, viewName[,mode,autoClose,x,y])`

5. Set the **`mode`** parameter to 2.
   This is the "Pop Up" mode.

   If the specified mode is 0-"use views properties" and the view has a pop-up defined in the Views Properties tab, then the default pop-up position is applied (which is at the center of the screen).

6. OPTIONAL
   If the parent window (the window where this view will be launched from) is a pop-up, then do one of the following:

   - Set the **`autoClose`** parameter to **0** if you do not want the parent window to automatically close once the action link is activated.

   or

   - Set **`autoClose`** to **1** if you want the parent window to automatically close once the action link is activated.

7. OPTIONAL
   To customize the placement of the pop-up view window, define the **`x`**- and **`y`-coordinates** of the top, left corner of the pop-up window.

   Example:
   ```
   ShowView("::MySolution.comp1", "defaultView",
   2,0,300,200)
   ```

   The coordinates you specify must place the top, left corner of the pop-up window *inside* the physical screen (or the coordinates will be overridden).

8. OPTIONAL
   Click **Set as Default** to specify this view as the default view.
9. Click **Add**.
   The name of the function appears in the Links list.
10. From the **File** menu, click **Save to repository**.

Closing Pop-up Views

By default, if the parent window is a pop-up itself, then it will remain open after the pop-up view is launched. To automatically close the pop-up parent window, use the following methods:

- For View links, enable the **Close parent Popup** option.

  For information on how to use this option, go to the topic, *Setting View Link Pop ups*.

- For Action links, define the **`autoClose`** parameter of the **`ShowView`** function.

  For information on how to use this option, go to the topic, *Setting View Link Pop ups*.

- Use the function **closeAllPopUpViews()**.

  For more information on this function, refer to the topic, *closeAllPopUpViews*, located in the XHQ Reference Guide.

- From an action link configuration, add the **closeWindow(1)** system function to the end of the list of functions.

  Example: eval("showView('::Top','circle',2,null,0,100)"), closeWindow(1)

  For more information on this function, refer to the topic, *closeWindow*, located in the XHQ Reference Guide.

Working with Embedded Views

When you use embedded views, you deal with component view hierarchies. Typically, there are four types of views in a hierarchy. The top-most view represents a *thumbnail* preview of the component. Next, you find views that give you a summary of information about the component. These are the same type of views that would be in a collection or flow view. Then, you have a level of views that provide even more details. And lastly, you have base views.

Component view hierarchies can start with very simple, base views. You can use these base views as building blocks to create more complex views that include the characteristics of the base view.

Creating such a hierarchy is extremely helpful. You can make changes to a base view and in turn, reflect the changes on all the other views that embed it.

As a solution best practice, resizing embedded views (which includes key views) is not recommended since it may result in an undesired sizing of the view. Resizing the embedded view affects every element in the view (including text and lines) proportionally. If an embedded view or a key view needs to have a dynamic resizing behavior, build it with animation to accommodate this behavior.

Base views are typically not exposed at the browser client level as views and are primarily used as tools for the application developer to reduce the cost of maintaining views.

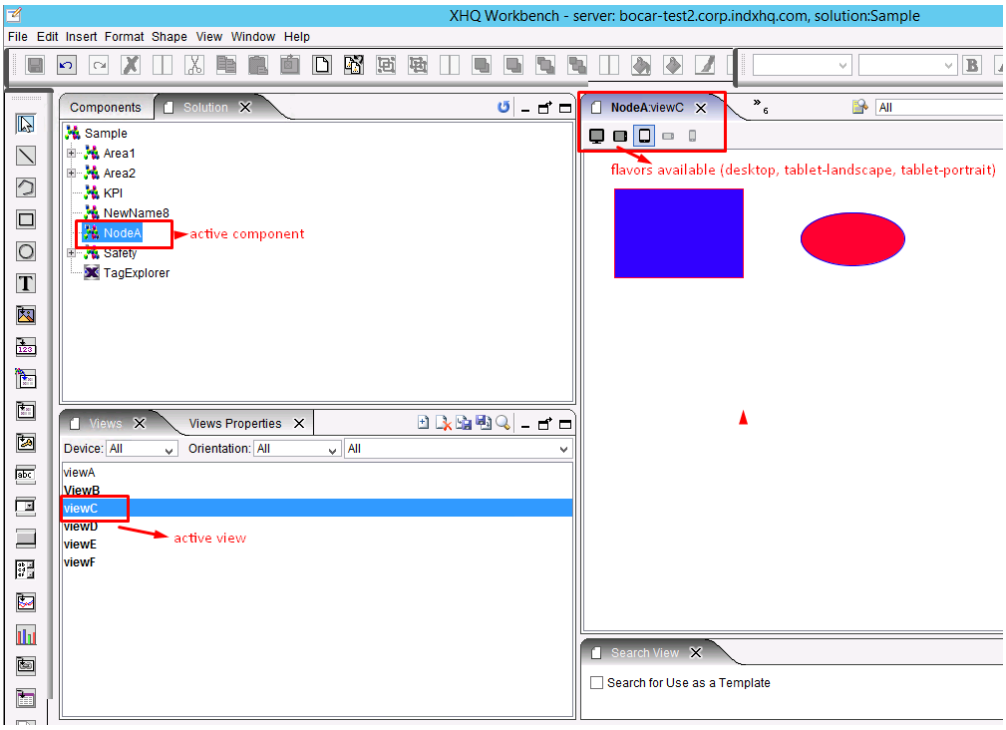
Now that you are familiar with the hierarchical structure of embedded views, the following sections will show you how to set navigation between the different levels.

The Views Tab

From the **Views** tab, you can set the device and orientation combinations. The *flavor* refers to the **device**-specific display (desktop, tablet, mobile phone) and the **orientation** type (landscape or portrait). This is also known as *view flavor* or *device-oriented view*.

Currently, there are five viewing *flavors*:

- Desktop
- Tablet-Landscape
- Tablet-Portait
- Phone-Landspace
- Phone-Portait



Example: View with 3 flavors - Desktop, Tablet-Landscape, Tablet-Portrait

At design-time in XHQ Workbench, a view will always have at least one flavor. At runtime, the view is loaded based on the available flavor.

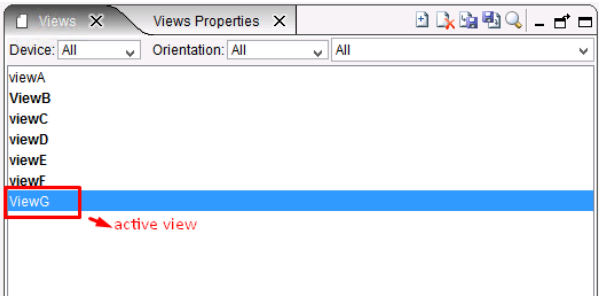
Important Things to Note

- For an existing solution, prior to this implementation, that already has multiple targeted flavors of configured views, no action is needed to fix those configurations. A visual indication in the workbench is shown for view properties with mismatch. For example, if a view has mismatched properties, it is highlighted in red and no property is displayed for that view. The user must correct this mismatching by re-saving the properties.
- In case of an import from a prior solution, the import ignores the device and orientation and proceeds sequentially. In case the device and orientation are available for non-desktop flavors, a log message is displayed indicating that the properties are updated to the view.
- At runtime, if a view is displayed in a device and orientation that has no flavor, then the view uses the fallback feature.

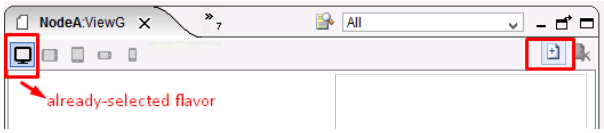
For more information on the fallback property, see the topic, *Fallback*, located in the XHQ Administrator's Guide.

To add view flavors

1. From the **Views tab**, select the view.

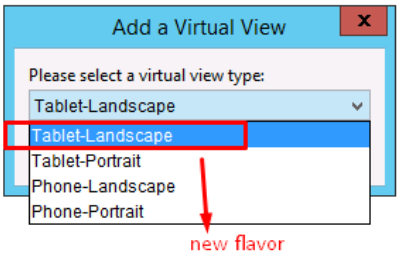


The view-specific tab appears in main Drawing Panel.



2. From the tab bar, click on the **Add icon** .

   The "Add a Virtual View" dialog appears.

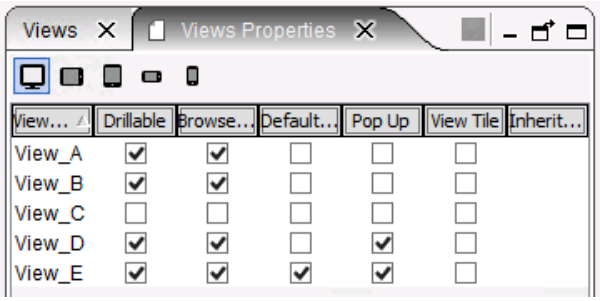


3. Click the drop-down arrow and select a view flavor.

Setting Views Properties

From the **Views Properties** tab, you can

- Identify drillable views.
- Select views that appear on the browser menu selection list.
- Designate a default view.
- Set pop-up views.
- Designate view tiles which are used to compose personalized views in the XHQ Solution Viewer.



Example: Views Properties tab

You access Views Properties by right-clicking on the component in the XHQ Workbench, and selecting the Views Properties command from the shortcut menu.

The View Name column lists all the views for the given component. The first time the Views Properties tab is launched, all views are drillable and appear on the browser menu.

Views that are listed but have a gray background (View Name column) are read-only. These views belong to a base component and therefore their view property settings cannot be edited.

For more information on base components, go to the topic, *About Base and Subclass Components*.

About Drillable Views

This feature allows a user to explore a view in a hierarchical manner, by pointing and clicking through a series of embedded views within a given component view.

From the Views Properties tab, you select which of the component's views are drillable. During runtime, when the user points to a view that is drillable, a rectangular blue line appears around the view element to indicate it is selectable.

Selectable View element

By default, the first time the Views Properties tab is launched, all views are drillable. So, if you have several layers of embedded views within a view, each embedded view will appear selectable. Normally, when using embedded views, you would enable all views to be drillable from the Views Properties tab.

Base views are an exception because you typically set them as non-drillable; nor do you allow them to appear on the browser context menu.

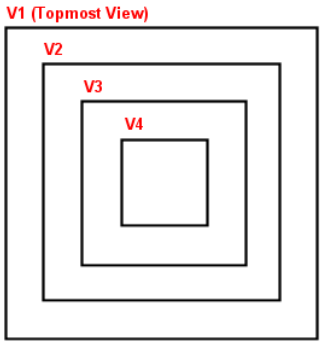
To make a view drillable or non-drillable

1. From the **XHQ Workbench**, **right-click on the component**, and from the shortcut menu, click **Views Properties**. The "Views Properties" panel appears.
2. Under **Drillable**, select the view(s) you want drillable by marking the appropriate checkbox. Or, deselect the view (s) you want non-drillable by clearing the checkbox.
3. Click **OK**.

Setting a non-drillable, embedded view

It is important to note that embedded views within a non-drillable view will also appear non-drillable. A non-drillable view behaves like a static view element (for example, a shape).

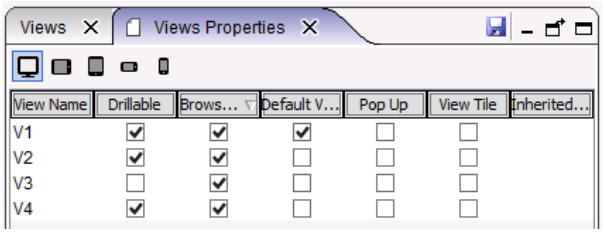
Consider the following example.



Embedded Views Example

Where V1 is the topmost view, and V4 is embedded in V3, V3 embedded in V2, and V2 embedded in V1.

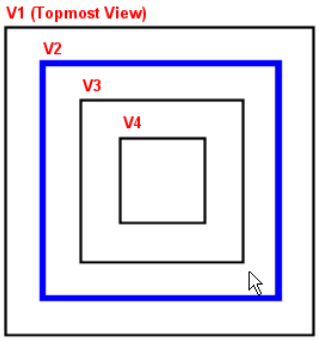
From the Views Properties panel, you set V3 as non-drillable by deselecting the checkbox.



Views Properties – Setting Drillable Options

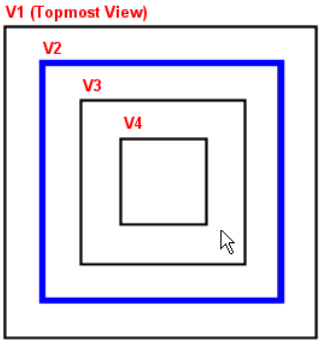
What will this look like during runtime?

When the user points to the area of V2, a blue outline appears around V2 indicating that it is drillable.



Pointing to a Drillable View (V2)

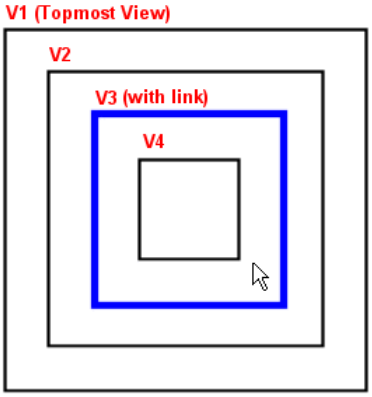
When the user points to the area of V3, an outline still appears around V2 but does not appear around V3, indicating that it is non-drillable. And, although V4 is set to drillable in the Views Properties tab, it too will appear non-drillable because the "containing" view, V3, is non-drillable.



V3 is non-drillable and V2 is drillable

During runtime, when you right-click a non-drillable view (in this case, V3) XHQ checks for the next view up that is drillable (which is V2) and shows its (V2s) available views and links in the given shortcut menu. And, although V3 is non-drillable, the right-click shortcut menu will display the default view of V2 as bold.

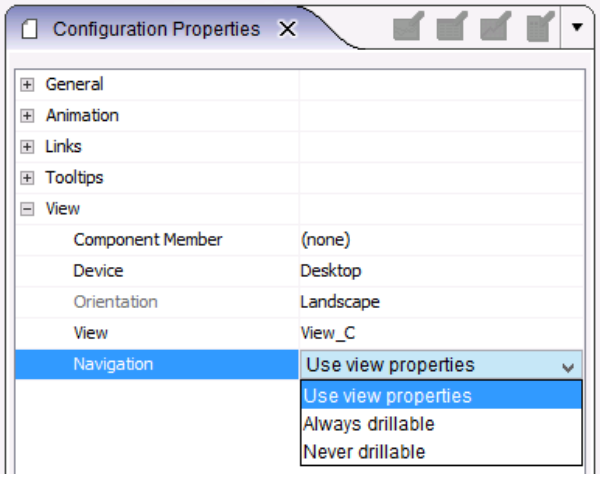
However, if the non-drillable view (V3) contains a view or URL link, then the blue outline does appear when you point to it, indicating that it is selectable. And clicking on it will take you to that link.



Pointing to a non-drillable view with a link.

Overriding the Views Properties - Drillable setting

You can override the Views Properties setting of an embedded view through the Configuration Properties tab. You simply select the embedded view and edit its Navigation setting.



Configuration Properties Tab – Navigation Settings

From the Navigation list

Select …	If you want to …
Use views properties	Use the settings defined in the Views Properties tab.
Always drillable	Set the view as always drillable, regardless of whether the view is, by default, drillable or not.
Never drillable	Set the view as never drillable, regardless of whether the view is, by default, drillable or not.

If you set a view as "Always drillable" and embed it (so it now becomes a child) into another view (the parent), which is set to "Never drillable", then the parent view's "Never drillable" setting overrides the child's "Always drillable" setting.

And, if you set a view as "Never drillable" and embed it into another view, which is set to "Always drillable", then the parent's "Always drillable" setting overrides the child's "Never drillable" setting.

To override the Views Properties setting

1. From the **XHQ Workbench**, **right-click on the component**.
   The shortcut menu appears.
2. Point to **Edit View**, and from the submenu, **select the view**.
3. **Click on the view element**.
   Selection handles appear around the view element.
4. From the **Configuration Properties tab**, expand **View**.
5. In the **Navigation** list, do one of the following:
   - Select **Always drillable**. This sets the view to drillable.

   or

   - Select **Never drillable**. This sets the view to non-drillable.
6. Click **Save to repository** and close XHQ Workbench.

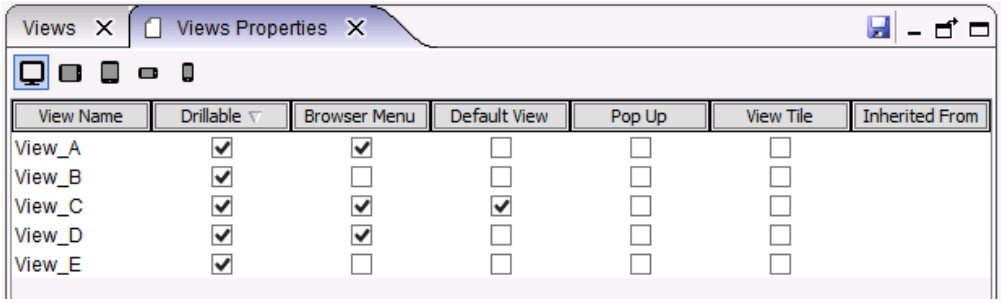
About Views on the Browser Menu

From the **Views Properties tab**, you select the views that appear in the right-click shortcut menu of the runtime client. Views that appear on this list are easily selected by the end-user.

The first time the Views Properties panel is launched, all views are set to appear on the shortcut menu. In most cases, this is the setting you would use. An exception would be the use of "base views", which are used internally and are usually not exposed to the end-user. To designate a base view, you would need to clear (deselect) the checkbox.

Consider the following example.

From the Views Properties panel, clear the checkbox next to View B and View E. Both views will no longer be listed on the right-click shortcut menu. The default view is View C.



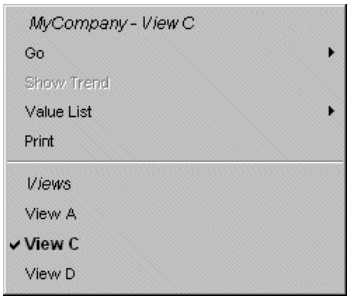
Views Properties – Setting Browser Menu and Default View Options

From the XHQ Solution Builder, in View mode (runtime), the right-click shortcut menu will look something like this:

Where:

- The views that were enabled (from the Browser Menu column of the Views Properties panel) appear in the middle panel under *Views*.
- The views that were disabled (not selected) appear in the bottom panel.
- A command separator separates the enabled and disabled views.
- The default view, View C, is in bold.
- The name of the current view appears at the top of the menu, and on the view list with a checkmark next to it.

From the browser client, during runtime, the right-click shortcut menu will look something like this:



In this mode, only the views that are enabled from the Views Properties tab are listed.

The checkmark indicates that the view is currently open.

To include a view on the shortcut menu list

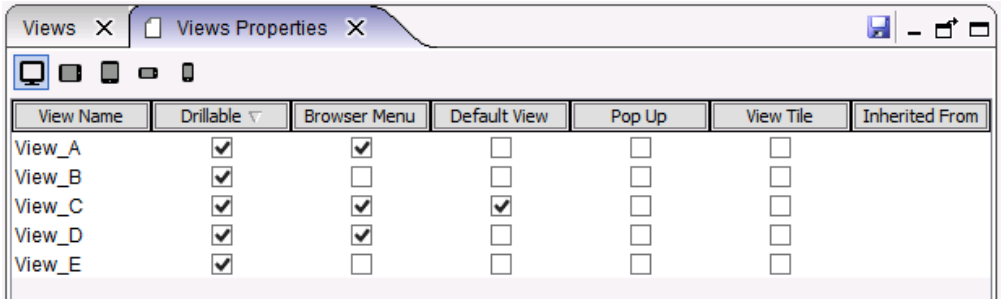
1. From the **XHQ Workbench**, **right-click on the component**, and from the shortcut menu, click **Views Properties**. The Views Properties panel appears.
2. Under **Browser Menu**, select the view(s) you want available by marking the appropriate checkbox. Or, deselect the view(s) you do not want listed by clearing the checkbox.
3. Click **OK**.

Overriding the Browser Menu Setting

In exceptional cases when you may need to override the Browser Menu selections you set from Views Properties, use view links. The view link will appear under the *Links* list in the shortcut menu even though you did not select it to appear on the Browser Menu, *Views* list.

Consider the following example.

Given: This set of Views Properties



Although View B is not selected in the image above, it appears on the Browser Menu as a view link.

Browser Menu with View Link

Setting the Default View

If views exist for a component, one will always be set as the default view. On the shortcut menu, views are listed alphabetically and the default view appears in **bold**.

There are two cases when you may not see a view in bold: if the default is not enabled (from the Browser Menu of the Views Properties tab); or if the default is a link.

When you add views for the first time to a component, the first view you create becomes the default. In this case, the default assignment is no longer dependent on the view name. If the current default view is deleted, however, then the first view on the list becomes the default.

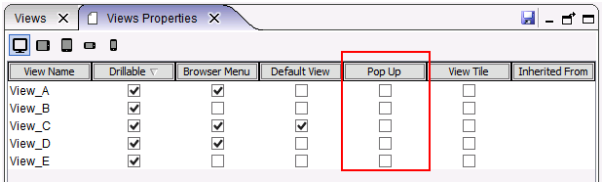
To set the default view through Views Properties

1. From the **XHQ Workbench**, **right-click on the component**, and from the shortcut menu, click **Views Properties**. The "Views Properties" panel appears.
2. Under **Default**, select your default view by marking the appropriate checkbox.
3. Click **OK**.

The XHQ System allows you to set a default view that has been excluded from the shortcut menu View list. And so, when you click on a component in the browser client, it is possible to access a view that does not appear on the shortcut menu.

Setting Pop-up Views from Views Properties

This Views Properties feature allows designated views to launch in a separate frame that is independent of the main view panel of the XHQ Solution Builder or the XHQ Solution Viewer (browser client).



Views Properties tab – Pop Up column

To set a pop-up view through Views Properties

1. From the **XHQ Workbench**, **right-click on the component**, and from the shortcut menu, click **Views Properties**. The "Views Properties" panel appears.
2. Under **Pop Up**, select the views to pop-up by marking the appropriate checkbox.
3. Click **OK**.

Viewing Pop-ups at Runtime

From the XHQ Solution Builder, pop-up links display the designated view in a small window that "pops up" on the screen. You can access these views either from the right-click shortcut menu or the navigation tree.

Pop-up View – from the XHQ Solution Builder

Views launched by view links designated as pop-ups, either directly (Mode = Pop Up) or indirectly (Mode = Use views properties, with Pop Up property selected), are displayed in a separate frame. These views are accessed from the right-click shortcut menu of the view panel.

This is also true of view links embedded in an element in the current active view.

Embedding Descendant Views

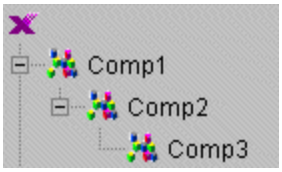
As stated in a previous topic, the concept of *parent, child,* and *grandchild* pertains to the nesting of components and views. Assume Component1 contains Component2, which contains Component3. Component1 is the *parent*. Component2 is therefore a *child* of Component1, while Component3 is a *grandchild* of Component1.

With this concept in mind, you can embed views that belong to other components further down in the solution hierarchy.

Consider this example on how to embed descendant views.

EXAMPLE: HOW TO EMBED DESCENDANT VIEWS

Given:

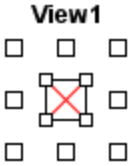


Where: Comp1 has View1

Comp2 has View2

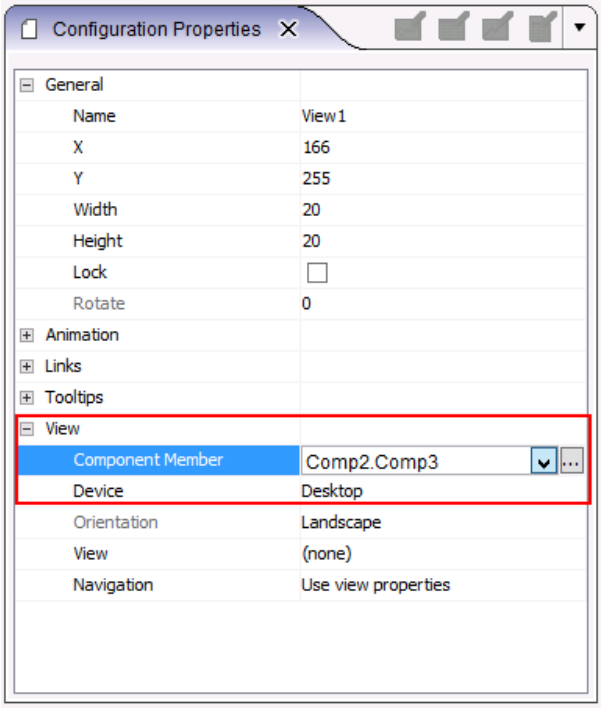
Comp3 has View3

1. From the **XHQ Workbench**, right-click on **Comp1** and, from the shortcut menu, **edit View1**.
   The XHQ Workbench appears.
2. Click the **Insert view tool** icon.
3. Place the cursor where you want to insert the view, and then click.
   A small square with selection handles appears.



4. Select the **Configuration Properties** tab.
5. Expand View.
6. In the **Component Member** field, enter the component path of Comp3 relative to Comp1.

   Enter: Comp2.Comp3



This action populates the View list with Comp3 views.

7. For the **View** field, select **View3**.
   This embeds View3 into View1.

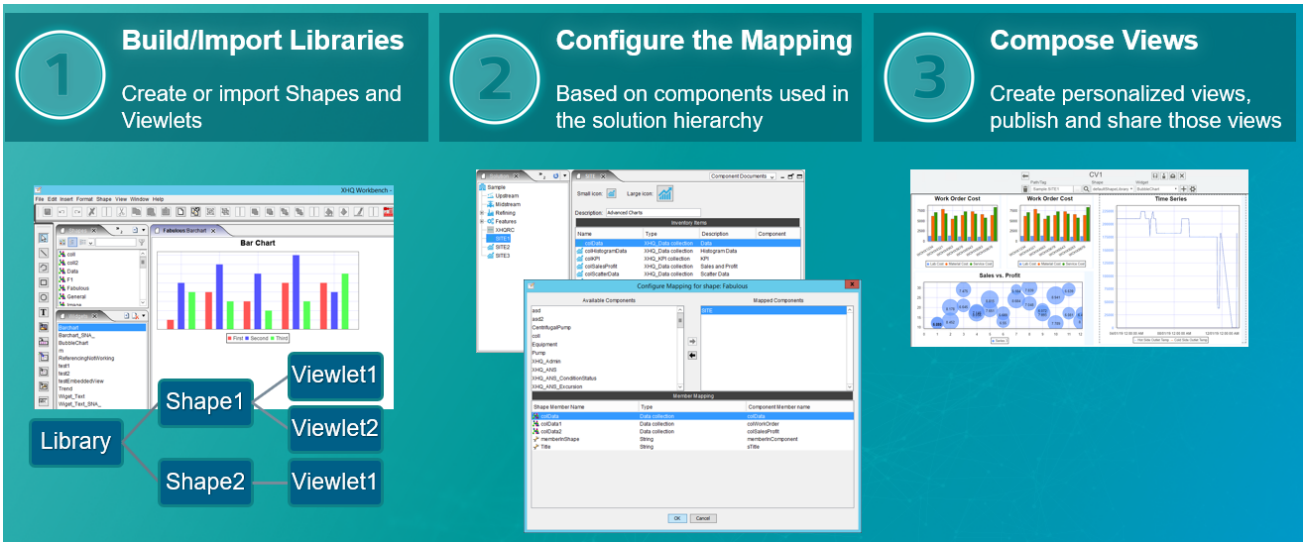
Working with the XHQ Visual Tile Composer

To work with the XHQ Visual Tile Composer (VTC), first get familiar with the following terms.

VTC Term Definitions

Term	Definition
Composed View (CV)	Also known as Tiled View or Tile Composed View (TCV). A CV is a view created using the VTC web-based tool. A CV is composed of "Tiles".
Library	In XHQ terms, it's analogous to a *Page*. A library is a collection of shapes. A user, with the right permissions, can create, delete, and modify an existing library. Libraries are portable from one solution to another where shapes with a library cannot reference any shape or component outside the library.
Mapping	Is the process of configuring the relationship between shapes and components. Multiple shapes can be mapped to a single component and vice-versa.
Shape	In XHQ terms, it's analogous to a *Component* or *Class*. A shape is created in the XHQ Workbench, similar to creating components. Each shape has members, like components. A shape belongs to a "Library" and is mapped to existing components.
Tile	Also known as a Live Visual Tile (LVT), Live Tile (LT) or a Widget. A tile is the building block an end user sees in the VTC. It which contains live and contextualized graphical elements. A tile is either a view that has the "Tile" view property checked, or a viewlet in a shape that is mapped to a component.
Viewlets	In XHQ terms, it's analogous to a *View*. A viewlet is a special type of view that is built using the XHQ Workbench. Viewlets belongs to a "Shape". Viewlets are more restrictive than XHQ views, in that keyed views and ENOs cannot be used in a viewlet.
Visual Tile Composer (VTC)	The VTC is a web-based tool to create, modify, view, and manage Composed Views. This runtime view building environment allows users with no XHQ Workbench background to build and assemble views from existing and predefined building blocks.

There are three main steps to using VTC:



1. Step 1: Build or import Libraries from the XHQ Workbench
2. Step 2: Map Shapes from the XHQ Workbench
3. Step 3: Build Composed Views from XHQ Solution Viewer

For information on the XHQ Visual Tile Composer and how to create a Tile Composed View, see the section, *Visual Tile Composer*, located in the XHQ Solution Viewer User's Guide.

Step 1: Building/Importing Libraries

In this section, you will learn how to:

1. Enable the Shapes and Viewlets tabs.
2. Build a Library of Shapes.
3. Work with Viewlets.
4. Resize Tiles.

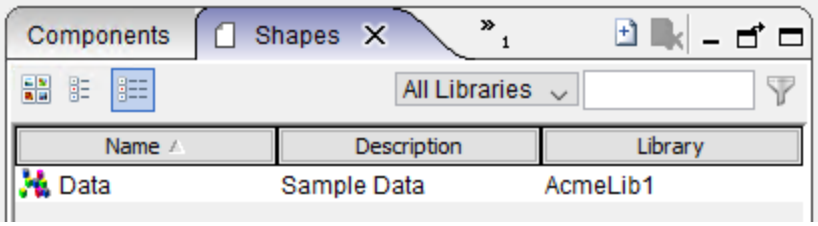
About Libraries, Shapes, Viewlets

- Shapes are comparable to XHQ Components and contain Viewlets.
- Viewlets are comparable to XHQ Views and contain visual elements.
- Libraries, shapes, and viewlets can be imported/exported.
- Shapes and viewlets are saved in the repos.

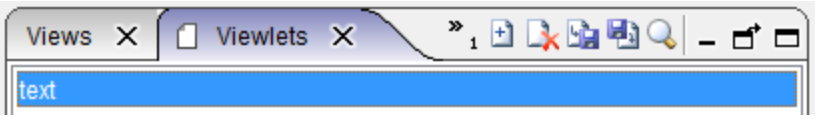
First, make sure the Shapes and Viewlets tabs are enabled.

To enable the Shapes and Viewlets tabs

1. From the **View menu**, select **Show Dockable** and click **Shapes**.
   The Shapes tab appears in the same panel as the Components tab.



2. Repeat and click **Viewlets**.
   The Viewlets tab appears in the same panel as the Views tab.

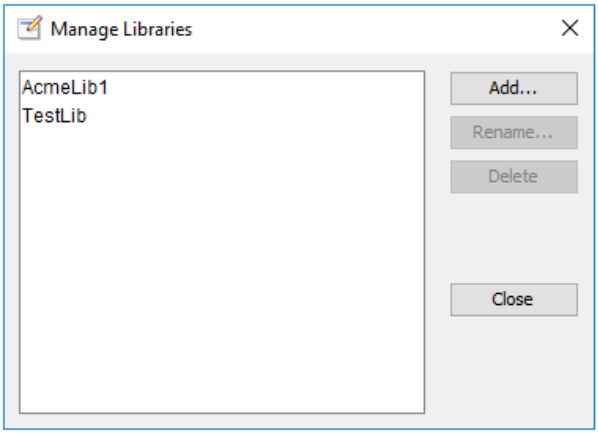


Once these tabs are enabled, you can start building the library.

Building a Library of Shapes

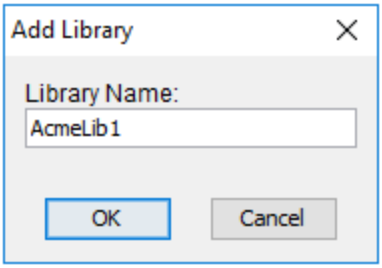
Before creating/importing shapes, you must first create the library.

From the **Manage Libraries** dialog, you can add, rename, and delete shape libraries.



To add a library

1. From the **File menu**, click **Manage Libraries**.
   The "Manage Libraries" dialog appears.
2. Click **Add**.
   The "Add Library" dialog appears.



3. Enter the **Library Name** and click **OK**.
4. Close the open dialogs.

To **import shapes**, go to the topic, *Shapes, Viewlets, and Mappings Export and Import*.

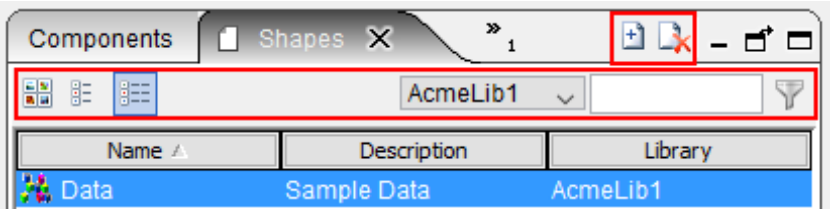
To **create shapes**, go to the topic, *Working with Shapes*.

After the library has been built, it's time to open the viewlet.

Working with Shapes

Creating a Shape is similar to creating an XHQ component. A shape is always associated with a library.

From the *Shapes Panel Editor*, you can add or delete shapes. You can also set how shapes are viewed or select a library from which to filter shapes.



Shapes Panel Editor

To **import shapes**, go to the topic, *Shapes, Viewlets, and Mappings Export and Import*.

Deleting a shape also deletes all mappings.

To create a shape

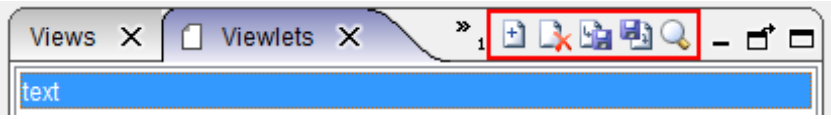
1. Select the library to which you want to add the new shape.
2. From the **Shape Panel Editor**, click the **New icon**.
   The "Add New Shape" dialog appears. The "Shape Library Name" automatically selects the active library.
3. Enter a **New Shape Name**.

Shape subclassing is similar to component subclassing. For more information, go to the topic, *About Base and Subclass Components*.

Working with Viewlets

Building Viewlets for a given shape is similar to building views for a given component.

From the *Viewlets Panel Editor*, you can create, delete, export, import, and search viewlets.



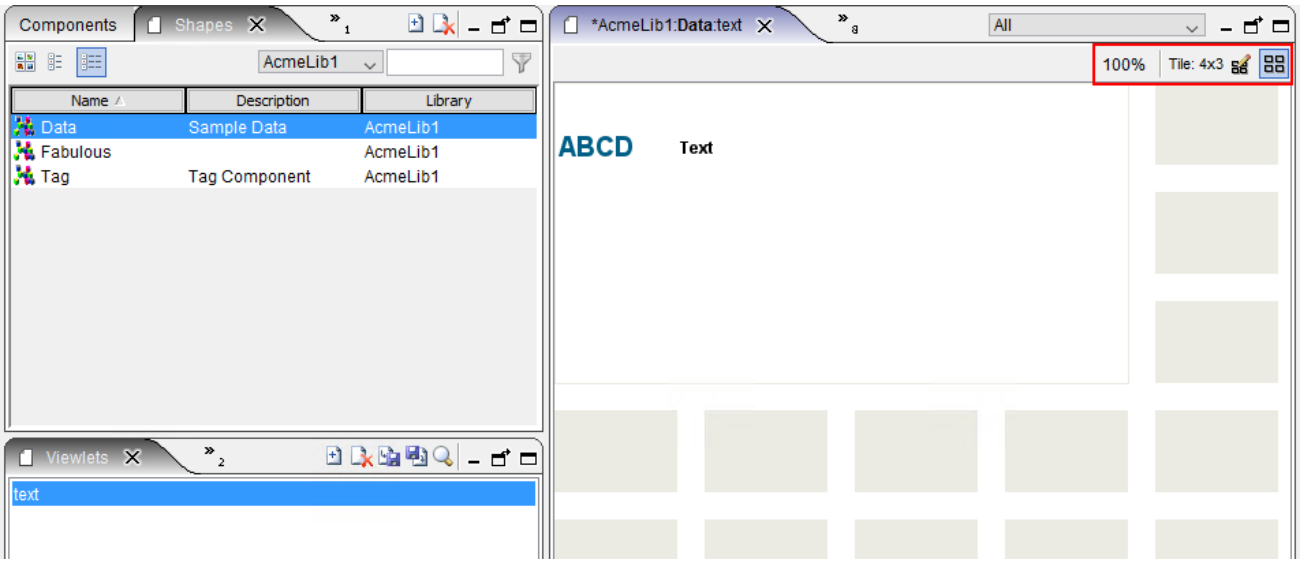
Viewlets Panel Editor

Due to library portability (exporting/importing between solutions), template use in viewlets is limited to templates that are in the same library. You will not be able to reference templates outside of the library.

Viewlets cannot include Keyed Views, View Instances, and ENO (Extended Namespace Objects) Tables. Again, this is so that libraries remain portable and not solution dependent.

The XHQ Workbench does not allow copy/paste actions when source and destination libraries are different. All paste options are disabled when the destination view is associated with a different shape library.

When you select a viewlet, it appears on the XHQ Workbench main panel.



Viewlet Grid Example

When sizing a viewlet, you measure the width (using the number of grid *columns*) by the height (using the number of grid *rows*). So, in the example above, the viewlet size is 4 grid columns by 3 grid rows, or 4x3 (as seen on the Panel Editor).

About the Grid

- The drawing area of the viewlet consists of 12 grids.
- The width of each grid is 90 pixels.
- The height of each grid is 60 pixels.
- The margin between each grid is 20 pixels.

To open a viewlet

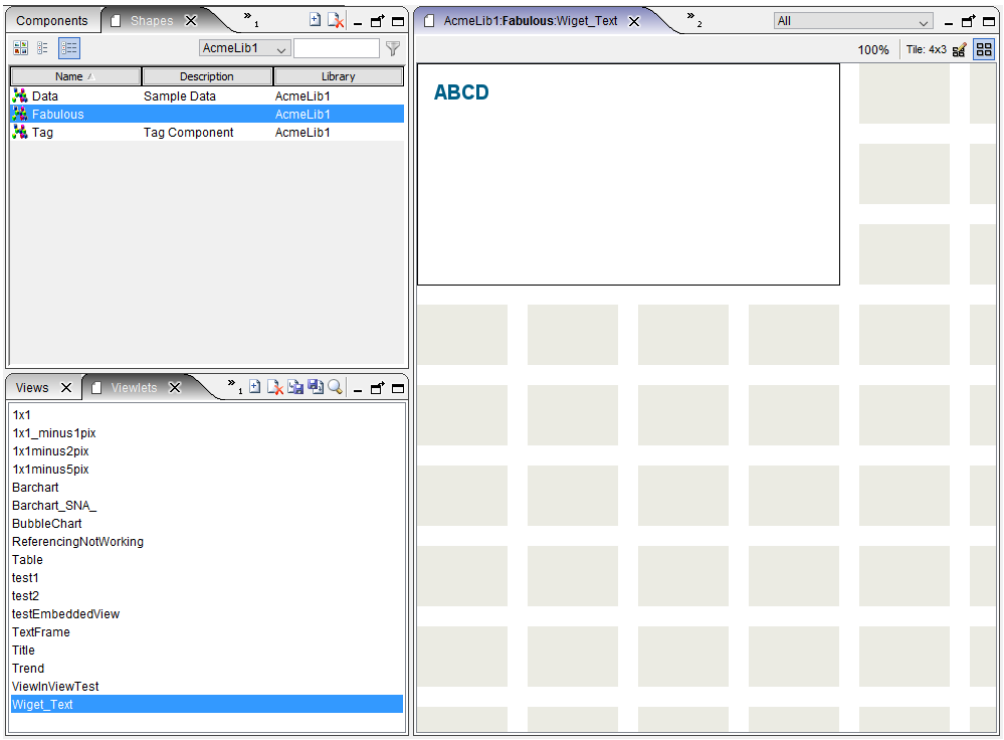
In the **Shapes tab**, select the shape and do one of the following:

- **Right-click** and, from the shortcut menu, select **Edit Viewlet** and click the viewlet.

or

- Go to the **Viewlets tab** and click the viewlet.

The tile appears as a tab in the main panel.



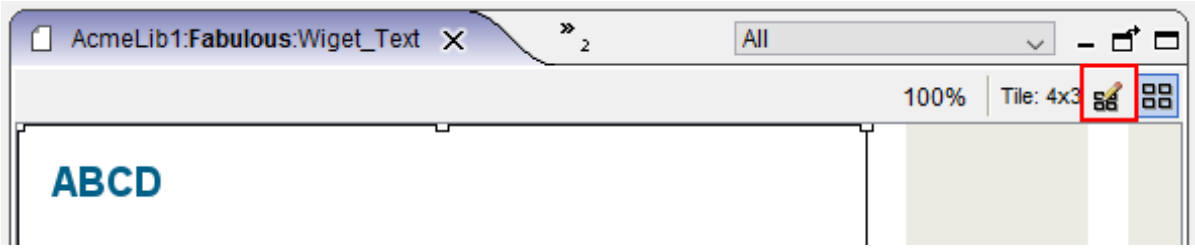
After building the viewlet, the final step is to work with the tile.

Working with Tiles

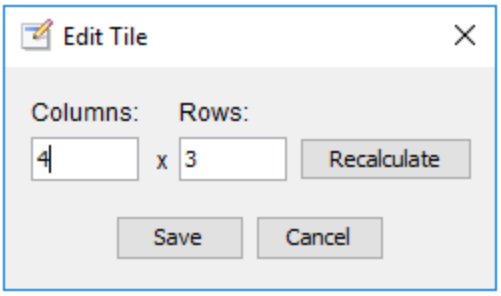
Viewlets are made up of Tiles. The tile is automatically sized on a grid system. The developer determines the tile size according to the grid.

To resize the tile

1. From the *Panel Editor*, click **Edit Tile**.



The "Edit Tile" dialog appears.



2. Enter the grid **Columns** and grid **Rows**.
3. *Optional*
   Click Recalculate to revert back to the original tile size.
4. Click **Save**.

This concludes *Step 1: Building/Importing Libraries*. The next step is to map shapes to the solution.

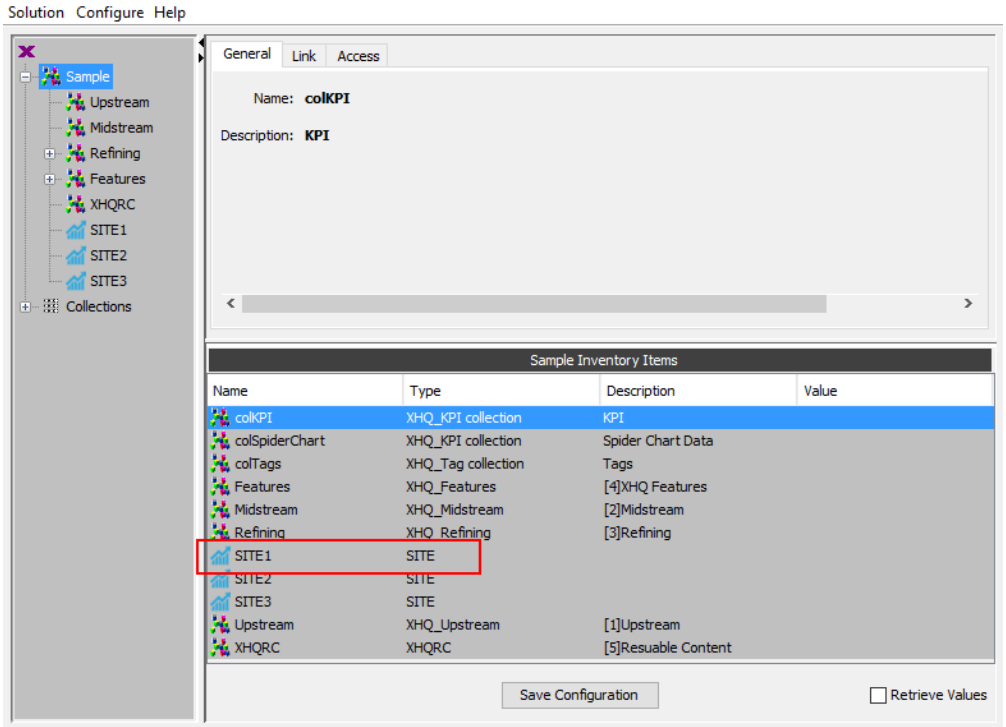
Step 2: Mapping Shapes

Mapping is the process of configuring the relationship between shapes and components.

This example builds on the procedures from Step 1.which showed you how to build a library, import a shape, and place a text tile.

EXAMPLE: Mapping components to shapes

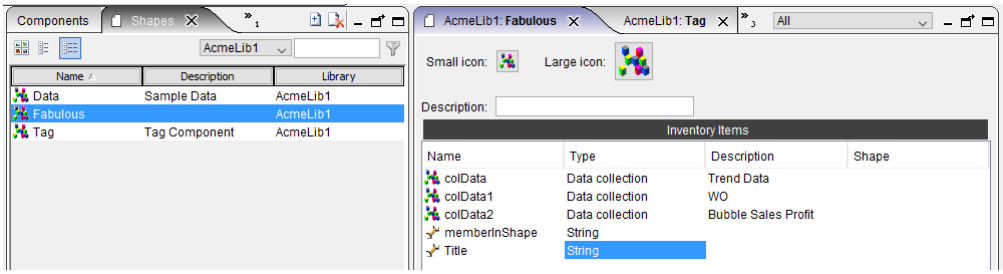
1. Open the **XHQ Solution Builder**.
2. Click the solution root **node**.
3. From the **Inventory Items** for node, locate the component to which you want to map and note the **Type**.



In this case, the component type for the SITE1 component instance is **SITE**.

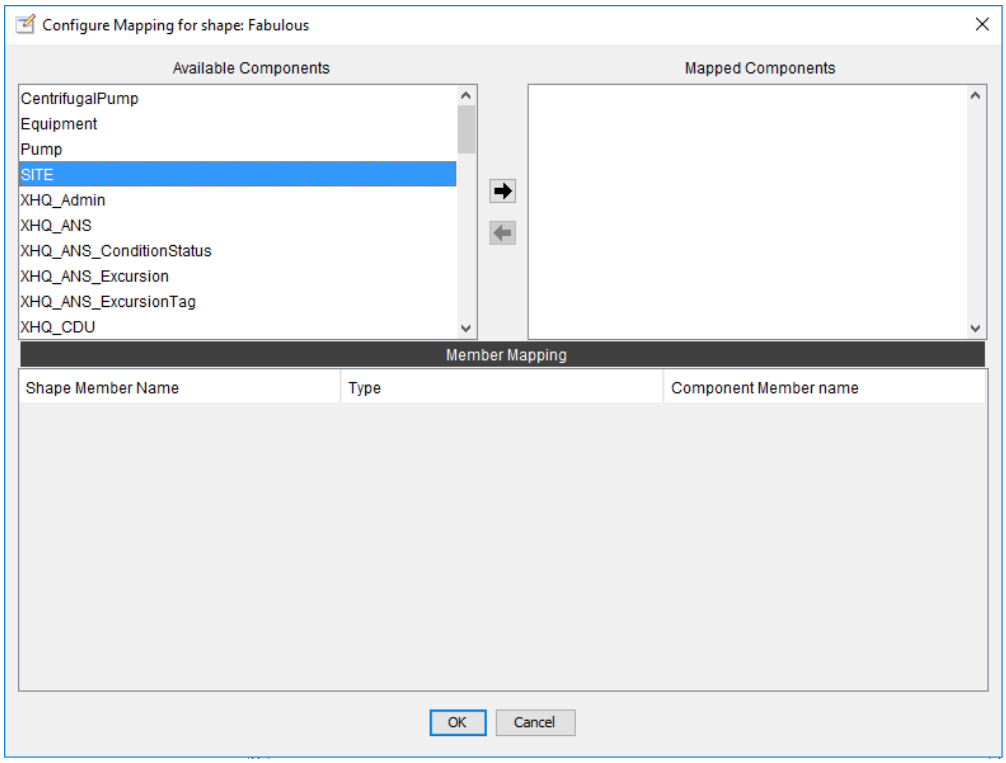
Next, map the "sTitle" member of the SITE1 component instance to the "Title" member of the shape that includes the text widget.

4. Go back to the XHQ Workbench.
5. In the **Shape tab**, double-click the shape that includes the text widget.
   This opens the Inventory Items for the shape.

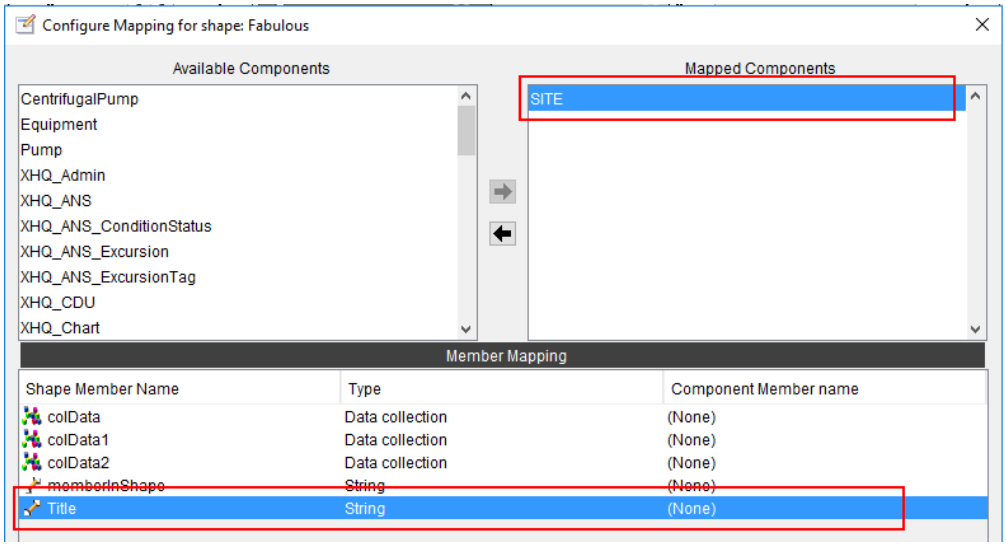


Note the "Title" member of type "String".

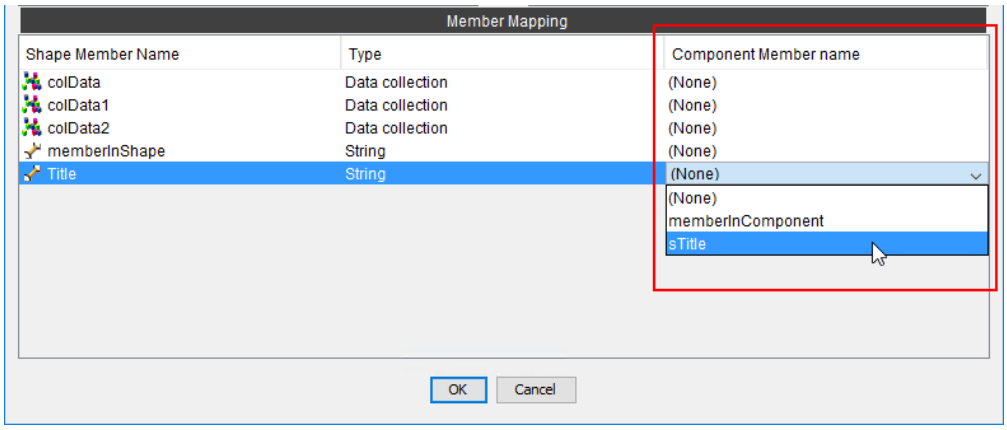
6. Next, right-click the same shape in step 5 and, from the shortcut menu, click **Configure Mapping**.
   The "Configure Mapping" dialog for the given shape appears.



7. Under **Available Components**, locate and select the component.
8. Click the **RIGHT arrow** to add it to the **Mapped Components** list.
9. Click the mapped component to view the **Member Mapping** list.



10. Select the **Title** member and under the **Component Member Name** column, **click the cell** to view an expanded members list.



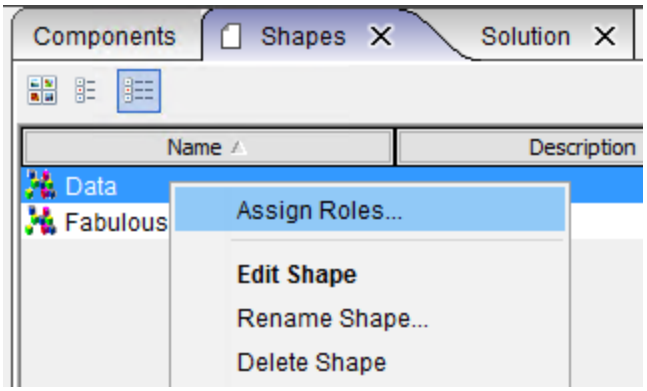
11. Select the **sTitle** member and click **OK**.

This concludes Step 2 of *Working with the XHQ Visual Tile Composer*. For the final step, you will create **Composed Views** from the XHQ Solution Viewer.

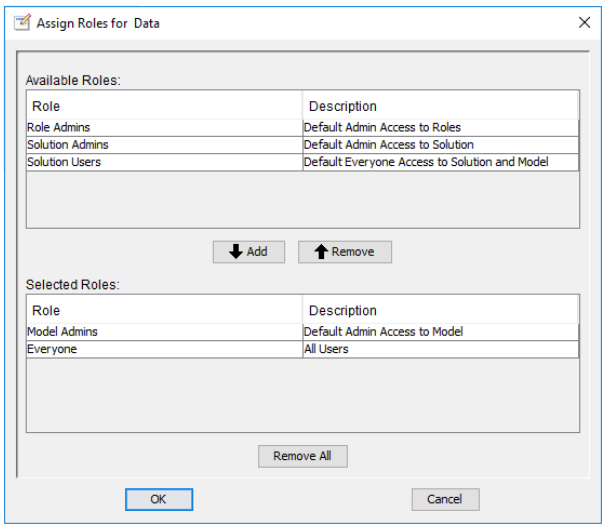
For information on the XHQ Visual Tile Composer and how to create a Tile Composed View, see the section, *Visual Tile Composer*, located in the XHQ Solution Viewer User's Guide.

About VTC Security

Like components, you can assign **roles** to shapes.



This launches the "Assign Roles" dialog box.



For more information on XHQ security, see the section, *About Roles*, located in the XHQ Administrator's Guide.

You must have access to the component to be able to create/edit/delete a mapping.

To assign **rights**, from the XHQ Workbench > View menu, select Role Administration.

The assignable XHQ rights for VTC are as follows:

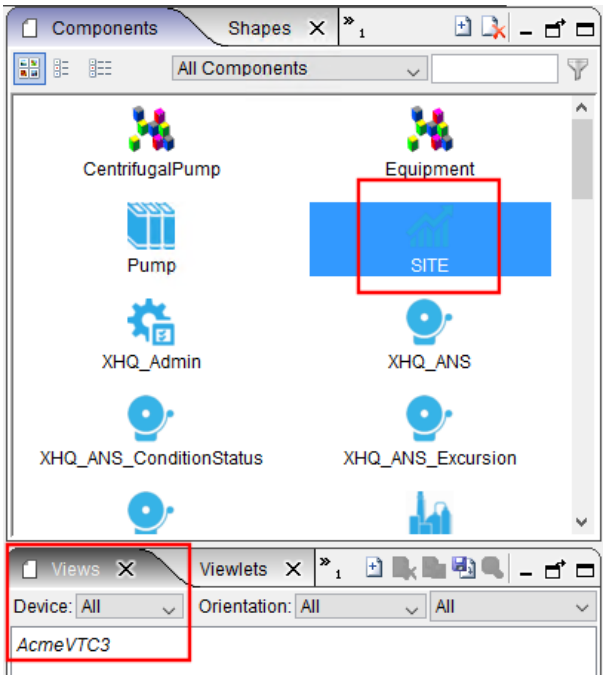
XHQ Right	Description
VTC Basic Access	This right allows the end-user to create new personalized views and share them with other users.
VTC Publishing	This enables the end-user to publish CVs from the VTC to the component.
View Editing	This right grants access to create/edit/import/delete views and edit view properties.

For more information on XHQ security, see the section, *Limiting Rights*, located in the XHQ Administrator's Guide.

About Published Composed Views (CVs)

These are Composed Views (CVs) that have been published from the web-based tool, XHQ Visual Tile Composer. Once a CV is published, it is treated like a typical XHQ view.

In the following example, the published CV, *AcmeVTC3*, appears in the view list of the component, *SITE*.

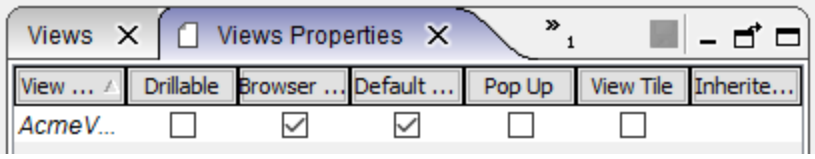


XHQ Workbench: Published Composed Views

You can delete/export/import/search this published CV from the XHQ Workbench.

Once a CV is deleted, it is **unpublished**.

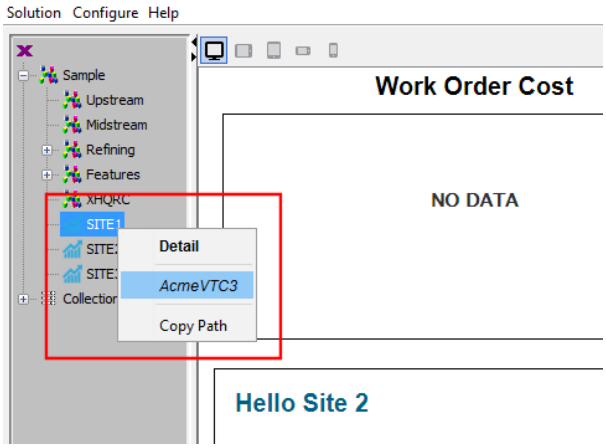
And like typical views, you can set **View Properties** for published CVs.



XHQ Workbench: View Properties for Published CV

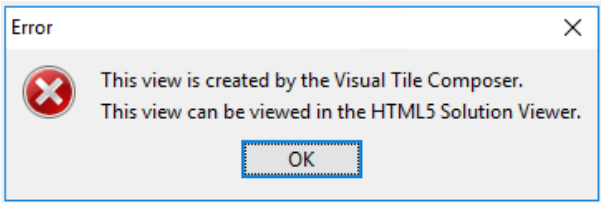
The end-user must be given certain rights to be able to create/edit/delete/publish. See the topic, *About VTC Security*.

From the XHQ Solution Builder, when you right-click on the solution node that includes a published CV, the CV is listed on the shortcut menu.



By default, when you click on the CV from the right-click shortcut menu, the CV loads. In this case, the `Use_Html_View` property in the `clientsettings.property` file is set to TRUE.

If set to FALSE, the following message appears when you try to open the CV:



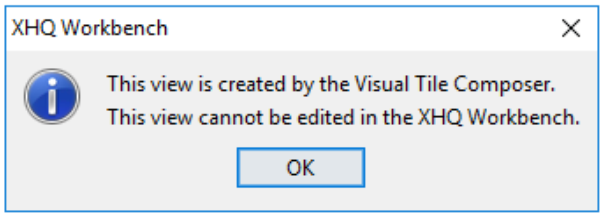
For more information, go to the topic, *About clientsettings.properties*, located in the XHQ Administrator's Guide.

If **View Security** is enabled, the published CV appears in the View Access tab of the XHQ Solution Builder.

For more information, go to the topic, *View Security*, located in the XHQ Administrator's Guide.

How Published CVs are different from typical Views

- Published CV names are *italicized*.
- You cannot edit published CVs from XHQ Workbench. The following message appears:



- You cannot embed a CV in a normal view.
- You cannot link to CVs.

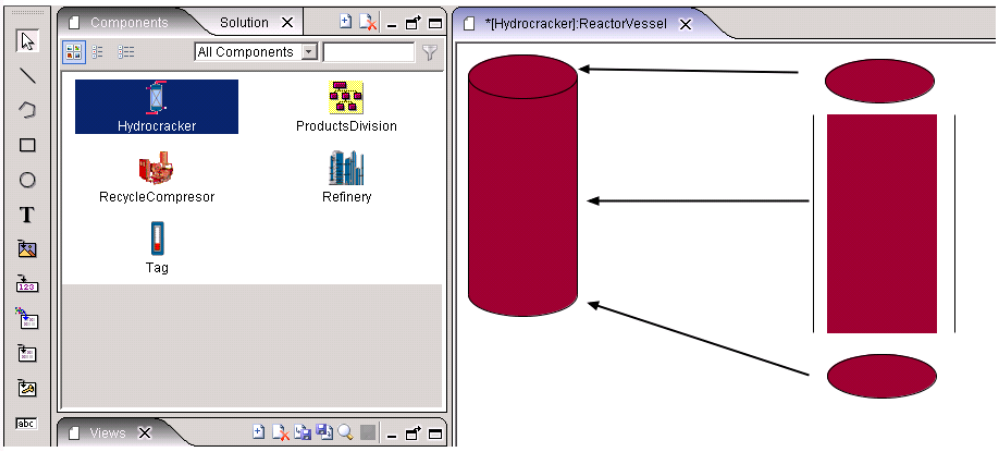
- You cannot add a CV as a template or as a View Instance.

Tutorial: Building Views

The following task is a continuation of the *Building Components* tutorial and refers to components created from that tutorial.

To create a view

1. Open the **XHQ WorkBench**.
2. From the **Components tab**, find the **Hydrocracker** component.
3. **Right-click** on the **Hydrocracker** component, and from the shortcut menu, click **New View.**
   The "New View" dialog box appears.
4. In the **View Name box**, enter `ReactorVessel` and click **OK**.
   The XHQ Workbench tab opens.
5. Use the **Ellipse tool** and the **Rectangle tool** to create two ellipses and a rectangle like those shown in the image below.

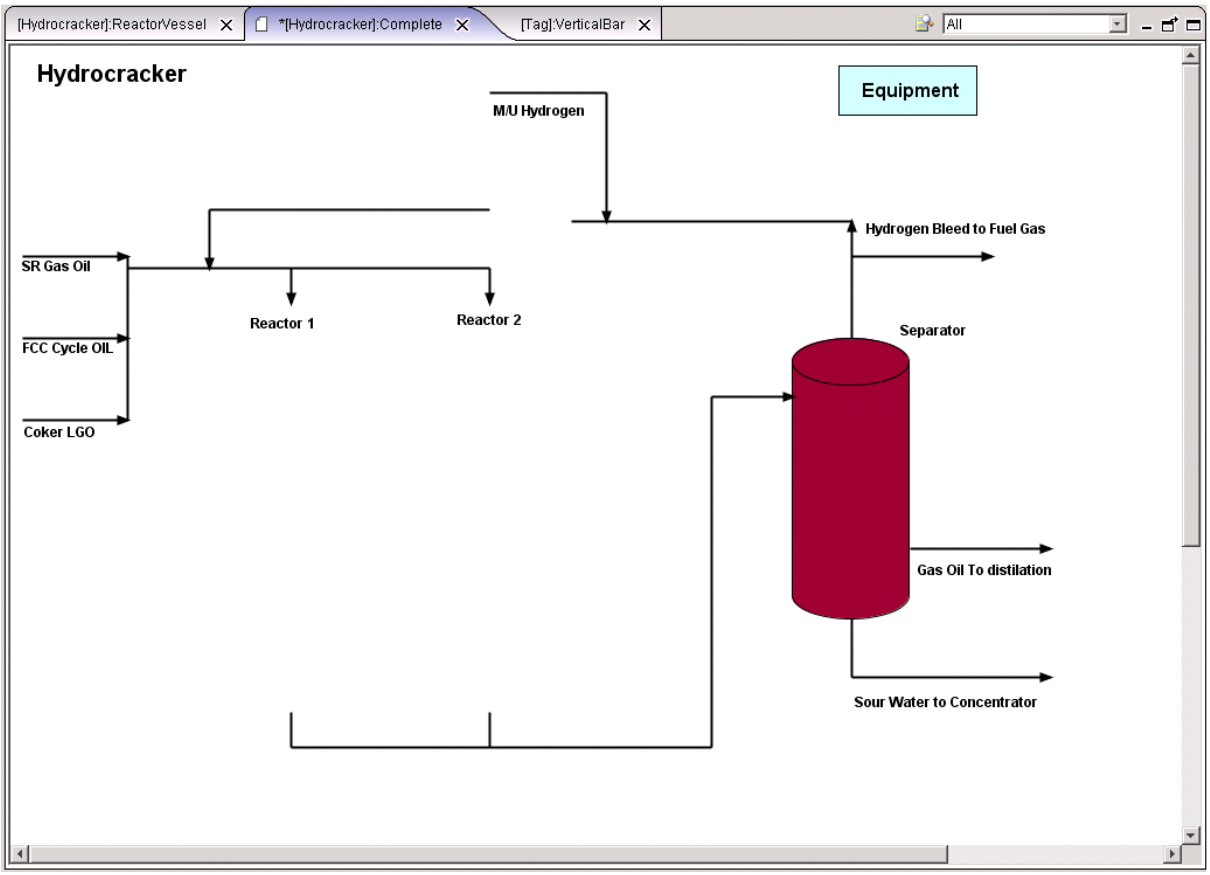


6. Using the **Line Tool** , create two lines for the cylinder edges.
7. **Adjust** the Fill color, the Line color, or the ordering of the view elements, as necessary.
8. **Group** the ellipses, the rectangle, and the lines.
9. Move the group to the upper left-hand corner of the view.
10. **Save** the view using CTRL+S and close the XHQ Workbench.

Next, you will be creating a process overview for the Hydrocracker unit.

To create a process overview for the Hydrocracker unit

1. From the XHQ Workbench, right-click on the **Hydrocracker** component, and from the shortcut menu, click **New View**.
   The "New View" dialog box appears.
2. In the **View Name** box, enter `Complete` and click **OK**.
   The XHQ Workbench tab opens.
3. Using the **Text tool**, add the title, `Hydrocracker`, to the view.
4. Using the **Line tool**, draw the lines with arrow heads as shown in the image below.

