



Hyper Historian

Accessing Historical Values as Live Data



APPLICATION NOTE

August 2014

Description: The Hyper Historian provides access to the historically logged data via a number of historical interfaces such as OPC HDA, OPC UA, and SQL, in addition it also provides the ability to access individual historical samples using a time context as live data via OPC DA and OPC UA.

Introduction

One of the key features of the Hyper Historian that distinguishes it is the ability to not only provide access to the historically logged data via a number of historical interfaces such as OPC HDA, OPC UA, and SQL, but to also access individual historical samples using a time context as live data via OPC DA and OPC UA.

This is especially useful for values calculated either as aggregates or as performance calculations in the Hyper Historian as they are not available by any other means.

Three common use cases for accessing the values using OPC DA or OPC UA are:

- To provide access the most up-to-date (calculated) value so that the user can see summary data in their GraphWorX64 displays
- To provide access the most up-to-date (calculated) value so that the ICONICS Alarm Server can generate the necessary events from the internally calculated values
- To provide replay access to the data, to in effect provide a means to what the operator would have seen when a particular event occurred

Accessing the Most Up-To-Date Value

Accessing the most up-to-date value of either directly logged, aggregated or calculated (via the performance calculations) values in GraphWorX64, the AlarmWorX64 Alarm Server or any other compatible product can be easily achieved by simply browsing to the tag that needs to be used.

GraphWorX64

To access the most up-to-date value from GraphWorX64:

1. Open GraphWorX64
2. Insert a Process Point from the Dynamics Ribbon
3. The Data Browser will automatically open
4. In the Navigation Tree select **Home** → **Historical Data** → **Hyper Historian** → **Emissions** → **Dust**
5. Click on OK and the following tag will be used as the DataSource in the Process Point dynamic:

HyperHistorian\\Configuration/Emissions/Dust

AlarmWorX64 Alarm Server

To access the most up-to-date value from the Alarm Server:

1. Open the Workbench
2. Select the Alarm Server provider
3. Expand **localhost** → **AwxServer** → **Configuration** → **Default**
4. Select **Add** → **Tag** from the Home Ribbon
5. Name this tag DustTag
6. Click on the Data Browser button (...) for the OPC Input
7. In the Navigation Tree select **Home** → **OPC Classic Servers** → **My Server** → **ICONICS.HyperHistorianDA.1** → **Most Recent** → **Emissions** → **Dust**
8. Click on OK and the following tag will be used as the OPC Input:
@ICONICS.HyperHistorianDA.1\Emissions/Dust.Value
9. Click apply

NOTE: It is possible to use both tags in either GraphWorX64 or the Alarm Server, however the examples above use the most efficient communication method for each.

Accessing Specific Historical Samples

The examples above provide access to the most up-to-date values, however to access historical data at specific times as live data, Hyper Historian must be provided with a Time Context.

The Time Context allows the user to specify for which time the live values should be retrieved. Hyper Historian allows the users to specify as many Time Contexts as they need, and then all tags that share that Time Context will show the values for the specified time within that context.

Data Replay Tags

To use Hyper Historian tags that expose a specific historical value at a given time use:

1. Open GraphWorX64
2. Insert a Process Point from the Dynamics Ribbon
3. The Data Browser will automatically open
4. In the Navigation Tree select **Home** → **Historical Data** → **Hyper Historian** → **Emissions** → **Dust** → **Data Replay**
5. Click on OK and the following tag will be used as the DataSource in the Process Point dynamic:
HyperHistorian\\Configuration/Emissions/Dust.%Context

The main difference from the example before is the addition of the **.%Context** at the end of the tag name, new contexts can be created as need be by simply typing in a new name instead of **Context**, however in most cases only a single context is required.



Hyper Historian

Accessing Historical Values as Live Data



APPLICATION NOTE

August 2014

Data Replay Time Tags

Once the necessary tags have been created on the display, an additional time tag needs to be added to allow the user to select a specific time

1. Insert a Data Entry from the Dynamics Ribbon
2. The Data Browser will automatically open
3. In the Navigation Tree select **Home** → **Historical Data** → **Hyper Historian** → **Data Replay** → **Time**
4. Click on OK and the following tag will be used as the DataSource in the Process Point dynamic:
HyperHistorian\\Configuration.%Context/Time

If a different context has been used, i.e. the word **Context** has been replaced by something else, ensure to use the same context is used here.

Runtime Usage

Once the tags have been configured put GraphWorX64 into runtime and enter the appropriate UTC time/date into the Data Replay Time Tag Data Entry field.

By default, before a value is entered, the context is set to show the most up-to-date value, once a date/time value has been entered once, it will show that value, to reset the context back to showing the most up-to-date value, the date/time value of 01/01/0001 00:00:00 has to be written.

Useful GraphWorX64 buttons

Besides explicitly selecting a date/time for which to show the data reply values, it is normally useful to have a few time control buttons to navigate through time.

All of the following examples are pick field actions of the type write value, they all have the same value for the DataSource property, but different values for the OnUpValue property or the WhileDownValue property:

All: Data Source

HyperHistorian\\Configuration.%Context/Time

Move forward by *n* seconds: OnUpValue

Change the value in the *fromseconds* to the appropriate number for your application

x= {{HyperHistorian\\Configuration.%Context/Time}} + fromseconds(1)

Move backward by *n* seconds: OnUpValue

Change the value in the *fromseconds* to the appropriate number for your application

x= {{HyperHistorian\\Configuration.%Context/Time}} - fromseconds(1)

Pause: OnUpValue

Switch the context from its real-time (most up-to-date) state to a paused state where it will pause at showing the current values
x= utcnow()

Resume: OnUpValue

Switch the context from its historical state back to real-time (most up-to-date)
x= todatetime("01/01/0001 00:00:00")

Fast forward by *n* seconds: WhileDownValue

Change the value in the *fromseconds* to the appropriate number for your application. The ExecuteWhileDown must be set to **True**, and the WhileDownInterval is set to **100**

x= {{HyperHistorian\\Configuration.%Context/Time}} + fromseconds(1)

Fast backward by *n* seconds: WhileDownValue

Change the value in the *fromseconds* to the appropriate number for your application. The ExecuteWhileDown must be set to **True**, and the WhileDownInterval is set to **100**

x= {{HyperHistorian\\Configuration.%Context/Time}} - fromseconds(1)