



About This Guide

---

---

ADO NET Data Provider

1

---

XHQ OPC UA Server

2




---

**XHQ**  
**Integrated Data Gateway**  
**Guide**

## 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.

 <b>DANGER</b>
Indicates that death or severe personal injury <b>will</b> result if proper precautions are not taken.
 <b>WARNING</b>
Indicates that death or severe personal injury <b>may</b> result if proper precautions are not taken.
 <b>CAUTION</b>
Indicates that minor personal injury can result if proper precautions are not taken.
<b>NOTICE</b>
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:

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

### Trademarks

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

### Disclaimer of Liability

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

Copyright © 1998-2019 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

<b>Table of Contents</b>	<b>5</b>
<b>About This Guide</b>	<b>9</b>
Conventions Used in This Guide	9
Visual Cues for Online Viewing	10
Related XHQ Product Documentation	11
Contacting Customer Support	13
General Feedback and Comments	14
<b>1   ADO.NET Data Provider</b>	<b>15</b>
Overview	15
Conventions Used in This Chapter	16
Connecting to an XHQ Data Source	17
To connect to an XHQ data source	17
Basic SQL Support	19
Data Types	19
Identifiers	19
Literals	20
Operators and Delimiters	20
Statements and Clauses	21
SELECT Statement	21
FROM clause	21
JOIN clause	22
GROUP BY clause	22
HAVING clause	22
ORDER BY clause	22
FETCH FIRST clause	22
WHERE clause	23
Operators	23
Functions	25
Using Tables	31
The Collection Data Category	31
Table Schema	31
Table Names	32
Supported Grammar	34

The Solution Metadata Category .....	35
The SOLUTION_INSTANCES Table .....	35
The SOLUTION_MEMBERS Table .....	36
The Solution Values Category .....	38
Schema for this category .....	38
Core Grammar for this Category .....	38
The SOLUTION_VALUES Table .....	39
The SOLUTION_VALUES_ACTUAL Table .....	39
The SOLUTION_VALUES_AT Table .....	40
The SOLUTION_VALUES_AVG Table .....	40
The SOLUTION_VALUES_COUNT Table .....	40
The SOLUTION_VALUES_DELTA Table .....	41
The SOLUTION_VALUES_FIRST Table .....	41
The SOLUTION_VALUES_INTEGRAL Table .....	41
The SOLUTION_VALUES_LAST Table .....	41
The SOLUTION_VALUES_MAX Table .....	42
The SOLUTION_VALUES_MIN Table .....	42
The SOLUTION_VALUES_SUM Table .....	42
The Solution Time-Series (Trend) Data Category .....	43
Schema for this category .....	43
Core Grammar for this Category .....	43
The SOLUTION_TRENDS_ACTUALS Table .....	44
The SOLUTION_TRENDS_ACTUALS_WITH_ADJACENT Table .....	44
The SOLUTION_TRENDS_FITS Table .....	44
The SOLUTION_TRENDS_TIMESLICED_AVG Table .....	45
The SOLUTION_TRENDS_TIMESLICED_INTERPOLATE Table .....	45
The SOLUTION_TRENDS_TIMESLICED_MAX Table .....	45
The SOLUTION_TRENDS_TIMESLICED_MIN Table .....	45
The Model Metadata Category .....	46
The MODEL_CLASSES Table .....	46
The MODEL_CLASS_ROLES Table .....	47
The MODEL_MEMBERS Table .....	47
The MODEL_ROLES Table .....	49
The MODEL_ROLE_PERMISSIONS Table .....	49
The MODEL_VIEWS Table .....	50
The MODEL_VIEW_PROPERTIES Table .....	51
The Tag Metadata Category .....	52
The TAG_COLLECTION Table .....	52

<b>2   XHQ OPC UA Server</b>	<b>53</b>
Server Initialization	53
How to Configure the Server	53
Port Assignments	53
To change the port assignments	53
Firewall Configuration	53
License Requirement	54
About the xhq_opcuaserver.exe.config file	54
To edit the <appSettings> tag	54
To edit the <log4net> tag	57
Security	58
To configure the server certificate issued by a Certificate Authority	58
To configure the client certificate	60
To trust a Certificate Authority	60
To change the default client certificate library	62
Special Security Configurations	62
To use Kerberos authentication	62
Reconnectivity	62
Getting Started	64
To start the application	64
XHQ OPC UA Server Features	65
Browser – DA and HDA Clients	65
Default Members	65
Primitive Member Mapping	65
Tag Object Mapping	66
Units of Measure (UoM)	66
READ – DA Client	67
WRITE – DA Client	67
READ – HDA Client	68
<b>Appendices</b>	<b>69</b>
A - Data Dictionary	70
SOLUTION_INSTANCES	70
SOLUTION_MEMBERS	71
SOLUTION_VALUES	72
SOLUTION_VALUES_ACTUAL	73
SOLUTION_VALUES_AT	74
SOLUTION_VALUES_AVG	75
SOLUTION_VALUES_COUNT	76

SOLUTION_VALUES_DELTA .....	77
SOLUTION_VALUES_FIRST .....	78
SOLUTION_VALUES_INTEGRAL .....	79
SOLUTION_VALUES_LAST .....	80
SOLUTION_VALUES_MAX .....	81
SOLUTION_VALUES_MIN .....	82
SOLUTION_VALUES_SUM .....	83
SOLUTION_TRENDS_ACTUALS .....	84
SOLUTION_TRENDS_ACTUALS_WITH_ADJACENT .....	85
SOLUTION_TRENDS_FITS .....	86
SOLUTION_TRENDS_TIMESLICED_AVG .....	87
SOLUTION_TRENDS_TIMESLICED_INTERPOLATE .....	88
SOLUTION_TRENDS_TIMESLICED_MAX .....	89
SOLUTION_TRENDS_TIMESLICED_MIN .....	90
MODEL_CLASSES .....	91
MODEL_CLASSE_ROLES .....	91
MODEL_MEMBERS .....	92
MODEL_ROLES .....	92
MODEL_ROLE_PERMISSIONS .....	93
MODEL_VIEWS .....	93
MODEL_VIEW_PROPERTIES .....	94
TAG_COLLECTION .....	94
B - Crystal Reports/Visual Studio Integration .....	95
To create a Crystal Report using an XHQ data source .....	95
C - Connecting the OPC Client .....	102
To connect using the OPC Foundation UA Client .....	102
To connect using a Classic OPC Client .....	103
D - Modifying xhqboot .....	106



# 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)
<b>Bold</b> is used for menu names, command options, and dialog box names in primary task procedures.	From the <b>XHQ Workbench</b> , go to the <b>Add</b> menu and click <b>New Component</b> .
<i>Italic</i> is used for glossary terms.	The first step in building this model is to develop reusable software building blocks, called <i>components</i> .
A <b>monospaced font</b> is used for program and code examples.	The subdirectory <code>\log</code> is automatically created below the location you choose. All log files are written to this subdirectory. <code>C:\XHQ</code>
Key combinations appear in uppercase, bold. If joined with a plus sign (+), press and hold the first key while you press the remaining keys.	<b>CTRL+B</b>
The .x (in italics) is used to indicate release numbers of a product.	Enable (by checking) the <b>Use Java x.x.x_xx for &lt;applet&gt;</b> 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 <i>About install.properties</i> topic located in the "Working with PROPERTIES Files" chapter of the <b>XHQ Administrator's Guide</b> .

## 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
<b>XHQ Administrator's Guide</b>  Provides the steps required to begin administering XHQ. It also covers security and access, property settings, redundancy, and localization.	Administrators
<b>XHQ ANS User's Guide</b>  Learn how to use and administer the XHQ Alert Notification System (XHQ ANS).	ANS Users, Administrators
<b>XHQ Backup and Recovery Guide</b>  Learn how to properly backup XHQ.	Administrators
<b>XHQ Connection Guide</b>  Provides information on injecting an XHQ-supported connector type and configuring the connection.	Connector Developers
<b>XHQ Developer's Guide</b>  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
<b>XHQ Getting Started</b>  Gives you step-by-step instruction on how to set up your model and solution.	Content, Connector, and Solution Developers
<b>XHQ Installation Guide</b>  Provides the system requirements, installation instructions, and upgrade information for the current release of the XHQ System.	Administrators
<b>XHQ Integrated Data Gateway Guide</b>  Includes information on the ADO.NET and the XHQ OPC UA Server.	Application Engineers, Integrators
<b>XHQ Performance Analytics Guide</b>  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
<b>XHQ Performance Management Guide</b>  Learn how to use Target Management to monitor performance indicators and eLogs to create shift reports.	Administrators, End Users
<b>XHQ Reference Guide</b>  Lists the functions and methods used in XHQ, and provides examples,	Content and Solution Developers

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

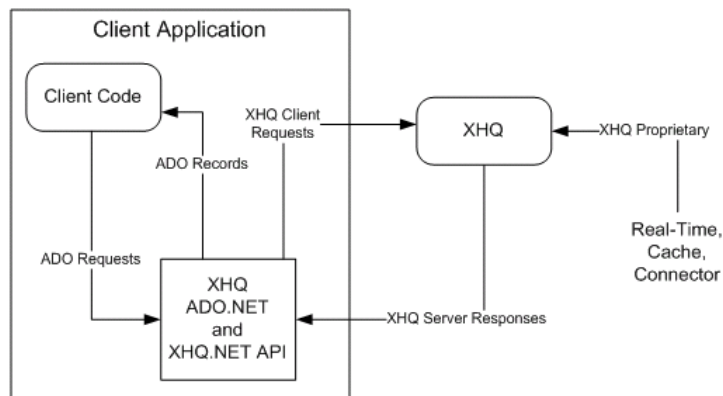
# 1 | ADO.NET Data Provider

The ADO.NET Data Provider is a connectivity tool that allows instant access to XHQ data through standard ADO.NET interfaces.

## Overview

The XHQ ADO.NET Data Provider is a SQL-like exposure of the XHQ Client Data API. For the XHQ System (or XHQ ADO.NET), it is a client-side tool used in the context of an ADO.NET Data Consumer such as a reporting package, Microsoft Visual Studio .NET, or some other Microsoft .NET-based custom program that uses ADO.NET Data Providers. This data provider communicates with the XHQ System through the XHQ Client Data API for Microsoft .NET (XHQ.NET API).

The diagram below shows the general architecture of XHQ ADO.NET in the context of a client application and the XHQ System.



*XHQ ADO.NET General Architecture*

## Conventions Used in This Chapter

The following formatting conventions are used to clearly describe the syntax of supported SQL statements (or similar constructs).



These conventions pertain only to the SQL syntax and do not apply to topic heading styles.

### *Syntax Formatting Convention*

Convention	Description
Mono-spaced font	Used to display the literal syntax of statements.
UPPERCASE	Used to represent SQL keywords.
<i>Italicized</i>	Used to represent an item that may be replaced with other literal syntax.
[ ]	Brackets indicate optional syntax items.
{ }	Braces indicate required syntax items.
	A vertical bar separates syntax items that are enclosed in brackets or braces. You can select either the item before the vertical bar or the item after the bar.
*	An asterisk represents zero or more repetitions of the preceding item.

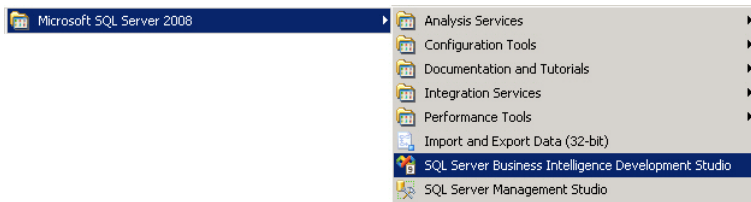


## Connecting to an XHQ Data Source

The following procedure gives you step-by-step instructions on how to connect to an XHQ data source and create a report with the use of the SQL Server Business Intelligence Developer.

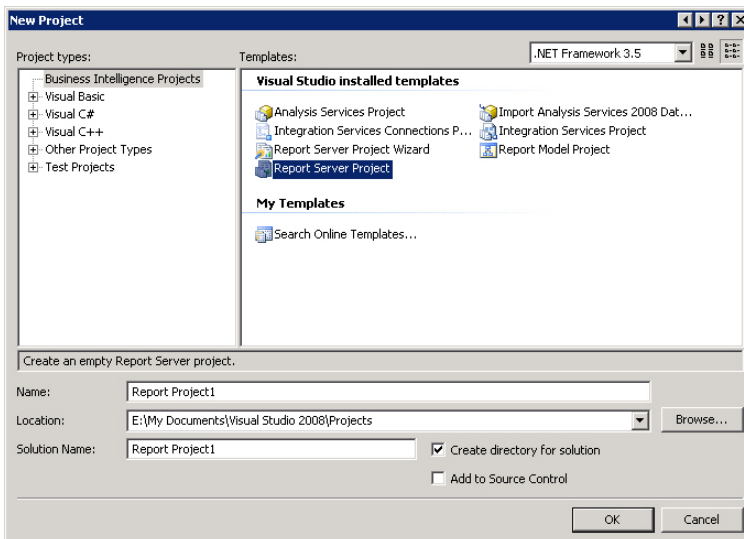
### To connect to an XHQ data source

1. From the **Start menu**, point to **Programs** then to **Microsoft SQL Server**, and then click **SQL Server Business Intelligence Development Studio**.

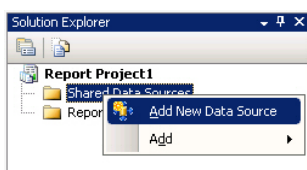


The Microsoft Visual Studio "Start Page" appears.

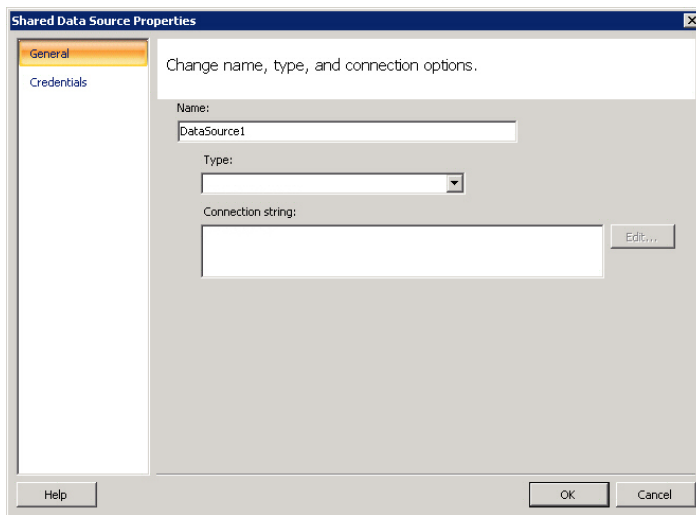
2. On the **File** menu, click **New Project**.  
The "New Project" dialog appears.
3. In the **Project Types** list, click (to highlight) **Business Intelligence Projects**.
4. In the **Templates** list, click **Report Server Project**.



5. Enter a **Name** and click **OK**.
6. From the **Solution Explorer**, locate the report project you just created.
7. Right-click the **Shared Data Source** folder, and from the shortcut menu, click **Add New Data Source**.



The "Shared Data Source Properties" dialog box appears.



8. In the **Type** list, select **XHQ Data Provider**.

9. In the **Connection string** box, type:

```
data source=<computername>; default time zone=XHQ Solution Server Time
```

*Where:*

- <computername> is the name of the XHQ data source provider.
- The **default** time zone parameter value is **XHQ Solution Server Time**, which sets both the input and output date/time to the time zone of the XHQ Solution Server. In general, this time zone notifies the Data Provider that all dates and/or times provided as an input or output is in this given time zone.



A complete list of time zones can be found in the registry at:

`HKLM\Software\Microsoft\WindowsNT\CurrentVersion\TimeZones`

Only UTC can be abbreviated. For all other time zones, enter the complete string. For example, Mountain Standard Time.



The ADO.NET Connect String has changed from prior releases. Refer to the XHQ ReadMe, located at the root directory of the XHQ installation media, for known upgrade issues

10. OPTIONAL

Use the following **Connection string parameters**:

- **trace level=<0-4>**

This outputs tracing information to the debug window, and optionally to the filename specified in the trace log file parameter. The trace levels are:

0 = None (off)	3 = Information
1 = Errors	4 = Verbose
2 = Warnings	

- **trace log file=<fully qualified path to log file>**

*Example:* A Connection string that uses the data source and trace parameters.

```
data source=xhqhost; trace level=3; trace log file=c:\xhq_ado.log
```

11. Click **OK**.

## Basic SQL Support

The XHQ ADO.NET Data Provider supports a limited subset of the SQL99 Standard, with limited support for table joins and other operations.



This topic makes reference to tables. For more information on tables, go to the topic, [Using Tables](#).

Also, with regards to SQL99, please refer to the SQL99 specification for additional information not documented in this guide (for example, reserved words).

## Data Types

The following SQL99 data types are supported.

Data Type	Description
BOOLEAN	Boolean logical data.
NVARCHAR	Localized (national) character data of varying length.
LONG	Integer number (64-bit integer). <b>Note:</b> This is NOT part of the SQL99 specification.
INTEGER	Integer number (32-bit integer) data.
SMALLINT	Integer number (approximately 16-bit integer) data.
REAL	Real number (floating-point number) data.
DATETIME	Date and time data.
INTERVAL	Time interval (time span) data.
TIMESTAMP	The syntax is: <code>TIMESTAMP 'yyyy-MM-dd hh:mm:ss.fffffffff'</code> <b>Note:</b> Fractional seconds are optional.

## Identifiers

There are two types of identifiers defined by the ADO.NET Data Provider: **static** and **dynamic** identifiers.

**Static identifiers** are used to identify tables and columns in the static schema. Static identifiers adhere to the SQL99 standard for identity rules paraphrased below:

- Identifiers no longer than 128 characters.
- Identifiers contain characters that are letters, numbers or symbols.
- Identifiers begin with a letter.
- Identifiers do not contain any spaces or special characters.
- Identifiers are never a reserved word or keyword.
- Identifiers are always unique within their scope.
- Identifiers may be quoted.

**Dynamic identifiers** are associated with collection data tables and are determined solely by the XHQ Model and XHQ Solution to which the connection has been made. In general, the dynamic identifiers for column names adhere to the same rules for static identifiers, with a couple of exceptions.

#### SPECIAL CASE: COLLECTION DATA TABLE NAMES

Collection data table names are defined by the name of the associated collection. Since these names follow the syntax of XHQ Solution paths, the table names necessarily violate some of the SQL99 rules for identity. The following are specific exceptions to the rules (which were given for static identifiers) with regards to dynamic identifiers for collection data table names.

- Identifiers may be **greater** than 128 characters.
- Identifiers **begin with a double-colon** and may have one or more periods in the name.

## Literals

The ADO.NET Data Provider supports numeric and string and/or character literals.

**Numeric literals** follow the SQL99 standard, which allow the following symbols for numeric literals:

0 1 2 3 4 5 6 7 8 9 + - \$ . E e

**String literals** also follow the SQL99 standard. All strings are enclosed with single quotes.

## Operators and Delimiters

The following table lists the set of supported operators and delimiters.

#### *Operators and Delimiters*

Symbol	Description
+	Addition operator (or unary positive operator)
-	Subtraction operator (or unary negative operator)
*	Multiplication operator
/	Division operator
=	Equality operator
<>	Inequality operator
<	Less-than operator
>	Greater-than operator
<=	Less-than or equal-to operator
>=	Greater-than or equal-to operator
(	Begin grouping delimiter
)	End grouping delimiter

Symbol	Description
,	List item separator
*	All items indicator
.	Identity qualifier separator
"	Quoted-identifier delimiter (must have matching open and close)
'	Character string delimiter (must have matching open and close)
+	Character string concatenation operator

## Statements and Clauses

The following statements and clauses are supported by the ADO.NET Data Provider.

### SELECT Statement

The primary statement supported by the ADO.NET Data Provider is the **SELECT** statement. The **SELECT** statement allows access to the various data in the defined tables and determines how the data is formatted and returned. There are limitations and requirements on how **SELECT** statements can be formed, depending on which table or tables are being queried.



For more information on tables, go to the topic, [Using Tables](#).

**Syntax:**     **SELECT** [**DISTINCT**] *select\_list*  
                   *from\_clause*  
                   [*join\_clause*]  
                   [*where\_clause*]  
                   [*order\_by\_clause*]

Where **select\_list** can be a *non\_aggregate\_list*:

- \*

Or, **select\_list** can be an *aggregate\_list*:

- *aggregate\_fn*[, *aggregate\_fn*]\*

### FROM clause

The **FROM** clause allows you to specify a table with an optional alias from which data is to be selected. The ADO.NET Data Provider supports only a single table reference in the **FROM** clause.

**Syntax:**                    **FROM** *table\_name* [**[AS]** *alias\_name*]

## JOIN clause

The **JOIN** clause allows one or more tables that support this operation to be joined together so that the intersection of the tables may be returned as a result set. The ADO.NET Data Provider only supports inner joins and left outer joins (both of which are only supported among specific tables on specific fields).



For more information on tables, go to the topic, [Using Tables](#).

**Syntax:** `[[LEFT|INNER] JOIN table [AS alias] ON criteria [[LEFT|INNER] JOIN table [AS alias] ON criteria]*`

## GROUP BY clause

The **GROUP BY** clause groups a selected set of rows into a set of summary rows by the values of one or more columns. Aggregate functions in the SELECT clause provide information about each group instead of individual rows.



The ADO.NET Data Provider only supports "grouping" on XHQ Collections.

**Syntax:** `GROUP BY column [, column]*`

## HAVING clause

The **HAVING** clause specifies a search condition for a GROUP BY aggregate clause.



The ADO.NET Data Provider only supports "having" on XHQ Collections.

**Syntax:** `HAVING <search condition>`

## ORDER BY clause

The **ORDER BY** clause specifies the order in which records are returned in the result set.



The ADO.NET Data Provider only supports "ordering" on any column or query.

**Syntax:** `ORDER BY column_or_position [ASC|ENDING] DESC [ENDING]] [, column_or_position [ASC|ENDING] DESC [ENDING]]*`

## FETCH FIRST clause

The **FETCH FIRST** clause causes the provider to return only the first <x> rows of the result set.

**Syntax:** `FETCH FIRST <x> ROWS ONLY`

## WHERE clause

The **WHERE** clause allows filtering of records from the result set and/or governs how the XHQ System searches for data to satisfy a request. Each table exposed by the ADO.NET Data Provider supports filtering on certain columns. Some tables may require that certain columns be constrained in the **WHERE** clause to construct a valid query.

In general, the syntax of the **WHERE** clause supports many combinations of conditional expressions.



For details on how each table supports the WHERE clause, go to the topic, [Using Tables](#).

The following sub-clauses may follow the **WHERE** keyword to form the full **WHERE** clause.

