SIEMENS

Siemens PLM Software: PD PA AE CIS XHQ

Standard XHQ Tag Sync Technical Specification

Revision 3.0

September 25, 2018

Document Change Summary

Revision	Sections	Remarks/Status	Initials	Date
1.0	All	Initial Version	EES	04/11/2018
2.0	3, 4 and 5	Replacing the Microsoft Command Line Transformation Utility by PowerShell script	EES	06/21/2018
3.0	4.4	Tag Sync Examples	JQ	09/25/2018

Proprietary Notice

Author

Siemens PLM Software
PD PA AE CIS XHQ
11000 Richmond Avenue, Suite 110
Houston, TX 770242, USA

Table of Contents

1 objectives 4

2 Reference documents 4

3 Tag Sync Standard Files 4

3.1 XHQTagSync.BAT 5

3.2 XHQTagSync.ps1 5

3.3 XHQTagSync.xsl 8

3.3.1 Tag Configuration Parameters 8

3.4 XHQTagSyncCustomVariables.xsl 11

3.5 XHQTagSyncJavaScripts.xsl 13

4 implementing xhq tag Synchronization 13

4.1 Step 1 – Receiving Tag metadata 13

4.2 Step 2 – Creating the Tag Collection 14

4.3 Step 3 – Using the XHQ Standard Tag Sync Files 16

4.4 Examples 16

4.4.1 XHQ Simulator Tags 16

4.4.2 XHQ Cache Tags 16

4.4.3 Database Tags 17

4.4.4 IP.21 Tags 17

4.4.5 PHD/PHDV Tags 17

4.4.6 PI Tags 18

4.4.7 PI AF Tags 19

4.4.8 OPC Tags 19

4.4.9 OPCUA Tags 19

4.4.10 Calculation Tags 20

5 Logging and Troubleshooting information 20

1 OBJECTIVES

The objective of this document is to describe how the Standard Tag Sync files are built. The document also defines how to configure the Tag Sync process in a XHQ Solution using the Standard Tag Sync files.

2 REFERENCE DOCUMENTS

The Table 2.1 describes the reference documents for this guide.

Table 2.1 - Reference documents

Type	Name
XHQ Document	XHQ Administrators Guide
XHQ Document	XHQ Connection Guide
XHQ Document	XHQ Developers Guide
XHQ Document	XHQ Getting Started
XHQ Document	XHQ Installation Guide
XHQ Document	XHQ Performance Analytics Guide
XHQ Document	XHQ Reference Guide
XHQ Document	XHQ Solution Design and Architecture

3 TAG SYNC STANDARD FILES

The Tag Sync Standard Files are composed by three XSL files, one BAT file and the PowerShell script file. These files are used to complete the XHQ Tag Synchronization process:

Table 3.1 – Tag Sync Files

File Name	Type	Description
XHQTagSync	BAT	Called by the XHQ Solution Server, this file performs tasks to parse the import XML, import and apply configurations and error handling.
XHQTagSync	XSL	Contains the Tag format templates to transform the exported XML files.
XHQTagSyncCustomVariables	XSL	Contains variables used to customize project-specific nomenclatures.
XHQTagSyncJavaScripts	XSL	Contains JavaScript functions that assist the XML transformation.

File Name	Type	Description
XHQTagSync	ps1	The TagSync.ps1 PowerShel script file is called by the TagSync.bat file, this file performs tasks to parse the import XML, import and apply configurations in XHQ and error handling.

3.1 XHQTagSync.BAT

After generating the XHQTagSync.xml file, the XHQ Solution Server calls the XHQTagSync.bat script, which is located in the %XHQ_SERVER_REPOS%\bin directory. This batch file call the PowerShell script files which is responsible for the XML transformation as well as error handling and logging.

The following screenshot shows how the error and logging handling are built in the XHQTagSync.bat file:

```
@echo off

set hour=%time:~0,2%
if "%hour:~0,1%" == " " set hour=0%hour:~1,1%
set min=%time:~3,2%
if "%min:~0,1%" == " " set min=0%min:~1,1%
set secs=%time:~6,2%
if "%secs:~0,1%" == " " set secs=0%secs:~1,1%
set msecs=%time:~9,2%
if "%msecs:~0,1%" == " " set msecs=0%msecs:~1,1%
set year=%date:~-4%
set month=%date:~4,2%
if "%month:~0,1%" == " " set month=0%month:~1,1%
set day=%date:~7,2%
if "%day:~0,1%" == " " set day=0%day:~1,1%

rem Reformatted date
set datetimef=%month%/%day%/%year% %hour%:%min%:%secs%.%msecs%:

echo %datetimef% XhqTagSync.bat running ...

echo %datetimef% Executing %XHQ_SERVER_REPOS%\bin\XhqTagSync.ps1

PowerShell -NoProfile -ExecutionPolicy Bypass -Command "& {Start-Process PowerShell -ArgumentList '-NoProfile
-ExecutionPolicy Bypass -File ""%XHQ_SERVER_REPOS%\bin\XHQTagSync.ps1""' -Verb RunAs}";

IF %ERRORLEVEL% EQU 0 echo %datetimef% XhqTagSync.ps1 - Executed. Check log files for details.
IF %ERRORLEVEL% NEQ 0 echo %datetimef% XhqTagSync.ps1 - Execution failed.
```

Figure 3-1 – TagSync.bat content

3.2 XHQTagSync.ps1

The XHQTagSync.bat calls the XHQTagSync.ps1 which is responsible for processing the XML files and import the new configuration into XHQ Server. It performs the following main tasks:

- Transforms the XHQTagSync.xml file in order to create an import XML file that specifies the tag configurations;
- Runs *xhqci_s importsoln* to import the configurations XML. Therefore, the XHQTagSync.bat file takes care of importing the new tag configurations to the solution.
- Runs xhqci_s configimportcomplete to inform the XHQ Solution Server that the import process is complete.

