

# Tatsoft Connectors

---

## FactoryStudio Real-time values

This connector is used to get real time values from FactoryStudio via the Tatsoft OPC DA server.

### Prerequisites

#### Verify Communication

To verify communication from Tatsoft's FactoryStudio, a generic OPC test client is provided to test the OPC Server.

This client is found on **XLReporter's** product CD under **Tools, OPC, OPC\_DA**. It can also be downloaded from [www.SmartSights.com](http://www.SmartSights.com).

To open, double-click **SampleClientDA.exe**. This opens the **OPC Quick Client** window.

To connect to an OPC server, select **Edit, New Server Connection** to open the **Server Properties** window. Select the Tatsoft OPC server and click **OK**.

Once the connection is made, select **Edit, New Group**. Specify **Name** and click **OK**.

Click on the group name created, and select **Edit, New Item**. This opens the **Add Items** window. Browse for tags and double click any to select. Once tag selection is complete click **OK** to return to the **OPC Quick Client** window.

All the selected tags appear along with their real time values, type, quality, and timestamp.

If at any point you experience an issue with this client, contact Tatsoft technical support to troubleshoot and correct these issues.

### Remote Communication

If XLReporter is not installed on the same machine as FactoryStudio, the workstation must also have the OPC core components installed. To determine if the core components are installed verify the following file exists:

- C:\Windows\SysWow64\OPCEnum.exe (64-bit OS)
- C:\Windows\system32\OPCEnum.exe (32-bit OS)

If the components are not installed, then they are provided in the XLReporter installation folder under *\_repairtools\OPC*. Alternatively, these can be downloaded from [www.opcfoundation.org](http://www.opcfoundation.org).

### Server Settings

To connect to FactoryStudio remotely both the machine where the server is running and the machine where the client is running must have matching Windows user accounts and the client must be logged in with a matching account.

In addition, on the machine with the OPC DA server, certain DCOM settings must be enabled. For details on what DCOM settings to enable, see [OPC and DCOM: 5 Things You Need to Know](#).

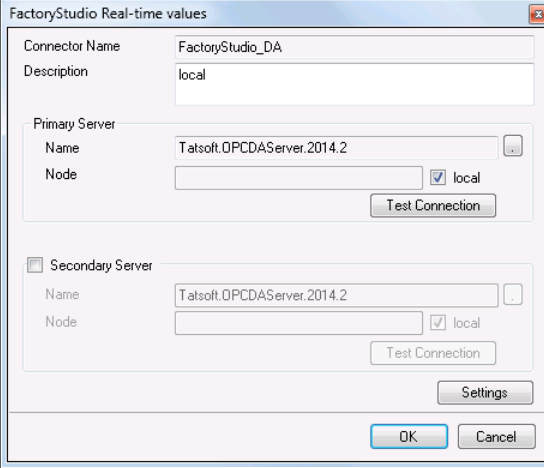
### Windows Firewall

If the Windows Firewall is enabled on the machine where FactoryStudio is running **TCP Port 135** must be opened for remote clients to connect.

## Connector

To configure the connector, from the **Project Explorer** select **Data, Connectors**.

- Click **Add**
- Select **Tatsoft, FactoryStudio Real-time values**
- Click **OK**



The screenshot shows a dialog box titled "FactoryStudio Real-time values". It contains the following fields and controls:

- Connector Name:** FactoryStudio\_DA
- Description:** local
- Primary Server:**
  - Name:** Tatsoft.OPCDAserver.2014.2 (with a browse button)
  - Node:** (with a dropdown menu and a checked "local" checkbox)
  - Test Connection:** button
- Secondary Server:** (checkbox is unchecked)
  - Name:** Tatsoft.OPCDAserver.2014.2 (with a browse button)
  - Node:** (with a dropdown menu and a checked "local" checkbox)
  - Test Connection:** button
- Settings:** button
- OK** and **Cancel** buttons at the bottom.

### Primary Server

These settings define the **Name** and **Node** of the OPC DA server. Use the browse button to browse the local or remote machine to select the Tatsoft OPCDA server.

Depending on the version of FactoryStudio installed the name of the server can vary. It is in the form:  
Tatsoft.OPCDAserver.Version

Use the **Test Connection** button to verify a connection to the server.