### Examples of Conditional Expressions

Conditional Expression	Example
Simple Boolean expression ( <i>comparison_expression</i> )	<i>expression comparison_operator expression</i>
Check for NULL ( <i>null_check_expression</i> )	<i>column_name</i> IS [NOT] NULL
Check for string LIKENESS ( <i>like_check_expression</i> )	[NOT] <i>column_name</i> LIKE ' <i>string_pattern</i> '
Check for ranges ( <i>between_check_expression</i> )	<i>expression</i> BETWEEN <i>expression</i> AND <i>expression</i>
Check for values in a set ( <i>in_check_expression</i> )	<i>expression</i> ( <i>expression</i> [, <i>expression</i> ]*)
General Boolean condition ( <i>boolean_expression</i> )	{ <i>comparison_expression</i>   <i>null_check_expression</i>   <i>like_check_expression</i>   <i>between_check_expression</i>   <i>in_check_expression</i> }
Logical expression ( <i>logical_expression</i> )	[ ( <i>boolean_expression</i> [{AND} <i>boolean_expression</i> ]*) ]
Complete WHERE clause ( <i>where_clause</i> )	WHERE <i>logical_expression</i> [{AND} <i>logical_expression</i> ]*

## Operators

The following tables list the supported operators.

### Arithmetic Operators

Symbol	Description	Symbol	Description
+	Adds the left operand to the right operand.	*	Multiplies the left operand by the right operand.

Symbol	Description	Symbol	Description
-	Subtracts the right operand from the left operand.	/	Divides the left operand by the right operand.

*String Operator*

Symbol	Description
+	Concatenates the right operand to the end of the left operand.

*Comparison Operators*

Symbol	Description	Symbol	Description
=	Equality operator	>	Greater-than operator
<>	Inequality operator	<=	Less-than or equal-to operator
<	Less-than operator	>=	Greater-than or equal-to operator

*Logical Operators*

Operator	Description
AND	TRUE if both Boolean expressions evaluate to TRUE.
BETWEEN	TRUE if the operand is with the supplied range.
IN	TRUE if the operand is equal to one of the list of expressions.
LIKE	TRUE if the operand matches the supplied pattern.
OR	TRUE if either Boolean expression evaluates to TRUE.

*Union Operator*

Operator	Description
UNION	This operator is used to combine the resulting set of two (or more) SELECT statements. Each SELECT statement within the UNION must have the same number of columns. The columns must have similar data types. And, the columns in each SELECT statement must be in the same order. <b>Note:</b> The UNION operator does not remove duplicate rows.

*Unary Operator*

Symbol	Description
-	The numeric operand is negative.



## Functions

The following functions are supported by the ADO.NET Data Provider.

### Supported Functions

Function	Description
<b>AVG</b>	<p>The <b>AVG</b> aggregate function computes the average value of a column in a table. This function works with numeric columns and the returned data type is <b>REAL</b>. Although <b>NULL</b> values are ignored for the calculation and indication of the presence of <b>NULL</b> values or if all values in the column are <b>NULL</b> can be determined by using the <b>QUALITY</b> functions on the result of the <b>AVG</b> function.</p> <p><b>Important:</b> This function does not support <b>ALL</b> or <b>DISTINCT</b> as found in the <b>SQL99</b> specification. This function is only supported on <b>XHQ Collection</b> queries.</p> <p><i>Syntax:</i> <code>AVG (column_name)</code></p>
<b>CAST</b>	<p>This function converts data of one type to that of another type. You may <b>CAST</b> any supported data type to other supported data type. However, some conversions may lead to an error. For example, casting a string literal or an <b>NVARCHAR</b> value that does not contain a properly formatted number to a <b>REAL</b> value can result in an error.</p> <p><i>Syntax:</i>  <code>CAST (expression AS data_type [(length)])</code></p>
<b>CHAR_LENGTH</b>	<p>This function returns the number of characters in a string type value or column.</p> <p><i>Syntax:</i>  <code>CHAR_LENGTH (string_expression)</code></p>
<b>COALESCE</b>	<p>This function returns the first non-null expression from its arguments. If all arguments are <b>NULL</b>, then <b>COALESCE</b> returns a <b>NULL</b>.</p> <p><i>Syntax:</i>  <code>COALESCE (expression [, ...n])</code></p> <p><i>Example:</i>  <pre>SELECT COALESCE (col1, col2, col3) FROM "::MyCollection"</pre></p>
<b>CONVERT_TO_UNITS</b>	<p>This function converts an input value from one unit of measure to another unit of measure. The return type is double.</p> <p><i>Syntax:</i>  <code>CONVERT_TO_UNITS (input_value, from_units, to_units)</code></p>
<b>COUNT</b>	<p>The <b>COUNT</b> aggregate function counts the number of non-<b>NULL</b> rows of a column in a table. Alternately, the <b>COUNT (*)</b> aggregate function counts the total number in a table. This function works with any type column and the returned data type is <b>INTEGER</b>.</p>

Function	Description
	<p>Although NULL values are ignored for the calculation and indication of the presence of NULL values or if all values in the column are NULL can be determined by using the QUALITY functions on the result of the COUNT function.</p> <p><b>Important:</b> This function does not support ALL or DISTINCT as found in the SQL99 specification. This function is only supported on XHQ Collection queries.</p> <p><i>Syntax:</i>COUNT(<i>column_name</i>) COUNT(*)</p>
CURRENT_DATE	<p>This returns the current date (no time portion) from the local system. The type is DATETIME.</p> <p><i>Syntax:</i> CURRENT_DATE</p>
CURRENT_TIME	<p>This returns the current time (no date portion) from the local system. The type is DATETIME.</p> <p><i>Syntax:</i> CURRENT_TIME</p>
CURRENT_TIMESTAMP	<p>This returns the current date (no time portion) from the local system. The type is DATETIME.</p> <p><i>Syntax:</i> CURRENT_TIMESTAMP</p>
DATEADD	<p>This returns TIMESTAMP value from an incoming TIMESTAMP value modified by the given date part and value.</p> <p><i>Syntax:</i> DATEADD(<i>date_part</i>, <i>integer_expression</i>, <i>date_expression</i>)</p> <p><i>Example:</i> DATEADD(MONTH, 3, TIMESTAMP '2009-01-01 00:00:00')</p>
DATEBEGIN	<p>Given a timestamp value, this returns the timestamp value with the time portion set to 00:00:00.</p>
DATEDIFF	<p>This returns an integer value, given a date part and two timestamp expressions. The return value represents the difference between the two timestamp expressions, expressed in date part units.</p> <p><i>Syntax:</i> DATEDIFF(<i>date_part</i>, <i>timestamp_expression_begin</i>, <i>timestamp_expression_end</i>)</p> <p><i>Example:</i> DATEDIFF(MONTH, TIMESTAMP '2009-01-01 00:00:00', TIMESTAMP '2009-03-01 00:00:00')</p>
DATEEND	<p>Given a timestamp value, this returns the timestamp value with the</p>

Function	Description																					
	time portion set to 23:59:59 (which is the last tick of the given date).																					
DEFAULT_TIMEZONE	This returns the value of the Default Time Zone connection string parameter. If no Default Time Zone connection string parameter is specified, it returns the current default value.																					
EXTRACT	<p>This returns a numeric (INTEGER) representation of year, month, day, hour, minute or second of a DATETIME type value.</p> <p><i>Syntax:</i></p> <p>EXTRACT(<i>extraction</i> FROM <i>date_time_expression</i>)</p> <p>The following table shows the various values for <i>extraction</i>.</p> <table><tr><th>extraction</th><th>Extracted Value</th><th>Range</th></tr><tr><td>SECOND</td><td>The seconds portion of the expression.</td><td>0-59</td></tr><tr><td>MINUTE</td><td>The minutes portion of the expression.</td><td>0-59</td></tr><tr><td>HOUR</td><td>The hours portion of the expression.</td><td>0-23</td></tr><tr><td>DAY</td><td>The day portion of the expression.</td><td>1-max for month</td></tr><tr><td>MONTH</td><td>The month portion of the expression.</td><td>1-12</td></tr><tr><td>YEAR</td><td>The year portion of the expression.</td><td>any</td></tr></table>	extraction	Extracted Value	Range	SECOND	The seconds portion of the expression.	0-59	MINUTE	The minutes portion of the expression.	0-59	HOUR	The hours portion of the expression.	0-23	DAY	The day portion of the expression.	1-max for month	MONTH	The month portion of the expression.	1-12	YEAR	The year portion of the expression.	any
extraction	Extracted Value	Range																				
SECOND	The seconds portion of the expression.	0-59																				
MINUTE	The minutes portion of the expression.	0-59																				
HOUR	The hours portion of the expression.	0-23																				
DAY	The day portion of the expression.	1-max for month																				
MONTH	The month portion of the expression.	1-12																				
YEAR	The year portion of the expression.	any																				
LOWER	<p>This returns a string with all lowercase characters converted from a supplied string expression.</p> <p><i>Syntax:</i> LOWER(<i>string_expression</i>)</p>																					
MIN	<p>The MIN aggregate function finds the minimum value of a column in a table. This function works with any numeric-type columns and the returned data type is the same type as the column. Although NULL values are ignored for the calculation and indication of the presence of NULL values or if all values in the column are NULL can be determined by using the QUALITY functions on the result of the MIN function.</p> <p><b>Important:</b> This function does not support ALL or DISTINCT as found in the SQL99 specification. This function is only supported on XHQ Collection queries.</p> <p><i>Syntax:</i> MIN(<i>column_name</i>)</p>																					
MAX	<p>The MAX aggregate function finds the maximum value of a column in a table. This function works with any numeric type columns and the returned data type is the same type as the column. Although NULL values are ignored for the calculation and indication of the presence of NULL values or if all values in the column are NULL can</p>																					

Function	Description
	<p>be determined by using the <code>QUALITY</code> functions on the result of the <code>MAX</code> function.</p> <p><b>Important:</b> This function does not support <code>ALL</code> or <code>DISTINCT</code> as found in the SQL99 specification. This function is only supported on XHQ Collection queries.</p> <p><i>Syntax:</i> <code>MAX(column_name)</code></p>
<b>POSITION</b>	<p>This returns an integer indicating the position of a pattern in a string or <code>NULL</code>, if the pattern cannot be found or either the pattern or the string is <code>NULL</code>.</p> <p><i>Syntax:</i>  <code>POSITION(pattern IN string_expression)</code></p>
<b>QUALITY_NAME , QUALITY_DESCRIPTION , QUALITY_SHORT</b>	<p>These <code>QUALITY</code> functions return the quality associated with an aggregate function, collection record set, or a column of collection record set. <b>Important:</b> These functions are not part of the SQL99 standard and are a special extension of the ADO.NET Data Provider. These functions can only be used in conjunction with an aggregate function or on collection data. Their data type is determined by the formatting argument.</p> <p><i>Syntax:</i> <code>QUALITY_NAME(selector)</code>  Type is <code>NVARCHAR</code></p> <p><code>QUALITY_DESCRIPTION(selector)</code>  Type is <code>NVARCHAR</code></p> <p><code>QUALITY_SHORT(selector)</code>  Type is <code>SMALLINT</code></p> <p>Where <i>selector</i> can be one of the following:</p> <ul style="list-style-type: none"> <li><code>*</code></li> <li><code>column_name</code></li> <li><code>COUNT (*)</code></li> <li><code>aggregate_function(column_name)</code></li> </ul>
<b>SOLUTION_SERVER_TIMEZONE</b>	<p>This returns the current time zone of the XHQ Solution Server.</p>
<b>SUBSTRING</b>	<p>This extracts a string from another string. The function returns <code>NULL</code>, if any input value is <code>NULL</code>.</p> <p><b>Note:</b> The ADO.NET Data Provider does not support the <code>COLLATE</code> option <code>SUBSTRING</code> function as specified by SQL99.</p> <p><i>Syntax:</i>  <code>SUBSTRING(string_expression FROM start_position [FOR char_count])</code></p>
<b>STDDEV_POP</b>	<p>The <code>STDDEV_POP</code> aggregate function computes the population standard deviation and returns the square root of the population variance.</p>

Function	Description
	<p><b>Important:</b> This function is only supported on XHQ Collection queries.</p> <p><i>Syntax:</i> STDDEV_POP(<i>column_name</i>)</p>
STDDEV_SAMP	<p>This aggregate function computes the cumulative sample standard deviation and returns the square root of the sample variance.</p> <p><b>Important:</b> This function is only supported on XHQ Collection queries.</p> <p><i>Syntax:</i> STDDEV_SAMP(<i>column_name</i>)</p>
SUM	<p>The SUM aggregate function computes the sum of all values of a column in a table. This function works with any type column and the returned data type is DECIMAL. Although NULL values are ignored for the calculation and indication of the presence of NULL values or if all values in the column are NULL can be determined by using the QUALITY functions on the result of the AVG function.</p> <p><b>Important:</b> This function does not support ALL or DISTINCT as found in the SQL99 specification. This function is only supported on XHQ Collection queries.</p> <p><i>Syntax:</i> SUM(<i>column_name</i>)</p>
TRIM	<p>This returns a string with characters removed from either or both ends of the supplied string expression. By default, the function removes spaces from the beginning and/or end of the string. However, a different string may be specified to remove other characters or strings from the target string.</p> <p><b>Important:</b> The LEADING, TRAILING, and BOTH parameters are not yet implemented. The TRIM function simply removes the leading and trailing space from a string.</p> <p><b>Note:</b> The ADO.NET Data Provider does not support the COLLATE option TRIM function as specified by SQL99.</p> <p><i>Syntax:</i> TRIM([<i>removal_string</i>] FROM] <i>string_expression</i>)</p>
UPPER	<p>This returns a string with all uppercase characters converted from a supplied string expression.</p> <p><i>Syntax:</i> UPPER(<i>string_expression</i>)</p>
UTC_OFFSET	<p>This function returns the time offset (in minutes) of the time value from UTC for the given timestamp/time zone combination. It accepts a timestamp value and a time zone name as parameters.</p> <p><i>Syntax:</i> UTC_OFFSET(TIMESTAMP, TIMEZONE)</p>
VAR_POP	<p>The VAR_POP aggregate function computes the population variance of a set of numbers after discarding nulls in the set.</p>

Function	Description
	<b>Important:</b> This function is only supported on XHQ Collection queries. <i>Syntax:</i> VAR_POP ( <i>column_name</i> )
<b>VAR_SAMP</b>	The VAR_SAMP aggregate function computes the sample variance of a set of numbers after discarding nulls in the set. <b>Important:</b> This function is only supported on XHQ Collection queries. <i>Syntax:</i> VAR_SAMP ( <i>column_name</i> )

## Using Tables

The tables that are supported by the data provider are divided into the following categories:

- Collection Data
- Solution Metadata
- Solution Values
- Solution Time-series (Trend) Data
- Model Metadata
- Tag Metadata

Each table category subsection provides a description of the table's purpose and lists the supported grammar.

### The Collection Data Category

Collection data tables are defined by the configuration of the XHQ Solution. Each global collection and link collection in the XHQ Solution namespace manifests as a unique table in the ADO.Net Data Provider. The name of each of these tables is defined by the name of the global collection or the link collection it represents.

#### Table Schema

The schema of each table is defined by the XHQ Model class from which the represented collection is based. Each primitive member of the class is represented by column of the same name in the table. The column data type is analogous to the primitive member's data type in the XHQ System. The following shows the XHQ System data type and the analogous collection table column data type.

##### Data Types

XHQ Data Type	Column Data Type
Boolean	BOOLEAN
Datetime	DATETIME
Decimal	DECIMAL
Double	FLOAT
Integer	INTEGER
Long	LONG
Real	REAL
String	NVARCHAR
Text	NVARCHAR

## Table Names

The name (identifier) of each of these tables is defined by the name of the collection. These identifiers may be very long, depending on the fully qualified name of the collection.



Since these identifiers include characters that may conflict with standard SQL99 delimiters or separators, these **identifiers must be enclosed by double quotes**.

### For Global Collections

Global collection table names have the form:

```
"::collection_name"
```

The parts of the name are defined as follows:

- The double-colon prefix indicates the standard XHQ Solution namespace for "top-level-instance".
- The *collection\_name* portion is the simple name of the global collection.

### For Link Collections

A link collection is a member of another object in the solution namespace that is linked to a global collection. The linked collection may impose extra constraints on the global collection data and act as a filter.

Link collection table names have the form:

```
"::root_name[.sub_member_name]*.collection_member_name"
```

The parts of the name are defined as follows:

- The double-colon prefix indicates the standard XHQ Solution namespace for "top-level-instance".
- The period indicates a delimiter between members in the path to the link collection.
- The *root\_name* portion represents the name of the root object in the XHQ Solution.
- The *[.sub\_member\_name]\** portion represents zero or more intervening member between the root object and the link collection member.
- The *collection\_member\_name* portion is the member name of the link collection.



### For Nested Collections

A nested collection is a member of another collection (global collection or link collection) that is linked to a global collection. The nested collection presents a collection of information in the context of a row of the parent collection.

Nested collection table names have the forms:

`"global_collection_name.nested_collection_member_name"`

`"link_collection_name.nested_collection_member__name"`

The parts of the name are defined as follows:

- The `global_collection_name` portion is described above under Global Collections.
- The `link_collection_name` portion is described above under Link Collections.
- The period indicates a delimiter between members in the path to the link collection.
- The `nested_collection_member__name` portion is the member name of the nested collection.



An ADO query for a Nested Collection is not directly possible. This complexity arises from the fact that the call provides a list of columns to select in the query but, in this one particular circumstance, the call would need to query every column from the parent query (even the columns that were not requested in the call itself). The required approach is to first query all members of the parent collection and then pass that as the collection value to the sub-query.

## Supported Grammar

Global collection tables and link collection tables support the following forms of **SELECT**.

Description	Syntax
Non-aggregate selection list or * for all columns	<code>SELECT [DISTINCT] non_aggregate_list from clause [where_clause] [order_by_clause]</code>
Aggregate-only selection	<code>SELECT aggregate_list from clause [where_clause] [order_by_clause]</code>



Sorting by expressions is currently not supported.



For information on statements and clauses, go to the topic, [Statements and Clauses](#).

Nested collection tables support the following two forms of **SELECT**.

Description	Syntax
Non-aggregate selection list or * for all columns	<code>SELECT [DISTINCT] non_aggregate_list FROM parent_collection_name [[AS] alias_name] JOIN nested_collection_name [[AS] alias_name] ON pc = nc [where_clause] [order_by_clause]</code>
Aggregate-only selection	<code>SELECT aggregate_list FROM parent_collection_name [[AS] alias_name] JOIN nested_collection_name [[AS] alias_name] ON pc = nc [where_clause] [order_by_clause]</code>

In the case of nested collection tables:

- The *parent\_collection\_name* may be any valid global or link collection table name.
- The *nested\_collection\_name* may be any valid nested collection table name. A nested collection may also be referred to by its parent collection alias in the JOIN clause.

*Example:*      `FROM "::EQUIPMENT" eq JOIN "eq.WORKORDERS" wo ON eq = wo`

## The Solution Metadata Category

Tables in this category provide access to metadata about the solution.

### The *SOLUTION\_INSTANCES* Table

This table gives details on *instances* in the XHQ Solution namespace. Instances are defined as objects, tags, or global collections. Each object, tag, and global collection has an entry in the **SOLUTION\_INSTANCES** table that describes each in detail.

The main purpose of this table is to be joined with the **SOLUTION\_MEMBERS** table or one of the **SOLUTION\_VALUES** or **SOLUTION\_TRENDS** tables when instance information is required in the returned result set. In addition, this table can also query the global collections, determine the tag collection, query the full tag set, or find a specific object or collection.

#### Schema

The **SOLUTION\_INSTANCES** table has the following columns.

Column Name	Description
OID	The ID of the instance.
NAME	The simple name of the instance.
FULLY_QUALIFIED_NAME	The fully-qualified path for the instance.
DESCRIPTION	The description of the instance.
INSTANCE_TYPE	The string that indicates whether it is an OBJECT or a COLLECTION.
TAG	The flag that indicates whether it is a Tag Object or a Tag Collection.
REVISION_ID	The revision ID of the instance.
CLASS_ID	The ID of the instance's class.
CLASS_REVISION_ID	The revision ID of the instance's class.
CLASS_NAME	The name of the instance's class.



For a full specification of the columns and keys, go to section, [Data Dictionary](#).

#### Core Grammar

The **SOLUTION\_INSTANCES** table may participate as a **right-hand table in a join** with the following tables:

- **SOLUTION\_MEMBERS**

The **JOIN** clause must be of the form (where *m* is the alias for the other table)

```
JOIN SOLUTION_INSTANCES i ON m.PARENT_OID = i.OID
```

- **SOLUTION\_VALUES** table type

The **JOIN** clause must be of the form (where *v* is the alias for the other table)

```
JOIN SOLUTION_INSTANCES i ON v.PARENT_OID = i.OID
```

- SOLUTION\_TRENDS table type

The **JOIN** clause must be of the form (where *t* is the alias for the other table)

```
JOIN SOLUTION_INSTANCES i ON t.PARENT_OID = i.OID
```

The **SOLUTION\_INSTANCES** table may be the sole **left-hand table in a query** and queried in the following ways.

- Find a specific object by ID:

```
SELECT {non_aggregate_list} FROM SOLUTION_INSTANCES WHERE OID = integer_expression
```

- Find a specific object by class name:

```
SELECT {non_aggregate_list} FROM SOLUTION_INSTANCES WHERE CLASS_NAME = string_expression
```



The columns from the **SOLUTION\_INSTANCES** table may not participate in any **ORDER BY** directive.

## The SOLUTION\_MEMBERS Table

This table gives details on members in the XHQ Solution namespace.

### Table Schema

The **SOLUTION\_MEMBERS** table has the following columns.

Column Name	Description
FULLY_QUALIFIED_NAME	The fully-qualified path to the member.
PARENT_PATH	The full path to the member's parent.
NAME	The name of the member.
PARENT_OID	The object ID number of the member's parent.
MID	The member ID number.
DESCRIPTION	The description of the member.
MEMBER_TYPE	The type of member (PRIMITIVE, OBJECT, or COLLECTION).
DATA_TYPE	The name of the class or primitive type of the member.
REFERENCE_OID	The object ID of the referenced object (if MEMBER_TYPE is OBJECT).
ALIAS_NAME	The alias of the member (only used when MEMBER_TYPE is PRIMITIVE).
ATTRIBUTE_TYPE	The attribute of the member (only used when MEMBER_TYPE is PRIMITIVE).
RESTRICTED	This indicates whether or not access to the member is restricted.

Column Name	Description
HIDDEN	This indicates whether or nor the member is hidden.
PROXY	This indicates whether or not the member is a proxy to another object (only used when MEMBER_TYPE is OBJECT).



For a full specification of the columns and keys, go to section, [Data Dictionary](#).

