

Description: Guide to setup a ReportWorX report to retrieve Hyper Historian data.

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

General Requirement: Installation of ReportWorX and Hyper Historian; logged Hyper Historian data available for query

Introduction

There are three ways to get Hyper Historian data in to a ReportWorX report. First, built into Hyper Historian is an HDA server and you can use the HDA Data Access Tag connection. The second way is to use Hyper Historian SQL provider in Open Database connection, which allows you to use common SQL queries to retrieve data. Finally, you can use the Hyper Historian OLEDB provider directly within Excel.

This application note will provide details on how to retrieve data in all three methods mentioned above. Before we can show you how to retrieve Hyper Historian data using ReportWorX, please make sure that Hyper Historian is running, and/or you have some logged data for retrieval. You can use the sample configuration if it will help. You should also create new Report and new empty Report Template in the ReportWorX Configurator.

HDA Data Access Tag

Hyper Historian is an HDA server and this allows us to browse OPC HDA tags in a report template.

1. Open your newly created Report Template
2. Select some area of cells and right-click in it
3. In the popup context menu, select Connect to a new Data Source → OPC HDA → HDA Data Access (DA) Tag

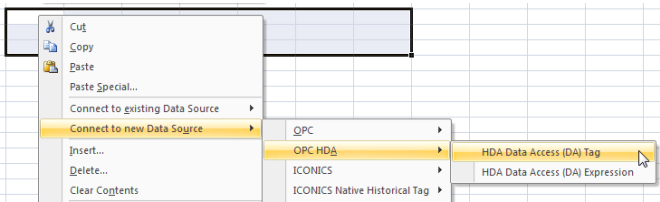


Figure 1 - Using HDA to get Hyper Historian Data

4. In the OPC HDA Connection Wizard, click on the browse button 
5. In the opened Unified Data Browser, double-click on the Hyper Historian HDA server and drill down the data logging groups to select the history tag(s) of your choice.

