GridWorX Viewer
Displaying Hyper Historian Data

APPLICATION NOTE

August 2014

Description: Displaying Hyper Historian Data in GridWorX Viewer.
General Requirement: GENESIS64 and Hyper Historian v10.7x or later installation, familiarity with Hyper Historian and GENESIS64, familiarity with the Hyper Historian SQL Query Engine.

Introduction

GridWorX Viewer was developed in order to visualize database data or OPC DA data through a data grid control. GridWorX64 can also read historical data stored in Hyper Historian database using the Hyper Historian OLE DB provider and SQL Query Engine.

This application note provides an example of how to read Hyper Historian data in a GridWorX Viewer. This application note also assumes that you have followed though the example in the Application Note titled *GENESIS64 – GridWorX64 Server, GENESIS64 – GridWorX64 Viewer*.

Before You Start

1. Before you start to read the data from Hyper Historian database make sure that Hyper Historian is properly set up and you are logging the data properly. If you have never configured Hyper Historian, you can start with the application note entitled, *Hyper Historian - Quick Start*.
2. Double check that Hyper Historian Core service is started (via traffic light or service manager). Please notice that the Hyper Historian engine has to be running to read the data from Hyper Historian database using the SQL Query Engine. For more details how to use SQL Query Engine see the related application note, *Hyper Historian - SQL Query Engine Quick Start*.
3. You should load the query from the stored procedure below into SQL Server Management Studio and test it to make sure that is accurate for your configuration. You can test it in Microsoft SQL Server Management Studio. For more details how to work with stored procedures in SQL Server see your SQL Server documentation or related application notes.

The stored procedure below needs to be created in your SQL Server (in the Hyper Historian configuration database). We will call this stored procedure from GridWorX64 Server to get our data.

NOTE: This stored procedure can actually be added into any database on the same server as the Hyper Historian Linked Server, not just the Hyper Historian configuration database, but the Hyper Historian configuration database is usually the best choice because it will be easier to find later if you need to make changes.

```
/****** Copyright ICONICS Inc. *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE PROCEDURE [dbo].[sp_GetRawData_perTag_perTimePeriod]
        @StartTime varchar(50),
        @EndTime varchar(50),
        @FolderName varchar(100),
        @TagName varchar(1000)
AS
SET NOCOUNT ON;
DECLARE @OpnQ varchar (1000)
DECLARE @SQL nvarchar(3000)

SET @OpnQ = 'SELECT [TAGNAME]
        ,[TIMESTAMP]
        ,[QUALITY]
        ,[VALUE]
        FROM ' + @FolderName + '.[RAWDATA]
        WHERE TAGNAME='''''+@TagName+''''' AND
TimeStamp >= '''''+@StartTime+''''' AND Timestamp <='''''
+@EndTime+''''''


SET @SQL = 'SELECT * FROM OPENQUERY (HH2,''' + @OpnQ+''')'

EXEC sp_executesql @SQL

GO
```

Connecting GridWorX64 Server to the Hyper Historian Database

1. Open up Workbench by going to **Start → All Programs → ICONICS → Workbench.**
2. In the Project Explorer, choose the **GridWorX64 Server** provider.
3. Under the GridWorX64 configuration database, right click on **SQL Server Connections** and click on **New Database Connection.**
4. Provide a name for the connection such as "Hyper Historian" and click on **Change Connection**.
5. Double click on SQL Server Connections and click on the button to open up Connection String dialog box. Connect to your Hyper Historian configuration database. Click OK when finished.
6. Back in the Connection Strings dialog, select the connection you just created and click OK.
7. Click "Test connection" to check if you can connect to database, then click Apply.

Configuring GridWorX64 Data Source

1. Right-click on the database connection you created and choose **New Data Source**.
2. Name this data source **GetRawData.**
3. In SELECT command tab, click on the pencil icon and choose **Stored Procedure Call.**
4. Browse for stored procedure which you created in the "Before You Start" section. It should be named "sp_GetRawData_perTag_perTimePeriod".