## Core Grammar

The **SOLUTION\_MEMBERS** table may participate as a **right-hand table in a join** with the following tables:

- **SOLUTION\_VALUES** table type

The **JOIN** clause must be of the format (where v is the alias for the other table):

```
JOIN SOLUTION_MEMBERS m ON v.FULLY_QUALIFIED_NAME = m.FULLY_QUALIFIED_NAME
```

- **SOLUTION\_TRENDS** table type

The **JOIN** clause must be of the format (where t is the alias for the other table):

```
JOIN SOLUTION_MEMBERS m ON t.FULLY_QUALIFIED_NAME = m.FULLY_QUALIFIED_NAME
```

The **SOLUTION\_MEMBERS** table may be the sole **left-hand table in a query** and queried in the following ways.

- Find specific members of a parent member or the count of these (with possible filtering):

```
SELECT {non_aggregate_list}FROM SOLUTION_MEMBERSWHERE PARENT_PATH = path_specifier_string[AND logical_expression]*
```

- Find a specific member based on the key or the count of these (0 or 1):

```
SELECT {non_aggregate_list}FROM SOLUTION_MEMBERSWHERE FULL_QUALIFIED_NAME = fq_n_string
```

The **SOLUTION\_MEMBERS** table may be the **left-hand table in a join** with the **SOLUTION\_INSTANCES** table.

The table may be queried in the following manner:

```
SELECT {non_aggregate_list}FROM SOLUTION_MEMBERS mJOIN SOLUTION_INSTANCES i ON m.PARENT_OID = i.OID[where_clause] [order_by_clause]
```

The **SOLUTION\_MEMBERS** table may also be the **middle table in a join** with a **SOLUTION\_VALUES\*** or **SOLUTION\_TRENDS\*** table and the **SOLUTION\_INSTANCES** table. The table may be queried in the following manner:

```
SELECT {non_aggregate_list}FROM SOLUTION_VALUES vJOIN SOLUTION_MEMBERS m ON v.FULLY_QUALIFIED_NAME = m.FULLY_QUALIFIED_NAMEJOIN SOLUTION_INSTANCES i ON m.PARENT_OID = i.OID[where_clause] [order_by_clause]
```

The following columns from the **SOLUTION\_MEMBERS** table may participate in any **ORDER BY** directive when the table is the only table in a **SELECT** or the left-hand table in a join.

- FULL\_QUALIFIED\_NAME
- PARENT\_PATH, NAME
- PARENT\_OID, MID

## The Solution Values Category

Tables in this category provide access to the primitive member value. The mode of access to the member values is determined by the specific table used. There are three subgroups of access:

- Current value of a member (**SOLUTION\_VALUES** table);
- Value of a member at specific time (**SOLUTION\_VALUES\_ACTUAL** and **SOLUTION\_VALUES\_AT** tables);
- Aggregate value of a member over a time range (all other tables).

### Schema for this category

All tables in this category have the following columns in common.

Column Name	Description
FULLY_QUALIFIED_NAME	The fully-qualified path to the member.
PARENT_PATH	The full path to the member's parent.
NAME	The name of the member.
PARENT_OID	The object ID number of the member's parent.
MID	The member ID number.
DATA_TYPE	The name of the class or primitive type of the member.
VALUE_INT	The integer representation of the value.
VALUE_BOOL	The Boolean representation of the value.
VALUE_REAL	The real number (floating-point) representation of the value.
VALUE_STRING	The text representation of the value.
QUALITY_NAME	The quality of the value represented as string.
QUALITY_DESCRIPTION	The description of the quality.
QUALITY_SHORT	The OPC-encoded quality of the value.
TIME_STAMP	The time-stamp of the value.

### Core Grammar for this Category

All tables in this category follow these common grammar rules.

- A value table must be the only table in a **SELECT** statement or the left-hand table in a **JOIN** with either the **SOLUTION\_MEMBERS** table or the **SOLUTION\_INSTANCES** table.
- A value table may participate as a right-hand table in a join with other value tables. The **JOIN** clause must be of the following format (where *v1* is the alias for the left-hand value table and *values\_table* is right-hand value table):

```
JOIN values_table v2 ON v1.FULLY_QUALIFIED_NAME = v2.FULLY_QUALIFIED_NAME
```

- Restrictions are imposed on the `WHERE` clause of all selections from the tables. In general, the `WHERE` clause must include conditions that uniquely identify specific primitive members.

The `WHERE` clause must include one of the following sets of identification criteria:

- `FULL_QUALIFIED_NAME = string_expression`
- `PARENT_PATH = string_expression AND NAME = string_expression`
- `PARENT_PATH = string_expression AND NAME LIKE string_pattern`
- `PARENT_PATH = string_expression AND NAME IN value_list`
- The `TIME_STAMP` column must *not* be specified in the `WHERE` clause of the `SOLUTION_VALUES` table but *must* be specified in the `WHERE` clause of the other tables and with specific comparison criteria.

The following columns from the **`SOLUTION_MEMBERS`** table are allowed in `ORDER BY` clauses when the table is the only table in a `SELECT` or the left-hand table in a join.

- `FULL_QUALIFIED_NAME`
- `PARENT_PATH`, `NAME`
- `PARENT_OID`, `MID`
- `TIME_STAMP`

### The `SOLUTION_VALUES` Table

This table provides access to current values of primitive members in an XHQ Solution.

#### Table Schema - Additional

In addition to the common columns for tables in this category, this table also includes the following columns.

Column Name	Description
<code>EXPIRATION_INTERVAL</code>	The expiration interval of the value.
<code>DATA_SOURCE</code>	The source of the value.
<code>UNITS</code>	The units of measure associated with the value.
<code>DESCRIPTION</code>	The description associated with the value.



For a full specification of the columns and keys, go to section, [Data Dictionary](#).

#### Core Grammar - Additional

In addition to the common Core Grammar for tables in this category, this table also adheres to the following rule:

- The `WHERE` clause must *not* include the `TIME_STAMP` column.

### The `SOLUTION_VALUES_ACTUAL` Table

This table gives access to primitive member values for a specific time. The last actual value recorded in history before the time requested is returned. If there is no appropriate value for the member and time requested, `NULL` is returned for the value and the `QUALITY` columns indicate that the quality of the value is `BAD`.

### Core Grammar - Additional

In addition to the common Core Grammar for tables in this category, this table also adheres to the following rule:

- The WHERE clause *must* include the TIME\_STAMP column and it must be compared to a specific time stamp:

```
TIME_STAMP = date_time_expression
```

### The SOLUTION\_VALUES\_AT Table

This table gives access to primitive member values for a specific time. An interpolated value from history at the time requested is returned. If there is no appropriate value for the member and time requested, NULL is returned for the value and the QUALITY columns indicate that the quality of the value is BAD.

### Core Grammar - Additional

In addition to the common Core Grammar for tables in this category, this table also adheres to the following rule:

- The WHERE clause *must* include the TIME\_STAMP column and it must be compared to a specific time stamp:

```
TIME_STAMP = date_time_expression
```

### The SOLUTION\_VALUES\_AVG Table

This table gives access to the average value for a primitive member over a specific time range. If the average cannot be calculated for the member in the time range requested, NULL is returned for the value and the QUALITY columns indicate that the quality of the value is BAD.

### Core Grammar - Additional

In addition to the common Core Grammar for tables in this category, this table also adheres to the following rule:

- The WHERE clause *must* include the TIME\_STAMP column and it must be compared to a specific time range:

```
TIME_STAMP BETWEEN date_time_expression AND date_time_expression
```

### The SOLUTION\_VALUES\_COUNT Table

This table gives access to the count of values recorded for a primitive member over a specific time range. If the count cannot be calculated for the member in the time range requested, NULL is returned for the value and the QUALITY columns indicate that the quality of the value is BAD.

### Core Grammar - Additional

In addition to the common Core Grammar for tables in this category, this table also adheres to the following rule:

- The WHERE clause *must* include the TIME\_STAMP column and it must be compared to a specific time range:

```
TIME_STAMP BETWEEN date_time_expression AND date_time_expression
```



### The SOLUTION\_VALUES\_DELTA Table

This table gives access to the delta (or change in value) recorded for primitive member over a specific time range. If there the delta cannot be calculated for the member and time range requested, NULL is returned for the value and the QUALITY columns indicate that the quality of the value is BAD.

#### Core Grammar - Additional

In addition to the common Core Grammar for tables in this category, this table also adheres to the following rule:

- The WHERE clause *must* include the TIME\_STAMP column and it must be compared to a specific time range:

```
TIME_STAMP BETWEEN date_time_expression AND date_time_expression
```

### The SOLUTION\_VALUES\_FIRST Table

This table gives access to the first recorded value for a primitive member in a specific time range. If no values can be found for the member and time range requested, NULL is returned for the value and the QUALITY columns indicate that the quality of the value is BAD.

#### Core Grammar - Additional

In addition to the common Core Grammar for tables in this category, this table also adheres to the following rule:

- The WHERE clause *must* include the TIME\_STAMP column and it must be compared to a specific time range:

```
TIME_STAMP BETWEEN date_time_expression AND date_time_expression
```

### The SOLUTION\_VALUES\_INTEGRAL Table

This table gives access to the integral of values for the primitive member over a specific time range. If the integral cannot be calculated for the member and time range requested, NULL is returned for the value and the QUALITY columns indicate that the quality of the value is BAD.

#### Core Grammar - Additional

In addition to the common Core Grammar for tables in this category, this table also adheres to the following rule:

- The WHERE clause *must* include the TIME\_STAMP column and it must be compared to a specific time range:

```
TIME_STAMP BETWEEN date_time_expression AND date_time_expression
```

### The SOLUTION\_VALUES\_LAST Table

This table gives access to the last recorded value for the primitive member in a specific time range. If no values can be found for the member and time range requested, NULL is returned for the value and the QUALITY columns indicate that the quality of the value is BAD.

#### Core Grammar - Additional

In addition to the common Core Grammar for tables in this category, this table also adheres to the following rule:

- The WHERE clause *must* include the TIME\_STAMP column and it must be compared to a specific time range:

```
TIME_STAMP BETWEEN date_time_expression AND date_time_expression
```

### The SOLUTION\_VALUES\_MAX Table

This table gives access to the maximum recorded value for the primitive member in a specific time range. If no values can be found for the member and time range requested, NULL is returned for the value and the QUALITY columns indicate that the quality of the value is BAD.

#### Core Grammar - Additional

In addition to the common Core Grammar for tables in this category, this table also adheres to the following rule:

- The WHERE clause *must* include the TIME\_STAMP column and it must be compared to a specific time range:

```
TIME_STAMP BETWEEN date_time_expression AND date_time_expression
```

### The SOLUTION\_VALUES\_MIN Table

This table gives access to the minimum recorded value for the primitive member in a specific time range. If no values can be found for the member and time range requested, NULL is returned for the value and the QUALITY columns indicate that the quality of the value is BAD.

#### Core Grammar - Additional

In addition to the common Core Grammar for tables in this category, this table also adheres to the following rule:

- The WHERE clause *must* include the TIME\_STAMP column and it must be compared to a specific time range:

```
TIME_STAMP BETWEEN date_time_expression AND date_time_expression
```

### The SOLUTION\_VALUES\_SUM Table

This table gives access to the sum of all values for the primitive member over a specific time range. If the sum cannot be calculated for the member and time range requested, NULL is returned for the value and the QUALITY columns indicate that the quality of the value is BAD.

#### Core Grammar - Additional

In addition to the common Core Grammar for tables in this category, this table also adheres to the following rule:

- The WHERE clause *must* include the TIME\_STAMP column and it must be compared to a specific time range:

```
TIME_STAMP BETWEEN date_time_expression AND date_time_expression
```

## The Solution Time-Series (Trend) Data Category

Tables in this category provide access to time-series data from primitive members. The mode of access to the member values is determined by the specific table used. The number of rows and the way in which each value is calculated is based on the time range specified. Additionally for time-sliced trends, the point count must be specified in the `WHERE` clause. The time range is divided by the point count and each interval is used to calculate a point value and returned as a separate row.

### Schema for this category

All tables in this category have the following columns in common.

Column Name	Description
FULLY_QUALIFIED_NAME	The fully-qualified path to the member.
PARENT_PATH	The full path to the member's parent.
NAME	The name of the member.
PARENT_OID	The object ID number of the member's parent.
MID	The member ID number.
DATA_TYPE	The name of the class or primitive type of the member.
VALUE_INT	The integer representation of the value.
VALUE_BOOL	The Boolean representation of the value.
VALUE_REAL	The real number (floating-point) representation of the value.
VALUE_STRING	The text representation of the value.
QUALITY_NAME	The quality of the value represented as string.
QUALITY_DESCRIPTION	The description of the quality.
QUALITY_SHORT	The OPC-encoded quality of the value.
TIME_STAMP	The time-stamp of the value.
POINT_COUNT	The target number of points in the trend.
ACTUAL_POINT_COUNT	The actual number of points in the trend.

### Core Grammar for this Category

All tables in this category have these common grammar rules.

- A trend table must be the only table in a `SELECT` statement or the left-hand table in a `JOIN` with either the `SOLUTION_MEMBERS` table or the `SOLUTION_INSTANCES` table.
- Restrictions are imposed on the `WHERE` clause of all selections from the tables. In general, the `WHERE` clause must include conditions that uniquely identify specific primitive members.

The `WHERE` clause must include one of the following sets of identification criteria:

- `FULL_QUALIFIED_NAME = string_expression`
- `PARENT_PATH = string_expression AND NAME = string_expression`
- `PARENT_PATH = string_expression AND NAME LIKE string_pattern`
- `PARENT_PATH = string_expression AND NAME IN value_list`
- The WHERE clause *must* include the `TIME_STAMP` column and it must be compared to a specific time range:  
`TIME_STAMP BETWEEN date_time_expression AND date_time_expression`
- The WHERE clause *must* include the `POINT_COUNT` column and it must be compared with a number greater than or equal to two (2) and less than or equal to 1000. The number of actual points (rows) returned is equal to or less than the number of points specified by this column.  
`POINT_COUNT = integer_expression`

The result set is always implicitly grouped by member and ordered by ascending time stamp. The following columns from the **SOLUTION\_TRENDS\*** tables may participate in the `ORDER BY` directive when the table is the only table in a `SELECT` or the left-hand table in a join.

- `FULL_QUALIFIED_NAME`
- `PARENT_PATH, NAME`
- `PARENT_OID, MID`

### The **SOLUTION\_TRENDS\_ACTUALS** Table

This table gives access to time-series data (trends) for the primitive member over a specific time range. The points returned are the actual recorded values over the specified time range up to the number of points specified. Please note that the actual number of points (rows) returned may be less than that specified depending on the historic data available in the system. Note also that the time range covered by the returned data may not cover the time range specified the total number of points in the time range is greater than that specified by the `POINT_COUNT` in the WHERE clause. The total number of points returned is limited to 1,000. If the trend cannot be obtained for the member and time range requested, no rows are returned for that member and time range.



For a full specification of the columns and keys, go to section, [Data Dictionary](#).

### The **SOLUTION\_TRENDS\_ACTUALS\_WITH\_ADJACENT** Table

This table has the same functionality as the **SOLUTION\_TRENDS\_ACTUALS** table with the additional behavior of returning a point before and/or after the requested time range, if available.

### The **SOLUTION\_TRENDS\_FITS** Table

This table gives access to time-series data (trends) for the primitive member over a specific time range. Up to 1,000 raw data values may be selected to reproduce the appearance of plotting raw data over the time range requested with the number of points specified. Please note that the actual number of points (rows) returned may be slightly less than that specified, depending on the data in the system and the algorithm used to "fit" the data to the plot. If the trend cannot be obtained for the member and time range requested, no rows are returned for that member and time range.

### *The SOLUTION\_TRENDS\_TIMESLICED\_AVG Table*

This table gives access to time-series data (trends) for the primitive member over a specific time range. The average value over each interval in the requested time range is returned as a separate row. If the trend cannot be obtained for the member and time range requested, no rows are returned for that member and time range.

### *The SOLUTION\_TRENDS\_TIMESLICED\_INTERPOLATE Table*

This table gives access to time-series data (trends) for the primitive member over a specific time range. An interpolated value in each interval over the requested time range is returned as a separate row. If the trend cannot be obtained for the member and time range requested, no rows are returned for that member and time range.

#### **Table Schema - Additional**

In addition to the common columns for tables in this category, this table also includes the following column.

Column Name	Description
INTERPOLATION_TYPE	The interpolation type of all the items in the time-series data (LINEAR or STEP).

### *The SOLUTION\_TRENDS\_TIMESLICED\_MAX Table*

This table gives access to time-series data (trends) for the primitive member over a specific time range. The maximum value in each interval over the requested time range is returned as a separate row. If the trend cannot be obtained for the member and time range requested, no rows are returned for that member and time range.

### *The SOLUTION\_TRENDS\_TIMESLICED\_MIN Table*

This table gives access to time-series data (trends) for the primitive member over a specific time range. The minimum value in each interval over the requested time range is returned as a separate row. If the trend cannot be obtained for the member and time range requested, no rows are returned for that member and time range.

## The Model Metadata Category

Tables in this category provide access to metadata for the model.

### The *MODEL\_CLASSES* Table

This table gives details on classes in the XHQ Model. Classes (also called components) are defined in the model to be used as objects, tags, or global collections in the solution. Each class has an entry in the **MODEL\_CLASSES** table that describes each in detail.

The main purpose of this table is to be joined with the **MODEL\_MEMBERS**, **MODEL\_VIEWS** or the **MODEL\_CLASS\_ROLES** table when class information is required in the returned result set.

#### Table Schema

The **MODEL\_CLASSES** table has the following columns.

Column Name	Description
CLASS_ID	The ID of the class.
REVISION_ID	The revision ID of the class.
NAME	The name of the class.
SUPER_CLASS_ID	The ID of the class' super class.



For a full specification of the columns and keys, go to section, [Data Dictionary](#).

#### Core Grammar

The **MODEL\_CLASSES** table may participate as a **right-hand table in a join** with the following tables:

- **MODEL\_MEMBERS**

The **JOIN** clause must be of the format (where *m* is the alias for the other table):

```
JOIN MODEL_MEMBERS i ON m.CLASS_ID = i.CLASS_ID[AND m.CLASS_REVISION_ID = i.REVISION_ID]
```

- **MODEL\_VIEWS**

The **JOIN** clause must be of the format (where *m* is the alias for the other table):

```
JOIN MODEL_VIEWS i ON m.CLASS_ID = i.CLASS_ID
```

- **MODEL\_CLASS\_ROLES**

The **JOIN** clause must be of the format (where *m* is the alias for the other table):

```
JOIN MODEL_CLASS_ROLES i ON m.CLASS_ID = i.CLASS_ID[AND m.CLASS_REVISION_ID = i.REVISION_ID]
```

The **MODEL\_CLASSES** table may be the sole **left-hand table in a query** and queried in the following ways.

- Find a specific class by name:

```
SELECT {non_aggregate_list} FROM MODEL_CLASSES WHERE NAME = class_name_string[AND logical_expression]*
```

- Find a specific class by ID:

```
SELECT {non_aggregate_list}FROM MODEL_CLASSESWHERE CLASS_ID = integer_expression[AND logical_expression]*
```

The `REVISION_ID` column may be used in the `WHERE` clause. The columns from the `MODEL_CLASSES` table may not participate in any `ORDER BY` directive.

## The `MODEL_CLASS_ROLES` Table

This table gives shows which roles are associated with which classes in the XHQ Model.

### Table Schema

The `MODEL_CLASS_ROLES` table has the following columns.

Column Name	Description
<code>CLASS_ID</code>	The ID of the class.
<code>CLASS_REVISION_ID</code>	The revision ID of the class.
<code>ROLE_ID</code>	The ID of the role.



For a full specification of the columns and keys, go to section, [Data Dictionary](#).

### Core Grammar

The `MODEL_CLASS_ROLES` table may be the sole **left-hand table in a query** and queried in the following way.

- Find specific role IDs of a class or the count of these (with possible filtering):

```
SELECT {non_aggregate_list}FROM MODEL_CLASS_ROLESWHERE CLASS_ID = integer_expression
[AND m.CLASS_REVISION_ID = i.REVISION_ID] [AND logical_expression]*
```

The `MODEL_CLASS_ROLES` table may be the **left-hand table in a join** with the `MODEL_CLASSES` and the `MODEL_ROLES` tables. The table may be queried in the following way:

```
SELECT {non_aggregate_list}FROM MODEL_CLASS_ROLES lJOIN MODEL_CLASSES c    ON l.CLASS_ID =
c.CLASS_ID    [AND l.CLASS_REVISION_ID = c.REVISION_ID]JOIN MODEL_ROLES r    ON l.ROLE_ID =
r.ROLE_ID [where_clause] [order_by_clause]
```

The following columns from the `MODEL_CLASS_ROLES` table may participate in any `ORDER BY` directive when the table is the only table in a `SELECT` or the left-hand table in a join.

- `ROLE_ID`
- `CLASS_ID`, `CLASS_REVISION_ID`

## The `MODEL_MEMBERS` Table

This table gives details on members in the XHQ Model.

## Table Schema

The **MODEL\_MEMBERS** table has the following columns.

Column Name	Description
CLASS_ID	The ID of the class to which the member belongs.
CLASS_REVISION_ID	The revision ID of the class.
MID	The member ID number.
NAME	The name of the member.
DESCRIPTION	The description of the member.
MEMBER_TYPE	The type of member (PRIMITIVE, OBJECT, or COLLECTION).
DATA_TYPE	The name of the class or primitive type of the member.
ATTRIBUTE_TYPE	The attribute of the member (only used when MEMBER_TYPE is PRIMITIVE).
RESTRICTED	This indicates whether or not access to the member is restricted.
HIDDEN	This indicates whether or not the member is hidden.



For a full specification of the columns and keys, go to section, [Data Dictionary](#).

## Core Grammar

The **MODEL\_MEMBERS** table may be the sole **left-hand table in a query** and queried in the following way.

- Find specific members of a class or the count of these (with possible filtering):

```
SELECT {non_aggregate_list} FROM MODEL_MEMBERS WHERE CLASS_ID = integer_expression [AND logical_expression] *
```

The **MODEL\_MEMBERS** table may be the **left-hand table in a join** with the **MODEL\_CLASSES** table. The table may be queried in the following way:

```
SELECT {non_aggregate_list} FROM MODEL_MEMBERS m JOIN MODEL_CLASSES c ON m.CLASS_ID = c.CLASS_ID [where_clause] [order_by_clause]
```

The following columns from the **MODEL\_MEMBERS** table may participate in any ORDER BY directive when the table is the only table in a SELECT or the left-hand table in a join.