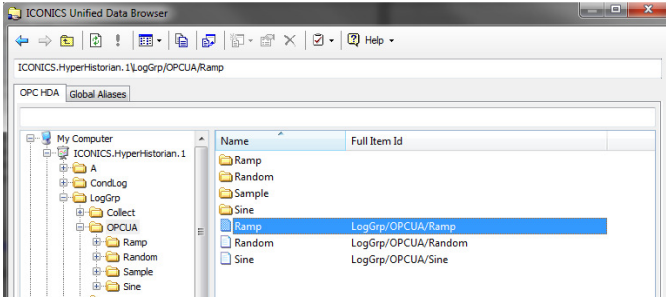


Figure 2 - Selecting an HDA Tag from the Hyper Historian HDA Server

6. Go through the OPC HDA Connection Wizard until you see the Tag group properties window as shown in Figure 3.

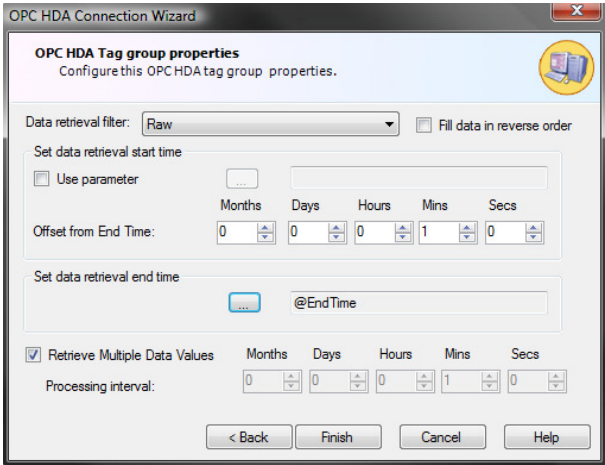


Figure 3 - Tag Group Properties

7. Set the Data Retrieval filter to 'Raw'
 8. Uncheck the Set data retrieval start time checkbox and set the Offset from End time field to 1 minute
 9. Click on the Set Data retrieval end time browse button '...' to create a new parameter that will specify to which date and time the report will retrieve data.
 10. In the Parameter Browser, click on the 'Add' button
 11. Change the name of the parameter to "EndTime"
 12. Select 'Automatically evaluate a date time' as the data type of the parameter
 13. Click on the Configure Parameter Details button and select 'Today' in the new window that appears
 14. Save and select the parameter by clicking on "OK" a few times.
- NOTE:** At this point, your window should look similar to Figure 3.
15. Click on finish to complete the data source creation.
 16. You should now see the Layout manager. Depending on the number or columns you highlighted when you created the data source, you may have different columns available. You can feel free to add or remove columns as necessary and assign the proper values from a data source to a column by

dragging it from the tree on the left hand side and dropping it in the desire cell. Once you are done, click on “OK” to save the changes and exit the Layout Manager.

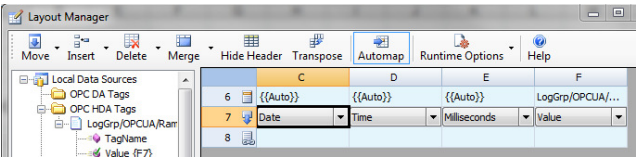


Figure 4 - Layout Manager

17. Save your report template and assign the template to a report
18. Once you have executed the report, open it and your results should display historical data for the last minute from when you executed the report.

Date	Time	Milliseconds	LogGrp/OPCUA/Ramp
6/11/2010 0:00	4:25:07 PM	492	28.624
6/11/2010 0:00	4:25:07 PM	993	29.624
6/11/2010 0:00	4:25:08 PM	555	30.75
6/11/2010 0:00	4:25:09 PM	118	31.874
6/11/2010 0:00	4:25:09 PM	680	33

Figure 5 - Hyper Historian Logged Data in a Report

Using an Open Database Connection

We can use the Hyper Historian SQL provider to retrieve Hyper Historian data as well. We'll need to connect to the Hyper Historian database using an Open Database connection in the report template and write an SQL query to retrieve the data.

Before we start editing the report template, we need to prepare the SQL queries. It is impossible to use the Simple or Advanced Query Wizard from the ReportWorX template because we do not have data stored in database tables; the data is actually in a proprietary format. We can only use the SQL provider to make a connection to the linked server, which will then allow us to query Hyper Historian data. For more details about the Hyper Historian data format and how it's possible to query data through SQL, please refer to the Hyper Historian help files.

We will not go through the details on how to create SQL queries for retrieving data from Hyper Historian in this document. If you need more information on this subject, please refer to the Application Note entitled *Hyper Historian – SQL Query Engine Quick Start*.

For an example in this document, we will assume that we have an SQL query similar to the one below:

```
select * from openquery(HH2, 'select * from
LogGrp.OPCUA.rawdata where tagname = ''Ramp''')
```

Now that we have this information, we will start to edit the report template.

1. Open a newly created Report Template and select some area of cells
2. Right-click on the selected cells and choose Connect to New Data Source → Open Database in the popup context menu.

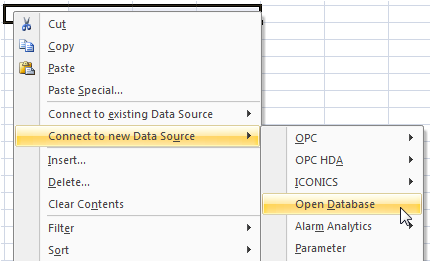


Figure 6 - New Open Database Data Source

3. Click on the ‘Browse’ button in Database Connection Configuration Wizard dialog
4. Right-click on ‘SQL Server Connections’ and select the ‘Insert New Connection’ option
5. Create a connection to your Hyper Historian configuration database.