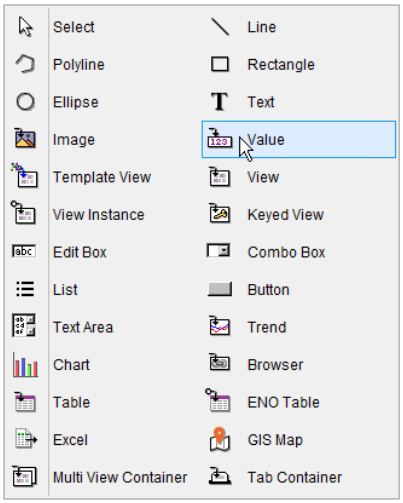


5. Using the **Text tool**, create labels for each location shown.
6. Using the **Rectangle tool**, draw a box in the upper right-hand corner of the view.
7. Set the **Fill color** to light blue.
8. Using the **Text tool**, place a text field in front of the blue box and enter **`Equipment`**.
9. Group the blue box and the text field.
10. Create the separator vessel, using steps 4 to 6.
11. Save the view.

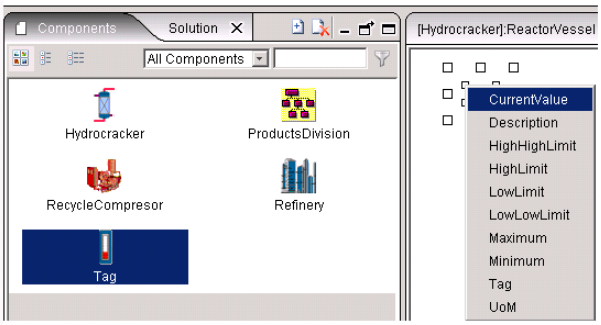
Next, you will be creating a view for the Tag component.

To create a view for the tag component

1. From the XHQ Workbench > Components tab, right-click the **Tag** component, and from the shortcut menu, click **New View.**
   The "New View" dialog box appears.
2. In the **View Name** box, enter **`ValueDisplay`** and click **OK.**
   The XHQ Workbench appears.
3. Right-click anywhere in the empty space and select the **Value** tool, from the shortcut menu.

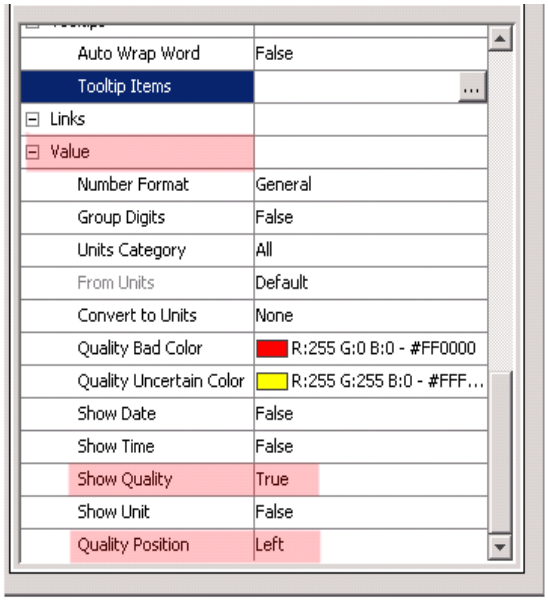


4. Click on the drawing area at the desired location for the value item.
   This adds a value field that references a primitive member of the Tag component.
5. From the list of primitive members, select **CurrentValue.**

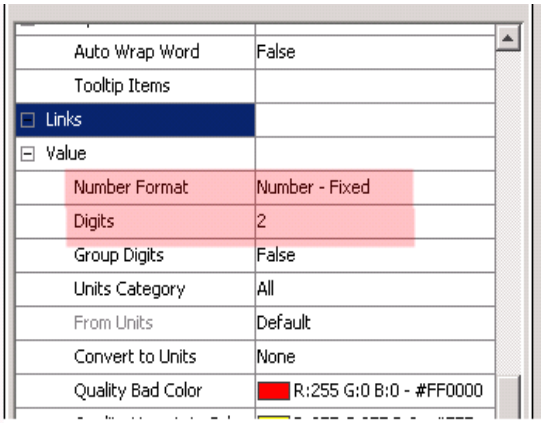


A value item placeholder appears.

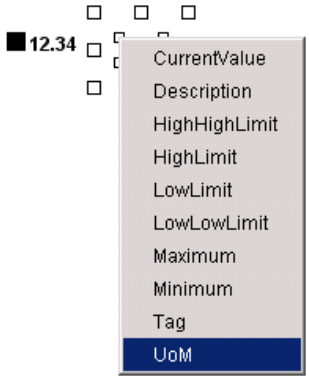
6. In the **Configuration Properties tab**, locate the Value properties, set **Show Quality** to **True** and **Quality Position** to **Left**



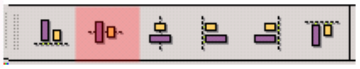
7. Next, set **Number Format** to **Number Fixed** and **Digits** to **2**



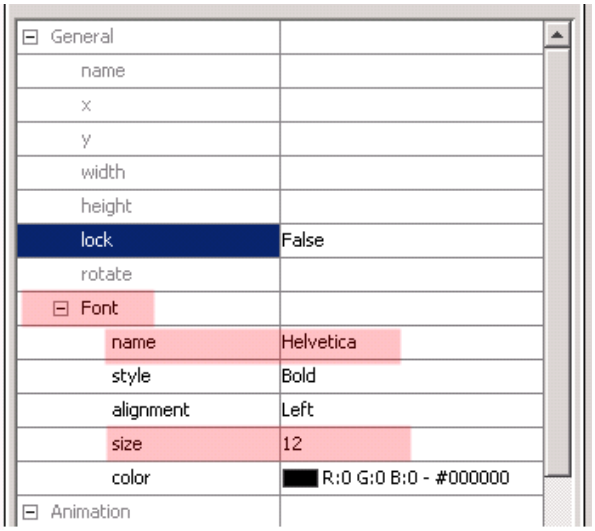
8. Set **Quality Settings** similar to in steps 6 and 7.
9. Repeat steps 3 through 5 to add another **Value** for the **UoM** primitive member and place it to the right of **CurrentValue**



10. Select both elements and, using the **Alignment tools**, align the two value items relative to their middles.



11. With both value items still selected, make sure that the font is **Helvetica** and the font size is 12.



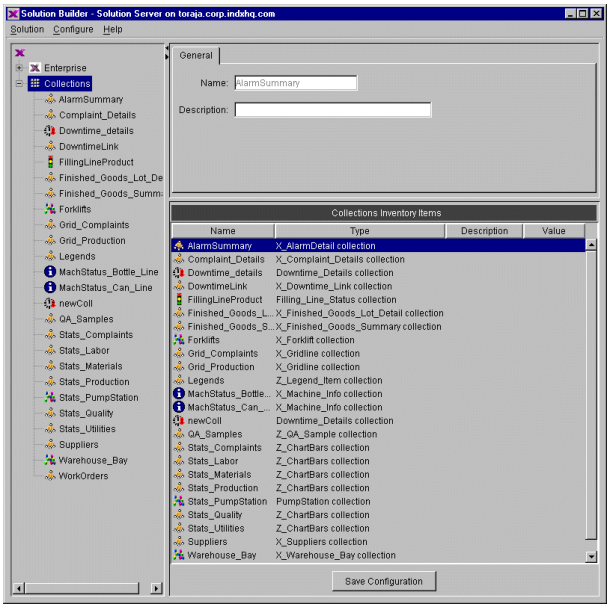
12. **Save** the view and close the XHQ Workbench tab.

8 | Working with Collections and Flow Views

Creating a Collection

Before you can configure a collection, you must have already developed a component with the appropriate member elements. These are normally the column names for a table of records. For example, a component called `FinishedGoods` could have the following members attached: `Name`, `Description`, `Quantity`, `Qty_Reserved`, `ProductID`, and `UnitofMeasure`.
The collection you create, `Goods`, may extract from a relational database a collection or set of records with fields that comprise some member elements in the `Finished Goods` component.

You only see the Collections node in the XHQ Solution Builder. The Collections node is not displayed at browser runtime.



Collections from the XHQ Solution Builder

When creating a new collection, the collection name you use must be **unique** among all the other collections. It must begin with an alpha character and may contain only alphanumeric or underscore characters. Do not use spaces and other special characters, including dashes and periods.

Collection member names are limited to 30 characters or less.

The component you select to associate with the collection must have the members that match the fields in the database that you will extract from. There may be other fields that you won't map in this collection, but which will be mapped from other databases to retrieve calculated values.

To expose collection contents, a collection component needs to have a primitive member configured as a

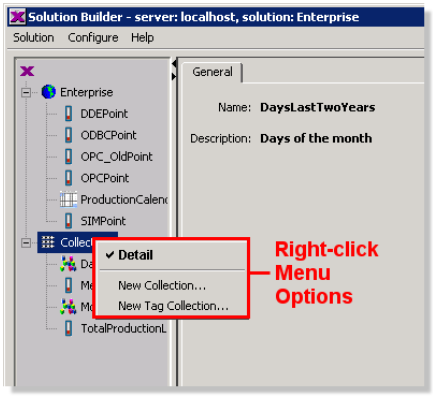
Key. The tag component to be used for the keyed view should be the one used in the tag collection. If the tag collection is NOT defined in the solution, the XHQ Workbench displays an error message when the keyed view is created. If the tag collection changes after the first keyed view is created in the XHQ Workbench, then you must restart the XHQ Workbench to reflect the new tag component.

Once a collection is created, it appears in the navigation tree of the XHQ Solution Builder. XHQ pulls the icon from the component and associates it with the collection. Collection members are displayed hierarchically in alphabetic order.

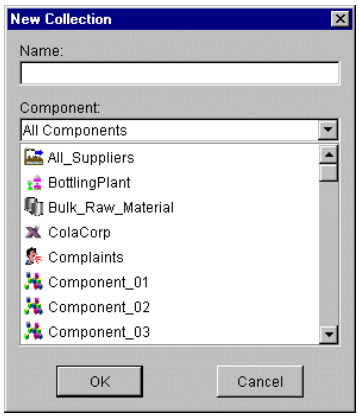
For more information about tag collections, go to the section, *Working with Tags*.

To create a collection

1. From the **XHQ Solution Builder**, in the navigation tree, right-click the **Collections** node.
   Three options are displayed on the shortcut menu: Detail, New Collection, and New Tag Collection.



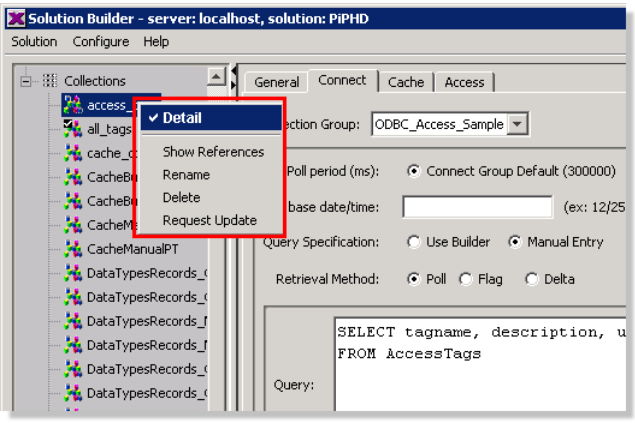
2. Click **New Collection**.
   The "New Collection Item" dialog box appears.



3. **Enter the name** of the new collection you are configuring in the **Name** box.
4. In the **Component** list, select the component you want to associate with the new collection.
   The component identifies the member element structure used to define a collection.
5. Click **OK**.

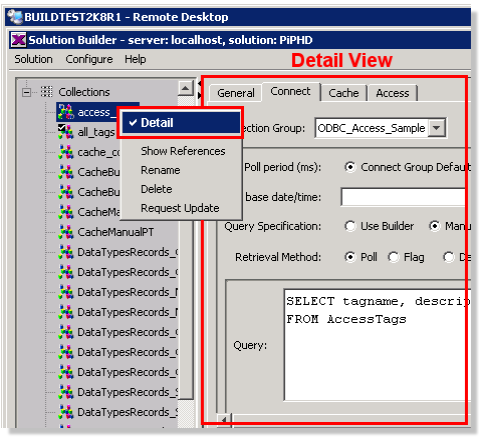
Working with Collections

The following options are available from the right-click shortcut menu.



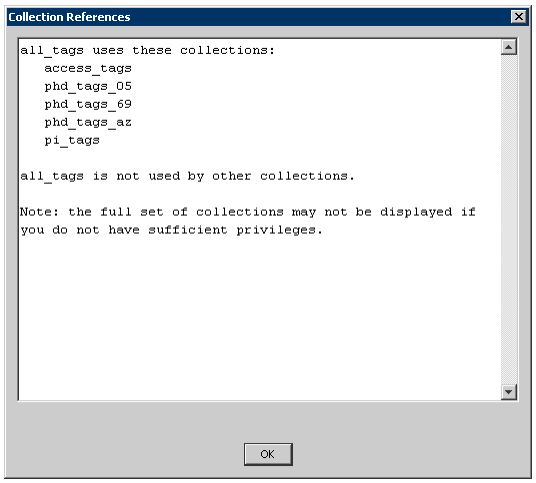
Collection Right-Click Menu

- **Detail**



This displays the Detail view, which consists of the Detail Tabs (General, Connect, and Cache) and the Edit panel. Use this option to configure the collection.

- **Show References**



This displays which collections reference, or is referenced by, the given collection.

- **Rename**
  This enables you to rename the collection.

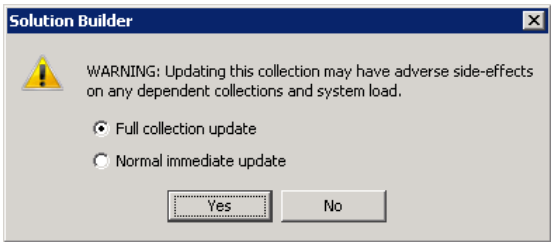
  Renaming the collection member invalidates collections that reference the current collection name.

- **Delete**
  This enables you to delete the collection.

  WARNING
  Deleting the collection in turn deletes all links to it from other objects in the solution.
  For example, deleting a collection that is configured as linked in another collection breaks the configuration for that collection. You must then manually reconfigure the collection to link it to the other collection in the Solution Builder, or re-import the master collection. The recommended way to do this is to export the configuration for both collections and then re-import both collections so that the re-link happens when the master collection is imported.

  See the topic, *XML to Preserve Collection Links*, for details on how to preserve these collection links.

- **Request Update**



  - **Full collection update**
    This returns all records and triggers a full update of the collection, which typically occurs on a main poll cycle.
  - **Normal immediate update**
    If the collection is a Delta collection, this performs an delta update. If the collection is a flag collection, then it performs a flag update. Updates are performed without changing the actual update schedule.

  Use the **Full collection update** if you are experiencing problems with missing updates (due to problems with a flag or delta updates), or if a significant change happens in the backend and the next scheduled poll cycle does not happen for some time.

  This operation is asynchronous. Although the client only sends the request, the actual time of the update is not guaranteed to be immediate. It depends on the current state of the connector. For example, if the connector is currently doing an update, then this request does not have an effect. The connector and the solution server logs provide the same messages at the specified log level as during the scheduled updates. The only other external sign of the complete update is the change of the collection time stamp in the XHQ Workbench or in the XHQ Solution Builder.

About Flag Collections

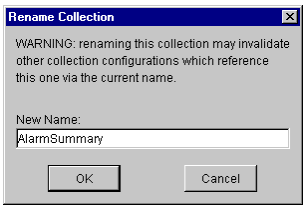
A flag collection retains the last flag value after an XHQ solution or a connector restart. So after a restart, flag queries do not automatically perform full updates through the base query. A full update is triggered only if there is a change to the flag value (that is, when the result of the "refresh query" differs from the last stored value).

If a full update is not required, the connector recalculates how many more additional flag polls are needed before submitting a "Refresh" request.

Like poll collections, activation during start-up is delayed for flag collections.

To rename a collection

1. From the **XHQ Solution Builder**, in the **Navigation tree** under **Collections**, right-click a collection.
   The collection shortcut menu appears.
2. Click **Rename**.
   The "Rename Collection" dialog box appears.



3. Enter the new name for the collection in the **New Name** field.

   WARNING
   As a result of changing the name, you may invalidate other collection configurations that access this collection name. Collection names are limited to 30 characters or less.
4. Click **OK**.

To delete a collection

If you chose to delete the collection, you will lose the collection and all links to it from other objects will be lost.

See the topic, *XML to Preserve Collection Links*, for details on how to preserve collection links.

1. From the **XHQ Solution Builder**, in the **Navigation tree** under **Collections**, right-click a collection.
   The collection shortcut menu appears.
2. Click **Delete**.
   A warning appears.

3. Do one of the following:
   - Click **Yes** to delete the collection.

   else,

   - Click **No**.

Ordering of Collections

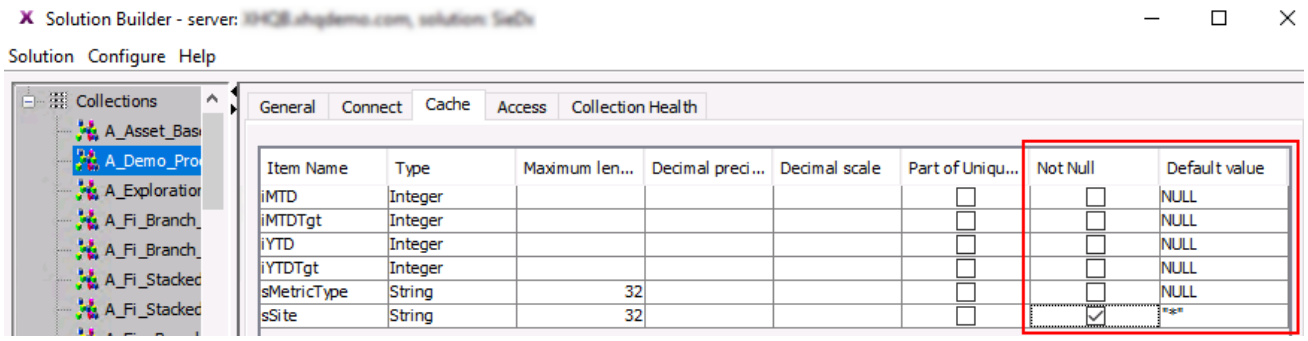
If you do not specify a sort order by using the Sort entry in the collection view parameters (through the View tab of the XHQ Workbench), then the results will be sorted according to any ORDER BY clause in the Link tab (of the XHQ Solution Builder) of the member collection. If there is no ORDER BY clause in the Link tab, then the order of the result rows is not defined and may change from request to request. Generally, you must specify a sort order in either the view parameters or the member collection Link tab to ensure reasonable results. If a sort order is specified in both the view parameters and the Link tab, the view parameter order takes precedence.

Using NOT-NULL to Improve Performance

To improve the performance of queries that use ORDER BY, you can specify collection members (columns) as NOT NULL. Previously, using ORDER BY in queries, even when properly indexed, caused a full table scan and sort when the index referenced nullable columns.

To improve performance using NOT-NULL

1. From the XHQ Solution Builder, select a **global collection**.
2. Click the **Cache tab**.
3. Enable NOT NULL for the given collection member by **checking the box**.



4. Enter a **Default value**.

About the Default Value

To specify a collection member as NOT NULL, you must assign a **Default value**. The Default value may be a valid XHQ expression (with the exception of expressions with references). And the data type of the Default value expression must match the data type of the collection member for which it is defined.

For members of the **String** or **Text** data type, checking the Not Null box automatically displays an asterisk enclosed in double quotes ("*") for the Default value. For members of the **DateTime** data type, checking the box automatically displays the Default value, Now(). All other data types display the Default value of 0 (zero).

As previously mentioned, you can change the Default value to a valid XHQ expression with matched data type.

Valid Data value expression examples:

For a Boolean type member, datePart("minute")>15

For a DateType type member, dateStart("week",Now())

Invalid Data value expression examples:

Any expression using a reference like rRef("tagName")

And direct references like ::Enterprise.Acme.vReal

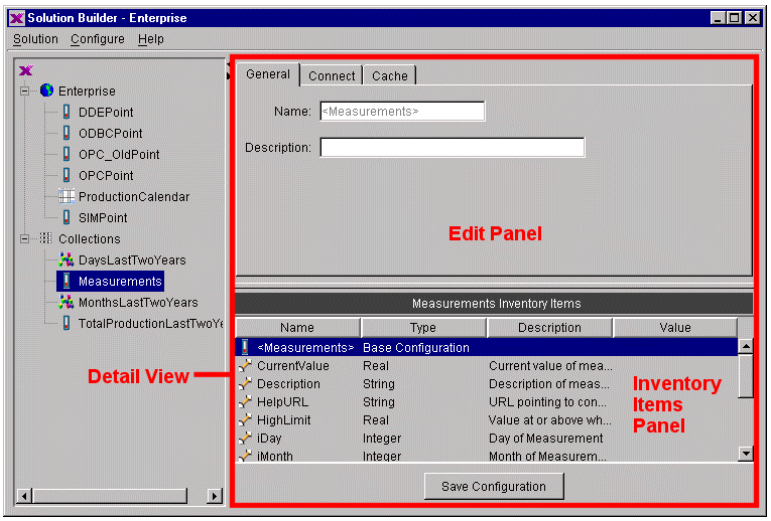
Configuring a Global Collection

Configuring a collection is a two-step process. First, you configure and map the global collection. For example, you might create an `Orders` global collection that consists of all the orders you might want to view from different components in the navigation tree.

Second, from any component in the tree from which you might want to look at a subset of orders (for example, the orders for the Southeast region), you point, or link, to the global collection and pull orders that make sense from the perspective of that component. Take, for example, two components, `Short-Range Planning` and `Long-Range Planning`. Each of those is linked into the `Orders` global collection to extract orders for different time periods.

Working in Detail View

The Detail View displays the Edit panel and the Inventory Items panel.



Detail View

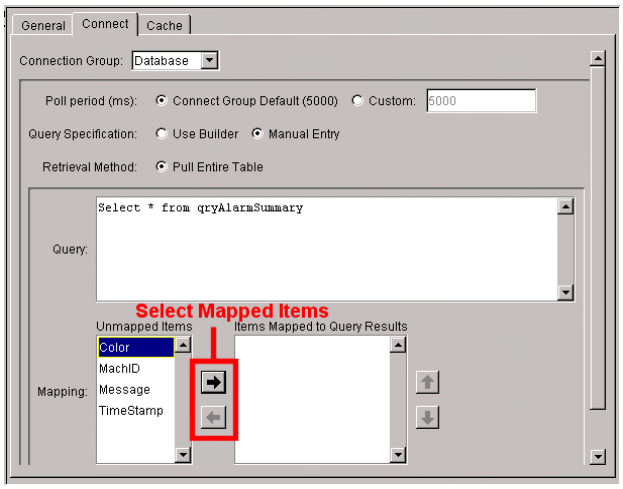
The name of the collection you are configuring shows at the top of the list of items in the Inventory Items panel and appears in brackets **< >**. The item is of type **Base Configuration**. This means you can select it in the Inventory Items panel and then use the Edit panel to configure the way it connects to the backend system and the way the cache is managed.

For more information on the Detail View tabs, refer to the topic, *Building and Configuring Solutions*.

Underneath the collection name is a list of members that mirrors exactly the items of the component associated with the collection. If you look at the component in the Component Details tab, you would see an identical listing of members.

To configure a collection

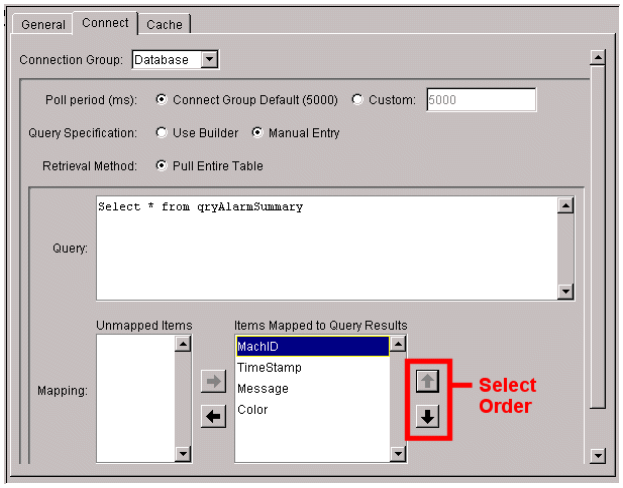
1. From the navigation tree, **click the collection** you are configuring.
   The "Detail View" tabs appear.
2. Click the **Connect** tab.
3. From the **Connection Group** list, select a connection group.
4. Under **Query Specification**, click **Manual Entry**.
   XHQ provides Query and Mapping areas that allow you to enter the SQL query statement that queries the backend database using the connection group and maps that data to the appropriate members. Component members appear in the Unmapped Items column.
5. In the **Query** box, **enter your SQL** statement query to extract data from the database.
6. In the **Unmapped Items** list box, select each member you included in your SQL query and **click on the right-pointing arrow**.
   This places the member in the Items Mapped to Query Results column.



7. If you inadvertently add a member you do not need, remove the member from the **Items Mapped to Query Results** box by selecting it and clicking the left-pointing arrow.

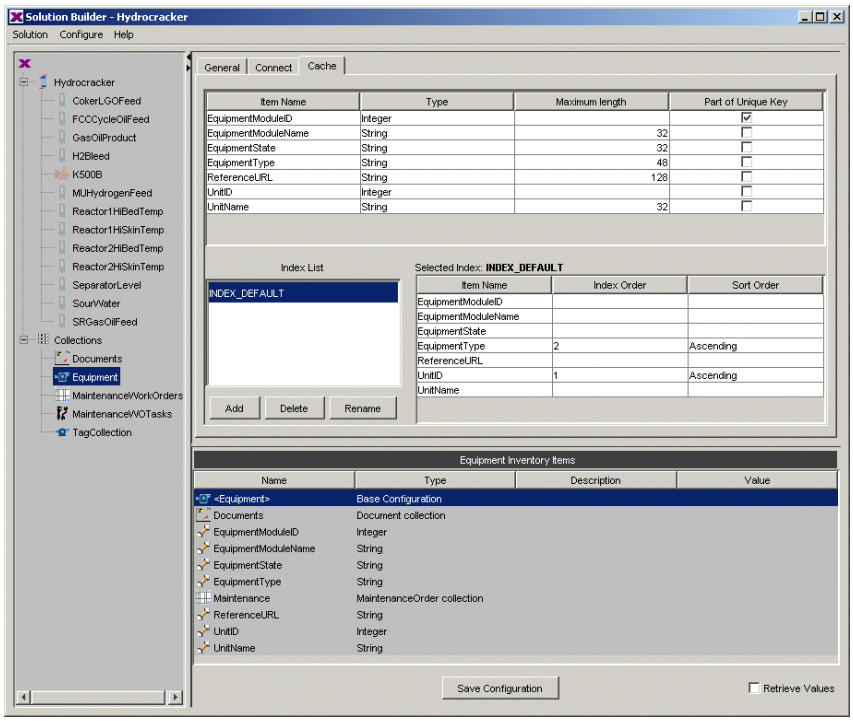
   You must order members in the Items Mapped to Query Results field box in precisely the same extraction order you entered in the Query field.

8. You can change the order of the items in the Items Mapped to Query Results field by selecting the member and clicking the upward- or downward-pointing arrow to move it up or down in the order.



   You do not have to map all available members. Some may have been built as "placeholders" for other types of queries not relating to this collection.

9. Click the **Cache** tab.
   The Cache tab allows you to configure index and unique key information so that the XHQ cache optimizes the data queries in real time. The Cache window displays a table of all the members assigned to the collection you are configuring. For each member, it allows you to change a string's maximum length and index and unique key information.



10. For String members, click in the **Maximum length** field next to the member you want to configure. This enables the field for manual entry.
11. Enter a value for the maximum length you want to save in the cache.
12. Click the **Index** check box next to the member you want to specify as an Index field. Designating an item as an Index means that item may be used to sort records.
13. Click the **Part of Unique Key** check box next to the member you want to specify as a *unique key*. A unique key is a unique field that identifies a record in a database. XHQ must know the Unique Key to query the database; therefore, you must designate an item as a Unique Key.

    Defining a Unique key does not mean that the Index key is defined. Both the Unique key and the Index key need to be defined in order to optimize the query.

14. Set the **Index**. Designating an item as an Index means that item may be used to sort records.

    For more information on how to set the Index key, go to the topic, *About the Index*.

15. Click **Save Configuration**.

Linking a Component to a Global Collection

After you have built a global collection and configured cache information, you then have to distribute the collection to all the components that need to look at the collection.

To link a component to a global collection

1. From the **XHQ Workbench**, **double-click the component** that you want to configure to "look at" a particular collection.
   The Component Details tab appears.
2. On the **Add** menu, click **Existing Component**.
   The "Add Existing Component Item" dialog box appears.
3. **Select the component** that was the basis for the global collection you built.
   This is a *placeholder component*, in that it identifies the structure of a collection (it is one of a type that will get many records of that type).
4. Under Dimension, click **Collection** to identify that there will be many of this type.
5. Click **OK**.

   You have now correctly configured a component in the XHQ Workbench with the appropriate structure to map the subset from the global collection at run-time. Next, you need to create an *instance* of that component in the XHQ Solution Builder to map the data at run-time. Typically, that instance is already present; you need to update to the latest components to reflect the modifications you have just performed for the collection subset.
6. In the navigation tree of the **XHQ Solution Builder**, select the instance of the component you just modified.
7. Select the collection placeholder component you want to link to a global collection.
8. Click the **Link** tab.
9. From the **Collection** drop-down list, select the global collection.
10. OPTIONAL
    Enter the SQL parameters that constrain the global collection in the `SELECT * FROM` text box.
11. Click **Save Configuration**.

You now have a correctly mapped instance of a component that knows how and where to extract its data when a view is built against the collection and called at run time.

Exposing Collection Records in the XHQ Solution Tree

The following configuration steps are required to expose the collection rows in the XHQ Model (Solution Tree).

Step 1	**Component configuration** Specify Key and Description columns.
Step 2	**Collection member configuration** Enable the "Expose Contents" option.
Step 3	**Solution configuration** Link collection members.

Step 1 - Component Configuration

The collection base component must have a String member that is associated with the **Key attribute**. The Key attribute is used to display and identify each record in the collection. If the Key value is not unique, then duplicate records are displayed.

In addition to the Key attribute, the collection base component must also have a String member that is associated with the **Description attribute**. The Description attribute is used to get the description for each record in the collection. At runtime, when you hover over a collection row item node in the navigation tree, the description appears in the Tooltip.

Consider the following example.

1. From the **XHQ Workbench**, create the component `EqItem`.
2. From the Component Details tab, on the **Add menu**, select **String**.
3. Under the **General tab**, in the **Name** box, enter the unique member name `Descr`.
4. For **Tag**, select **Description**.

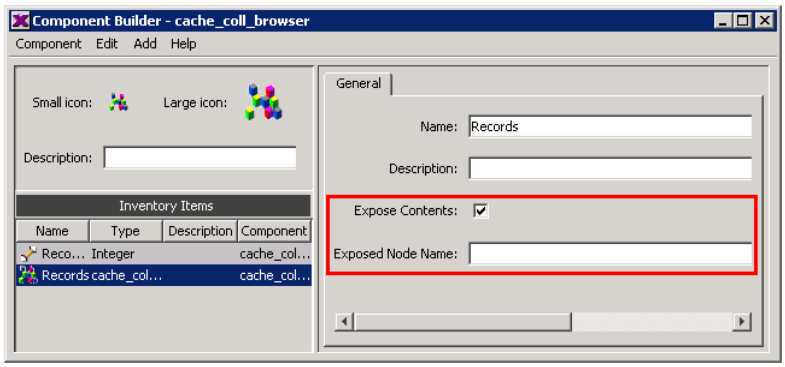
   Next, add a String member that is associated with the Key attribute.

5. On the **Add menu**, select **String**.
6. Under the **General tab**, in the **Name** box, enter the unique member name `EquipName`.
7. For **Tag**, select **Key**.
8. **Save** your changes.

Step 2 - Collection Member Configuration

To expose XHQ collection members, which by default are not displayed in the XHQ navigation model, two configurable properties have been added to the General tab of the Component Details tab:

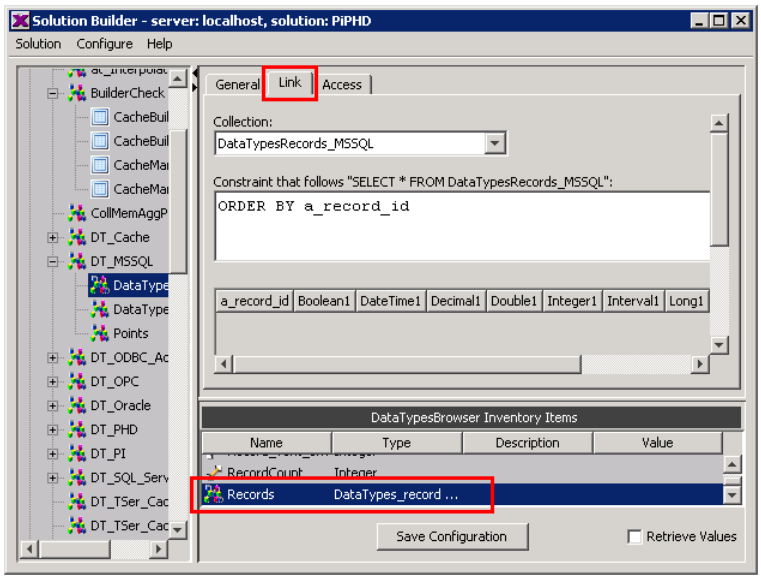
- **Expose Contents**
  When checked, the collection member is displayed in the model, with the collection row items as its children.
- **Exposed Node Name**
  Enter the collection member name that appears in the model. If left blank, then the collection member itself is not shown in the model. All its children, however, show directly under the collection member's parent node in the model.



Component Details tab, General Tab

Step 3 - Solution Configuration

The collection member must be linked in the solution. From the XHQ Solution Builder, locate the collection member and configure the Link tab.



XHQ Solution Builder, Link Tab

The link collection constraint syntax allows full expression support. There are **two** forms of syntax that allow a link constraint to obtain data from XHQ members to make the constraint dynamic:

```
%membername% and {xhq-expr expression-to-evaluate}
```

The first form obtains the value of the member and substitutes it in the constraint before the query is issued. The second form allows a full XHQ expression.

Here is an example using both forms:

```
WHERE equipment = %sEqName% AND workorder LIKE {xhq-expr "WO#" + StringValue(iWONumber) +
"%"}
```

The first form is the original simple form. If it has BAD quality data, it treats the comparison (no matter what the operator is) as IS NULL. The second form, however, instead causes the query to be invalid if BAD data is encountered with any expression that is more complex than a single member.

The "BAD, Not Initialized" quality is treated as a special condition and causes the query to be delayed until data is received.

So, if you are using the full expression form and it is desired to handle BAD data, use the XHQ expression features to assist you. The following example illustrates the difference. In this example, the member `sVal1` is a String member with an expression "X" and `sVal2` is an unconfigured String member.

```
WHERE col = %sVal1% OR col = %sVal2%
```

Would be translated into SQL as "`WHERE col = 'X' OR col IS NULL`". Using the full expression syntax, the result would be the same for the simple case.

```
WHERE col = {xhq-expr sVal1} OR {xhq-expr sVal2}
```

However, in the earlier example, if the `iWONumber` is not configured, it results in bad quality data and an invalid constraint. Note the following solution server log example:

```
* * * * INVALID CONSTRAINT * * * * block [LIKE EXPRESSION: "WO#"  + StringValue
(iWONumber) + "%"] has input of unacceptable quality: null]
```

In order to have a **valid** constraint with unconfigured or BAD input, you must use the **`iff`** and **`quality()`** methods:

```
WHERE equipment = %sEqName% AND workorder LIKE {xhq-expr iff(quality(iWONumber)
==BAD,"","WO#"  + StringValue(iWONumber) + "%")}
```

This will have a similar effect and produce the following when iWONumber is BAD:

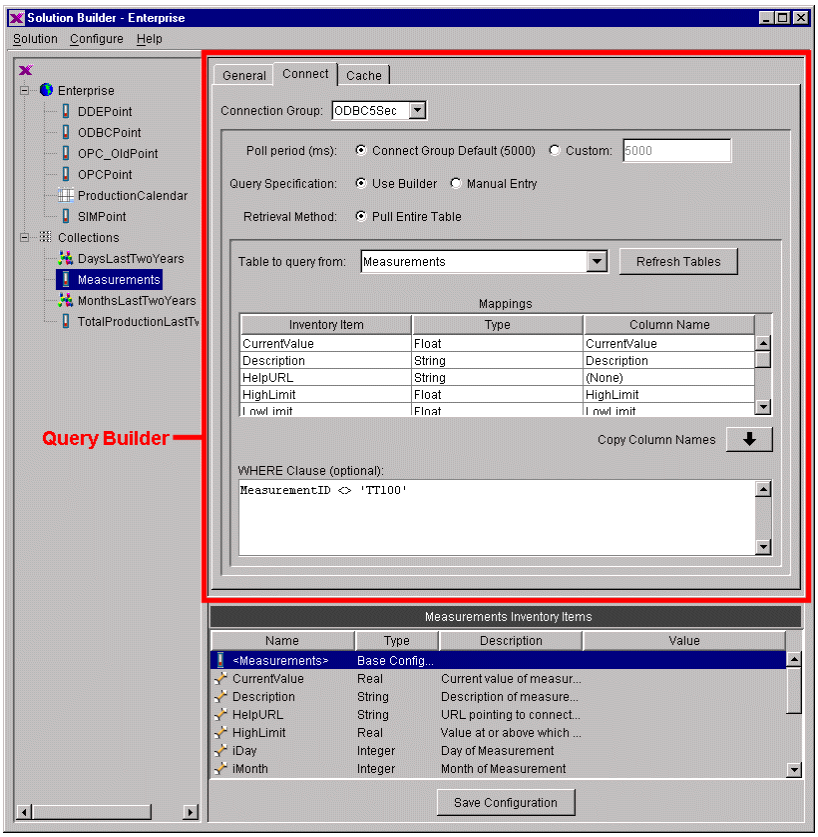
```
WHERE equipment = 'MILL101' AND workorder LIKE ''
```

Things to Note about the Runtime Behavior

- In the NavBar, collection records appear under the collection member node.
- Multiple levels of nested collections are supported.
- Runtime behaviors for general model nodes apply to collection member nodes and collection row item nodes.
- To change the maximum collection row items to display in the navigation model, set the `MaxCollectionItemsInModel` global setting property. The default value is 50.

Using the Query Builder

The query builder provides a simple interface for configuring **simple** queries against ODBC or Oracle sources. It supports any query that directly uses the columns of a single table or view. You access the query builder by selecting a collection inventory item and clicking on the Connect tab.



Query Builder Interface

The Query Builder works on Oracle 8.0 and above. For queries against Oracle 7.3.4, select **Manual Entry** to manually enter complete SQL statements.

To use the query builder

1. From the **XHQ Solution Builder**, select the collection.
2. Click the **Connect** tab.
3. From the **Connection Group** list, select an ODBC or an Oracle connection group.
   A list of Inventory Items appear in Mappings.
4. For **Poll Period**, do one of the following:
   - Click **Connect Group Default** to use the poll period set from the Connection Group dialog box. The default value appears in parenthesis.

   or

   - Click **Custom** to set a poll period (in milliseconds) that is different from the connection group default.

5. Under **Query Specification**, do one of the following:
   - For users of Oracle 8.0 and above, you can either **Use Builder** or select **Manual Entry** for advanced queries.

   or

   - For users of Oracle 7.3.4, select **Manual Entry** to manually enter your SQL statements.
6. In **Table to query from**, select the name of the table.
   This allows you to map columns from the database to members in the collection.
7. Click **Refresh Tables**.
   This allows you to receive recently updated data.
8. Under **Column Name**, click inside the field.
   A drop-down list appears, displaying the table's column names.
9. From the drop-down list, **select a column name** to associate with the inventory item.
10. Repeat as needed.
11. OPTIONAL
    Click **Copy Column Names** to automatically transfer the column names you identified onto the WHERE Clause text box.
12. Click **Save Configuration**.

About Flow Views

A *flow view* is a view of a set of collection members – members are the explicit values owned by a particular component. For a collection the number of members is determined by a query at runtime. For a single member of a collection, you can specify how it should be rendered in a view. For example you can show the individual fields from each record in a row of a database table as a rich display containing text, animated images, and so forth. When creating a flow view, you specify how you want to arrange the multiple views of all the collection members when they are retrieved. For the example, the composite view of all the individual views of each row in the table is a flow view.

Flow views are only visible from the View panel of a collection view. You cannot see them in the navigation tree.

See the topic, *Additional Global Settings*, located in the XHQ Administrator's Guide, for details on the `MaxCollectionCells` property (which limits the number of cells in a flow view).

Building Views Against Collections

In XHQ, a collection gathers data from a backend source, usually a database or collection of objects, and returns that information into a *global collection*. At run-time, an instance of a component in the tree links to extract a subset of the record sets from a global collection. For example, you could build a global collection containing orders for the entire United States. Then, in the instance of a **`State`** component, such as **`Arizona`**, you would configure it to link to the United States global collection and select the pertinent orders for Arizona.

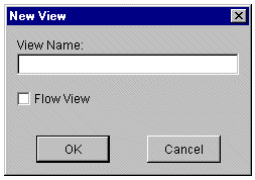
Suppose you wanted to create a view of all orders generated by the Arizona region in an enterprise. You would follow the following three steps to create the collection view.

Step 1	Build a singleton view against a primitive component. For example, `One_SalesOrder`.
Step 2	Build a flow view against the primitive component. For example, `Many_SalesOrders`.
Step 3	Build a view in the target component, and in it, insert the primitive's flow view. For example, `Regional_View` of the `State` component contains the `Many_SalesOrders` flow view.

By building a view and flow view at the primitive component level, you can reuse the primitive flow view of **`SalesOrder`** in other components other than the instances of **`State`** components.

EXAMPLE: BUILDING A VIEW AGAINST A COLLECTION

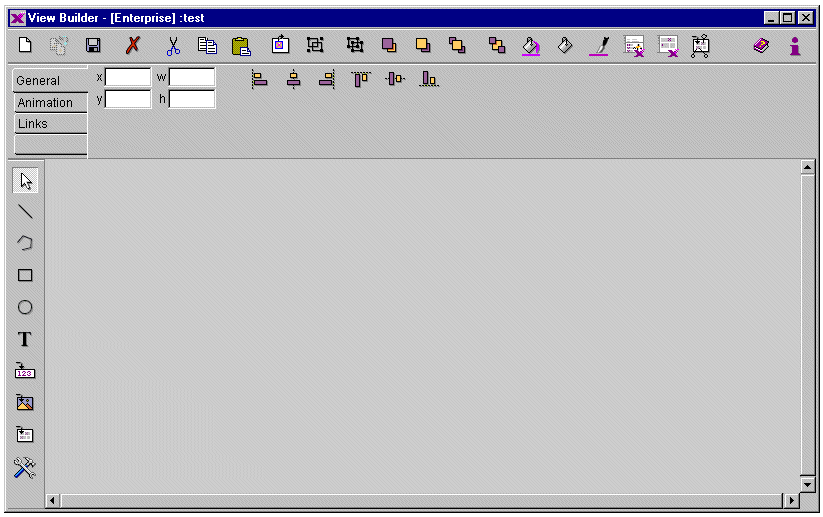
1. From the **XHQ Workbench**, locate the primitive component that defines the member structure of the global collection.
   In this case, **`SalesOrder`**.
2. Right-click the primitive component and from the shortcut menu, click **New View**.
   The "New View" dialog box appears.



3. Enter the **view name**.
   Because this view is a single set of collection values (and consequently not a flow view), use a descriptive name for the new view like **`One_SalesOrder`**.

   Do not select the Flow View selection box.

4. Click **OK**.
   The XHQ Workbench appears.



5. Click the **Value** tool.
   The cursor changes to the insertion cursor.
6. **Click on the drawing area** at the desired location for the value item.
   The component's primitive members are listed in a shortcut menu at the insertion point.
7. From the shortcut menu, select the primitive member whose value you want to appear on the screen.
   A value item placeholder appears.
8. Click on the **Value** tab and configure the value.
9. Click the **Animation** tab and define the **Set**, **Using**, and **Evaluating** values for each of the available collection elements.
   All available collection value names are displayed in the evaluating drop-down list.
10. After you have finished defining the view for a single set of collection values for the primitive component, click the **Save to repository** button.

You now have a view of what one order looks like. Next, you will build a flow view of the collection **`Many_SalesOrders`** against the primitive component **`SalesOrder`**.

> Any view elements you create in the collection set view will repeat in the full collection view you create against the set view.

11. Right-click the primitive component (**`SalesOrder`**), and from the shortcut menu, click **New View**.
    The "New View" dialog box appears.
12. Enter the flow view name in the **View Name** box.
    In this case, using a descriptive name for the new view like **`Many_SalesOrders`**.
13. Click **Flow View**, and then click **OK**.
    The XHQ Workbench appears.
14. Define its visual appearance (for example, background color) using the XHQ Workbench commands.
15. In the **Edit Panel**, click the **Flow View** tab.
16. From the **View** drop-down list, select a single view.
    In this case, **`One_SalesOrder`**.
17. **Enter the number of rows and columns** you want the collection view to contain in the Rows and Columns fields.
    The default rows and columns amount is 1. To change the number, highlight the old value, type a new one, and then press **Enter**.

18. Click **Use Previous/Next Buttons** if, when viewing the collection, you want to browse through sets of collection members.
    If you do not click this box, the collection view will display only one set of collection members.
19. Click **Save to repository**.

You have now created a view of what many orders looks like against the primitive component.

20. In the **XHQ Workbench**, select the parent component where the collection flow view will be seen.
    In this case, **`State`**.
21. Right-click and from the shortcut menu, select **New View**.
    The "New View" dialog box appears.
22. Enter the new view name in the **View Name** box.
    In this case, using a descriptive name for the new view like **`Regional_View`**.
23. Leave the Flow View checkbox **clear** and click **OK**.
    The XHQ Workbench appears.
24. Click the **Insert View** icon and position the point where you want to insert the flow view.
25. In the **Edit panel**, click the **View** tab.
26. From the **Component** drop-down list, select **`SalesOrder`**.
27. From the **View** drop-down list, select **`Many_SalesOrders`**.

    The Rows and Columns selection boxes are disabled because you can only insert one flow view.

29. Define its visual appearance (for example, background color) using the XHQ Workbench commands
30. Click **Save to repository**.

By following this example in the **`Regional_View`** of the **`State`** component, you can also insert a flow of **`One_SalesOrder`** view from the **`SalesOrder`** component. However, you have less reusability, since only **`State`** components would then have the **`One_SalesOrder`** flow view. For example, using this technique you would have to repeat steps to present a **`One_SalesOrder`** flow in a new **`Western_States`** component view.

View-Based Collection Filtering

This feature allows you to configure collection filters for view-in-views and place controls on the view that can filter results. With these controls, you can also initiate a collection query. These controls enable your end-user to change the values associated with the filters at runtime.

Configuring view-based collection filtering involves the following basic steps.

Step 1	**Determine Collection Filter Criteria** What is your filtering criteria? What options do you want to give the end-user? How do you want to initiate a collection query?
Step 2	**Create and Configure the Control** Controls allow the end-user to select, or possibly enter, values that effect the collection filter behavior. Through the XHQ Workbench, you create the controls that either define the filter or start a query.
Step 3	**Configure the Collection Filter** A collection filter is made up of one or more *criterion*. In this step, you will learn how to add, modify, and delete collection filter criteria.
Step 4	**Execute the Collection Query** The collection query is executed during runtime in the XHQ Solution Builder or the XHQ Solution Viewer.

For more information on controls, refer to the topic, *Adding Controls*.

Determining Your Collection Filter Criteria

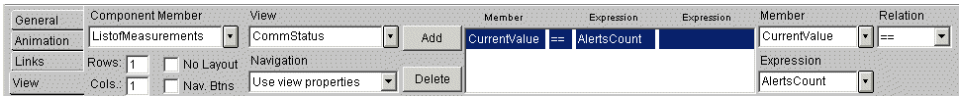
Prior to creating your collection filter, you must define the purpose for the filter.

Consider the following questions.

- What data do you want to filter?
- How the filter will be used?
- How do you want your end-user to use the filter?
- How will the collection query be initiated?
- What will the query interface look like? What controls do you want to use?

Configuring the Collection Filter

After creating your controls, the next step is to configure the collection filter. This is done by defining the filter *criteria* through the **View tab** of the Edit panel in the XHQ Workbench.

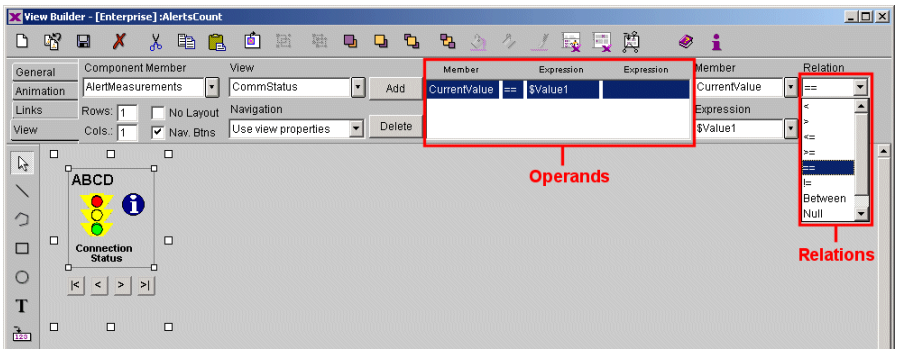


XHQ Workbench Tab Example

A collection filter is made up of one or more criterion. An empty criterion is added to the filter and its various elements are modified afterwards. Therefore, you can:

- Add criterion to a collection filter.
  You can add one or more.
- Modify the criterion once it is added to the collection filter.
  This is done by specifying and modifying either an operand or a relation.
- Delete a criterion from a collection filter.

Criteria are composed of up to four elements: one to three *operands* and a *relation*. The operands are entered as expressions and the relation is chosen from a given list of supported relations.



View Tab – Operands and Relations

About Operands

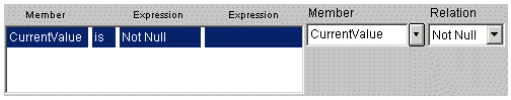
Operands provide the values that are used to compare against the values in the collection for the purpose of filtering or ordering the collection records. Operands are defined using expressions that may contain data references that must be within the context of the component containing the collection member (not the collection itself) or variable references that are defined on the view itself.

About Relations

Criteria relations define the logical relationship between operands in the criteria structure.

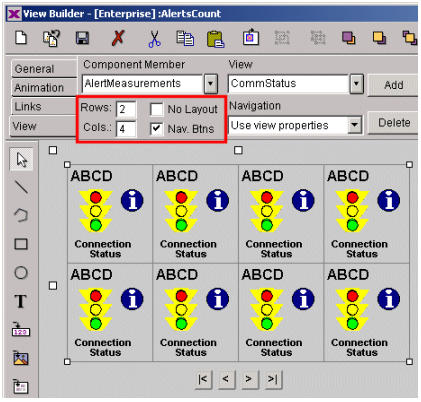
Operand Relationships

Symbol	Meaning	Description
>	greater than	Placed between two operands to determine if the first value is greater than the second value.
<	less than	Placed between two operands to determine if the first value is less than the second value.
>=	greater than or equal to	Placed between two operands to determine if the first value is greater than or equal to the second value.
<=	less than or equal to	Placed between two operands to determine if the first value is less than or equal to the second value.

Symbol	Meaning	Description
==	equal to	Placed between two operands to determine if the first value is equal to the second value.
!=	not equal to	Placed between two operands to determine if the first value is not equal to the second value.
Between	between	Placed between two operands to determine if the value is between the two values. This is inclusive. For example, "3 between 10" is seen as "3≤x≤10" and includes both "3" and "10".
Null	is null	Sets Member operand as Null.
Not Null	is not null	Sets Member operand as Not Null. 

Collection View Layout

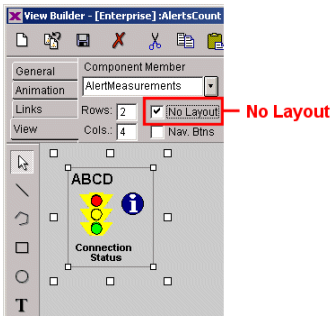
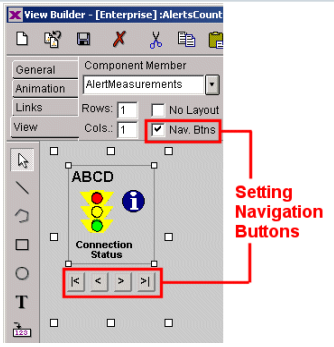
You can set the number of rows and columns as well as insert navigation buttons to the collection view.



View Tab – Collection View Placement Controls

Collection View Placement

Control Name	Description
Rows	Enter the number of rows.
Cols	Enter the number of columns.

No Layout	Select (check) to apply no layout. Overlays the view of each record on top of one another (each record's view is placed in the same location). Typically, animation configurations are used to set the location at runtime.	
Nav. Btns	Select (check) to insert navigation buttons that allow you to scroll through the set of collection views.	

|<	Scrolls to the **First** view.
<	Scrolls to the **Previous** view.
>	Scrolls to the **Next** view.
>|	Scrolls to the **Last** view.

Executing the Collection Query

With view-based collection filtering, you can execute a collection query without requiring a change to the collection. For example, a control can be linked to a view-in-view's collection filter and used to execute the collection query from within a view.

In the case of a button control, you can explicitly associate it with a view-in-view's collection filter. When the end-user presses the button control at runtime, it will cause the view-in-view to reevaluate its criteria and then execute the collection query.

The combo box and edit controls may also be used when a button control is not associated with a view-in-view's collection filter. In general, any control variable used in an expression that is part of a criteria operand will cause a collection query to be executed. So, an end-user can change the value of a control linked with a collection filter and execute the collection query.

9 | Working with Tags

A *tag* is the unique identification of a single process variable. It is a measured or calculated data element that is uniquely identified in the historians (such as OSI PI, Honeywell PHD, or AspenTech IP.21). It contains the actual value, as well as the meta-data (or descriptors), of the process variable. Meta-data can include items such as the tagID, the TagName, the high limit, and the low limit.

Each tag value needs to be explicitly defined and administered as an instance of an XHQ point or time-series data value. Tag data sources can consist of several thousands of tags. Therefore, given the large quantity of tags typically involved, the manual construction and administration of these point instances can be overwhelming. In most cases, only a small number of these tags are actually being referenced in frequency. Therefore, to simplify the task of building and administrating large number of tag references within XHQ solutions, XHQ has an in-built functionality called *tag synchronization*.

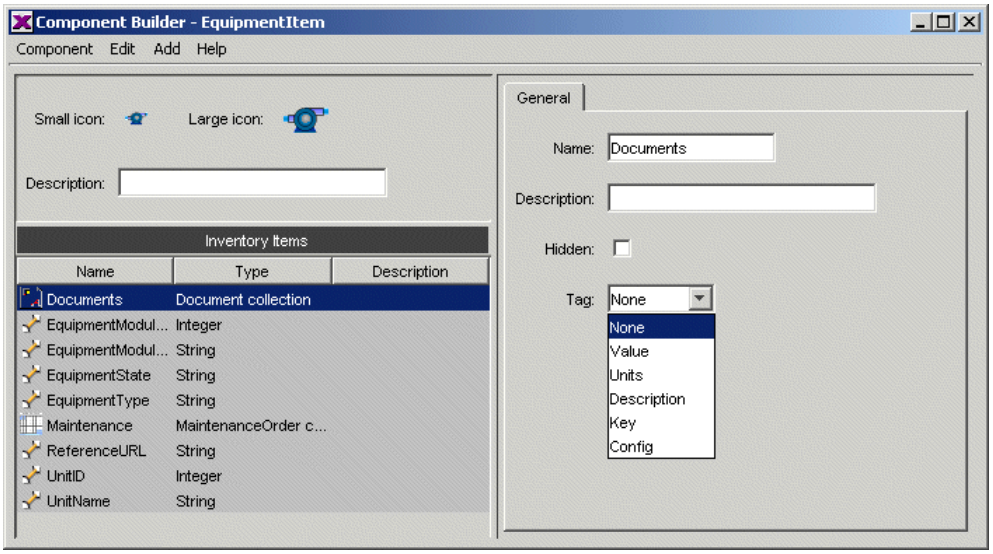
For more information on tag synchronization, go to the topic, *Tag Synchronization*, located in the XHQ Solution Design and Architecture.

How Tags Are Configured

Tag configuration is done from the component level. **Primitive** component member can be configured to have the following tag attributes:

- **None**
  This is the default. No tag attribute is applied to the component member.
- **Value**
  If Value is selected for a member, then that member will be used as the default pen for trending. You can, however, have multiple selections with this tag attribute; in which case, the default pen for trending will be the member with the alias assigned to it (which by default is the one with the first member ID).
- **Units**
  If Unit is selected for a member, then that member will be used to hold the tag unit. Setting this attribute is OPTIONAL.
- **Description**
  If Description is selected for a member, then that member is used to hold the tag's description string. Setting this attribute is OPTIONAL.
- **Key**
  This is the Tag name or ID. Only one member of a tag component can have the Key attribute. This member must be of the type String.
- **Config**
  This attribute is used as a place holder for extra information needed to configure a tag. Typically, it is used to group tags under different categories and determine whether a particular tag needs configuration update.

You configure the tag by setting the Tag Attribute from the General tab of the Component Details tab.



Component Details tab – Tag Configuration

You create tag components the same way you would any other component, with the exception of some additional settings.

Things to consider when configuring tag components

- A tag component consists of only primitive members. Component members are ignored.
- You must configure one of the string members to have the **Key** tag attribute.
- The tag component should have value members for all data types (Real, Integer, String, and Boolean) that are required in the solution.
- All value members must be assigned the **Value** tag attribute.
- The tag attributes, **Units** and **Description**, are optional.
- The **Config** tag attribute is optional. This is used in the solution to determine whether a particular tag needs configuration update.

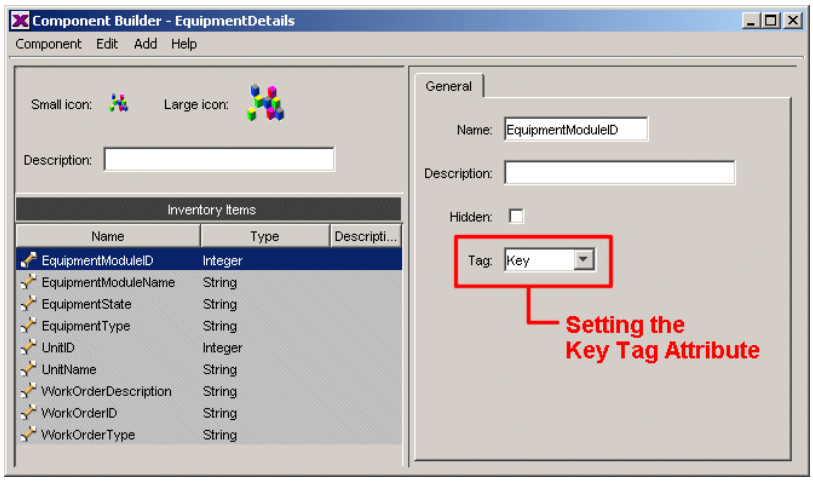
About Tag Collections

A tag collection is a type of global collection that defines the tags within the solution. Tag collections play an important role in tag synchronization. To use tag synchronization, a tag collection must be created that includes all necessary tag meta-data as well as real-time data members.

For more information on tag synchronization, go to the topic, *Tag Synchronization*, located in the XHQ Solution Design and Architecture.

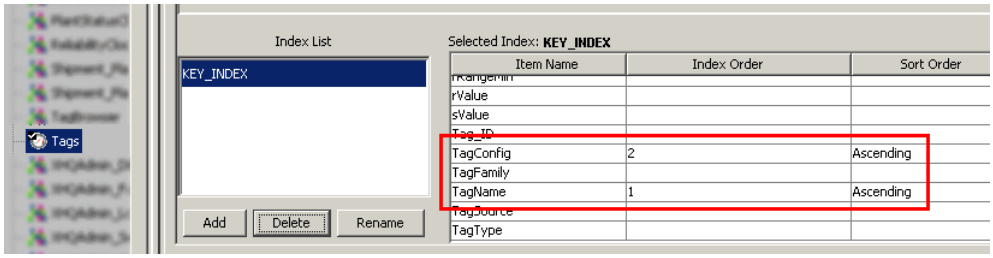
Things to consider when creating your tag collection

- You can only create one tag collection in the solution.
- For the minimum requirement configuration, the component you select to associate with the tag collection must have a primitive member with the **Key** tag attribute set.



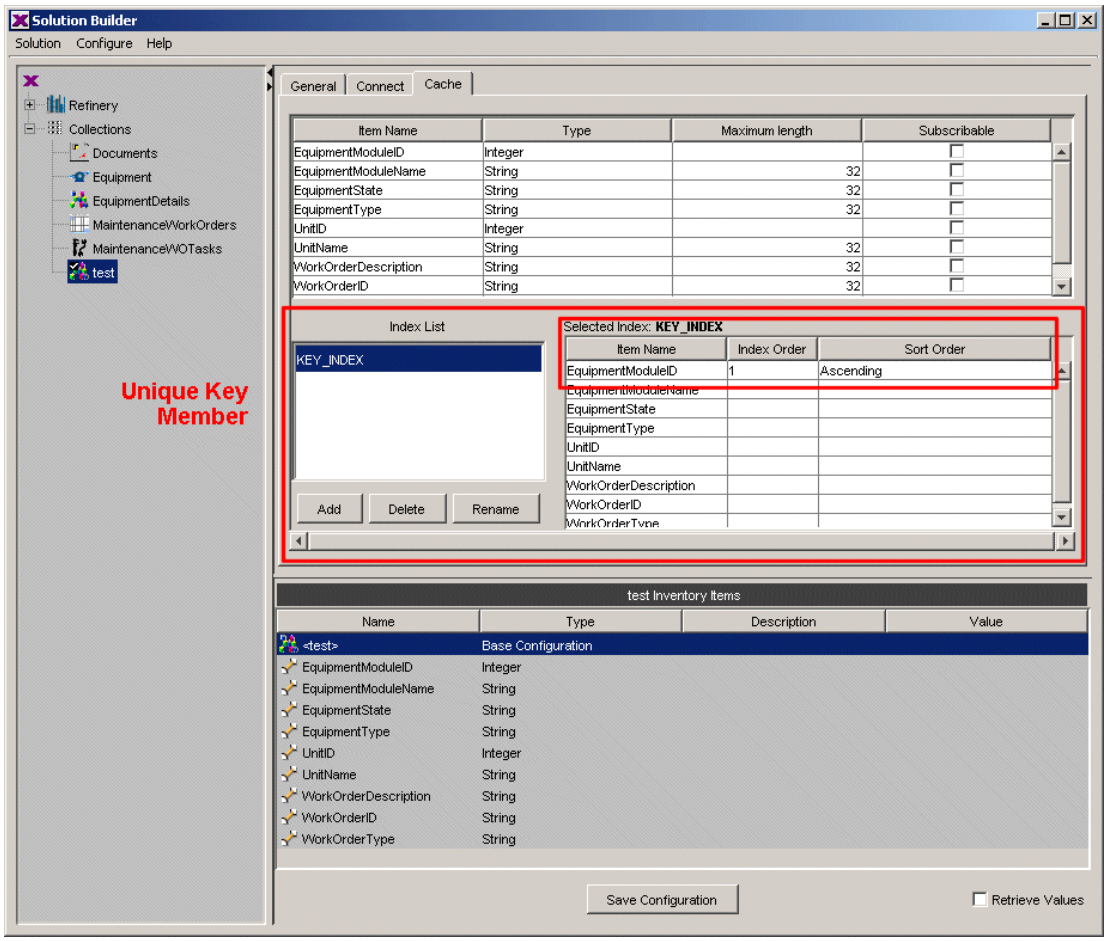
Component Details tab: Setting the Key Tag Attribute

- To expose collection contents, a collection component needs to have a primitive member configured as a **Key**. The tag component to be used for the keyed view should be the one used in the tag collection. If the tag collection is NOT defined in the solution, the XHQ Workbench displays an error message when the keyed view is created. If the tag collection changes after the first keyed view is created in the XHQ Workbench, then you must restart the XHQ Workbench to reflect the new tag component.
- In tag collections that have multiple indexes, add a composite index to optimize data retrieval. When you create the composite index, set the index and sort orders for each of the items (columns) to which you want to apply the composite index.



XHQ Solution Builder: Adding a Composite Index

- Once it is created, the tag collection automatically sets its member with the **Key** tag attribute as the only member of a **Unique Key**. The *unique key* is the tag identifier. It is used to uniquely identify a record within the collection. Only one member of the tag collection can be defined as the unique key.



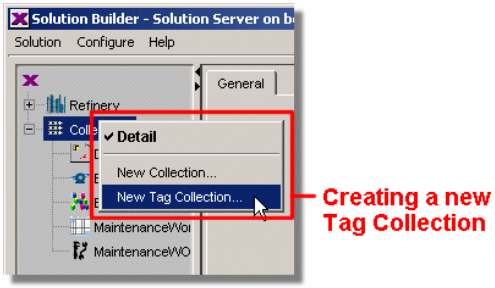
XHQ Solution Builder, Cache Tab: Unique Key Collection Member

- You can rename a tag collection. This will not effect the tags in the solution.
- A tag collection can be a *cache collection*. If multiple tag servers exist, a global collection can be created for each server; these global collections can be joined into one tag collection (a cache collection).
- When you delete a tag collection, a warning message appears asking you to confirm the deletion. You will delete ALL tag objects and all links to it from other objects in the solution.

View references to deleted tags are not removed.

To create a tag collection

1. In the **XHQ Solution Builder** navigation tree, right-click the **Collections** node.



2. From the shortcut menu, select **New Tag Collection**.

   You can only create one tag collection in the solution. Therefore, if one has already been added, this shortcut menu option will be disabled.

The "New Tag Collection" dialog box appears.

3. In the **Name** box, enter the name of the new tag collection.
4. In the **Component** list, you must select a component with a member that has the **Key attribute**.

   If the component you select does not have a member with the Key attribute, an error message appears, stating that "the configuration of the object is invalid."

5. Click **OK**.
   The new tag collection appears on the navigation tree.

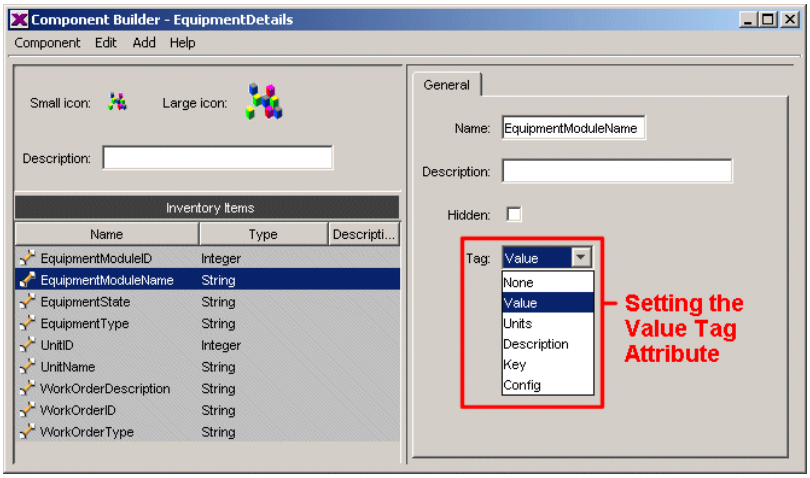
   To distinguish the tag collection from all other collections, a small check mark ( ) appears just above the collection icon.