### Secondary Server

These settings define the (optional) secondary Tatsoft to connect to if a connection to the **Primary Server** fails.

### Settings

For information on the specific settings, see the **DATA CONNECTIVITY, OPC** document.

## Verify Data Communication

To verify communication to FactoryStudio, open the **Project Explorer** and select the **Tools** tab. Launch the **System Check** application.

- Click **Add**
- Choose the *FactoryStudio Real-time values* connector from the dropdown list
- Click the pushbutton ([...]) next to Items to open the **Tag Browser** window.
- Select one or more tags and click **OK**
- Click **Start** to verify the communication

---

## FactoryStudio Historical values

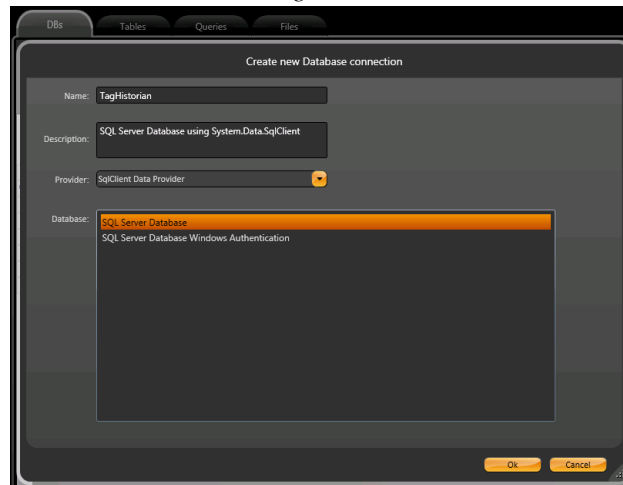
This connector is used to retrieve historical data logged to a relational database.

### Set up FactoryStudio

FactoryStudio logs historical data by defining the **TagHistorian** and a **Historian Table** to log. By default, the **TagHistorian** dataset is configured to log to an internal server. For third party applications to access historical data, the **TagHistorian** dataset must be configured to store data to a database like Microsoft SQL Server or MySQL.

To make this change, from the FactoryStudio menu, open the project. From the menu select **Edit, Datasets**.

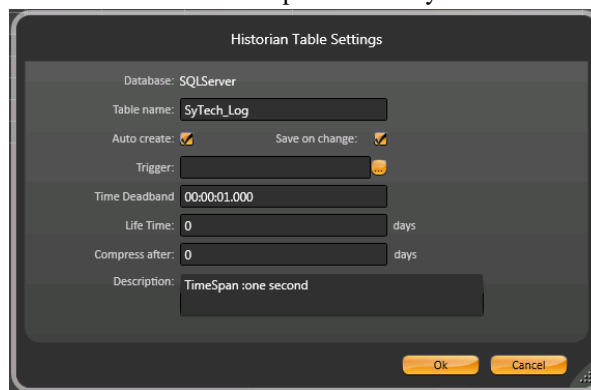
Rename the existing **TagHistorian** dataset. Then, on the **DBs** tab select **Create new....** When defining the new database connection, set the **Name** to *TagHistorian*.



### Define Tags to Log

From the menu select **Edit, Tags**. Under the **Objects** tab, specify each tag you wish to log.

On the **Historian** tab, click **New** to define a **Historian Table** to log. The **Life Time** setting controls how much data is preserved. It can be set to 0 to preserve every record.



### Prerequisites

#### Verify Database

To verify the database, open the tool provided by the database to view tables and columns. If this is a Microsoft SQL Server database, the SQL Server Management Studio is the tool to use.

