SIEMENS

XHQ

Administrator's Guide

About This Guide

Starting XHQ 1

Security Access and Privileges 2

Solution Deployment and Access 3

XHQ Audit Trail 4

Working with PROPERTIES Files 5

Using the xhqci Utility 6

Using xhqdr with IP.21 7

Troubleshooting Connector Backlogs 8

XHQ Redundancy 9

XHQ Solution Viewer Administration 10

About the XHQ Web Files 11

Localization and XHQ 12

Distributed Solutions 13

Miscellaneous Topics 14

Appendix A

06/2019 V6.0.0.1

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 Copyrights and Trademarks 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 © 1998-2019 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 Product Lifecycle Management 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 2008, Windows Server 2012, Windows Server 2016, 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 **18**
Conventions Used in This Guide 18
Visual Cues for Online Viewing 19
Related XHQ Product Documentation 20
Contacting Customer Support 22
General Feedback and Comments 23
1 | Starting XHQ **24**
Starting and Stopping the XHQ Solution 24
To start or stop the solution 25
To restart the solution 25
Accessing the Server 25
To point the development client to a solution 25
About the xhqboot Batch File 26
To execute the xhqboot batch file 27
To access xhqboot help 28
xhqboot Topic Descriptions 29
xhqboot help actions 30
xhqboot help dirs 30
xhqboot help format 30
xhqboot help syntax 31
xhqboot help classic 32
xhqboot help classpath.append 32
xhqboot help classpath.prepend 32
xhqboot help custom.modules 33
xhqboot help db.retries 33
xhqboot help db.timeout 33
xhqboot help ex_server 34
xhqboot help ex_ans 34
xhqboot help ex_router 34
xhqboot help ex_delay 34
xhqboot help javaopts 35
xhqboot help memory 35
xhqboot help modules 35
xhqboot help params.append 36

xhqboot help params.replace 36
xhqboot help sequence 37
xhqboot help shutdown.dat 37
xhqboot help startup.dat 37
xhqboot help xhqboot.properties 37
xhqboot help xhqbootmods.dat 37
How to run a batch file using the rep_run.exe program 38

2 | Security, Access, and Privileges 40
XHQ Development Clients Security 40
About XHQ Security 40
How Passwords Are Handled 40
Applying Access and Privileges 41
Accessing the Role Administration Utility 41
About Roles 42
Predefined Roles 42
About Security Modes 44
Using Classic Security 44
Using Active Directory Security 44
To configure the web server for Active Directory security 44
To configure IE for Active Directory security 45
To enable Active Directory security from the XHQ Workbench 46
To view a solution, with Active Directory security enabled, from a browser 46
How security works with xhqci 47
Managing Roles 48
To create a new role 48
To edit an existing role 49
More About Groups 51
To map a predefined role to a group 53
Deleting Roles 55
To delete a role 55
Limiting Rights 56
To grant/deny Rights 58
About Rights and Predefined Roles 59
About Visual Tile Composer Access 62
About Security and File Recovery 62
Refreshing Roles 63
To refresh XHQ Roles using the Role Administration utility 64
Setting Permissions 65

Permissions for the XHQ Workbench – Main Menu 65
Permissions for the XHQ Workbench – Shortcut Menu 66
Permissions for View Building 67
Permissions for the Solution Builder – Main Menu 67
Permissions for Global Collection Access 68
Managing Component Access 69
To assign a role (or roles) to a component 69
Exporting and Importing Component Roles 70
About Role Mapping 71
Role Mapping: Defining the XML format 71
The Role Mapping DTD 73
Configuring End User Access (for the Solution Admins role) 74
Kerberos Authentication 77
About XHQ and Kerberos 77
Setting up the XHQ Server 77
Step 1 - Addressing Pre-requisites 78
To force the use of TCP packets 78
Web Servers 78
To force the use of TCP packets 79
XHQ System Pre-requisites 80
Step 2 - Choosing a Service Principal Name (SPN) 80
Step 3 - Creating a Unique Domain User Account 81
To create an AD domain user account 81
Step 4 - Mapping the SPN 82
Step 5 - Configuring the XHQ Service 83
To configure the XHQ Service to the SPN user 83
Step 6 - Verifying Kerberos Use 83
Troubleshooting Kerberos 83
Checking XHQ 83
Using the kerbtray utility 83
Debug Settings 85
To set the debugging environment 85
To enable Kerberos event logging on a specific computer 85
Testing the XHQ connection 86
Application Server Log Messages 86
XHQ Web Service Impersonation and Delegation 87
End-user impersonation 87
End-user delegation 88

Changing Group Membership 91
Cross-forest Authentication 93
View Security 94
Assigning View Roles 94
Requirements 96
Exporting/Importing 96
View Flavors 96
To set view security for views 97
XHQ Web Server Architecture and Security 98
Configuring SSL in IIS for HTTPS 99
Enabling SSL in IIS 99
To enable SSL in IIS 99
Deploying the Root Certificate 101
Using the XHQ Install Certificate Utility 101
To deploy the root certificate on XHQ server and client machines 102
Jetty and IIS 102
To reconfigure IIS 102
Configuring SSL for XhqWse Web Service 106
To enable the XhqWse web service to work with SSL (HTTPS) 106
SSL Access for the XHQ Applet and Web Applications 109
To configure the SSL certificate for the indx virtual directory 109
SSL Access for the XhqWse Web Service 109
Updating WAKA Links to use HTTPS 110
Updating Client Applications to use TLS 111
XHQ Server Communication Security 112
Background 112
Changing XHQ Port Configuration 113
To change the configuration of the XHQ ports 114
Using the XHQ Portmap Utility 115
Portmap Configuration File 116
Troubleshooting 118

3 | Solution Deployment and Access 120
Defining the Solution 120
To define a solution and attach the single top node to the solution environment 121
To rename a solution 122
Deleting a Solution 123
To completely remove the solution 123
Accessing Your Solution 124

To open the solution from a browser 124
To access a specific view directly 124

4 | XHQ Audit Trail 125
Launching XHQ Audit Trail 125
Audit Trail for XHQ Core 126
Database Format 126
Audit CSV Format 127
Audit Trail for XHQ Performance Management Applications 129
Viewing XHQ Audit Trail 131
Setting Filters 132
To change Start and End dates 132
To reset filters 133
To export filtered audit records 133

5 | Working with .PROPERTIES Files 134
Using globalsettings.properties 137
Property Settings for XHQ HTML5 Solution Viewer 138
Property Settings for the Trend Viewer 139
Property Settings for the Navigation Bar 140
To change a NavBar property 140
Using Custom Images 140
To replace a default image with a custom image 142
Showing or Hiding the Logo 143
To show/hide the logo 143
Placement Customization 144
To place the logo in the West region 144
To place the favorite panel in the South region 145
Showing or Hiding Button Bar Icons 145
Creating Custom Buttons 146
NavBar Clock Properties 146
Alarm Update Setting 147
Changing the NavBar Background 147
Setting the NavBar Icons Style 147
Changing the Interface Type 148
Editing the Interface Type Attributes 148
Editing Navigation Settings 151
To customize the tab order for component members 152
To set the number of component member tabs to display at once 152

To set a maximum number of Trends/Views to a favorites list 152
To set the number of items on the Explorer drop-down list 153
To display the NavBar tabs in the appropriate locale using component member descriptions 154
Server Storage of Preferences 155
To set the server storage of preferences 155
Using JavaScript to Get Client Time 156
getClientTimeAsString 156
getClientTimeAsLong 156
getClientTimeAsDate 156
Setting Historical Navigation Values 157
Showing Tooltips 158
Solution Viewer Progress Indicator 160
Working with the View Background Color 160
Browser Control 161
View and ENO Group Instances 162
Setting global properties for View Usage Statistics 163
Setting global properties for Date and Time Formats 164
How the parameter DefaultTimeStampFormat is used 166
How the parameter DefaultDateOnlyFormat is used 167
How the parameter DefaultTimeOnlyFormat is used 168
24-hour versus 12-hour Format 168
Enabling a Historian – History Tab 169
Configuring Caches 171
View Load Performance 172
Redundancy 173
For Aggregates 174
Visual Tile Composer Properties 175
Grid Dimensions 175
View Load Performance 175
Additional Global Settings 177
Using globaleditingsettings.properties 184
Attribute Key 187
Using solutionsettings.properties 189
Connector Restart Properties 194
Aggregated Values Cache Properties 195
Tag Sync Property 197
Historization Property 197
Using cachesettings.properties 198

Using modelsettings.properties 199
Fallback 199
Using routersettings.properties 201
About clientvariablesettings.properties 201
About clientsettings.properties 201
XHQ Alert Notification System (ANS) Properties 202
About xhqboot.properties 203
Starting The Servers 203
About install.properties 204
To create the install.properties file 204
About jetty.properties 205
About monitoring.properties 206
About storage.properties 207
About xhq_service.properties 209
To create the xhq_service.properties file 210
About the XHQ View Server and viewserversettings.properties 212
Setting Properties 212
Session Timeouts 214
To change the session timeouts 214
Tuning Database Performance (XHQ DB Settings Utility) 215
About database.custom.properties 216

6 | Using the xhqci Utility 220

About the xhqci Commands 221
Common Parameters to xhqci Commands 222
xhqci Access and Security 222
Using the Command-Line 223
To execute the xhqci batch file 223
Exporting a Solution 224
Importing a Solution 227
Removing a Configuration 227
Configuring the Tag Time-series Data Type By Import 228
Exporting Components 229
Importing Components 230
Exporting Component Roles 232
Importing Component Roles 233
Exporting Roles Mappings 234
Importing Role Mappings 235
Exporting View Tile Composer (VTC) Views 236

Importing View Tile Composer (VTC) Views 238
Purging View Tile Composer (VTC) Views 239
Revert Units of Measure 240
Exporting Units of Measure (UoM) 241
Importing Units of Measure (UoM) 242
Exporting Connector Definitions 243
Importing Connector Definitions 244
Exporting Connector Groups 245
Importing Connector Groups 246
Exporting Connector Processes 247
Importing Connector Processes 248
Exporting Solution Definitions 249
Importing Solution Definitions 250
Importing Alert Definitions 251
Exporting Connection Groups to be Deleted 252
Exporting Connection Processes to be Deleted 252
Exporting the Model to be Deleted 253
Exporting Deleted Role Mappings 254
Exporting Solution Collections to be Deleted 255
Deleting Instances Created in Extended Namespace 256
Exporting a Database 257
Setting Configuration Import Complete 258
Updating a Solution 259
Solution Information 260
Model Information 262
Refreshing Roles 263
Renaming Components 264
To rename components using the command-line 264
Using ViewStats 265
To use the viewstats command-line 266
Compressing the Solution 267
Starting/Stopping a Connector 268
Restoring Storage Mode 269
Looking at the XHQ License 270
Managing Data Quality 271
Updating Client Notification 272
Performing a Tag Reconciliation 274
Using Collection Tools 275

Online Backup 277
Setting a Tag Synchronization Processing Delay 279
Deleting Global Collections 280
Deleting Classes 281
Validating A Repos 282
Resolving Aliases to Paths 283
Related Global Properties 284
Checking the Health of the XHQ Server 285
Historical Evaluation 286
Clearing Audit Records 288
Additional XHQ Server Actions 289

7 | Using xhqdr with IP.21 293
Getting the First Historical Record 294
Transferring History from XHQ to IP.21 296
How Migration Works 296
Deleting Tags from the XHQ Data Recorder 297
Working with Aliases 298

8 | Troubleshooting Connector Backlogs 299
What are Backlogs? 299
Common Causes of Backlogs 300
Steps to troubleshooting the backlog 302

9 | XHQ Redundancy 304
How is Redundancy Applied? 304
How Redundancy Works in XHQ 305
How to Configure Redundancy 306
Defining Host Names 306
To define the primary and backup server host names 306
Setting Global Properties for Redundancy 307
Monitoring the Connection Status 308
Accessing Server Information 308
To access server information from the Solution Viewer 308
How the XHQ Client Handles Connection Change Events 309
ANS and Redundancy 310
Editing the config.json File 310
To edit the config.json file 310

10 | XHQ Solution Viewer Administration 311
Using XHQ Web Diagnostics 311

To run the XHQ Key-Pair Utility ... 316
Usage Stats ... 317
Debugging by Session ID ... 320
Disabling HTTPS Enforcement ... 320
To disable HTTPS enforcement ... 320
Setting Cookie Expiration ... 321
To set the cookie expiration ... 321
Using the XHQ Solution Viewer with the Oracle JVM ... 321
Setting the Maximum Length for the Trend Title ... 322
To set the Trend title maximum length ... 322
To change the Trend title maximum length ... 322
Solution Viewer Display Resolution ... 323
Embedded Trends and Double Buffering ... 323
Anti-aliasing in the XHQ system ... 323
Working with High-Resolution Displays ... 324
To increase the Oracle JVM heap size ... 324

11 | About the XHQ Web Files ... 325
The XHQ Web Root Directory ... 325
NavBar-related Files and Directories ... 326
Customized Web Script Files ... 327
trendsettings.js ... 327
appletsettings.js ... 327
svsettings.js ... 328
styles.css ... 329
About the /indx/custom Virtual Directory ... 330
To stop a web site prior to replacing (switching) the repos folder ... 330
How to Make Custom Configurations ... 331
If the Web Server is on the SAME machine as the XHQ Server ... 331
Setting the Request Mode ... 332
To set the request mode ... 332

12 | Localization and XHQ ... 333
Setting the XHQ Workbench Locale ... 333
Placing the Language Files ... 334
To place the language file ... 335
Displaying Kanji ... 336
To display the NavBar tabs in the appropriate locale using component member descriptions ... 336
To properly display Kanji (from the server) ... 337

Displaying Localized Calendar Formats 338
To set the localized calendar format for the high-performance database 338
About Locale and Arithmetic Expressions 339
Using Halfwidth or Fullwidth Forms 339
To create the properties file 339
To use halfwidth symbols 339

13 | Distributed Solutions 340
Overview 340
About Realms 341
About the Model Manager 341
Solution Management 342
Using the XHQ Solution Administration Utility 343
To access the Solution Administration Utility 345
To add a realm 345
To add a solution server 346
Configuring the XHQ Administration Utility 353
To access the XHQ Administration Utility 353
To manage the realms 354
Managing Servers in Distribution 356
To add a server to the realm 356
To remove a server from the realm 356
To move a server to another realm 356
To change the hostname of a server 357
Migrating an Existing Solution to Distribution 358

14 | Miscellaneous Topics 359
About XHQ Tray 361
About the Digital Certificate 362
About XHQ 64-bit 363
Technical Details of the 64-bit Operation 364
The XHQ Service Configuration 364
The Windows XHQ Service 366
To start or stop the XHQ_service from the Control Panel 367
To start or stop the XHQ_service from a Command Prompt 367
The XHQ Keep Alive Service 368
XHQ Service and WAKA Configuration 370
XHQ Performance and Anti-virus Software 371
Optimal Server Settings 372

To adjust scheduling for improved performance ... 372
Virtual Server Environments ... 373
Additional Security Considerations ... 374
Clickjacking ... 374
To prevent clickjacking ... 374
Cross-site Scripting ... 374
About XHQ and Active Directory ... 375
About the License Manager ... 375
XHQ Logs ... 375
About the Log Directory ... 375
About the Solution Viewer Log ... 375
Load View Performance Logging ... 376
IIS Logging ... 377
Customizing the Development Client Logs and Repos Directory Locations ... 378
More about the launchsettings.bat files ... 378
JDBC Fetch Size ... 379
About Hyper-Threading Technology and the XHQ System ... 380
Using the debug.js File ... 381
To enable logging and set the log level ... 381
XHQ Workbench File Recovery ... 383
About Security and File Recovery ... 383
For XHQ Applications Using AJAX ... 384
To configure the IIS Manager for XHQ applications using AJAX ... 384
About the XHQ Internal Cache ... 385
Improved Performance with Storage Updates ... 386
Using the smck Utility ... 386
To upgrade with compressed DAT files using the smck utility ... 386
Changing the XHQ High-Performance Database Password ... 387
About the Password Utility (pwutil) ... 387
To change the individual user's password for the High-Performance Database ... 390
To change all XHQ passwords ... 391
Maintenance of the Embedded Database TEMP Tablespace ... 392
About XHQTempRecreate.bat ... 393
Disabling the Data Recorder ... 394
To disable the XHQ Data Recorder service ... 394
To set the global property to hide the XHQ Data Recorder option ... 395

Appendix ... 396
Date and Time Formats ... 396

Text 397
Number 397
Year 397
Month 397
General Time Zone 397
RFC 822 Time Zone 398
Pattern Examples 398
Valid Date/Time Formats 399
Kerberos Glossary Terms 400

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**
The *.x* (in italics) is used to indicate release numbers of a product.	Enable (by checking) the **Use Java *x.x.x_xx* for <applet>** option.
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.

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**.

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`

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.

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.

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
XHQ Visual Composer Guide Provides end-user information for the XHQ Visual Composer and associated programs, which are used in the development of presentation content.	Content Developers

Contacting Customer Support

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 (default: Americas):

- Americas (15-1 GMT)
- Europe (8-17 GMT)
- Asia (1-10 GMT)

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

- USA (Americas)
- Germany (Europe)
- Singapore (Asia)

Example Americas: *Implies coverage from 7:00 AM to 5:00 PM, Pacific Time, Monday to Friday, excluding US national holidays.*

Silver Support: 24/5

24 x 5 hours support. 24 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 (default: Americas):

- Americas
- Europe
- Asia

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

- California/USA (Americas)
- Germany (Europe)
- Singapore (Asia)

Example Americas: *Implies coverage from midnight on Sunday until midnight on Friday, Pacific Time, Monday to Friday*.

Gold Support: 24/7

24 x 7 hours support. 24 hours per day, 7 days per week.

- **Postal Mail**

  Siemens Product Lifecycle Management 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 Product Lifecycle Management 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.

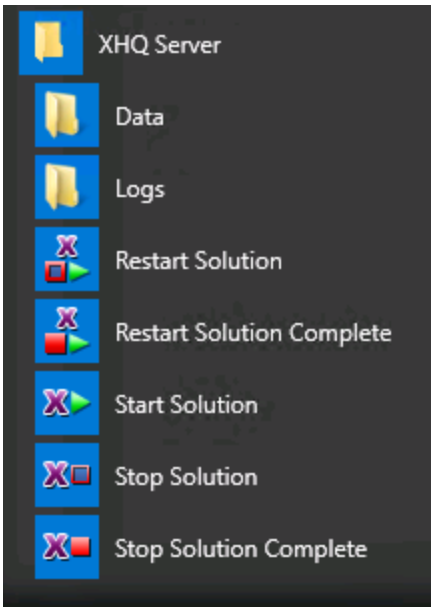
1 | Starting XHQ

Starting and Stopping the XHQ Solution

Before you begin, you will need to learn how to start and stop the XHQ solution.

If you added Start/Stop/Restart solutionshortcut icons to your desktop or Quick Launch toolbar, you may have to replace them after upgrading the XHQ system.

Stopping or restarting the XHQ Server performs a reset of IIS services.



Start Menu – XHQ Server Commands

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.
	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.

See also the topic, *Troubleshooting XHQ Start-up*, located in the Appendix section of the **XHQ Installation Guide**.

To start or stop the solution

- From the **Start menu**, go to **XHQ Server** and click **Start Solution**.
  The Start Server MS-DOS window appears.
- To stop servers, from the **Start menu**, go to **XHQ Server**and click **Stop Solution**.
  The Stop Server MS-DOS window appears.

To restart the solution

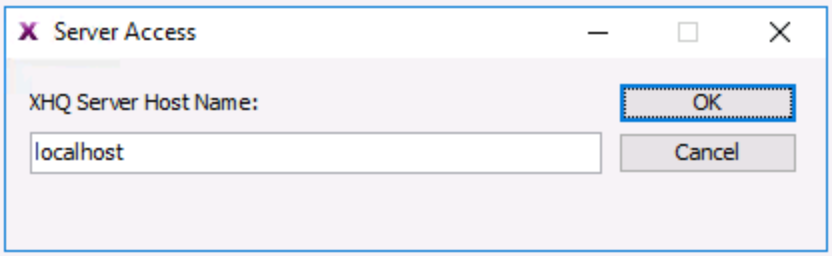
From the **Start menu**, go to **XHQ Server** and click **Restart Solution**.
This automatically stops and starts the XHQ Server.

Accessing the Server

In this section, you will learn how to point the development client to a solution using the **Server Access** utility.

To point the development client to a solution

1. Start the **XHQ Server**.
2. From the **Start menu**, go to **XHQ Development** and click **Server Access**.
   The "Server Access" dialog displays.



3. For **XHQ Server Host Name**, enter the **TCP/IP host name** of the enterprise server.
   Typically, if you installed the Enterprise Server on the local machine, the host name is `localhost`.
4. Click **OK**.

About the xhqboot Batch File

The **`xhqboot.bat`** batch file links the XHQ start/restart/shutdown process to a repository, which allows for smoother upgrades. By running the **`xhqboot.bat`** batch file through a DOS Command Prompt, you can start, shutdown, or restart the XHQ System on any XHQ System server machine.

The shortcuts in `Start menu > XHQ Server`, as well as the XHQ Service, also use `xhqboot.bat` to perform the above actions.

The **`xhqboot`** batch file is located in the **`%XHQ_SERVER_HOME%\bin`** directory, which by default is `C:\Program Files (x86)\XHQ\XHQ Server\bin`.

For a list of XHQ environment variables and their default paths (values), go to the topic, *Distribution of XHQ Server and devClient Files*, located in the XHQ Installation Guide.

When you execute the batch file, the following command options appear.

Syntax:

```
xhqboot startup|shutdown|restart [classic][complete][restricted][test|window]
xhqboot create standard|ans|router
xhqboot help topics|?
xhqboot help [topic]
```

Brackets [ ] indicate that the parameter is optional.

xhqboot Commands

Parameter	Description
startup	Starts the XHQ System.
shutdown	Stops the XHQ System.
restart	Stops and then starts the XHQ System.
create	Creates the `xhqboot.properties` file, if not present.
help	Lists command descriptions.

xhqboot Command Options

Parameter	Description
classic	In classic mode, `%XHQ_SERVER_HOME%\bin\startup.dat` and `%XHQ_SERVER_HOME%\bin\shutdown.dat` are strictly used; custom scripts are not produced.
complete	Completely stops all embedded services (for example, database and web service) and IIS (the entire web server, not just XHQ Web Root). *Example:* `xhqboot shutdown complete`
restricted	Restricts scripts to standard modules only.
test	Produces scripts only – do not invoke. May not be combined with the window parameter.
window	Displays output in a separate window. This option may only be used for startup, restart, or shutdown operations, and may not be combined with the test parameter.

Parameter	Description
	Selecting text in this output window does not cause `xhqboot` to pause (or hang). Note, closing this window does not stop `xhqboot`, you will simply not see the output. *Visual Cues:* • Error messages appear in red. • The output window title matches the `xhqboot` function (for example, XHQ Restart Solution Complete). • The active taskbar icon and the output window icon (upper left-hand corner of the window) match the server shortcut for the `xhqboot` function.
topics	Lists help topics.
?	Lists help topics.
topic	Enter a specific command topic.

The primary task of `xhqboot` is to generate customized `startup.dat` and `shutdown.dat` scripts based on its configuration. These scripts are then read by `rep_startup.exe` and `rep_shutdown.exe`. To produce repository-specific start and shutdown sequences, `xhqboot` uses the following files:

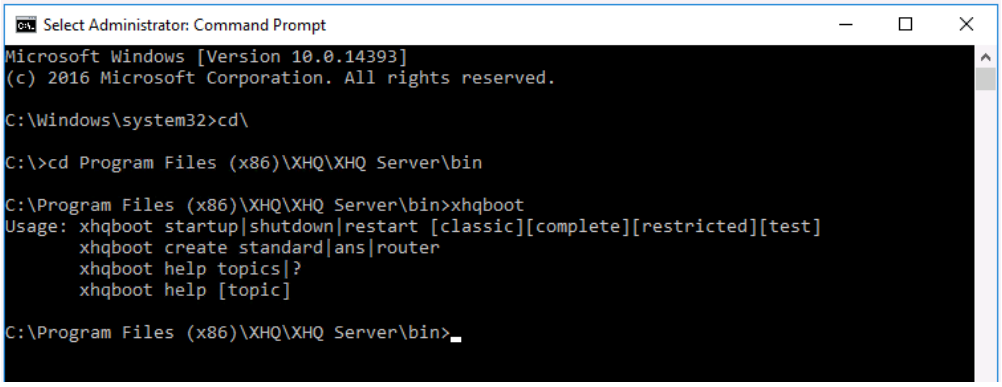
File	Description
%XHQ_SERVER_HOME%\bin\startup.dat	The start-up template.
%XHQ_SERVER_HOME%\bin\shutdown.dat	The shut-down template.
%XHQ_SERVER_REPOS%\xhqboot.properties	Customization properties.
%XHQ_SERVER_REPOS %\xhqbootmods.dat	Custom modules.
%XHQ_SERVER_REPOS %\tmp\startup.dat	Generated start-up script.
%XHQ_SERVER_REPOS %\tmp\shutdown.dat	Generated shut-down script.

Typically, the configuration installed with the XHQ System is sufficient for normal use. However, if your application of the XHQ System requires other programs to be started after the XHQ Solution Server is running, for example, you may have to customize the configuration. The primary customization file is **`xhqboot.properties`**. This file contains property settings that govern how the start-up and shut-down scripts are generated. An optional configuration file, `xhqbootmods.dat`, allow you to specify custom modules (programs) that can be invoked during the start-up and/or shut-down sequence of the XHQ System.

Do not leave blank lines in `xhqboot` files. Either comment out blank lines or delete them.

To execute the xhqboot batch file

1. Run a **Command Prompt** as an **Administrator** and enter the location for the **`%XHQ_SERVER_HOME%\bin`** directory.
2. After the prompt, type **`xhqboot.`**
   A list of supported xhqboot commands appears.



3. Use the following syntax example to shutdown and start XHQ:

```
@echo off
call "%XHQ_SERVER_HOME%\bin\xhqboot.bat" shutdown
. . .
call "%XHQ_SERVER_HOME%\bin\xhqboot.bat" startup
```

By default, the shutdown process for the XHQ System does not shutdown the Cache/ANS XHQ High-Performance Database (the embedded database). If you would like to shutdown the Cache/ANS XHQ High-Performance Database along with the XHQ System, you must use the "complete" option with the "shutdown" command (see the following example).

4. *OPTIONAL*
   Use the following syntax example to shutdown and start XHQ with the Cache/ANS XHQ High-Performance Database:

```
@echo off
call "%XHQ_SERVER_HOME%\bin\xhqboot.bat" shutdown complete
. . .
call "%XHQ_SERVER_HOME%\bin\xhqboot.bat" startup
```

To access xhqboot help

Do one of the following:

- To view the **list of help topics**, enter the following sequence after the location path and the prompt:

  `xhqboot help ?`

  or

- To view the **syntax for a command**, enter the following after prompt:

  `xhqboot help <command topic name>`

xhqboot Topic Descriptions

The xhqboot help topics fall into four of the following categories.

General	
actions	Supported action names in properties files.
dirs	Directories and contents used by xhqboot.
format	Template/module file formatting information.
syntax	Property name syntax information.
topics	List of help topics.

Properties	
classic	Specifies operating in "classic mode"
classpath.append	The Java™ class path to append.
classpath.prepend	The Java™ class path to prepend.
custom.modules	The list of custom modules to use.
javaopts	The Java™ options to insert.
memory	The memory settings to use (-Xmx options).
modules	The list of standard modules to use.
params.append	The parameters to append to the invocation line.
params.replace	The parameters to replace on the invocation line.
sequence	The sequence setting for a custom module.

Files	
shutdown.dat	The template or resulting script for shutdown action.
startup.dat	The template or resulting script for startup action.
xhqboot.properties	The settings for xhqboot processing.
xhqbootmods.dat	Custom modules for xhqboot to process and place in a script.

Example Configurations	
ex_ans	Sample xhqboot.properties for an XHQ Server with ANS.
ex_delay	Sample xhqboot configuration files to delay startup.
ex_router	Sample xhqboot.properties for router-only (web server node).
ex_server	Sample xhqboot.properties for a standard XHQ server.

xhqboot help actions

The following are supported action names in properties files:

Action Name	Description
`shutdown`	Stop XHQ processes.
`startup`	Start XHQ processes.

xhqboot help dirs

The following are various directories used by xhqboot.

Directory	Description
`%XHQ_SERVER_HOME%\bin`	Location of `startup.dat` and `shutdown.dat` templates.
`%XHQ_SERVER_REPOS%`	Location of `xhqboot.properties` and `xhqbootmods.dat`.
`%XHQ_SERVER_REPOS%\tmp`	Location of generated scripts.

xhqboot help format

These are the template/module file formatting information.

```
[section_name]
# comments
! message line written to screen during startup/shutdown
line1    console priority waittime
line2    program string for run
line3    debug filename string (.out / .err appended)
```

Things to Note

- The **`section_name`** tag can be any name that does not conflict with a standard module name or another custom module name.
- You can have any number of comment or message lines.
- **`line1`**
  Must have exactly three numbers, usually 0 0 5, representing the following three items:
  1. **`console`**
     Is a flag indicating if a console window should be started to show execution of the command. Setting this to a 1 causes a separate console window to be started. Setting this value to a 0 does not start a new window.
  2. **`priority`**
     Is not supported for custom modules and must be set to 0.
  3. **`waittime`**
     Is the number of seconds for the xhqboot process to wait before continuing with the next command. This is normally set to 5 to give process enough time to start.
- **`line2`**
  is the command line to be run. This can be any valid command-line for the operating system.

- **`line3`**
  is the name of a file in the **`%XHQ_SERVER_LOGS%`** directory where output is written. This name must not conflict with any other name that is used by standard modules or other custom modules. You may leave this line blank if you do not wish to capture the processes output.
- A module can have any number of three-line command groups in it. However, the format must adhere strictly to the three-line specification above. In other words, be very careful about inserting blank lines where you do not intend to do so and so forth.

xhqboot help syntax

The property name syntax information. The general form of a property setting is as follows:

```
[[module.]action.]baseproperty=value
```

Where:

- **`baseproperty`** is the base property name (see topics)
- **`value`** is the setting for the property
- **`action`** is indicates the property is specific to that action (see actions)
- **`module`** is indicates the property is specific to a module

Additional Things to Note

- Specifying a property by only its base property name means the setting is valid for all actions and modules.
- Specifying a property by only for a specific action (shutdown or startup) means the setting valid for all modules while performing the action.
- Specifying a property by only for a specific module and action means the setting valid for that module while performing the action.
- The only restriction on the value is that a backslash (\) must be "escaped" that is preceded by another backslash (\\). This is important for paths on Windows computers since these contain backslash characters to delimit directories.
- For example, to specify that the directory c:\dev\test.jar be appended to the solution server's startup command line, you would use the following:

  ```
  xhq_sserver.startup.classpath.append=c:\dev\test.jar
  ```
- To include a Java™ option to every startup line, you would use:

  ```
  startup.javaopts=-Dfoo="bar"
  ```

xhqboot help classic

This specifies operating in "classic mode".

Value: True or false.

Applies to: All processing for all actions or specific actions.

Things to Note

- When this property is set, the template file is used directly for the indicated action. No modifications are performed and no custom modules loaded.
- When an action is prepend to the property (for example, `shutdown.classic`), the setting refers only to script processing during that action.
- Module names are not supported in the syntax of this property.

xhqboot help classpath.append

This is the Java™ class path to append.

Value: Semi-colon-separated list of directories or jar files.

Applies to: Standard modules only.

Special feature: Specific settings concatenated with more general ones.

The directories and/or jar files are appended to the standard `-classpath` setting in the command line. The family of `classpath.append` properties works by concatenating more general settings with more specific settings, with values separated by a semi-colon. In other words, if there is a module specific property (for example, `xhq_sserver.startup.classpath.append`) and a general setting (`classpath.append`), the two values are used together for the solution server start-up. The value of `classpath.append` is applied to all other command lines.

Example:

```
classpath.append=c:\\jars\\a.jar
startup.classpath.append=c:\\jars\\b.jar
xhq_cserver.startup.classpath.append=c:\\jars\\c.jar
```

Results in:

- All (Java-oriented) command lines during start up or shut down including the `c:\jars\a.jar` at the end of the classpath.
- All command lines during start up including `c:\jars\b.jar;c:\jars\a.jar` in the `-classpath` .
- The xhq_cserver comand line during start up including `c:\jars\c.jar;c:\jars\b.jar;c:\jars\a.jar` in the `-classpath` .

xhqboot help classpath.prepend

This is the Java™ class path to prepend.

Value: Semi-colon-separated list of directories or jar files.

Applies to: Standard modules only.

Special feature: Specific settings concatenated with more general ones.

The directories and/or jar files are prepended to the standard `-classpath` setting in the command line. The family of `classpath.prepend` properties works by concatenating more general settings with more specific settings, with values separated by a semi-colon. In other words, if there is a module specific property (for example, `xhq_sserver.startup.classpath.prepend`) and a general setting (`classpath.prepend`), the two values are used together for the solution server start-up. The value of `classpath.append` is applied to all other command lines.

Example:

```
classpath.prepend=c:\\jars\\a.jar
startup.classpath.prepend=c:\\jars\\b.jar
xhq_cserver.startup.classpath.prepend=c:\\jars\\c.jar
```

Results in:

- All (Java-oriented) command lines during start up or shut down including the "`c:\jars\a.jar`" at the beginning of the classpath.
- All command lines during start up including `c:\jars\b.jar;c:\jars\a.jar` in the `-classpath` .
- The `xhq_cserver` comand line during start up including `c:\jars\c.jar;c:\jars\b.jar;c:\jars\a.jar` in the `-classpath` .

xhqboot help custom.modules

The list of custom modules to use.

Value: Comma-separated list of custom module names.

Applies to: Custom modules only.
Custom module names are defined in the `xhqbootmods.dat` file.

Example: `startup.custom.modules=init,checksoln`

xhqboot help db.retries

The maximum number of retries to connect to the database if no connection can be established on the initial attempt. If no connection to the database can be established at the end of the final retry, the start-up sequence is halted. The xhqboot log file `<%XHQ_SERVER_LOGS%xhqbboot.out>` details the problems encountered while attempting to connect to the database.

Value: A number from 1 to 20.

Applies to: Start-up sequence only when using the XHQ High-Performance Database.

Default: 12

Example: `db.retries=12`

xhqboot help db.timeout

The time in milliseconds to wait between database connection attempts. A delay is used between each attempt to connect to the database only if the connection error is recoverable. For example, if the database is not yet available because it is not finished initializing, the next attempt is delayed by the specified time-out.

Value: A number from 0 to 300000.

Applies to: Start-up sequence only when using the XHQ High-Performance Database.

Default: 10000 (10 seconds)

Example: `db.timeout=10000`

xhqboot help ex_server

The standard XHQ Server.

```
%XHQ_SERVER_REPOS\xhqboot.properties:
#----------------------------------
# STANDARD XHQ SERVER CONFIG
modules=xhq_eserver,xhq_router,xhq_cserver,xhq_sserver
#----------------------------------
```

xhqboot help ex_ans

The standard XHQ Server with XHQ ANS.

```
%XHQ_SERVER_REPOS\xhqboot.properties:
#----------------------------------
# XHQ SERVER WITH ANS CONFIG
modules=xhq_eserver,xhq_router,xhq_cserver,xhq_sserver,xhq_aserver
#----------------------------------
```

xhqboot help ex_router

The special router-only configuration.

```
%XHQ_SERVER_REPOS\xhqboot.properties:
#----------------------------------
# ROUTER-ONLY NODE (for web-server on a separate node)
modules=xhq_router
xhq_router.startup.params=-printonly
#----------------------------------
```

xhqboot help ex_delay

The delayed start-up configuration.

```
%XHQ_SERVER_REPOS\xhqboot.properties:
#----------------------------------
# XHQ SERVER WITH DELAYED STARTUP
modules=xhq_eserver,xhq_router,xhq_cserver,xhq_sserver
startup.custom.modules=delay
#----------------------------------
%XHQ_SERVER_REPOS\xhqbootmods.dat:
#----------------------------------
# XHQ SERVER WITH DELAYED STARTUP
[delay]
!
! Delaying for 10 seconds...
!
0 0 10
%XHQ_SERVER_HOME%\bin\rep_sleep.exe 10
#----------------------------------
```

xhqboot help javaopts

These are the Java™ options to insert.

Value: List of valid Java™ command-line options.

Applies to: Standard modules only.

Special feature: Specific settings concatenated with more general ones.

Options are inserted into the command line. The family of `javaopts` properties works by concatenating more general settings with more specific settings. In other words, if there is a module specific property (for example, `xhq_sserver.startup.javaopts`) and a general setting (`javaopts`), the two values are used together for the solution server startup. The value of `javaopts` is applied to all other command lines.

Example:

```
javaopts=-verbose
startup.javaopts=-Dfoo="bar"
xhq_cserver.startup.javaopts=-Ddir="c:\\bin"
```

Results in:

- All (Java-oriented) command lines during start up or shut down including the `-verbose` option.
- All command lines during start up including `-Dfoo="bar" -verbose`.
- The `xhq_cserver` comand line during start up including `-Ddir="c:\\bin" -Dfoo="bar" -verbose` .

xhqboot help memory

These are the memory settings to use (-Xmx option).

Value:Maximum Java heap size.

Applies to: Standard modules only.

Setting a memory property overrides the Java heap size for the process specified in the template file. Typically, a memory property for specific modules during the startup is set. It is not recommended to set the non-specific action memory property or general memory property.

Example: `xhq_sserver.startup.memory=1200m`

xhqboot help modules

These are the list of standard modules to use.

Value: Comma-separated list of standard module names.

Applies to: Standard modules only.

These are the names of the standard modules:

`xhq_appserver`	XHQ Application Server
`xhq_aserver`	XHQ ANS Alert Server
`xhq_auditserver`	XHQ Audit Server
`xhq_cserver`	XHQ Cache Server
`xhq_dbhist`	XHQ Data Recorder
`xhq_eserver`	XHQ Enterprise (Model) Server

`xhq_historian`	XHQ Historian Server
`xhq_postscript`	Last few message/comment lines in template
`xhq_preamble`	Beginning few message/comment lines in template
`xhq_router`	XHQ Router Server
`xhq_rdsserver`	XHQ Relational Data Store Load Server
`xhq_sserver`	XHQ Solution Server (default, if no modules specified)
`xhq_ssopt`	XHQ Solution Server Storage Optimizer

Note that `xhq_preamble` and `xhq_postscript` are always included regardless of the value assigned to the modules. Standard modules required by those listed in the modules property are also included.

Example: `modules=xhq_sserver,xhq_aserver`

xhqboot help params.append

These are the parameters to append to the invocation line.

Value: Module-specific command-line parameters.

Applies to: Standard modules only.

Special feature: Specific settings concatenated with more general ones.

See also: `params.replace`

The command-line parameters are appended to the end of the module invocation command line. The family of `params.append` properties works by concatenating more general settings with more specific settings. In other words, if there is a module specific property (for example, `xhq_sserver.startup.params.append`) and a general setting (`params.append`), the two values are used together for the solution server startup. The value of `params.append` is applied to all other command lines.

Example:

```
params.append=-x1
startup.params.append=-x2
xhq_cserver.params.params.append=-x3
```

Results in:

- All (Java-oriented) command lines during start up or shut down including the -x1 at the end of the parameter list.
- All command lines during start up including -x2 -x1 at the end.
- The `xhq_cserver` command line during start up including -x3 -x2 -x1 at the end.

xhqboot help params.replace

These are the parameters to replace on the invocation line.

Value: Module-specific command-line parameters.

Applies to: Standard modules only.

The command-line parameters are replaced by the value specified by the property setting.

See also: `params.append`

Example: `xhq_cserver.params.params.replace=-pt33`

Results in: The `xhq_cserver` command-line during start up only having -pt33 as a parameter.

xhqboot help sequence

This is the sequence setting for a custom module.

Value: Position relative to another module.

Applies to: Custom modules only.

You must fully supply the module and action when using the sequence property. The value of the property is a plus (+) or a minus (-) followed by the name of another module (usually a standard module). This indicates when the custom module is invoked during the action sequence. A plus means after and a minus means before.

Example: `checksoln.startup.sequence=+xhq_sserver`

xhqboot help shutdown.dat

This is the template or resulting script for shutdown action. The template file is located in **`%XHQ_SERVER_HOME%\bin`**. The generated script file is located in **`%XHQ_SERVER_REPOS%\tmp`**.

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

You can examine the generated script file by using the test command-line parameter when running **`xhqboot`**.

xhqboot help startup.dat

This is the template or resulting script for startup action. The template file is located in **`%XHQ_SERVER_HOME%\bin`**. The generated script file is located in **`%XHQ_SERVER_REPOS%\tmp`**.

WARNING
Do not modify the `startup.dat` template file.

You can examine the generated script file by using the test command-line parameter when running **`xhqboot`**.

xhqboot help xhqboot.properties

These are the settings for **`xhqboot`** processing. This file is located in **`%XHQ_SERVER_REPOS%`**. This is a standard Java™ properties file and adheres to the following general syntax:

```
# comment line
propertyname=value
```

xhqboot help xhqbootmods.dat

These are the custom modules for **`xhqboot`** to process and place in script. This file is located in **`%XHQ_SERVER_REPOS%`**. It allows you to define custom modules to be run during start-up or shutdown. This is similar to the `startup.dat` and `shutdown.dat` template files in format.

How to run a batch file using the rep_run.exe program

The **xhqbootmods.dat** would be modified as follows. The custom module is called **xhqbatchcmd** in this example. Please note how quotes are used on the command-line.

The **rep_run** command launches a command and then waits until it finishes. Then, it writes a value to shared memory that indicates the process is finished and **rep_start** continues with the next command. The **-1** in the first line of the module definition indicates that it will wait until signaled in such a way.

If the batch file does not need to complete before the next command starts, then **rep_detach** can be used instead. This launches the program in the background and allows **rep_start** to continue with the next item.

Do not leave blank lines in xhqbootmods.dat. Either comment it or delete it.

EXAMPLE: XHQBOOTMODS.DAT FILE

```
  [comments]
#
# Repository custom boot modules
#
#   [section_name]
#
# comments
#   ! echo'd during startup
#   line1    console priority waittime
#   line2    program string for run
#   line3    debug file string (.out / .err appended)
#
[xhqbatchcmd]
!
! Running batch.bat...
!
0 0 -1
%XHQ_SERVER_HOME%\bin\rep_run "%XHQ_SERVER_REPOS%\bin\batch.bat"
```

In addition, *modify **xhqboot.properties** as shown in this example*. You need to know which module (for example, xhq_sserver) you want to start the command before (-) or after (+).

EXAMPLE: XHQBOOT.PROPERTIES FILE

```
# GENERATED BY XHQBOOT Sun Jan 23 05:32:08 PST 2005
#----------------------------------
# STANDARD XHQ SERVER CONFIG
modules=xhq_sserver
#----------------------------------

# The following was added to test a custom module (batch.bat)

##################################################
# Custom modules from custom templates
##################################################

# Configuration for custom modules (%XHQ_SERVER_REPOS%\xhqbootmods.dat)

# Indicate which custom modules should be used:
startup.custom.modules=batch

# The "batch" custom module should start after the "xhq_sserver" module
batch.startup.sequence=+xhq_sserver
```

For more information, refer to the topic, *About xhqboot.properties*.

2 | Security, Access, and Privileges

- XHQ Development Clients Security
- Configuring SSL in IIS for HTTPS
- XHQ Server Communication Security

XHQ Development Clients Security

As an information system, XHQ delivers valuable business information to people throughout an organization. In most cases, the information is freely available to help users in their decision-making. However, for various reasons, some information may be sensitive and its use should be limited to a selected audience.

About XHQ Security

The security features in XHQ emphasize preventing unauthorized users from accessing information within XHQ. In addition, you can control access to administrative functions to protect your XHQ solutions. Role Administrators can limit the use of development tools to authorized users.

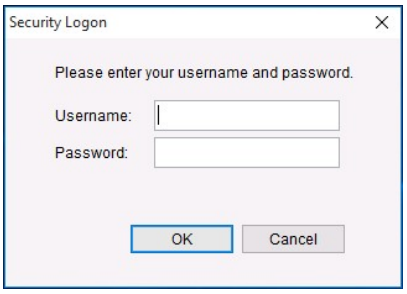
For information on Classic and Active Directory security, go to the topic *About Security Modes*.

Things to Note

- You can set security on **Extended Namespace Objects (ENO)** through the solution configuration import (since the "Roles" attribute is supported). An ENO is any component instance or group created in the extended namespace.
- Security settings may be applied to modeled group members and the standard XHQ security model applies.
- For associations, there is currently no way to specify configurations directly to objects and groups that are not represented by modeled members.

How Passwords Are Handled

Security is applied as the user logs on to either the browser client or to a designated development client application (for example, the XHQ Solution Builder). As soon as the user types in their security information, a username and password, it is encrypted.



Security Log on Dialog Box

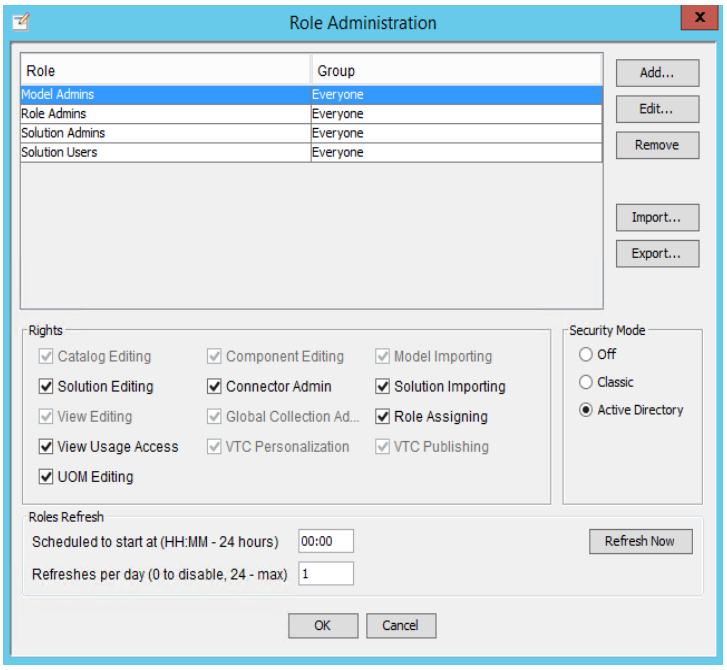
Applying Access and Privileges

Prior to setting user access and privileges, consider the following dependencies and requirements.

- XHQ relies on the operating system's *user database*. This database houses security information such as user account names and passwords, and the security policy settings.
- The users and groups are defined using existing services provided by the operating system (for example, Windows Active Directory Users and Computers).
- The user has the option of saving log on information. In order to use this feature in the browser client, cookies must be enabled on the browser.

Accessing the Role Administration Utility

The Role Administration utility allows you to manage roles, set access and privilege rights, and enable security. To access this utility, from the **XHQ Workbench**, go to the **View menu** and click **Role Administration**.



Role Administration Dialog Box

About Roles

Roles are typically functional role names such as 'Engineer', 'Operator', 'Manager', or 'Customer'. XHQ roles and mapping information are stored on the Enterprise Server.

Roles map to Windows groups and are stored in the local user database of the Enterprise Server or on a network domain controller. XHQ allows you to choose groups from either the domain or local groups on the XHQ Enterprise Server. A domain group can contain only domain members, whereas a local group can have both local and domain members.

Enterprise servers that are not members of a Windows domain must use local groups.

For XHQ requirements with regards to network domain controllers, see the topic, *Additional System Specifications and Restrictions*, located in the XHQ Installation Guide.

In general, the domain group is used. From an administrator's standpoint, you would want to choose a domain group because user accounts can be centrally managed.

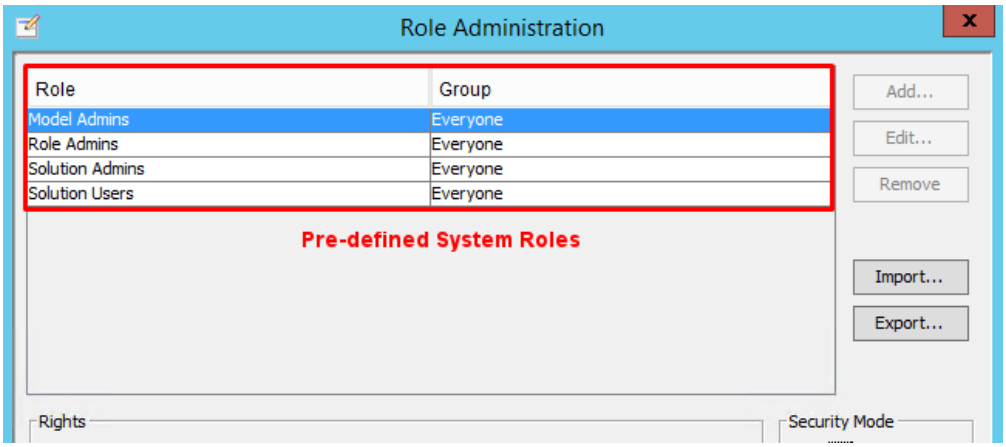
If the domain group is too general, however, then you can create groups locally, using the accounts in the domain.

XHQ roles must map to a Windows group that already exists. Depending on your model, you may have to create new groups that correspond to the roles you assign. This must be done prior to creating your roles through the Role Administration utility.

Accounts that are not in valid groups for XHQ roles are not eligible for impersonation.

Predefined Roles

In the Role Administration utility, there are four, "predefined" system roles: **Model Admins**, **Role Admins**, **Solution Admins** and **Solution Users**.



Role Administration Utility: List of pre-defined system roles and corresponding groups

These pre-defined system roles cannot be deleted. Initially, these roles are mapped to "Everyone." This "Everyone" group includes any user with a valid login for the domain or the Enterprise Server.

- Only Role Admins can access the Role Administration dialog to administer and assign roles and rights.
- Model Admins have the rights to assign roles to individual components within the XHQ Workbench.
- Solution Admins can perform solution imports and make changes to member access settings.

Initially, Role Admins, Model Admins, and Solution Admins are set to "Everyone". Therefore, to secure a system, you must map these predefined roles to another group that contains a limited set of users.

Unlike the previous administrator roles, the **Solution Users** role does not have any special rights, but is simply a pre-configured role you can use for the majority of users.

To define a **Super-User**, someone who has **all rights and open access**, you assign the same group to all three of the predefined administrator roles: Role Admins, Model Admins, and Solution Admins.

When removing users from a domain, please be aware that synchronization does not occur immediately. Consider the following example.

You have a domain user who is a member of two groups: A and B. In group A, the user is given Solution Admin access. In group B, however, the user is given Solution User access. The user is then removed from group A; in which case, the user should no longer have administrative privileges.

The time it takes for this change to take place (the synchronization, or the convergence time) does not occur immediately. Depending on how your domain is set-up (for example, the size of the domain, the number of domain users, the number of domain controllers, and so forth), the period before synchronization occurs could last several minutes to several days.

Should you need changes to occur immediately, please contact your domain administrator to "force" synchronization.

About Security Modes

In order to map a role or change access rights, select either **Classic** or **Active Directory** security mode.

Security Mode Options

If you select . . .	Then . . .
Classic	Security is on, but integrated security is not enabled.
Active Directory	Security is on and integrated security is enabled. This is the default.

Using Classic Security

If Classic Security is **enabled**, the user is **prompted** for a user name and password each time he/she attempts to launch either the browser client (the XHQ Solution Viewer) or a designated development client application (for example, the XHQ Solution Builder).

Using Active Directory Security

If Active Directory security is **enabled**, the user is **not prompted** for a user name and password when attempting to launch either the browser client or a designated development client application. In this case, the current Windows user information on the client computer is used for authentication. This is a secure form of authentication because the user name and password are not sent across the network.

When using integrated security with the XHQ Solution Viewer (browser client), you must use Microsoft IIS as the web server and Internet Explorer as the browser.

For a list of supported versions, go to the topic, *System Requirements*, located in the XHQ Installation Guide.

However, to enable Active Directory security, you must first execute the following tasks OUTSIDE of the XHQ System:

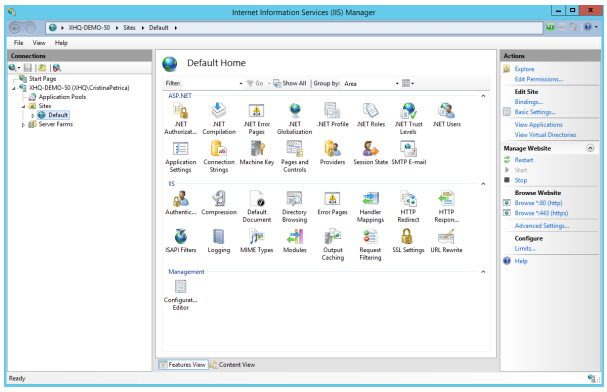
- From the **Web Server**,
  Disable Anonymous Access, and enable Integrated Windows Authentication. See the topic, *To configure the web server for Active Directory security*.
- From **Internet Explorer**,
  Configure User Authentication. See the topic, *To configure IE for Active Directory security*.

XHQ Active Directory security supports nested groups but with one exception:
Due to a limitation of the Win32 API, the XHQ system cannot retrieve members of a global group from the domain server if those members are also "groups." In this case, roles that have global groups use a scheme of comparing the role group name with the global groups to which the user belongs.

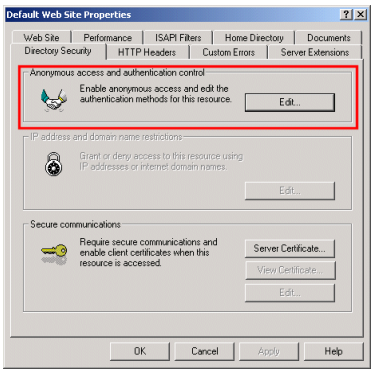
Once these two tasks are complete, then you can enable security from the **Role Administration** dialog box of the XHQ Workbench. See the topic, *To enable Active Directory security from the Workbench*.

To configure the web server for Active Directory security

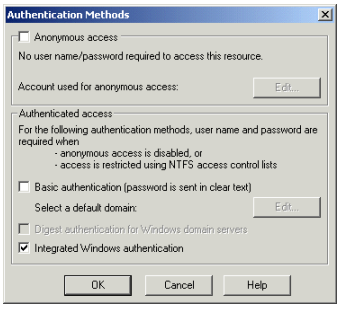
1. Launch the **Internet Service Manager**.
2. Expand Internet Information Services to **Default Web Site**.



3. **Right-click** on Default Web Site and, from the menu, click **Properties**.
   The "Default Web Site Properties" dialog box appears.
4. Select the **Directory Security tab**.



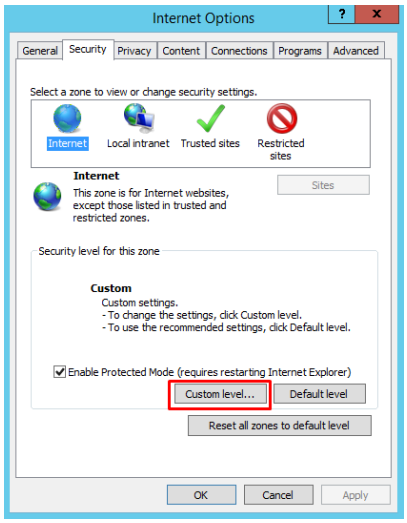
5. Under **Anonymous access and authentication control**, click the **Edit** button.
   The "Authentication Methods" dialog box appears.



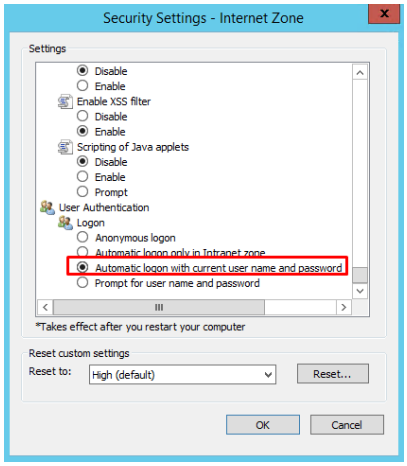
6. Disable (deselect) Anonymous access, and enable (select) **Integrated Windows Authentication**.
7. Click **OK** to save your changes and close the dialog boxes.
8. **Save** the changes to IIS.

To configure IE for Active Directory security

1. From the **Tools** menu, click **Internet Options**.
   The "Internet Options" dialog box appears.



2. Select the **Security tab** and click the **Custom Level** button.
   The "Security Settings" dialog box appears.



3. Scroll down to **User Authentication** and enable (select) **Automatic logon with current username and password**.
4. Click **OK** to save your changes and close the dialog boxes.

To enable Active Directory security from the XHQ Workbench

1. From the **View** menu, click **Role Administration**.
   The "Role Administration" dialog box appears.
2. Under **Security Mode**, click **Active Directory** and click **OK**.

To view a solution, with Active Directory security enabled, from a browser

When Active Directory security is enabled, `xhq.aspx` or xhqNavbar.aspx (if using the XHQ NavBar) is used instead of `xhq.html` or `xhqNavBar.html` to view the solution from a browser.

1. Open your browser.
2. In your browser's URL field, type the web server location for your solution.
   The address can either be a fully qualified web page URL or an IP address.

Example: **Of server name/html pages**

`http://localhost/indx/xhqNavbar.aspx`

Example: **Of IP addresses**

`http://205.6.35.224/indx/xhqNavbar.aspx`

If the XHQ Solution Viewer Components is installed, you can also launch your solution directly by accessing the `xhqNavbar.aspx` file from the browser. This file is located in the XHQ Web Root directory, for which the default path is `C:\Program Files (x86)\XHQ\XHQ Web Root\indx\xhqNavbar.aspx`.

How security works with xhqci

Here are some things to note with regards to how `xhqci` works with Classic/Active Directory security.

If Security (either Classic or Active Directory) is ENABLED:

- And a username is provided as a command-line parameter, `xhqci` follows the **Classic security model** and expects a password. Both username and password are then sent to the server.
- And a username is not provided, `xhqci` follows the **Active Directory security model** and acquires the logged in user's username from the system. Only the username is sent to the server; a password is not sent.

Managing Roles

In this section, you will learn how to:

- Map a predefined role to a group(s).
- Create a new role.
- Edit an existing role.
- Delete a role.

Important Things to Note

- In order to map, create, edit, or delete a role, security must be enabled.
- XHQ roles must map to Windows groups that already exists. You may have to create new groups that correspond to the roles you assign. This must be done prior to creating new roles. If necessary, work with your system administrator to get this done.

To create a new role

The roles you create must map to existing Windows groups. These roles are stored as part of the XHQ repository.

Only **Role Admins** can access the Role Administration utility.

1. Launch the **XHQ Workbench**.
2. From the **View** menu, click **Role Administration**.
   The "Role Administration" dialog box appears.

In order to add a new role, select either Classic or Active Directory security. Setting the security enables the command buttons and the "Rights" group box.

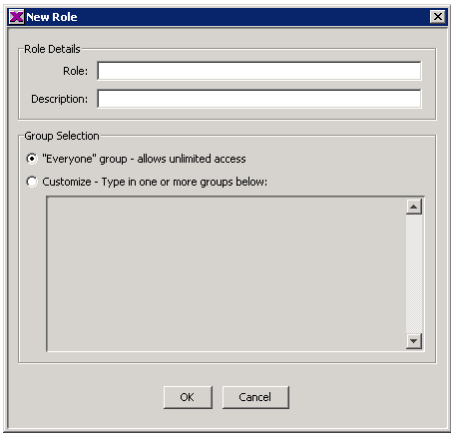
3. Under **Security Mode**, do one of the following:

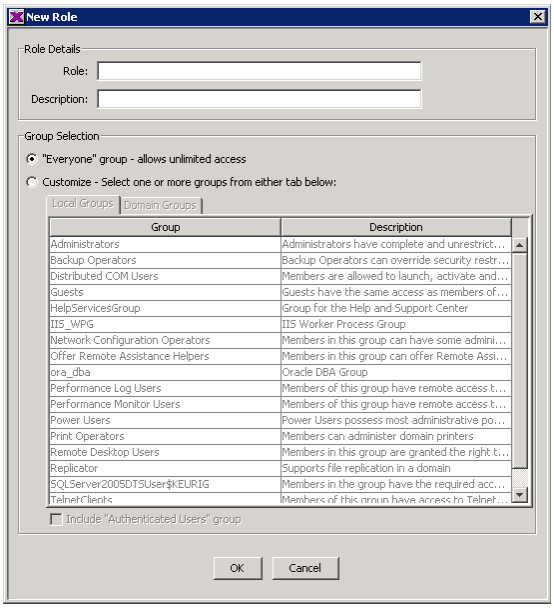
Click this . . .	If you want . . .
Classic	Security enabled, but integrated security disabled.
Active Directory	Both classic security and integrated security enabled. This is the default.

4. Click **Add**.
   Depending on the **`DisableRoleSecurityGroupsTable`** global property setting, one of the following "New Role" dialog boxes will appear.

If **`DisableRoleSecurityGroupsTable`**=TRUE (which is the default):

If **`DisableRoleSecurityGroupsTable`**=FALSE:





For more information on why there is a difference in the Groups option, go to the topic, *More About Groups*.

5. Enter the **Role** name.
6. Enter a Description.
7. Under **Group Selection**, do one of the following:
   - Click **Everyone**.

   or

   - Click **Customize**, then enter or select (depending on your setting) a Group (or Groups) that map to the role.
8. OPTIONAL
   Select (check) Include "Authenticated Users" group.
9. Click **OK** to save and close the dialog box.
   This returns you to the "Role Administration" dialog box. Note, the new role is added to the Role table.
10. Click **OK**.
    A message appears.
11. Click **OK** and **restart** XHQ Enterprise Server.

To edit an existing role

Only **Role Admins** can access the Role Administration Utility.

1. Launch the **XHQ Workbench**.
2. From the **View** menu, click **Role Administration**.
   The "Role Administration" dialog box appears.

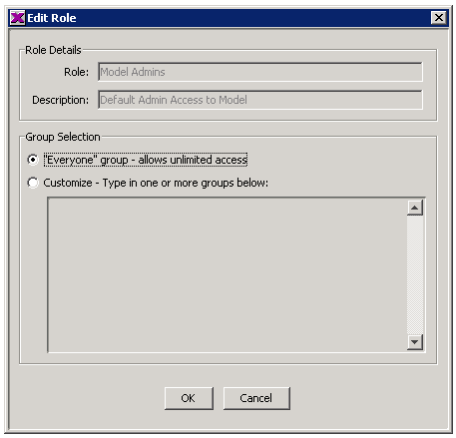
   In order to edit an existing role, you must select either Classic or Active Directory security. Setting the security enables the command buttons and the "Rights" group box.

3. Under **Security Mode**, do one of the following:

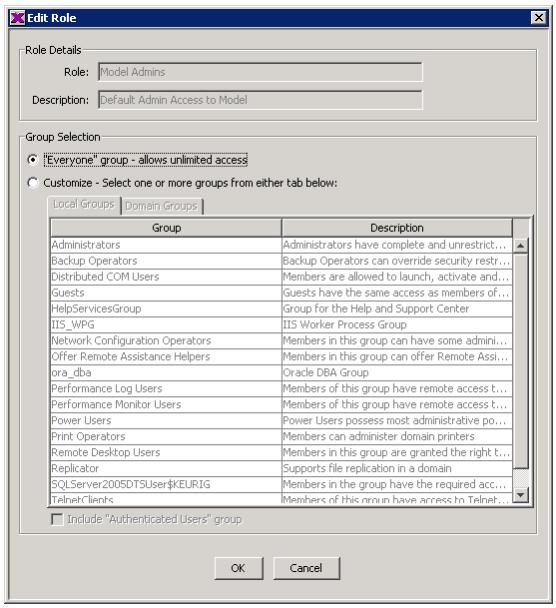
Click this . . .	If you want . . .
Classic	Security enabled, but integrated security disabled.
Active Directory	Both classic security and integrated security enabled. This is the default.

4. Click **Add**.
   Depending on the **`DisableRoleSecurityGroupsTable`** global property setting, one of the following "Edit Role" dialog boxes will appear.

If **`DisableRoleSecurityGroupsTable`**=TRUE (which is the default):



If **`DisableRoleSecurityGroupsTable`**=FALSE:



For more information on why there is a difference in the Groups option, go to the topic, *More About Groups*.

5. Under **Group Selection**, do one of the following:
   - Click **Everyone**.

or

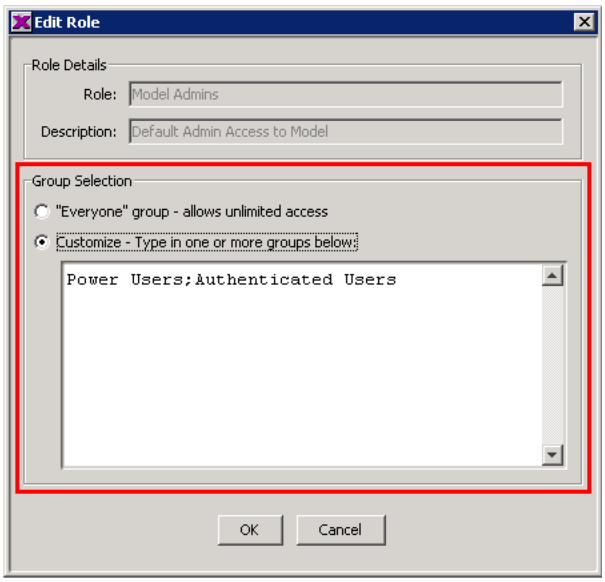
- Click **Customize**, then enter or select (depending on your setting) a Group (or Groups) that map to the role.

6. OPTIONAL
   Select (check) **Include "Authenticated Users"** group.
7. Click **OK** to save and close the dialog box.
   This returns you to the "Role Administration" dialog box.
8. Click **OK**.
   A message appears.
9. Click **OK** and **restart** XHQ Enterprise Server.

More About Groups

With security enabled, you can allow access for all users (Everyone) or limit access to a select few (Customize option) when adding or editing a role.

By default, the **Customize** option is a text box where you can enter valid group names.



Edit Role: Customize option as a text box

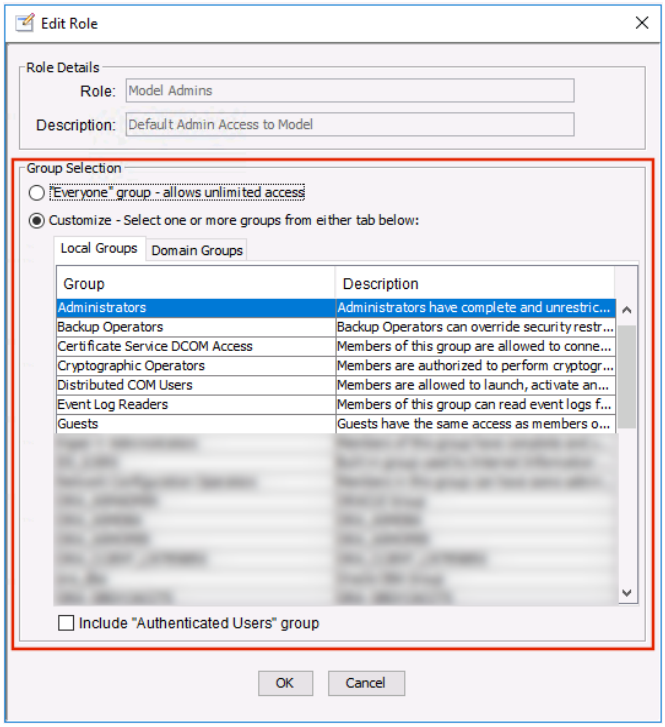
Use a **semi-colon** to separate each group name, and use a `D\` prefix to indicate a **domain group**.

Local group names do not require a prefix; the `L\` prefix is optional.

Examples:

```
Operators;D\Domain Users;Authenticated Users
D\Domain Admins;L\Users
```

If, however, you are unfamiliar with the names of existing Windows groups, you can set the Customize option so that, instead of a text box, it displays a table that automatically displays all existing groups. This is done by setting the **`DisableRoleSecurityGroupsTable`** global property to FALSE.



Edit Role: With DisableRoleSecurityGroupsTable global property set to False

To map to a **Domain Group**, click the Domain Groups tab. For **Authenticated Users**, select (check) the option at the bottom of the table.

For more information on setting the `DisableRoleSecurityGroupsTable` global property, go to the topic, *Working with PROPERTIES Files*. See the topics on *Using globalsettings.properties* and *Additional Global Settings*.

Additional Things to Note About Groups

- An XHQ role may be associated with zero, one, or more Windows-defined group(s).
- If a role is associated with **Everyone**, then the role is applicable to any valid user.
- If a role is associated with one or more groups, the user must be a member of at least one of the groups in order for the role to apply to that user.
- To indicate multiple groups when the Customize option is a **text box**, use semi-colons to separate each group name. When the Customize option displays a **table**, use the SHIFT key to select a cluster of consecutive groups or the CTRL key to select individual groups.

To map a predefined role to a group

Only **Role Admins** can access the Role Administration Utility.

1. Launch the **XHQ Workbench**.
2. From the **View** menu, click **Role Administration**.
   The "Role Administration" dialog box appears.

In order to map a predefined role, you must select either Classic or Active Directory security. Setting the security enables the command buttons and the "Rights" group box.

3. Under **Security Mode**, do one of the following:

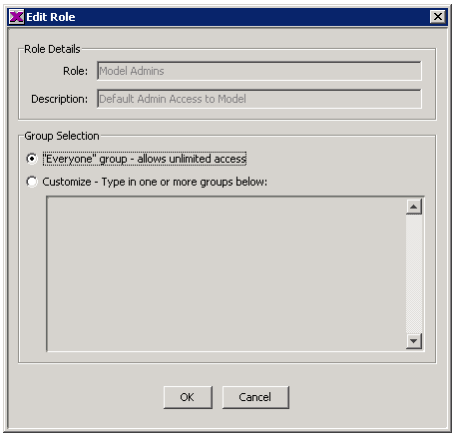
Click this . . .	If you want . . .
Classic	Security enabled, but integrated security disabled.
Active Directory	Both classic security and integrated security enabled. This is the default.

4. Select a predefined role.

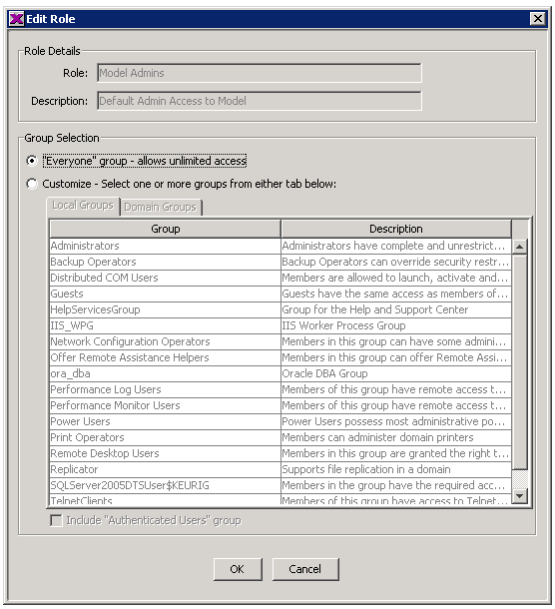
   For information on predefined roles, go to the topic, *About Roles*.

5. Click **Edit**.
   Depending on the **`DisableRoleSecurityGroupsTable`** global property setting, one of the following "Edit Role" dialog boxes will appear.

If **`DisableRoleSecurityGroupsTable`**=TRUE (which is the default):



If **`DisableRoleSecurityGroupsTable`**=FALSE:



For more information on why there is a difference in the Groups option, go to the topic, *More About Groups*.

6. Under **Group Selection**, do one of the following:
   - Click **Everyone**.

   or

   - Click **Customize**, then enter or select (depending on your setting) a Group (or Groups) that map to the role.
7. OPTIONAL
   Select (check) Include "Authenticated Users" group.
8. Click **OK** to save and close the dialog box.
   This returns you to the "Role Administration" dialog box.
9. Click **OK**.
   The following message appears.

10. Click **OK** and **restart** XHQ Enterprise Server.

Deleting Roles

You must be careful when deleting roles. If you delete a role that has already been selected by a component/member, then any access that role was granted will no longer be available.

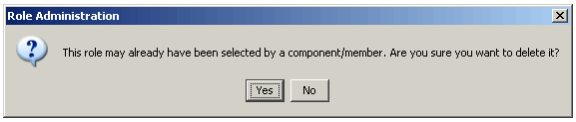
For example, if roles A and B had access to a node in the solution and B was deleted, only members of A and of Solution Admins will have access to the solution.

Predefined system roles cannot be deleted.

To delete a role

Only **Role Admins** can access the Role Administration Utility.

1. Launch the **XHQ Workbench**.
2. From the **View** menu, click **Role Administration**.
   The "Role Administration" dialog box appears.
3. From the Role table, select (highlight) the role and click **Remove**.
   A message appears.



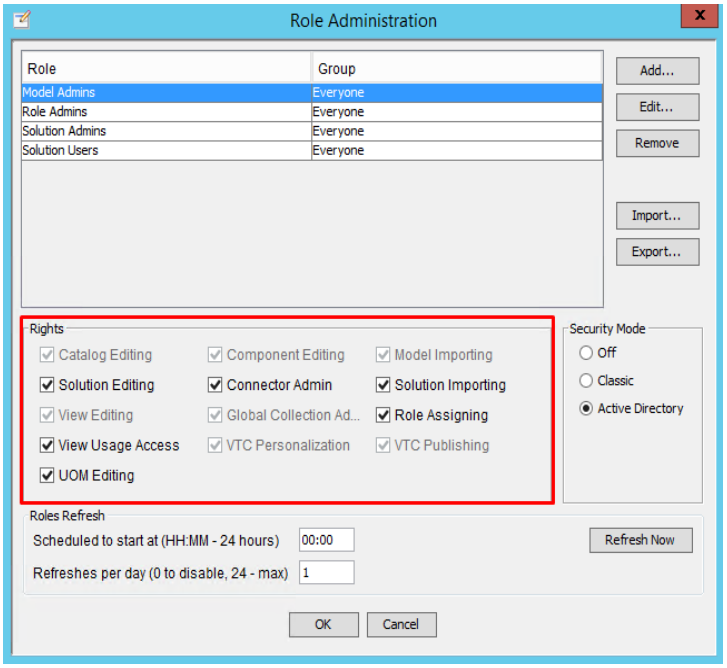
4. Click **Yes**.
   This returns you to the "Role Administration" dialog box. Note, the role is removed from the Role table.

Limiting Rights

Depending on your security model, you may want to limit access to one or more XHQ tools. Initially, access is granted to "Everyone" for all predefined roles.

XHQ **Rights** are set using the Role Administration Utility.

Only **Role Admins** can access the Role Administration Utility.