- NAME
- CLASS\_ID, MID



## The MODEL\_ROLES Table

This table gives details on roles in the XHQ Model. Each role has an entry in the **MODEL\_ROLES** table that describes each in detail. The main purpose of this table is to be joined with the **MODEL\_CLASS\_ROLES** when class information is required in the returned result set.

### Table Schema

The **MODEL\_ROLES** table has the following columns.

Column Name	Description
ROLE_ID	The ID of the role.
NAME	The name of the role.
DESCRIPTION	The description of the role.
OS_GROUP	The operating system group associated with the role.
IS_LOCAL	This indicates whether or not the group is locally defined.



For a full specification of the columns and keys, go to section, [Data Dictionary](#).

### Core Grammar

The **MODEL\_ROLES** table may participate as a **right-hand table in a join** with the following table:

- **MODEL\_CLASS\_ROLES**

The **JOIN** clause must be of the format (where *m* is the alias for the other table):

```
JOIN MODEL_CLASS_ROLES r ON m.ROLE_ID = r.ROLE_ID
```

The **MODEL\_ROLES** table may be the sole **left-hand table in a query** and queried in the following ways.

- Find all roles:

```
SELECT {non_aggregate_list} FROM MODEL_ROLES
```

The columns from the **MODEL\_ROLES** table may not participate in any **ORDER BY** directive.

## The MODEL\_ROLE\_PERMISSIONS Table

This table shows which permissions are associated with which roles in the XHQ Model.

### Table Schema

The **MODEL\_ROLE\_PERMISSIONS** table has the following columns.

Column Name	Description
ROLE_ID	The ID of the role.
NAME	The name of the permission.



For a full specification of the columns and keys, go to section, [Data Dictionary](#).

### Core Grammar

The **MODEL\_ROLE\_PERMISSIONS** table may be the sole left-hand table in a query and queried in the following way.

- Find specific permissions of a role or the count of these (with possible filtering):

```
SELECT {non_aggregate_list}FROM MODEL_ROLE_PERMISSIONS WHERE ROLE_ID = integer_expression[AND logical_expression]*
```

The **MODEL\_ROLE\_PERMISSIONS** table may be the **left-hand table in a join** with the **MODEL\_ROLES** table. The table may be queried in the following way:

```
SELECT {non_aggregate_list}FROM MODEL_ROLE_PERMISSIONS pJOIN MODEL_ROLES r ON p.ROLE_ID = r.ROLE_ID [where_clause] [order_by_clause]
```

The following columns from the **MODEL\_ROLE\_PERMISSIONS** table may participate in any ORDER BY directive when the table is the only table in a SELECT or the left-hand table in a join.

- ROLE\_ID
- NAME

### The MODEL\_VIEWS Table

This table gives details on views in the XHQ Model.

### Table Schema

The **MODEL\_VIEWS** table has the following columns.

Column Name	Description
CLASS_ID	The ID of the class to which the view belongs.
VIEW_ID	The view ID number.
NAME	The name of the view.
DESCRIPTION	The description of the view.



For a full specification of the columns and keys, go to section, [Data Dictionary](#).

### Core Grammar

The **MODEL\_VIEWS** table may be the sole left-hand table in a query and queried in the following way.

- Find specific views of a class or the count of these (with possible filtering):

```
SELECT {non_aggregate_list}FROM MODEL_VIEWS WHERE CLASS_ID = integer_expression[AND logical_expression]*
```

The **MODEL\_VIEWS** table may be the **left-hand table in a join** with the **MODEL\_CLASSES** table. The table may be queried in the following way:

```
SELECT {non_aggregate_list}FROM MODEL_VIEWS vJOIN MODEL_CLASSES c      ON v.CLASS_ID =
c.CLASS_ID[where_clause] [order_by_clause]
```

The following columns from the **MODEL\_VIEWS** table may participate in any ORDER BY directive when the table is the only table in a SELECT or the left-hand table in a join.

- NAME
- CLASS\_ID, MID

## The MODEL\_VIEW\_PROPERTIES Table

This table shows which properties are associated with which views in the XHQ Model.

### Table Schema

The **MODEL\_VIEW\_PERMISSIONS** table has the following column.

Column Name	Description
CLASS_ID	The ID of the class to which the member belongs.
VIEW_ID	The view ID number.
NAME	The name of the property.
DESCRIPTION	The description of the property.



For a full specification of the columns and keys, go to section, [Data Dictionary](#).

### Core Grammar

The **MODEL\_VIEW\_PROPERTIES** table may be the sole **left-hand table in a query** and queried in the following way.

- Find specific properties of a view or the count of these (with possible filtering):

```
SELECT {non_aggregate_list}FROM MODEL_VIEW_PROPERTIESWHERE CLASS_ID = integer_
expression[AND l.VIEW_ID = integer_expression] [AND logical_expression]*
```

The **MODEL\_VIEW\_PROPERTIES** table may be the **left-hand table in a join** with the **MODEL\_VIEWS** table. The table may be queried in the following way:

```
SELECT {non_aggregate_list}FROM MODEL_VIEW_PROPERTIES pJOIN MODEL_VIEWS v      ON p.VIEW_ID
= v.VIEW_ID [where_clause] [order_by_clause]
```

The following columns from the **MODEL\_VIEW\_PROPERTIES** table may participate in any ORDER BY directive when the table is the only table in a SELECT or the left-hand table in a join.

- CLASS\_ID, VIEW\_ID
- NAME

## The Tag Metadata Category

Tables in this category provide access to metadata for the tag collection.

### The TAG\_COLLECTION Table

This table is an alias for the actual tag collection table. It allows you to easily retrieve solution-independent tag metadata without knowing the name of the actual tag collection.

#### Table Schema

The **TAG\_COLLECTION** table has the following column.

Column Name	Description
KEY	The name of the tag.
UNITS	The unit of measurement of the tag value.
CONFIGURATION	The configuration for the tag.
DESCRIPTION	The description of the tag.



For a full specification of the columns and keys, go to section, [Data Dictionary](#).

## 2 | XHQ OPC UA Server

### Server Initialization



For pre-requisites and installation instructions, refer to the [XHQ Installation Guide](#).

### How to Configure the Server

#### Port Assignments

By default, the XHQ OPC UA Server uses port **62540 for HTTP** connection and port **62541 for TCP** connection. If other applications use either of these ports, change the port assignments to avoid conflicts.

#### To change the port assignments

1. Open the **XhqServer.Config.xml** file.



Typically, this configuration file is found in the `\Program Files\XHQ\XHQ OPC UA Server` directory.

2. Edit the URLs for the **http** and **tcp** protocols.

```
<ServerConfiguration>
<BaseAddresses>
  <ua:String>http://localhost:62540/Siemens/XHQ-OPC-UA-Server</ua:String>
  <ua:String>opc.tcp://localhost:62541/Siemens/XHQ-OPC-UA-Server</ua:String>
</BaseAddresses>
</ServerConfiguration>
```

3. **Save** the **.xml** file.

#### Firewall Configuration



The firewall must be configured to allow the chosen ports to receive connections from the OPC UA Clients.

For more information, see the topic, [How to Configure the Server](#).

## License Requirement

The XHQ OPC UA Server requires special licensing for XHQ and is not generally available. If you are interested in obtaining a license, please contact your Siemens XHQ sales representative.



The XHQ Solution Server license must permit the use of the XHQ OPC UA Server. If it is not permitted, the OPC Server logs the following warning:

OPC Server not authorized in license check.

## About the `xhq_opcuaserver.exe.config` file

To configure the XHQ OPC UA Server, change the **<appSettings>** tags on the `xhq_opcuaserver.exe.config` file.



If you perform a server re-install without erasing or moving the old application configuration file, the old file is left intact.



Other configurations in the `xhq_opcuaserver.exe.config` file are related to third-party libraries used by the application (such as OPC Foundation and log4net). Do not change these settings except in the cases specified in this guide.

## To edit the **<appSettings>** tag

1. From the `xhq_opcuaserver.exe.config` file, locate the **<appSettings>** tag.

Note the following example.

```
<appSettings>
<add key="XHQAddress" value="localhost" />
<add key="TagFolderName" value="XHQ Tags" />
<add key="SolutionFolderName" value="XHQ Objects" />
<add key="UOMEnabled" value="True" />
<add key="WaitCreatingTags" value="True" />
<add key="WatchdogTimeFrame" value="60000" />
<add key="ReconnectTimeFrame" value="60000" />
<add key="ReconnectAttempts" value="10" />
<add key="UpdateRateForTags" value="60" />
<add key="ProxyServerSubjectName" value="UA COM Proxy Server" />
</appSettings>
```

2. Change the `value` attribute to any of the following settings.

Setting	Description
<b>ProxyServerSubjectName</b>	<p>Change the <code>value</code> attribute to the client certificate subject name of a Proxy Server that may be used for legacy OPC connections.</p> <p>The default value is <b>UA COM Proxy Server</b> (which is the subject name of the OPC Foundation Proxy Server).</p>
<b>ReconnectAttempts</b>	<p>Change the <code>value</code> attribute to the number of reconnect attempts before quitting.</p> <p>The default value is <b>10</b>.</p> <p>The minimum value is 1 and the maximum value is 100000.</p>
<b>ReconnectTimeFrame</b>	<p>Change the <code>value</code> attribute to the number of <b>milliseconds</b> between two reconnection attempts with the XHQ Server.</p> <p>The default value is <b>60000</b>.</p> <p>The minimum value is 30000 and the maximum value is 600000.</p>
<b>SolutionFolderName</b>	<p>Change the value attribute to the name of the folder that holds the <b>Solution Members</b>.</p> <p>The default value is <b>XHQ Objects</b>.</p>
<b>TagFolderName</b>	<p>Change the <code>value</code> attribute to the name of the folder that holds the <b>Tag Collection</b>.</p> <p>The default value is <b>XHQ Tags</b>.</p>

Setting	Description
<b>UOMEnabled</b>	<p>Set the <code>value</code> attribute to <b>True</b> to enable the UoM feature, or <b>False</b> to disable it.</p> <p>The default value is <b>True</b>.</p>
<b>UpdateRateForTags</b>	<p>Change the <code>value</code> attribute to the number of <b>minutes</b> that the Tag collection is updated.</p> <p>The default value is <b>60</b>.</p> <p>The minimum value is 15 and the maximum value is 60000.</p>
<b>WaitCreatingTags</b>	<p>If the attribute is set to <b>True</b>, then the OPC Server waits for the XHQ Tags to be loaded before accepting any connections by OPC Client(s). Should a client attempt to connect to the OPC Server while it is still loading the XHQ Tags, a <code>BadServerHalted</code> error message is returned.</p> <p><b>Note:</b> The OPC Server will only respond when <u>all</u> Tags have been read and loaded. Depending on the number of Tags, this may take a considerable amount of time.</p> <p>If the attribute is set to <b>False</b>, the <code>BadServerHalted</code> error message appears until the OPC Server establishes a connection with the XHQ Solution Server. The XHQ Tagsloading runs concurrently in the background. So, should a client attempt to connect to the OPC Server before all the XHQ Tags are retrieved (loaded), a <code>BadWaitingForInitialData</code> error message is returned. After all the Tags are loaded, use the <b>Browse</b> command from the Client to show the elements.</p> <p>The default value is <b>True</b>.</p>
<b>WatchdogTimeFrame</b>	<p>Change the <code>value</code> attribute to the number of <b>milliseconds</b> that is used by the Watchdog, which checks the connection between the XHQ Server and the XHQ OPC UA Server if there is no activity.</p> <p>The default value is <b>60000</b>.</p> <p>The minimum value is 30000 and the maximum value is 600000.</p>
<b>XHQAddress</b>	<p>Change the <code>value</code> attribute to the IP address, or hostname, of the XHQ Server that is running the OPC Server. If it is in the same machine, use <b>localhost</b>.</p> <p>The default value is <b>localhost</b>.</p>



## To edit the <log4net> tag

The OPC UA Server generates two log files: one for logging errors and the second for information about the Server activities.

1. From the **xhq\_opcuaserver.exe.config** file, locate the **<log4net> tag**.

Note the **highlighted text** in the following example.

```
<log4net>
<root>
  <level value="INFO"/>
  <appender-ref ref="xhqopcuaserver"/>
  <appender-ref ref="xhqopcuaserverlog"/>
</root>
<appender name="xhqopcuaserver"
  type="XhqOpcServer.XhqOpcServerFileAppender">
  <file type="log4net.Util.PatternString"
    value="%env{XHQ_SERVER_LOGS}\\xhqopcuaserver.out"/>
  </appender>
<appender name="xhqopcuaserverlog"
  type="XhqOpcServer.XhqOpcServerFileAppender">
  <file type="log4net.Util.PatternString"
    value="%env{XHQ_SERVER_LOGS}\\xhqopcuaserverlog.out"/>
  </appender>
</log4net>
```

2. Change the **value** attribute to the following settings.

Setting	Description
<b>File</b>	<p>Change the <b>value</b> attribute to the desired path and name for the log file. Use double backslashes ("\\") to separate directories and the key <code>%env{&lt;name of environment variable&gt;}</code> if an environment variable is used. If this variable does not exist, the log is placed in the <b>log</b> folder of the installation path.</p> <p>The default value is <code>%env{XHQ_SERVER_LOGS}\\xhqopcuaserverlog.out</code>.</p>
<b>Level</b>	<p>Change the <b>value</b> attribute to increase or decrease the log level. The valid values are: ALL, DEBUG, INFO, WARN, ERROR, FATAL, and OFF.</p> <p>The default value is <b>INFO</b>.</p>

## Security

Security for the XHQ OPC UA Server is based on data encryption, authentication, and certificates. To properly configure the Server security, the following tasks must be executed.



### If XHQ security is enabled...

- For the XHQ Server, the **Active Directory** security type must be enabled. The Classic security type is not supported.
- The XHQ OPC UA Server respects user roles and security constraints set in the XHQ Workbench and the XHQ Solution Builder with regards to reading, browsing, and writing access.
- The XHQ OPC UA Server does not allow cross domain connections. In this case, the OPC UA Server and all OPC UA Clients must be in the same domain.
- The XHQ OPC UA Server only allows connections from clients that have proper credentials. Anonymous access is not allowed.
- The XHQ OPC UA Server must be executed by a user with full access to the XHQ model and solution. If not, the application may not work properly.
- The OPC UA Clients may pass credentials with NTLM or Kerberos. For more information, see the topic, *Special Security Configurations*.



See the section, *Security, Access, and Privileges*, located in the XHQ Administrator's Guide, for more information on XHQ security.

### About the self-signed server certificate

The application automatically creates a self-signed certificate, which can be used to run the OPC Server. However, the OPC Clients must manually trust the OPC Server (see the topic, *To configure the client certificate, for more information*). The self-signed certificate is not issued by a certification authority. In order to connect to the server, the client must trust the certificate.

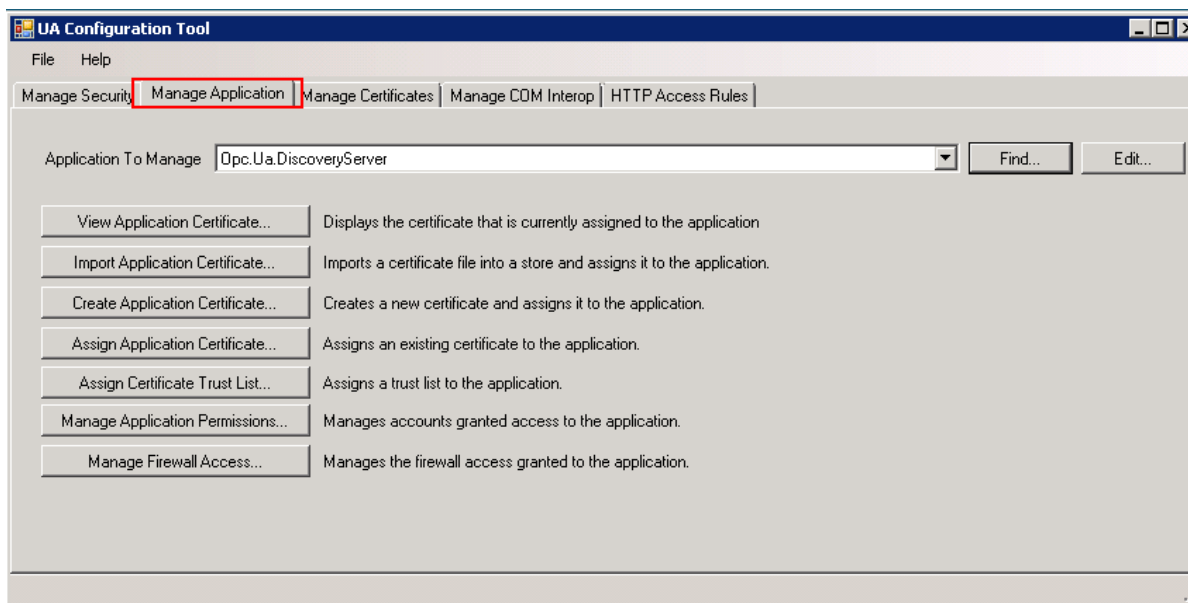
### To configure the server certificate issued by a Certificate Authority

1. Request a certificate from the **Certificate Authority** (for example, VeriSign) for the XHQ OPC UA **Server**. Private and Public Keys are generated with extensions **.pfx** and **.der**, respectively.

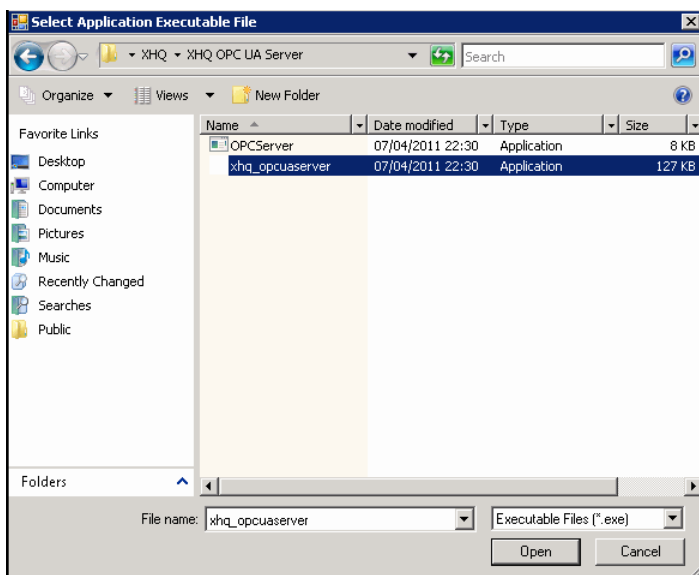


The certificate DNS name is the name of the machine that hosts the XHQ OPC UA Server.

2. From the server machine, open the **UA Configuration Tool**.
3. Click the **Manage Application** tab.



4. Click **Find**.  
The "Select Application Executable File" dialog appears.

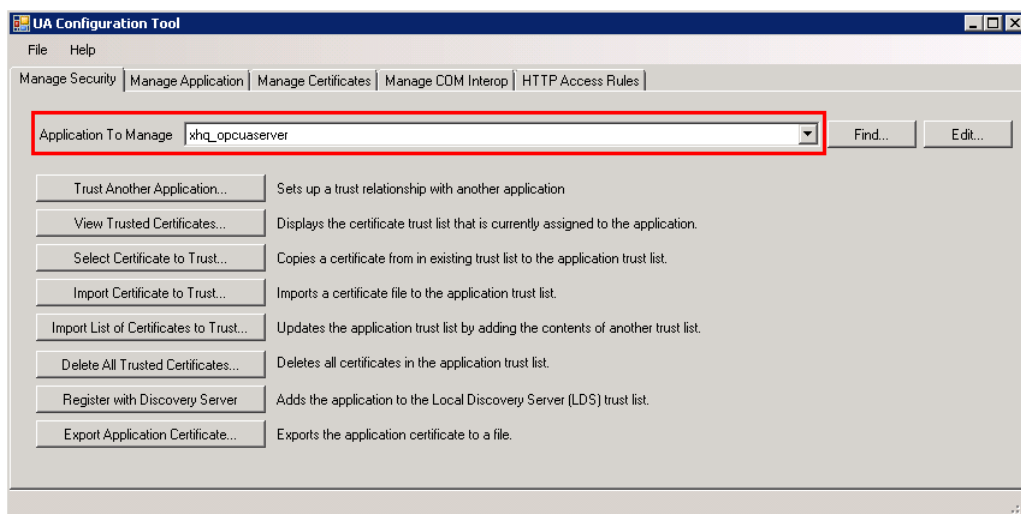


5. Locate and select the **xhq\_opcuaserver.exe** file, which is the XHQ OPC UA Server executable file.
6. Click **Open**.  
The "Modify Application Information" dialog box appears.
7. For **Configuration File**, click **Browse**.
8. Select the **XhqServer.Config.xml** file and click **OK**.  
This returns you to the "UA Configuration Tool" > "Manage Application" tab.
9. Click **Import Application Certificate**.
10. Select the **Private Key file** (the **.pfx** file) generated in step 1.
11. Click **OK**.

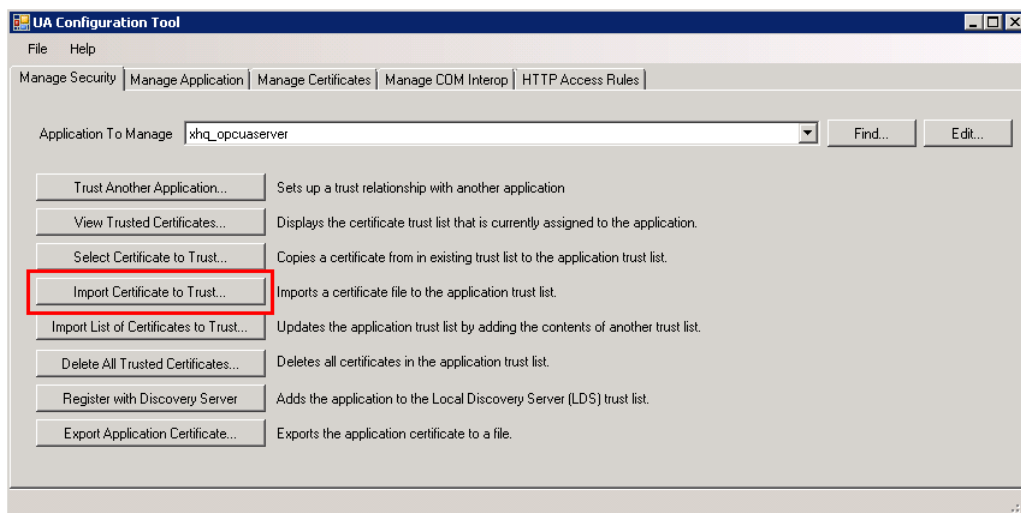
## To configure the client certificate

For each client, complete the following procedure.