This file also contains error handling and logging functionalities:

- Error Handling
  - Minimize the impact of run-time errors;
  - Starts the next task only when the previous one has finished successfully, preventing the solution from running *xhqci_s configimportcomplete* without having successfully completed the *xhqci_s importsoln* step.
- Logging
  - Describes when the process failed and helps to find where the problem occurred.
- File backup and cleanup
  - Copy and rename the new configuration files;
  - Delete old data XML files from tmp folder older than $daysBack variable;
  - Provides different levels of loggin information: ("DEBUG","INFO","WARN","ERROR","FATAL");
- Email functionality
  - Sends email notification for errors during the TagSync process.

The Figure 3-2 shows the steps of the XHQTagSync.bat file:

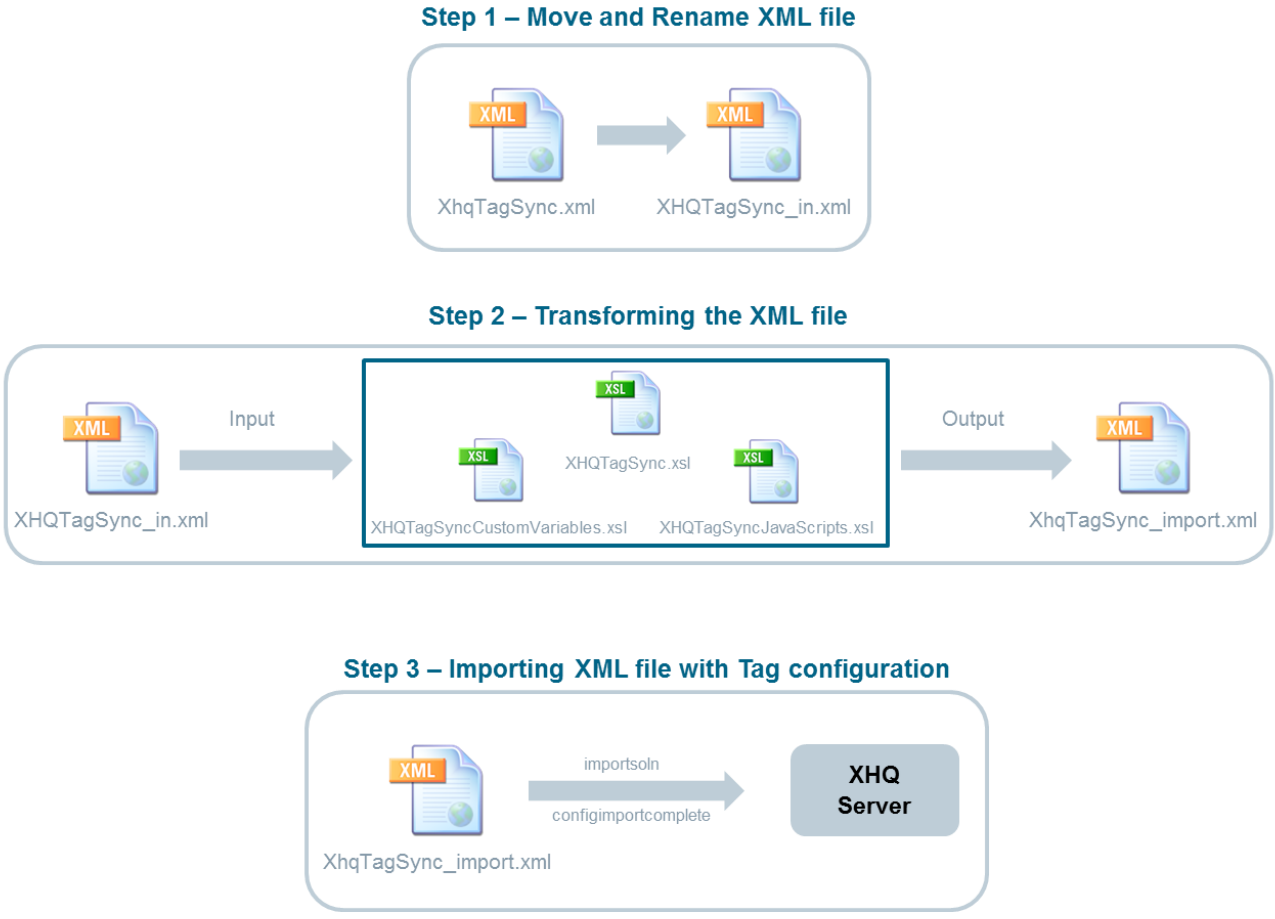

Figure 3-2 – XHQTagSync.ps1 steps

The XHQTagSync.ps1 script allows the customization of certain parameters related to Logging information, email notification and XML file cleanup settings as follow:

- Log Settings
  - $logFile = "$env:XHQ_LOGS\XHQTagSyncSteps.out"
  - $logLevel = "DEBUG" # ("DEBUG","INFO","WARN","ERROR","FATAL")
  - $logSize = 5mb # 30kb
  - $logCount = 2

Table 3.2 – Log settings

Variable Name	Description	Possible values/examples
$logFile	Log File path\name to be created with the PowerShell script output.	"$env:XHQ_LOGS\XHQTagSyncSteps.out"
$logLevel	Defines the log level.	"DEBUG","INFO","WARN","ERROR","FATAL"
$logSize	Defines the output log file size.	5mb
$logCount	Defines how many different log files should be created.	2

- Email notification Settings
  - $flagSendemails = 0
  - $emailFromAddress = "fromAddress@example.com"
  - $emailToAddress = "toAddress@example.com"
  - $emailSubject = "[XHQ TagSync Process] - Process Failed - Action Required"
  - $smtpServer = "smtp.example.com"

Table 3.3 – Email notification settings

Variable Name	Description	Possible values/examples
$flagSendemails	Flag that defines whether email notifications should be sent or not.	0 or 1
$emailFromAddress	Defines the email address for the sender.	"fromAddress@example.com"
$emailToAddress	Defines the email address of the recipients.	"toAddress@example.com"
$emailSubject	Defines the email subject.	"[XHQ TagSync Process] - Process Failed - Action Required"
$smtpServer	SMTP server address.	"smtp.example.com"

Table 3.4 – XML Backup Files Cleanup Settings

Variable Name	Description	Possible values/examples
$daysBack	Integer value that represents the number of days the backup files should be stored.	7

3.3 XHQTagSync.xsl

The TagSync.xsl contains the XSL templates to transform the XhqTagSync.xml to a formatted XML file, named as XhqTagSync_import.xml, which will be imported by the XHQ Solution Server.