Role Administration Utility – Assignable Rights

Assignable XHQ Rights

XHQ Rights	Descriptions
Catalog Editing	This right grants access to the XHQ Workbench, Manage Pages, export model, export views, Search Views, and Generate Summaries. This right does not grant access to read components and views in XHQ Workbench. **Important**: **Component Editing** is required to create a new component.
Solution Editing	This right grants access to the XHQ Solution Builder. Allows read access to the solution configurations, including the global collections. Allows solution tree configuration editing (excluding the global collection), Update to Latest Components, and VTC exports.
View Editing	This right grants access to edit views, create view, delete views, edit view properties, create new view, and import view. **Important**: You must also have **Catalog Editing** rights in order to launch the XHQ Workbench. **Note**: If a role has "View Editing" rights and is also associated with a specific component, then this role will be able to edit the given component's views and

XHQ Rights	Descriptions
	view properties, which includes the ability to delete views. The "Component Editing" right is not required to have "View Editing" rights.
View Usage Access	This right grants access to the view usage reports.
UOM Editing	This right grants the ability to create, delete, or modify units of measure and categories. **Note**: Currently, this is only supported through `xhqci` commands.
Component Editing	This right grants access to create a new component and edit/delete/rename an existing component. **Important:** You must also have **Catalog Editing** rights in order to launch the XHQ Workbench. And you must have Model Importing rights to be able to import components. **Note**: If a role has "Component Editing" rights and is also associated with a specific component, then this role will be able to edit the given component, which includes the ability to delete it.
Connector Admin	This right grants access to add, edit, remove, import and export connection processes and groups. It also grants access to the Connector Administration utility, allowing you to add, edit, or remove connector types. **Important**: You must also have **Solution Editing** rights in order to launch the XHQ Solution Builder and use the Connection Group and Connection Process dialogs.
Global Collection Admin	This right enables the user to create, rename, delete, show references, or request update for any global collection. **Important**: You must also have **Solution Editing** rights in order to launch the XHQ Solution Builder.
VTC Personalization	This right grants basic access to the XHQ Visual Tile Composer. Users can only create non-reusable, non-published Tile Composed Views (TCVs).
Model Importing	This right allows the user to import models, including their associated views and configurations. **Important**: You must also have **Catalog Editing** rights in order to launch the XHQ Workbench. **Note**: Model Importing rights grant the user the permission to delete and update components, which includes the associated views, view properties, and assigning pages to components. It is also possible to delete the root solution with this permission. A user cannot delete or update a component without having access to the component.
Solution Importing	This right allows the user to import the solution tree and global collection configurations. **Important**: You must also have **Solution Editing** rights in order to launch the XHQ Solution Builder. A user cannot import the configuration for a secured path, member, or global

XHQ Rights	Descriptions
	collection to which the user does not have access.
Role Assigning	This right allows the user to assign roles to components, using the right-click shortcut menu command from the XHQ Workbench.
VTC Publishing	This right grants access to the XHQ Visual Tile Composer. Users can create re-usable TCVs and publish TCVs.

To grant/deny Rights

Only **Role Admins** can access the Role Administration Utility.

1. Launch the **XHQ Workbench**.
2. From the **View** menu, click **Role Administration**.
   The "Role Administration" dialog box appears.

In order to edit an existing role, you must select either Classic or Active Directory security. Setting the security enables the command buttons and the "Rights" group box.

3. Under **Security Mode**, do one of the following:

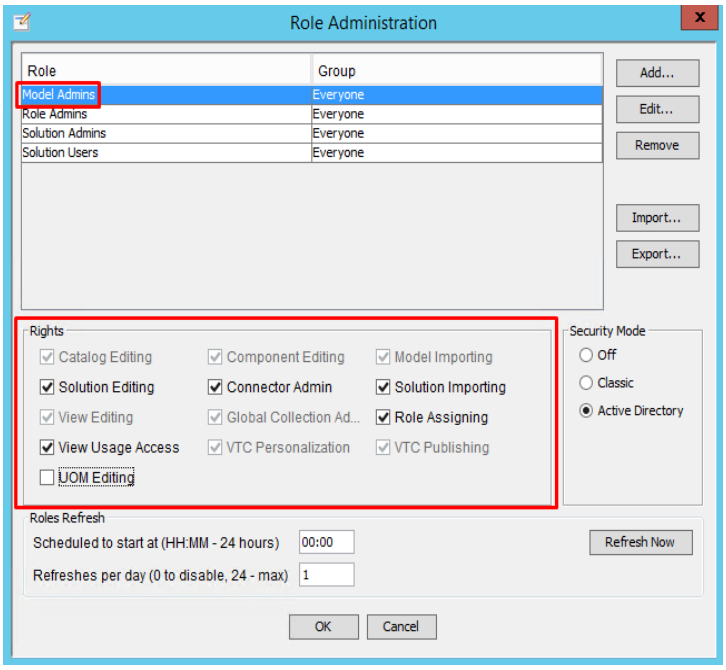
Click this . . .	If you want . . .
Classic	Security enabled, but integrated security disabled.
Active Directory	Both classic security and integrated security enabled. This is the default.

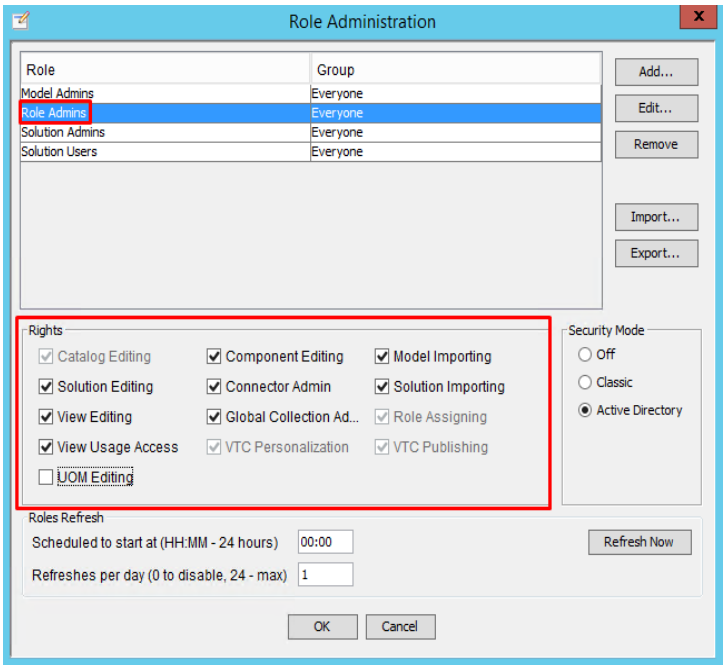
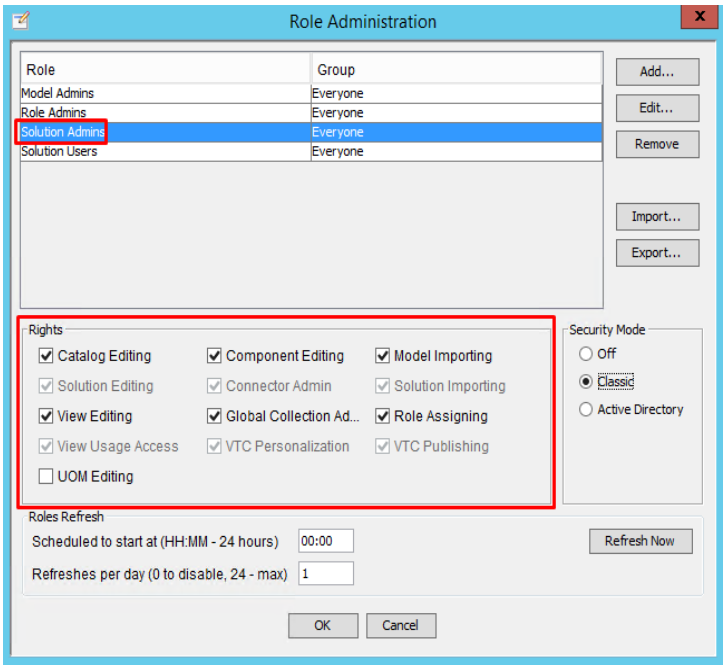
4. Select a **Role**.
5. Under **Rights**, select (to grant access to) or deselect (to deny access to) the items for the given role.
6. Click **OK**.
   A message appears, verifying that security has been enabled and that you must restart the Enterprise Server.
7. Click **OK** and **restart** the XHQ Enterprise Server.

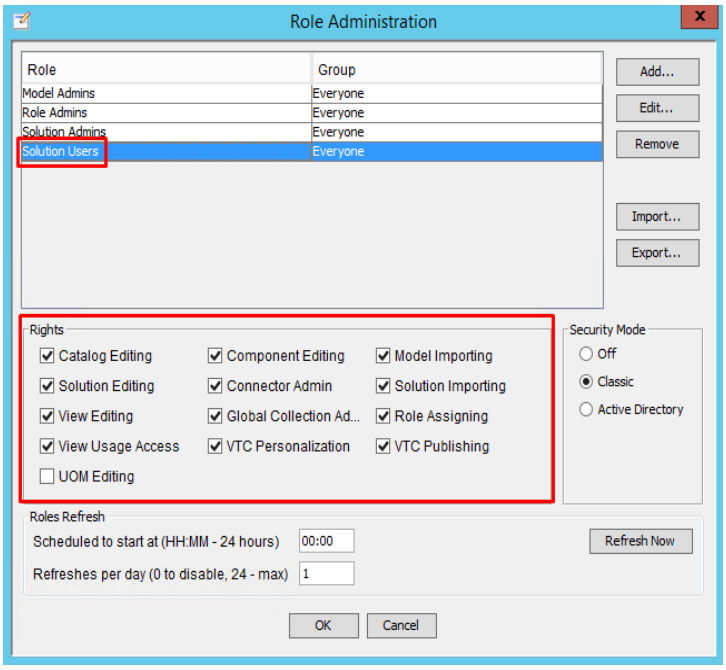
About Rights and Predefined Roles

There are four, "predefined" system roles: **Model Admins**, **Role Admins**, **Solution Admins** and **Solution Users**. Each of these roles is associated with a set of default, "permanent" rights that cannot be disabled.

Rights Associated with Predefined System Roles

Role	Asssociated Rights (Default Setting)
Model Admins Permanent Rights: Catalog Editing View Editing Component Editing Global Collection Admin Model Importing VTC Personalization VTC Publishing	 For XHQ Workbench, the Model Admins predefined role has "Role Assigning" rights even if the box is left unchecked. For `xhqci importroles`, however, the "Role Assigning" right has to be explicitly checked.

Role	Asssociated Rights (Default Setting)
Role Admins Permanent Rights: Catalog Editing Role Assigning VTC Personalization VTC Publishing	
Solution Admins Permanent Rights: Solution Editing View Usage Access Connector Admin Solution Importing VTC Personalization VTC Publishing Audit Trail Admin	

Role	Asssociated Rights (Default Setting)
Solution User Permanent Rights: <None>	

About Visual Tile Composer Access

To enable the Visual Tile Composer (VTC) icon on the Solution Viewer NavBar, the user must be granted either VTC Personalization or VTC Publishing rights (in addition to setting the `ShowTileComposerLink` global property).

With **VTC Personalization**, users can only create non-reusable, non-published Tile Composed Views (TCVs).

With **VTC Publishing**, users can create re-usable TCVs and/or publish TCVs.

A user with **VTC Personalization** permission cannot edit/delete VTC views published by other users.

For more information, see the following:

- XHQ Developer's Guide > *Working with View Tiles and the Visual Tile Composer*
- XHQ Solution Viewer User's Guide > *Visual Tile Composer*

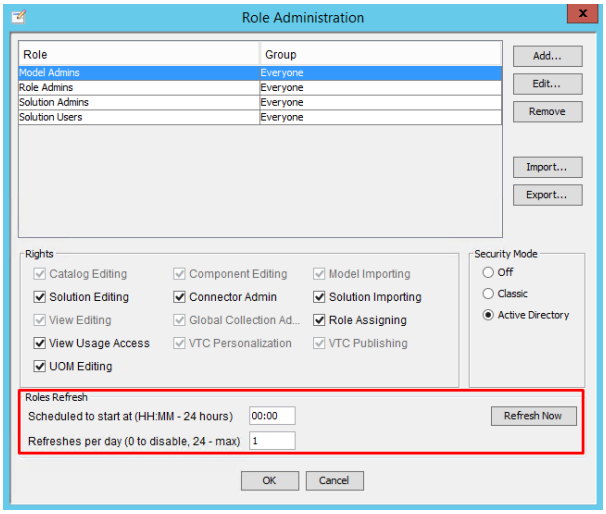
About Security and File Recovery

XHQ features a file recovery mechanism for the XHQ Workbench. If you lose a network link while working in either of these environments, a message appears indicating that the network connection has been broken, and your work will be saved to a recovery file on your local machine. Prior to retrieving recovery files, however, XHQ automatically performs an access/security check. If security has been enabled, the following may occur:

- The XHQ Workbench will launch only if the user is a Model Admin or has Component Editing permissions on the component.
- The XHQ Workbench will launch only if the user is a Model Admin or has View Editing permissions on the component.

Refreshing Roles

Setting security (Classic or Active Directory) enables the **Roles Refresh** options. You can set a daily schedule to refresh the XHQ roles, or elect to force an immediately refresh by clicking **Refresh Now**.



Roles Refresh

Important Things to Note

- You must have Role Admins permissions.
- In addition to this utility, you may also use the `xhqci` command to force a roles refresh or set the daily refresh schedule. The `xhqci` command to force a roles refresh is also available through a Web Service.

  For details on the xhqci command, see the topic, *Refreshing Roles*.

  For details on the XHQ Web Services, see the topic, *Web Services*, located in the XHQ SDK Reference Guide.

- For Kerberos authentication, if a user's group membership changes in the Active Directory and the user has already been authenticated by XHQ in a previous session, a roles refresh is required in order for the group membership change to take effect.

  For details, see the topic, *Kerberos Authentication*.

To refresh XHQ Roles using the Role Administration utility

1. From the **Role Administration**, set the **Security Mode** to either **Classic** or **Active Directory**.
   This enables the "Roles Refresh" group.

Setting a daily roles refresh schedule

2. For **Scheduled to start at**, enter the time when the first refresh of the day occurs, using the format **`HH:MM`**.
   - `HH` is the **hour** in 24-hour format.
   - `MM` is the **minutes**.

   The default is one refresh daily at midnight.

3. For **Refreshes per day**, enter the daily refresh frequency.

   Zero (0) means no automatic refresh.

   The number of refreshes should be kept to a minimum to avoid unnecessary cache purging and refreshing.

Forcing an immediate refresh of XHQ roles

4. Click **Refresh Now**.
5. Click **OK**.

Setting Permissions

Permissions for the XHQ Workbench – Main Menu

The following table lists selected **main menu** commands from the **XHQ Workbench** and the permissions allowed if security is enabled and rights are assigned.

If permissions have not been granted, then menu commands are disabled or an "Access Denied" message appears.

XHQ Workbench Menu Command	Who Has Permissions?
[To launch XHQ Workbench]	Model Admins role *or* Catalog Editing rights enabled
File Menu -> New Components	Model Admins role *or* Catalog Editing rights enabled
File Menu -> Import Components	Model Admins role *or* Catalog Editing rights enabled
File Menu -> Manage Pages	Model Admins role *or* Catalog Editing rights enabled
View Menu -> Role Administration	Role Admins only

Permissions for the XHQ Workbench – Shortcut Menu

The following table lists selected **right-click shortcut menu** commands from the **XHQ Workbench** and the permissions allowed if security is enabled and rights are assigned.

If permissions have not been granted, then menu commands are disabled or an "Access Denied" message appears.

XHQ Workbench Shortcut Menu Command	Who Has Permissions?
Assign Roles	Model Admins role *or* Role Assigning rights enabled **Note:** This option is enabled only when Security is enabled.
Set Page	Model Admins role *or* Catalog Editing rights enabled
Edit Component Rename Component Delete Component	Model Admins role *or* Component Editing rights enabled and the role must be assigned to the given component.
New View Edit View Import View Delete View	Model Admins role *or* View Editing rights enabled and the role must be assigned to the given component.
View Properties	Model Admins role *or* View Editing rights enabled and the role must be assigned to the given component.

Permissions for View Building

The following table lists selected button bar options from the **XHQ Workbench** and the permissions allowed if security is enabled and rights are assigned.

If permissions have not been granted, then menu commands are disabled or an "Access Denied" message appears.

Button Bar Options	Who Has Permissions?
Launch Component Builder	Model Admins role *or* View Editing rights enabled and the role must be assigned to the given component.
Launch View Builder	Model Admins role *or* View Editing rights enabled and the role must be assigned to the given component. **Note:** If the selected view element is NOT an embedded view, the button is disabled.

Permissions for the Solution Builder – Main Menu

The following table lists selected **main menu** commands from the **Solution Builder** and the permissions allowed if security is enabled and rights are assigned.

If permissions have not been granted, then menu commands are disabled or an "Access Denied" message appears.

Solution Builder Menu Command	Who Has Permissions?
[To launch Solution Builder]	Solution Admins role *or* Solution Editing rights enabled
Solution Menu -> Import Configuration	Solution Admins role *or* Solution Importing rights enabled
Configure Menu -> Connection Groups	Solution Admins role *or* Connector Admin rights enabled
Configure Menu -> Connection Processes	Solution Admins role *or*

Solution Builder Menu Command	Who Has Permissions?
	Connector Admin rights enabled

Permissions for Global Collection Access

The following table lists selected right-click shortcut menu commands when pointing to a **global collection** from the Solution Builder and the permissions allowed if security is enabled and rights are assigned.

In order to create, rename, or delete collections, you must have:

- Global Collection Admin rights

or

- Component Editing rights and the role you belong to is assigned to the component that is associated with the given collection.

If permissions have not been granted, then menu commands are disabled or an "Access Denied" message appears.

Collections Shortcut Menu Command	Who Has Permissions?
New Collection	Global Collection Admin rights enabled *or* Component Editing rights enabled and the role must be assigned to the given component that is associated with the collection.
Rename	Global Collection Admin rights enabled *or* Component Editing rights enabled and the role must be assigned to the given component that is associated with the collection.
Delete	Global Collection Admin rights enabled *or* Component Editing rights enabled and the role must be assigned to the given component that is associated with the collection.
Save Configuration	Global Collection Admin rights enabled *or* Component Editing rights enabled and the role must be assigned to the given component that is associated with the collection.

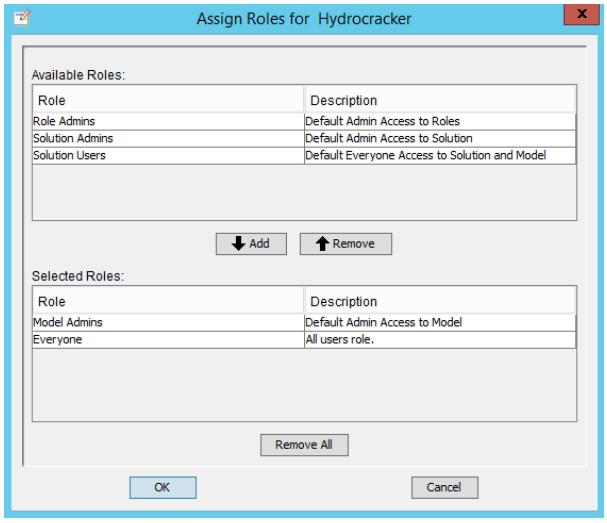
Managing Component Access

The Role Assigning right allows you to associate components to one or more roles for the purposes of editing the component's contents. This includes the component detailed definition, views of the component, and global collections associated with the component.

By default, a component is associated with the group "Everyone", and so initially all users are able to assign roles to a component.

To assign a role (or roles) to a component

1. With security enabled, you must have **Role Assigning** rights.
2. From the **XHQ Workbench**, right-click on a given component.
   The right-click shortcut menu appears.
3. Click **Assign Roles**.
   The "Assign Roles for [Component Name]" dialog box appears.



4. From the list of **Selected Roles**, do one of the following:
   - Click **Add** to add a role to the list of roles associated with the given component.

   or

   - Click **Remove** to delete a role from the list of roles associated with the given component.

   or

   - Click **Remove All** to delete all roles, except Model Admins, from the list.
5. Click **OK**.

Exporting and Importing Component Roles

From the Role Administration Utility, you have the option of importing or exporting component roles.

To learn how to export or import component roles using the **command-line**, go to the topic, *Using the Command-Line to Export/Import Component Roles*, located in the XHQ Developer's Guide.

The export file includes all roles assigned for all components. The Model Admin role is implicitly included in the export file against all components. This means that even though the export file does not explicitly include the Model Admin role, it is automatically assumed that the role is assigned to all components.

The import file also contains all of the components and the role assignment(s) for each component. As with the export file, the Model Admin role is also [implicitly] assigned to all components.

You must have both **Role Assigning** and **Model Importing** rights to import component roles.

If an invalid role is encountered in the import file, then the role is ignored and the import process continues. If all roles for a given component are invalid, then the only role assigned to that component will be the Model Admin role.

When editing the XML import file, use tools specifically designed for XML editing. Do not use text editors (such as Notepad) to edit because they do not properly save files in UTF-8 format.

About Role Mapping

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.

Role Mapping: Defining the XML format

The outermost tag in any security (or role mapping) XML document that is imported to or exported from XHQ is **`XHQ:RoleMap`**. All the attributes to this node-list tag provide information regarding the XHQ environment.

Attributes of the XHQ:RoleMap Tag

Attribute	Description	
xmlns:XHQ	The URL to the XHQ web site.	
esHost	The name of the enterprise server machine.	
esTime	The time, in milliseconds, on the enterprise server when the import process begins (since the 'epoch').	
esFormattedTime	The time on the enterprise server when the import process begins.	
xhqVersion	The XHQ Platform release version.	
xhqBuild	The XHQ Platform build number.	
securityMode	If this attribute is set to:	Then:
	off	Security is turned off.
	classic	Security is on, but integrated security is not enabled.
	active directory	Security is on and integrated security is enabled.
	<nothing>	Security defaults to "off".
	Note: This value for the `securityMode` attribute is **case in-sensitive**.	

Each XHQ Role is represented as an **`XHQ:RoleMapEntry`** node-list tag.

Attributes of the XHQ:RoleMapEntry Tag

Attribute	Description
name	The name of the XHQ Role.
group	The name of the corresponding Windows group.

Attribute	Description	
local	If this attribute is set to:	Then:
	true	The Windows group is local to the Model Server host.
	false	The Windows group is NOT local to the Model Server host.
	Note: This value for the `local` attribute is **case in-sensitive**.	

For each right granted to an XHQ Role an **`XHQ:Right`** node tag is created.

Attributes of the XHQ:Right Tag

Attribute	Description
name	The name of the XHQ Rights, which may be one of the following: • Catalog_Editing • Solution_Editing • View_Editing • View_Usage_Access • Component_Editing • Connector_Admin • Global_Collection_Admin • Model_Importing • Solution_Importing • Role_Assigning
	See Also: For a detailed description of these rights, go to topic *Limiting Rights*.

The Role Mapping DTD

The DTD (Document Type Definition) for the security (role mapping) 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: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>
```

Example: XML document for Role Mapping

Given:

The XHQ role **`RoleAdmins`** is assigned to the domain group **`dom01\xhqadmins`** and all rights are assigned to it.

In this case, the XML export of this **`RoleAdmins`** role looks like the following:

```
<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:RoleMapEntryname="RoleAdmins"
group="dom01\xhqadmins"
local="false">
<XHQ:Right name="catalog_editing"/>
<XHQ:Right name="component_editing"/>
<XHQ:Right name="view_editing"/>
<XHQ:Right name="view_usage_access"/>
<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>
```

Configuring End User Access (for the Solution Admins role)

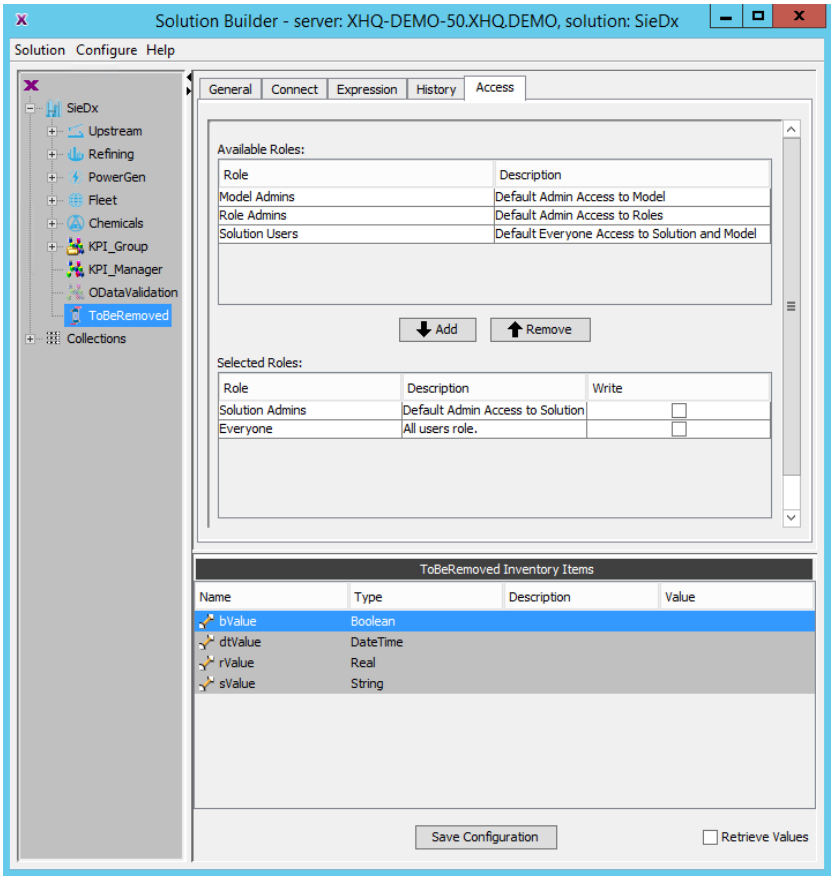
As a **Solution Admin**, you can assign access to the following solution elements:

- Individual components
- Primitive members
- Link collections
- Global collections
- Global collection members

Important Things to Note

- The Access tab is disabled for all other users that do not belong to the Solution Admins role.
- Security cannot be configured for the root of the solution.

You restrict access to these elements through the XHQ Solution Builder. Simply click on the element from the Navigation Panel and the Access tab appears in the Detail panel of the XHQ Solution Builder.



XHQ Solution Builder – Access List

The roles on the Access List reflect the roles configured through the Role Administration Utility. Here are some possible role examples and their expected access outcome.

Role Examples

Roles listed	Description of who has access
"Solution Admins" only	Only members of Solution Admins have access.
"Solution Admins", "PlantMgrs", "Execs"	Any members of Solution Admins, PlantMgrs or Execs have access. Users with only membership in Solution Users would not have access.

Example: Granting access to PlantMgrs role only

The PlantMgrs role was created using the Role Administration Utility.

1. Launch the **Solution Builder**.
2. From the navigation panel, select a solution element (for example, a component, a primitive member, a global collection, etc.).
3. Click on the **Access tab**.
4. Click **Remove All** to clear the Selected Roles list, leaving only the Solution Admins role.
5. From the list of Available Roles, select PlantMgrs and click Add.
   The PlantMgrs role is added to the list of Selected Roles.
6. Click **Save Configuration**.
7. OPTIONAL
   From the Solution menu, click Log Out.
   You will be asked if you want to log on as another user.
8. Click **No**.
   The XHQ Solution Builder closes.

Besides the Solution Admins, only members of the PlantMgrs role will be able to access this solution element.

Example: Denying access to all users

In this example, only Solution Admins have complete access.

1. Launch the **Solution Builder**.
2. From the navigation panel, select a solution element (for example, a component, a primitive member, a global collection, and so forth).
3. Click on the **Access tab**.
4. Click **Remove All** to clear the Selected Roles list, leaving only the Solution Admins role.

   You cannot deny access to Solution Admins.

5. Click **Save Configuration**.
6. OPTIONAL
   From the Solution menu, click Log Out.
   You will be asked if you want to log on as another user.

7. Click **No**.
   The XHQ Solution Builder closes.

Kerberos Authentication

Kerberos facilitates user authentication over an un-trusted network of disparate systems. XHQ Servers can use Kerberos for **Single, Sign-On (SSO) authentication**. This is also known as integrated log-in since XHQ takes advantage of an earlier Windows log-in to securely establish a user's identity without asking the user for a password again.

Kerberos authentication is too large a subject to cover in this guide. Additional information can be found on the Internet or in many of the books written on Kerberos.

About XHQ and Kerberos

This version of XHQ supports Kerberos V5 and leverages the Java GSS API (and JAAS) for this support. By default, XHQ installs and works in a non-Kerberos mode.

All connections to XHQ requires either NTLM (NT LAN Manager) or Kerberos, depending on the setting of the XHQ Server. There is, however, no fallback from Kerberos to NTLM.

Setting up the XHQ Server

The following steps must be carried out to explicitly prepare and configure each XHQ Server for Kerberos use.

For This Task	
Step 1	Addressing Pre-requisites
Step 2	Choosing a Service Principal Name (SPN)
Step 3	Creating a Unique User Account
Step 4	Mapping the SPN
Step 5	Configuring the XHQ Service to the SPN user
Step 6	Verifying Kerberos Use

A list of **Kerberos glossary terms** can be found in the *Appendix*.

Step 1 - Addressing Pre-requisites

To address possible issues with the UDP (User Datagram Protocol) for tickets requests and with the Kerberos ticket size, changes must be made to Windows Registry.

TCP vs. UDP

When a server delegates to another server, it needs to request a ticket on behalf of the client. But when the client is a member of *approximately* 120 groups, the ticket becomes too large to send reliably with UDP. To force TCP (Transmission Control Protocol) for tickets requests, a registry change is needed on the server.

A limitation on the UDP packet size may cause an error message at domain logon similar to this:

```
Event Log Error 5719
Source NETLOGON

No Windows Domain Controller is available for domain Domain. The following error
occurred:

There are currently no logon servers available to service the logon request.
```

Additionally, the Netdiag tool may display the following error messages:

```
Error message 1
DC list test . . . . . . . . . . . : Failed [WARNING] Cannot call DsBind to
COMPUTERNAMEDC.domain.com (159.140.176.32). [ERROR_DOMAIN_CONTROLLER_NOT_FOUND]
Error message 2
Kerberos test. . . . . . . . . . . : Failed [FATAL] Kerberos does not have a ticket for
MEMBERSERVER$.]
```

To force the use of TCP packets

1. From the server, go to **Start**, click **Run**, and enter **`regedit`**.
   The Registry Editor appears.
2. Go to **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters**.
3. Check if the following parameter has been applied; if not, add/change accordingly.

Parameter	Type	Value
MaxPacketSize	REG_DWORD	1 (decimal)

   The MaxPacketSize is the maximum size for a UDP packet. Setting this to "1" forces the use of TCP packets.

4. **Restart** the server.

Web Servers

When users are members of many groups, the Kerberos ticket size becomes large. Since the Kerberos ticket is included in the web requests, the IIS server may reject the request because of the request being too large. To avoid this, configure the following registry setting on the web server.

To force the use of TCP packets

1. From the web server, go to **Start**, click **Run**, and enter **`regedit`**.
   The Registry Editor appears.
2. Go to **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\HTTP\Parameters**.
3. Check if the following parameter has been applied; if not, add/change accordingly.

Parameter	Type	Value
MaxFieldLength	REG_DWORD	65534 (decimal)
MaxRequestBytes	REG_DWORD	65534 (decimal)
MaxClientRequestBuffer	REG_DWORD	32767 (decimal)
UrlSegmentMaxLength	REG_DWORD	4096 (0x00001000)

4. **Restart** the web server.

XHQ System Pre-requisites

- The XHQ Server that you wish to configure for Kerberos authentication must be in a **working** state, meaning it has already been installed and configured (without Kerberos), and runs successfully.
- You must use **Windows Active Directory** as your directory service.
- The Login Server is obtained through the system environment variable `LOGONSERVER`.
- Only **Windows-based clients** can use Kerberos authentication. The Windows client system needs to be in a domain (not a workgroup) and the user needs to be logged on as a **domain** account (logging on to a Windows client system as a local account will not work).
- All of the client systems should be in the same domain unless you follow additional Windows configuration steps for cross-realm Kerberos authentication.

  These steps are outside the scope of this document. Consult the Windows Server product documentation for details.

- Windows-based XHQ systems must adhere to the documented XHQ system requirements for the server and clients. See the topic, *System Requirements*, located in the XHQ Installation Guide for details.

Step 2 - Choosing a Service Principal Name (SPN)

The SPN should be in the form:

`XHQ/HOSTNAME.example.com@EXAMPLE.COM`

Where:	**XHQ**	Is the XHQ Service Class. It is in **uppercase**.
	HOSTNAME	Is the server host name. The hose name must be an undistinguished (no dots) DNS name. It is in **uppercase**. The Internet standards (Request for Comments) for protocols mandate that component hostname labels may contain only the ASCII letters "a" through "z" (in a case-insensitive manner), the digits "0"' through "9", and the hyphen ("-"). No other symbols, punctuation characters, or white space are permitted. Specifically, host names must not contain underscores ("_").
	example.com	Is the server domain. It is in **lowercase**.
	EXAMPLE.COM	Is the realm. It is in **uppercase**. The Kerberos realm is based on the DNS Domain which is obtained through the environment variable `USERDNSDOMAIN` (if available), or by other means for accounts where this variable is not present.

The SPN is **case-sensitive**.

`HOSTNAME.example.com` is also commonly referenced to as being a FQDN (Fully Qualified Domain Name).

Step 3 - Creating a Unique Domain User Account

You must create a unique *Active Directory* (AD) domain user account in the same domain as the XHQ Server to This user account acts as the Kerberos security user for XHQ on a specific machine and is the "identity" behind the SPN. This account is mapped to the SPN (as documented in Step 4) and, more importantly, has a password that, through some hashing, serves as a symmetric encryption/decryption key for parts of the Kerberos tickets.

The domain account and password mentioned above are defined while setting up the XHQ Server.

Important Things to Note

- The AD domain user account should not be a regular user. That is, do not use your "personal" domain account.
- The "identity" user account must be **unique** and not be used for purposes other than acting as a Kerberos identity for a single XHQ Server. XHQ servers must not be sharing this same identity user.
- The "identity" user account must not be the same user account that might have been set up for other purposes, such as XHQ support remote access. This restriction is needed because the user account set up for XHQ support purposes will likely have its Active Directory `UserPrincipalName` attribute set up in a way that is incompatible with the XHQ Kerberos user account.
- The name of this user account should be in the form: **`XHQ-`*`HOSTNAME`***, where *`HOSTNAME`* is the name of the machine that hosts the XHQ services.

  In general, the hostname is similar to the short SPN name chosen above with a dash ("-") in place of the slash ("/").

- When setting the XHQ-*HOSTNAME* user's password, set it to never expire.
- An underscore in the user account name is not recommended due to known issues that pertain to third-party vendors/products.
- This "identity" user account name is **case-sensitive**.
- Local user accounts may also authenticate when Kerberos is enabled; however, in this case, basic authentication is used.

To create an AD domain user account

1. Open the **Active Directory Users and Computers** tool on a Domain Controller in the XHQ Server Windows domain.
2. **Create a Windows domain user** (not computer) account that will be used for the SPN.
   This account must be in the same domain as that of the XHQ system.
3. Enter the unique "identity" user account name.
4. Set the **Password to never expire**.
5. **Enter a password** for that account and confirm it.
   Remember this password as it will be needed later.

The following steps may be needed to conform to corporate IT site policies (with regards to how these types of accounts need to be configured).

6. From the **Active Directory Users and Computers** tool, select and right-click the account you just created, and then select **Properties**.
7. Click the **Account tab** and set any needed options.

Step 4 - Mapping the SPN

After the Kerberos "identity" user account is created, it must be mapped to the proper SPNs. This is done by using the **Microsoft `setspn` utility**, which is available on Windows Server.

Example: Mapping the user account to the SPN using the setspn utility

In this example, the SPN is mapped (using the `setspn` utility) for the machine that will hosts XHQ. The SPN consists of the following information:

- The host name is **NODE100**;
- The FQDN is **NODE100.corp.xyz.com**;
- The Kerberos realm is **CORP.XYZ.COM**;
- The domain account user name is **XHQ-NODE100**.

See *Step 2 Choosing a Service Principal Name SPN* for details on the SPN format.

From a command-line, run the **`setspn` utility**, and enter the following commands:

```
setspn -a XHQ/NODE100.corp.xyz.com@CORP.XYZ.COM XHQ-NODE100
ktpass /princ XHQ/NODE100.corp.xyz.com@CORP.XYZ.COM /pass <password> /ptype KRB5_NT_
PRINCIPAL /mapuser XHQ-NODE100@CORP.XYZ.COM
```

Where `<password>` is the actual password used by the `xhq_service` account.

If you enter these lines on an XHQ System but do not have Domain Administration rights then the `setspn` command fails, displaying an error message similar to the following:

```
C:\>setspn -a XHQ/jfg jaastest

Registering ServicePrincipalNames for CN=JAAS Test,OU=Users IndX
Groups and Special Accounts,DC=corp,DC=indxhq,DC=com

XHQ/jfg

Failed to assign SPN on account 'CN=JAAS Test,OU=Users IndX Groups
and Special Accounts,DC=corp,DC=indxhq,DC=com', error 0x2098/8344
-> Insufficient access rights to perform the operation.
```

Step 5 - Configuring the XHQ Service

For current keytab users:

Note, the keytab file is not needed for Microsoft integrated Kerberos authentication. Therefore, use the following procedure to configure the XHQ Service to the SPN user (this enables the updated enhanced Kerberos support).

To configure the XHQ Service to the SPN user

1. Go to the location defined by `%XHQ_SERVER_HOME%`, locate the **keytab file**, and **rename** it.

   This ensures that the keytab file is not used.
2. **Shutdown** the **XHQ Service**.
3. Change the **logon credentials** of the XHQ Service to the SPN user (`DOMAIN\XHQ-NODENAME`).
4. Add that SPN user to the **local Administrators group**.
5. **Start** the **XHQ Service**.

Step 6 - Verifying Kerberos Use

The last step in this process is to **start XHQ** and **verify** that the Model server is using Kerberos authentication (and not NTLM).

If problems persist, continue to the topic, *Troubleshooting Kerberos*.

Troubleshooting Kerberos

Troubleshooting problems with Kerberos can be complex, given the number of computers and amount of software that can be involved. As such, this topic contains possible tools to troubleshoot the problems you may be experiencing.

Checking XHQ

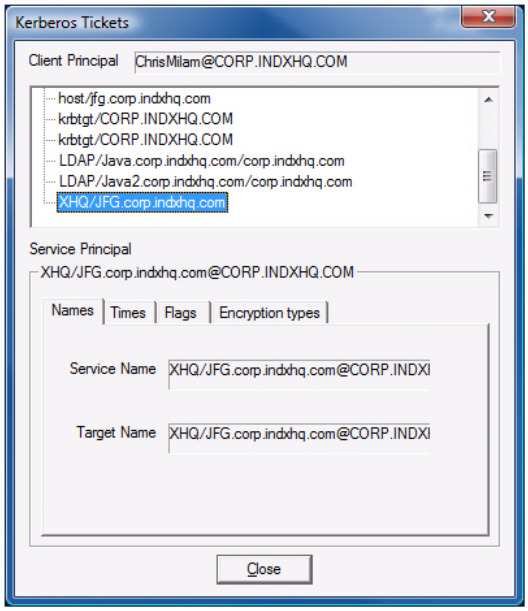
The first thing to try if encountering problems is to make sure that XHQ works without Kerberos. If problems persist, then troubleshoot XHQ before attempting to debug problems with Kerberos.

If it has been established that XHQ works with username / password, then test XHQ using its "integrated security" setting. Make sure that the workstation running XHQ has been logged on under Windows as a domain user.

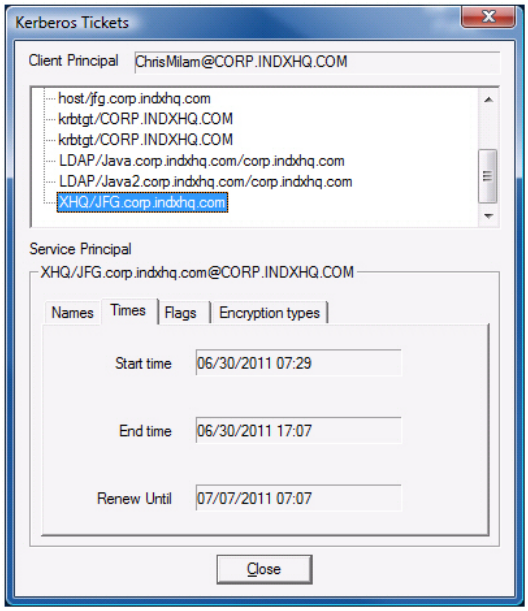
Using the kerbtray utility

The **`kerbtray` utility** displays ticket information for a given server running the Kerberos protocol.

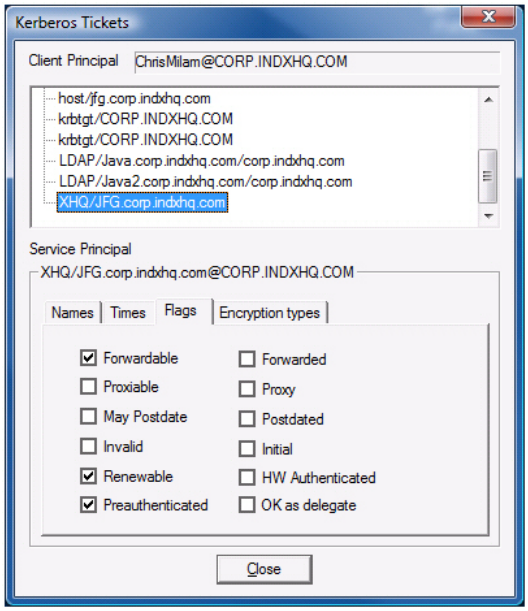
The klist utility (tickets command) is a command-line utility that also provides Kerberos ticket information, but is not as detailed as the kerbtray utility.



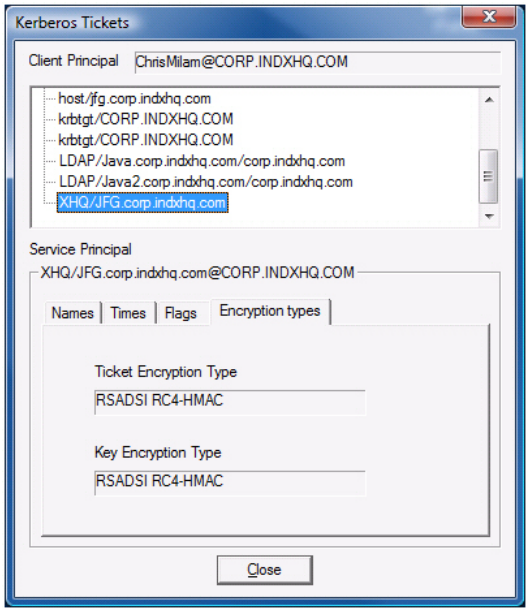
The **Names tab** displays the Service and Target names for the ticket selected. The Service name is the account principle for the service requested. The Target name is the service for which the ticket was requested.



The **Times tab** shows the period of time for which the selected ticket is valid.



The **Flags tab** enables you to determine the ticket's status and intended use.



The **Encryption Types tab** provides ticket encryption type (that is, the algorithm used to encrypt the Kerberos ticket) and the key encryption type (which is the encryption type used with the session key).

Debug Settings

The following XHQ log-related settings should be temporarily used if Kerberos-related issues need to be debugged with the assistance of the XHQ Customer Support Team.

To set the debugging environment

1. From the **`modelsettings.properties`** file (which, by default, is stored at the location specified by the environment variable `%XHQ_SERVER_REPOS%`), set the following properties:

```
net.indx.util.syslog.sysloglevel=2
net.indx.util.syslog.maxlogsizemb=50
net.indx.util.syslog.displaythreadname=true
```

2. **Reboot** XHQ.

Again, these settings should be **temporarily** used for the debug period only.

To enable Kerberos event logging on a specific computer

1. Start the **Registry Editor**.
2. **Add** the following registry value:

`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters`

Where:	**Registry Value=**	LogLevel
	Value Type=	REG_DWORD
	Value Data=	0x1

3. If the **`Parameters`** subkey does not exist, create it.
4. **Exit** the Registry Editor.

Kerberos-related events are found in the system log.

Remove this registry value when it is no longer needed so that performance is not degraded on the computer. Also, you can remove this registry value to disable Kerberos event logging on a specific computer.

Testing the XHQ connection

To test a basic connection to XHQ using .NET API, use the **`Siemens.Xhq.TestConnect.exe`** test program.

Consider the following example.

```
c:> Siemens.Xhq.TestConnect /host jfg

IKVM INIT: cli.System.NotImplementedException: The method or operation is not
implemented.
Debug is  true storeKey false useTicketCache false useKeyTab false doNotPrompt true
ticketCache is null isInitiator true KeyTab is null refreshKrb5Config is true principal
is null tryFirstPass is false useFirstPass is false storePass is false clearPass is false
[XHQNTLoginModule] user name = xhquser
[XHQNTLoginModule] user SID = S-1-5-31-1945604384-1365774696-1373009395-4990
[XHQNTLoginModule] user domain = INDX
AUTH: Attempt connect to XHQ/JFG.corp.indxhq.com@CORP.INDXHQ.COM
AUTH: Got client credentials
AUTH: Got client context
AUTH: Pass #1 - top of loop to connect to XHQ/JFG.corp.indxhq.com@CORP.INDXHQ.COM
AUTH: Will send token of size 1470 from initSecContext.
AUTH: uid = 3
AUTH: token len = 103
AUTH: read input token of size 103 for processing by initSecContext
AUTH: Pass #2 - top of loop to connect to XHQ/JFG.corp.indxhq.com@CORP.INDXHQ.COM
AUTH: ClientAuth returned: Request succeeded.

Success!
Connected to XHQ Solution Server on jfg as INDX\xhquser
```

Application Server Log Messages

The Application Server log records authentication errors, as seen in the following example.

```
06/17/2011 15:49:35.518: Error authenticating user: NT AUTHORITY\SYSTEM
06/17/2011 15:49:35.518: Cause of error: Unable to authenticate XHQ user.
06/17/2011 15:49:36.597: Error authenticating user: NT AUTHORITY\SYSTEM
06/17/2011 15:49:36.597: Cause of error: Unable to authenticate XHQ user.
```

The **`Error authenticating user`** message occurs because the Keep Alive service is, by default, configured with the local system account. In a Kerberos environment, it must be configured with a valid domain user account. Configuring the Keep Alive service with a valid domain user account fixes these errors.

XHQ Web Service Impersonation and Delegation

This topic summarizes end-user impersonation and delegation with respects to the XHQ Web Service.

> For details on impersonation and delegation with Windows Communication Foundation (WCF), refer to the following site:
> `http://msdn.microsoft.com/en-us/library/ms730088.aspx`

End-user impersonation

In the case of the **ASP.NET web service** (.ASMX), there is no change (either declarative or programmatic) for the web service client. However, in the case where there are multiple "hops" or nodes between the end-user (such as the Internet Explorer invoker) and the service, each "hop" (the intermediate web application or service) needs to allow **impersonation** to ensure the original caller credential propagation. This is done declaratively in the **`web.config` file** by adding the following line:

```
<identity impersonate="true"/>
```

In the case of the **WCF service**, the impersonation is more complex on the server-side. In this case, the client that invokes the web service needs to allow the impersonation level. This can be done one of two ways:

- **Programmatically in the client code**, using the service proxy.

  Consider the following example.

```
//Invoke the XHQ Web Service
XHQWebServiceClient client = new XHQWebServiceClient();

//Change the AllowedImpersonationLevel
client.ClientCredentials.Windows.AllowedImpersonationLevel =
System.Security.Principal.TokenImpersonationLevel.Impersonation;

//invoke web service operation
Boolean status = client.IsSolutionAccessible(host);
client.Close();
```

- **Declaratively in the client `config` file** (for example, `web.config`, `app.config`, or, in the case of the `WcfTestClient` application, `Client.dll.config`).

  Consider the following example.

```
<system.serviceModel>
……
<client>
  <endpoint address="http://localhost/indx/xhqwse/XhqWse.svc"
binding="basicHttpBinding" bindingConfiguration="BasicHttpBinding_XHQ Web Service"
contract="XhqWseReference.XHQWebService" name="BasicHttpBinding_XHQ Web Service"
behaviorConfiguration = "ImpersonationBehavior" />
</client>
<behaviors>
  <endpointBehaviors>
<behavior name="ImpersonationBehavior">
  <clientCredentials>
    <windows allowedImpersonationLevel = "Impersonation" />
  </clientCredentials>
</behavior>
  </endpointBehaviors>
</behaviors>
</system.serviceModel>
```

Both approaches are implemented in the **`TestXhqWse` web application** that is provided as a convenience for testing the WCF service.

For more information on the `TestXhqWse` web application, go to the topic, *Testing the Web Service* located in the **XHQ SDK Reference Guide**.

If the **`WcfTestClient.exe`** application (which is provided by Microsoft) is preferred for testing the WCF service (which requires impersonation in order for WCF operations to work), modify the `Client.dll.config` file by appending the XML lines shown previously in the Declarative example.

End-user delegation

If delegation is required (that is, if the client and WCF service are on separate machines and/or multiple hops are needed, AND if XHQ uses Kerberos authentication), the previous descriptions for the programmatic and declarative methods of invoking the web service (for impersonation) are still valid but the following minor changes are needed for delegation

Associated accounts and client machines must both be enabled for delegation.

Consider the following examples.

- For the **programmatic** method:

```
//Invoke the XHQ Web Service
XHQWebServiceClient client = new XHQWebServiceClient();

//Change the AllowedImpersonationLevel
client.ClientCredentials.Windows.AllowedImpersonationLevel =
System.Security.Principal.TokenImpersonationLevel.Delegation;

//invoke web service operation
Boolean status = client.IsSolutionAccessible(host);
client.Close();
```

- For the **declarative** method:

```
<system.serviceModel>
……
<client>
  <endpoint address="http://localhost/indx/xhqwse/XhqWse.svc"
binding="basicHttpBinding" bindingConfiguration="BasicHttpBinding_XHQ Web Service"
contract="XhqWseReference.XHQWebService" name="BasicHttpBinding_XHQ Web Service"
behaviorConfiguration = "ImpersonationBehavior" />
</client>
<behaviors>
  <endpointBehaviors>
<behavior name="ImpersonationBehavior">
  <clientCredentials>
     <windows allowedImpersonationLevel = "Delegation" />
  </clientCredentials>
</behavior>
  </endpointBehaviors>
</behaviors>
</system.serviceModel>
```

Important Things to Note about Delegation

- All servers must be enabled for delegation.
  See the topic, *EXAMPLE Configuring the server to be trusted for delegation*.
- The Web Server requires delegation enabled if Kerberos is used.
- Domain accounts must be enabled for delegation.
- Domain accounts, under which you run the web services, must have an SPN registered in Active Directory.
- The client request call must permit delegation.

For additional information, see these articles:
http://msdn.microsoft.com/en-us/library/ms998355.aspx
http://msdn.microsoft.com/en-us/library/ff647404.aspx#paght000023_delegation

EXAMPLE: DELEGATION FROM A WCF CLIENT

Given:

`AllowedImpersonationLevel` is applied to Impersonate for the client endpoint behavior.

Impersonation scenario:

```
<behaviors>
<endpointBehaviors>
  <behavior name="NewBehavior">
    <clientCredentials>
<windows allowedImpersonationLevel=" Impersonation" />
    </clientCredentials>
  </behavior>
</endpointBehaviors>
</behaviors>
```

The impersonated client credential needs to access resources across machines. So, `AllowedImpersonationLevel` is set as Delegate as shown in the example below.

Delegation scenario:

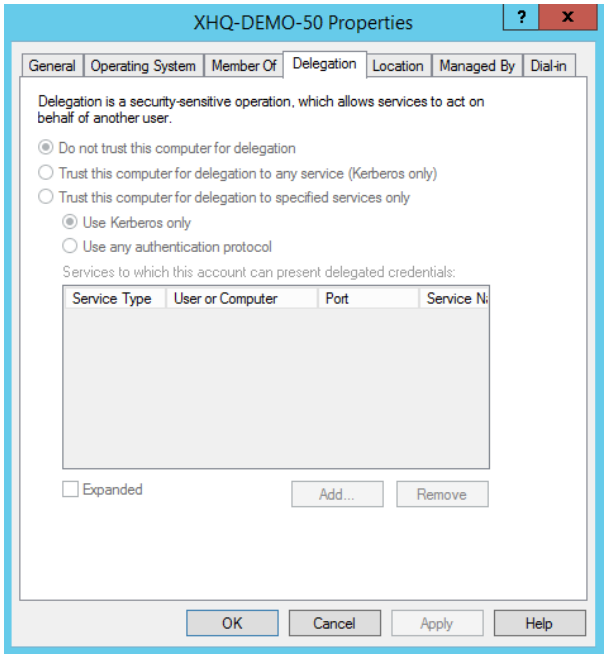
```
<behaviors>
<endpointBehaviors>
  <behavior name="NewBehavior">
    <clientCredentials>
<windows allowedImpersonationLevel=" Delegation" />
    </clientCredentials>
  </behavior>
</endpointBehaviors>
</behaviors>
```

For the delegation scenario, enable the WCF host or the identity under which WCF service is running delegation.

EXAMPLE: CONFIGURING THE SERVER TO BE TRUSTED FOR DELEGATION

All servers must be enabled for delegation.

1. From the **Start** menu, open **Administrative Tools**, and click **Active Directory Users and Computers**.
2. In the left pane, expand the domain and click **Computers**.
   A list computers in this domain appears in the right pane.
3. In the right pane, go to the server, right-click and then click **Properties**.
4. Click the **Delegation tab**.
5. If the WCF server is running under **a built-in machine account** (for example, Network Service) or **a domain account**, click **Trust this computer for delegation to any service (Kerberos only)**.



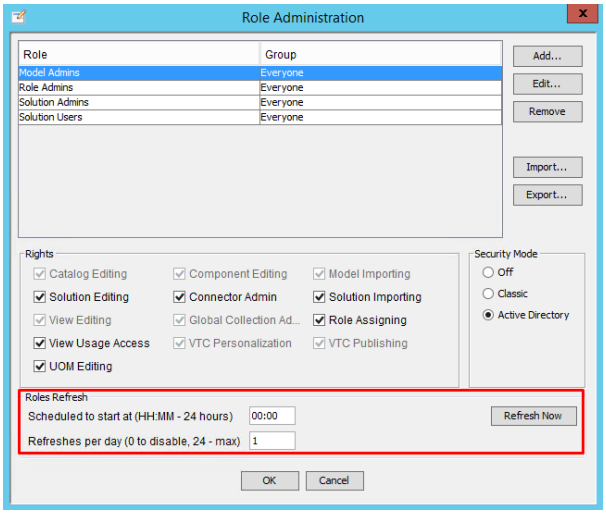
6. Click **OK**.
7. Exit **Active Directory Users and Computers**.

Changing Group Membership

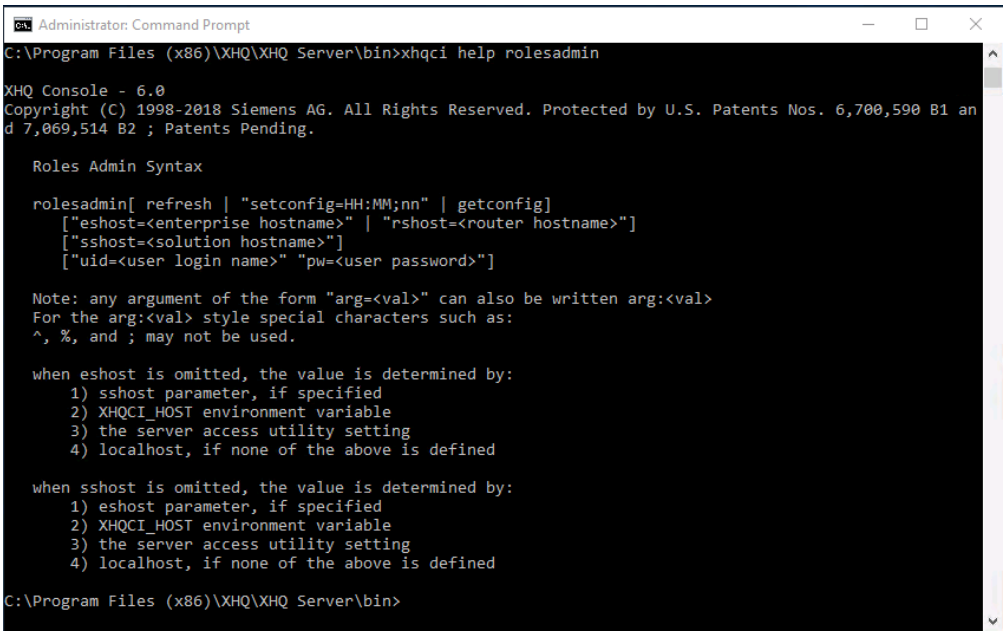
If a user's group membership changes in the Active Directory and the user has already been authenticated by XHQ in a previous session, two things must occur in order for the group membership change to take effect:

- The Role-to-Group mappings must be refreshed;
- The Active Directory needs to update and propagate the changes.

You can manually force a refresh of XHQ roles or set a refresh schedule through the **Role Administration** utility of the XHQ Workbench, or by setting the **`xhqci` command**.



Role Administration Utility - Roles Refresh



xhqci rolesadmin Command-line

Important Things to Note

- To execute the `xhqci rolesadmin` command, you must have **Role Admins permissions**.
- The `xhqci rolesadmin refresh` command is also available through a **Web Service**.
- The number of refreshes should be kept to a minimum to avoid unnecessary cache purging and refreshing of this information. Considering that large Active Directory configurations can take a long time to propagate changes or that other changes are often communicated to users as "next day", refreshing a few times a day should be adequate.

For details on the Role Administration Utility, see the topic, *Accessing the Role Administration Utility*.

For details on the xhqci command, see the topic, *Refreshing Roles*

For details on the XHQ Web Services, see the topic,*Web Services*, located in the **XHQ SDK Reference Guide**.

Cross-forest Authentication

Cross-forest authentication allows you to configure the Model Server to contact additional domains and/or a specific LDAP (Lightweight Directory Access Protocol) server when authenticating users.

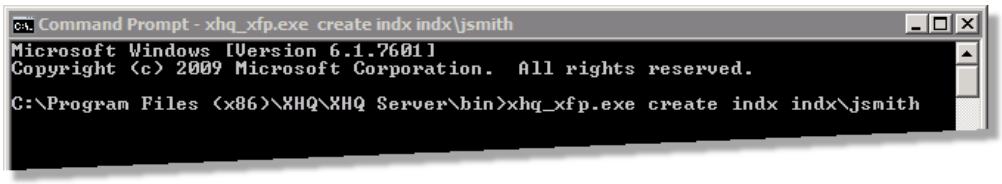
To configure, run the cross-forest configuration utility, **`xhq_xfp.exe`**, **on the XHQ Server**. This utility manages the entries (create, modify, delete, list) in the **`xfp.dat`** file.

Commands:

```
create <domain or server name> <username>
update <domain or server name> <username>
delete <domain or server name>
list
```

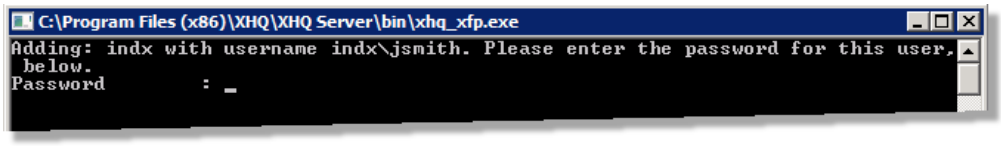
Example:

```
create indx indx\jsmith
```



Important Things to Note

- The **`xhq_xfp.exe`** file is located in the **bin** directory of **`%XHQ_SERVER_HOME%`**.
- The specified `<username>` must be in either of the following formats:
  - **The User Principal Name Format**
    username@example.domain.com
  - **The Down-level Logon Name Format**
    DOMAIN\username
- The **`create`** and **`update`** commands prompt for a **password**.



- The user specified as the XFP authentication user should be treated like a **Windows Service account**. Therefore,
  - The **password should never expire**.
  - The account should not be used for any other purpose.
  - The account must have read access to LDAP/Active Directory.

To query foreign LDAP servers, appropriate credentials must be provided to authenticate with the given LDAP server. These credentials, as well as the name of the domain/LDAP server, are stored in the **`xfp.dat`** file, which is encrypted using Windows Data Protection API.

Windows Data Protection is recommended by Microsoft as best practice for handling encrypted data for service uses such as this.

View Security

In this section, you will learn how to control access to the component view for any instance of the component in the solution.

The XHQ View Security only works for integrated and user/password authentication.

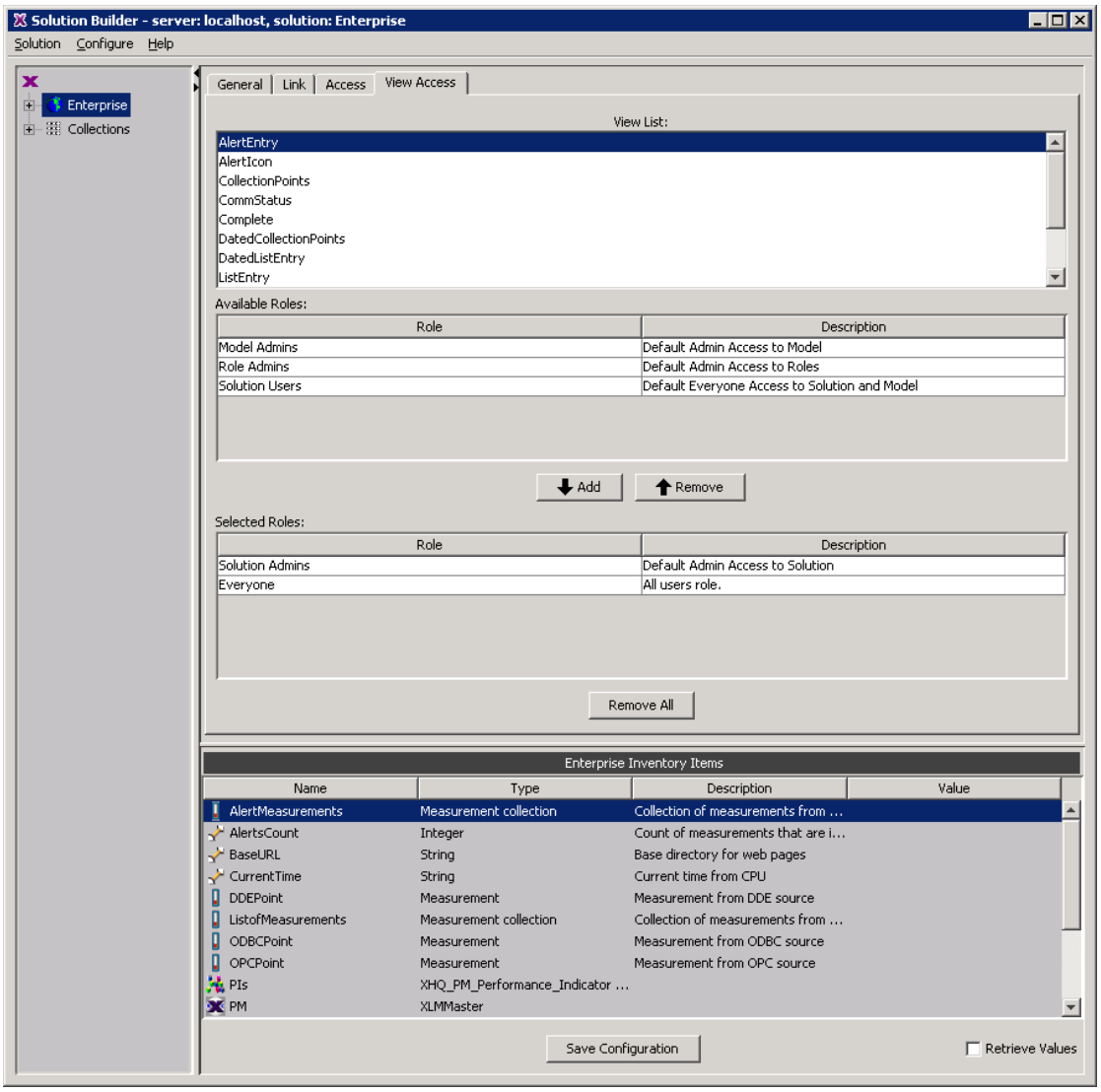
See the topic, *Requirements*, for more information.

Assigning View Roles

You can assign roles to a particular view using the **View Access tab** of the component configuration in the Solution Builder. The View Access tab appears when a **component member** is select from the Inventory Items panel of the Solution Builder.

The View Access tab is not available for primitive members.

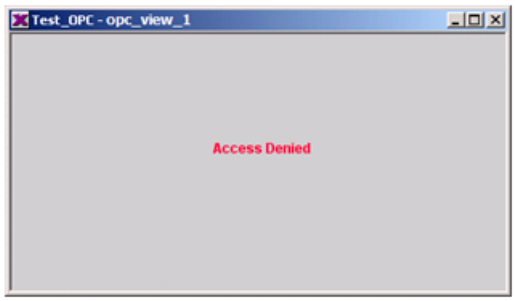
In the View Access tab, a list of component views appears. Select a view to which you want to assign roles.



Solution Builder, View Access Tab

Important Things to Note

- View role configurations are stored in the `\repos\Solution\%solution_name%\DataConfigs.dat` file.
- If an instance does not have view roles assignment, then it is available to all roles.
- If a view is deleted, but view roles exist for the view, then it is ignored (without any error message) and is removed from the file on the next save.
- If a user is not given access to the given view, then the view is:
  - Not shown in the list of views in the Solution Builder.
  - Not shown in the list of views in the NavBar.
  - Not shown in the list of views in the VTC.
- If a user navigates to a view they have not been given access to, then an "Access Denied" message appears.



- If a user opens a view that contains an embedded view to which the user does not have access to, then the embedded view appears blank.
- Exposed collections cannot have view security.

Requirements

Only **Solution Administrator roles** are able to use the View Access tab.

The Solution Administrators group has access to all views.

As mentioned earlier, View Security only works when authentication and security are enabled. In addition, the `EnableViewSecurity` property must be set to true in the globalsettings.properties file.

Exporting/Importing

The View Access is automatically added to the export XML document if security is enabled. Upon import, it is again part of the XML document.

TIP: The following is an example of the security entry in the exported XML document. It may be omitted in the import to provide compatibility with legacy documents.