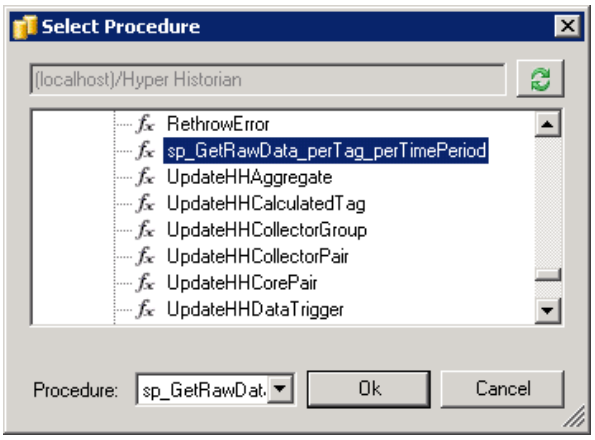
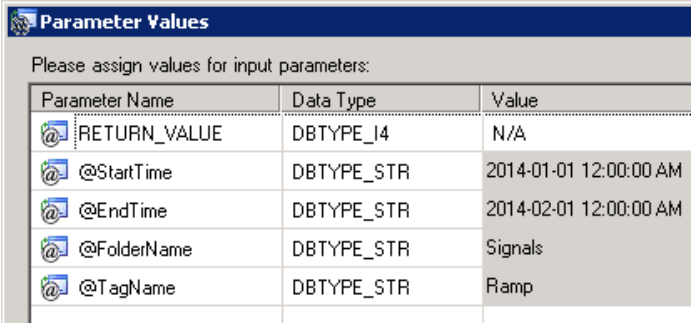


Figure 1 - Selecting the stored procedure

5. After you have selected the stored procedure the "Schema Metadata Auto-Discovery" dialog will appear, asking if you'd like to execute the command now to get the schema columns. Click **Yes, Execute the Command now…** When prompted, fill in parameter values appropriate for your Hyper Historian configuration. The parameters may look like Figure 2.

NOTE: Be sure to use the Name of your group and not the Display Name.

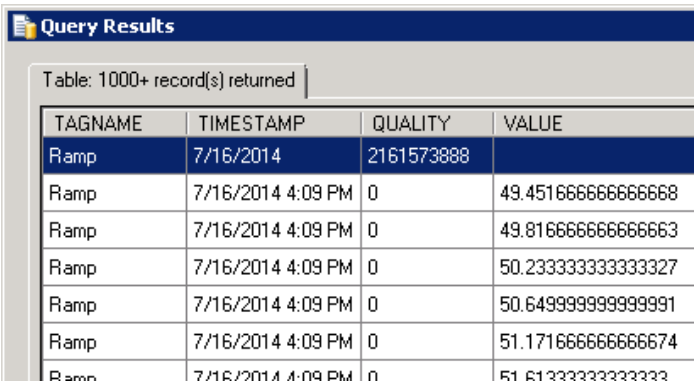
NOTE: If you have more than one level of folders, the additional levels should be added to the TagName parameter. So if you have a tag named MyTag nested in Folders named Folder1 → Folder2, then your @FolderName parameter would be "Folder1" @TagName would be "Folder2/MyTag". See the help files or application notes for more details on the correct values for the SQL Query Engine.



Parameter Name	Data Type	Value
RETURN_VALUE	DBTYPE_I4	N/A
@StartTime	DBTYPE_STR	2014-01-01 12:00:00 AM
@EndTime	DBTYPE_STR	2014-02-01 12:00:00 AM
@FolderName	DBTYPE_STR	Signals
@TagName	DBTYPE_STR	Ramp

Figure 2 - Example parameters

6. Once you are done you can run test query by clicking on the **Test SELECT Command** button. You can use the same parameters that were used for the schema auto-discovery, or different parameters. A successful query should return output that looks like Figure 3.



