MergeWorX – Using MergeWorX to Insert Data into Hyper Historian

APPLICATION NOTE

September 2013

Description: Using MergeWorX to import CSV data into Hyper Historian
OS Requirement: Vista x64/ Windows 7 x64/ Windows 8 x64/ Windows Server 2008 x64/ Windows Server 2008 R2 x64/ Windows Server 2012
General Requirement: Installation of GENESIS64 v10.80 and basic knowledge of Hyper Historian and GENESIS64

Introduction

MergeWorX is a tool for automatic or manual inserting of data into the Hyper Historian (or any OPC UA history write capable server). It incorporates a plugin technology which allows for the creation of a custom .NET plugin module to read data from any desired source. This plugin is configured from the ICONICS Workbench and doesn't require a custom configurator to be created.

MergeWorX is an event driven tool and uses ICONICS Unified Data Manager Triggers for this purpose. The Unified Data Manager provides both time and data driven triggers. This allows both fully automatic data merges as well as manually driven data merges triggered by an OPC values data change.

As triggering is set on plugins level, this allows for multiple instances of the same plugin, each trigged by its own trigger, each processing data from a different IO/factory floor device.

This application note will show how to use the provided MergeWorX CSV plugin to merge data into existing Hyper Historian log files. This will cover merging data without any pre-processing on it.

Pre-requisites

To be able to import external data into Hyper Historian using MergeWorX, the following items are required:

1. Hyper Historian has to be logging during the time when the MergeWorX plugin is running, as shown in Figure 1.

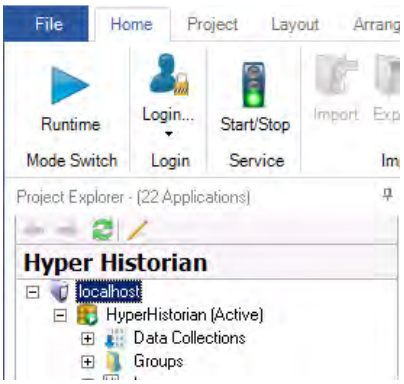

Figure 1 - Hyper Historian Running

2. Hyper Historian needs to have a data logged for the time period which MergeWorX is planning to merge.

   For instance if a new .HHD file is created every day, and MergeWorX is looking to import data from 9 a.m. to 1 p.m. then the log file will need to have been created at 12 a.m. for the merge to succeed, an example file is shown below.

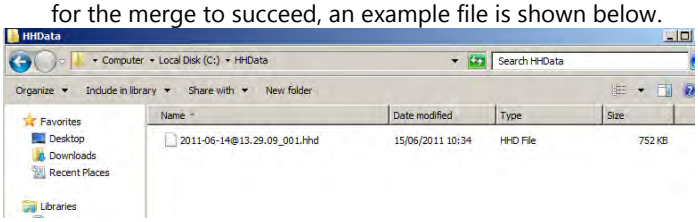

Figure 2 - Example HHD File

3. Hyper Historian HDA editing must be enabled.

   This is found in the Hyper Historian Configurator under the following section: Node Setup and Redundancy > Local Logging Server. In the General section the Disable HDA Editing checkbox will need to be unchecked.

CSV File

MergeWorX CSV files have 3 ways to define how values are to be merged. However the following specifications have to be adhered to:

- Keep one empty line between each type and each point/value block.
- A delimiter character must be present, this can be configured in the plugin settings however by default it is ",".

The recommended way of constructing the CSV file is as follows:

The CSV file will consist of two parts; a header definition line and the data lines. The header will define the order of the columns, and consists of the DataPoint, Value, Timestamp, Quality and ValueType, of which the Timestamp, Quality and ValueType are optional. The following table offers a recommendation on how the CSV file should be formed.

DataPoint,Value,Timestamp, Quality, ValueType
Location1\PLC\Temp,24,2011/06/14 12:00:00,0,Int32
Location1\PLC\Temp,23,2011/06/14 13:00:00,0,Int32
Location1\PLC\Temp,22,2011/06/14 14:00:00,0,Int32

The DataPoint refers to the point name used to identify the data, MergeWorX offers processing capabilities on imported data so if they are to be utilized then the DataPoint must be the input name of the MergeWorX tag created. If no pre-processing is required then the DataPoint will need to have the full browse path of the Hyper Historian tag that the data belongs to.

The timestamp must be formed in the YYYY/MM/DD HH:MM:SS format, however if it isn't specified then the UTC time when parsing the file will be used.

The quality codes conform to the ICONICS FrameWorX specifications and have to be specified by numeric value. Leaving Quality field empty results in use of default quality code defined in configuration, the following are the most common quality codes, however a full list of codes can be provided if required:

Bad Quality: **134217728**
Good Quality: **0**

The ValueType field will most commonly be written as its string value; however it can also be given its numeric enumeration. If the field is left empty it will attempt to parse value to an appropriate type, defaulting to the string type if it fails.

Empty = 0,
Boolean = 3,
Char = 4,
SByte = 5,
Byte = 6,
Int16 = 7,
UInt16 = 8,
Int32 = 9,
UInt32 = 10,
Int64 = 11,
UInt64 = 12,
Single = 13,
Double = 14,
DateTime = 16,
String = 18

Following types are not allowed, using those results in string value:

Object = 1,
DBNull = 2,
Decimal = 15

Trigger Setup

Before creating a MergeWorX plugin the user will need a method of executing the action, these triggers are created in the Unified Data Manager and are accessed in the Workbench.