Setting Subscribable Members

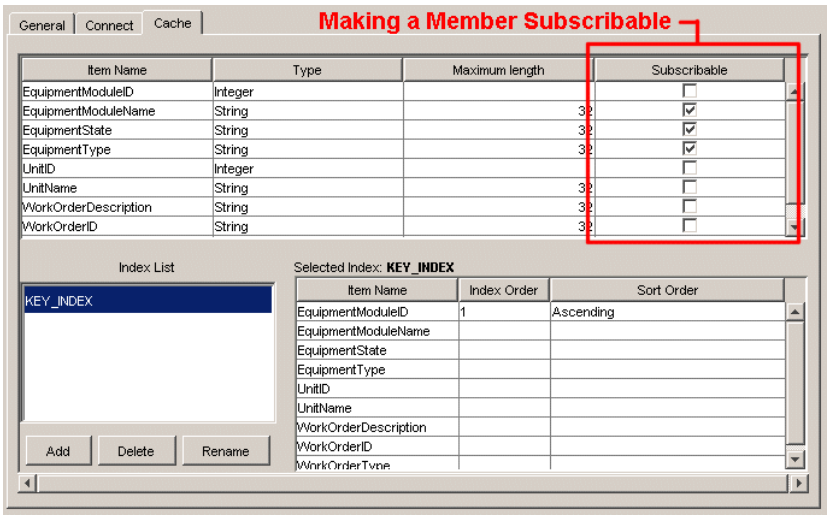
Within the tag, there are two types of data members:

- Members that have their own input configurations. These are *subscribable* members.
- And members that acquire their value from a corresponding column of the Tag collection, in a row where the name of the tag matches the value of the Tag collection member with the Key tag attribute.

Every member with the **Value** tag attribute set (via Component Builder) is by default subscribable. You can set additional collection members as subscribable from the **Cache** configuration panel of the Solution Builder.



Component Details tab: Setting the Value Tag Attribute



XHQ Solution Builder: Making a Collection Member Subscribable

Tags and Views

In this section, you will learn how to:

- Create one view for all your tags regardless of the Value primitive's data type;
- Embed tag views using the Keyed View option;
- Build controllers without having to model them in the solution.

Dealing with Multiple Data Types

You solution may have tags with different data type values (Integer, Real, String, or even Boolean). In order to resolve this, your tag component must have one member with either an **Input** or an **Expression** configuration. Unfortunately, it is not always possible to determine, during development, which member needs to be displayed. The following example shows you resolve this problem.

EXAMPLE: USING THE "STACKING" TECHNIQUE

1. From the XHQ Workbench, **stack** all necessary value members on top of each other in the view drawing panel.
2. For each member, set **Animation** to **Show** the **Value** using the `isconfigured(member_name)` expression.

> Members that do not have either an **Input** or an **Expression** configuration receive a value of `BAD_NOT_CONFIGURED` quality upon subscription. The `isconfigured()` function returns FALSE with either a `BAD_NOT_CONFIGURED` or a `BAD_NON_EXISTENT` quality, and TRUE otherwise. Therefore, only values from appropriately configured members are displayed.

Using this technique, you can use one view for all your tags regardless of the Value primitive's data type.

About Keyed Views

Because tags are not modeled (they are not members of other components in the solution), tag views cannot be embedded in the usual way. To embed a tag view, you will need to use the **Keyed View tool** located on the XHQ Workbench Tools Palette.

Basically, a *keyed view* is a view of an object where the object is identified by its key in a tag collection.

Embedding a tag view is similar (in concept) to inserting a template.
And, like embedded views, trends and trend tables are supported.
Keyed views are also drillable.

After you place the tool on the Drawing panel, the **Keyed View tab** becomes available. Pick a tag component, or a component that extends the tag component, and then select a view to embed. In the **Key Value** text box, enter an expression that results in the name of the desired tag.

During runtime, whenever this view is selected, the expression is processed and its result is sent to the XHQ Solution Server, which resolves the given name into an ID of the tag. If the resolution is successful, the client generates the subscription to all displayed members with the resolved object ID.

To insert a keyed view

1. Launch the **XHQ Workbench**.
2. From the **Tools palette**, click the **Insert Keyed View** tool and place the tool on the Drawing panel.

   A keyed view element placeholder appears.
3. From the **Edit panel**, click the **Keyed View tab**.
4. To select a **Tag Component**, click on the browse button next to the text box.
   The "Select Component" dialog box appears.
5. From the list, select a tag component, or a component that extends a tag component, and then click **OK.**
6. Under **View**, select the view to embed.
7. In the **Key Value** text box, do one of the following:

- **Enter an expression** that results in the name of the desired tag.

or

- Click the **browse** button, and from the Tag Selector dialog that appears, **select a unique key value** and click **OK**.

8. Select a **Navigation** setting.

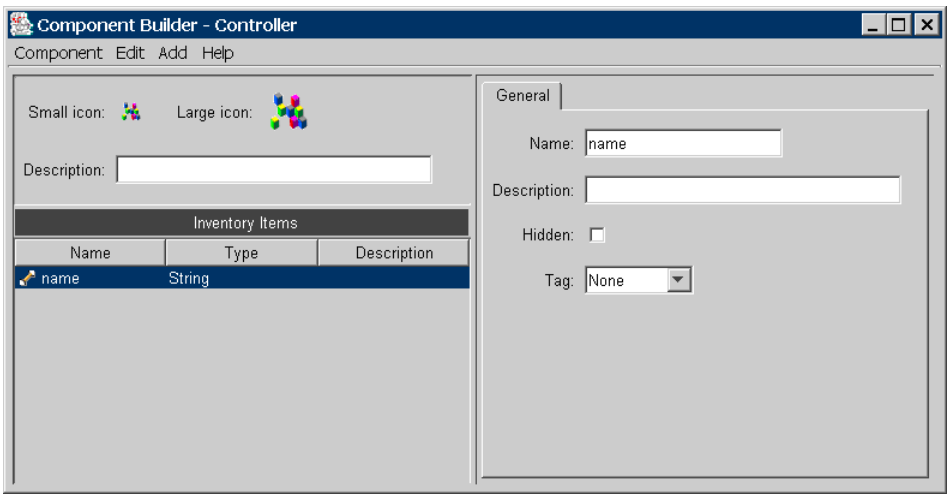
Select …	If you want to …
Use views properties	Use the settings defined in the Views Properties tab
Always drillable	Set the view as always drillable, regardless of whether the view is, by default, drillable or not.
Never drillable	Set the view as never drillable, regardless of whether the view is, by default, drillable or not.

9. **Save** the view.

Working with Composed Tag Views

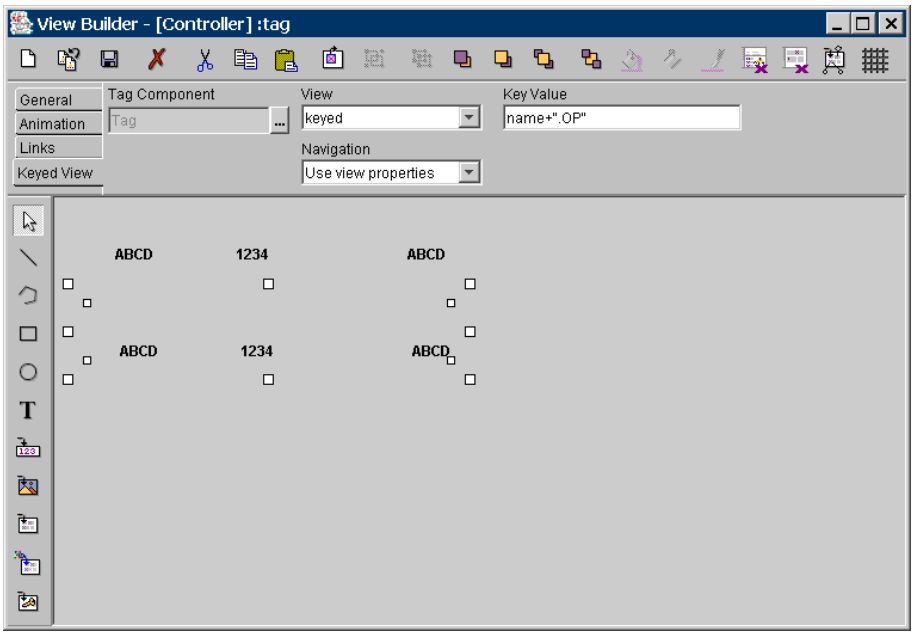
Some solutions have a component which is composed of more that one tag (in typical situations, it is composed of three tags). This component is called a **Controller**. All tags in the Controller have the same name, but are distinguished by a give suffix (for example, .OP, .SP, and .IP). The Controller consists of views which are compositions of views of its tags. Therefore, a Controller can be built without having to model it in the solution.

The Controller is a component with only one member, Name, which references the base part of embedded tags.

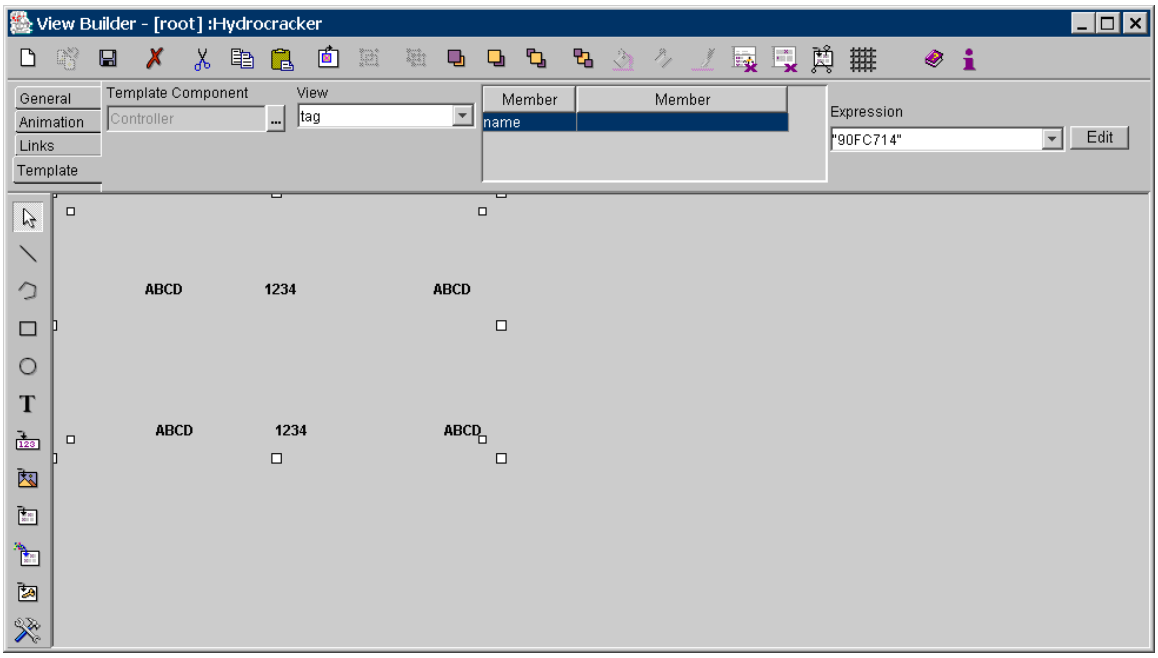


Component Details tab for Controller

The Controller's view consists of "keyed views" of its tags, where the Name member is used in the expression to properly resolve the name of each embedded tag.



The Controller's view is then embedded in the appropriate view as a template. The expression for its Name member results in the base of the desired tag composition. This name is then passed to the expressions of keyed views embedded in the Controller's view with an appropriate suffix, so all necessary linking is complete.



Using the Tag Pickers

Tag Picker is a generic term used to refer to the user interface that allows you to retrieve and view tag collection data. The Tag Picker launches in **two** different modes (dialog boxes); both modes share a similar construct but have different functions:

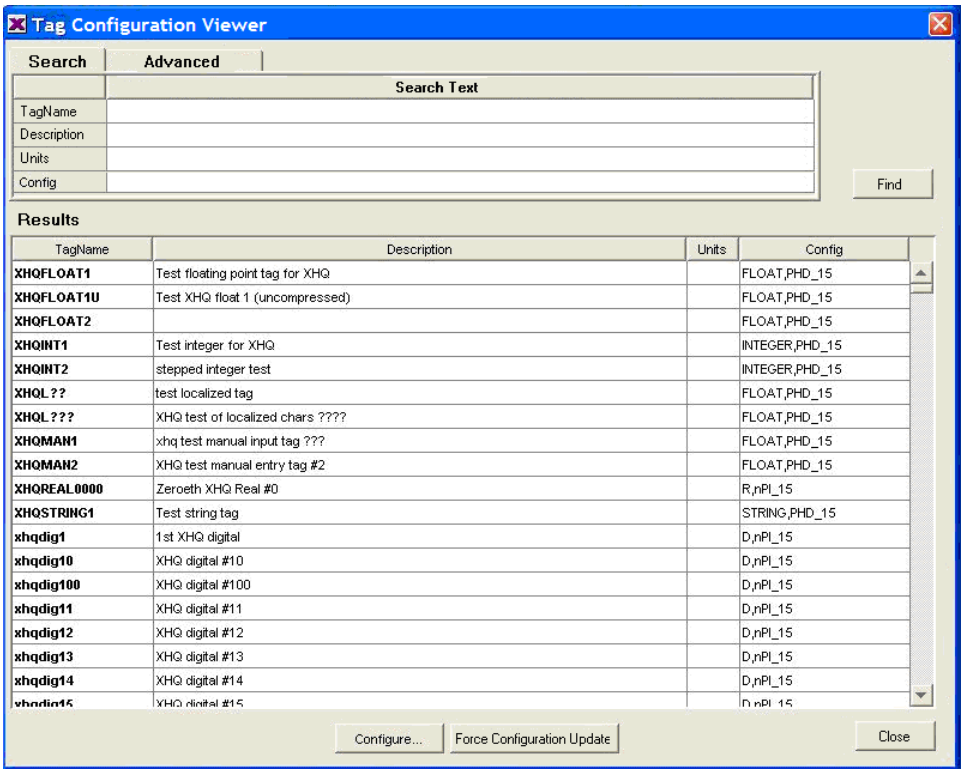
- **The Tag Configuration Viewer**
  This dialog box allows you to view or edit tag configurations, and is launched from the XHQ Solution Builder.
- **The Tag Selector**
  This dialog box allows you to pick a tag name for trending or keyed view, and is launched from the XHQ Workbench and the Trend Viewer.

The Tag Configuration Viewer and the Tag Selector are "Tag Pickers." A key difference between the two is that the Tag Configuration Viewer does not return a value whereas the Tag Selector does.

The Tag Configuration Viewer

The Tag Configuration Viewer retrieves and displays tags and associated information, allowing you to inspect, modify, and even troubleshoot certain aspects of the tag's information and configuration. In this function, it is important that all of the tag collection columns are visible.

When launched from the XHQ Solution Builder, the Tag Configuration Viewer connects to the solution running on the same machine as the model server and allows the solution developer to view and edit tag configurations.



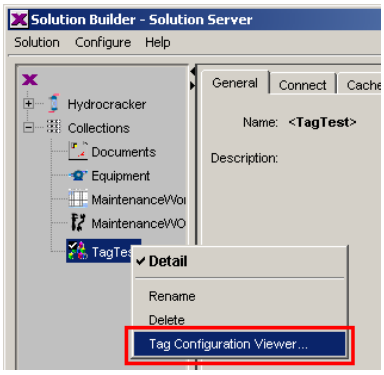
Tag Configuration Viewer launched from the XHQ Solution Builder

To launch the Tag Configuration Viewer from the XHQ Solution Builder

1. Create a tag collection.

   See the topic *To create a tag collection*.

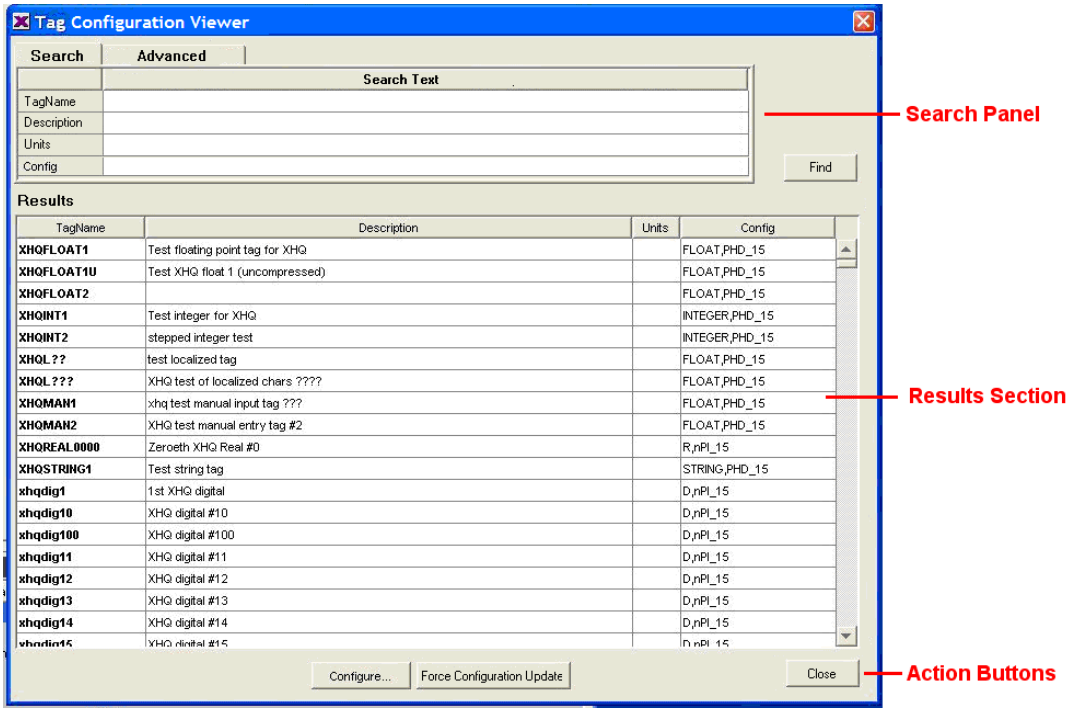
2. Right-click on the tag collection node.



3. From the shortcut menu, select **Tag Configuration Viewer**.
   The "Tag Configuration Viewer" dialog box appears.

About the Tag Configuration Viewer Dialog Box

The Tag Configuration Viewer dialog box consists of three areas: the Search panel, the Results section, and the bottom row action buttons.



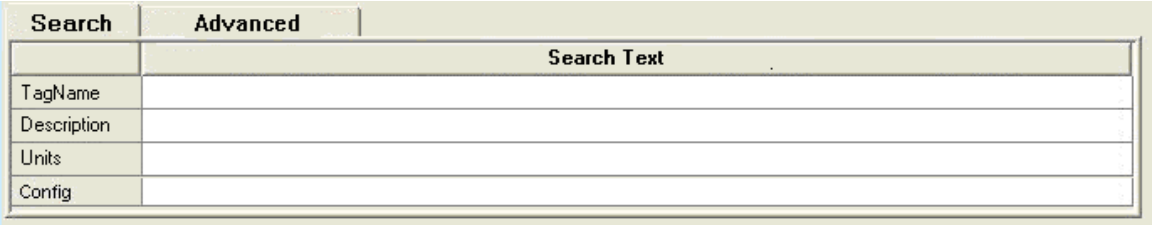
Tag Configuration Viewer Dialog Box

The Search Panel

From this panel, there are two options to set your search requirements, using the basic or the advanced search technique.

Basic Searching

Basic searching is provided through the **Search** tab, which consists of all non-subscribable members. In the example image below, four fields (or columns) are defined: TagName (or Key), Description, Units, and Config.



Example: Basic Search Tab

For each of the fields, you can set the **Search Text,** or keep it blank, or use any combination. Search text is **case-sensitive**.

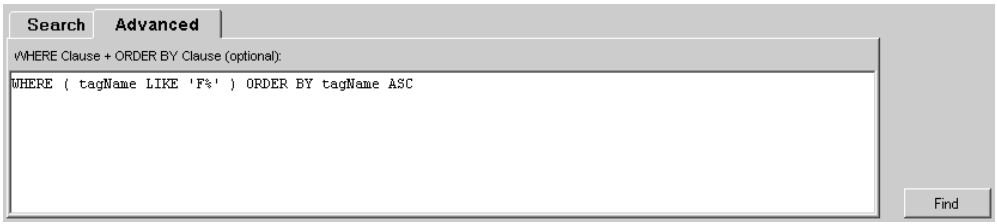
If Search Text is left **blank**, then the WHERE clause is omitted for that item. This allows you to have NULL values in columns.

If it is **not blank**, then the system automatically adds a wildcard character to the beginning and end of the string you enter.

Advanced Searching

From the **Advanced** search tab, you can manually modify/create the search requirements using database syntax.

This option is for advanced SQL users.



Example: Advanced Search Tab

Some additional things for you to consider when using Advanced searching:

- Changes made in the "Advanced" tab are not reflected in the basic "Search" tab.
- The WHERE clause is the logical equivalent of the search text set using the basic "Search" tab; the SQL WHERE clause, however, overrides the search text set from the "Search" tab.
- Search text specified in the basic "Search" tab form a "x LIKE SearchText" construct.

When entering or editing the query criteria, always click **Find**, located next to the Search panel, to apply the current set of search requirements, retrieve the appropriate results, and display them in the Results section.

To enter (or edit) search text

1. Under the Search Text column, click **inside** the cell.
2. **Enter** (or edit) a string expression and click **Enter**.

   Search text is case-sensitive.

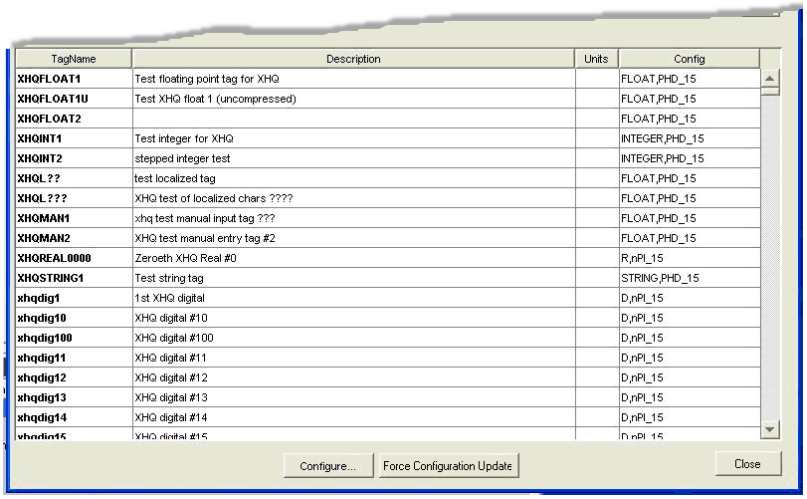
3. Click **Find**, located next to the Search panel, to apply the current set of search requirements, retrieve the appropriate results, and display them in the Results section.

The Results Section

The results that appears in this table are based on the current content of the tag collection configuration in the solution, as well as the search requirements you set.

This table displays data for all non-subscribable members. The first column, which represents the collection tag name, is the unique key column. In this column, the tag names are in bold font.

By default, you must first click the **Find** button in order to display results in the Results table.



TagName	Description	Units	Config
XHQFLOAT1	Test floating point tag for XHQ		FLOAT,PHD_15
XHQFLOAT1U	Test XHQ float 1 (uncompressed)		FLOAT,PHD_15
XHQFLOAT2			FLOAT,PHD_15
XHQINT1	Test integer for XHQ		INTEGER,PHD_15
XHQINT2	stepped integer test		INTEGER,PHD_15
XHQL??	test localized tag		FLOAT,PHD_15
XHQL???	XHQ test of localized chars ????		FLOAT,PHD_15
XHQMAN1	xhq test manual input tag ???		FLOAT,PHD_15
XHQMAN2	XHQ test manual entry tag #2		FLOAT,PHD_15
XHQREAL0000	Zeroeth XHQ Real #0		R,nPI_15
XHQSTRING1	Test string tag		STRING,PHD_15
xhqdig1	1st XHQ digital		D,nPI_15
xhqdig10	XHQ digital #10		D,nPI_15
xhqdig100	XHQ digital #100		D,nPI_15
xhqdig11	XHQ digital #11		D,nPI_15
xhqdig12	XHQ digital #12		D,nPI_15
xhqdig13	XHQ digital #13		D,nPI_15
xhqdig14	XHQ digital #14		D,nPI_15
xhqdig15	XHQ digital #15		D,nPI_15

Example: Results Section for Tag Configuration Viewer

Action Buttons

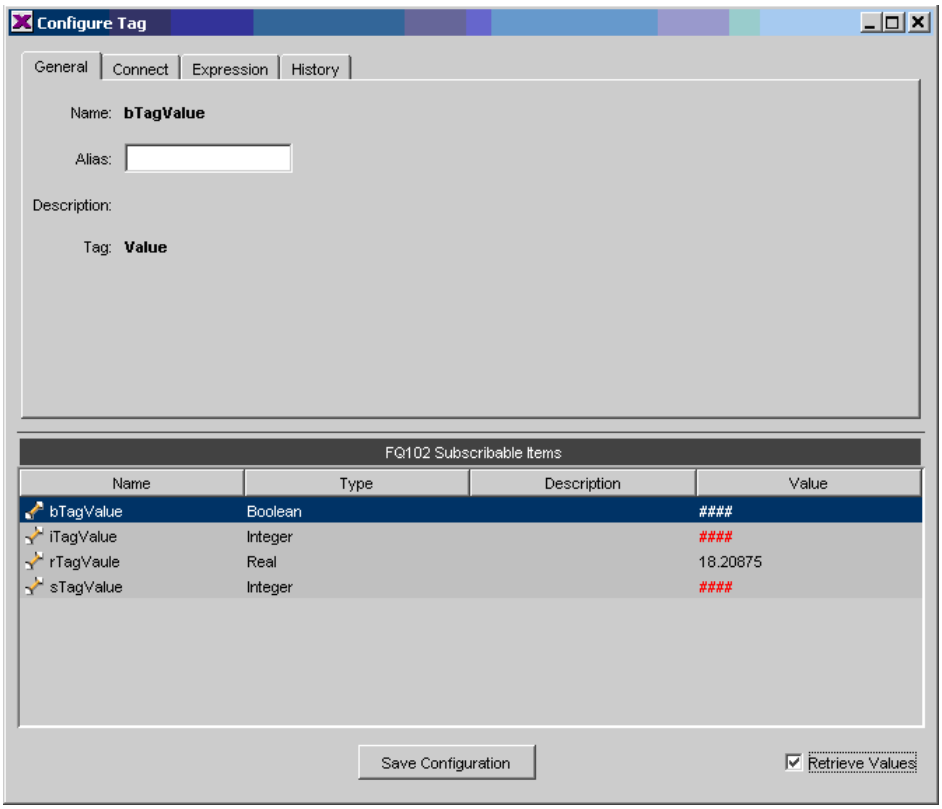
The buttons on the bottom row of the Tag Configuration Viewer dialog box have the following states and actions.

Button Descriptions

Button	Initial State	Is Enabled when	Action on Click
Find	Enabled	N/A (always active)	Retrieves the list of tags from the collection to display in the Results table.
Configure	Disabled	A row is selected.	Launches (or activates) the "Configure Tag" dialog box.
Force Configuration Update	Disabled	A row (or set of rows) is selected.	Forces the solution server to generate an XML for all currently selected rows; invokes the standard Tag Sync batch file.
Close	Enabled	N/A (always active)	Closes the Tag Configuration Viewer.

The Configure Tag Dialog Box

This dialog box allows you to view or edit a single tag configuration for subscribable members.



Example: Configure Tag

There are two ways to launch this dialog from the Tag Configuration Viewer collection data table:

- Select a row and click the "Configure" button.
- Double-click inside the cell of a row.

Once launched, the Configure Tag dialog is populated with the given tag configuration. Editing the tag configuration is similar to editing a component instance in the solution.

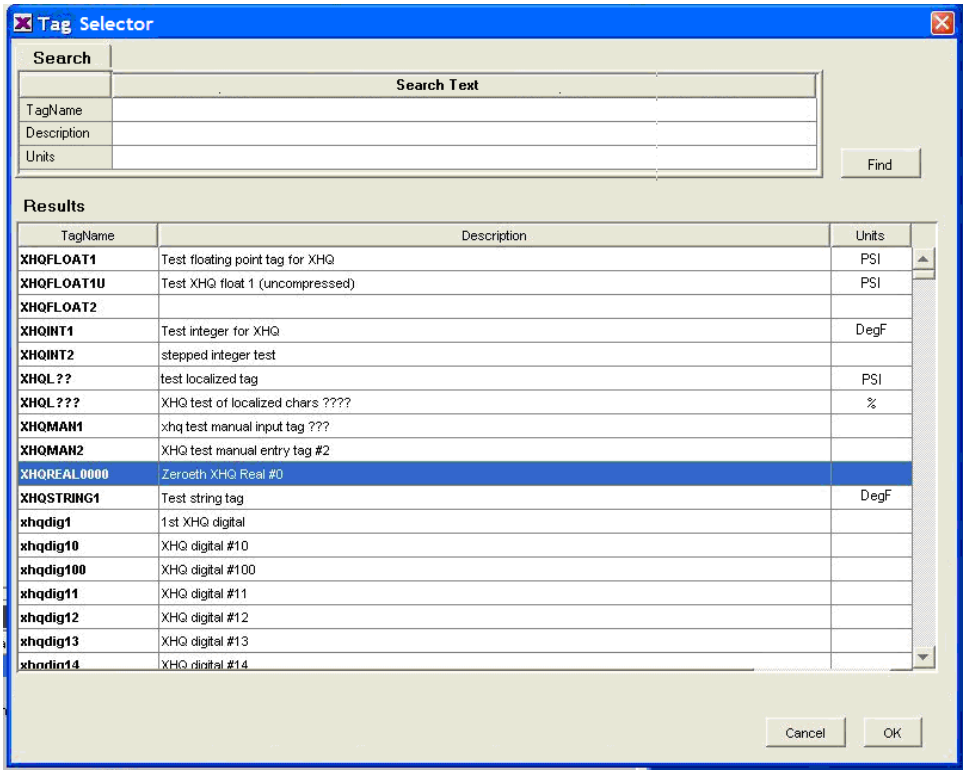
Tag synchronization provides batch tag configurations through an automated tag configuration XML import. Therefore, editing the tag configuration is typically useful in development environments where the automated tag configuration XML import process might not be functioning properly.

For more information on tag synchronization, go to the topic, *Tag Synchronization*, located in the XHQ Solution Design and Architecture.

Selecting (checking) the **Retrieve Values** checkbox displays live data for tags with proper configurations. Use this option to validate the tag configuration.

The Tag Selector

The primary purpose of the Tag Selector is to provide a means to select tags. When used from the XHQ Trend Viewer, the Tag Selector allows you to select multiple rows so that multiple tags can be returned to the trender for plotting. When used from the XHQ Workbench, it allows you to select the tag to be displayed by a Keyed View or a Value Tag animation.



Tag Selector

Unlike the Tag Configuration Viewer (which launches from the XHQ Solution Builder), the Tag Selector does not have the "Configure" and "Force Configuration Update" options at the bottom of the dialog box.

About Value Tag Animations

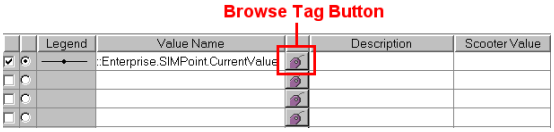
Value Tag animations are the only means of doing client-side expressions that contain multiple tag values. For example, you create a Value Tag animation for tag 1, `$Value1`, and another for tag 2, `$Value2`. Then the expression would simply be `$Value1 + $Value2`.

Things to note

- The Value Tag animations for tag 1 and tag 2 may be hidden if their actual values are not necessary to the view.

Launching the Tag Selector from the Trend Viewer

When launched from the XHQ Trend Viewer, the Tag Selector allows users to select the tag to be trended. Clicking on the Browse Tag button, next to the Value Name column, launches the Tag Selector. The Tag Selector displays the tag collection.



Launching the Tag Picker from the XHQ Trend Viewer

After selecting a tag name and clicking the OK button in the Tag Selector, the **Value Name** field is populated with the tag name you selected.

You must set the global property **`ShowTagBrowserButton=true`** in order to see this button.

For more information on setting the aggregation intervals, refer to the topic, *Using globalsettings.properties*, located in the XHQ Administrator's Guide.

Launching the Tag Selector from the XHQ Workbench

When launched from the XHQ Workbench, the Tag Selector allows application developers to select the tag to be displayed in either a Keyed View or a Value Tag animation.

To launch the Tag Selector from the XHQ Workbench (for Keyed Views)

To learn more about Keyed Views, go to the topic, *About Keyed Views*.

1. From the **Tools palette** of the XHQ Workbench, click the **Insert Keyed View** tool and place the tool on the Drawing panel.
   A keyed view element placeholder appears.
2. Click the **Keyed View property**.
3. Next to the **Key Value** text box, click the browse button.
   The "Tag Selector" dialog box appears.
4. From the Tag Selector, **select a unique key value** and click **OK**.
   The "Key Value" field is populated with the unique key value you selected.

To launch the Tag Selector from the XHQ Workbench (for Value Tag Animation)

For more information, go to the topic, *Using Animation*.

1. From the **Tools palette** of the XHQ Workbench, click the **Value tool** and place the value on the Drawing panel.
   A value element placeholder appears.
2. Click the **Animation property**.
3. Click **Add** and, from the shortcut menu, select an animation from the list of available options.
   The animation you select is displayed in the "Set" column of the Animation table.

   The options listed on this drop-down shortcut menu depends on the type of view element you are animating.

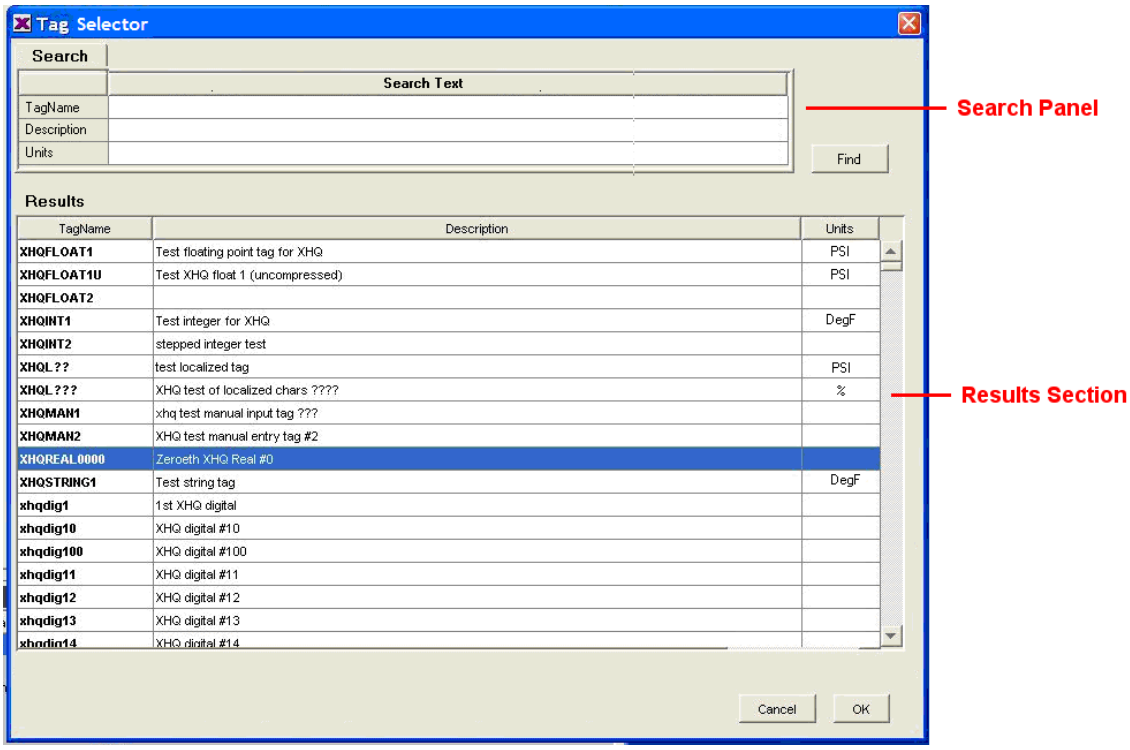
4. From the **Using** drop-down list, select the **Value Tag** animation type.
   The animation type you select is displayed in the "Using" column of the Animation table.
5. Next to the **Tag Component** text box, click the browse button.
   The "Select Tag Component" dialog box appears.
6. **Select** from the list of tag components available and then click **OK**.
7. From the **Key Value** text box, do one of the following:
   - Click the **browse button**. The "Tag Selector" dialog box appears; **select a unique key value** and click **OK**. The "Key Value" field is then populated with the unique key value you selected.

   or

   - **Enter a string** expression that results in the name of the desired tag.
8. From the **Evaluating** drop-down list, **select a tag component member** to drive the animation.
   The members listed are those contained in the tag component for which you are building the current view.
9. **Save** the view.

About the Tag Selector Dialog Box

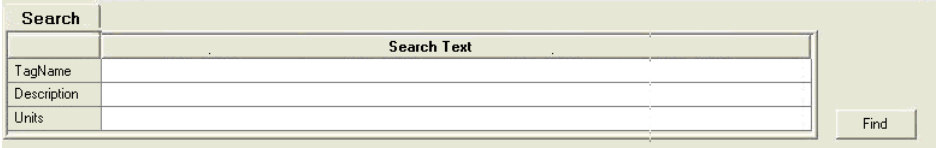
The Tag Selector dialog box consists of two areas: the Search panel and the Results section.



Tag Selector Dialog Box

The Search Panel

The **Search** tab consists of three defined fields: TagName (or Key), Description, and Units.



Example: Search Tab for Tag Selector

For each of the fields, you can set a **Search Text** (or keep it blank, or use any combination). Search text is **case-sensitive**.

If Search Text is left **blank**, then the WHERE clause is omitted for that item. This allows you to have NULL values in columns.

If it is **not blank**, then the system automatically adds a wildcard character to the beginning and end of the string you enter.

To enter (or edit) search text

1. Under the Search Text column, click **inside** the cell.
2. **Enter** (or edit) a string expression and click **Enter**.

   Search text is case-sensitive.

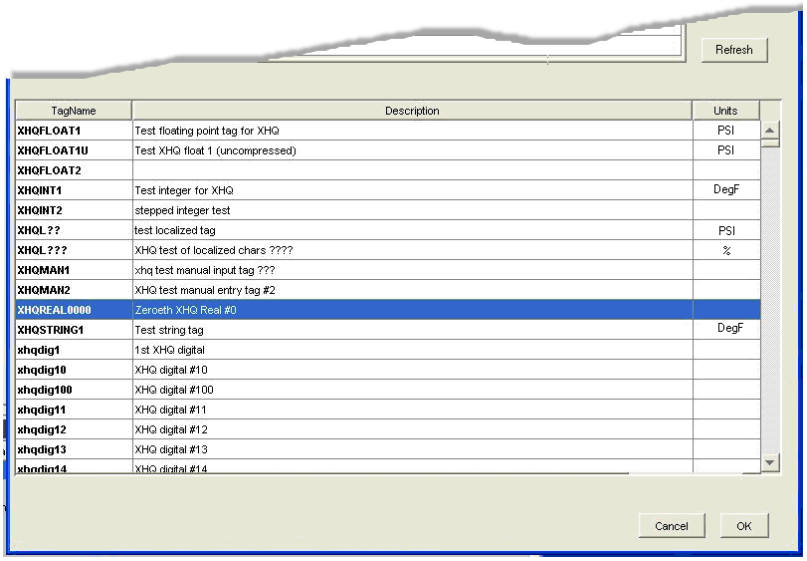
3. Click **Find**, located next to the Search panel, to apply the current set of search requirements, retrieve the appropriate results, and display them in the Results section.

The Results Section

The results that appears in this table are based on the current content of the tag collection configuration in the solution, as well as the search requirements you set.

This table displays data for all non-subscribable members. The first column, which represents the collection tag name, is the unique key column. In this column, the tag names are in bold font.

By default, you must first click the **Find** button in order to display results in the Results table.



Example: Results Table for Tag Selector

When launched from the **Trend Viewer**, select a tag name and click the OK button. The "Value Name" field of the Trend Viewer is then populated with the tag name you selected.

When launched from the **XHQ Workbench**, the Tag Selector allows application developers to select the tag to be displayed in either a Keyed View or a Value Tag animation.

For more information, go to topics: *Launching the Tag Selector from the Trend Viewer*, or *Launching the Tag Selector from the XHQ Workbench*.

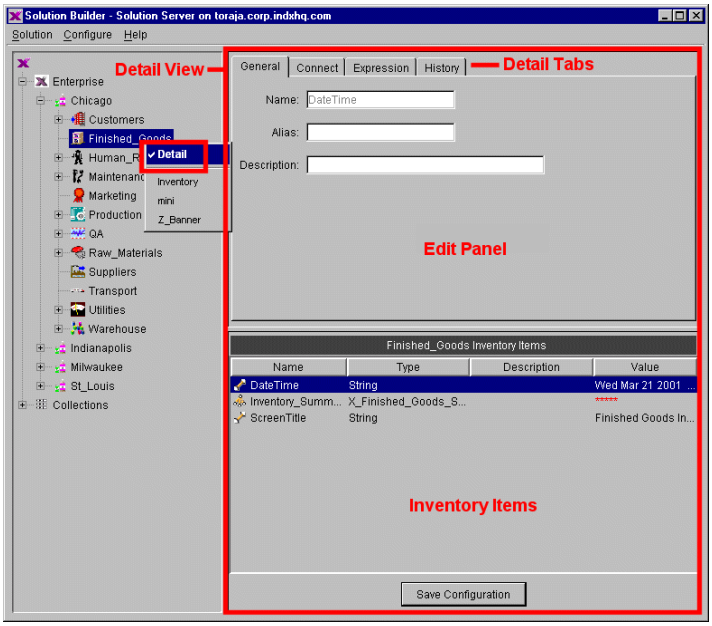
10 | The XHQ Solution Builder

The XHQ Solution Builder is part of the XHQ development environment that allows you to build a solution by using the components in the XHQ Workbench. The XHQ Solution Builder provides the environment for you to declare how the actual instances of the components in a solution operate, and how they extract data from backend data sources. A *solution* is a logical grouping of software components that represents the physical or logical elements of a business. Solutions are made up of components containing connections, data, events, history, views, and other relevant information.

In this section, you will learn how to work with the XHQ Solution Builder user interface and menu commands.

Touring the XHQ Solution Builder Environment

The XHQ Solution Builder launches a standard layout for preparing the structural design of a new solution, or for displaying the structure of an existing solution.



XHQ Solution Builder – Detail View

The XHQ Solution Builder is a standalone client and opens in an independent window that, in Detail view mode, consists of the Navigation panel, the Inventory Items panel, and the Edit panel.

A menu bar appears at the top of the XHQ Solution Builder window. Depending on the item you select in the Navigation or Inventory Items panels, the tabs in the Edit panel change to reflect appropriate context.

Prior to building your solution, the connector type you plan to use should already be injected.

For more information on connectors, refer to the *XHQ Connection Guide*.

To launch the XHQ Solution Builder

1. On the taskbar, click the **Start** button, point to **Programs**, and then, point to **XHQ devClient**.
2. Click the XHQ Solution Builder.

If security has been enabled, you will be prompted for a username and password.

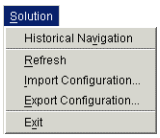
For more information on XHQ security, see the section, *Security, Access, and Privileges*, located in the XHQ Administrator's Guide.

XHQ Solution Builder Menu Commands

This section describes all of the XHQ Solution Builder menus and their respective commands.

XHQ Solution Builder Menu Toolbar

The Solution Menu Commands

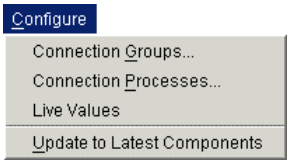


Solution Menu, XHQ Solution Builder

Solution Menu Commands

Menu	Commands	Action
Solution	**Historical Navigation**	Allows you to select a historical time to apply to XHQ views and to force refresh updates at incremental time periods.
	Refresh	Refreshes the current views.
	Import Configuration	Imports configurations for selected connection group(s).
	Export Configuration	Exports configurations for selected connection group(s).
	Exit	Exits the XHQ Solution Builder.

The Configure Menu Commands



Configure Menu, XHQ Solution Builder

Configure Menu Commands

Menu	Commands	Action
Configure	**Connection Groups**	Adds, edits, or removes a connection group.
	Connection Processes	Adds, edits or removes a connection process.
	Live Values	Displays current values in the Inventory Items panel.
	Update to Latest Components	Updates the solution to include the latest changes to all components in the structure. **Note:** An **Update To Latest** (UTL) does not disable `dbexport`. A DB export automatically follows a UTL when namespace changes have occurred.

The Help Menu Commands

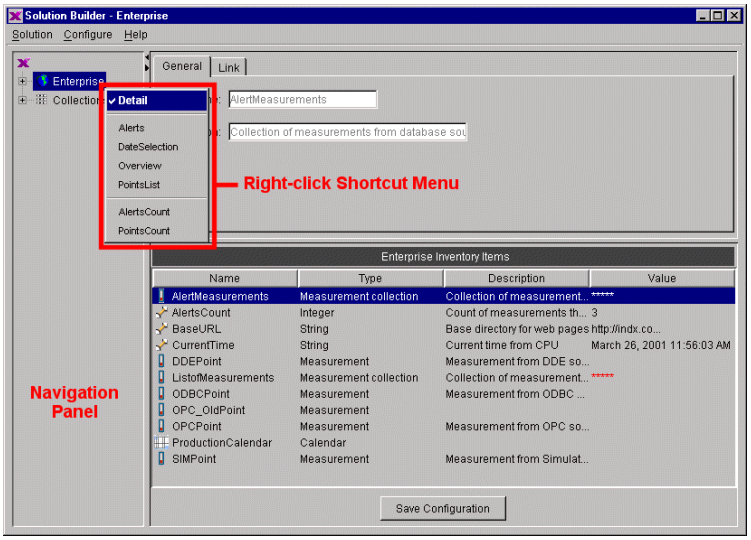
Help Menu Commands

Menu	Commands	Action
Help	Help Topics	Displays task and reference related Help topics for the XHQ Solution Builder.
	About Solution Builder	Displays version, build, and copyright information.

Using the Navigation Panel

The Navigation panel provides a tree-style view of the solution. Using names and/or icons, the tree-style view shows all navigation nodes within a solution. A node can be a component, a member component, or a configured collection. By pointing and clicking, you can navigate through the tree and select and display views for a node. The order of the nodes and leafs in the navigator is dictated by the definition and configuration of components.

When you select a node by double-clicking, you see a list of inventory items associated with that component in the Inventory Items panel. You can right-click on a node for an alphabetical listing of all the views for that node. At development time, the structural view (the Detail view) of a component is the default view, which appears first.



Navigation Panel, Right-click Shortcut Menu

When you click on any node in the navigation panel, the selected node highlights, and the Detail view changes to reflect information corresponding to the selected node. However, at runtime, the Detail view is not presented in the list, and the view that shows at the top of the right-click view list becomes the default view.

You can resize the Navigation panel to provide more viewable area for the navigation tree. You can also click the left or right arrow icon on the split bar (between the navigation and the detail panels) to allow either panel to occupy the whole window.

To select a node or a component in the navigation panel

Click on the icon representing a node or component.
The XHQ Solution Builder defaults to the Detail view, which consists of the Inventory Items and Edit panels for the selected node or component.

To expand or collapse a node in the navigation panel

Do one of the following:

- Double-click on the icon representing a node or component to expand or collapse that node in the tree. When expanded, any component contained within a component or node is displayed directly below the selected icon.

or

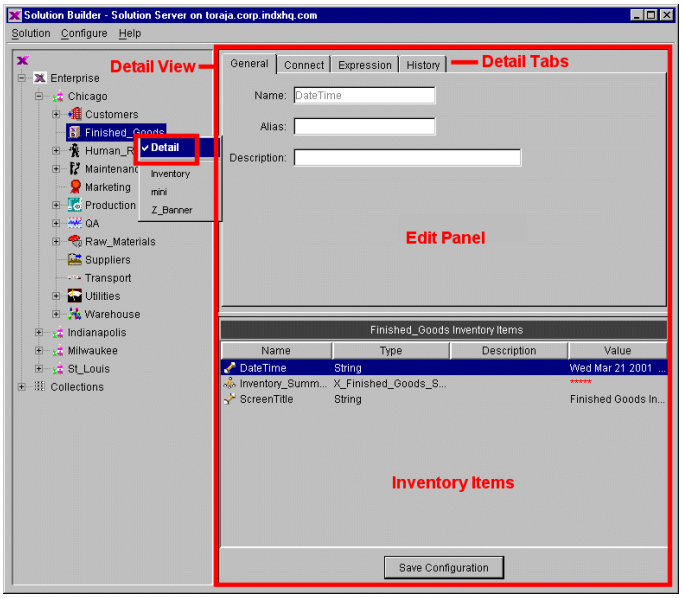
- Click the plus (+) sign next to the node icon to expand or the minus (-) sign to collapse.

To resize the Navigation panel

1. Point anywhere along the vertical split bar between the navigation and detail panels.
   The pointer changes.
2. Click down and drag the split bar to any position.

About The Detail View

The Detail view consists of two panels, the Inventory Items panel and the Edit panel.



XHQ Solution Builder: Detail View

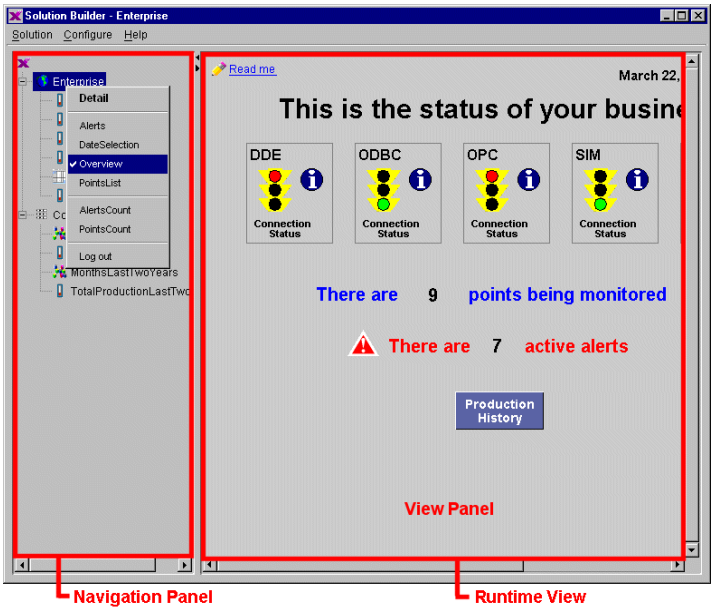
In the **Inventory Items panel**, you see a detail list view of the node or component represented by the selected icon in the navigation tree. Details include name, type, description, and value. Above the Inventory Items panel you see the Edit panel for the item selected in the inventory list. The Detail tabs organize the editing functions. You can perform different editing operations on the selected member of that tree node by switching from one tab to another.

For more information on the Detail view and the Edit panel tabs, refer to the section, *Building and Configuring Solutions*.

Accessing Views

By default, the XHQ Solution Builder displays the Navigation and Detail views. However, the XHQ Solution Builder can also show any of the available views for a component or node. To display a view, simply select the node from the navigation panel and right-click. A shortcut menu appears. Select from the views that are available.

The view submenu is similar to the one for the XHQ Workbench. For more information, go to the topic, *View Submenu Grouping*.

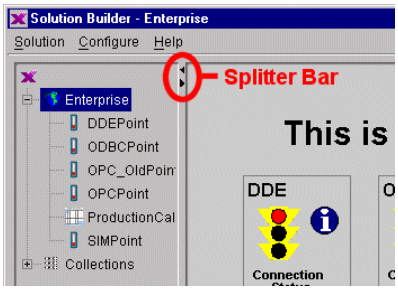


Displaying a View in the XHQ Solution Builder

Note that the **Detail view** is always indicated at the top of the view list in the right-click shortcut menu. Selecting any view in the list results in its display in the view panel.

To work in either navigation or detail view

1. Note the right and left arrows at the top of the vertical split bar.



2. Do one of the following:
   - Point and click on the arrow pointing **LEFT** (<). The **Detail view** occupies the entire window.

   or

   - Point and click on the arrow pointing **RIGHT** (>). The **Navigation view** occupies the entire window.
3. To **revert back** to both views showing, click on the **opposite arrow**.

About Tooltips

Tooltips are notes that appear on the screen when you mouse over a graphical image or a value item within a view.

For graphical image Tooltips:

- The **first** (top) line is the name of the component against which the current view is built.
- And, if the image is "drillable", a **second** line appears displaying the name of the view that launches if you select the link. A rectangular blue line appears around the image, indicating it is selectable.

For value item Tooltips:

- Depending on your view, the **first** line displays either the name of the component against which the current view is built or an expression relating to the given value item.
- The **second** line displays the Value.
- The **third** line displays the Time.
- The **fourth** line displays the Quality.

The behavior of Tooltips is the same in the browser runtime environment as it is in the XHQ Solution Builder.

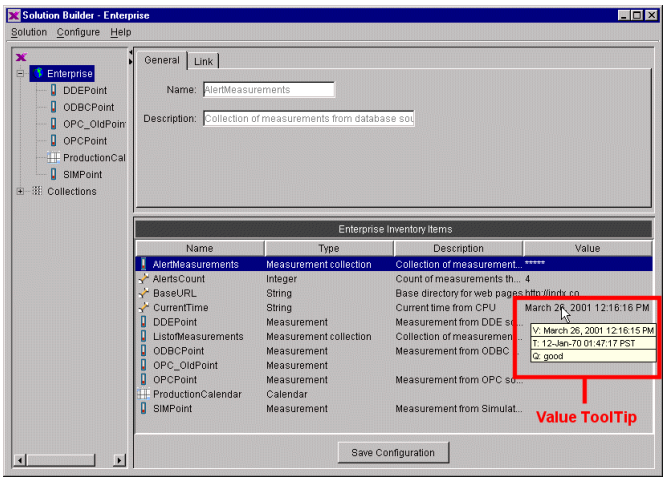
You can also customize Tooltips. For more information, go to the topic, *Customizing Tooltips*.

About the Tag Tooltip

If the value item is associated with a component of the type **Tag**, the appropriate primitives are set as **Units** and **Description**. Therefore, the second and third lines display Units and Description, respectively. If either Units or Description are not set at the time of configuration, then the corresponding Tooltip line does not display.

About the Value Tooltip

Tooltips are also available from the Detail view when you mouse-over a Value under the Inventory Items panel. In this case, the Tooltip displays the VTQ (Value, Time, Quality) for the given item.



Good Value Tooltip Example

For a list of BAD quality indicators, go to the topic, *About Quality*.

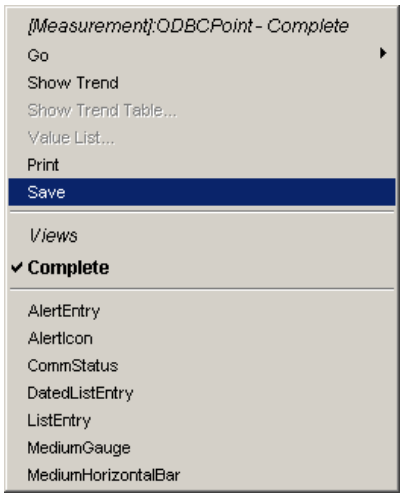
Saving Views

You can save a view in either **.jpeg**, or **.png** format. The file is saved at the location specified by the environment variable **%APPDATA%\XHQ\XHQ devClient\repos\temp\print**.

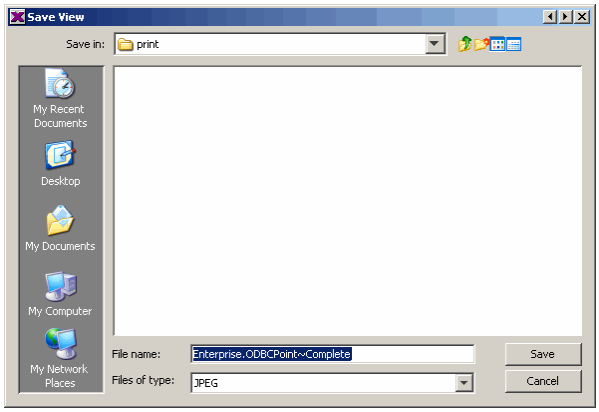
Example Path: `C:\Documents and Settings\johnsmith\Application Data\XHQ\XHQ devClient\repos\temp\print\acme.png`

To save a view from the XHQ Solution Builder

1. With the view displayed in the View Panel of the XHQ Solution Builder, **right-click** on the view.
   The right-click shortcut menu appears.



2. Click **Save**.
   The "Save View" dialog box appears.



3. Do one of the following:
   - Accept the default file name and click **Save**.

   or

   - **Enter** a file name and click **Save**.

Accessing Trends

In order to launch the XHQ Trend Viewer from the XHQ Solution Builder, you must have WRITE access to the `%XHQ_SERVER_REPOS%\temp` folder. In addition, the global property `webserver` must be defined for the given web server.

Before launching the XHQ Trend Viewer from the XHQ Solution Builder, you must verify that the item's **connection** and **history** are configured properly. This is done through the Details view of the XHQ Solution Builder.

To verify an item's connection and history

1. From the **Navigation panel** of the XHQ Solution Builder, navigate to and select the component.
   The Detail view for that component appears.
2. From the **Inventory Items** panel, click the **item**.
   The Detail view tabs appear.
3. Click the **Connect tab**.
4. Verify the **Connection Group**.
5. Click the **History tab**.
6. Verify the **Historian**.
7. If changes were made, click **Save Configuration**.
8. **Repeat** steps 1 through 7 for each item.

Your next step is to select each value item to view and launch the XHQ Trend Viewer.

To select a value item and launch the XHQ Trend Viewer

1. Using the **Navigation panel** of the XHQ Solution Builder, **open the view** that contains the value item.
2. **Point the mouse** cursor directly on the value item (where the values are updating) and **right-click**.
   The right-click shortcut menu appears.
3. From the shortcut menu, click **Select**.
   This adds the value item to the trend list. Once you have selected a value item, the commands "Show Trend", Show Trend Table", and "Value List" become available.
4. **Repeat** steps 1 through 3 for each value item to trend.
   You can add up to sixteen values to view in the XHQ Trend Viewer simultaneously.

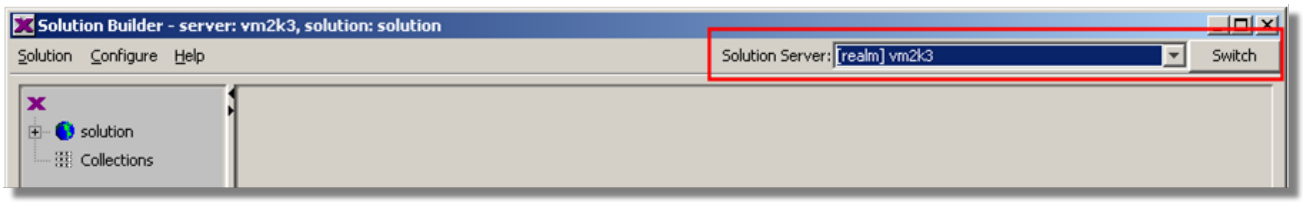
   If you add more than sixteen values, the "oldest" (the first one selected) is omitted to make room for the "newest" (the most recently added).

5. Point to a value item you selected, right-click, and then click **Show Trend**.
   The XHQ Trend Viewer appears.

   To learn how to use the XHQ Solution Viewer, see the *XHQ Solution Viewer User's Guide*.

Accessing Solution Servers

If the XHQ Solution is associated with multiple Solution Servers, Then the XHQ Solution Builder displays an additional box that enables you to select from a list of solution servers (which were added through the Solution Administration Utility).



XHQ Solution Builder: Selecting a Solution Server

The XHQ Solution Builder automatically connects to the previous server with a successful connection. If the Solution Server you choose is unavailable, an error message appears.

From the navigation tree, to identify which nodes are local and which are remote, the icon is appended with an overlay of the letter "L" for *link* or the letter "R" for *remote*.

For more information on the Solution Administration Utility and distributed solutions, refer to the topic, *Distributed Solutions*, located in the XHQ Administrator's Guide.

11 | Building and Configuring Solutions

In this section, you will learn how to effectively use the XHQ Solution Builder to:

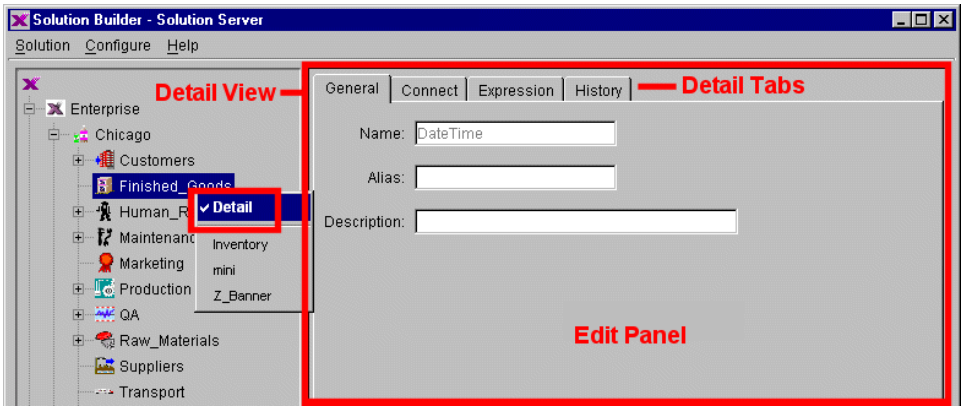
- Modify and adjust a solution component by entering configuration parameters in the fields of the Detail View tabs.
- Use expressions (for example, arithmetic formulas and Boolean logic) to configure and extend a component.
- Configure history using the XHQ Data Recorder.

XHQ Solution Builder Terms to Know

Terms	Definition
Association	An *association* is a relation between items in the namespace. An association can be made from one item (such as an object, a collection, a group, or a tag) and any other item. The association has a relation name. The associated object has a unique name within the context of the relation.

Using Detail View Tabs

A *component* is a shell or a structure that comes to life in a solution with data from the backend. The component in a solution, also called an *instance*, behaves in a way that is defined by the combination of its parent and its own specific attributes. The properties you enter for a solution component are the characteristics that make it uniquely different from all other similar components. Simple examples are name and description, but more complex properties include specific connections and history. These properties are logically grouped in the Detail tabs of the Edit panel.

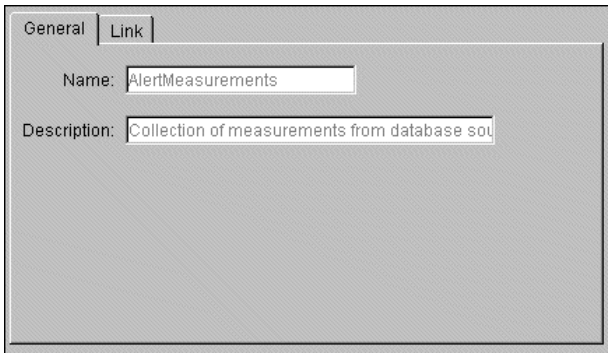
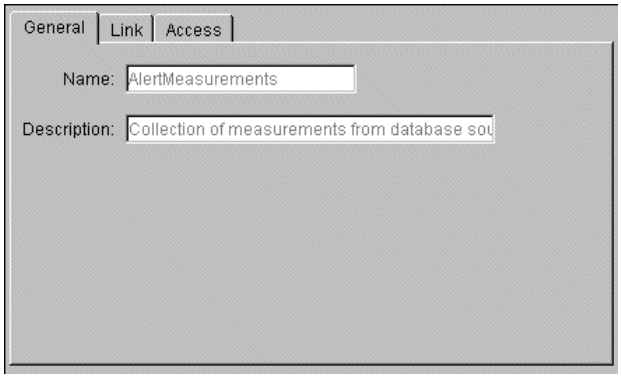
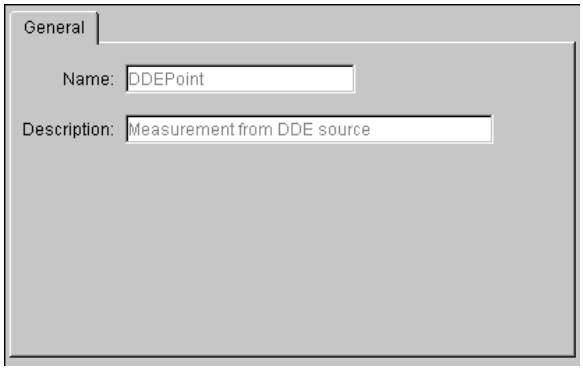


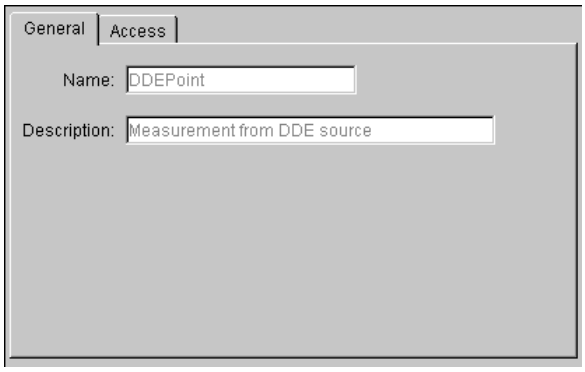
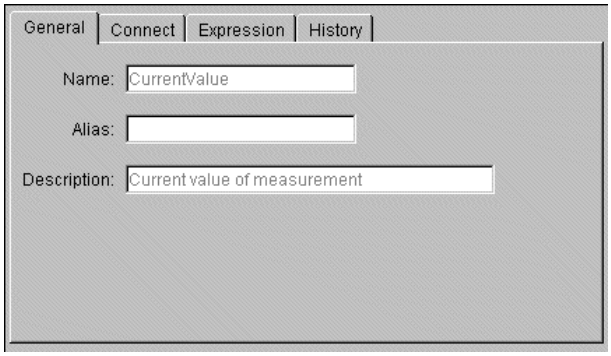
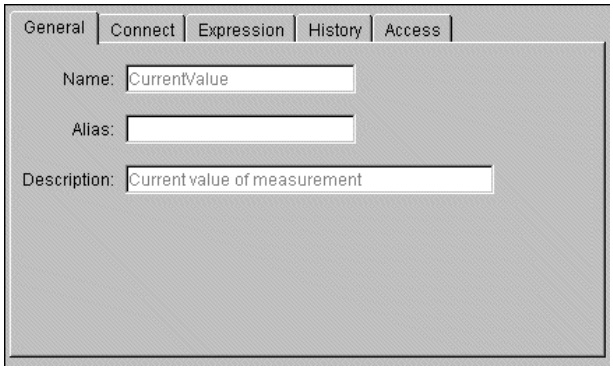
Detail Tabs

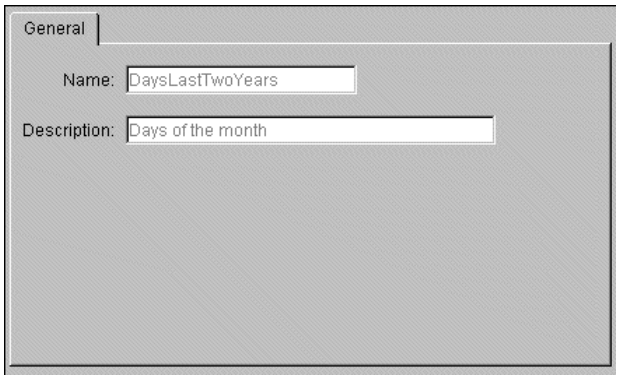
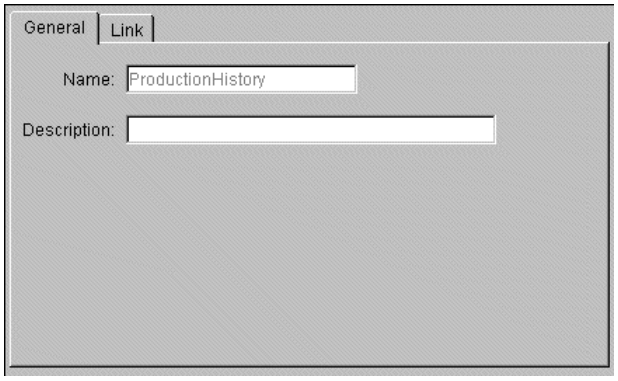
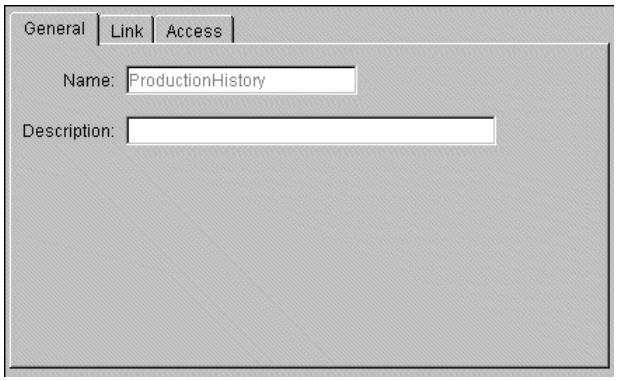
Generally, to view the properties of a selected component instance, you select the parent component in the Navigation panel and then choose the desired component from the Inventory Items panel. By selecting any member in the Inventory Items panel you see the properties for that member displayed in the Edit panel.