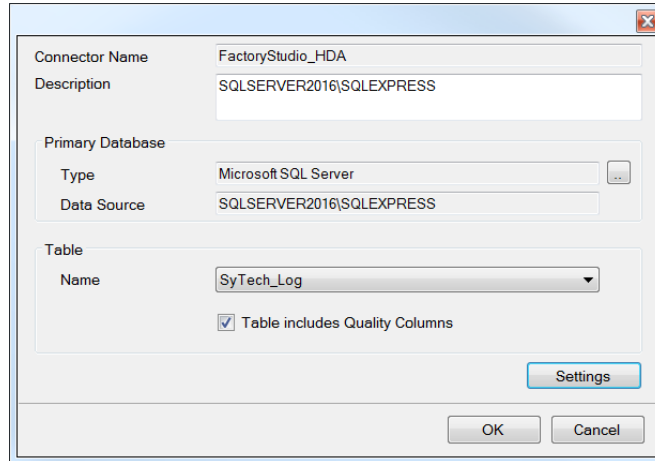
Once opened, locate the table that matches the **Table name** specified in FactoryStudio and select the *TOP 1000 rows*.

If you are experiencing issues connecting or retrieving data, contact Tatsoft technical support.

## Connector

To configure the connector, from the **Project Explorer** select **Data, Connectors**.

- Click **Add**
- Select **Tatsoft, FactoryStudio Historical values**
- Click **OK**



The screenshot shows a dialog box for configuring a connector. The fields are as follows:

Connector Name	FactoryStudio_HDA
Description	SQLSERVER2016\SQLEXPRESS
Primary Database	
Type	Microsoft SQL Server
Data Source	SQLSERVER2016\SQLEXPRESS
Table	
Name	SyTech_Log
<input checked="" type="checkbox"/> Table includes Quality Columns	

Buttons: Settings, OK, Cancel

### Primary Database

This defines a connection to the database where the **FactoryStudio Historian** is logging data. A browse button [...] is provided to define.

### Table

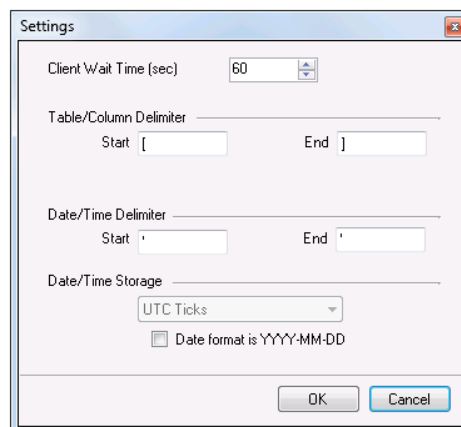
After connecting to the **Primary Database** where FactoryStudio is logging the historical data, select the **Table Name** where the data is being logged.

If quality is logged along with the values, the **Table includes Quality Columns** should be checked so that quality can be factored in on any calculations performed.

Note, if FactoryStudio logs data to multiple tables, each table must be set up as its own connector.

### Settings

The **Settings** button opens the **Settings** dialog that defines characteristics of the database that are used to retrieve data.



The screenshot shows a dialog box for settings. The fields are as follows:

Client Wait Time (sec)	60
Table/Column Delimiter	
Start [	End ]
Date/Time Delimiter	
Start '	End '
Date/Time Storage	
UTC Ticks	
<input type="checkbox"/> Date format is YYYY-MM-DD	

Buttons: OK, Cancel

Typically, these settings are defaulted correctly based on the **Primary Database**.

If queries timeout, increase the **Client Wait Time**.

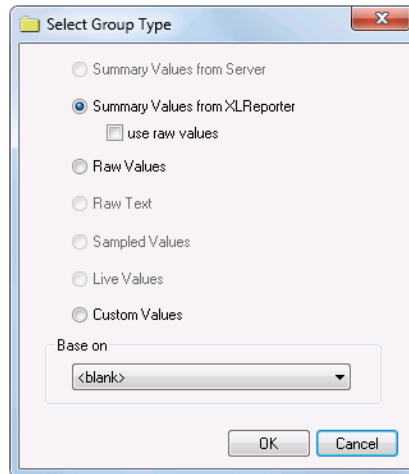
The delimiter and timestamp settings are typically filled in automatically for the database and can be modified for other databases.

The **Date/Time Storage** settings define how timestamps are stored in the database.

## Data Group

The following describes the historical data group settings specific to the **FactoryStudio Historical Values** connector.

### Group Types



For **FactoryStudio Historical Values** the following group types are available:

#### **Summary Values from XLReporter**

This group type retrieves sampled values from the database and performs calculations on those samples for reporting.