The XHQTagSync.xsl file references the XHQTagSyncCustomVariables.xsl and XHQTagSyncJavaScripts.xsl which will be explained in the following chapters.

3.3.1 Tag Configuration Parameters

The Tag Configuration parameters are used to configure the Tag member in XHQ. It must be created in the right order and must follow the values restrictions.

The Figure 3-3 exemplifies parameters for a Tag which uses the XHQ Simulator connector and a PHDV server to write values:

```
<Tag
    name="Teste_SimulatorPHDV">
    <TagMember
        name="sTagname"
        value="Teste_SimulatorPHDV"
    />
    <TagMember
        name="sTagConfig"
        value="XHQSIM##XHQ_Sim##Real####TagName##########################PHDV##true####PHDV##PHDVTAGNAME####RANDOM##110##130##2##300##1"
    />
</Tag>
```

Figure 3-3 – XHQTagSync.bat steps

The correct order is specified in Table 3.5.

Table 3.5 – Tag Configuration Parameters

Num	Name	Description	Possible Values	Mandatory
0	Tag Group Name	The custom Tag group names.	Must be included in the XHQTagSyncJavaScripts.xsl file.	Yes
1	XHQ Connection Group	The name of the connection group which the Tag is associate with.	Must be the exactly XHQ Connection Group name.	Yes
2	Data Type	The data type of the tag.	Possible values are configured in the XHQTagSyncCustomVariables.xsl.	Yes
3	XHQ Security Roles	List of roles to be applied to the Tag primitive member.	Must contain exactly the same name of the XHQ security roles, separated by “;”.	No

Num	Name	Description	Possible Values	Mandatory
4	BackEndTagName	Name of the back end tag that the XHQ tag is being connected to.	String value.	Yes
5	Digital State Alias	Tag name/alias of the Digital state tag being configured (this is project-specific).	String value.	No
6	SQL Query / Path	SQL Query that goes in the Connect tab or the point Path used by XHQ Connector Tag type.	String value which contains SQL query or a valid XHQ member Path.	No
7	Interpolation Type	Interpolation type of the Cache and Database Tag types.	Integer value, where: 0 - Normal; 1 - Time Series Linear; 2 - Time Series Stepped. Default value is: 1.	No
8	Pool Interval	Interval time in milliseconds which goes in the Connect tab (used ONLY for CACHE group connector).	Integer value where minimum is 60000.	No
9	Access Path	Path that goes in Connect tab for Simatic IT and OPC connection group types.	String value.	Yes[1]
10	HDA Item	String Value that goes in the Connect tab for Simatic IT and OPC connection group types.	String value.	Yes[1]
11	DA Property/HDA Attribute/Tag Attribute	String Value that goes in the Connect tab for Simatic IT and OPC connection group types. Or Tag Attribute used by the IP21 Tag Types	String value.	Yes[1]
12	Server Side Expression	Expression that goes in the Expression tab.	Must be a valid XHQ Server Side expression.	No
13	Schedule Enabled	Defines whether expression scheduling is enabled.	Enabled = 1, Disabled = 0, Default value is 0.	No
14	Schedule Base Time	Defines a schedule base time for the calculation.	Valid date string. Format: MM/dd/yyyy HH:mm:ss (24h), e.g. 12/24/2007 23:00:00.	No
15	Schedule Poll Period	Defines a schedule poll rate.	Values are in milliseconds. Minimum value: 60000.	No
16	Recorder Name	Recorder name that goes in the History tab.	Possible values are: Database, IP21, PHD, PHDV, and PI. Default value is: Database.	No
17	Record Data Enable	Boolean that goes in the History tab.	Possible values: true or false. Default value is: false.	No

[1] Mandatory for specific Tag Types.

Num	Name	Description	Possible Values	Mandatory
18	Recorder Marker Mode	Recorder Marker Mode that goes in the History tab.	Possible values: Undefined, On or Off.	
19	Database Recorder Interpolation Type	Defines the Data Recorder interpolation type.	Possible values are: Default, Stepped or Linear.	No
20	Target Connection Group	Defines the XHQ Connection Group which is to be used for write back operations. This connector in XHQ must be configured with "-allowWrite".	String value.	Yes[2]
21	Target Tag Name	Defines the PIMS write back tag name in which to save history.	String value.	Yes[2]
22	Target Write Mode	Write mode specific for PI connections.	Integer value. Possible values are: 8 - Insert Duplicates, 3 - Error Duplicates, 5 - Replace Duplicates, 6 - Replace On Duplicates.	No
23	Simulator Type	Defines the XHQ simulator type.	Possible values are: RANDOM, COUNT or SINEWAVE.	Yes[3]
24	Simulator Low Limit	Defines the lowest value the simulator will use.	Numeric value.	Yes[3]
25	Simulator High Limit	Defines the highest value the simulator will use.	Numeric value.	Yes[3]
26	Simulator Data Rate	Defines the data rate update in seconds.	Numeric value.	Yes[3]
27	Simulator Sine Period	Defines the sine period.	Numeric value.	Yes[3]
28	Simulator Deviation	Defines the fluctuation rate of the change.	The value must range from 0.00 to 1.00.	Yes[3]
29	PIAF Path	Defines the Path used by the PIAF Tag Type.	String value.	Yes[4]
30	PIAF GUID	Defines the GUID used by the PIAF Tag Type.	String value.	Yes[4]
31	SNMP OID	Defines the OID (Object Identifier) used by the SNMP Tag Type.	String value.	Yes[5]
32	OPC UA DA ItemID or NodeID flag	Indicates whether the OPC UA DA element is specified via ItemID or NodeID.	Possible values are: ITEMID or NODEID	Yes[6]
33	OPC UA DA ItemID or NodeID	Defines the OPC UA DA ItemID or NodeID address.	String value.	Yes[6]
34	OPC UA HDA ItemID or NodeID flag	Indicates whether the OPC UA HDA element is specified via ItemID or NodeID.	Possible values are: ITEMID or NODEID	Yes[6]

[2] Mandatory if the variable Record Data Enable is true.
[3] Mandatory if it is a Simulator Tag Type.
[4] Mandatory if it is a PIAF Tag Type.
[5] Mandatory if it is a SNMP Tag Type.
[6] Mandatory if it is an OPC UA Tag Type.