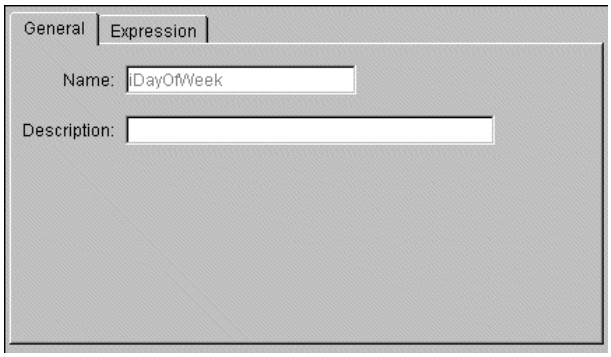
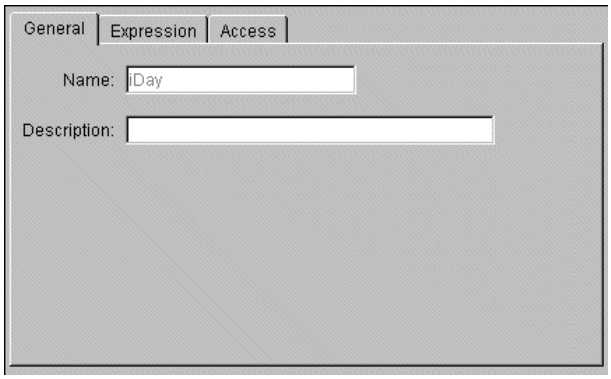
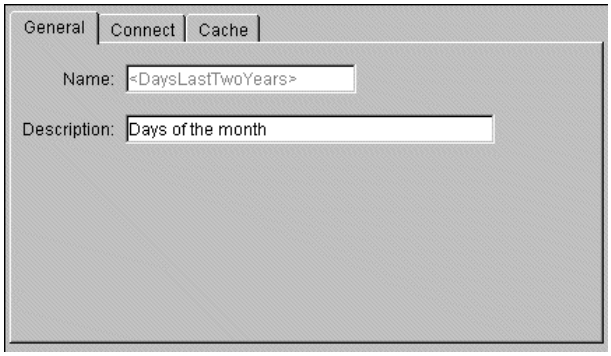
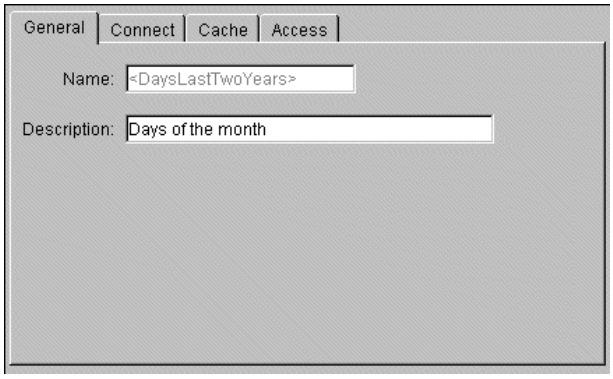
Depending on the item type and whether or not security is enabled, the Detail View offers the following tabs.

Tab Display

If you select this type	This tab will display
Component	**General, Link [Access]**  With Security Enabled: 
Component Member	**General [Access]**  With Security Enabled:

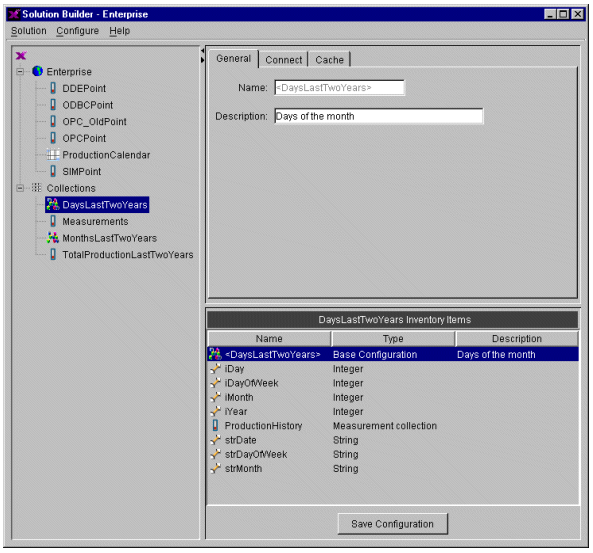
If you select this type	This tab will display
	
Primitive Member	**General (with alias), Connect, Expression, History [Access]**  With Security Enabled: 
Collection	**General**

If you select this type	This tab will display
	
Collection Member	**General, Link [Access]**  With Security Enabled: 
Collection Primitive Member	**General, Expression [Access]**

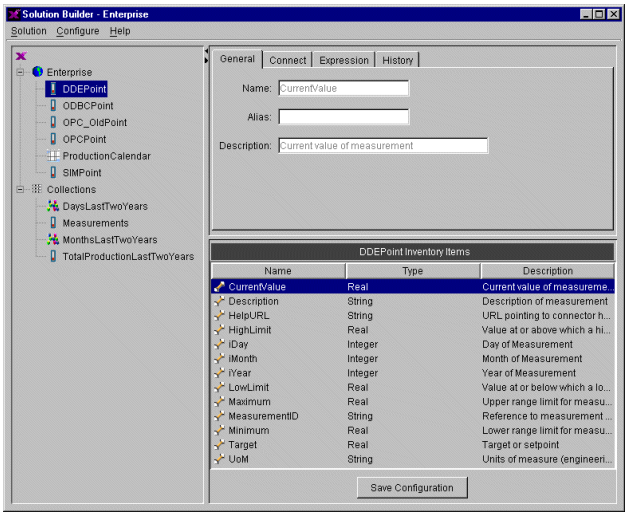
If you select this type	This tab will display
	 With Security Enabled: 
Collection Base Configuration	**General, Connect, Cache [Access]**  With Security Enabled: 

The General Tab

The General tab displays the name and description of the selected member. You provide the name and description when you add components in the Component Details tab.



General Tab Example for Base Configuration



General Tab Example for Primitive Member

Using Aliases

If you select a primitive member, note that the Alias field appears on the General tab of the Edit panel. You use the Alias field to assign another globally unique name to the primitive member. The alias name provides a way for the system to access that primitive member. Unlike member names, which can be the same in different components, an alias is a unique "global" name for a solution; it can only be used once.

Alias names require single quotes (either forward or backward quote).

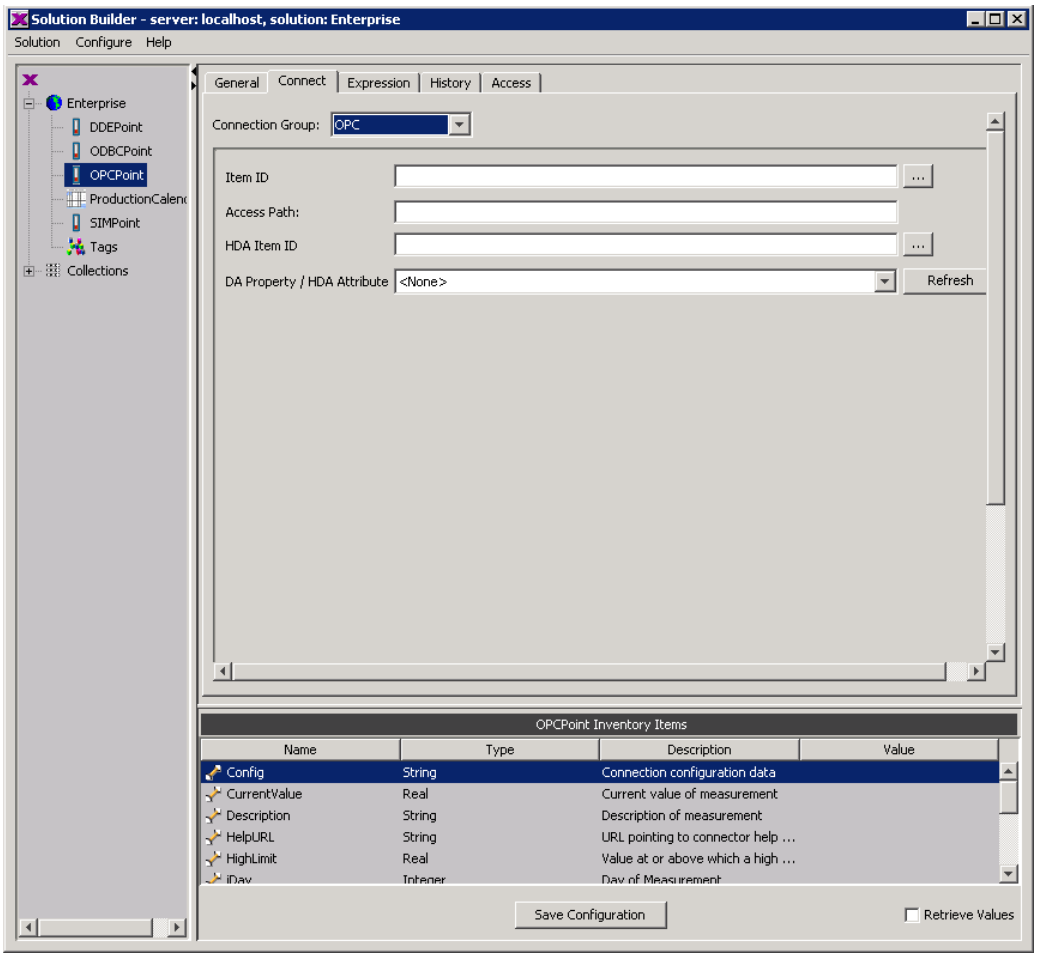
To assign an alias name to a primitive member

1. From the navigation panel of the **XHQ Solution Builder**, **select the node** which contains the primitive member to which you want to assign an alias name.
   The General tab appears in the Edit panel.
2. From the **Inventory Items** panel, select **a primitive member** (Boolean, Integer, Real, or String type) to which you want to assign an alias name.
3. In the **General** tab, enter the alias name you want to assign to the member in the **Alias** text box.
4. Click **Save Configuration**.

The Connect Tab

The Connect tab shows information for the connections made to an external data source. Whereas the connector definition indicates general properties, the Connect tab makes specific assignments to physical addresses in other systems or databases.

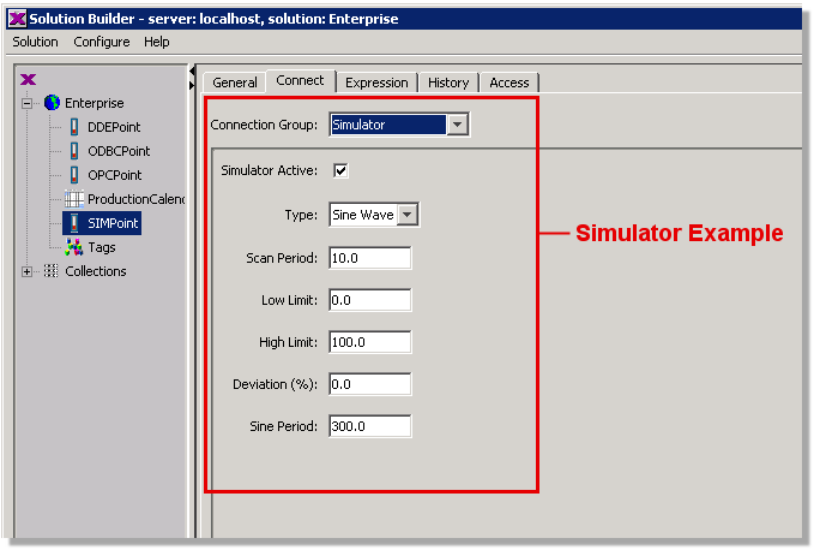
The Connect tab appears when you select either a primitive or base configuration type.



Connect Tab, Primitive Type example

Using the Simulator Connection Group

XHQ includes a built-in simulator that generates data according to parameters you specify. You can use the Simulator to test the operation of components that will normally collect data from external systems, or set up prototypes that can simulate real-life applications.



Connect Tab, Simulator

Simulator Controls

Item	Action
Simulator Active	Select to engage the Simulator.
Type: Count	Generates data in even increments between the low and high limits.
Type: Random	Randomly generates data between the low and high limits specified.
Type: Sine Wave	Generates data between the low and high limits following a sine wave path.
Scan Period (sec)	Defines the time between updates, specified in seconds. You can also specify milliseconds in decimal form. For example, if you enter 0.5, the scan period is 500 milliseconds, or one-half second.
Low Limit	Designates the minimum value generated by the Simulator.
High Limit	Designates the maximum value generated by the Simulator.
Deviation (%)	Defines the percent amount of change between the values generated.
Sine Period (ms)	Defines the length of the sine wave, specified in milliseconds. Applies to Sine Wave type only.

Selecting the Query Type for Data Member Configuration

The JDBC-based connectors (Oracle, ODBC, SQL, PostgreSQL) and the internal XHQCache connector allow you to set the following query type options:

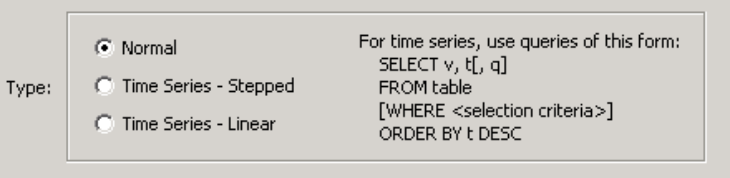
- **Normal**
  This is a normal query, not a time-series query. The member value is set to the first value returned by the query. And unless the historian configuration is made against this member through the History tab, the member is not trendable in any meaningful way.

  The Normal query type is for a point for which history has been configured; the Time Series query type is used when history is not configured.

- **Time Series – Stepped**
  This is a time-series query. The relational database being queried is used as a historian. Without making a history configuration on the member, it is still possible to get a trend against that member. The values between any two actual points in the trend are interpolated to be the same as the value of the earlier actual point. The last value is retained until a new value is received. This abrupt change to the new value creates the stepped appearance.

- **Time Series – Linear**
  This is a time-series query. The relational database being queried is used as a historian. Without making a history configuration on the member, it is still possible to get a trend against that member. The values between any two actual points in the trend are interpolated to fall on the straight line connecting the points.

  Selecting Stepped versus Linear affects how the endpoints are interpolated for various time aggregation functions.



Example: Query type options for JDBC-based and XHQCache connectors

Things to Note

- Again, the query types mentioned above are only relevant for the JDBC-based connectors (Oracle, ODBC, SQL, PostgreSQL) and the internal XHQCache connector.
- OPC, PHD, PI, IP.21, and XHQ connectors acquire the interpolated type values directly from the backend.
- For the SIMATIC IT Historian and the XHQ Data Recorder, the interpolation type options (stepped or linear) are available from the History tab. (In this case, the Normal query does not apply.)
- For new data member configurations using either a JDBC-based connector or the XHQCache connector, the **Normal** query type is selected by default. For pre-existing time-series configurations, the **Time Series – Linear** query type is selected by default for Real, Double, and Decimal types, and **Time Series – Stepped** for all other data types.
- In addition to visualization, the interpolation type (stepped, linear) may also affect how functions are calculated. For example, functions such as at(), first(), and last() may be different for the stepped or linear interpolation type.

  For more information on functions and expressions, go to the *XHQ Reference Guide*.

About Time-Series

The time-series feature allows you to access historical data stored in a relational database (RDB). Through a special point configuration available for most RDB sources, both the most recent (current) value and historical data can be retrieved from a single RDB source. The RDB source must have data that has value and time columns in a single table or view.

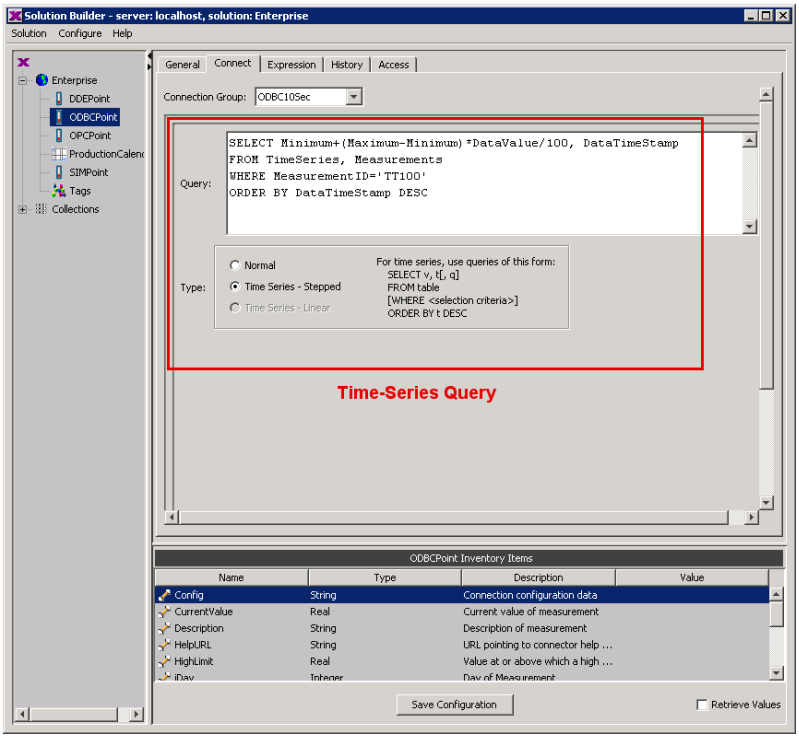
If a time-series interpolation type option is selected, the RDB connector parses the query supplied for retrieving the current value and uses it as the basis for generating the appropriate query for historical data. Historical queries are generated when trending, for historical data retrieval, and for time-series aggregate functions. Current value queries are executed according to the poll rate set for the given connection group.

The query format is as follows. Brackets indicate that it is optional.

```
SELECT value_column, time_column[, quality_column]
FROM table_name
[WHERE some_constraint]
ORDER BY time_column DESC
```

Deviations from the suggested query format may return unpredictable results.

A **quality column** is optional. For more information, go to the topic, *About the Quality Column*.



XHQ Solution Builder, Time-Series Query

The ORDER BY time_column DESC is used in the query since the most recent value is assumed to be the first record in the result set. If you do not specify ORDER BY time_column DESC, there is no way to know which record of the query's result set will be first. For historical data queries, the ORDER BY clause specified in the current value query is not used and is replaced by a generated ORDER BY time_column clause.

When historical data is needed, a query is generated for a given time span and sent to the data source. The query made to the RDB source expects ACTUALS for the source's historical data. If BEST FITS historical data is requested, the connector converts the actual data returned in the result set from the RDB. Trending in XHQ requests BEST FITS historical data.

About the Quality Column

As seen in the following query format, the quality column in brackets indicates that it is **optional**.

```
SELECT value_column, time_column[, quality_column]
FROM table_name
[WHERE some_constraint]
ORDER BY time_column DESC
```

Quality values from the RDB are expected as GOOD, BAD, or UNCERTAIN in string format. If any other values for quality are received, the quality will be returned as UNCERTAIN. If quality is not part of your time-series query, all quality will be reported as GOOD.

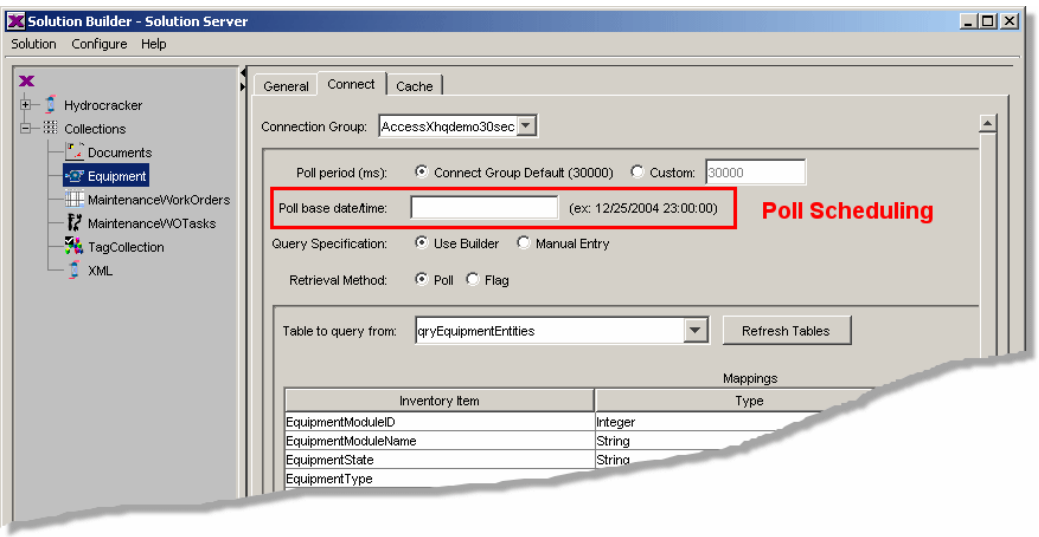
The following quality values have corresponding **numeric** values.

Quality numeric value
BAD = 0
BAD_CONFIG_ERROR = 4
BAD_NOT_CONNECTED = 8
BAD_DEVICE_FAILURE = 12
BAD_SENSOR_FAILURE = 16
BAD_LAST_KNOWN_VALUE = 20
BAD_COMM_FAILURE = 24
BAD_OUT_OF_SERVICE = 28
BAD_NOT_SUPPORTED = 32
BAD_CONNECTOR_NOT_RUNNING = 44
UNCERTAIN = 64
GOOD = 192

About the Poll Scheduling Option

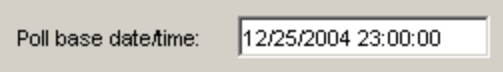
Use the poll schedule option for connections using selected connectors that allows the time of day of the poll to be controlled. It is currently available for collection connections using the MSSQL, ODBC, Oracle, SAP3, or XML connectors.

To set the poll schedule, select a collection and then click the **Connect tab**.



Poll Scheduling Option

In the text box labeled **Poll base date/time**, enter the poll schedule using the following format. The date/time you specify must be in the past.



Poll Schedule Format

If the base date/time field is left blank, connection poll scheduling is done as before (execute a poll on startup and at every period thereafter).

Poll Schedule – Blank

If a base date/time is entered, then connection polls occur at connector process start-up and then at multiples of the period from the specified base date and time.

For example, with a period of 8 hours and a poll schedule of 4/9/2005 8:00:00, polls would occur at 8AM, 4PM, and 12AM each day. With a period of 48 hours and the same base time, polls would occur at 8AM every other day. With a period of 7 days, polls would occur at 8AM each Monday (because 4/9/2005 is a Monday).

If the poll period is 2 hours or longer, and divides evenly into 24 hours or is a multiple of 24 hours, then scheduling is automatically adjusted for Daylight Saving Time (DST) changes. This means that for poll periods that result in easily predictable poll times (like the examples above), the poll occurs at the same local clock times in the summer and winter. Practically speaking, the above rule adjusts for DST with poll periods of 2, 3, 4, 6, 8, 12, or 24 hours, or some multiple of 24 hours.

Things to Note

- The base date/time must be set on individual collection connections. There is no option to set it at the connection group level.
- To prevent the DST adjustment, set the poll period to a value that is slightly different than the desired value. For example, set a custom poll period to 86400001 ms instead of 86400000 for a one day period.
- Avoid scheduling DST-adjusted polls such that they occur during the "ambiguous times" caused by setting the clock back (1AM to 2AM in most of the US). For example, a poll scheduled to run at 1:30AM may run twice on the "fall back" date, since 1:30AM occurs twice on that day.
- DST adjustment forces the specified poll period to be lengthened or shortened on dates when the DST changes occur. The lengthened poll on the fall DST change date may cause collections with poll periods of 2 or 3 hours to expire to UNCERTAIN quality for a short time.

About Retrieval Methods

There are four types of Retrieval Methods available.

- **Poll**
  Executes a query from the entire table at the given poll rate.
- **Flag**
  Executes the given query each time a value changes. Enter the given query in the **Refresh** text box that appears below the WHERE clause text box.

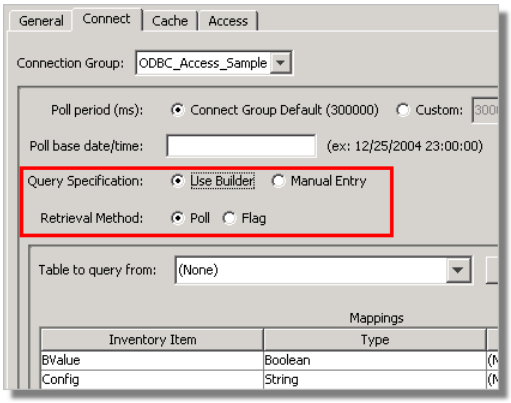
  The flag query is also referred to as the refresh query.

- **Delta**
  Provides fast and efficient updates for collections with new records.

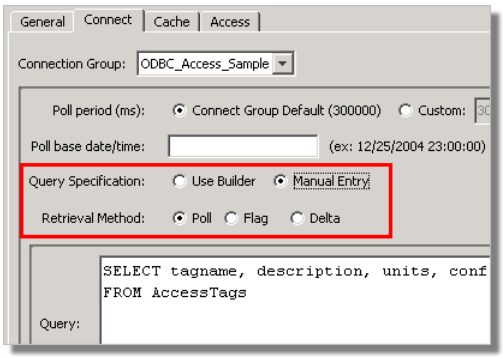
  The Delta retrieval method is available for JDBC-based connections (Oracle, ODBC, SQL, PostgreSQL) using Manual Entry queries.

- **Pass Through**
  Passes queries to the backend data source and results are delivered to the user; content is ***not cached*** within the XHQ solution server.

  Pass-through queries are not available for the ODBC connector.



ODBC Connection Group > Use Builder



ODBC Connection Group > Manual Entry

About the Flag (Refresh) Query

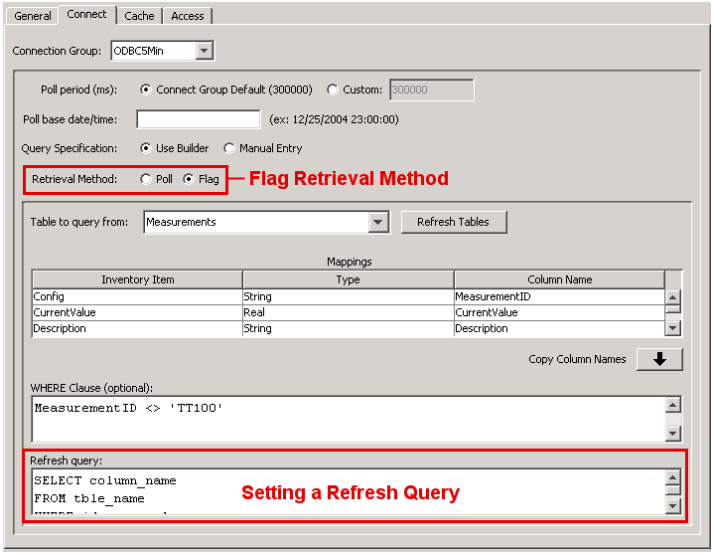
Used for the Oracle, the ODBC, the MSSQL, and the JDBC connectors, the refresh query is similar to a point configuration query in that a single column is evaluated. The column is read as a string and, depending on its value, will result in the following actions.

Result Set Actions

Result Set	Resulting Action
Null or **no rows** returned.	No refresh occurs.
First time a row is returned.	The value of the first column is saved and a refresh is performed using the base query.
Subsequent times a row is returned.	If the value of the first column is different from the previously saved value, then the new value is saved and a refresh is performed.

Again, the flag query is also referred to as the refresh query.

A *refresh* causes the base query to execute one time, which updates the base table in the XHQ cache. You can configure the base query by using the Query Builder or by manually entering your SQL statement.



Example: Setting a Refresh Query

To set a refresh query, select the **Flag** Retrieval Method to execute the given query each time a value changes. Then enter the given query in the **Refresh** text box that appears below the WHERE clause text box.

Important Things to Note

At the specified poll interval a flag query is executed. The flag query returns data for a single column only. Because it could be polled often, keep the query simple. The results of the query is then compared with the previous result.

If the resulting value (there is a record in the column) exists and it does not match the previous value, then the main query is executed and the typical full-update of the collection is processed.

The main query provides actual collection data. If there is more than one (1) record flag query, it only checks the first one. For predictable results, it is recommended that you sort data.

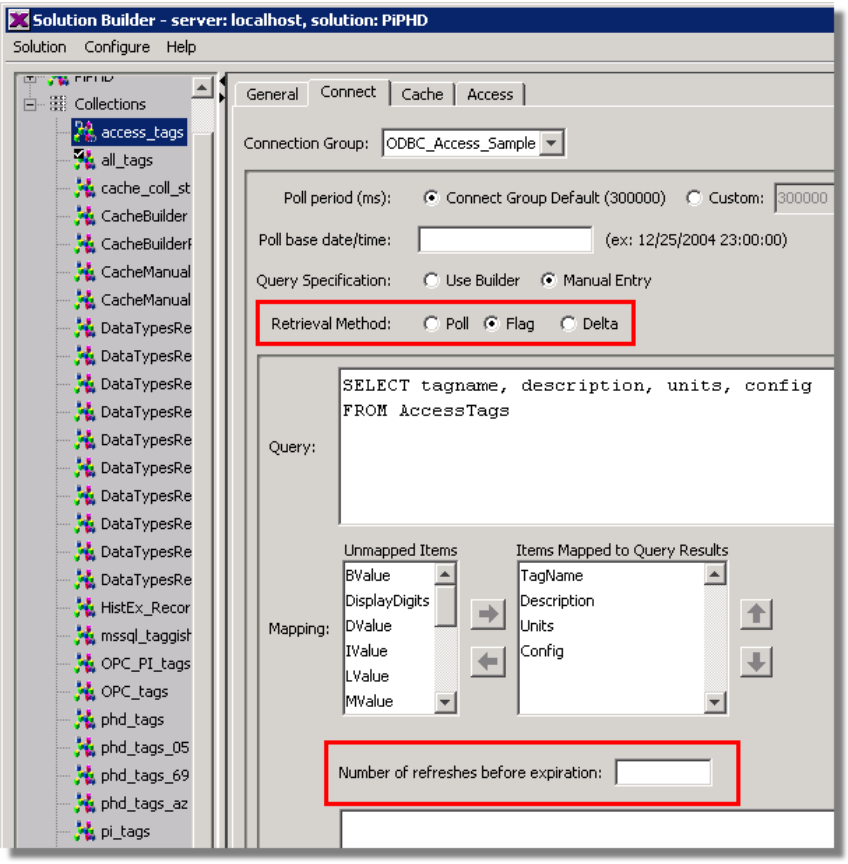
If the result is empty (there are no records in the specified column) or the resulting value matches the previous one, then the connector refreshes the collection to prevent it from expiring.

The resulting value is not persisted; therefore, a connector restart will always trigger the main query.

Refreshes before Collection Expiration

Currently, the poll rate determines the rate for a flag (refresh) query. This period is also the expiration period for the collection. As a result, the connector is forced to refresh at this rate, or the collection expires.

To override this, set the **Number of refresh before expiration** parameter, which specifies the number of poll intervals before the collection expires. In this case, the refresh request is sent to the server only once, for each number of flag queries.



Setting the "Number of refresh before expiration" parameter for a Flag Query

EXAMPLE: SETTING THE REFRESH PARAMETER FOR A COLLECTION

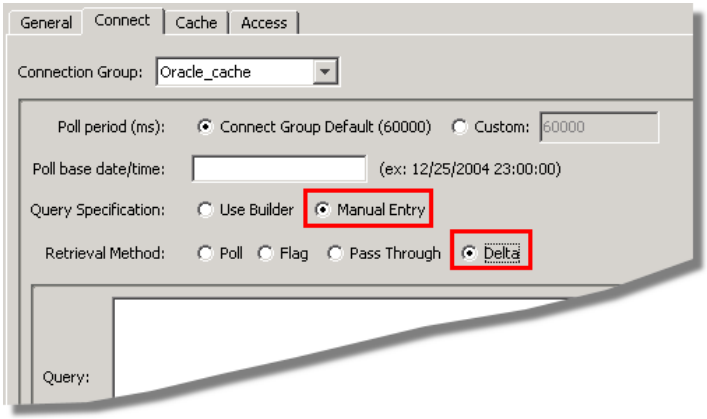
For a given collection, the flag poll rate is set to 10 seconds. The **Number of refreshes before expiration** parameter is set to 360.

In this case, the connector sends only one refresh request for every 360 flag queries. The collection then expires after 10 (seconds) multiplied by 360 (queries), which is 3600 seconds (or 1 hour).

This scenario assumes that there is no new data. As soon as the full update occurs, the counter restarts from the beginning.

About Delta Updates

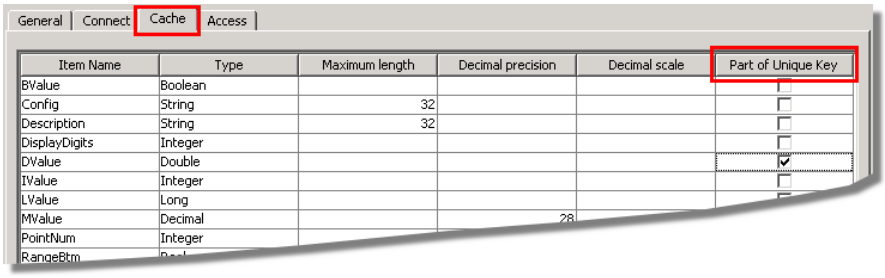
The Delta retrieval method is available for JDBC-based connections using **Manual Entry queries**. It provides fast and efficient updates for collections with new records.



Delta Retrieval Method

Important Things to Note

- The Delta Retrieval type requires that a **unique key** be defined from the **Cache tab** of the global collection.



- The delta update is not synchronized with the main poll period. Rather, the delta update runs at its intervals (which are an offset) and skips if the main update is scheduled to run at about the same time.
- The **collection timestamp** does not get updated with every delta interval. If the delta query does not find any changes, then an update is not sent to the XHQ Solution Server. Because the expiration interval for a delta collection is equal to the main poll period, there is no need for a refresh message if the delta has no changes. However, if the delta determines that there are changes, then the collection timestamp is updated.
- The time of the variables changes when **Delta Query** or **Delta delete query** returns non-empty results. If both queries return empty results, the time is not updated.

When using the Delta option, the following need to be defined:

- **Delta Period**
  The connector polls the database for new updates at this interval. The main poll period is used only for full collection updates and is set to very large intervals, for example in days or weeks. The Delta period, however, can be set to run frequently since it polls during new updates. In other words, it can be set more frequently because results can be as little as no records to only a few records (compared to the full update, which may retrieve millions of records). The Delta period is not based on the "Poll base date/time." Rather, it is based on when the process is started. The "Poll base date/time" is only relative to the poll period.

- **Delta Field**
  A member of the XHQ collection that is used to determine which collection column is used as the filter element for the Delta request.

The Delta Field can be mapped only to an Integer, a Long, or a Date member of the collection. In general, it is mapped to a column in the backend table that uniquely defines a set of records, such as a record date or a record ID. This information is used to filter new records from the records that were retrieved during previous queries.

- **Delta Query**
  This is the query used during the Delta poll requests. The main query is used only for a full request during server start-up or at full query poll intervals. The Delta query is similar to the main query, using the same order of elements in the `SELECT` statement. However, it also defines a Delta filter in the query so only the newest rows are selected.

  Example: `WHERE ID > {<XHQ_LAST_DELTA>}`

  This WHERE clause, `{<XHQ_LAST_DELTA>}`, can be used multiple times within the delta query. For example, you can include it in all the nested SELECT statements within the delta query.

  When either the main or the delta query is being processed, the connector retains the largest value of the current column (member) specified as a Delta Field. At the time of execution, the connector substitutes all the `<XHQ_LAST_DELTA>` keys in the delta query with the last saved Delta Field value.

- **Delta delete query**
  To track deletes, enter a SQL statement (if this box is left empty, deletes are not tracked).

  Scratch Table Requirements
  To use the delta delete query, a **scratch table** must first be created in the backend database. (Note, XHQ does not automatically create this scratch table nor does XHQ make modifications to it. XHQ only reads data from it.) In addition, a trigger in the database must also be created so that when records are deleted from the main table(s), it is documented in the scratch table.
  The scratch table must have a column of type **`Datetime`** (the delete date column), which indicates the time of the record deletion. It should also have columns for ALL columns used in the normal Delta new/update configuration. The delete date column and the column(s) identified as (parts of) the unique key must have actual values; all other columns can be set to null.

Important things to note about the SQL statement

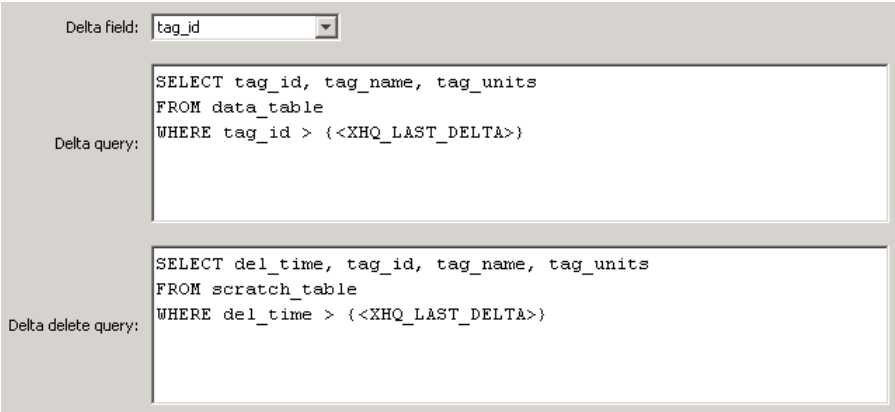
- The first item in the SELECT statement must be a delete date column.
- After the delete date column, enter the same set of items in the same order specified in the Delta query.
- The FROM clause must reference the scratch table.
- The WHERE clause references the delete date column.
- The {<XHQ_LAST_DELTA>} values for the Delta query and the Delta delete query are independent. Although they use the same syntax, the values are stored separately. The {<XHQ_LAST_DELTA>} for the **Delta query** is for filtering *new* or *changed* records in the *main* table. For the **Delta delete query**, however, {<XHQ_LAST_DELTA>} is for filtering *deleted* records in the *scratch* table.

Consider the following example.

Given: The main table is **`data_table`**. It contains the columns **`tag_id`** (which is the unique key), **`tag_name`**, and **`tag_units`**. The scratch table is **`scratch_table`**, and it contains the columns **`del_time`**, **`tag_id`**, **`tag_name`**, and **`tag_units`**.

Query:
```
SELECT tag_id, tag_name, tag_units
FROM data_table
```

The Delta retrieval is configured as follows:

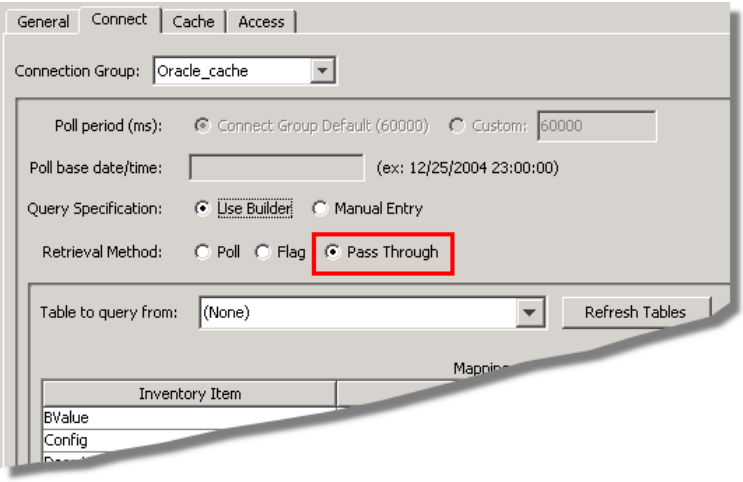


In the Delta delete query, the first item in the SELECT statement list is a delete date column (`del_time`). After this delete date column, the same set of items listed in the Delta query SELECT statement is entered, in the same order (`tag_id, tag_name, tag_units`). Note the difference of the WHERE clause in both the Delta query and the Delta delete query. The Delta delete query references the delete date column (`del_time`).

In this case, only the del_time and tag_id (the unique key) columns must have data. The other columns may have null values.

About Pass-Through Queries

In a pass-through query, requests are "passed" to the backend data source and results are delivered to the user; content is ***not cached*** within the XHQ Solution Server. Pass-through queries allow you to set a query against very large tables without the need to store large amounts of data. This retrieval method is an option from the Connect tab of the XHQ Solution Builder. Pass-through queries are only available for the Oracle, SQL, SAP3 and Documentum backend data sources; it is not available for ODBC.



Pass-through Query Retrieval Method

WARNING
Please note, every client that views collection data from a pass-through query will generate a separate query to the backend data source. This may cause performance issues. In addition, pass-through queries are directly executed by the backend system in response to user actions. If the query is complex or is executed often, this may place a heavy load on the backend system.

Additional things to note about pass-through queries

- Do not use column aliases in the SELECT list of the statement you enter in the global collection Connect tab. They are not supported.
- For SQL, the innermost query (configured in the global collection) can contain anything that is valid for the backend database, including JOINs, UNIONs, OUTER JOINs, function calls.
- To minimize the load on the backend, use pass-through queries for infrequently accessed views that need to draw a relatively small selection of records from a potentially large set of relational data.
- If the records must be presented in a particular order, use Sort(s) in the view.

Consider the following example.

EXAMPLE: WHEN TO USE A PASS-THROUGH QUERY

You need to provide users with the ability to view work order detail information for the last year. This record set typically consists of several hundred thousand large records containing all of the details associated with each work order. This content is only accessed on occasion, when users drill through to a specific record for more details.

In this case, it would be inefficient to maintain this large, infrequently accessed data set within the XHQ cache. Instead, a pass-through query that executes when the user drills through to details is a more efficient option.

- Ad hoc constraints are added to the pass-through query and are not applied by the client.
- The number of records returned is limited by the number of records that the client is able to display. Therefore, if a client that can only display 20 rows makes a pass-through query against a million-row table, then only 20 rows are returned.
- Rollups and aggregate calculations on pass-through queries are currently not supported.
- SAP3 supports Ad Hoc but not link constraints.
- Documentum queries do not use a nested `SELECT`.
- The XHQ Solution Viewer is idle while it waits for the pass-through request to finish.
- A complex query against a large table may take longer than 30 to 40 seconds to return, depending on the responsiveness of the backend.

For SQL Pass-Through Queries

The following information in this section is specifically for the SQL pass-through.

The query specified in the global collection connection is executed to produce some result set. The linked collection constraint is applied to that result set to produce another result set and then finally, the client view filters and sorts are applied to that to produce the final result set. This is actually done by forming a single nested query that is passed to the backend SQL engine, which optimizes it however it sees fit.

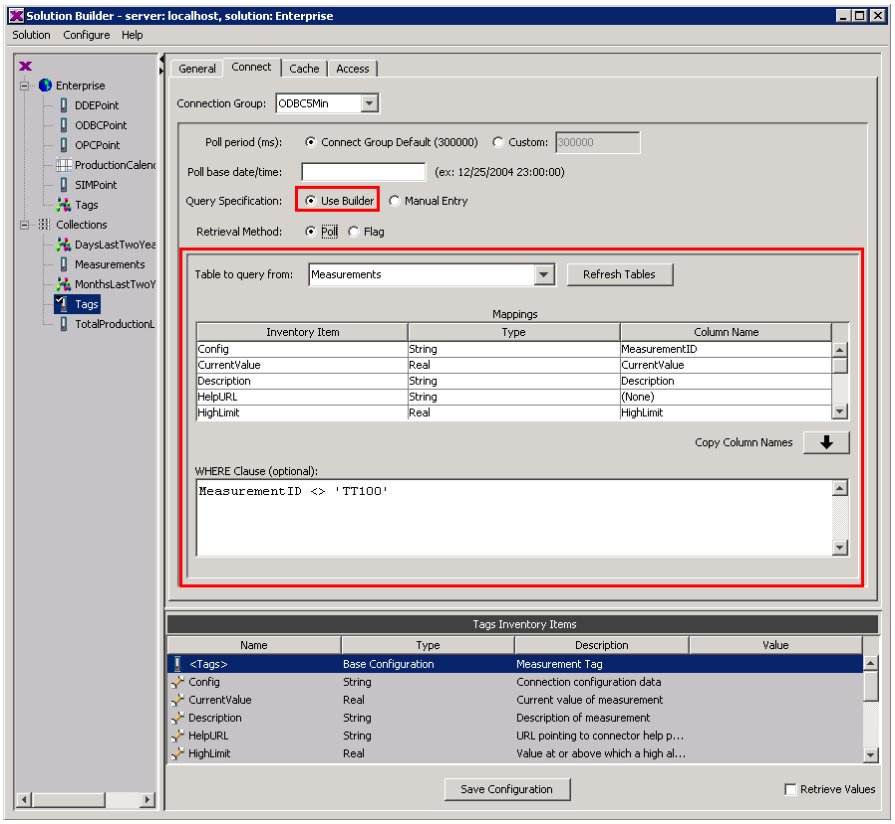
The following is an example of a query passed to the backend:

```
SELECT view columns FROM
  (
    SELECT view columns FROM
      (
        SELECT whatever FROM backend tables WHERE whatever
      )
    WHERE linked collection constraints
  )
WHERE view specified constraints
ORDER BY view specified sorts
```

Again, this example works on any backend that supports this type (SQL-92) of nested queries; Oracle and SQL do, but ODBC does not (or at least, ODBC to Microsoft Access does not). Additional things to note, SAP3 supports Ad Hoc but not link constraints and Documentum queries are done without a nested select.

Using the Query Builder

The query builder provides a simple interface for configuring **simple** queries against ODBC or Oracle sources. It supports any query that directly uses the columns of a single table or view. You access the query builder by selecting a collection inventory item and clicking on the Connect tab.

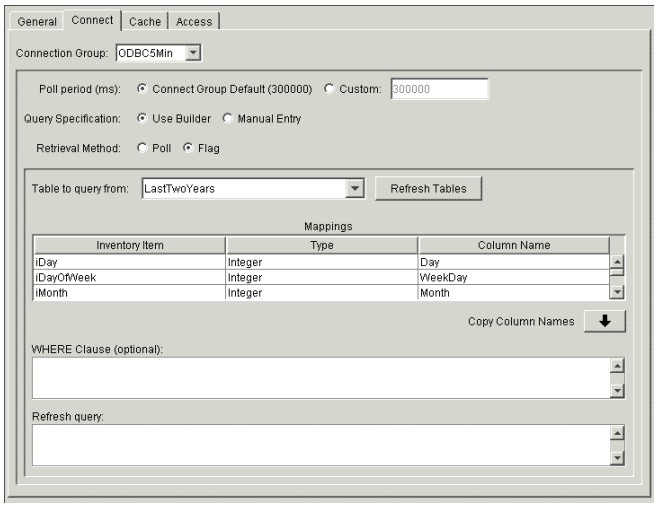


Query Builder Interface

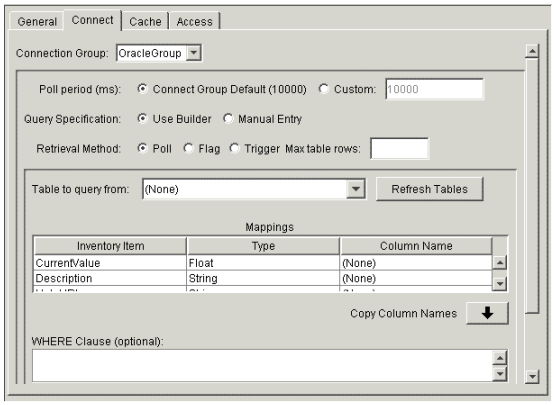
The Query Builder works on Oracle 8.0 and above. For queries against Oracle 7.3.4, select **Manual Entry** to manually enter complete SQL statements.

To use the query builder

1. From the **XHQ Solution Builder**, select the collection.
2. Click the **Connect** tab.
3. From the **Connection Group** list, select an ODBC or an Oracle connection group.
   A list of Inventory Items appear in Mappings.



Connect Tab - ODBC Example



Connect Tab - Oracle Example

4. For **Poll Period**, do one of the following:
   - Click **Connect Group Default** to use the poll period set from the Connection Group dialog box. The default value appears in parenthesis.

   or

   - Click **Custom** to set a poll period (in milliseconds) that is different from the connection group default.
5. Under **Query Specification**, do one of the following:
   - For users of Oracle 8.0 and above, you can either **Use Builder** or select **Manual Entry** for advanced queries.

   or

   - For users of Oracle 7.3.4, select **Manual Entry** to manually enter your SQL statements.

6. For **Retrieval Method**, do one of the following:
   - Click **Poll** to execute a query from the entire table at the set poll rate given in step 4.

   or

   - Click **Flag** to execute the given query each time a value changes. Enter the given query in the **Refresh** text box that appears below the WHERE clause text box.

   or

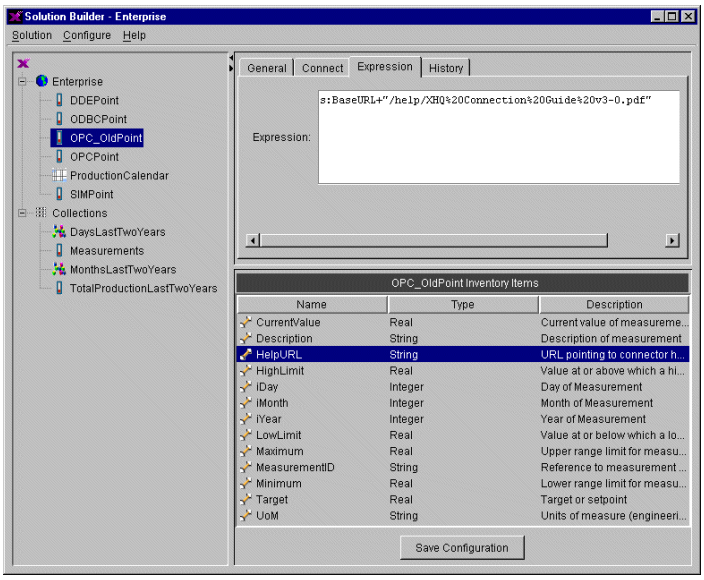
   - Click **Pass Through** to execute the given query each time a query request is made.

     Pass-through queries are not available for the ODBC connector.

7. In **Table to query from**, select the name of the table.
   This allows you to map columns from the database to members in the collection.
8. Click **Refresh Tables**.
   This allows you to receive recently updated data.
9. Under **Column Name**, click inside the field.
   A drop-down list appears, displaying the table's column names.
10. From the drop-down list, **select a column name** to associate with the inventory item.
11. Repeat as needed.
12. OPTIONAL
    Click **Copy Column Names** to automatically transfer the column names you identified onto the **WHERE Clause** text box.
13. Click **Save Configuration**.

The Expression Tab

Use the Expression tab to perform a calculation against data returned from the backend system.



Expression Tab Example, Primitive Type

About Expressions

An *expression* is a simple way to configure and extend a component using arithmetic formulas and Boolean logic. Normally, you enter the appropriate XHQ variable names when configuring a component in the XHQ Solution Builder (*server-side expression*) or "pointing to" a component in the XHQ Workbench (*client-side expression*). You can also enter an expression in place of such variables.

In every expression, there are one or more *operands*, which are the elements to be calculated, which may be separated by calculation *operators*. The simplest form of an expression consists of a single literal constant or variable. In that case, the operand is without an operator. More commonly, an expression performs a calculation or operation, such as simple addition and subtraction, or more complex operations, such as relational, bitwise, logical, and complex math operations.

XHQ calculates the expression from left to right, according to a specific order for each operator in the formula. The evaluation of an expression performs one or more operations, yielding a result.

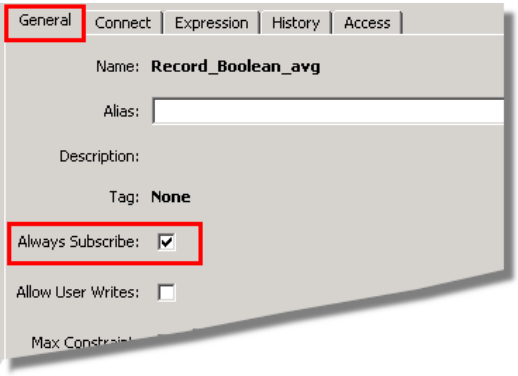
When you enter a zero (0) before a number in a server-side expression, XHQ reads it as an octal (base-8).

If you enter . . .	XHQ reads it as . . .
010	8
012	10

The *XHQ Reference Guide* explains the types of functions you may want to perform using expressions, provides examples of how you create them, and provides tables listing all the operators, or functions, and math and string functions XHQ supports.

Expressions with Primitive Members

For any expression that contains a primitive member, the primitive member must be subscribed so that the expression gets evaluated. (The subscription can be done either by scheduling the expression or by historizing the member.) In addition, the **Always Subscribe flag** (located in the General tab of the XHQ Solution Builder) needs to be enabled on the member.



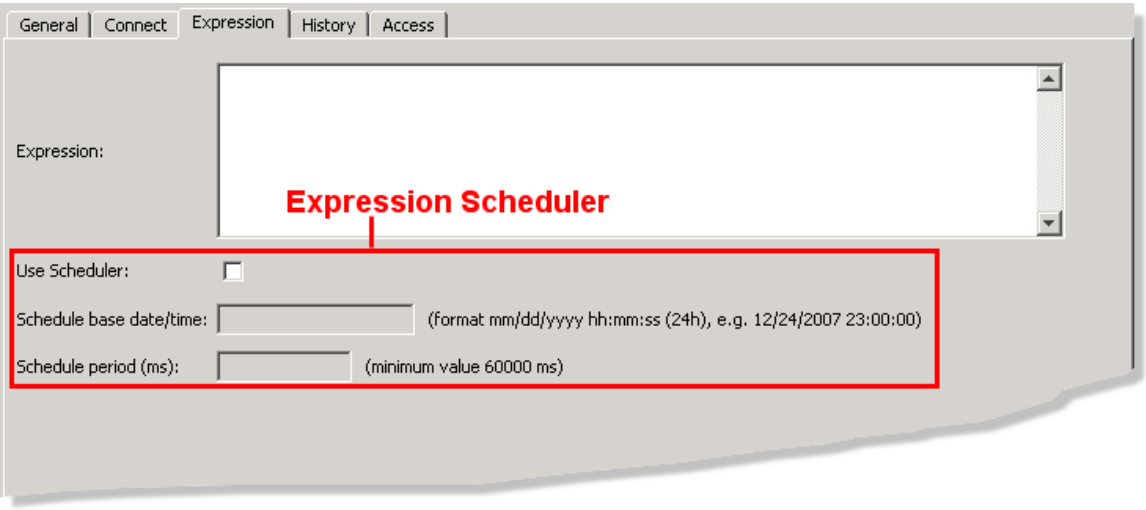
Enabling Always Subscribe in the XHQ Solution Builder

In cases where expressions need to be evaluated to retrieve and display values, subscribing the member and enabling this flag ensures that the values are always visible, even on an XHQ Server restart.

The Server-side Expression Scheduler

The Server-side Expression Scheduler provides a scheduling mechanism for any type of server-side expression. Its main purpose is the scheduling of time-depended historical aggregates.

The Scheduler has the flexibility to allow periodic updates for various time intervals, such as once per hour or once per day.



XHQ Solution Builder: Expression Scheduler

How the Scheduler Works

When the Scheduler is enabled, the expression is (re-)evaluated at every given period, starting from the specified base date and time. If the expression configuration is changed but not the schedule, the value is recalculated on the next scheduled update.

Scheduled expressions are also recalculated when XHQ is restarted and history is not presently configured. If history is configured and XHQ is restarted, then the server attempts to retrieve a value for the last scheduled evaluation time from the configured history.

Related solution setting property:
`HistoryManagement.LastHistoryValueRequestWaitTimeMins`

For more information, go to the topic, *Using solutionsettings.properties*, located in the XHQ Administrator's Guide.

From the `solutionsettings.properties` file, you may set the following property:
`SolutionServer.useLegacyScheduledExprEval`

If this property value is set to **`true`**, the Scheduler uses the current value of referenced members at the time of evaluation.

If set to **`false`**, the Scheduler attempts to use historical values based on the scheduled time. If history is not enabled for the referenced members in the expression, then the member's current value is used.

In the latter case (=false), the expression is evaluated at every given period regardless of the possibility that some members of the expression may expire sooner than the scheduled period.

The `SolutionServer.useLegacyScheduledExprEval` property is only read at **start-up**.

Possible Scenarios

Scenario	Input	Input-No Backend History	Event Expression	Scheduled Expression	History
Unconfigured					
BackendValue	X				
RawCurrentValue		X			
CurrentScaledValue	X		X		
HistorizedScaledValue	X		X		X
ConstantExpression			X		
HistorizedConstant			X		X
ScheduledExpression				X	
HistorizedSchedExpr				X	X
EventExpression			X		
HistorizedEventExpr			X		X

Referenced values come from history or , if history is not available, the current value.

Result of Scheduled Expression

Referenced Member	Best Possible Resulting Quality	Referenced Value
Unconfigured	BAD, Not Configured	N/A
BackendValue	GOOD	History Value (ACTUAL)
RawCurrentValue	BAD, ??? (Based on the backend)	History Value (ACTUAL)
CurrentScaledValue	GOOD	Current Value
HistorizedScaledValue	GOOD	History Value (ACTUAL)
ConstantExpression	GOOD	Current Value
HistorizedConstant	GOOD	History Value (ACTUAL)
ScheduledExpression	GOOD	Current Value
HistorizedSchedExpr	GOOD	History Value (ACTUAL)
EventExpression	GOOD	Current Value
HistorizedEventExpr	GOOD	History Value (ACTUAL)

About Elimination Gaps

Should gaps occur when checking scheduled expressions, the following message appears:

WARNING: Evaluation of scheduled expressions had 3 gaps during the last 20 seconds. 1 gap was filled. 2 gaps were ignored.

Consequently, because of the high load, some evaluated items have been rescheduled with gaps during the last statistic interval.

For **filled** gaps, the scheduling of items with enabled history was corrected. All items were evaluated and stored to history for each scheduled time.

For **ignored** gaps, the missing calculations of items that shouldn't be written into the history (the "visual" points) were omitted losslessly.

Adjusting for Daylight Saving Time

If the expression schedule period is 2 hours or longer, and divides evenly into 24 hours or is a multiple of 24 hours, then scheduling is automatically adjusted for Daylight Saving Time (DST) changes. This means that for scheduled periods that result in easily predictable evaluation times (like the examples above), the expression evaluation occurs at the same local clock times in the summer and winter. Practically speaking, the above rule adjusts for DST with schedule periods of 2, 3, 4, 6, 8, 12, or 24 hours, or some multiple of 24 hours.

To prevent the DST adjustment, set the schedule period to a value that is slightly different than the desired value. For example, set the schedule period to 86400001 ms instead of 86400000 for a one-day period.

Important Things to Note about DST Adjustments

- Avoid scheduling DST-adjusted periods such that they occur during the "ambiguous times" caused by setting the clock back (1AM to 2AM in most of the U.S.). For example, an expression evaluation scheduled to run at 1:30AM

may run twice on the "fall-back" date, since 1:30AM occurs twice on that day.

- DST adjustment forces the specified schedule period to be lengthened or shortened on dates when the DST changes occur.

To access and set the Expression Scheduler

1. From the **XHQ Solution Builder**, select a component inventory item.
2. Click the **Expression tab**.
3. In the Expression box, **enter a valid expression**.
4. To activate the Scheduler, **check Use Scheduler**.

   Once the Scheduler is activated, the Schedule base date/time and Schedule period fields are automatically populated with the following default values:

   - A base date of the current day and a base time of midnight;
   - A period of 3600000 milliseconds (which is one hour).

5. For **Schedule base date/time**, enter the starting date/time for scheduling.

   When entering a base date/time, use the following format:

   `MM/dd/yyyy HH:mm:ss`

   Once the Scheduler is activated (step 4), this field cannot be empty.

6. For **Schedule period (ms)**, enter the scheduling period in milliseconds.

   The minimum period allows is 60000 milliseconds (which is one minute). Once the Scheduler is activated (step 4), this field cannot be empty.

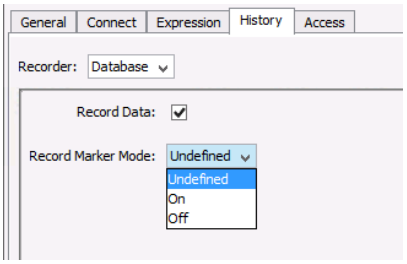
7. Click **Save Configuration**.

The History Tab

In the History tab, you configure parameters to indicate the circumstances under which you save history. You may also connect XHQ to third-party sources of time-series data for the purposes of retrieving history.

You cannot configure the storage parameters for third-party historians within the XHQ environment.

For more information, go to the topic, *Listing Third party Historians*.



History Tab: XHQ Data Recorder

To use the XHQ Data Recorder, for **Recorder**, select **Database**. The XHQ Data Recorder is the only supported data recorder with XHQ and is covered in detail in the following sections.

The XHQ Data Recorder is automatically installed with XHQ and is enabled by default. If the XHQ Data Recorder is not required, you can disable it to reduce the required physical memory footprint on the XHQ server by about 300 MB.

For details on how to disable the XHQ Data Recorder, go to the topic, *Disabling the Data Recorder*, located in the XHQ Administrator's Guide.

About the XHQ Data Recorder

The XHQ Data Recorder is a separate process similar to the XHQ Solution and Enterprise servers. It starts and stops together with the XHQ system. The XHQ Data Recorder runs under the **`xhq_dbhist.exe`** Windows process.

Things to Note

- The XHQ Data Recorder is not intended to be a process historian. It is meant to be a lightweight data recorder and a transition storage solution for integrated solution needs, especially for layered applications such as XHQ Performance Management.

  Use cases that have high throughput, high performance requirements, high update frequencies, or large data volumes must be resolved by using an external process historian.

- The XHQ Data Recorder does not provide advanced user management or advanced diagnostic capabilities.
- The Data Recorder is designed for recording hundreds or thousands of points, and should not be used for more than 10,000 points. The minimum change period supported is one minute (60000 ms), but it is expected that only a small number of historized data members are updating at this frequency. The bulk of the points being recorded update much less frequently. An example is included below.

- There is no maximum storage period enforced or throttling of the storage process for the historian. There is also no compression of the stored data.
- The database size required for a period of two years is about 28 GB and about 140 GB for a coverage period of 10 years, calculated on the basis of 3000 recorded tags.

  The XHQ Data Recorder usage, either alone or due to Performance Management, means that it requires storage space that depends on the number of records being stored. The data recorder was designed and documented to address a use case of storing a typical maximum of 3000 primitives at a variable storage rate between 1 minute and 1 week, for a period of 2 years, each record being about 42 bytes, and this was calculated to result in a maximum size of about 28 GB pure data over two years with the following usage scenario:

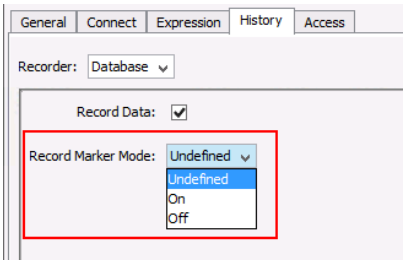
  - Total No. of primitives: 3000
  - Percentage of primitives stored every minute (%): 20
  - Percentage of primitives stored every hour (%): 50
  - Percentage of primitives stored every day (%): 0
  - Percentage of primitives stored every week (%): 10
  - Period of time for storage (in years): 2
  - Record size for one primitive data (in bytes): 42
  - Primitives stored every minute: 600
  - Primitives stored every hour: 1500
  - Primitives stored every day: 600
  - Primitives stored every week: 300
  - Total database size (in GB): 27.63

  This results in a physical database size of 2 to 3 times this data volume; so, between 50 GB and 70 GB is expected.
- The XHQ Data Recorder is not dependent on the XHQ Alert Notification System (ANS); that is, ANS does not need to be installed for the XHQ Data Recorder to work.
- In **mixed mode**, the XHQ Data Recorder is **online**.

For more information on these modes, go to the topic, *Solution Export to XHQ ANS Database Tool*, located in the XHQ ANS User's Guide.