By default, summary values are calculated time weighted, and values are propagated based on the last known value. However, to change this so that summary values are calculated strictly on the data returned check **use raw values**.

#### **Raw Values**

This group retrieves every value logged to the database between the start and end time specified.

#### **Custom Values**

This option opens the Database Group builder where a query can be configured to retrieve data from any table in the database.

## Group Settings

### Setup Tab

#### Retrieval (Summary Values for XLReporter Group)

The **Retrieval** settings define how data is retrieved for the calculations selected for the group. The following settings are available:

- **Retrieval Mode**  
This setting defines how data is retrieved from the historian. For FactoryStudio only *Raw Values* are available.
- **Lead Time**  
The amount of time (in seconds) to retrieve data before the start time.

### Filters Tab

If the **Perform by Server** option is checked, any filter configured in this tab is put into the *WHERE* clause of the query sent to the database to retrieve data for the group. Otherwise, the configured filtering is performed by the reporting engine after the values are returned. It is recommended to leave this setting checked as the performance is much better.

## Verify the Data Connector

From the **XLReporter Project Explorer** select, **Tools, Connector Groups**

Select the *FactoryStudio Historical values* connector and then select **Add**.

- Set the **Type Raw Values** and click **OK**.

On the **Columns** tab of the group, select the tag **Name(s)**.

Select **Preview**, pick a *Start* date and click **Refresh**.

---

## FactoryStudio Alarms

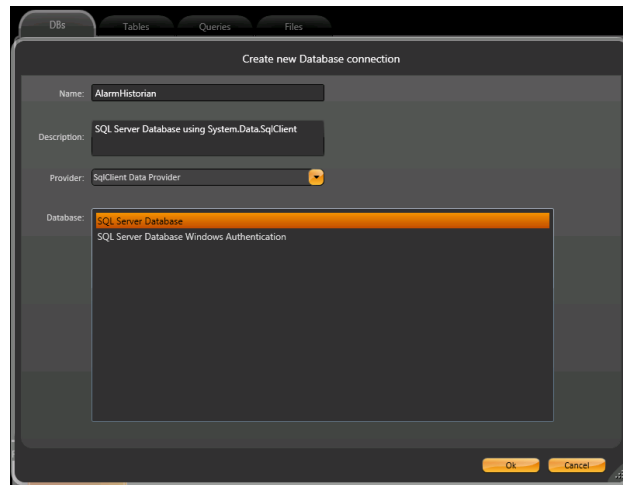
This connector is used to retrieve alarm data logged to a relational database.

### Set up FactoryStudio

FactoryStudio logs Alarm data by defining the **AlarmHistorian** dataset. Just like with historical data, by default, alarm data is logged to an internal server. To make alarm data accessible to third party applications, the AlarmHistorian dataset must be redefined to log to a database such as Microsoft SQL server or MySQL.

To make this change, from the **FactoryStudio** menu, open the project. From the menu select **Edit, Datasets**.

Rename the existing **AlarmHistorian** dataset. Then, on the **DBs** tab, select **Create new....** When defining the new database connection, set the **Name** to *AlarmHistorian*.



Alarms can then be configured by selecting **Edit, Alarms**.

### Prerequisites

#### Verify Database

To verify the database, open the tool provided by the database to view tables and columns. If this is a Microsoft SQL Server database, the SQL Server Management Studio is the tool to use.

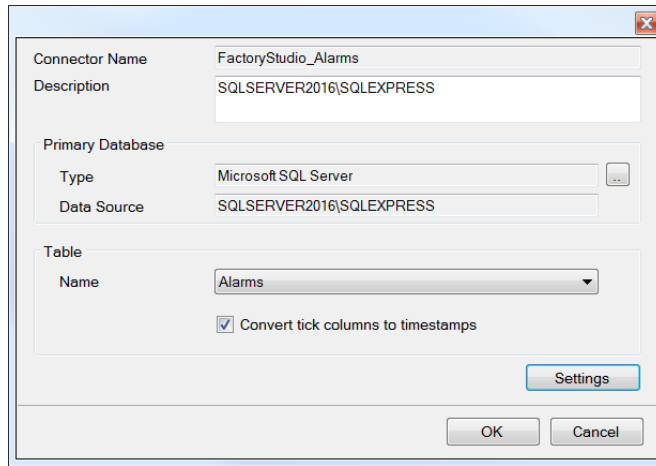
Once opened, locate the table that matches the **Table name** specified in for alarms FactoryStudio and select the *TOP 1000 rows*.