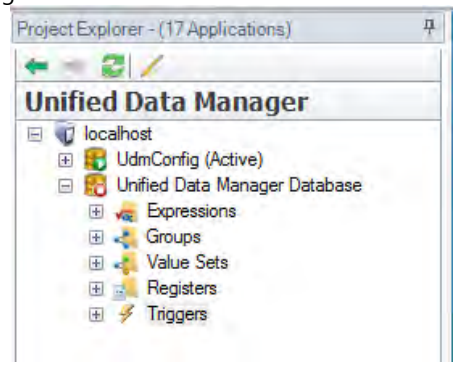

Figure 3 - UDM Triggers

The first type of trigger which can be used are Data Triggers, and these are either based on OPC tags or Static Registers, and are used primarily for manually starting the merge action.

A UDM register can be created in which the user will be able to write to, to drive the trigger. The following image is a data trigger which is triggered when the data tag changes value, as illustrated in Figure 4.

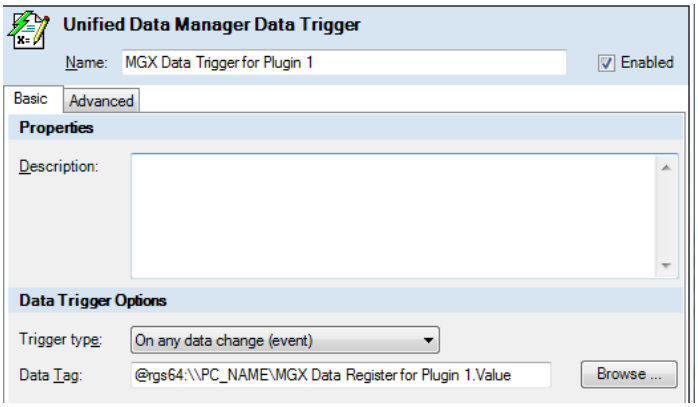

Figure 4 - UDM Data Trigger

The second type of trigger available is a Time Trigger and these are used for periodically triggering the merge action at a configurable interval. The following image illustrates a time trigger which is scheduled to execute every minute, as shown in Figure 5.

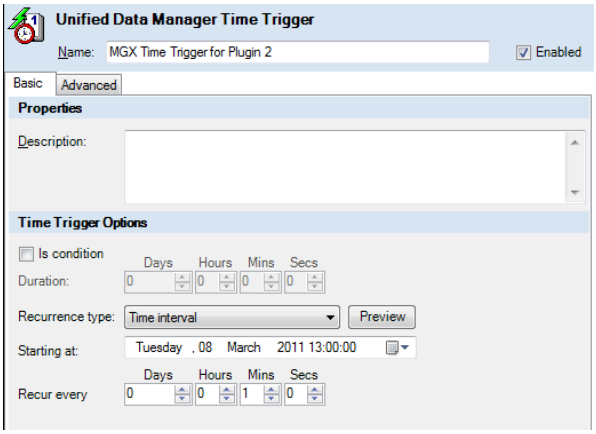

Figure 5 - Time Trigger

Plugin Setup

After the trigger has been created, the MergeWorX plugin will need to be created from inside the MergeWorX configurator. A new plugin is created by expanding the tree to the Plugins folder, right-clicking and selecting New Plugin.

In the general section the default plugin type will be set to CSV, however you will need to browse to the trigger point setup in the previous section. Figure 6 illustrates how the trigger point will appear after browsing.

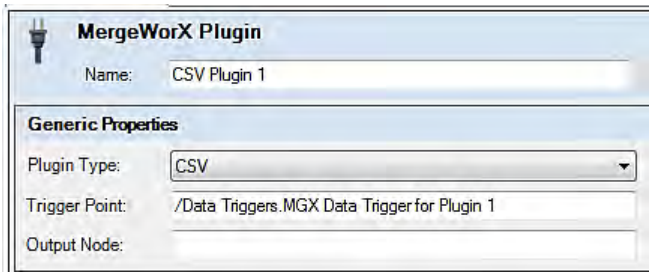

Figure 6 - Trigger Point

The output node is primarily for specifying a remote Hyper Historian installation, which won't be covered in this document.

After the trigger point has been selected, the user will need to configure the locations of the file(s) to be imported along with the action to be applied against processed files, as illustrated in **Error! Reference source not found.**.

Figure 7 - Plugin Configuration

The default plugin configuration settings should suffice, however the recommended location for the Input CSV files is:

C:\ProgramData\ICONICS\MergeWorX\Plugins\CSV\Source

The recommended location for the processed archive files to be moved to is:

C:\ProgramData\ICONICS\MergeWorX\Plugins\CSV\Processed

However these locations can be changed if desired, once these options are applied the advanced configuration options can be changed if required. Once these settings are all applied the traffic light can be clicked to start the service.

Example CSV File Types

Type 1
@DataPoint,opc.tcp://[Your HH PC]:7011/Hyper Historian\\Configuration/LogGrp/OPCUA/Sine Value,Timestamp
10,2011-03-08 5:14:00 PM
100,2011-03-08 5:15:00 PM
1000,2011-03-08 5:16:00 PM

Type 2
DataPoint,Value,Timestamp
opc.tcp://[Your HH PC]:7011/Hyper Historian\\Configuration/LogGrp/OPCUA/Sine,10,2011-03-08 5:14:00 PM
opc.tcp://[Your HH PC]:7011/Hyper Historian\\Configuration/LogGrp/OPCUA/Sine,100,2011-03-08 5:15:00 PM
opc.tcp://[Your HH PC]:7011/Hyper Historian\\Configuration/LogGrp/OPCUA/Sine,1000,2011-03-08 5:16:00 PM

Saving either of these as a .csv file will be able to be imported, provided the names are adapted to your settings.