Record Marker Mode

This marks the start/stop (or interruption) of normal recording and points out such anomalies in data flow for a **point configuration**.



Record Marker Modes

In general, the modes are:

- **Undefined**
  This point configuration does not override the specified default marker defined by the global solution setting property.
- **On**
  Marker is recorded.
- **Off**
  Marker is not recorded.

These mode are used **in conjunction with** the solution setting property, **`HistoryManagement.markerValueMode`**. The latter determines the global system setting.

A combination of the point configuration and the server properties enables you to either define all the points the same way by setting the global solution setting property, or set exceptions by configuring a specific point using the Record Marker mode. However, the ability to set the point configuration depends on the global property setting.

For example, if the global property is set to `FORCE_MARKER_VALUE`, the individual point configuration is ignored.

- **`HistoryManagement.markerValueMode = FORCE_MARKER_VALUE_ON`**
  Point configuration is ignored. The marker is always ON for all points.
- **`HistoryManagement.markerValueMode = FORCE_MARKER_VALUE_OFF`**
  Again, point configuration is ignored, The marker is alwasy OFF for all points.

To configure the marker mode for a point, the global property must be set to either of the following:

- **`HistoryManagement.markerValueMode = DEFAULT_MARKER_VALUE_ON`**
  All marker values are recorded, unless the individual point is configured otherwise.
- **`HistoryManagement.markerValueMode = DEFAULT_MARKER_VALUE_OFF`**
  All marker values are not recorded, unless the individual point is configured otherwise.

Related solution setting property: `HistoryManagement.markerValueMode`

For more information, go to the topic, *Using solutionsettings.properties*, located in the XHQ Administrator's Guide.

Storing Data Streams

Data tables are generated during installation time for a predefined number of years, which is a configurable value. The default value is 13 years (3 years in the past for data backfilling and 10 years into the future). The XHQ Data Recorder can store future values, limited only by the configured end date. The underlying database used by the XHQ Data Recorder requires no maintenance operations, other than backup.

The storing mechanism for a primitive data stream is *event-based*, so when the value of the primitive changes, the primitive's VTQ is saved into the recorder database.

Primitives with the following XHQ data types can be stored in the XHQ Data Recorder:

- Boolean
- Integer
- Long
- Real
- Double
- Decimal
- DateTime
- Interval
- String

The primitives' stored quality in the database is OPC HDA 2.0 specification compliant.

The default value of 13 years (which again is 3 years in the past for data backfilling and 10 years into the future) can be modified by running the **`hist_date_adjuster.bat`** script.

To modify the default value of 13 years

1. Open the **Windows Command Prompt** (`CMD`) as **Administrator**, run the change directory (`cd`) command:
   `cd /D "%XHQ_DB_REPOS%\..\dbadmin\XHQ\create"`
2. Call the script and provide the XHQ password.
   `hist_date_adjuster xhqpassword`
3. Enter the new start and end dates, using the given date format.
4. Click **ENTER** to continue and exit the command prompt window.
   A message appears verifying that the dates have been modified.

Retrieving Data Streams

The Data Recorder retrieves a trend containing the raw values from the database, from which it derives the **AT** and **ACTUAL** values. Similarly, the Data Recorder calculates the aggregate functions from the raw data contained in a result set generated from a database query. The Data Recorder supports the following **aggregate functions**:

- average
- minimum
- maximum
- sum
- delta
- integral
- count
- first
- last

Log Files

The Data Recorder log files are located in the same folder as all other XHQ Server log files, which is **`<XHQ_SERVER_HOME>\log`**. The log filenames are **`datarecorderlog1.out`** and **`datarecorder.out`**.

Configuration Properties

The configuration properties of the Data Recorder are stored in the **`datarec.properties`** file in the **XHQ repos folder**, which is typically located in the **`C:XHQ\data\repos`** directory. With this file, you can specify a log level for debugging the recorder. For example, using the same XHQ log syntax used in the Solution Server, you can set the following properties:

- net.indx.util.syslog.HistorianServer=1
- net.indx.util.syslog.HistoryController=3
- net.indx.util.syslog.DbHistoryAccess=4
  or
  net.indx.util.syslog.sysloglevel=4

In addition, the Data Recorder uses the following database property specified in the **`storage.properties`** file:

- `datarec.db.poolsize`
  The default value is 50, the **minimum** value is 10, and the **maximum** value is 100.

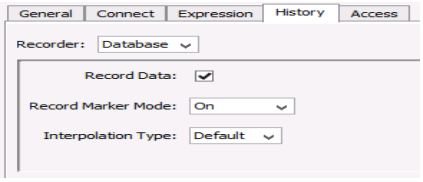
- `datarec.cache.maxTagsCount`
  The maximum count of tags that can be stored into the history cache. The value must be **greater than zero** (0) and must be **greater than** the datarec.cache.removeCount value. The default value is 10000.
- `datarec.cache.removeCount`
  The count of tags that are removed when the cache reaches its end. The value must be **greater than zero** (0) and **less than** the datarec.cache.maxTagsCount value. The default value is 100.

These properties have default values so the recorder can operate without having to explicitly specify any of the above mentioned properties and should only be specified in order to enter a value that deviates from the default value.

Using the XHQ Data Recorder

The XHQ Data Recorder stores primitive values, from the XHQ Platform Layer and the Application Layer (XHQ Performance Management), in a relational database. The XHQ Data Recorder allows you to store real-time data with minimum external maintenance and retrieve data for aggregation and time-series trending.

You access the XHQ Data Recorder through the History tab of the XHQ Solution Builder Detail View.



XHQ Solution Builder: XHQ Data Recorder

XHQ Data Recorder Controls

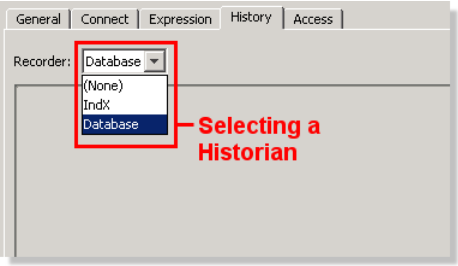
Item	Action
Record Data	Enables data collection.

To store real-time values for an XHQ primitive

1. From the **XHQ Solution Builder**, select the primitive member (for which you want to store real-time values) on the **Inventory Items list**.
2. In the **Edit panel**, click the **History tab**.
3. In the **Recorder** list box, select **Database**.
4. Then, select (check) the **Record Data** checkbox.
5. Click **Save Configuration**.

Listing Third-party Historians

The following task displays a third-party historian in the Recorder drop-down list of the History tab.



Recorder Drop-down List

There are two main steps to this task:

- Setting the `*HistoryTabEnabled` global property for the given connector.
- Adding the `-allowStore` parameter to the given connector process command-line.

Currently, the global property is available for the PI, PHD, IP.21, and OPC connectors. The **`-allowStore`** parameter enables the connector to write to the backend third-party historian.

For more information on the global properties, go to the topic, *Enabling a Historian – History Tab*, located in the XHQ Administrator's Guide.

For more information on the connection process command-line parameters, go to the topic, "How to edit the Command-line," located in the given connector's section of the XHQ Connection Guide.

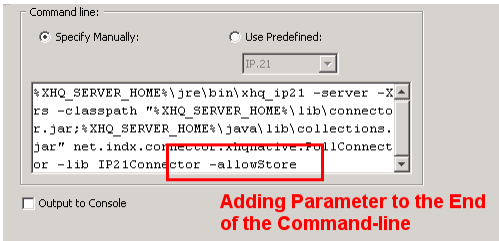
To display a third-party historian in the Recorder drop-down

1. Edit the **`globalsettings.properties`** file by adding the given history tab property and setting it to **true**. The `globalsettings.properties` file is stored at the location specified by the environment variable `%XHQ_SERVER_REPOS%` on the XHQ Enterprise Server (which is typically `<systemdrive>:\XHQ\data\repos`). Use a text editor, such as Notepad, to edit this file.

   Example: For PI, set `PIHsitoryTabEnabled = true`

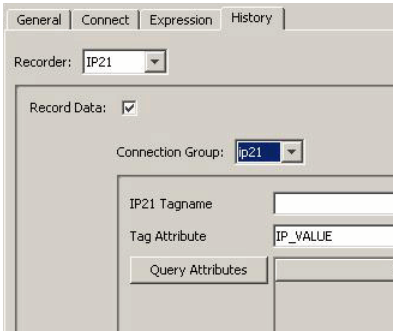
   Property names are case-sensitive.

2. **Save** the `globalsettings.properties` file and **restart** the server.
   Your next step is to edit the connection process for the given connector type/historian.
3. From the **XHQ Solution Builder**, in the **Configure** menu, click **Connection Processes**.
   The "Connection Processes" dialog box appears.
4. Select the connection process associated to the connector type/historian and click **Edit**.
5. In the **Command line** group, under **Use Predefined**, select the connector type.
   This displays the command-line for the given connector in the text box.
6. Click **Specify Manually**, and **type `-allowStore`** at the end of the command-line.

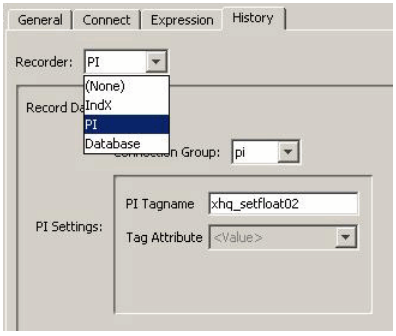


7. Click **OK** to close the "Edit Connection Process" dialog box, and acknowledge subsequent messages.
8. Click **Close [X]** to exit the "Connection Processes" dialog box.

The historian is now listed in the Recorder drop-down list of the History tab.



IP.21 Historian Example



PI Historian Example

History Tab vs Connect Tab: Displaying the Trend

If a point is configured in the **History tab**, then the historical data is taken from the recorder (connector) selected in the History tab.

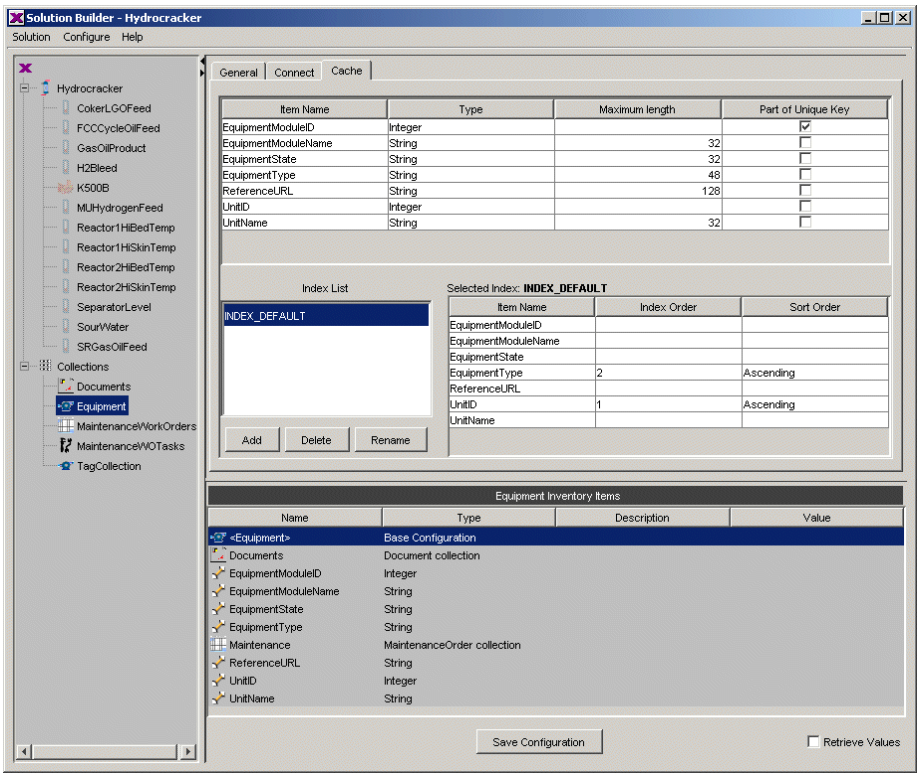
If a point is configured in the **Connect tab** and the History tab is set to `(None)`, then historical data (trends and aggregates) is taken from the connector selected in the Connect tab.

If a point is configured in the History tab and the Connect tab, then historical data is taken from the recorder selected in the History tab and the subscription of the current data is taken from the connector selected in the Connect tab.

For example, you configure history from the History tab so that all historical requests go to a given historian and set a specific tag within this historian. By chance, you happen to set a different historian from the Connect tab. In this case, a trend displays data set from the History tab, and not from the Connect tab.

The Cache Tab

The Cache tab allows you to configure index and unique key information so that the XHQ cache optimizes the data queries in real time. It consists of a table of all the members assigned to the collection you are querying. You can designate a string's maximum length, unique key, and index information.



Cache Tab Example, Base Configuration

About the Unique key

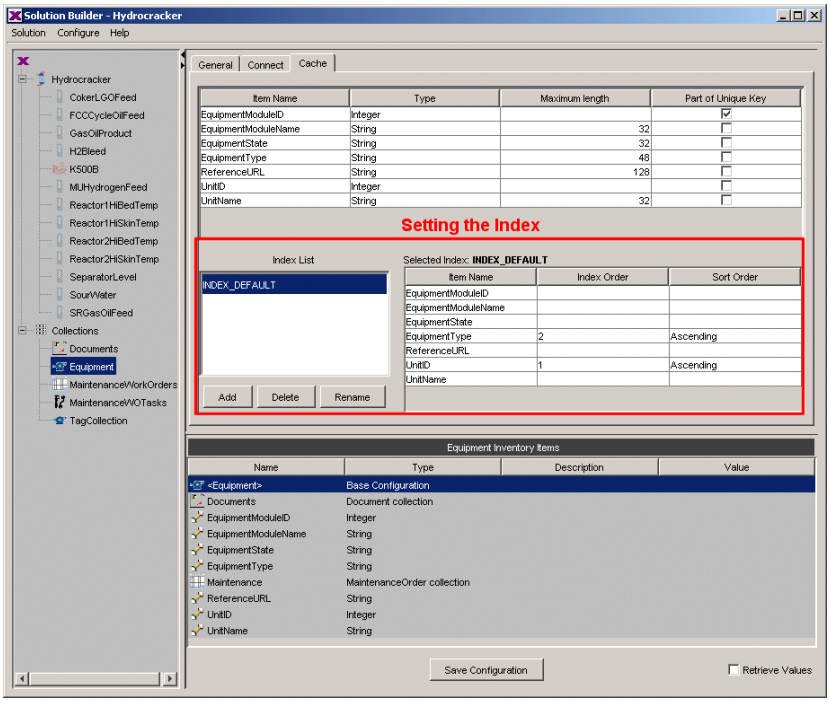
A *unique key* is used to uniquely identify a record in the collection. In XHQ, setting unique keys allows the system to compare new records with existing records, match them, and update the client (as is the case for event-based updates in collections). By assigning key fields, XHQ can identify a record that you have drilled to from a collection view. Without it, the server cannot make the association between the data you drilled to and the updated data for that same record (to the system, it "looks" just like another record). If you are using event updates to the collection, you must identify the unique key fields.

In some cases, you may have to assign *multiple* unique keys to uniquely identify a record. For example, if you had a table of people, setting just the `last_name` field alone may not be unique. You would probably have to set the `last_name`, `first_name`, and `date_of_birth` fields to uniquely identify each person. Alternatively, a *single* key identifying each person by their social security number may be sufficient.

Defining a Unique key does not mean that the Index key is defined. Both the Unique key and the Index key need to be defined in order to optimize the query.

About the Index

Designating a collection member as an *index* speeds up sorting and filtering based on that item. Setting indexes is optional and may not have a noticeable performance benefit for small record sets. With larger sets, however, having multiple indexes can improve performance.



XHQ Solution Builder, Cache Configuration: Setting the Index

The Index group box lists and defines all indexes for the current collection. Initially, if there is no index defined for the collection, the Index list will be empty.

When you select an index from the list, you can set the index and sort orders for any of the collections members. To set the **Index Order** for a collection member, click inside the appropriate cell in the "Index Order" column. Then, click on the down arrow and a drop-down list appears. This list consists of numbers from 1 to "n" (where "n" is the number of collection members). It also includes an empty (blank) field, which is the default. The empty field indicates that the collection member is not part of the index. When you select a number, you set the sort priority for the given collection members. The members with the index order of "1" get sorted first, followed by members with the index order of "2", and so forth.

Collection members cannot have the same index order. A message will appear when you try to "Save Configuration", stating that you have duplicate index order priorities.

Next, you will need to define the **Sort Order**. You can choose either "Ascending" or "Descending" order. Once you set the index order for a collection member, "Ascending" automatically appears in the next cell, under the "Sort Order" column. To switch to "Descending" order, click inside the cell and then on the down arrow.

To optimize the query, define both the Unique key and the Index key.

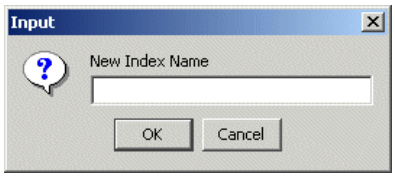
About Multiple Indexes

The use of multiple indexes gives the solution developer greater control over how collection members are sorted, and thus improves the performance of collections.

As stated above, you will not be able to revert back to the old index once you make any changes or add indexes. Therefore, if you plan to go back to an older version of the XHQ platform, make sure that you **save a copy of the old repos**.

To set the index(es) for a collection

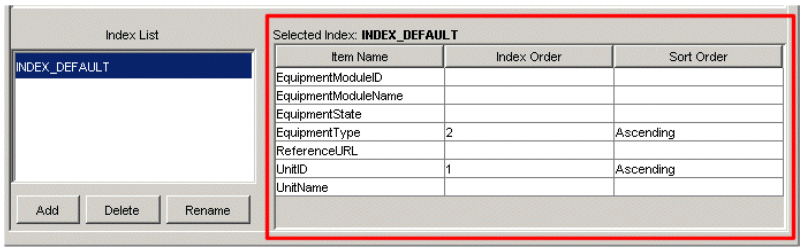
1. From the **XHQ Solution Builder**, select a **collection** and then click the **Cache** tab.
2. From the Index group box, click **Add**.
   The "Add New Index" dialog box appears.



3. Enter the **Index Name** and click **OK**.

   The index name must follow SQL 99 specifications. Therefore, spaces or special characters are not allowed. Also, it is recommended that you keep the index name to 13 characters or less.

   The new index appears in the "Index List".

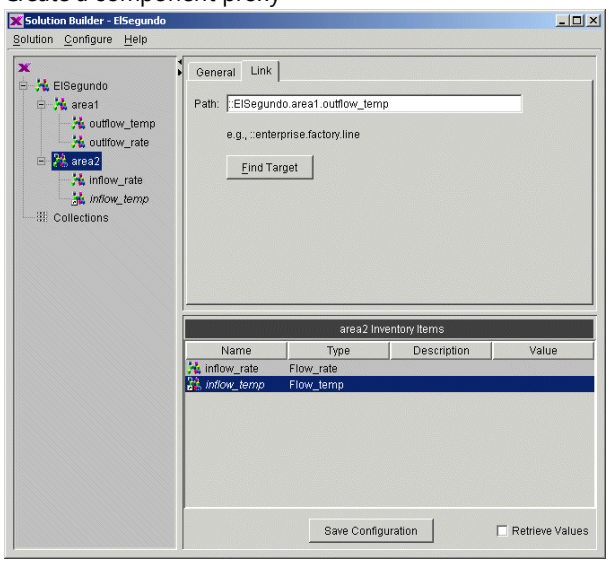


4. From the selection table, select a collection member.
5. In the **Index Order** column, click **inside** the cell next to the collection member name.
   A down arrow appears in the cell.
6. Click on the down arrow and select a **number**.
   This number defines the sort priority for the given collection member. Once you set the index order for a collection member, "Ascending" automatically appears in the next cell, under the "Sort Order" column.
7. OPTIONAL
   To change the sort order, click inside the cell, click the down arrow and select either "Ascending" or "Descending".
8. Click **Save Configuration**.

The Link Tab

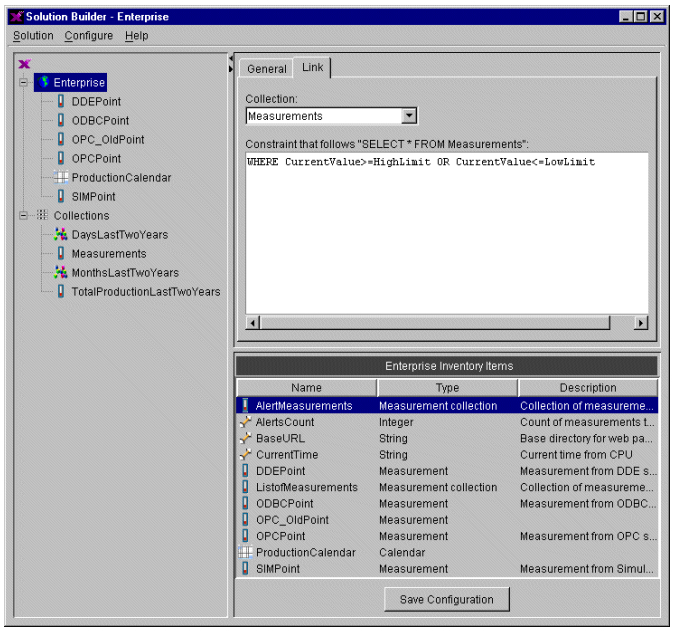
The Link tab allows you to:

- Create a component proxy



Link Tab – Creating a component proxy

- Create a new global collection



Link Tab – Creating a new global collection

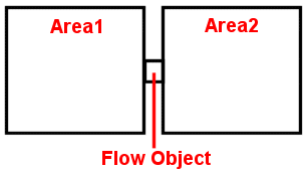
Creating a Proxy

By definition, a *proxy* in XHQ is a link to an actual, or real, component instance that acts like a "virtual" instance of the same component class as the real component instance. Proxies allow the same component instance to appear in multiple places within the solution tree. This makes configuration and editing simple since you will only have to do it from one place – the real component. Changes to the real component are automatically applied to all its proxies. Proxies are particularly useful for components that are logically shared by multiple parents.

Consider the following example.

EXAMPLE: CREATING A PROXY

In this scenario, Area1 and Area2 share a common flow object.



Example Scenario

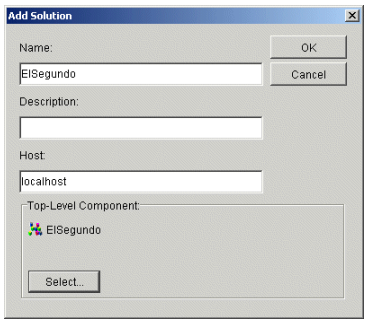
Step 1 – Create the components

Components for this Example

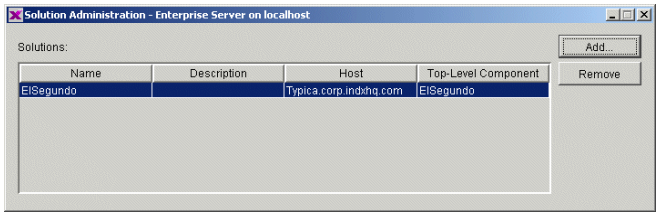
Component Member	Primitive Member Name	Type
Flow_temp	temp	Integer
Flow_rate	rate	Real
Area1	outflow_rate	Flow_rate
	outflow_temp	Flow_temp
Area2	inflow_rate	Flow_rate
	inflow_temp	Flow_temp
ElSegundo	area1	Area1
	area2	Area2

Step 2 – Deploying the Solution

1. Click **Start**, point to **Programs**, then to **XHQ Development**, then to **Administration**, and click **Solutions**.
   The "Solution Administration" dialog box appears, presenting a list of all the defined solutions.
2. Click the **Add** button.
   The "Add Solution" dialog appears.



3. Enter **`Acme1`**.
4. In the **Host box**, enter the name of the XHQ Solution Server.
   Typically, if you installed the Solution Server on the local machine, the host name is `localhost`.
5. Click **Select to choose the top-level component**.
   A list containing all components in the XHQ Workbench is displayed.
6. In the **Components list**, select **`Acme1`** and click **OK**.
7. Click **OK** to close the "Add Solution" dialog box.

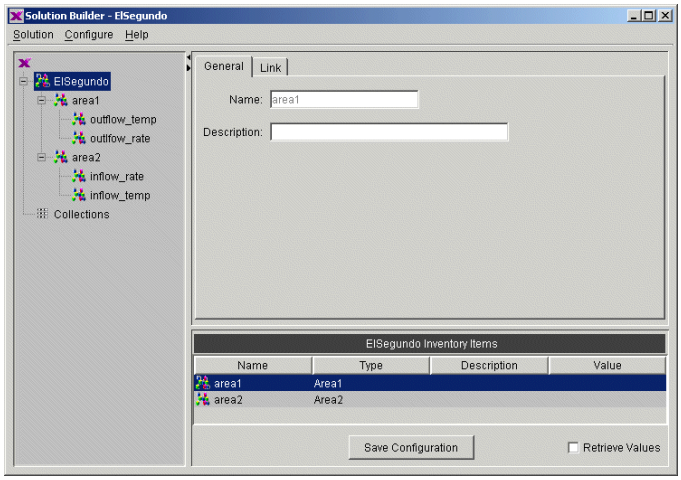


8. Click the **Close** icon [X] to exit the "Solution Administration" dialog box.
9. **Restart** the XHQ Solution Server.

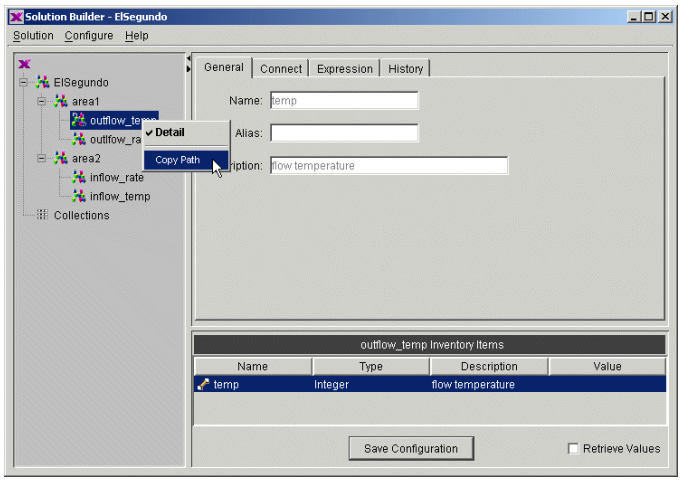
Step 3 – Creating the proxy

In this scenario, **outflow_temp** is the "real" component instance, and **inflow_temp** is the proxy.

1. Launch the XHQ Solution Builder and **expand the solution tree**.



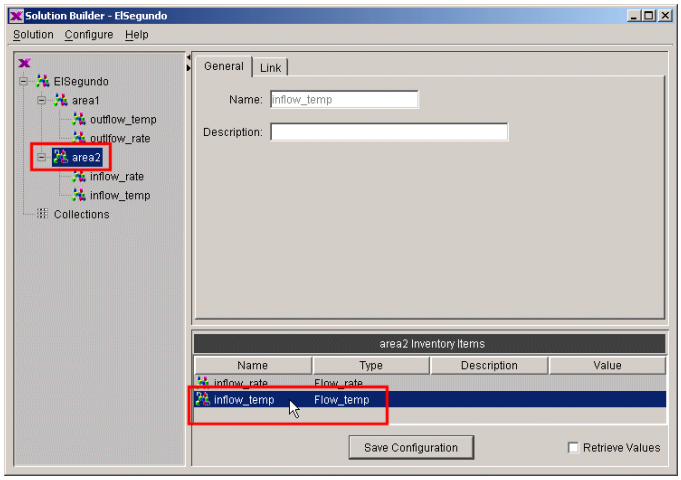
2. Right-click on **outflow_temp** (which is the real component instance).
   The right-click menu appears.



3. Click **Copy Path**.

> When assigning proxies, it is actually the parent component that contains the proxy. Therefore, the proxy is set from the parent component.

4. Select **area2** (which is the parent of the proxy).
5. From the **Inventory Items** list, select **inflow_temp** (which is the proxy for outflow_temp).



6. Click the **Link tab** and in the **Path** text box, **paste the path** you copied in step 3.

In this case, the path is `::Acme1.area1.outflow_temp`

7. From the solution tree and the **Inventory Items list**, note how the proxy name is displayed in italics. In addition, the icon includes a small arrow.

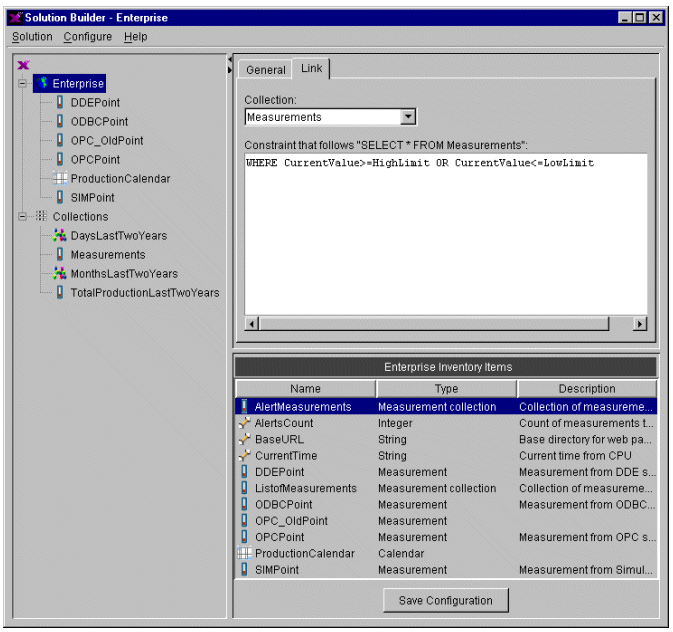
> During runtime in the XHQ Solution Viewer, there is no distinction between "real" components and proxies. Proxies are displayed like real components.

For more complex solutions, finding the real component to the proxy may be difficult, especially if the component is located under several levels. In this case, use the **Find Target** button to easily locate the real component.

8. From the solution tree, **select the parent** of the proxy and, from the Inventory Items list, **select the proxy**.
9. Click the **Link tab** and click **Find Target**.
   The solution tree automatically expands to the location of the real component.

Creating a Global Collection

The Link tab lets you select the global collection to which you want to link a "placeholder" component. It also lets you enter SQL parameters that constrain the global collection.



Link Tab Example, Component Member

Using Secured Members in the Constraint Clause

The following example demonstrates the expected behavior of a linked collection constraint using a secure member in the constraint clause.

Example:

You create a global collection and grant access to one of the columns (ColumnA) to Solution Admins only. You create a linked collection with the following link constraint:

```
where ColumnA like '%name%'
```

Expected Behavior:

Although ColumnA is secured, this does not impact the constraint (nor make it invalid) since it is intended to be set-up by users with the permission to do so. The values of ColumnA are secured for viewing.

For more information, go to the section, *Working with Collections and Flow Views*.

To create a new global collection from the Link tab

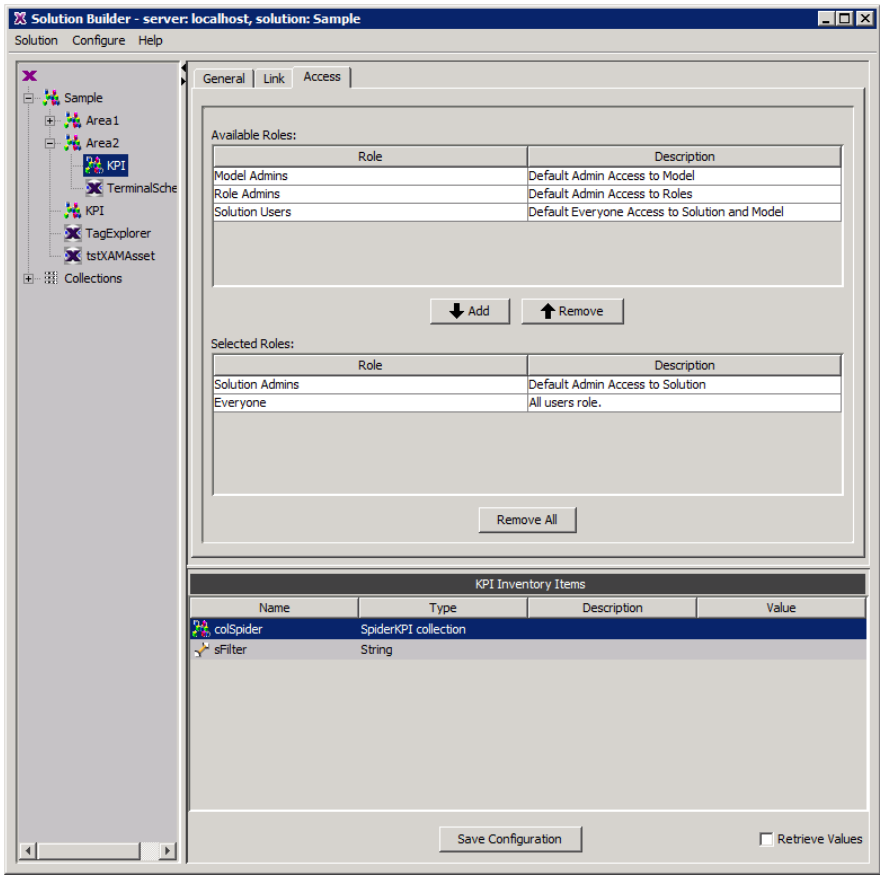
1. From the **XHQ Solution Builder**, **select** a collection member.
2. Click the **Link** tab.
3. From the **Collection** drop-down list, click **New Collection**.
   The "New Collection" dialog box appears.
4. **Enter a name** and click **OK**.
   The new collection appears under the Collections node.

The Access Tab

The Access tab allows the **Solution Admin** to restrict access to the following elements:

- Individual components
- Primitive members
- Link collections
- Global collections
- Global collection members

For information, go to the topic, *Configuring End-user Access*, located in the XHQ Administrator's Guide.

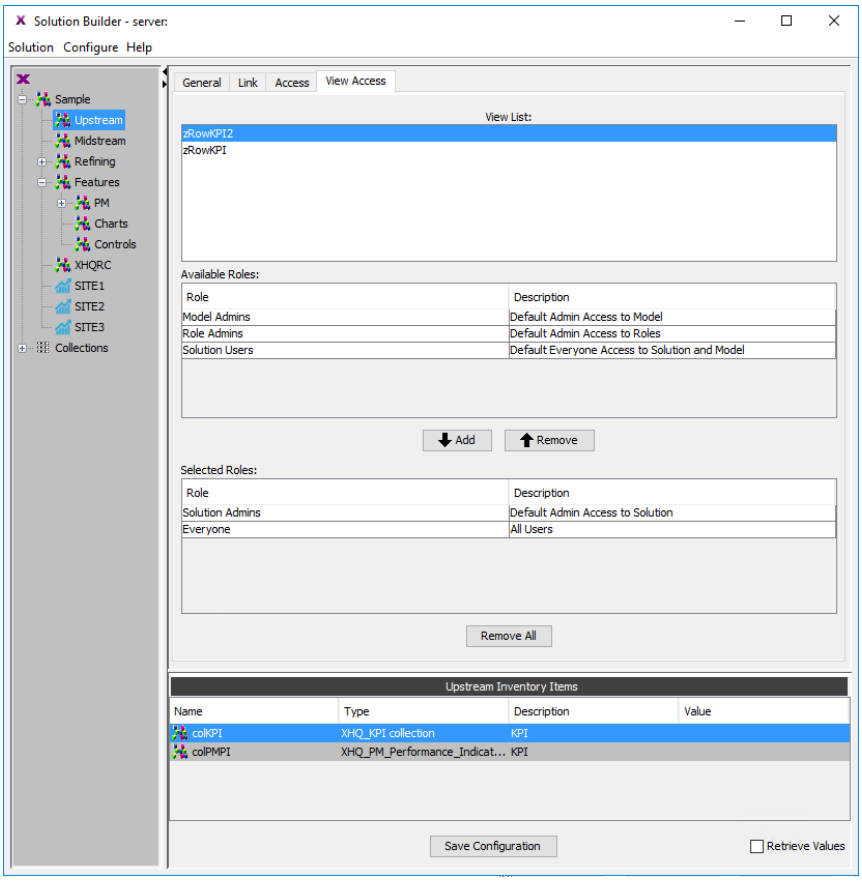


Access Tab Example

The View Access Tab

The View Access tab allows the **Solution Admin** to configure the access permission to the view by assigning roles to the given view. By default, this tab is disabled.

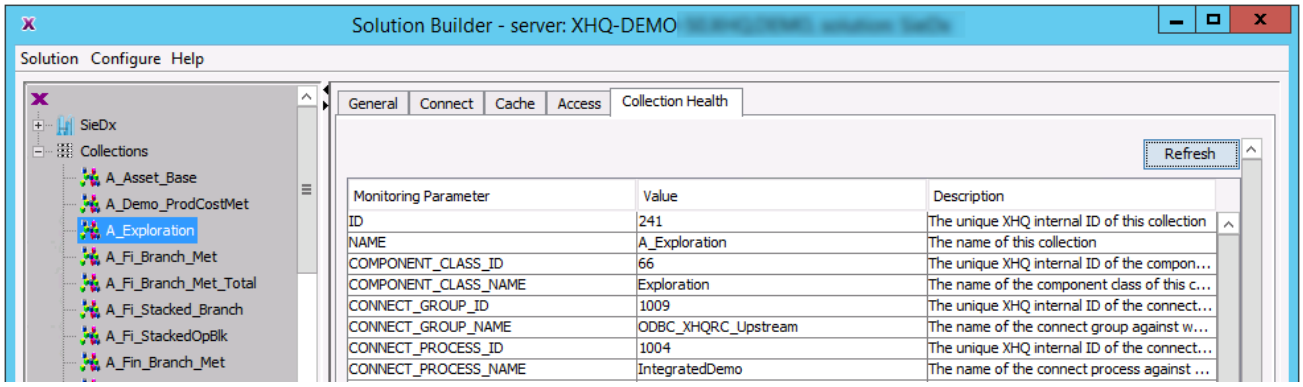
To enable the View Access tab, refer to the topic, *View Security*, located in the XHQ Administrator's Guide.



View Access Tab Example

The Collection Health Tab

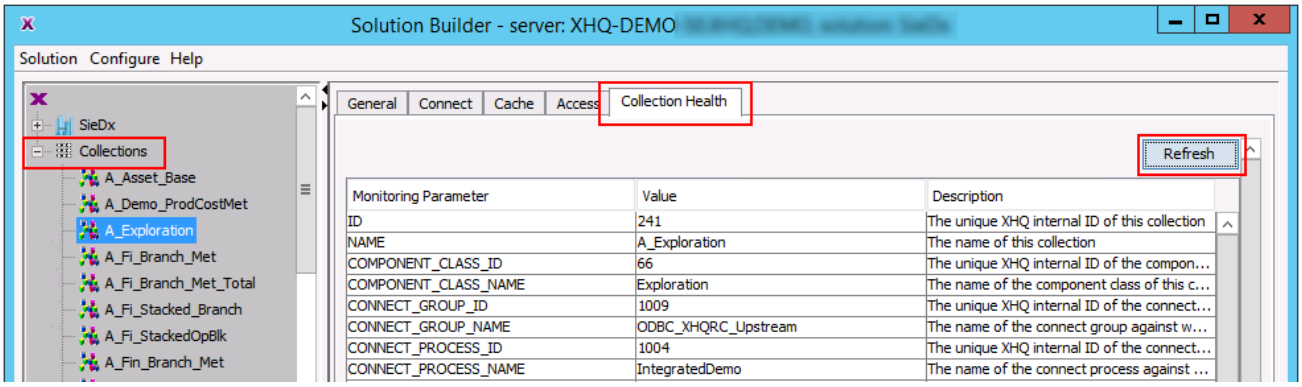
The Collection Health tab enables you to monitor and analyze global collections.



Collection Health Tab Example

To monitor the collection health

1. From the navigation panel, select a **collection**.
2. Click the **Collection Health** tab.
3. Click **Refresh**.



The following monitored parameter values are displayed, enabling you to see the current status of the global collection as a whole.

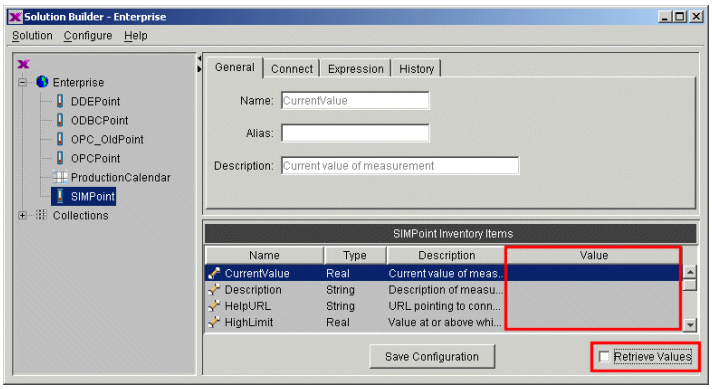
Monitoring Parameters

Parameter	Description
ID	The unique XHQ internal ID of this collection.
NAME	The name of this collection.
COMPONENT_CLASS_ID	The unique XHQ internal ID of the component class of this collection
COMPONENT_CLASS_NAME	The name of the component class of this collection.
CONNECT_GROUP_ID	The unique XHQ internal ID of the connect group against which this collection is configured.

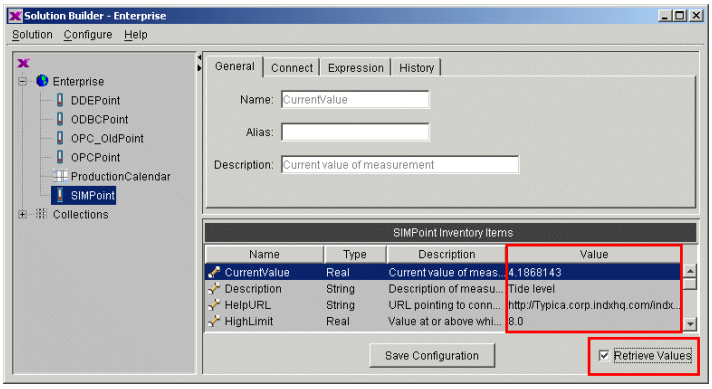
Parameter	Description
CONNECT_GROUP_NAME	The name of the connect group against which this collection is configured.
CONNECT_PROCESS_ID	The unique XHQ internal ID of the connect process against which this collection is configured.
CONNECT_PROCESS_NAME	The name of the connect process against which this collection is configured.
TAG_COLLECTION	Is this collection the tag collection? *Possible Values*: 0 - No, 1 - Yes
USES_CACHE_CONNECTOR	Is this collection configured to use the cache connector? *Possible Values*: 0 - No, 1 - Yes
EXPRESSIONS_ON_MEMBERS	Does the configuration of this collection include expressions? *Possible Values*: 0 - No, 1 - Yes
LAST_UPDATE_TIME	The last time at which this collection was updated.
LAST_UPDATE_DURATION	The amount of time it last took for this collection to update.
AVG_UPDATE_DURATION	The average amount of time it has taken for this collection to update in this server session.
LAST_UPDATE_TIME_CASCADED	The last time at which this collection and all its dependents were updated.
LAST_UPDATE_DURATION_ CASCADED	the amount of time it last took for this collection and all its dependents to update.
AVG_UPDATE_DURATION_ CASCADED	The average amount of time it took for this collection and all its dependents to update.
UPDATE_STATE	The code for the update status of this collection. *Possible Values*: 0 - "Not Finished Initializing", 1 - "Currently Being Updated", 2 - "Completed Update", 3 - "Available For Query", 4 - "Not Yet Populated"
ROWS_IN_UPDATE	The number of rows in the latest update of this collection.
EXPIRATION_TIME	The time at which this collection will expire.
QUALITY	The quality of this collection encoded as an OPC byte.
EXPIRED	Is this collection expired? *Possible Values*: 0 - No, 1 - Yes
BLOCKED_FOR_TAG_SYNC	Is the update of the tag collection blocked for a tag sync operation? *Possible Values*: 0 - No, 1 - Yes
RECONCILE_TAGS_REQUESTED	Has tag reconciliation been requested for the tag collection? *Possible Values*: 0 - No, 1 - Yes
RECONCILE_TAGS_LAST_ REQUESTED	The time at which tag reconciliation was last requested for the tag collection.
COLLECTION_INTEGRITY	The code for the integrity of this collection. *Possible Values*: 0 - "Good", 1 - "Bad Query in Query Builder", 2 - "Bad Manual Query", 3 - "Failed Insert", 4 - "Circular Reference"

Displaying Real-time Values

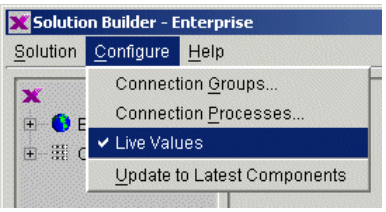
There are two ways to display live values for given inventory items: by enabling the **Retrieve Values** checkbox at the bottom of the view panel and by the **Live Values** command under the Configure menu.



Retrieve Values – Default Setting



Retrieve Values – Enabled



Live Values Command

Group Members in the XHQ Solution Builder

The XHQ Solution Builder is capable of displaying group members of components as defined in the XHQ Workbench. See *Example 1*.

For more information about Group members, see the topic, *Adding Group Members*.

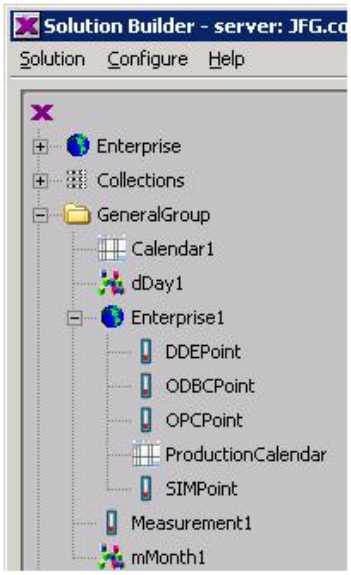
The XHQ Solution Builder can also display group members of components as defined in the XHQ Workbench. See *Example 2*.

EXAMPLE 1

Two Groups have been added to the solution at the top level: **General Group** (with no type defined) and **Measurement Group** (of the type "Measurement"). From the navigation tree of the XHQ Solution Builder, these two groups are displayed after the root node (Enterprise) and the Collections node.

XHQ Solution Builder: Navigation Tree

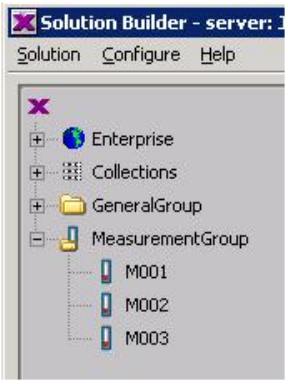
Because not type was defined for the General Group, when you expand the GeneralGroup node, you see an instance of each type of component class defined in the solution.



GeneralGroup Node Expanded

The Enterprise "root" object has a duplicate under the GeneralGroup. This is permissible with the extended namespace.

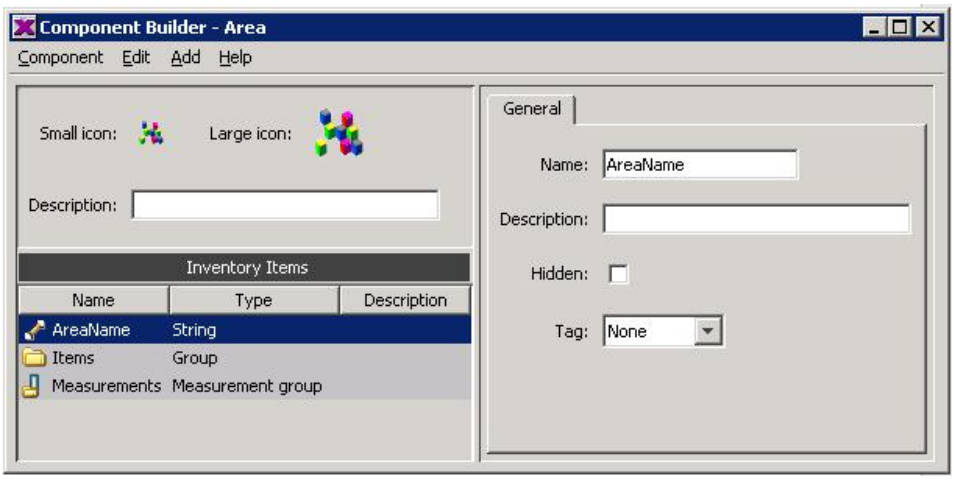
Expanding the MeasurementGroup node shows a set of Measurement-type objects. The components of other types are not allowed in this group since the type is defined to be "Measurement".



MeasurementGroup Node Expanded

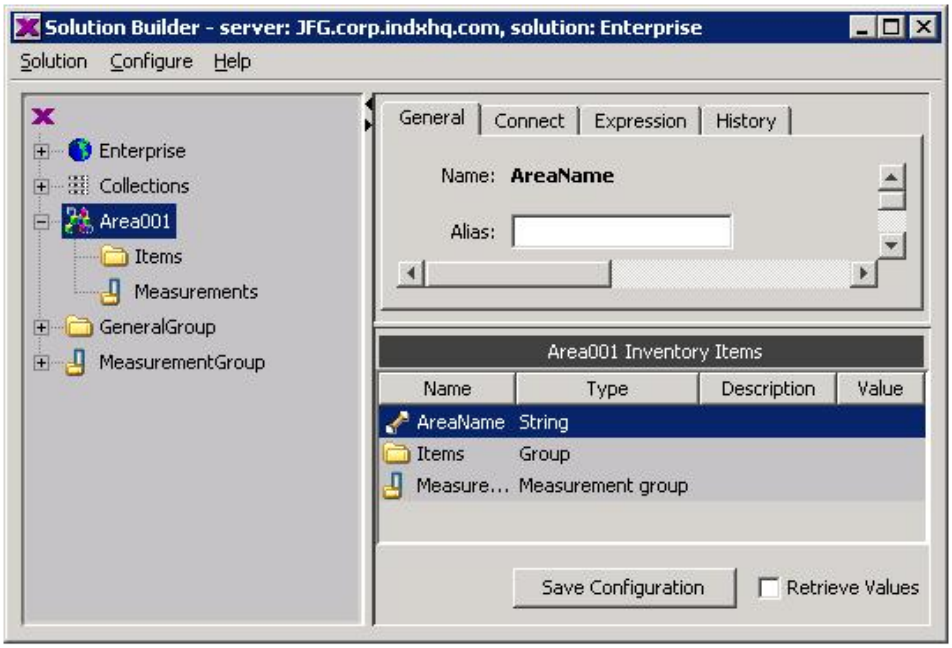
EXAMPLE 2

In this example, an instance of the Area component class is added from the Component Details tab.



Component Details tab: Area Component Class

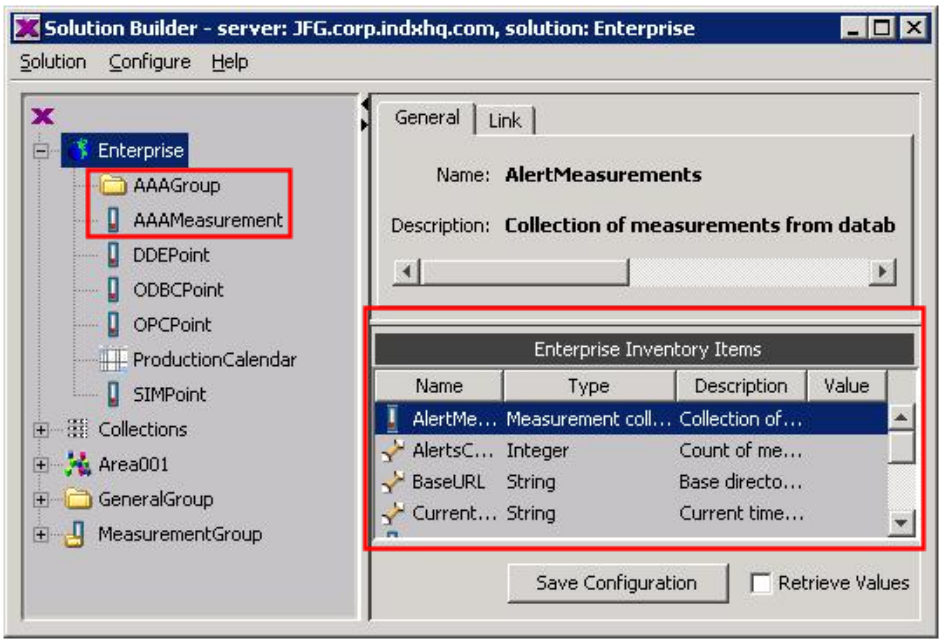
As a result, a new top-level Area component has been added to the XHQ Solution Builder.



XHQ Solution Builder: Area Node

Inventory Items and Child Objects

The XHQ Solution Builder shows modeled members (inventory items) of components in the **Inventory Items panel**. However, child objects or groups added to the solution are only shown in the navigation tree. The following example shows child objects (AAAGroup and AAAMeasurement) that have been added to the classic root node.

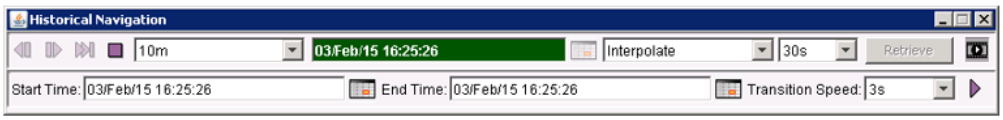


Note how the AAAGroup and AAAMeasurement items are not displayed in the Inventor Items panel.

Historical Viewing and Navigation

This XHQ feature allows you to select a historical time to apply to XHQ views and to force refresh updates at incremental time periods.

Historical mode is enabled by simply launching the Historical Navigation Toolbar. This command is located in the **Solution menu** of the **XHQ Solution Builder**.



Historical Navigation Toolbar

You can also access the Historical Navigation Toolbar from the XHQ Solution Viewer, using the applet's **`openHistoricalNavigation()`** method, or from the XHQ Navigation Bar.

When retrieving historical data, the Internet Explorer browser XHQ NavBar time matches the historical XHQ NavBar time and not the computer time. This is the expected behavior.

Once historical mode is enabled, the view automatically shows its data at the historical (or reference) time you entered. In fact, any view you navigate to will display its data in the given historical time.

In historical mode, values that are GOOD quality but are not historical data, such as collection data, will display as UNCERTAIN to indicate that the given values are not historical in nature.

For historical viewing, the following view items appear:

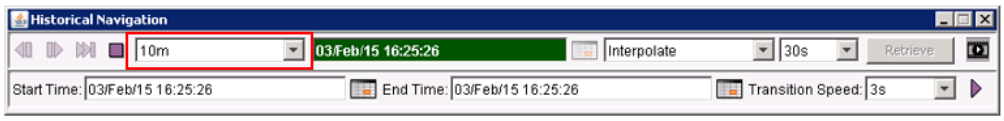
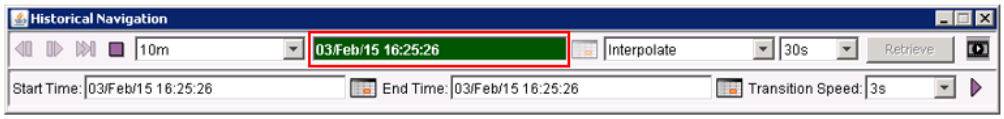
- Values and their timestamps.
- Values based on client-side expressions.
- Time-dependent functions.
  For example: `Date(),Now(),DateAdd("day", 1)`, etc.
- Historical aggregate methods.
  For example: `.avg(DateAdd("date",-1), Now())`
- Embedded trend-in-a-view.
- Historized members that are based on server-side expressions.

All other items are not affected by historical viewing and will appear in real-time.

To disable historical mode, either close the historical navigation toolbar or click the STOP toggle button (see description in the following table).

The historical navigation toolbar contains VCR-type buttons that allow you to do the following.

Historical Navigation Toolbar Button Descriptions

Control Descriptions
The **Jog back** (reverse) button moves the Select Time back according to the given jog increment.   **Note:** There is a delay time of 500ms between jog button presses before the data in view is actually retrieved. This delay prevents constant view data retrieval, which may increase network traffic.
The **Jog forward** button moves the Select Time forward according to the given jog increment.   **Note:** There is a delay time of 500ms between jog button presses before the data in view is actually retrieved. This delay prevents constant view data retrieval, which may increase network traffic.
The **Jog Now** button moves the Reference Time to the current date and time (Now).
Starts and stops Historical Viewing.   When this toggle button is in the "down" (depressed) state, STOP, then historical mode is disabled and the XHQ data within the views is shown in real-time. In this mode, all other controls in the navigation toolbar is disabled. When this button is returned to the "up" state, START, then historical viewing is again enabled and the XHQ data within the views is shown at the historical time entered.
The **Jog Increment** selector:    Allows you to either: • Select from a predetermined set of jog increment times. The default values are **10s** (seconds), **10m** (minutes), **2h** (hours), **5d** (days), **5w** (weeks). *or* • Select a given jog increment in **s** (seconds), **m** (minutes), **h** (hours), **d** (days), or **w** (weeks). You can edit the pre-determined times listed by setting the `JogIncrementList` global property. For more information, go to the topic, *Setting Historical Navigation Values*, located in the XHQ Administrator's Guide.
The **Reference Time** control:    Allows you to enter a valid reference date and time for the historical viewing. You simply replace (type over) the existing date/time with a new date/time and then press ENTER. The first time the historical navigation toolbar is launched, it displays the current date/time.

Control Descriptions

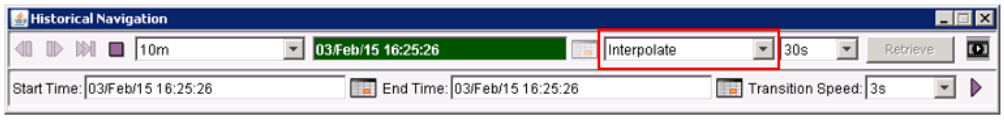
Note: If the reference time is invalid, it is reset to the current time and the historical mode is then enabled.

The **Date/Time Picker** (if available):

Click on the calendar icon to launch the Date/Time Picker.

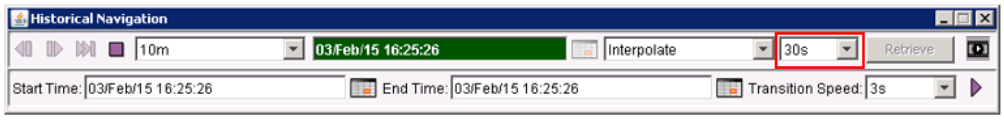
This also allows you to change the Reference Time. You have the option of manually editing the given date/time in the topmost text box or, if available, clicking on the Picker icon and using the calendar to select a specific day and/or time. You must enter a valid reference date and time.

The **Aggregation Type** selector:



Contains these options: Interpolate, Average, Min, Max, and Raw. Interpolate is the default.

The **Aggregation Interval** selector:



Is a pre-populated drop-down list with the default of 30 seconds (30s). Select from the given list or enter your own.

Note: XHQ splits the aggregation interval (ΔT) around the reference time (T) and asks the backend to send the aggregate over the time interval (T-ΔT, T+ΔT). Depending on the backend system, the timestamp associated with the aggregate value returned will vary.
For a PHD backend system, the timestamp returned with the aggregate value is the beginning of the aggregation interval (that is, T-ΔT).

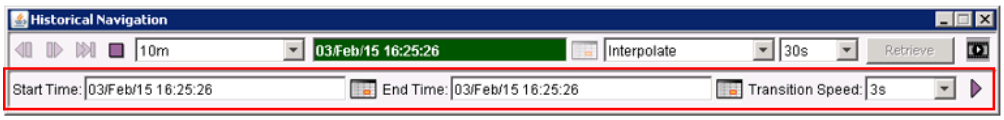
You can edit the pre-determined times listed by setting the `AggregateAverageList` global property. For more information, go to the topic, *Setting Historical Navigation Values*, located in the XHQ Administrator's Guide.

The **Toggle Historical Movie Mode** button:

When clicked, extends the Historical Navigation dialog to include the start time, end time, and transition speed of the

Control Descriptions

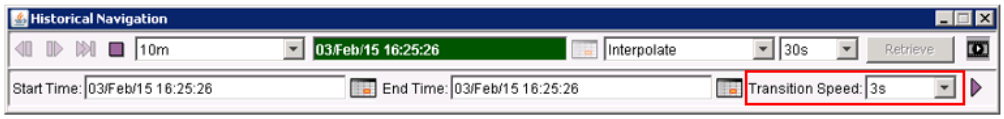
slide show.



In Movie Mode:

The **Start Time** is the time in which the Historical Viewing movie (or slide show) starts playing. The **End Time** is the time in which the slide show stops playing.

The **Transition Speed** determines the number of seconds it takes to transition from one slide to the next slide.



Select from a pre-determined set of transition times in seconds.

You can edit the pre-determined times listed by setting the `TransitionSpeedList` global property. For more information, go to the topic, *Setting Historical Navigation Values*, located in the XHQ Administrator's Guide.

The **Play** button:

When pressed, this toggles to the **Pause** button:

The Historical Navigation Toolbar is a *floating* toolbar. You can drag it to another location as well as resize the toolbar.

To resize the toolbar, move the pointer over any boundary until it changes to a double-headed arrow. Then drag the edge of the toolbar to the desired size.

For information on using the historical mode function `isInHistoricalMode(memberName)`, refer to the section,*Miscellaneous Functions*, located in the XHQ Reference Guide.

For information on using the `startHistoryNavigationSlideShow`, refer to the section,*System Functions*, which is also located in the XHQ Reference Guide.

Historical Viewing and JavaScript Functions

These JavaScript functions allow you to determine and set the current time and mode for historical viewing.

- `onHistMode(histMode, histTime, jogIncrement)`
- `setHistMode(histMode, histTime, jogIncrement)`

For more information, go to the topic, *JavaScript Functions*, located in the XHQ Reference Guide.

URL Parameters Support for Historical Viewing

The following URL parameters support historical viewing in a web page.

Example:

`http://localhost/indx/xhqnavbar.aspx?histmode=yes&jogincrement=30000&t2=1321402159000`

Historical Viewing Parameter Descriptions

Parameter	Description
histmode	**no** – historical mode **disabled** (default) **yes** – historical mode **enabled**
jogincrement	The jog increment in **seconds**. **Note:** If this is not specified, the globally set jog increment is used.
t2	The reference time, in **milliseconds**, to use for historical query. **Note:** If this is not specified, the current time is used.

12 | Exporting and Importing

Model Export and Import

XHQ allows you to export selected components from the model and import them into another model. When you *export*, you package all icons and inventory items associated with the given component. You also have the option of exporting views and embedded components. The exported content is saved as a `.zip` file.

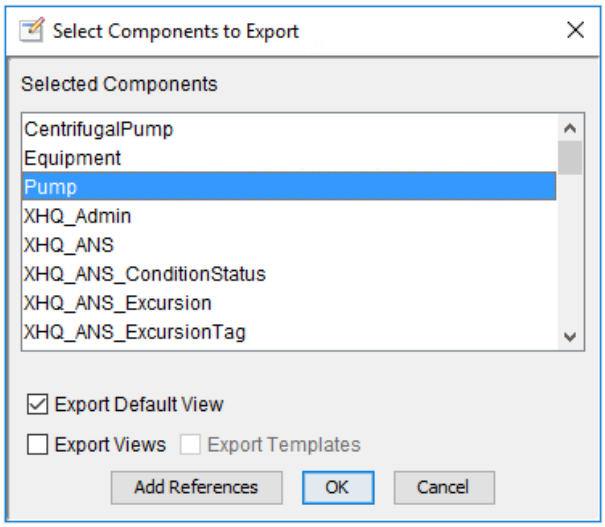
Importing involves opening an exported `.zip` and selecting the component(s) to include into your model.

Important Things to Note

- To import components from the XHQ Workbench, you must have Catalog Editing rights.
- In order to **import components**, you must have Model Importing AND Component Editing rights.
- In order to **import components with their views**, user must have Model Importing, Component Editing, AND View Editing rights.
- You can only export/import from the "All Components" page of the XHQ Workbench > Component tab. You cannot export/import from a user-defined XHQ Workbench page.

Exporting Components

You can select one or more components from the model and export to a `.zip` file. This file describes the containment structure (hierarchy) of components and their icons. As an option, the file may also contain data about related component views.



Select Components to Export Dialog Box

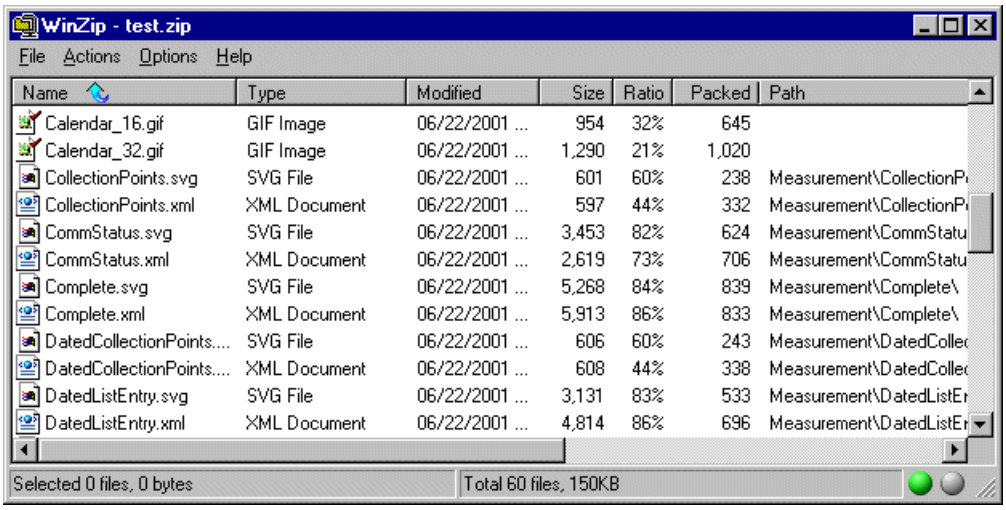
Component data that is exported includes component names, descriptions, member names, member descriptions, and member data types. View data that is exported includes bitmapped graphics, view structure, and animation data.

As mentioned, you can **Export Views** associated with the components you select, as well as templates. You also have the option to **Add References**, which allows you to include all descendents.

About the Exported ZIP File

The exported **`.zip`** file consists of the following:

- A model `.xml` file that contains the exported component structure.
- (For any view you export) An `.xml` file (data binding information) and bitmap files in `.gif` or `.jpg` format (for each bitmap image contained in the view).
- The small (16 x 16 pixels) and large (32 x 32 pixels) icons in `.gif` or `.jpg` format.