Num	Name	Description	Possible Values	Mandatory
35	OPC UA HDA ItemID or NodeID	Defines the OPC UA HDA ItemID or NodeID address.	String value.	Yes[6]

3.4 XHQTagSyncCustomVariables.xsl

The XHQTagSyncCustomVariables.xsl file contains the custom variables used to customize the process.

An example of custom variable values can be found at Figure 3-4:

```
    <msxsl:script implements-prefix="XHQ" language="JScript">
        <![CDATA[
    //Variable used to define if this TagSync Config file will have write access
    var writeModeEnable = "TRUE";

    //Tag Group Names per Type - Include here the name of the tag group created in your project
    var sPHDTagType = "PHD;PHDV";
    var sPITagType = "PI;PISDK;PI64";
    var sIP21TagType = "IP21";
    var sOPCTagType = "OPC";
    var sSIMATICITTagType = "SIMATICIT";
    var sXHQCACHETagType = "XHQCACHE;XHQCacheGroup";
    var sCALCTagType = "CALC";
    var sDATABASETagType = "SQLServer;Oracle;ODBC;SQLExpress";
    var sXHQSIMULATORTagType = "XHQSIM";
    var sXHQCONNECTORTagType = "XHQConnector";
    var sOPCUATagType = "OPCUA";
    var sSNMPTagType = "SNMP";
    var sPIAFTagType = "PIAF";

  //Tag component name - Include here the name of the Tag component created in your solution
    var xhqTagComponentName = "Tags";

    //Value primitive member name by type - Include here the name of the primitive members by type created in your Tag component
    var iMemberName     = "iCurrentValue";
    var rMemberName     = "rCurrentValue";
    var sMemberName     = "sCurrentValue";
    var dMemberName     = "dCurrentValue";
    var lMemberName     = "lCurrentValue";
    var bMemberName     = "bCurrentValue";
    var dtMemberName    = "dtCurrentValue";

    //Value TagClass Type names - Include here the TagClass Type names received from Historians
    var IntegerTypeNames    = "Int;Integer;I";
    var RealTypeNames       = "Real;R;Float";
    var StringTypeNames     = "String;str;Text";
    var DoubleTypeNames     = "Double;dob;db";
    var LongTypeNames       = "Long;L;l";
    var BoolTypeNames       = "Bool;Boolean;B";
    var DateTimeTypeNames   = "Date;Time;Datetime;dt";
    ]]></msxsl:script>
</xsl:stylesheet>
```

Figure 3-4 – Custom variable values example

The Table 3.6 shows the variables and its description:

Table 3.6 – Custom variables

Variable Name	Description	Possible Values
writeModeEnable	Variable used to define if the Tag Sync process will have write access in the Write Back servers.	Possible values: true or false.

Variable Name	Description	Possible Values
sPHDTagType	A list of XHQ Connection Group names related to the PHD Tag type.	A list of string values separated by ";".
sPITagType	A list of XHQ Connection Group names related to the PI Tag type.	A list of string values separated by ";".
sIP21TagType	A list of XHQ Connection Group names related to the IP21 Tag type.	A list of string values separated by ";".
sOPCTagType	A list of XHQ Connection Group names related to the OPC Tag type.	A list of string values separated by ";".
sSIMATICITTagType	A list of XHQ Connection Group names related to the Simatic IT Tag type.	A list of string values separated by ";".
sXHQCACHETagType	A list of XHQ Connection Group names related to the XHQ Cache Tag type.	A list of string values separated by ";".
sCALCTagType	A list of XHQ Connection Group names related to the Calculated Tag type.	A list of string values separated by ";".
sDATABASETagType	A list of XHQ Connection Group names related to the Relational Database Tag type.	A list of string values separated by ";".
sXHQSIMULATORTagType	A list of XHQ Connection Group names related to the XHQ Simulator Tag type.	A list of string values separated by ";".
sXHQCONNECTORTagType	A list of XHQ Connection Group names related to the XHQ Connector Tag type.	A list of string values separated by ";".
sOPCUATagType	A list of XHQ Connection Group names related to the OPC UA Tag type.	A list of string values separated by ";".
sSNMPTagType	A list of XHQ Connection Group names related to the SNMP Tag type.	A list of string values separated by ";".
sPIAFTagType	A list of XHQ Connection Group names related to the PI AF Tag type.	A list of string values separated by ";".
xhqTagComponentName	The name of the Tag component created in the Solution.	String value.
iMemberName	The name of the primitive member related to the integer value.	String value.
rMemberName	The name of the primitive member related to the real value.	String value.
sMemberName	The name of the primitive member related to the string value.	String value.
dMemberName	The name of the primitive member related to the double value.	String value.
lMemberName	The name of the primitive member related to the long value.	String value.
bMemberName	The name of the primitive member related to the boolean value.	String value.
dtMemberName	The name of the primitive member related to the Datetime value.	String value.
IntegerTypeNames	A list with the type names related to the "Data Type" variable described in Table 3.5.	A list of string values separated by ";". e.g.: "Int;Integer;I".
RealTypeNames	A list with the type names related to the "Data Type" variable described in Table 3.5.	A list of string values separated by ";". e.g.: "Real;R;Float".