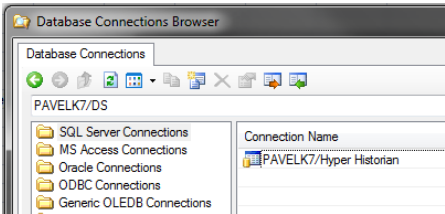


Figure 7 - SQL Server Connection to the HH Configuration Database

6. Select the newly created connection and click on “OK”. This should bring you back to the Data Source Configuration Wizard.
7. In the Database Query dialog on the next screen, click on ‘Edit’ and select ‘Custom Query’

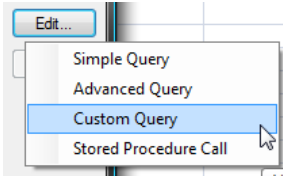


Figure 8 - Custom Query

8. Copy and Paste the SQL query that you created before editing template into the Custom Query dialog
9. Click OK on information dialog
10. Click on the ‘Test’ button to see if you are getting historical data. Once you are done testing, click on Next



ReportWorX - Reporting on Hyper Historian Data



APPLICATIONS NOTE

July 2011

11. You can change the Data Source name if you wish, when you are done click on the Finish button
12. In the Layout Manager, select the columns of data that you are interested in displaying.

	J	K	L	M
6	{{Auto}}	{{Auto}}	{{Auto}}	{{Auto}}
7	TAGNAME	TIMESTAMP	QUALITY	VALUE
8				

Figure 9 - Layout Manager

13. When you are done configuring the layout of the data, click on OK to close Layout manager.
14. Save your report template and exit Excel.
15. Assign the template to a report and execute the report
16. When you open the executed report, you should see the historical data for the queried tag.

TAGNAME	TIMESTAMP	QUALITY	VALUE
Ramp	6/8/2010 15:17	2157707264	
Ramp	6/10/2010 14:33	2157641728	
Ramp	6/10/2010 14:33	0	77
Ramp	6/10/2010 14:33	0	78.124
Ramp	6/10/2010 14:33	0	79.124
Ramp	6/10/2010 14:33	0	80.25
Ramp	6/10/2010 14:33	0	81.25
Ramp	6/10/2010 14:33	0	82.374
Ramp	6/10/2010 14:33	0	83.5
Ramp	6/10/2010 14:33	0	84.624

Figure 10 - HDA Data in Report

SQL Query with ReportWorX Parameters

You can define your SELECT SQL query with a WHERE condition for Timestamp, and with this condition specify the time frame for retrieving Hyper Historian data.

```
select * from openquery(HH2, 'select * from LogGrp.OPCUA.rawdata where tagname = ''Ramp''') WHERE timestamp between convert(datetime, '6/09/2011 2:33:00 PM') and convert(datetime, '6/10/2011 2:33:00 PM') ORDER BY timestamp DESC
```

NOTE: It is not possible to use 'between' command or 'order by' command inside red select. This select is sent to HH and needs to be translated by HH. These commands are not implemented in v 9.2. The query like this can slow down the performance.

Additionally you can use ReportWorX parameters for this improved SQL query and bind parameter to Timestamp values. With this binding, you'll be able to define time frame of your Hyper Historian HDA data dependently on report start.

```
select * from openquery (HH2, 'select * from LogGrp.OPCUA.rawdata where tagname = ''Ramp''') WHERE timestamp between convert(datetime, @StartTime) and convert(datetime, @EndTime) ORDER BY timestamp DESC
```

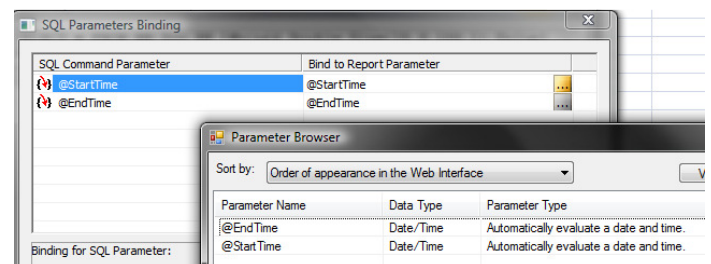


Figure 11 - Parameter Mapping with SQL Query