```
<XHQ:ViewSecurity
    xhqComponent="Test_OPC"
    xhqView="OPC_View_2"
    roles="Model Admins; Users of Remote Client "
/>
```

Important things to note about Import View Security

- The default settings for view security allows **Everyone** and **Solution Admin** roles.
- If `rolesAllowed="%none"`, then the view security reverts to using default security settings.
- If `XHQ:ViewSecurity` does not exist in the XML, then view security remains unchanged.

View Flavors

View security is applied at the view level, regardless of flavor. The view security settings are the same for all flavors of a view irrespective of the device and orientation type.

For more information on view flavors, go to the topic, *The Views Tab*, located in the XHQ Developer's Guide

Important Things to Note

- For an existing security setting, prior to the view flavor implementation, that already has multiple flavors of a configured view, security is based on the Desktop.
- If a desktop view flavor doesn't exist, then the security configuration for the flavors is ignored and the user must reconfigure it.

To set view security for views

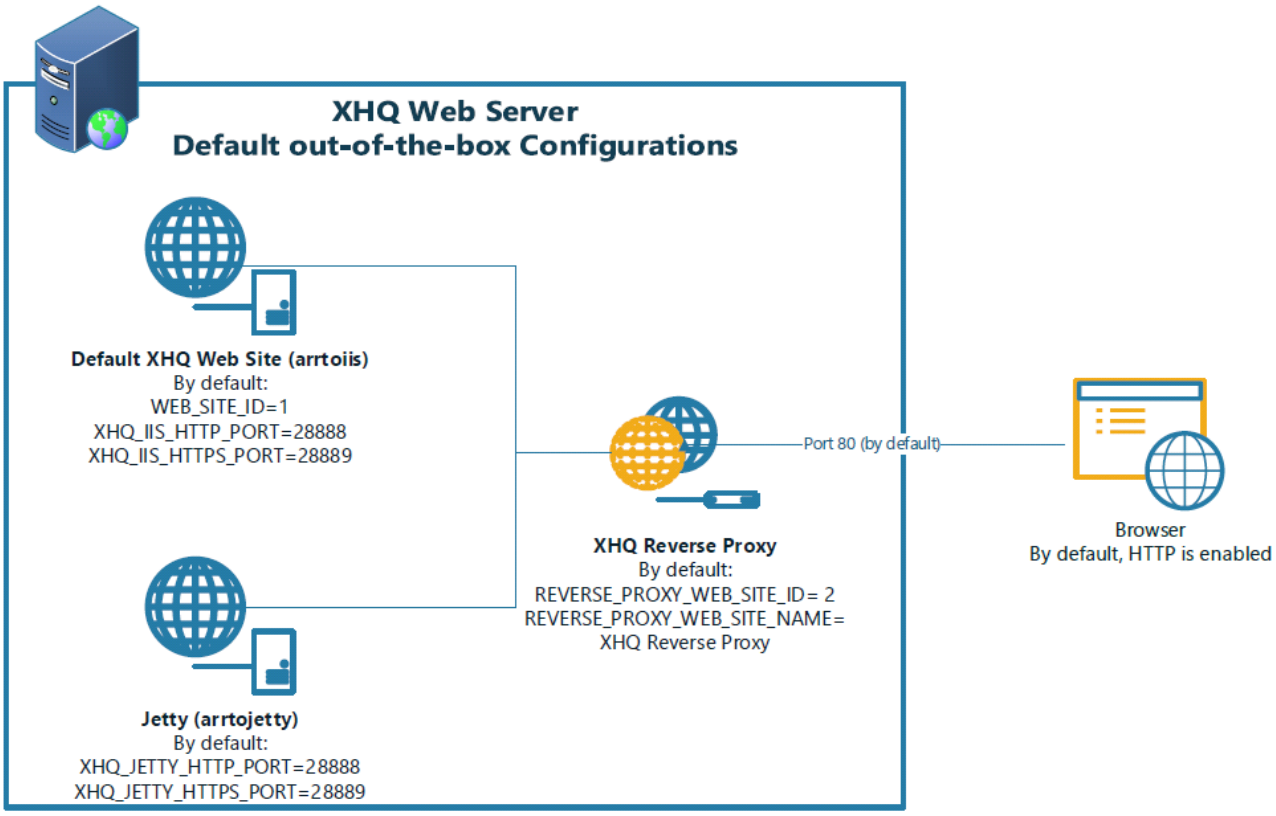
1. Enable **Integrated Security (Active Directory)** and join a domain.
2. From the `globalsettings.properties` file, set the **`EnableViewSecurity`** property to **true**.

For more information, go to the topic, *Using globalsettings properties*.

XHQ Web Server Architecture and Security

The updated IIS architecture on the XHQ Server makes the XHQ HTML5 Solution Viewer robust and effective, especially with more than 100 users viewing simultaneously. This update introduces the IIS Reverse Proxy, which is referred to as **XHQ Reverse Proxy**, and **Jetty** .

Consider the following diagram:



Important Things to Note

- The front port for XHQ continues to be port 80 for HTTP and 443 for HTTPS request.
- The **XHQ Reverse Proxy** handles all the incoming requests and routes all the requests to either the IIS **Default XHQ Web Site** or to the **Jetty**.
- The Jetty serves the XHQ HTML5 Solution Viewer dynamic content while all other requests are handled by the Default XHQ Web Site.

The XHQ Reverse Proxy must remain with **anonymous authentication**. All authentications are passed to the IIS Default XHQ Web Site and Jetty accordingly.

The installation and configuration of IIS and Jetty web servers are handled by the XHQ installer. On a **fresh XHQ install**, all the port bindings and web site IDs can be changed through the `install.properties` file. On an **upgrade** from a release prior to XHQ 6.0.0.1 only the ports and site IDs for XHQ Reverse Proxy and Jetty can be changed.

When upgrading from an older release of XHQ, you will need to *reconfigure IIS* to be able to use the XHQ HTML5 Solution Viewer.

The virtual directory, `/indx/resources` (which points to `%XHQ_WEB_DATA%\conf\resources\public`), must have **Anonymous Access**, which is the default installation.

Configuring SSL in IIS for HTTPS

By default, XHQ is installed without HTTPS (HTTP over SSL) enabled. It is recommended, as a best practice, to enable HTTPS. The use of HTTPS protects against eavesdropping and man-in-the-middle attacks.

To setup HTTPS, a certificate is required. The certificate technically can be a "Self-Signed" certificate, however, it is recommended to have a certificate from a certificate signing authority (CA), which can be a home grown trusted CA or a commercial CA. The certificate typically has a trusted root certificate which is a public certificate used in the client machines to trust the certificates signed by the CA.

In summary, you need to have:

1. Server Certificate for IIS on the XHQ Server.
2. Root Certificate deployed on server and client machines.

The root certificate is required on the client machine running the XHQ Solution Viewer. Not having the root certificate deployed results in warning messages in the browser.

Typically, the root certificate is deployed as a trusted certificate in client machines for the browser usage. To deploy the root certificate on the client machine, refer to the following link (Manage Trusted Root Certificates):

https://technet.microsoft.com/en-us/library/cc754841(v=ws.11).aspx

The steps to configure XHQ to use HTTPS are as follows:

1. Enable SSL in IIS.
2. Deploy the root certificate.
3. Enable the XhqWse web service to work with SSL (HTTPS).
4. Update WAKA links to use HTTPS.
5. Update client applications to use TLS 1.2 or 1.1.

Enabling SSL in IIS

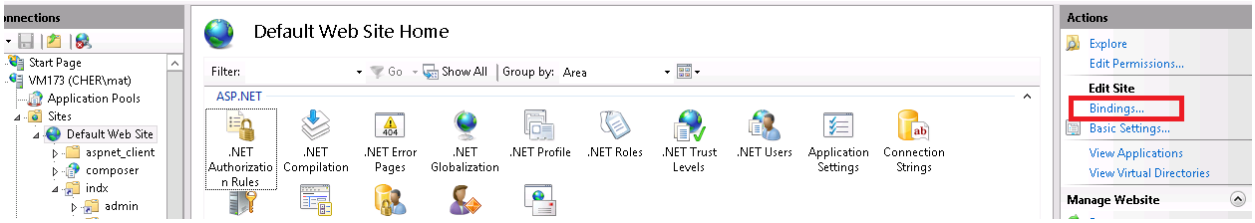
To use the secure **https** protocol, you must enable SSL in IIS.

The following procedure assumes that your site already has a certificate assigned to it. This certificate is used to encrypt and decrypt the information that is transferred over the network.

To enable SSL in IIS

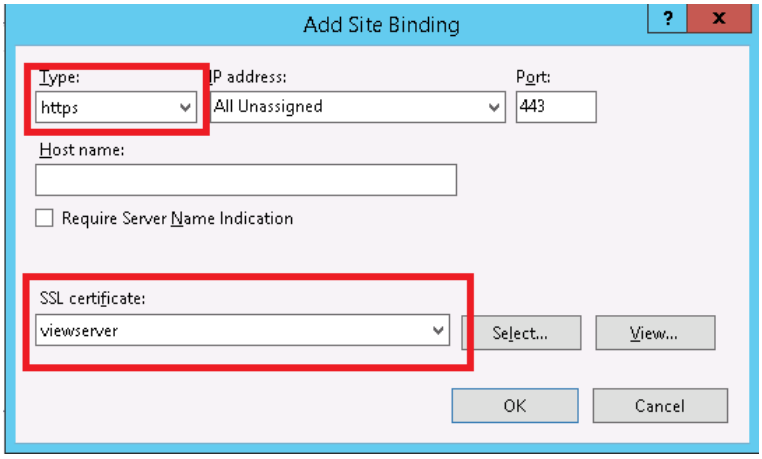
1. Log on to the Web Server machine as an **Administrator**.
2. From the **Start** menu, search for **IIS Manager**.
3. In the **Connections pane** on the left, click on the name of your server.
4. Click on **Server Certificates**.

5. In the **Actions pane**, click **Import**.
6. Select the **server certificate**.
   This is the certificate signed by the CA.
7. Click **OK**.
8. In the **Connections pane**, expand **your computer name**, expand **Sites**, and then click **Default**.
9. In the **Actions pane**, under **Edit Site**, click **Bindings**.



The "Site Bindings" dialog appears.

10. Click **Add**.
    The "Add Site Binding" dialog appears.



11. For **Type**, select **https**.
12. For **SSL Certificate**, select your certificate.
13. Click **OK**.
    This returns you to the "Site Bindings" dialog.
14. Click **Close**.
    This returns you to the IIS Manager.
15. In the **Connections pane**, expand **Sites**, and select **Default Web Site**.
16. In the IIS group, double-click **SSL Settings**.
17. Check **Require SSL**.
18. In the **Action pane**, click **Apply**.
    IIS is now running with HTTPS.

In case the root certificate is not in the "Trusted Root Certification Authorities" for both the clients and the server machines, you must deploy the root certificate in the trusted certification authorities and run `xhq_installcert`.

For details on xhq_installcert, refer to *Deploy the Root Certificate*.

For the **XHQWSE**, follow the instruction in *Configuring SSL for XhqWse Web Service*.

Deploying the Root Certificate

The following procedure assumes that you already have the root certificate for the Certificate Signing Authority that signed the certificate used to configure the IIS.

When you set-up HTTPS on a server, the Trusted Root CA certificate must be available on that server. The Trusted Root CA certificate must also be available on the JRE of every devClient that connects to that server. This can be achieved by using xhq_installcert.

You need to deploy the root certificate to server and client machines if the certificate is signed by a certificate signing authority where their root certificate is not in the "Trusted Root Certification Authorities" (for example, if you have your local signing certificate signing authority).

The root certificate is needed if HTTPS/SSL is enabled with "Required SSL" on IIS, which is the recommended setting.

Using the XHQ Install Certificate Utility

The XHQ Install Certification Utility (**xhq_installcert**) is used to deploy the root certificate on the XHQ Server and client machines. The utility is available under `%XHQ_DEV_HOME%\bin`.

This tool uses the Java keytool command interface to deploy the root certificate for the Certificate Authority signing the certificate used in the IIS to enable SSL/HTTPS.

Syntax:

```
xhq_installcert [/a.lias <alias>] [/p.assword <password>] [/f.ile <cert file>]
xhq_installcert /help
xhq_installcert /?
```

Example:

```
xhq_installcert /file c:\rootCA.cer
```

Parameter	Description
/alias	The name under which the certificate is to be stored in the JRE keystore. If this option is not specified, or is specified but no value is specified with the /alias switch, then "XHQCA" will be used.
/password	The password to the JRE keystore. If this option is not specified, or is specified but no value is specified with the /password switch, then the Java default will be used.
/file	The file containing the certificate (or certificate chain) for the Certification Authority that signed the certificate configured for HTTPS use. If no file is specified, but an alias or password are, then any previously installed certificate stored with the specified alias and password will be displayed.
/help *or* /?	Displays the help message.

To deploy the root certificate on XHQ server and client machines

1. Log on to the machine (XHQ server or client machines) where you want to deploy the root certificate.
2. Open the XHQ Shell.
3. Use **`xhq_installcert`** to deploy the certificate.

   Example:

   ```
   >xhq_installcert /file c:\rootCA.cer
   ```

Jetty and IIS

Jetty serves the XHQ HTML5 Solution Viewer dynamic content while all other requests are handled by the **Default XHQ Web Site**. All requests go through the **XHQ Reverse Proxy**. The XHQ Reverse proxy routes the requests to either Jetty or the Default XHQ Web Site.

The XHQ installer configures Jetty with the default settings. *If you opt in changing the defaults, update the install.properties prior to installing XHQ*.

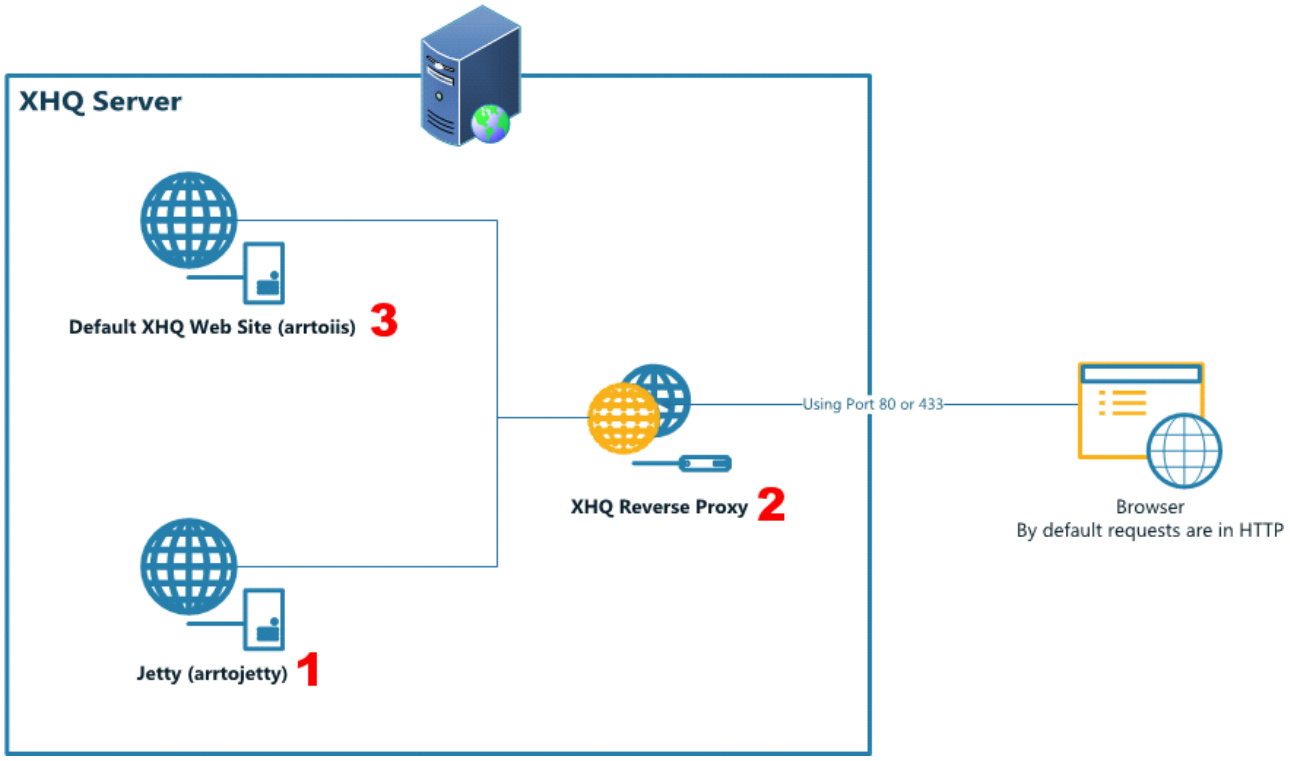
If you need to change the jetty settings after installing XHQ then jetty.properties need to be updated and based on the update you may need to update IIS accordingly.

The following procedure describes how to update the IIIS configurations.

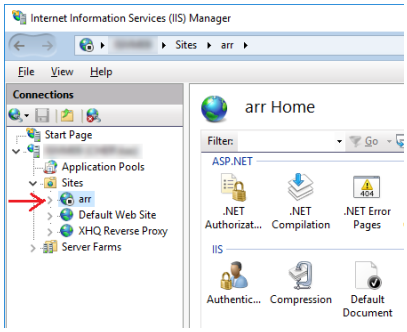
The XHQ Reverse Proxy must remain with **anonymous authentication**.

To reconfigure IIS

This procedure shows you how to configure IIS manually. The XHQ installer takes care of all the steps mentioned here and added as a reference.



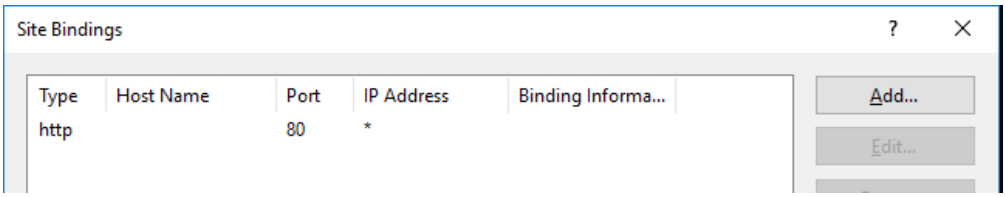
1. Open the **Internet Information Services (IIS) Manager**.
2. In the **Connections pane**, expand the **XHQ Server** > **Sites**.
3. Locate and **delete** the **arr website**.



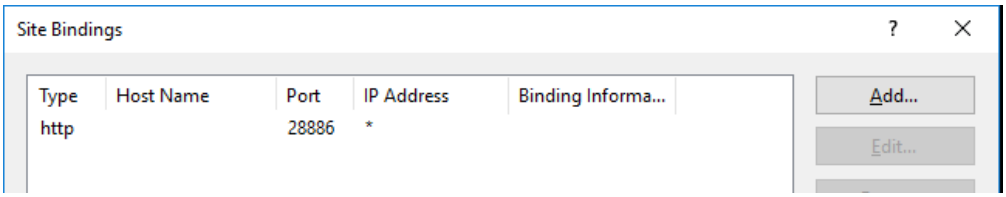
4. Under **Sites**, click **XHQ Reverse Proxy** (2).

   If stopped, the XHQ Reverse Proxy needs to be started.
5. In the **Actions pane**, under **Manage Website**, click **Start**.

   Next, you will set the HTTP port on the XHQ Reverse Proxy and the Default Web Site.
6. With **XHQ Reverse Proxy** still selected, in the **Actions pane**, under **Edit Site**, click **Bindings**.
   The "Site Bindings" dialog appears.
7. Edit/add **HTTP** and set the value to **80**.

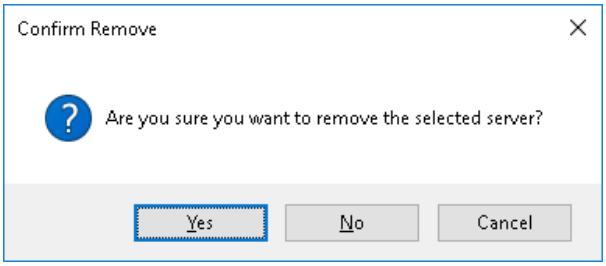


8. Click **Close**.
9. Under **Sites**, click **Default Web Site** (3).
10. In the **Actions pane**, under **Edit Site**, click **Bindings**
11. Edit/add **HTTP** and set the value to **28886**.



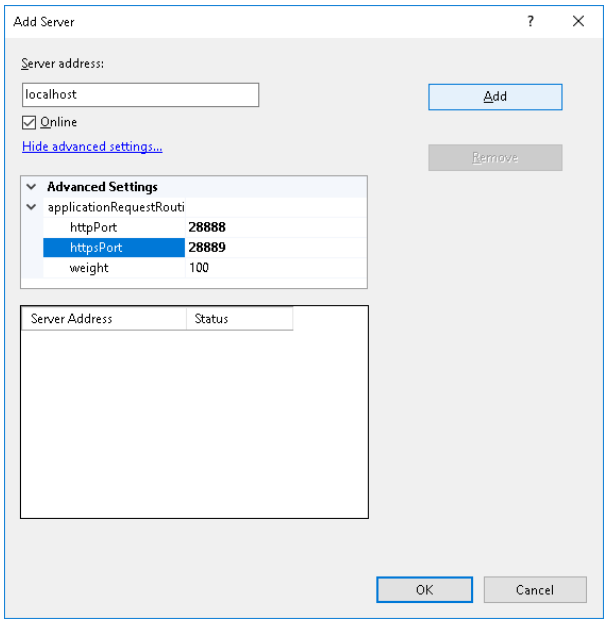
12. Click **Close**.

    Next, you will recreate the server in the Jetty Web Farm.
13. In the **Connections pane**, expand the **XHQ Server** > **Server Farms**.
14. Expand **arrtojetty** (1)> **Servers** and right-click **localhost**.
15. Click **Remove Server**.
    A message appears asking you to confirm removal.



16. Click **Yes**.
17. Right-click **localhost** again, and click **Add Server**.
    The "Add Server" dialog appears.
18. For **Server address**, enter **localhost**.
19. Show **Advanced Settings**.
20. Set **httpPort** and **httpsPort** to the same values set in the `jetty.properties` file.

Go to the topic, *About jetty.properties*. for the values to the corresponding properties, **`xhq.jetty.http.port`** and **`xhq.jetty.https.port`**.



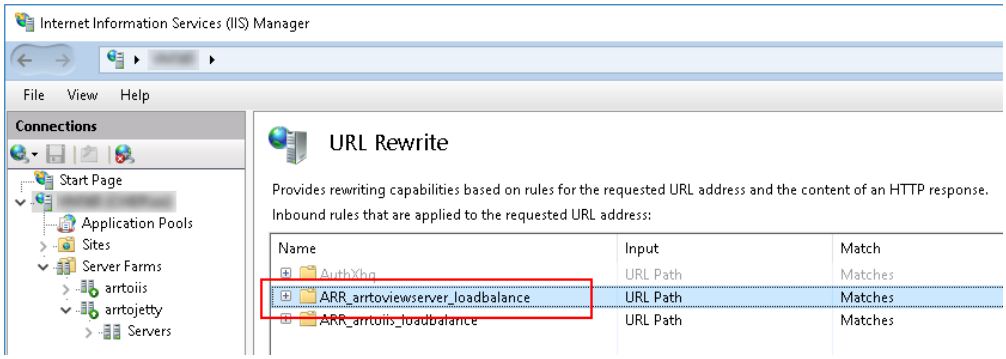
21. Click **Add** and then **OK**.

    In these final steps, you will modify the rewrite rules.

22. In the **Connections pane**, select the **XHQ server**.

23. From the **Workspace**, under the **IIS** group, click **URL Rewrite**.
    The "URL Rewrite" page appears.

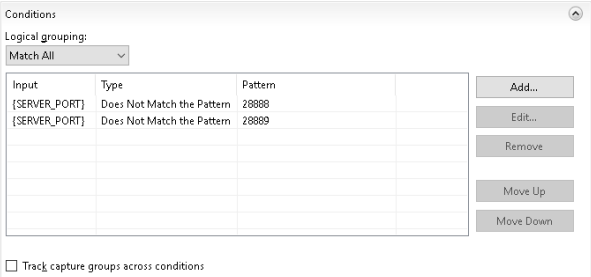
24. Double-click the **ARR_arrtoviewserver_loadbalance** rule.



    The "Edit Inbound Rule" page appears.

25. Scroll down to the **Conditions group**.

26. Edit the HTTP and HTTPS port conditions according to *step 20*.



27. From the **Action pane**, click **Apply**.
28. **Restart** the XHQ Server.

Configuring SSL for XhqWse Web Service

By default, XHQ installs the XhqWse web service with only HTTP enabled. The HTTP configuration for the WCF service is found in the `%XHQ_WEB_HOME%\webapps\xhqwse\web.config` file.

The code example below is a section from the `web.config` file that contains the HTTP (port 80) configuration. The highlighted text shows where changes need to be made to allow for SSL (HTTPS).

```
<system.serviceModel>
    <bindings>
      <basicHttpBinding>
        <binding name="XhqWseBinding" maxReceivedMessageSize="4194304">
          <security mode="TransportCredentialOnly">
            <transport clientCredentialType="Windows"/>
          </security>
          <readerQuotas
             maxArrayLength="4194304"
        />
        </binding>
      </basicHttpBinding>
</bindings>
    <services>
      <service behaviorConfiguration="Indx.Xhq.Wse.Service1Behavior"
name="Indx.Xhq.Wse.XhqWse">
        <endpoint address="" binding="basicHttpBinding"
bindingConfiguration="XhqWseBinding" contract="Indx.Xhq.Wse.IXhqService">
          <identity>
            <dns value="localhost"/>
          </identity>
        </endpoint>
      </service>
    </services>
    <behaviors>
      <serviceBehaviors>
        <behavior name="Indx.Xhq.Wse.Service1Behavior">
          <!-- To avoid disclosing metadata information, set the value below to false and
remove the metadata endpoint above before deployment -->
          <serviceMetadata httpGetEnabled="true"/>
          <!-- To receive exception details in faults for debugging purposes, set the
value below to true.  Set to false before deployment to avoid disclosing exception
information -->
          <serviceDebug includeExceptionDetailInFaults="true"/>
        </behavior>
      </serviceBehaviors>
    </behaviors>
  </system.serviceModel>
</configuration>
```

To enable the XhqWse web service to work with SSL (HTTPS)

1. Open the **`web.config`** file for the XhqWse Web Service, located in the `%XHQ_WEB_HOME%\webapps\xhqwse` directory (which by default is `<systemdrive>:\Program Files (x86)\XHQ\XHQ Web Root\webapps\xhqwse`).

2. Find the **security mode tag** and change the value to **`Transport`**.

```
<security mode="Transport">
```

3. Find the **endpoint address tag** and set the value using the FQDN for the server that hosts the WCF service.

```
<endpoint address="https://<MACHINE_FQDN>:8887/indx/xhqwse/XhqWse.svc"
```

4. OPTIONAL
   Find the **serviceMetadata tag** and make it accessible through HTTPS by adding the attribute `httpsGetEnabled="true"`.

```
<serviceMetadata httpGetEnabled="false" httpsGetEnabled="true"/>
```

This results in the following HTTPS configuration:

```
<system.serviceModel>
    <bindings>
      <basicHttpBinding>
        <binding name="XhqWseBinding" maxReceivedMessageSize="4194304">
          <security mode="Transport">
            <transport clientCredentialType="Windows"/>
          </security>
          <readerQuotas
             maxArrayLength="4194304"
        />
        </binding>
      </basicHttpBinding>
</bindings>
    <services>
      <service behaviorConfiguration="Indx.Xhq.Wse.Service1Behavior"
name="Indx.Xhq.Wse.XhqWse">
        <endpoint address="https://corp.acme.com:8887/indx/xhqwse/XhqWse.svc"
binding="basicHttpBinding" bindingConfiguration="XhqWseBinding"
contract="Indx.Xhq.Wse.IXhqService">
          <identity>
            <dns value="localhost"/>
          </identity>
        </endpoint>
      </service>
    </services>
    <behaviors>
      <serviceBehaviors>
        <behavior name="Indx.Xhq.Wse.Service1Behavior">
          <!-- To avoid disclosing metadata information, set the value below to false
and remove the metadata endpoint above before deployment -->
          <serviceMetadata httpGetEnabled="false" httpsGetEnabled="true"/>
          <!-- To receive exception details in faults for debugging purposes, set the
value below to true.  Set to false before deployment to avoid disclosing exception
information -->
          <serviceDebug includeExceptionDetailInFaults="true"/>
        </behavior>
      </serviceBehaviors>
    </behaviors>
  </system.serviceModel>
</configuration>
```

The XhqWse clients, including the Microsoft WCFTestClient utility, will also need to use the security mode Transport and the new endpoint address in their given configuration files.

For example:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <system.serviceModel>
        <bindings>
            <basicHttpBinding>
                <binding name="BasicHttpBinding_XHQ Web Service"
closeTimeout="00:01:00"
                    openTimeout="00:01:00" receiveTimeout="00:10:00"
sendTimeout="00:01:00"
                    allowCookies="false" bypassProxyOnLocal="false"
hostNameComparisonMode="StrongWildcard"
                    maxBufferSize="65536" maxBufferPoolSize="524288"
maxReceivedMessageSize="65536"
                    messageEncoding="Text" textEncoding="utf-8"
transferMode="Buffered"
                    useDefaultWebProxy="true">
                    <readerQuotas maxDepth="32" maxStringContentLength="8192"
maxArrayLength="16384"
                        maxBytesPerRead="4096" maxNameTableCharCount="16384"
/>
                    <security mode="Transport">
                        <transport clientCredentialType="Windows"
proxyCredentialType="None"
                            realm="" />
                        <message clientCredentialType="UserName"
algorithmSuite="Default" />
                    </security>
                </binding>
            </basicHttpBinding>
        </bindings>
        <client>
            <endpoint
address="https://corp.acme.com:8887/indx/xhqwse/XhqWse.svc"
                binding="basicHttpBinding"
bindingConfiguration="BasicHttpBinding_XHQ Web Service"
                contract="XHQWebService" name="BasicHttpBinding_XHQ Web
Service" />
        </client>
    </system.serviceModel>
</configuration>
```

The TestXhqwse sample web application only works with HTTP and, by default, is not configured to support SSL.

Web References

The following link shows how to configure SSL for IIS 7.0:

http://learn.iis.net/page.aspx/144/how-to-set-up-ssl-on-iis-7/

The following links shows how to configure and use self generated test certificates for client testing the SSL support if a real certificate, issued by a trusted authority (for example, VeriSign), is not available:

http://www.codeproject.com/KB/WCF/WCFSSL.aspx

http://msdn.microsoft.com/en-us/library/ff648498.aspx

SSL Access for the XHQ Applet and Web Applications

As with the `XhqWse` web service, XHQ installs by default the XHQ Applet and web applications (XHQ Alert Notification System (XHQ ANS), Target Management (TM), eLogs, and View Stats) with only HTTP (port 80) enabled. In order to enable SSL (HTTPS on port 443) for the XHQ Applet and these web applications, you must first configure the SSL certificate for the `indx` virtual directory (`https://localhost/indx`).

To configure the SSL certificate for the indx virtual directory

1. Open the **Internet Information Services (IIS) Manager**.
2. Navigate to and click **Default Web Site**.
   The "Default Web Site Home" panel appears.
3. Under **IIS**, click **SSL Settings**.
   The "SSL Settings" panel appears.
4. In the **SSL Settings panel**, check **Require SSL**.
   This applies the SSL settings to all XHQ applications, which include the XHQ Applet, XHQ ANS, XHQ View Statistics Utility, and the XHQ Performance Management (PM) web applications.
5. Update the XHQ View Statistics Utility, PM Administration, TM, and eLogs links for HTTPS (SSL) for the Web Application Keep Alive (WAKA) service in the **`XHQWebAppKeepAlive.xml`** file (which is located in `%XHQ_SERVER_REPOS%\conf\services\xhq_webappkeepalive` folder).

SSL Access for the XhqWse Web Service

By default, XHQ installs the XhqWse web service with only HTTP enabled. The HTTP configuration for the WCF service is found in the `%XHQ_WEB_HOME%\webapps\xhqwse\web.config` file.

The code example below is a section from the `web.config` file that contains the HTTP (port 80) configuration. The highlighted text shows where changes need to be made to allow for SSL (HTTPS).

```
<system.serviceModel>
    <bindings>
      <basicHttpBinding>
        <binding name="XhqWseBinding" maxReceivedMessageSize="4194304">
          <security mode="TransportCredentialOnly">
            <transport clientCredentialType="Windows"/>
          </security>
          <readerQuotas
             maxArrayLength="4194304"
        />
        </binding>
      </basicHttpBinding>
</bindings>
    <services>
      <service behaviorConfiguration="Indx.Xhq.Wse.Service1Behavior"
name="Indx.Xhq.Wse.XhqWse">
        <endpoint address="" binding="basicHttpBinding"
bindingConfiguration="XhqWseBinding" contract="Indx.Xhq.Wse.IXhqService">
          <identity>
            <dns value="localhost"/>
          </identity>
        </endpoint>
```

```
      </service>
    </services>
    <behaviors>
      <serviceBehaviors>
        <behavior name="Indx.Xhq.Wse.Service1Behavior">
          <!-- To avoid disclosing metadata information, set the value below to false and
remove the metadata endpoint above before deployment -->
          <serviceMetadata httpGetEnabled="true"/>
          <!-- To receive exception details in faults for debugging purposes, set the
value below to true.  Set to false before deployment to avoid disclosing exception
information -->
          <serviceDebug includeExceptionDetailInFaults="true"/>
        </behavior>
      </serviceBehaviors>
    </behaviors>
  </system.serviceModel>
</configuration>
```

Updating WAKA Links to use HTTPS

The **`XHQWebAppKeepAlive.xml`** file is located in `%XHQ_SERVER_REPOS%\conf\services\xhq_webappkeepalive` folder.

For XhqWse:

The link for HTTP for the Web Application Keep Alive (WAKA) service in the **`XHQWebAppKeepAlive.xml`** file is:

```
<Site name="XhqWse">
    <Settings>
      <Setting name="FastInterval" value="2" />
      <Setting name="Interval" value="10" />
      <Setting name="Debug" value="true" />
    </Settings>
    <Paths>
<Path name="XhqWse Service"      value="http://localhost/indx/xhqwse/Default.aspx" />
    </Paths>
</Site>
```

The link for HTTPS (SSL) for WAKA in the **`XHQWebAppKeepAlive.xml`** file is as follows.

```
<Site name="XhqWse">
    <Settings>
      <Setting name="FastInterval" value="2" />
      <Setting name="Interval" value="10" />
      <Setting name="Debug" value="true" />
    </Settings>
    <Paths>
<Path name="XhqWse Service"
value="https://localhost/indx/xhqwse/Default.aspx?useSSL=true" />
    </Paths>
</Site>
```

Depending on how the SSL certificate has been issued (for which machine name, and so forth), localhost may need to be replaced with the FQDN.

For ViewStats:

```
<Site name="ViewStats">
    <Settings>
```

```
      <Setting name="FastInterval" value="2" />
      <Setting name="Interval" value="20" />
      <Setting name="Debug" value="true" />
    </Settings>
    <Paths>
      <Path name="ViewStats Home Page"
value="https://localhost/indx/viewstats/Default.aspx" />
      <Path name="ViewStats Main Page"
value="https://localhost/indx/viewstats/MainPage.aspx" />
    </Paths>
</Site>
```

Depending on how the SSL certificate has been issued (for which machine name, and so forth), `localhost` may need to be replaced with the FQDN.

Updating Client Applications to use TLS

When SSL is used to access a XHQ web service (via HTTPS) and either the TLS 1.2 or 1.1 protocol is the only security transport protocol enabled, there are several ways to allow client applications to use TLS 1.2 or 1.1.

Method 1

1. Compile your client application against .NET Framework 4.6 (or later).
   a. From **Visual Studio**, open your project's property pages.
   b. In the **Application tab**, change the **Target Framework**.
2. Next, open the **`web.config`** file for the client application, located in the **`%XHQ_WEB_HOME%\webapps\xhqwse`** directory (which by default is `<systemdrive>:\Program Files (x86)\XHQ\XHQ Web Root\webapps\xhqwse`).
3. Make sure that the .NET Framework 4.6 (or later) is set as the HTTP runtime target framework

   Example:

   ```
   <system.web>
     <httpRuntime targetFramework="4.6.1"/>
   </system.web>
   ```

Method 2

1. On the client machine, run **`RegEdit.exe`**.
2. Go to `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\ .NETFramework\v4.0.30319`.
3. Add a **DWORD (32-bit)** value named **`SchUseStrongCrypto`**, and set it to **1**.
   This flag causes ServicePointManager.SecurityProtocol to be initialized to Tls | Tls11 | Tls12.

Method 3

When your client application starts up, turn on TLS 1.2 by adding the following code snippet to the web service client's app code (in the actual client invoker app).

```
ServicePointManager.SecurityProtocol |= SecurityProtocolType.Tls12;
```

XHQ Server Communication Security

Background

All XHQ modules and services communicate through Transport layer Security (TLS), using the Advanced Encryption Standard (AES) algorithm. XHQ generates a self-signed certificate that is used for the socket communications.

By default, XHQ uses the User Ports in conformance with IANA RFC6553. The XHQ default port assignment for the XHQ modules are referred to as ***Default Ports***. The *Legacy Ports* refer to the port assignment prior to the current version of XHQ.

These ports are used for XHQ **internal** communication and typically reside on the same machine. For **external** communication with clients, the Model Server and Solution Server need to be enabled on the firewall for successful communication.

For the list of port assignments, go to the topic, *Portmap Configuration File*.

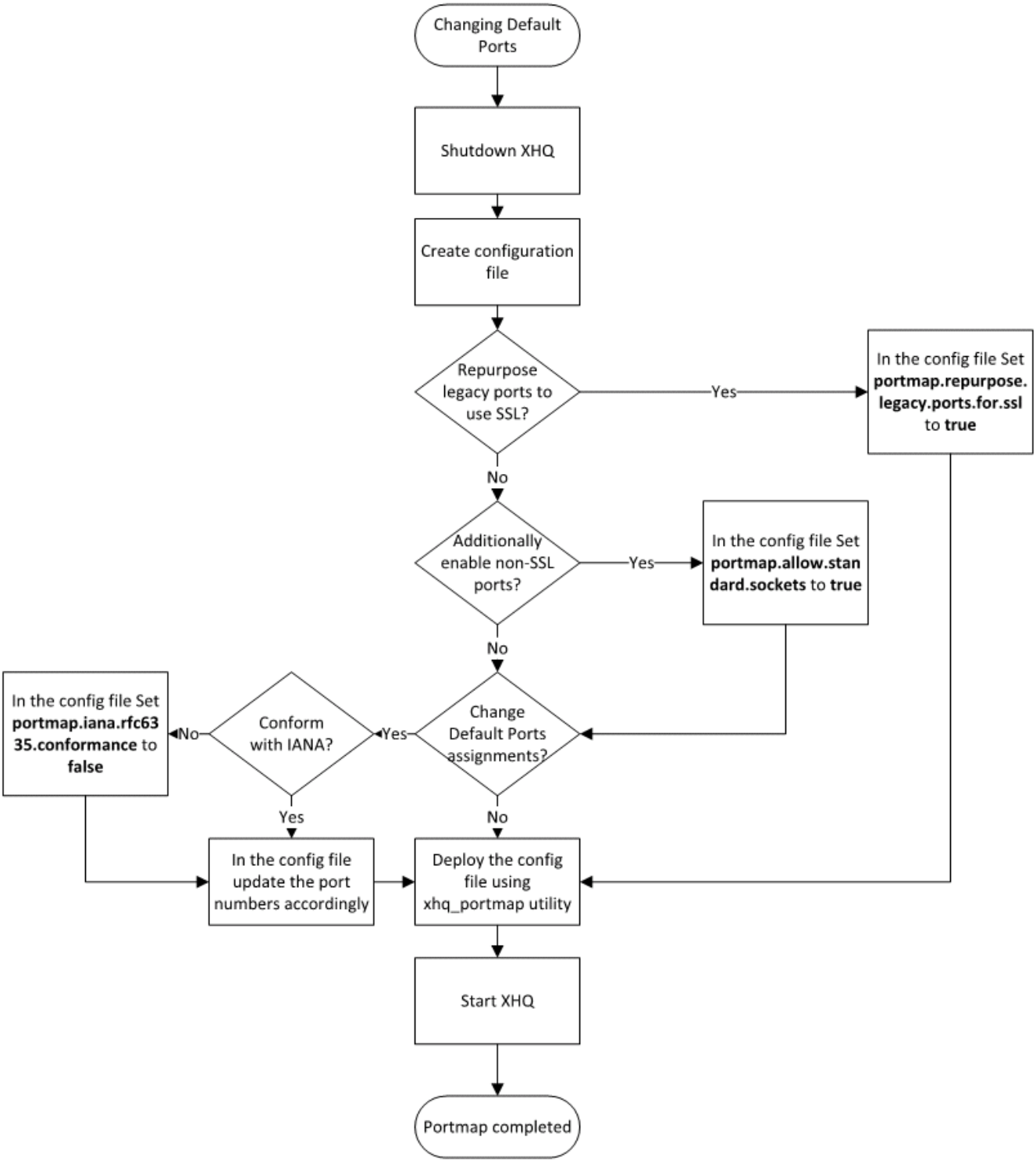
For port assignments related Jetty, refer to the topics, *XHQ Web Server Architecture and Security* or *About jetty.properties*.

By default, *Legacy Ports* are disabled and not used in any communication. Though it is possible to enable the legacy ports, however, *it is discouraged* (this option will be removed in a subsequent XHQ release).

Enabling the Legacy Ports allows earlier releases of XHQ Clients to connect to the XHQ Server in parallel with the Default Ports. For security reasons, however, we recommend you upgrade your existing clients, which leaves the Default Ports in place and the Legacy Ports disabled. A typical scenario where you may opt to enable the Legacy Ports is during an upgrade of XHQ in a mixed environment (where you have earlier releases of XHQ) and you are using the XHQ Tiered Connector.

The subsequent section describes how to update the configuration for the ports assignment.

Changing XHQ Port Configuration



The XHQ Portmap file deployed to the following location must not be edited directly:

`%XHQ_WEB_DATA%\conf\resources\public\portmaps\xhqportmap.properties`

To change the configuration of the XHQ ports

1. **Shutdown** XHQ.
2. Create a **portmap configuration** file.
   Refer to the topic, *Portmap Configuration File*, for possible configurations.
3. **Deploy** the portmap configuration file using the *XHQ Portmap Utility*.

Using the XHQ Portmap Utility

To execute the XHQ Portmap Utility (**`xhq_portmaputil`**), from XHQ Shell or from a CMD run as Administrator, go to the **`%XHQ_SERVER_HOME%\bin`** directory (which by default is `C:\Program Files (x86)\XHQ\XHQ Server\bin`). Run the **`xhq_portmaputil.bat`** file. This utility allows the XHQ Administrator to deploy and test the port configurations on the XHQ Server.

Syntax:

```
xhq_portmaputil /deploy [<config_file_path>]
xhq_portmaputil /test [<config_file_path>]
xhq_portmaputil /show
xhq_portmaputil [/help]
```

Example:

```
xhq_portmaputil /deploy c:\ports\xhqportmap.properties
```

Things to Note Regarding Syntax

- Parameters enclosed in **square brackets** "[ ]" are *optional*.
- Parameter values in **angle brackets** "<>" must be substituted with literal value.

  For example, the parameter, `<config_file_path>`, may be substituted with `C:\myPortMapTemp\PMC.txt`.
- Parameters may not include spaces unless enclosed in **double quotes**.
- Running this utility without any parameters shows command usage.

Parameter	Description
config_file_path	The full path to the configuration file. If the path contains **spaces**, then enclose the full path in **quotes**. Refer to the topic, *Portmap Configuration File*, for possible configurations. *Examples:* • `\\myShare\myPortMapConfig\PortMapConfig.properties` • `C:\myPortMapTemp\PMC.txt` • `"C:\my port map\portmap.config"`
/deploy	Deploys a Portmap based on the configuration properties file specified. If no file parameter is given, the default Portmap is deployed. *Deployment may only be done when XHQ is not running. Requires Administrator rights.* *Use any of the following*: • /deploy • /dep • /d
/test	Tests the given configuration file and displays the resulting Portmap (if valid) or error (for invalid file). If no file parameter is supplied, the currently deployed Portmap is tested instead. This option only tests a Portmap configuration and may be used when XHQ is either running or shutdown. *Use any of the following*: • /test

Parameter	Description
	• /t
/show	Shows verbose settings of the currently deployed PortMap. Copyright and "DO NOT EDIT" comments are suppressed in the output. *Use any of the following*: • /show • /s
/help	Displays the help message and exit. *Use any of the following*: • /help • /h

The virtual directory, `/indx/resources` (which points to `%XHQ_WEB_DATA%\conf\resources\public`), must have **Anonymous Access**, which is the default installation.

Portmap Configuration File

This configuration file defines the port configurations to be deployed in XHQ Server. There is no restriction on what type this file can be, so it can be saved with a `.txt` or a `.properties` extension. This configuration file must be validated by the **`xhq_portmaputil`** before being deployed to the XHQ Server.

The table below summarizes the port assignments (the port number for the TLS/SSL socket for the given module) and the property name to assign. By default, you can assign any available port on the XHQ Server in the User Ports range between 1024-49151.

Port 5109 is no longer used.

It is possible to specify a **unique port** using the Java Remote Method Invocation (RMI). Refer to the PHD or PI connector chapters of the XHQ Connection Guide.

XHQ Module Port Assignments

XHQ Module	Default Port	Legacy Port	Property Name for Port
Model Server	25100	5100	MODEL_SSL_PORT
Solution Server	25101	5101	SOLUTION_SSL_PORT
Cache Server	25102	5102	CACHE_SSL_PORT
Alert Server	25105	5105	ANS_SSL_PORT
Router Server	25108	5108	ROUTER_SSL_PORT
RDSL Server	25110	5110	RDS_SSL_PORT
Data Recorder	25112	5112	DB_HISTORIAN_SSL_PORT
Application Server	25115	5115	APP_SERVER_SSL_PORT

XHQ Module	Default Port	Legacy Port	Property Name for Port
Audit Server	25116	5116	AUDIT_SSL_PORT
View Server	25117	5117	VIEWSERVER_SSL_PORT
Connectors Start Port*	25120	5120	CONNECTOR_BASE_SSL_PORT
HTTPS	443		
HTTP	80		

To change the HTTP port from which the XHQ View Server listens for HTTP traffic, refer to the topic, *.About jetty.properties*.

How to display port information

There are a couple of ways to obtain port information:

- Open the **Windows Resource Monitor**. Click the **Network tab** and then expand **Listening Ports**. Note the processes prefixed by "xhq_" and the related port number.
- Or use the **XHQ Portmap Utility** with the `/show` parameter enabled.

The table below shows additional properties that can be set in this configuration file.

Portmap Configurations Properties

Property	Default Value	Description
portmap.iana.rfc6335.conformance	true	Specifies whether the port mapping should or should not conform to the IANA RFC6553 User Ports range.
portmap.allow.standard.sockets	false	Allows legacy ports to be opened for handling plain (non-SSL) sockets in the servers.
portmap.repurpose.legacy.ports.for.ssl	false	Repurpose the legacy ports for standard sockets for use as the SSL sockets ports. Setting this to true automatically sets portmap.iana.rfc6335.conformance and portmap.allow.standard.sockets to false. All other configuration properties in the port map configuration file are ignored when this property is set to true.

Comments are allowed in the configuration file.

To add a comment, start the line with the hash ("#") symbol.

Example:

#This is a comment.

XHQ Portmap Configuration File Example

```
portmap.repurpose.legacy.ports.for.ssl = false
# Repurpose the legacy ports for standard sockets for use as the SSL sockets ports.
```

```
#Setting this to true automatically sets portmap.iana.rfc6335.conformance and
#portmap.allow.standard.sockets to false.
# All other configuration properties in the port map configuration file
# are ignored when this property is set to true.
portmap.iana.rfc6335.conformance = true
# Specifies whether the port mapping should or should not conform
#to the IANA RFC6553 User Ports range. Default is true.
portmap.allow.standard.sockets = false
#Allows legacy ports to be opened for handling plain (non-SSL) sockets in the servers.
#Default is false.
#========================================
# SERVER MODULE PORTS FOR TLS/SSL SOCKETS
#========================================
MODEL_SSL_PORT = 25100
# Model Server: The port number for the TLS/SSL socket for the Model Server.
SOLUTION_SSL_PORT = 25101
# Solution Server: The port number for the TLS/SSL socket for the Solution Server.
CACHE_SSL_PORT = 25102
# Cache Server: The port number for the TLS/SSL socket for the Cache Server.
ANS_SSL_PORT = 25105
# Alert Server: The port number for the TLS/SSL socket for the Alert Server.
ROUTER_SSL_PORT = 25108
# Router Server: The port number for the TLS/SSL socket for the Router Server.
RDS_SSL_PORT = 25110
# RDSL Server: The port number for the TLS/SSL socket for the RDSL Server.
DB_HISTORIAN_SSL_PORT = 25112
# Data Recorder: The port number for the TLS/SSL socket for the Data Recorder.
APP_SERVER_SSL_PORT = 25115
# Application Server: The port number for the TLS/SSL socket for the Application Server.
AUDIT_SSL_PORT = 25116
# Audit Server: The port number for the TLS/SSL socket for the Audit Server.
VIEWSERVER_SSL_PORT = 25117
# View Server: The port number for the TLS/SSL socket for the View Server.
CONNECTOR_BASE_SSL_PORT = 25120
# Connector Processes: The base port number for the TLS/SSL socket given to a new
#connector - this number is increased by one for each new connector that is added
#to the solution.
```

Troubleshooting

For this Issue...	Do This...
XHQ is not starting up	Make sure the configured ports are not used by any other application. To check the current port mapping, run the following command: **`xhq_portmaputil /test`** You can update the port mapping using XHQ Portmap Utility.
Unable to connect to the XHQ Server remotely	Make sure your firewall configuration allows access to the Model/Enterprise Server, the Solution Server and Model Server. Additionally, make sure the `indx/resources/keystores` and `indx/resources/portmaps` have anonymous access in IIS.
Revert back to the default ports	To revert back to the default ports and configurations, use the following command-line with Admin privileges: **`xhq_portmaputil /deploy`**

For this Issue...	Do This...
	This command-line reverts back to defaults settings.

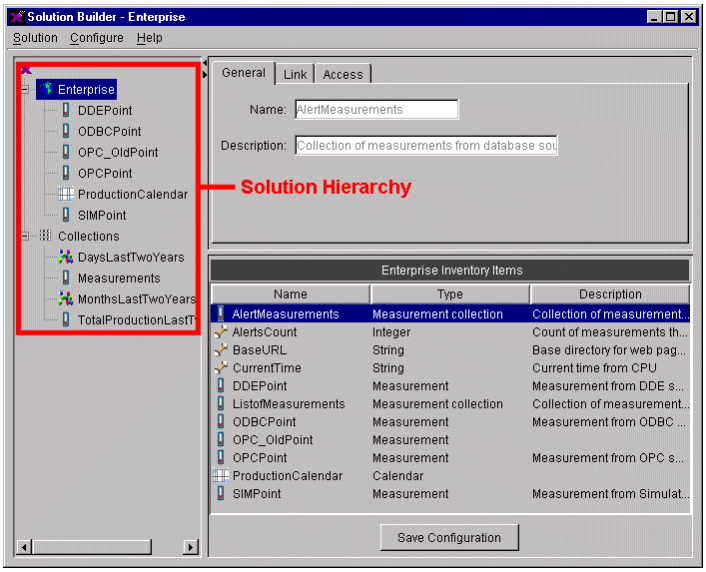
3 | Solution Deployment and Access

To deploy a solution, you define the top node and then attach it to the host computer on which it will run. Once you have defined a top node, you must restart the solution server. The solution server then recognizes the newly-defined top node and its solution, and builds the appropriate navigational tree by creating the instances of all its components and their members. Each time you change the top node, you create a new solution.

> SIDEBAR: COMPATIBILITY OF MULTIPLE VERSIONS OF THE XHQ APPLET
>
> There should not be a compatibility issue with running different versions of the XHQ applet. For example, pointing at web server A, then at web server B, then back to A should not require a new download of the XHQ applet from A even if there is a version mismatch between the applets that A and B are serving.

Defining the Solution

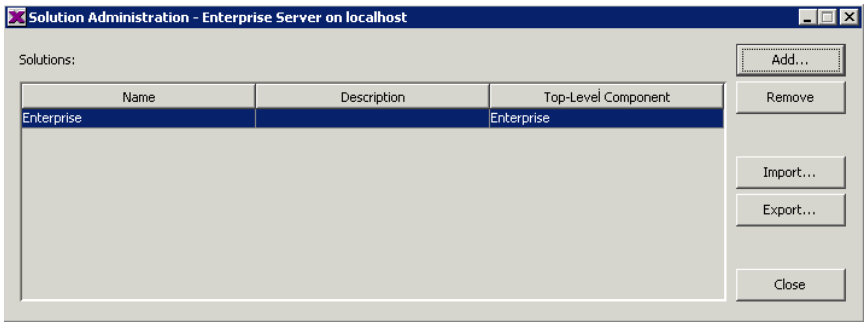
The top node of any solution identifies a unique solution. Positioned underneath the top node are the various organizational nodes and the informational components that fully represent the information being collected from back-end systems. The arrangement of its nodes describes a solution hierarchy and explicitly defines its operation and navigation. The solution hierarchy that you build appears as the run-time navigation tree.



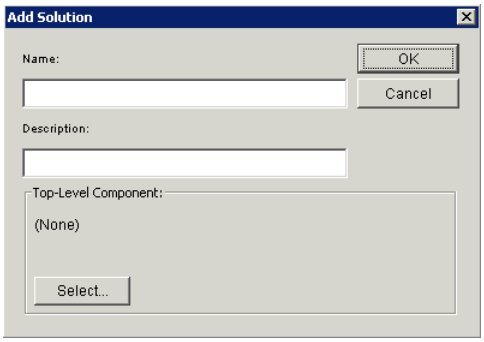
Solution Hierarchy

To define a solution and attach the single top node to the solution environment

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 a descriptive name for the solution in the **Name** box.

   Example: `Phazer_Electronics`

4. OPTIONAL
   Enter a description of the solution in the **Description** box.
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 the component and click **OK**.
7. Click **OK** to close the "Add Solution" dialog box.
8. Click **Close** to exit the "Solution Administration" dialog box.
9. **Restart** the Solution Server.

To rename a solution

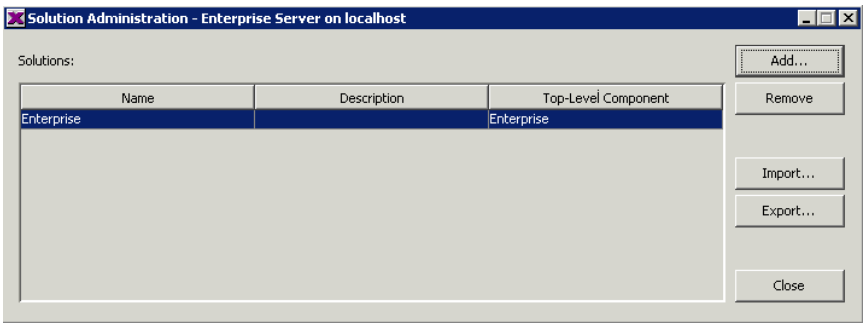
1. Backup the XHQ solution.
2. Export all **Connection Processes**.
3. Export all **Connection Groups**.
4. Export all **configurations**.
5. Locate and edit the **XML configuration file**.
6. **Search** for the existing root name you want to change (for example, ::Acme1), and **replace** it with the new root name.
7. **Save** the XML configuration file.
8. Click **Start**, point to **Programs**, then to **XHQ Development**, then to **Administration**, and click **Solutions**.
   The "Solution Administration" dialog box appears, listing all of the defined solutions.
9. **Select** the solution you wish to change and click **Edit**.
10. **Rename** the solution.
11. Click **OK**.
12. **Close** the Solution Administration dialog box.
13. **Stop** the XHQ Server.
14. Open the **`repos`** directory, which is typically stored at the location specified by the environment variable **`%XHQ_SERVER_REPOS%`**.
15. Open the Solution directory.
16. **Delete** the folder(s) in the Solution directory.
17. **Re-start** XHQ.
18. Launch the XHQ Solution Builder.
19. Import the **Connection Processes**.
20. Import the **Connection Groups**.
21. Import the **new configuration XML file** (see steps 5-7 above).
22. **Validate** the solution.

Deleting a Solution

Once you have defined and deployed a solution, you cannot edit its deployment, or change its hierarchical structure. You can remove the solution. However, the XHQ Solution Builder general configurations, such as history deployment and connector assignments, are not deleted. Therefore, to **completely delete a solution**, you need to remove the solution and delete the solution database.

To completely remove the solution

1. Click **Start**, point to **Programs**, then to **XHQ devClient**, then to **Administration**, and click **Solutions**.
   The "Solution Administration" dialog box appears, listing all of the defined solutions.



2. **Select a solution** from the list and click **Remove**.
   The name of the solution is removed from the list.
3. Next, remove the solution database:

```
<install directory>\XHQ\data\repos\Solution\<solution name>…
```

Accessing Your Solution

Once you have deployed a solution, you can then enable your browser to access that solution. You access an XHQ solution in the same way you access other web-based applications. And, if the XHQ Web Root feature is installed, you can also launch your solution directly by accessing the **`.html`** file (or **`.aspx`** if Active Directory security is enabled) from Windows Explorer.

The following procedures require the **XHQ Web Root feature** to be installed on the computer you are using.

For more information on the XHQ Web Root support files, refer to the topic, *About the XHQ Web Files*.

To open the solution from a browser

1. Open your **browser**.
2. In the browser's URL field, type the web server location for your solution.

   The address can either be a **fully qualified** web page URL or an IP address. The URL directs your browser to the appropriate XHQ web server and a particular `.html` and/or `.aspx` page.

   Examples: **Of server name/web page**

   `http://localhost/indx/xhqNavbar.aspx`

   `http://localhost/indx/AcmePlant.html`

   Examples: **Of IP addresses**

   `http://205.6.35.224/indx/xhqNavbar.aspx`

   `http://205.6.35.224/indx/AcmePlant.html`

   If this is the first visit to the page, a *certificate* will launch prior to the log on prompt. This certificate verifies the authenticity of the XHQ Solution Viewer.
3. Click **Enter**.

   The solution launches in the browser.

To access a specific view directly

1. Open your **browser**.
2. In the address field, enter the **fully qualified URL** for the view.

   Example URLs: **For the XHQ Solution Viewer applet**

   `http://localhost/indx/xhqnavbar.aspx/?solutionHome=Root.GIS~Markers_PassThrough`

   For the HTML5 XHQ Solution Viewer

   `http://localhost/indx/sv/#::Root/GIS/~Markers_PassThrough`
3. Click **Enter**.

   The specific view launches in the browser.

4 | XHQ Audit Trail

The XHQ Audit Trail is a subsystem of XHQ. It is designed to provide the web user interface access to audit records without logging into the XHQ Server. You can then filter the audit records by start date, end date, module name, record type, solution name, action, host name and audit record key word.

Launching XHQ Audit Trail

Step 1 - Start the XHQ Application Server

To use XHQ Audit Trail, you must *enable the XHQ Application Server*.

Step 2 - Create the auditsettings.properties File

This properties file is required by XHQ Audit Trail. It allows you to set general user preferences for XHQ Audit Trail. This file is not automatically included in an XHQ installation. Therefore, you must create, edit, and then save it to the location specified by the environment variable `%XHQ_SERVER_REPOS%`.

1. Using a text editor, **create** the `auditsettings.properties` file.
2. **Add** the following property:
   `audit.server.storage.mode=db`

   Only the `audit.server.storage.mode` property is required to launch XHQ Audit Trail.

   For a list of configurable XHQ Audit Trail properties go to the topic, *Audit Trail for XHQ Core*.
3. **Save** the file to `%XHQ_SERVER_REPOS%` (which by default is `%SystemDrive%\XHQ\data\repos`).
4. **Restart XHQ Server**.

Step 2 - Launch XHQ Audit Trail

Do one of the following:

- Launch **XHQ Performance Management**. Under **Configuration** > **XHQ Audit Trail**, click **browse**. This launches the main XHQ Audit Trail interface.

or

- From a supported browser, go to **`http://<server_name>/indx/audit/index.html`**

  Where `<server_name>` is the name of the machine on which the XHQ Web Server is running.

  Example URLs:
  - `http://acme_one/indx/audit/index.html`
  - `http://localhost/indx/audit/index.html`

Audit Trail for XHQ Core

The Audit Trail feature enables XHQ to store critical system actions. The Solution Server and the Model Server both send audit messages to the Audit Server, which stores these messages to CSV files. Audit log files are stored in "`%XHQ_SERVER_REPOS%"/audit`. All files and folders are automatically created when needed by the Audit Server.

The following configuration properties are located in the **`auditsettings.properties`** file.

For additional information on **`auditsettings.properties`**, go to the topic, *Launching XHQ Audit Trail.*

Configuration Properties

Property	Description
`audit.server.storage.mode=[db|file|none]`	This property switches the Audit Server storage mode. *Requires a restart of the XHQ system.* *Parameters:* • db – database storage (the default) • file – CSV files storage • none – turn off audit
`audit.server.showUserName=[true|false]`	This property displays/hides the user name in the audit. *Parameters:* • true – default , show user name • false – hide user name *This property is required to filter by User Name.*
`audit.server.storage.file.separator=[STRING]`	This property defines the separator for the CSV file. The default value is a comma (",").
`audit.server.storage.file.maxRecords=1000`	This property sets the maximum number of records in one CSV file. If the total number of records exceeds this set maximum, then the Audit Server creates a new CSV file. The default value is 1000.

Database Format

During the XHQ installation process, a new table, **`XHQ_AUDIT_LOG`**, is created in the XHQ schema. A view is also created, **`XHQ_AUDIT_LOG_V`**.

Both table and view have the following format:

```
LOG_ID NUMBER(38,0) NOT NULL,
TIMESTAMP DATE DEFAULT sysdate NOT NULL,
USER_NAME VARCHAR2(512 BYTE) NOT NULL,
REALM_NAME VARCHAR2(256 BYTE) NOT NULL,
SOLUTION_NAME VARCHAR2(256 BYTE) NOT NULL,
HOST VARCHAR2(256 BYTE) NOT NULL,
ACTION VARCHAR2(16 BYTE) NOT NULL,
OBJECT_TYPE VARCHAR2(64 BYTE) NOT NULL,
```

```
OBJECT_NAME VARCHAR2(64 BYTE) NOT NULL,
MODULE_IDNUMBER(38,0) NOT NULL,
```

Important Things to Note

- To retrieve data, use:
  `Select * from XHQ_AUDIT_LOG_V;`
- If the property `audit.server.showUserName` is set to false, then the `USER_NAME` field displays "???" characters.
- The `REALM_NAME` and `SOLUTION_NAME` fields may also display "???" characters if empty.

Audit CSV Format

Audit messages are written to the CSV file, which uses the following format:

`Date Time, User Name, Realm Name, Solution Name, Host Name, Action, Object Type, Object Name`

This is an example of an Audit CSV file:

```
21/08/2014
17:14:40.158,CHER\vmb,Sample,Sample,vm70.cher.rus.ugasoft.net,CREATE,CONNECTION_
GROUP,PISDK_GROUP
21/08/2014
17:15:17.141,CHER\vmb,Sample,Sample,vm70.cher.rus.ugasoft.net,UPDATE,CONNECTION_
PROCESS,mssql_coll
21/08/2014 17:15:51.930,CHER\vmb,Sample,Sample,vm70.cher.rus.ugasoft.net,CREATE,GLOBAL_
COLLECTION,::mssqlCollection
21/08/2014 17:16:00.854,CHER\vmb,Sample,Sample,vm70.cher.rus.ugasoft.net,RENAME,GLOBAL_
COLLECTION,::mssqlCollection-> ::mssqlCollectionTemp
21/08/2014
17:16:33.778,CHER\vmb,Sample,Sample,vm70.cher.rus.ugasoft.net,CREATE,CONNECTION_
PROCESS,PI_PROCESS_READ
21/08/2014 17:16:55.606,CHER\vmb, , ,vm70.cher.rus.ugasoft.net,CREATE,PRIMITIVE_
MEMBER,Top.Pump1
```

Audit Coverage

Server	Object		Create	Update	Delete	Rename	Move
Model	Component		x	x	x	x	not supported
	Member	Primitive	x	x	x	x	not supported
		Collection	x	x	x	x	not supported
		Component	x	x	x	x	not supported
		Group	x	x	x	x	not supported
	View		x	x	x	not supported	not supported
	View property		not supported	x	not supported	not supported	not supported
	VTC View		not supported	not supported	not supported	not supported	not supported
	Role		x	x	x	x	not supported
	Security mode		not supported	x	not supported	not supported	not supported
	Page		x	x	x	x	not supported

Audit Coverage

Server	Object		Create	Update	Delete	Rename	Move
Solution	Connector		x	x	x	x	not supported
	Connection	process	x	x	x	x	not supported
		group	x	x	x	x	not supported
	Member	group	x	not supported	x	x	x
		component	x	not supported	x	x	x
	Collection	Global	x	not supported	x	x	not supported
		Tag	x	not supported	x	x	not supported
	Virtual Tags		x	x	x	not supported	not supported
	Configuration	Alias	x	x	x	not supported	not supported
		Data Control	x	x	x	not supported	not supported
		Connect (INPUT)	x	x	x	not supported	not supported
		Expression	x	x	x	not supported	not supported
		History	x	x	x	not supported	not supported
		Access	not supported	x	not supported	not supported	not supported
		View Access	not supported	x	not supported	not supported	not supported
		Link (PROXY)	x	x	x	not supported	not supported
		Link (RDB_LINK)	x	x	x	not supported	not supported
		Cache (RDB_BASE)	not supported	x	not supported	not supported	not supported
	Update to the latest		x	not supported	x	not supported	not supported

Audit Trail for XHQ Performance Management Applications

The following are audit trail specifications for XHQ Performance Management web applications.

Audit Trail for XHQ Performance Management

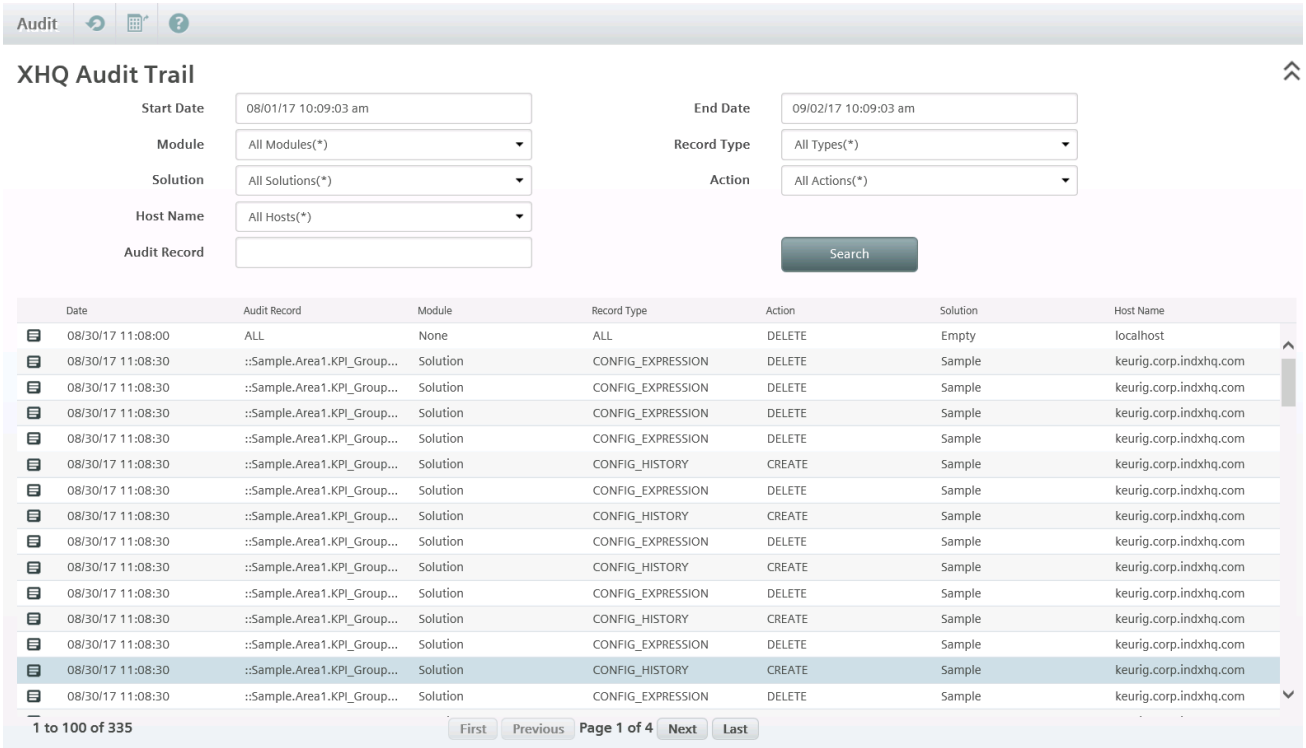
PM App	Feature	Create	Update	Delete	Import	Backfill
ANS	Alert Condition Definition	x	x	x		x
	Definition List	x	x	x		
	Subscription	x	x	x		
	Subscription List	x	x	x		
	Code Key	x	x	x		
	List Section	x	x	x		
	Notification Routing	x	x	x		
	Role Permissions		x			
	Condition Definition Security Roles	x		x		
	Interval	x	x	x		
	Schedule	x	x	x		
	Template	x	x	x		
	User	x	x	x		
	Imported Definitions				x	
	Imported Reason Codes				x	
	Deleted Definitions via Import				x	

Audit Trail for XHQ Performance Management

PM App	Feature	Create	Update	Delete	Import	Backfill
eLogs	Association Types	x	x	x		
	Unit of Measure	x	x	x		
	Condition Type	x	x	x		
	Condition	x	x	x		
	Condition List	x	x	x		
	Default Set	x	x	x		
	Multi Default Set	x	x	x		
	Generic User	x		x		
	Import Configurations				x	
	Operational Log	x	x			
	Log Status		x			
	Parameter Type	x	x	x		
	Parameter Group	x	x	x		
	Routine Parameter	x	x	x		
	Routine		x			
	Report Template	x	x	x		
	Shift Report	x	x			
Admin	Severity	x	x	x		
	Limit Class	x	x	x		
	Type	x	x	x		
	Sub-type	x		x		
	Role Types		x			
	Role Sub-types		x			
	Role Paths		x			
	Reason Code	x	x	x		
TM	Performance Indicator	x	x	x		
	Import PIs				x	
LO	Event Definition	x	x			
	Slice Reason		x			
RDSL	Subscribed Collection	x	x	x		
View Stats	Stats Report	x	x	x		

Viewing XHQ Audit Trail

There are two sections to the XHQ Audit Trail interface: the upper **Filter** section and the lower **Searched Audit Records** section.



XHQ Audit Trail Interface

By default, the Audit Trail displays all audit records that were logged from the previous month to the current time.

The Audit Trail does not capture the Solution name is there are changes in the model.

Setting Filters

Set the following filters to narrow your records search.

XHQ Audit Trail Filters

Filter	Description
Start Date	The start time and date for the report query.
End Date	The end time and date for the report query.
Module	Filter the audit records by XHQ module. Only the modules that appear in the database will be listed. Module options may include: “Admin”, “Alert Notification System”, “eLogs”, “Lost Opportunity”, “Model”, “Relational Data Store Load”,” Solution”, “System”, “Target Management”, “View Stats”, and “None”.
Record Type	The record type depends on the Module selected.
Solution	Filter the records by Solution. Determined by the available repos on the XHQ Server.
Action	There are five action you can filter records by (depending on the actions that appear in the database): CREATE, UPDATE, DELETE, IMPORT, or BACKFILL
Host Name	The hostname comes from the machine name that accessed XHQ.
Audit Record	This is a simple wildcard expression that is used to filter on the audit records. The default is empty, which means that all audit records are to be included. For example, you may narrow the scope by using a string, "Lo". In this case, only audit records contain the substring "Lo" are shown.

To change Start and End dates

1. Go to the **`%XHQ_WEB_HOME%\webapps\audit\app\config`** folder (which by default is `<systemdrive>:\Program Files (x86)\XHQ\XHQ Web Root\webapps\audit\app\config`).
2. Open the **`config.json`** file and locate the following snippet:

```
"audit": {
    "PAGE_SIZE": 100,
    "ROW_HEIGHT": 22,
    "HEADER_HEIGHT": 25
    "DEFAULT_SUBTRACT_PERIOD": {
        "MONTHS": 0,
        "DAYS": 0,
        "HOURS": 12
    }
    "DEFAULT_ADD_PERIOD": {
        "MONTHS": 0,
        "DAYS": 1,
        "HOURS": 0
    }
}
```

The **default value** for the START date (DEFAULT_SUBTRACT_PERIOD) is 12 hours before NOW. The default for the END date (DEFAULT_ADD_PERIOD) is 1 day after NOW.

3. Change the MONTHS, DAYS, or HOURS parameters.

   Example to change START date:

```
"audit": {
    "PAGE_SIZE": 100,
    "ROW_HEIGHT": 22,
    "HEADER_HEIGHT": 25
    "DEFAULT_SUBTRACT_PERIOD": {
        "MONTHS": 2,
        "DAYS": 5,
        "HOURS": 4
    }
    "DEFAULT_ADD_PERIOD": {
        "MONTHS": 0,
        "DAYS": 1,
        "HOURS": 0
    }
}
```

4. **Save** and close **`config.json`**.
5. **Clear the cache** and **refresh the browser** to apply the new parameters.

To reset filters

From the XHQ Audit Trail top navigation bar, click the **Reset icon**.

To export filtered audit records

1. Set the audit filters and click **Search**.
   The table of filtered audit records appears.
2. From the XHQ Audit Trail top navigation bar, click the **Export icon**.

5 | Working with .PROPERTIES Files

By editing the XHQ `.properties` files, you can set user preferences, change default settings, or enhance system performance. A `.properties` file is a text file that can be created and/or maintained with a text editor, such as Notepad.

The table below lists the various `.properties` files that you can edit in the XHQ System.

For a list of XHQ environment variables and their default paths (values), go to the topic, *Distribution of XHQ Server and devClient Files*, located in the XHQ Installation Guide.

List of XHQ .PROPERTIES Files

Filename	Location	Description
ans.properties	`%XHQ_SERVER_REPOS%`	Use this file to set the user preferences for the XHQ ANS system. **See Also**: For more information on this file, go to the topic, *XHQ Alert Notification System ANS Properties*.
appletsettings.properties	`%XHQ_SERVER_REPOS%`	Use this properties file to define settings for the applet. For example, to show/hide the XHQ ANS icon on the XHQ NavBar, set `var xansInstalled` to either true (show) or false (hide). **See Also**: The topic, *To show or hide the NavBar ANS icon*, located in the XHQ ANS User's Guide. **Important:** This file is not included with the XHQ install. It is created by the user and saved in the given location.
auditsettings.properties	`%XHQ_SERVER_REPOS%`	Allows you to set general user preferences for XHQ Audit Trail. **Important:** This file is not included with the XHQ install. It is created by the user and saved in the given location. **See Also**: For more information, go to the topic, *Launching XHQ Audit Trail*.
cachesettings.properties	`%XHQ_SERVER_REPOS%`	Allows you to set the debug log level on the cache server and the interval between executions of Java's "garbage collection" process in the XHQ Cache Server. **Important:** This file is not included with the XHQ install. It is created by the user and saved in the given location. **See Also**: For a complete list of configurable properties, go to the topic, *Using cachesettings.properties*.
clientsettings.properties	`%APPDATA%\XHQ\XHQ devClient\repos`	This file retains user preferences for the XHQ Development Client, such as fill color, text fonts, etc., used in the XHQ Workbench. **See Also**: For more information on this file, go to the topic, *About clientsettings.properties*.

Filename	Location	Description
clientvariablesettings.properties	`%XHQ_SERVER_REPOS%`	Use this properties file to define client variables for the applet. **See Also**: For more information on this file, go to the topic, *About clientvariablesettings.properties*.
edge.properties	`%ProgramFiles(x86)%\XHQ\XHQ Install Files`	This file enables you to configure the XHQ Edge Installer. **See Also**: For details, see the topic, *About edge.properties*, located in the XHQ Installation Guide.
globaleditingsettings.properties	`%XHQ_SERVER_REPOS%`	This file allows you to edit the preferences, or settings, that define the graphical attributes of various View building features (such as the color palette, text size, fill and line color, and so forth). **See Also**: For a complete list of configurable properties, go to the topic, *Using globaleditingsettings.properties*.
globalsettings.properties	`%XHQ_SERVER_REPOS%`	Allows you to set solution-wide options, such as preferences for Historical Navigation, the background color, and aggregation intervals. This file is included with the XHQ install and contains default settings. **See Also**: For a complete list of configurable properties, go to the topic, *Using globalsettings.properties*.
install.properties	`%ProgramFiles(x86)%\XHQ\XHQ Install Files`	This file enables you to configure the XHQ installer. **See Also**: For details, see the chapter, *The install.properties File*, located in the XHQ Installation Guide.
modelsettings.properties	`%XHQ_SERVER_REPOS%`	Allows you to set the debug log level on the enterprise server, and the interval for memory tests for Java garbage collection process in the Model (Enterprise) Server. **Important:** This file is not included with the XHQ install. It is created by the user and saved in the given location. **See Also**: For a complete list of configurable properties, go to the topic, *Using modelsettings.properties*.
monitoring.properties	`%XHQ_SERVER_REPOS%`	Use this properties file to customize the tests done during the health check, or to disable the testing (allowing the test to pass). **See Also**: For more information on this file, go to the topic, *About monitoring.properties*.
routersettings.properties	`%XHQ_SERVER_REPOS%`	**Important:** This file is not included with the XHQ install. It is created by the user and saved in the given location. **See Also**: For a complete list of configurable properties, go to the topic, *Using routersettings.properties*.

Filename	Location	Description
solutionsettings.properties	`%XHQ_SERVER_REPOS%`	Allows you to set the debug log level on the solution server, the interval between executions of Java's "garbage collection" process in the XHQ Solution Server (optimizing memory usage). This file also provides for other tunable parameters that deal with solution server performance. **Important:** This file is not included with the XHQ install. It is created by the user and saved in the given location. **See Also**: For a complete list of configurable properties, go to the topic, *Using solutionsettings.properties*.
storage.properties	`%XHQ_SERVER_REPOS%`	Use this properties file to define how connections are made to the external database. **See Also**: For more information on this file, go to the topic, *About storage.properties*.
viewserversettings.properties	`%XHQ_SERVER_REPOS%`	Use this properties file to control XHQ View Server settings. **See Also**: For more information on this file, go to the topic, *About the XHQ View Server and viewserversettings.properties*.
xhq_service.properties	`%XHQ_SERVER_HOME%\bin`	Use this properties file to control various XHQ Service behaviors. **Important:** This file is not included with the XHQ install. It is created by the user and saved in the given location. **See Also**: For more information on this file and how to create it, go to the topic, *About xhq_service.properties*.
xhqboot.properties	`%XHQ_SERVER_REPOS%`	This file contains property settings that govern how the start-up and shutdown scripts are generated. For example, if your application of the XHQ System requires other programs to be started after the XHQ Solution Server is running, then use this file to customize the configuration. **See Also**: For more information on this file, go to the topic, *About xhqboot.properties*.

Using globalsettings.properties

The **`globalsettings.properties`** file allows you to set solution-wide options. This file is stored at the location specified by the environment variable `%XHQ_SERVER_REPOS%` on the XHQ Enterprise Server (which by default is `<systemdrive>:\XHQ\data\repos`). Use a text editor to edit this file.

Property names are **case-sensitive**.

Example: globalsettings.properties file

Property Settings for XHQ HTML5 Solution Viewer

These are global properties related to the XHQ HTML5 Solution Viewer.

Global Property	Description
`sv.ShowDeviceOrientationOptions`	When true, shows the option to choose the device and orientation from the XHQ HTML5 Solution Viewer Button Bar. *Default*: `sv.ShowDeviceOrientationOptions = false`
`sv.ZoomPercentMax`	Sets the percentage for the *maximum* zoom level for the XHQ HTML5 Solution Viewer. *Default*: `sv.ZoomPercentMax = 200`
`sv.ZoomPercentMin`	Set the percentage for the *minimum* zoom level for the XHQ HTML5 Solution Viewer. *Default*: `sv.ZoomPercentMin = 100`
`sv.ShowEarlyAccessFeatures`	When true, enables the alpha features, which are intended for customer evaluation. *Default*: `sv.ShowEarlyAccessFeatures = false`
`sv.ShowPreloader`	When true, enables the view preloader. *Default*: `sv.ShowPreloader = true`

Property Settings for the Trend Viewer

Global Property	Description
`InteractiveTrenderDefaultHeight`	Customizes the Interactive Trender window height (in pixels). *Default:* InteractiveTrenderDefaultHeight=768
`InteractiveTrenderDefaultWidth`	Customizes the Interactive Trender window width (in pixels). *Default:* InteractiveTrenderDefaultWidth=1024
`ShowTagBrowserButton`	To show the Browse Tag button in the Trend Viewer (Interactive Trender), set this property to **true**; else, *Default:* ShowTagBrowserButton=false **Note:** When clicked, the Browse Tag button launches the Tag Selector dialog box. For more information on the Tag Selector, go to the topic, *Working with Tags*, located in the XHQ Developer's Guide.
`DisableTrenderPin`	When adjusting time parameters, this property enables (shows) the push pin icon if false, and disables (hides) it if true. By disabling the push pins, the user can easily set the start and end times for the trend, regardless of the duration. *Default:* DisableTrenderPin=false For more information, go to the topic, *Adjusting Time Parameters*, located in the XHQ Trend Viewer User's Guide.
`AggregateAverageList`	This sets the list of selectable refresh periods for the XHQ Trend Viewer (s=seconds, m=minutes, h=hours, d=days, w=weeks). Note, however, that only values greater than 60 seconds and less than 1 day are acceptable for **auto refresh**. *Default:* AggregateAverageList = 30s;5m;2h;5d;5w

Property Settings for the Navigation Bar

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

To change a NavBar property

1. Find the **`globalsettings.properties`** file and disable the READ ONLY property.
   This file is stored at the location specified by the environment variable `%XHQ_SERVER_REPOS%` on the Enterprise Server (which is typically `C:\XHQ\data\repos`).
2. Open the **`globalsettings.properties`** file in a text editor (such as Notepad).
3. Add (or find and edit) the property.

   Example: `NavbarTextMouseOverColor=255,0,0`

   Property names are **case-sensitive**.

4. **Save** and **close** the text editor.
5. **Start** a new session of the **Solution Viewer** (browser client).

Using Custom Images

If you are using custom images, make sure that all images are in `.GIF` format. In addition, you may need to create an **`images`** folder in the **`repos`** directory under **`\XHQ\data`** (that is, `%XHQ_SERVER_REPOS%\images`) to house all custom images.

Each custom image needs to be the same size as the default image it replaces. All button images must be 22 x 22 pixels, logos must be 53 x 18 pixels, and navigation interface images (such as the tabs) must be 60 x 16 pixels in dimension.

Example: Store the custom image in the **`images`** folder of the **`repos`** directory, then edit the **`globalsettings.properties`** file.
*Default:*NavbarLogoImg=indxsmlogoicon.gif
Custom: NavbarLogoImg=my_logo.gif

Image Description	Default Image	Global Property	Required Dimension (w x h, in pixels)
Navigation interface "Path" image	PATH	NavbarPathImg	60 x 16
Navigation interface "Current" image	CURRENT	NavbarCurrentImg	60 x 16
Navigation interface "Down" image	DOWN	NavbarDownImg	60 x 16
Navigation interface "Views" image	VIEWS	NavbarViewsImg	60 x 16
Corporate logo image (located in the NavBar Button Bar)		NavbarLogoImg	53 x 18
Corporate logo image (located in the West Region)		NavbarLogoImg	no restriction

Image Description	Default Image	Global Property	Required Dimension (w x h, in pixels)
Active "Backward" image		NavbarBackwardImg1	22 x 22
Inactive "Backward" image		NavbarBackwardImg3	22 x 22
Mouseover "Backward" image		NavbarBackwardImg2	22 x 22
Active "Home" image		NavbarHomeImg1	22 x 22
Mouseover "Home" image		NavbarHomeImg2	22 x 22
Active "Forward" image		NavbarForwardImg1	22 x 22
Inactive "Forward" image		NavbarForwardImg3	22 x 22
Mouseover "Forward" image		NavbarForwardImg2	22 x 22
Mouseover "Reset Filters" image		NavbarRefreshImg1	22 x 22
Active "Reset Filters" image		NavbarRefreshImg2	22 x 22
Mouseover "History" image		NavbarHistoryImg1	22 x 22
Active "History" image		NavbarHistoryImg2	22 x 22
Mouseover "Alert" image		NavbarAlertImg1	22 x 22
Active "Alert" image		NavbarAlertImg2	22 x 22
Mouseover "Print" image		NavbarPrintImg1	22 x 22
Active "Print" image		NavbarPrintImg2	22 x 22
Mouseover "Send Email" image		NavbarEmailImg1	22 x 22
Active "Send Email" image		NavbarEmailImg2	22 x 22
Mouseover "Toggle Scrollbars" image		NavbarToggleImg1	22 x 22
Active "Toggle Scrollbars" image		NavbarToggleImg2	22 x 22
Mouseover "User Setting" image		NavbarUserImg1	22 x 22

Image Description	Default Image	Global Property	Required Dimension (w x h, in pixels)
Active "User Setting" image		NavbarUserImg2	22 x 22
Mouseover "Add View to My Views" image		NavbarViewsImg1	22 x 22
Active "Add View to My Views" image		NavbarViewsImg2	22 x 22
Mouseover "Make this view your Home View" image		NavbarMakeHomeImg1	22 x 22
Active "Make this view your Home View" image		NavbarMakeHomeImg2	22 x 22
Mouseover "Solution Viewer Help" image		NavbarHelpImg1	22 x 22
Active "Solution Viewer Help" image		NavbarHelpImg2	22 x 22

To replace a default image with a custom image

About Custom Logos
The filename of your custom logo must use alphanumeric characters only.

1. Go to the location specified by the environment variable `%XHQ_SERVER_REPOS%` on the Enterprise Server (which is typically `C:\XHQ\data\repos`).
2. Create an **`images`** folder.

   Example: `XHQ\data\repos\images`

3. Copy your custom image into this **`images`** folder.

   Custom image dimensions MUST BE THE SAME as the default image being replaced.