Variable Name	Description	Possible Values
StringTypeNames	A list with the type names related to the “Data Type” variable described in Table 3.5.	A list of string values separated by “;”. e.g.: "String;str;Text".
DoubleTypeNames	A list with the type names related to the “Data Type” variable described in Table 3.5.	A list of string values separated by “;”. e.g.: "Double;dob;db”.
LongTypeNames	A list with the type names related to the “Data Type” variable described in Table 3.5.	A list of string values separated by “;”. e.g.: "Long;L;l".
BoolTypeNames	A list with the type names related to the “Data Type” variable described in Table 3.5.	A list of string values separated by “;”. e.g.: "Bool;Boolean;B".
DateTimeTypeNames	A list with the type names related to the “Data Type” variable described in Table 3.5.	A list of string values separated by “;”. e.g.: "Date;Time;Datetime;dt".

3.5 XHQTagSyncJavaScripts.xsl

The XHQTagSyncJavaScripts.xsl file contains the JavaScript functions responsible for supporting the XHQTagSync.xsl templates. It also contains functions responsible for parsing the TagConfig data and reading the custom variables values created in the XHQTagSyncCustomVariables.xsl file.

4 IMPLEMENTING XHQ TAG SYNCHRONIZATION

This section describes how to implement the Tag Synchronization process in a XHQ solution. These are sample steps. In other words, the procedure can be modified according to the solution requirements.

4.1 Step 1 – Receiving Tag metadata

Create XHQ Connection Process and XHQ Connection Group to connect to the tag metadata source. Then create a Global collection to receive the metadata as follows:

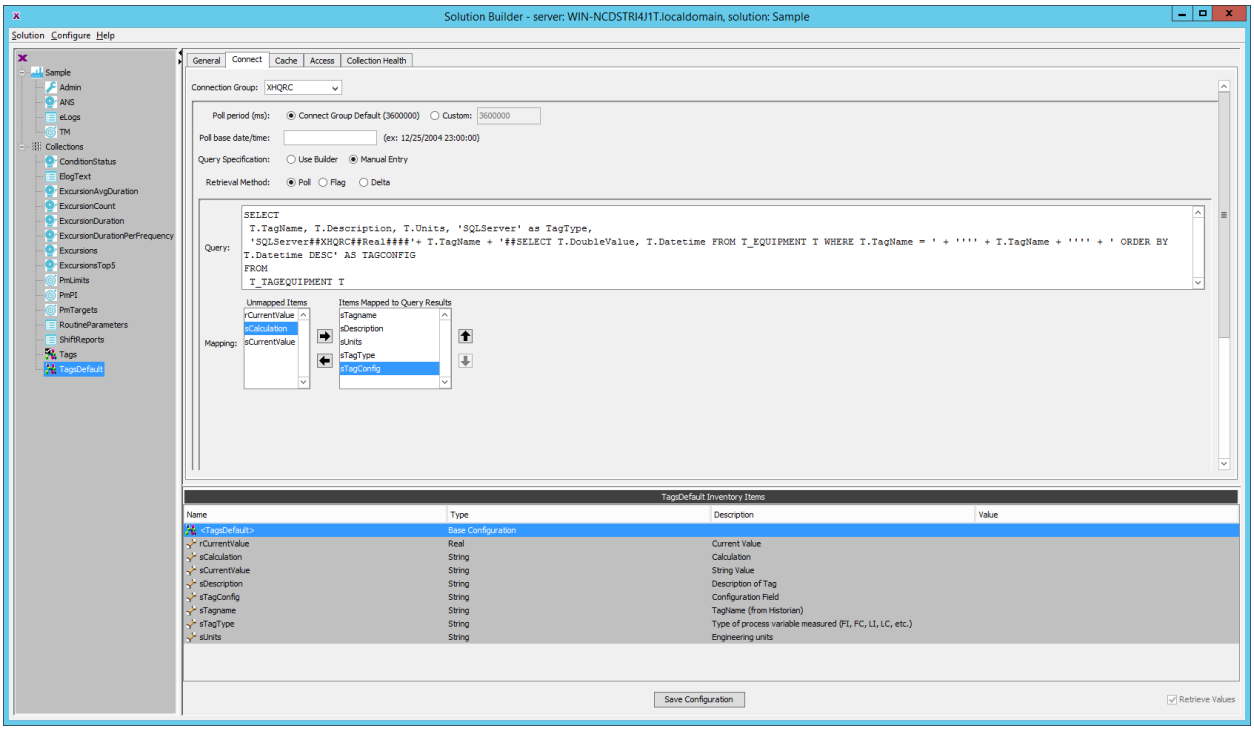

Figure 4-1 – Configuring a Tag metadata collection

4.2 Step 2 – Creating the Tag Collection

After receiving the Tag metadata information, an XHQ Tag Collection must be configured in order to allow the Tag Sync process works.

1. Create a Tag component using the XHQ Workbench and configure the Tag properties of the primitive members correctly:
    a. **None**: This is the default. No tag attribute is applied to the component member.
    b. **Value**: If Value is selected for a member, then that member will be used as the default pen for trending. You can, however, have multiple selections with this tag attribute; in which case, the default pen for trending will be the member with the alias assigned to it (which by default is the one with the first member ID).
    c. **Units**: If Unit is selected for a member, then that member will be used to hold the tag unit. Setting this attribute is OPTIONAL.
    d. **Description**: If Description is selected for a member, then that member is used to hold the tag's description string. Setting this attribute is OPTIONAL.
    e. **Key**: This is the Tag name or ID. Only one member of a tag component can have the Key attribute. This member must be of String type. In the Tags collection, this column must necessarily have unique values.

f. **Config**: This attribute is used as a place holder for extra information needed to configure a tag. Typically, it is used to group tags under different categories and determine whether a particular tag needs configuration update.