1. Obtain the **OPC Client Public Key** with the **.der** extension.
2. From the server machine, open the **UA Configuration Tool**.
3. From the **Manage Security tab**, next to the **Application to Manage** box, click the **down arrow**.



4. In the drop-down list, select **xhq\_opcuaserver**.
5. Click **Import Certificate to Trust**.

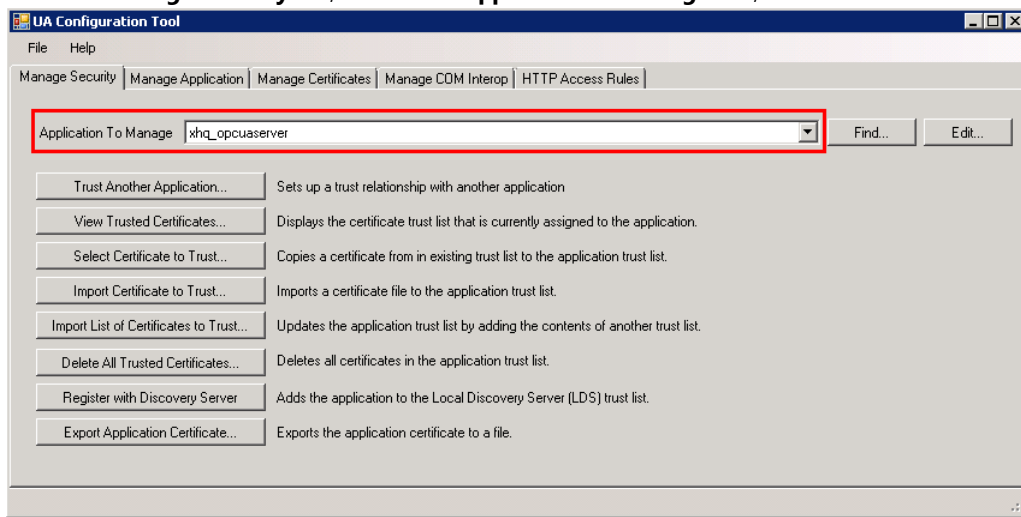


6. Select the client **Public Key file** (the **.der** file) generated in step 1.
7. Click **OK**.

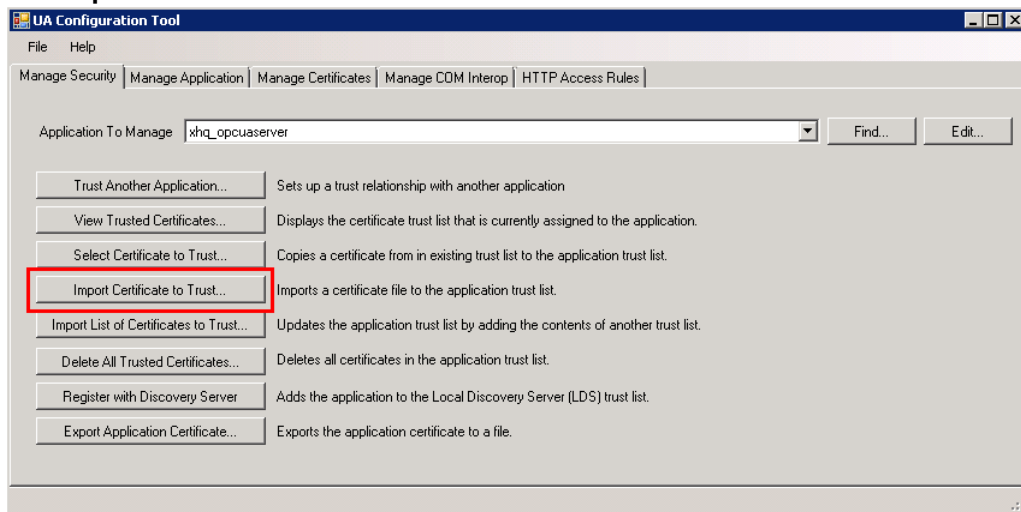
## To trust a Certificate Authority

This process is needed to avoid manually placing the client's certificates inside the Trusted Peer Certificates folder. By doing this task, the XHQ OPC UA Server automatically trusts all certificates issued by a Certificate Authority.

1. Obtain the **Certificate Authority Public Key** (the **.der** file) and the **Certificate Revocation List** (the **.cr1** file).
2. From the **server**, launch the **UA Configuration Tool**.
3. From the **Manage Security** tab, next to the **Application to Manage** box, click the **down arrow**.



4. In the drop-down list, select **xhq\_opcuaserver**.
5. Click **Import Certificate to Trust**.



6. Select the **Certificate Authority Public Key** (the **.der** file).
7. Click **OK**.
8. Open the **Trusted Peer Certificate** directory.
9. Do one of the following:
  - If the **.cr1** folder **exists** within the Trusted Peer Certificate directory, then **continue to step 10**.
  - Else, **create a new folder** under the Trusted Peer Certificate store directory and name it **cr1**.
10. Place the **Certificate Revocation List** (the **.cr1** file) inside the **cr1** folder.



If the Certificate Authority revokes or un-revokes any of its certificates, the **.cr1** file should be replaced by the **most recent** file. The server then rejects connections from clients that are using revoked certificates.

## To change the default client certificate library

Change the path defined by the **StoreType** and **StorePath** tags in the **XhqServer.Config.xml** file (which is located in the OPC UA Server installation folder).

*Example* From this:

```
<TrustedPeerCertificates>
  <StoreType>Windows</StoreType>
  <StorePath>LocalMachine\UA Applications</StorePath>
</TrustedPeerCertificates>
```

To this:

```
<TrustedPeerCertificates>
  <StoreType>Directory</StoreType>
  <StorePath>D:\YourPath\XHQOPCUAServerTrustedListFolder
</StorePath>
</TrustedPeerCertificates>
```



If a given client certificate is not in the client certificate store of the XHQ OPC UA Server, connections from this client is not allowed.

## Special Security Configurations

The XHQ OPC UA Server accepts either NTLM or Kerberos credentials. NTLM authentication does not require additional configuration; however, Kerberos authentication requires the following security configuration.

## To use Kerberos authentication

1. From a command prompt, **create an SPN** (Service Principal Name) for the XHQ OPC UA Server by using the **setspn.exe** utility, replacing [OPC-SERVER-HOSTNAME] and [OPC-SERVER-USERNAME] with the OPC UA Server hostname and username, respectively.

*Example:* `setspn -s [OPC-SERVER-HOSTNAME]/xhq_opcuaserver [OPC-SERVER-USERNAME]`

2. In the **XhqServer.Config.xml** file (which is located in the OPC UA Server installation folder), locate **IssuerEndpointUrl** and replace OPC-SERVER-HOSTNAME with the OPC UA Server hostname.

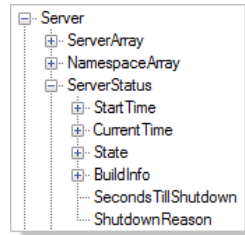
*Example:*

```
<ua:UserTokenPolicy>
  <ua:TokenType>IssuedToken_3
</ua:TokenType>
  <ua:IssuedTokenType>http://docs.oasis-open.org/wss/oasis-wss-kerberos-
token-profile-1.1
</ua:IssuedTokenType>
  <ua:IssuerEndpointUrl>OPC-SERVER-HOSTNAME/xhq_opcuaserver
</ua:IssuerEndpointUrl>
</ua:UserTokenPolicy>
```

## Reconnectivity

The XHQ OPC UA Server is prepared to handle connection problems with OPC Clients and the XHQ Solution Server. To track the connection status, monitor the following Tags:

- SecondsTillShutdown
- ShutdownReason



## Getting Started

### To start the application

1. Check the `xhq_opcuaserver.exe.config` file to see if the XHQ Server is **running**.
2. Open the **Windows Command Prompt** (CMD) as **Administrator**.
3. Run the following change directory (cd) command:

```
cd server_path
```

Where `server_path` is the path to the XHQ OPC UA Server.

4. To start the application, type **OPCServer startup**.  
This starts the `xhq_opcauserver.exe` process, which creates the OPC Tree.



The server becomes available for connection as soon as it retrieves all the Tags from the XHQ Solution Server. If a client tries to connect with the XHQ OPC UA Server before the server becomes available, it received a `BadServerHalted` return code.

To alter this behavior, edit the `WaitCreatingTags` setting (see the topic, [To edit the appSettings tag](#)).



If Windows UAC (User Access Control) is enabled, run the XHQ OPC UA Server with **administrative privileges** in order to correctly generate logging files and connect to XHQ.

5. To shutdown the server, execute **OPCServer shutdown**.

### Important Things to Note

- To configure an auto-start using `xhqboot`, go to the topic, [Modifying xhqboot](#).
- If not integrated into `xhqboot`, then the OPC Server must be restarted after a repos swap. (This is automatic with `xhqboot` due to the XHQ Server start-up/shutdown procedure.)



## XHQ OPC UA Server Features

For demonstrations purposes, the features are shown using the OPC Foundation UA Clients. If a feature is the same for both clients, DA and HDA, the feature example will use only the first client.

### Browser – DA and HDA Clients

The OPC Foundation UA Client browses the OPC Tree in the same way as any other Classic OPC Client does. In this section, is shown how the XHQ Solution Tree is mapped to the OPC Tree.



The DA/HDA Proxy software has no "on demand" capability; therefore, it will take a long time to build XHQ Tags for a large repository.

### Default Members

There are three components at the first level of the OPC Tree.

#### Server

- This component is always created by the OPC UA Server to expose some information about the server.
- It only appears in UA Clients.

#### XHQ Objects

- The mapped XHQ Solution Server into the OPC Tree.
- Maps all XHQ Member Groups, Groups, Member Objects, Objects, Primitive Member and Tag Objects in the XHQ Solution Tree.
- Only XHQ Tag Objects and Primitive Members not inside a Tag Object have a special mapping ([see the topic, Tag Object Mapping](#)).
- This folder name can be changed in the `xhq_opcuaserver.exe.config` file ([see the topic, Server Initialization](#)).

#### XHQ Tags

- Maps all members of the Tag Collection that have a valid Alias (not Null).
- This folder name can be changed in the `xhq_opcuaserver.exe.config` file ([see the topic, Server Initialization](#)).

### Primitive Member Mapping

A **Primitive Member** is mapped to the OPC Tree in the same way as in the XHQ Solution Tree. The difference is that it has a child with a name of **Value** and has the capability to be monitored and/or written to by an OPC Client.

## Tag Object Mapping

In order for a **Member Object** to be considered as a **Tag Object**, it must:

- Have one configured member with an attribute key;
- Have one member with an attribute value;
- Be inside the Tag Collection.

Primitive Members of a Tag Object do not appear directly in the OPC Tree. The Primitive Member marked as **Value** and has an alias configured is represented by a child of the Tag Object named `Value`. The remaining XHQ primitive members are mapped as OPC properties and are appended under the **Value** member.

The Tag Object uses the value of the Primitive Member marked as **Key** as its name.



When mapping an OPC UA node to a decimal value in XHQ, its data type is `Double`. This can result in the loss of precision.

## Units of Measure (UoM)

The UoM feature depends on the following:

- The `xhq_opcuaserver.exe.config` file must have the `UOMEnabled` key marked as `True`.
- The XHQ version must have UoM capability.
- The **Member Object** must have at least one member marked as **Value** in the XHQ Workbench. Only these members can be converted with UoM.
- The **Member Object** must have a member marked as **Units** in the XHQ Workbench that corresponds to the default Tag's unit. If the Member Object is a Tag Object, the same rule applies.
- The unit must exist in the **XHQ UoM configuration file**, and its **Sort Order** must be greater than zero.
- Only the **Primitive Members** of a numeric data type marked as **Value** in the XHQ Workbench are valid for UoM. The valid data types are: Integer, Long, Real, Double, and Decimal.
- The OPC UA protocol does not accept the Decimal data type. This type is currently mapped to Double.

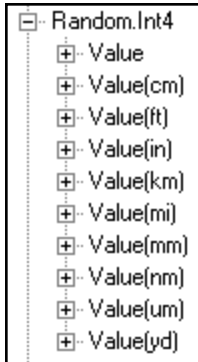


Precision errors of Decimal/Double conversions may occur in this case.

- The server does not support writing to the UoM nodes when the default value is an Integer or a Long. In this case, UoM WRITE is only possible to the base UoM unit.



The XHQ Server only records the default value. Non-default/default conversions can yield inconsistencies caused by rounding values to Integer.



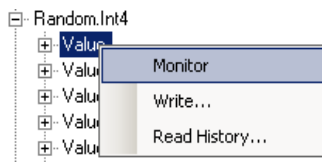
With UoM, the **Value** member and its properties are replicated for each unit that is in the same group. A READ or WRITE operation has the values converted.

### Important Things to Note

- If the conversion results in a data type overflow at one member, its value is with the maximum/minimum value of the corresponding data type, and its quality is **BadOutOfRange**.
- If a WRITE operation receives a value that is greater/less than the maximum/minimum value of the data type, it is not written to XHQ and the OPC Server sends a **BadOutOfRange** error message to the client.
- A legacy OPC Client may read, write, or monitor UoM members if the **NodeId** is known. A BROWSE operation shows only the "Value" member and does not show the UoM siblings.

## READ – DA Client

When a **Value** member is monitored, the XHQ Server will subscribe to the matching back-end of the Primitive Member. It starts to update and provide the values for the OPC Client as a VTQ.



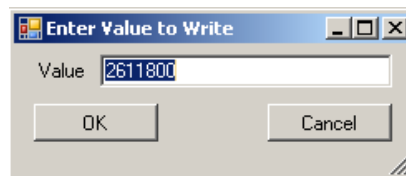
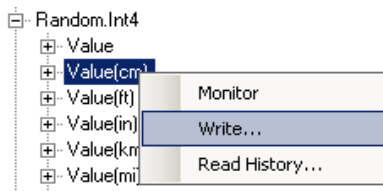
Value	Quality	Timestamp
26118	Good	07:28:24.000

### Important Things to Note

- DateTime and Timestamp information is in the UTC format in the OPC UA standard. However, some OPC UA clients may show this information converted to the local time zone.
- If a proxy component link is linked to another component in the Solution Builder during the OPC Server execution time when its value is being monitored, you will need to re-browse, cancel monitoring, and initiate a new monitoring in order to fetch current data. Otherwise, quality remains as **BadDataUnavailable**.

## WRITE – DA Client

When a **Value** member is written, the matching XHQ Primitive Member is changed. The value may be stored, if it is configured for it.

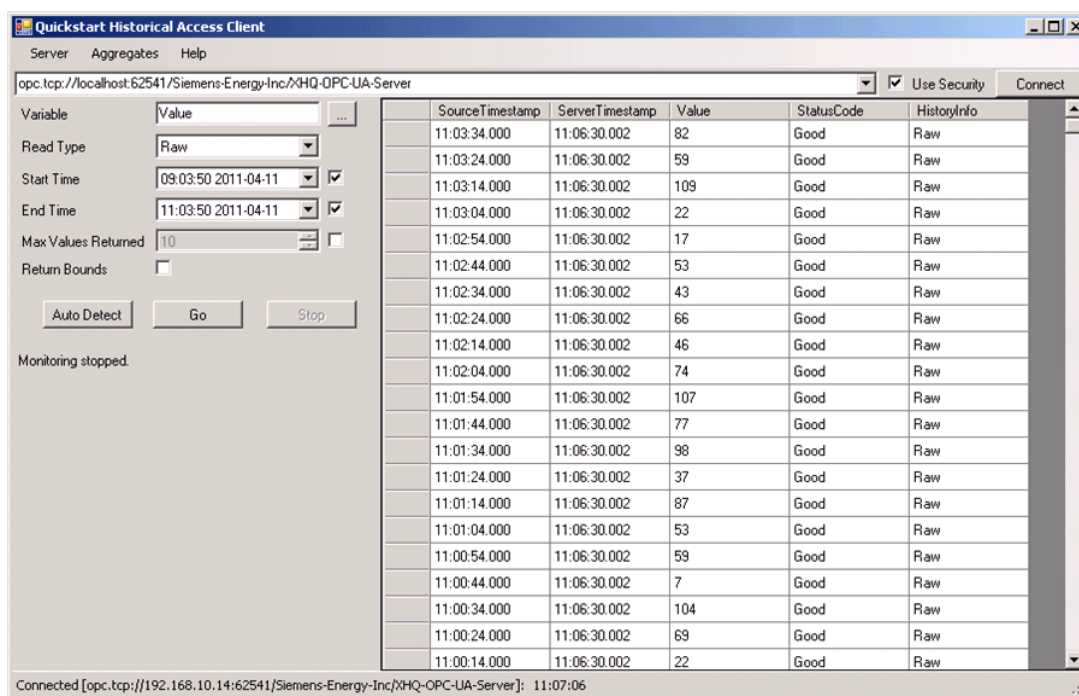


The XHQ Primitive Member must have the **Allow User Writes** attribute marked. Re-monitor the tag so changes take effect.

## READ – HDA Client

To read the historical information of an XHQ Primitive Member with the HDA Client, the following are needed:

- The variable of the OPC Tree
- The READ type
- The Start and End times



*Historical Access Client Dialog*

# Appendices

## Section Contents

---

- A - [Data Dictionary](#)
- B - [Crystal Reports Visual Studio Integration](#)
- C - [Connecting the OPC Client](#)
- D - [Modifying xhqboot](#)

## A - Data Dictionary

### SOLUTION\_INSTANCES

#### SOLUTION\_INSTANCES

This table contains all information about objects, global collections and tags in the XHQ Solution Server namespace.

#### Constraints

Primary Key: OID

Unique: FULLY\_QUALIFIED\_NAME

Column Name	Nulls?	Type	Description
OID	NO	INTEGER	The instance ID number.
NAME	NO	NVARCHAR()	The name of the instance.
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully qualified name of the instance.
REVISION_ID	NO	INTEGER	The revision ID of the instance.
DESCRIPTION	YES	NVARCHAR()	A description of the instance.
CLASS_ID	NO	INTEGER	The class ID of the instance.
CLASS_REVISION_ID	NO	INTEGER	The revision ID of the class.
CLASS_NAME	NO	NVARCHAR()	The class name of the instance.
INSTANCE_TYPE	NO	NVARCHAR()	The type of instance (OBJECT, COLLECTION).
TAG	NO	BOOLEAN	True if the instance type is COLLECTION (it is the TAG collection).

## SOLUTION\_MEMBERS

### SOLUTION\_MEMBERS

This table contains all information about members of instances in the XHQ Solution Server namespace.

#### Constraints

Primary Key: OID  
 Foreign Key: PARENT\_OID->SOLUTION\_INSTANCES.OID  
 Foreign Key: REFERENCE\_OID->SOLUTION\_INSTANCES.OID  
 Unique:

Column Name	Nulls?	Type	Description
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully-qualified path to the member.
PARENT_PATH	YES	NVARCHAR()	The full path to the member's parent.
NAME	NO	NVARCHAR()	The name of the member.
PARENT_OID	NO	INTEGER	The object ID number of the member's parent.
MID	NO	SMALLINT	The member ID number.
DESCRIPTION	YES	NVARCHAR()	A description of the member.
MEMBER_TYPE	NO	NVARCHAR()	The type of member (PRIMITIVE, OBJECT, COLLECTION).
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
REFERENCE_OID	YES	INTEGER	The object ID of the referenced object if MEMBER_TYPE is OBJECT.
ALIAS_NAME	YES	NVARCHAR()	The alias of the member (only used when MEMBER_TYPE is PRIMITIVE).
ATTRIBUTE_TYPE	YES	NVARCHAR()	The attribute of the member (only used MEMBER_TYPE is PRIMITIVE).
RESTRICTED	NO	BOOLEAN	Indicates whether or not access to the member restricted.
HIDDEN	NO	BOOLEAN	Indicates whether or not the member is hidden.
PROXY	NO	BOOLEAN	Indicates whether or not the member is a proxy to another object (only used MEMBER_TYPE is OBJECT).

## SOLUTION\_VALUES

### SOLUTION\_VALUES

This table contains current values from the XHQ Solution server.

#### Constraints

Primary Key: FULLY\_QUALIFIED\_NAME

Unique:

Foreign Key: FULLY\_QUALIFIED\_NAME  
->SOLUTION\_MEMBERS.FULLY\_QUALIFIED\_NAME

Foreign Key: PARENT\_OID->SOLUTION\_INSTANCES.OID

Column Name	Nulls?	Type	Description
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully-qualified path to the member.
PARENT_PATH	YES	NVARCHAR()	The full path to the member's parent.
NAME	NO	NVARCHAR()	The name of the member.
PARENT_OID	NO	INTEGER	The object ID number of the member's parent.
MID	NO	SMALLINT	The member ID number.
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
VALUE_INT	YES	INTEGER	The integer representation of the value.
VALUE_BOOL	YES	BOOLEAN	The Boolean representation of the value.
VALUE_REAL	YES	REAL	The real number (floating-point) representation of the value.
VALUE_STRING	YES	NVARCHAR()	The text representation of the value.
QUALITY_NAME	NO	NVARCHAR()	The quality of the value represented as string.
QUALITY_DESCRIPTION	NO	NVARCHAR()	A description of the quality.
QUALITY_SHORT	NO	SMALLINT	The OPC-encoded quality of the value.
DATA_SOURCE	NO	NVARCHAR()	The source of the value.
TIME_STAMP	YES	DATETIME	The time stamp of the value.
EXPIRATION_INTERVAL	YES	INTERVAL	The expiration interval of the value.
UNITS	YES	NVARCHAR()	The units of measure associated with the value.
DESCRIPTION	YES	NVARCHAR()	The description associated with the value.



## SOLUTION\_VALUES\_ACTUAL

### SOLUTION\_VALUES\_ACTUAL

This table contains values from the XHQ Solution server most closely corresponding to the time specified in the WHERE clause.

#### Constraints

Primary Key: FULLY\_QUALIFIED\_NAME

Unique:

Foreign Key: FULLY\_QUALIFIED\_NAME  
->SOLUTION\_MEMBERS.FULLY\_QUALIFIED\_NAME

Foreign Key: PARENT\_OID->SOLUTION\_INSTANCES.OID

Column Name	Nulls?	Type	Description
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully-qualified path to the member.
PARENT_PATH	YES	NVARCHAR()	The full path to the member's parent.
NAME	NO	NVARCHAR()	The name of the member.
PARENT_OID	NO	INTEGER	The object ID number of the member's parent.
MID	NO	SMALLINT	The member ID number.
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
VALUE_INT	YES	INTEGER	The integer representation of the value.
VALUE_BOOL	YES	BOOLEAN	The Boolean representation of the value.
VALUE_REAL	YES	REAL	The real number (floating-point) representation of the value.
VALUE_STRING	YES	NVARCHAR()	The text representation of the value.
QUALITY_NAME	NO	NVARCHAR()	The quality of the value represented as string.
QUALITY_DESCRIPTION	NO	NVARCHAR()	A description of the quality.
QUALITY_SHORT	NO	SMALLINT	The OPC-encoded quality of the value.
DATA_SOURCE	NO	NVARCHAR()	The source of the value.
TIME_STAMP	YES	DATETIME	The time stamp of the value.
EXPIRATION_INTERVAL	YES	INTERVAL	The expiration interval of the value.
UNITS	YES	NVARCHAR()	The units of measure associated with the value.
DESCRIPTION	YES	NVARCHAR()	The description associated with the value.