TAGNAME	TIMESTAMP	QUALITY	VALUE
Ramp	7/16/2014	2161573888	
Ramp	7/16/2014 4:09 PM	0	49.451666666666668
Ramp	7/16/2014 4:09 PM	0	49.816666666666663
Ramp	7/16/2014 4:09 PM	0	50.233333333333327
Ramp	7/16/2014 4:09 PM	0	50.649999999999991
Ramp	7/16/2014 4:09 PM	0	51.171666666666674
Ramp	7/16/2014 4:09 PM	0	51.613333333333333

Figure 3 - Test results

7. Remember to click **Apply** when done.

Reading Data in a GridWorX Viewer

1. Open **GraphWorX64**.
2. From the **Controls** ribbon, add a **GridWorX Viewer** to your display.
3. Double click on the GridWorX Viewer control to bring up the configuration dialog box.
4. Click on **Grid** from the tree on the left.
5. Under **Source** tab, click on to create a new source. Choose **Data Set Tag Subscription**.
6. The tag browser will appear. Browse for **Databases → SQL Server → Hyper Historian → Data Sources → GetRawData.** Then click OK.
7. You may get an error telling you the schema could not be retrieved. Ignore this error for now – click **OK**.

8. In the Select Data Tag field, fill in the parameter values after each equal sign. Use double quotation marks around each parameter value. A final Data Tag string should look something like this:

db:Hyper Historian.GetRawData<@StartTime="2014-01-01 12:00:00 AM", @EndTime="2014-01-02 12:00:00 AM", @FolderName="Signals", @TagName="Ramp">

9. If the "Data Tag Fields retrieved" table did not populate already, hit the green refresh button.
10. When fully configured, your Source tab should look similar to Figure 4.

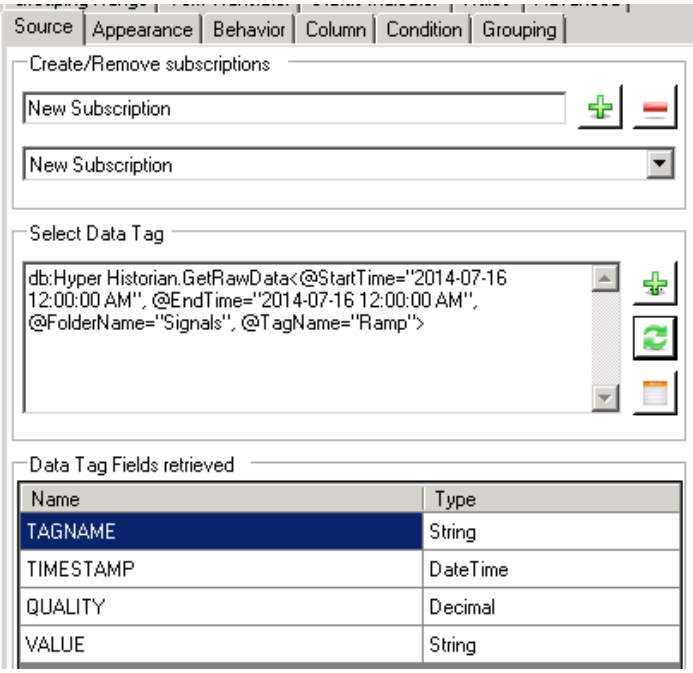


Figure 4 - GridWorX Viewer Source tab

11. Click Close to save your changes.
12. Go to runtime and see the table with Hyper Historian data.

Using Global Aliasing for Dynamic Parameters

Rather than hard-coding the values into the GridWorX Viewer, global aliases can be specified as parameter values, allowing the user to dynamically change the tag or time range of the viewer in runtime.

1. Edit your previously-created GridWorX Viewer, select the Grid, go to the Source tab, and replace the tag name value with a global alias, as shown below. Make sure to leave the quotation marks around the alias.

db:Hyper Historian.GetRawData<@StartTime="2014-01-01 12:00:00 AM", @EndTime="2014-01-02 12:00:00 AM", @FolderName="Signals", @TagName="<#TagName#>">

2. Add a pair of buttons to your display. Both buttons should have Set Global Alias pick actions, and should set the value for the TagName alias. Each button should set the value with a different tag name from your logging group, like Sine and Ramp.

   NOTE: You may add the TagName alias into the Global Aliasing provider, but this step is optional.

3. Go into runtime and click the alias buttons. Your GridWorX Viewer should reload with data for the new tag.