4. Find the **`globalsettings.properties`** file and disable the READ ONLY property.
   This file is also stored at the location specified by the environment variable `%XHQ_SERVER_REPOS%` on the Enterprise Server (which is typically `C:\XHQ\data\repos`).
5. Open the **`globalsettings.properties`** file in a text editor (such as Notepad).
6. Add (or find and edit) the appropriate property and set it equal to the name of your image.

   Example: NavbarLogoImg=my_logo.gif

7. **Save** and close the text editor.
8. **Restart** the server and open a new session of the XHQ Solution Viewer (browser client).

Showing or Hiding the Logo

Show/Hide Logo

Global Property	Description
`NavbarShowLogo`	If **true**, show logo. If **false**, hide logo. *Default:* NavbarShowLogo=true
`NavbarLogoImg`	This sets the logo image. *Default:* NavbarLogoImg=indxsmlogoicon.gif

To show/hide the logo

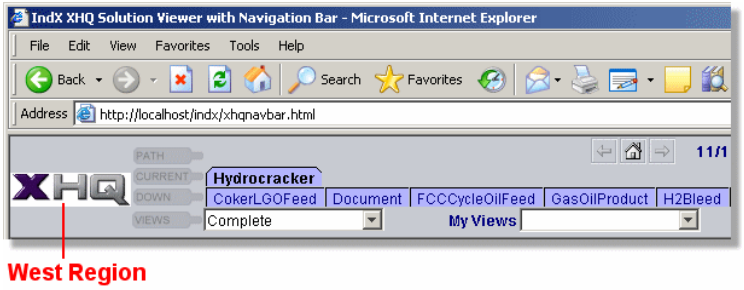
1. Find the **`globalsettings.properties`** file and disable the READ ONLY property.
   This file is stored at the location specified by the environment variable `%XHQ_SERVER_REPOS%` on the Enterprise Server (which is typically `C:\XHQ\data\repos`).
2. Open the **`globalsettings.properties`** file in a text editor (such as Notepad).
3. Add (or find and edit) the **`NavbarShowLogo`** property and set its value.
4. Save and close the text editor.
5. Open a new session of the XHQ Solution Viewer (browser client).

Placement Customization

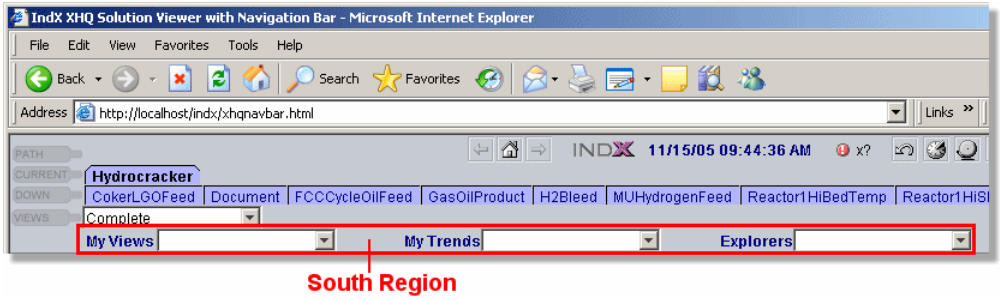
The XHQ NavBar panel is divided into three regions: West, South, and Main.

There are two placement customizations currently supported:

- Placing the logo in the **West** region.



- Placing the **Favorites** in the **South** region.



To place the logo in the West region

1. Open the **`globalsettings.properties`** file in a text editor (such as Notepad) and set the following global properties to the given values.
   - NavbarLogoPosition=west
   - NavbarWestVerticalOffset=0

   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.

2. **Save** and close the text editor.
3. Open a new session of the XHQ Solution Viewer (browser client).

To place the favorite panel in the South region

1. Open the **`globalsettings.properties`** file in a text editor (such as Notepad) and set the following global properties to the given values.
   - NavbarFavoritePanelPosition=south
   - NavbarSouthHorizontalOffset=50

   Use the `NavbarSouthHorizontalOffset` 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.

2. **Save** and close the text editor.
3. Open a new session of the XHQ Solution Viewer (browser client).

Showing or Hiding Button Bar Icons

Show/Hide Button Bar Icons

Global Property	Description
`NavbarShowMail`	Show/hide the **Send Mail** icon on the NavBar Button Bar. If **true**, show icon. If **false**, hide icon. *Default:* NavbarShowMail=true
`NavbarShowPrint`	Show/hide the **Print** icon on the NavBar Button Bar. If **true**, show icon. If **false**, hide icon. *Default:* NavbarShowPrint=true
`NavbarShowToggle`	Show/hide the **Toggle Scrollbars** icon on the NavBar Button Bar. If **true**, show icon. If **false**, hide icon. *Default:* NavbarShowToggle=true
`ShowTileComposerLink`	Show/hide the **Visual Tile Composer** icon on the NavBar Button Bar. If **true**, show icon. If **false**, hide icon. *Default:* ShowTileComposerLink=true
`NavbarShowHelp`	Show/hide the **Solution Viewer Help** icon on the NavBar Button Bar. If **true**, show icon. If **false**, hide icon. *Default:* NavbarShowHelp=true

Creating Custom Buttons

Custom buttons allow you to extend existing XHQ NavBar action. The global property to use for custom buttons is **`NavbarCustomActions`**. Remember to use an image that is 22 x 22 pixels in dimension, which is the default image size for XHQ NavBar buttons, and to place the .GIF image in the custom images directory for the NavBar (typically, this is under the `\XHQ\data\repos\images` directory). Custom buttons are then placed before the "Help" button of the NavBar button bar.

To create a custom button, there are four elements you will need to configure:

- **<tooltip>**
  The tooltip for the custom button.
- **<image1>**
  The active image for the button.
- **<image2>**
  The mouseover image for the button.
- **<action>**
  The JavaScript function to invoke when the button is pressed.

The global property will then have the following format and value:

`NavbarCustomActions=[<tooltip>|<image1>|<image2>|<action>]`

Example: Either of the following formats are valid.

NavbarCustomActions= [Hello Tom|tom-button.gif|tom-over-button.gif|ExternalHelloWorld ('Tom')][Hello Dan|dan-button.gif|dan-over-button.gif|ExternalHelloWorld('Dan')]

or

NavbarCustomActions=

[Hello Tom|tom-button.gif|tom-over-button.gif|ExternalHelloWorld('Tom')]\

[Hello Dan|dan-button.gif|dan-over-button.gif|ExternalHelloWorld('Dan')]

NavBar Clock Properties

Clock Properties

Global Property	Description
`NavbarShowClock`	If **true**, show clock. If **false**, hide clock. *Default:* NavbarShowClock=true
`NavbarClockFormat`	This is the NavBar clock format setting. NavbarClockFormat=DefaultTimeStampFormat *Default:* NavbarClockFormat=MM/dd/yyy hh:mm:ss (which is the default time stamp format)
`NavbarClockTextColor`	This is the NavBar clock text color setting. *Default:* NavbarClockTextColor=0,0,128 This is an RGB value corresponding to the color Navy Blue.
`NavbarClockRefreshRate`	This is the NavBar clock refresh rate setting (in milliseconds).

Global Property	Description
	Default: NavbarClockRefreshRate=1000 The default value of 1000 means 1000 milliseconds. The clock, therefore, refreshes once every second.

Alarm Update Setting

This flag only applies if you have the XHQ Alert Notification System (ANS) installed.

Global Property	Description
NavbarAlertUpdateRate	The poll rate to retrieve an active alert count from the application server. The default value is 60,000 milliseconds (ms). *Example:* NavbarAlertUpdateRate=30000

Changing the NavBar Background

The NavBar background color is set using RGB values. By default, the background image is not set. When a background image is specified, it overwrites the background color.

Global Property	Description
`NavbarBgColor`	The background color. *Default:* NavbarBgColor=204,204,204
`NavbarBgImage`	The background image. You will need to add this property to the `globalsettings.properties` file, and then save the file. *Example:* NavbarBgImage=indx_background.gif For more information, see the topic, *Using Custom Images*.

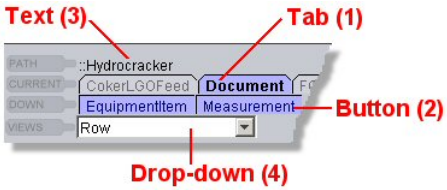
Setting the NavBar Icons Style

If you prefer the classic NavBar icons look (which appears in XHQ 4.6 and older), then set the following property.

Global Property	Description
`UITheme`	Sets the style for the NavBar icons. *Example:* UITheme=classic If UITheme is not defined (which is the default) or is not valid, then the updated NavBar icons style is used. If UITheme is set to classic, then the NavBar icons style in XHQ 4.6 (or older) is used.

Changing the Interface Type

For each navigation level (path, current, down, or view), you can use one of the four predefined interface types: text, button, tab, or dropdown.



To change the interface type for a particular navigation level, set the global property to a value associated with the given interface type ("1" for tab; "2" for button; "3" for text; "4" for drop-down).

The drop-down type is not supported for the path level.

By default, each navigation level is set to the following interface types.

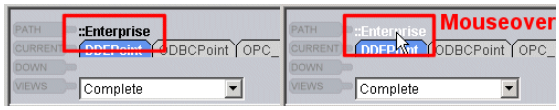
Navigation Level	Default Type	Global Property	Default Value
Path	text	NavbarPathComponentType	3
Current	tab	NavbarCurrentComponentType	1
Down	button	NavbarDownComponentType	2
View	drop-down	NavbarViewComponentType	4

Editing the Interface Type Attributes

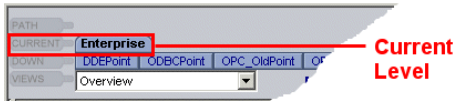
NavBar interface types are made up of graphical elements, consisting of lines, edges, text, and fill color. From the **`globalsettings.properties`** file, you can modify the tab or button color by specifying a valid RGB value.

The text on the tab is dynamically generated by the browser. Tab labels are actual XHQ solution tree information.

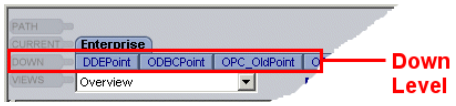
Path Level Properties

Global Property	Description
`NavbarTextMouseOverColor`	"Path" level text mouseover color. *Default:* NavbarTextMouseOverColor=150,150,150  *Path TextPath Text Mouseover*
`NavbarTextFgColor`	"Path" level text color. *Default:* NavbarTextFgColor=0,0,0

Current Level Properties

Global Property	Description
`NavbarTabFgColor`	"Current" level navigation tab text foreground color (for unselected tabs). *Default:* NavbarTabFgColor=105,105,105 
`NavbarTabSelectedFgColor`	Selected "Current" level navigation tab text foreground color. *Default:* NavbarTabSelectedFgColor=0,0,0
`NavbarTabMouseOverColor`	"Current" level navigation tab mouseover color (for unselected tabs). *Default:* NavbarTabMouseOverColor=200,200,200
`NavbarTabSelectedMouseOverColor`	Selected "Current" level navigation tab mouseover color. *Default:* NavbarTabSelectedMouseOverColor=200,200,200
`NavbarTabBgColor`	"Current" level navigation tab background color (for unselected tabs). *Default:* NavbarTabBgColor=210,210,220
`NavbarTabSelectedBgColor`	Selected "Current" level navigation tab background color. *Default:* NavbarTabSelectedBgColor=180,180,250

Down Level Properties

Global Property	Description
`NavbarBtnFgColor`	"Down" level navigation button text foreground color (for unselected buttons). *Default:* NavbarBtnFgColor=0,0,128 
`NavbarBtnSelectedFgColor`	Selected "Down" level navigation button text foreground color. *Default:* NavbarBtnSelectedFgColor=0,0,0
`NavbarBtnMouseOverColor`	"Down" level navigation button mouseover color (for unselected buttons). *Default:* NavbarBtnMouseOverColor=200,200,200
`NavbarBtnSelectedMouseOverColor`	Selected "Down" level navigation button mouseover color. *Default:* NavbarBtnSelectedMouseOverColor=200,200,200

Global Property	Description
`NavbarBtnBgColor`	"Down" level navigation button background color (for unselected buttons). *Default:* NavbarBtnBgColor=180,180,255
`NavbarBtnSelectedBgColor`	Selected "Down" level navigation button background color. *Default:* NavbarBtnSelectedBgColor=100,100,255

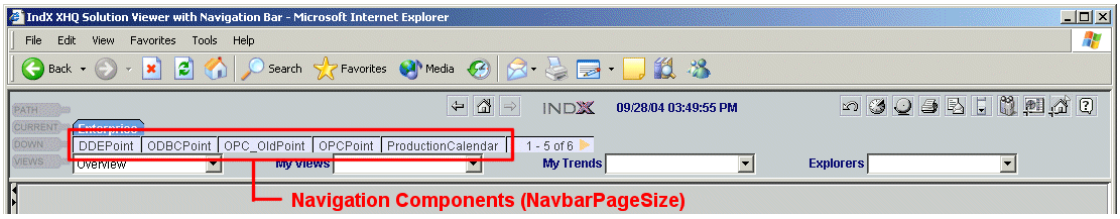
Additional NavBar Properties

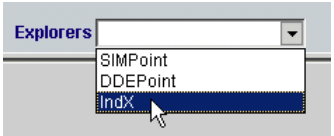
Global Property	Description
`NavbarLabelTextColor`	This is the NavBar label color for the "My Views," "My Trends," and "Explorers" drop-down box labels. *Default:* NavbarLabelTextColor=0,0,128

Editing Navigation Settings

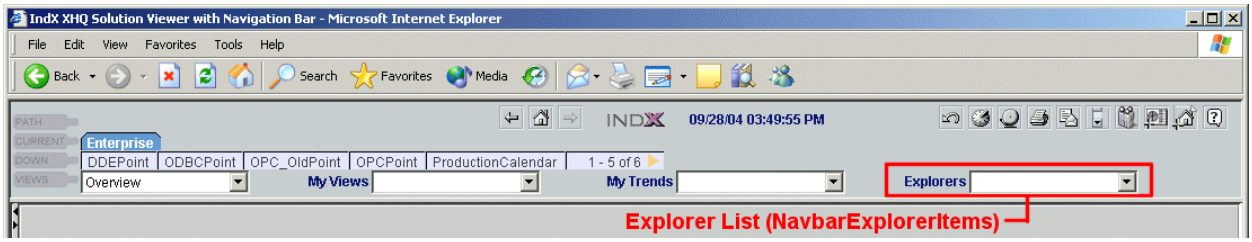
Global Property	Description	Example
NavbarPageSize	The number of navigation components (tabs) displayed in a page. If the total number is greater than the page size, a scrollable component appears at the end. The default size is 10.	NavbarPageSize=7

The NavbarPageSize property sets the number of component member tabs to display at once



NavbarExplorerItems	The list of predefined view links and URLs. Each entry uses the format: <display_name>|<view_link> *or* <display_name>|<URL> *or* <display_name>|<JavaScript> View links are separated by a comma "," or a comma/line continue character combination ",\".	NavbarExplorerItems=SIMPoint|::Enterprise. SIMPoints~Alerts,DDEPoint|::Enterprise. OPC Point~Complete,IndX|http://www.siemens.com /indx This example results in a list containing the following: 

The NavbarExplorerItems property sets the items listed in the Explorer drop-down list



Additional Global Properties

Global Property	Description
`DisableNavbarAutoResize`	If **true**, then the automatic XHQ NavBar resize feature is disabled. If **false**, then the feature is enabled.

Global Property	Description
	Default: DisableNavbarAutoResize=false
`UseClassicNavigation`	For the Solution Viewer (**browser client**) only. If **false** (default), when you right-click on the View panel, the shortcut menu does not include "Go" and "View" from the submenu. If **true**, when you right-click on the View panel, the shortcut menu does include "Go" and "View" from the submenu. *Default:* UseClassicNavigation=false

To customize the tab order for component members

1. From the XHQ Workbench, **select the component**.
   The Component Details tab appears.
2. Under **Inventory Items**, **select a component member**.
3. In the **General** property, in the **Description** text box, **enter the order number** enclosed by square brackets. The brackets and the order numbers can be anywhere in the description field (beginning, end, or middle).

4. **Save** your changes and close the Component Details tab.
5. From the **XHQ Solution Builder**, click **Update to Latest**.

To set the number of component member tabs to display at once

1. Find the **`globalsettings.properties`** file and disable the READ ONLY property.
   The `globalsettings.properties` file allows you to set solution-wide options. This file is stored at the location specified by the environment variable `%XHQ_SERVER_REPOS%` on the Enterprise Server (which is typically `C:\XHQ\data\repos`).
2. Open the **`globalsettings.properties`** file in a text editor (such as Notepad).
3. Add (or find and edit) the **`NavbarPageSize`** property.

   Example: `NavbarPageSize=7`

4. **Save** and close the text editor.
5. Start a new session of the XHQ Solution Viewer (browser client).

To set a maximum number of Trends/Views to a favorites list

You must have access to the `globalsettings.properties` file in order to do the following procedure. If you do not have access, please see your XHQ Administrator.

1. Open the **`globalsettings.properties`** file in a text editor (such as Notepad).
   The `globalsettings.properties` file allows you to set solution-wide options. This file is stored at the location

specified by the environment variable `%XHQ_SERVER_REPOS%` on the Enterprise Server (which is typically `C:\XHQ\data\repos`).

2. To change the maximum number of trends to list, find and edit the **`MaxSavedTrends`** property.

   Example: `MaxSavedTrends=10`

3. To change the maximum number of views to list, find and edit the **`MaxSavedViews`** property.

   Example: `MaxSavedViews=10`

4. **Save** and close the text editor.
5. Start a new session of the XHQ Solution Viewer.

To set the number of items on the Explorer drop-down list

1. Open the **`globalsettings.properties`** file in a text editor.
   This file is stored at the location specified by the environment variable `%XHQ_SERVER_REPOS%` on the Enterprise Server (which is typically `C:\XHQ\data\repos`).
2. Add (or find and edit) the **`NavbarExplorerItems`** property, using the format: `<display_name>|<view_link` *OR* `url` *OR* `JavaScript>`. For multiple items, use one of the following to separate each item.
   - Use a comma "," if you are listing a few items.

   Example 1: NavbarExplorerItems=SIMPoint|::Enterprise.SIMPoints~Alerts,OPCPoint|::Enterprise.OPC Point~Complete,IndX|http://www.siemens.com/indx

   Results in a list containing the following:

   SIMPoint
   OPCPoint
   IndX

or

   - Use the comma and line continue character combination ",\" if you are listing many items.

   Example 2: NavbarExplorerItems= Overview|::Enterprise~Overview,\
   DDEPoint|::Enterprise.OPCPoint~Complete,\
   SIMPoint|::Enterprise.SIMPoints~Alerts,\
   Temps|http://docweb3.indx.net/temperatures/stats.aspx,\
   Calendar|http://www.indx.net/dates.htm,\
   ... ,\
   ... ,\
   XHQ|http://www.siemens.com/mes

   Results in a list containing the following:

   Overview
   OPCPoint
   SIMPoint
   Temps
   Calendar
   ...
   ...
   XHQ

   Example 3: NavbarExplorerItems=SIMPoint|::Enterprise.SIMPoint~Alerts,\
   XHQ|http://www.siemens.com/mes,\

AnotherView|javascript:showView("::"+uservariable_path,"Alerts")

3. **Save** and **close**.
4. **Start** a new session of the XHQ Solution Viewer (browser client).

To display the NavBar tabs in the appropriate locale using component member descriptions

Component and component member names are not localized. However, the Description field (associated with each component member) is and will display in the given language (if a description is given).

1. Place the appropriate language file onto the web server at the site that is designed to use the given language.

   The procedure to do this can be found in the topic, *Placing the Language Files*.

2. Open the **`globalsettings.properties`** file in a text editor (such as Notepad).
   The `globalsettings.properties` file allows you to set solution-wide options. This file is stored at the location specified by the environment variable `%XHQ_SERVER_REPOS%` on the Enterprise Server (which is typically `C:\XHQ\data\repos`).
3. Find and edit, or add, the following property:

```
UseComponentMemberDescription=true
```

   The default setting for the `UseComponentMemberDescription` property is false (in which case, the non-localized component member name is displayed on the NavBar tabs).

4. **Save** and close the text editor.
5. Start a new session of the XHQ Solution Viewer.

Server Storage of Preferences

Use the `globalsettings.properties` file to set server storage of preferences. You must first add:

`DefaultUseServerPrefs=true`

to the `globalsettings.properties` file. Then, **save** the file and **restart the server**. If this property does not appear in the `globalsettings.properties` file, then the default of:

`DefaultUseServerPrefs=false`

is assumed, in which case the cookie is saved on the client-side.

You must have access to the `globalsettings.properties` file in order to do the following procedure. If you do not have access, please see your XHQ Administrator.

To set the server storage of preferences

1. Open **`globalsettings.properties`** file.
   The `globalsettings.properties` file allows you to set solution-wide options. This file is stored at the location specified by the environment variable `%XHQ_SERVER_REPOS%` on the Enterprise Server (which is typically `C:\XHQ\data\repos`).
2. Type **`DefaultUseServerPrefs=true`**.
3. Save **`globalsettings.properties`** and **restart** the server.

Using JavaScript to Get Client Time

There are three APIs that can be called from JavaScript. Each takes into account the following factors:

- The global time zone setting;
- The global time format;
- And whether the browser client is running in historical mode when returning the time.

The property for the global time zone setting is **`UseClientTimeZone`**, which has a default value of TRUE.

Although the `UseClientTimeZone` property is currently not listed on the `globalsettings.properties` file, it has the default value of TRUE. To change the value to FALSE, you will have to edit the `globalsettings.properties` file and set `UseClientTimeZone=false`.

The property for the global time format is **`DefaultTimeStampFormat`**. The default value for the timestamp format is **`MM/dd/yy HH:mm:ss a`**.

getClientTimeAsString

Returns a string that represents the current time (historical or server) using the proper time zone and formatting.

Syntax: `String getClientTimeAsString()`

getClientTimeAsLong

Returns a long that represents the current time (historical or server) using the proper time zone and formatting.

Syntax: `long getClientTimeAsLong()`

getClientTimeAsDate

Returns a date that represents the current time (historical or server) using the proper time zone and formatting .

Syntax: `Date getClientTimeAsDate()`

Setting Historical Navigation Values

The following historical navigation values can be set from the **`globalsettings.properties`** file.

For more information on historical navigation, go to the topic, *Historical Viewing and Navigation*, located in the XHQ Solution Viewer User's Guide.

Global Property	Description
`DefaultJogIncrement`	This is the default jog increment that appears in the historical navigation toolbar when it is first launched. *Default:* DefaultJogIncrement=10m
`DefaultTimeStampFormat`	The timestamp format for Tooltip, the Excel trend data output, and historical navigation. This must be a valid Java date/time format. **Note:** This property can be set to any value accepted by the SimpleDateFormat Java class with one exception: XHQ currently does not accept the default for SimpleDateFormat. **See Also:** For a list of valid Date/Time formats, see the Appendix. *Default:* DefaultTimeStampFormat=MM/dd/yy/ HH:mm:ss a
`HistoricalNavigationPanelLeft`	An integer value, this is the top-left X-position of the Historical Navigation toolbar when it is first launched. *Default:* HistoricalNavigationPanelLeft=250
`HistoricalNavigationPanelTop`	An integer value, this is the top-left Y-position of the Historical Navigation toolbar when it is first launched. *Default:* HistoricalNavigationPanelTop=0
`JogIncrementList`	This sets the increment values available from the Jog Increment Selector control of the historical navigation toolbar. *Default:* JogIncrementList=10s;10m;2h;5d;5w
`MinimumAllowedJogIncrement`	This is the smallest jog increment a user can enter in the Jog Increment Selector control. *Default:* MinimumAllowedJogIncrement=5s
`AggregateAverageList`	This sets the aggregate values available from the Aggregate Interval control of the historical navigation toolbar. *Default:* AggregateAverageList=30s;5m;2h;5d;5w
`TransitionSpeedList`	This sets the increment values available from the Transition Speed control of the historical navigation toolbar. *Default:* TransitionSpeedList=3s;5s;10s;15s;30s

Showing Tooltips

The following properties can be applied to Tooltips.

Global Property	Description
`DisableToolTipInView`	This allows you the option to turn off runtime Tooltips. If **true**, then Tooltips are **disabled**. If **false**, then Tooltips are **enabled**. *Default:* DisableToolTipInView=false
`ShowSourceForToolTip`	This controls whether or not the source of the value is displayed in the view Tooltip. *Default:* ShowSourceForToolTip=false
`ShowToolTipForValueMemberOnly`	If true, Tooltips are shown only for members marked as the Value member of the component. When set to true, the Tooltip is shown only for members of the tag component (the typical use case). *Default:* ShowToolTipForValueMemberOnly=false
`ShowValueForToolTip`	If false, which is the default, the value entry is hidden. If true, the value entry is shown in the Tooltip pop-up. When set to true, an additional line appears in the Tooltip, starting with "V:". *Default:* ShowValueForToolTip=false
`TagMappingTooltipFormat`	For the XY Chart, this customizes the Tag Mapping Tooltip that appears when a user hovers over a point at runtime. *Default:* TagMappingTooltipFormat="Domain: %1$s Time: %2$s Value: %3$.3f Desc: %4$s Range: %5$s Time: %6$s Value: %7$.3f Desc: %8$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)
`UseClassicTooltips`	This allows you to set the following Tooltip options. If **false** (default), then Tooltips are enabled only for the **data item**. If **true**, then Tooltips are also enabled for **embedded views** and **links**. *Default:* UseClassicTooltips=false

Solution Viewer Progress Indicator

Each time a view is being loaded, the Solution Viewer broadcasts the load progress through the status bar or a progress indicator box.

To view the indicator on the status bar, the browser status bar must be enabled.

The progress indicator box provides an additional visible indication of the view load progress. This indicator box launches after a given delay, which by default is set to 4 seconds. This timer delay can be used to avoid displaying progress indications unless the view load is taking longer than a deemed amount of time typical for the local client environment.

To edit the delay setting for the pop-up indicator box, set the value (which is in **milliseconds**) of the following global property.

Global Property	Description
`ProgressBarDelayTimer`	To launch the progress indicator box after a given delay in **milliseconds**. The default is 4000 milliseconds (or 4 seconds). *Default:* `ProgressBarDelayTimer = 4000` To launch the progress indicator box immediately upon the start of the view load, set this value to 0.

Working with the View Background Color

From the **`globalsettings.properties`** file, you can modify the view background color by specifying a valid RGB value. You can also determine whether or not to print the background color.

Global Property	Description
`PrintBackground`	If **true**, then the background color is printed with the view. If **false**, then the background color is not printed. *Default:* PrintBackground=false
`ViewBackgroundColor`	This is the view background color. Enter a valid RGB value. *Default:* ViewBackgroundColor=204,204,204

Browser Control

The following properties can be applied to the XHQ Browser Control.

For more information on the browser control (for example, the differences between default and non-default), see the topic, *Embedding Browser Content*, located in the XHQ Developer's Guide.

Global Property	Description
`BrowserControlPaintMethod`	Options: • native The browser control uses the native paint method. Web content is best rendered in this mode. This is the default value. • java The browser control uses the Java paint method. This provides the best mode for user interaction. • java_native This is a combination of both native and Java paint methods. *Default:* BrowserControlPaintMethod = java_native
`DefaultBrowserRenderingEngine`	Sets the system default rendering engine. *Example:* DefaultBrowserRenderingEngine = Modern Generation

View and ENO Group Instances

The following properties can be applied to both View and ENO Group instances.

For more information on **view instances**, see the topic, *Using the XHQ View Instance*, located in the XHQ Developer's Guide.

For more information on **ENO group instances**, see the topic, *Using an ENO Group as the Data Source*, located in the XHQ Developer's Guide.

Global Property	Description
`DisableExtendedNamespaceUI`	Set the value to false to display ENO in the solution navigation tree. *Default:* DisableExtendedNamespaceUI = true
`EnableInstanceItem`	Set the value to true to display the ENO Table and View Instance tools in the XHQ Workbench Tools Pallete. *Default:* EnableInstanceItem = false

Setting global properties for View Usage Statistics

Global Property	Description
`ViewStatsUpdateTimer`	This property sets the frequency (in milliseconds) in which the client-side updates the view access records before sending this collection of records to the XHQ Solution Server. The default is set to 1 hour. *Default:* ViewStatsUpdateTimer=3600000 **Note:** When the client is closed, the collection of view access records are "pushed" to the XHQ Solution Server before the client exits.
`LogViewAccessUserId`	By default, the user ID is collected and stored for the view access record. If **true**, then the user ID is collected. If **false**, then the user ID is not collected. *Default:* LogViewAccessUserId=true

For more information, go to the topic, *View Usage Statistics*, located in the XHQ Solution Viewer User's Guide.

Setting global properties for Date and Time Formats

The global properties to set date and time formats are:

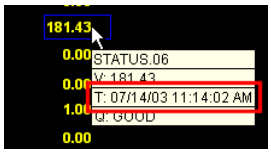
Global Property	Description
`DateFormatSymbolsLanguage`	This allows you to use a language other than the one specified by the regional setting when formatting dates. This requires a valid ISO-639 language code. If left blank, or an invalid language code is given, then the system's default regional setting is used.
`DefaultTimeStampFormat`	The timestamp format for value Tooltip, the Excel trend data output, the trend table, and historical navigation, in addition to the XHQ Trend Viewer "Start" and "End" times. This must be a valid Java date/time format. *Default:* DefaultTimeStampFormat=MM/dd/yy/ HH:mm:ss a **Important:** hh:mm:ss (with or without the "a" at the end) always displays the 12-hour format. **See Also:** For more information on Java date/time formats, refer to the topic on *Date and Time Formats*.
`DefaultDateOnlyFormat`	This sets the date format for the x-axis label of the Trend Chart and the Scooter time for the XHQ Trend Viewer (XHQ Interactive Trender). This must be a valid Java date/time format. *Default:* DefaultDateOnlyFormat=MM/dd/yy **Important:** This property is for the XHQ Trend Viewer (or XHQ Interactive Trender) only, and is not used for embedded trends. The date format for embedded trends can only be set from the XHQ Workbench. **See Also:** For more information on Java date/time formats, refer to the topic on *Date and Time Formats*.
`DefaultTimeOnlyFormat`	This sets the time format for the x-axis label of the Trend Chart and the Scooter time for the XHQ Trend Viewer (XHQ Interactive Trender). This must be a valid Java date/time format. *Default:* DefaultTimeOnlyFormat=hh:mm:ss a **Important:** hh:mm:ss (with or without the "a" at the end) always displays the 12-hour format. This property is for the XHQ Trend Viewer (or XHQ Interactive Trender) only, and is not used for embedded trends. The date format for embedded trends can only be set from the XHQ Workbench. **See Also:** For more information on Java date/time formats, refer to the topic on *Date and Time Formats*.
`DateFormatSymbolsLanguage`	This allows you to use a language other than the one specified by the regional setting when formatting dates.

Global Property	Description
	This requires a valid ISO-639 language code. If left blank, or an invalid language code is given, then the system's default regional setting is used.
`UseClientTimeZone`	If this property is not listed on your globalsettings.properties file, then the default value of "true" is assumed. To change the setting to "false", you must add this property to the globalsettings.properties file, set it to "false" and then save the file. *Default:* UseClientTimeZone=true When UseClientTimeZone=false, do not use the time zone component in the date/time format for the client. **WARNINGS:** The XHQ Platform does not support clearing (deselecting) the "Automatically adjust clock for daylight saving changes" checkbox due to an external issue with Java 1.1.x. To make sure that this checkbox is selected, open the **Control Panel**, click **Date and Time**. The "Date and Time Properties" dialog box appears. Click the **Time Zone** tab. If you set this property to true and then change the time zone on the machine, you will need to restart the XHQ system.

How the parameter DefaultTimeStampFormat is used

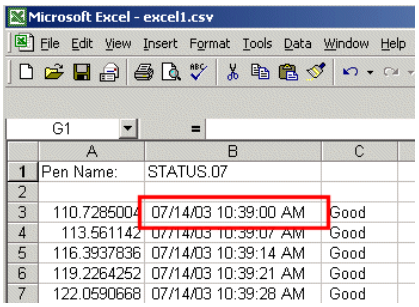
Use the `DefaultTimeStampFormat` to set the timestamp format for the following:

- The Tooltip for a value within a view.



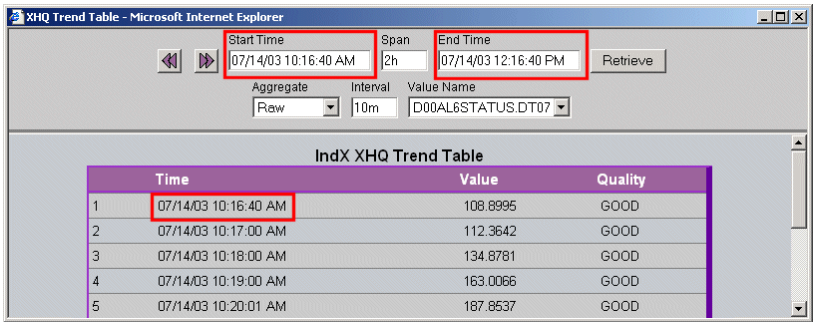
Tooltip Timestamp

- The Excel trend data output.



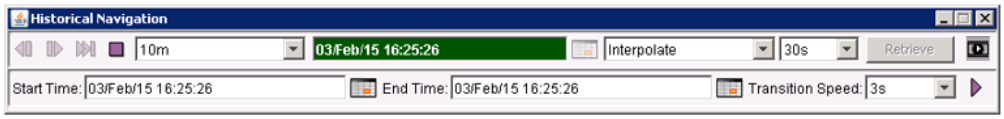
Excel data Timestamp

- The Trend Table "Start Time", "End Time", and "Time" column.



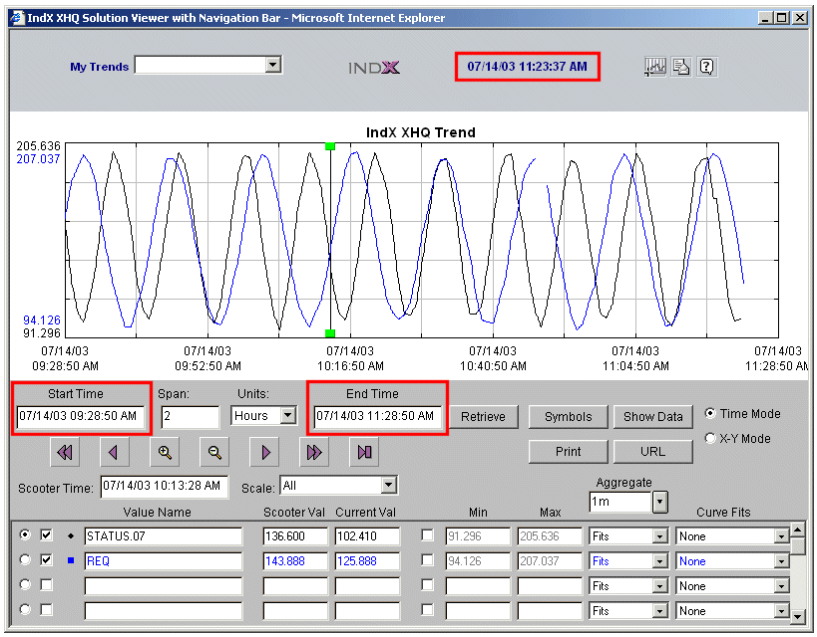
Trend Table Timestamps

- The Historical Navigation bar.



Historical Navigation Timestamp

- The XHQ Trend Viewer Date/Time label, the "Start Time", and the "End Time".

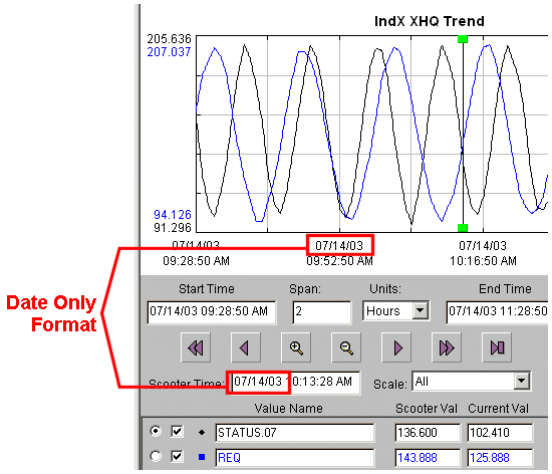


XHQ Trend Viewer

How the parameter DefaultDateOnlyFormat is used

The `DefaultDateOnlyFormat` property sets the date format for:

- The x-axis label of the Trend Chart for the XHQ Trend Viewer.
- The date portion of the "Scooter Time" for the XHQ Trend Viewer.

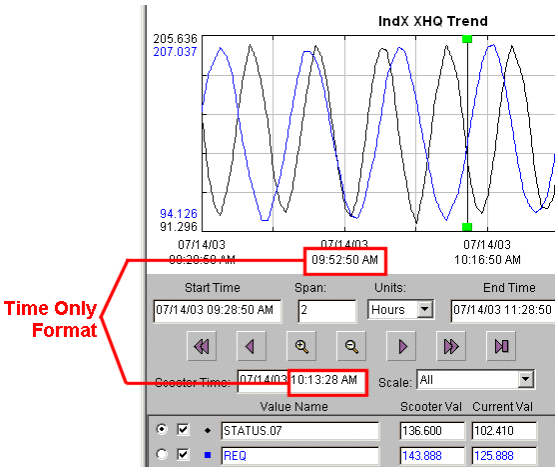


This property is for the XHQ Trend Viewer (or XHQ Interactive Trender) only, and is not used for embedded trends. The date format for embedded trends can only be set from the XHQ Workbench.

How the parameter DefaultTimeOnlyFormat is used

The `DefaultTimeOnlyFormat` property sets the time format for:

- The x-axis label of the Trend Chart for the XHQ Trend Viewer.
- The time portion of the "Scooter Time" for the XHQ Trend Viewer.



This property is for the XHQ Trend Viewer (or XHQ Interactive Trender) only, and is not used for embedded trends. The date format for embedded trends can only be set from the XHQ Workbench.

24-hour versus 12-hour Format

For the global properties, the **24-hour** format is: **HH:mm:ss a**, where "a" is optional. In the 24-hr format, using the capital "H" is required. If both "HH" and "a" are specified, then the time property returns in 24-hour format with either an "am" or a "pm" after it.

The **12-hour** format is: **hh:mm:ss a**.

To display the 12-hour format, use the lower case "h".

The format hh:mm:ss with or without the "a" at the end, always displays the 12-hour format.

Enabling a Historian – History Tab

These properties enable you to select the given historian from the Recorder drop-down list of the XHQ Solution Builder **History tab**.

Recorder Drop-down List

In addition to setting this property, you must also add the **`-allowStore`** parameter to the command-line for the given connector process.

For more information, go to the topic, "How to edit the Command-line," located in the given connector's section of the XHQ Connection Guide.

Also, see the topic, *Using the Command-Line to Export/Import Component Roles*, located in the XHQ Developer's Guide.

Global Property	Description
`IP21HistoryTabEnabled`	If **false** (which is the default), the IP.21 historian is not available for selection, write-back is disabled. If **true**, the IP.21 historian is available from the Recorder drop-down list of the History tab. *Default:* IP21HistoryTabEnabled = false
`MindSphereHistoryTabEnabled`	If **false** (which is the default), the MindSphere IoT historian is not available for selection, write-back is disabled. If **true**, the MindSphere IoT historian is available from the Recorder drop-down list of the History tab. *Default:* MindSphereHistoryTabEnabled = false
`OPCHistoryTabEnabled`	If **false** (which is the default), the OPC historian is not available for selection. If **true**, the OPC historian is available from the Recorder drop-down list of the History tab. *Default:* OPCHistoryTabEnabled = false **Important:** Currently, write-back support is for OPC HDA only.
`PHDHistoryTabEnabled` `PHDVHistoryTabEnabled`	If **false** (which is the default), the PHD/PHDV historian is not available for selection, write-back is disabled. If **true**, the PHD/PHDV historian is available from the Recorder drop-down list of the History tab. *Defaults:* PHDHistoryTabEnabled = false PHDVHistoryTabEnabled = false
`PIHistoryTabEnabled`	If **false** (which is the default), the PI/PISDK historian is not available

Global Property	Description
	for selection, write-back is disabled. If **true**, the PI/PISDK historian is available from the Recorder drop-down list of the History tab. *Default:* PIHistoryTabEnabled = false

Configuring Caches

Global Property	Description
`IObjectCacheSize`	This cache contains information about component instances at runtime. *Default:* IObjectCacheSize =1000 (Which translates to 1000 instances of IObject objects.)
`ObjectInfoCacheSize`	This cache contains information about component metadata at runtime. *Default:* ObjectInfoCacheSize =10000 (Which translates to 10000 instances of ObjectInfo objects.)
`ObjectIDCacheSize`	This cache contains information about component IDs at runtime. *Default:* ObjectIDCacheSize =10000
`ViewTableIDCacheSize`	This cache contains information about view metadata. *Default:* ViewTableIDCacheSize =10000
`ViewCacheMegs`	This is the amount of memory for IObjects. *Default:* ViewCacheMegs =5 (in megabytes) **Note:** If ViewCacheMegs=0, then there is no view cache.
`ViewCacheLocation`	There are two methods to set the view cache location. **Method 1:** Specify the full path of the base location. Two backslashes are required after the drive letter. Note the format in the following example. *Example 1:* ViewCacheLocation=C:\\XHQViewCache\\<user_name> Where <user_name> is the current login user name during runtime. **Method 2:** Set the location to <user_profile>, then at runtime, the path of <user_profile> is retrieved and is used as the view cache directory. *Example 2:* ViewCacheLocation=<user_profile> *Default (which is blank):* ViewCacheLocation= The default is ViewCacheLocation= [blank]. This property file uses the Java property `user.home` to determine the user's profile location (which is C:\Documents and Settings\<userid>\). The cache is stored in the default directory, `%APPDATA%\Indx`. **Important:** You must have write access to the directory you specify.

View Load Performance

These property values should only be changed on the advice of the XHQ Customer Support Team.

Global Property	Description
`AggregatesBulkTimer`	This is the timer value (in milliseconds) for the aggregate bulking thread. *Default:* AggregatesBulkTimer=100 To disable this property, set the value to 0.
`RollupsBulkTimer`	This is the timer value (in milliseconds) for the rollups bulking thread. *Default:* RollupsBulkTimer=100 To disable this property, set the value to 0.
`CollectionsBulkTimer`	This is the timer value (in milliseconds) for the collection query bulking thread. *Default:* CollectionsBulkTimer=100 To disable this property, set the value to 0.
`TrendsBulkTimer`	This is the timer value (in milliseconds) for the trends bulking thread. *Default:* TrendsBulkTimer=150 To disable this property, set the value to 0.
`CancelBulkRequests`	If **true** (the default), this enables the cancellation of bulk requests that are in progress for a view when the user clicks on a different view. If **false**, this disables the cancellation of bulk requests that are in progress for a view when the user clicks on a different view. *Default:* CancelBulkRequests=true

Redundancy

The following properties can be applied to a redundant solution.

Global Property	Description
`ServerConnectionMonitorTimer`	This sets the number of **milliseconds** when the Connection Controller checks the connection status of each of the four Connection Monitors. *Default:* ServerConnectionMonitorTimer = 30000
`RedundantMessageDisplayTimer`	When a switch in server occurs, a message box appears. This property sets the duration (in **milliseconds**) this message is displayed. *Default:* RedundantMessageDisplayTimer = 5000
`RedundantSwitchBackTimer`	This sets a delay (in **milliseconds**) when switching back from the backup solution server to the primary solution server. This delay allows the primary solution server to update and/or validate the cache server content. *Default:* RedundantSwitchBackTimer = 300000
`EnableConnectionMonitorDebug`	If **true**, the Connection Monitor produces debug messages in a given log file. Else, the default is **false**. *Default:* EnableConnectionMonitorDebug = false

For Aggregates

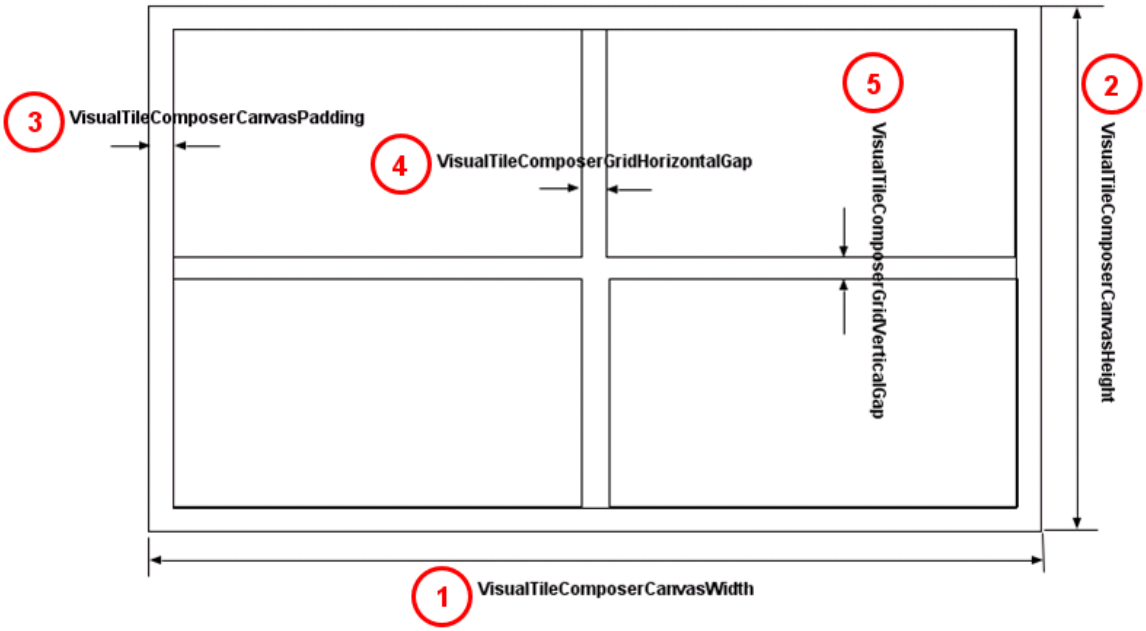
The following properties can be applied to aggregates.

Global Property	Description
`AggregateAverageList`	This sets the aggregation interval values available from the XHQ Trend Viewer. (s=seconds, m=minutes, h=hours, d=days, w=weeks) *Default:* AggregateAverageList = 30s;5m;2h;5d;5w
`AggregatesThrottleTimer`	This is the timer value (in seconds) for throttling aggregates calls. *Default:* AggregatesThrottleTimer = 300
`DefaultAggregateAverage`	This property is applied when the aggregate average for a time-sliced pen in a trend is invalid. *Default:* DefaultAggregateAverage = 30m Consider the following situations. **Case 1:** In an Interactive Trender, a pen's aggregate type is changed from FITS to INTERPOLATE. In this case, the aggregate average is set to the `DefaultAggregateAverage` value. **Case 2:** In both the Embedded Trend and Interactive Trender, a pen has multiple references and the aggregate type is FITS or RAW. This is an invalid state. In this case, the aggregate type is forced to INTERPOLATE at runtime, and the `DefaultAggregateAverage` value is used.

Visual Tile Composer Properties

Grid Dimensions

Set the following properties to edit dimensions for the VTC grid and canvas components.



Visual Tile Composer Grid Dimensions

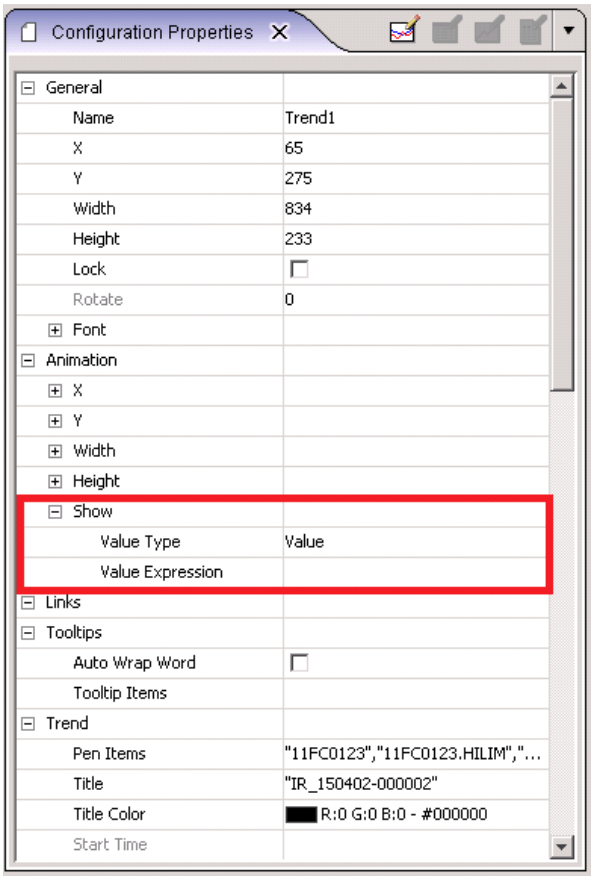
Image Ref. No.	Global Property	Default Value (pixels)
1	`VisualTileComposerCanvasWidth`	1080
2	`VisualTileComposerCanvasHeight`	616
3	`VisualTileComposerCanvasPadding`	10
4	`VisualTileComposerGridHorizontalGap`	4
5	`VisualTileComposerGridVerticalGap`	4
	`VisualTileComposerGridColumns`	10
	`VisualTileComposerGridRows`	10

Reference numbers map to the image below. The default values for these properties are in pixels (with the exception of the columns and rows values).

View Load Performance

Set the property `EnableLazyViewLoad` to **`true`** to improve view loading. In this case, the embedded view is only loaded when the "Show" animation option (View Configuration Properties) value is also set to true.

This property applies to non-VTC views as well.



View Configuration Properties: Show Animation Option

Likewise, for views in a group item, the views are only loaded when the "Show" animation option value is set to true on the group item. And the template view column in the table control is only loaded when the "Show" animation is set to true on the table control.

Set `EnableLazyViewLoad` to **`false`** to disable this feature. The default value to this property is `true`.

Additional Global Settings

The following table lists additional settings you can set from the **`globalsettings.properties`** file.

Global Property	Description
`AllowComboBoxMultipleSelection`	If true, then all combo boxes in every view have multiple selection enabled. *Default:* AllowComboBoxMultipleSelection=false For more information, go to the topic, *Multiple Selection Combo Box* located in the XHQ Developer's Guide.
`ConvertCRtoLF`	When set to **true** (the default), all carriage returns, <CR>, are converted to <LF> in GUI text areas.
`DatabaseHistorianEnabled`	When set to **false**, the XHQ Data Recorder option (from the History tab of the XHQ Solution Builder) is hidden and cannot be selected. *Default:* DatabaseHistorianEnabled =true **Important:** Setting this global property is one of two steps in disabling the Data Recorder. For more information, go to the topic, *Disabling the Data Recorder*.

`DefaultTrendDecimalDigits`	This allows the administrator to configure a default number of decimal digits to display when the number of decimal digits is not explicitly set in XHQ.
	This property is used to set the format of default-formatted values:
	• Displayed in the Trend Table;
	• Exported to an Excel file via the Show Data > Excel function;
	• Displayed in the Legend Panel of the XHQ Interactive Trender, which includes:
	  ◦ Current value field
	  ◦ Scooter value field(s)
	  ◦ Multi-Scooter Aggregate values (avg, min, max, and so forth).
	Example:
	DefaultTrendDecimalDigits=n
	Where (n) is the default number of decimal digits to display, ranging from 0 to 8.
	Additional things to note:
	• This property only pertains to pen configurations using the General scale format and is overridden by the Number-Scientific and Number-Fixed formats.

Global Property	Description
	• If this property setting is absent, the shipping default is set to 2 decimal digits. • If the parameter is set to zero (0), then the defaulted values are displayed as an integer with no decimal point. • Values less than zero default to zero, and values greater than 8 default to 8.
`DefaultUseServerPrefs`	Set this property to true if you want to save user preferences on a server instead of the client. If this property does not appear in the globalsettings.properties file, then the following default is assumed (in which case the cookie is saved on the client side): *Default:* DefaultUseServerPrefs=false To change the default setting, edit the globalsettings.properties file by adding this property and setting it to **true**. Then, **save** the file and **restart** the server.
`DisableEmbeddedTrendDoubleBuffering`	Set this property to true only if you are using a Barco display screen to view an embedded trend or if you are experiencing an "Out of Memory" error when launching an embedded trend.
`DisableExtendedNamespaceUI`	If **false**, it enables the visualization of rootless (un-modeled) objects in the XHQ Solution Builder. If **true** (which is the default), this feature is disabled.
`DisableRoleSecurityGroupsTable`	If **true**, then the Role Security Groups table is **disabled**. When selecting a Role Security group, a text box appears, allowing you to enter a valid group. If **false**, then the Role Security Groups table is **enabled**. When selecting a Role Security group, a selection table appears. *Default:* DisableRoleSecurityGroupsTable=true
`DisableWriteConfigUI`	This controls the display of any write-oriented configuration user interfaces such as the new options on the General tab for primitives or the Write column in the Access tab. *Default:* DisableWriteConfigUI=true
`EnableAntiAlias`	Anti-aliasing makes graphics and text easier to read by shading the pixels along the borders of the graphical elements. To disable the anti-alias option, set EnableAntiAlias=false. *Default:* EnableAntiAlias=true
`EnableClientVariableQuality`	This enables the client variable to have quality. By default, the quality of client variable is always good. *Default:* EnableClientVariableQuality=false
`EnableLazyViewLoad`	Set to true to improve view loading. In this case, the

Global Property	Description
	embedded view is only loaded when the "Show" animation option (View Configuration Properties) value is also set to true. Likewise, for views in a group item, the views are only loaded when the "Show" animation option value is set to true on the group item. And the template view column in the table control is only loaded when the "Show" animation is set to true on the table control. Set to false to disable this feature. *Default:* EnableLazyViewLoad=true
`EnableLegacyXAnimationLogic`	If set to true, reverts back to legacy logic of handling a value item x-coordinate animation, which addresses justification of data. *Default:* EnableLegacyXAnimationLogic=false
`EncryptDataChannel`	If set to true, then the data sent through the data channels between the client (which can be the XHQ Solution Builder or the XHQ Solution Viewer) and the XHQ Solution Server is encrypted. *Default:* EncryptDataChannel=true
`FilterCaseSensitive`	Sets the case sensitivity for the collection query filter (ad-hoc criteria filter) combo or edit boxes. *Default:* FilterCaseSensitive=true To make the input to these controls **case-insensitive**, set this to **false**.
`IEBrowserFont`	Displays Kanji on a non-Japanese machine. Set IEBrowserFont to either MS Gothic of MS Mincho. (Make sure that either font is available under Windows.) *Example:* IEBrowserFont=MS Gothic *Default (which is **blank**):* IEBrowserFont=
`LoadTreeNodesOnExpand`	If set to **true**, the children of a node are loaded only when that node is expanded. *Default:* LoadTreeNodesOnExpand=false
`LogViewAccessUserId`	This allows you to hide or display the username from the View Statistics utility. If **true**, then the username is **displayed**. If **false**, then the username is **hidden**. *Default:* LogViewAccessUserId=false
`LogViewStatistics`	Set this property to **true** to automatically log view statistics to the browser client log file when the user changes views. *Default:* LogViewStatistics=false

Global Property	Description
`MailLinkExpirationDays`	This property sets the expiration (in the number of days) for the e-mailed URL link for Trends and Charts. Once expired, the Trend/Chart configuration file is deleted. When you try to access the link after the expiration date, a blank trend/chart appears. *Default:* MailLinkExpirationDays=30
`MaxCollectionCells`	This property sets the maximum number of flow views allowed for a collection display. *Default:* MaxCollectionCells=80
`MaxFireCriteriaMonitorTimer`	Configures the collection criteria monitor timer. Each collection has a criteria monitor timer that monitors the quality of all criteria items. By default, it is set to check the quality every one second. The timer stops when the quality of all criteria items are GOOD. If the quality of some criteria items are always BAD (for example, the default value is not configured), then the timer keeps checking every one (1) second. This can negatively impact performance. *Default:* MaxFireCriteriaMonitorTimer=3
`MaxTrendFreeTime`	The maximum space allotted for adding new data points to an automatically updated trend chart (in seconds). This sets the maximum blank space at the right side of a trend, and has the effect of making long duration trends contain more data and less empty space for new incoming points. By setting the time that new points will be added to an existing plot before jogging forward and redrawing the entire trend, it also determines the interval at which the trend will be completely refreshed. *Default:* MaxTrendFreeTime=600
`MaxUndoStackSize`	From the XHQ Workbench, this property sets the maximum number of actions that you can undo. By default, you can undo up to 25 actions. To change this limit, set this property to a value between 1 and 100. *Default:* MaxUndoStackSize=25
`ReconnectAttemptsCount`	The maximum number of reconnection attempts. The default is 5. *Default:* ReconnectAttemptsCount=5
`ReconnectAttemptsInterval`	The delay (in milliseconds) between reconnection attempts. The default value is 10000. *Default:* ReconnectAttemptsInterval=10000
`RequestMaxArrayLength`	This property sets the maximum **number of elements** in the

Global Property	Description
	data array that is sent to the server. *Default:* RequestMaxArrayLength=10000
`RequestMaxDataSize`	This property sets the maximum size (in **bytes**) to request from the data array. *Default:* RequestMaxDataSize=1600000
`ShowDistributeModelMenu`	If TRUE, a menu item appears in the XHQ Workbench for the Model Administration on the Model Manager server. If FALSE (which is the default), then the XHQ Workbench menu item to distribute the model is hidden. *Default:* ShowDistributeModelMenu=false
`ShowLastRowWhenPaging`	If TRUE (which is the default), in collection paging, the last row in the previous page is shown as the first row in the next page. If FALSE, the last row in the previous page is not shown as the first row in the next page. *Default:* ShowLastRowWhenPaging=true
`ShowRedXForInvalidKeyedView`	Setting this property to FALSE hides the red "X" that appears for a keyed view when the tag is invalid. *Default:* ShowRedXForInvalidKeyedView=true
`SolutionHostForCache`	For example, this is useful for terminal server environments. This forces the use of the given solution host name. *Default (which is **blank**):* SolutionHostForCache=
`UseBlueBoxForExcelControl`	If TRUE, a rectangular blue outline appears when you mouse-over the Excel control item. In addition, the cursor changes from the hand to the select cursor. By default, this property is set to FALSE. *Default:* UseBlueBoxForExcelControl=false
`UseComponentMemberDescription`	If you are using the XHQ NavBar, you can also set the labels of the NavBar tabs to display in the appropriate language by setting this global property to TRUE. By setting this property to TRUE, the component member description is displayed instead of the component member name. **Note:** Component and component member names are not localized. However, the Description field (associated with each component member) is and will display in the given language (if a description is given). *Default:* UseComponentMemberDescription=false When set to the default value of FALSE, the non-localized component member name is displayed on the NavBar tabs. **For XHQ Performance Management**

Global Property	Description
	When set to TRUE, the following occurs in the "Performance Indicator Browser" page: • The Anchor filter is disabled. However, you can still click the Browse button to a select a path to use as a filter, which launches the "Anchor Selector" dialog. • If the Metric component is a path, then the Metric is displayed by description (if a description is set) or by name (if the description is not set). • Anchor components are displayed by description (if a description is set) or by name (if the description is not set). And the following occurs in the "Anchor Selector" dialog: • The "Search" field is disabled; you can only select a path to use as a filter. • The Path tree only shows components, not primitives or tags. • Components are displayed by description (if description is set) or by name (if description is not set). And the following occurs in the "Performance Indicator Details" page: • The Anchor path picker fields are disabled. To edit the path, click the Browser button. The "Value Selector" dialog appears. • The "Search" field is disabled; you can only select a path to use as a filter. • The Path tree only shows components, not primitives or tags. • Components are displayed by description (if description is set) or by name (if description is not set).
`UseFontAlias`	This property maps one of the three font names (Courier, Helvetica, TimesRoman) you assign during design-time (XHQ Workbench) to a general font name that displays during runtime (Solution Viewer, browser client). *Example:* `UseFontAlias = Courier|Monospaced;Helvetica|SansSerif; TimesRoman|Serif` The value for this property is a list of font pairs separated by a semi-colon (;). The first font in the pair is one of the three given font names available during design-time: Courier, Helvetica, or TimesRoman. This is then followed by a pipe or vertical bar (|), which is then followed by the second font name to which the first font will be mapped.

Global Property	Description
	Important: You must enter font names exactly as they are given in the system. For example, because the font "Times New Roman" appears in the XHQ Workbench as "TimesRoman", you must use the latter "TimesRoman". Also, it is recommended that the font name you map to is generally available to all clients (such as Monospaced, SansSerif, or Serif). Should you use a font name that the system does not recognize, a default font will be assigned. Note, font styles are case insensitive, and spaces added before or after a font name are ignored. The default value for this property is <empty>. *Default (which is blank):* UseFontAlias =
`UseLog10BaseFormat`	To set the Trend Viewer Scale Type to log10-based format, set this property to true.
`WebServer`	The web server name. This property must be defined in order to launch the Trender from the Solution Builder. *Default:* WebServer=localhost
`ZoomValues`	Sets the zoom selection values for the right-click shortcut menu of the XHQ Solution Viewer. Use the decimal equivalent of the percentage, for example, use 0.5 for 50%. Use a semi-colon to separate the values. *Default:* ZoomValues=0.5;0.75;1.0;1.5;2.0;2.5;3.0;3.5;4.0

Using globaleditingsettings.properties

This file allows you to edit the preferences, or settings, that define the graphical attributes of various XHQ Workbench features (such as the color palette, text size, fill and line color, and so forth). Consider, for example, the color of a BAD value item. By default (when you first launch the XHQ Workbench after a brand new XHQ install), the color for a BAD value item is red (the RGB equivalent is 255,0,0). Such attributes are pre-determined and are referred to as the *XHQ defaults*.

What if the content views you were developing required a different BAD value color? Well, you can customize these settings, according to your needs:

- By editing the **`clientsettings.properties`** file, which is typically located in the `Documents and Settings\<user>\Application Data\XHQ\XHQ devClient\repos` folder on the user's local machine (referred to as the *User defaults*).

or

- By editing the **`globaleditingsettings.properties`** file, which is typically located in the <install directory>:`\XHQ\data\repos` folder on the server (referred to as the *Solution defaults*).

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

`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).

Editing the **`globaleditingsettings.properties`** file allows you to set solution-wide XHQ Workbench preferences. The following table lists the XHQ Workbench settings you can edit.

Default Property Settings

Feature	Property Defaults
For Value Items	ValueItem_textColor=0,0,0
	ValueItem_fillColor=255,255,255
	ValueItem_badColor=255,0,0
	ValueItem_uncertainColor=255,255,0
	ValueItem_fontName=Helvetica
	ValueItem_fontSize=12
	ValueItem_fontStyle=1*
	ValueItem_fontJustify=0*
	ValueItem_fill=false
	ValueItem_decimalPlace=0
	ValueItem_valueFormat=0*
For Text Items	TextItem_fontName=Helvetica
	TextItem_fontSize=12

Feature	Property Defaults
	TextItem_fontStyle=1*
	TextItem_fontJustify=0*
	TextItem_textColor=0,0,0
	TextItem_fillColor=255,255,255
	TextItem_fill=false
For Line Items	LineItem_lineWidth=3*
	LineItem_lineColor=0,0,0
	LineItem_lineStyle=0*
	LineItem_lineArrowhead=0*
	LineItem_lineArrowheadStyle=0*
For Polyline Items	PolylineItem_lineWidth=1*
	PolylineItem_lineColor=0,0,0
	PolylineItem_fillColor=255,255,255
	PolylineItem_fill=false
	PolylineItem_lineStyle=0*
	PolylineItem_lineArrowhead=0*
	PolylineItem_lineArrowheadStyle=0*
For Rectangles	RectangleItem_lineColor=0,0,0
	RectangleItem_fillColor=255,255,255
	RectangleItem_fill=true
	RectangleItem_lineWidth=1*
For Ellipses	EllipseItem_fill=true
	EllipseItem_fillColor=255,255,255
	EllipseItem_lineColor=0,0,0
	EllipseItem_lineWidth=1*
For Edit Boxes	EditBoxItem_fontName=Helvetica
	EditBoxItem_fontSize=12
	EditBoxItem_fontStyle=1*
	EditBoxItem_fontJustify=0*
	EditBoxItem_textColor=0,0,0
	EditBoxItem_fillColor=255,255,255
	EditBoxItem_fill=true
For Combo Boxes	ComboBoxItem_fontName=Helvetica
	ComboBoxItem_fontSize=12
	ComboBoxItem_fontStyle=1*
	ComboBoxItem_fontJustify=0*

Feature	Property Defaults
	ComboBoxItem_textColor=0,0,0
	ComboBoxItem_fillColor=255,255,255
	ComboBoxItem_fill=true
For Buttons	ButtonItem_fontName=Helvetica
	ButtonItem_fontSize=12
	ButtonItem_fontStyle=1*
	ButtonItem_fontJustify=0*
	ButtonItem_textColor=0,0,0
	ButtonItem_fillColor=192,192,192
	ButtonItem_fill=true
For Text Areas	TextAreaItem_fontName=Helvetica
	TextAreaItem_fontSize=12
	TextAreaItem_fontStyle=1*
	TextAreaItem_fontJustify=0*
	TextAreaItem_textColor=0,0,0
	TextAreaItem_fillColor=255,255,255
	TextAreaItem_VScrollBar=false
	TextAreaItem_fill=false
	TextAreaItem_badColor=255,0,0
	TextAreaItem_uncertainColor=255,255,0
For Trends	TrendItem_fillColor=255,255,255
	TrendItem_fill=true
	TrendItem_fontName=Helvetica
	TrendItem_fontSize=12
For Solution-defined Colors	Solution_defined_color_0=-1,-1,-1
	Solution_defined_color_1=-1,-1,-1
	Solution_defined_color_2=-1,-1,-1
	Solution_defined_color_3=-1,-1,-1
	Solution_defined_color_4=-1,-1,-1
	Solution_defined_color_5=-1,-1,-1
	Solution_defined_color_6=-1,-1,-1
	Solution_defined_color_7=-1,-1,-1
	Solution_defined_color_8=-1,-1,-1
	Solution_defined_color_9=-1,-1,-1
	Solution_defined_color_10=-1,-1,-1
	Solution_defined_color_11=-1,-1,-1

Feature	Property Defaults
	Solution_defined_color_12=-1,-1,-1 Solution_defined_color_13=-1,-1,-1 Solution_defined_color_14=-1,-1,-1 Solution_defined_color_15=-1,-1,-1
Lock Indicator Color	Lock_Indicator_Color=64,64,64
Point Handles Color	Sub_Selection_Handle_Color=200,200,200
Font Selection	ViewBuilder_fonts=<fontname-1>,<fontname-2>, ... ,<fontname-n> The value is a comma-separated list of font names. These fonts appear in the XHQ Workbench font drop-down list after the three default fonts and in the order they are listed. *Example*: ViewBuilder_fonts=Arial,Tahoma,Courier New,Times New Roman
**See Attribute Key*	

Attribute Key

Attribute	Options
fontStyle	0 – plain 1 – bold 2 – italic 3 – bold, italic
fontJustify	0 – left align 1 – center align 2 – right align
valueFormat	0 – general format 1 – number scientific format 2 – number fixed format
lineStyle	0 – solid 2 – dotted 3 – dashed
lineArrowhead	0 – no arrowhead 1 – arrowhead at origin 2 – arrowhead at end 3 – arrowheads at both ends
LineArrowheadStyle	0 – open arrow 1 – solid arrow

Attribute	Options
lineWidth	number of pixels

Using solutionsettings.properties

The **`solutionsettings.properties`** file allows you to set the debug log level on the solution server, the interval for memory tests for Java garbage collection, and other tunable parameters that deal with solution server performance. This file is stored at the location specified by the environment variable **`%XHQ_SERVER_REPOS%`** on the Enterprise Server (which is typically **`C:\XHQ\data\repos`**).

This file is not included with the XHQ install and must be created by the user. Changes to these parameters **require server restart**.

The following is a list of configurable properties and their default values.

- **`net.indx.util.syslog.sysloglevel=0`**
  This property sets the debug logging level on the solution server.
- **`net.indx.util.syslog.maxlogsizemb=5`**
  This property sets the maximum size of the log file in megabytes.
- **`net.indx.util.syslog.logextension=.out`**
  This is the extension to the full log filename.
- **`net.indx.util.syslog.logfilename=solutionlog`**
  This is the base to the full log filename. It is followed by the number 1 or 2 (for example, `solutionlog1.out`). This log file is located in the `%XHQ_SERVER_LOGS%` directory (which is typically under `C:\XHQ\log`). Once the first log file reaches the specified maximum size, a second log file is created (in this case, `solutionlog2.out`), and the logging is switched to it.
- **`net.indx.repository.solution.server.collectioncontroller.allowautoreconcile=true`**
  This enables automatic tag reconciliation on start-up.
- **`net.indx.repository.solution.server.collectioncontroller.statsupdatethreshold=1`**
  This property sets restrictions on the Solution Server when collecting database statistics on back-end collections that have high update frequencies. The value of this property is the collection update frequency in minutes and may be an integer value from 1 to 10. The default frequency is 1 minute, as given in the above example.

  You may adjust this property during runtime. Simply edit the `solutionsettings.properties` file and save it. Then, open the XHQ Shell and run `xhqci servertool updatelogging` to re-read the properties file.
- **`net.indx.repository.solution.server.rollupcache.rollupcachesize=2000`**
  The rollup cache keeps processed rollup queries in memory for performance optimization.
- **`net.indx.repository.solution.server.datamanager.suppresstagcollectionquality=false`**
  This controls the display of the tag collection quality. A value of "false" (which is the default) indicates no quality suppression.

  When the solution server starts and the setting is "true", the following message is written to the log file:

  ```
  INFO: Suppressing tag collection quality changes according to configuration setting
  ```

  When quality has been suppressed, a similar message like the following appears:

  ```
  WARNING: ::Allocates collection has expired but quality has not been changed to reflect this.
  ```

  The reason for this message should be checked in case there is a problem with access to the network or backend system(s) which drive the tag collection.

  There is also an xhqci command to change the setting at runtime, which does not require a server restart. For more information, go to the topic, *Managing Data Quality*.

- **`net.indx.repository.solution.server.datamanager.totallistmax=0`**
  This property controls the total number of elements that will be kept in memory. This number represents all the nodes in the solution tree including components, primitive members, and collections. Setting this value to the default of "0" makes the most efficient memory configuration.
- **`net.indx.repository.solution.server.datamanager.expirationcoefficient=1.0`**
  This property provides the ability to tune expiration intervals. The minimum as well as the default value for this parameter is 1.0. To increase the current expiration interval by 25%, set the value to 1.25; to double it, set it to 2.0.
- **`net.indx.repository.solution.server.datamanager.allowdelayedactivation=true`**
  This allows collections to delay normal, initial activation if the persisted data has not expired. If this property is true (which is the default), or if it is not specified in the `solutionsettings.properties` file, the connector uses the last poll and expiration date to determine whether activation and/or scheduling can be delayed and how that delay is handled. If set to false, collection activations are never delayed at system start-up, regardless of the expiration state (which is the behavior in XHQ versions prior to 4.0). This applies to polled connector collections and does not apply to either XHQ Cache Connector collections or Pass-through collections. Currently, only JDBC-based connectors (Oracle, ODBC, SQL, PostgreSQL) support this feature.

  In a normal activation, during system or connector start-up, the corresponding collection starts its update regardless of whether is has expired or not. A delayed activation allows for system load optimization. If a connector started, and there is a collection configured for that given connector, then the system calculates the next update time based on the expiration time of the collection.

- **`net.indx.repository.solution.server.datamanager.clientsortfirst=true`**

  Priority is now given to the sorting order done by the XHQ Solution Viewer over the sorting configurations in the XHQ Workbench and XHQ Solution Builder. To revert back to the *classic sorting behavior*, set this property to *false*.
- **`net.indx.repository.solution.server.datamanager.logspecialupdates=true`**
  This enables the logging of special writes.
- **`net.indx.repository.solution.server.datamanager.unsubscribedelay=1800000`**
  This value determines how long the point stays active after all clients have unsubscribed from it.

  Units: milliseconds (ms)

  Default: 1800000 (which is 30 minutes)

  When deactivated, the point is removed from its connector polling list and its object in the XHQ Solution Server is erased.

  The activation process may take some time, especially during heavy load. Once a point is active, its current value is readily available. Delaying unsubcription increases memory use in the XHQ Solution Server and the connector. It also adds load to the backend historian. However, it improves the user experience when clients open a view. The data appears much faster in the view.
- **`net.indx.repository.solution.server.solutionserver.numprocessthreads=10`**
  Recommended only for application engineers, this property controls the number of threads used to process most requests to the server and is typically used for performance tuning.

  This value should be greater than the expected number of backend collections/connection processes handling collections.

- **`net.indx.repository.solution.server.solutionserver.allowapiclients=true`**
  This allows the web services to access the solution.
- **`net.indx.repository.solution.server.solutionserver.collectionqueuesize=5`**
  Recommended only for application engineers, this property may be used to control collection memory utilization on the solution server during collection updates.
- **`net.indx.repository.solution.server.solutionserver.streamingChunkTimeout=30000`**

Adjusts the default timeout for the XHQ connector. This is to prevent having a timeout on the target server. Specifically, when a `-DuseStream` switch is used, this prevents the source server from waiting too long on a request.

Units: milliseconds (ms)

Min: 2000, Max: 60000

Default: 30000

- **`net.indx.repository.solution.server.solutionstorage.dataconfigsclustersize=1024`**
  Recommended only for application engineers, this property is read during system start-up and is used to enhance performance.
- **`net.indx.repository.solution.server.writequeue.insertbuffermb=2`**
  The buffer size for the collection queue (in MB). The maximum is 4; the minimum is 1.

  Constraint: 1 <= insertbuffermb <= 4(2*default)
- **`net.indx.repository.solution.server.writequeue.memorylimitmb=768`**
  The total amount of memory allowed in the collection queue (in MB). The maximum is 1280. The minimum is the product of net.indx.repository.solution.server.solutionserver.collectionqueuesize * net.indx.repository.solution.server.writequeue.insertbuffermb. If memorylimitmb + insertbuffermb is greater than the maximum value, then no collection is allowed to insert data until the sum of these values fall below the max value.

  Constraints:
  - MIN <= memorylimitmb <= 1280
    where MIN is defined as solutionserver.collectionqueuesize * writequeue.insertbuffermb
  - memorylimitmb >= 50*insertbuffermb.
- **`net.indx.repository.solution.server.activationthread.unsubscribedelay=5000`** (or 5 seconds)
  This delays the unsubscription of a point from its connector. A deactivation delay optimizes performance as well as backend data source load. Each activation requests new data from thebackend immediately. Setting the value to "0" results in no delay in retrieval, increasing the load on the data source. There is no maximum limit.
- **`net.indx.repository.solution.server.transactionlog.enablelog=true`**
  When set to true, this enables transaction logging; when set to false, logging is disabled.
- **`net.indx.util.gcinvoker.gcinterval=10000`**
  The interval (in ms) at which GCInvoker checks whether it should start full garbage collection. It depends on the relative amount of free memory versus configured threshold.
- **`net.indx.util.gcinvoker.gcminfreepct=10`**
  If less than this percentage of the total heap is free, the GCInvoker will trigger a full garbage collection.

  These parameters limit the number of elements stored in memory. The Server will maintain the specified maximum of elements by replacing previously loaded elements with newly loaded elements from persisted files.

- **`net.indx.repository.solution.server.securitymanager.suppressCircularReferenceWarnings=false`**

  When set to true, warnings about circular references in expressions are suppressed while checking permissions to read data. The default is false, which means these warnings are not suppressed.
- **`HistoryManagement.markerValueMode`**

  Controls the historization of a MARKER value.