## SOLUTION\_VALUES\_AT

### SOLUTION\_VALUES\_AT

This table contains values from the XHQ Solution server interpolated at the time specified in the WHERE clause.

#### Constraints

Primary Key: FULLY\_QUALIFIED\_NAME

Unique:

Foreign Key: FULLY\_QUALIFIED\_NAME  
->SOLUTION\_MEMBERS.FULLY\_QUALIFIED\_NAME

Foreign Key: PARENT\_OID->SOLUTION\_INSTANCES.OID

Column Name	Nulls?	Type	Description
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully-qualified path to the member.
PARENT_PATH	YES	NVARCHAR()	The full path to the member's parent.
NAME	NO	NVARCHAR()	The name of the member.
PARENT_OID	NO	INTEGER	The object ID number of the member's parent.
MID	NO	SMALLINT	The member ID number.
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
VALUE_INT	YES	INTEGER	The integer representation of the value.
VALUE_BOOL	YES	BOOLEAN	The Boolean representation of the value.
VALUE_REAL	YES	REAL	The real number (floating-point) representation of the value.
VALUE_STRING	YES	NVARCHAR()	The text representation of the value.
QUALITY_NAME	NO	NVARCHAR()	The quality of the value represented as string.
QUALITY_DESCRIPTION	NO	NVARCHAR()	A description of the quality.
QUALITY_SHORT	NO	SMALLINT	The OPC-encoded quality of the value.
DATA_SOURCE	NO	NVARCHAR()	The source of the value.
TIME_STAMP	YES	DATETIME	The time stamp of the value.
EXPIRATION_INTERVAL	YES	INTERVAL	The expiration interval of the value.
UNITS	YES	NVARCHAR()	The units of measure associated with the value.
DESCRIPTION	YES	NVARCHAR()	The description associated with the value.

## SOLUTION\_VALUES\_AVG

### SOLUTION\_VALUES\_AVG

This table contains average values from the XHQ Solution Server.

#### Constraints

Primary Key: FULLY\_QUALIFIED\_NAME

Unique:

Foreign Key: FULLY\_QUALIFIED\_NAME  
->SOLUTION\_MEMBERS.FULLY\_QUALIFIED\_NAME

Foreign Key: PARENT\_OID->SOLUTION\_INSTANCES.OID

Column Name	Nulls?	Type	Description
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully-qualified path to the member.
PARENT_PATH	YES	NVARCHAR()	The full path to the member's parent.
NAME	NO	NVARCHAR()	The name of the member.
PARENT_OID	NO	INTEGER	The object ID number of the member's parent.
MID	NO	SMALLINT	The member ID number.
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
VALUE_INT	YES	INTEGER	The integer representation of the value.
VALUE_BOOL	YES	BOOLEAN	The Boolean representation of the value.
VALUE_REAL	YES	REAL	The real number (floating-point) representation of the value.
VALUE_STRING	YES	NVARCHAR()	The text representation of the value.
QUALITY_NAME	NO	NVARCHAR()	The quality of the value represented as string.
QUALITY_DESCRIPTION	NO	NVARCHAR()	A description of the quality.
QUALITY_SHORT	NO	SMALLINT	The OPC-encoded quality of the value.
TIME_STAMP	YES	DATETIME	The time stamp of the value.
DATA_SOURCE	NO	NVARCHAR()	The source of the value.
EXPIRATION_INTERVAL	YES	INTERVAL	The expiration interval of the value.
UNITS	YES	NVARCHAR()	The units of measure associated with the value.
DESCRIPTION	YES	NVARCHAR()	The description associated with the value.

## SOLUTION\_VALUES\_COUNT

### SOLUTION\_VALUES\_COUNT

This table contains counts from the XHQ Solution Server.

#### Constraints

Primary Key: FULLY\_QUALIFIED\_NAME

Unique:

Foreign Key: FULLY\_QUALIFIED\_NAME  
->SOLUTION\_MEMBERS.FULLY\_QUALIFIED\_NAME

Foreign Key: PARENT\_OID->SOLUTION\_INSTANCES.OID

Column Name	Nulls?	Type	Description
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully-qualified path to the member.
PARENT_PATH	YES	NVARCHAR()	The full path to the member's parent.
NAME	NO	NVARCHAR()	The name of the member.
PARENT_OID	NO	INTEGER	The object ID number of the member's parent.
MID	NO	SMALLINT	The member ID number.
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
VALUE_INT	YES	INTEGER	The integer representation of the value.
VALUE_BOOL	YES	BOOLEAN	The Boolean representation of the value.
VALUE_REAL	YES	REAL	The real number (floating-point) representation of the value.
VALUE_STRING	YES	NVARCHAR()	The text representation of the value
QUALITY_NAME	NO	NVARCHAR()	The quality of the value represented as string.
QUALITY_DESCRIPTION	NO	NVARCHAR()	A description of the quality.
QUALITY_SHORT	NO	SMALLINT	The OPC-encoded quality of the value.
TIME_STAMP	YES	DATETIME	The time stamp of the value.
DATA_SOURCE	NO	NVARCHAR()	The source of the value.
EXPIRATION_INTERVAL	YES	INTERVAL	The expiration interval of the value.
UNITS	YES	NVARCHAR()	The units of measure associated with the value.
DESCRIPTION	YES	NVARCHAR()	The description associated with the value.

## SOLUTION\_VALUES\_DELTA

### SOLUTION\_VALUES\_DELTA

This table contains changes in values over time from the XHQ Solution Server.

#### Constraints

Primary Key: FULLY\_QUALIFIED\_NAME

Unique:

Foreign Key: FULLY\_QUALIFIED\_NAME  
->SOLUTION\_MEMBERS.FULLY\_QUALIFIED\_NAME

Foreign Key: PARENT\_OID->SOLUTION\_INSTANCES.OID

Column Name	Nulls?	Type	Description
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully-qualified path to the member.
PARENT_PATH	YES	NVARCHAR()	The full path to the member's parent.
NAME	NO	NVARCHAR()	The name of the member.
PARENT_OID	NO	INTEGER	The object ID number of the member's parent.
MID	NO	SMALLINT	The member ID number.
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
VALUE_INT	YES	INTEGER	The integer representation of the value.
VALUE_BOOL	YES	BOOLEAN	The Boolean representation of the value.
VALUE_REAL	YES	REAL	The real number (floating-point) representation of the value.
VALUE_STRING	YES	NVARCHAR()	The text representation of the value.
QUALITY_NAME	NO	NVARCHAR()	The quality of the value represented as string.
QUALITY_DESCRIPTION	NO	NVARCHAR()	A description of the quality.
QUALITY_SHORT	NO	SMALLINT	The OPC-encoded quality of the value.
TIME_STAMP	YES	DATETIME	The time stamp of the value.
DATA_SOURCE	NO	NVARCHAR()	The source of the value.
EXPIRATION_INTERVAL	YES	INTERVAL	The expiration interval of the value.
UNITS	YES	NVARCHAR()	The units of measure associated with the value.
DESCRIPTION	YES	NVARCHAR()	The description associated with the value.

## SOLUTION\_VALUES\_FIRST

### SOLUTION\_VALUES\_FIRST

This table contains the first value in the time span from the XHQ Solution Server.

#### Constraints

Primary Key: FULLY\_QUALIFIED\_NAME

Unique:

Foreign Key: FULLY\_QUALIFIED\_NAME  
->SOLUTION\_MEMBERS.FULLY\_QUALIFIED\_NAME

Foreign Key: PARENT\_OID->SOLUTION\_INSTANCES.OID

Column Name	Nulls?	Type	Description
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully-qualified path to the member.
PARENT_PATH	YES	NVARCHAR()	The full path to the member's parent.
NAME	NO	NVARCHAR()	The name of the member.
PARENT_OID	NO	INTEGER	The object ID number of the member's parent.
MID	NO	SMALLINT	The member ID number.
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
VALUE_INT	YES	INTEGER	The integer representation of the value.
VALUE_BOOL	YES	BOOLEAN	The Boolean representation of the value.
VALUE_REAL	YES	REAL	The real number (floating-point) representation of the value.
VALUE_STRING	YES	NVARCHAR()	The text representation of the value.
QUALITY_NAME	NO	NVARCHAR()	The quality of the value represented as string.
QUALITY_DESCRIPTION	NO	NVARCHAR()	A description of the quality.
QUALITY_SHORT	NO	SMALLINT	The OPC-encoded quality of the value.
TIME_STAMP	YES	DATETIME	The time stamp of the value.
DATA_SOURCE	NO	NVARCHAR()	The source of the value.
EXPIRATION_INTERVAL	YES	INTERVAL	The expiration interval of the value.
UNITS	YES	NVARCHAR()	The units of measure associated with the value.
DESCRIPTION	YES	NVARCHAR()	The description associated with the value.

## SOLUTION\_VALUES\_INTEGRAL

### SOLUTION\_VALUES\_INTEGRAL

This table contains integral value in the time span from the XHQ Solution Server.

#### Constraints

Primary Key: FULLY\_QUALIFIED\_NAME

Unique:

Foreign Key: FULLY\_QUALIFIED\_NAME  
->SOLUTION\_MEMBERS.FULLY\_QUALIFIED\_NAME

Foreign Key: PARENT\_OID->SOLUTION\_INSTANCES.OID

Column Name	Nulls?	Type	Description
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully-qualified path to the member.
PARENT_PATH	YES	NVARCHAR()	The full path to the member's parent.
NAME	NO	NVARCHAR()	The name of the member.
PARENT_OID	NO	INTEGER	The object ID number of the member's parent.
MID	NO	SMALLINT	The member ID number.
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
VALUE_INT	YES	INTEGER	The integer representation of the value.
VALUE_BOOL	YES	BOOLEAN	The Boolean representation of the value.
VALUE_REAL	YES	REAL	The real number (floating-point) representation of the value.
VALUE_STRING	YES	NVARCHAR()	The text representation of the value.
QUALITY_NAME	NO	NVARCHAR()	The quality of the value represented as string.
QUALITY_DESCRIPTION	NO	NVARCHAR()	A description of the quality.
QUALITY_SHORT	NO	SMALLINT	The OPC-encoded quality of the value.
TIME_STAMP	YES	DATETIME	The time stamp of the value.
DATA_SOURCE	NO	NVARCHAR()	The source of the value.
EXPIRATION_INTERVAL	YES	INTERVAL	The expiration interval of the value.
UNITS	YES	NVARCHAR()	The units of measure associated with the value.
DESCRIPTION	YES	NVARCHAR()	The description associated with the value.

## SOLUTION\_VALUES\_LAST

### SOLUTION\_VALUES\_LAST

This table contains last value in the time span from the XHQ Solution Server.

#### Constraints

Primary Key: FULLY\_QUALIFIED\_NAME

Unique:

Foreign Key: FULLY\_QUALIFIED\_NAME  
->SOLUTION\_MEMBERS.FULLY\_QUALIFIED\_NAME

Foreign Key: PARENT\_OID->SOLUTION\_INSTANCES.OID

Column Name	Nulls?	Type	Description
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully-qualified path to the member.
PARENT_PATH	YES	NVARCHAR()	The full path to the member's parent.
NAME	NO	NVARCHAR()	The name of the member.
PARENT_OID	NO	INTEGER	The object ID number of the member's parent.
MID	NO	SMALLINT	The member ID number.
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
VALUE_INT	YES	INTEGER	The integer representation of the value.
VALUE_BOOL	YES	BOOLEAN	The Boolean representation of the value.
VALUE_REAL	YES	REAL	The real number (floating-point) representation of the value.
VALUE_STRING	YES	NVARCHAR()	The text representation of the value.
QUALITY_NAME	NO	NVARCHAR()	The quality of the value represented as string.
QUALITY_DESCRIPTION	NO	NVARCHAR()	A description of the quality.
QUALITY_SHORT	NO	SMALLINT	The OPC-encoded quality of the value.
TIME_STAMP	YES	DATETIME	The time stamp of the value.
DATA_SOURCE	NO	NVARCHAR()	The source of the value.
EXPIRATION_INTERVAL	YES	INTERVAL	The expiration interval of the value.
UNITS	YES	NVARCHAR()	The units of measure associated with the value.
DESCRIPTION	YES	NVARCHAR()	The description associated with the value.



## SOLUTION\_VALUES\_MAX

### SOLUTION\_VALUES\_MAX

This table contains the maximum values from the XHQ Solution Server.

#### Constraints

Primary Key:	FULLY_QUALIFIED_NAME
Unique:	
Foreign Key:	FULLY_QUALIFIED_NAME ->SOLUTION_MEMBERS.FULLY_QUALIFIED_NAME
Foreign Key:	PARENT_OID->SOLUTION_INSTANCES.OID

Column Name	Nulls?	Type	Description
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully-qualified path to the member.
PARENT_PATH	YES	NVARCHAR()	The full path to the member's parent.
NAME	NO	NVARCHAR()	The name of the member.
PARENT_OID	NO	INTEGER	The object ID number of the member's parent.
MID	NO	SMALLINT	The member ID number.
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
VALUE_INT	YES	INTEGER	The integer representation of the value.
VALUE_BOOL	YES	BOOLEAN	The Boolean representation of the value.
VALUE_REAL	YES	REAL	The real number (floating-point) representation of the value.
VALUE_STRING	YES	NVARCHAR()	The text representation of the value.
QUALITY_NAME	NO	NVARCHAR()	The quality of the value represented as string.
QUALITY_DESCRIPTION	NO	NVARCHAR()	A description of the quality.
QUALITY_SHORT	NO	SMALLINT	The OPC-encoded quality of the value.
TIME_STAMP	YES	DATETIME	The time stamp of the value.
DATA_SOURCE	NO	NVARCHAR()	The source of the value.
EXPIRATION_INTERVAL	YES	INTERVAL	The expiration interval of the value.
UNITS	YES	NVARCHAR()	The units of measure associated with the value.
DESCRIPTION	YES	NVARCHAR()	The description associated with the value.

## SOLUTION\_VALUES\_MIN

### SOLUTION\_VALUES\_MIN

This table contains the minimum values from the XHQ Solution Server.

#### Constraints

Primary Key: FULLY\_QUALIFIED\_NAME

Unique:

Foreign Key: FULLY\_QUALIFIED\_NAME  
->SOLUTION\_MEMBERS.FULLY\_QUALIFIED\_NAME

Foreign Key: PARENT\_OID->SOLUTION\_INSTANCES.OID

Column Name	Nulls?	Type	Description
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully-qualified path to the member.
PARENT_PATH	YES	NVARCHAR()	The full path to the member's parent.
NAME	NO	NVARCHAR()	The name of the member.
PARENT_OID	NO	INTEGER	The object ID number of the member's parent.
MID	NO	SMALLINT	The member ID number.
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
VALUE_INT	YES	INTEGER	The integer representation of the value.
VALUE_BOOL	YES	BOOLEAN	The Boolean representation of the value.
VALUE_REAL	YES	REAL	The real number (floating-point) representation of the value.
VALUE_STRING	YES	NVARCHAR()	The text representation of the value.
QUALITY_NAME	NO	NVARCHAR()	The quality of the value represented as string.
QUALITY_DESCRIPTION	NO	NVARCHAR()	A description of the quality.
QUALITY_SHORT	NO	SMALLINT	The OPC-encoded quality of the value.
TIME_STAMP	YES	DATETIME	The time stamp of the value.
DATA_SOURCE	NO	NVARCHAR()	The source of the value.
EXPIRATION_INTERVAL	YES	INTERVAL	The expiration interval of the value.
UNITS	YES	NVARCHAR()	The units of measure associated with the value.
DESCRIPTION	YES	NVARCHAR()	The description associated with the value.

## SOLUTION\_VALUES\_SUM

### SOLUTION\_VALUES\_SUM

This table contains the sum of the values from the XHQ Solution Server.

#### Constraints

Primary Key: FULLY\_QUALIFIED\_NAME

Unique:

Foreign Key: FULLY\_QUALIFIED\_NAME  
->SOLUTION\_MEMBERS.FULLY\_QUALIFIED\_NAME

Foreign Key: PARENT\_OID->SOLUTION\_INSTANCES.OID

Column Name	Nulls?	Type	Description
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully-qualified path to the member.
PARENT_PATH	YES	NVARCHAR()	The full path to the member's parent.
NAME	NO	NVARCHAR()	The name of the member.
PARENT_OID	NO	INTEGER	The object ID number of the member's parent.
MID	NO	SMALLINT	The member ID number.
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
VALUE_INT	YES	INTEGER	The integer representation of the value.
VALUE_BOOL	YES	BOOLEAN	The Boolean representation of the value.
VALUE_REAL	YES	REAL	The real number (floating-point) representation of the value.
VALUE_STRING	YES	NVARCHAR()	The text representation of the value.
QUALITY_NAME	NO	NVARCHAR()	The quality of the value represented as string.
QUALITY_DESCRIPTION	NO	NVARCHAR()	A description of the quality.
QUALITY_SHORT	NO	SMALLINT	The OPC-encoded quality of the value.
TIME_STAMP	YES	DATETIME	The time stamp of the value.
DATA_SOURCE	NO	NVARCHAR()	The source of the value.
EXPIRATION_INTERVAL	YES	INTERVAL	The expiration interval of the value.
UNITS	YES	NVARCHAR()	The units of measure associated with the value.
DESCRIPTION	YES	NVARCHAR()	The description associated with the value.

## SOLUTION\_TRENDS\_ACTUALS

### SOLUTION\_TRENDS\_ACTUALS

This table contains the actual time-series values from the XHQ Solution Server.

#### Constraints

Primary Key: FULLY\_QUALIFIED\_NAME, TIME\_STAMP

Unique:

Foreign Key: FULLY\_QUALIFIED\_NAME  
->SOLUTION\_MEMBERS.FULLY\_QUALIFIED\_NAME

Foreign Key: PARENT\_OID->SOLUTION\_INSTANCES.OID

Column Name	Nulls?	Type	Description
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully-qualified path to the member.
PARENT_PATH	YES	NVARCHAR()	The full path to the member's parent.
NAME	NO	NVARCHAR()	The name of the member.
PARENT_OID	NO	INTEGER	The object ID number of the member's parent.
MID	NO	SMALLINT	The member ID number.
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
VALUE_INT	YES	INTEGER	The integer representation of the value.
VALUE_BOOL	YES	BOOLEAN	The Boolean representation of the value.
VALUE_REAL	YES	REAL	The real number (floating-point) representation of the value.
VALUE_STRING	YES	NVARCHAR()	The text representation of the value.
QUALITY_NAME	NO	NVARCHAR()	The quality of the value represented as string.
QUALITY_DESCRIPTION	NO	NVARCHAR()	A description of the quality.
QUALITY_SHORT	NO	SMALLINT	The OPC-encoded quality of the value.
TIME_STAMP	NO	DATETIME	The time stamp of the value.
POINT_COUNT	NO	INTEGER	The target number of points in the trend.
ACTUAL_POINT_COUNT	NO	INTEGER	The actual number of points in the trend.

## SOLUTION\_TRENDS\_ACTUALS\_WITH\_ADJACENT

### SOLUTION\_TRENDS\_ACTUALS\_WITH\_ADJACENT

This table contains the actual time-series values plus the values immediately before and after the time span, if available from the XHQ Solution Server.

#### Constraints

Primary Key: FULLY\_QUALIFIED\_NAME, TIME\_STAMP

Unique:

Foreign Key: FULLY\_QUALIFIED\_NAME  
->SOLUTION\_MEMBERS.FULLY\_QUALIFIED\_NAME

Foreign Key: PARENT\_OID->SOLUTION\_INSTANCES.OID

Column Name	Nulls?	Type	Description
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully-qualified path to the member.
PARENT_PATH	YES	NVARCHAR()	The full path to the member's parent.
NAME	NO	NVARCHAR()	The name of the member.
PARENT_OID	NO	INTEGER	The object ID number of the member's parent.
MID	NO	SMALLINT	The member ID number.
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
VALUE_INT	YES	INTEGER	The integer representation of the value.
VALUE_BOOL	YES	BOOLEAN	The Boolean representation of the value.
VALUE_REAL	YES	REAL	The real number (floating-point) representation of the value.
VALUE_STRING	YES	NVARCHAR()	The text representation of the value.
QUALITY_NAME	NO	NVARCHAR()	The quality of the value represented as string.
QUALITY_DESCRIPTION	NO	NVARCHAR()	A description of the quality.
QUALITY_SHORT	NO	SMALLINT	The OPC-encoded quality of the value.
POINT_COUNT	NO	INTEGER	The target number of points in the trend.
ACTUAL_POINT_COUNT	NO	INTEGER	The actual number of points in the trend.
TIME_STAMP	NO	DATETIME	The time stamp of the value.

## SOLUTION\_TRENDS\_FITS

### SOLUTION\_TRENDS\_FITS

This table contains the time-series values appropriate for a visual plot.

#### Constraints

Primary Key: FULLY\_QUALIFIED\_NAME, TIME\_STAMP

Unique:

Foreign Key: FULLY\_QUALIFIED\_NAME  
->SOLUTION\_MEMBERS.FULLY\_QUALIFIED\_NAME

Foreign Key: PARENT\_OID->SOLUTION\_INSTANCES.OID

Column Name	Nulls?	Type	Description
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully-qualified path to the member.
PARENT_PATH	YES	NVARCHAR()	The full path to the member's parent.
NAME	NO	NVARCHAR()	The name of the member.
PARENT_OID	NO	INTEGER	The object ID number of the member's parent.
MID	NO	SMALLINT	The member ID number.
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
VALUE_INT	YES	INTEGER	The integer representation of the value.
VALUE_BOOL	YES	BOOLEAN	The Boolean representation of the value.
VALUE_REAL	YES	REAL	The real number (floating-point) representation of the value.
VALUE_STRING	YES	NVARCHAR()	The text representation of the value.
QUALITY_NAME	NO	NVARCHAR()	The quality of the value represented as string.
QUALITY_DESCRIPTION	NO	NVARCHAR()	A description of the quality.
QUALITY_SHORT	NO	SMALLINT	The OPC-encoded quality of the value.
TIME_STAMP	NO	DATETIME	The time stamp of the value.
POINT_COUNT	NO	INTEGER	The target number of points in the trend.
ACTUAL_POINT_COUNT	NO	INTEGER	The actual number of points in the trend.