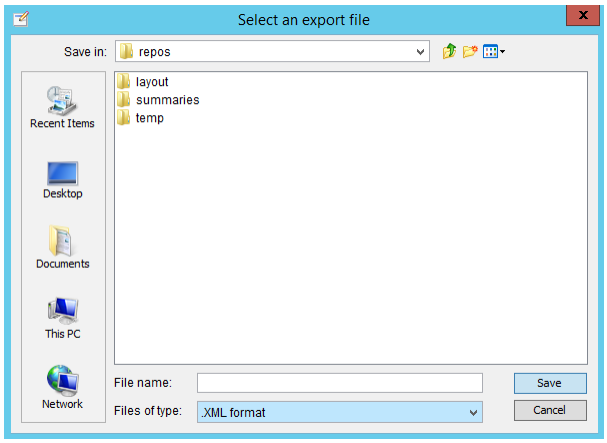
Name	Type	Modified	Size	Ratio	Packed	Path
Calendar_16.gif	GIF Image	06/22/2001 ...	954	32%	645	
Calendar_32.gif	GIF Image	06/22/2001 ...	1,290	21%	1,020	
CollectionPoints.svg	SVG File	06/22/2001 ...	601	60%	238	Measurement\CollectionP
CollectionPoints.xml	XML Document	06/22/2001 ...	597	44%	332	Measurement\CollectionP
CommStatus.svg	SVG File	06/22/2001 ...	3,453	82%	624	Measurement\CommStatu
CommStatus.xml	XML Document	06/22/2001 ...	2,619	73%	706	Measurement\CommStatu
Complete.svg	SVG File	06/22/2001 ...	5,268	84%	839	Measurement\Complete\
Complete.xml	XML Document	06/22/2001 ...	5,913	86%	833	Measurement\Complete\
DatedCollectionPoints....	SVG File	06/22/2001 ...	606	60%	243	Measurement\DatedColle
DatedCollectionPoints....	XML Document	06/22/2001 ...	608	44%	338	Measurement\DatedColle
DatedListEntry.svg	SVG File	06/22/2001 ...	3,131	83%	533	Measurement\DatedListE
DatedListEntry.xml	XML Document	06/22/2001 ...	4,814	86%	696	Measurement\DatedListE

Example of an unzipped export file

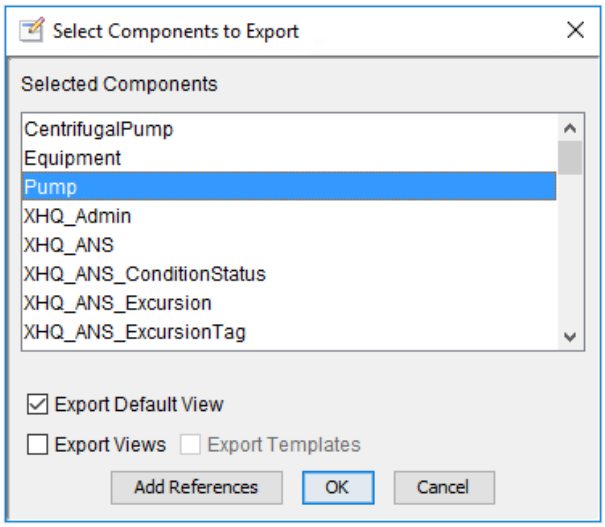
To export components

1. On the **File menu**, click **Export Components** and confirm the component export by clicking **Yes**.
   The "Select an export file" dialog box appears.

   The default directory is **`%APPDATA%\XHQ\XHQ devClient\repos`**.



2. **Enter a file name** and click **Save**.
   The "Select Components to Export" dialog box appears.



3. Select the component(s) to export.

   Use the **SHIFT** key to select a group of components or the **CTRL** key to select individual components.

4. OPTIONAL
   Select the views or templates to export.

6. OPTIONAL
   Click **Add References** to include embedded components in the export. (These are components that are directly associated with component members.)
   If you are exporting a subclass component, clicking "Add References" adds the base component. However, if you are exporting a base component, "Add References" does not add the subclass component(s).

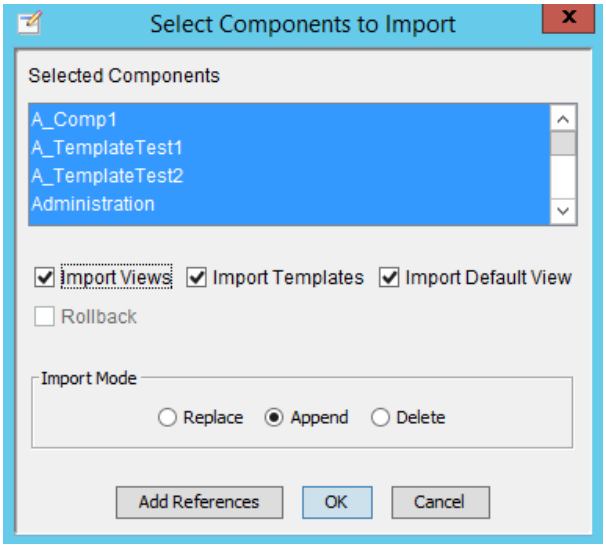
   For more information on base and subclass components, go to the topic, *About Base and Subclass Components*.

7. Click **OK**.

Importing Components

To import into your model, you simply open an exported `.zip` file and select the component(s) to include into your current XHQ Workbench. From a security standpoint, only Solution Administrators can import.

WARNING
Make sure to **backup your system** before importing components. There is no way to restore the previous state of imported components and views except by using a separate backup of the previous repos.



Select Components to Import Dialog Box

Like exporting, you can also import **Views and Templates** associated with the components you select. You also have the option to **Add References** and select an **Import Mode**, which allows you to include all component descendents.

Once import is competed, a dialog box appears displaying a log of all imported items. This information is also available in a separate log file located in **`%XHQ_SERVER_LOGS%\catalog.log`**.

About Component Subclassing and Importing Your Model

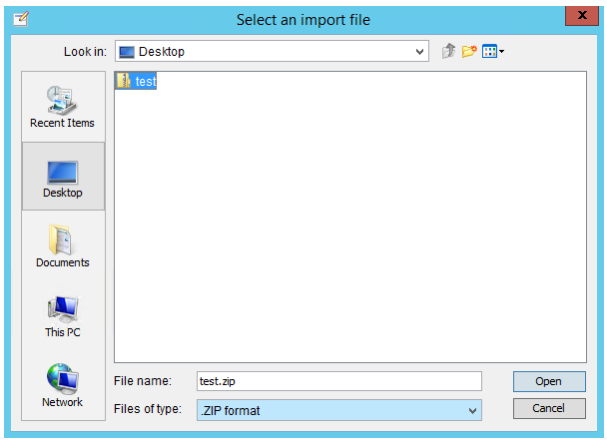
Here are some rules to consider when importing your model.

- A component defined as a subclass cannot be changed to extend a different component or changed to a base component.
- Subclasses may be defined before base classes.
- Subclasses may extend exactly ONE component.
- Subclass member and view name must be different from their base class member and view names.

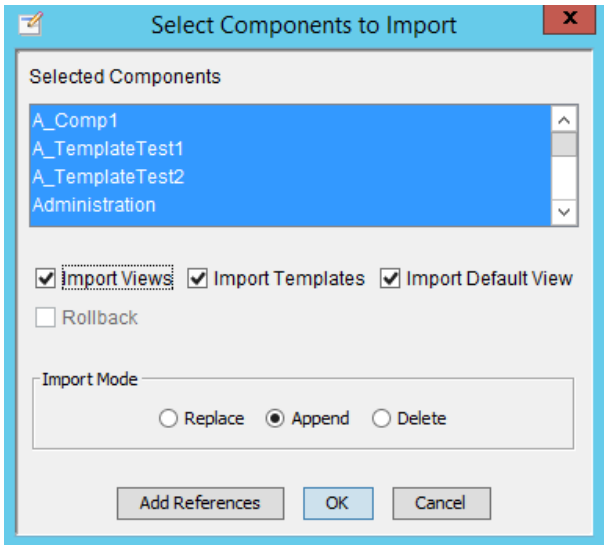
To import components

To import components, you must either belong to the **Model Admin role** OR have **Model Import permissions**.

1. **Backup** your model and solution.
2. From the **XHQ Workbench**, on the **File menu**, click **Import Components**.
   An important message appears.
3. To continue with the import, click **Yes**.
   The "Select a .ZIP import file" dialog box appears.



4. **Navigate** to and **select** the `.zip` file, and then click **Open**.
   The "Select Components to Import" dialog box appears.



5. Select the component(s) to import.

   Once you select a component from this list, the "Import Views" option is enabled.

6. OPTIONAL
   Click **Import Views** to include the views associated with the components you selected.

7. OPTIONAL
   After selecting **Import Views**, click **Import Templates** to include the template components associated with the views you selected.
8. OPTIONAL
   Click **Rollback** to indicate that the import is a rollback.
9. Select an **Import Mode**.
   - **Replace**
     This is the default mode. Any components included in the model import file will replace matching components in the XHQ Workbench.
   - **Append**
     Component members included in the model import file will either replace (existing matching members) or be added to the XHQ Workbench. Any components that existed in the XHQ Workbench, but whose definition does not appear in the model file are left alone. Rollbacks are not supported in this mode.
   - **Delete**
     All components included in the import file are deleted, regardless of whether or not they contain members.
10. OPTIONAL
    Click **Add References** to include embedded components in the import.
11. To start the import, click **OK**.

Once the import process is complete, the icons for the component(s) you selected appear in the XHQ Workbench.

Copying Repositories

When copying repositories using the export/import method, it is important to note that the two repositories operate independently. Changes you make in one repos are not automatically applied when you import into the other repos. You must explicitly apply those changes in the other repos.

For example, you copy ReposA using export/import into ReposB. You rename a component in ReposA (from “Comp1” to “Comp2”). You need the same change to be applied to ReposB. You cannot simply re-import into ReposB. Doing so does not rename Comp1 to Comp2 in ReposB. Instead, Comp2 is treated like a new, named component.

Any collection member or component member based on the original component in ReposB will also remain of that type.

To properly account for the component rename in ReposB, you must do one of the following prior to re-importing:

- From ReposB, go to the XHQ Workbench and rename “Comp1” to “Comp2”. Then re-import.

OR

- From the latest exported `.zip` file, open `model.xml` and locate the element for “Comp2”.

  Change:

  `name=”Comp2”`

  to

  `name=”Comp1”`

  And add the tag to the element:

  `renameTo=”Comp2`

  Example:

```
<XHQ:Component
 name="Comp1"
extends=""
subClassing="true"
page="Acme Model"
 renameTo="Comp2"
desc=""
...
</XHQ:Component>
```

  Save `model.xml` and re-import.

XHQ Model Export: List of supported XML Elements

See also the topic, *DTD Validation Structure*.

XML Element	`XHQ:Model`	
Description	This is the root element of the document. It contains one or more child elements. Required: YES	
Attribute(s)	`xmlns:XHQ=http://www.siemens.com/indx`	
		Is required and must be defined this way. This attribute defines the XHQ namespace.
	`esHost`	Identifies the Enterprise server hostname, and is generated by an export from the XHQ Workbench.
	`esTime`	Identifies the time of export (as an integer), and is generated by an export from the XHQ Workbench. This attribute is not used during import.
	`esFormattedTime`	Identifies the time of export (as a readable string) and is generated by an export from the XHQ Workbench. This attribute is not used during import.
	`xhqVersion`	Identifies the version of XHQ and is generated by an export from the XHQ Workbench. This attribute is not used during import.
	`xhqBuild`	Identifies the build version of XHQ and is generated by an export from the XHQ Workbench. This attribute is not used during import.

Child Element(s)	`XHQ:Component`
	`XHQ:ModifiedComponent`
	`XHQ:DeletedComponent`	

Restrictions	This element appears only once in the document, as the root element.	

XML Element	`XHQ:Component`	
Description	This element describes an XHQ Workbench component. Required: NO	
Attribute(s)	`name`	Describes the name of the component.
	`extends`	Describes the name of the base component, if any.
	`page`	Describes the name of the page to which this component belongs.
	`desc`	Indicates the description of the component.

XML Element	`XHQ:Component`	
Child Element(s)	XHQ:ComponentRef	XHQ:CollectionRef
	XHQ:Group	XHQ:boolean
	XHQ:integer	XHQ:long
	XHQ:float	XHQ:double
	XHQ:decimal	XHQ:datetime
	XHQ:interval	XHQ:string
	XHQ:text	
Restrictions	None	

XML Element	`XHQ:ComponentRef`	
Description	This element describes a component that is a member (child) of another component. Required: NO	
Attribute(s)	`name`	Describes the name of the member (child) component.
	`usage`	Indicates the description of the member component.
	`type`	Describes the type of the member (child) component.
	`hidden`	Indicates if this member component is hidden or not. It can be true or false.
	`tagAttribute`	Describes the tag attribute of the member (child) component. This always has the value of "None".
Child Element(s)	None	
Restrictions	Can only be part of `XHQ:Component`.	

XML Element	`XHQ:CollectionRef`	
Description	This element describes a collection of components that is a member (child) of another component. Required: NO	
Attribute(s)	`name`	Describes the name of the member collection.
	`usage`	Indicates the description of the member collection.
	`type`	Describes the type of the member (child) collection.
	`hidden`	Indicates if this member collection is hidden or not. It can be true or false.
	`tagAttribute`	Describes the tag attribute of the member collection. This always has the value of "None".

XML Element	`XHQ:CollectionRef`
Child Element(s)	None
Restrictions	Can only be part of `XHQ:Component`.

XML Element	`XHQ:Group`	
Description	This element describes a group that is a member (child) of another component. Required: NO	
Attribute(s)	`name`	Describes the name of the member group.
	`usage`	Indicates the description of the member group.
	`type`	Describes the type of the member group.
	`hidden`	Indicates if this member group is hidden or not. It can be true or false.
Child Element(s)	None	
Restrictions	Can only be part of `XHQ:Component`.	

XML Element	`XHQ:boolean`	
Description	This element describes a component data member that is of type Boolean.** Required: NO	
Attribute(s)	`name`	Describes the name of the data member.
	`usage`	Indicates the description of the data member.
	`hidden`	Indicates if this data member is hidden or not; it can be true or false.
	`tagAttribute`	Describes the tag attribute of the member collection. **Note:** If the data member is of type "String" or "Text", this attribute can have the following values: None|Value|Config|Description|Key|Units. For all the rest of data member types it has only two values "None" or "Value".
Child Element(s)	None	
Restrictions	None	

** The elements XHQ:integer, XHQ:long, XHQ:float, XHQ:double, XHQ:decimal, XHQ:datetime, XHQ:interval, XHQ:string, and XHQ:text all have the same attributes as XHQ:boolean. As such, refer to the table above for attribute descriptions.

XML Element	`XHQ:ModifiedComponent`	
Description	This element describes a component that has been modified and that may contain deleted members and/or views. This element syntax is used in deleting components through import. Required: NO	
Attribute(s)	`name`	Describes the name of the component that is modified.
Child Element(s)	`XHQ:DeletedMembersList` `XHQ:DeletedViewsList`	
Restrictions	This element syntax is used in deleting components through import.	

XML Element	`XHQ:DeletedMembersList`	
Description	This element describes a list of members for a certain component that are to be deleted through XML import. Required: NO	
Attribute(s)	`name`	Describes the name of the list of members to be deleted.
Child Element(s)	`XHQ:DeletedMember`	
Restrictions	None	

XML Element	`XHQ:DeletedViewsList`	
Description	This element describes a list of views for a certain component that are to be deleted through XML import. Required: NO	
Attribute(s)	`name`	Describes the name of the list of views to be deleted.
Child Element(s)	`XHQ:DeletedView`	
Restrictions	None	

XML Element	`XHQ:DeletedMember`	
Description	This element describes the component member that is to be deleted through XML import. Required: NO	
Attribute(s)	`name`	Describes the name of the component member that is to be deleted.
Child Element(s)	None	
Restrictions	None	

XML Element	`XHQ:DeletedView`	
Description	This element describes the view that is to be deleted through XML import. Required: NO	
Attribute(s)	`name`	Describes the name of the component view that is to be deleted.
Child Element(s)	None	
Restrictions	None	

XML Element	`XHQ:DeletedComponent`	
Description	This element describes the component that is to be deleted through XML import. Required:	
Attribute(s)	`name`	Describes the name of the component that is to be deleted.
Child Element(s)	None	
Restrictions	None	

Importing the Model Using XHQ Web Services

The XHQ Web Services includes three web methods that enable you to invoke the xhqci command `importmodel` asynchronously through XHQ Web Services.

For more information, see the topic, *Web Services*, located in the XHQ SDK Reference Guide.

Solution Export and Import

XHQ allows you to export solution member configurations and import them into another model. When importing, all configurations in the given file are imported, unless you choose to limit which connection group configuration to include.

Exporting a Solution Configuration

With XHQ, you can export one or more solution member configurations to an .XML document format.

The following table displays the type of data that is exported.

Exported Data

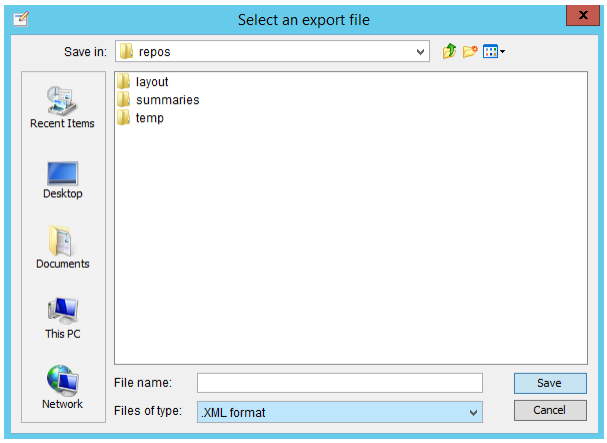
For this type . . .	These parameters are exported
XHQ data members	Access, Alias, Connection, History and Expressions parameters
XHQ component members – single dimension	Access parameters
XHQ component members – collection dimension	Access and Link parameters
Collections	Access, Connection, and Cache parameters

When exporting, you can select which connection group configurations to include. You also have the option to **Include Descendants**, which allows you to export all the component members within the node you select to export.

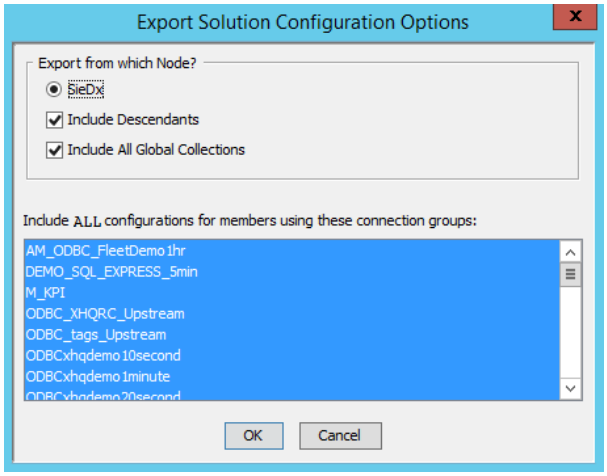
To export a solution configuration

1. From the **XHQ Solution Builder** navigation panel, select a node.
2. On the **Solution menu**, click **Export Configuration**.
   The "Select an XML export file" dialog box appears.

   The default directory that opens from this dialog box is **%XHQ_DEV_REPOS%**.



3. **Enter a file name** and click **Save**.
   The "Export Solution Configuration Options" dialog box appears.



4. Under **Export from which Node**, select a node.

   The nodes listed include the root and possibly a component instance (only if one was selected in step 1). For example, in the image above, there are two nodes to choose from: the root named "Enterprise" and a component instance named "ODBCPoint."

5. OPTIONAL
   Click **Include Descendants** to include all the component members of the given node.
6. Select which connection group configurations to include.

   Use the **SHIFT** key to select a group of items or the **CTRL** key to select individual items. Press **CTRL+A** to select all items.

7. OPTIONAL
   Select **(None)** to export nodes that do not have connection configurations.
8. Click **OK**.
   A status dialog box appears momentarily, showing the configurations being exported.
9. OPTIONAL
   To cancel the export, click Cancel from this export status dialog box.

Importing a Solution Configuration

When importing, all configurations in the given file are imported, unless you choose to limit which connection groups to include.

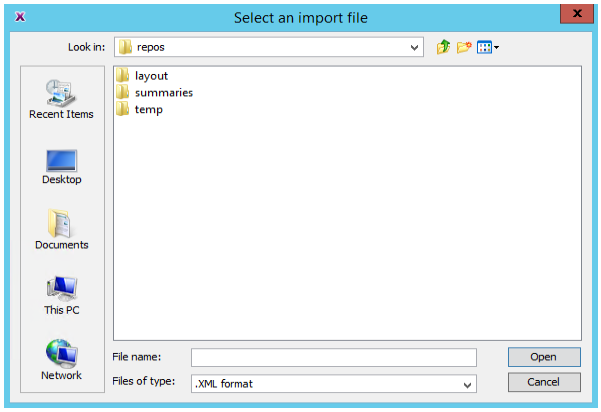
All existing configurations matched during the import are replaced. If there are configuration attributes not specified in the import, they are left preserved in the solution. For example, an imported configuration might not specify history settings. So, if the configuration you import does not include history settings, the existing history configurations are left intact.

In addition, if a collection is referenced in the import file, but it does not exist in the destination solution, then a placeholder for the new collection is automatically created.

To import a solution configuration

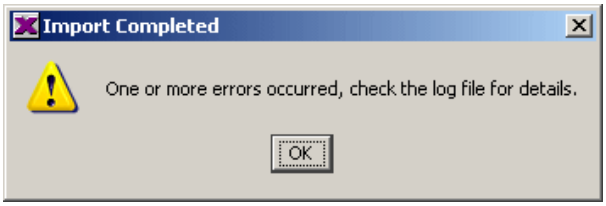
If security is enabled, you must be a member of the solution administrator's role in order to import configurations.

1. From the **XHQ Solution Builder**, on the **Solution menu**, click **Import Configuration**.
   The "Select an XML import file" dialog box appears.



2. **Navigate** to and **select** the file, and then click **Open**.
   The "Import Solution Configuration Options" dialog box appears.
3. Select which connection group configurations to import.
4. Click **OK**.
   A status dialog box appears momentarily, showing the configurations being imported.
5. OPTIONAL
   To cancel the export, click Cancel from this export status dialog box.

   If a configuration failed to import, a message appears.



   Details of the import failure are noted in a file located in `%XHQ_`

SERVER_LOGS%.

XHQ Solution Documents

Configuring an XML File for Import

Each major element of the XML solution document specifies a configuration for an XHQ collection, collection member, or solution member.

When editing the XML import file, use tools specifically designed for XML editing. Do not use text editors (such as Notepad and WordPad) to edit because they do not properly save files in UTF-8 format.

To configure your XML document as an XHQ solution document, follow these rules.

- The document must be a well-formed XML document.
- The root element of your document must be the `XHQ:SolutionConfig` element.
- Elements and attributes that are not recognized will be ignored.
- Values are *case-sensitive* to document element and attribute **labels**.
- Values are *case-**in**sensitive* to document element and attribute **values**.
- All required attributes and child elements must contain valid configuration values.
- Solution members referenced in the solution document must already exist in your solution. Therefore, the XHQ model must be defined prior to import.

  Example: In order to import a configuration for "::Enterprise.DDEPoint.CurrentValue", the solution member should already exist.

In addition, for **values**:

- Ampersand (&) characters must be replaced with "&"
- Quote (") characters must be replaced with """
- Less than (<) characters must be replaced with "<"
- Greater than (>) characters must be replaced with ">"

XHQ Solution Export: List of supported XML Elements

See also the topic, *DTD Validation Structure*.

XML Element	`XHQ:SolutionConfig`
Description	This is the root element of the document. It contains one or more child elements. Required: YES
Attribute(s)	`xmlns:XHQ=http://www.siemens.com/indx`
	Is required and must be defined this way. This attribute defines

XML Element	`XHQ:SolutionConfig`	
		the XHQ namespace.
	`ssHost`	Identifies the Solution server hostname, and is generated by an export from the Solution Builder. This attribute is not used during import.
	`rootName`	Identifies the root component name of the solution. This attribute is not used during import.
	solutionRevID	Identifies the solution revision ID, and is generated by an export from the Solution Builder. This attribute is not used during import.
	`ssTime`	Identifies the time of export (as an integer), and is generated by an export from the Solution Builder. This attribute is not used during import.
	ssFormattedTime	Identifies the time of export (as a readable string) and is generated by an export from the Solution Builder. This attribute is not used during import.
	`xhqVersion`	Identifies the version of XHQ and is generated by an export from the Solution Builder. This attribute is not used during import.
	`xhqBuild`	Identifies the build version of XHQ and is generated by an export from the Solution Builder. This attribute is not used during import.

Child Elements	`XHQ:Collection`
	`XHQ:Component`
	`XHQ:Group`
	`XHQ:DataMember`
	`XHQ:ComponentMember`
	`XHQ:GroupMember`
	`XHQ:CollectionMember`
	`XHQ:DeletedCollection`	

Restrictions	This element appears only once in the document, as the root element.	

XML Element	XHQ:Collection	
Description	This element describes a collection configuration. Required: NO	
Attribute(s)	`xhqPath`	Describes the XHQ path for the configuration. This path point to a Collection, Collection member, or Solution member.
	`xhqComponent`	Describes the constraint placed on a Collection for the configuration.
	`usage`	Describes the collection.
	`tagCollection`	Specifies the tag collection.
	`roles`	Describes the role(s) required to access the member. The roles for this configuration are set exactly to what is specified, or left alone if the field is blank. If %none is specified, all roles are removed.
	`writes`	For each of the roles enumerated in the "roles" attribute, it indicates whether that specific role has the permissions to write to this collection or not. This is a semi-colon separated string of Booleans (for example, `true;true;false`).
	`connectionGroup`	Describes the connection group assigned to the configuration.

Child Elements	`XHQ:ConnectProp`
	`XHQ:CacheProp`
	`XHQ:CacheIndexList`	

Restrictions	The collection cannot be created (if needed) unless the `xhqComponent` attribute specifies a valid XHQ component.	

XML Element	XHQ:Component	
Description	This element describes a component configuration. Required: NO	
Attribute(s)	`xhqPath`	Describes the XHQ path for the configuration. This path point to a Collection, Collection member, or Solution member.
	`xhqComponent`	Describes the constraint placed on a Collection for the configuration.
	`usage`	Describes the component.
	`roles`	Describes the role(s) required to access the member. The roles for this configuration are set

XML Element	`XHQ:Component`	
		exactly to what is specified, or left alone if the field is blank. If %none is specified, all roles are removed.
Child Elements	None	
Restrictions	The component cannot be created (if needed) unless the `xhqComponent` attribute specifies a valid XHQ component.	

XML Element	`XHQ:Group`	
Description	This element describes a group configuration. Required: NO	
Attribute(s)	`xhqPath`	Describes the XHQ path for the configuration. This path point to a Collection, Collection member, or Solution member.
	`xhqComponent`	Describes the constraint placed on a Collection for the configuration.
	`usage`	Describes the group.
	`roles`	Describes the role(s) required to access the member. The roles for this configuration are set exactly to what is specified, or left alone if the field is blank. If %none is specified, all roles are removed.
Child Elements	None	
Restrictions	The group cannot be created (if needed) unless the `xhqComponent` attribute specifies a valid XHQ component.	

XML Element	`XHQ:DataMember`	
Description	This element describes a solution or collection data member configuration. Required: NO	
Attribute(s)	`xhqPath`	Describes the XHQ path for the configuration. *Example:* ::Enterprise.OPCPoint.CurrentValue
	`xhqComponent`	Describes the constraint placed on a Collection for the configuration.
	`xhqMember`	Describes the member of a tag, in case the attribute specified in the "xhqComponet" is an XHQ Tag. This is used mostly with XHQ Tag Sync.
	`usage`	Describes the data member.

XML Element	XHQ:DataMember	
	alias	Specifies the alias for this configuration. Use "%none" to remove the alias from the configuration.
	datacontrolOptimizeAuto	Indicates if the data control optimizer is enabled or not. It can be true or false.
	datacontrolAllowWrite	Indicates if the user can write to the point described by this Data Member. It can be true or false.
	datacontrolConstrainMinimum	Indicates if the constraint for the minimum value written to this point is enabled or not. It can be true or false.
	datacontrolMinimum	The minimum numerical value allowed to be written to the point.
	datacontrolConstrainMaximum	Indicates if the constraint for the maximum value written to this point is enabled or not. It can be true or false.
	datacontrolMaximum	The maximum numerical value allowed to be written to the point.
	datacontrolConstrainMask	Indicates if the constraint for the mask value written to this point is enabled or not. This used only for the "String" and "Text" types. It can be true or false.
	datacontrolMask	The mask value to be applied to the point. This is used only for the "String" and "Text" types.
	expression	Describes the expression assigned to the configuration.
	expressionSchedulerEnabled	Indicates if the expression scheduler for this primitive is enabled or not. It can be true or false.
	expressionScheduleBaseTime	Describes the expression scheduler base time.
	expressionSchedulePeriod	Describes the expression scheduler period in milliseconds.
	roles	Describes the role required to access the XHQ member. If not specified, the existing role in the solution is preserved.
	writes	For each of the roles enumerated in the "roles" attribute, it indicates whether that specific role has the permissions to write to this point or not. This is a semi-colon separated string of Booleans (for example,

XML Element	XHQ:DataMember	
		true;true;false).
	historian	Describes the historian used for this configuration. The supported values are "Data Recorder", "PI", "IP21", and "PHD". Use "%none" to remove the historian from the configuration.
	historianRecord	Describes the historian record setting used for this configuration. If TRUE, history is recorded to this item.
	hitorianCompression	Describes the historian compression method used for this configuration. The only supported value is "Boxcar-Backslope".
	historianDeviation	Describes the historian deviation setting used for this configuration.
	historianMinTime	Describes the historian minimum time setting used for this configuration.
	historianMaxTime	Describes the historian maximum time setting used for this configuration.
	historianName	Indicates the type of the external historian used to store the values (for example, Data Recorder, PI, or PHD).
	historianConnectGroup	Indicates the connection group assigned to the historian.
	connectionGroup	Describes the connection group assigned to the configuration.
Child Elements	XHQ:HistorianConnectProp XHQ:ConnectProp	
Restrictions	None	

XML Element	XHQ:ComponentMember	
Description	This element describes a solution or collection component member (for a single dimension) configuration. Required: NO	
Attribute(s)	xhqPath	Describes the XHQ path for the configuration. *Example:* ::Enterprise.OPCPoint.CurrentValue
	linkPath	Describes the path to a proxy configuration.
	xhqComponent	Describes the constraint placed on a Collection

XML Element	XHQ:ComponentMember	
		for the configuration.
	usage	Describes the component member.
	roles	Describes the role required to access the XHQ member. If not specified, the existing role in the solution is preserved.
Child Elements	None	
Restrictions	None	

XML Element	XHQ:GroupMember	
Description	This element describes a group member configuration. Required:	
Attribute(s)	xhqPath	Describes the XHQ path for the configuration. *Example:* ::Enterprise.OPCPoint.CurrentValue
	xhqComponent	Describes the constraint placed on a Collection for the configuration.
	usage	Describes the group member.
	roles	Describes the role required to access the XHQ member. If not specified, the existing role in the solution is preserved.
Child Elements	None	
Restrictions	None	

XML Element	XHQ:CollectionpMember	
Description	This element describes a component member (collection dimension) configuration. If the xhqPath is not valid, but eh xhqComponent is defined with a valid component, then XHQ will attempt to create the collection. Required: NO	
Attribute(s)	xhqPath	Describes the XHQ path for the configuration. *Example:* ::Collections.OPCPoint.CurrentValue
	xhqComponent	Describes the constraint placed on a Collection for the configuration.
	usage	Describes the role required to access the XHQ member. If not specified, the existing role in the solution is preserved.
	roles	Describes the role required to access this XHQ

XML Element	XHQ:CollectionpMember	
		collection. If not specified, the existing role in the solution is preserved. It is a semi-colon separated string list of role names.
	linkCollection	Describes the collection to which the configuration is linked. Use "%none" to remove the link form the configuration.
	linkConstraint	Describes the constraint placed on a collection for the configuration. Use "%none" to remove the constraint from the configuration.
Child Elements	None	
Restrictions	None	

XML Element	XHQ:ConnectProp	
Description	This element describes a property supported by the connection group specified in the connectionGroup. Required: NO	
Attribute(s)	name	A property name recognized by the connection group's connector.
	value	A value associated with the property name.
Child Elements	None	
Restrictions	This element is valid only for XHQ:DataMember and XHQ:Collection elements that specify a valid connectionGroup element, and the connector must support import/export capability.	

XML Element	XHQ:HistorianConnectProp	
Description	This element describes a Historian connection property. Required: NO	
Attribute(s)	name	Describes the name of the historian connection property.
	value	A value associated with the property name.
Child Elements	None	
Restrictions	This element is only valid for XHQ:DataMember.	

XML Element	XHQ:CacheProp
Description	This element describes a cache setting for a XHQ:Collection member (column).

XML Element	`XHQ:CacheProp`	
	Required: NO	
Attribute(s)	`name`	Describes the name of the collection member (column).
	`cacheMaxLength`	Describes the maximum string lengths for a collection's cache settings.
	`cachePrecision`	Indicates the cache precision value.
	`cacheScale`	Indicates the cache scale value.
	`subscribable`	Indicates if this member's cache property is subscribable or not. It can be true or false.
	`cacheUniqueKey`	Indicates if this member cache property is a unique key or not. It can be true or false.
Child Elements	None	
Restrictions	None	

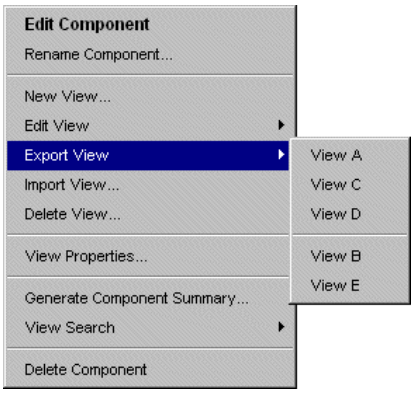
XML Element	`XHQ:CacheIndexList`	
Description	This element describes a cache index list setting for a `XHQ:Collection`. Required: NO	
Attribute(s)	`name`	Describes the name of the cache index for the collection.
Child Elements	`XHQ:CacheIndexListMember`	
Restrictions	None	

XML Element	`XHQ:CacheIndexListMember`	
Description	This element describes a collection member (column) that is part of cache index list setting for a collection `XHQ:CacheIndexList`. Required: NO	
Attribute(s)	`name`	Describes the name of the collection member (column) that is part of the cache index.
	`sort`	Describes the sorting (Ascending or Descending) type for the collection member (column).
Child Elements	None	
Restrictions	None	

XML Element	`XHQ:DeletedCollection`	
Description	This element describes a collection that is to be deleted through XHQ import. Required: NO	
Attribute(s)	`name`	Describes the name of the collection to be deleted.
Child Elements	None	
Restrictions	This element is only valid for `XHQ:SolutionConfig`.	

View Export and Import

XHQ lets you export views and their associated animations as files to be stored in any directory. You can then import those files into any running XHQ model. The XHQ Workbench is used to perform importing and exporting functions.



Right-click Shortcut Menu, Export View Command

To animate correctly, a view needs data sources (component primitive members). Consequently it makes most sense to assign an imported view to a component that shares a similar data structure. If you import a view to a component with identically named members, the animations are automatically re-established. If, however, the data members are different, the *Substitute Reference* dialog box is displayed that allows you to re-link the imported view's animations to the alternative primitive members of the target component.

XHQ only allows you to export and import completed views. You cannot import/export partial views or selections from views.

You can export the XHQ view to any directory on your network. Currently, the default directory is **`%APPDATA%\XHQ\XHQ devClient\repos`**.

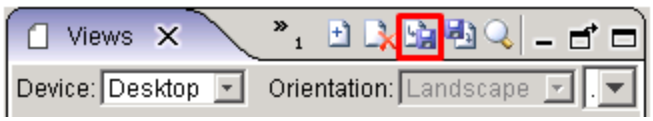
XHQ export filenames are in the following format:

```
<component class>_<view name>.view
```

Export/import of a native view (`.view` format) is **not supported between releases**. In this case, you may only export/import XML.

To export a view in XHQ Workbench

1. Launch **XHQ Workbench**.
2. In the **Components** tab, select a component with a view.
3. In the **Views** tab, select the view to export.
4. Select a **Device** from the drop-down list.
5. Depending on the device type, select an **Orientation**.
6. Click **Export**.



The "Export View" dialog appears.

7. Enter the **`.view`** or **`.xml`** file name and select where you want to save the file.
8. Click **OK**.

To import a view into XHQ Workbench

1. From **XHQ Workbench**, in the **Components** tab, select the component into which you want to import the view.
2. In the **Views** tab, click **Import**.
   The "Import View" dialog appears.
3. Locate the **`.view`** or **`.xml`** file you exported and click **Open**.
   The "Import View" dialog appears.
4. For **Rename View**, enter a new name.
5. Select a **Device and Orientation**.
6. Click **OK**.
   The "Substitute References" dialog appears.
7. OPTIONAL
   Select **New Members**.
8. Click **OK**.

Exporting a view containing a GISMap control

When you export a view containing a GISMap control with any of the following conditions, you will need to manually copy resources to the destination environment to which you are importing:

- If the Image provider is used and the images are stored in the repos;
- If Marker icon images are stored in the repos;
- If the Shapefile provider is used and the shapefile ZIP file is stored in the repos.

Exporting and Importing XHQ Animations (XANIs)

In XHQ, you can export and import the animated clipart. These animated clipart, called XANI (pronounced "zany") files, can be exported as files you can store in any directory. You can then import these clipart files into any running XHQ system. You use the XHQ Workbench to initiate all importing and exporting functions.

To animate correctly, a clipart element needs data sources (component primitive members). Consequently it makes most sense to assign an imported clipart element to a component that shares a similar data structure. When you export a clipart animation, its established backend links are broken. When you import clipart to a component, you must re-bind each animated element of the clipart to the primitive members of the target component in order for it to animate.

If a XANI containing a value item is inserted into a flow view, then all other items EXCEPT for that value item is inserted.

You can export the XANI to any directory on your network. Currently, the default directory is `%APPDATA%\XHQ\XHQ devClient\repos`. And the default XANI export filename is **`indx.xani`** (XANI files have the `.xani` extension).

To export a XANI

1. From the **XHQ Workbench**, **right-click on the component** containing the clipart you want to export.
2. On the shortcut menu, click **Edit View**, and then select the view.
   The XHQ Workbench appears.
3. Click the **Selection** tool.
4. Click and drag the mouse around the animated clipart element you want to export.
   As the mouse moves, a line appears, indicating the border of your selection.
5. Release the mouse button.
   Handles appear around the selected view element.
6. Click the **Export Selection to file** icon on the button bar.

   The "Export Selection" dialog box appears.
7. Click **Save** to save the exported animated clipart file into the default network or browse the network to find the location for the file.
   XHQ assigns the default name **`indx.xani`** to the clipart file. You can re-name the file.

To import a XANI

1. From the **XHQ Workbench**, **select the component** containing the view you want to import to, and right-click.
2. On the shortcut menu, click **Edit**, and then select the view.
   The XHQ Workbench appears.
3. Click on the **Insert Image** tool.
   The cursor changes to the insertion cursor.
4. Click in the Drawing panel to indicate where you want to import the XANI file.
   The "Select Image" dialog box appears.
5. **Navigate** to the animated clipart you want to import, **select it** and click **OK**.
   The animated clipart appears in the XHQ Workbench.

To re-bind individual elements in an imported XANI

1. Click the **Ungroup** button to separate the XANI (animated clipart) elements.

   Usually, there are many elements grouped into one animated clipart element. You will need to repeat steps 3 through 7 for each element of the imported animated clip art.

2. Select an element in the clipart group and click the **Animation** tab.
3. Click on **Add**.
   A drop-down list is displayed that contains supported set options.
4. Select one display characteristic (or animation) from the list of available options.
   The display characteristic (animation) you selected appears under the Set column of the Animation table.
5. From the **Using** drop-down list, left click to select the type of animation.
   The animation type you selected appears under the Using column of the Animation table.
6. Do one of the following:
   - From the **Evaluating** drop-down list, select the primitive member to drive the animation.
     The members listed are those contained in the component for which you are building the current view.

   or

   - If you want to drive the animation from a primitive member that belongs to a component further down in the solution hierarchy, such as a grandchild, you can enter in the name of the component in the Evaluating text box, using the following format and click **Enter**.

```
component1.component2.member
```

   You can include any number of nested primitive component members.

7. Depending upon your previous selections, you may be required to enter particular configuration parameters that match your choice of display characteristic (animation) and animation type.
8. Repeat steps 3 through 7 for each element of the imported animated clipart.
9. Click **Save**.

Selecting Configurations to Export or Import

When importing/exporting, you can also selectively import or export the following configurations:

- Role mapping/security
- Solution definition
- Connector definition
- Connection process
- Connection group

This can be done one of two ways: through their respective XHQ dialog box or through the use of their given command-line.

In the following sections, you will learn how:

- To access the dialogs to import/export.
- The XML document for the import/export is formatted.

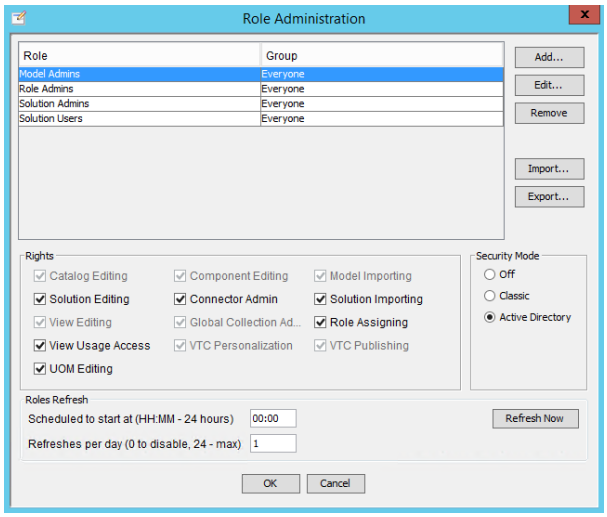
Role Mapping Configuration Import/Export

For security purposes, the XHQ Platform uses Windows groups to map to XHQ roles. The group mapped to a role may be a domain group or a group local to the model server host machine. Only the user with the **`RoleAdmins`** role may import security configurations.

About Passwords
Password values are not exported or imported in text. They are encrypted immediately and recorded in the XML export file as an encrypted string representation. They are decrypted only when needed.

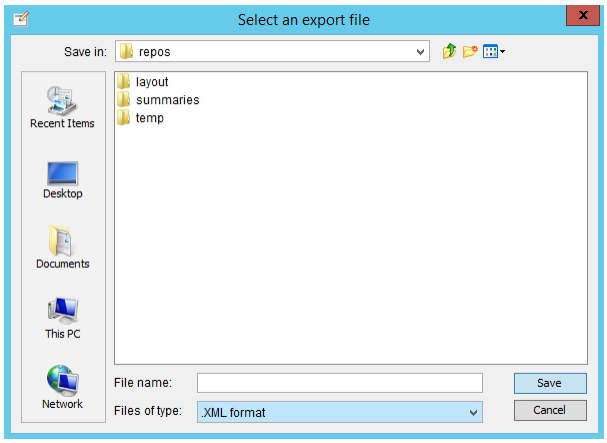
To export an XHQ security (mapped roles) configuration

1. From the **XHQ Workbench**, click the **Options menu**, and then click **Role Administration**.
   The "Role Administration" dialog box appears.



2. Click **Export**.
   The "Select an XML export file" dialog box appears.

   The default directory that opens from this dialog box is **`%APPDATA%\XHQ\XHQ devClient\repos`**.



3. **Enter a file name** and click **Save**.
   The "Select Role Mappings to Export" dialog box appears.

4. **Select the role(s)** to export and then click **OK**.

   Use the **SHIFT** key to select a group of items or the **CTRL** key to select multiple items. Press **CTRL+A** to select all items.

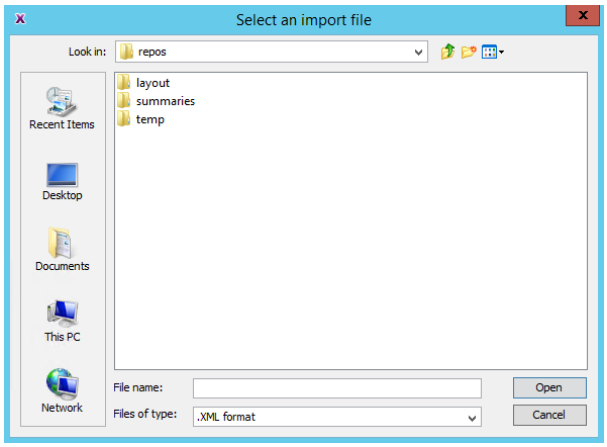
A message appears stating that the export completed without error.

For more information on the XML document, go to the topic, *Role Mapping: Defining the XML format*, located in the XHQ Administrator's Guide.

For an example of an XML document, go to the topic, *EXAMPLE: XML document for Role Mapping*.

To import an XHQ security configuration

1. From the **XHQ Workbench**, click the **Options menu**, and then click **Role Administration**.
   The "Role Administration" dialog box appears.
2. Click **Import**.
   The "Select an XML import file" dialog box appears.



3. **Navigate** to and **select** the file, and then click **Open**.
   The "Select Role Mappings to Import" dialog box appears.
4. **Select the role(s)** to import and then click **OK**.
   A message appears stating that the import was completed.

XHQ Role Mapping Export: List of supported XML Elements

See also the topic, *DTD Validation Structure*.

XML Element	`XHQ:RoleMap`	
Description	This is the root element of the document. It contains one or more child elements. Required: YES	
Attribute(s)	`xmlns:XHQ=http://www.siemens.com/indx`	
		The URL to the web site. This attribute defines the XHQ namespace. This is required and must be defined this way.
	`esHost`	Identifies the Enterprise server hostname.
	`esTime`	Identifies the time on the Enterprise server of export (as an integer).
	esFormattedTime	Identifies the time of export on the Enterprise server (as a readable string).
	`xhqVersion`	Identifies the version of XHQ.
	`xhqBuild`	Identifies the build version of XHQ.

XML Element	`XHQ:RoleMap`	
	securityMode	Identifies the current security setting. • **Classic** Security is on; integrated security is disabled. • **Active Directory** Both classic and integrated security is enabled. This is the default.
Child Element(s)	`XHQ:RoleMapEntry`	

XML Element	`XHQ:RoleMapEntry`	
Attribute(s)	`name`	The name of the XHQ role.
	`group`	The name of the corresponding Windows group.
	`local`	If **True**, then the Windows group is local to the Enterprise server host. Else, it is **False**. **Note:** Attribute values are case insensitive.
Child Element(s)	`XHQ:Right`	

XML Element	`XHQ:Right`	
Attribute(s)	`name`	Options: • Catalog_Editing • Component_Editing • View_Building • Solution_Editing • Connector_Administration • Global_Collection_Administration • Model_Importing • Solution_Importing • Role_Assigning
Child Element(s)	None	

EXAMPLE: EXPORT XML OF ROLEADMINS

Given:

The **`RoleAdmins`** XHQ role is assigned to the domain group named **`dom1\xhqadmins`**. All rights are assigned to the RoleAdmins role.

```
<XHQ:RoleMap
xmlns:XHQ="http://www.siemens.com/indx"
esHost="localhost"
esTime="0"
esFormattedTime="Jan 1, 1970 12:00:00 AM GMT"
xhqVersion="4.0"
xhqBuild="17"
securityMode="off">
<XHQ:RoleMapEntry
name="RoleAdmins"
group="dom1\xhqadmins"
local="false">
<XHQ:Right name="catalog_editing"/>
<XHQ:Right name="component_editing"/>
<XHQ:Right name="view_building"/>
<XHQ:Right name="solution_editing"/>
<XHQ:Right name="connector_administration"/>
<XHQ:Right name="global_collection_administration"/>
<XHQ:Right name="model_importing"/>
<XHQ:Right name="solution_importing"/>
<XHQ:Right name="role_assigning"/>
</XHQ:RoleMapEntry>
</XHQ:RoleMap>
```

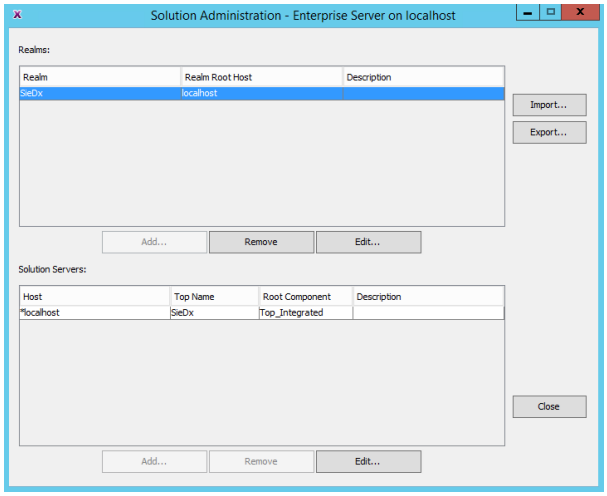
Solution Definition Import/Export

The configuration for a solution definition includes the following:

- Solution name
- Solution description
- Name of the Solution Server
- Name of the top-level component of the solution

To export a solution definition

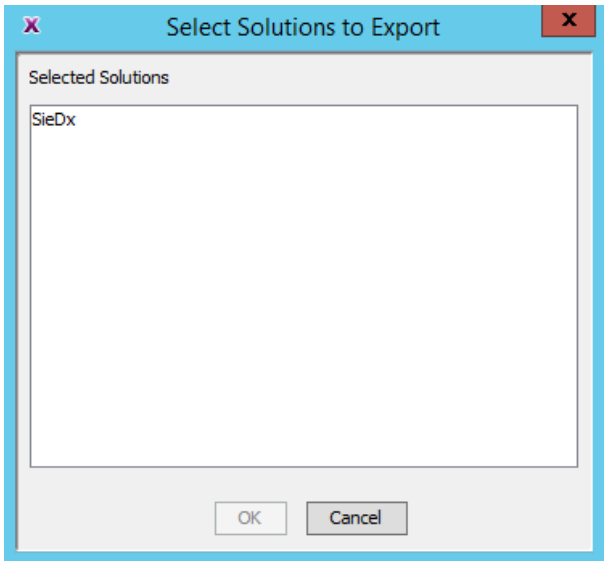
1. Click **Start**, point to **Programs**, then to **XHQ Development**, then to **Administration**, and click **Solutions**.
   The "Solution Administration" dialog box appears.



2. Click **Export**.
   The "Select an XML export file" dialog box appears.

   The default directory that opens from this dialog box is **`%APPDATA%\XHQ\XHQ devClient\repos`**.

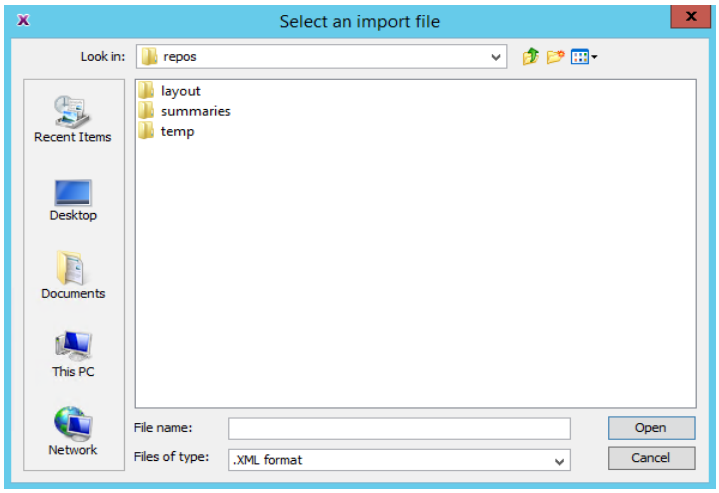
3. **Enter a file name** and click **Save**.
   The "Select Solutions to Export" dialog box appears.



4. **Select the solution** to export and then click **OK**.
   A message appears stating that the export completed without error.

To import a solution definition

1. Click **Start**, point to **Programs**, then to **XHQ Development**, then to **Administration**, and click **Solutions**.
   The "Solution Administration" dialog box appears.
2. Click **Import**.
   The "Select an XML import file" dialog box appears.



3. **Select** the XML import file and click **Open**.

XHQ Solution Definition Export: List of supported XML Elements

See also the topic, *DTD Validation Structure*.

XML Element	`XHQ:SolutionDefList`	
Description	This is the root element of the document. It contains one or more child elements. Required: YES	
Attribute(s)	`xmlns:XHQ=http://www.siemens.com/indx`	
		The URL to the web site. This attribute defines the XHQ namespace. This is required and must be defined this way.
	`esHost`	Identifies the Enterprise server hostname.
	`esTime`	Identifies the time on the Enterprise server of export (as an integer).
	esFormattedTime	Identifies the time of export on the Enterprise server (as a readable string).
	`xhqVersion`	Identifies the version of XHQ.
	`xhqBuild`	Identifies the build version of XHQ.
Child Element(s)	`XHQ:SolutionDef`	

XML Element	XHQ:SolutionDef	
Attribute(s)	name	The name of the XHQ solution.
	descr	A description of the solution.
	solnHost	The name of the machine on which the Solution server runs.
	top	The name of the component class of the top-level component to the solution.
Child Element(s)	None	

EXAMPLE: EXPORT XML OF SOLUTION DEFINITION

Given:

A solution named **Solution1** with a top-level component of the **ROOT** class.
The Solution server is on a machined named **host1.corp.indxhq.com**.

```
<XHQ:SolutionDefList
xmlns:XHQ="http://www.siemens.com/indx"
esHost="localhost"
esTime="0"
esFormattedTime="Jan 1, 1970 12:00:00 AM GMT"
xhqVersion="4.0"
xhqBuild="17">
<XHQ:SolutionDef
name="Solution1"
descr=""
solnHost="host1.corp.indxhq.com"
top="ROOT"
</XHQ:SolutionDef>
</XHQ:SolutionDefList>
```

Connector Definition Import/Export

Only a user that belongs to a role with **Connector Administration** rights may import or export connector definition configurations.

XHQ Connector Definition Export: List of supported XML Elements

See also the topic, *DTD Validation Structure*.

XML Element	`XHQ:ConnectorDefList`	
Description	This is the root element of the document. It contains one or more child elements. Required: YES	
Attribute(s)	xmlns:XHQ=http://www.siemens.com/indx	
		The URL to the web site. This attribute defines the XHQ namespace. This is required and must be defined this way.
	`ssHost`	Identifies the Solution server hostname.
	`ssTime`	Identifies the time on the Solution server of export (as an integer).
	`ssFormattedTime`	Identifies the time of export on the Solution server (as a readable string).
	`xhqVersion`	Identifies the version of XHQ.
	`xhqBuild`	Identifies the build version of XHQ.
Child Element(s)	`XHQ:ConnectorDef`	

XML Element	`XHQ:ConnectorDef`	
Attribute(s)	`name`	The name of the XHQ connector definition.
	`descr`	A description of the definition.
	`cimClass`	The name of the CIMInfo class.
	`cimJar`	The name of the CIM jar.
	`cpmClass`	The name of the main class of the connector process
	`cpmJar`	The name of the CPM jar.
	`type`	The pre-defined connector type. For example, ODBC.
Child Element(s)	None	

The attributes, `cimClass`, `cimJar`, `cpmClass`, and `cpmJar`, are legacy attributes and are no longer in use.

EXAMPLES: EXPORT XML OF CONNECTOR DEFINITION

Given:

An JDBC connector definition named **JDBC** with a description of "JDBC (Predefined Type)".

```
<XHQ:ConnectorDefList
xmlns:XHQ="http://www.siemens.com/indx"
ssHost="localhost"
ssTime="1500465601688"
ssFormattedTime="Jul 19, 2017 3:00:01 PM GMT+03:00"
xhqVersion="5.1.0.0"
xhqBuild="1053">
<XHQ:ConnectorDef
name="JDBC"
descr="JDBC (Predefined Type)"
type="JDBC"
/>
</XHQ:ConnectorDefList>
```

Given:

An ODBC connector definition named **ODBCPoint** with a description of "The ODBC Connector Type".

The following example uses the legacy attributes, cimClass, cimJar, cpmClass, and cpmJar, that are no longer in use.

```
<XHQ:ConnectorDefList
xmlns:XHQ="http://www.siemens.com/indx"
ssHost="localhost"
ssTime="0"
ssFormattedTime="Jan 1, 1970 12:00:00 AM GMT"
xhqVersion="4.0"
xhqBuild="17">
<XHQ:ConnectorDef
name="ODBC"
descr="The ODBC Connector Type"
cimClass="net.indx.indxconnector.db.odbc.cim.ODBCCIMInfo"
cimJar="odbccim.jar"
cpmClass="net.indx.indxconnector.db.jdbc.server.ODBCServer"
cpmJar="jdbc.jar"/>
</XHQ:ConnectorDef>
</XHQ:ConnectorDefList>
```

Connector Process Import/Export

Only a user that belongs to a role with **Connector Administration** rights may import or export connector process configurations.

XHQ Connector Process Export: List of supported XML Elements

See also the topic, *DTD Validation Structure*.

XML Element	`XHQ:ConnectProcessList`	
Description	This is the root element of the document. It contains one or more child elements. Required: YES	
Attribute(s)	`xmlns:XHQ=http://www.siemens.com/indx`	
		The URL to the web site. This attribute defines the XHQ namespace. This is required and must be defined this way.
	`ssHost`	Identifies the Solution server hostname.
	`ssTime`	Identifies the time on the Solution server of export (as an integer).
	ssFormattedTime	Identifies the time of export on the Solution server (as a readable string).
	`xhqVersion`	Identifies the version of XHQ.
	`xhqBuild`	Identifies the build version of XHQ.
Child Element(s)	`XHQ:ConnectorProcess`	

XML Element	`XHQ:ConnectProcess`
Description	The text for this tag is the command-line used to start the process.

XML Element	XHQ:ConnectProcess	
Attribute(s)	name	The name of the XHQ connect process.
	descr	A description of the process.
	manual	If **true**, then the process starts manually. If **false**, it starts automatically, which is the default.
	console	If **true**, the process logs results to a console window. If **false**, it does not log to a console window, which is the default.
	specUser	If **true**, the process runs under different credentials than that of the XHQ service. Else, it is **false**, which is the default.
	user	The name of the user under whose credentials the process runs. **Note:** This is ignored if specUser is false.
	password	An encrypted string containing the password of the user under whose credentials the process runs. **Note:** This is ignored if specUser is false.
Child Element(s)	None	

EXAMPLE: EXPORT XML OF CONNECTOR PROCESS

Given:

An ODBC connector process named **ODBCProc** with a description of "The ODBC Connector Process".

```
<XHQ:ConnectorProcessList
xmlns:XHQ="http://www.siemens.com/indx"
ssHost="localhost"
ssTime="0"
ssFormattedTime="Jan 1, 1970 12:00:00 AM GMT"
xhqVersion="4.0"
xhqBuild="17">
<XHQ:ConnectorProcess
name="ODBCProc"
descr="The ODBC Connector Process"
manual="false"
console="false"
specUser="true"
user="indx\test"
password="1002(encrypted)">
%XHQ_SERVER_HOME%\jre\bin\xhq_odbc -server -Xrs
-classpath "%XHQ_SERVER_HOME%\lib\connector.jar;
%XHQ_SERVER_HOME%\lib\jdbc.jar;
%XHQ_SERVER_HOME%\java\lib\collections.jar
net.indx.indxconnector.db.jdbc.server.DbServer
-driver sun.jdbc.odbc.JdbcOdbcDriver -url jdbc:odbc:
</XHQ:ConnectProcess>
</XHQ:ConnectProcessList>
```

Connector Group Import/Export

Only a user that belongs to a role with **Connector Administration** rights may import or export connector group configurations.

XHQ Connector Group Export: List of supported XML Elements

See also the topic, *DTD Validation Structure*.

XML Element	`XHQ:ConnectGroupList`	
Description	This is the root element of the document. It contains one or more child elements. Required: YES	
Attribute(s)	`xmlns:XHQ=http://www.siemens.com/indx`	
		The URL to the web site. This attribute defines the XHQ namespace. This is required and must be defined this way.
	`ssHost`	Identifies the Solution server hostname.
	`ssTime`	Identifies the time on the Solution server of export (as an integer).
	ssFormattedTime	Identifies the time of export on the Solution server (as a readable string).
	`xhqVersion`	Identifies the version of XHQ.
	`xhqBuild`	Identifies the build version of XHQ.
Child Element(s)	`XHQ:ConnectGroup`	

XML Element	`XHQ:ConnectGroup`	
Attribute(s)	`name`	The name of the XHQ connect group.
	`descr`	A description of the group.
	`definition`	The name of the XHQ connector definition associated with this group.
	`process`	The name of the XHQ connector process associated with this group.
	`cimClass`	The name of the CIMInfo class.
	`cimJar`	The name of the CIM jar.
	`cfgClass`	The name of the class the encapsulates the connect group configuration.
Child Element(s)	`ConnectGroupConfig`	

XML Element	XHQ:ConnectGroupConfig	
Attribute(s)	name	The name of the connector-specific configuration.
	value	The value of the configuration
Child Element(s)	None	

EXAMPLE: EXPORT XML OF CONNECTOR GROUP

Given:

An ODBC connector definition named **ODBCPoint** with a connect group named **ODBCGroup**, which has a description of "The ODBC Connector Group". It is associated with the **ODBCProc** connection process.

```
<XHQ:ConnectGroupListxmlns:XHQ="http://www.siemens.com/indx"
ssHost="localhost"
ssTime="0"
ssFormattedTime="Jan 1, 1970 12:00:00 AM GMT"
xhqVersion="4.0"
xhqBuild="17">
<XHQ:ConnectGroup
name="ODBCGroup"
descr="The ODBC Connector Group"
type="ODBC"
process="ODBCProc"
cimClass="net.indx.indxconnector.db.odbc.cim.ODBCCIMInfo"
cimJar="odbccim.jar"
cfgClass="net.indx.indxconnector.odbc.cim.
ODBCConnectNameConfig”>
<XHQ:ConnectGroupConfig
name="pollrate"
value="10000"/>
<XHQ:ConnectGroupConfig
name="coherent"
value="false"/>
</XHQ:ConnectGroup>
</XHQ:ConnectGroupList>
```

Import/Export Delete

This topic describes the following XHQ import/export functionalities:

- The XML syntax for deleting collections, roles, XHQ Workbench components, members and views;
- The use of the XHQ export and import mechanism to leverage the XML delete syntax;
- The DTD validation for importing XML used for Solution configurations and Model (components and views).

The DTD validation structure can be found in the topic, *DTD Validation Structure*.

XML Supported Delete Syntax

The XML syntax (tag) for **deleting collections** from the XHQ Solution Server is:

```
<XHQ:DeletedCollection
name="<COLLECTION_NAME>"
>
```

Example:

```
<XHQ:DeletedCollection
name="AMAssets"
/>
```

The "XHQ:DeletedCollection" element tags can be used either in a XHQ standard import/export Solution Configs XML file, in conjunction with the rest of the XHQ supported element tags, or in a separate XML file that contains only the deleted collections tag elements. Here's a simple example of a valid XML file that can be used, through XHQ Solution Configs import, to delete XHQ collections:

```
<XHQ:SolutionConfig
   xmlns:XHQ="http://www.siemens.com/indx"
   ssHost="localhost"
   rootName="Enterprise"
   solutionRevID="3677"
   ssTime="1254339578223"
   ssFormattedTime="Sep 30, 2009 12:39:38 PM PDT"
   xhqVersion="4.0"
   xhqBuild="211">
   <XHQ:DeletedCollection
      name="AMAssets"
   />
   <XHQ:DeletedCollection
      name="LMXANSExcursions"
   />
   <XHQ:DeletedCollection
      name="MMWorkOrders"
   />
   <XHQ:DeletedCollection
      name="PI_Tags"
   />
   <XHQ:DeletedCollection
      name="Tags"
   />
</XHQ:SolutionConfig>
```

The XML syntax (tag) for **deleting role mappings** is:

```
<XHQ: DeletedRoleMapEntry
```

```
name="<ROLE_NAME>"
/>
```

Example:

```
<XHQ: DeletedRoleMapEntry
name="TestRole"
/>
```

The "XHQ: DeletedRoleMapEntry" element tags can be used either in a XHQ standard role import/export XML file, in conjunction with the rest of the XHQ role supported element tags, or in a separate XML file that contains only the deleted roles tag elements. Here's a simple example of a valid XML file that can be used, through XHQ role import (using the XHQ Workbench UI for roles or XHQCI role import command), to delete XHQ roles:

```
<XHQ:RoleMap
   xmlns:XHQ="http://www.siemens.com/indx"
   esHost="localhost"
   esTime="1254533277528"
   esFormattedTime="Oct 2, 2009 6:27:57 PM PDT"
   xhqVersion="4.0"
   xhqBuild="211"
   securityMode="integrated">
   <XHQ:DeletedRoleMapEntry
      name="Role1"
   />
   <XHQ:DeletedRoleMapEntry
      name="Role2"
   />
   <XHQ:DeletedRoleMapEntry
      name="Role3"
   />
</XHQ:RoleMap>
```

The XML syntax (tag) for **deleting XHQ Workbench components** is:

```
<XHQ: DeletedComponent
name="<COMPONENT_NAME>"
/>
```

Example:

```
<XHQ: DeletedComponent
name="xpak1"
/>
```

The XML syntax (tag) for deleting XHQ Workbench component members and/or views is:

```
<XHQ:ModifiedComponent
name="<COMPONENT_NAME>"
<XHQ:DeletedMembersList>
<XHQ:DeletedMember
name="<COMPONENT_MEMBER_NAME>"
/>

<XHQ:DeletedMember
name="<COMPONENT_MEMBER_NAME>"
/>
</XHQ:DeletedMembersList>
<XHQ: DeletedViewsList>
<XHQ: DeletedView
name="<VIEW_NAME>"
/>

<XHQ: DeletedView
```

```
name="< VIEW _NAME>"
/>
</XHQ: DeletedViewsList >
</XHQ:ModifiedComponent>
```

Example:

```
<XHQ:ModifiedComponent
name="Component_3">
<XHQ:DeletedMembersList>
<XHQ:DeletedMember
name="Member1"
/>
<XHQ:DeletedMember
name=" Member1"
/>
</XHQ:DeletedMembersList>
<XHQ:DeletedViewsList>
<XHQ:DeletedView
name="view1"
>
<XHQ:DeletedView
name="view2"
/>
<XHQ:DeletedView
name="view3"
/>
</XHQ:DeletedViewsList>
</XHQ:ModifiedComponent>
```

The components deletion syntax can be used either in a XHQ standard model import/export XML file, in conjunction with the rest of the XHQ role supported element tags, or in a separate XML file that contains only the deleted roles tag elements. In both cases the name of the XML file remains unchanged "model.xml" and is part of a zip file.