  Allowable property values (*all strings maybe upper or lower case*):

- DEFAULT_MARKER_VALUE_ON - Unless a primitive has its own configuration a MARKER value should be recorded.
- DEFAULT_MARKER_VALUE_OFF - Unless a primitive has its own configuration a MARKER value should not be recorded.
- FORCE_MARKER_VALUE_ON - MARKER values are recorded for all historized primitives, the primitive's own configuration is ignored.
- FORCE_MARKER_VALUE_OFF - MARKER values are not recorded for all historized primitives, the primitive's own configuration is ignored.

- **`DelayedCalculationQueue.threadsPoolSize=10`**

  Sets number of threads for asynchronious calculation. Type: number, Min: 1 , Max: 1000 , Default: 10.

- **`DelayedCalculationQueue.dcqLogEnabled=false`**

  Enables the monitor logger that logs the queue status at each interval defined by the DelayedCalculationQueue.dcqLogIntervalSec property. Note, DCQ logging is automatically enabled if the net.indx.util.syslog.DelayedCalculationQueue property is set to 4 or has debug flag set.

  Type: Boolean

  Default: false

- **`DelayedCalculationQueue.dcqLogIntervalSec=600`**

  Sets interval (in seconds) between queue status log.

  Type: Number

  Min: 10 , Max: 3600

  Default: 600

- **`SolutionServer.useLegacyScheduledExprEval=false`**

  Allow the Scheduler to suppress all other re-evaluation mechanism for the server-side expression.

  About Elimination Gaps

  With `SolutionServer.useLegacyScheduledExprEval=false` in the `solutionsettings.properties` file, 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.

- **`ExprItem.AllowLateBinding=all`**

  Disables/enables support for late-binding references in server-side expressions. If a late-binding reference is encountered in an expression at runtime where it is not allowed, then the result of the expression is BAD and no usable value is returned.

  Allowable property values:

  - **all**
    Late-binding references are allowed in all areas where expressions are supported. This is the default value for this

property.

- **none**
  Late-binding references are not allowed in any area where expressions are supported.
- **arealist**
  Late-binding references are only allowed in a defined comma-separated list of areas. The list can include one or more of the following areas:
  - **pm** - Expressions on primitive members.
  - **gc** - Expressions on global collection primitive members.
  - **ue** - User-managed expressions (API user-managed expressions, ANS conditional alerts, on-demand evaluated expressions).

  Example of `arealist` *value for global collection and user-managed areas*:

  ```
  ExprItem.AllowLateBinding=gc,ue
  ```

  Late-bound references in link collection constraint expressions are not supported. Although link collection constraint expressions will compile late-bound references, at runtime, however, this results in restricted access to the collection link and rollups that have these.

- **`ExprItem.LogDisallowedLateBinding=false`**

  If true, occurrences of late-binding use when not allowed are logged in the solution log. The default value is false.
- **`HistoryManagement.LastHistoryValueRequestWaitTime=1`**

  Specifies the wait time (in minutes) between attempts to retrieve a value from history. The default value is 1 minute (which is also the minimum value).
- **`maxCollectionStreamingThreads=12`**
  For the XHQ Connector. This property limits the maximum number of threads available for streaming purposes. If this limit is exceeded, then the request returns an OVERLOAD error.

  Min: 1 , Max: N/A

  Default: 12
- **`retryIfOverloadedCount=1`**
  For the XHQ Connector. This property sets the number of retries to stream the collection from the remote server should the server be overloaded with streaming requests.

  Min: 1 (which indicates no retry), Max: 100

  Default: 1
- **`retryIfOverloadedIntervalMs=60000`**
  For the XHQ Connector. This property sets the interval (in milliseconds) between retries.

  Min: 10000 , Max: 3600000

  Default: 60000

Connector Restart Properties

The following is a list of configurable properties that control how frequently and how many times a **connector restart** occurs. Their default values are given, followed by min/max values.

- **`connectormanager.maxrestartattempts=5`**
  (min = 1, max = 10)
  This controls how many times a connector can be restarted in the interval.
- **`connectormanager.restartcountresetdelay=3600000`**
  (min = 60000, max = no limit)
  This defines in milliseconds how long a connector must wait to be restarted after the maximum restarts has been reached. The interval starts when the first restart attempt is made.
- **`connectormanager.minrestartdelay=300000`**
  (min = 60000, max = no limit)
  This is how long (in milliseconds) a connector must wait between each restart.
- **`connectormanager.processcheckinterval=60000`**
  (min = 10000, max = no limit)
  The interval (in milliseconds) at which the connectors are checked to see if a restart is required.
- **`connectorcontroller.collectiontimeout=180000`**
  (min = 1000, max = 600000)
  This value, in milliseconds, overrides the standard time-out for collections (which by default is 60 seconds).
- **`connectorcontroller.maxtimeouts=3`**
  (min = 1, max = 1000)
  The maximum number of timeouts that can occur before the connector process is automatically shut down.
- **`connectorcontroller.generaltimeoutinterval=60000`**
  (min = 5000, max = 600000)
  The timeout interval (in milliseconds) for most calls to the connector process. This is also the pass-through timeout parameter.
- **`connectorcontroller.extendedtimeoutinterval=60000`**
  (min = greater of [10000 or generaltimeoutinterval], max = 1200000)
  The timeout interval (in milliseconds) for trend and aggregate calls to the connector process.

These properties are shown using the **short name format**. Although the long name format is still supported, it is discouraged from use.

Example of long name format:

net.indx.repository.solution.server.connectormanager.processcheckinterval=60000

Aggregated Values Cache Properties

The Aggregated Values Cache reduces the number of backend requests for aggregate values (like average, min, and max) for the same tag and for the same time period. The value would likely be the same unless the history is modified in the backend. By caching values instead of going to the backend, there is a significant gain in performance in cases of late-binding expressions and other calculations.

The Aggregate Values Cache has the following configurable properties. Their default values are given as well as min/max values (if applicable).

All properties are case insensitive and presented with capitalization for readability.

- `AggregateValuesCache.useCache=true`

  Configures whether or not to use the cache. If true, the cache is on.
- `AggregateValuesCache.maxTagsInCache=100000`

  (min=20000, max=10000000)

  Configures how many tags (primitive members) can be stored in the cache at the same time. When the maximum is reached, the oldest accessed tag is removed from the cache.
- `AggregateValuesCache.maxValuesPerTag=20`

  (min=10, max=10000)

  Configures how many aggregate values per tag can be stored in the cache at the same time. Each aggregate value is defined by a unique combination of trendMode, startTime, endTime, and aggregateMode parameters. When the maximum is reached, the oldest accessed value is removed.
- `AggregateValuesCache.numberOfWritesPerHour=4`

  (min=0, max=6)

  Configures how many times per hour, at equals intervals, the cache is written to the file. The file is used only to initialize the cache on XHQ start-up.
- `AggregateValuesCache.minutesInHourToStartWriting=10`

  (min=0, max=59)

  Configures an hourly offset (in minutes) when to start the cycle of writing to the file. This avoids always writing at the start of the hour.

  Consider the following example.

  Given (using the default values):

  - `AggregateValuesCache.numberOfWritesPerHour=4`
  - `AggregateValuesCache.minutesInHourToStartWriting=10`

  Expected Results:

  The cache is written to the file on the 10th, 25th, 40th, and 55th minute of each hour.

- `AggregateValuesCache.expirationIntervalInMinutes=360`

  (60 hours; min=10, max=365*24*60)

  Configures how long the value put in the cache is considered valid. After expiration, the value is purged and retrieved from the backend when needed.

Things to Note

- The cache file location is `%XHQ_SERVER_REPOS_TMP%\AggregateValues.cache`.
- If any configuration change is made for a primitive member, all its cached aggregate values are purged.
- If a primitive member is deleted by the UTL, all its cached aggregate values are purged.
- Only values with non-BAD quality are stored in the cache.
- If on start-up the Solution Server determines that the cache has a different version, the cache is ignored and will be overwritten on the next write.
- If on start-up the maximum number of tags allowed is less than what is in the cache file, only the values for the first max tags are read and the rest is ignored.
- If on start-up the maximum number of values per tags is less than what is in the cache file, all values are read but the normal limitation logic will purge earlier-read values to keep the current maximum.
- If on start-up while the cache is being read an exception occurs, then the read process is stopped. At this point, the tags and values already initialized in the cache will stay.
- When a value is put in the cache, the time it entered the cache is also stored.
- When a value is retrieved from the cache, it is verified that it has not stayed in the cache longer than the configured expiration period. If it has stayed longer than the expiration period, it is ignored and purged from the cache. The new value is then retrieved from the backend and is put in the cache.

Tag Sync Property

The following is a configurable property for tag synchronization. The value given (in minutes) is the default.

- **`TagSync.initialProcessingDelay=5`**
  This property causes the tag sync processing thread to delay any tag sync request from the time the XHQ Solution Server receives the request until the given delay (in minutes) has expired.

Setting the `TagSync.initialProcessingDelay` property to a positive number causes a delay of that many minutes. Sending a 0 (zero) cancels the delay and allows any queued request to be processed.

Example: With a processing delay of 5 minutes (which is the default), the corresponding message in the log is:

```
05/21/2009 15:43:18.759: INFO: Tag Sync initial processing delay set to
5 minutes.
```

Therefore, any tag sync request is delayed for 5 minutes (`05/21/2009 15:48:18.759`).

Historization Property

The following is a configurable property to enable or disable historization. The value given is the default.

- **`HistoryManagement.enableRecording=true`**
  With the value set to **true**, historization (recording) is enabled. If set to **false**, recording to the backend historians is disabled for the entire solution. The solution acts as if History configuration is not defined.

This property can be refreshed by the same command that refreshes log level properties, namely `xhqci servertool updateLogging`. For more information on this command, go to the topic, *Additional XHQ Server Actions*.

Using cachesettings.properties

The **`cachesettings.properties`** file allows you to set the debug log level on the cache server and the interval for memory tests for Java garbage collection. This file is stored at the location specified by the environment variable `%XHQ_SERVER_REPOS%` on the Enterprise Server (which is typically **`C:\XHQ\data\repos`**).

This file is not included with the XHQ install and must be created by the user. Changes to these parameters require server **restart**.

The following is a list of configurable properties and their default values.

- **`net.indx.util.syslog.sysloglevel=0`**
  This property sets the debug logging level on the cache server.
- **`net.indx.util.syslog.maxlogsizemb=5`**
  This property sets the maximum size of the log file in megabytes.
- **`net.indx.util.syslog.logextension=.out`**
  This is the extension to the full log filename.
- **`net.indx.util.syslog.logfilename=cachelog`**
  This is the base to the full log filename. It is followed by the number 1 or 2 (for example, `cachelog1.out`). This log file is located in the `%XHQ_SERVER_LOGS%` directory (which is typically under `C:\XHQ\log`). Once the first log file reaches the specified maximum size, a second log file is created (in this case, `cachelog2.out`), and the logging is switched to it.
- **`net.indx.repository.solution.cache.cacheserver.numProcessThreads=10`**
  This property controls how many threads are created for processing requests.
- **`net.indx.repository.solution.cache.cacheserver.connectionPool=10`**
  This property controls how many JDBC connections are established in the JDBC connection pool.
- **`net.indx.repository.solution.cache.cacheserver.userStorageProperties=false`**
  This property causes the cache server to use the database configuration settings found in the common storage properties file `%XHQ_SERVER_REPOS%\storage.properties`.

Using modelsettings.properties

The **`modelsettings.properties`** file allows you to set the debug log level on the enterprise server, and the interval for memory tests for Java garbage collection. This file is stored at the location specified by the environment variable `%XHQ_SERVER_REPOS%` on the Enterprise Server (which is typically **`C:\XHQ\data\repos`**).

This file is not included with the XHQ install and must be created by the user. Changes to these parameters require server **restart**.

The following is a list of configurable properties and their default values.

- **`net.indx.util.syslog.sysloglevel=0`**
  This property sets the debug logging level on the Enterprise server.
- **`net.indx.util.syslog.maxlogsizemb=5`**
  This property sets the maximum size of the log file in megabytes.
- **`net.indx.util.syslog.logextension=.out`**
  This is the extension to the full log filename.
- **`net.indx.util.syslog.logfilename=modellog`**
  This is the base to the full log filename. It is followed by the number 1 or 2 (for example, `modellog1.out`). This log file is located in the `%XHQ_SERVER_LOGS%` directory (which is typically under `C:\XHQ\log`). Once the first log file reaches the specified maximum size, a second log file is created (in this case, `modellog2.out`), and the logging is switched to it.
- **`net.indx.repository.model.server.modelserver.userRefreshRate=360`**
  This property controls the rate (in minutes) at which the users' XHQ roles are refreshed, based on the re-evaluation of the users' group membership. The range is from 1 minute to 35791; the default is 360 (which is 6 hours). If a value greater than the maximum of 35791 is entered, the refresh rate automatically defaults to 35791. To disable the refresh, set the value to 0 (zero) or less.

  This refresh setting is only read during system start-up.

Fallback

The *fallback* feature determines the behavior at runtime in terms of device and orientation type whenever the flavor view for the runtime device and orientation is unavailable. The navigation content (that is, views and nodes) is always identical in all devices. The fallback feature ensures there is always a view that is displayed irrespective of the specifically-chosen flavor of a view, built or not for the device in question.

Property	Description
`ModelServer.desktopFirstFallback`	This property controls if the solution viewer shifts/falls back to the desktop view first or consider it as the last option. By default, this setting is false. Additionally, by default, if there is a mobile flavor view, then the fallback displays the mobile view first. *Default:* ModelServer.desktopFirstFallback = false
`ModelServer.orientationFirstFallback`	This property controls which orientation is displayed if the fallback is first with the same orientation or with the same device. By default, this property is false. This means the fallback is first with the same device. In both cases, if there is no available mobile flavor, the

Property	Description
	fallback will display the desktop flavor. *Default:* ModelServer.orientationFirstFallback = false

There is no additional scaling based on the fallback mechanism. For example, falling back to a desktop flavor in a phone-portrait device shows the desktop flavor with no additional scale factors. This behaves as if the user copied the desktop flavor's content to the phone-portrait flavor.

The user can use the zooming and panning features to view the rest of the view in case the view is larger than the mobile viewing area.

Using routersettings.properties

The **routersettings.properties** file is stored at the location specified by the environment variable %XHQ_SERVER_REPOS% on the Enterprise Server (which is typically **C:\XHQ\data\repos**).

This file is not included with the XHQ install and must be created by the user. Changes to these parameters require server **restart**.

The following is a list of configurable properties and their default values.

- **net.indx.util.syslog.sysloglevel=0**
  This property sets the debug logging level on the router server.
- **net.indx.util.syslog.maxlogsizemb=5**
  This property sets the maximum size of the log file in megabytes.
- **net.indx.util.syslog.logextension=.out**
  This is the extension to the full log filename.
- **net.indx.util.syslog.logfilename=routerlog**
  This is the base to the full log filename. It is followed by the number 1 or 2 (for example, routerlog1.out). This log file is located in the %XHQ_SERVER_LOGS% directory (which is typically under C:\XHQ\log). Once the first log file reaches the specified maximum size, a second log file is created (in this case, routerlog2.out), and the logging is switched to it.

About clientvariablesettings.properties

To set client variables for the applet, **create** (if one is currently not present) the **clientvariablesettings.properties** file in the directory defined by the %XHQ_SERVER_REPOS% environment variable.

The following are examples of configurable properties.

Examples:

```
FORMAT_DATE=dd-MMM-yyyy
FORMAT_TIME=HH:mm
FORMAT_DATETIME=dd-MMM-yyyy HH:mm
LOCATION_PATH=::Refinery
LOCATION_VIEW=Sample View
TagDetail_ViewPath=::Refinery.TagBrowser
```

About 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.

This file can be found in the `repos` directory immediately below the install location selected for the development client, which is typically `\Documents and Settings\<user>\Application Data\XHQ\XHQ devClient\repos`.

XHQ Alert Notification System (ANS) Properties

The XHQ ANS properties files allow you to set user preferences for XHQ ANS. There are three files:

- The **`ans.properties`** file
  Use this file to set general user preferences.
- The **`ansmail.properties`** file
  Use this file to format a notification message sent through a **standard SMTP** channel.
- The **`ansauthmail.properties`** file
  Use this file to format a notification message sent through an **authenticated SMTP** channel.

These files are stored at the location specified by the environment variable `%XHQ_SERVER_REPOS%` (which is typically `C:\XHQ\data\repos`). A copy of these files (with default values) can also be found in the `XHQ\data\repos_empty` directory. You can use a text editor, such as Notepad, to edit this file.

- Property names are case-sensitive.
- Upon making changes to and saving these files, you must restart the XHQ server.

For a complete list of XHQ ANS properties, go to the topic, *Using the ANS Properties Files*, located in theXHQ ANS User's Guide.

About xhqboot.properties

The **`xhqboot.properties`** file contains property settings that govern how the start-up and shutdown scripts are generated. This file is stored at the location specified by the environment variable `%XHQ_SERVER_REPOS%` on the XHQ Enterprise Server (which is typically **`C:\XHQ\data\repos`**).

The default content of `xhqboot.properties` **varies** depending on which repos (that come with the product) you use.

Starting The Servers

To start the 1) **XHQ Solution Server** and the 2) **Alert Notification System (ANS) Server**, set the following line (in bold):

```
# GENERATED BY XHQBOOT Sun Jan 23 05:32:08 PST 2005
#----------------------------------
# STANDARD XHQ SERVER CONFIG
modules=xhq_aserver
#----------------------------------
```

To start the **XHQ Application Server** in addition to the XHQ Solution Server and the XHQ ANS Server, do **one** of the following.

- Either add **`xhq_appserver`** as a module (in bold):

```
# GENERATED BY XHQBOOT Sun Jan 23 05:32:08 PST 2005
#----------------------------------
# STANDARD XHQ SERVER CONFIG
modules=xhq_aserver,xhq_appserver
#----------------------------------
```

- Or, replace the default module with the following (in bold):

```
# GENERATED BY XHQBOOT Sun Jan 23 05:32:08 PST 2005
#----------------------------------
# STANDARD XHQ SERVER CONFIG
modules=xhq_appserver
#----------------------------------
```

- Or, remove the **`module`** line altogether:

```
# GENERATED BY XHQBOOT Sun Jan 23 05:32:08 PST 2005
#----------------------------------
# STANDARD XHQ SERVER CONFIG
#----------------------------------
```

In this case, an **empty** `xhqboot.properties` file with no **`module`** line starts the XHQ Solution Server, the XHQ ANS Server, and the XHQ Application Server. This is the default.

Save your changes to the **`xhqboot.properties`** file.

About install.properties

The `install.properties` file contains property settings that enable you to configure the XHQ installer.

For **details**, see the chapter, *The install.properties File*, located in the XHQ Installation Guide.

Things to Note

- The `install.properties` file must be located in the `%ProgramFiles(x86)%\XHQ\XHQ Install Files` directory, regardless of where you install XHQ.
- Log files are automatically written to the `%ProgramFiles(x86)%\XHQ\XHQ Install Logs` directory. The filenames are in the format: `XHQ_Install_<timestamp>.log`.

To create the install.properties file

1. Create the `%ProgramFiles(x86)%\XHQ\XHQ Install Files` directory.

   Example: `C:\Program Files (x86)\XHQ\XHQ Install Files`

2. Using a text editor (such as Notepad), **create** the `install.properties` file.
3. **Save** this file into the directory you created in step 1.

About jetty.properties

This file is located in the `%XHQ_SERVER_DATA%\conf\resources\local\properties` folder (which by default is `C:\XHQ\data\conf\resources\local\properties`).

To edit `jetty.properties`, you must contact the *XHQ Customer Support Team* for specific instructions.

Property	Description
`xhq.jetty.http.enabled`	Determines if HTTP for jetty is enabled. *Default*: `xhq.jetty.http.enabled=true`
`xhq.jetty.http.port`	Determines the HTTP port number from which the View Server listens for HTTP traffic. *Default*: `xhq.jetty.http.port=28888`
`xhq.jetty.https.enabled`	Determines if HTTPS for jetty is enabled. *Default*: `xhq.jetty.https.enabled=true`
`xhq.jetty.https.port`	Determines if HTTPS port number from which the View Server listens for HTTP traffic. *Default*: `xhq.jetty.https.port=28889`
`xhq.jetty.negotiate.enabled`	Determines if Negotiate authentication protocol is enabled. Set this property to false if the XHQ Service is started by a different user than Local System. This avoids issues with Kerberos authentication and falls back to NTLMv2. *Default*: `xhq.jetty.negotiate.enabled=true`

About monitoring.properties

The **`monitoring.properties`** file contains property settings that customize the tests done during the health check, or disables the testing (allowing the test to pass). This file is stored at the location specified by the environment variable `%XHQ_SERVER_REPOS%` on the Enterprise Server (which is typically `C:\XHQ\data\repos`).

To activate the property, "uncomment" the given line by removing the number sign (#) at the beginning of the line.

XHQ Module	Property	Description
XHQ Model Server (xhq_eserver)	`xhq_eserver.healthcheck.disabled`	Setting this property to true disables the health check tests, allowing the xhqci healthcheck command to skip the testing of this module.
	Example: `xhq_eserver.healthcheck.disabled=true`	
	`xhq_eserver.healthcheck.class`	This class property specifies the component class with which to perform the health check tests. If this property is not defined, then no component class testing is done.
	Example: `xhq_eserver.healthcheck.class=Measurement`	
XHQ Solution Server (xhq_sserver)	`xhq_sserver.healthcheck.disabled`	Setting this property to true disables the health check tests, allowing the xhqci healthcheck command to skip the testing of this module.
	Example: `xhq_sserver.healthcheck.disabled=true`	
	`xhq_sserver.healthcheck.collection`	This collection property specifies the global collection with which to perform the health check tests. If this property is not defined, then no collection-specific testing is done.
	Example: `xhq_sserver.healthcheck.collection=Measurements`	
	`xhq_sserver.healthcheck.testpoint`	This property specifies the member/tag/alias with which to perform the health check tests. If this property is not defined, then no value/trend testing is done.
	Example: `xhq_sserver.healthcheck.testpoint=::Enterprise.SIMPoint.CurrentValue`	
XHQ Cache Server (xhq_cserver)	`xhq_cserver.healthcheck.disabled`	Setting this property to true disables the health check tests, allowing the xhqci healthcheck command to skip the testing of this module.
	Example: `xhq_cserver.healthcheck.disabled=true`	

About storage.properties

The **`storage.properties`** file is the common storage properties file and is found at the location specified by the environment variable `%XHQ_SERVER_REPOS%` on the Enterprise Server (which is typically **`C:\XHQ\data\repos`**). It defines how any connection is made to the external database and includes the storage mode for the solution and model servers.

Warning
Do not edit/change these storage properties unless instructed to do so by the XHQ Customer Support Team.

The following is a list of properties and their default values.

- **`storage.type=mixed`**
  This property enables/disables the database storage facility.

  The mixed mode is used by ALL advanced XHQ features (such as XHQ ANS, XHQ Performance Management, Reporting Services, and so forth).

  The supported value for this property is:

  - **`mixed`**
    Which maintains XHQ ANS configuration and information in the XHQ High-Performance Database.

- **`cache.db.poolsize`**
  This property sets the size of the database connection pool for the cache server and is set to an internal XHQ default value.

  If the `storage.properties` configuration file contains an explicit setting for the **`cache.db.poolsize`** parameter (which was also known as the `db.poolsize.cache` parameter for XHQ 4.5 SP1 B143 and earlier releases), then this setting must be removed from the configuration file to ensure that the **XHQ internal default** is used.

  In test situations, a setting of 10 was too low and caused problems. And, although a setting value of 50 was also often used for XHQ 4.0/4.1, this caused problems in XHQ 4.5 SP1. **In both these cases, upgrading to XHQ 4.5 SP1 (or later), required the removal of this parameter setting from the `storage.properties` configuration file.**

  Depending on which value is greater, the –cp command-line parameter may override this property setting. For example, if you set `cache.db.poolsize=100` and `-cp` to `50`, then the pool size is the greater of the two, which in this case is 100. If you set `cache.db.poolsize=20` and `-cp` to `50`, then the pool size is 50.

In addition, the Data Recorder uses the following properties:

- `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.

About xhq_service.properties

This properties file contains settings that control various XHQ Service behaviors.

Things to Note

- The settings in this file override environment variable settings of the same name.
- This file is read at start-up of the XHQ Service.
- The examples given in the table below show **default** values.

Property	Description
`XHQ_AUTOSTART`	When set to true, the XHQ System automatically starts when the XHQ Service is started (for example, at system boot). *Default:* `XHQ_AUTOSTART=true`
`XHQ_SERVER_VERBOSE`	This property is for debugging purposes. It allows for more information to be sent to the Windows Application Event Log during the course of operation. *Default:* `XHQ_SERVER_VERBOSE=false`
`XHQ_AUTO_RESTART_ON_FAIL`	When set to true, the XHQ System is restarted when an XHQ Server process stops unexpectedly. *Default:* `XHQ_AUTO_RESTART_ON_FAIL=true`
`XHQ_HEALTH_CHECK_INTERVAL`	The interval (in minutes) in which the XHQ Service runs a health check test of the XHQ System. The value can be any positive integer in the range of 1 to 10. To disable the health check, set the interval value to 0. The default interval is 5 minutes. If a health check failure is detected, it is reported to the Windows Application Event Log and the service takes the action defined by the `XHQ_HEALTH_CHECK_FAIL_ACTION` property. *Default:* `XHQ_HEALTH_CHECK_INTERVAL=5`
`XHQ_HEALTH_CHECK_TIMEOUT`	The number of seconds that the XHQ Service is allowed for a health check test to complete. If the health check does not complete within this allotted time, it is considered a failed test. The default timeout is 120 seconds. The value can be any positive integer in the range of 5 seconds to the value of `(XHQ_HEALTH_CHECK_INTERVAL*60)/2` seconds. If `XHQ_HEALTH_CHECK_TIMEOUT` is greater than `XHQ_HEALTH_CHECK_INTERVAL`, then `XHQ_HEALTH_CHECK_INTERVAL` is set to `2*XHQ_HEALTH_CHECK_TIMEOUT`. *Default:* `XHQ_HEALTH_CHECK_TIMEOUT=120`
`XHQ_HEALTH_CHECK_FAIL_ACTION`	The action taken by the XHQ Service should the health check operation returns a failure. Options are: • **none** No action is taken; reports failure to the event log (default).

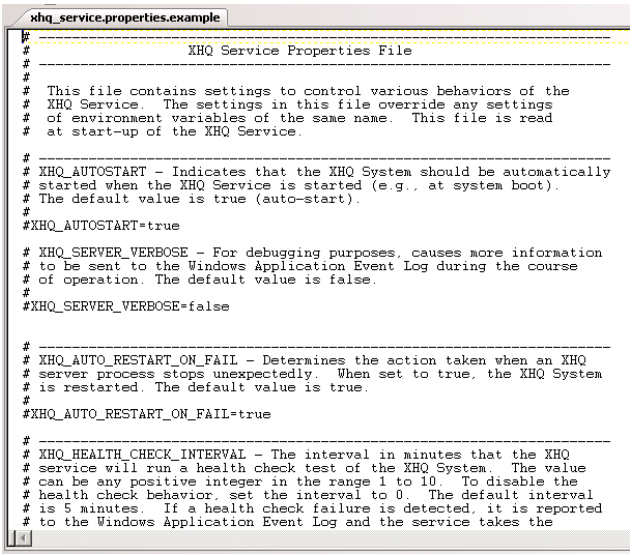
Property	Description
	• **shutdown** Shuts down the XHQ System; useful for redundant system set-ups. • **restart** Restarts the XHQ System. *Default:* `XHQ_HEALTH_CHECK_FAIL_ACTION=none`

This file is not included with the XHQ install. It is created by the user and saved in the given location.

To create the xhq_service.properties file

1. Go to the location `%XHQ_SERVER_HOME%\bin` (which by default is the `C:\Program Files (x86)\XHQ\XHQ Server\bin` directory).
2. Find the **`xhq_service.properties.example`** file, **copy** and **paste** it into the same `bin` directory.
3. **Rename** the copied file to **`xhq_service.properties`**.
4. Open `xhq_service.properties`.

Each line in this file has been "commented out" with the hashtag symbol.



```
xhq_service.properties.example
# ------------------------------------------------------------------------
#                    XHQ Service Properties File
# ------------------------------------------------------------------------
#
#  This file contains settings to control various behaviors of the
#  XHQ Service.  The settings in this file override any settings
#  of environment variables of the same name.  This file is read
#  at start-up of the XHQ Service.

# ------------------------------------------------------------------------
# XHQ_AUTOSTART - Indicates that the XHQ System should be automatically
# started when the XHQ Service is started (e.g., at system boot).
# The default value is true (auto-start).
#
#XHQ_AUTOSTART=true

# XHQ_SERVER_VERBOSE - For debugging purposes, causes more information
# to be sent to the Windows Application Event Log during the course
# of operation. The default value is false.
#
#XHQ_SERVER_VERBOSE=false


# ------------------------------------------------------------------------
# XHQ_AUTO_RESTART_ON_FAIL - Determines the action taken when an XHQ
# server process stops unexpectedly.  When set to true, the XHQ System
# is restarted. The default value is true.
#
#XHQ_AUTO_RESTART_ON_FAIL=true

# ------------------------------------------------------------------------
# XHQ_HEALTH_CHECK_INTERVAL - The interval in minutes that the XHQ
# service will run a health check test of the XHQ System.  The value
# can be any positive integer in the range 1 to 10.  To disable the
# health check behavior, set the interval to 0.  The default interval
# is 5 minutes.  If a health check failure is detected, it is reported
# to the Windows Application Event Log and the service takes the
```

5. **Remove the comment hashtag** in front of the property(ies) you want to activate.

xhq_service.properties.txt

```
# ------------------------------------------------------------------------
#                    XHQ Service Properties File
# ------------------------------------------------------------------------
#
#  This file contains settings to control various behaviors of the
#  XHQ Service.  The settings in this file override any settings
#  of environment variables of the same name.  This file is read
#  at start-up of the XHQ Service.

# ------------------------------------------------------------------------
# XHQ_AUTOSTART - Indicates that the XHQ System should be automatically
# started when the XHQ Service is started (e.g., at system boot).
# The default value is true (auto-start).
#
XHQ_AUTOSTART=true

# XHQ_SERVER_VERBOSE - For debugging purposes, causes more information
# to be sent to the Windows Application Event Log during the course
# of operation. The default value is false.
#
XHQ_SERVER_VERBOSE=false


# ------------------------------------------------------------------------
# XHQ_AUTO_RESTART_ON_FAIL - Determines the action taken when an XHQ
# server process stops unexpectedly.  When set to true, the XHQ System
# is restarted. The default value is true.
#
XHQ_AUTO_RESTART_ON_FAIL=true

# ------------------------------------------------------------------------
# XHQ_HEALTH_CHECK_INTERVAL - The interval in minutes that the XHQ
# service will run a health check test of the XHQ System.  The value
# can be any positive integer in the range 1 to 10.  To disable the
# health check behavior, set the interval to 0.  The default interval
# is 5 minutes.  If a health check failure is detected, it is reported
# to the Windows Application Event Log and the service takes the
```

6. **Save** the file.
7. **Restart** the XHQ Server.

About the XHQ View Server and viewserversettings.properties

The XHQ View Server is a core XHQ Server module that runs at the same level as the XHQ Model and Solution Servers. It allows browsers that are compatible with the latest HTML5 standards to access XHQ views, providing an alternative to the XHQ applet. Technically, the XHQ View Server reads data through the XHQ Model and Solution Servers and converts it into suitable HTML5 format.

Things to Note

- The XHQ View Server module is delivered via the **`viewserver.jar` file**, which is installed in the same location as all the other XHQ core server jar files.
- The XHQ View Server runs under the same user as the XHQ Solution and Model Servers. It is started by XHQ Services.
- The XHQ View Server **starts automatically** with all other XHQ core servers.
- The XHQ View Server can be configured using the **`viewserversettings.properties`** file. Like the other XHQ core servers, it uses standard properties to control log level, log file size, and so forth.
- The XHQ View Server **log file** is named `viewserver.out` by default.

Setting Properties

Use the `viewserversettings.properties` file to configure the **XHQ View Server**.

WARNING

Do not modify any of these settings in this initial release without first consulting the XHQ Customer Support Team to ensure that the change is supported.

XHQ View Server Settings Properties

Property	Description
viewserver.model.server	The IP address of the XHQ Model Server. *Default*: viewserver.model.server = localhost
viewserver.solution.server	The IP address of the XHQ Solution Server. *Default*: viewserver.solution.server = localhost
viewserver.userSessionInactiveTimeoutSeconds	Session expiration time-out. When the session expires, all resources are freed. Minimum value is -1 (the session never expires); maximum value is 604800 (1 week). *Default*: viewserver.userSessionInactiveTimeoutSeconds = 600 (which is 10 minutes) For additional information, go to the topic, *Session Timeouts*.
viewserver.numProcessThreads	The number of threads that serve an API request queue. Note: IIS requests use this queue. *Default*: viewserver.numProcessThreads = 400
viewserver.subscriptionsChunkSize	Limits the number of subscriptions sent in one chunk.

Property	Description
	• Unit: items • Minimum value: 1 • Maximum value: 10000 *Default*: viewserver.subscriptionsChunkSize = 1000
viewserver.subscriptionsInterval	Sets the interval (in ms) between chunk sending attempts. • Unit: milliseconds • Minimum value: 30 • Maximum value: 10000 *Example*: If the interval is set to 1000 (ms), then chunks (of up to 100 items each, which is the default) will be sent one time per second. *Default*: viewserver.subscriptionsInterval = 300

The following **Client Logger properties** are also defined in the `viewserversettings.properties` file.

XHQ View Server Settings Properties - Client Logger

Property	Description
clientlogger.logExtension	This is the format extension for the log file. *Default*: clientlogger.logExtension = .out
clientlogger.logFileName	This is the base to the full log file name. It is followed by the number 1 or 2 (for example, viewserverlog1.out). This log file is located in the `%XHQ_SERVER_LOGS%` directory, which is typically `C:\XHQ\log`. Once the first log file reaches the specified maximum size, a second log file is created (in this case, viewserver2.out), and the logging is switched to it. *Default*: clientlogger.logFileName = viewserverlog
clientlogger.maxLogSizeMb	The maximum size of the log file in **megabytes**. *Default*: clientlogger.maxLogSizeMb = 5
clientlogger.displayThreadName	If **true**, displays the Thread name in the logs. *Default*: clientlogger.displayThreadName = true
clientlogger.displayChannelName	If **true**, displays the Channel name in the logs. *Default*: clientlogger.displayChannelName = true
clientlogger.displayUserName	If **true**, displays the User name in the logs. *Default*: clientlogger.displayUserName = false
clientlogger.numLogFiles	The number of log files. *Default*: clientlogger.numLogFiles = 10

You can also specify log levels by adding the **`logLevel` parameter** to the URL for the XHQ HTML5 Solution Viewer. The `logLevel` value options are:

- **SEVERE**
  Logs only critical errors.
- **WARNING**
  Logs critical errors and warnings.
- **ALL**
  Logs all system messages.

To disable logging, simply remove the `logLevel` parameter from the URL.

Example:

```
http://localhost/indx/sv/?logLevel=ALL
```

You can specify the independent log level for each component by adding the **`logType` parameter** to the URL and setting the value to *FancyWindow*.

Example:

```
http://localhost/indx/sv/?logType=FancyWindow
```

Session Timeouts

The default value for the timeout session is 600 seconds (or 10 minutes). This may lead to a high (and unnecessary) number of sessions accumulated in the XHQ View Server. Therefore, we recommended you use a value **not less than** 180 seconds (or 3 minutes) and **not more than** 300 seconds (or 5 minutes).

To change the session timeouts

1. Open the **Internet Information Services (IIS) Manager**.
2. In the **Connections pane**, expand the XHQ server and click **Application Pools**.
   A list of application pools appears in the main panel.
3. Locate and right-click on **xhqsvPool**.
   The "Advanced Settings" dialog appears.
4. Scroll down to **Recycling**, find **Regular Time Interval (minutes)**, and **click inside the cell** to edit.
5. Delete 1740 and **enter 180**.
6. Verify that the **Private Memory Limit** and the **Virtual Memory Limit** settings (also under Recycling) are **both set to zero** (0).
7. Go to **`%XHQ_SERVER_REPOS%`**, which by default is `C:\XHQ\data\repos`.
8. Create (or open, if it already exists) **`viewserversettings.properties`** file.
   This properties file will be saved in the repos folder.
9. **Add** the following property:
   `viewserver.userSessionInactiveTimeoutSeconds=180`
10. **Save** the properties file.

Tuning Database Performance (XHQ DB Settings Utility)

The use of this properties file requires prior approval by the XHQ Customer Support Team.

Use the `XHQ_DB_Settings` command-line utility to change the High-Performance Database settings for performance improvement.

From `%XHQ_DBMS_HOME%\dbadmin\settings` (which is typically, `<systemdrive>:\XHQ\XHQDB\dbadmin\settings`), run the following:

`> XHQ_DB_Settings [size xhq_instance]`

Where:

- **`size`** is one of the following: small, medium, large, custom, custom_silent, or display.

  The **small**/**medium**/**large** options set the predefined sizing for the High-Performance Database. The default is small.

  For **custom**, the settings are taken from the *database.custom.properties* file; **custom_silent** is similar, but does not prompt the user for confirmation.

  For **display**, the current settings are displayed.
- **`xhq_instance`** is the XHQ High-Performance Database instance name. The options are: `XHQ`, `XHQCACHE`, or `XHQHIST`.

Example: To display the current database settings for the `XHQCACHE` instance.

```
> XHQ_DB_Settings display xhqcache
```

For the **legacy** High-Performance Database, run the following:

`> XHQ_DB_Settings [instance sys-password size [redo_log_directory]]`

Where:

- **`instance`** is the XHQ instance: `XHQ`, `XHQCACHE`, or `XHQHIST`.
- **`sys_password`** is the password for the sys user and is needed to perform the database backup.
- **`size`** is the database size. The options are: small, medium, large, custom, or display.

  For **custom** , the settings are taken from the *database.custom.properties* file.

  For **display**, the current settings are displayed.

  For **small**/**medium**/**large**, the *settings in this table* are used
- **`redo_log_directory`** is OPTIONAL. It is the online redo log.

Small/Medium/Large DB Settings

	sga_max_size sga_target	pga_aggregate_target	memory_max_target memory_target	processes	job_queue_process	open_cursors
Small	1024M	1024M	0	325	10	600
Medium	2048M	2048M	0	500	16	1000
Large	3072M	3072M	0	800	24	1200

About database.custom.properties

The use of this properties file requires prior approval by the XHQ Customer Support Team.

The `database.custom.properties` file contains custom property settings for the High-Performance Database (or embedded database). This file is used by the command-line utility, `XHQ_DB_Settings` to tune the embedded database performance.

This file is not included with the XHQ install. It needs to be created by the user and saved in the `%XHQ_DBMS_HOME%\dbadmin\settings` folder (which is typically, `<systemdrive>:\XHQ\XHQDB\dbadmin\settings`). An **example file** is included in this folder and can be used as the basis for the `database.custom.properties` file. Simply copy and paste it, then rename the file by removing the "`.example`" at the end.

Example of database.custom.properties file

```
##########################################################################################
# XHQ instance settings
# Note: Please send changes to XHQ support for approval before applying these
# settings.
#
# sga_max_size memory_max_target
# sga_target pga_aggregate_target memory_target processes job_queue_process open_cursors
# Small 1024M 1024M 0 325 10 600
# Medium 2048M 2048M 0 500 16 1000
# Large 3072M 3072M 0 800 24 1200
##########################################################################################
##########################################################################################
# memorytarget: Megabyte. Must be smaller than memory max target.
# ex: 1024
##########################################################################################
xhq.database.xhq.memorytarget=0
##########################################################################################
# memorymaxtarget: Megabyte. Sum of pgaaggregatetarget + sgamaxsize
# ex: 1024
##########################################################################################
xhq.database.xhq.memorymaxtarget=0
##########################################################################################
# pgaaggregatetarget: Megabyte
# ex: 1024
##########################################################################################
xhq.database.xhq.pgaaggregatetarget=1024
##########################################################################################
# sgamaxsize: in Megabyte
# ex: 1024
##########################################################################################
xhq.database.xhq.sgamaxsize=1024
##########################################################################################
# sgatarget: in Megabyte
# ex: 1024
##########################################################################################
xhq.database.xhq.sgatarget=1024
##########################################################################################
# processes: integer value
# ex: 100, 225
##########################################################################################
xhq.database.xhq.processes=325
##########################################################################################
# logsize: in Megabyte
```

```
# ex: 100 or 200
##########################################################################################
#xhq.database.xhq.logsize=300
##########################################################################################
# jobqueueprocesses: integer value
# ex: 10, 16, or 24
##########################################################################################
xhq.database.xhq.jobqueueprocesses=10
##########################################################################################
# opencursors: integer value
# ex: 600, 1000, or 1200
##########################################################################################
xhq.database.xhq.opencursors=600
##########################################################################################
# logbuffer: in MegaByte
# ex: 5, 10
##########################################################################################
#xhq.database.xhq.logbuffer=10
##########################################################################################
# XHQHIST instance settings
##########################################################################################
##########################################################################################
# memorytarget: Megabyte. Must be smaller than memory max target.
# ex: 1024
##########################################################################################
xhq.database.xhqhist.memorytarget=0
##########################################################################################
# memorymaxtarget: Megabyte. Must be more than pgaaggregatetarget + sgamaxsize
# ex: 1024
##########################################################################################
xhq.database.xhqhist.memorymaxtarget=0
##########################################################################################
# pgaaggregatetarget: Megabyte
# ex: 1024
##########################################################################################
xhq.database.xhqhist.pgaaggregatetarget=1024
##########################################################################################
# sgamaxsize: in Megabyte
# ex: 1024
##########################################################################################
xhq.database.xhqhist.sgamaxsize=1024
##########################################################################################
# sgatarget: in Megabyte
# ex: 1024
##########################################################################################
xhq.database.xhqhist.sgatarget=1024
##########################################################################################
# processes: integer value
# ex: 100, 225
##########################################################################################
xhq.database.xhqhist.processes=325
##########################################################################################
# logsize: in Megabyte
# ex: 100 or 200
##########################################################################################
#xhq.database.xhqhist.logsize=300
##########################################################################################
# jobqueueprocesses: integer value
# ex: 10, 16, or 24
##########################################################################################
xhq.database.xhqhist.jobqueueprocesses=10
```

```
##########################################################################################
# opencursors: integer value
# ex: 600, 1000, or 1200
##########################################################################################
xhq.database.xhqhist.opencursors=600
##########################################################################################
# logbuffer: in MegaByte
# ex: 5, 10
##########################################################################################
#xhq.database.xhqhist.logbuffer=10
##########################################################################################
# XHQCACHE instance settings
##########################################################################################
##########################################################################################
# memorytarget: Megabyte. Must be smaller than memory max target.
# ex: 1024
##########################################################################################
xhq.database.xhqcache.memorytarget=0
##########################################################################################
# memorymaxtarget: Megabyte. Must be more than pgaaggregatetarget + sgamaxsize
# ex: 1024
##########################################################################################
xhq.database.xhqcache.memorymaxtarget=0
##########################################################################################
# pgaaggregatetarget: Megabyte
# ex: 1024
##########################################################################################
xhq.database.xhqcache.pgaaggregatetarget=1024
##########################################################################################
# sgamaxsize: in Megabyte
# ex: 1024
##########################################################################################
xhq.database.xhqcache.sgamaxsize=1024
##########################################################################################
# sgatarget: in Megabyte
# ex: 1024
##########################################################################################
xhq.database.xhqcache.sgatarget=1024
##########################################################################################
# processes: integer value
# ex: 100, 225
##########################################################################################
xhq.database.xhqcache.processes=325
##########################################################################################
# logsize: in Megabyte
# ex: not appliable
##########################################################################################
xhq.database.xhqcache.logsize=100
##########################################################################################
# jobqueueprocesses: integer value
# ex: 10, 16, or 24
##########################################################################################
xhq.database.xhqcache.jobqueueprocesses=10
##########################################################################################
# opencursors: integer value
# ex: 600, 1000, or 1200
##########################################################################################
xhq.database.xhqcache.opencursors=600
##########################################################################################
# logbuffer: in MegaByte
# ex: 5, 10
```

```
##########################################################################################
#xhq.database.xhqcache.logbuffer=10
```

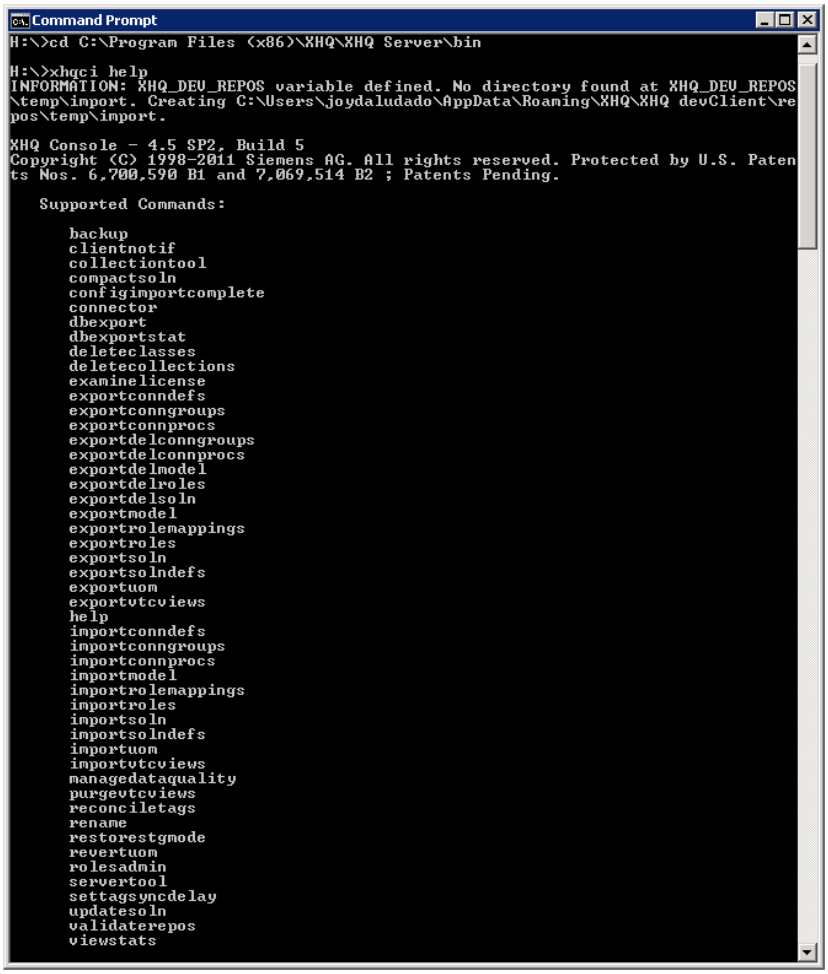
6 | Using the xhqci Utility

By running the **`xhqci.bat`** utility through a DOS command prompt, you can perform the following tasks.

Commands:

- Exporting a Solution
- Importing a Solution
- Exporting Components
- Importing Components
- Exporting Component Roles
- Importing Component Roles
- Exporting Roles Mappings
- Importing Role Mappings
- Exporting View Tile Composer VTC Views
- Importing View Tile Composer VTC Views
- Purging View Tile Composer VTC Views
- Exporting Units of Measure UoM
- Importing Units of Measure UoM
- Revert Units of Measure
- Exporting Connector Definitions
- Importing Connector Definitions
- Exporting Connector Groups
- Importing Connector Groups
- Exporting Connector Processes
- Importing Connector Processes
- Exporting Solution Definitions
- Importing Solution Definitions
- Exporting Connection Groups to be Deleted
- Exporting Connection Processes to be Deleted
- Exporting the Model to be Deleted
- Exporting Deleted Role Mappings
- Exporting Solution Collections to be Deleted
- Exporting a Database
- Setting Configuration Import Complete
- Refreshing Roles
- Renaming Components
- Updating a Solution
- Solution Information
- Model Information
- Using ViewStats
- Compressing the Solution
- Starting Stopping a Connector
- Restoring Storage Mode
- Looking at the XHQ License
- Managing Data Quality
- Updating Client Notification
- Performing a Tag Reconciliation
- Collection Tool
- Online Back up
- Setting a Tag Synchronization Processing Delay
- Deleting Global Collections
- Deleting Classes
- Validating A Repos
- Resolving Aliases to Paths
- Checking the Health of the XHQ Server
- Historical Evaluation
- Clearing Audit Records
- Additional XHQ Server Actions

About the xhqci Commands



```
Command Prompt
H:\>cd C:\Program Files (x86)\XHQ\XHQ Server\bin

H:\>xhqci help
INFORMATION: XHQ_DEV_REPOS variable defined. No directory found at XHQ_DEV_REPOS
\temp\import. Creating C:\Users\joydaludado\AppData\Roaming\XHQ\XHQ devClient\re
pos\temp\import.

XHQ Console - 4.5 SP2, Build 5
Copyright (C) 1998-2011 Siemens AG. All rights reserved. Protected by U.S. Paten
ts Nos. 6,700,590 B1 and 7,069,514 B2 ; Patents Pending.

   Supported Commands:

      backup
      clientnotif
      collectiontool
      compactsoln
      configimportcomplete
      connector
      dbexport
      dbexportstat
      deleteclasses
      deletecollections
      examinelicense
      exportconndefs
      exportconngroups
      exportconnprocs
      exportdelconngroups
      exportdelconnprocs
      exportdelmodel
      exportdelroles
      exportdelsoln
      exportmodel
      exportrolemappings
      exportroles
      exportsoln
      exportsolndefs
      exportuom
      exportvtcviews
      help
      importconndefs
      importconngroups
      importconnprocs
      importmodel
      importrolemappings
      importroles
      importsoln
      importsolndefs
      importuom
      importvtcviews
      managedataquality
      purgevtcviews
      reconciletags
      rename
      restorestgmode
      revertuom
      rolesadmin
      servertool
      settagsyncdelay
      updatesoln
      validaterepos
      viewstats
```

Supported xhqci Commands

Things to Note Regarding Syntax

- The **pipe symbol** "|" in between parameters indicate an either/or choice.
- Parameters enclosed in **square brackets** "[ ]" are *optional*.
- Parameter values in **angle brackets** "<>" must be substituted with literal value.
- Any argument in the form of **`arg=<val>`**, where `arg` is the parameter and `<val>` is the value, can also be written **`arg:<val>`**.
- Parameters may not include spaces unless enclosed in **double quotes**.

  Example: `xhqci exportsoln "file=C:\My Files\Solution Export.xml"`

Common Parameters to xhqci Commands

These are parameters common to all `xhqci` command-line syntax.

Syntax:

```
<command-line>
<parameters specific to the command-line>
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Common Command-Line Parameters

Parameter	Description
eshost	The hostname for the Enterprise Server. If the parameter is not specified, then this value is determined by: 1. The sshost parameter, if specified; 2. The XHQCI_HOST environment variable; 3. The Server Access Utility setting; 4. The localhost default if none of the above is defined. **Note:** Use either eshost or rshost (not both).
rshost	The hostname for the router. **Note:** Use either eshost or rshost (not both).
sshost	The hostname for the Solution Server. This is the machine where the solution you want to update is located. If the parameter is not specified, then this value is determined by: 1. The eshost parameter, if specified; 2. The XHQCI_HOST environment variable; 3. The Server Access Utility setting; 4. The localhost default if none of the above is defined.
uid	The user login name. Administrator privilege is required to run the `xhqci` utility with credentials. **Note:** This parameter is not required if XHQ security is enabled. If omitted, then the credentials of the user invoking this command are used for authentication.
pw	The user password. Administrator privilege is required to run the `xhqci` utility with credentials. **Note:** This parameter is not required if XHQ security is enabled. If omitted, then the credentials of the user invoking this command are used for authentication.

xhqci Access and Security

Administrator privilege is required to run the `xhqci` utility with credentials. To use credentials other than the user who is logged in, you must define the **`uid`** and **`pw`** parameters.

Details on how `xhqci` works with either **Classic** or **Active Directory** security enabled can be found in the topic, *About Security Modes*.

Using the Command-Line

The first thing you will need to do is learn how to execute the **`xhqci.bat`** batch file, which is typically located in the `\Program Files\XHQ\XHQ Server\bin` folder.

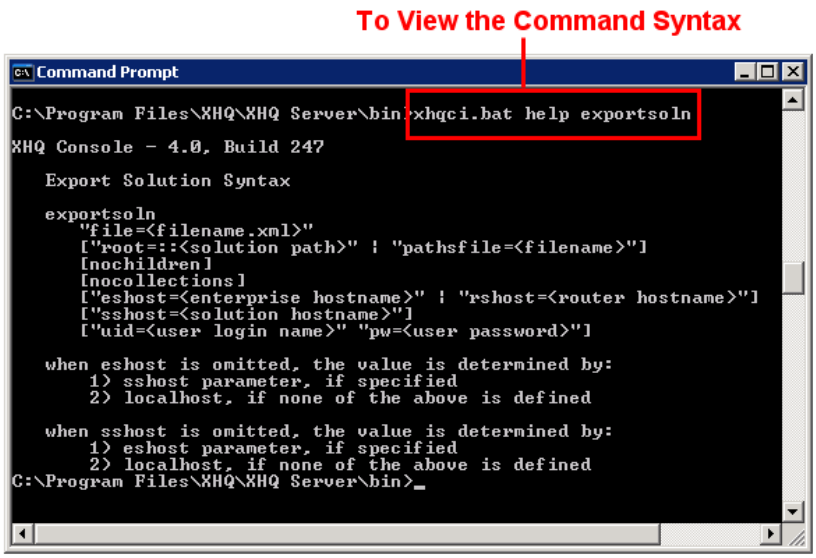
The batch file `xhqci_s.bat` is used on the Server for processes that run only on the Server (for example, tag synchronization). The `xhqci.bat` file is used from the Development Clients. Both batch files support the same commands.

To execute the xhqci batch file

1. From a DOS prompt, enter the location for the `\Program Files (x86)\XHQ\XHQ Server\bin` directory.
2. After the prompt, type **xhqci help**.
   A list of supported commands appears.
3. To view the syntax for a command, enter the following sequence after the location path and the prompt:

```
xhqci help <command name>
```

Example: `C:\Progam Files\XHQ\XHQ Server\bin>xhqci help exportsoln`



Example: To View Syntax: for exportsoln Command

Exporting a Solution

Syntax:

```
exportsoln
"file=<filename.xml>"
["alltags" | "tagsfile=file_name" | "tags=tag1,tag2"]
["tagmembers=all" | "tagmembers=unconfigured" | "tagmembers=configured"]
[tagsonly]
["root=::<solution path>" | "pathsfile=<filename>"]
[nochildren]
[nocollections]
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Example 1:

Given: Security is enabled.
All tags with configured tag members are exported; the solution tree is not exported.

```
>xhqci exportsoln "file=C:\Documents and Settings\jsmith\Application
Data\XHQ\XHQ devClient\repos\acmesoln.xml" "alltags"
"tagmembers=configured" tagsonly "rshost=router1" "sshost=server1"
"uid=jsmith" "pw=enterprise"
```

Example 2:

Given: Security is enabled.
Only myRoot is exported.

```
>xhqci exportsoln "file=C:\Documents and Settings\jsmith\Application
Data\XHQ\XHQ devClient\repos\acmesoln.xml" "root=::mySolution.myRoot"
nochildren "eshost=server1" "sshost=server1" "uid=jsmith"
"pw=enterprise"
```

Example 3:

Given: Security is enabled.
The component myRoot and all "children" of myRoot are exported.

```
>xhqci exportsoln "file=C:\Documents and Settings\jsmith\Application
Data\XHQ\XHQ devClient\repos\acmesoln.xml" "root=::mySolution.myRoot"
"rshost=router1" "sshost=server1" "uid=jsmith" "pw=enterprise"
```

Example 4:

Given: Security is disabled.
The component myRoot and all "children" of myRoot are exported.

```
>xhqci exportsoln "file=C:\Documents and Settings\jsmith\Application
Data\XHQ\XHQ devClient\repos\acmesoln.xml" "root=::mySolution.myRoot"
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci CommandsCommon Parameters to xhqci Commands*.

Command-Line Parameters for exportsoln

Parameter	Description
file	The filename of the solution you are exporting. The supported file format is XML. Typically, the .xml file is exported to the `\Documents and Settings\<user>\Application Data\XHQ\XHQ devClient\repos` directory.
<tags to export>	Use one of the following:

Parameter	Description
	• **alltags** Exports all tags. • **tagsfile=file_name** Exports a list of tags from the given .txt file. Create the .txt file. List each tag on a new line. • **tags=tag1_name,tag2_name,tagn_name** Exports the tags listed. This list is comma-dellimited. **Important**: If no parameter is provided, then the tag configuration will not be included in the export. It takes about three hours to export one million tags. So if you need to export all the tag configurations, you may want to have this running in the background.

tagmembers	To use this parameter, one of the following <tags to export> parameters must be defined: alltags, tagsfile, or tags.
	Options:
	• **all**
	All tag value members are exported. This is the default.
	• **configured**
	Only tag value members with configurations are exported.
	• **unconfigured**
	Only tag value members with no configurations are exported.
	Example:
	There are three tags (tag1, tag2, tag3) with one value member. There is no configuration for the value member for tag2, while tag1 and tag3 value members have configuration.
	The following command lists the value member of tag2 that is not configured:
	`>xhqci exportsoln "file=C:\test\tagSolnExport.xml" "tags=tag1,tag2,tag3" "tagmembers=unconfigured"`
	The following command list the value members for tag1 and tag3:
	`>xhqci exportsoln "file=C:\test\tagSolnExport.xml" "tags=tag1,tag2,tag3" "tagmembers=configured"`

tagsonly	To use this parameter, a <tags to export> parameter (alltags, tagsfile, or tags) must be defined. If this parameter is applied, then the solution tree is not exported. If this parameter is not used (which is the default), then both the tags and solution tree are exported.
root	The path for your solution's root. The path must begin with two colon characters (::). The default is the top-level solution node.
pathsfile	The single path of an object in the model per line. It can be any valid filename. **Important**: Redundant export information is produced if the `nochildren` option is not specified and the `pathsfile` specified is a child of another.
nochildren	When this is included in the syntax, only the root component is exported. If excluded, then the root component and all it's "children" are exported.
nocollections	When this is included in the syntax, then the global collections are not exported. If excluded, then all global collections are exported.

If you use a pathsfile that contains global collections and `nocollections` is not added to the command-line, then exportsoln exports all global collections AND then additionally re-exports the collections in the pathsfile.

For example, if the following pathsfile is used and the `nocollections` parameter is not added to the command-line:

```
::COLLECTIONS.Collection4
::COLLECTIONS.Collection5
```

Then note the collections exported below. Notice that collections 4 and 5 are exported twice.

```
::COLLECTIONS.Collection1
::COLLECTIONS.Collection2
::COLLECTIONS.Collection3
::COLLECTIONS.Collection4
::COLLECTIONS.Collection5
::COLLECTIONS.Collection4
::COLLECTIONS.Collection5
```

Importing a Solution

Syntax:

```
importsoln
"file=<filename.xml>"
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Example 1: *Given*: Security is disabled.
The solution acmesoln.xml is imported.

```
>xhqci importsoln "file=C:\Documents and Settings\jsmith\Application
Data\XHQ\XHQ devClient\repos\acmesoln.xml"
```

Example 2: *Given*: Security is enabled.
The solution acmesoln.xml is imported.

```
>xhqci importsoln "file=C:\Documents and Settings\jsmith\Application
Data\XHQ\XHQ devClient\repos\acmesoln.xml" "eshost=server1"
"sshost=server1" "uid=jsmith" "pw=enterprise"
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for importsoln

Parameter	Description
file	The full system path for the solution you are importing. The supported file format is .xml. Typically, the .XML file is exported to the `\Documents and Settings\<user>\Application Data\XHQ\XHQ devClient\repos` directory.

When importing a global collection, if the current global collection has "connect" configured, but the one in the import does not, the existing "connect" configuration should be removed after the import. To remove a configuration from an item, it must be explicitly configured with the value **%none**. See the following topic, *Removing a Configuration*, for details.

Removing a Configuration

In general, the solution import replaces any configuration associated with a member, collection, or group, if the configuration element is specified in the import file. If the configuration element is not specified (not part of the import), then it is left alone and may need to be removed. To remove any type of configuration (expression, connection, security, and so forth) after a solution import, explicitly set the item to be removed to the **%none** value.

The following example shows an Integer member with a connection configuration when exported:

```
<XHQ:DataMember
   xhqPath="::PiPHD.at_Interpolation.PHD_Int_linear"
   xhqComponent="at_Interpolation"
   usage=""
   connectionGroup="PHD_15">
   <XHQ:ConnectProp
      name="TAG_NAME"
      value="string:&quot;XHQINT1&quot;"
   />
   <XHQ:ConnectProp
      name="TAG_ATTRIBUTE"
      value="string:&quot;&lt;None&gt;&quot;"
   />
   <XHQ:ConnectProp
      name="POLL_PERIOD"
      value="int:0"
   />
</XHQ:DataMember>
```

To remove the connection configuration, it is set to **%none**:

```
<XHQ:DataMember
   xhqPath="::PiPHD.at_Interpolation.PHD_Int_linear"
   xhqComponent="at_Interpolation"
   usage=""
   connectionGroup="%none">
</XHQ:DataMember>
```

Configuring the Tag Time-series Data Type By Import

To configure a Tag's time-series data type (for example, "Normal," "Time Series Linear," or "Time Series Stepped") through an import (using, for example, **`xhqci importsoln`**), note the following:

- For the XHQCache connector, the associated `XHQ:ConnectProp` node name is TS_INTERP_TYPE.
- For SQL-based connectors (such as Oracle, MSSQL, or SQL Server), the associated `XHQ:ConnectProp` node name is INTERP_TYPE.

So, should you get the following error:

`ERROR: SolnRoot.SubNode1.Tag1: Connection attributes failed to import: Attribute TS_INTERP_TYPE: invalid attribute,`

try INTERP_TYPE in the import file instead of TS_INTERP_TYPE.

Exporting Components

Syntax:

```
exportmodel
"file=<ZIP file>"
["root=<component name>" [nochildren]] {default=root component}
[noviews] [notemplates]
[dir=<directory name>]
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Example 1: *Given*:Security is disabled.
Only the component myRoot is exported.
No views are exported.

```
>xhqci exportmodel "file=C:\Documents and Settings\jsmith\Application
Data\XHQ\XHQ devClient\repos\acmemodel.zip" "root=myRoot" nochildren
noviews
```

Example 2: *Given*:Security is enabled.
The component myRoot and all "children" of myRoot is exported.
No views are exported.

```
>xhqci exportmodel "file=C:\Documents and Settings\jsmith\Application
Data\XHQ\XHQ devClient\repos\acmemodel.zip" noviews "rshost=router1"
"sshost=server1" "uid=jsmith" "pw=enterprise"
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line to Export Parameters

Parameter	Description
file	The full system path for the ZIP file you are exporting.
root	The name of the root component to export.
nochildren	This is only available if a root parameter has been given. When this is included in the syntax, only the root component is exported. If excluded, then the root component and all it's "children" are exported.
noviews	When this is included in the syntax, no views are exported. If excluded, then all views are exported.
notemplates	When this is included in the syntax, no templates are exported. If excluded, then all templates are exported.
dir	The name of the directory to export the model to. **Note:** The **`dir`** parameter produces the same result as the **`file`** parameter; the only difference is that all the information is in a directory instead of a ZIP file.

Importing Components

In order to **import components**, you must have Model Importing AND Component Editing rights.

In order to **import components with their views**, you must have Model Importing, Component Editing AND View Editing rights.

Syntax:

```
importmodel
"file=<ZIP file>"
["root=<component name>" [nochildren]] {default=all components}
[[noviews] [notemplates]] | [rollback]]
[append | delete]
[dir=<directory name>]
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Example 1:

Given: Security is disabled.
The component myRoot and all "children" of myRoot are imported.
No views are imported.

```
>xhqci importmodel "file=C:\Documents and
Settings\jsmith\Application Data\XHQ\XHQ
devClient\repos\acmemodel.zip" "root=myRoot" noviews
```

Example 2:

Given: Security is enabled.
Only the component myRoot is imported.

```
>xhqci importmodel "file=C:\Documents and
Settings\jsmith\Application Data\XHQ\XHQ
devClient\repos\acmemodel.zip" "root=myRoot" nochildren rollback
append "eshost=server1" "sshost=server1" "uid=jsmith"
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for importmodel

Parameter	Description
file	The full system path for the ZIP file you are importing.
root	The name of the root component to import.
nochildren	This is only available if a root parameter has been given. When this is included in the syntax, only the root component is imported. If excluded, then the root component and all it's "children" are imported.
noviews	When this is included in the syntax, no views are imported. If excluded, then all views are imported.
notemplates	When this is included in the syntax, no templates are imported. If excluded, then all templates are imported.
append	Component members included in the model import file will either replace (existing

Parameter	Description
	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.
dir	The name of the directory to import the model from. **Note:** The `dir` parameter produces the same result as the `file` parameter; the only difference is that all the information is in a directory instead of a ZIP file.

Exporting Component Roles

Currently, only the export/import of component roles using the command-line utility is supported. Through this utility, you can export component role definitions to an .XML file that is separate from the XHQ Workbench export file.

The export file includes all roles assigned for all components. The Model Admin role is implicitly included in the export file against all components. This means that even though the export file does not explicitly include the Model Admin role, it is automatically assumed that the role is assigned to all components.

Syntax:

```
exportroles
"file=<XML file>"
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Example 1: *Given*: In this example, a qualified path is given for the location of the .XML output file.

```
>xhqci exportroles "file=C:\Documents and Settings\jsmith\Application
Data\XHQ\XHQ devClient\repos\roles.xml" "eshost=localhost"
"sshost=localhost" "uid=jsmith" "pw=enterprise"
```

Example 2: *Given*: In this example, "roles.xml" will be saved under the current working directory (for example, c:).

```
>xhqci exportroles "file=roles.xml" "eshost=localhost"
"sshost=localhost" "uid=johnsmith" "pw=enterprise"
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line to Export Roles Parameters

Parameter	Description
file	The full system path for the .XML file you are exporting. (See Example 2 above for an alternative.)

Importing Component Roles

Currently, only the export/import of component roles using the command-line utility is supported. Through this utility, you can import component role definitions from an .XML file that is separate from the XHQ Workbench or import file. The import file contains all of the components and the role assignment for each component. The Model Admin role is implicitly assigned to all components. This means that even though the import file does not explicitly include the Model Admin role, it is automatically assumed that the role is assigned to all components.

If an invalid role is encountered in the import file, then the role is ignored and the import process continues. If all roles for a given component are invalid, then the only role assigned to that component will be the Model Admin role.

You must have both **Role Assigning** and **Model Importing** rights to import component roles.

Syntax:

```
importroles
"file=<XML file>"
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Example 1: *Given*: In this example, a qualified path is given for the location of the .XML output file.

```
>xhqci importroles "file=C:\Documents and Settings\jsmith\Application
Data\XHQ\XHQ devClient\repos\roles.xml" "eshost=localhost"
"sshost=localhost" "uid=jsmith" "pw=enterprise"
```

Example 2: *Given*: Security is enabled. In this example, "roles.xml" will be saved under the current working directory (for example, c:).

```
>xhqci importroles "file=roles.xml" "eshost=localhost"
"sshost=localhost" "uid=jsmith" "pw=enterprise"
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line to Import Roles Parameters

Parameter	Description
file	The full system path for the .XML file you are importing. (See Example 2 above for an alternative.)

Exporting Roles Mappings

Syntax:

```
exportrolemappings
"file=<XML file>"
["<comma separated names of items to export> | ALLCONFIGS"]
{default ALLCONFIGS}
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for exportrolemappings

Parameter	Description
file	The filename of the XML file exported.
ALLCONFIGS	All role mappings are exported.

Importing Role Mappings

The `importrolemappings` command allows the mapping of Windows User Groups to XHQ Security Roles. The specific permissions granted to each of these roles may also be specified when those mapping are made. This is done through the XML file that is provided as a parameter on the command-line.

For more information on XHQ Security, see the section, *Security Access and Privileges*.

For information on how the XML document is structured, see the topic, *Exporting and Importing Component Roles*.

Syntax:

```
importrolemappings
"file=<XML file>"
["<comma separated names of items to import> | ALLCONFIGS"]
{default ALLCONFIGS}
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for importrolemappings

Parameter	Description
file	The filename of the XML file exported.
ALLCONFIGS	All role mappings are imported.

Exporting View Tile Composer (VTC) Views

Non-reusable VTC views can only be exported by using this xhqci command (or through the XHQ Solution Builder).

When using the XHQ Workbench to export VTC views, only reusable views can be exported, regardless of whether they are published or private.

Syntax:

```
exportvtcviews
"file=<ZIP filename>"
["components=<component views>"]
["paths=<view paths>"]
[private | published | all] {default=all}
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Example 1: *Given*: Export ALL private VTC views for all users, and all components (reusable views), and all paths (single views).

```
>xhqci exportvtcviews "file=vtcExport.zip" private
```

Example 2: *Given*: Export ALL published AND private VTC views for all users, and all components (reusable views), and all paths (single views).

```
>xhqci exportvtcviews "file=vtcExport.zip" all
```

Or, because the "all" parameter is the default, you can simply enter:

```
>xhqci exportvtcviews "file=vtcExport.zip"
```

Example 3:

```
>xhqci exportvtcviews "file=vtcExport.zip"
"components=Component1,View1,View2; Component2; Component3,View8"
"paths=::Enterprise, ViewA, ViewB; ::Enterprise.SIMPoint" published
"rshost=Router1" "sshost=server1" "uid=johnsmith" "pw=enterprise"
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for exportvtcviews

Parameter	Description
file	The filename of the resulting .ZIP file to export.
components	A semi-colon separated list of components for which to export the re-usable VTC views. For each component listed, you can also specify which view(s) to export. Note the following example format. *Example 1::* `xhqci exportvtcviews "file=vtcExport.zip" "components=Component1,View1,View2; Component2; Component3,View8" published` For `Component1`, `View1` and `View2` are exported, provided that they are published VTC views. For `Component2`, all published VTC views for this component are exported.

Parameter	Description
	For `Component3,View8` is exported, provided that it is a published VTC view. If the specified view is private, then the user name of the view owner must be specified in curly brackets. Note the example below. If the specified view is marked for deletion, then append the `#del` text to the end of the view name. Note the example below. *Example 2::* `xhqci exportvtcviews "file=vtcExport.zip" "components=Measurement,View1 {acme\johnsmith}#del;Calendar"` If a list of views is not defined for the component, then all the VTC reusable views are exported for that given component.

paths	A semi-colon separated list of model paths for which to export the single VTC views. For each absolute path listed, you can also specify which view(s) to export. Note the following example format.
	Example 1::
	`xhqci exportvtcviews "file=vtcExport.zip" "paths=::Enterprise, ViewA, ViewB; ::Enterprise.SIMPoint" published`
	For the first absolute path listed (`::Enterprise`), `ViewA` and `ViewB` are exported, provided that they are published VTC views.
	For the second absolute path listed (`::Enterprise.SIMPoint`), all published VTC views in this path are exported.
	If the specified view is private, then the user name of the view owner must be specified in curly brackets. Note the example below.
	If the view is marked for deletion, then a append the `#del` text to the end of the view name. Note the example below.
	Example 2::
	`xhqci exportvtcviews "file=vtcExport.zip" "components=Measurement,View1 {acme\johnsmith}#del;Calendar" "paths=::Enterprise,View5#del"`
	If a list of views is not defined for the model path, then all the VTC single views are exported for that given path.

private | published | all	The view access type selectors are: private – Exports only private VTC views. published – Exports only published VTC views. all – Exports ALL VTC views. This is the default if no view access type selector is defined. *Example:* `xhqci exportvtcviews "file=vtcExport.zip" "components=Component1,View1,View2" "paths=::Enterprise.SIMPoint" published` In this example, the "published" view type selector is defined. However, if `View1` is a private view, it is not exported.

Importing View Tile Composer (VTC) Views

To import private VTC views, you have to belong to the **Solution Admins** predefined role.

Although VTC Publishing and VTC Personalization both grant rights to run the `importvtcviews` command, the private views of the user are not imported.

Syntax:

```
importvtcviews
"file=<ZIP filename>"
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Example:

```
>xhqci importvtcviews "file=vtcExport.zip"
"rshost=Router1" "sshost=server1" "uid=johnsmith" "pw=enterprise"
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for importvtcviews

Parameter	Description
file	The filename of the .ZIP file to import.

Purging View Tile Composer (VTC) Views

Purges (deletes) all VTC views from the system.

Syntax:

```
purgevtcviews
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Revert Units of Measure

Reverts to the original/default, system-defined, units of measure, by deleting all user defined units of measure.

Syntax:

```
revertuom
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Exporting Units of Measure (UoM)

Syntax:

```
exportuom
"file=<filename.xml>"
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Example:

```
>xhqci exportuom "file=uomExport.xml"
"rshost=Router1" "sshost=server1" "uid=johnsmith" "pw=enterprise"
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for exportuom

Parameter	Description
file	The XML file that contains the Units of Measure to be exported

Importing Units of Measure (UoM)

Syntax:

```
importuom
[replace]
"file=<filename.xml>"
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for importuom

Parameter	Description
replace	If the **`replace`** parameter is used, the categories and units present in the current Model server are all deleted and replaced with the ones in the XML file. If the **`replace`** parameter is not used, then the import operation performs a modification, renaming, or deletion for the categories and units that are present in the Model server.
file	The XML file that contains the Units of Measure to be imported.

Exporting Connector Definitions

Syntax:

```
exportconndefs
"file=<XML filename>"
["<comma separated names of items to export> | ALLCONFIGS"]
{default ALLCONFIGS}
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for exportconndefs

Parameter	Description
file	The filename of the XML file to be exported.
ALLCONFIGS	All connector definitions are exported.

Importing Connector Definitions

Syntax:

```
importconndefs
"file=<XML filename>" | "file=<ZIP filename>" | "dir=<directory name>"
["<comma separated names of items to import> | ALLCONFIGS"]
{default ALLCONFIGS}
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for importconndefs

Parameter	Description
file	The filename of either the XML or ZIP file to be imported.
dir	The name of the directory to import the connector definitions from. **Note:** The **`dir`** parameter produces the same result as the **`file`** parameter; the only difference is that all the information is in a directory instead of a ZIP file.
ALLCONFIGS	All connector definitions are imported.

Exporting Connector Groups

Syntax:

```
exportconngroups
"file=<XML file>"
["<comma separated names of items to export> | ALLCONFIGS"]
{default ALLCONFIGS}
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for exportconngroups

Parameter	Description
file	The filename of the XML file exported.
ALLCONFIGS	All connector groups are exported.

Importing Connector Groups

Syntax:

```
importconngroups
"file=<XML file>"
["<comma separated names of items to import> | ALLCONFIGS"]
{default ALLCONFIGS}
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for importconngroups

Parameter	Description
file	The filename of the XML file to be imported.
ALLCONFIGS	All connector groups are imported.

Exporting Connector Processes

Syntax:

```
exportconnprocs
"file=<XML file>"
["<comma separated names of items to export> | ALLCONFIGS"]
{default ALLCONFIGS}
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for exportconnprocs

Parameter	Description
file	The filename of the XML file exported.
ALLCONFIGS	All connector processes are exported.

Importing Connector Processes

Syntax:

```
importconnprocs
"file=< XML file>"
["<comma separated names of items to import> | ALLCONFIGS"]
{default ALLCONFIGS}
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for importconnprocs

Parameter	Description
file	The filename of the XML file to be imported.
ALLCONFIGS	All connector processes are imported.

Exporting Solution Definitions

Syntax:

```
exportsolndefs
"file=<XML file>"
["<comma separated names of items to export> | ALLCONFIGS"]
{default ALLCONFIGS}
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for exportsolndefs

Parameter	Description
file	The filename of the XML file exported.
ALLCONFIGS	All solution definitions are exported.

Importing Solution Definitions

Syntax:

```
importsolndefs
"file=< XML file>"
["<comma separated names of items to import> | ALLCONFIGS"]
{default ALLCONFIGS}
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for importsolndefs

Parameter	Description
file	The filename of the XML file to be imported.
ALLCONFIGS	All solution definitions are imported.