## SOLUTION\_TRENDS\_TIMESLICED\_AVG

### SOLUTION\_TRENDS\_TIMESLICED\_AVG

This table contains the average values for each time slice over the time span.

#### Constraints

Primary Key: FULLY\_QUALIFIED\_NAME, TIME\_STAMP

Unique:

Foreign Key: FULLY\_QUALIFIED\_NAME  
->SOLUTION\_MEMBERS.FULLY\_QUALIFIED\_NAME

Foreign Key: PARENT\_OID->SOLUTION\_INSTANCES.OID

Column Name	Nulls?	Type	Description
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully-qualified path to the member.
PARENT_PATH	YES	NVARCHAR()	The full path to the member's parent.
NAME	NO	NVARCHAR()	The name of the member.
PARENT_OID	NO	INTEGER	The object ID number of the member's parent.
MID	NO	SMALLINT	The member ID number.
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
VALUE_INT	YES	INTEGER	The integer representation of the value.
VALUE_BOOL	YES	BOOLEAN	The Boolean representation of the value.
VALUE_REAL	YES	REAL	The real number (floating-point) representation of the value.
VALUE_STRING	YES	NVARCHAR()	The text representation of the value.
QUALITY_NAME	NO	NVARCHAR()	The quality of the value represented as string.
QUALITY_DESCRIPTION	NO	NVARCHAR()	A description of the quality.
QUALITY_SHORT	NO	SMALLINT	The OPC-encoded quality of the value.
TIME_STAMP	NO	DATETIME	The time stamp of the value.
POINT_COUNT	NO	INTEGER	The target number of points in the trend.
ACTUAL_POINT_COUNT	NO	INTEGER	The actual number of points in the trend.

## SOLUTION\_TRENDS\_TIMESLICED\_INTERPOLATE

### SOLUTION\_TRENDS\_TIMESLICED\_INTERPOLATE

This table contains the interpolation of values at each time slice over the time span.

#### Constraints

Primary Key: FULLY\_QUALIFIED\_NAME, TIME\_STAMP

Unique:

Foreign Key: FULLY\_QUALIFIED\_NAME  
->SOLUTION\_MEMBERS.FULLY\_QUALIFIED\_NAME

Foreign Key: PARENT\_OID->SOLUTION\_INSTANCES.OID

Column Name	Nulls?	Type	Description
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully-qualified path to the member.
PARENT_PATH	YES	NVARCHAR()	The full path to the member's parent.
NAME	NO	NVARCHAR()	The name of the member.
PARENT_OID	NO	INTEGER	The object ID number of the member's parent.
MID	NO	SMALLINT	The member ID number.
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
VALUE_INT	YES	INTEGER	The integer representation of the value.
VALUE_BOOL	YES	BOOLEAN	The Boolean representation of the value.
VALUE_REAL	YES	REAL	The real number (floating-point) representation of the value.
VALUE_STRING	YES	NVARCHAR()	The text representation of the value.
QUALITY_NAME	NO	NVARCHAR()	The quality of the value represented as string.
QUALITY_DESCRIPTION	NO	NVARCHAR()	A description of the quality.
QUALITY_SHORT	NO	SMALLINT	The OPC-encoded quality of the value.
INTERPOLATION_TYPE	YES	NVARCHAR()	The interpolation type of all the items in the time-series data. (LINEAR or STEP).
TIME_STAMP	NO	DATETIME	The time stamp of the value.
POINT_COUNT	NO	INTEGER	The target number of points in the trend.
ACTUAL_POINT_COUNT	NO	INTEGER	The actual number of points in the trend.



## SOLUTION\_TRENDS\_TIMESLICED\_MAX

### SOLUTION\_TRENDS\_TIMESLICED\_MAX

This table contains the maximum values in each time slice over the time span.

#### Constraints

Primary Key: FULLY\_QUALIFIED\_NAME, TIME\_STAMP

Unique:

Foreign Key: FULLY\_QUALIFIED\_NAME  
->SOLUTION\_MEMBERS.FULLY\_QUALIFIED\_NAME

Foreign Key: PARENT\_OID->SOLUTION\_INSTANCES.OID

Column Name	Nulls?	Type	Description
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully-qualified path to the member.
PARENT_PATH	YES	NVARCHAR()	The full path to the member's parent.
NAME	NO	NVARCHAR()	The name of the member.
PARENT_OID	NO	INTEGER	The object ID number of the member's parent.
MID	NO	SMALLINT	The member ID number
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
VALUE_INT	YES	INTEGER	The integer representation of the value.
VALUE_BOOL	YES	BOOLEAN	The Boolean representation of the value.
VALUE_REAL	YES	REAL	The real number (floating-point) representation of the value.
VALUE_STRING	YES	NVARCHAR()	The text representation of the value.
QUALITY_NAME	NO	NVARCHAR()	The quality of the value represented as string.
QUALITY_DESCRIPTION	NO	NVARCHAR()	A description of the quality.
QUALITY_SHORT	NO	SMALLINT	The OPC-encoded quality of the value.
TIME_STAMP	NO	DATETIME	The time stamp of the value.
POINT_COUNT	NO	INTEGER	The target number of points in the trend.
ACTUAL_POINT_COUNT	NO	INTEGER	The actual number of points in the trend.

## SOLUTION\_TRENDS\_TIMESLICED\_MIN

### SOLUTION\_TRENDS\_TIMESLICED\_MIN

This table contains the minimum values in each time slice over the time span.

#### Constraints

Primary Key: FULLY\_QUALIFIED\_NAME, TIME\_STAMP

Unique:

Foreign Key: FULLY\_QUALIFIED\_NAME  
->SOLUTION\_MEMBERS.FULLY\_QUALIFIED\_NAME

Foreign Key: PARENT\_OID->SOLUTION\_INSTANCES.OID

Column Name	Nulls	Type	Description
FULLY_QUALIFIED_NAME	NO	NVARCHAR()	The fully-qualified path to the member.
PARENT_PATH	YES	NVARCHAR()	The full path to the member's parent.
NAME	NO	NVARCHAR()	The name of the member.
PARENT_OID	NO	INTEGER	The object ID number of the member's parent.
MID	NO	SMALLINT	The member ID number.
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
VALUE_INT	YES	INTEGER	The integer representation of the value.
VALUE_BOOL	YES	BOOLEAN	The Boolean representation of the value.
VALUE_REAL	YES	REAL	The real number (floating-point) representation of the value.
VALUE_STRING	YES	NVARCHAR()	The text representation of the value.
QUALITY_NAME	NO	NVARCHAR()	The quality of the value represented as string.
QUALITY_DESCRIPTION	NO	NVARCHAR()	A description of the quality.
QUALITY_SHORT	NO	SMALLINT	The OPC-encoded quality of the value.
TIME_STAMP	NO	DATETIME	The time stamp of the value.
POINT_COUNT	NO	INTEGER	The target number of points in the trend.
ACTUAL_POINT_COUNT	NO	INTEGER	The actual number of points in the trend.

## MODEL\_CLASSES

### MODEL\_CLASSES

This table contains all information about classes in the XHQ Model Server.

#### Constraints

Primary Key: CLASS\_ID, REVISION\_ID

Foreign Key: SUPER\_CLASS\_ID->MODEL\_CLASSES.CLASS\_ID

Column Name	Nulls?	Type	Description
CLASS_ID	NO	INTEGER	The class ID number.
NAME	NO	NVARCHAR()	The name of the class.
REVISION_ID	NO	INTEGER	The revision ID of the class.
DESCRIPTION	YES	NVARCHAR()	A description of the class.
SUPER_CLASS_ID	YES	INTEGER	The class ID of the super class.

## MODEL\_CLASS\_ROLES

### MODEL\_CLASS\_ROLES

This table contains all information about roles of classes in the XHQ Model Server.

#### Constraints

Primary Key: CLASS\_ID, REVISION\_ID, ROLE\_ID

Foreign Key: CLASS\_ID, CLASS\_REVISION\_ID  
->MODEL\_CLASSES.CLASS\_ID, MODEL\_CLASSES.REVISION\_ID

Foreign Key: ROLE\_ID->MODEL\_ROLES.ROLE\_ID

Unique:

Column Name	Nulls?	Type	Description
CLASS_ID	NO	INTEGER	The ID of the view's class.
CLASS_REVISION_ID	NO	INTEGER	The revision ID of the class.
ROLE_ID	NO	INTEGER	The role ID number.

## MODEL\_MEMBERS

### MODEL\_MEMBERS

This table contains all information about members of classes in the XHQ Model Server.

#### Constraints

Primary Key: CLASS\_ID, CLASS\_REVISION\_ID, MID

Foreign Key: CLASS\_ID, CLASS\_REVISION\_ID  
->MODEL\_CLASSES.CLASS\_ID, MODEL\_CLASSES.CLASS\_REVISION\_ID

Unique:

Column Name	Nulls?	Type	Description
CLASS_ID	NO	INTEGER	The ID of the member's class.
CLASS_REVISION_ID	NO	INTEGER	The revision ID of the class.
MID	NO	SMALLINT	The member ID number.
NAME	NO	NVARCHAR()	The name of the member.
DESCRIPTION	YES	NVARCHAR()	A description of the member.
MEMBER_TYPE	NO	NVARCHAR()	The type of member (PRIMITIVE, OBJECT, COLLECTION).
DATA_TYPE	NO	NVARCHAR()	The name of the class or primitive type of the member.
ATTRIBUTE_TYPE	YES	NVARCHAR()	The attribute of the member (only used MEMBER_TYPE is PRIMITIVE).
RESTRICTED	NO	BOOLEAN	Indicates whether or not access to the member restricted.
HIDDEN	NO	BOOLEAN	Indicates whether or not the member is hidden.

## MODEL\_ROLES

### MODEL\_ROLES

This table contains all information about roles in the XHQ Model Server.

#### Constraints

Primary Key: ROLE\_ID

Unique: NAME

Column Name	Nulls?	Type	Description
ROLE_ID	NO	INTEGER	The role ID number.
NAME	NO	NVARCHAR()	The name of the role.
DESCRIPTION	YES	NVARCHAR()	A description of the class.
OS_GROUP	NO	NVARCHAR()	The operating system group associated with the role.
IS_LOCAL	NO	BOOLEAN	Indicates whether or not the group is "locally" defined.

## MODEL\_ROLE\_PERMISSIONS

### MODEL\_ROLE\_PERMISSIONS

This table contains all information about permissions of roles in the XHQ Model Server.

#### Constraints

Primary Key: ROLE\_ID, NAME

Foreign Key: ROLE\_ID->MODEL\_ROLES.ROLE\_ID

Unique:

Column Name	Nulls?	Type	Description
ROLE_ID	NO	INTEGER	The ID of the role.
NAME	NO	NVARCHAR()	The name of the permission.

## MODEL\_VIEWS

### MODEL\_VIEWS

This table contains all information about views of classes in the XHQ Model Server.

#### Constraints

Primary Key: CLASS\_ID, VIEW\_ID

Foreign Key: CLASS\_ID->MODEL\_CLASSES.CLASS\_ID

Unique:

Column Name	Nulls?	Type	Description
CLASS_ID	NO	INTEGER	The ID of the view's class.
VIEW_ID	NO	INTEGER	The view ID number.
NAME	NO	NVARCHAR()	The name of the view.
DESCRIPTION	YES	NVARCHAR()	A description of the view.

## MODEL\_VIEW\_PROPERTIES

### MODEL\_VIEW\_PROPERTIES

This table contains all information about properties of views in the XHQ Model Server.

#### Constraints

Primary Key: CLASS\_ID, VIEW\_ID, NAME  
 Foreign Key: CLASS\_ID->MODEL\_CLASSES.CLASS\_ID  
 Foreign Key: VIEW\_ID->MODEL\_VIEWS.VIEW\_ID  
 Unique:

Column Name	Nulls?	Type	Description
CLASS_ID	NO	INTEGER	The ID of the view's class.
VIEW_ID	NO	INTEGER	The view ID number.
NAME	NO	NVARCHAR()	The name of the property.
DESCRIPTION	YES	NVARCHAR()	A description of the property.

## TAG\_COLLECTION

### TAG\_COLLECTION

This table contains all the tag names in the XHQ Solution Server.

#### Constraints

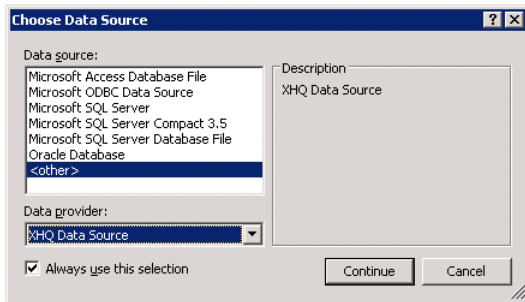
Primary Key: KEY  
 Foreign Key:  
 Foreign Key:  
 Unique:

Column Name	Nulls?	Type	Description
KEY	NO	NVARCHAR()	The name of the tag.
UNITS	NO	NVARCHAR()	The unit of measurement of the tag value..
CONFIGURATION	NO	NVARCHAR()	The configuration for the tag.
DESCRIPTION	NO	NVARCHAR()	A description of the tag.

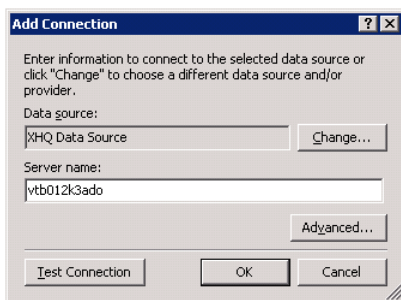
## B - Crystal Reports/Visual Studio Integration

To create a Crystal Report using an XHQ data source

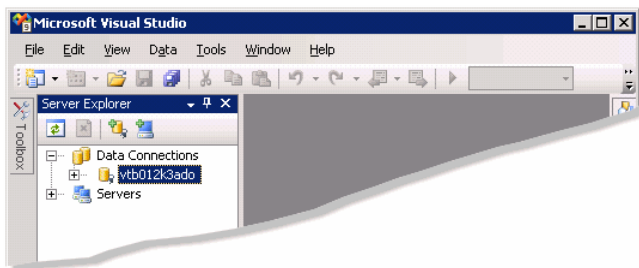
1. Start **Microsoft Visual Studio .NET**.
2. On the **Tools** menu, click **Connect to Database**.  
The "Choose Data Source" dialog appears.



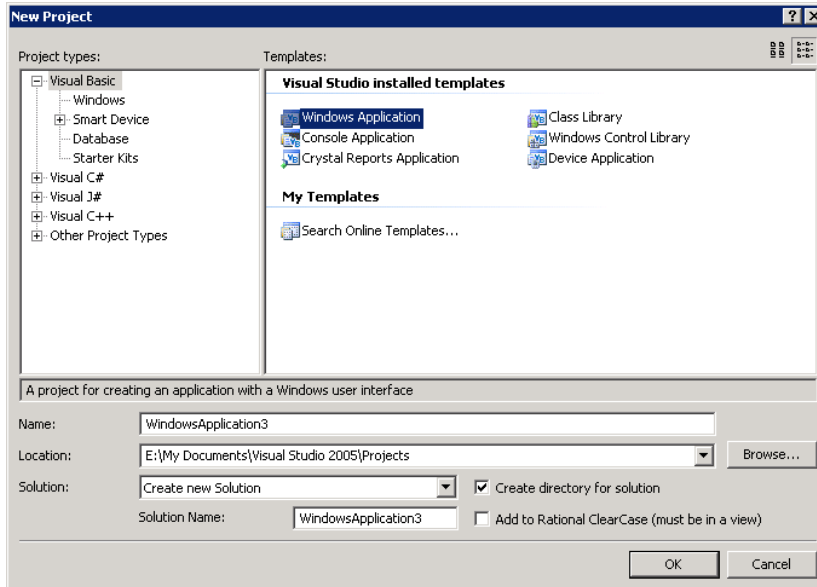
3. In the **Data source** box, select **<other>**.
4. In the **Data provider** list, select **XHQ Data Source** and click **Continue**.  
The "Add Connection" dialog appears.



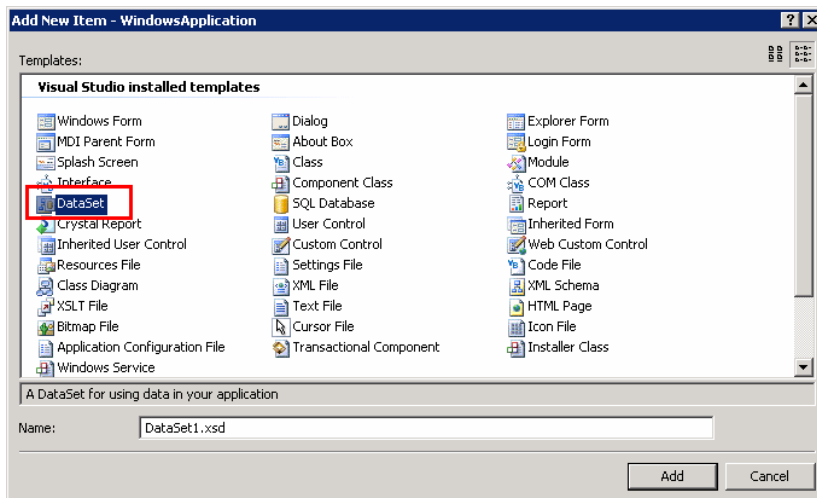
5. In the **Server name** list, **type the host name of your XHQ Server** and click **OK**.  
The new XHQ data connection appears in the Server Explorer pane of Visual Studio.



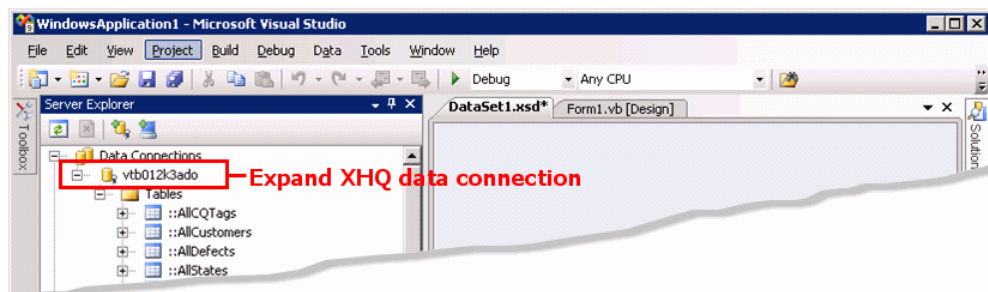
6. On the **File** menu, click **New Project**.  
The "New Project" dialog appears.



7. In the **Project types** box, click (to highlight) **Visual Basic**.
8. In the **Templates** box, click **Windows Application** and then click **OK**.
9. On the **Project** menu, click **Add New Item**.  
The "Add New Item" dialog appears.

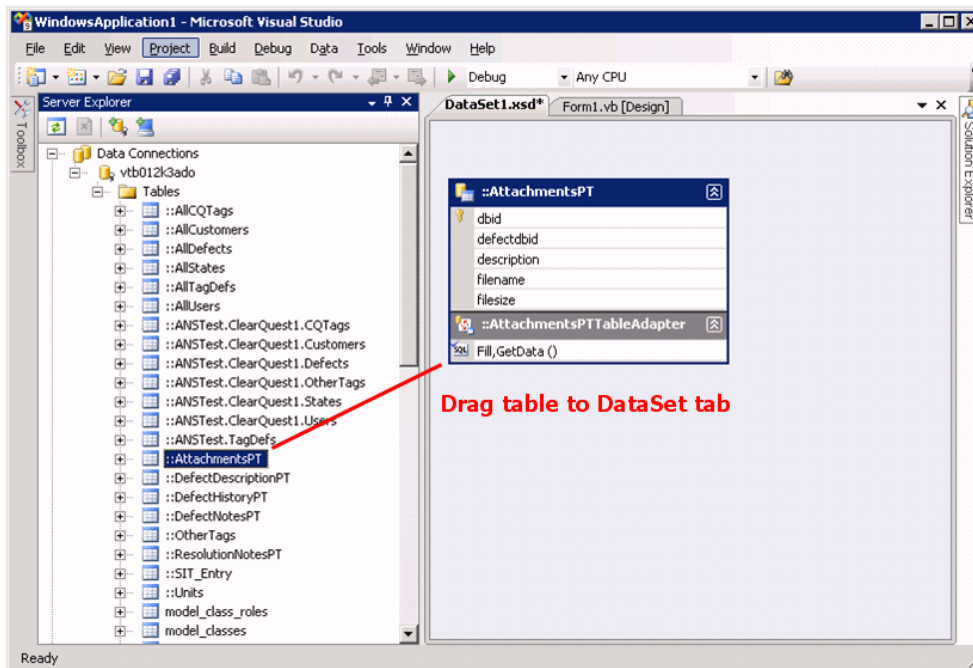


10. Select **DataSet** template, accept the default name, and click **Add**.  
This returns you to the "WindowsApplication" window of Visual Studio, with the "DataSet1.xsd" tab open.
11. In the **Server Explorer** pane, **expand the XHQ data connection**.

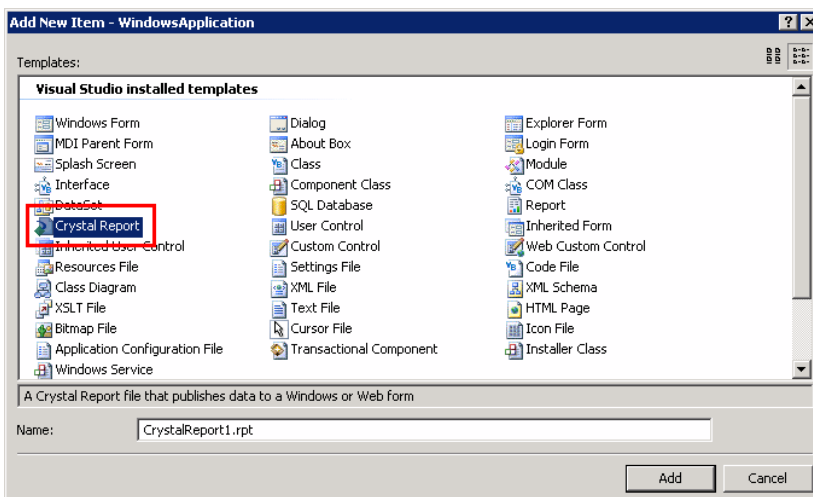




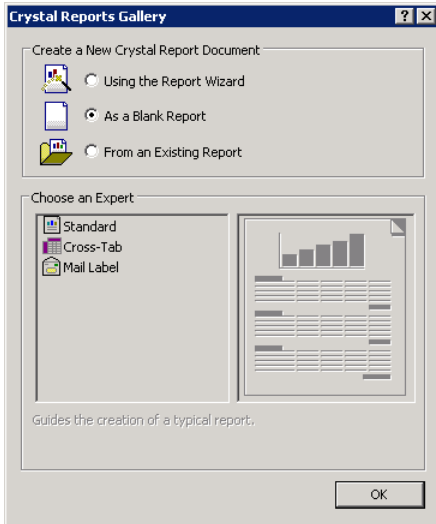
12. Drag a table from the **Server Explorer** pane to the **DataSet1.xsd** tab.