Here is a simple example of a valid "model.xml" file that can be used, through XHQ model import (using the XHQ Workbench UI or XHQCI role import command), to delete XHQ components, members and their associated views:

```
<XHQ:Model
   xmlns:XHQ="http://www.siemens.com/indx"
   esHost="localhost"
   esTime="1255051173"
   esFormattedTime="Oct 8, 2009 6:19:33 PM PDT"
   xhqVersion="4.0"
   xhqBuild="211">
   <XHQ:DeletedComponent
      name="xpak1"
   />
   <XHQ:DeletedComponent
      name="xpak2"
   />
   <XHQ:DeletedComponent
      name="xpak3"
   />
   <XHQ:DeletedComponent
      name="Tata"
   />
   <XHQ:ModifiedComponent
      name="comp2">
      <XHQ:DeletedMembersList>
         <XHQ:DeletedMember
            name="memdf"
         />
         <XHQ:DeletedMember
```

```
            name="asda"
         />
         <XHQ:DeletedMember
            name="asdfsd"
         />
      </XHQ:DeletedMembersList>
   </XHQ:ModifiedComponent>
   <XHQ:ModifiedComponent
      name="Component_3">
      <XHQ:DeletedMembersList>
         <XHQ:DeletedMember
            name="dsf"
         />
         <XHQ:DeletedMember
            name="dfasd"
         />
      </XHQ:DeletedMembersList>
      <XHQ:DeletedViewsList>
         <XHQ:DeletedView
            name="view1"
         />
         <XHQ:DeletedView
            name="view2"
         />
         <XHQ:DeletedView
            name="view3"
         />
      </XHQ:DeletedViewsList>
   </XHQ:ModifiedComponent>
   <XHQ:ModifiedComponent
      name="comp1">
      <XHQ:DeletedMembersList>
         <XHQ:DeletedMember
            name="mem1"
         />
         <XHQ:DeletedMember
            name="merm2"
         />
         <XHQ:DeletedMember
            name="mem3"
         />
      </XHQ:DeletedMembersList>
   </XHQ:ModifiedComponent>
   <XHQ:ModifiedComponent
      name="para3">
      <XHQ:DeletedViewsList>
         <XHQ:DeletedView
            name="borvie.putoi"
         />
         <XHQ:DeletedView
            name="DFASFASF"
         />
      </XHQ:DeletedViewsList>
   </XHQ:ModifiedComponent>
   <XHQ:ModifiedComponent
      name="para2">
      <XHQ:DeletedViewsList>
         <XHQ:DeletedView
            name="454"
         />
         <XHQ:DeletedView
```

```
            name="454"
         />
         <XHQ:DeletedView
            name="34"
         />
      </XHQ:DeletedViewsList>
   </XHQ:ModifiedComponent>
</XHQ:Model>
```

Import and Export Use of the XML Delete Syntax

For Export

There are two ways (scenarios) in which a user can export or generate XML with "delete" syntax:

- Use the **`xhqci`** commands;
- Use the **XHQ Administration Utility** to execute a **Repos Comparison**.

There are three `xhqci` commands which can generate and export XML for deleting components, collections and roles:

- `exportdelmodel`
- `exportdelroles`
- `exportdelsoln`

For more information, go to the topic, *Using the xhqci Utility*, located in the XHQ Administrator's Guide.

The most convenient way to generate the delete XML syntax is by executing a **Repos Comparison** using the **XHQ Administration Utility**.

For more information, go to the topic, *The XHQ Administration Utility*, located in the XHQ Backup and Recovery Guide.

For Import

Using the XHQ Workbench, XHQ Solution Builder, or `xhqci` commands to import the XML that contains the delete XML syntax for model components, roles, and collections.

For more information, go to the topic, *Using the xhqci Utility*, located in the XHQ Administrator's Guide.

Using the Command-line to Import/Export

XHQ allows you to import or export using a DOS command-line. This is done by running the **`xhqci.bat`** *batch file* through a DOS command prompt. Batch files primarily consist of a group (or batch) of DOS commands that, when executed sequentially, perform a certain task.

For the list of commands that involve the component model and component role, go to the section, *Using the xhqci Utility*, located in the XHQ Administrator's Guide.

See also the topic, *DTD Validation Structure*.

By running the **`xhqci.bat`**, you can perform the **export/import** of the following:

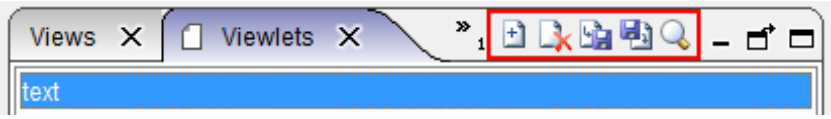
- Solution Configurations
- Model Configurations
- Component Roles
- Connector Definitions
- Connector Processes
- Connector Groups
- Solution Definitions
- View Tile Composer Views
- Units of Measure

As well as the **export** of the following:

- Connection Groups to be Deleted
- Connection Processes to be Deleted
- Model to be Deleted
- Role Mappings to be Deleted
- Solution Collections to be Deleted
- Database

Shapes, Viewlets, Mappings Export and Import

Both VTC Shapes and Mappings export/import options are on the XHQ Workbench **File menu**. Viewlets export/import options are found on the **Viewlets Panel Editor**.



Viewlets Panel Editor

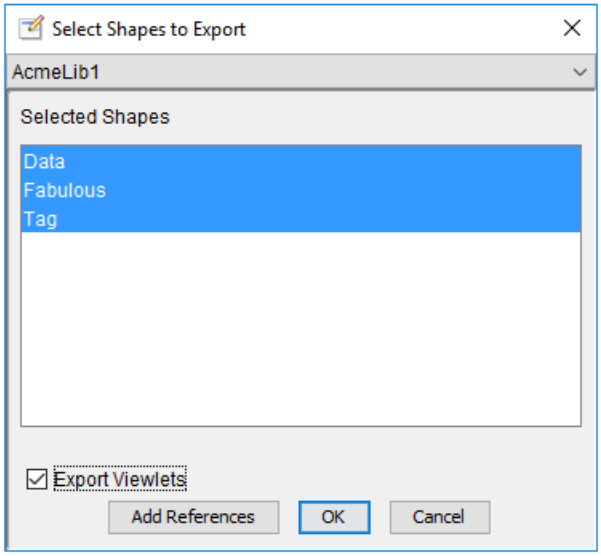
To export shapes

1. From the **File menu**, click **Export Shapes** and confirm the export by clicking **Yes**.
   The "Select an export file" dialog appears.
2. For **Save in**, click the **DOWN arrow** and locate the directory to which you want to save the ZIP export file.

   Best-Practice: Naming the Export File

   Name the export file the same as the library from which the shape is being exported. This makes it easy when importing these shapes into a different library.

3. Enter a **File name** and click **Save**.
   The "Select Shapes to Export" dialog appears.
4. At the top left of the dialog, click the **DOWN arrow** and **select the Library** from which you want to export.
5. In the Selected Shapes box, select (to **highlight**) the shape file(s).

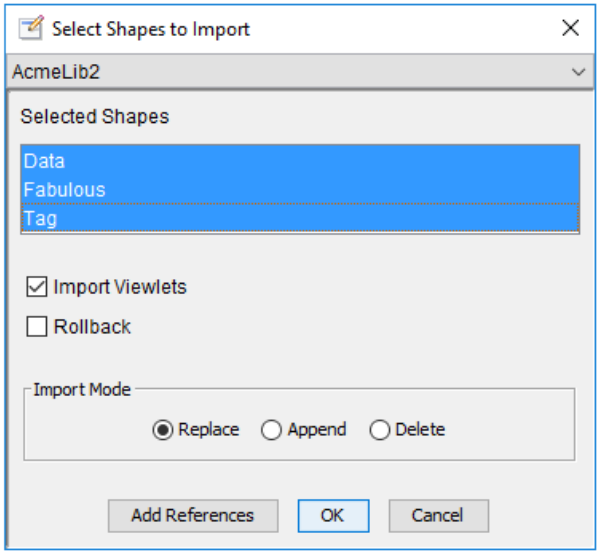


6. Check **Export Viewlets** and click **OK**.

To import shapes to a library

1. Select the library to which you want to import the shape.
2. From the **File menu**, click **Import Shapes** and confirm the import by clicking **Yes**.
   The "Select an import file" dialog appears.
3. For **Look in**, click the **DOWN arrow** and locate the exported ZIP file.

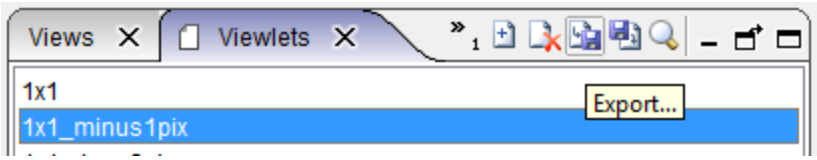
4. Select the ZIP file and click **Open**.
   The "Select Shapes to Import" dialog appears.
5. At the top left of the dialog, click the **DOWN arrow** and **select the Library** to which you want to import.
6. In the Selected Shapes box, select (to **highlight**) the shape file(s).



7. Check **Import Viewlets** and click **OK**.

To export viewlets

1. From the Shapes tab, select the shape from which you want to export the viewlet.
   This shows the list of viewlets on the Viewlets tab.
2. From the Viewlets tab, select the viewlet to export.
3. From the **Viewlets Panel Editor**, click the **Export** icon.



   The "Export Viewlet" dialog appears.
4. For **Save in**, click the **DOWN arrow** and locate the directory to which you want to save the .VIEW export file.
5. Click **Save**.
   A message appears verifying where the .VIEW export file was saved.
6. Click **OK**.

To import viewlets

There are two ways to import viewlets:

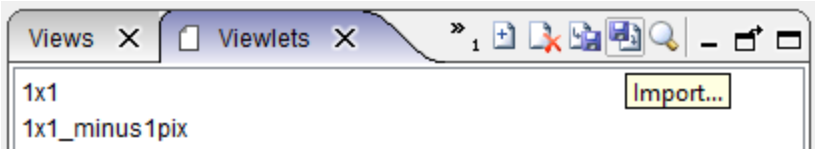
- From the Shapes tab, right-clicking on the shape from which you want to import the viewlet and , from the shortcut menu, click Import Viewlet.

or

- Select the shape to which you want to import the viewlet and, from the Viewlets Panel Editor, click Import icon.

The following procedure uses the latter of the two ways.

1. From the Shapes tab, select the shape to which you want to import the viewlet.
   This shows the list of viewlets on the Viewlets tab.
2. From the **Viewlets Panel Editor**, click the **Import** icon.



   The "Import Viewlet" dialog appears.
3. For **Look in**, click the **DOWN arrow** and locate the exported .VIEW file.

   VIEW files with the following view elements are not imported:
   - Keyed views
   - View instances
   - ENO tables
4. Select the .VIEW file and click **Open**.
   The "Import Viewlet" dialog appears.
5. *Optional*
   Rename the viewlet and click OK.
   A message appears verifying a successful import.

The newly imported viewlet is listed on the Viewlets tab.

To export mappings

1. From the **File menu**, click **Export Mappings**.
   The "Select an export file" dialog appears.
2. For **Save in**, click the **DOWN arrow** and locate the directory to which you want to save the XML export file.
3. Enter a **File name** and click **Save**.

To import mappings

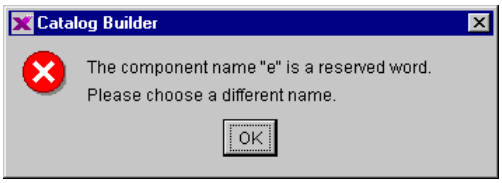
1. From the **File menu**, click **Import Mappings** .
   The "Select an import file" dialog appears.
2. For **Look in**, click the **DOWN arrow** and locate the exported XML file.
3. Select the XML file and click **Open**.

Appendices

- A - Reserved Words
- B - Supported Characters
- C - Application Techniques
- D - DTD Validation Structure
- E - XML to Preserve Collection Links
- F - Chart Mapping and Data Sets
- G - Shortcut Keys
-

A - Reserved Words

The following lists are words that should be avoided in **component, component member, view,** and **exposed collection node** names because they are *reserved words*, words that have a specific meaning to XHQ. If you use a reserved word, you will receive a notification error asking you to choose a different name.



Reserved Word Error

Reserved Words: Component and View Names

Component and View Names Reserved Words			
aux	com1	com2	com3
com4	com5	com6	com7
com8	com9	con	lpt
lpt1	lpt2	lpt3	lpt4
lpt5	lpt6	lpt7	lpt8
lpt9	nul	prn	prn1
prn2	prn3	prn4	

Reserved Words: Component Member Names

Component Member Names Reserved Words			
absolute	action	add	all
allocate	alter	and	any
are	as	asc	assertion
at	authorization	aux	avg
bad	begin	between	bigint
bit	bit_length	boolean	both
by	call	cascade	cascaded
case	cast	catalog	char
char_length	character	character_length	check
clike	close	coalesce	collate
collation	column	com1	com2

Component Member Names Reserved Words			
com3	com4	com5	com6
com7	com8	com9	commit
con	connect	connection	constraint
constraints	continue	convert	corresponding
count	create	cross	current
current_date	current_time	current_timestamp	current_user
cursor	date	day	deallocate
dec	decimal	declare	default
deferrable	deferred	delete	desc
describe	diagnostics	disconnect	distinct
domain	double	drop	e
else	end	endexec	escape
except	exception	exec	execute
exists	explain	external	extract
false	fetch	first	float
for	foreign	found	from
full	get	getcurrentconnection	global
go	good	goto	grant
group	having	hour	identity
immediate	in	index	indicator
initially	inner	input	insensitive
insert	int	integer	intersect
into	is	isolation	join
key	language	last	leading
left	level	like	limit
local	long	longint	lower
lpt	lpt1	lpt2	lpt3
lpt4	lpt5	lpt6	lpt7
lpt8	lpt9	ltrim	match
max	min	minute	money
month	national	natural	nchar
next	no	not	nul

Component Member Names Reserved Words			
null	nullif	numeric	nvarchar
octet_length	of	on	online
only	open	option	or
order	outer	output	overlaps
pad	partial	pi	prepare
preserve	primary	prior	privileges
prn	prn1	prn2	prn3
prn4	procedure	public	read
real	references	relative	restrict
revoke	right	rlike	rollback
rollup	rows	rtrim	runtimestatistics
schema	scroll	second	select
sequence	session	session_user	set
size	slike	smallint	some
space	sql	sqlcode	sqlerror
sqlstate	start	step	substr
substring	sum	system_user	table
temporary	text	then	this
time	timestamp	timezone_hour	timezone_minute
tinyint	to	trailing	transaction
translate	translation	trim	true
type	uid	uncertain	union
unique	unknown	update	upper
usage	user	using	value
values	varchar	varying	view
when	whenever	where	with
work	write	year	zone

Reserved Symbols

The following are symbols that should be avoided in **component, component member,** and **exposed collection node** names.

Reserved Symbols	
[(left bracket)	] (right bracket)

B - Supported Characters

The following is the list of supported characters that can be used in alias references. If the identifier is not quoted, it must start with an alphabetic character followed by any of the following:

- An alphanumeric
- An underscore
- A period

XHQ Supported Characters

Character Description	Symbol
Upper case letters	**A** through **Z**
Lower case letters	**a** through **z**
Digits	**0** through **9**
Comma	,
Period	.
Semicolon	;
Colon	:
Tilde	~
Question Mark	?
Exclamation Point	!
At Sign	@
Pound (or Number) Sign	#
Dollar Sign	$
Percent Sign	%
Caret	^
Less Than	<
Greater Than	>
Ampersand	&
Asterisk	*
Left (or Open) Parenthesis	(
Right (or Close) Parenthesis	)
Underscore	_
Dash (or Hyphen or Minus Sign)	-
Plus Sign	+
Equals Sign	=

Character Description	Symbol
Forward Slash (or Division Sign)	/
Double Quotes	“
Single Quote (or Apostrophe)	‘ or ’ **Note:** If a forward/open single quote is used to bracket the string, then the string can include a backward/close quote, and vice versa.
Space (or Blank)	

C - Application Techniques

Improving Collection Queries

Previously, in XHQ solutions, you can populate a combo box with a column in a filtered collection by creating a pseudo-hidden collection view (by using a standard technique that creates an XHQ view called `zWhiteSpace` in which a white square blends in with the background). This pseudo-hidden collection view is configured with proper filtering criteria and then it's collection view variable (for example, `$View1`) is linked to the combo box.

The problem with this approach is that the pseudo-hidden collection view triggers a collection query at runtime, which can cause performance issues. You can set the "Show" animation to "false" on the collection view (causing it to be hidden) so as not to trigger the collection query. But the filtering criteria defined on the now hidden collection view will not be applied to the combo box.

To fix this problem, XHQ now suppresses the excess collection queries when a combo box is associated with filter criteria.

When a collection view is hidden, the evaluation of the filtering criteria is now broadcasted to other interested parties like the combo box or aggregate functions. This addresses the combo box issue and will also address the same issue when using aggregate functions such as `$view1.count()` to count the collection records with some filter criteria.

Sorting should never be applied to the "hidden" collection view. Sorting defined on the hidden collection view does not have any effect on the sorting order of the content displayed in the combo box. The combo box always overrides the sorting criteria defined on the collection view before it queries the collection. The same also applies to aggregate functions but, in this case, the XHQ server discards/removes the sort when executing the aggregate functions. Aggregate functions only use the WHERE clause defined in the criteria.

Querying GOOD Criteria

The following technique can be used to force the collection to query only when all criteria are good. This can be useful for some of the slower populating criteria.

Add a **`Show property`** to the collection view that displays only when the criterion is good.

Example: `Show = Quality($ComboBox1)==GOOD`

The query only fires if the ComboBox has a value and could be applied when the value of the ComboBox is non-editable.

Using the Collection Data Combo Box

A *collection data combo box* is a combo box that is populated by a column in a given collection. To configure a collection data combo box, you can simply select a collection member from the collection drop-down list in the `ComboBox` tool tab of the XHQ Workbench.

For more information, go to the topic *Using the Combo Box*, located in the "Working with XHQ Workbench Tools" chapter of this guide.

To apply filtering criteria to the content in the drop-down, you must first create a collection view. Then, define all the filtering criteria on the collection view.

Next, link the collection view to the combo box by selecting the collection view variable (for example, `$View1`) from the collection drop-down list in the `ComboBox` tool tab of the XHQ Workbench. As an option, you can set the "Show" animation to "false" on the collection view to hide the collection view during runtime.

If the collection view is hidden during runtime, it will not trigger the collection query, but all criteria defined on the collection are applied to the combo box when the combo box queries the collection.

Using a Collection Aggregate Method

A similar technique can also be applied to the collection aggregate method on which filter criteria are to be applied. For example, instead of using `<collection>.count()`, use `$View1.count()`, where `View1` is the collection view that has all filtering criteria defined. Again, set the "Show" animation to "false" on the collection view if you want to hide the collection view during runtime.

In this case, if the collection view is hidden during runtime, it will not trigger the collection query. But all criteria defined on the collection are applied to the collection aggregate method when the aggregate method sends a collection rollup request.

D - DTD Validation Structure

Things to Note

- The **DTD validator lines**, which are located at the very beginning of each XHQ XML file allows validating the XML structure and syntax in two scenarios:
  a. At the import back time in XHQ;
  b. With any XML IDE that supports DTD validation (i.e. Altova, Microsoft DTD editor, etc.) to validate the XML before trying to import the data into XHQ.
- **Namespaces** are not needed in the export/import XML and, because they cannot be validated, are not supported in the DTD. To fix an XSL that is causing a parsing error, explicitly exclude namespaces by using the **`exclude-result-prefixes`** attribute in `xsl:styesheet` or `xsl:transform` elements.

Format: `exclude-result-prefixes="namespace_1 namespace_2 ... namespace_n"`

Each namespace listed is separated by a blank space.

Example:

```
<xsl:transform version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:XHQ="http://www.siemens.com/indx"
xmlns:msxsl="urn:schemas-microsoft-com:xslt"
xmlns:SITBUFn=<http://www.indxhq.com>
exclude-result-prefixes="SITBUFn msxsl XHQ">
```

The Solution DTD

The DTD (Document Type Definition) for the solution definition XML configuration is as follows. This DTD defines the document structure of the XML document by providing a list of valid elements and attributes.

```
<!ELEMENT XHQ:SolutionConfig
((XHQ:Collection|XHQ:Component|XHQ:Group|XHQ:DataMember|XHQ:ComponentMember|XHQ:GroupMemb
er|XHQ:CollectionMember)*,XHQ:DeletedCollection*)>
<!ATTLIST XHQ:SolutionConfig xmlns:XHQ       CDATA #IMPLIED>
<!ATTLIST XHQ:SolutionConfig ssHost          CDATA #IMPLIED>
<!ATTLIST XHQ:SolutionConfig rootName        CDATA #IMPLIED>
<!ATTLIST XHQ:SolutionConfig solutionRevID   CDATA #IMPLIED>
<!ATTLIST XHQ:SolutionConfig ssTime          CDATA #IMPLIED>
<!ATTLIST XHQ:SolutionConfig ssFormattedTime CDATA #IMPLIED>
<!ATTLIST XHQ:SolutionConfig xhqVersion      CDATA #IMPLIED>
<!ATTLIST XHQ:SolutionConfig xhqBuild        CDATA #IMPLIED>

<!ELEMENT XHQ:Collection (XHQ:ConnectProp*,XHQ:CacheProp*,XHQ:CacheIndexList*)>
<!ATTLIST XHQ:Collection xhqPath CDATA        #REQUIRED>
<!ATTLIST XHQ:Collection xhqComponent CDATA     #REQUIRED>
<!ATTLIST XHQ:Collection usage CDATA        #IMPLIED>
<!ATTLIST XHQ:Collection tagCollection (true|false)        #IMPLIED>
<!ATTLIST XHQ:Collection roles CDATA        #IMPLIED>
<!ATTLIST XHQ:Collection writes CDATA        #IMPLIED>
<!ATTLIST XHQ:Collection connectionGroup CDATA        #IMPLIED>

<!ELEMENT XHQ:Component EMPTY>
<!ATTLIST XHQ:Component xhqPath CDATA             #REQUIRED>
<!ATTLIST XHQ:Component xhqComponent CDATA            #REQUIRED>
<!ATTLIST XHQ:Component usage CDATA            #IMPLIED>
```

```
<!ATTLIST XHQ:Component roles CDATA        #IMPLIED>

<!ELEMENT XHQ:Group EMPTY>
<!ATTLIST XHQ:Group xhqPath CDATA             #REQUIRED>
<!ATTLIST XHQ:Group xhqComponent CDATA            #REQUIRED>
<!ATTLIST XHQ:Group usage CDATA            #IMPLIED>
<!ATTLIST XHQ:Group roles CDATA        #IMPLIED>

<!ELEMENT XHQ:DataMember (XHQ:HistorianConnectProp*,XHQ:ConnectProp*)>
<!ATTLIST XHQ:DataMember xhqPath CDATA        #REQUIRED>
<!ATTLIST XHQ:DataMember xhqComponent CDATA     #REQUIRED>
<!ATTLIST XHQ:DataMember xhqMember CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember usage CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember alias CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember datacontrolOptimizeAuto (true|false)        #IMPLIED>
<!ATTLIST XHQ:DataMember datacontrolAllowWrite (true|false)        #IMPLIED>
<!ATTLIST XHQ:DataMember datacontrolConstrainMinimum (true|false)        #IMPLIED>
<!ATTLIST XHQ:DataMember datacontrolMinimum CDATA       #IMPLIED>
<!ATTLIST XHQ:DataMember datacontrolConstrainMaximum (true|false)        #IMPLIED>
<!ATTLIST XHQ:DataMember datacontrolMaximum CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember datacontrolConstrainMask (true|false)        #IMPLIED>
<!ATTLIST XHQ:DataMember datacontrolMask CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember expression CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember expressionSchedulerEnabled (true|false)        #IMPLIED>
<!ATTLIST XHQ:DataMember expressionScheduleBaseTime CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember expressionSchedulePeriod CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember roles CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember writes CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember historian CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember historianRecord (true|false)        #IMPLIED>
<!ATTLIST XHQ:DataMember historianCompression CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember historianDeviation CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember historianMinTime CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember historianMaxTime CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember historianName CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember historianConnectGroup CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember connectionGroup CDATA        #IMPLIED>

<!ELEMENT XHQ:ComponentMember EMPTY>
<!ATTLIST XHQ:ComponentMember xhqPath CDATA             #REQUIRED>
<!ATTLIST XHQ:ComponentMember linkPath CDATA        #IMPLIED>
<!ATTLIST XHQ:ComponentMember xhqComponent CDATA            #REQUIRED>
<!ATTLIST XHQ:ComponentMember usage CDATA            #IMPLIED>
<!ATTLIST XHQ:ComponentMember roles CDATA        #IMPLIED>

<!ELEMENT XHQ:GroupMember EMPTY>
<!ATTLIST XHQ:GroupMember xhqPath CDATA             #REQUIRED>
<!ATTLIST XHQ:GroupMember xhqComponent CDATA            #REQUIRED>
<!ATTLIST XHQ:GroupMember usage CDATA            #IMPLIED>
<!ATTLIST XHQ:GroupMember roles CDATA        #IMPLIED>

<!ELEMENT XHQ:CollectionMember EMPTY>
<!ATTLIST XHQ:CollectionMember xhqPath CDATA             #REQUIRED>
<!ATTLIST XHQ:CollectionMember xhqComponent CDATA            #REQUIRED>
<!ATTLIST XHQ:CollectionMember usage CDATA            #IMPLIED>
<!ATTLIST XHQ:CollectionMember roles CDATA        #IMPLIED>
<!ATTLIST XHQ:CollectionMember linkCollection CDATA        #IMPLIED>
<!ATTLIST XHQ:CollectionMember linkConstraint CDATA        #IMPLIED>

<!ELEMENT XHQ:ConnectProp EMPTY>
<!ATTLIST XHQ:ConnectProp name CDATA             #REQUIRED>
```

```
<!ATTLIST XHQ:ConnectProp value CDATA            #REQUIRED>

<!ELEMENT XHQ:HistorianConnectProp EMPTY>
<!ATTLIST XHQ:HistorianConnectProp name CDATA             #REQUIRED>
<!ATTLIST XHQ:HistorianConnectProp value CDATA            #REQUIRED>

<!ELEMENT XHQ:CacheProp EMPTY>
<!ATTLIST XHQ:CacheProp name CDATA             #REQUIRED>
<!ATTLIST XHQ:CacheProp cacheMaxLength CDATA            #IMPLIED>
<!ATTLIST XHQ:CacheProp cachePrecision CDATA            #IMPLIED>
<!ATTLIST XHQ:CacheProp cacheScale CDATA            #IMPLIED>
<!ATTLIST XHQ:CacheProp subscribable (true|false)            #IMPLIED>
<!ATTLIST XHQ:CacheProp cacheUniqueKey (true|false)            #IMPLIED>

<!ELEMENT XHQ:CacheIndexList (XHQ:CacheIndexListMember*)>
<!ATTLIST XHQ:CacheIndexList name CDATA        #REQUIRED>

<!ELEMENT XHQ:CacheIndexListMember EMPTY>
<!ATTLIST XHQ:CacheIndexListMember name CDATA        #REQUIRED>
<!ATTLIST XHQ:CacheIndexListMember sort CDATA        #REQUIRED>

<!ELEMENT XHQ:DeletedCollection EMPTY>
<!ATTLIST XHQ:DeletedCollection name CDATA            #REQUIRED>
```

The Model DTD

The DTD for the model definition XML configuration is as follows. This DTD defines the document structure of the XML document by providing a list of valid elements and attributes.

```
<!ELEMENT XHQ:Model (XHQ:Component*,(XHQ:ModifiedComponent|XHQ:DeletedComponent)*)>
<!ATTLIST XHQ:Model xmlns:XHQ       CDATA #IMPLIED>
<!ATTLIST XHQ:Model esHost          CDATA #IMPLIED>
<!ATTLIST XHQ:Model esTime          CDATA #IMPLIED>
<!ATTLIST XHQ:Model esFormattedTime CDATA #IMPLIED>
<!ATTLIST XHQ:Model xhqVersion      CDATA #IMPLIED>
<!ATTLIST XHQ:Model xhqBuild        CDATA #IMPLIED>

<!ELEMENT XHQ:Component ((XHQ:ComponentRef|XHQ:CollectionRef|XHQ:Group|XHQ:boolean|
XHQ:integer|XHQ:long|XHQ:float|XHQ:double|XHQ:decimal|
XHQ:datetime| XHQ:interval|XHQ:string|XHQ:text)*)>
<!ATTLIST XHQ:Component name CDATA        #REQUIRED>
<!ATTLIST XHQ:Component extends CDATA     #IMPLIED>
<!ATTLIST XHQ:Component page CDATA        #IMPLIED>
<!ATTLIST XHQ:Component desc CDATA        #IMPLIED>

<!ELEMENT XHQ:ComponentRef EMPTY>
<!ATTLIST XHQ:ComponentRef name CDATA             #REQUIRED>
<!ATTLIST XHQ:ComponentRef usage CDATA            #IMPLIED>
<!ATTLIST XHQ:ComponentRef type CDATA             #REQUIRED>
<!ATTLIST XHQ:ComponentRef hidden (true|false)    "false">
<!ATTLIST XHQ:ComponentRef tagAttribute (None)       #IMPLIED>

<!ELEMENT XHQ:CollectionRef EMPTY>
<!ATTLIST XHQ:CollectionRef name CDATA             #REQUIRED>
<!ATTLIST XHQ:CollectionRef usage CDATA            #IMPLIED>
<!ATTLIST XHQ:CollectionRef type CDATA             #REQUIRED>
<!ATTLIST XHQ:CollectionRef hidden (true|false)    "false">
```

```
<!ATTLIST XHQ:CollectionRef tagAttribute (None)       #IMPLIED>

<!ELEMENT XHQ:Group EMPTY>
<!ATTLIST XHQ:Group name CDATA             #REQUIRED>
<!ATTLIST XHQ:Group usage CDATA            #IMPLIED>
<!ATTLIST XHQ:Group type CDATA             #REQUIRED>
<!ATTLIST XHQ:Group hidden (true|false)    "false">

<!ELEMENT XHQ:boolean EMPTY>
<!ATTLIST XHQ:boolean name CDATA             #REQUIRED>
<!ATTLIST XHQ:boolean usage CDATA            #IMPLIED>
<!ATTLIST XHQ:boolean hidden (true|false)     "false">
<!ATTLIST XHQ:boolean tagAttribute (None|Value) "None">

<!ELEMENT XHQ:integer EMPTY>
<!ATTLIST XHQ:integer name CDATA             #REQUIRED>
<!ATTLIST XHQ:integer usage CDATA            #IMPLIED>
<!ATTLIST XHQ:integer hidden (true|false)     "false">
<!ATTLIST XHQ:integer tagAttribute (None|Value) "None">

<!ELEMENT XHQ:long EMPTY>
<!ATTLIST XHQ:long name CDATA             #REQUIRED>
<!ATTLIST XHQ:long usage CDATA            #IMPLIED>
<!ATTLIST XHQ:long hidden (true|false)     "false">
<!ATTLIST XHQ:long tagAttribute (None|Value) "None">

<!ELEMENT XHQ:float EMPTY>
<!ATTLIST XHQ:float name CDATA             #REQUIRED>
<!ATTLIST XHQ:float usage CDATA            #IMPLIED>
<!ATTLIST XHQ:float hidden (true|false)     "false">
<!ATTLIST XHQ:float tagAttribute (None|Value) "None">

<!ELEMENT XHQ:double EMPTY>
<!ATTLIST XHQ:double name CDATA             #REQUIRED>
<!ATTLIST XHQ:double usage CDATA            #IMPLIED>
<!ATTLIST XHQ:double hidden (true|false)     "false">
<!ATTLIST XHQ:double tagAttribute (None|Value) "None">

<!ELEMENT XHQ:decimal EMPTY>
<!ATTLIST XHQ:decimal name CDATA             #REQUIRED>
<!ATTLIST XHQ:decimal usage CDATA            #IMPLIED>
<!ATTLIST XHQ:decimal hidden (true|false)     "false">
<!ATTLIST XHQ:decimal tagAttribute (None|Value) "None">

<!ELEMENT XHQ:datetime EMPTY>
<!ATTLIST XHQ:datetime name CDATA             #REQUIRED>
<!ATTLIST XHQ:datetime usage CDATA            #IMPLIED>
<!ATTLIST XHQ:datetime hidden (true|false)     "false">
<!ATTLIST XHQ:datetime tagAttribute (None|Value) "None">

<!ELEMENT XHQ:interval EMPTY>
<!ATTLIST XHQ:interval name CDATA             #REQUIRED>
<!ATTLIST XHQ:interval usage CDATA            #IMPLIED>
<!ATTLIST XHQ:interval hidden (true|false)     "false">
<!ATTLIST XHQ:interval tagAttribute (None|Value) "None">

<!ELEMENT XHQ:string EMPTY>
<!ATTLIST XHQ:string name CDATA             #REQUIRED>
<!ATTLIST XHQ:string usage CDATA            #IMPLIED>
<!ATTLIST XHQ:string hidden (true|false)     "false">
<!ATTLIST XHQ:string tagAttribute (None|Value|Config|Description|Key|Units) "None">
```

```
<!ELEMENT XHQ:text EMPTY>
<!ATTLIST XHQ:text name CDATA             #REQUIRED>
<!ATTLIST XHQ:text usage CDATA            #IMPLIED>
<!ATTLIST XHQ:text hidden (true|false)     "false">
<!ATTLIST XHQ:text tagAttribute (None|Value|Config|Description|Key|Units) "None">

<!ELEMENT XHQ:ModifiedComponent (XHQ:DeletedMembersList?,XHQ:DeletedViewsList?)>
<!ATTLIST XHQ:ModifiedComponent name CDATA        #REQUIRED>

<!ELEMENT XHQ:DeletedMembersList (XHQ:DeletedMember+)>

<!ELEMENT XHQ:DeletedViewsList (XHQ:DeletedView+)>

<!ELEMENT XHQ:DeletedMember EMPTY>
<!ATTLIST XHQ:DeletedMember name CDATA             #REQUIRED>

<!ELEMENT XHQ:DeletedView EMPTY>
<!ATTLIST XHQ:DeletedView name CDATA             #REQUIRED>

<!ELEMENT XHQ:DeletedComponent EMPTY>
<!ATTLIST XHQ:DeletedComponent name CDATA             #REQUIRED>
```

Component DTD (used in the "model.xml") validator

```
<!ELEMENT XHQ:Model (XHQ:Component*,(XHQ:ModifiedComponent|XHQ:DeletedComponent)*)>
<!ATTLIST XHQ:Model xmlns:XHQ       CDATA #IMPLIED>
<!ATTLIST XHQ:Model esHost          CDATA #IMPLIED>
<!ATTLIST XHQ:Model esTime          CDATA #IMPLIED>
<!ATTLIST XHQ:Model esFormattedTime CDATA #IMPLIED>
<!ATTLIST XHQ:Model xhqVersion      CDATA #IMPLIED>
<!ATTLIST XHQ:Model xhqBuild        CDATA #IMPLIED>

<!ELEMENT XHQ:Component
((XHQ:ComponentRef|XHQ:CollectionRef|XHQ:Group|XHQ:boolean|XHQ:integer|XHQ:long|XHQ:float
|XHQ:double|XHQ:decimal|XHQ:datetime|XHQ:double|XHQ:interval|XHQ:string|XHQ:text)*)>
<!ATTLIST XHQ:Component name CDATA        #REQUIRED>
<!ATTLIST XHQ:Component extends CDATA     #IMPLIED>
<!ATTLIST XHQ:Component page CDATA        #IMPLIED>
<!ATTLIST XHQ:Component desc CDATA        #IMPLIED>

<!ELEMENT XHQ:ComponentRef EMPTY>
<!ATTLIST XHQ:ComponentRef name CDATA             #REQUIRED>
<!ATTLIST XHQ:ComponentRef usage CDATA            #IMPLIED>
<!ATTLIST XHQ:ComponentRef type CDATA             #REQUIRED>
<!ATTLIST XHQ:ComponentRef hidden (true|false)    "false">
<!ATTLIST XHQ:ComponentRef tagAttribute (None)       #IMPLIED>

<!ELEMENT XHQ:CollectionRef EMPTY>
<!ATTLIST XHQ:CollectionRef name CDATA             #REQUIRED>
<!ATTLIST XHQ:CollectionRef usage CDATA            #IMPLIED>
<!ATTLIST XHQ:CollectionRef type CDATA             #REQUIRED>
<!ATTLIST XHQ:CollectionRef hidden (true|false)    "false">
<!ATTLIST XHQ:CollectionRef tagAttribute (None)       #IMPLIED>

<!ELEMENT XHQ:Group EMPTY>
<!ATTLIST XHQ:Group name CDATA             #REQUIRED>
<!ATTLIST XHQ:Group usage CDATA            #IMPLIED>
```

```
<!ATTLIST XHQ:Group type CDATA             #REQUIRED>
<!ATTLIST XHQ:Group hidden (true|false)    "false">

<!ELEMENT XHQ:boolean EMPTY>
<!ATTLIST XHQ:boolean name CDATA             #REQUIRED>
<!ATTLIST XHQ:boolean usage CDATA            #IMPLIED>
<!ATTLIST XHQ:boolean hidden (true|false)     "false">
<!ATTLIST XHQ:boolean tagAttribute (None|Value) "None">

<!ELEMENT XHQ:integer EMPTY>
<!ATTLIST XHQ:integer name CDATA             #REQUIRED>
<!ATTLIST XHQ:integer usage CDATA            #IMPLIED>
<!ATTLIST XHQ:integer hidden (true|false)     "false">
<!ATTLIST XHQ:integer tagAttribute (None|Value) "None">

<!ELEMENT XHQ:long EMPTY>
<!ATTLIST XHQ:long name CDATA             #REQUIRED>
<!ATTLIST XHQ:long usage CDATA            #IMPLIED>
<!ATTLIST XHQ:long hidden (true|false)     "false">
<!ATTLIST XHQ:long tagAttribute (None|Value) "None">

<!ELEMENT XHQ:float EMPTY>
<!ATTLIST XHQ:float name CDATA             #REQUIRED>
<!ATTLIST XHQ:float usage CDATA            #IMPLIED>
<!ATTLIST XHQ:float hidden (true|false)     "false">
<!ATTLIST XHQ:float tagAttribute (None|Value) "None">

<!ELEMENT XHQ:double EMPTY>
<!ATTLIST XHQ:double name CDATA             #REQUIRED>
<!ATTLIST XHQ:double usage CDATA            #IMPLIED>
<!ATTLIST XHQ:double hidden (true|false)     "false">
<!ATTLIST XHQ:double tagAttribute (None|Value) "None">

<!ELEMENT XHQ:decimal EMPTY>
<!ATTLIST XHQ:decimal name CDATA             #REQUIRED>
<!ATTLIST XHQ:decimal usage CDATA            #IMPLIED>
<!ATTLIST XHQ:decimal hidden (true|false)     "false">
<!ATTLIST XHQ:decimal tagAttribute (None|Value) "None">

<!ELEMENT XHQ:datetime EMPTY>
<!ATTLIST XHQ:datetime name CDATA             #REQUIRED>
<!ATTLIST XHQ:datetime usage CDATA            #IMPLIED>
<!ATTLIST XHQ:datetime hidden (true|false)     "false">
<!ATTLIST XHQ:datetime tagAttribute (None|Value) "None">

<!ELEMENT XHQ:interval EMPTY>
<!ATTLIST XHQ:interval name CDATA             #REQUIRED>
<!ATTLIST XHQ:interval usage CDATA            #IMPLIED>
<!ATTLIST XHQ:interval hidden (true|false)     "false">
<!ATTLIST XHQ:interval tagAttribute (None|Value) "None">

<!ELEMENT XHQ:string EMPTY>
<!ATTLIST XHQ:string name CDATA             #REQUIRED>
<!ATTLIST XHQ:string usage CDATA            #IMPLIED>
<!ATTLIST XHQ:string hidden (true|false)     "false">
<!ATTLIST XHQ:string tagAttribute (None|Value|Config|Description|Key|Units) "None">

<!ELEMENT XHQ:text EMPTY>
<!ATTLIST XHQ:text name CDATA             #REQUIRED>
<!ATTLIST XHQ:text usage CDATA            #IMPLIED>
<!ATTLIST XHQ:text hidden (true|false)     "false">
```

```
<!ATTLIST XHQ:text tagAttribute (None|Value|Config|Description|Key|Units) "None">

<!ELEMENT XHQ:ModifiedComponent (XHQ:DeletedMembersList?,XHQ:DeletedViewsList?)>
<!ATTLIST XHQ:ModifiedComponent name CDATA        #REQUIRED>

<!ELEMENT XHQ:DeletedMembersList (XHQ:DeletedMember+)>

<!ELEMENT XHQ:DeletedViewsList (XHQ:DeletedView+)>

<!ELEMENT XHQ:DeletedMember EMPTY>
<!ATTLIST XHQ:DeletedMember name CDATA             #REQUIRED>

<!ELEMENT XHQ:DeletedView EMPTY>
<!ATTLIST XHQ:DeletedView name CDATA             #REQUIRED>

<!ELEMENT XHQ:DeletedComponent EMPTY>
<!ATTLIST XHQ:DeletedComponent name CDATA             #REQUIRED>
```

Also, the XML element syntax for components has been changes in order to enable DTD validation (the old XML tax syntax did not allow such validation) and also to allow numerical names or names that start with numbers for the XHQ Workbench components. Each component element is now called <XHQ:Component> and also a "name" attribute has been added to the structure.

Example:

```
<XHQ:Component
      name="comp1"
      extends=""
      page="Perf Mgt"
      desc="Test componet">
.
.
.
</XHQ:Component>
```

DTD validator for the XHQ Views

```
<!ELEMENT XHQ:Animation
((XHQ:DrawItem|XHQ:ComponentItem|XHQ:TemplateItem|XHQ:KeyedItem|XHQ:ButtonItem|XHQ:ComboB
oxItem|XHQ:EditBoxItem|XHQ:BrowserItem|XHQ:GroupItem|XHQ:TextAreaItem|XHQ:TrendItem|XHQ:V
alueItem)*)>
<!ATTLIST XHQ:Animation xmlns:XHQ CDATA       #IMPLIED>
<!ATTLIST XHQ:Animation esHost CDATA             #IMPLIED>
<!ATTLIST XHQ:Animation esTime CDATA             #IMPLIED>
<!ATTLIST XHQ:Animation xhqVersion CDATA         #IMPLIED>
<!ATTLIST XHQ:Animation xhqBuild CDATA           #IMPLIED>
<!ATTLIST XHQ:Animation class_id CDATA            #IMPLIED>
<!ATTLIST XHQ:Animation flow_view (true|false)    #REQUIRED>
<!ATTLIST XHQ:Animation drillable (true|false)    #REQUIRED>
<!ATTLIST XHQ:Animation navigable (true|false)    #REQUIRED>
<!ATTLIST XHQ:Animation default (true|false)      #REQUIRED>
<!ATTLIST XHQ:Animation pop_up (true|false)       #IMPLIED>
<!ATTLIST XHQ:Animation view_tile CDATA           #IMPLIED>
<!ATTLIST XHQ:Animation view_name CDATA           #REQUIRED>
<!ATTLIST XHQ:Animation comment CDATA       #IMPLIED>
```

```
<!ELEMENT XHQ:DrawItem (XHQ:Variable?,
(XHQ:BreakpointAnimation|XHQ:PrecisionAnimation|XHQ:RangeAnimation|XHQ:ValueAnimation|XHQ
:ValueTagAnimation)*,(XHQ:ShowLink|XHQ:UrlLink|XHQ:ViewLink)
*,XHQ:Scale?,XHQ:Lock?,XHQ:TooltipData?)>
<!ATTLIST XHQ:DrawItem id CDATA           #REQUIRED>

<!ELEMENT XHQ:ComponentItem (XHQ:Variable,
(XHQ:BreakpointAnimation|XHQ:PrecisionAnimation|XHQ:RangeAnimation|XHQ:ValueAnimation|XHQ
:ValueTagAnimation)*,(XHQ:ShowLink|XHQ:UrlLink|XHQ:ViewLink)
*,XHQ:Scale,XHQ:Lock,XHQ:TooltipData?,XHQ:eComponent?,XHQ:Expression?,XHQ:Criteria*)>
<!ATTLIST XHQ:ComponentItem id CDATA           #REQUIRED>
<!ATTLIST XHQ:ComponentItem ini_bound_width CDATA           #REQUIRED>
<!ATTLIST XHQ:ComponentItem ini_bound_height CDATA           #REQUIRED>
<!ATTLIST XHQ:ComponentItem view_name CDATA           #REQUIRED>
<!ATTLIST XHQ:ComponentItem comp_name CDATA           #REQUIRED>
<!ATTLIST XHQ:ComponentItem variable CDATA           #IMPLIED>
<!ATTLIST XHQ:ComponentItem drill_state (true|false)       #IMPLIED>
<!ATTLIST XHQ:ComponentItem apply_trigger_to_initial_population (true|false)
#IMPLIED>
<!ATTLIST XHQ:ComponentItem rows CDATA           #REQUIRED>
<!ATTLIST XHQ:ComponentItem columns CDATA           #REQUIRED>
<!ATTLIST XHQ:ComponentItem use_browse_buttons (true|false)       #REQUIRED>
<!ATTLIST XHQ:ComponentItem layout (true|false)       #REQUIRED>

<!ELEMENT XHQ:TemplateItem (XHQ:Variable,
(XHQ:BreakpointAnimation|XHQ:PrecisionAnimation|XHQ:RangeAnimation|XHQ:ValueAnimation|XHQ
:ValueTagAnimation)*,(XHQ:ShowLink|XHQ:UrlLink|XHQ:ViewLink)
*,XHQ:Scale,XHQ:Lock,XHQ:TooltipData?,XHQ:ComponentViewMember*)>
<!ATTLIST XHQ:TemplateItem id CDATA           #REQUIRED>
<!ATTLIST XHQ:TemplateItem drill_state (true|false)       #IMPLIED>
<!ATTLIST XHQ:TemplateItem view_name CDATA           #REQUIRED>
<!ATTLIST XHQ:TemplateItem comp_name CDATA           #REQUIRED>
<!ATTLIST XHQ:TemplateItem variable CDATA           #IMPLIED>

<!ELEMENT XHQ:KeyedItem (XHQ:Variable?,
(XHQ:BreakpointAnimation|XHQ:PrecisionAnimation|XHQ:RangeAnimation|XHQ:ValueAnimation|XHQ
:ValueTagAnimation)*,(XHQ:ShowLink|XHQ:UrlLink|XHQ:ViewLink)
*,XHQ:Scale?,XHQ:Lock?,XHQ:TooltipData?,XHQ:KeyedValue?)>
<!ATTLIST XHQ:KeyedItem id CDATA           #REQUIRED>
<!ATTLIST XHQ:KeyedItem drill_state (true|false)       #IMPLIED>
<!ATTLIST XHQ:KeyedItem view_name CDATA           #REQUIRED>
<!ATTLIST XHQ:KeyedItem comp_name CDATA           #REQUIRED>
<!ATTLIST XHQ:KeyedItem variable CDATA           #IMPLIED>

<!ELEMENT XHQ:ButtonItem (XHQ:Variable,
(XHQ:BreakpointAnimation|XHQ:PrecisionAnimation|XHQ:RangeAnimation|XHQ:ValueAnimation|XHQ
:ValueTagAnimation)*,(XHQ:ShowLink|XHQ:UrlLink|XHQ:ViewLink)
*,XHQ:Scale,XHQ:Lock,XHQ:TooltipData?,XHQ:Expression?)>
<!ATTLIST XHQ:ButtonItem id CDATA           #REQUIRED>

<!ELEMENT XHQ:ComboBoxItem (XHQ:ValueListItem*,XHQ:Variable,
(XHQ:BreakpointAnimation|XHQ:PrecisionAnimation|XHQ:RangeAnimation|XHQ:ValueAnimation|XHQ
:ValueTagAnimation)*,(XHQ:ShowLink|XHQ:UrlLink|XHQ:ViewLink)
*,XHQ:Scale,XHQ:Lock,XHQ:TooltipData?,XHQ:CollectionDataType?)>
<!ATTLIST XHQ:ComboBoxItem id CDATA           #REQUIRED>
<!ATTLIST XHQ:ComboBoxItem datatype (value_list|collection_data)       #REQUIRED>
<!ATTLIST XHQ:ComboBoxItem editable (true|false)       #REQUIRED>
<!ATTLIST XHQ:ComboBoxItem name_type
(integer|real|string|boolean|long|double|decimal|datetime|interval|text)  #REQUIRED>
```

```
<!ELEMENT XHQ:EditBoxItem (XHQ:Variable,
(XHQ:BreakpointAnimation|XHQ:PrecisionAnimation|XHQ:RangeAnimation|XHQ:ValueAnimation|XHQ
:ValueTagAnimation)*,(XHQ:ShowLink|XHQ:UrlLink|XHQ:ViewLink)
*,XHQ:Scale,XHQ:Lock,XHQ:TooltipData?)>
<!ATTLIST XHQ:EditBoxItem id CDATA           #REQUIRED>
<!ATTLIST XHQ:EditBoxItem name_type
(integer|real|string|boolean|long|double|decimal|datetime|interval|text)  #REQUIRED>

<!ELEMENT XHQ:BrowserItem (XHQ:Variable,
(XHQ:BreakpointAnimation|XHQ:PrecisionAnimation|XHQ:RangeAnimation|XHQ:ValueAnimation|XHQ
:ValueTagAnimation)*,(XHQ:ShowLink|XHQ:UrlLink|XHQ:ViewLink)
*,XHQ:Scale,XHQ:Lock,XHQ:TooltipData?,browser_trigger_expr?,browser_url_expr?)>
<!ATTLIST XHQ:BrowserItem id CDATA           #REQUIRED>

<!ELEMENT XHQ:GroupItem (XHQ:Variable?,
(XHQ:BreakpointAnimation|XHQ:PrecisionAnimation|XHQ:RangeAnimation|XHQ:ValueAnimation|XHQ
:ValueTagAnimation)*,(XHQ:ShowLink|XHQ:UrlLink|XHQ:ViewLink)
*,XHQ:Scale?,XHQ:Lock?,XHQ:TooltipData?,
(XHQ:DrawItem|XHQ:ComponentItem|XHQ:TemplateItem|XHQ:KeyedItem|XHQ:ButtonItem|XHQ:ComboBo
xItem|XHQ:EditBoxItem|XHQ:BrowserItem|XHQ:GroupItem|XHQ:TextAreaItem|XHQ:TrendItem|XHQ:Va
lueItem)*)>
<!ATTLIST XHQ:GroupItem id CDATA           #REQUIRED>
<!ATTLIST XHQ:GroupItem ini_bound_width CDATA           #IMPLIED>
<!ATTLIST XHQ:GroupItem ini_bound_height CDATA           #IMPLIED>

<!ELEMENT XHQ:TextAreaItem (XHQ:Variable,
(XHQ:BreakpointAnimation|XHQ:PrecisionAnimation|XHQ:RangeAnimation|XHQ:ValueAnimation|XHQ
:ValueTagAnimation)*,(XHQ:ShowLink|XHQ:UrlLink|XHQ:ViewLink)
*,XHQ:Scale,XHQ:Lock,XHQ:TooltipData?)>
<!ATTLIST XHQ:TextAreaItem id CDATA           #REQUIRED>
<!ATTLIST XHQ:TextAreaItem uncertain_color CDATA     #REQUIRED>
<!ATTLIST XHQ:TextAreaItem bad_color CDATA           #REQUIRED>
<!ATTLIST XHQ:TextAreaItem vertical_scrollbar (true|false)           #REQUIRED>

<!ELEMENT XHQ:TrendItem (XHQ:Variable,
(XHQ:BreakpointAnimation|XHQ:PrecisionAnimation|XHQ:RangeAnimation|XHQ:ValueAnimation|XHQ
:ValueTagAnimation)*,(XHQ:ShowLink|XHQ:UrlLink|XHQ:ViewLink)
*,XHQ:Scale,XHQ:Lock,XHQ:TooltipData?,XHQ:Pen*,XHQ:EndTime?,XHQ:StartTime?,XHQ:Duration?,
XHQ:Title?,XHQ:Y_Label?,XHQ:Y2_Label?,common_min?,common_max?,right_common_min?,right_
common_max?,XHQ:TickIncrement?,XHQ:LabelAtEveryXTicks?)>
<!ATTLIST XHQ:TrendItem id CDATA           #REQUIRED>
<!ATTLIST XHQ:TrendItem unit (weeks|days|hours|minutes|seconds)  #REQUIRED>
<!ATTLIST XHQ:TrendItem aggregate_avg CDATA           #REQUIRED>
<!ATTLIST XHQ:TrendItem show_symbols (true|false)          #REQUIRED>
<!ATTLIST XHQ:TrendItem show_right_scale (true|false)          #IMPLIED>
<!ATTLIST XHQ:TrendItem show_left_scale (true|false)          #IMPLIED>
<!ATTLIST XHQ:TrendItem right_scale_format CDATA          #IMPLIED>
<!ATTLIST XHQ:TrendItem left_scale_format CDATA          #IMPLIED>
<!ATTLIST XHQ:TrendItem min_right_margin CDATA          #IMPLIED>
<!ATTLIST XHQ:TrendItem min_left_margin CDATA          #IMPLIED>
<!ATTLIST XHQ:TrendItem mode (Time|XY)          #REQUIRED>
<!ATTLIST XHQ:TrendItem show_date (true|false)          #REQUIRED>
<!ATTLIST XHQ:TrendItem show_time (true|false)          #REQUIRED>
<!ATTLIST XHQ:TrendItem hide_horizontal_grid_line (true|false)          #IMPLIED>
<!ATTLIST XHQ:TrendItem hide_vertical_grid_line (true|false)          #IMPLIED>
<!ATTLIST XHQ:TrendItem hide_xaxis_ticks (true|false)          #IMPLIED>
<!ATTLIST XHQ:TrendItem hide_yaxis_ticks (true|false)          #IMPLIED>
<!ATTLIST XHQ:TrendItem border_color CDATA          #IMPLIED>
<!ATTLIST XHQ:TrendItem left_scale_color CDATA          #IMPLIED>
<!ATTLIST XHQ:TrendItem right_scale_color CDATA          #IMPLIED>
```

```
<!ATTLIST XHQ:TrendItem left_scale_label_color CDATA         #IMPLIED>
<!ATTLIST XHQ:TrendItem right_scale_label_color CDATA         #IMPLIED>
<!ATTLIST XHQ:TrendItem title_color CDATA         #IMPLIED>
<!ATTLIST XHQ:TrendItem x_axis_color CDATA         #IMPLIED>
<!ATTLIST XHQ:TrendItem date_format CDATA           #REQUIRED>
<!ATTLIST XHQ:TrendItem time_format CDATA           #REQUIRED>
<!ATTLIST XHQ:TrendItem left_scale_log_type (Linear|Log10|Log)  #IMPLIED>
<!ATTLIST XHQ:TrendItem right_scale_log_type (Linear|Log10|Log)  #IMPLIED>
<!ATTLIST XHQ:TrendItem right_scale (all|none|common-auto|common-manual)  #IMPLIED>
<!ATTLIST XHQ:TrendItem scale (all|none|common-auto|common-manual)  #REQUIRED>

<!ELEMENT XHQ:ValueItem (XHQ:Variable,
(XHQ:BreakpointAnimation|XHQ:PrecisionAnimation|XHQ:RangeAnimation|XHQ:ValueAnimation|XHQ
:ValueTagAnimation)*,(XHQ:ShowLink|XHQ:UrlLink|XHQ:ViewLink)
*,XHQ:Scale,XHQ:Lock,XHQ:TooltipData?)>
<!ATTLIST XHQ:ValueItem id CDATA           #REQUIRED>
<!ATTLIST XHQ:ValueItem uncertain_color CDATA           #REQUIRED>
<!ATTLIST XHQ:ValueItem bad_color CDATA           #REQUIRED>
<!ATTLIST XHQ:ValueItem show_date (true|false)         #REQUIRED>
<!ATTLIST XHQ:ValueItem show_time (true|false)         #REQUIRED>
<!ATTLIST XHQ:ValueItem show_quality (true|false)         #REQUIRED>
<!ATTLIST XHQ:ValueItem show_value (true|false)         #REQUIRED>
<!ATTLIST XHQ:ValueItem decimal_places CDATA           #REQUIRED>
<!ATTLIST XHQ:ValueItem group_digits (true|false)         #REQUIRED>
<!ATTLIST XHQ:ValueItem value_layout (time_top|time_bottom|time_right|time_left)
#REQUIRED>
<!ATTLIST XHQ:ValueItem quality_layout (quality_left|quality_right)  #REQUIRED>
<!ATTLIST XHQ:ValueItem time_format CDATA           #REQUIRED>
<!ATTLIST XHQ:ValueItem date_format CDATA           #REQUIRED>
<!ATTLIST XHQ:ValueItem time_format_datetime CDATA           #REQUIRED>
<!ATTLIST XHQ:ValueItem date_format_datetime CDATA           #REQUIRED>
<!ATTLIST XHQ:ValueItem value_format CDATA           #REQUIRED>

<!ELEMENT XHQ:Pen (XHQ:Description?,XHQ:Value?,XHQ:Min?,XHQ:Max?,XHQ:MasterPen?)>
<!ATTLIST XHQ:Pen thickness CDATA     #REQUIRED>
<!ATTLIST XHQ:Pen style CDATA     #REQUIRED>
<!ATTLIST XHQ:Pen symbol CDATA           #REQUIRED>
<!ATTLIST XHQ:Pen interpolation_interval CDATA           #IMPLIED>
<!ATTLIST XHQ:Pen aggregate_type (Fits|Average|Min|Max|Interpolate|Raw)  #REQUIRED>
<!ATTLIST XHQ:Pen scale_format CDATA     #REQUIRED>
<!ATTLIST XHQ:Pen value_format CDATA     #REQUIRED>
<!ATTLIST XHQ:Pen scale_precision CDATA     #REQUIRED>
<!ATTLIST XHQ:Pen value_precision CDATA     #REQUIRED>
<!ATTLIST XHQ:Pen tolerance CDATA     #REQUIRED>
<!ATTLIST XHQ:Pen interpolation_type (default|linear|stepped)  #IMPLIED>
<!ATTLIST XHQ:Pen scale_assigment (left|right)  #IMPLIED>
<!ATTLIST XHQ:Pen future_data (true|false)     #REQUIRED>
<!ATTLIST XHQ:Pen display_values (true|false)     #REQUIRED>
<!ATTLIST XHQ:Pen color CDATA     #IMPLIED>

<!ELEMENT XHQ:Description (XHQ:Expression)>

<!ELEMENT XHQ:Value (XHQ:Expression)>

<!ELEMENT XHQ:Min (XHQ:Expression)>

<!ELEMENT XHQ:Max (XHQ:Expression)>

<!ELEMENT XHQ:MasterPen (XHQ:Expression)>

<!ELEMENT XHQ:EndTime (XHQ:Expression)>
```

```
<!ELEMENT XHQ:StartTime (XHQ:Expression)>

<!ELEMENT XHQ:Duration (XHQ:Expression)>

<!ELEMENT XHQ:Title (XHQ:Expression)>

<!ELEMENT XHQ:Y_Label (XHQ:Expression)>

<!ELEMENT XHQ:Y2_Label (XHQ:Expression)>

<!ELEMENT common_min (XHQ:Expression)>

<!ELEMENT common_max (XHQ:Expression)>

<!ELEMENT right_common_min (XHQ:Expression)>

<!ELEMENT right_common_max (XHQ:Expression)>

<!ELEMENT XHQ:TickIncrement (XHQ:Expression)>

<!ELEMENT XHQ:LabelAtEveryXTicks (XHQ:Expression)>

<!ELEMENT browser_trigger_expr (XHQ:Expression)>
<!ATTLIST browser_trigger_expr apply_trigger_to_initial_load (true|false)
#REQUIRED>

<!ELEMENT browser_url_expr (XHQ:Expression)>

<!ELEMENT XHQ:KeyedValue (XHQ:Expression)>

<!ELEMENT XHQ:eComponent (XHQ:Expression)>

<!ELEMENT XHQ:Criteria (XHQ:Expression*)>
<!ATTLIST XHQ:Criteria name CDATA           #REQUIRED>
<!ATTLIST XHQ:Criteria relation CDATA           #REQUIRED>

<!ELEMENT XHQ:ValueListItem EMPTY>
<!ATTLIST XHQ:ValueListItem label CDATA           #REQUIRED>
<!ATTLIST XHQ:ValueListItem value CDATA           #REQUIRED>

<!ELEMENT XHQ:ComponentViewMember (XHQ:Expression?)>
<!ATTLIST XHQ:ComponentViewMember name CDATA           #REQUIRED>
<!ATTLIST XHQ:ComponentViewMember xmlns:XHQ CDATA           #IMPLIED>

<!ELEMENT XHQ:CollectionDataType (XHQ:Expression*)>
<!ATTLIST XHQ:CollectionDataType sortdescending (true|false)       #IMPLIED>

<!ELEMENT XHQ:Variable (XHQ:Expression?)>
<!ATTLIST XHQ:Variable name CDATA           #REQUIRED>
<!ATTLIST XHQ:Variable name_type CDATA           #REQUIRED>
<!ATTLIST XHQ:Variable xmlns:XHQ CDATA           #IMPLIED>

<!ELEMENT XHQ:Scale EMPTY>
<!ATTLIST XHQ:Scale enable_horizontal_scale (true|false)       #REQUIRED>
<!ATTLIST XHQ:Scale enable_vertical_scale (true|false)       #REQUIRED>
<!ATTLIST XHQ:Scale xmlns:XHQ CDATA           #IMPLIED>

<!ELEMENT XHQ:Lock EMPTY>
<!ATTLIST XHQ:Lock enable_lock (true|false)       #REQUIRED>
<!ATTLIST XHQ:Lock xmlns:XHQ CDATA           #IMPLIED>
```

```
<!ELEMENT XHQ:Expression EMPTY>
<!ATTLIST XHQ:Expression data CDATA       #IMPLIED>

<!ELEMENT XHQ:TooltipData (XHQ:TooltipItem*)>
<!ATTLIST XHQ:TooltipData auto_word_wrap (true|false)       #REQUIRED>
<!ATTLIST XHQ:TooltipData max_char_num_per_line CDATA       #REQUIRED>

<!ELEMENT XHQ:TooltipItem (XHQ:TooltipItemValueExpr?)>
<!ATTLIST XHQ:TooltipItem name CDATA           #REQUIRED>

<!ELEMENT XHQ:TooltipItemValueExpr (XHQ:Expression)>

<!ELEMENT XHQ:BreakpointAnimation
(XHQ:Expression,XHQ:Breakpoint*,XHQ:BreakpointProperty*)>
<!ATTLIST XHQ:BreakpointAnimation property CDATA           #REQUIRED>

<!ELEMENT XHQ:Breakpoint EMPTY>
<!ATTLIST XHQ:Breakpoint type (double)       #REQUIRED>
<!ATTLIST XHQ:Breakpoint value CDATA       #REQUIRED>

<!ELEMENT XHQ:BreakpointProperty EMPTY>
<!ATTLIST XHQ:BreakpointProperty type (color)       #REQUIRED>
<!ATTLIST XHQ:BreakpointProperty value CDATA       #REQUIRED>

<!ELEMENT XHQ:PrecisionAnimation (XHQ:Expression)>
<!ATTLIST XHQ:PrecisionAnimation property CDATA           #REQUIRED>

<!ELEMENT XHQ:RangeAnimation (XHQ:Expression)>
<!ATTLIST XHQ:RangeAnimation low_value CDATA           #REQUIRED>
<!ATTLIST XHQ:RangeAnimation low_prop_value CDATA           #REQUIRED>
<!ATTLIST XHQ:RangeAnimation high_value CDATA           #REQUIRED>
<!ATTLIST XHQ:RangeAnimation high_prop_value CDATA           #REQUIRED>
<!ATTLIST XHQ:RangeAnimation property CDATA           #REQUIRED>

<!ELEMENT XHQ:ValueAnimation (XHQ:Expression)>
<!ATTLIST XHQ:ValueAnimation property CDATA           #REQUIRED>

<!ELEMENT XHQ:ValueTagAnimation (XHQ:Expression,XHQ:KeyedValue?)>
<!ATTLIST XHQ:ValueTagAnimation comp_name CDATA           #REQUIRED>
<!ATTLIST XHQ:ValueTagAnimation property CDATA           #REQUIRED>

<!ELEMENT XHQ:ShowLink (XHQ:Expression?)>
<!ATTLIST XHQ:ShowLink link_name CDATA           #REQUIRED>
<!ATTLIST XHQ:ShowLink default CDATA           #REQUIRED>

<!ELEMENT XHQ:UrlLink (XHQ:Expression?)>
<!ATTLIST XHQ:UrlLink link_name CDATA           #REQUIRED>
<!ATTLIST XHQ:UrlLink default CDATA           #REQUIRED>
<!ATTLIST XHQ:UrlLink description CDATA           #REQUIRED>

<!ELEMENT XHQ:ViewLink (XHQ:Expression?)>
<!ATTLIST XHQ:ViewLink link_name CDATA           #REQUIRED>
<!ATTLIST XHQ:ViewLink default CDATA           #REQUIRED>
<!ATTLIST XHQ:ViewLink mode (0|1|2|3|4|5)       #REQUIRED>
<!ATTLIST XHQ:ViewLink auto_close_parent_popup (0|1)       #REQUIRED>
<!ATTLIST XHQ:ViewLink custom_popup_position (true|false)       #REQUIRED>
<!ATTLIST XHQ:ViewLink x CDATA           #IMPLIED>
<!ATTLIST XHQ:ViewLink y CDATA           #IMPLIED>
<!ATTLIST XHQ:ViewLink view_name CDATA           #REQUIRED>
```

Role Mapping DTD

The DTD for the Role Mapping (security) XML configuration is as follows.

```
<!ELEMENT XHQ:RoleMap (XHQ:RoleMapEntry?)>
<!ATTLIST XHQ:RoleMap xmlns:XHQ       CDATA #IMPLIED
                      esHost          CDATA #IMPLIED
                      esTime          CDATA #IMPLIED
                      esFormattedTime CDATA #IMPLIED
                      xhqVersion      CDATA #IMPLIED
                      xhqBuild        CDATA #IMPLIED
                      securityMode    (off|classic|integrated) "off">

<!ELEMENT XHQ:RoleMapEntry (XHQ:Right?)>
<!ATTLIST XHQ:RoleMapEntry name  CDATA        #REQUIRED
                           group CDATA        #REQUIRED
                           local (true|false) #REQUIRED>

<!ELEMENT XHQ:Right EMPTY>
<!ATTLIST XHQ:Right name (catalog_editing|component_editing|view_building|solution_
editing|connector_administration|global_collection_administration|model_
importing|solution_importing|role_assigning) #REQUIRED
                    grant (true|false) #REQUIRED>
```

Solution Definition DTD

The DTD for the Solution Definition XML configuration is as follows.

```
<!ELEMENT XHQ:SolutionDefList (XHQ:SolutionDef?)>
<!ATTLIST XHQ:SolutionDefList xmlns:XHQ       CDATA #IMPLIED
                              esHost          CDATA #IMPLIED
                              esTime          CDATA #IMPLIED
                              esFormattedTime CDATA #IMPLIED
                              xhqVersion      CDATA #IMPLIED
                              xhqBuild        CDATA #IMPLIED>

<!ELEMENT XHQ:SolutionDef EMPTY>
<!ATTLIST XHQ:SolutionDef name     CDATA #REQUIRED
                          descr    CDATA #IMPLIED
                          solnHost CDATA #IMPLIED
                          top      CDATA #REQUIRED>
```

Connector Definition DTD

The DTD for the Connector Definition XML configuration is as follows.

```
<!ELEMENT XHQ:ConnectorDefList (XHQ:ConnectorDef*)>
<!ATTLIST XHQ:ConnectorDefList xmlns:XHQ CDATA #IMPLIED>
<!ATTLIST XHQ:ConnectorDefList ssHost CDATA #IMPLIED>
<!ATTLIST XHQ:ConnectorDefList ssTime CDATA #IMPLIED>
<!ATTLIST XHQ:ConnectorDefList ssFormattedTime CDATA #IMPLIED>
<!ATTLIST XHQ:ConnectorDefList xhqVersion CDATA #IMPLIED>
<!ATTLIST XHQ:ConnectorDefList xhqBuild CDATA #IMPLIED>
<!ELEMENT XHQ:ConnectorDef EMPTY>
<!ATTLIST XHQ:ConnectorDef name CDATA #REQUIRED>
<!ATTLIST XHQ:ConnectorDef descr CDATA #IMPLIED>
<!ATTLIST XHQ:ConnectorDef cimClass CDATA #IMPLIED>
<!ATTLIST XHQ:ConnectorDef cimJar CDATA #IMPLIED>
<!ATTLIST XHQ:ConnectorDef cpmClass CDATA #IMPLIED>
<!ATTLIST XHQ:ConnectorDef cpmJar CDATA #IMPLIED>
<!ATTLIST XHQ:ConnectorDef type CDATA #IMPLIED>
]>
```

Connector Process DTD

The DTD for the Connector Process XML configuration is as follows.

```
<!ELEMENT XHQ:ConnectProcessList (XHQ:ConnectProcess?)>
<!ATTLIST XHQ:ConnectProcessList xmlns:XHQ       CDATA #IMPLIED
                                 ssHost          CDATA #IMPLIED
                                 ssTime          CDATA #IMPLIED
                                 ssFormattedTime CDATA #IMPLIED
                                 xhqVersion      CDATA #IMPLIED
                                 xhqBuild        CDATA #IMPLIED>

<!ELEMENT XHQ:ConnectProcess (#PCDATA)>
<!ATTLIST XHQ:ConnectProcess name     CDATA        #REQUIRED
                             descr    CDATA        #IMPLIED
                             manual   (true|false) "false"
                             console  (true|false) "false"
                             specUser (true|false) "false"
                             user     CDATA        #IMPLIED
```

```
                            password CDATA        #IMPLIED>
```

Connector Group DTD

The DTD for the Connector Group XML configuration is as follows.

```
<!ELEMENT XHQ:ConnectGroups (XHQ:Group?)>
<!ATTLIST XHQ:ConnectGroups xmlns:XHQ       CDATA #IMPLIED
                            ssHost          CDATA #IMPLIED
                            ssTime          CDATA #IMPLIED
                            ssFormattedTime CDATA #IMPLIED
                            xhqVersion      CDATA #IMPLIED
                            xhqBuild        CDATA #IMPLIED>

<!ELEMENT XHQ:ConnectGroup (XHQ:ConnectorConfig?)>
<!ATTLIST XHQ:ConnectGroup name       CDATA        #REQUIRED
                           descr      CDATA        #IMPLIED
                           definition CDATA        #REQUIRED
                           process    CDATA        #IMPLIED
                           cimClass   CDATA        #IMPLIED
                           cimJar     CDATA        #IMPLIED
                           cfgClass   CDATA        #REQUIRED>

<!ELEMENT XHQ:ConnectGroupConfig EMPTY>
<!ATTLIST XHQ:ConnectGroupConfig name      CDATA        #REQUIRED
                                 value     CDATA        #REQUIRED>
```

Units of Measure DTD

The DTD for the Units of Measure XML configuration is as follows.

```
<!ELEMENT XHQ:UnitsOfMeasure (XHQ:CategoriesList?,XHQ:UoMList?)>
<!ATTLIST XHQ:UnitsOfMeasure xmlns:XHQ       CDATA #IMPLIED>
<!ATTLIST XHQ:UnitsOfMeasure esHost          CDATA #IMPLIED>
<!ATTLIST XHQ:UnitsOfMeasure esTime          CDATA #IMPLIED>
<!ATTLIST XHQ:UnitsOfMeasure esFormattedTime CDATA #IMPLIED>
<!ATTLIST XHQ:UnitsOfMeasure xhqVersion      CDATA #IMPLIED>
<!ATTLIST XHQ:UnitsOfMeasure xhqBuild        CDATA #IMPLIED>

<!ELEMENT XHQ:CategoriesList (XHQ:Category*,XHQ:RenamedCategory*,XHQ:DeletedCategory*)>

<!ELEMENT XHQ:Category EMPTY>
<!ATTLIST XHQ:Category name CDATA        #REQUIRED>
<!ATTLIST XHQ:Category baseUnits CDATA  #REQUIRED>

<!ELEMENT XHQ:RenamedCategory EMPTY>
<!ATTLIST XHQ:RenamedCategory oldName CDATA        #REQUIRED>
<!ATTLIST XHQ:RenamedCategory newName CDATA        #REQUIRED>

<!ELEMENT XHQ:DeletedCategory EMPTY>
<!ATTLIST XHQ:DeletedCategory name CDATA        #REQUIRED>

<!ELEMENT XHQ:UoMList (XHQ:UoM*,XHQ:RenamedUoM*,XHQ:DeletedUoM*)>

<!ELEMENT XHQ:UoM EMPTY>
<!ATTLIST XHQ:UoM name CDATA        #REQUIRED>
<!ATTLIST XHQ:UoM definition CDATA  #IMPLIED>
<!ATTLIST XHQ:UoM baseUnits CDATA  #REQUIRED>
```

```
<!ATTLIST XHQ:UoM sortOrder CDATA  #IMPLIED>
<!ATTLIST XHQ:UoM factorA CDATA    #REQUIRED>
<!ATTLIST XHQ:UoM factorB CDATA    #IMPLIED>
<!ATTLIST XHQ:UoM factorC CDATA    #REQUIRED>
<!ATTLIST XHQ:UoM factorD CDATA    #IMPLIED>

<!ELEMENT XHQ:RenamedUoM EMPTY>
<!ATTLIST XHQ:RenamedUoM oldName CDATA        #REQUIRED>
<!ATTLIST XHQ:RenamedUoM newName CDATA        #REQUIRED>

<!ELEMENT XHQ:DeletedUoM EMPTY>
<!ATTLIST XHQ:DeletedUoM name CDATA        #REQUIRED>
```

DTD for a Tag Sync XML File Import

Adding the Solution Configuration DTD validation at the beginning of the tag sync file can help improve the tag sync import performance.