Importing Alert Definitions

This command imports alert definitions configured in an XML file into XHQ ANS.

Syntax:

```
ansimport
"file=<XML file>"
[autoCreateLists]
[updateDefsById]
[importSubsc]
[importLOConfigs]
["anshost=<ANS host name>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for ansimport

Parameter	Description
file	The filename of the XML file to be imported.
autoCreateLists	Automatically creates Definition and Subscription Lists.
updateDefsById	Updates Definitions and Subscriptions using definition IDs.
importSubsc	Imports Subscriptions.
importLOConfigs	Imports Lost Opportunity configurations.
anshost	The ANS host name. If `anshost` is omitted, the value is then determined by: 1. The `anshost` parameter, if specified; 2. The `XHQCI_HOST` environment variable; 3. The Server Access Utility setting; 4. And, if none of the above apply, `localhost`.

Exporting Connection Groups to be Deleted

A file, containing a list of connection groups that are to be deleted, is exported.

Syntax:

```
exportdelconngroups
"file=<XML file>"
["conngroups=<connection groups>"]
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for exportdelconngroups

Parameter	Description
file	The filename of the XML file exported.
conngroups	A comma-separated list of connection groups to be deleted.

Exporting Connection Processes to be Deleted

A file, containing a list of connection processes that are to be deleted, is exported.

Syntax:

```
exportdelconnprocs
"file=<XML file>"
"process=<connection processes to be deleted>"
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for exportdelconnprocs

Parameter	Description
file	The filename of the XML file exported.
process	A **comma-separated list** of connection processes to be deleted.

Exporting the Model to be Deleted

A file, containing the Model (components, component members, and views) that is to be deleted, is exported.

Syntax:

```
exportdelmodel
"file=<ZIP file>"
["components=<components to be deleted>"]
["members=<members to be deleted>"]
["views=<views to be deleted>"]
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for exportdelmodel

Parameter	Description
file	The filename of the .ZIP file exported.
components	A **comma-separated** list of components to be deleted.
members	**Semi-colon-separated** lists of components members to be deleted. The first element in each list is the component name, followed by a **comma-separated** list of each component member(s) to be deleted. Note the following example format. *Example:* `xhqci exportdelmodel "file=modelDelete.zip" "members=Component1,MemberA,MemberB; Component2,MemberC; Component3` For Component1, MemberA and MemberB are to be deleted. For Component2, only MemberC is to be deleted. For Component3, all members are deleted.
views	**Semi-colon-separated** lists of components views to be deleted. The first element in each list is the component name, followed by a **comma-separated** list of each component view(s) to be deleted. Note the following example format. *Example:* `xhqci exportdelmodel "file=modelDelete.zip" "views=Component1,View1,View2; Component2,View3; Component3` For Component1,View1 and View2 are to be deleted. For Component2, only View3 is to be deleted. For Component3, all views are deleted.

For more information, go to the topic, *Import\Export Delete* located in the XHQ Developer's Guide.

Exporting Deleted Role Mappings

A file, containing a list of role mappings that are to be deleted, is exported.

Syntax:

```
exportdelroles
"file=<XML file>"
["<comma separated list of role names to be deleted> |
rolesfile=<filename>"]
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for exportdelroles

Parameter	Description
file	The filename of the XML file exported.
rolesfile	The role name to be deleted.

For more information, go to the topic, *Import\Export Delete* located in the XHQ Developer's Guide.

Exporting Solution Collections to be Deleted

A file, containing the solution collections that are to be deleted, is exported.

Syntax:

```
exportdelsoln
"file=<XML file>"
["colslist=<solution collections to be deleted>"
     | "colsfile=<filename>"]
["pathslist=<paths to be deleted>" | "pathsfile=<filename>"]
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for exportdelsoln

Parameter	Description
file	The filename of the XML file exported.
colslist	A comma-separated list of solution collection to be deleted.
colsfile	The solution collection to be deleted.
pathslist	A comma-separated list of instances (objects) to be deleted, using the full pathname. See the topic, *Deleting Instances Created in Extended Namespace*, for more information.
pathsfile	The file that contains a list of paths to be deleted; one path per line. See the topic, *Deleting Instances Created in Extended Namespace*, for more information.

For more information, go to the topic,*Import\Export Delete* located in the XHQ Developer's Guide.

Deleting Instances Created in Extended Namespace

Use the **`exportdelsoln`** command to designate which objects are to be deleted during the import of the resulting XML file.

Syntax:

```
exportdelsoln
"file=<XML file>"
["pathlist=<instance pathnames>" | "pathfile=<filename>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for exportdelsoln

Parameter	Description
file	The filename of the XML file exported.
pathlist	A comma-separated list of instances (objects) to be deleted, using the **full pathname**. *Example*: `"pathlist=::Enterprise.View1.Object2,::Enterprise.View5.Object3"`
pathfile	The file that contains a list of paths to instances that are to be deleted; **one path per line**. *Example*: `"pathfile=f:\del_ins.txt"`
colslist	A comma-separated list of solution collection to be deleted. See the topic, *Exporting Solution Collections to be Deleted*, for more information.
colsfile	The solution collection to be deleted. See the topic, *Exporting Solution Collections to be Deleted*, for more information.

Exporting a Database

For dbexport

To initiate DB export, use the following syntax.

Syntax:

```
dbexport
["force=true"]
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

An Update To Latest (UTL) does not disable dbexport. A DB export automatically follows a UTL when namespace changes have occurred.

For dbexportstat

To **query DB export status**, use the following syntax.

Syntax:

```
dbexportstat
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Setting Configuration Import Complete

To signal that the configuration import is complete, use the following syntax.

Syntax:

```
configimportcomplete
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.
For information on how to invoke `configimportcomplete` from the `XHQTagSync.bat` script, go to the topic, *Step 5 – Cycle Completion*, located in the XHQ Solution Design and Architecture.

Updating a Solution

This command updates a solution to the latest component model.

Syntax:

```
updatesoln
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Example 1: *Given*: Security is enabled.

```
>xhqci updatesoln "rshost=Router1" "sshost=server1" "uid=johnsmith"
"pw=enterprise"
```

Example 2: *Given*: Security is disabled; therefore, eshost=localhost and sshost=localhost.

```
>xhqci updatesoln
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Solution Information

The following xhqci command helps you determine:

- If an XHQ Server is a member of a distributed solution;
- The solution's root path;
- If the solution object is valid or exists.

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Is the XHQ Server a member of a distributed solution?

Syntax:

```
solutioninfo getDistributionMode
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Returns:

- If it is a member of a distributed solution, then the following output appears:

  `Mode=DISTRIBUTED; Realm=realmName`

  Where **`realmName`** is the name of the realm to which the server belongs.
- Else, the following output appears:

  `Mode=STANDALONE`

What is the solution's root path?

Syntax:

```
solutioninfo getRootPath
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Returns:

```
Root path=::AcmeEnterprise
```

Is the solution object valid? Does it exist?

Syntax:

```
solutioninfo validateObject
"path=<object Path>"
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Returns:

- If it is **valid** or if it **exits**, then the following appears:

  `Object ID=objectID`
- Else, if the object is not valid or does not exist, the following output appears:

  `Object ID Not Found`

Command-Line Parameters for solutioninfo validateObject

Parameter	Description
path	The path of an object in the solution.

Model Information

The following xhqci command outputs a comma-separated list of XHQ Workbench pages.

Syntax:

```
modelinfo getPages
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Results:

```
Pages: pageName1, pageName2, pageName3, ... pageNameN
Total pages returned = N
```

Where **`pageNameN`** is the name of the page defined in the model and **`N`** is the total number of pages in the model.

Refreshing Roles

Enables to you to set a schedule to refresh XHQ roles. To execute this command, you must have Role Admins permissions.

Syntax:

```
rolesadmin
[refresh | setconfig=HH:MM;nn | getconfig]
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters

Parameter	Description
refresh	Forces a refresh of XHQ roles. **Note**: The xhqci rolesadmin refresh command is also available through a *Web Service*.
setconfig	Sets a schedule when the first refresh of the day occurs and refresh frequency. The default is one refresh daily at midnight. The format is `HH:MM;nn` • "HH" is the hour in 24-hour format • "MM" is the minutes • "nn" is the number of refreshes per day; 0 (zero) means no automatic refresh. The number of refreshes should be kept to a minimum to avoid unnecessary cache purging and refreshing.
getconfig	Retrieves current settings.

Renaming Components

Should you have problems with component or component member names, a renaming utility is provided with XHQ that allows batch operation to either:

- Rename any occurrence in all components with a specific member name;
- Or, rename any occurrence in a specific component with a specific member name.

Syntax:

```
rename
"file=<filename>"
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line to Rename Parameters

Parameter	Description
file	The full system path for the file. Supported line formats: <comp name> <new comp name> <comp name> <old member> <new member> <*> <old member> <new member>

Typically, the only parameter you will need is "file=<filename>". However, if security is enabled, the "uid" and "pw" parameters are required. And, if the server is not on the local machine, the standard hostname parameters are needed as well.

The *rename* command processes an ASCII file. Each line on the file contains 2 or 3 strings.

To rename components using the command-line

1. From a DOS prompt, enter the location for the `\Program Files\XHQ\XHQ Server\bin` directory.
2. At the prompt, type the rename model help command.

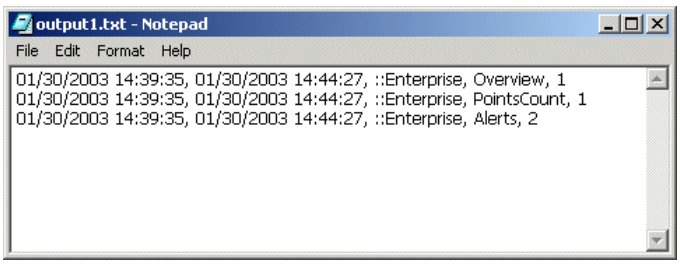
   Example: `xhqci help rename`

3. Do one of the following.
   - To rename a component, use:
     `<old component name> <new component name>`
   - To rename a component's member:
     `<component name> <old member name> <new member name>`
   - To globally replace component member names:
     `<*> <old member name> <new member name>`
     This searches all components, and any matching member name is replaced with the new one.

Using ViewStats

This command creates an output text file that contains a list of all the views that have been visited prior to running the viewstats utility, as well as the number of times the view was visited.

Your solution must be running in order to use the viewstats command line.



Example: – ViewStats Output File

As seen in the example above, the output text file consists of the following information.

Column	Description	Example
1	Initially, this is the date and time the solution server was started. However, after the first viewstat retrieval, this will display the date and time that the last retrieval occurred.	01/30/2003 14:39:35
2	The time that the viewstat retrieval was executed.	01/30/2003 14:44:27
3	The fully qualified path name for the object.	::Enterprise.DDEpoint
4	The view name.	Alerts
5	The number of times the view was loaded.	2

Syntax:

```
viewstats
"file=<filename.txt>"
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Examples::

```
>xhqci viewstats "file=output1.txt"
>xhqci viewstats "file=%XHQ_SERVER_REPOS%\output2.txt"
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for ViewStats

Parameter	Description
file	The output text file. This file is placed in the same directory where the command is executed. For example, if you run the command from `C:\Program Files\XHQ`, then the output file is placed in the `C:\Program Files\XHQ` directory. *Example:* From a command prompt, the following command line is entered:

Parameter	Description

	`C:\Program Files\XHQ>xhqci viewstats "file=output1.txt"`
	
	The output text file is placed in the `C:\Program Files\XHQ` directory.
	
	You can also use the `file` **parameter** to indicate a specific directory to place the output text file.
	
	Example:
	
	From a command prompt, the following command line is entered:
	
	`C:\Program Files\XHQ>`
	`xhqci viewstats "file=%XHQ_SERVER_REPOS%\output2.txt"`
	
	The output text file is placed in the location specified by the environment variable `%XHQ_SERVER_REPOS%`, which is typically `C:\XHQ\data\repos`.
	
	Important: If a file already exists with the same file name, it will be overwritten with the new viewstats information.

To use the viewstats command-line

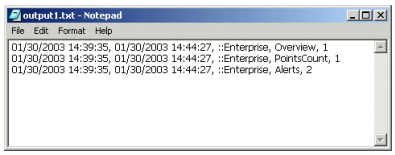
1. From a DOS prompt, enter the location for the `\Program Files\XHQ\XHQ Server\bin` directory.

   Your solution must be running in order to use the viewstats command line.

2. After the prompt, type the viewstats command.

   Example: `xhqci viewstats "file=output1.txt"`

3. Click **ENTER**.
4. Go to the `\Program Files\XHQ\XHQ Server\bin` directory.
5. Open your output text file.
   This launches a text editor with the viewstats output.



Compressing the Solution

This command compresses the solution.

Syntax:

```
compactsoln
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Example 1: *Given*: Security is enabled.

```
>xhqci compactsoln "rshost=Router1" "sshost=server1" "uid=johnsmith"
"pw=enterprise"
```

Example 2: *Given*: Security is disabled; therefore, eshost=localhost and sshost=localhost.

```
>xhqci compactsoln
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Starting/Stopping a Connector

This command allows you to start or stop a connector process from the command-line. With this feature, you can manage the order that your connect processes start. This reduces the overall startup time.

Syntax:

```
connector
"conn=<connector_name>" [start|stop]
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Example 1: *Given*: Starting the connector process "phd_connector" and security is enabled.

```
>xhqci connector "conn=phd_connector" start "rshost=Router1"
"sshost=server1" "uid=johnsmith" "pw=enterprise"
```

Example 2: *Given*: Stopping the connector process "opc1" and security is disabled; therefore, eshost=localhost and sshost=localhost.

```
>xhqci connector "conn=opc1" stop
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for Connector Start/Stop

Parameter	Description
conn	The connector process name.
start | stop	Enter start or stop to either start or stop the connector process.

Restoring Storage Mode

To restore the storage mode, use the following syntax.

Syntax:

```
restorestgmode
["server=model"|"server=soln"]
["eshost=<enterprise hostname>"|"rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for restorestgmode

Parameter	Description
server	Restores the storage mode for either the model or solution server. *Options*: • model • soln

Looking at the XHQ License

When this command is executed, a message is sent notifying the solution server that a new license file is in place and ready for examination.

For more information on the XHQ License and how to active an updated license without an XHQ restart, go to the topic, *License Manager*, located in the XHQ Installation Guide.

Syntax:

```
examinelicense
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Managing Data Quality

This command allows you to suppress (or restore) data quality for specific members of the XHQ Solution. By default, it acts on the Tag collection by suppressing a quality change to that collection. This prevents intermittent bad quality from being propagated to all the tags. You can also suppress quality management for changes caused by expiration.

The `managedataquality` command works with any solution item you can address by path or alias.

Syntax:

```
managedataquality
["item=<member path>"]
["mask=all" | "mask=expiration"]
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Example:

```
>xhqci managedataquality "item=::acme1.plant1.pump1" "mask=all"
"rshost=Router1" "sshost=server1" "uid=johnsmith" "pw=enterprise"
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for managedataquality

Parameter	Description
item	This is the Tag collection. Otherwise, use the fully qualified path of the solution member whose quality should be managed. *Default:* item = the Tag collection
mask	• **`all`** This suppresses all quality except GOOD. This is the default value. • **`expiration`** This only suppresses quality changes caused by expired values. *Default:* mask = all

Updating Client Notification

This command allows you to either clear or update the current "client notification" for XHQ runtime clients. The file options ("file" and "localfile") define an existing file from which the message will be read.

The existing message file must reside in either the local (client) machine or at the location specified by the environment variable `%XHQ_SERVER_REPOS%` on the Enterprise Server (which is typically `XHQ\XHQ Server\repos`).

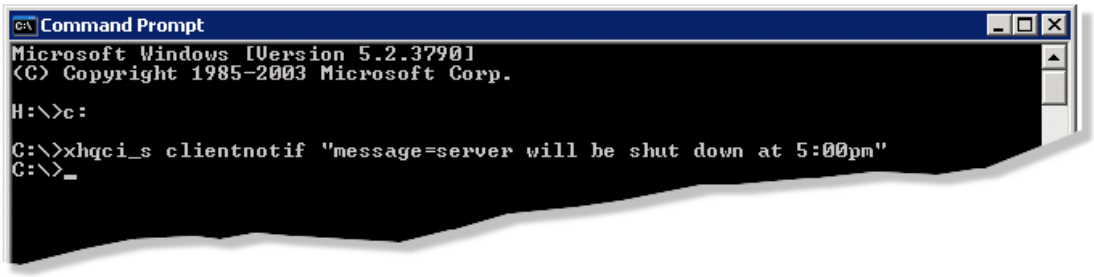
Syntax:

```
clientnotif
"message=<text of message>"
"file=<filename>"
"localfile=<localfilepath>"
reset
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

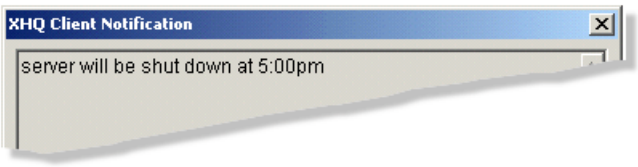
Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for clientnotif

Parameter	Description
message	Any text entered after the equal sign is sent to the solution server, which then becomes the new notification message that is broadcast to clients.
file	Enter the name of the file that contains the notification message. **Important**: This file, which is accessed by the solution server, must reside in the `%XHQ_SERVER_REPOS%` directory (which is typically `C:\XHQ\data\repos`).
localfile	The xhqci command reads the contents of the file identified by this parameter and sends the text to the solution server. This, in turn, becomes the new notification message that is broadcast to runtime clients.
reset	Clears the current message. **Note**: The default state is not to broadcast a notification message to runtime clients.



Setting a client notification



Notification Example:

Performing a Tag Reconciliation

This feature reconciles the internal list of available tag objects (within the XHQ Solution Server) with the list of tag records shown by the tag collection. To match these two sets, tag objects are created or deleted.

Syntax:

```
reconciletags
[reset]
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for reconciletags

Parameter	Description
reset	This option causes any existing tags to be reconfigured with current configurations from the collection.

Using Collection Tools

This command provides the following tools for use with XHQ global collections: **Update** and **Dependencies**.

The collectiontool command does not work with pass-through or un-/mis-configured collections.

For Collection(s) Update

Use this tool to immediately update one or more global collections.

Syntax:

```
collectiontool update
["style=<s>"]
collname [collname...]
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Any argument in the form `"arg=<val>"` can also be written `arg:<val>`. For example, `collectiontool update "style=full" Test1` can also be written `collectiontool update style:full Test1`.

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for collectiontool update

Parameter	Description
update	The update tool.
style	The style parameter controls how the collection is updated. Use one of the following options: • **full** A full style update is a complete update of the collection, as if the collection had just been re-configured or the connecter the collection uses had just started. • **normal** A normal style update is simply an out-of-band request to perform a normal poll. The **default** is normal. **Note:** For each style, the exact action taken during an update depends on what type of collection is used. For polled (normal) collections or XHQCache connector collections, both the normal and full style updates result in the same actions. However, for a flag or delta collection, the normal and full style updates take different actions.
collname	The global collection name. **Note:** For multiple global collections, use a **space** to separate each collection name. *Example:* `xhqci collectiontool update Measurements AllTags`

For Collection Dependencies

Use this tool to generate an XML file/output of collection dependencies.

Syntax:

```
collectiontool dependencies
["file=<XML file>"]
["bom=<b>]
["dtd=<d>"]
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Again, parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for collectiontool dependencies

Parameter	Description
file	The path to the XML output file. The default is to console.
bom	Indicates whether or not to include the UTF-8 BOM (Byte Order Mark) in the XML output. **Note:** This parameter is dependant on the file parameter. So, unless the file parameter is used, the bom option is ignored. Use one of the following options: • **include** • **none** The default is to include the UTF-8 BOM in the XML output. *Example:* (using the unquoted argument technique described in the previous page) `xhqci collectiontool dependencies file:c:\colldeps.xml bom:none` In this example, the `colldeps.xml` file is generated with no BOM at the beginning.
dtd	Indicates whether or not to include the DTD in the output. Use one of the following options: • **include** • **none** • **only** The default is to include the DTD in the output. *Example:* (using the unquoted argument technique described in the previous page) `xhqci collectiontool dependencies dtd:only` In this example, only the DTD of the XML is printed to the console.

If no arguments are defined for this command-line, then the XML output is printed to the console.

Online Backup

Backup the XHQ solution without having to shutdown the server.

Syntax:

```
backup
"directory=<backup directory>"
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Security Requirements

In order to execute the `backup` command, you must be an XHQ Solution Admin as well as have Windows user access to both the physical XHQ repos folder and the destination folder.

XHQ Service (and all XHQ processes) must run as a specific user, as recommended in this guide. If XHQ Service is configured to run as Network or security is turned off in the XHQ repository (which are both NOT recommended), then the `xhqci backup` command is considered unsafe in a network environment because authorization cannot be performed properly. This could allow a malicious user to copy files from the repos folder on the XHQ Server machine to a folder on the network that is accessible from the XHQ Server machine. Certain files in the XHQ Repository folder may contain sensitive information that may be exposed to unauthorized users if copied in this manner.

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for backup

Parameter	Description
directory	This is the directory to which the repos backup files are copied. This command can either backup to a directory or create a ZIP file of the given directory. **Important:** Initially, this folder must be empty. The directory name must **not end with a slash**. You cannot use the root directory as the backup directory.

Consider the following examples:

Example 1 - Directory

```
>xhqci backup "directory=C:\XHQ\myXHQDataBackup"
```

The directory, `\XHQ\myXHQDataBackup`, is created and the backup files are copied to this directory.

Example 2 - Zip File

```
>xhqci backup "directory=C:\XHQ\myXHQDataBackup.zip"
```

In this second example, the directory, `\XHQ\myXHQDataBackup`, is created. After the backup files are copied to this directory, it is zipped, creating a **`myXHQDataBackup.zip`** file. Then, the original directory, `\XHQ\myXHQDataBackup`, is deleted.

Once a backup is initiated using this command, the Solution Server receives the backup request and the following sequence occurs:

1. If the specified backup directory does not exist, the Solution Server creates it. The given backup directory must be empty and have proper `WRITE` permissions for the Solution Server.
2. The Model Server receives the backup request and immediately blocks all component and view `EDIT` requests, and copies all files and subdirectories from the `\repos\Model` directory to the specified backup directory. Once the Model directory has been copied to the backup directory, all `EDIT` requests are unblocked, even if the Solution Server has not completed the backup process.
3. TagSync operations are delayed.
4. All `EDIT` requests are blocked from performing solution editing requests.

   Blocking these `EDIT` request does not block `READ` or `WRITE` operations.

5. All `*.dat` files from the active solution in the `\repos\Solution` directory are copied.

   If there are other solutions defined, they will NOT be backed up.

After the `\repos\Model` and `\repos\Solution` directories are backed up, the Solution Server then backs up all other files and subdirectories in the `\repos` directory.

You may also use the XHQ Administration Utility to backup your repos. For more information, see the topic, *Online Backup*, located in the XHQ Backup and Recovery Guide.

Setting a Tag Synchronization Processing Delay

Setting this command delays any Tag Sync request until the delay is expired.

Syntax: settagsyncdelay

[minutes]

["eshost=<enterprise hostname>" | "rshost=<router hostname>"]

["sshost=<solution hostname>"]

["uid=<user login name>" "pw=<user password>"]

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for settagsyncdelay

Parameter	Description
minutes	The number of minutes to delay the processing of pending Tag Sync operations. Enter a value from the range of **1** to **1440** (which is one day). **Note:** Enter a **0** (zero) or no argument to clear any delay setting. Enter a **-1** (minus one) to indefinitely delay all Tag Sync processing. When the -1 setting is used, the following message is written to the log: "WARNING: Tag Sync processing has been DISABLED – use settagsyncdelay command to resume."

Deleting Global Collections

This command deletes one or more global collections from the XHQ Solution.

Syntax:

```
deletecollections
[/deletetags]
collectionname1 [collectionname2...]
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Things to Note

- For multiple collections, use a **space** to separate each collection identifier.
- Collection identifiers are case-insensitive.
- If a collection identifier is incorrectly referenced, an error message appears and the incorrect collection entry is skipped.

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for deletecollections

Parameter	Description
/deletetags	This enables the deletion of the given tag collection.
collectionname	The name or numeric ID (OID) of the global collection to be deleted, also known as the *collection identifier*.

Deleting Classes

This command deletes one or more component classes from the XHQ Model.

Syntax:

```
deleteclasses
[classname...]
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for deleteclasses

Parameter	Description
classname	The name (or CID) of the component class to be deleted. **Note:** For multiple classes, use a **space** to separate each CID.

EXAMPLE: DELETING CLASSES

Given:

`ClassA`, `ClassB` (which is a subclass of `ClassA`), and `ClassC`

Delete all three classes.

In order to delete a class, all its subclasses must also be deleted.

1. At the prompt, type the following command-line:

   `xhqci deleteclasses ClassB ClassA ClassC`

2. Click **ENTER**.

Validating A Repos

This command performs a repos validation.

Syntax:

```
validaterepos
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Example Output: For a repos that is not corrupt.

```
Validating XHQ Model . . .
No corrupt components were found.

Validating XHQ Solution . . .
Found 0 corrupt collections.
Found 0 corrupt object components.
Found 0 corrupt configs.
```

Resolving Aliases to Paths

For valid aliases, this command returns the corresponding path associated with the member.

The generated output file is primarily used by the `xhqdr` utility. For more information on this utility, go to the topic, *Using xhqdr with IP.21*.

Syntax:

```
resolveAliases
"infile=<filename.txt>"
"outfile=<filename.xml>" | "outfile=<filename.csv>"
["csvdelim=<delimiter>"]
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Example 1:

```
>xhqci resolveAliases "infile=C:\myaliases.txt" "outfile=C:\aliasdb.csv"
"csvdelim=,"
```

Given myaliases.txt:

```
abc123
11FC103
```

Results in the following aliasdb.csv file:

```
abc123,::MySolution.Long.Path.To.Member
11FC103,::MySolution.OtherPath.To.OtherMember
```

Example 2:

```
>xhqci resolveAliases "infile=acmealiases.txt" "outfile=acmealiasdb.xml"
```

Given acmealiases.txt:

```
Acme2015
301CF
```

Results in the following acmealiasdb.xml file:

(Note, the DTD and namespaces were purposely left out of the following example.)

```
<?xml version="1.0" encoding="UTF-8"?>
<aliases>
<alias><name>Acme2015</name><path>::AcmeSolution.Long.Path.To.Member</path>
<alias><name>301CF</name><path>::AcmeSolution.OtherPath.To.OtherMember</path>
</aliases>
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Parameter	Description
infile	The input `.txt` file that lists the aliases, **one alias per line**.
outfile	The formatted text output file, in either `.xml` or `.csv` format, depending on which one you specify. It lists each alias and the corresponding path. **Note**: If there are undefined aliases, the output file is still generated and an error message appears.

Parameter	Description
cvsdelim	This optional command-line parameter defines the delimiter for the formatted `.csv` data. The **default** delimiter is the **TAB** character. **Important**: The caret symbol (^) and the percent symbol (%) cannot be used as delimiters.

Related Global Properties

There are two global properties related to the `resolveAliases` command: **`RequestMaxArrayLength`** and **`RequestMaxDataSize`**.

Global Properties Related to resolveAliases

Property	Description	Default Setting
RequestMaxArrayLength	This property sets the maximum **number of elements** in the data array that is sent to the server.	RequestMaxArrayLength=10000
RequestMaxDataSize	This property sets the maximum size (in **bytes**) to request from the data array.	RequestMaxDataSize=1600000

For more information on global properties, go to the topic, *Using globalsettings.properties*.

Checking the Health of the XHQ Server

This command checks the health of the XHQ Server during a restart of the XHQ System. In addition, this command may also be used to cancel a purge request or obtain the current status.

Syntax:

```
healthcheck ["timeout=#[m|s]>"][verbose][<module>...]
healthcheck decode <n>
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Examples:

```
>xhqci healthcheck "timeout=2m"
>xhqci healthcheck decode %ERRORLEVEL%
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters for healthcheck

Parameter	Description
#	The number of minutes (m) or seconds (s) for the test.
verbose	Log test steps to console.
module	A standard module (XHQ Server Process) name: `xhq_eserver`, `xhq_cserver`, `xhq_sserver`. The default is to check all.
decode	Decode the exit status <n> from a previous invocation.

Historical Evaluation

This mechanism evaluates the primitive's expression for a given time and stores the results to history. Historical evaluation behaves like a regular evaluation with the exception of switching Now() to a given time.

There are two ways to evaluate: **`path`**, which uses the direct path or alias, or **`exprfile`**, which references an external file. The latter is also referred to as Free-form Evaluation.

Syntax 1:

```
evaluate
"path=<member path or alias>"
"timestamps=<ts1;ts2...>"|"tsfile=<filepath>"
["mode=<series|scheduler|interval>"]
["interval=<interval value>"]
["basetime=<basetime>"]
[record]
[silent]
["outfile=<filepath>"]
["outdelim=<delimiter>"]
["tsformat=<timestamp format>"]
["qtext=<none|general|detailed>"]
["qnumeric=<none|dec|hex>"]
```

Example 1:

```
xhqci evaluate "path=TK101V" "mode=interval" interval="60"
"timestamps=2015-03-16 06:06:06;2015-03-16 07:06:06" record
```

Syntax 2:

```
evaluate
"exprfile=<filepath>"
"timestamps=<ts1;ts2...>"|"tsfile=<filepath>"
["anchorpath=<path to component>"]
["mode=<series|interval>"]
["interval=<interval value>"]
["basetime=<basetime>"]
["outfile=<filepath>"]
["outdelim=<delimiter>"]
["tsformat=<timestamp format>"]
["qtext=<none|general|detailed>"]
["qnumeric=<none|dec|hex>"]
```

Example 2:

```
xhqci evaluate "exprfile=expression.txt" "mode=series"
"tsfile=times.txt"
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Command-Line Parameters

Parameter	Description
path	The path to the primitive member or alias with expression evaluation.

Parameter	Description
exprfile	The path to the file that contains the expression.
anchorpath	Indicates the expression's reference component. All relative paths in the expression are evaluated in this context.
mode	*Options*: • **`series`** Specify re-evaluation time manually, one-by-one. • **`scheduler`** Specify start and end times. All times in between are calculated based on Schedule interval. • **`interval`** Specify start and end times. All times in between are calculated based on given interval parameter.
interval	When "interval" mode is specified, this parameter calculates the times based on the start and end times. Basically, it is the number of **seconds** between times specified in evaluations. The minimum value is 60 seconds.
basetime	The exact timestamps that are used for the evaluation process are calculated based on this "basetime". For example, The first timestamp is: `basetime + interval * 1` The second: `basetime + interval * 2` And so forth, to the Nth: `basetime + interval * N`
timestamps	The given time(s) to re-evaluate the expression.
tsfile	The path to the file that contains the given time(s) to re-evaluate the expression.
record	Records the evaluation results and stores them in the historian specified by the member specified by the "path" mode.
silent	This flag is only valid when both the "path" and "record" parameters are specified. In "silent" mode, evaluation results are not returned.
outdelim	The output field delimiter character. By default, it is the tab character.
outfile	The path to the file that is used for the output of expression calculation.
tsformat	The format string for timestamps. The default format is yyyy-mm-dd HH:MM:SS.sss.
qtext	The quality description output field. The options are none, general, or detailed (which is the default).
qnumeric	The numeric quality output field. The options are none (which is the default), dec, or hex.

Clearing Audit Records

This command enables you to remove all audit messages from the audit server in a single operation.

Syntax:

```
clearaudit
["eshost=<enterprise hostname>"|"rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Additional XHQ Server Actions

The **`servertool`** command enables you to carry out the following tools for use with the XHQ Servers.

Syntax:

```
servertool <tool>
["eshost=<enterprise hostname>" | "rshost=<router hostname>"]
["sshost=<solution hostname>"]
["uid=<user login name>" "pw=<user password>"]
```

Examples:

```
>xhqci servertool updateLogging
>xhqci servertool getSolutionRevId "eshost=server1" "sshost=server1"
"uid=jsmith" "pw=enterprise"
```

Parameters that deal with hostnames and user access are common to all xhqci commands. Go to the topic, *Common Parameters to xhqci Commands*.

Servertool actions

Parameter	Description

tool	Use one of the following actions/operations:
	• **cacheSchemaValidation**
	Queries or controls the cache schema validation behavior at XHQ restart. Basically, this allows the administrator to force cache schema validation upon restart, to check the status of cache validation at the next restart, or to reset the cache validation behavior at start-up to the default behavior.
	Syntax:
	`servertool cacheSchemaValidation [status|default|force]`
	Attributes for this parameter are:
	○ **`default`**
	Uses standard start-up logic to determine if full validation is required.
	Output:
	`xhqci servertool cacheSchemaValidation default`
	`Accessing XHQ as user 'DOMAIN\xhquser'`
	`Cache schema validation: Validation will not be forced on restart unless XHQ is shutdown abnormally.`
	○ **`force`**
	Forces validation on XHQ restart.
	Output:
	`xhqci servertool cacheSchemaValidation force`
	`Accessing XHQ as user 'DOMAIN\xhquser'`
	`Cache schema validation: Validation will be forced on restart of XHQ.`
	○ **`status`**
	Shows the current setting of the cache schema validation on XHQ restart (force or default). The status operation is assumed if none is supplied.
	Output, example 1 (when no prior invocation of this command is used with `force` or prior

Parameter	Description

invocation of `default` is used):

```
xhqci servertool cacheSchemaValidation status
Accessing XHQ as user 'DOMAIN\xhquser'
Cache schema validation: Validation will not be forced on
restart unless XHQ is shutdown abnormally.
```

Output, example 2 (when prior invocation of this command is used with `force`):

```
xhqci servertool cacheSchemaValidation status
Accessing XHQ as user 'DOMAIN\xhquser'
Cache schema validation: Validation will be forced on restart
of XHQ.
```

- **purgeUserAuthCache**
  Purges cached user authorization information.

  Important: Only Role Admins are allowed to use this command. If the user does not have Role Admins permissions, then the following message is returned:

  "Failed to purge user auth cache: Access is denied."

  Attributes for this parameter are:

  - `*all*`
    This purges for all cached users.

    ```
    xhqci servertool purgeUserAuthCache *all*
    ```

    Successfully purged cached user authorization data.
  - `domain\user1 [domain\user2…]`
    This purges the cached authorization information for the given users. You must use the domain\account format when specifying the user(s).

    ```
    xhqci servertool purgeUserAuthCache acme\mmartian
    acme\fleghorn
    ```

    Successfully purged cached user authorization data for specified users.
- **purgeAggregateCache**
  Purges the aggregate cache for a given XHQ point or the entire cache.

  Attributes for this parameter are:

  - `all`
    This purges the entire cache.

    ```
    xhqci servertool purgeAggregateCache all
    ```
  - `"path=pointName1,pointName2,..."`
    This purges the cache for specified points in the XHQ solution. The path may be a **full solution path** or an **alias**. The path must be to points for calculated aggregates and not points that have expressions.

    ```
    xhqci servertool purgeAggregateCache
    "path=alias1,alias2,alias3"
    ```
- **updateLogging**
  Which updates logging for the XHQ Enterprise, Solution, and Cache servers by force reading the existing log properties file for each server.

Parameter	Description

	• **dumpStacks**
	Which dumps the thread stack (thread instrumentation) for the XHQ Solution, Cache, and Data Recorder servers.
	• **getModelChanges**
	Which retrieves the number of modified components in the XHQ Workbench.
	• **getModelRevId**
	Which retrieves the Model Server revision ID.
	• **getSolutionRevId**
	Which retrieves the Solution Server revision ID.
	• **modeldist**
	The distributed model. Attributes for this parameter are:
	◦ **`-targethost`**
	This is the fully-qualified host name of the server that receives the Model update. The syntax is:
	`xhqci servertool modeldist "targethost=<fully-qualified host name>"`
	◦ **`-realm`**
	This is the realm name for the solution that receives the Model update. The syntax is:
	`xhqci servertool modeldist "realm=<realm_name>"`
	• **link**
	To link a server in the distributed solution (specified by its FQDN) to a node in the solution tree (specified by the path).
	Attributes for this parameter are:
	◦ **`path`**
	The path must specify an object on the root server and the xhqci must be connected to the root server.
	◦ **`server`**
	This is the fully-qualified domain name (FQDM) of the server in the distributed solution.
	The syntax is:
	`xhqci servertool link "path=<full_path>" "server=<FQDN>"`
	• **unlink**
	To unlink a server from the distributed solution.
	The attribute for this parameter is:
	◦ **`path`**
	The path must specify the server link that needs to be unlinked.
	`xhqci servertool unlink "path=<full_path>"`

Buffered write-back servertool actions

Parameter	Description

tool	The following servertool actions provide visibility and control over the history buffers.
	• **getHistoryBuffersStatus**
	Prints the status of the buffers for the selected Connection group.
	The syntax is:
	`getHistoryBuffersStatus "groupname=<name of the connection group>"`
	
	• **purgeHistoryBuffer**
	Purges (cleans, removes all items) the buffer for the selected Connection group.
	The syntax is:
	`purgeHistoryBuffer "groupname=name of the connection group>"`
	
	• **flushHistoryBuffer**
	Flushes (forces the buffer to record to the back-end) buffer for the selected Connection group.
	The syntax is:
	`flushHistoryBuffer "groupname=<name of the connection group>"`

7 | Using xhqdr with IP.21

Prior to using this command, refer to the topic, *Setting Up the Environment*, located in the IP.21 Connector chapter of the XHQ Connection Guide. It provides information on how to set-up the XHQ Server with the IP.21 client libraries, which are needed for the `xhqdr` utility to work properly.

The general syntax and command descriptions are as follows.

Syntax: **`xhqdr [/v|/i|/n] command [parameter]`**

Output Flags:

`/v` - Verbose mode (output to screen and log).

`/i` - Interactive mode (don't send output to log).

`/n` - Interactive mode with no feedback (don't send output to screen or log).

default (with no flag) - Output to the log but not the screen.

Supported Commands

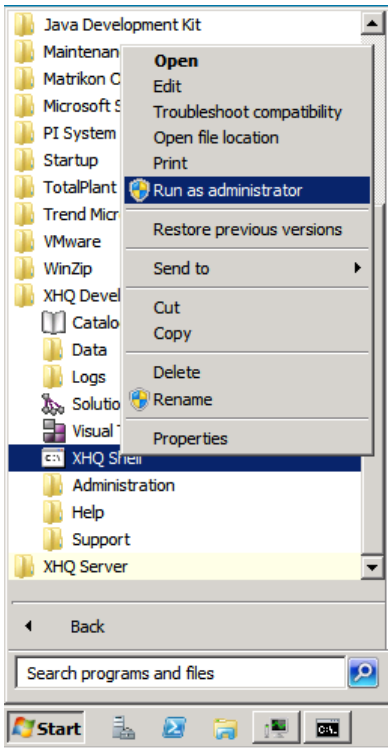
- **`delhistag`**
  Deletes tags from the Data Recorder.
- **`first`**
  Gets the first historical record.
- **`migratetoip21`**
  Transfers history from XHQ to IP.21.
- **`help`**
  Launches the xhqdr command help. To view the help for each of these supported commands within the XHQ Shell, enter:

  `xhqdr help <command>`

  Example: `xhqdr help first`

Important Things to Note

- You must have **administrative privileges** to execute the `xhqdr` batch file.
- The **`xhqdr`** command must be run in a command shell like the **XHQ Shell**.



- To view the help for each command within the command shell, enter:

  `xhqdr help <command>`

  Example: `xhqdr help first`

Getting the First Historical Record

To get the first historical record for the XHQ tag, run the following command.

Syntax:

```
xhqdr [/v|/i|/n] first "xhqtag=<xhq-full-primitive-member-path>"
["aliasFile=<filename>"]
```

Parameters:

`xhqtag` - The **full path** of the primitive member, whether located in the XHQ information model or from a tag.

For example:

```
::RootLevel.Area.rSomePrimitiveMember
::XHQTagTypeReal.rCurrentValue
::XHQTagTypeString.sCurrentValue
::XHQTagTypeInteger.iCurrentValue
```

`aliasfile` - Points to the XML file that contains a list of aliases and their associated file paths. This parameter is *optional*. For more information, see the topics, *Working with Aliases* and *Resolving Aliases to Paths*.

Output Flags:

`/v` - Verbose mode (output to screen and log).

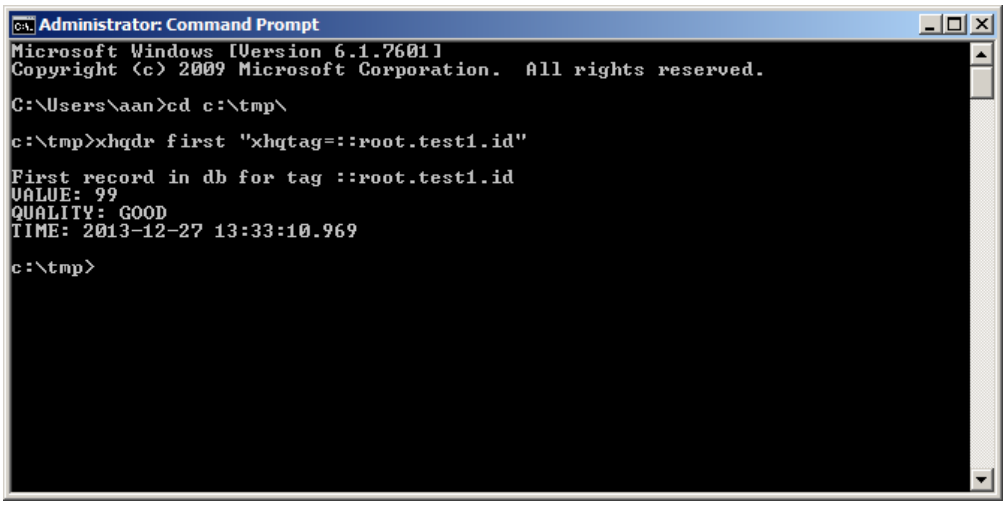
`/i` - Interactive mode (don't send output to log).

`/n` - Interactive mode with no feedback (don't send output to screen or log).

default (with no flag) - Output to the log but not the screen.

Examples:

```
xhqdr first "xhqtag=::root.test1.id"
```




```
xhqdr first "xhqtag=alias1" "aliasfile=C:\alias.xml"
```

Transferring History from XHQ to IP.21

To transfer historical data from XHQ to the IP.21 tag, use the following command.

Syntax:

```
xhqdr [/v|/i|/n] migratetoip21 "max=<size-of-chunk>" "start=<start-time>"
      "end=<end-time>" "xhqtag=<full-primitive-member-path>"
      "ip21srv=<server-name>" "ip21grp=<group-name>" "ip21tag=<tag-name>"
      ["aliasFile=<filename>"]
```

Parameters:

`max` - The **number of records** to read as a chunk from the XHQ Data Recorder.

`start` - The **start time** to read (FORMAT 'MM/DD/YYYY HH:MI:SS').

`end` - The **end time** to read (FORMAT 'MM/DD/YYYY HH:MI:SS').

`xhqtag` - The **full path** of the primitive member, whether located in the XHQ information model or from a tag. For example:

```
::RootLevel.Area.rSomePrimitiveMember
::XHQTagTypeReal.rCurrentValue
::XHQTagTypeString.sCurrentValue
::XHQTagTypeInteger.iCurrentValue
```

`ip21srv` - The IP.21 **historian**.

`ip21grp` - The IP.21 **group**.

`ip21tag` - The IP.21 **tag**.

`aliasfile` - Points to the XML file that contains a list of aliases and their associated file paths. This parameter is *optional*. For more information, see the topics, *Working with Aliases* and *Resolving Aliases to Paths*.

Output Flags:

`/v` - Verbose mode (output to screen and log).

`/i` - Interactive mode (don't send output to log).

`/n` - Interactive mode with no feedback (don't send output to screen or log).

default (with no flag) - Output to the log but not the screen.

Examples:

```
xhqdr migratetoip21 "max=1000" "start=12/31/2013 23:00:01" "end=01/01/2014
03:00:01" "xhqtag=::root.test1.id" "ip21srv=ip21" "ip21grp=200"
"ip21tag=test_5"
```

```
xhqdr migratetoip21 "max=1000" "start=12/31/2013 00:00:01" "end=12/31/2013
10:00:01" "xhqtag=alias1" "ip21srv=ip21server" "ip21grp=200"
"ip21tag=test_5" "aliasfile=C:\alias.xml"
```

How Migration Works

During data transfer, the `xhqdr` utility first reads values from the XHQ Data Recorder. For each retrieved value, the utility checks if there is a record in IP.21 with the same timestamp.

When such a record exists (matching timestamp) ,it compares the Value and Quality of both the XHQ Data Recorder record and the IP.21 record. If the Value and Quality are the same, no new recording for the point occurs. However, if **either** the Value or the Quality from IP.21 **differs** from the XHQ Data Recorder, **both are replaced** with the Value and Quality from the XHQ Data Recorder.

If the record does not exist in the IP.21, then the record from the XHQ Data Recorder is recorded.

After each recording, the utility reads the recorded value back and compares it with the record from the XHQ Data Recorder. This double-check ensures that what is recorded matches the original record.

Depending on the IP.21 load, the recorded value may not be available for immediate reading after the recording. In this case, the `xhqdr` utility may make several attempts to read the value back from IP.21, increasing the wait time between each attempt. If the utility is unable to retrieve a recorded value, it stops the migration process and displays an error message.

This process may be started multiple times for the same period of time in cases when the migration fails.

It's important to note that the `xhqdr` utility will not migrate a value if the **data type** from IP.21 differs from that of the XHQ Data Recorder. For example, a Double XHQ Data Recorder value will not migrate to an Analog IP.21 tag. However, a Long XHQ Data Recorder value will migrate to Discrete IP.21.

Deleting Tags from the XHQ Data Recorder

To delete tags, use the **`xhqdr delhistag`** command.

Before running this command,

- DO stop the XHQ solution.
- DO NOT stop the XHQ services or DO NOT do a complete shutdown.

Syntax:

```
xhqdr [/v|/i|/n] delhistag [-all]
  ["paths=<comma-separated-path-list>"|"pathsfile=<filename>"]
  ["aliasFile=<filename>"]
```

Parameters:

`all` - To delete all tags from XHQ.

`paths` – To delete tags in the path list.

`pathsfile` – To remove tags listed in the file.

`aliasfile` - Points to the XML file that contains a list of aliases and their associated file paths. This parameter is *optional*. For more information, see the topics, *Working with Aliases* and *Resolving Aliases to Paths*.

Output Flags:

`/v` - Verbose mode (output to screen and log).

`/i` - Interactive mode (don't send output to log).

`/n` - Interactive mode with no feedback (don't send output to screen or log).

default (with no flag) - Output to the log but not the screen.

Example:

```
xhqdr /v delhistag "paths=alias1,::rootid1" "aliasfile=C:\alias.xml"
```

Working with Aliases

When using XHQ aliases, add the **`aliasFile`** and `xhqtag` parameters to the given `xhqdr` command (`first`, `migratetoip21`, or `delhistag`).

Syntax:

```
xhqdr <command name> "xhqtag=<xhq tag name>"
  ["aliasfile=<filename>"]
```

Parameters:

`xhqtag` - Is the XHQ tag name.

`aliasfile` - Points to the XML file that contains a list of aliases and their associated file paths. This parameter is optional.

The alias XML file can be generated by the **`xhqci resolveAliases`** command. For more information, see the topic, *Resolving Aliases to Paths*.

Example:

```
xhqdr first "xhqtag=Alias1" "aliasfile=C:\alias.xml"
```

8 | Troubleshooting Connector Backlogs

This section provides explanations for the main causes of XHQ connector backlogs and some troubleshooting steps that have proven useful in diagnosing backlog-related issues.

What are Backlogs?

Backlogs occur when a Global Collection was trying to process against a back-end source but could not do so.

When a backlog occurs, a log entry is recorded in the log file for the relevant XHQ connector. The following is an example backlog log entry:

```
06/05/2006 07:50:01.052: [Thread-0] Server backlogged, rescheduling poll for oid=835
```

Backlog log entries are often accompanied by one or more of the following log entries.

```
06/04/2006 16:40:39.933: [Connector-1] Exception attempting to close connections
06/04/2006 16:40:39.933: [Connector-1] Io exception: Connection reset
12/07/2005 13:00:39.524: [Connector-1] No more data to read from socket
12/07/2005 13:00:39.524: [Connector-1] DbConnectionProcessor: Problem processing query:
<SQL query typically follows>
```

Other typical messages are:

Message	Description
`Server backlogged, rescheduling poll for oid=xyz`	This message is generated by the JDBC connector when the Solution Server denies a request from the connector to start another data update for a particular collection. The Solution Server denies a new update cycle because the corresponding collection is still in the update state and has yet to complete the previous cycle.
`<SchedulerAdapter>: WARNING: Backlog on item`	This message may be generated by poll-based connectors, such as PHD or IP21. This message is generated for point data when the connector determines that the next time for processing this point is already passed. In this case, the scheduler recalculates the next cycle to be in the future. If the calculation of the next time is in the future and there was a previously reported backlog, then the connector reports that the backlog has ended (see the next message).
`<SchedulerAdapter>: Backlog ended on item`	This condition typically happens when there are too many scheduled items for a single connector and a significant number of synchronous requests, such as Trend and Aggregate requests. The recommendation is to split all data points to several connectors, and run the PHD connector in multi-process mode.

Common Causes of Backlogs

Backlogs typically occur because the connector cannot acquire and/or maintain a connection to the relevant data source to execute the associated query.

Backlogs result in unfulfilled data requests, which have not been serviced within at least one poll period of an XHQ connector. If a collection is unable to process due to a backlog, the collection data becomes `UNCERTAIN` after one poll period after the expiration of one poll period, or even `BAD` after the expiration of two poll periods.

For more information, go to the topic, *About Data Quality Validation*, located in the XHQ Connection Guide.

XHQ administrators must consider the following potential original causes of backlogs:

- Back-end server configuration problems;
- XHQ server sizing or load problems;
- Back-end Historian configuration problems;
- Customer network problems;
- Customer network limitations (bandwidth, latency, configuration);
- XHQ problems (configuration, bugs).

Problems with the XHQ server configuration and/or back-end server configuration and/or network-related connector issues are the most frequently identified causes of backlogs.

Backlogs preceded by one or more I/O errors strongly suggest a network-related problem and other indicators may be available in the system logs (for example, indications of problems such as packet loss). Network issues can prevent timely completion of the query.

The IT department, database administrators, historian server administrators, and so forth, may need to be involved in order to resolve backlogs caused by network problems. For instance, they may need to perform network or database traces to assist in identifying the problem.

Another example of how backlogs can occur is if too many collections are configured against a single Connection Process and the related connection pool is set too small.

An occasional backlog message may occur with little or no impact to system performance and this can be due to system dynamics such as sporadic, high demand being placed upon a back-end data source. This is especially true in system configurations where multiple third-party products are requesting data from the same source, or where the system design or configuration is not laid out to adequately meet sporadic peak loads.

Occasionally, the XHQ Server can also cause backlogs. After evaluating potential non-XHQ causes of backlogs, and if the XHQ System is still suspected to be the cause of any backlog issue, contact the XHQ Customer Support team for assistance with the analysis.

Consider the following *example*.

EXAMPLE: BACKLOG TROUBLESHOOTING

Given:

The Global Collection is not scheduled. It polls every 30 minutes, and it takes 25 minutes to complete the query.

In this scenario, if the first poll occurs at 12:00 pm, then the next poll occurs at 12:55 pm (12:00 pm + 25 minutes processing time + 30 minute poll period).

At 12:55 pm, other connector activity occurs that prevents this Global Collection from polling the back-end source.

The Connection Process has a certain number of connections available in the pool, and the number of collections that poll back-end sources exceeds this number, which prevents this Global Collection from being able to use a connection in the pool.

A backlog message is written to the Connection Process log file to reflect that the query could not be processed. At this point the first poll period expiration has passed and the data may become `UNCERTAIN`. If another poll period expires, the data may become `BAD`.

If this Connection Process has too few connections in the pool associated with it for the number of Global Collections that use it, as a precaution, increase the connections in the pool, or create a new Connection Process that would have its own connection pool. The decision to do this depends upon the type, availability, dependability, and configuration of the back-end data source.

Steps to troubleshooting the backlog

Step 1	Cold restart of the XHQ Server.
Step 2	Ensure the XHQ Server has the recommended OS, RAM memory, and CPU configuration. See the topic, *System Requirements*, located in the XHQ Installation Guide for details.
Step 3	Consider the effect(s) of additional software installed on the XHQ Server. Such software may contribute to or even cause the problem.

Step 4	Check the solution log.
	Open the solution log in `%XHQ_SERVER_LOGS%` (either `solutionlog1.out` or `solutionlog2.out`) and check if it contains logged messages similar to the following:
	`07/11/2014 17:55:09.739: GCInvoker: forced GC @ 6% free memory. Now 92% free (used/total=6475K/87392K)`
	If the GC Invoker (garbage collection) struggles to free enough memory for normal operations, the XHQ Solution Server heap size may need to be increased. To do this, add the following line to the `%XHQ_SERVER_REPOS%\xhqboot.properties` file:
	`xhq_sserver.startup.memory=1024m`
	Note: Setting the memory to 1024m may not work on all servers or configurations. This is especially true if the Asian Language Pack Extensions are installed on the server.
	On XHQ startup, if this setting produces an error in `%XHQ_SERVER_LOGS%\xhqboot.log` resulting in the XHQ Solution Server not reserving enough heap, then drop the value of this setting iteratively by 64m and restart XHQ.
	Tip: Try 896m, and so forth, until an appropriate setting for that particular XHQ Server is found such that the server starts without further backlogs.

Step 5	Start a continuous ping against the back-end server associated with the backlogging connector. Log the output to a file. Open the **Windows Command Prompt** (`CMD`) as **Administrator**, enter: `ping -l 1440 -t server > ping.log` **Note:** Please note that routers may treat ping requests (echoes) with lower priority or drop them so this test is less dependable if the network path contains a router. Check the `ping.log` file periodically to see if either of the following key messages are found: Request Timed Out or Destination Host Unreachable. Depending on the tolerance of the back-end data source to network errors, even a single one of these network issues can cause sporadic backlogs, especially for very complex or large scale queries to a backend server. The ping test can be combined with trace routes and standard network troubleshooting utilities. The network and hardware issues identified and resolved over the years included faulty, misconfigured, or under-powered back-end servers, bad or intermittently bad NIC cards, misconfigured routers and/or bridges, poorly configured databases often with improper indexing, over-utilization of back-end sources by other third-party clients in parallel with XHQ access, and so forth. **Important:** These kinds of network tests or the installation of certain test utilities usually require prior notification to the network or IT administrator(s) in many cases to avoid causing issues with the network or security policies or being misinterpreted as hacking

attempts.

Step 6 Check the connection pool.

If a Connection Process is defined to have a certain number of connections in the pool, and the number of collections that poll back-end sources exceeds this number, backlogs may occur. Increase the connections in the pool for a backlogging Connection Process. Try increasing the pool in increments of 5 over a few days and see if the backlogs are eliminated. Stop and restart the Connection Process after each change.

See the *XHQ Connection Guide* for more information.

Step 7 Increase the poll period of any backlogged connector to two, or perhaps even four, times the current poll period. For example, if the poll period is one hour, increase it to four hours. Also, consider using custom or scheduled poll periods.

See the *XHQ Connection Guide* for more information.

Step 8 Address the Global Collections, Connection Processes, Connection Groups.

If a Connection Process encounters backlogs and has multiple Global Collections configured against it, then set up separate Connection Processes for each Global Collection so that each Connection Process is associated with only one Global Collection. This allows the Connection Process to devote itself fully to the polling of data for only one Global Collection and helps to isolate the problematic Global Collection from all others. This new configuration may directly solve the problem such that no backlogs occur.

Important: It is mandatory that a suitable backup of the repos is made, validated, and archived prior to any such solution reconfiguration.

Consider the following example.

Old Configuration	New Configuration
Connection Process A Connection Group A.1	Connection Process A Connection Group A.1
Global Collection A Connection Group A.2	Global Collection A
Global Collection B	Connection Process B* Connection Group A.2
	Global Collection B

* *This is the newly created Connection Process.*

Note: Even if this configuration does not solve the backlog problem, it can still be useful for troubleshooting purposes because the one Global Collection associated with the Connection Process is also the only Global Collection that creates log entries in the associated Connection Process log file. No other Global Collection log entries is outputted to this Connection Process log file.

9 | XHQ Redundancy

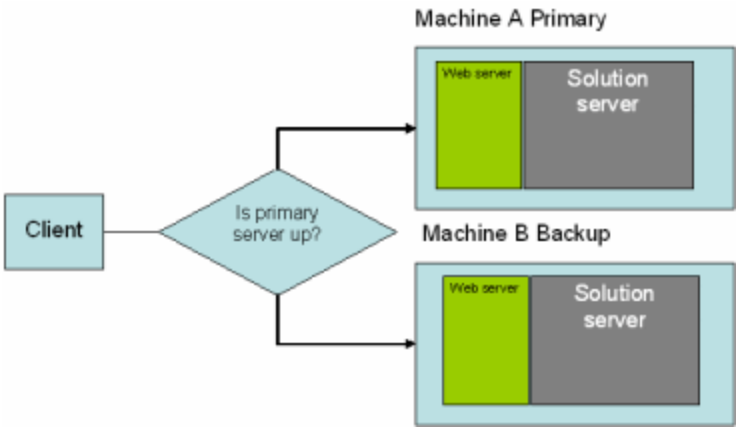
XHQ Redundancy helps to make sure that XHQ users do not experience downtime due to an XHQ Server, a Web server, or a Solution server failure. In this case, *Redundancy is only for the applet-based XHQ Solution Viewer*.

This feature allows a defined subset of your solution to continue running even after the XHQ Server and/or the XHQ solution becomes unavailable due to a planned, or even an unplanned, network downtime (for example, shutting down the XHQ solution for a scheduled patch update or XHQ upgrade).

This initial version is intended for use in specific scenarios, such as displaying XHQ solutions on central big board displays, where those solution displays need to be made highly available for operations personnel.

How is Redundancy Applied?

In XHQ, redundancy is achieved through the XHQ Solution Viewer (the browser client) by using an identical solution on two different machines: the **Primary server** and the **Backup server**. If the Client detects that the primary server is down, it switches to the backup server. It automatically switches back once it detects that the primary server is up and running again.

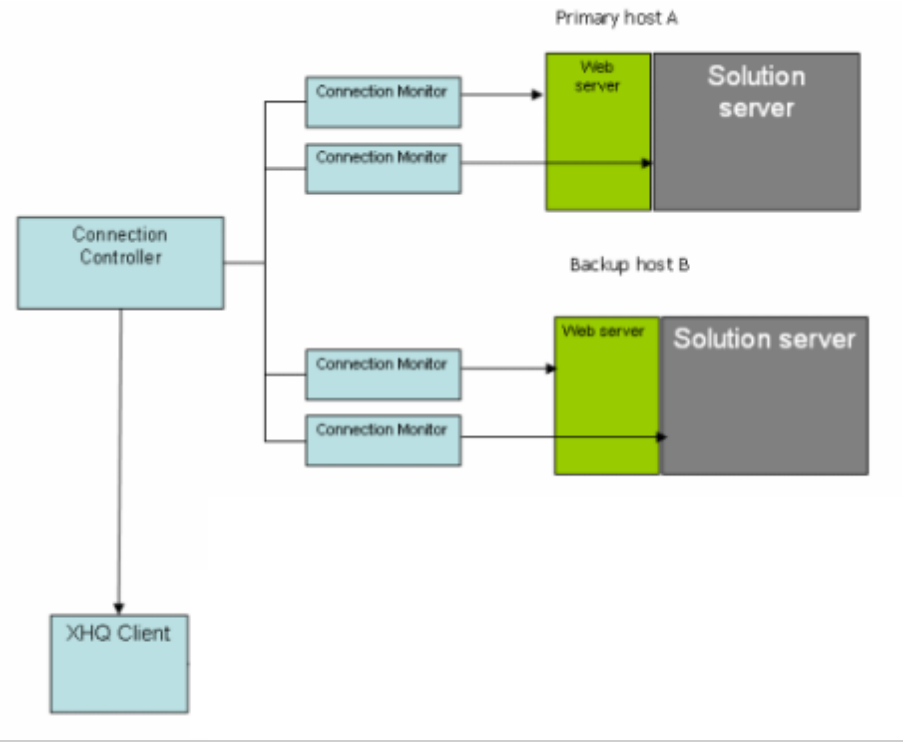


Using Identical Solutions on Two Separate Servers

How Redundancy Works in XHQ

If redundancy is configured, a Connection Controller object is created along with four Connection Monitor objects:

- One monitors the Primary Web server;
- One monitors the backup Web server;
- One monitors the Primary Solution server;
- And one monitors the backup Solution server.



If the Primary Solution server is up, the connection to the Primary Solution server is established. Otherwise, the connection to the backup Solution server is established. If both the Primary and backup Solution servers are down, the XHQ Client (or XHQ Solution Viewer) exits.

How to Configure Redundancy

To enable the redundancy feature, you will need to define the host names for both primary and backup servers in the **`appletsettings.js`** file.

Defining Host Names

To define the primary and backup server host names

1. Browse to the **`%XHQ_WEB_DATA%\repos\conf\web`** directory (which is typically `C:\XHQ\data\repos\conf\web`).
   This is the location of the customized XHQ web script files.

   For more information, go to the topic, *Customized Web Script Files*.

2. Open the **`appletsettings.js`** file.
   Use a text editor, such as Notepad.
3. Define the **host names** using the following variables:

```
primaryHostname = '<machineA>';
backupHostname = '<machineB>';
```

Where `<machineA>` and `<machineB>` are the computer names.

Example:

```
primaryHostname = 'acme1';
backupHostname = 'acme2';
```

4. **Save and close** `appletsettings.js`.
5. Launch the **XHQ Solution Viewer** (the browser client).

Once the XHQ Solution Viewer is started and the connection to the Web server is established, the applet begins to download and starts the initial connection to the Solution server.

Classic XHQ Solution vs. Redundant XHQ Solution
If the `primaryHostname` and the `backupHostname` are not defined, the solution is a *Classic* XHQ Solution. On the other hand, if they are defined, then the solution is a *Redundant* XHQ Solution.

From this point, the Connection Controller object is created and the Connection Monitor objects begin to monitor the Web server and the Solution server on both Primary and Backup hosts.

For more information on the Connection Controller and Monitor objects, go to the topic, *How Redundancy Works in XHQ*.

Improving Performance
Initially, launch the XHQ Solution Viewer (browser client) on both the Primary and the Backup machines. This will cache the applet and all supporting `.html` and `.js` files on both machines. Should the client ever need to switch to the backup web server because the primary web server is down, this step saves from having to reload the applet and all `.html`/`.js` support files.

Setting Global Properties for Redundancy

The **`globalsettings.properties`** file allows you to set solution-wide options. This file is stored at the location specified by the environment variable `%XHQ_SERVER_REPOS%` on the Enterprise Server (which is typically `C:\XHQ\data\repos`). You can use a text editor, such as Notepad, to edit this file.

For more information, go to the topic, *Using globalsettings.properties*.

The following properties can be applied to a redundant solution.

Global Property	Description
`ServerConnectionMonitorTimer`	This sets the number of **milliseconds** when the Connection Controller checks the connection status of each of the four Connection Monitors. *Default:* ServerConnectionMonitorTimer = 30000 **WARNING:** Do not use a value that is less than the default of 30 seconds. A lower setting needs approval by the XHQ Customer Support team for the specific configuration in use. Otherwise, any value less than the default is not supported due to the potential negative impact of excessive heartbeat checks (on the network, the WAN performance, and the XHQ Server) and interference with general performance optimizations in client-server communications.
`RedundantMessageDisplayTimer`	When a switch in server occurs, a message box appears. This property sets the duration (in **milliseconds**) this message is displayed. *Default:* RedundantMessageDisplayTimer = 5000
`RedundantSwitchBackTimer`	This sets a delay (in **milliseconds**) when switching back from the backup solution server to the primary solution server. This delay allows the primary solution server to update and/or validate the cache server content. *Default:* RedundantSwitchBackTimer = 300000
`EnableConnectionMonitorDebug`	If **true**, the Connection Monitor produces debug messages in a given log file. Else, the default is **false**. *Default:* EnableConnectionMonitorDebug = false

Monitoring the Connection Status

After the initial connections to the Web and Solution servers are established, the Connection Controller checks the connection status every 3000 milliseconds of each of the four Connection Monitors. If the Connection Monitors detect a change in either the Web or Solution server connection, they notify the Connection Controller, which in turn sends a connection change alert to the XHQ Client.

To customize the number of milliseconds the Connection Controller checks the connection status, use the global property **`ServerConnectionMonitorTimer`**. For more information, go to the topic, *Setting Global Properties for Redundancy*.

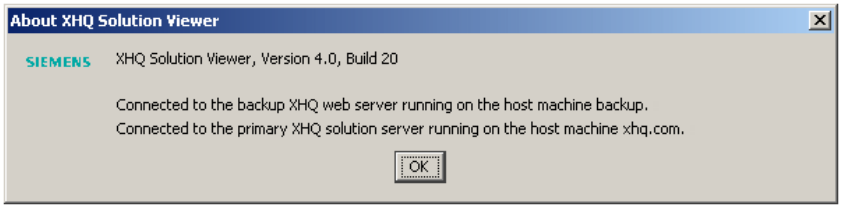
Accessing Server Information

From the XHQ Solution Viewer (browser client), you can quickly access server information by using the following right-click shortcut menu option.

To access server information from the Solution Viewer

1. In the **View panel** of the XHQ Solution Viewer, **right-click** and, from the shortcut menu, **select About XHQ Solution Viewer**.

The "About XHQ Solution Viewer" dialog box appears.



2. Click **OK** to close the dialog box.

How the XHQ Client Handles Connection Change Events

If the XHQ Client receives a connection change alert from the Connection Controller, the following events occur, depending on the given connection status of either/both the Web server or/and the Solution server.

What to expect and when

Expected behaviors	Web server connection change	Solution server connection change	BOTH the Web and Solution servers are down
All pop-up windows are closed.	✔	✔	✔
The current object path and view name is saved in the cookie.	✔		
The applet is reloaded from the given Web server specified in the event.	✔		
The XHQ splash screen will not be displayed on the applet.	✔		
The view saved in the cookie is also reloaded.	✔		
The client view cache is switched to the current web server.	✔		
If classic XHQ security** is used, an authorization dialog box appears prompting for re-authentication.	✔		
Historical mode is persisted.		✔	
User preferences are not migrated.		✔	
Connection to the Solution and Model servers are stopped.		✔	
New connections are established on both the given Solution and Model servers specified in the event.		✔	
The current view is refreshed.		✔	
An error message is displayed.			✔
All current connections are stopped.			✔
The XHQ Client is closed.			✔

** For more information on XHQ security, go to the topic. *Security Access and Privileges*.

ANS and Redundancy

In this initial version, XHQ ANS (which is the XHQ Alert Notification System) must only run on the primary XHQ Server. As such, if the XHQ client is connected to the backup XHQ web server and the primary web server comes back online, the XHQ client **does not** automatically fall back to the primary web server. Instead, you must manually configure the backup web server using the following procedure so that the XHQ ANS web application running on the backup web server can continue to communicate with the XHQ ANS server and XHQ ANS database running on the primary XHQ Server. This also requires the **primary** host machine to start its TNS listener to allow a database connection from another machine, which in this case is the backup web server.

Therefore, to use XHQ ANS with a redundant solution, you must edit the `config.json` file from the **backup machine**.

Editing the config.json File

To edit the config.json file

1. From the **backup server**, go to the `%XHQ_WEB_HOME%\webapps\ans\app\config` folder (which by default is `<systemdrive>:\Program Files (x86)\XHQ\XHQ Web Root\webapps\ans\app\config`).
2. Open the **`config.json`** file and locate the following snippet:

```
"application": {
    "version": "1.0",
    "serverHost": "localhost",
    "webAPIServerHost": "localhost",
    "serviceBase": "/indx/xhqwi/api/ans/",
    "impServiceBase": "/indx/xhqwi/api/ans/",
    "coreServiceBase": "/indx/xhqwi/api/core/",
    "adminServiceBase": "/indx/xhqwi/api/admin/",
    "loServiceBase": "/indx/xhqwi/api/lo/",
    "loggerServiceBase": "/indx/xhqwi/api/core/",
    "preferredLanguage": "en",
    "fallbackLanguage": "en"
  },
```

3. **Replace** `localhost` (which is highlighted in the example above) with the **primary host machine name**.
4. **Save** and close the **`config.json`** file.

10 | XHQ Solution Viewer Administration

Using XHQ Web Diagnostics

Running these diagnostic tests can help identify a configuration or network problem that is preventing a successful launch of the XHQ Solution Viewer.

To access the XHQ Web Diagnostics page, enter the following URL from a browser:

```
http://<servername>/indx/diagnostics.html
```

Where `<servername>` is the name of the Solution Server.

Test	Description	Compatibility
Diagnostic Test 02	Request a page that uses client-side JavaScript to generate and display a simple message.	
Diagnostic Test 03	Request a simple signed Java applet	
Diagnostic Test 04	Request a simple signed Java applet that is included in a large 'padded' JAR file	
Diagnostic Test 05	Request a simple signed Java applet as part of the Standard XHQ JAR file	
Diagnostic Test 06	Request a Java applet that reads and reports Java System Properties and Windows Environment Variables	
Diagnostic Test 07	Request a Java applet that sends and retrieves information to and from JavaScript	
Diagnostic Test 08	Request a Java applet that uses JavaScript to open a window	
Diagnostic Test 09	Request a Java applet that uses JFreeChart to make a chart	
Diagnostic Test 10	Request a Java applet that uses JNA to read and report Windows Environment Variables	
Diagnostic Test 11	Request a Java applet that downloads a Java Properties file and from it reads, for each server module, its unsecured and secured port numbers	
Diagnostic Test 12	Request a Java applet that downloads a Java Key Store and reads the Certificate therein	
Diagnostic Test 13	Request a Java applet that connects to the XHQ Server and reads and reports the Global Properties	
Diagnostic Test 14	Request a page that uses WebSocket protocol to establish communication with HTML5 Solution Viewer to generate and display properties.	

XHQ Web Diagnostics Page

Perform the following diagnostic tests **in the order given**. In each case, a successful test displays the information requested.

All tests show the minimum Java family as well as the XHQ version and browser name.

XHQ Web Diagnostic Tests

Test	Description	Results
Test 1	Launch the XHQ Web Diagnostics page.	Displays the XHQ Web Diagnostics page.
Test 2	Request a page that uses client-side JavaScript to generate and display a simple message.	Displays the browser name, the XHQ version and the minimum Java version.
Test 3	Request a simple signed Java applet.	Displays the browser name, the XHQ version, the minimum Java version, and a message confirming a successful test. Should the system fail this initial test, an additional test link is provided: "Request the simple applet JAR

Test	Description	Results
		through HTTP." This checks if you can request and download a copy of the JAR file directly from the server to the client via HTTP.
Test 4	Request a simple signed Java applet that is included in a large *padded* JAR file.	Displays the browser name, the XHQ version, the minimum Java version, and a message confirming a successful test. Should the system fail this initial test, an additional test link is provided: "Request the 'padded' simple applet JAR through HTTP." This checks if you can request and download a copy of the JAR file directly from the server to the client via HTTP.
Test 5	Request a simple signed Java applet as part of the standard XHQ JAR file.	Displays the browser name, the XHQ version, the minimum Java version, and a message confirming a successful test. Should the system fail this initial test, an additional test link is provided: "Request the standard XHQ JAR through HTTP." This checks if you can request and download a copy of the JAR file directly from the server to the client via HTTP.
Test 6	Request a Java applet that reads and reports Java system properties.	Displays selected applet parameters that are specified with PARAM-tags under the OBJECT-tag that creates the applet: 1. The name of the applet 2. Whether the applet is signed. 3. The JARS included in the OBJECT-tag 4. The versions of these JARS In addition to the Java System Properties, it also shows the values of the clients Environment Variables (as determined by the Java System.getenv() method). Should the system fail this initial test, an additional test link is provided: "Request the standard XHQ JAR through HTTP." This checks if you can request and download a copy of the JAR file directly from the server to the client via HTTP.
Test 7	Request a Java applet that sends and retrieves information to and from JavaScript.	Tests the inter-communication between the Java applet and the JavaScript engine of the browser. It sends the Applet Class Name, and a randomly generated number back and forth between the two. The applet tests the values sent against those received back to see that they match and reports this. Both values should match. The test does nothing with the System Properties or

Test	Description	Results
		with Applet Parameters.
Test 8	Request a Java applet that uses JavaScript to open a window.	Loads a simple Java applet that does nothing but display a brief message and a button. With the exception of this button, the applet used is the same as the one in the simple applet tests. To run the test, press the button. If a page that simply says "Hello World!" opens in front of the applet page, your system has passed the test.
Test 9	Request a Java applet that uses JFreeChart to make a chart.	Loads a Java applet that: • Creates a simple pie chart that displays a random distribution of energy use values. • Allows you to open the pie chart and access a PDF containing an image of the chart and a table of the data. • Displays a text area containing version and copyright information about the JFreeChart and iText packages that do the charting and PDF handling. If this chart and information displays, and clicking on the chart opens the PDF, your system has passed the test.
Test 10	Request a Java applet that uses JNA to read and report Windows environment variables.	Tests the use of JNA to make Windows native method calls in Java applets. The test loads a simple Java applet that reads and compares the following in a scrolling text window: • The Windows Environment Variables from Windows native method calls using JNA. • The Windows Environment Variables retrieved with Java's built-in Sytstem.getenv() method. If this comparison displays, and the items match, your system has passed the test.
Test 11	Request a Java applet that downloads a Java Properties file, and from it reads the unsecured and secured port numbers for each server module.	Tests the ability of the applet to retrieve a Java Properties file that specifies which port numbers are used for unsecured communication and which are used for secured communication for each server module. If the mappings of server modules to unsecured and secured port numbers appear in the test box, then your system has passed the test. Note, these are default port mappings. To display **custom** port mappings, run the XHQ Port Map Utility.
Test 12	Request a Java applet that downloads a Java Key Store and reads the Certificate	Tests the ability of the applet to retrieve an X.509 Certificate that it could use to open SSL sockets for

Test	Description	Results
	therein.	communication with the server. However, no actual SSL socket is opened in this test. To display the Certificate information, run the XHQ Key-Pair Utility. If the Certificate issuer appears as the title and the details of the Certificate are displayed, then your system has passed the test.
Test 13	Request a Java applet that connects to the XHQ Server and reads and reports the global properties.	Tests connectivity with the core XHQ Servers. The test loads a simple Java applet that reads and reports the following in a scrolling text window: • The host name and port number reported by the XHQ Solution Server. • The global settings returned by the Model Server (or the defaults if the connection to the Model Server fails). If all of these items display, your system has passed the test.
Test 14	Request a page that uses WebSocket protocol to establish communication with HTML5 Solution Viewer to generate and display properties. **Important**: Before executing this diagnostic, make sure the XHQ HTML5 Solution Viewer process, **`xhq_viewserver.exe`**, is running.	Checks whether the browser supports WebSockets protocol and displays the results in the Log window. If it does, then it establishes communication with the XHQ HTML5 Solution Viewer using WebSockets protocol. If the communication is successful, it will display the properties that were returned by the Solution Viewer in the Log window.

Important Things to Consider

- By default, only Diagnostic Test 1 (the initial HTML test) is enabled; Diagnostic Tests 2-14 are disabled.
- Use the **`diagnostics.js`** script file to enable/disable the diagnostic tests.

  This file is located in the **`%XHQ_WEB_DATA%\repos\conf\web`** folder (the virtual directory is **`/indx/custom`**).

  Make sure that you only edit the diagnostics script file located in the path above. If this file does not exist in this path, then create a **new diagnostics.js** file. (In particular, **do not copy** the file of the same name located in %XHQ_WEB_HOME%\webapps\diagnostics folder.) Then, from this new file, set the variable(s) you want to enable/disable.