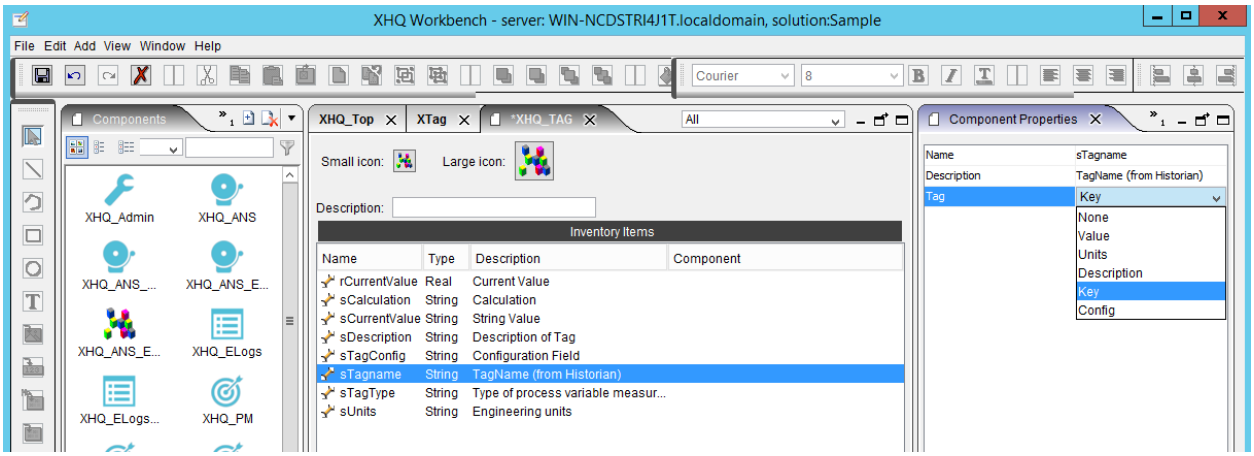

Figure 4-2 – Configuring a Tag component

2. Create a Tag Collection using the Solution Builder:

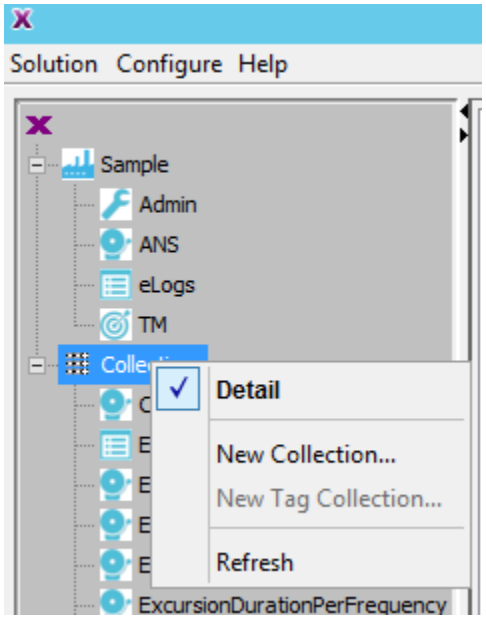

Figure 4-3 – Creating the XHQ Tag Collection

3. Configure the Tag Collection to receive data from the Tag metadata collection created in the item 4.1.

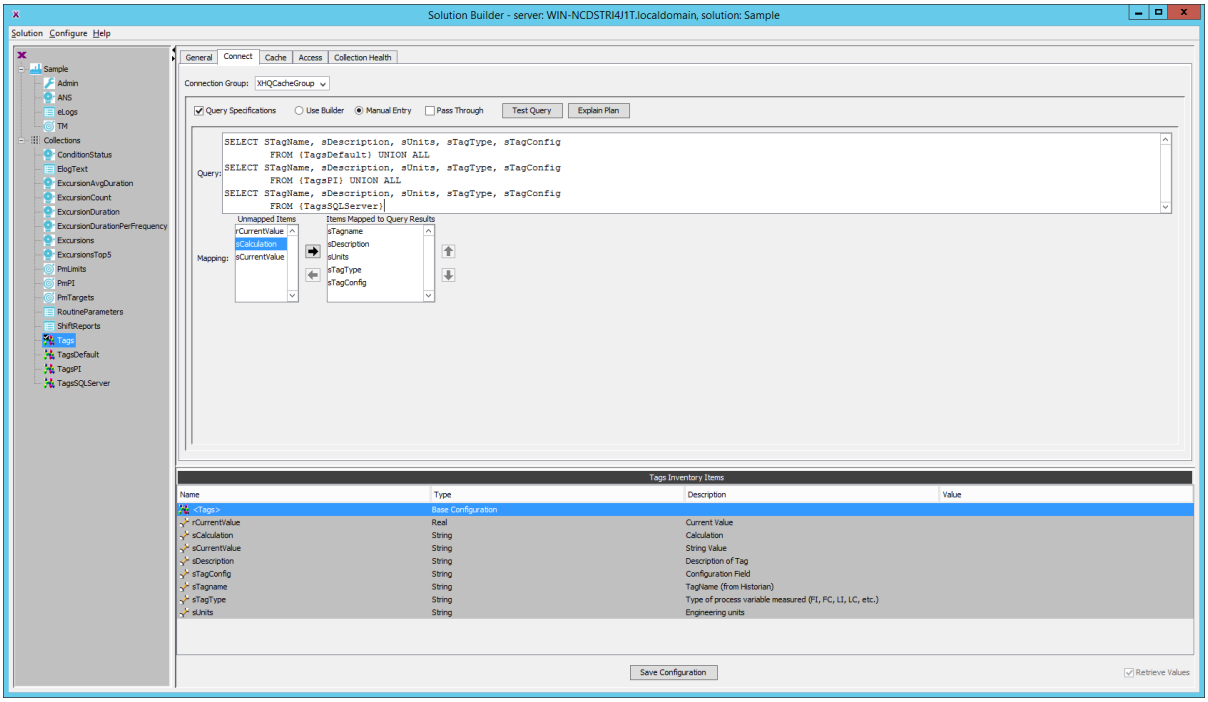

Figure 4-4 – Tag Collection configuration

4.3 Step 3 – Using the XHQ Standard Tag Sync Files

Copy the XHQ Standard Tag Sync Files to the %XHQ_SERVER_REPOS%\bin folder.