For an example, see the topic, *DTD for a Tag Sync XML File Import*, located in the Appendix of the XHQ Solution Design and Architecture.

E - XML: Preserving Collection Links

If you delete a collection that is configured as linked to another collection, the configuration for that collection is then broken. You must then manually reconfigure the collection to link it to the other collection in XHQ Solution Builder, or re-import the master collection.

The recommended way to do this is to export the configuration for both collections and then re-import both collections so that the re-link occurs when the master collection is imported. The following XML example demonstrates how to export both collections and re-import back in one import operation and, in turn, preserving the link.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!-- Created by XHQ Solution Builder Export on Sep 15, 2009 1:06:59 PM PDT -->
<!-- Begin DTD Validation section - DO NOT MODIFY THIS SECTION -->
<!DOCTYPE XHQ:SolutionConfig[
<!ELEMENT XHQ:SolutionConfig
(XHQ:Collection|XHQ:Component|XHQ:Group|XHQ:DataMember|XHQ:ComponentMember|XHQ:GroupMembe
r|
XHQ:CollectionMember)*>
<!ATTLIST XHQ:SolutionConfig xmlns:XHQ       CDATA #IMPLIED>
<!ATTLIST XHQ:SolutionConfig ssHost          CDATA #IMPLIED>
<!ATTLIST XHQ:SolutionConfig rootName        CDATA #IMPLIED>
<!ATTLIST XHQ:SolutionConfig solutionRevID   CDATA #IMPLIED>
<!ATTLIST XHQ:SolutionConfig ssTime          CDATA #IMPLIED>
<!ATTLIST XHQ:SolutionConfig ssFormattedTime CDATA #IMPLIED>
<!ATTLIST XHQ:SolutionConfig xhqVersion      CDATA #IMPLIED>
<!ATTLIST XHQ:SolutionConfig xhqBuild        CDATA #IMPLIED>

<!ELEMENT XHQ:Collection (XHQ:ConnectProp*,XHQ:CacheProp*,XHQ:CacheIndexList*)>
<!ATTLIST XHQ:Collection xhqPath CDATA        #REQUIRED>
<!ATTLIST XHQ:Collection xhqComponent CDATA     #REQUIRED>
<!ATTLIST XHQ:Collection usage CDATA        #IMPLIED>
<!ATTLIST XHQ:Collection tagCollection (true|false)        #IMPLIED>
<!ATTLIST XHQ:Collection roles CDATA        #IMPLIED>
<!ATTLIST XHQ:Collection writes CDATA        #IMPLIED>
<!ATTLIST XHQ:Collection connectionGroup CDATA        #IMPLIED>

<!ELEMENT XHQ:Component EMPTY>
<!ATTLIST XHQ:Component xhqPath CDATA             #REQUIRED>
<!ATTLIST XHQ:Component xhqComponent CDATA            #REQUIRED>
<!ATTLIST XHQ:Component usage CDATA            #IMPLIED>
<!ATTLIST XHQ:Component roles CDATA        #IMPLIED>

<!ELEMENT XHQ:Group EMPTY>
<!ATTLIST XHQ:Group xhqPath CDATA             #REQUIRED>
<!ATTLIST XHQ:Group xhqComponent CDATA            #REQUIRED>
<!ATTLIST XHQ:Group usage CDATA            #IMPLIED>
<!ATTLIST XHQ:Group roles CDATA        #IMPLIED>

<!ELEMENT XHQ:DataMember (XHQ:HistorianConnectProp*,XHQ:ConnectProp*)>
<!ATTLIST XHQ:DataMember xhqPath CDATA        #REQUIRED>
<!ATTLIST XHQ:DataMember xhqComponent CDATA     #REQUIRED>
<!ATTLIST XHQ:DataMember xhqMember CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember usage CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember alias CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember datacontrolOptimizeAuto (true|false)        #IMPLIED>
<!ATTLIST XHQ:DataMember datacontrolAllowWrite (true|false)        #IMPLIED>
<!ATTLIST XHQ:DataMember datacontrolConstrainMinimum (true|false)        #IMPLIED>
<!ATTLIST XHQ:DataMember datacontrolMinimum CDATA       #IMPLIED>
<!ATTLIST XHQ:DataMember datacontrolConstrainMaximum (true|false)        #IMPLIED>
```

```
<!ATTLIST XHQ:DataMember datacontrolMaximum CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember datacontrolConstrainMask (true|false)        #IMPLIED>
<!ATTLIST XHQ:DataMember datacontrolMask CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember expression CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember expressionSchedulerEnabled (true|false)        #IMPLIED>
<!ATTLIST XHQ:DataMember expressionScheduleBaseTime CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember expressionSchedulePeriod CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember roles CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember writes CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember historian CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember historianRecord (true|false)        #IMPLIED>
<!ATTLIST XHQ:DataMember historianCompression CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember historianDeviation CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember historianMinTime CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember historianMaxTime CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember historianName CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember historianConnectGroup CDATA        #IMPLIED>
<!ATTLIST XHQ:DataMember connectionGroup CDATA        #IMPLIED>

<!ELEMENT XHQ:ComponentMember EMPTY>
<!ATTLIST XHQ:ComponentMember xhqPath CDATA        #REQUIRED>
<!ATTLIST XHQ:ComponentMember linkPath CDATA        #IMPLIED>
<!ATTLIST XHQ:ComponentMember xhqComponent CDATA        #REQUIRED>
<!ATTLIST XHQ:ComponentMember usage CDATA        #IMPLIED>
<!ATTLIST XHQ:ComponentMember roles CDATA        #IMPLIED>

<!ELEMENT XHQ:GroupMember EMPTY>
<!ATTLIST XHQ:GroupMember xhqPath CDATA        #REQUIRED>
<!ATTLIST XHQ:GroupMember xhqComponent CDATA        #REQUIRED>
<!ATTLIST XHQ:GroupMember usage CDATA        #IMPLIED>
<!ATTLIST XHQ:GroupMember roles CDATA        #IMPLIED>

<!ELEMENT XHQ:CollectionMember EMPTY>
<!ATTLIST XHQ:CollectionMember xhqPath CDATA        #REQUIRED>
<!ATTLIST XHQ:CollectionMember xhqComponent CDATA        #REQUIRED>
<!ATTLIST XHQ:CollectionMember usage CDATA        #IMPLIED>
<!ATTLIST XHQ:CollectionMember roles CDATA        #IMPLIED>
<!ATTLIST XHQ:CollectionMember linkCollection CDATA        #IMPLIED>
<!ATTLIST XHQ:CollectionMember linkConstraint CDATA        #IMPLIED>

<!ELEMENT XHQ:ConnectProp EMPTY>
<!ATTLIST XHQ:ConnectProp name CDATA        #REQUIRED>
<!ATTLIST XHQ:ConnectProp value CDATA        #REQUIRED>

<!ELEMENT XHQ:HistorianConnectProp EMPTY>
<!ATTLIST XHQ:HistorianConnectProp name CDATA        #REQUIRED>
<!ATTLIST XHQ:HistorianConnectProp value CDATA        #REQUIRED>

<!ELEMENT XHQ:CacheProp EMPTY>
<!ATTLIST XHQ:CacheProp name CDATA        #REQUIRED>
<!ATTLIST XHQ:CacheProp cacheMaxLength CDATA        #IMPLIED>
<!ATTLIST XHQ:CacheProp cachePrecision CDATA        #IMPLIED>
<!ATTLIST XHQ:CacheProp cacheScale CDATA        #IMPLIED>
<!ATTLIST XHQ:CacheProp subscribable (true|false)        #IMPLIED>
<!ATTLIST XHQ:CacheProp cacheUniqueKey (true|false)        #IMPLIED>

<!ELEMENT XHQ:CacheIndexList (XHQ:CacheIndexListMember*)>
<!ATTLIST XHQ:CacheIndexList name CDATA        #REQUIRED>

<!ELEMENT XHQ:CacheIndexListMember EMPTY>
<!ATTLIST XHQ:CacheIndexListMember name CDATA        #REQUIRED>
```

```
<!ATTLIST XHQ:CacheIndexListMember sort CDATA        #REQUIRED>

]>
<!-- End DTD Validation section -->
<XHQ:SolutionConfig
   xmlns:XHQ="http://www.siemens.com/indx"
   ssHost="localhost"
   rootName="AlisoViejo"
   solutionRevID="133"
   ssTime="1253045219360"
   ssFormattedTime="Sep 15, 2009 1:06:59 PM PDT"
   xhqVersion="4.0"
   xhqBuild="217">
   <XHQ:Collection
      xhqPath="::COLLECTIONS.colAEmbedded"
      xhqComponent="MaintenanceOrder"
      usage=""
      roles="Solution Admins;Everyone"
      writes="false;false"
      connectionGroup="Odbc_60min">
      <XHQ:ConnectProp
         name="DELTA_PERIOD"
         value="int:0"
      />
      <XHQ:ConnectProp
         name="WHERE_STATEMENT"
         value="string:&quot;&quot;"
      />
      <XHQ:ConnectProp
         name="TS_REQUEST"
         value="bool:false"
      />
      <XHQ:ConnectProp
         name="USE_BUILDER"
         value="bool:true"
      />
      <XHQ:ConnectProp
         name="USE_PASS_THROUGH"
         value="bool:false"
      />
      <XHQ:ConnectProp
         name="COLUMN_NAMES"
         value="string
[]:EquipmentModuleID;EquipmentModuleName;WorkOrderStatus;SubmitDate;Symptoms;UnitName;
WorkOrderDescription;WorkOrderID;WorkOrderType"
      />
      <XHQ:ConnectProp
         name="POLL_PERIOD"
         value="int:0"
      />
      <XHQ:ConnectProp
         name="USE_CUSTOM_POLL"
         value="bool:false"
      />
      <XHQ:ConnectProp
         name="MAPPED_MEMBERS"
         value="string
[]:EquipmentModuleID;EquipmentModuleName;Status;SubmitDate;Symptoms;UnitName;
WorkOrderDescription;WorkOrderID;WorkOrderType"
      />
      <XHQ:ConnectProp
```

```
         name="TABLE_NAME"
         value="string:&quot;qryMaintenanceOrders&quot;"
      />
      <XHQ:ConnectProp
         name="USE_REFRESH"
         value="bool:false"
      />
      <XHQ:ConnectProp
         name="USE_DELTA"
         value="bool:false"
      />
      <XHQ:ConnectProp
         name="VERSION"
         value="int:1"
      />
      <XHQ:CacheProp
         name="Comments"
         cacheMaxLength="32"
         cacheUniqueKey="false"
      />
      <XHQ:CacheProp
         name="EndTime"
         cacheMaxLength="32"
         cacheUniqueKey="false"
      />
      <XHQ:CacheProp
         name="EquipmentModuleName"
         cacheMaxLength="32"
         cacheUniqueKey="false"
      />
      <XHQ:CacheProp
         name="Status"
         cacheMaxLength="32"
         cacheUniqueKey="false"
      />
      <XHQ:CacheProp
         name="SubmitDate"
         cacheMaxLength="32"
         cacheUniqueKey="false"
      />
      <XHQ:CacheProp
         name="Symptoms"
         cacheMaxLength="32"
         cacheUniqueKey="false"
      />
      <XHQ:CacheProp
         name="UnitName"
         cacheMaxLength="32"
         cacheUniqueKey="false"
      />
      <XHQ:CacheProp
         name="WorkOrderDescription"
         cacheMaxLength="32"
         cacheUniqueKey="false"
      />
      <XHQ:CacheProp
         name="WorkOrderID"
         cacheMaxLength="32"
         cacheUniqueKey="false"
      />
      <XHQ:CacheProp
```

```
      name="WorkOrderType"
      cacheMaxLength="32"
      cacheUniqueKey="false"
   />
</XHQ:Collection>
<XHQ:DataMember
   xhqPath="::COLLECTIONS.colAEmbedded.Comments"
   xhqComponent="MaintenanceOrder"
   usage=""
   roles="Solution Admins;Everyone"
   >
</XHQ:DataMember>
<XHQ:DataMember
   xhqPath="::COLLECTIONS.colAEmbedded.EquipmentModuleID"
   xhqComponent="MaintenanceOrder"
   usage=""
   roles="Solution Admins;Everyone"
   >
</XHQ:DataMember>
<XHQ:DataMember
   xhqPath="::COLLECTIONS.colAEmbedded.EndTime"
   xhqComponent="MaintenanceOrder"
   usage=""
   roles="Solution Admins;Everyone"
   >
</XHQ:DataMember>
<XHQ:DataMember
   xhqPath="::COLLECTIONS.colAEmbedded.EquipmentModuleName"
   xhqComponent="MaintenanceOrder"
   usage=""
   roles="Solution Admins;Everyone"
   >
</XHQ:DataMember>
<XHQ:DataMember
   xhqPath="::COLLECTIONS.colAEmbedded.Status"
   xhqComponent="MaintenanceOrder"
   usage=""
   roles="Solution Admins;Everyone"
   >
</XHQ:DataMember>
<XHQ:DataMember
   xhqPath="::COLLECTIONS.colAEmbedded.SubmitDate"
   xhqComponent="MaintenanceOrder"
   usage=""
   roles="Solution Admins;Everyone"
   >
</XHQ:DataMember>
<XHQ:DataMember
   xhqPath="::COLLECTIONS.colAEmbedded.Symptoms"
   xhqComponent="MaintenanceOrder"
   usage=""
   roles="Solution Admins;Everyone"
   >
</XHQ:DataMember>
<XHQ:DataMember
   xhqPath="::COLLECTIONS.colAEmbedded.UnitName"
   xhqComponent="MaintenanceOrder"
   usage=""
   roles="Solution Admins;Everyone"
   >
</XHQ:DataMember>
```

```
   <XHQ:DataMember
      xhqPath="::COLLECTIONS.colAEmbedded.WorkOrderDescription"
      xhqComponent="MaintenanceOrder"
      usage=""
      roles="Solution Admins;Everyone"
      >
   </XHQ:DataMember>
   <XHQ:DataMember
      xhqPath="::COLLECTIONS.colAEmbedded.WorkOrderID"
      xhqComponent="MaintenanceOrder"
      usage=""
      roles="Solution Admins;Everyone"
      >
   </XHQ:DataMember>
   <XHQ:DataMember
      xhqPath="::COLLECTIONS.colAEmbedded.WorkOrderType"
      xhqComponent="MaintenanceOrder"
      usage=""
      roles="Solution Admins;Everyone"
      >
   </XHQ:DataMember>
   <XHQ:Collection
         xhqPath="::COLLECTIONS.colA"
         xhqComponent="EquipmentItem"
         usage=""
         roles="Solution Admins;Everyone"
         writes="false;false"
         connectionGroup="Odbc_60min">
         <XHQ:ConnectProp
            name="DELTA_PERIOD"
            value="int:0"
         />
         <XHQ:ConnectProp
            name="WHERE_STATEMENT"
            value="string:&quot;&quot;"
         />
         <XHQ:ConnectProp
            name="TS_REQUEST"
            value="bool:false"
         />
         <XHQ:ConnectProp
            name="USE_BUILDER"
            value="bool:true"
         />
         <XHQ:ConnectProp
            name="USE_PASS_THROUGH"
            value="bool:false"
         />
         <XHQ:ConnectProp
            name="COLUMN_NAMES"
            value="string
[]:EquipmentModuleID;EquipmentModuleName;EquipmentState;EquipmentType;ReferenceURL;UnitID
;UnitName"
         />
         <XHQ:ConnectProp
            name="POLL_PERIOD"
            value="int:0"
         />
         <XHQ:ConnectProp
            name="USE_CUSTOM_POLL"
            value="bool:false"
```

```
         />
         <XHQ:ConnectProp
            name="MAPPED_MEMBERS"
            value="string
[]:EquipmentModuleID;EquipmentModuleName;EquipmentState;EquipmentType;ReferenceURL;UnitID
;UnitName"
         />
         <XHQ:ConnectProp
            name="TABLE_NAME"
            value="string:&quot;qryEquipmentEntities&quot;"
         />
         <XHQ:ConnectProp
            name="USE_REFRESH"
            value="bool:false"
         />
         <XHQ:ConnectProp
            name="USE_DELTA"
            value="bool:false"
         />
         <XHQ:ConnectProp
            name="VERSION"
            value="int:1"
         />
         <XHQ:CacheProp
            name="EquipmentModuleName"
            cacheMaxLength="32"
            cacheUniqueKey="false"
         />
         <XHQ:CacheProp
            name="EquipmentState"
            cacheMaxLength="32"
            cacheUniqueKey="false"
         />
         <XHQ:CacheProp
            name="EquipmentType"
            cacheMaxLength="32"
            cacheUniqueKey="false"
         />
         <XHQ:CacheProp
            name="ReferenceURL"
            cacheMaxLength="32"
            cacheUniqueKey="false"
         />
         <XHQ:CacheProp
            name="UnitName"
            cacheMaxLength="32"
            cacheUniqueKey="false"
         />
      </XHQ:Collection>
      <XHQ:DataMember
         xhqPath="::COLLECTIONS.colA.EquipmentModuleID"
         xhqComponent="EquipmentItem"
         usage=""
         roles="Solution Admins;Everyone"
         >
      </XHQ:DataMember>
      <XHQ:DataMember
         xhqPath="::COLLECTIONS.colA.EquipmentModuleName"
         xhqComponent="EquipmentItem"
         usage=""
         roles="Solution Admins;Everyone"
```

```
         >
      </XHQ:DataMember>
      <XHQ:DataMember
         xhqPath="::COLLECTIONS.colA.EquipmentState"
         xhqComponent="EquipmentItem"
         usage=""
         roles="Solution Admins;Everyone"
         >
      </XHQ:DataMember>
      <XHQ:DataMember
         xhqPath="::COLLECTIONS.colA.EquipmentType"
         xhqComponent="EquipmentItem"
         usage=""
         roles="Solution Admins;Everyone"
         >
      </XHQ:DataMember>
      <XHQ:DataMember
         xhqPath="::COLLECTIONS.colA.ReferenceURL"
         xhqComponent="EquipmentItem"
         usage=""
         roles="Solution Admins;Everyone"
         >
      </XHQ:DataMember>
      <XHQ:DataMember
         xhqPath="::COLLECTIONS.colA.UnitID"
         xhqComponent="EquipmentItem"
         usage=""
         roles="Solution Admins;Everyone"
         >
      </XHQ:DataMember>
      <XHQ:DataMember
         xhqPath="::COLLECTIONS.colA.UnitName"
         xhqComponent="EquipmentItem"
         usage=""
         roles="Solution Admins;Everyone"
         >
      </XHQ:DataMember>
      <XHQ:CollectionMember
         xhqPath="::COLLECTIONS.colA.MaintenanceOrders"
         xhqComponent="MaintenanceOrder"
         usage=""
         roles="Solution Admins;Everyone"
         linkCollection="::Collections.colAEmbedded"
         linkConstraint="WHERE EQUIPMENTMODULENAME = &apos;K500B&apos;"
      />
</XHQ:SolutionConfig>
```

F - Chart Mapping and Data Sets

The data processed by each chart must be properly set-up in the collection. When configuring charts (in particular, mapping columns) the data requirements are dependent on the chart type.

Data Mapping Summary

Data	Bar	Bubble	Candlestick	Gantt	Histogram	Pareto
Domain	Numeric					
Range	Numeric					
Time			Date			
Value	Numeric				Numeric	Numeric
Key						
Category	String					String
TaskName				String		
SubTask				String		
StartDate				Date		
EndDate				Date		
%Complete				Numeric		
X		Numeric (X-value)				
Y		Numeric (Y-value)				
Z		Numeric (diameter)				
High			Numeric			
Low			Numeric			
Open			Numeric			
Close			Numeric			
Volume			Numeric			
Annotation	String		String	String		String
Tooltip	String	String	String	String		String
Color	String	String				String
Label Color	String/Integer	String/Integer		String/Integer		String/Integer
Label Font	String	String		String		String
Parent Category						

Data Mapping Summary - Continued

Data	Pie	Polar	Spider Web	Time Series	Treemap	Waterfall	XY
Domain		Numeric (angular, in degrees)					Numeric
Range		Numeric (radial position)					Numeric
Time				Date			
Value	Numeric		Numeric	Numeric		Numeric	
Key	String						
Category			String		String	String	
TaskName							
SubTask							
StartDate							
EndDate							
%Complete							
X							
Y							
Z							
High							
Low							
Open							
Close							
Volume							
Annotation				String		String	String
Item					String		
Box Size					Numeric		
Shade					Numeric		
Exception					Integer		
Tooltip	String	String	String	String	String	String	String
Color	String (hex colors) Integer	String	String	String			String

Data	Pie	Polar	Spider Web	Time Series	Treemap	Waterfall	XY
	(integer colors)						
Label Color				String/Integer		String/Integer	String/Integer
Label Font				String		String	String
Parent Category					String		

For 3D Charts, go to *About 3D Charts*.

For the Bar Chart

The Bar chart requires **Categorical** or **Numerical** data.

Categorical Data

Data Mapping

Column	Data Type
Value	Numeric
Category	String
OPTIONAL Annotation	String

Data Set Example – Categorical Data

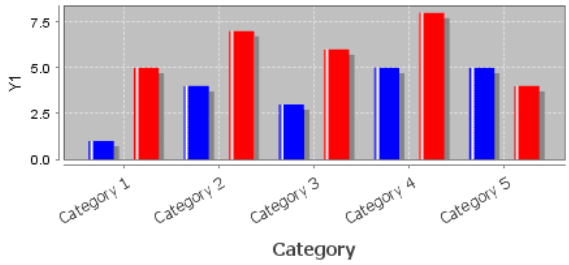
Given the following data set with **two series**:

Bar Chart Categorical Data - Series ABar Chart Categorical Data - Series B

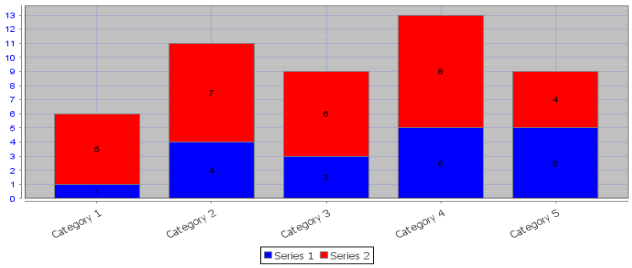
Value A	Category A	Value B	Category B
1	Category 1	5	Category 1
4	Category 2	7	Category 2
3	Category 3	6	Category 3
5	Category 4	8	Category 4
5	Category 5	4	Category 5

Results in the following chart.

With **Render** set to **Bar3D**:



With **Render** set to **BarStacked** (and both series belong to the same group):



Numerical Data

Data Mapping

Column	Data Type
Domain	Numeric
Range	Numeric
OPTIONAL Annotation	String

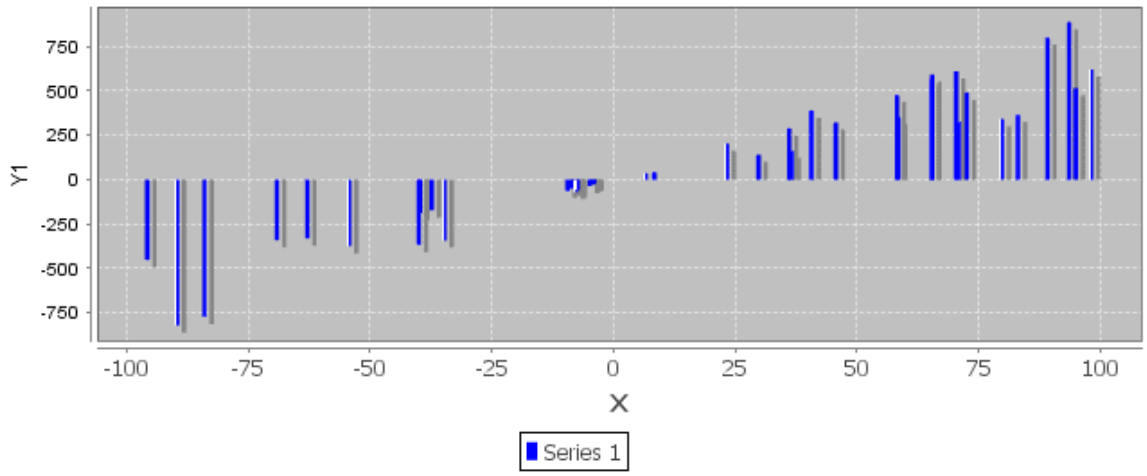
Data Set Example – Numerical Data

Given the following data set for a **single** series:

Bar Chart Numerical DataBar Chart Numerical Data - Continued

Domain	Range	Domain	Range
40.69	389.61	58.51	353.77
-34.74	-342.69	29.8	141.14
-95.81	-452.89	70.41	610.68
89.15	801.03	-54.31	-375.05
94.95	514.91	23.27	202.56
36.59	162.36	-62.92	-331.35
65.4	567.1	-9.47	-62.31
-4.74	-31.35	-84.06	-773.11
-39.74	-187.56	-4.93	-34.49

Results in the following chart:



For the Bubble Chart

The Bubble chart requires the following columns for data mapping.

Data Mapping

Column	Data Type
X	Numeric The **X-value** of the bubble.
Y	Numeric The **Y-value** of the bubble.
Z	Numeric The **diameter** of the bubble.

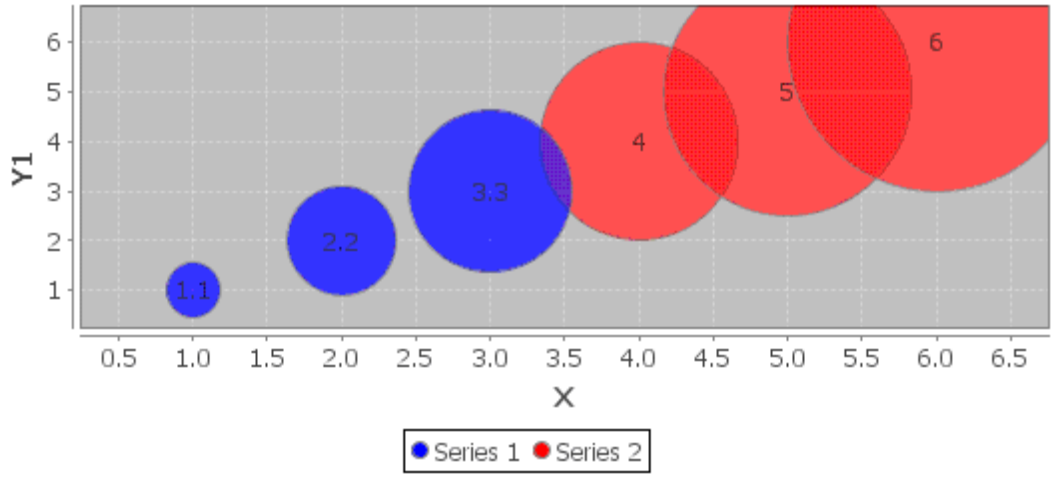
Data Set Example – Bubble Chart

Given the following data set with **two series**:

Bubble Chart Data - Series 1Bubble Chart Data - Series 2

X1	Y1	Z1	X2	Y2	Z2
1	1	1.1	4	4	4
4	2	2.2	5	5	5
3	3	3.3	6	6	6

Results in the following chart:



For the Candlestick Chart

The Candlestick chart requires the following columns for data mapping.

Data Mapping

Column	Data Type
Time	Date
High	Numeric
Low	Numeric
Open	Numeric
Close	Numeric
OPTIONAL Volume	Numeric
OPTIONAL Annotation	String

This chart only allows one (1) series.

Data Set Example – Candlestick Chart

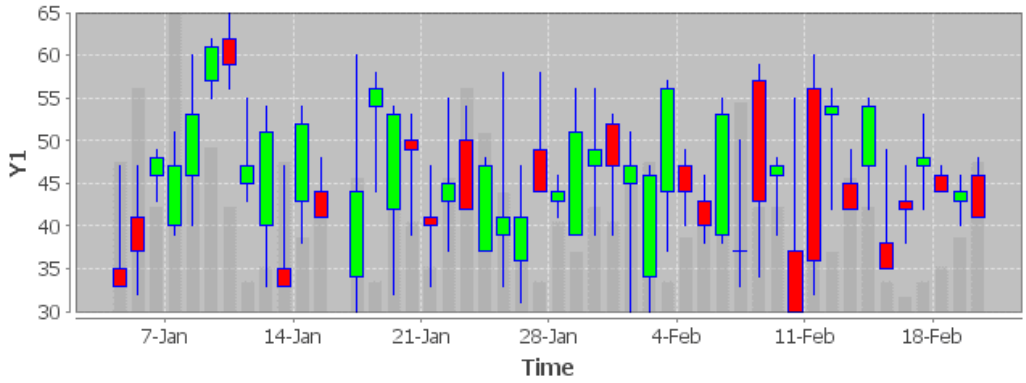
Given the following data set for a **single** series:

Candlestick Chart Data

Time	High	Low	Open	Close	Volume
2001-01-04 12:00:00	47	33	35	33	100
2001-01-05 12:00:00	47	32	41	37	150

Time	High	Low	Open	Close	Volume
2001-01-06 12:00:00	49	43	46	48	70
2001-01-07 12:00:00	51	39	40	47	200
2001-01-08 12:00:00	60	40	46	53	120
2001-01-09 12:00:00	62	55	57	61	110
2001-01-10 12:00:00	65	56	62	59	70
2001-01-11 12:00:00	55	43	45	47	20
2001-01-12 12:00:00	54	33	40	51	30
2001-01-13 12:00:00	47	33	35	33	100
2001-01-14 12:00:00	54	38	43	52	50
2001-01-15 12:00:00	48	41	44	41	80
2001-01-17 12:00:00	60	30	34	44	90
2001-01-18 12:00:00	58	44	54	56	20
2001-01-19 12:00:00	54	32	42	53	70
2001-01-20 12:00:00	53	39	50	49	60
2001-01-21 12:00:00	47	33	41	40	30
2001-01-22 12:00:00	55	37	43	45	90
2001-01-23 12:00:00	54	42	50	42	150
2001-01-24 12:00:00	48	37	37	47	120
2001-01-25 12:00:00	58	33	39	41	80
2001-01-26 12:00:00	47	31	36	41	40
2001-01-27 12:00:00	58	44	49	44	20
2001-01-28 12:00:00	46	41	43	44	60
2001-01-29 12:00:00	56	39	39	51	40
2001-01-30 12:00:00	56	39	47	49	70
2001-01-31 12:00:00	53	39	52	47	60

Results in the following chart:



For the Gantt Chart

The Gantt chart requires the following columns for data mapping.

Data Mapping

Column	Data Type
TaskName	String
StartDate	Date
EndDate	Date
%Complete	Numeric
OPTIONAL SubTask	String
OPTIONAL Annotation	String

This chart only allows one (1) series.

Things to Note

- A SubTask, with a given TaskName, cannot appear before the main task.

  Using the following Gantt chart data example, the subtask, `Design Phase/Design 1`, cannot appear before the main task, `Design Phase`.
- The StartDate must be less than or equal to the EndDate.

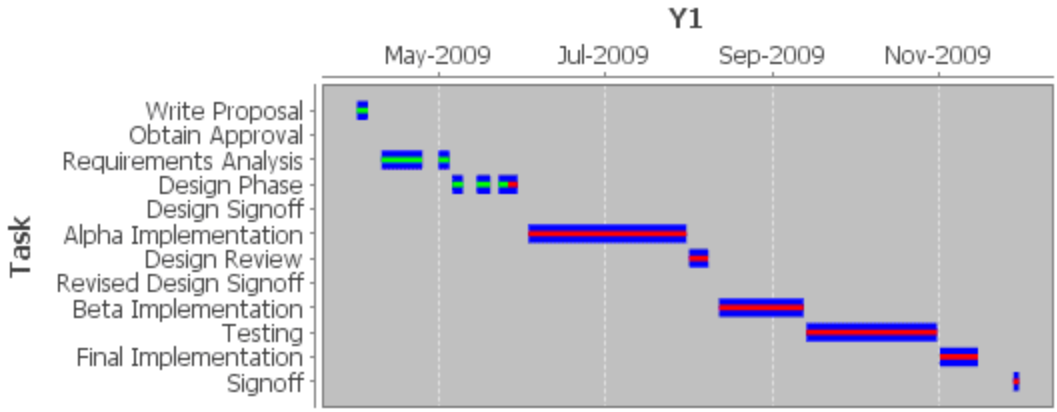
Data Set Example – Gantt Chart

Given the following data set for a **single** series:

Gantt Chart Data

TaskName	SubTask	StartTime	EndTime	%Complete
Write Proposal		2009-04-01	2009-04-05	1
Obtain Approval		2009-04-09	2009-04-09	1
Requirements Analysis		2009-04-10	2009-05-05	0
Design Phase		2009-05-06	2009-05-30	0
Design Signoff		2009-06-02	2009-06-02	0
Alpha Implementation		2009-06-03	2009-07-31	0
Design Review		2009-08-01	2009-08-08	0
Revised Design Signoff		2009-08-10	2009-08-10	0
Beta Implementation		2009-08-12	2009-09-12	0
Testing		2009-09-13	2009-10-31	0
Final Implementation		2009-11-01	2009-11-15	0
Signoff		2009-11-28	2009-11-30	0
Requirements Analysis	Req 1	2009-04-10	2009-04-25	1
Requirements Analysis	Req 2	2009-05-01	2009-05-05	1
Design Phase	Design 1	2009-05-06	2009-05-10	1
Design Phase	Design 2	2009-05-15	2009-05-20	1
Design Phase	Design 3	2009-05-23	2009-05-30	0.5

Results in the following chart:



For the Histogram Chart

The Histogram chart requires the following column for data mapping.

Data Mapping

Column	Data Type
Value	Numeric

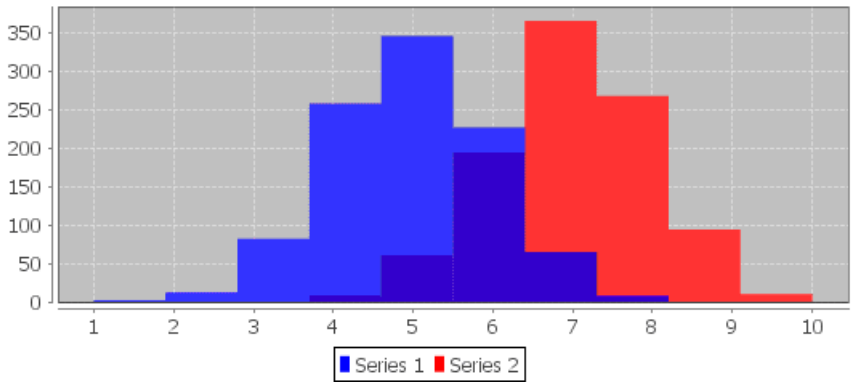
Data Set Example – Histogram Chart

Given the following data set with **two series** (with the number of Bins set to 10):

Histogram Chart Data - Series 1Histogram Chart Data - Series 2

Value1	Value2
5.26	6.86
5.54	5.33
4.35	6.01
3.97	6.45
5.54	5.02
5.65	6.04
5.2	7.8
4.13	7.29
5.58	8.89
4.38	6.98
3.9	6.53
6.5	7.86
5.51	8.37
4.96	7.64
5.43	7.5
5.28	8.16
3.14	7.24
2.51	7.79
5.35	8.86
3.31	5.95

Results in the following chart:



For the Pareto Chart

The Pareto chart requires the following columns for data mapping.

Data Mapping

Column	Data Type
Value	Numeric
Category	String
OPTIONAL Annotation	String

This chart only allows one (1) series.

Data Set Example – Pareto Chart

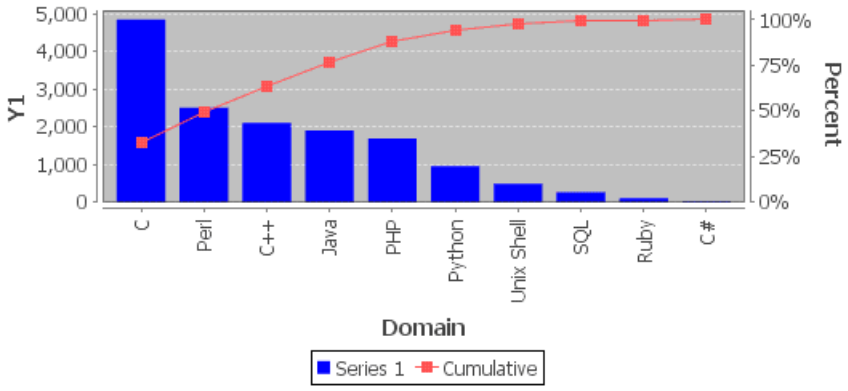
Given the following data set for a **single** series:

Pareto Chart Data

Value	Category
4843	C
2098	C++
26	C#
1901	Java
2507	Perl
1689	PHP
948	Python
100	Ruby

Value	Category
263	SQL
485	Unix Shell

Results in the following chart:



For the Pie Chart

The Pie chart requires the following columns for data mapping.

Data Mapping

Column	Data Type
Value	Numeric
Key	String
OPTIONAL Color	String

This chart only allows one (1) series. Also, a negative value for "Value" is not supported.

Data Set Example – Pie Chart

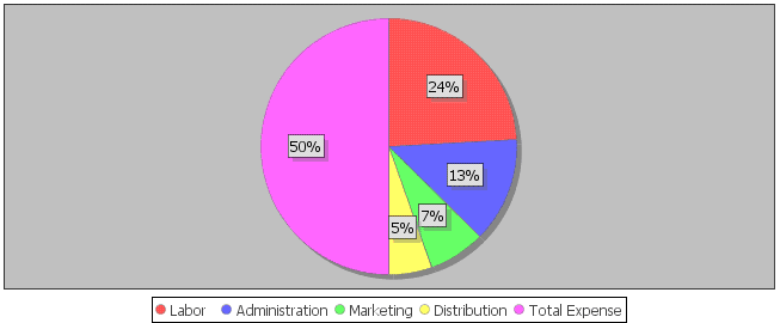
Given the following data set for a **single** series:

Pie Chart Data

Value	Key
15.76	Labor
8.66	Administration

Value	Key
4.71	Marketing
3.51	Distribution
32.64	Total Expense

Results in the following chart:



For the Polar Chart

The Polar chart requires the following columns for data mapping.

Data Mapping

Column	Data Type
Domain	Numeric The angular value in degrees.
Range	String The radial position.

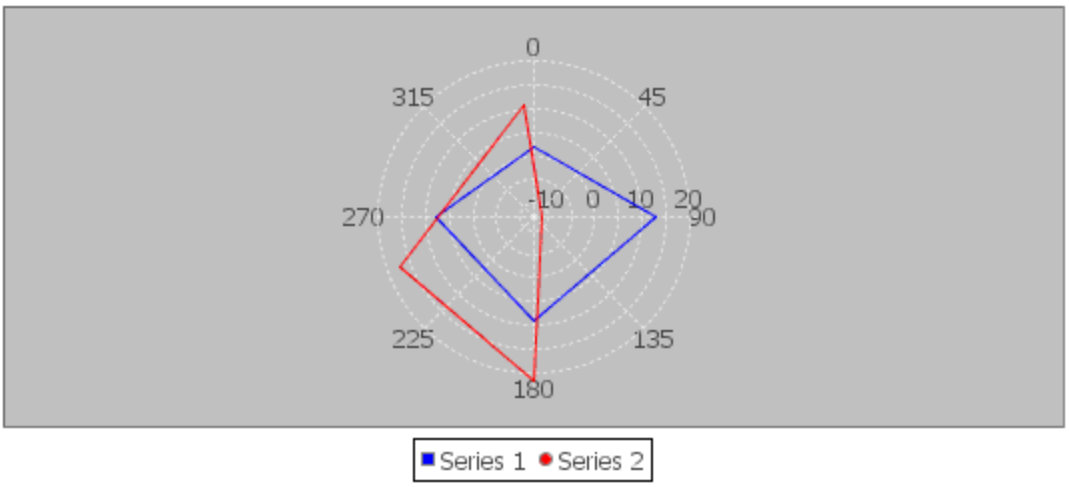
Data Set Example – Polar Chart

Given the following data set with **two series**:

Polar Chart Data - Series 1Polar Chart Data - Series 2

Domain1	Range1	Domain2	Range2
0	2	90	-11.2
90	13	180	21.4
180	9	250	17.3
270	8	355	10.9

Results in the following chart:



For the Spider Web Chart

The Spider Web chart requires the following columns for data mapping.

Data Mapping

Column	Data Type
Value	Numeric
Category	String

Data Set Example – Spider Web Chart

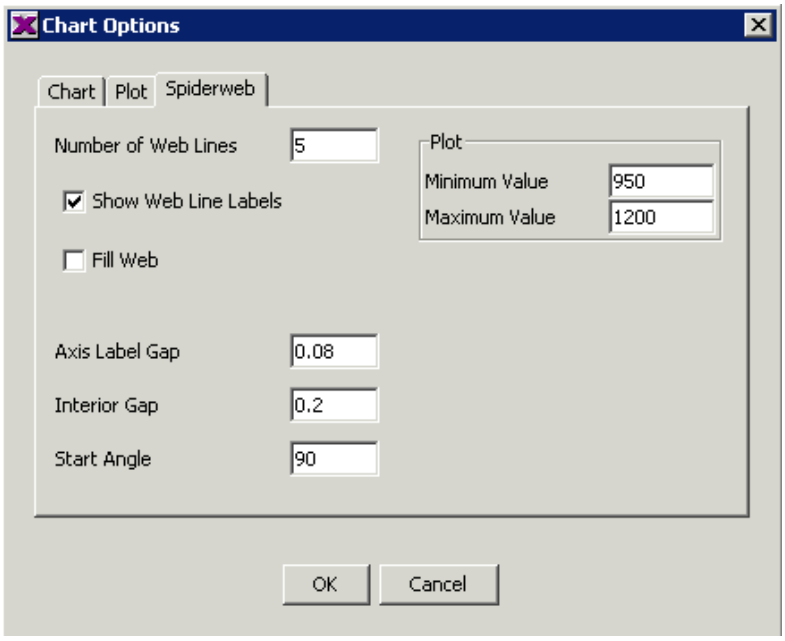
Given the following data set with **two series**:

Spider Web Chart Data - Series 1Spider Web Chart Data - Series 2

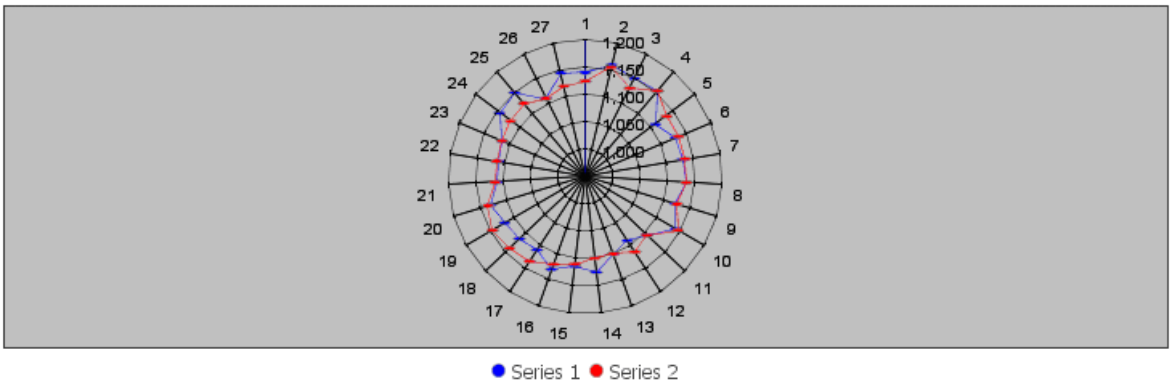
Value1	Category1	Value2	Category2
1140	1	1125	10
1160	2	1155	20
1150	3	1130	30
1155	4	1155	40
1110	5	1135	50
1130	6	1135	60
1130	7	1135	70
1135	8	1135	80
1120	9	1125	90

Value1	Category1	Value2	Category2
1140	10	1145	100
1105	11	1105	110
1090	12	1115	120
1100	13	1100	130
1125	14	1100	140
1115	15	1110	150
1130	16	1120	160
1110	17	1135	170
1115	18	1140	180
1120	19	1145	190
1130	20	1135	200
1110	21	1115	210
1110	22	1115	220
1115	23	1115	230
1145	24	1120	240
1150	25	1125	250
1110	26	1110	260
1145	27	1120	270

With the following **Chart Options**:



Results in the following chart:



For the Time Series Chart

The Time Series chart requires the following columns for data mapping.

Data Mapping

Column	Data Type
Time	Date
Value	Numeric
Annotation (optional)	String

Data Set Example – Time Series Chart

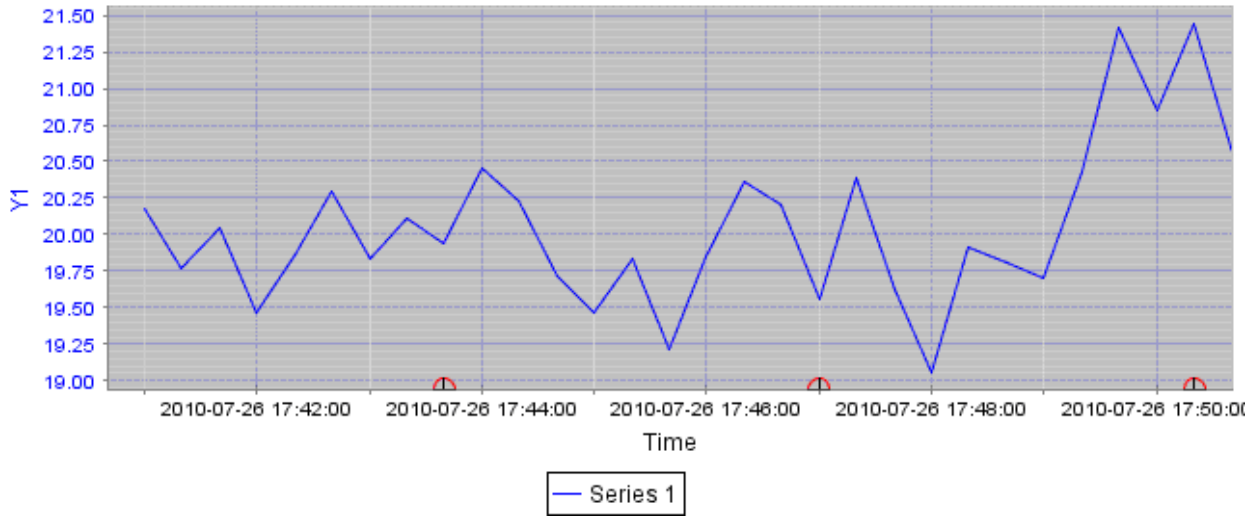
Given the following data set for a **single** series and an annotation:

Time Series Chart Data

Time	Value	Annotation
2010-07-23 17:46:00	42.59	
2010-07-23 17:46:20	43.53	
2010-07-23 17:46:40	42.42	
2010-07-23 17:47:00	41.96	Annotation value 3200
2010-07-23 17:47:20	40.85	
2010-07-23 17:47:40	40.77	
2010-07-23 17:48:00	42.25	
2010-07-23 17:48:20	40.75	
2010-07-23 17:48:40	40.99	

Time	Value	Annotation
2010-07-23 17:49:00	42.74	
2010-07-23 17:49:20	41.5	
2010-07-23 17:49:40	42.78	
2010-07-23 17:50:00	43.31	
2010-07-23 17:50:20	42.16	Annotation value 3210
2010-07-23 17:50:40	41.29	
2010-07-23 17:51:00	39.41	
2010-07-23 17:51:20	38.15	
2010-07-23 17:51:40	37.41	
2010-07-23 17:52:00	35.76	
2010-07-23 17:52:20	34.88	
2010-07-23 17:52:40	33.32	
2010-07-23 17:53:00	34.93	
2010-07-23 17:53:20	33.34	
2010-07-23 17:53:40	33.85	Annotation value 3220
2010-07-23 17:54:00	34.42	
2010-07-23 17:54:20	36.04	
2010-07-23 17:54:40	34.48	
2010-07-23 17:55:00	34.98	
2010-07-23 17:55:20	36.68	
2010-07-23 17:55:40	35.98	
2010-07-23 17:56:00	35.47	
2010-07-23 17:56:20	33.96	

Results in the following chart:



For the Treemap Chart

The Treemap chart requires the following columns for data mapping.

Data Mapping

Column	Data Type
Category	String
Item	String
Box Size	Numeric
OPTIONAL Shade	Numeric
OPTIONAL Exception	Integer
OPTIONAL ToolTip	String
OPTIONAL Parent Category	String

This chart only allows one (1) series.

Things to Note

- If the **Shade** column is not provided, the treemap uses the color specified by the Item's background color. If the Shade column is provided and the **Item Color Logic** is set to **Color Table**, then the treemap uses the colors from the color table to create a gradient pattern of colors (based on the range you set for the minimum and maximum Shade values). This results in a gradient spectrum of colors, starting from the minimum value color to the maximum value color.

- If **Exception** is set to one (1), then the treemap uses the color specified in the Exception group. Setting this value to one overrides the Shade column value.
- Unless a custom **Tooltip** is provided, the default ToolTip (with Item Label and Item Value) is displayed.

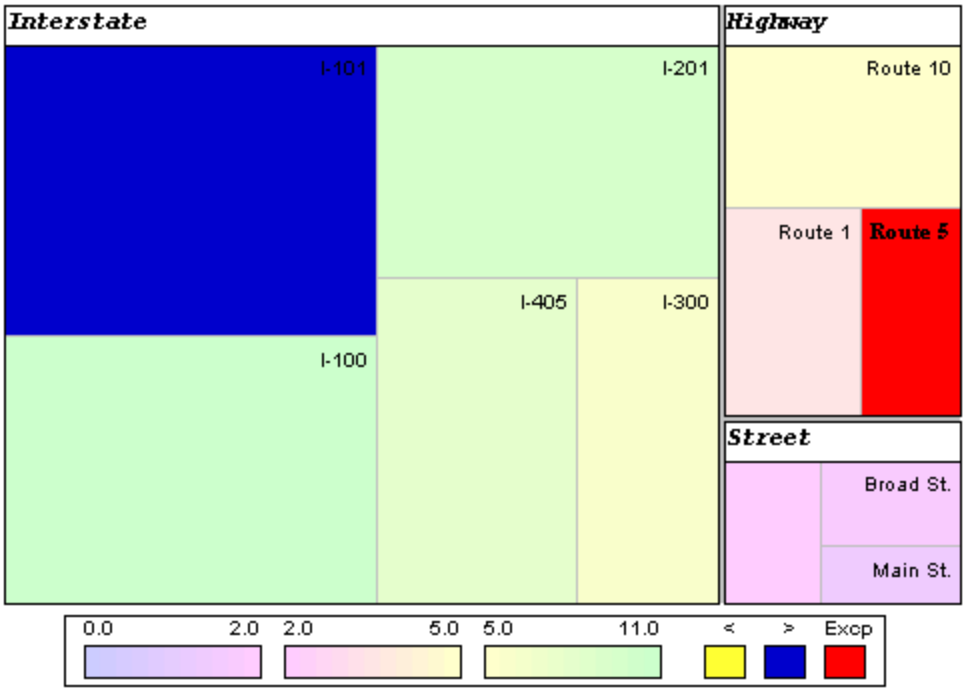
Data Set Example – Treemap Chart

Given the following data set for a **single** series:

Treemap Chart Data

Category	Item	BoxSize	Shade	Exception	Tooltip
Highway	Route 1	3.3	3.3		Dangerous
Highway	Route 5	2.4	2.4	1	
Interstate	I-300	5.3	5.3		
Interstate	I-405	7.5	7.5		
Interstate	I-100	10.5	10.5		
Interstate	I-201	9.2	9.2		
Interstate	I-101	11.5	11.5		Traffic Slow
Street		1.1	1.1		
Street		1.6	1.6		
Street		1.9	1.9		
Highway		4.5	4.5		

Results in the following chart:



For the Waterfall Chart

The Waterfall chart requires the following columns for data mapping.

Data Mapping

Column	Data Type
Value	Numeric
Category	String

This chart only allows one (1) series.

Data Set Example – Waterfall Chart

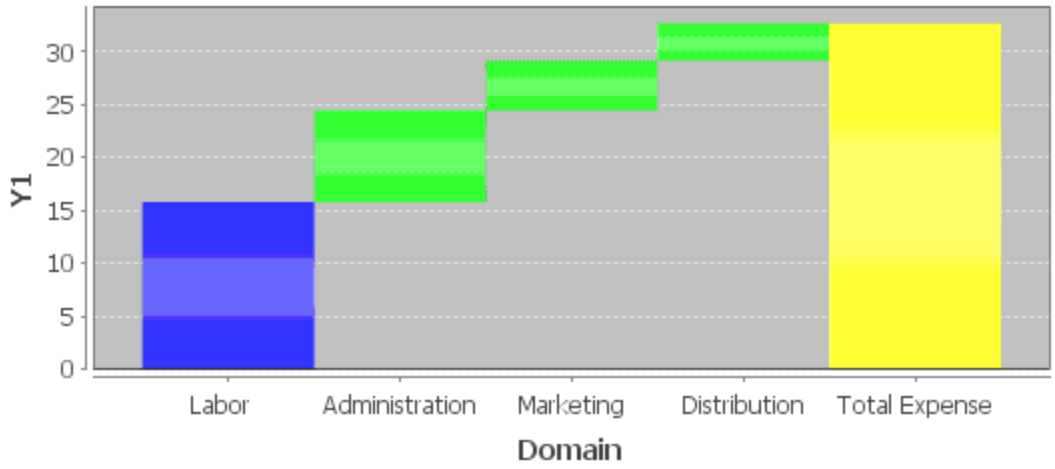
Given the following data set for a **single** series:

Waterfall Chart Data

Value	Category
15.76	Labor
8.66	Administration

Value	Category
4.71	Marketing
3.51	Distribution
32.64	Total Expense

Results in the following chart:



For the XY Chart

The XY chart requires the following columns for data mapping.

Data Mapping

Column	Data Type
Domain	Numeric
Range	Numeric
Annotation (optional)	String

Data Set Example – XY Chart

Given the following data set for a **single** series and **no** annotations:

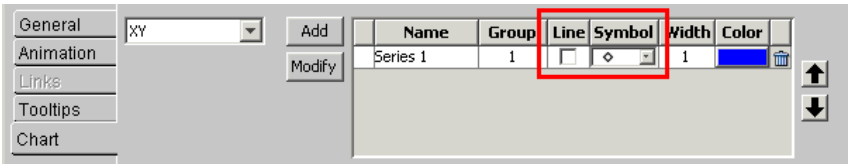
XY Chart Data

Domain	Range
40.68	389.61
-34.74	-342.69

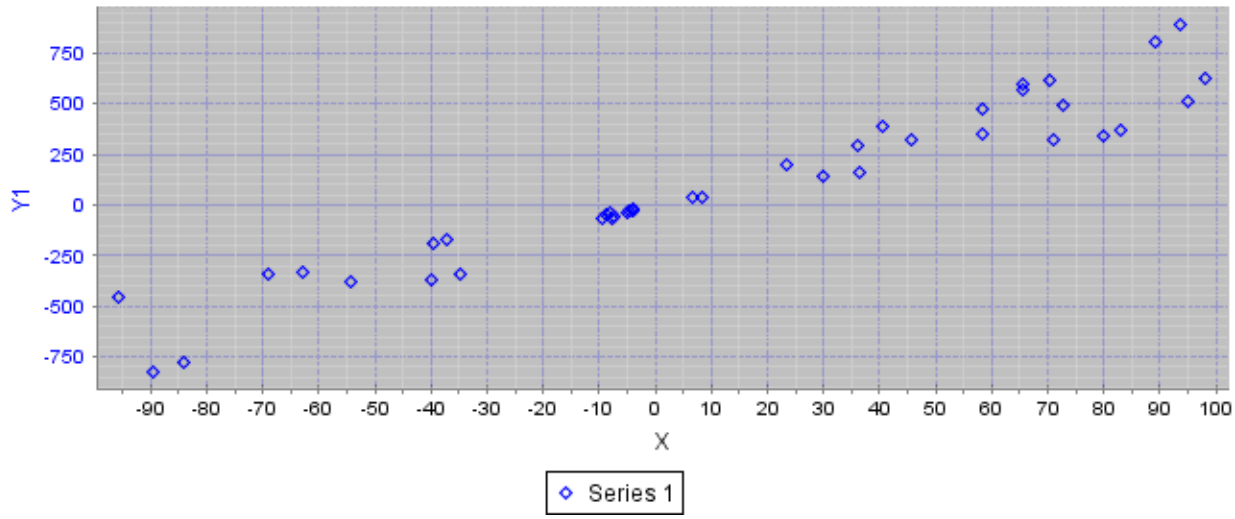
Domain	Range
-95.81	-452.89
89.15	801.03
94.95	514.91
36.59	162.36
65.4	567.1
-4.74	-31.35
-39.74	-187.56
58.51	353.77
29.8	141.14
70.41	610.68
-54.31	-375.05
23.27	202.56
-62.92	-331.35
-9.47	-62.31
-84.06	-773.11
-4.93	-34.49

With the following **Series** properties:

- No line
- Diamond symbol



Results in the following chart:



G - Shortcut Keys

Shortcut keys are key combinations, such as **CTRL+C**, that when pressed at the same time perform a particular command, in this case the **Copy** action.

Shortcut Keys

Menu	Command	Shortcut Key Combination
File	Save to repository	CTRL + S
	Revert to saved	CTRL + R
	Export	CTRL + E
	Launch Component Builder	F8
	Launch View Builder	F9
	View Details Report	CTRL + SHIFT + P
	Exit	
Edit	Undo	CTRL + Z
	Redo	CTRL + Y
	Cut	CTRL + X
	Copy	CTRL + C
	Paste	CTRL + V
	Delete	Delete
Add	Integer	SHIFT + I
	Long	SHIFT + L
	Real	SHIFT + R
	Boolean	SHIFT + B
	Double	SHIFT + D
	Decimal	SHIFT + M
	Date/Time	SHIFT + T
	Interval	SHIFT + V
	String	SHIFT + S
	Text	SHIFT + X
	Group	SHIFT + G
	Existing Component	SHIFT + C
	New Component	SHIFT + N
Insert	Select	SHIFT + S
	Line	SHIFT + L
	Polyline	SHIFT + P
	Rectangle	SHIFT + R
	Ellipse	SHIFT + E

Menu	Command	Shortcut Key Combination
	Text	SHIFT + T
	Value	SHIFT + V
	Image	SHIFT + I
	View	SHIFT + W
	View Instance	CTRL + SHIFT + W
	Keyed View	SHIFT + K
	Template	SHIFT + M
	Edit Box	CTRL + SHIFT + E
	Combo Box	CTRL + SHIFT + C
	Button	CTRL + SHIFT + B
	Text Area	CTRL + SHIFT + T
	Trend	SHIFT + D
	Browser	SHIFT + B
	Chart	SHIFT + C
	Table	SHIFT + A
	ENO Table	CTRL + SHIFT + A
	Excel	SHIFT + X
	GIS Map	SHIFT + G
Format	Enable Fill	CTRL + K
	Fill Color	CTRL + F3
	Line Color	CTRL + F4
	Text Color	CTRL + F5
	Bold	CTRL + B
	Italic	CTRL + I
	Align Left	ALT + SHIFT + L
	Align Center	ALT + SHIFT + C
	Align Right	ALT + SHIFT + R
Shape	Group	CTRL + G
	Ungroup	CTRL + SHIFT + G
	Bring to Front	CTRL + ALT + SHIFT + Up
	Send to Back	CTRL + ALT + SHIFT + Down
	Move Forward	CTRL + SHIFT + Up
	Move Backward	CTRL + SHIFT + Down
	Align:	
	Lefts	CTRL + ALT + L
	Centers (Vertically)	CTRL + ALT + C

Menu	Command	Shortcut Key Combination
	Rights	CTRL + ALT + R
	Tops	CTRL + ALT + T
	Middles (Horizontally)	CTRL + ALT + M
	Bottoms	CTRL + ALT + B
	Rotate Right	CTRL + SHIFT + Right
	Rotate Left	CTRL + SHIFT + Left
	Flip Horizontal	CTRL + H
	Flip Vertical	CTRL + J
	Configure Grid	ALT + SHIFT + G
Help	Help Topics	F1
	About XHQ Workbench	SHIFT + F1