- To **enable all** diagnostic tests, set the **ENABLE_SUITE variable** to **true**; else, set to false to disable.
- To **disable individual diagnostic tests**, first enable all (using the ENABLE_SUITE variable) then (for each of the diagnostic tests you want disabled) set the variable to **true**.

  Example: `var PADDED_APPLET_TEST_DISABLED=true;`

  Currently, the variables are:

  - SIMPLE_JAVASCRIPT_TEST_DISABLED
  - ASP_TEST_DISABLED

- SIMPLE_APPLET_TEST_DISABLED
- PADDED_APPLET_TEST_DISABLED
- SIMPLE_APPLET_TEST_FROM_THE_STANDARD_XHQ_JAR_DISABLED
- SYSTEM_APPLET_TEST_DISABLED
- JAVASCRIPT_APPLET_TEST_DISABLED
- OPEN_WINDOW_TEST_DISABLED
- JFREECHART_APPLET_TEST_DISABLED
- JNA_APPLET_TEST_DISABLED
- SERVER_CONNECT_APPLET_TEST_DISABLED
- STATIC_JSP_TEST_DISABLED
- ACTIVE_JSP_TEST_DISABLED

The specific variable for disabling a given diagnostic test is named after the title that you see displayed at the top of the page. The naming convention for the variable is as follows:

a) Take the title as seen at the top of the test page (less the ": Disabled" suffix, where present).

b) Replace spaces with underscores.

c) Convert entire name to uppercase.

d) Append the name with "_DISABLED".

- This diagnostics page can be controlled to only output unsafe details. Otherwise, it simply does a test and indicates passed or failed. To re-enable **output unsafe server details**, place a file named ALLOW_UNSAFE_SERVER_DETAILS in the diagnostics directory, which is typically found under:

  `%XHQ_WEB_HOME%\webapps\diagnostics`

  The file name must match exactly. The contents of the file are not important and can be empty.

  The following example can be executed at an **administrator command-line prompt**.

  To create ALLOW_UNSAFE_SERVER_DETAILS, enter:

  ```
  C:\> echo.
  > "%XHQ_WEB_HOME%\webapps\diagnostics\ALLOW_UNSAFE_SERVER_DETAILS"
  ```

  Note that the **path must be quoted** because `%XHQ_WEB_HOME%` typically contains spaces (for example, `C:\Program Files (x86)\XHQ\XHQ Web Root`)

  Next, confirm that the file exists using `dir`:

  ```
  C:\> dir "%XHQ_WEB_HOME%\webapps\diagnosticsALLOW_UNSAFE_SERVER_DETAILS"
  Volume in drive C has no label.
  Volume Serial Number is 9CAB-B488
  Directory of C:\Program Files (x86)\XHQ\XHQ Web Root\webapps\diagnostics
  10/26/2016 11:27 AM 3 ALLOW_UNSAFE_SERVER_DETAILS
  1 File(s) 3 bytes
  0 Dir(s) 59,106,791,424 bytes free
  ```

To run the XHQ Key-Pair Utility

Use this task with Test 12 to generate a self-singed Private Key and X509 Certificate. This utility places the Certificate in a keystore that is accessible to the XHQ Web Server.

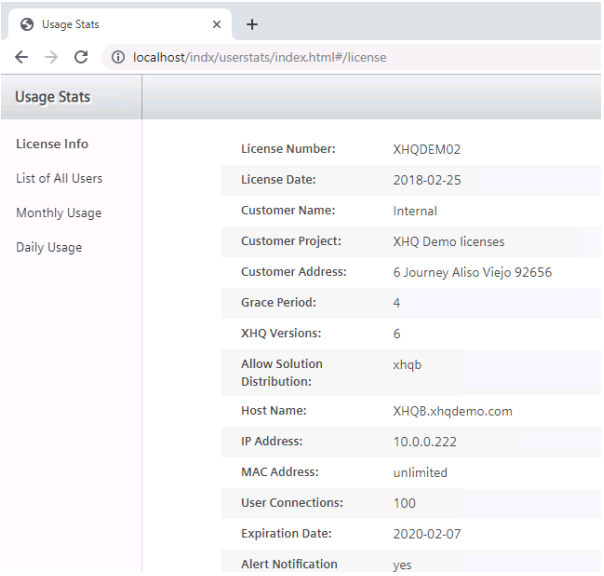
1. Open an **XHQ Shell** on the server in **Administrative** mode.
2. Type **`xhq_keypairutil /g`**.

Usage Stats

XHQ Usage Stats consists of the following four pages:

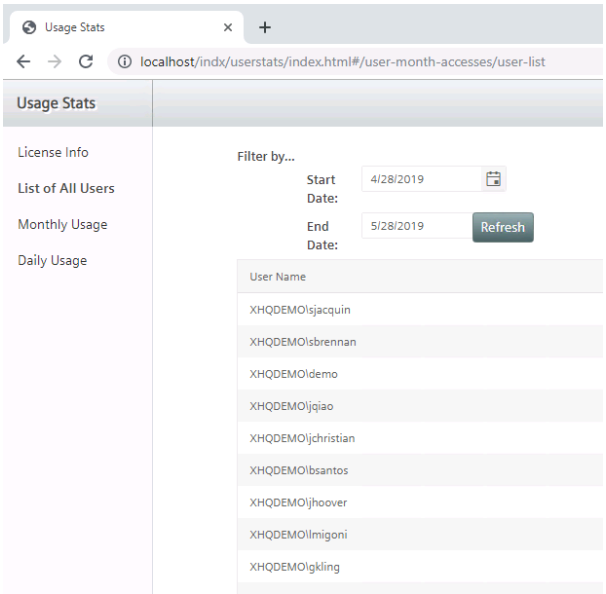
- **License Information**

  This page displays the current XHQ license information (license number, date, and so forth).



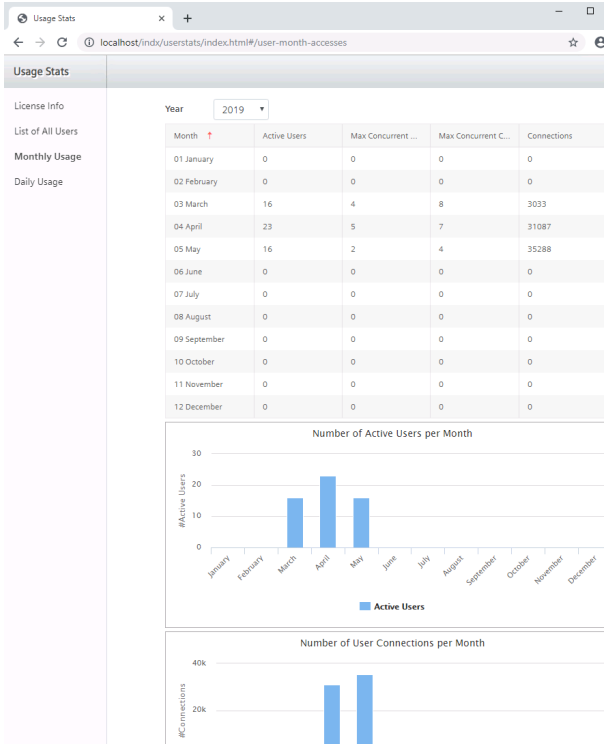
- **List of All Users**

  This page displays the distinct users during a given period of time.



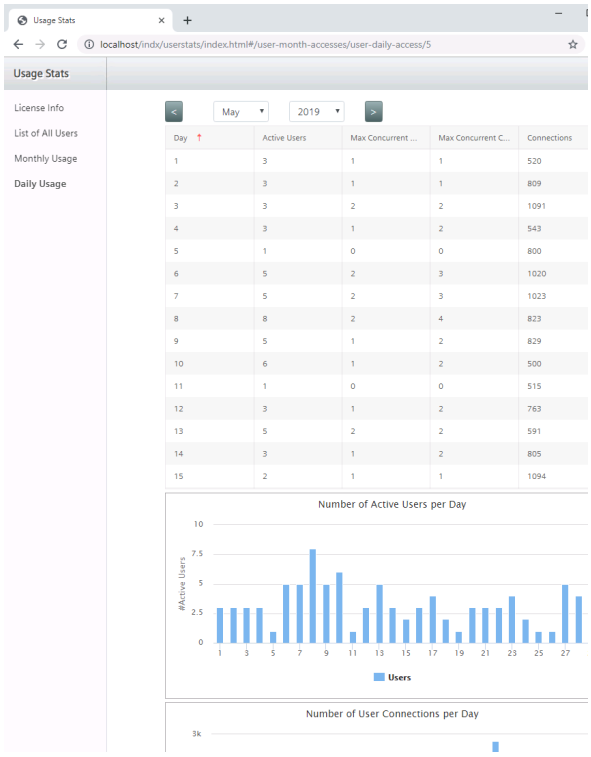
- **Monthly Usage**

  This page displays the number of active users, the number of concurrent users, the number of concurrent connections, and the number of connections by month for the selected year.



- **Daily Usage**

  This page displays the number of active users, the number of concurrent users, the number of concurrent connections, and the number of connections per day for the selected month.



You must have **Audit Admin** permissions or be in the **Solution Admin** group to access XHQ Usage Stats.

Also, **user names** provided are **case-sensitive**. So `ACME1\JoeSmith` and `ACME1\joesmith` are treated as two distinct users.

To access the XHQ Usage Stats homepage, go to **`http://<server_name>/indx/userstats/index.html`**.

Example: `http://localhost/indx/userstats/index.html`

Debugging by Session ID

There are times you may need to debug the XHQ Solution Viewer by session ID and in such a manner that the debug window does not appear for end-users.

Use the following URL parameter to troubleshoot a given session:

```
xhqnavbar.aspx?debug=true
```

Example: `http://localhost/indx/xhqnavbar.aspx?debug=true`

Debug Parameter Descriptions

Parameter	Description
debug	**false** – debugging by session ID is **disabled** (default). **true** – debugging by session ID is **enabled**.

Disabling HTTPS Enforcement

To stop HTTPS enforcement, you will need to remove the `requireSSL` attribute.

To disable HTTPS enforcement

1. Go to the **`%XHQ_WEB_HOME%\webapps\sv`** folder.
2. Open the **`web.conf`** file in a text editor.
3. Locate the line:
   ```
   <httpCookies httpOnlyCookies="false" requireSSL="true" />
   ```
4. **Delete** the highlighted attribute to change the line to this:
   ```
   <httpCookies httpOnlyCookies="false" />
   ```
5. **Save** **`web.conf`**.

Setting Cookie Expiration

The number of days the XHQ Applet cookie is set to expire is defined in the **`appletsettings.js`** file. The default value is five (5) days.

If a non-integer or a negative value or no value is specified, the default of five (5) days is assumed.

To set the cookie expiration

1. Browse to the `%XHQ_WEB_DATA%\repos\conf\web` directory, which by default is `C:\XHQ\data\repos\conf\web`.
   This is the location of the customized XHQ web script files.

   For more information, go to the topic, *Customized Web Script Files*.

2. Open the **`appletsettings.js`** file.
   Use a text editor, such as Notepad.
3. Define the **days until expiration** by setting the following variable to a **non-negative integer**:

```
defaultCookieExpDays = <number of days>;
```

Where `<number of days>` is the number of days until the cookie expires.

Example: `defaultCookieExpDays = 1000;`

4. **Save and close** `appletsettings.js`.

Using the XHQ Solution Viewer with the Oracle JVM

Currently, in order to load the XHQ Solution Viewer (browser client), the Oracle Java Virtual Machine (JVM) must be installed and enabled for use by the browser client. If the Oracle JVM is not available, you will not be able to run the XHQ Solution Viewer.

For a list of supported JVM versions, go to the topic, *Additional System Specifications and Restrictions*, located in the XHQ Installation Guide.

To determine the Oracle JVM version you currently have, open a Command Prompt and enter **`java -version`**.

Setting the Maximum Length for the Trend Title

For XHQ HTML5 Solution Viewer

The maximum length of the trend title allowed in My Trends is defined in the **`trendsettings.js`** file. The **default value is 164** characters.

For more information, see the topic, *trendsettings.js*.

To set the Trend title maximum length

1. **Browse** to the `%XHQ_SERVER_REPOS%\conf\web` directory, which by default is `C:\XHQ\data\repos\conf\web`.
   This is the location of the customized XHQ web script files.
2. Using a text editor, create the **`trendsettings.js`** file at this location.
3. **Set** the following variable:

   ```
   maxTitleLength = <maximum number of characters>;
   ```

   Example: `maxTitleLength = 255;`
4. **Save and close** `trendsettings.js`.

For the Applet-based XHQ Solution Viewer

The maximum length of the trend title allowed in My Trends is defined in the **`appletsettings.js`** file. The default value is also 164 characters.

To change the Trend title maximum length

1. **Browse** to the `%XHQ_WEB_DATA%\repos\conf\web` directory, which by default is `C:\XHQ\data\repos\conf\web`.
   This is the location of the customized XHQ web script files for the Applet-based XHQ Solution Viewer.

   For more information, go to the topic, *Customized Web Script Files*.
2. **Open** the **`appletsettings.js`** file.
   Use a text editor, such as Notepad.
3. **Set** the following variable:

   ```
   maxTitleLength = <maximum number of characters>;
   ```

   Example: `maxTitleLength = 255;`
4. **Save and close** `appletsettings.js`.

Solution Viewer Display Resolution

The XHQ Platform supports an XHQ Solution Viewer resolution of up to 1600 x 1280. Success with higher resolution displays, such as the 16640 x 3072 Barco display wall, is dependent on the system configuration in use (for instance, the Oracle JVM version, the graphics driver, the Microsoft Operating System drivers like DirectDraw, and so forth).

The following sections describe two global properties that are associated with display resolution:

- `DisableEmbeddedTrendDoubleBuffering`
- `EnableAntiAlias`

On some hardware platforms, the combination of disabling both of these global properties can result in poor performance of embedded trends.

You can elect to disable double-buffering (by setting the property to "true") and/or anti-aliasing (by setting the property to "false"), but do so with caution and only after consulting the XHQ Customer Support team.

For more information, go to the topic, *Working with PROPERTIES Files*.

Embedded Trends and Double Buffering

The XHQ system uses *double-buffering* for embedded trends to enhance the end-user experience. Double-buffering prevents the application from appearing sluggish and eliminates image flickering, resulting in smoother screen updates. Although it improves the refresh performance for embedded trends, double-buffering can increase memory consumption.

The default setting is `DisableEmbeddedTrendDoubleBuffering=`**`false`**.

Here are a couple of instances when you may want to consider disabling double-buffering:

- If you are using a high resolution display, such as a Barco display wall, to view several large embedded trends.
- If you are getting an "Out of Memory" error when displaying embedded trends.

Anti-aliasing in the XHQ system

By default, the XHQ system uses *anti-aliasing* when displaying views from a browser client. The anti-aliasing technique produces graphics and text that are easier to read by shading the pixels along the borders of these elements, making curves appear smoother and less jagged.

The default setting for the global property is `EnableAntiAlias=true`. This is the recommended setting due to the enhanced quality of the graphics and text.

The use of anti-aliasing affects system performance due to the increased CPU load required. The impact depends on the graphic element(s) being processed.

Working with High-Resolution Displays

If you are using a high-resolution display, such as the 16640 x 3072 Barco display wall, you must manually increase the Oracle JVM default memory heap size (on the client with the display) or the display size is limited by the default JVM heap memory setting to a lower value (such as 4.75 x 2.2 screens).

Use the following procedure to help you change this setting.

The XHQ Platform provides the required Oracle plug-in, but it cannot programmatically change the plug-in default heap size. You must have access to the plug-in control panel (on the relevant client) to change the heap size setting.

To increase the Oracle JVM heap size

1. Launch the **Control Panel** and click **Java Plug-in**.
   The Java Plug-in Control Panel appears.
2. Click the **Advanced tab**.
3. In the **Java Runtime Parameters** box, enter **`-Xmx<number>m`** (where <number> it the heap size in MB).

   Example: `-Xmx128m`

   The number of megabytes you specify depends on the memory available on the client and any JVM-specific limitations on the maximum value for the installed version.
4. Click **Apply** and **close** the Java Plug-in Control Panel.
5. **Restart** the browser client.

11 | About the XHQ Web Files

The XHQ Web Root Directory

During the XHQ Server installation, you have the option of installing the XHQ Web Root feature. These files serve the XHQ Solution Viewer. If this feature is installed, the web support files are copied to the local machine in the location specified by the environment variable **`%XHQ_WEB_HOME%\indx`**, which by default is the `C:\Program Files (x86)\XHQ\XHQ Web Root\indx` directory. However, the feature will function only if a web server (for example, IIS) is available on the local machine.

Folder names are in brackets [].

Directory contents of %XHQ_WEB_HOME%\indx		
[help]	solution_viewer.jnlp	xhqMissingPlugin.html
[images]	trendTableControls.html	xhqNav.asp
[lib]	trendTableDisplay.html	xhqNav.html
[scripts]	web.config	xhqNavbar.asp
[stylesheets]	xhq.asp	xhqNavbar.aspx
buildInfo.js	xhq.aspx	xhqNavbar.html
bv.html	xhq.html	xhqTrend.asp
debug.js	xhqChart.asp	xhqTrend.aspx
default.html	xhqChart.aspx	xhqTrend.html
diagnostics.html	xhqChart.html	xhqTrendPrefs.html
error.html	xhqChartPrefs.html	xhqTrendTable.html
ServerClock.class	xhqManagePrefs.html	

The following folders are in a different location specified by the environment variable **`%XHQ_WEB_HOME%\webapps`**, which is by default the `C:\Program Files (x86)\XHQ\XHQ Web Root\webapps` directory.

Directory contents of %XHQ_WEB_HOME%\webapps		
[admin]	[elogs]	[viewstats]
[ans]	[pmws]	[xhqadminws]
[answs]	[rds]	[xhqam]
[common]	[sv]	[xhqodata]
[composer]	[testxhqwse]	[xhqwi]
[diagnostics]	[tm]	[xhqwse]

NavBar-related Files and Directories

During the installation of the XHQ System, the following XHQ NavBar-related files and folders are copied onto the web server location for your solution, **`%XHQ_WEB_HOME%\indx`**, which by default is the `C:\Program Files (x86)\XHQ\XHQ Web Root\indx` directory.

Navigation Bar Files and Directories

Location	File or Directory Name	Description
`XHQ Web Root\indx\`	xhqManagePrefs.html xhqNavbar.aspx xhqNavbar.html	Files related to the XHQ NavBar.
`XHQ Web Root\indx\images\`	xhqImages	The xhqImages folder includes all images used by the XHQ Solution Viewer and XHQ NavBar.
`XHQ Web Root\indx\scripts\`	xhqScripts\applet xhqScripts\navigation	The applet folder includes the JavaScript files related to the XHQ applet, such as the `xhq.js` file. The navigation folder includes all the JavaScript files that contain functions for generating navigation events.
`XHQ Web Root \indx\stylesheets\`	xhqStylesheets	The xhqStylesheets folder holds all stylesheets provided with XHQ.

Customized Web Script Files

To protect your customized web scripts from being overwritten during an upgrade, make sure you use the following designated file locations.

The XHQ HTML5 Solution Viewer uses the following file to configure settings for the Interactive Trender:

- trendsettings.js

This JavaScript file is located in the **%XHQ_SERVER_REPOS%\conf\web** directory, which by default is `C:\XHQ\data\repos\conf\web`.

The following files are used for configuring the launch of the Applet-based XHQ Solution Viewer and reside in the **%XHQ_WEB_DATA%\repos\conf\web** directory:

- appletsettings.js
- svsettings.js
- styles.css

These files are required prior to launching the XHQ Solution Viewer. Therefore, they are not kept as Java properties files since there is no Java program running when the settings defined in these files are needed.

trendsettings.js

The **trendsettings.js** file is located in the **%XHQ_SERVER_REPOS%\conf\web** directory, which be default is `C:\XHQ\data\repos\conf\web`.

trendsettings.js Settings

Variable Setting	Description
`sMailTrendSubjectPrefix`	The prefix for the subject of the Trender e-mails. *Default:* `sMailTrendSubjectPrefix` = "XHQ Trend Of"
`maxTitleLength`	The maximum number of characters for the Interactive Trender title. *Default*: `maxTitleLength` = 164 See also, *Setting the Maximum Length for the Trend Title*.

appletsettings.js

The **appletsettings.js** file includes settings leading up to the creation of the applet. Assignments defined in this file (located in the **%XHQ_WEB_DATA%\repos\conf\web** directory) **override** the default assignments defined in the **appletsettings.js** file located at the `%XHQ_WEB_HOME%\indx\scripts` web path. This file currently includes the following settings.

appletsettings.js Settings

Variable Setting	Description
`missingPluginPage`	The value of this variable is a JavaScript function that presents a message to the

Variable Setting	Description
	XHQ user indicating that the Java plug-in must be installed in order for the XHQ Solution Viewer to work.
`blankTrend[0]`	The name of a blank trend. *Default:* `blankTrend[0]` = "Blank Trend"
`sMailTrendSubjectPrefix`	The prefix for the subject of trend e-mails. *Default:* `sMailTrendSubjectPrefix` = "XHQ Trend Of"
`solName`	The name of the solution. *Default*: `solName` = "" (Which results in the name of the first solution on the enterprise server being used.)
`esHost`	The Enterprise server host name. *Default*: `esHost` = "" (Which selects the host on which the Enterprise server is running.)
`ssHost`	The Solution server host name. *Default*: `ssHost` = "" (Which selects the host on which the Solution server is running.)
`rsHost`	The Router server host name. *Default*: `rsHost` = "" (Which selects the host on which the Router server is running.)
`useRouter`	A flag that, when true, indicates that the Router Server (specified by rsHost) should be used to find the enterprise and solution servers. *Default:* `useRouter` = false
`xansEnabled`	A flag that, when true, enables support for XHQ ANS on machines where XHQ ANS is installed. If XHQ ANS is installed, the default is `xansEnabled` = true. Otherwise, `xansEnabled` = false.

See also the following topics:

- *Defining Host Names*
- *Setting Cookie Expiration*
- *Using the XHQ Solution Viewer with the Oracle JVM*

svsettings.js

To load custom scripts for the XHQ HTML5 Solution Viewer, use **`dynamicLoadScript()`** and/or **`document.write()`** in the **`svsettings.js`** file.

You have to **create** the **`svsettings.js`** file. To do this, go to the `repos\conf\web\` directory, copy the existing `appletsettings.js` file, and rename this new file **`svsettings.js`**. Edit **`svsettings.js`** accordingly.

Here is a complete example of the `svsettings.js` entry:

```
onAppletInitListener = function(){
var applet = getApplet();
applet.setClientVariable("ClientVariableX", "1234");
}
function reset() {
var applet = getApplet();
applet.setClientVariable("ClientVariableX", "0");
}
dynamicLoadScript('custom/custom01.js');
document.write('<script type="text/javascript" src="custom/custom02.js"></script>');
```

styles.css

This file defines the component styles of the XHQ NavBar. The styles defined in this file (located in the `%XHQ_WEB_DATA%\repos\conf\web` directory, which by default is `C:\XHQ\data\repos\conf\web`) override those defined in the `xhqstyle.css` file located at the `%XHQ_WEB_HOME%\indx\stylesheets\xhqStyleSheets` web path.

About the /indx/custom Virtual Directory

During the XHQ installation, the virtual folder **`/indx/custom`** is created so that the customized settings described in the previous section are available to the web server.

The `repos\conf\web` folder is the physical path of this virtual directory.

To access the solution-specific configuration files within web documents, such as `xhqNavbar.html`, you will need to reference this virtual directory.

Note the following example.

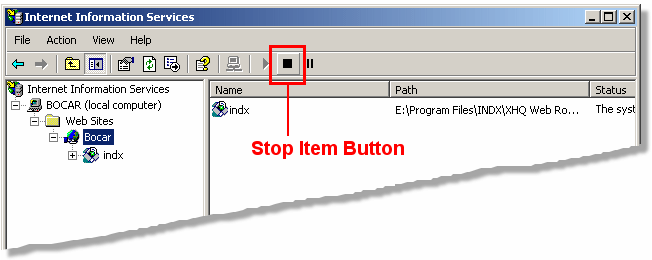
```
<script type="text/javascript" src="custom/appletsettings.js"> </script>
```

If you need to replace (switch-out) the repos folder, you must first stop the web site where the `/indx/custom` virtual directory exists.

See the topic, *To stop a web site prior to replacing switching the repos folder*.

To stop a web site prior to replacing (switching) the repos folder

1. Open the **Internet Information Services (IIS) Manager**.
2. **Expand** the local computer and Web Sites.
3. **Select** (highlight) the web site in which the `/indx/custom` virtual directory exists.
4. Click the **Stop Item** button.



Your next step is to replace the repos folder. Then, go back to the IIS Manager, select the same web site in step 3. Select (highlight) the web site in which the `/indx/custom` virtual directory exists, and click the Start Item button.

How to Make Custom Configurations

Depending on location of your web server in relation to the XHQ Server, use the following to customize the desired settings file in the **`%XHQ_WEB_DATA%\repos\conf\web`** directory, which by default is `C:\XHQ\data\repos\conf\web`.

If the Web Server is on the SAME machine as the XHQ Server

The XHQ install program will:

- Set up the **`%XHQ_WEB_DATA%\repos\conf\web`** folder.
- Write examples to the **`%XHQ_WEB_DATA%\repos\conf\web\examples`** folder.
- Create the **`/indx/custom`** virtual directory that points to the **`%XHQ_WEB_DATA%\repos\conf\web`** folder.

To make custom configurations when the web server is on the same machine as the XHQ server, do one of the following:

- Copy the desired file from the **`\conf\web\examples`** folder into the **`\conf\web`** folder and modify the file as needed.

or

- Create the desired file, such as the **`appletsettings.js`** file, in the **`\conf\web`** folder (if an empty file is not already present) and add your custom settings.

If you need to replace (switch-out) the repos folder, you must first stop the web site where the `/indx/custom` virtual directory exists.

See the topic, *To stop a web site prior to replacing switching the repos folder*.

Setting the Request Mode

Use the following procedure to define the transportation protocol between the XHQ View Server and the XHQ HTML5 Solution Viewer.

To set the request mode

1. Open the **`web.config`** file, which by default is in the `%SystemDrive%\Program Files (x86)\XHQ\XHQ Web Root\webapps\sv` folder.
2. Locate the **`<appSettings>`** tag.

```
Web.config
1   <?xml version="1.0" encoding="UTF-8"?>
2   <configuration>
3     <location path="SVRouterHandler.ashx">
10      <location path="SVRouterHandlerRest.ashx">
15    <system.web>
22    <system.webServer>
49    <appSettings>
50        <add key="debug" value="false" />
51    <!--<add key="requestMode" value="socket"/> -->
52        <add key="requestMode" value="web" /> <!--request modes = web, socket.
          'web' for REST, 'socket' for WS-->
53    </appSettings>
54  </configuration>
```

3. For **`requestMode`**, enter either of the following **value**:
   - "web"
     For requests through **web services**. This is the default.

   or

   - "socket"
     For requests through **websockets**.
4. **Save** `web.config`.

To change the reconnect parameters, go to the topic, *Additional Global Settings*, and see the `ReconnectAttemptsCount` and `ReconnectAttemptsInterval` properties.

12 | Localization and XHQ

Things to Note about Localization

- Currently, the following features are not supported on Asian locales (but will be supported in an upcoming XHQ release).

Kerberos Authentication	Performance Management
OPC UA Server	Units of Measure
XHQ SDK	Charting
XHQ IDG	Table Control

- With regards to import/export, because UTF filenames are used, they can be unreadable unless a suitable program (such as 7-Zip File Manager) is used.

Setting the XHQ Workbench Locale

To set the XHQ Workbench locale, open the **`clientsettings.properties`** file (which, by default, is located in the `%XHQ_DEV_HOME%\repos` directory).

Add the line: **`Workbench_Locale=<locale_abbreviation>`**

Where: **`<locale_abbreviation>`** is one of the following abbreviations:

Abbreviation	Language
en	English
de	German
ar	Arabic
es	Spanish
fr	French
it	Italian
ja	Japanese
ko	Korean
pt	Portuguese
ru	Russian
th	Thai
zh	Chinese

Default: `Workbench_Locale=en`

Example for German: `Workbench_Locale=de`

Placing the Language Files

To launch the localized version of the **html-based dialogs** from the XHQ Solution Viewer (for example, the "User Preference" dialog box from the XHQ NavBar), put the appropriate language file onto the web server at the site that is designed to use the given language.

The default language file is English and additional language files are available for French, Italian, Spanish, German, Portuguese, Arabic, Russian, Chinese, Japanese, Korean, and Thai.

The "User Preference" Dialog Box

To place the language file

Do the following procedure for both the production system and any development systems.

1. Go to the **%XHQ_SERVER_REPOS%** directory (which is typically `<systemdrive>:\XHQ\data\repos`) and navigate to the **conf\web\i18n** folder.
2. **Copy** the *language-specific* **xhqI18nVariables_xx_XX.js** file.

   Example: For *Italian*, select the `xhqI18nVariables_it_IT.js` file.

Filename	ISO-639 Language Codes		ISO-3166 Country Codes	
xhql18nVariables_ar_SA.js	ar	Arabic	SA	Saudi Arabia
xhql18nVariables_de_DE.js	de	German	DE	Germany
xhql18nVariables_en_US.js	en	English	US	United States
xhql18nVariables_es_ES.js	es	Spanish	ES	Spain
xhql18nVariables_fr_FR.js	fr	French	FR	France
xhql18nVariables_it_IT.js	it	Italian	IT	Italy
xhql18nVariables_ja_JP.js	ja	Japanese	JP	Japan
xhql18nVariables_ko_KR.js	ko	Korean	KR	Korea
xhql18nVariables_pt_PT.js	pt	Portuguese	PT	Portugal
xhql18nVariables_ru_RU.js	ru	Russian	RU	Russia
xhql18nVariables_th_TH.js	th	Thai	TH	Thailand
xhql18nVariables_zh_CN.js	zh	Chinese (Simplified, Traditional)	CN	China

3. Go to the **%XHQ_WEB_HOME%\indx\scripts** directory (which by default is `C:\Program Files (x86)\XHQ\XHQ Web Root\indx\scripts`) and paste this file.

In the next step, you will replace the existing *original* `xhqI18nVariables.js` file. If you want to keep a backup copy of this *original* .js file, then rename it to, for example, `xhqI18nVariables_bkup.js` before doing step 4.

4. Then, **rename** the **xhqI18nVariables_xx_XX.js** file you pasted in step 3 to **xhqI18nVariables.js** by removing the locale-specific suffix.

Although component names are not localized, the Description field associated with each component is and will display in the given language. Therefore, if you are using the XHQ NavBar, you can also set the labels of the XHQ NavBar tabs to display in the given locale by setting the **UseComponentMemberDescription** global property to TRUE. By setting this property to TRUE, the component description is displayed instead of the component name.

Displaying Kanji

To properly view Japanese contents in Kanji, set the following global property:

IEBrowserFont=MS Gothic

For more information, go to the topic, *Using globalsettings.properties*.

If you are using the XHQ NavBar, you can also set the labels of the XHQ NavBar tabs to display in Kanji by setting the **UseComponentMemberDescription** global property to TRUE. By setting this property to TRUE, the component member description is displayed instead of the component member name.

Component and component member names are not localized. However, the Description field (associated with each component member) is and will display in the given language (if a description is given).

To display the NavBar tabs in the appropriate locale using component member descriptions

The following procedure applies to any of the supported languages.

1. Place the appropriate language file onto the web server at the site that is designed to use the given language.

   The procedure to do this can be found in the topic, *Placing the Language Files*.

2. Open the **globalsettings.properties** file in a text editor (such as Notepad).
   The `globalsettings.properties` file allows you to set solution-wide options. This file is stored at the location specified by the environment variable **%XHQ_SERVER_REPOS%** on the Enterprise Server (which is typically C:\`XHQ\data\repos`).
3. Find and edit, or add, the following property:

   UseComponentMemberDescription=true

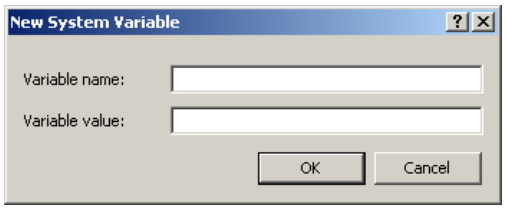
   The default setting for the **UseComponentMemberDescription** property is false (in which case, the non-localized component member name is displayed on the XHQ NavBar tabs).

4. **Save** and close the text editor.
5. Start a new session of the XHQ Solution Viewer.

If the Solution Server is acting as a client to Oracle (through the Oracle connector or the Oracle cache) and you experience a problem with displaying Kanji, use the following procedure to create the necessary environment variable.

To properly display Kanji (from the server)

1. From the machine where the XHQ Solution Server resides, open the **Control Panel** and click on **System**.
   The "System Properties" dialog box appears.
2. In the **Advanced** tab, click **Environment Variables**.
3. Under **System variables**, click **New**.
   The "New System Variable" dialog box appears.



4. In the **Variable name** box, enter `NLS_Lang`.
5. In the **Variable value** box, enter **JAPANESE_JAPAN.JA16SJIS**.
6. Click **OK** to close through the dialog boxes.

Displaying Localized Calendar Formats

The following procedure enables the XHQ High-Performance Database to display localized calendar formats. This is done by creating a new environment variable.

This procedure must be done prior to installing the XHQ Server. If the server is already installed, you must shut it down prior to making these changes and re-install.

Currently, this feature only supports the UNITED ARAB EMIRATES localized calendar format.

To set the localized calendar format for the high-performance database

1. From the **Control Panel**, click **System**.
   The "System Properties" dialog box appears.
2. In the **Advanced** tab, click **Environment Variables**.
   The "Environment Variables" dialog box appears.
3. Under **System variables**, click **New**.
   The "New System Variable" dialog box appears.
4. For **Variable name**, enter **`XHQ_DB_TERRITORY`**.
5. For **Variable value**, enter **`UNITED_ARAB_EMIRATES`**.

   Capitalize both Variable name and value.

6. Click **OK** through the dialogs to save your changes.
7. **Install** (or re-install) the XHQ Server.

In this particular example, the expected results are as follows:

This database parameter . . .	Changes from this . . .	To this . . .
NLS_TERRITORY	AMERICA	UNITED ARAB EMIRATES
NLS_CALENDAR	NULL	English Hijrah

To set the calendar format back to US, shutdown the server, change the **Variable value** from `UNITED_ARAB_EMIRATES` to `AMERICA` and re-install the XHQ Server.

About Locale and Arithmetic Expressions

It is important to note that locale-specific arithmetic expression syntax is not supported in the XHQ Expression language. Therefore, do not use a comma as a decimal point separator in a server-side expression.

> The XHQ Expression language is a specified programming language similar to C or Visual Basic in that the syntax does not change depending on the programmer's locale.

Consider the following.

Correct Syntax:`10 * 2.7`

Incorrect Syntax:`10 * 2,7`

The arithmetic expression `10 * 2,7` does not result in a product of 27 in XHQ. Instead, the digits to the right of the comma are ignored. So, this expression results in a product of 20, which is incorrect.

Using Halfwidth or Fullwidth Forms

Graphic symbols (for example, Kanji in Japanese) are either halfwidth or fullwidth characters. A halfwidth character takes up half the width of a fullwidth character. By default, XHQ uses fullwidth characters.

> The Unicode block range U+FF00-FFEF consists of fullwidth and halfwidth symbols (forms).

> The properties file used in the following procedure is not included with the XHQ install and must be created by the user. Changes to these parameters require server **restart**.

To create the properties file

1. Using a text editor (such as Notepad), **create** the `customtranslations.properties` file.
2. **Save** the file in location specified by the environment variable `%XHQ_SERVER_REPOS%` on the server (which by default is `%SystemDrive%\XHQ\data\repos`).

To use halfwidth symbols

1. Open the `customtranslations.properties` file.
2. Enter the following key/value parameter for **halfwidth** symbols:

```
My_Trends=\uff8f\uff72\uff84\uff9a\uff9d\uff84\uff9e
```

> The **default fullwidth** key/value parameter is
> `My_Trends=\u30de\u30a4\u30c8\u30ec\u30f3\u30c9`

3. **Save** the properties file.
4. **Restart** the server.

13 | Distributed Solutions

In this section, you will learn how to add, deploy, and maintain multiple XHQ solutions as a single solution.

Important Things to Note

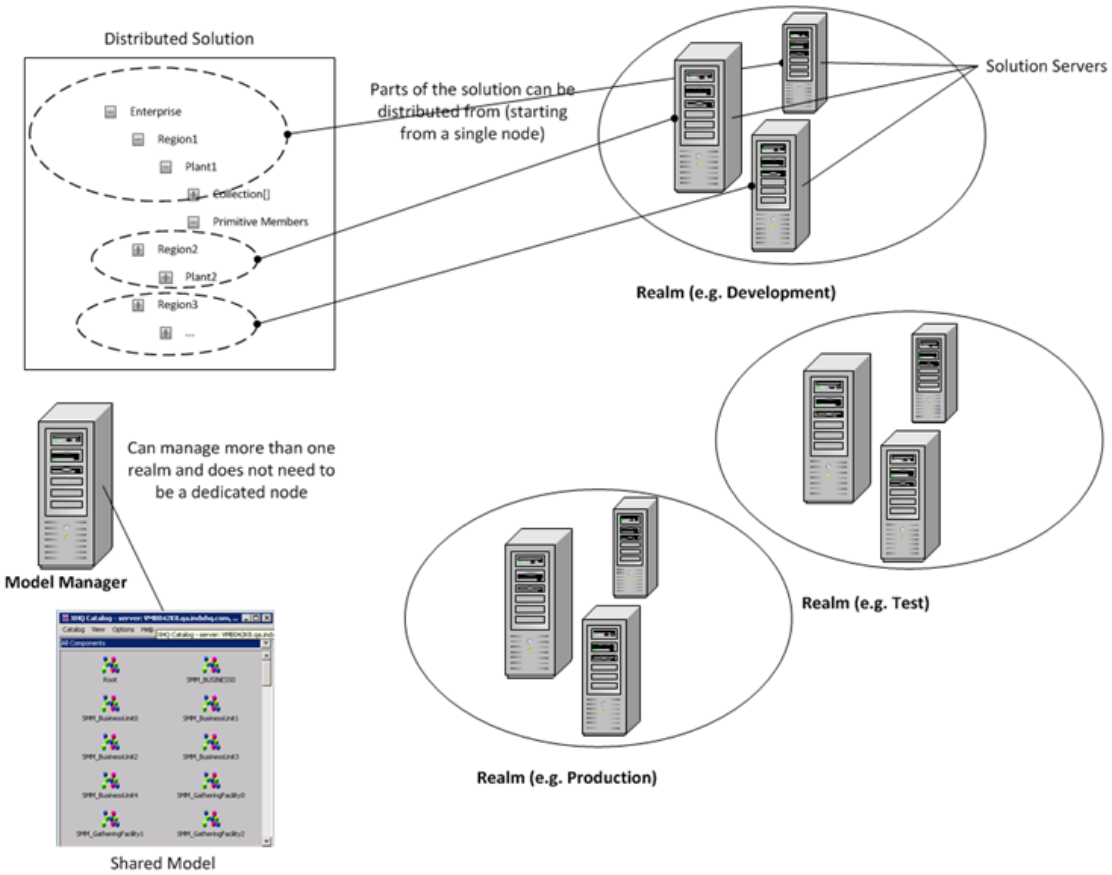
- A license key is required to enable this feature.

  For information on how to acquire the license key, go to the topic, *The License Manager*, located in the XHQ Installation Guide.

- In this release phase, a limited distribution is supported. In this case, two servers inclusive of the server hosting Root solution (the root server).
- Converting an existing solution into a distributed solution using the XHQ Solution Administration Utility is not supported.

Overview

The following image illustrates how this works.



Distributed Solutions Architecture

To protect the integrity of the model data, distribution is done from a **single point**. Distribution may be done from the XHQ Administration Utility, various xhqci commands, or web services.

For more information on distribution using web services, see the topic, *Using the Administrative Web Service, XhqAdminWs*, located in the **XHQ SDK Reference Guide**.

Distributed Solutions Terminology

Term	Definition
Model Manager	This is the server that controls the distribution of the model between the distributed solution servers.
Realm	A set of XHQ Servers that together share a solution and information model.
Remote Objects	These are objects processed on the remote server.

About Realms

A **Realm** is set of XHQ Servers that together share a solution and information model (which consists of XHQ Workbench components, views, security roles, and units of measure. The solution is distributed across the realm such that each server can host a ﬁpartﬂ of the solution tree. Any object or member in the solution tree can be accessed from any server within the XHQ solution. This includes objects, primitive members and member collections (or linked collections).

Multiple realms are supported and can share a single model.

Things to Note

- Any Solution Server in a realm can serve as an entry point, for external clients/applications, to the distributed solution hierarchy within that realm.
- Connections are local to the specific Solution Server that hosts them.
- Global collections, Tags, and Aliases are no longer global in scope. They can only be accessed inside of the Solution Server that hosts them.

For information on realm administration using the XHQ Administration Utility, go to the topic, *To manage the realms*.

About the Model Manager

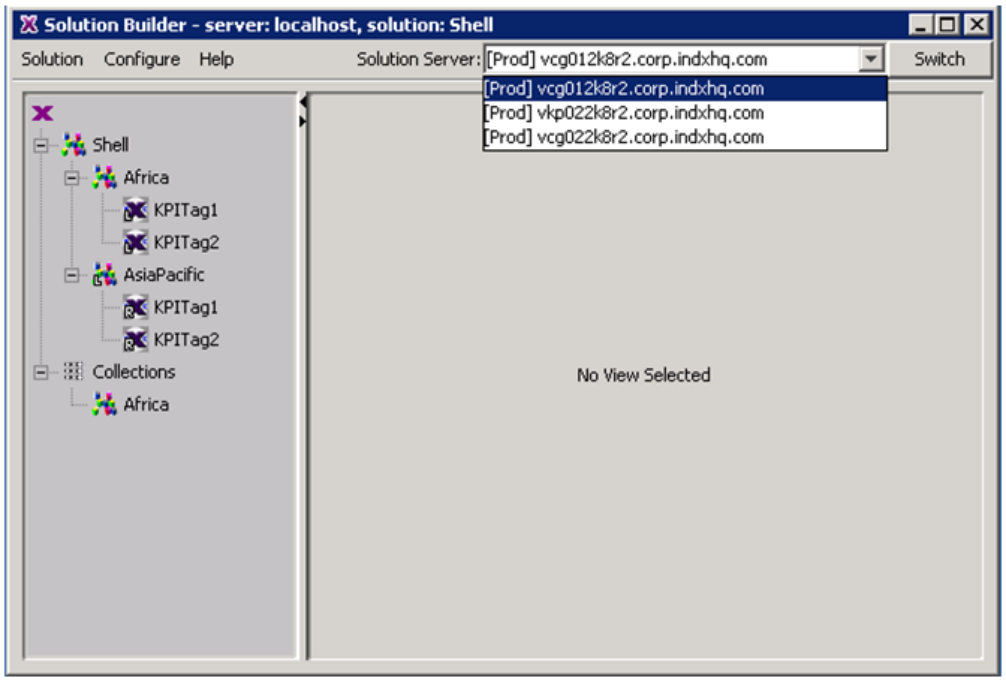
The **Model Manager** is a server that manages the configuration of the Distributed Model. From this node:

- The realm(s) and servers are defined.
- The Model is maintained and distributed.

Multiple realms can be managed. For example, you may have a Development, a Test, and a Production realm (as shown in the Distributed Solutions Architecture image). Consequently, a single model is used for all realms.

Solution Management

The XHQ Solution Builder can be pointed to any Solution Server within any realm that is part of the Distributed Solution.



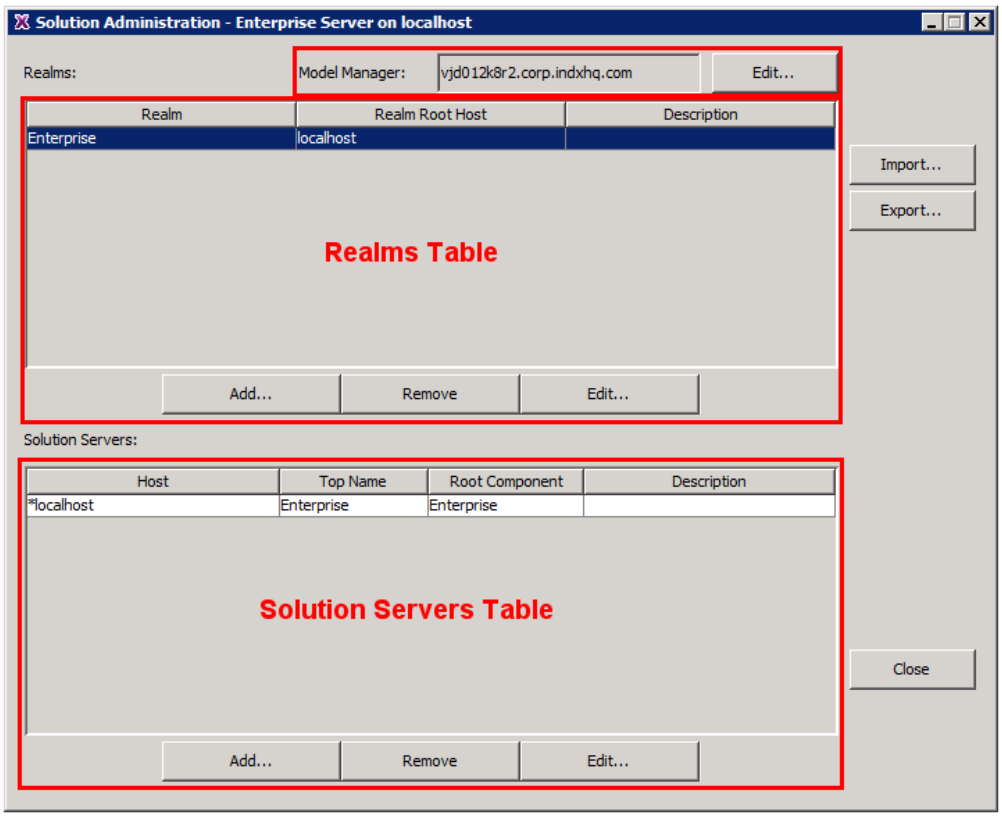
XHQ Solution Builder Example

Things to Note

- If Distributed Solutions are detected, the XHQ Solution Builder displays the Solution Server box, which allows you to switch between the Solution Servers.
- From the navigation tree, to identify which nodes are local and which are remote, the icon is appended with a letter overlay. The letter "L" stands for *link*. It refers to the root of another server's tree in the Distributed Solution.
  The letter "R" is for *remote*. The given object or member is hosted by another server that is not the server to which the XHQ Solution Builder is connected. Therefore, the object or member cannot be configured since configuration is only allowed for local items.

Using the XHQ Solution Administration Utility

The XHQ Solution Administration Utility allows you to add solutions to a distributed configuration, as well as set the XHQ Solution Servers associated with the given solution.



XHQ Solution Administration Utility User Interface

XHQ Solution Administration Utility Interface

Control	Description
Model Manager	Set the **Model Manager** for this realm. This is the name of the server running the Model Manager for this solution. Although the default value is `localhost`, it actually cannot be used in a distributed solution. `localhost` may only be used for a single realm/single server configuration. **Note:** Use the full name of the server (for example, `ModelManager.acmeSite.enterprise.com`) or use `localhost`. If the solution is a "stand-alone" solution, use `localhost`. Things to Note: • The Model Manager server name and the server host name **must be fully-qualified domain names** (FQDN). • If a solution is not defined in the Solution Administration Utility,

Control	Description
	then it is not possible to edit the Model Manager. • If the Model Manager is changed to an invalid host (that is, the host does not exist or the host has the wrong solution), then the XHQ Solution Administration Utility becomes locked in the Read-Only mode and you are no longer able to make changes. To unlock the Model, see the steps below under "Important Things to Note." • You may create multiple realms using the same Model Manager. These realms will have the same model (XHQ Workbench).
Realms Table:	
Realm	The name of the set of distributed solutions to which the given solution belongs (for example, Production, Development, Test, and so forth). **Note:** Because solution names may not be unique, realms help in uniquely identifying solutions.
Realm Root Host	The root host server name.
Solution Servers Table:	
Host	This is the name of the server in which the Solution Server resides. Use the fully-qualified domain name (FQDN). An asterisk (*) in front of the host name indicates that the server is the dedicated root of the distributed solution. For a portable, single solution/single server configuration, the server name may be "localhost."
Top Name	This is the member of the top-level component for the given server. **Note:** Each solution must have at least one server that is mapped to the <root> member of the solution. In the case of a "stand-alone" solution, the server must be mapped to the <root> member and set to "localhost."
Root Component	The name of the root object for this server. Note the following: • If this server is a solution root server, this name defines the name of the entire distributed solution. • Else, if this server is just a server in the realm, this name is the name of the node in the solution tree after the server is linked in the overall solution. However, if the server is not linked, this name is the local solution name.

Important Things to Note

- If a server is not a Model Manager, then all XHQ Workbench and XHQ Solution Administration Utility functions are "Read Only"
- To unlock the Model, run the **`xhqci debugmodel`** command and select command **3 – Set As Model Manager** with an attribute of **true**.

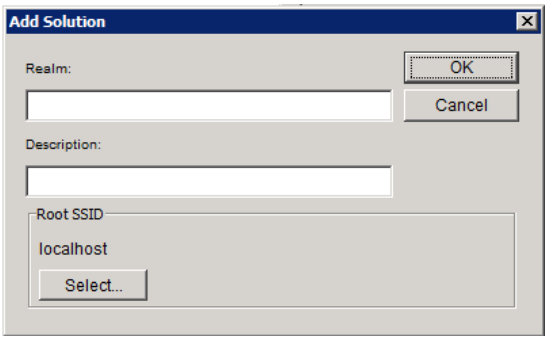
- One server must always point to the <root> member of the solution.

To access the Solution Administration Utility

From the **Start menu**, go to **All Programs** > **XHQ Development** > **Administration** and click **Solutions**.

To add a realm

1. From the XHQ Solution Administration Utility, under the **Realm** table, click **Add**.
   The "Add Solution" dialog box appears.

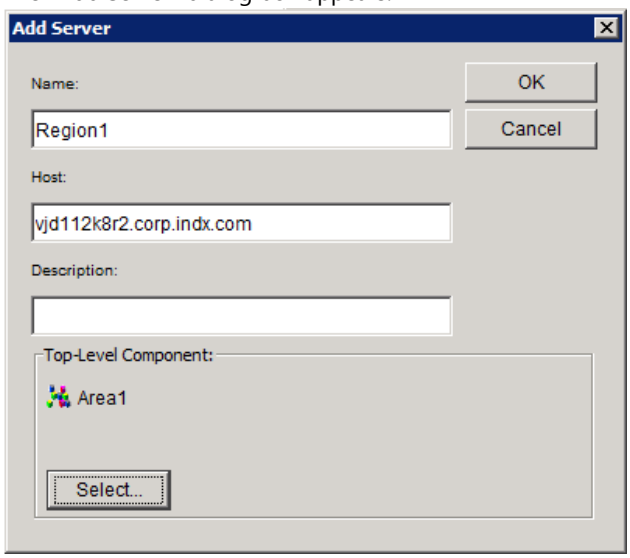


3. Enter the **Realm** to which this solution belongs.
   This is the unique XHQ name.
4. Enter a **Description**.
5. Click **Select** to choose the **Top-Level Component** of the solution.
6. Click **OK**.

Your next step is to add a solution server.

To add a solution server

1. Under the **Solution Servers** table, click **Add**.
   The "Add Server" dialog box appears.

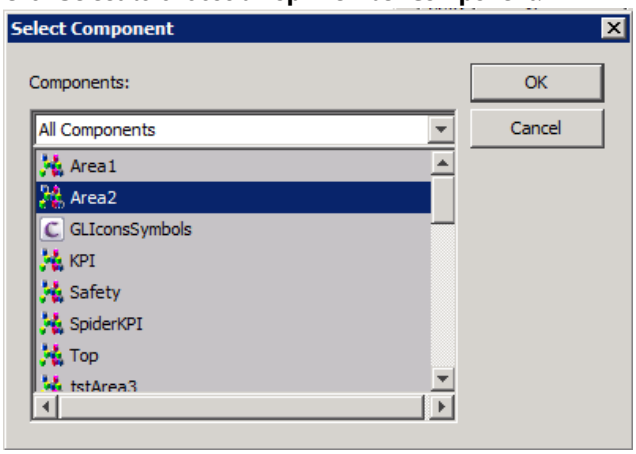


2. Enter the server **Name**.

   This is the name of the root object for this server. Note the following:

   - If this server is a solution root server, this name defines the name of the entire distributed solution.
   - Else, if this server is just a server in the realm, this name is the name of the node in the solution tree after the server is linked in the overall solution. However, if the server is not linked, this name is the local solution name.

3. Enter the unique **Host** name of the server where the Solution Server resides.
   Use the full name of the server (for example, `ModelManager.acmeSite.enterprise.com`) or use `localhost`.
4. Enter a **Description**.
5. Click **Select** to choose a **Top-Member Component**.



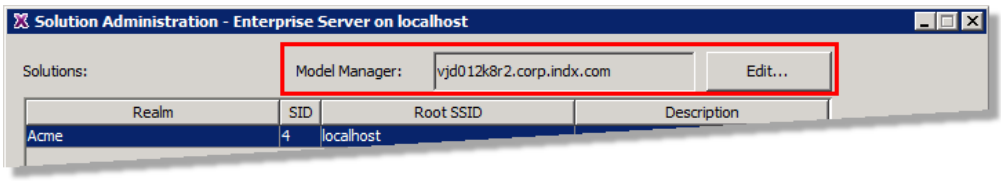
   Each solution must have at least one server that is mapped to the <root> member of the solution. In the case of a "stand-alone" solution, the server must be mapped to the

<root> member and set to "localhost".

6. Click **OK**.

Important Things to Note

- If the XHQ Solution is associated with multiple solution servers, Then the XHQ Solution Builder displays an additional box that enables you to identify the Model Manager.



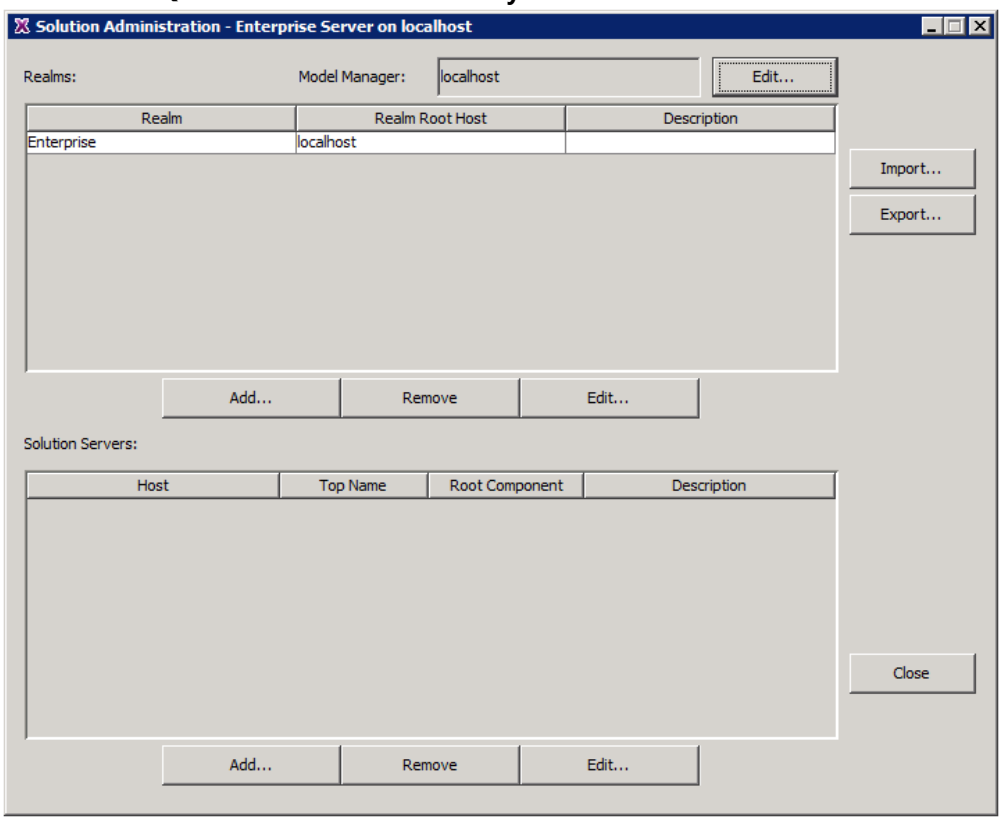
XHQ Solution Builder: Selecting a Model Manager

- The XHQ Solution Builder automatically connects to the previous server with a successful connection. If the Model Manager you choose is unavailable, an error message appears.
- Remote objects are marked with an arrow.

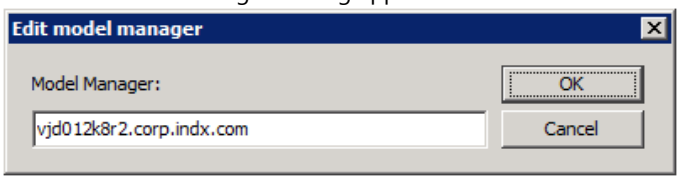
EXAMPLE: SOLUTION ADMINISTRATION CONFIGURATION

In this example, you will learn how to set-up a root server with two addition solution servers. **First, you must define the solution and start it on the root server.** Then, continue with the following steps.

1. Launch the **XHQ Solution Administration Utility**.



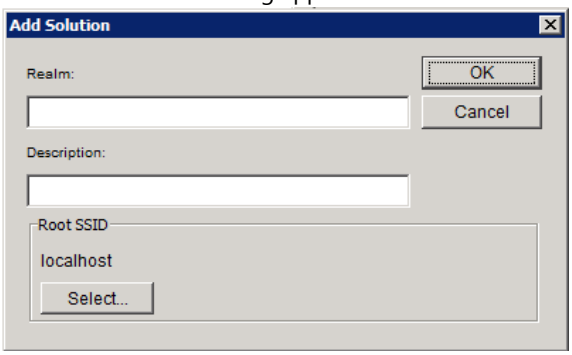
2. At the top of the dialog, for **Model Manager**, click **Edit**.
   The "Edit Model Manager" dialog appears.



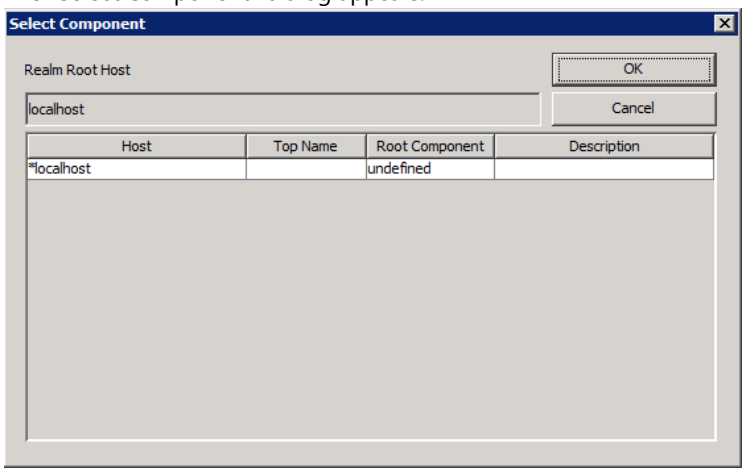
To use Distributed Solutions, the Model Manager cannot be set to localhost.
Any server in the distribution could be the Model Manager, not only the root server.

3. Enter a **fully qualified domain name** (FQDN) and click **OK**.
4. Click **OK** for the subsequent messages and return to the XHQ Solution Administration Utility.

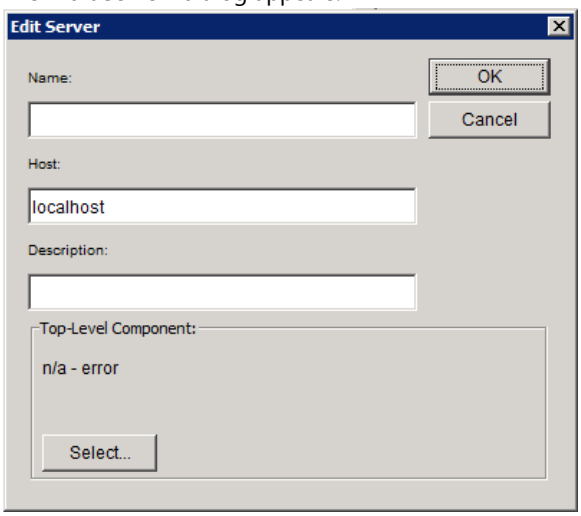
5. Under the **Realms** table, click **Add**.
   The "Add Solution" dialog appears.



6. Enter a **Realm** and click **Select**.
   The "Select Component" dialog appears.



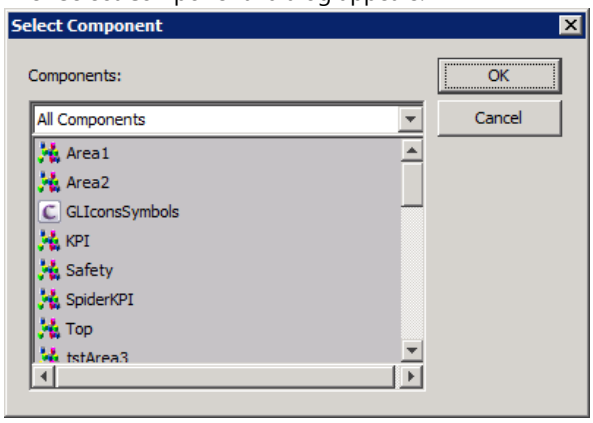
7. Select the **Root Host** and click **OK**.
8. In the **Solutions table**, select (to highlight) the root solution.
9. From the **Solution Servers table**, select (to highlight) the **Solution Server associated with the root server**.
10. Directly below the Solution Servers table, click **Edit**.
    The "Edit Server" dialog appears.



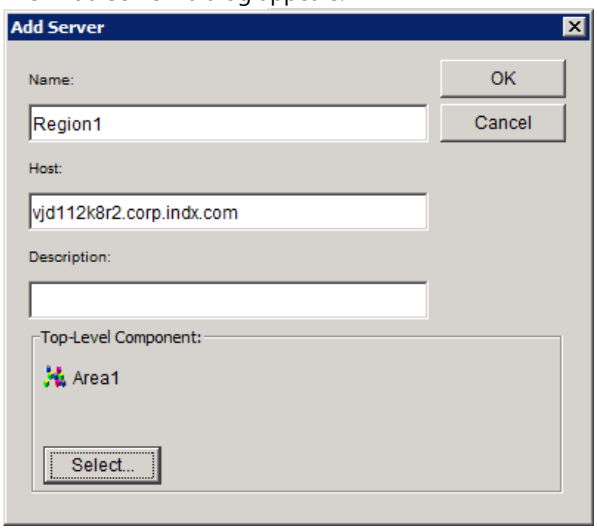
11. Enter the root server **Name**.
    This root server name defines the name of the Distributed Solution.
12. For **Host**, enter a fully-qualified domain name (FQDN).

    This Host does not have to be the same as the Model Manager.
    The Model Manager does not have to be a part of the Distributed Solution.

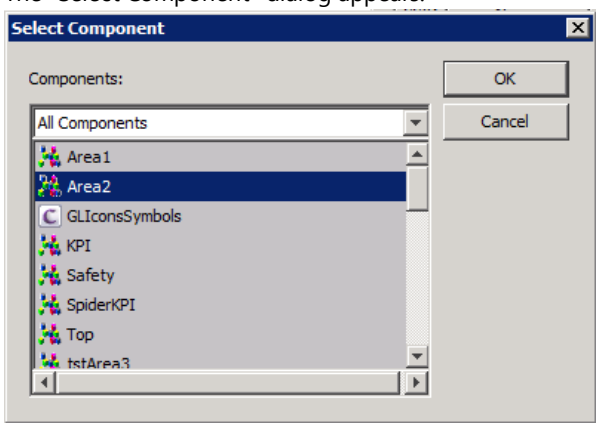
13. For **Top-Level Component**, click **Select**.
    The "Select Component" dialog appears.



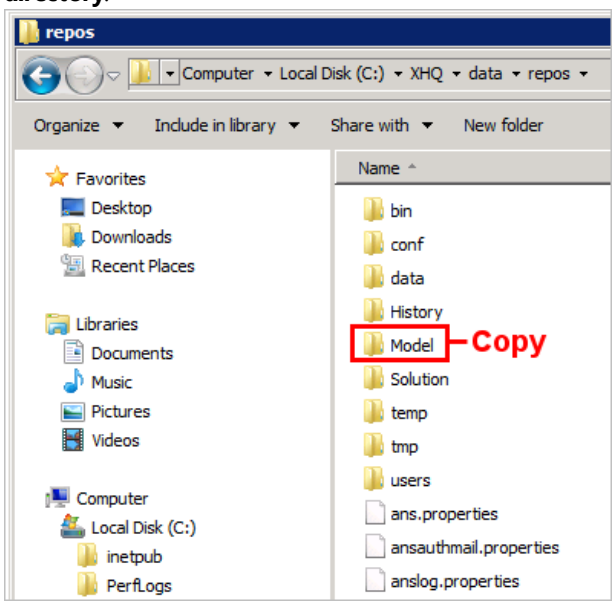
14. **Select** the top-level component and click **OK**.
    The next step is to add two more solution servers, each with different top-level components.
15. Below the Solution Servers table, click **Add**.
    The "Add Server" dialog appears.



16. Enter the (first of two) server **Name** and **Host** and click **Select**.
    The "Select Component" dialog appears.



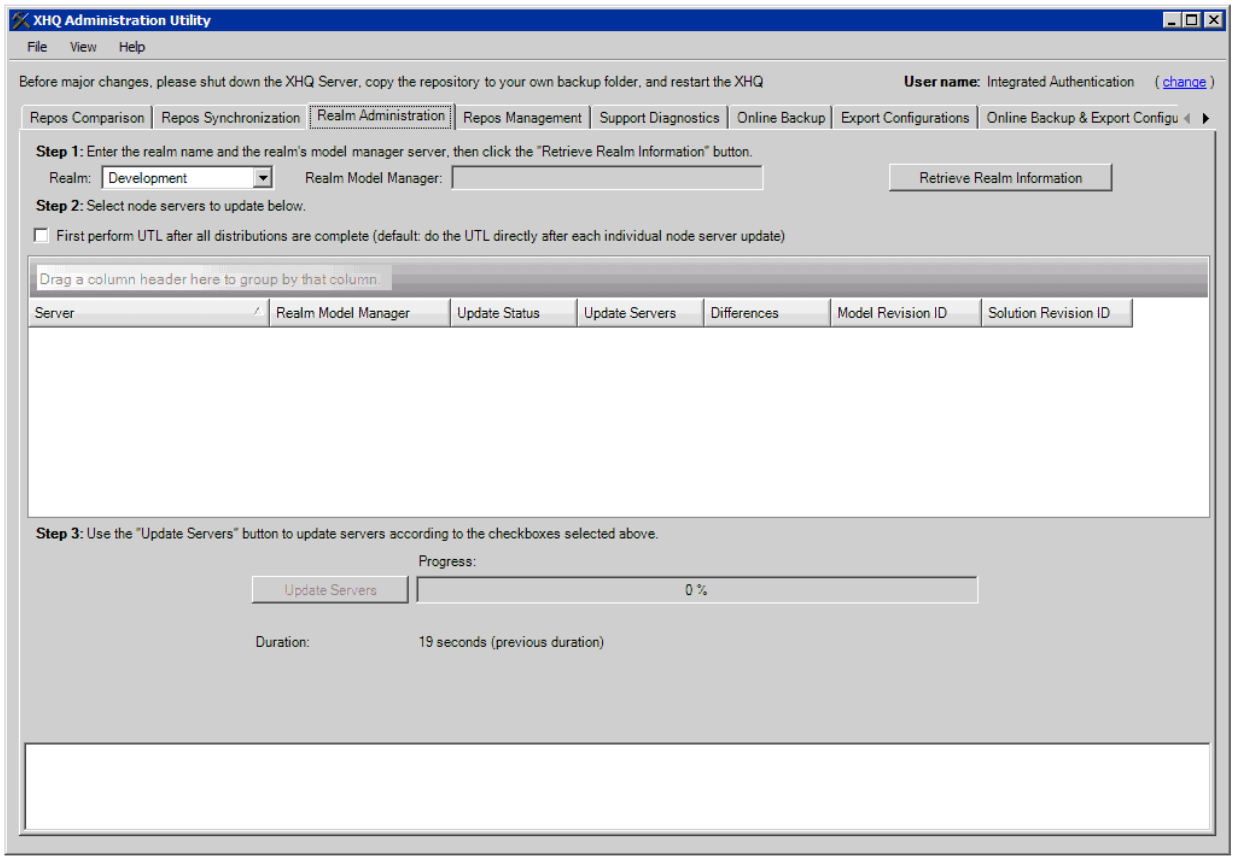
17. Click **OK**.
    This returns you to the XHQ Solution Administration Utility.
18. **Repeat steps 15-17** for the second of the two servers; entering a different Name and Host, and selecting a different top-level component.
    The servers defined in this example distribution are named "Region1" and "Region2".
19. Close the XHQ Solution Administration Utility and **stop** the Model Manager server.
    You may or may not have the Model Manager as part of the Distributed Solution. In either case, the Model Manager server must be stopped in order to copy the Model directory to ALL specified (added) servers in the distribution.
    Next, you must manually replace the Model directory of ALL specified servers (which, in this example, are the two servers you added) with the Model directory of the Model Manager server.
20. To each of the added servers, install the empty repos (which, by default, can be found in the `C:\XHQ\data\` directory).
21. Then, from the **Model Manager**, go to the XHQ **repos** directory (`%XHQ_SERVER_REPOS%`) and **copy the Model directory**.



22. To each of the servers defined in the distribution, go to the repos directory and replace the existing Model directory with the one copied from the Model Manager.
23. For the root server and each server defined in the distribution, set the global property **`DisableExtendedNamespaceUI`** to **false**.
24. **Start** the servers.

Configuring the XHQ Administration Utility

Once the XHQ Solution Administration Utility has been configured for distributed solutions, your next step is to manage them using the XHQ Administration Utility.



XHQ Administration Utility: Realm Administration Tab

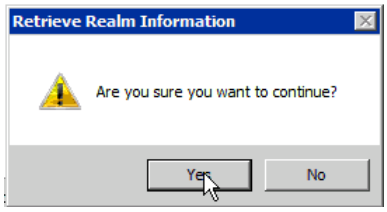
The Realm Administration tab is only enabled in the XHQ Administration Utility if distributed solutions are detected.

To access the XHQ Administration Utility

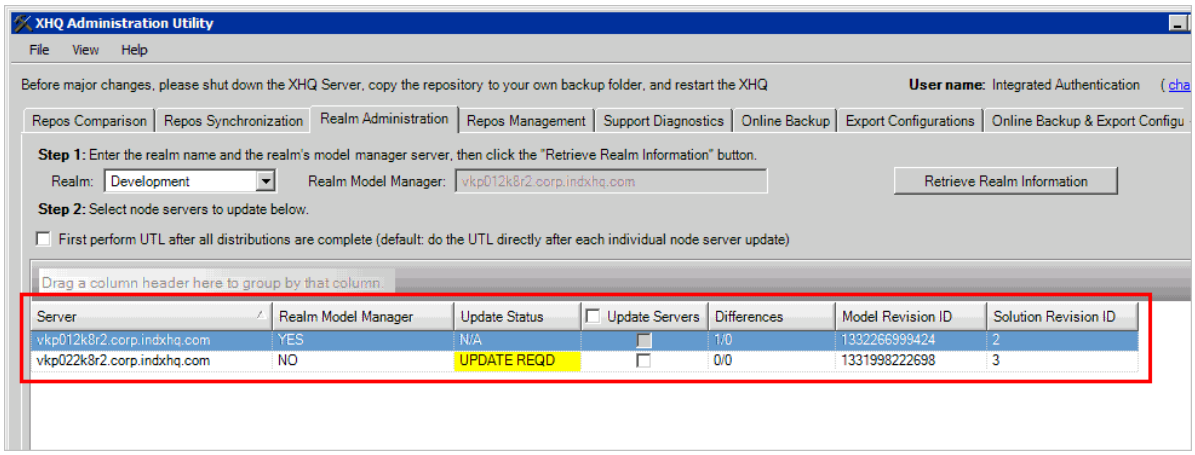
From the **Start menu**, go to **All Programs** > **XHQ Development** > **Administration** and click **Administration Utility**.

To manage the realms

1. From the XHQ Administration Utility, click the **Realm Administration tab**.
2. Select a **Realm**.
3. Click **Retrieve Realm Information**.
   A message appears, asking you to confirm this action.



4. Click **Yes**.
   When the retrieval is complete, a list of node servers appears in the table.

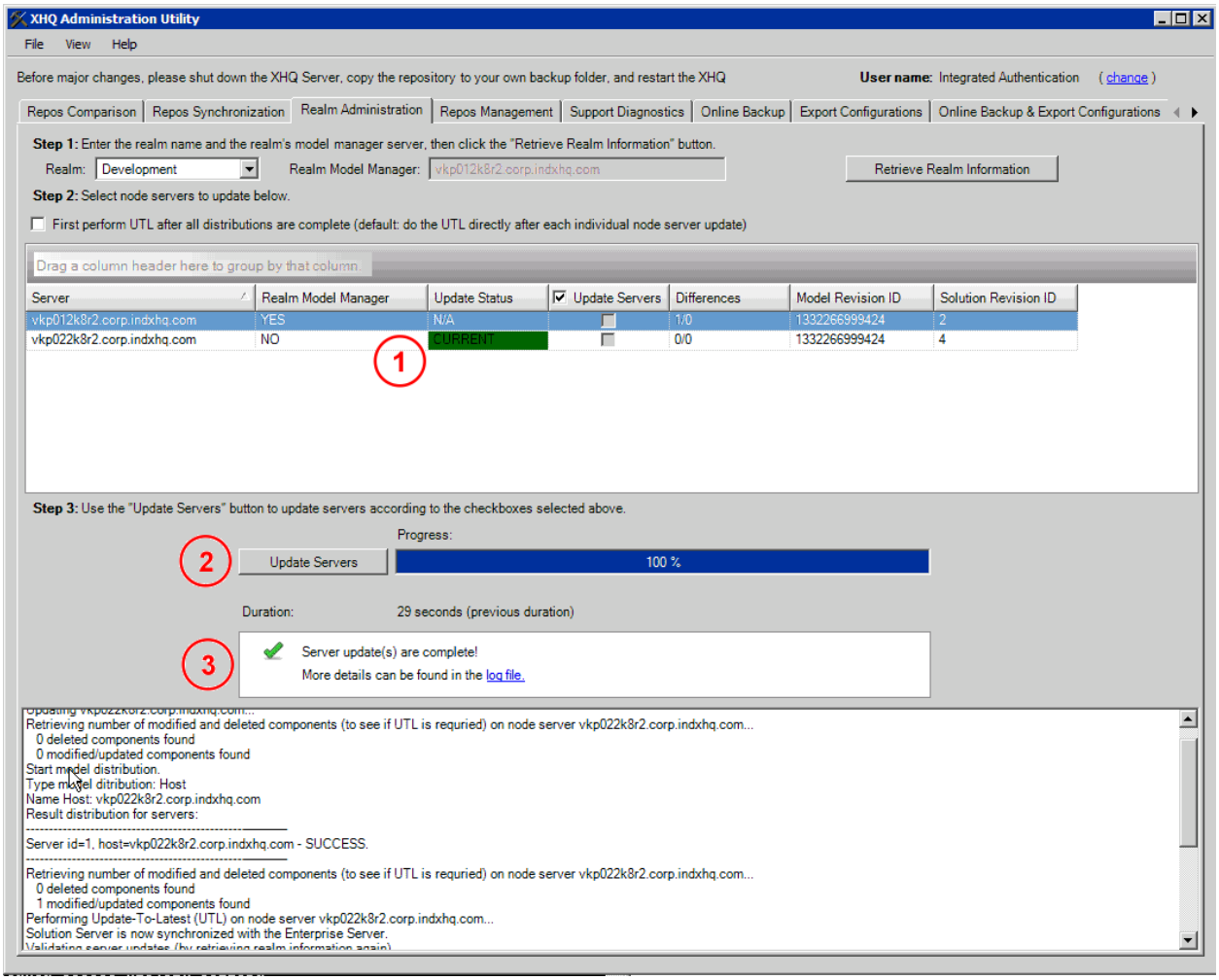


The possible **Update Status** results are:

- **N/A**
  For the Realm Model Manager server.
  - **UPDATE REQD** (color-coded yellow)
    An update is required for the given server.
  - **CURRENT** (green)
    The server is current.
  - **ERROR**(red)
    You must check the solution manually to determine why the error is occurring.

