

Description: This application note describes how to configure and use the TrendWorX32 SQL Query Engine.

OS Requirement: Windows XP / Windows Vista / Windows 7 / Windows 8 / Windows Server 2003 / Windows Server 2008 / Windows Server 2008 R2 / Windows Server 2012

General Requirement: GENESIS32 version 9.2 or later.

Prerequisites

In order to use the TrendWorX32 SQL Query Engine it is necessary to have an active TrendWorX32 Logger configuration and some logged data. See the “TrendWorX32 Logger – Data Logging Quick Start” application note for information on how to create a TrendWorX32 Logger configuration.

If you would like to query an instance of TrendWorX32 that is running on another computer please make sure GenBroker is running and properly configured on that computer. The TrendWorX32 SQL Query Engine uses GenBroker communication to speak with remote TrendWorX32 OPC HDA servers. For more information on how to configure GenBroker, please see the application notes on that subject.

Introduction

The TrendWorX32 Logger records data into a database, but the format of this database is not easy to query. It was optimized for space, not for ease of access. The TrendWorX32 SQL Query Engine allows you to use common SQL Data Manipulation Language (DML) queries to retrieve and edit data. This data can then be displayed in any SQL client, like Microsoft Excel, the ICONICS Data Mining OPC Server or Microsoft SQL Management Studio.

The following clients can use the TrendWorX32 SQL Query Engine (this is not an exhaustive list):

- ReportWorX Configurator
- Data Mining Configurator
- MS Excel (2003, 2007)
- MS SQL Server (2005, 2008)
- MS OLE DB RowsetViewer

Using the Data Mining Configurator

This example will show you how display a list of tags from your logging groups in GraphWorX32 or another OPC-DA client using the Data Mining Configurator. The example assumes you are using the default “VCRCConfiguration” TrendWorX32 configuration database. If you are using a different configuration you may need to change some of the names in the example.

Configuring the Data Mining Configurator

1. Open the Data Mining Configurator by going to Start → Programs → ICONICS Tools → Data Mining Configurator.
2. Right click on Databases and select New → Database Connection.
3. Type “TWX SQL” into the Connection Name field.
4. Set the Connection Type to “Generic OLE DB data source”.
5. Click on the Advanced button
6. Select first tab (Provider) and select “ICONICS TrendWorX32 SQL Query Engine”.
7. Click Next.
8. Fill in the Data Source with the name of the computer where your TrendWorX32 Logger is located. If it is on the same machine as your Data Mining Configurator you can leave the Data Source field blank.
9. Select your database group from the initial catalog dropdown and your logging group from the initial schema dropdown (these lists may take a few seconds to populate).
10. Click OK, then in the Data Mining Configurator click Apply.
11. Double-click on “TWX SQL” in the tree to expand it.
12. Right click on Data Sources and select New → Datasource.
13. Type “Command_Tags” into the Name field.
14. In the Specify Dataset section, select SQL Command/Stored Procedure and check Enable Multi-rows Dataset.
15. Click the Edit button.
16. Enter this SQL query into the SQL Command Editor dialog, then click OK, then Apply:

```
SELECT *
FROM "VCRWorX32 Group"."VCRWorX32 Logging Group".Tags
```

NOTE: If you are not using the example VCRCConfiguration replace “VCRWorX32 Group” with the name of your database group and “VCRWorX32 Logging Group” with the name of your logging group.

17. Click the Test SQL button then click Test SQL Query. You should see the list of tags in that logging group. You can click Close when done.

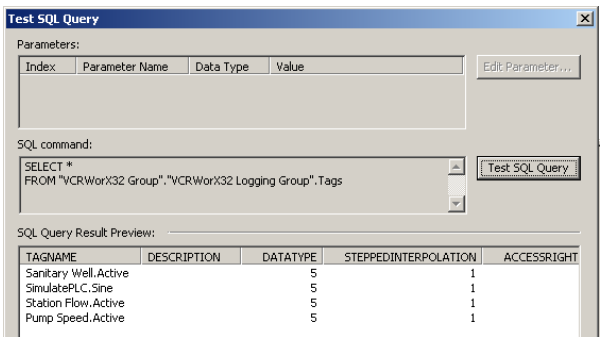


Figure 1 - Testing the SQL Query

Getting Data in GraphWorX32

1. Start GraphWorX32 by going Start → Programs → ICONICS GENESIS32 → GraphWorX32 → GraphWorX32.
2. In the ActiveX toolbar click on the ICONICS Data Mining Control . This will insert an ActiveX into your display.
3. Double-click on the Data Mining ActiveX in the display to open its properties.
4. Set Data Source Type to Data Mining.