If you are experiencing issues connecting or retrieving data, contact Tatsoft technical support.

### Connector

To configure the connector, from the **Project Explorer** select **Data, Connectors**.

- Click **Add**
- Select **Tatsoft, FactoryStudio Alarms**
- Click **OK**



### Primary Database

This defines a connection to the database where the **FactoryStudio** is logging alarms. A browse button [...] is provided to define.

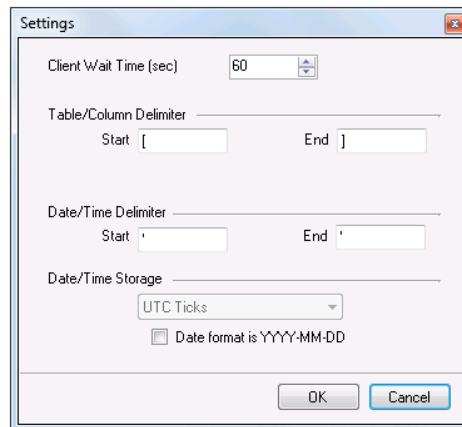
### Table

After connecting to the **Primary Database** where FactoryStudio is logging alarms, select the **Table Name** where the alarms are being logged.

In the alarms table, there are many “time” columns that are logged as UTC ticks which may not be readable to a user. To convert these columns to show as local timestamps, check the **Convert tick columns to timestamps** setting.

### Settings

The **Settings** button opens the **Settings** dialog that defines characteristics of the database that are used to retrieve data.



Typically, these settings are defaulted correctly based on the **Primary Database**.

If queries timeout, increase the **Client Wait Time**.

The delimiter and timestamp settings are typically filled in automatically for the database and can be modified for other databases.

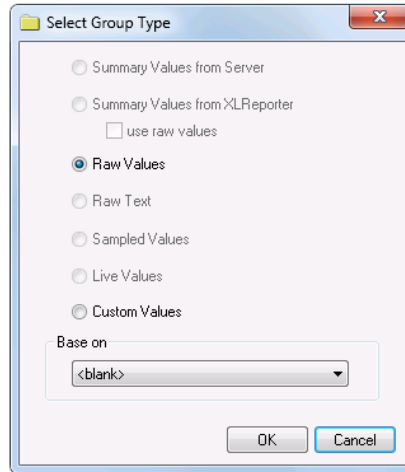
The **Date/Time Storage** settings define how timestamps are stored in the database.



## Data Group

The following describes the historical data group settings specific to the **FactoryStudio Alarms** connector.

### Group Types



For **FactoryStudio Alarms** the following group types are available:

#### **Raw Values**

This group retrieves every alarm logged to the database between the start and end time specified.

#### **Custom Values**

This option opens the Database Group builder where a query can be configured to retrieve data from any table in the database.

### Group Settings

#### **Filters Tab**

If the **Perform by Server** option is checked, any filter configured in this tab is put into the *WHERE* clause of the query sent to the database to retrieve data for the group. Otherwise, the configured filtering is performed by the reporting engine after the values are returned. It is recommended to leave this setting checked as the performance is much better.

## Verify the Data Connector

From the **XLReporter Project Explorer** select, **Tools, Connector Groups**

Select the *FactoryStudio Alarms* connector and then select **Add**.

- Set the **Type Raw Values** and click **OK**.

On the **Columns** tab of the group, select the columns to show alarm data.

Select **Preview**, pick a *Start* date and click **Refresh**.

Information in this document is subject to change without notice. SmartSights, LLC assumes no responsibility for any errors or omissions that may be in this document. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the prior written permission of SmartSights, LLC.

Copyright 2000 - 2024, SmartSights, LLC. All rights reserved.

XLReporter® is a registered trademark of SmartSights, LLC.

Microsoft® and Microsoft Excel® are registered trademarks of Microsoft, Inc.  
All registered names are the property of their respective owners.