5. If an **update** is required for a given server, under the **Update Servers** column, check the box for that server.
6. OPTIONAL
   Above the table is a checkbox which enables a UTL action before all distributions are complete (instead of doing the UTL after each individual server update).
7. Click **Update Servers**.

Upon completion, note the following:



Update Servers Complete Indicators

Item	Description
1	The update status of non-realm model manager servers is "Current".
2	The progress indicator shows 100% completion.
3	In the backup message box: • A check mark appears if the servers were successfully updated. • A warning icon appears if corruptions were found or if the update was unsuccessful. For details of this process, click the log file link or scroll the contents of the output pane, which displays the status during this process.

Managing Servers in Distribution

Use the following procedures to manage the servers in distribution.

To add a server to the realm

1. **Configure** the new server and **install** the appropriate version of **XHQ**.

   Do not start XHQ yet. If started, shut it down.

2. Create an **empty repos** on the newly added server.
3. From this empty repos, **delete** the **Model** folder.
4. From the **Model Manager server**, add the new server to the realm by using the XHQ Solution Administration Utility or by importing.
5. On the Model Manager server, use **`xhqci backup`** with a **non-ZIP** destination.
6. **Copy** the Model directory from the backup to the repos on the added server.
7. **Start XHQ** on the added server.
8. From the **Model Manager**, distribute the model to the realm (to all servers in the realm).

   For the newly added server, the Distribution may report that the model is up-to-date.

At this point, all the running servers in the realm will update their lists of servers and attempt to establish a connection with the newly added server. Your next step is to build and link the new server to the distributed solution, as needed.

To remove a server from the realm

1. **Unlink** the server from the distributed solution.
2. From the **Model Manager server**, remove the server from the realm by using the XHQ Solution Administration Utility or by importing.
3. From the **Model Manager**, distribute the model to the realm (to all servers in the realm).

   The Distribution will also distribute the solution definition change to the removed server.

All running servers in the realm will update their lists of servers and drop connections to the removed server. The removed server will also update its list of servers. Based on being removed from the realm, the removed server will restart. It will start with an empty solution, since its solution is not defined any more.

To move a server to another realm

1. From the **Model Manager server**, remove the server from the realm and add it to another realm by using the XHQ Solution Administration Utility or by importing.
2. From the ***Model Manager***, distribute the model to both realms (to all servers in both realms).

   Since the moved server is considered to be removed from its original realm, this server may receive two Model Distribution requests; the second one will report that it is up-to-date.

All running servers in the first realm (where the server was removed) will update their lists of servers and drop connections to the removed server. The removed server will also update its list of servers. The removed server will restart and find itself in the new realm. It will establish connections with all servers in the new realm and create its solution as the new one. All servers in the second realm (where the server is added) will also update their lists of servers and establish connections to the added server once it restarts. Next, build and link the new server to the distributed solution, as needed.

In all cases, if one (or several) of the servers in the affected realm is not running, it will not receive the Model Distribution in time. The server will require the Model Distribution once it has started.

To change the hostname of a server

1. **Unlink** the server to be changed from the distributed solution.
2. **Configure** the new server and **install** the appropriate version of **XHQ**.

   Do not start XHQ yet. If started, shut it down.

3. Create an **empty repos** on the newly added server.
4. From this empty repos, **delete** the **Model** folder.
5. On the **Model Manager server**, change the hostname of the server to the new one by using the XHQ Solution Administration Utility or by importing.
6. On the Model Manager server, use **`xhqci backup`** with a **non-ZIP** destination.
7. **Copy** the Model directory from the backup to the repos on the added server.
8. **Start XHQ** on the new server.
9. From the **Model Manager**, distribute the model to the realm (to all servers in the realm).

   For the newly added server, the Distribution may report that the model is up-to-date.

At this point, all the running servers in the realm will update their lists of servers and attempt to establish a connection with the server based on its new hostname. They will also drop connections to the old host.

The XHQ Server with the removed hostname will no longer be accepted by the other servers in the realm and can be reconfigured.

You next step is to build and link the new server to the distributed solution, as needed.

Migrating an Existing Solution to Distribution

To migrate an existing solution to distribution, you will need to re-create the repos as a distributed solution.

1. First, **copy** the repos.
2. Then, using the **XHQ Administration Utility**, **export** all the **configuration files**.
3. **Stop** the solution.
4. Go to location of the repos copy directory.
5. **Delete** the Model and Solution folders.
6. **Restart** the solution.
7. First, **import** the **`xhq_1_of_7_repos_model_roles.xml`** configuration file (from step 2).
8. Then, **restart the XHQ server** before you continue to import the other configuration files.
9. Next, **import** the **`xhq_2_of_7_repos_model_with_templates_with_views_add_references.zip`** configuration file.
10. Using the XHQ Solution Administration Utility, manually configure the distributed solution.

    For more information on how this is done, see the topic, *EXAMPLE Solution Administration Configuration*.

11. Continue with the import of the following configuration files:
    - `xhq_4_of_7_repos_solution_connectors.zip`
    - `xhq_5_of_7_repos_solution_connection_processes.xml`
    - `xhq_6_of_7_repos_solution_connection_groups.xml`
    - `xhq_7_of_7_repos_solution_configurations.xml`

    You will notice that the `xhq_3_of_7_repos_solution_defintion.xml` file is not imported. This file includes the old Solution Administration configuration and is not needed for this migration process.

14 | Miscellaneous Topics

Section Contents

About XHQ Tray **361**

About the Digital Certificate **362**

About XHQ 64-bit **363**

Technical Details of the 64-bit Operation 364

The XHQ Service Configuration 364

The Windows XHQ Service **366**

The XHQ Keep Alive Service **368**

XHQ Service and WAKA Configuration **370**

XHQ Performance and Anti-virus Software **371**

Optimal Server Settings **372**

Virtual Server Environments **373**

Additional Security Considerations **374**

Clickjacking 374

Cross-site Scripting 374

About XHQ and Active Directory **375**

About the License Manager **375**

XHQ Logs **375**

About the Log Directory 375

About the Solution Viewer Log 375

Load View Performance Logging 376

IIS Logging 377

Customizing the Development Client Logs and Repos Directory Locations **378**

More about the launchsettings.bat files 378

JDBC Fetch Size **379**

About Hyper-Threading Technology and the XHQ System **380**

Using the debug.js File **381**

XHQ Workbench File Recovery **383**

About Security and File Recovery 383

For XHQ Applications Using AJAX 384

About the XHQ Internal Cache 385

Improved Performance with Storage Updates 386

Using the smck Utility 386

Changing the XHQ High-Performance Database Password 387

About the Password Utility (pwutil) 387

Maintenance of the Embedded Database TEMP Tablespace 392

About XHQTempRecreate.bat 393

Disabling the Data Recorder 394

About XHQ Tray

There are three major modes of operation for XHQ Tray on an XHQ Server:

- **Local Development Client**
  Shows server state in the tray icon and displays XHQ development tools. This does not require elevated or Administrator level.
- **Administrator**
  All functionality of Development Client mode and allows control of the XHQ Service and XHQ Server processes. Requires elevated or Administrator level access.
- **Remote Host Development Client**
  Acts like XHQ Tray on an XHQ Development Client installation. This does not require elevated or Administrator level.

To launch XHQ Tray with Administrator functionality, do any of the following:

- Right-click on `%XHQ_SERVER_HOME%\bin\xhq_tray.exe` and select **Run as administrator**.
- Run the `%XHQ_SERVER_HOME%\bin\xhq_tray_admin.exe`.
- Run either `xhq_tray` or `xhq_tray_admin` from an XHQ Shell that has been started in elevated mode.

About the Digital Certificate

XHQ contains a *digital certificate*that certifies the ownership of a public key by the named subject of the certificate.

This allows others (that is, relying parties) to trust the signatures or assertions made by the private key that corresponds to the certified public key.

Like other certificates of this type, the trusted certificate used in XHQ is time-limited by the signing authority. **This means that it must be periodically updated to remain valid**. This normally occurs during an XHQ upgrade.

About XHQ 64-bit

XHQ leverages the 64-bit architecture of Windows Server. The following XHQ Java-based processes run as 64-bit processes on a 64-bit JRE.

The following list is informative only and is subject to change at any time.

XHQ Server Processes:

- xhq_appserver.exe
- xhq_appsrvend.exe
- xhq_aserver.exe
- xhq_asrvend.exe
- xhq_cserver.exe
- xhq_csrvend.exe
- xhq_dbhist.exe
- xhq_dbhistend.exe
- xhq_eserver.exe
- xhq_esrvend.exe
- xhq_rdsserver.exe
- xhq_rdssrvend.exe
- xhq_routend.exe
- xhq_router.exe
- xhq_sserver.exe
- xhq_ssopt.exe
- xhq_ssrvend.exe

XHQ Connector Processes:

- xhq_jdbc.exe
- xhq_jdbccon.exe
- xhq_msql.exe
- xhq_msqlcon.exe
- xhq_odbc64.exe
- xhq_odbc64con.exe
- xhq_ora.exe
- xhq_oracon.exe
- xhq_sim.exe
- xhq_simcon.exe
- xhq_snmp.exe
- xhq_snmpcon.exe
- xhq_sqlserver.exe
- xhq_sqlservercon.exe
- xhq_xhq.exe
- xhq_xhqcon.exe
- xhq_xml.exe
- xhq_xmlcon.exe

The difference in name for the ODBC Connector Process (`xhq_odbc64.exe`) is required because both the 32- and 64-bit versions may have to run on a 64-bit installation simultaneously due to the way ODBC drivers are segregated into 32- and 64-bit versions. You can only load 32-bit ODBC drivers into 32-bit processes and 64-bit drivers into 64-bit processes. Having the alternate name allows you to select which version to run by changing the process name in the connection configuration.

XHQ Miscellaneous Utilities:

- xhq_xhqci.exe
- xhq_xhqci_s.exe
- xhq_xhqci_scon.exe
- xhqtagsync.bat

The "XHQ Miscellaneous Utilities" are included to allow the XHQ Tag Sync to run in 64-bit mode.

Technical Details of the 64-bit Operation

The 64-bit support depends on two execution environments being available on a 64-bit operating system. The 64-bit versions of Windows have two different **`Program Files`** directories: one is for native 64-bit applications, and the other is for legacy 32-bit applications. There are two environment variables associated with these paths.

- (For 64-bit applications)

  `%ProgramFiles% = "C:\Program Files"`

- (For 32-bit applications)

  `%ProgramFiles(x86)% = "C:\Program Files (x86)"`

The installation on a 64-bit operating system also configures the 64-bit XHQ Server environment. The XHQ 64-bit support environment is comprised of a 64-bit JRE and some 64-bit versions of XHQ DLLs and a system environment variable. The 64-bit JRE is located in the `%XHQ_SERVER_HOME_X64%\jre\bin` directory and the 64-bit XHQ DLLs are located in the `%XHQ_SERVER_HOME_X64%\bin` directory. The environment variable, `%XHQ_SERVER_HOME_X64%`, defaults to `"%ProgramFiles%\XHQ\XHQ Server"` and is required for support of 64-bit XHQ.

Do not remove or modify these environment variables after XHQ has configured them.
Doing so, results in a system with an unsupported configuration.

So, for example, when XHQ is installed on a 64-bit operating system, the following environment variables are defined:

XHQ_SERVER_HOME_X64 Which defaults to `%ProgramFiles%\XHQ\XHQ Server`	For 64-bit server and connector processes
XHQ_SERVER_HOME Which defaults to `%ProgramFiles(x86)%\XHQ\XHQ Server`	For 32-bit applications, server, and connector processes

Compare this to when XHQ is installed on a 32-bit operating system. Only XHQ_SERVER_HOME is defined and defaults to `%ProgramFiles%\XHQ\XHQ Server`.

The XHQ Service Configuration

The XHQ Service uses a .PROPERTIES file to configure various items. This takes precedent over certain environment variables. To locate this configuration file, go to **`%XHQ_SERVER_HOME%\bin\xhq_service.properties`**.

This file is not in the `repos` directory since it is related to the software configuration on the machine.

The following environment variables and/or properties are used to configure the XHQ Service:

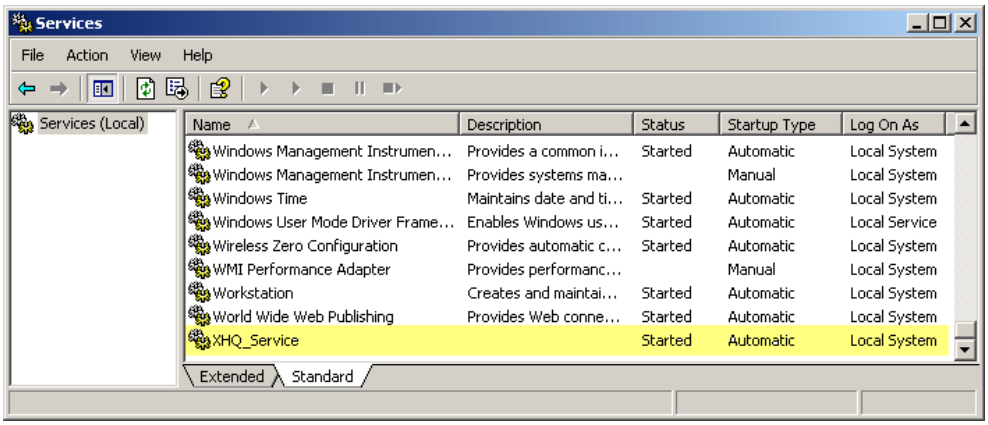
Environment Variable/Property	Description
`XHQ_SERVER_HOME`	Determines the location of the (32-bit) XHQ Server. This must be defined as a **system environment variable**; it is not recognized as a property.
`XHQ_SERVER_HOME_X64`	Determines the location of the 64-bit XHQ Server.

Environment Variable/Property	Description
	This must be defined as a **system environment variable**; it is not recognized as a property. Only read if the operating system is 64-bit.
XHQ_AUTO_RESTART_ON_FAIL	Determines the action taken when an XHQ Server process stops unexpectedly. When set to true, the XHQ System is restarted. The default value is `true`, in which case the XHQ System is restarted.
XHQ_HEALTH_CHECK_INTERVAL	The interval in minutes that the XHQ Service runs a health check test of the XHQ System. The value can be any positive integer in the range from 1 to 10. To disable the health check behavior, set the interval to 0 (zero). The default interval is 5 minutes. **Note:** If a health check failure is detected, it is reported to the Windows Application Event Log and the service takes the action defined by the XHQ_HEALTH_CHECK_FAIL_ACTION property.
XHQ_HEALTH_CHECK_FAIL_ACTION	Indicates the action taken by the XHQ Service if the health check operation returns a failure. The following settings are defined for this property: • **none** No action is taken. The failure is reported to event log. This is the default. • **shutdown** Shuts down the XHQ System. This is useful for redundant system setups. • **restart** Restarts the XHQ System.
XHQ_SERVER_VERBOSE	For debugging purposes. This causes more information to be sent to the Windows Application Event log during the course of operation. The default value is `false`.
XHQ_X64_PROC_NAMES	Defines the list of approved processes that the service launches as 64-bit applications. The list is delimited by any of the following: space, comma, semi-colon, or newline (\n). This property is only read when the XHQ Service is running on a 64-bit Windows Operating System. There is no default value for this property. **Note:** The `XHQ_X64_PROC_NAMES` environment variable can be manipulated. This allows you to further restrict which processes can be launched as 64-bit versions.

Also refer to the topic, *The Windows XHQ Service*, for more information.

The Windows XHQ Service

The XHQ System installs a service called **`XHQ_Service`** in the Windows services.



Windows Services - XHQ_Service

Important Things to Note

- The user that is configured to run XHQ Service needs LOCAL administrative rights to the system. If the user is changed to a DOMAIN user, then the domain user needs to be in the LOCAL Admin group.
- The account used for the XHQ Keep Alive must be the same used for the XHQ Service.
- The XHQ Service account must also have access rights to XHQ.

Also refer to the topics: *The XHQ Keep Alive Service*, or *XHQ Service and WAKA Configuration*.

For additional information regarding the **group policy setting** as it relates to the XHQ service user, refer to the topic, *Log On Options for the Connection Process*, located in the XHQ Connection Guide.

The following section describes in detail what happens when the `XHQ_Service` is stopped and started.

Stopping and starting the XHQ_Service is only recommended for backing up the XHQ System prior to an upgrade.

When the `XHQ_Service` is **stopped**, the following steps occur.

1. The `XHQ_Service` is stopped:
   - From either the Control Panel or a Command Prompt (see the following procedures).
   - Or when the host machine is shutdown and the `XHQ_Service` receives a `SHUTDOWN` event.
2. The service status is changed to `STOP PENDING`.
3. Depending on how `XHQ_Service` was stopped in step 1,
   - If the service was stopped from the Control Panel or the Command Prompt, then the **`xhqboot shutdown complete`** command is run.
   - Or, if the host machine was shutdown, then the **`xhqboot shutdown`** command is run.
4. The service then waits for the shutdown to complete.
5. The service sends a `SHUTDOWN` message to the `rep_init` process through Windows interprocess communications (IPC).

6. The service waits for the `rep_init` process to exit.
7. Next, the service reports the status as `STOPPED`.
8. The service exits and every process associated with the XHQ Service is stopped.

When the `XHQ_Service` is **started**, these steps occur.

1. The service status is changed to `START PENDING`.
2. The service checks to see if the XHQ repository communication layer (`XHQ rep`) support is initialized.
3. If it is not, then the service runs the `rep_init` process.
4. The server waits a small amount of time, and checks again to see if `XHQ rep` support is initialized.
5. Next, any of the following occurs:
   - If `XHQ rep` support is initialized, then the server proceeds to step 6.
   - If `XHQ rep` support is still not initialized, the server continues to wait and check (see step 4) ten more times.
   - If `XHQ rep` support is still not available after ten checks, the `XHQ_Service` exits with a failure.
6. Once the `XHQ rep` support is initialized, then the following command is run on a separate thread:

```
xhqboot startup
```

7. The `XHQ_Service` reports the status as `STARTED`.
8. The `XHQ_Service` waits for the `xhqboot` command to finish and reports if the command returns an error code.

To start or stop the XHQ_service from the Control Panel

1. From the **Control Panel**, click **Administrative Tools**, and then click **Services**.
   The "Services" dialog box appears.
2. Locate **`XHQ_Service`**.
3. **Right-click** on `XHQ_Service` and select **Start** or **Stop**.

To start or stop the XHQ_service from a Command Prompt

From a Command Prompt, enter:

- `NET START XHQ_Service`

or

- `NET STOP XHQ_Service`

The XHQ Keep Alive Service

The XHQ Keep Alive Service (**`XHQ_WebAppKeepAlive`**) pings web sites on a continual basis to improve load times. It is installed by default in the *Start* state.

The domain user account for the XHQ Keep Alive Service **must be the same** used for the XHQ Service.

This service uses a configuration file, to which you can enter parameters and sites (including paths).

The current configuration file (as well as a sample configuration file) is, by default, located in the **`XHQ\data\repos\conf\services\xhq_webappkeepalive`** folder.

Also refer to the topic, *XHQ Service and WAKA Configuration*, for more information.

```
  <?xml version="1.0" encoding="utf-8" ?>
- <Configuration>
  - <Settings>
      <Setting name="Debug" value="true" />
      <Setting name="LogPath" value="%XHQ_SERVER_LOGS%" />
    </Settings>
  - <Site name="Composer">
    - <Settings>
        <Setting name="FastInterval" value="2" />
        <Setting name="Interval" value="20" />
        <Setting name="Debug" value="true" />
      </Settings>
    - <Paths>
        <Path name="Composer" value="http://localhost/composer/PortalPage/VisualComposer/Composer.aspx" />
        <Path name="Data Source Manager"
          value="http://localhost/composer/PortalPage/VisualComposer/Configuration.aspx" />
      </Paths>
    </Site>
  + <Site name="PM">
  + <Site name="TM">
  + <Site name="eLogs">
  + <Site name="ViewStats">
  </Configuration>
```

Global Settings

Site Settings

Path Settings

XHQ Keep Alive Service Configuration File

Section	Parameters	Description
Global Settings	• **Debug**	This value is either true or false. If true, logs are written to the log file.
	• **LogPath**	This is the path to the log file(s), or the environment variable pointing to the location where the log file(s) is (are) written. *Example:* `name="LogPath" value="%XHQ_SERVER_LOGS%"`
Site Settings	These settings are specifically for the given site. All sites are processed *asynchronously*. For example, the sites Composer, PM, TM, eLogs, and ViewStats each run in its own thread. So, when the XHQKeepAlive service is started, the first path from each site is processed at the same time. Paths within a site are processed *synchronously*, or, one after another.	
	• **name**	This *attribute* is a string that describes the overall site. *Example:*

Section	Parameters	Description
		`<Site name="Composer">`
	• **FastInterval**	This is a number (in **minutes**) between pings. It is the interval time that is used by the site whenever a ping to a site fails. The site continues to use this fast interval time until the ping is successful (when the normal Interval time is then used).
	• **Interval**	This is a number (in **minutes**) between pings. *Example:* `name="Interval" value="15"`
	• **Debug**	This value is either true or false. If set, it overrides the Debug setting in the Global Settings section. *Example:* `name="Debug" value="true"`
Paths Settings	Each given site can contain zero or more Path elements. Each Path has a name and a value attribute.	
	• **name**	This parameter describes the path value. *Example:* `name="Composer Home Page"`
	• **value**	The URL to the web site to ping. **Tip:** For best load time performance, the path should be to a view on the site where resources (such as back-end connections, libraries, and so forth) are cached into the application pool so that they can be used by the next user. *Example:* `value="http://localhost/composer"`

XHQ Service and WAKA Configuration

The XHQ Service and XHQ Web Application Keep Alive (WAKA) Service must be configured with the same user account and can be configured in one of the following two scenarios.

To modify the WAKA XML configuration file for the remote/distributed XhqWse Web Service, refer to the topic, *Web Services Setup*, located in the XHQ SDK Reference Guide.

SCENARIO 1: RECOMMENDED CONFIGURATION FOR GENERAL XHQ USE AND/OR XHQ SECURITY ENABLED

Configure and run the XHQ Service and the XHQ WAKA Service processes as a **valid domain user** (also known as the "XHQ Service account"), which also has local machine administration rights (is also a local administrator).

This ensures that all child processes of the XHQ Service and XHQ WAKA Service, including for example the XHQTagSync.bat, is launched with the account associated with the XHQ Service. This is a "best practice" for configuring XHQ systems.

To change the account setting, open the Services administrative tool. Locate and right-click XHQ_Service. From the shortcut menu, select "Properties." In the "Properties" dialog box, click the "Log On" tab. Select a valid user account. Do the same for the XHQ WAKA Service.

Things to Note

- In this scenario, the **NT AUTHORITY\SYSTEM (Local System)** account has no implicit XHQ rights (role membership) unless explicitly granted by the XHQ administrator. To enable a Local System process to access XHQ, NT AUTHORITY\SYSTEM must be in a group that is explicitly mapped to the XHQ role(s) where access is to be granted.

SCENARIO 2: NON-RECOMMENDED CONFIGURATION FOR GENERAL XHQ USE (OUT OF THE BOX DEFAULT) OR WHEN XHQ SECURITY IS DISABLED

Configure and run the XHQ Service and the XHQ WAKA Service processes as the **NT AUTHORITY\SYSTEM (Local System)** account (also known as the "XHQ Service account") which has local machine administration rights (is also a local administrator). This is the "out of the box" configuration and should only be used until such time that a valid domain user has been configured and can be leveraged instead. This configuration is supported because it is understood that a domain user might not be viable at the time of installation.

In this configuration scenario, all child processes of the XHQ Service and XHQ WAKA Service, including for example the Tag Sync process, are launched with the account associated with the XHQ Service, that is the **NT AUTHORITY\SYSTEM (Local System)**.

Note: If XHQ security is disabled, no layered applications (such as VTC, Alert Notification System, View Statistics, Performance Management, and so forth) can be used These require XHQ security to be enabled.

Things to Note

- The NT AUTHORITY\SYSTEM (Local System) account is a powerful account that has full access to the computer. The Local System account does not have any rights to access the network. When network access is necessary, Local System uses the account Domain\computername$.

- NT AUTHORITY\SYSTEM can be added to machine local groups but the DOMAIN\COMPUTER$ account cannot be added.
- NT AUTHORITY\SYSTEM is not a real user but a "well-known SID" (S-1-5-18) and acts more like a "group"
- Any process connecting to XHQ from a remote machine is subject to the XHQ role mapping rules.
- "Local System" processes from remote machines cannot access or use XHQ.

XHQ Performance and Anti-virus Software

For anti-virus software users (such as Norton AntiVirus), exclude the following XHQ directories from being scanned, so as to avoid potential performance degradation.

From the XHQ Client

- The view cache directory

From the XHQ Server

- XHQ Server
- All XHQ-related web folder subdirectories.
  For example, in `C:\Inetpub\`
- The XHQ program file location.
  For example, `C:\Program Files\XHQ`
- The XHQ data file location.
  For example, `C:\XHQ`

Because virus scan usage on an XHQ Server causes a large performance hit (CPU, disk I/O), the following approach is recommended (if viable for your case):

- At the very least, remove the on-access scan and exclude as many XHQ data and cache directories as possible from the server scan.
- Schedule a batch scan of theXHQ Server during server off-peak hours (for example, at night).
- Ensure that the server is locked down against any type of end-user write access that could deposit files on the server. Standard administrative procedures must be in place so that files cannot be uploaded to the XHQ Server (for example, so that they could be offered to users over the web).
- Manually scan all files before they are uploaded to the server.

Optimal Server Settings

This topic discusses the optimal server settings for application performance and network use.

The Windows operating system includes a setting that determines how long individual threads are allowed to run on the CPU before a context-switch occurs and the next thread is serviced. This amount of time is referred to as a *quantum*. Windows also has a setting which lets you choose how processor *quanta* are shared between foreground and background processes. Typically for a server, it is not desirable to allow the foreground program to have more CPU time allocated to it than background processes. That is, all applications and their processes running on the server should be given equal contention for the CPU.

Refer to Microsoft support information with regards to the various server settings for application performance and network utilization.
`http://technet2.microsoft.com` *or* `http://support.microsoft.com`

The following recommended procedure shows you how to adjust the processor scheduling so that all programs receive equal amounts of processor time.

This is the preferred setting for most servers.

To adjust scheduling for improved performance

1. From the **Control Panel**, click **System**.
   The "System Properties" dialog box appears.
2. Click the **Advanced** tab, and under **Performance** click **Settings**.
   The "Performance Options" dialog box appears.
3. Click the **Advanced** tab.
4. In the **Processor scheduling** group, click **Background services**.
5. Click **OK** through the subsequent dialogs to apply and save your changes.
6. **Close** Control Panel.

Virtual Server Environments

XHQ supports installation and use of virtual servers running in a virtual server environment. Currently, the **VMware ESX Server** and the **Microsoft Hyper-V** are supported.

For a complete list of supported versions,go to the topic, *Supported Virtual Servers*, located in the XHQ Installation Guide.

Virtual servers are approved for the purpose of consolidating XHQ **development servers only**. Make sure that the XHQ licensing also covers use of XHQ on a virtual server. In most cases, additional XHQ licensing is required since the underlying machine or the complete virtual machine cluster must be licensed due to the underlying embedded database.

XHQ production servers are not supported without the prior explicit approval of the planned configuration by the XHQ Customer Support Team (due to the impact of underlying hardware or software issues on multiple production systems at the same time, and the risk of performance-related issues).

Installation pre-requisites do not change due to running XHQ on a virtual machine. As far as XHQ is concerned, it is a physical machine, so the current sizing recommendations apply. XHQ currently does not recommend use of iSCSI-based storage systems, but instead recommends high performance Fiber Channel-based storage.

For a complete list of system requirements, see the topic, *Requirements and Pre-requisites*, located in the XHQ Installation Guide.

Things to Consider

When setting up such a solution for running virtual XHQ server systems, you must consider the following:

- The ability to assign adequate real memory to the virtual machine (as defined in the system pre-requisites).
- The ability to assign adequate CPU bandwidth (through many virtual CPUs) to the machine such that the machine *believes* it has the required number of physical CPUs and cores, for example, when running. In addition, the server must be able to use all virtual CPUs in a multiprocessor configuration when running.
- The lack of performance degradation due to virtualization when the I/O (for example, the disk activity) increases on the virtual machine.

  This is an issue when using VMware Professional on a desktop (as compared to using VMware ESX Server or Microsoft Hyper-V) which is why VMware Professional is not a supported customer configuration.

- Virtual machines must support the XHQ Server operating system.

  Refer to the topic, *Requirements and Pre-requisites*, located in the XHQ Installation Guide for the currently supported XHQ Server operating system(s).

- A defined process for the creation and management of snapshots and backup/recovery.

Consider the following example.

Four virtual XHQ development systems are to be run on one physical server. Ideally, each of the four virtual machines has 16 GB of memory. In this case, a machine with 64 GB of memory and 4 CPU eight-core is a typical physical server configuration used to host the virtual machines. Each machine would get eight cores assigned.

In general, having enough memory is more critical than having a maximal CPU configuration.

Additional Security Considerations

Clickjacking

In this type of attack, multiple transparent / opaque layers are placed on top of each other in order to trick the user to unknowingly perform an action within the application (for example, visiting a malicious web site).

OWASP provides a good example of a clickjacking attack:

"Imagine an attacker who builds a web site that has a button on it that says 'click here for a free iPod'. However, on top of that web page, the attacker has loaded an iframe with your mail account, and lined up exactly the 'delete all messages' button directly on top of the 'free iPod' button. The victim tries to click on the 'free iPod' button but instead actually clicked on the invisible 'delete all messages' button. In essence, the attacker has 'hijacked' the user's click, hence the name Clickjacking."

To prevent clickjacking

It is recommended you visit the site, https://www.owasp.org/index.php/Clickjacking_Defense_Cheat_Sheet, for a comprehensive list of methods to combat clickjacking.

To protect against clickjacking for all sites in IIS 7, add the **CSP directive** (as noted in the recommended link) to the IIS configuration file, `applicationHost.config`. In addition, Internet Explorer 11 users must also add the **X-Frame-Options directive**.

1. Navigate to `%windir%\system32\inetsrv\config\`.
2. Locate and right-click on the **applicationHost.config file**.
3. **Open** the file in a text editor.
4. Locate the **configuration > system.webServer > httpProtocol > customHeaders tag** and insert the following:

   ```
   <add name="Content-security-Policy" value="frame-ancestors siteName1" />
   <add name="X-Frame-Options" value="ALLOW-FROM siteName1" />
   ```

   Where: **siteName1** is your local intranet site that hosts an embedded XHQ web page in an iFrame element. It can either be a server name or a web site. You can enter as many site names as needed, separated by a space.

   Example for CSP: `value="frame-ancestors https://siteName1.com https://siteName2.com" />`

   Example for X-Frame-Options: `value="ALLOW-FROM https://siteName1.com https://siteName2.com" />`

5. **Save** and close the file.
6. **Restart the IIS Server** to validate these changes.

Cross-site Scripting

Due to potential security risks, cross-site scripting is prohibited.

Search the web for more information on cross-site scripting.

About XHQ and Active Directory

During installation, the installation routine determines which operating system the Enterprise Server is running on and installs the appropriate DLL (Dynamic Link Library) in the **`%XHQ_SERVER_HOME%\bin`** directory, which by default is `C:\Program Files (x86)\XHQ\XHQ Server\bin`.

The Active Directory-based DLL with the filename **`librep.dll`** is installed in the above directory.

About the License Manager

XHQ is shipped with license management software. A license key is required to activate the licensed software based on the product options purchased.

For information on how to acquire the license key, go to the topic, *The License Manager*, located in the XHQ Installation Guide.

XHQ Logs

About the Log Directory

WARNING
Do not delete the DB directory, which is by default located in `%XHQ_LOGS%`. Doing so makes the XHQ system unusable. In addition, do not delete the XHQ, XHQCACHE, or the XHQHIST subfolders located in the DB directory.

As stated in the WARNING message above, do not delete the subfolders in this directory, specifically the XHQ, XHQCACHE, or the XHQHIST subfolders. You may delete the XHQ log files (which are the `.OUT` files generated by the XHQ servers and connectors), **but not the folders**.

If you delete the `%XHQ_LOGS%\DB` folder, the **XHQ Server fails to start**. In this case, to fix this error, you must manually create the following directories:

- `%XHQ_LOGS%\DB\XHQ`
- `%XHQ_LOGS%\DB\XHQHIST`
- `%XHQ_LOGS%\DB\XHQCACHE`

About the Solution Viewer Log

Only one Solution Viewer log file is created, even if you are viewing multiple solutions on the same client machine. The view cache creates separate directories for each solution being viewed in the single log file.

The default log file is **`browser.log`**. To generate a custom browser log file, use the **logname** parameter to append the URL that launches XHQ.

Example: `http://localhost/indx/sv/?logname=browser1.log`

In this example, the Solution Viewer uses the `browser1.log` file instead of the default log file.

Load View Performance Logging

To log the load view performance, add the parameter, `perflog=true`, to the XHQ Solution Viewer URL.

Example:

`http://localhost/indx/sv/?perflog=true#::root/~viewA`

The output log file, **`viewserverclientlog1.out`**, is located in the `%XHQ_LOGS%` directory, which by default is `<systemdrive>:\XHQ\log`.

Example Output Log:

```
"view name":"::SieDx.~trend",
"view load time":7772,
"rendering time":18,
"requests time":219,
"evaluate time":7454,
"collections":"0 (0)",
"aggregates":0,
"subscriptions":"1 (1)",
"trend requests":2,
"received elements":114,
"processed elements":117,
"sent data size":13155,
"received data size":711953,
"sent requests":89,
"received requests":88
```

Field	Description
view name	The name of the loaded view. *Example*: `"::SieDx.~trend"`
view load time	The overall time the view takes to load. *Unit*: milliseconds (ms)
rendering time	The amount of time it takes to render elements to the page. *Unit*: milliseconds (ms)
requests time	The amount of time it takes for the server to answer a request. *Unit*: milliseconds (ms)
evaluate time	The amount of time spent on evaluating all the expressions. *Unit*: milliseconds (ms)
collections	The number of distinct collection members used in the view, followed by the number of collection requests in parentheses. In other words: (all collections (sent collections)) *Format*: Number (Number) *Example*: `"0 (0)"`
aggregates	The number of aggregate requests.
subscriptions	The total number of subscriptions followed by the number of subscriptions on Tag members in parentheses.

Field	Description
	In other words: (all subscriptions (the number of subscriptions of tags)) *Format*: Number (Number) *Example*: `"1 (1)"`
trend requests	The number of Trend data requests for the view.
received elements	The number of elements received from the server.
processed elements	The number of processed elements for the view.
sent data size	The size of the data sent to the server. *Unit*: bytes
received data size	The size of the data received from the server, including metadata. *Unit*: bytes
sent requests	The number of requests sent to the server.
received requests	The number of request responses received from the server.

IIS Logging

To **enable logging** of all requests (successful and failed), open the **Windows Command Prompt** (`CMD`) as **Administrator** and run the following commands:

1. `%systemdrive%`
2. `cd %systemroot%\System32\inetsrv`
3. `appcmd.exe set config -section:system.webServer/httpLogging -dontLog:"false" /commit:apphost`
4. `appcmd.exe set config -section:system.webServer/httpLogging -selectiveLogging:"LogAll" /commit:apphost`

To **disable logging** of successful requests (default log level), open the **Windows Command Prompt** (`CMD`) as **Administrator** and run the following commands:

1. `%systemdrive%`
2. `cd %systemroot%\System32\inetsrv`
3. `appcmd.exe set config -section:system.webServer/httpLogging -dontLog:"false" /commit:apphost`
4. `appcmd.exe set config -section:system.webServer/httpLogging -selectiveLogging:"LogError" /commit:apphost`

Customizing the Development Client Logs and Repos Directory Locations

By default, the folders used by the XHQ Development Client to write its logs and application data are found under the path defined in the standard Windows environment variable for user application data, **`%APPDATA%`**. More specifically:

- Logs are located in the `%APPDATA%\XHQ\XHQ devClient\logs` directory;
- Data is located in the `%APPDATA%\XHQ\XHQ devClient\repos` directory.

For additional information on the XHQ environment variables, go to the topic *Distribution of XHQ Server and devClient Files* located in the **XHQ Installation Guide**.

Usually, these defaults are the best choice for writing logs and data. It is, however, possible to customize these locations by configuring one or both of the following batch files:

- `%XHQ_DEV_HOME%\bin\launchsettings.bat`
  Machine-wide folder customizations may be defined in this batch file.
- `%APPDATA%\XHQ\XHQ devClient\repos\conf\launchsettings.bat`
  Individual custom folders may be defined in this batch file.

These files do not exist by default, and need to be manually created and saved in the given directory.

Setting custom folders is as simple as defining either the `%XHQ_DEV_LOGS%` or the `%XHQ_DEV_REPOS%` environment variable (or both) in one of the batch files. For example:

```
set XHQ_DEV_LOGS=C:\XHQ\devLogs
set XHQ_DEV_REPOS=C:\XHQ\devRepos
```

If both batch files are configured, and a setting is made in one conflicts with a setting made in the other, the value set in `%APPDATA%\XHQ\XHQ devClient\repos\conf\launchsettings.bat` overrides the value set in `%XHQ_DEV_HOME%\bin\launchsettings.bat`.

More about the launchsettings.bat files

Custom folders defined in either of the `launchsettings.bat` files affect all XHQ Development Client tools (XHQ Workbench, XHQ Solution Builder, XHQCI, and so forth) whether launched from the Start menu, the XHQ Shell, or the XHQ Tray Utility. The Start menu shortcuts for the log and data folders also open the custom folders as defined in the `launchsettings.bat` files.

A change made in the `launchsettings.bat` file takes effect once the client tools are started after the change. At which point, all client tools, shortcuts, and so forth, start writing logs, and reading and writing data to the new custom location. Please note that client preferences and configurations are kept in files in the data folder for the XHQ Development Client. For example, the settings made with the Server Access utility are stored in a file in the XHQ Development Client data folder. When you customize the XHQ Development Client data folder, you will lose these preferences and configurations unless you move the files that contain them. As such, it is not recommended that the data folder location be changed frequently or without ample preparation.

JDBC Fetch Size

Fetch size is a performance hint that tells the JDBC driver how many rows should be fetched from the database when more rows are needed by the client. Increasing this value can improve overall query performance, especially with large result sets. For example, the default for Oracle is ten.

XHQ leverages the default driver settings for fetch size. While it is possible to change/tune this parameter for performance improvements, such changes need to be carefully made and well tested. Setting a JDBC fetch size that is too large or too small can degrade performance or cause memory problems. In one documented case, the fetch size was changed to 100000. This resulted in the XHQ cache server running out of heap memory. However, in the same circumstance, changing the fetch size to 1024 worked fine. Extreme care is needed when setting the size above 10000. To avoid out-of-memory issues, changing the size to 1024 or 10000 are typical use cases and would be validated in performance analysis scenarios. The optimal value varies, depending on the solution requirements.

About Hyper-Threading Technology and the XHQ System

Enabling Hyper-Threading Technology on the **client** can significantly improve system performance. The XHQ System only supports the use of the Hyper-Threading Technology on **client** operating system versions if the computer configuration is supported by Microsoft with hyper-threading enabled.

Hyper-threading on the XHQ Server is not recommended due to possible performance degradation.

For additional information, see the Microsoft white paper "Windows Support for Hyper-Threading Technology" available at:

http://www.microsoft.com/whdc/system/CEC/HT-Windows.mspx

EXAMPLE: HOW TO ENABLE HYPER-THREADING

The following is a typical example based on one version of the DELL Optiplex 270 system.

1. Enter the PC BIOS during startup.

   Refer to your PC manual for instructions on how this is done.

2. Select **Performance** and then **Hyper-Threading**.
3. If the "On" background is gray, then Hyper-Threading is enabled.
   If Hyper-Threading is not enabled, then continue with the following.
4. Press **Enter** to modify this option.
5. Select **On**.

   The background is green when selected.

6. Press **Enter**.
7. Press **Esc**.
8. Select **Save/Exit**.

Using the debug.js File

The `debug.js` file is used to capture JavaScript errors that occur, and display the debug messages in a pop-up window. The `debug.js` file is located in the **`%XHQ_WEB_HOME%\indx`** directory, which by default is `C:\Program Files (x86)\XHQ\XHQ Web Root\indx`.

To enable logging and set the log level

1. Go to the **`%XHQ_WEB_HOME%\indx`** directory.
2. Open the `debug.js` file in a text editor.
3. Locate the `g_LOG_TO_CONSOLE` variable and set it to **true**.

```
debug.js

66
67  //*****************************************************************************
68  // The following defines the different log levels.
69  //*****************************************************************************
70  var g_SHOW_NO_MSGS              = 0;
71  var g_SHOW_ERROR_MSGS           = 1;
72  var g_SHOW_WARNING_MSGS         = 2;
73  var g_SHOW_INFORMATION_MSGS     = 3;
74  var g_SHOW_FILE_OPEN_MSGS       = 4;
75  var g_SHOW_FUNCTION_ENTRY_MSGS  = 5;
76  var g_SHOW_LINE_DEBUG_MSGS      = 6;
77  var g_SHOW_ALL_MSGS             = : true;
78
79  // *** SET the log level to one of the values shown above.
80  var g_LOG_LEVEL = g_SHOW_WARNING_MSGS;
81
82  // *** SET this variable to true to enable logging.
83  var g_LOGGING_ENABLED = true;
84
85  // *** SET the next variable to true to enable the logging console.
86  var g_LOG_TO_CONSOLE = true;
87
88  // *** SET the next variable to true to enable logging to the status bar.
89  var g_LOG_TO_STATUS_BAR = false;
90
```

4. Locate the `g_LOG_LEVEL` variable and set the value to a given log level.

```
debug.js

66
67  //*****************************************************************************
68  // The following defines the different log levels.
69  //*****************************************************************************
70  var g_SHOW_NO_MSGS              = 0;
71  var g_SHOW_ERROR_MSGS           = 1;
72  var g_SHOW_WARNING_MSGS         = 2;
73  var g_SHOW_INFORMATION_MSGS     = 3;
74  var g_SHOW_FILE_OPEN_MSGS       = 4;
75  var g_SHOW_FUNCTION_ENTRY_MSGS  = 5;
76  var g_SHOW_LINE_DEBUG_MSGS      = 6;
77  var g_SHOW_ALL_MSGS             = : true;
78
79  // *** SET the log level to one of the values shown above.
80  var g_LOG_LEVEL = g_SHOW_WARNING_MSGS;
81
```

Log Levels

Example: `var g_LOG_LEVEL = g_SHOW_WARNING_MSGS;`

5. **Save** and close and file.
6. **Embed** (place the contents of) this file inside all HTML or ASP files that contain any of the `logMessage` functions.

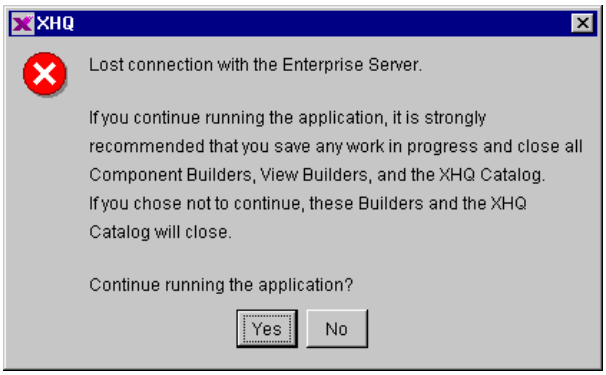
As a result, opening any of these HTML or ASP files launches a debug pop-up window, which displays the debug messages.

For more information on logMessage functions, see the topic, *Log Functions*, located in the XHQ Reference Guide.

XHQ Workbench File Recovery

XHQ features a file recovery mechanism for the XHQ Workbench. If you lose a network link while working in either of these environments, a message appears indicating that the network connection has been broken, and your work will be saved to a recovery file on your local machine.

Copying to Local File System



Lost Connection Message

In the event you experience the loss of a network connection while operating multiple editing sessions, all sessions are saved to recovery files. The system presents you with the option to recover the files. If you choose "No", the recovery files are deleted. If you respond "Yes", a new session of the XHQ Workbench opens for each of the available recovery files. The next time you try to save any recovered file, the system will attempt to write to the network, saving the file to the server rather than locally. If a network connection cannot be established, the system overwrites the recovery file on the local machine.

About Security and File Recovery

Prior to retrieving recovery files, XHQ automatically performs an access/security check. If security has been enabled, the following may occur:

- The XHQ Workbench > Component Details tab will launch only if the user is a Model Admin or has Component Editing permissions on the component.
- The XHQ Workbench > View Drawing panel will launch only if the user is a Model Admin or has View Editing permissions on the component.

For more information on XHQ security, go to the topic, *Security Access and Privileges*.

For XHQ Applications Using AJAX

For XHQ applications (such as the View Statistics Utility) that use AJAX, the following option needs to be configured as described, in order for the applications to function correctly.

To configure the IIS Manager for XHQ applications using AJAX

1. From **IIS Manager**, in the **Connections** pane, expand the **Sites** node and **select the web site that contains the XHQ application** (which is typically, Default Web Site).
2. In **Features View**, on the **Default Web Site Home** page, double-click **Handler Mappings**.
3. Under the **Enabled** group, **select** (to highlight) the handler associated with the **.axd** extension.
4. In the **Action** pane, click **Edit**.
   The "Edit Script Map" dialog appears.
5. Click **Request Restrictions**.
6. From the **Mapping** tab, **uncheck** the **Invoke handler only if request is mapped to** option.
   By default, this option is unchecked.
7. Click **OK**.
8. Repeat steps 3-7 for each handler associated with the **.axd** extension.
9. From the **IIS Manager**, in the **Connections** pane, **right-click** the server and **restart IIS**.

About the XHQ Internal Cache

The XHQ internal Cache allows configuration of index and unique key information so that the XHQ cache optimizes data queries in real-time. It consists of a table of all the members assigned to the collection being queried. A string's maximum length, unique key, and index information can be defined.

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.

The use of multiple indexes gives the solution developer greater control over how collection members are sorted, and thus improves the performance of collections.

Pass-through queries are *passed* to the back-end data source and results are delivered to the user; in this case, content is not cached within the XHQ Solution Server.

Therefore, this "temporary" cache leverages the embedded database (except for pass-through) which can:

- Reduce the load on the back-end systems by serving up the information to multiple readers;
- Control the back-end refresh activity through definable poll periods per collection;
- Increase performance drastically when configured optimally;
- And combine information from different data sources.

Improved Performance with Storage Updates

As a result of optimizations, the overall storage size is smaller. Storage updates can "squeeze" a normal XHQ **`DataConfigs.dat`** file down to less than 25% of its original size. As a result of this compression, there is a noticeable improvement in start-up performance.

Once a repository is started with this current version of the software or if **`smck -C`** has been used to modify the DAT files, the repository **cannot** be started with an earlier version of XHQ.

Using the `smck` Utility

The **smck** utility converts and compresses the DAT files in a repository after upgrading and before starting the XHQ Platform. This ensures the greatest amount of compression on start-up (otherwise, it only happens on configuration changes).

To upgrade with compressed DAT files using the smck utility

1. **Shutdown** the XHQ solution.

   Example: For this example, the XHQ solution is `ABC`.

2. **Backup** the contents of the repository directory to a safe place.

   Example: `%XHQ_SERVER_REPOS%\Solution\ABC`

3. Open the **Windows Command Prompt** (`CMD`) as **Administrator**.
4. Run the change directory (`cd`) command and go to the storage directory location.

   Example: `C:> cd /d "%XHQ_SERVER_REPOS%\Solution\ABC"`

5. From the prompt, enter the following commands to immediately convert the larger storage files.

```
smck -Cu ClassDefs Classes Collections Components Tags DataConfigs
```

Option names are **case-sensitive**.
Executing these commands will take several minutes per DAT file.

Example:

```
C:> smck -Cu ClassDefs Classes Collections Components
Tags DataConfigs
```

Changing the XHQ High-Performance Database Password

This procedure requires you to **shutdown and restart** the XHQ High-Performance Database.

A **valid** password:

- Is a minimum of 8 ASCII printable characters;
- Is a maximum of 30 ASCII printable characters;
- Must contain at least one **uppercase** letter;
- Must contain at least one **lowercase** letter;
- Must contain at least one **number**;
- Must contain at least one of the following **special characters** (these are the only special characters allowed):

  { (open curly bracket)
  } (close curly bracket)
  [(open straight bracket)
  ] (close straight bracket)
  $ (dollar sign)
  ! (exclamation mark)
  - (dash)
  + (plus sign)
  # (number sign)
  ? (question mark)
  ~ (tilde)
  _ (underscore)

  Again, all other characters (such as @, ^, &, >, <, |, ") are not allowed;
- Does not contain any spaces;
- Complies with the convention for selecting secure passwords at your company, or in accordance with reasonable industry practice.

The password is **case-sensitive**.

About the Password Utility (pwutil)

Performing the task of changing the password in the XHQ High-Performance Database is the main role of the **`pwutil`** utility. The batch file is located in `%XHQ_DBMS_HOME%\dbadmin\bin` directory (which by default is `<systemdrive>:\XHQ\XHQDB\dbadmin\bin`).

Syntax:

```
pwutil /<operation>[[ ]<modifier>]
       [/i[nteractive]
       [/v[erbose]]
       <user_token>...
pwutil [/(h[elp] | ?)]
```

Where:

```
operation  = (r[eset] | t[est])
 modifier  = (ns|nosync)
user_token = <database>.<user_name>[=<password>]
```

(with wildcard) = (<database>| *).*[=<password>]

You may use the dash '-' instead of the forward slash, '/'.

Operations

Parameter	Description
r[eset]	Changes the saved passwords of one to many users.
t[est]	Tests whether one to many users can connect with their saved passwords.

Operation Modifier

Parameter	Description
(ns|nosync)	*Valid only with the r[eset] operation.* Ordinarily, r[eset] reconfigures each specified user's database to ensure that the user can connect to the database using the new password. The (ns|nosync) modifier suppresses that behavior. If a user of the XHQAM database is specified, the (ns|nosync) modifier is assumed for that user even if not specified.

User Token

Parameter	Description
<database>	One of the recognized database names: XHQ, XHQCACHE, XHQHIST, or XHQAM.
<user_name>	One of the recognized user names in the recognized databases. • For XHQ: SYS, System, postgres, xhq, xhqro, xhqstats, xhqapps, xhqapps_r, xhqapps_w • For XHQCACHE: SYS, System, postgres, xhqcache • For XHQHIST: SYS, System, postgres, xhqhist • For XHQAM: xhq
<password>	*Required with the r[eset] operation.* Invalid otherwise.

User Token with Wildcard

Parameter	Description
<database>.*=<password> with r[eset]	Changes all passwords for all users of the specified database to the single password provided to the right of the equals sign.
<database>.* with t[est]	For all users of the specified database, tests whether they can connect to the specified database with their saved passwords.
.=<password> with r[eset]	Changes all passwords for all users in all databases to the single password provided to the right of the equals sign.
. with t[est]	For all users of all databases, tests whether they can connect with their saved

Parameter	Description
	passwords.

If a wildcard user token is specified, then no other user token may be specified, not even another wildcard user token.

Also, you cannot use a wildcard for the database along with the name of a specific user. There is no user name common to all databases.

Interactive Mode

Parameter	Description

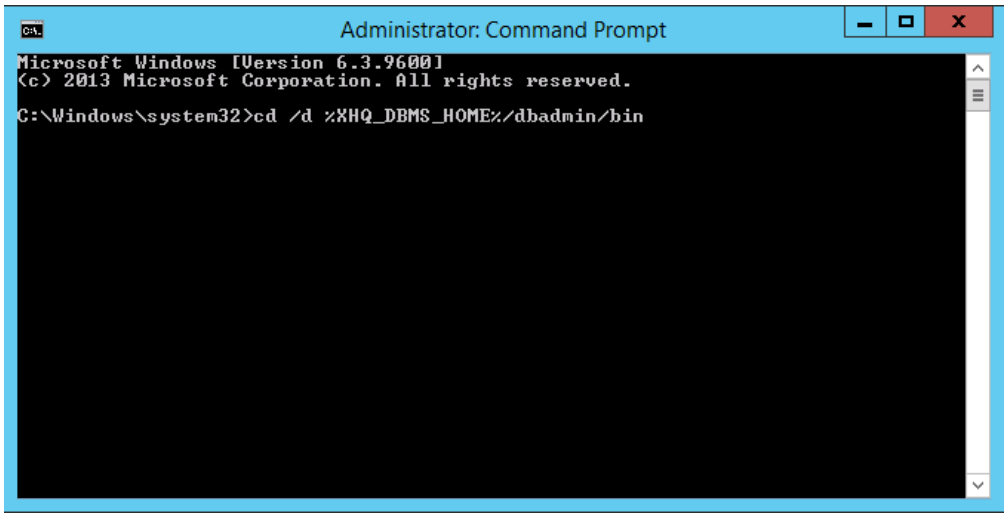
`i[nteractive]`	*A single operation and at least one user token is required.* You cannot specify a user token without first specifying an operation.
	If i[nteractive] mode is specified, then **`pwutil`** prompts for any missing required information. If no required information is missing, then i[nteractive] has no effect.
	It is also possible, using i[nteractive] mode, to avoid including sensitive information in a command line. This is done by piping input into **`pwutil`** when it is running in `i[nteractive]` mode.
	Consider the following example.
	Given: The file, **`pw.txt`**, which contains the following lines:
	`reset`
	`nosync`
	`*.*=NewPassword!99`
	The following command would reset every saved user password for every database to `NewPassword!99` without also reconfiguring the database with the new password.
	`type pw.txt | pwutil /i`

Verbose Logging

Parameter	Description
`v[erbose]`	Verbose logging provides more detailed information that describes where the operation is in its processing. If an operation completes successfully, **`pwutil`** normally responds as follows. **`r[eset]:`** An in-line list of users with changed passwords. (And an in-line list of any users with passwords that could not be changed.) **`t[est]:`** An in-line list of users that successfully connected with their saved passwords. (And an in-line list of any users that could not connect with their saved passwords.)

To change the individual user's password for the High-Performance Database

1. **Stop** the XHQ Solution.
2. Open the **XHQ Shell** *or* the **Windows Command Prompt** (CMD) as **Administrator**.
3. Run the change directory (`cd`) command and go to the `%XHQ_DBMS_HOME%\dbadmin\bin` directory (which by default is `<systemdrive>:\XHQ\XHQDB\dbadmin\bin`).



4. Run the following commands, each bullet item is a separate call to `pwutil`:
   - `pwutil /r xhq.sys=NewPassword!99`
   - `pwutil /r xhq.system=newpassword xhq.xhq=newpassword xhq.xhqro=newpassword xhq.xhqstats=newpassword xhq.xhqapps=newpassword xhq.xhqapps_r=newpassword xhq.xhqapps_w=newpassword`
   - `pwutil /rns xhqam.xhq=newpassword`
   - `pwutil /r xhqcache.sys=newpassword`
   - `pwutil /r xhqcache.system=newpassword xhqcache.xhqcache=newpassword`
   - `pwutil /r xhqhist.sys=newpassword`
   - `pwutil /r xhqhist.system=newpassword xhqhist.xhqhist=newpassword`

   Command Format

   `pwutil /r <instance-1>.<user-1>=<pw-1> <instance-2>.<user-2>=<pw-2> <instance-3>.<user-3>=<pw-3>...<instance-n>.<user-n>=<pw-n>`

   Where: r = reset

   To test whether the password stored matches what is needed to log onto the database instance as a given user, enter:

   `pwutil /t <instance>.<user>`

   Where: t - test

5. **Restart** the XHQ Solution.

To change all XHQ passwords

1. **Stop** the XHQ Solution.
2. Do one of the following:
   - Open the **XHQ Shell**.

   or

   - Open the **Windows Command Prompt** (`CMD`) as **Administrator** and then run the change directory (`cd`) command and go to the `%XHQ_DBMS_HOME%\dbadmin\bin` directory (which by default is `<systemdrive>:\XHQ\XHQDB\dbadmin\bin`).
2. Run the following command:
   - `pwutil /r *.*=NewPassword!99`

   To test whether the password stored matches what is needed to log onto the database instance as a given user, enter:

   `pwutil /t *.*`

   Where: t - test

5. **Start** the XHQ Solution.

Maintenance of the Embedded Database TEMP Tablespace

After an XHQ installation, the TEMP tablespace re-creation script is located in **`%XHQ_DBMS_HOME%\dbAdmin\bin`**, which by default, this is the `C:\XHQ\XHQDB\dbadmin\bin` directory.

This script batch file permits the XHQ Administrator to essentially delete and recreate the embedded database TEMP tablespace.

The following can cause the TEMP tablespace to grow large:

1. Oracle performs a complex table sort, due to an ORDER BY or GROUP BY operation.
2. Oracle performs a hash join.
3. The existence of a global temporary table.
   This is not supported and should not occur on XHQ Systems.
4. Bad pass-through queries.

Sometimes a large TEMP tablespace problem can be avoided by properly indexing XHQ collections, on columns that are frequently involved in sorts or ORDER BY or GROUP BY operations. In general, proper indexing can also dramatically increase performance.

Another reason to delete and recreate the TEMP tablespace, for example, is due to poorly written custom SQL that runs recursively, thus resulting in a very large TEMP tablespace that uses all the free space on the hard disk. Since the TEMP tablespace does not automatically shrink, a means is needed to reset it to a reasonable default size.

This method should only be performed after an xhqboot shutdown is performed (which shuts down XHQ, but leaves the embedded database running).

One problem with this method is that some free space is required on the hard disk in order to perform the resize operation. Unfortunately, free space may no longer be available. To that end, the script now allows specification of a max size for the TEMP tablespace as an optional second parameter. This parameter should be set to the maximum free space available on the hard disk minus a few hundred MB (as an example).

The first parameter to the script is the embedded database password.

XHQ does not support setting the TEMP tablespace max size to being much lower than the available hard disk free space and, by default, wants it unlimited for the reason that an XHQ internal operation always expects to have TEMP tablespace available and hitting a hard limit could result in data loss or subsequent system problems.

The real and preferred solution to this issue is to adequately test any custom SQL to ensure it behaves correctly and does not run out of control, or in an unexpected manner, and use up the TEMP tablespace. And do this before putting the SQL on the production system.

Currently, it is not possible to determine and separate out internal operations and customer SQL. If the TEMP tablespace ever hits this max size, it is critical to analyze why and to address the offending SQL to ensure system stability and performance is not impacted by any recurrence.

About XHQTempRecreate.bat

This **`XHQTempRecreate`** command is only used with the **LEGACY** High-Performance Database (embedded database).

This batch script drops and recreates the embedded database (XHQ High-Performance Database) TEMP tablespace when it does not shrink on its own, back to its normal size of about 1GB of physical space.

The following is an example of how the script is called, with the embedded database password being passed to it.

```
C:\WINDOWS\system32\cmd.exe
X:\dev\src\database\creation\install>XHQTempRecreate.bat xhq_
```

Example: **`X:\dev\src\database\creation\install>XHQTempRecreate.bat xhq`**

A warning then appears, telling you that this script will stop the XHQ Server and related databases (associated processes).

```
C:\WINDOWS\system32\cmd.exe - XHQTempRecreate.bat xhq
##################################################
#------------------------------------------------
#
#  WARNING!!! This will STOP your XHQ Server
#      and related database(s).
#
#  Press [ENTER] to continue or [CTRL+C] to cancel
#
#------------------------------------------------
##################################################
Press any key to continue . . . _
```

Press `CTRL+C` to cancel the script, or press `ENTER` (or any key) to continue.

If the process **successfully completes** without any errors, the following message appears:

```
##################################################
#------------------------------------------------
#
#    TEMP Tablespace recreated.
#
#------------------------------------------------
##################################################
```

Exit the command window ant **restart** the XHQ Server.

If **errors** were found, the following message appears:

```
C:\WINDOWS\system32\cmd.exe
#########################################
#
# Errors found in recreate_temp.log
# please review or send to Support
#
#########################################
```

Send the **`recreate_temp.log`** file to the XHQ Customer Support Team.

Disabling the Data Recorder

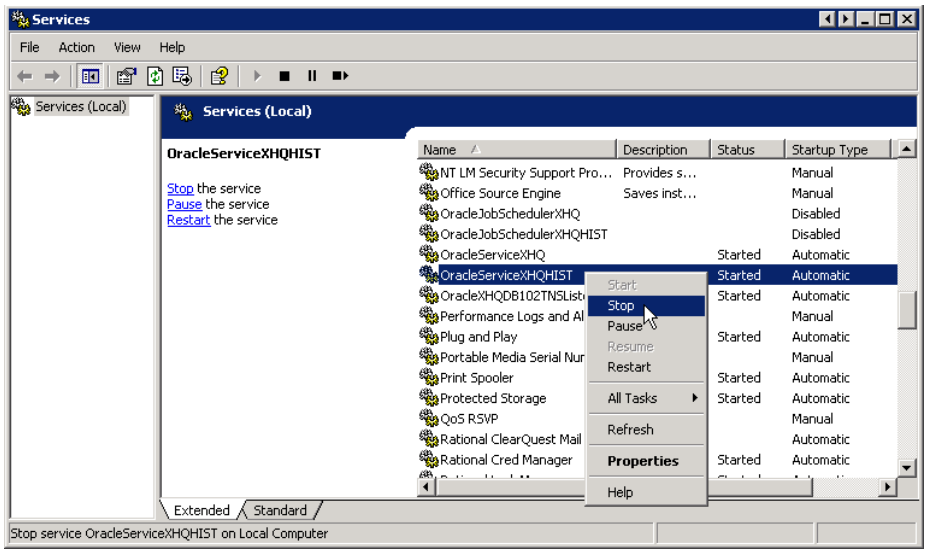
The XHQ Data Recorder is automatically installed with XHQ and is enabled by default. If the XHQ Data Recorder is not required, you can **disable** it to reduce the required physical memory footprint on the XHQ Server by about 300 MB.

Disabling the XHQ Data Recorder does not have any impact on data previously stored in the XHQ Data Recorder. This data can be accessed by re-enabling the XHQ Data Recorder.
Note, however, that the XHQ Installer requires the XHQ Data Recorder service to be enabled. If it is disabled, the installation process is halted.

There are two steps to disabling the XHQ Data Recorder. First you must disable the XHQ service associated with the XHQ Data Recorder. Then you will need to set a global property to remove the XHQ Data Recorder as an option from the History tab of the XHQ Solution Builder.

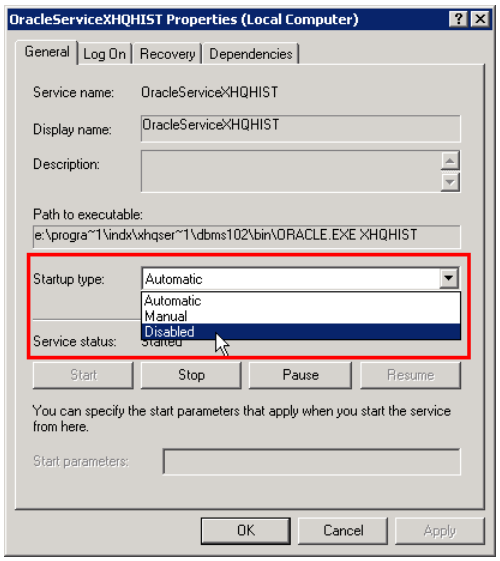
To disable the XHQ Data Recorder service

1. From the **Control Panel**, launch **Administrative Tool** and then **Services**.
2. From the list of Services, right-click on `OracleServiceXHQHist` and, from the shortcut menu, click **Stop**.



3. Right-click on `OracleServiceXHQHist` and click **Properties**.
   The "Properties" dialog box appears.
4. In the **General tab**, for **Startup type**, select **Disabled** from the drop-down list.

   You must use the Disabled type. The Manual type does not sufficiently disable the Data Recorder at startup.



5. **Restart** XHQ.

Your next step is the define the global property that hides theXHQ Data Recorder option. This avoids accidental attempted use while the XHQ Data Recorder is disabled.

To set the global property to hide the XHQ Data Recorder option

1. Locate the **`globalsettings.properties`** file and disable the READ ONLY property.
   This file is stored at the location specified by the environment variable `%XHQ_SERVER_REPOS%` on the Enterprise Server (which is typically `C:\XHQ\data\repos`).
2. Open the **`globalsettings.properties`** file in a text editor (such as Notepad).
3. Add the **`DatabaseHistorianEnabled`** property and set the value to **false**.

   Example: DatabaseHistorianEnabled = false

   Property names are **case-sensitive**.

4. **Save** and **close** the text editor.

To re-enabled the XHQ Data Recorder, you simply set the **`OracleServiceXHQHist`** service startup type back to Automatic, set the **`DatabaseHistorianEnabled`** global property to true, and restart XHQ.

Appendix

Date and Time Formats

The following is an excerpt from the JavaDocs site:

http://docs.oracle.com/javase/6/docs/api/java/text/SimpleDateFormat.html

Date and time formats are specified by date and time pattern strings. Within date and time pattern strings, unquoted letters from `'A'` to `'Z'` and from `'a'` to `'z'` are interpreted as pattern letters representing the components of a date or time string. Text can be quoted using single quotes (`'`) to avoid interpretation. `"''"` represents a single quote. All other characters are not interpreted; they're simply copied into the output string during formatting or matched against the input string during parsing.

The following pattern letters are defined.

All other characters from 'a' to 'z' and from 'A' to 'Z' are reserved.

Letter	Date/Time Component	Output	Examples
G	Era designator	Text	AD
y	Year	Year	1996; 96
M	Month in year	Month	July; Jul; 07
w	Week in year	Number	27
W	Week in month	Number	2
D	Day in year	Number	189
d	Day in month	Number	10
F	Day of week in month	Number	2
E	Day in week	Text	Tuesday; Tue
a	Am/pm marker	Text	PM
H	Hour in day (0-23)	Number	0
k	Hour in day (1-24)	Number	24
K	Hour in am/pm (0-11)	Number	0
h	Hour in am/pm (1-12)	Number	12
m	Minute in hour	Number	30
s	Second in minute	Number	55
S	Millisecond	Number	978
z	Time zone	General time zone	`Pacific Standard Time; PST; GMT-`

Letter	Date/Time Component	Output	Examples
			08:00
Z	Time zone	RFC 822 time zone	-08:00

Text

For formatting, if the number of pattern letters is 4 or more, the full form is used; otherwise a short or abbreviated form is used if available. For parsing, both forms are accepted, independent of the number of pattern letters.

Number

For formatting, the number of pattern letters is the minimum number of digits, and shorter numbers are zero-padded to this amount. For parsing, the number of pattern letters is ignored unless it's needed to separate two adjacent fields.

Year

For formatting, if the number of pattern letters is 2, the year is truncated to 2 digits; otherwise it is interpreted as a number.

For parsing, if the number of pattern letters is more than 2, the year is interpreted literally, regardless of the number of digits. So using the pattern "MM/dd/yyyy", "01/11/12" parses to Jan 11, 12 A.D.

For parsing with the abbreviated year pattern ("y" or "yy"), `the ServerClock applet` must interpret the abbreviated year relative to some century. It does this by adjusting dates to be within 80 years before and 20 years after the time the `the ServerClock applet` instance is created. For example, using a pattern of "MM/dd/yy" and a `the ServerClock applet` instance created on Jan 1, 1997, the string "01/11/12" would be interpreted as Jan 11, 2012 while the string "05/04/64" would be interpreted as May 4, 1964. During parsing, only strings consisting of exactly two digits will be parsed into the default century. Any other numeric string, such as a one digit string, a three or more digit string, or a two digit string that isn't all digits (for example, "-1"), is interpreted literally. So "01/02/3" or "01/02/003" are parsed, using the same pattern, as Jan 2, 3 AD. Likewise, "01/02/-3" is parsed as Jan 2, 4 BC.

Month

If the number of pattern letters is 3 or more, the month is interpreted as text; otherwise, it is interpreted as a number.

General Time Zone

Time zones are interpreted as text if they have names. For time zones representing a GMT offset value, the following syntax is used:

```
    GMTOffsetTimeZone:
            GMT Sign Hours : Minutes
    Sign: one of
            + -
    Hours:
            Digit
            Digit Digit
```

```
    Minutes:
            Digit Digit
    Digit: one of
            0 1 2 3 4 5 6 7 8 9
```

Hours must be between 0 and 23, and *Minutes* must be between 00 and 59. The format is locale independent and digits must be taken from the Basic Latin block of the Unicode standard.

RFC 822 Time Zone

For formatting, the RFC 822 4-digit time zone format is used:

```
    RFC822TimeZone:
            Sign TwoDigitHours Minutes
    TwoDigitHours:
            Digit Digit
```

TwoDigitHours must be between 00 and 23. Other definitions are as for general time zones.

Pattern Examples

The following examples show how date and time patterns are interpreted in the U.S. locale. The given date and time are 2001-07-04 12:08:56 local time in the U.S. Pacific Time time zone.

Date and Time Pattern	Result
"yyyy.MM.dd G 'at' HH:mm:ss z"	2001.07.04 AD at 12:08:56 PDT
"EEE, MMM d, ''yy"	Wed, Jul 4, '01
"h:mm a"	12:08 PM
"hh 'o''clock' a, zzzz"	12 o'clock PM, Pacific Daylight Time
"K:mm a, z"	0:08 PM, PDT
"yyyyy.MMMMM.dd GGG hh:mm aaa"	02001.July.04 AD 12:08 PM
"EEE, d MMM yyyy HH:mm:ss Z"	Wed, 4 Jul 2001 12:08:56 -0700
"yyMMddHHmmssZ"	010704120856-0700

Valid Date/Time Formats

The following are valid formats in XHQ.

MM/dd/yy h:mm:ss.SSS a z	MM/dd/yy H:mm:ss	MM/dd/yyyy H:mm z
MM/dd/yy h:mm:ss.SSS a	MM/dd/yyyy h:mm:ss a z	MM/dd/yyyy H:mm
MM/dd/yy H:mm:ss.SSS z	MM/dd/yyyy h:mm:ss a	MM-dd-yy h:mm a z
MM/dd/yy H:mm:ss.SSS	MM/dd/yyyy H:mm:ss z	MM-dd-yy h:mm a
MM/dd/yyyy h:mm:ss.SSS a z	MM/dd/yyyy H:mm:ss	MM-dd-yy H:mm z
MM/dd/yyyy h:mm:ss.SSS a	MM-dd-yy h:mm:ss a z	MM-dd-yy H:mm
MM/dd/yyyy H:mm:ss.SSS z	MM-dd-yy h:mm:ss a	MM-dd-yyyy h:mm a z
MM/dd/yyyy H:mm:ss.SSS	MM-dd-yy H:mm:ss z	MM-dd-yyyy h:mm a
MM-dd-yy h:mm:ss.SSS a z	MM-dd-yy H:mm:ss	MM-dd-yyyy H:mm z
MM-dd-yy h:mm:ss.SSS a	MM-dd-yyyy h:mm:ss a z	MM-dd-yyyy H:mm
MM-dd-yy H:mm:ss.SSS z	MM-dd-yyyy h:mm:ss a	MM/dd/yy
MM-dd-yy H:mm:ss.SSS	MM-dd-yyyy H:mm:ss z	MM/dd/yyyy
MM-dd-yyyy h:mm:ss.SSS a z	MM-dd-yyyy H:mm:ss	MM-dd-yy
MM-dd-yyyy h:mm:ss.SSS a	MM/dd/yy h:mm a z	MM-dd-yyyy
MM-dd-yyyy H:mm:ss.SSS z	MM/dd/yy h:mm a	h:mm:ss a
MM-dd-yyyy H:mm:ss.SSS	MM/dd/yy H:mm z	H:mm:ss
MM/dd/yy h:mm:ss a z	MM/dd/yy H:mm	h:mm a
MM/dd/yy h:mm:ss a	MM/dd/yyyy h:mm a z	H:mm
MM/dd/yy H:mm:ss z	MM/dd/yyyy h:mm a	

You can set the timestamp for XHQ ToolTips and Excel trend data output using the global property **`DefaultTimeStampFormat`**.

For details, go to the topic, *Working with PROPERTIES Files*.

Kerberos Glossary Terms

This section lists definitions or descriptions of Kerberos-related terms and abbreviations.

Term Definitions and Descriptions

Term/Abbreviation	Definition/Description
Authenticate	To establish identity. In Kerberos, this is usually done initially by presenting some credential, such as a password, when the client logs onto the system. After this initial authentication, Kerberos allows the use of tickets and encrypted timestamp information to reliably establish the identity.
Client	An application that wishes to authenticate itself to a Service like XHQ. In regards to using Kerberos with XHQ, a client might be the XHQ applet or applications built using the XHQ Client API, XHQ ADO.NET API, or XHQ Web Services.
Delegation	The server process can impersonate the client's security context on remote systems.
Impersonation	The server process can impersonate the client's security context on its local system. The server cannot impersonate the client on remote systems.
Keytab	A key table. This is a secure table of user names and passwords that may be used by services to identity themselves to the KDC.
KDC	*Key Distribution Center* A central server known to both a Kerberos client and service that supplies Kerberos authentication services. The KDC knows passwords for both the client and the service and acts as an intermediary between the two. It is responsible for issuing Kerberos tickets. In the case of using Kerberos with XHQ, the KDC is a Windows domain controller using Active Directory.
NTLM	*NT LAN Manager* In a Windows network, NTLM is a suite of Microsoft security protocols that provides authentication, integrity, and confidentiality to users. Microsoft no longer recommends using NTLM in applications. Kerberos has replaced NTLM as the default authentication protocol in an Active Directory-based, Single Sign-On (SSO) scheme. NTLM is still widely used in situations where a domain controller is not available or is unreachable. For example, NTLM would be used if a client is not Kerberos capable, the server is not joined to a domain, or the user is remotely authenticating over the web.
Realm	The users, systems, SPNs, and so on, whose security is controlled by a KDC. For XHQ, this is equivalent to a Windows Active Directory domain (for example, EXAMPLE.COM).
Replay Attack	An attempt by some "bad guy" to re-use parts of previous messages that have passed over an un-trusted network to falsely identify himself as some other user. Kerberos protects itself from such attacks by encrypting timestamp and other information with its tickets and raising an authentication error if a particular ticket appears again with the same

Term/Abbreviation	Definition/Description
	timestamp.
Service	A server-based application running on a particular system that can use Kerberos authentication to verify the identity of its clients. The XHQ Server is such a service.
Service Ticket	A small piece of encrypted data issued by the KDC that can be used by a client to identify itself to a particular service, like XHQ. This ticket may be used many times by the client whenever it wishes to make requests to that service, although it usually has a limited lifetime of 8 or so hours.
SPN	*Service Principal Name* A name that identifies a particular Kerberos service that is registered by the KDC. The client uses an SPN to identify the Kerberos service to which it would like to authenticate itself. In regards to XHQ, this identifier is usually something like XHQ/HOSTNAME.example.com@EXAMPLE.COM where *HOSTNAME* is the host name of the XHQ server system.
TGT	*Ticket Granting Ticket* A small piece of encrypted data issued by the KDC that is given to either a client or service after it has successfully authenticated itself to the KDC (for example, by initially logging onto a client system or providing a service with a cached password). A TGT may then be used to identity itself when a client or service wishes to use some Kerberos function.