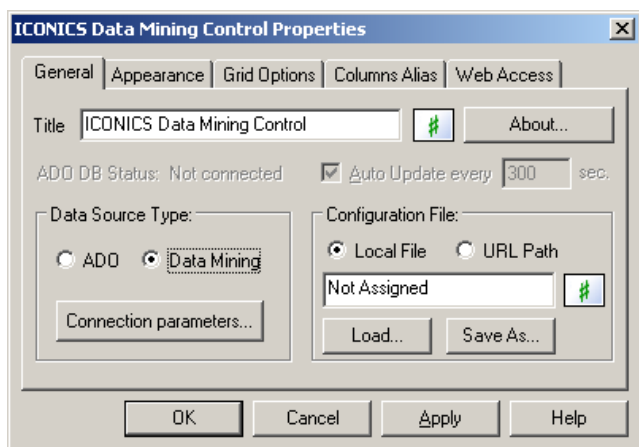


Figure 2 - Data Mining Control Properties

5. Click the Connection Parameters button.
6. Click the Add Tags button.
7. Select the Database Access tab.
8. Expand Databases → TWX SQL → Command_Tags, double-click on @@Recordset, then click OK.
9. Click OK until you are back in GraphWorX32.
10. Go to runtime. After a few seconds you should see the list of tags from your logging group.

ICONICS Data Mining Control				
	TAGNAME	CRIP	TATY	INTER
1	Sanitary Well.Active	5	1	
2	SimulatePLC.Sine	5	1	
3	Station Flow.Active	5	1	
4	Pump Speed.Active	5	1	

Figure 3 - Results in GraphWorX32

Totalizer over Data Mining

This section will describe how to use the Data Mining Configurator and the TrendWorX32 SQL Query engine to

create a process point that displays totalized data. It assumes you have completed the previous Data Mining example.

1. In the Data Mining Configurator, expand to Databases → TWX SQL.
2. Right-click on Data Sources and select New → Datasource.
3. Name the new data source “Totalizer”.
4. Select “SQL Command/Stored Procedure”.
5. Check the Enable Multi-rows Dataset.
6. Click the Edit button.
7. Enter the following query into the box, then hit OK:

```
SELECT SUM(value)
FROM "VCRWorX32 Group"."VCRWorX32 Logging
Group".RawData
WHERE tagname = ?
AND timestamp >= dateadd(mi, -5, NowUTC)
AND timestamp <= NowUTC
```

8. Click the Add button to add a parameter.
9. Name the parameter “Tagname” and keep the type as VB_BSTR.
10. Hit OK, then Apply. Your data source should look similar to the figure below.

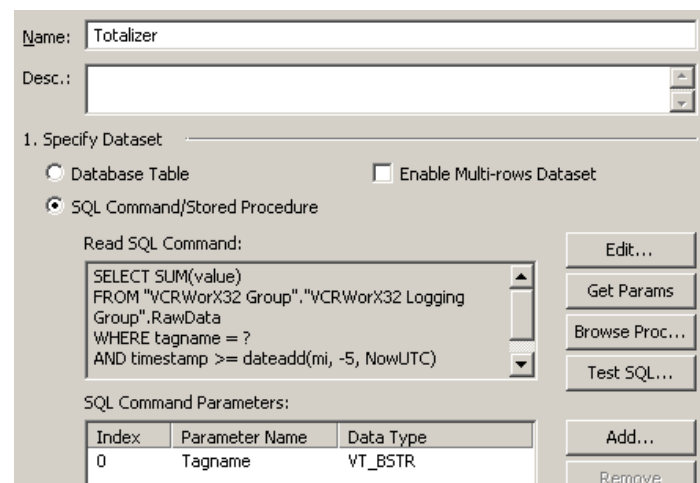


Figure 4 - Totalizer Data Source

11. Click the “Test SQL” button to test the query.
12. Double-click the Tagname parameter and enter a value of “SimulatePLC.Sine”, then click OK.
13. Hit the Test SQL Query button. You should see a result similar to the figure below.

NOTE: You may want to change the collection rate of your logging group when doing this test. The Sine tag changes in such a way that it’s easy to get a 0 sum over a period of time. If your collection rate is set to something uneven, like 1 second and 5 miliseconds, you’re more likely to get a nonzero sum.