4.4 Examples

These examples were made based on XHQ Sample repos and some dependent configurations which are not included in standard Sample repos. And they might not work in customer solution and they are for developer reference only. The purpose of these examples are for reference only and cannot be used for production prior to changes tailored for customer solution.

4.4.1 XHQ Simulator Tags

Dependencies: Cache collection TagsDefault; XHQ Simulator Connection Group: XHQ_Sim, XHQ Data Recorder needs to be enabled.

```
SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), sTagType, NVL(sCalculation, N'-'), sTagConfig
FROM {TagsDefault}
```

4.4.2 XHQ Cache Tags

Dependencies: Global collections: **TagsDefault** and **XHQ_Data**; XHQ Cache Connection Group: **XHQ_Sim**.

```
SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'real', NVL(sCalculation, N'-'),
'XHQCache##XHQCacheGroup##real####' || sTagName || '####select rValue, dtDateTime from {XHQ_Data} where
sName = '''|| sTagname || ''' order by dtDateTime desc####0'
FROM {TagsDefault}
```

4.4.3 Database Tags

Dependencies: Global collections: **TagsDefault**; XHQ ODBC Connection Group: **ODBCTagValue_1min**; Table: **TagValues** table or view with sTagName, rValue and dtDateTime columns

```
SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'real', NVL(sCalculation, N'-'),
'ODBC##ODBCTagValue_1min##real####' || sTagName || '####select rValue, dtDateTime from TagValues where
sTagName = '''|| sTagname || ''' order by dtDateTime desc'
FROM {TagsDefault}
```

4.4.4 IP.21 Tags

1. TagSync w/o Writeback

   Dependencies: Global collections: **TagsDefault**; IP.21 connection group: **IP21TagValue_1min**.

   ```
   SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'real', NVL(sCalculation, N'-'),
   'IP21##IP21TagValue_1min##Real####' || sTagName || '###############AW_LEDPTG'
   FROM {TagsDefault}
   ```

2. TagSync w/o Writeback for Digital State tags

   Dependencies: Global collections: **TagsDefault**; IP.21 connection group: **IP21TagValue_1min**.

   ```
   SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'real', NVL(sCalculation, N'-'),
   'IP21##IP21TagValue_1min##Real####' || sTagName || '##DS_' || sTagName || '#############AW_LEDPTG'
   FROM {TagsDefault}
   ```

3. TagSync w/ Writeback to IP.21 Historian

   Dependencies: Global collections: **TagsDefault**; IP.21 connection group: **IP21TagValue_1min** and **IP21TagValueWriteback_1min**. IP.21 Writeback needs to be enabled in XHQ.

   ```
   SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'real', NVL(sCalculation, N'-'),
   'IP21##IP21TagValue_1min##Real####' || sTagName ||
   '###############AW_LEDPTG##########IP21##true##On####IP21TagValueWriteback_1min##WB_' || sTagName
   FROM {TagsDefault}
   ```

4. TagSync w/ Writeback to IP.21 Historian for Digital State tags

   Dependencies: Global collections: **TagsDefault**; IP.21 connection group: **IP21TagValue_1min** and **IP21TagValueWriteback_1min**. IP.21 Writeback needs to be enabled in XHQ.

   ```
   SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'real', NVL(sCalculation, N'-'),
   'IP21##IP21TagValue_1min##Real####' || sTagName || '##DS_'|| sTagName ||
   '#############AW_LEDPTG##########IP21##true##On####IP21TagValueWriteback_1min##WB_' || sTagName
   FROM {TagsDefault}
   ```

4.4.5 PHD/PHDV Tags

1. TagSync w/o Writeback

   Dependencies: Global collections: **TagsDefault**; PHDV connection group: **PHDTagValue_1min**.

   ```
   SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'Real', NVL(sCalculation, N'-'),
   N'PHDV##PHDTagValue_1min##Real####' || sTagname
   FROM {TagsDefault}
   ```

2. TagSync w/o Writeback for Digital State tags

Dependencies: Global collections: **TagsDefault**; PHDV connection group: **PHDTagValue_1min**.

```
SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'Real', NVL(sCalculation, N'-'),
N'PHDV##PHDTagValue_1min##Real####' || sTagname || '##DS_' || sTagName
FROM {TagsDefault}
```

3. TagSync w/ Writeback to PHD Historian
   Dependencies: Global collections: **TagsDefault**; PHDV connection group: **PHDTagValue_1min** and **PHDTagValueWriteback_1min**. PHD Writeback needs to be enabled in XHQ.

```
SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'Real', NVL(sCalculation, N'-'),
N'PHDV##PHDTagValue_1min##Real####' || sTagName ||
'########################PHDV##true##On####PHDTagValueWriteback_1min##WB_' || sTagName
FROM {TagsDefault}
```

4. TagSync w/ Writeback to PHD Historian for Digital State tags
   Dependencies: Global collections: **TagsDefault**; PHDV connection group: **PHDTagValue_1min** and **PHDTagValueWriteback_1min**. PHD Writeback needs to be enabled in XHQ.

```
SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'Real', NVL(sCalculation, N'-'),
N'PHDV##PHDTagValue_1min##Real####' || sTagName || '##DS_' || sTagName ||
'########################PHDV##true##On####PHDTagValueWriteback_1min##WB_' || sTagName
FROM {TagsDefault}
```

4.4.6 PI Tags

1. TagSync w/o Writeback
   Dependencies: Global collections: **TagsDefault**; PI connection group: **PITagValue_1min**.

```
SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'Real', NVL(sCalculation, N'-'),
N'PI##PITagValue_1min##Real####' || sTagName
FROM {TagsDefault}
```