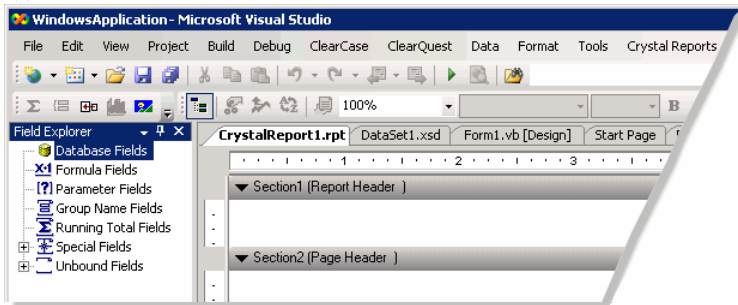
13. Save the DataSet.
14. On the **Project** menu, click **Add New Item**.  
The "Add New Item" dialog appears.



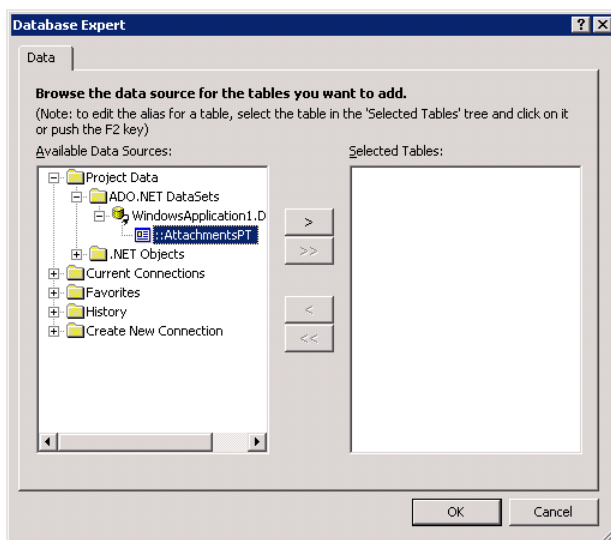
15. Select Crystal Report template, accept the default name, and click **Add**.  
The "Crystal Reports Gallery" dialog appears.



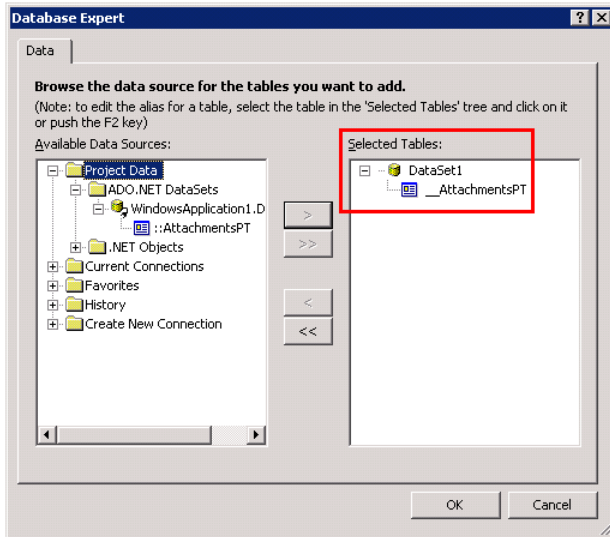
16. Click **As a Blank Report** and click **OK**.  
The Design view of the report appears, with the Field Explorer pane to the left.



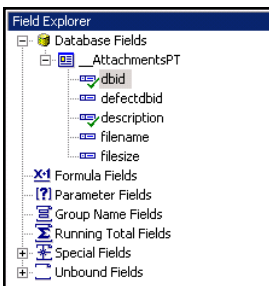
17. In the **Field Explorer** pane, right-click **Database Fields**, and from the shortcut menu, select **Database Expert**.  
The "Database Expert" dialog appears.



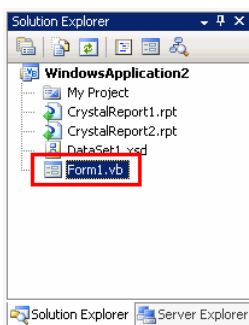
18. In the **Available Data Sources** box, drill-down into **Project Data**, **ADO.NET DataSets**, **<YourProjectName>.Dataset1** to the given **table** you selected in **step 12**.  
19. **Select** (to highlight) **the table name** and **click the right arrow >** to move this table to the **Selected Tables** box.



20. Click **OK**.  
This returns you to the Design view of the report.
21. In the **Field Explorer** pane, expand **Database Fields**.

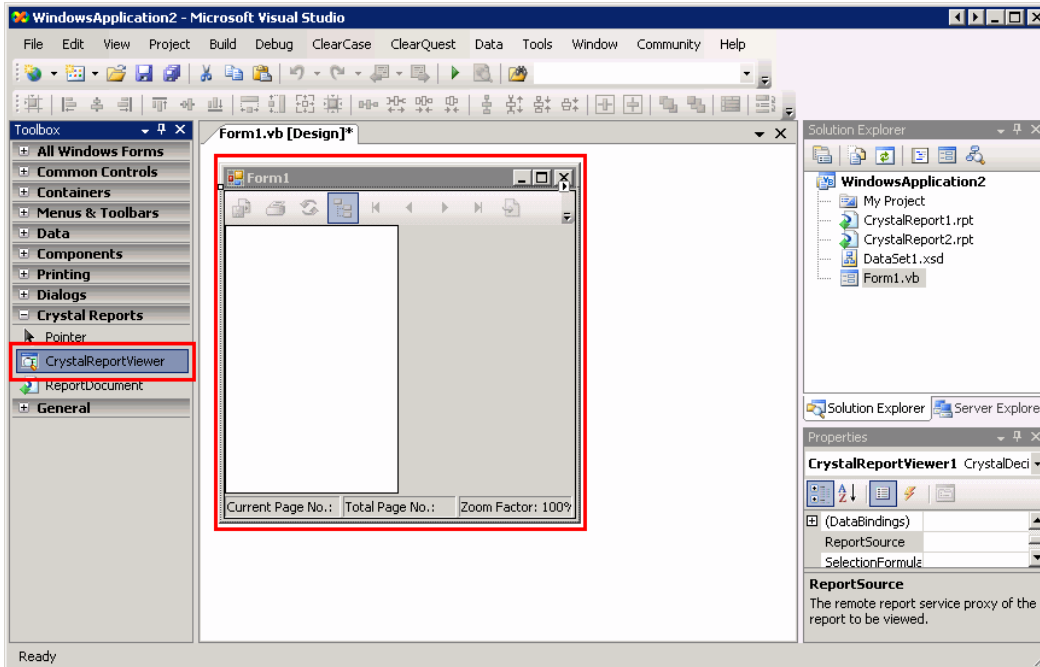


22. Select the desired field from your given table and drag it to a section in the main report.
23. Repeat for each field.
24. In the **Solution Explorer** pane, **double-click Form1.vb** file.

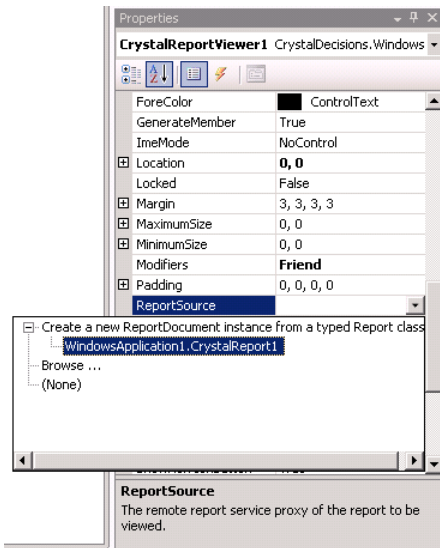


The "Form1.vb [Design]" tab appears.

25. In the **Toolbox** pane, under **Crystal Reports**, select (to highlight) **CrystalReportViewer**.
26. **Drag** the **CrystalReportViewer** control onto **Form1**.



27. Click the **CrystalReportViewer1** object on Form1 and, from the **Properties** pane, click the **ReportSource** property. The "Create a new ReportDocument instance from a typed Report class" pop-up appears.



28. Click **WindowsApplication1.CrystalReport1**. This becomes the value for the ReportSource property.
29. In the **Solution Explorer** pane, **right-click the Form1.vb** object and, from the shortcut menu, click **View Code**.
30. **Insert** the following code to the **Form1 class**, starting on line 2.

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles MyBase.Load
        Dim ta As DataSet1TableAdapters.__AttachmentsPTTableAdapter = New
        DataSet1TableAdapters.__AttachmentsPTTableAdapter()
        Dim ds As DataSet1.__AttachmentsPTDataTable = ta.GetData()
        CrystalReport11.SetDataSource(CType(ds, DataTable))
End Sub
```



This code will differ, depending on the table you selected in your DataSet.

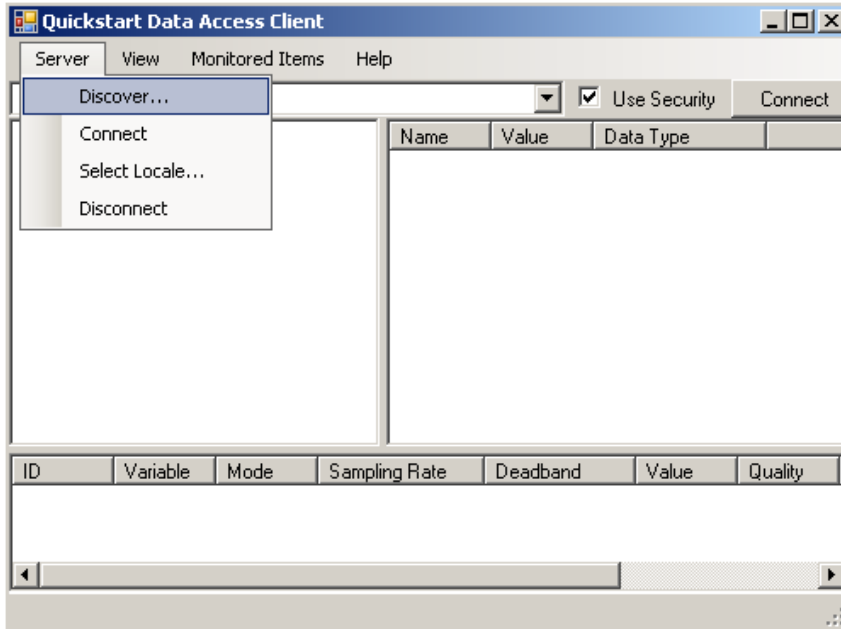
```
1 Public Class Form1
2     Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
3         Dim ta As DataSet1TableAdapters.__AttachmentsPTTableAdapter = New DataSet1TableAdapters.__AttachmentsPTTableAdapter()
4         Dim ds As DataSet1.__AttachmentsPTDataTable = ta.GetData()
5         CrystalReport11.SetDataSource(CType(ds, DataTable))
6     End Sub
7 End Class
```

31. Run your program.

## C - Connecting the OPC Client

### To connect using the OPC Foundation UA Client

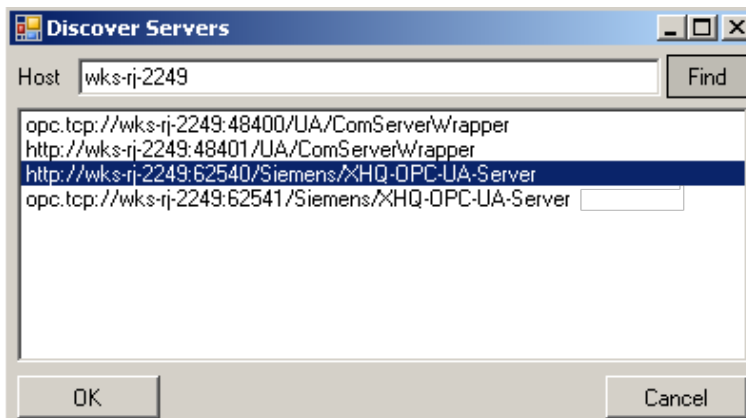
1. From the **OPC Foundation UA Client**, click the **Server** tab, and select **Discover**.



The "Discover Servers" dialog box appears.

2. Enter a **Host** name or an IP address.
3. Click **Find**.

A list of links appears in the box.



4. Select a link with the **format**:

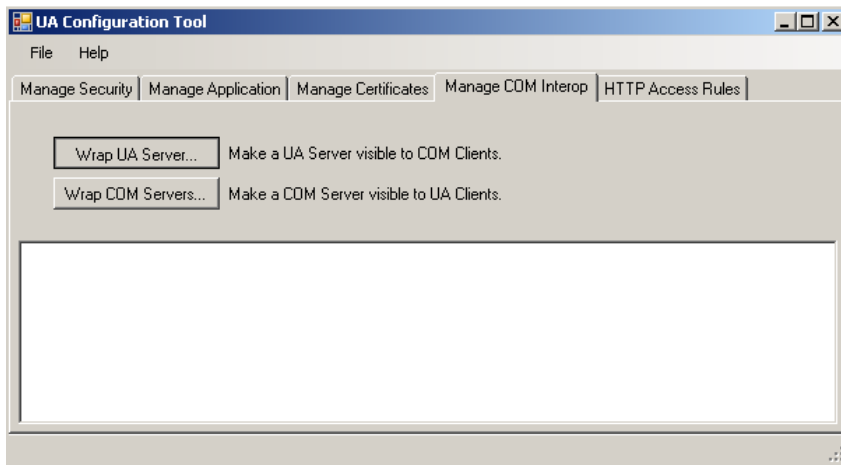
`<protocol>://<server_name>:<port>/Siemens/XHQ-OPC-UA-Server`

*Example:* `http://wks-rj-2249:62540/Siemens/XHQ-OPC-UA-Server`

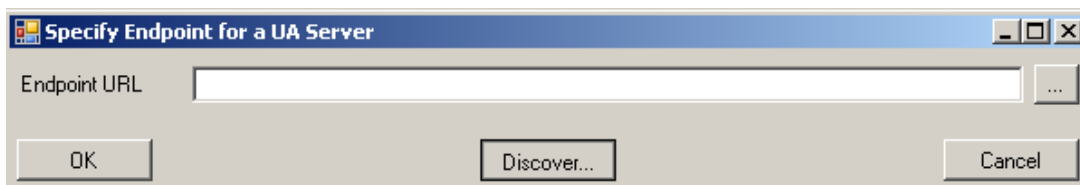
5. Click **OK**.

## To connect using a Classic OPC Client

1. Launch the **UA Configuration Tool** to provide a proxy for COM communications.
2. Click the **Manage COM Interop** tab.

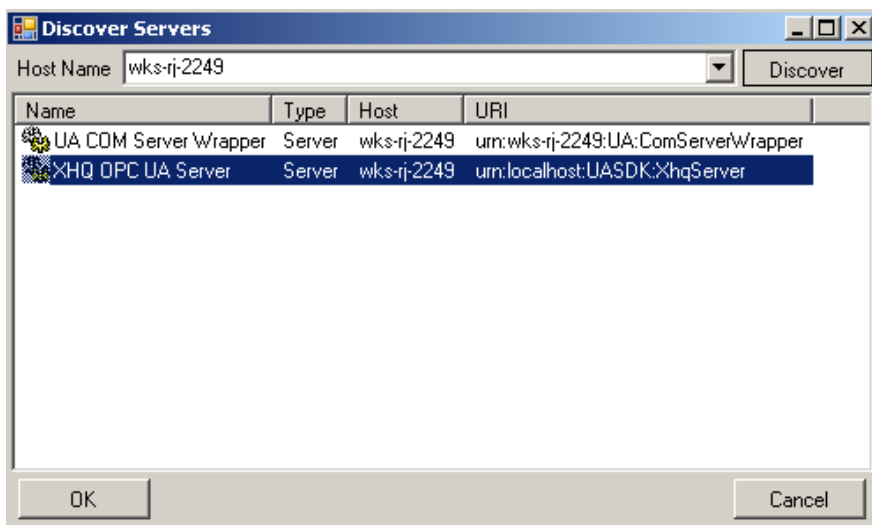


3. Click **Wrap UA Server**.  
The "Specify Endpoint for a UA Server" dialog box appears.

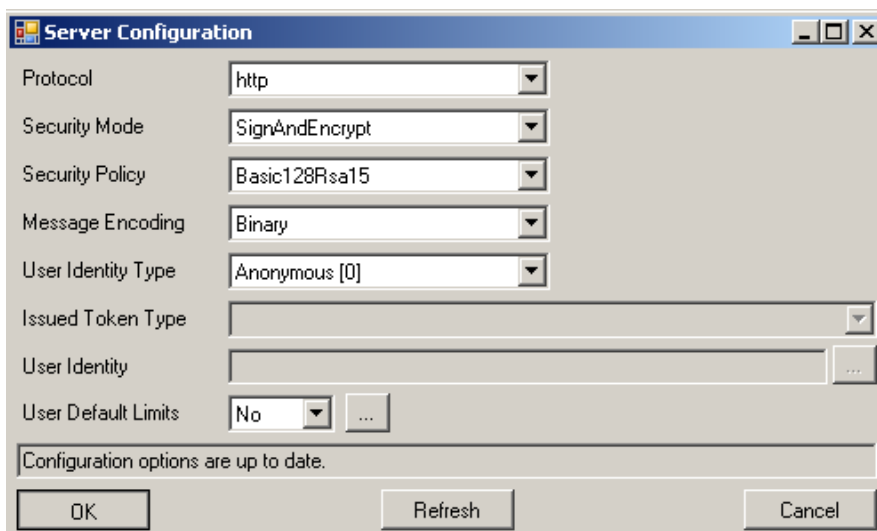


4. Click **Discover**.

The "Discover Servers" dialog box appears.



5. Enter a **Host Name** or an IP address.
6. Click **Discover**.
7. Select (highlight) the **XHQ OPC UA Server** and click **OK**.  
The "Server Configuration" dialog box appears.

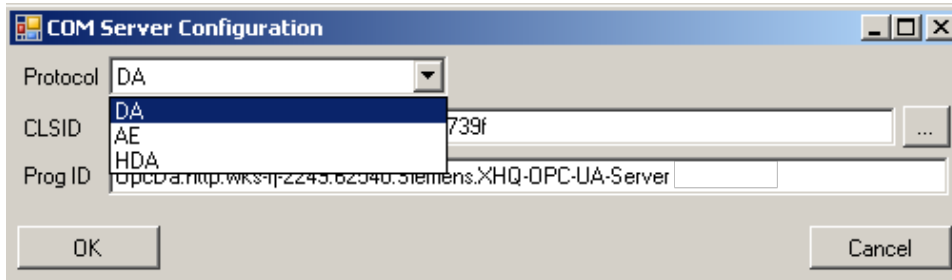


8. Accept the default configuration and click **OK**.  
This returns you back to the "Specify Endpoint for a UA Server" dialog box.



9. Click **OK**.

The "COM Server Configuration" dialog box appears.



10. Select the protocol and click **OK**.



Currently only DA and HDA are available.

11. Navigate to the %CommonApplicationData%\OPC Foundation\ComPseudoServers\ directory.

12. Locate and **open the OPC Server configuration file**, which has the following format:

<OPC-Server-Clsid>.xml



This OPC Server configuration file is created at runtime. The filename generated is in [hexadecimal](#).

*Example:* 3b09332b-c9c0-47c6-b5c3-78f27b6471f3.xml

13. Locate the **transport quota configuration** and change it from this:

```
<d2p1:MaxArrayLength>65535</d2p1:MaxArrayLength>  
<d2p1:MaxMessageSize>4194304</d2p1:MaxMessageSize>
```

To this (note highlighted text):

```
<d2p1:MaxArrayLength>4194304</d2p1:MaxArrayLength>  
<d2p1:MaxMessageSize>16777216</d2p1:MaxMessageSize>
```

14. Open the Classic OPC Client and connect as you would with a Classic OPC Server.

*Example:* The connecting link for this example should be:

OpcDa.http.wks-rj-2249.62540.Siemens.XHQ-OPC-UA-Server

## D - Modifying xhqboot

The primary task of **xhqboot** is to generate customized `startup.dat` and `shutdown.dat` scripts based on its configuration.



### Important Things to Note

- For `xhqboot.properties`:  
Do not leave any extra blank lines in `xhqboot` files. Extra line breaks cause the start-up process to break. Either comment-out blank lines or delete them.  
  
To start OPC UA on `xhqboot`, either the `XHQ_Service` service must be restarted or the server must be rebooted after the XHQ OPC UA Server install. The `XHQ_Service` system context is not aware of the `XHQ_OPCUA_HOME` environment variable after the installation.
- For `xhqbootmods.dat`:  
Leave exactly one extra blank line at the end of the `xhqbootmods.dat` file.

### EXAMPLE: XHQBOOT.PROPERTIES

```
#-----
# STANDARD XHQ SERVER CONFIG -- Leave this segment blank to start all modules by
# default.
###modules=*
#-----
#
# Configuration for custom modules (%XHQ_SERVER_REPOS%\xhqbootmods.dat)
#
# Indicate which custom modules should be used:
startup.custom.modules=xhq_OPCUA_Start
shutdown.custom.modules=xhq_OPCUA_Stop
#
# The "xhq_OPCUA_Start" custom module start after the "xhq_aserver" module
# NOTE: if the xhq_aserver modules is not used, use xhq_sserver reference
# instead
xhq_OPCUA_Start.startup.sequence=+xhq_aserver
#
# The "xhq_OPCUA_Stop" custom module stop before the "xhq_aserver" module
# NOTE: if the xhq_aserver modules is not used, use xhq_sserver reference
# instead
xhq_OPCUA_Stop.shutdown.sequence=-xhq_aserver
```



For more information on the `xhqboot.properties` file, refer to the topic, [Working with .PROPERTIES Files](#), located in the XHQ Administrator's Guide.

#### EXAMPLE: XHQBOOTMODS.DAT

---

```
[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)
#
[xhq OPCUA_Start]
!
! Starting OPC UA Server
!
0 0 5
"%XHQ_OPCUA_HOME%\OPCServer.exe" startup
%XHQ_SERVER_LOGS%\OPCUAServer_startup
[xhq OPCUA_Stop]
!
! Stopping OPC UA Server
!
0 0 5
"%XHQ_OPCUA_HOME%\OPCServer.exe" shutdown
%XHQ_SERVER_LOGS%\OPCUAServer_shutdown
```



Leave **one blank line at the end** of the `xhqbootmods.dat` file



For more information on the `xhqbootmods.dat` file, refer to the topic, [Starting the XHQ Platform](#), located in the XHQ Administrator's Guide.