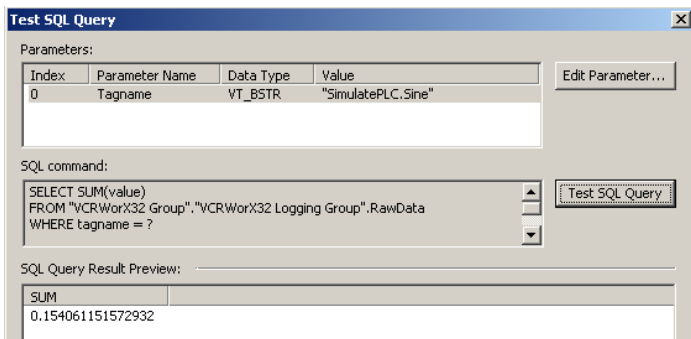


Figure 5 - Query Test Results

Creating a Data Item

This optional section describes how to create a data item, which is a way to hard-code the value of the Tagname parameter for the data source we created above.

1. Right-click on the Totalizer data source and select Create New Data Item. Name it “Sine”
2. Double-click on the Tagname parameter and fill in a value of “SimulatePLC.Sine”, then hit OK.
3. Hit the “...” button near Column Name and pick the Sum item.
4. Hit Apply.

Getting Totalizer into Data Mining ActiveX

1. Start GraphWorX32 by going Start → Programs → ICONICS GENESIS32 → GraphWorX32 → GraphWorX32.
2. Click the Process Point button (), or go to Dynamics → Intrinsic → Process Point, then click in the display to add a process point.
3. Click the Data Tags button.
4. Go to the Database Access tab and click once on Databases → TWX SQL → Data Sources → Totalizer → @@Column (don’t double-click). This path should appear in the unified data browser:

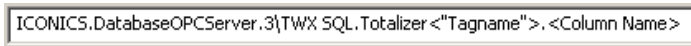


Figure 6 - Path for @@Column Item

5. Replace “Tagname” with “SimulatePLC.Sine and replace “Column” with “SUM”. Your path should now look like this:

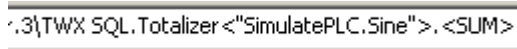


Figure 7 – Part of the Finished Path

6. Click OK to all dialogs.

7. If you completed the “Creating a Data Item” optional steps above, add another process point, but this time browse for Databases → TWX SQL → Data Sources → Totalizer → Sine.
8. Go into runtime. Both process points should show the same value, which is the sum of the samples collected for SimulatePLC.Sine in the last five minutes.

Using the SQL Query Engine as a Linked Server

This section will guide you through adding the TrendWorX32 SQL Query Engine as a linked SQL server.

Setting DCOM permissions

1. Go to Start → Run and type “dcomcnfg”
2. Expand Component Services → Computers → My Computer → DCOM Config.
3. Right-click on ICONICS Raw Data Provider Host and select Properties.
4. Go to the Security tab and click on the first Edit button.
5. Click Add, then Advanced.
6. If the “From this location” field does not say your local computer name, click on Locations and pick your local computer.
7. Click Find Now.
8. Double-click on Network Service then hit OK.
9. Make sure the Allow box is checked for Local Launch and Local Activation then hit OK.
10. Repeat steps 3-8 for the next Edit button, but this time make sure the Allow box for Local Access is checked.
11. Click OK to all dialogs.
12. Go to the Processes tab of your Task Manager and kill the RawDataProviderHost.exe process if it was running. Alternately, you could restart your machine.

Adding linked server

If you do not have Workbench32 installed please see the application note entitled *TrendWorX32 Logger - Adding a Linked Server to SQL Server*. If you do have Workbench32 installed, follow these steps to easily add your linked server.

1. Start Workben32 from Start → Programs → ICONICS Workbench32 → ICONICS Workbench32
2. Select the TrendWorX32 Logger provider

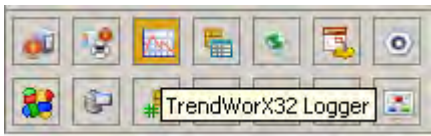


Figure 8 - TrendWorX32 Logger Provider

3. Click the SQL Query Engine Setup button in the toolbar.



Figure 9 - SQL Query Engine Setup Button

4. In section 1, type in or select the name of the SQL Server where you want to create the Linked Server and fill in the credentials.
5. In section 2, give a name for your linked server.
6. In section 3, select or type in the name of the machine where the TrendWorX32 HDA server is running.