2. TagSync w/o Writeback for Digital State Tags
   Dependencies: Global collections: **TagsDefault**; PI connection group: **PITagValue_1min**.

```
SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'Real', NVL(sCalculation, N'-'),
N'PI##PITagValue_1min##Real####' || sTagName || '##DS_' || sTagName
FROM {TagsDefault}
```

3. TagSync w/ Writeback to PI Historian
   Dependencies: Global collections: **TagsDefault**; PI connection group: **PITagValue_1min** and **PITagValueWriteback_1min**. PI Writeback needs to be enabled in XHQ.

```
SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'Real', NVL(sCalculation, N'-'),
N'PI##PITagValue_1min##Real####' || sTagName ||
'########################PI##true##On####PITagValueWriteback_1min##WB_' || sTagName || '##5'
FROM {TagsDefault}
```

4. TagSync w/ Writeback to PI Historian for Digital State tags
   Dependencies: Global collections: **TagsDefault**; PI connection group: **PITagValue_1min** and **PITagValueWriteback_1min**. PI Writeback needs to be enabled in XHQ.

```
SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'Real', NVL(sCalculation, N'-'),
```

```
N'PI##PITagValue_1min##Real####' || sTagName || '##DS_' || sTagName ||
'######################PI##true##On####PITagValueWriteback_1min##WB_' || sTagName || '##5'
FROM {TagsDefault}
```

4.4.7 PI AF Tags

1. TagSync w/o Writeback

   Dependencies: Global collections: **TagsDefault**; PI AF connection group: **PIAFTagValue_1min**.

```
SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'Real', NVL(sCalculation, N'-'),
N'PIAF##PIAFTagValue_1min##Real####' || sTagName ||
'#####################################################' ||
'\\Sample\Refining\DIST\CDU1\Valve1\Status\PV' || sTagName
FROM {TagsDefault}
```

2. TagSync w/ Writeback to PI Historian

   Dependencies: Global collections: **TagsDefault**; PI AF connection group: **PIAFTagValue_1min** and **PIAFTagValue Writeback_1min**. PI Historian Writeback needs to be enabled in XHQ.

```
SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'Real', NVL(sCalculation, N'-'),
N'PIAF##PIAFTagValue_1min##Real####' || sTagName ||
'######################PI##true##On####PITagValueWriteback_1min##WB_' || sTagName ||
'##5################\\Sample\Refining\DIST\CDU1\Valve1\Status\PV\' || sTagName
FROM {TagsDefault}
```

4.4.8 OPC Tags

1. TagSync w/o Writeback

   Dependencies: Global collections: **TagsDefault**; OPC connection group: **OPCTagValue_1min**; OPC Writeback needs to be enabled in XHQ.

```
SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'Real', NVL(sCalculation, N'-'),
N'OPC##OPCTagValue_1min##Real####' || sTagname || '##########' || sTagName || '##HDA_' || sTagName
|| '##<None>'
FROM {TagsDefault}
```

2. TagSync w/ Writeback to OPC Server

   Dependencies: Global collections: **TagsDefault**; OPC connection group: **OPCTagValue_1min** and **OPCTagValue Writeback_1min**; OPC Writeback needs to be enabled in XHQ.

```
SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'Real', NVL(sCalculation, N'-'),
N'OPC##OPCTagValue_1min##Real####' || sTagname || '##########' || sTagName || '##HDA_' || sTagName
|| '##<None>##########OPC##true##On####OPCTagValueWriteback_1min##WB_' || sTagName
FROM {TagsDefault}
```

4.4.9 OPCUA Tags

1. TagSync w/o Writeback

   Dependencies: Global collections: **TagsDefault**; OPCUA connection group: **OPCUATagValue_1min**.

```
SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'Real', NVL(sCalculation, N'-'),
N'OPCUA##OPCUATagValue_1min##Real####' || sTagName ||
'#############################################################Processvalue\' || sTagName ||
'#Value####Processvaluearchive\' || sTagName || '#Value'
FROM {TagsDefault}
```

2. TagSync w/ Writeback to OPC Server
   Dependencies: Global collections: **TagsDefault**; OPCUA connection group: **OPCUATagValue_1min** and **OPCTagValue Writeback_1min**; OPC Writeback needs to be enabled in XHQ.

```
SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'Real', NVL(sCalculation, N'-'),
N'OPCUA##OPCUATagValue_1min##Real####sTagName########################OPC##true##On####OPCTagValueW
riteback_1min##WB_' || sTagName || '########################Processvalue\' || sTagName ||
'#Value####Processvaluearchive\' || sTagName || '#Value'
FROM {TagsDefault}
```

4.4.10 Calculation Tags

1. TagSync w/o Writeback
   Dependencies: Global collections: **TagsDefault**.

```
SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'Real', NVL(sCalculation, N'-'),
N'CALC####Real####################(random_Real( ) + 12.3456)/2##1##12/24/2007 23:00:00##60000'
FROM {TagsDefault}
```

2. TagSync w/ Writeback to XHQ Historian
   Dependencies: Global collections: **TagsDefault**; PI Connection: **PITagValueWriteback_1min**.

```
SELECT sTagname, NVL(sDescription,N'-'), NVL(sUnits,N'-'), N'real', NVL(sCalculation, N'-'),
N'CALC####Real####################(random_Real( ) + 12.3456)/2##1##12/24/2007
23:00:00##60000##PI##true##On####PITagValueWriteback_1min##' || sTagName || '##5'
FROM {TagsDefault}
```

5 LOGGING AND TROUBLESHOOTING INFORMATION

The Log information about the XHQ Tag Sync process is available in the following files stored at the location specified by the environment variable %XHQ_SERVER_LOGS%:

Table 5.1 – Log Files

Log File Name	Description
tagsynclog1.out	Contains detailed information regarding the Solution Tag updates: ADD, DELETE and UPDATE events.
xhqci_s.out	Contains output information from the XHQCI commands executed during the Tag Sync processes.
XHQTagSync.out	Contains information of the steps executed by the XHQTagSync.BAT file.
XHQTagSyncSteps.out	Contains information of the steps executed by the XHQTagSync.BAT file.