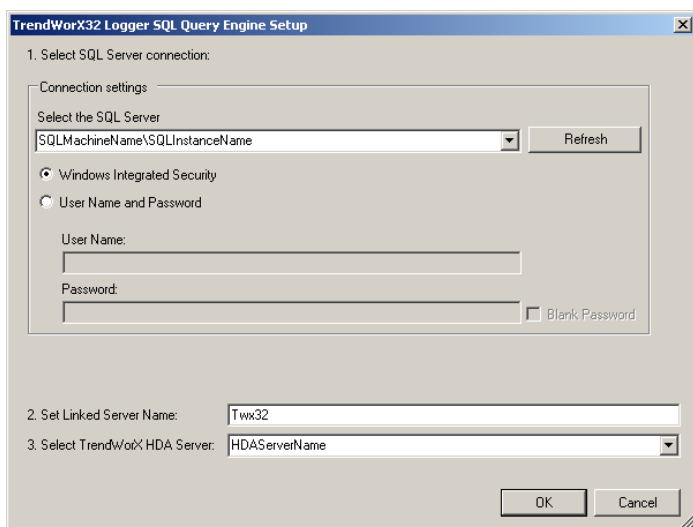


Figure 10 - TrendWorX32 Logger SQL Query Engine Setup window

7. Click OK when you are done. This may take a moment. You will receive a message when the process is complete. You should now have a linked server available in your SQL Server Management Studio.

Building SQL queries

Getting Basic Information

To get a list of all of the database groups (catalogs) in your configuration, run this query:

```
EXEC sp_catalogs Twx32
```

To get a list of available tables you can query, run the query below:

```
EXEC sp_tables_ex Twx32
```

Using OPENQUERY

For the fastest performance it's recommended to use the OPENQUERY function to query the linked Twx32 server. When using the OPENQUERY function your queries should look something like this:

```
SELECT *
FROM OPENQUERY (Twx32,
'SELECT *
FROM "VCRWorX32 Group"."VCRWorX32 Logging
Group".RawData
WHERE tagname = 'SimulatePIC.Sine'
AND timestamp >= dateadd(mi, -5, NowUTC)
AND timestamp <= NowUTC')
```

You can write your queries directly, but the performance may be slower, you may not have access to some of the special keywords used by the SQL Query Engine (like "NowUTC"), you will need to start all of your table names with "Twx32.", and some of the columns will need to be converted to different data types.

For these reasons we highly recommend using the OPENQUERY syntax given above. The rest of this application note will assume you are using the OPENQUERY function.

Tags and RawData Tables

When writing SQL queries for your Twx32 linked server you will usually be querying one of these two tables:

```
DBGroup.LogGroup.Tags
DBGroup.LogGroup.RawData
```

"DBGroup" and "LogGroup" will be the names of your database group and logging group, respectively. These must be surrounded by square brackets if their names contain spaces.

The **Tags** table will contain all of the tags that are in the designated logging group. For example, to get all of the tags in a logging group named "VCRWorX32 Logging Group" which is in a database group named "VCRWorX32 Group" you would use this query:

```
SELECT *
FROM OPENQUERY (Twx32,
'SELECT *
FROM Twx32."VCRWorX32 Group"."VCRWorX32
Logging Group".Tags')
```

The **RawData** table contains the raw logged data for the tags in the designated logging group. It contains these columns:

- Tagname
- Timestamp
- Quality
- Value



TrendWorX32 Logger – SQL Query Engine Quick Start



APPLICATIONS NOTE

October 2013

As an example, to get all data logged for the SimulatePLC.Sine tag between January 1st and February 1st you'd use this query:

```
SELECT *
FROM OPENQUERY (Twx32,
'SELECT *
FROM "VCRWorX32 Group"."VCRWorX32 Logging
Group".RawData
WHERE Tagname = 'SimulatePLC.Sine'
AND Timestamp > '2010-01-01'
AND Timestamp < '2010-02-01''')
```

It's recommended that you always include some sort of Timestamp filter in your queries to avoid creating very large queries that take a long time to process.

Reading the Quality

The Query column in the RawData table is a number that may not be immediately readable to a user. There are several functions that will translate this number into something more meaningful.

QualityToStr() returns "Good", "Bad", or "Uncertain" based on the inputted quality value. **IsGood()**, **IsBad()**, and **IsUncertain()** all return true or false based on whether the input represents that quality type.

This example builds on the last example above except it renders the query column readable to a user.

```
SELECT *
FROM OPENQUERY (Twx32,
'SELECT TAGNAME, TIMESTAMP,
QualityToStr(QUALITY) AS QUAL, VALUE
FROM "VCRWorX32 Group"."VCRWorX32 Logging
Group".RawData
WHERE Tagname = 'SimulatePLC.Sine'
AND Timestamp > '2010-01-01'
AND Timestamp < '2010-02-01''')
```

NOTE: You can find more information about the syntax and more examples in the product documentation.

Using the SQL Query Engine in other ICONICS Applications

You can find more information about using the SQL Query Engine in application notes such as *TrendWorX32 - Using SQL Query Engine in BridgeWorX and ReportWorX* and *TrendWorX32 - Using SQL Query Engine in ScriptWorX2010*.