Data Management Magnotix Overview

System Overview

Industrial Strength Data Replication for OT

Iintroduction

IT-OT Convergence is driving a need to extract value from existing industrial data historian, DCS and SCADA solutions. Magnotix is a software solution used to transfer, transform and validate OT data between systems using flow-based workflow configuration. Functionality is encapsulated in nodes which can be interconnected in numerous ways by wiring input and output together. Nodes consist of data sources, data endpoints, data analysis and data manipulation. The base Magnotix installation consist of over 30 nodes geared towards liberating stranded industrial data. Data can be retrieved from and/or sent to the data connectors shown in the diagram below:

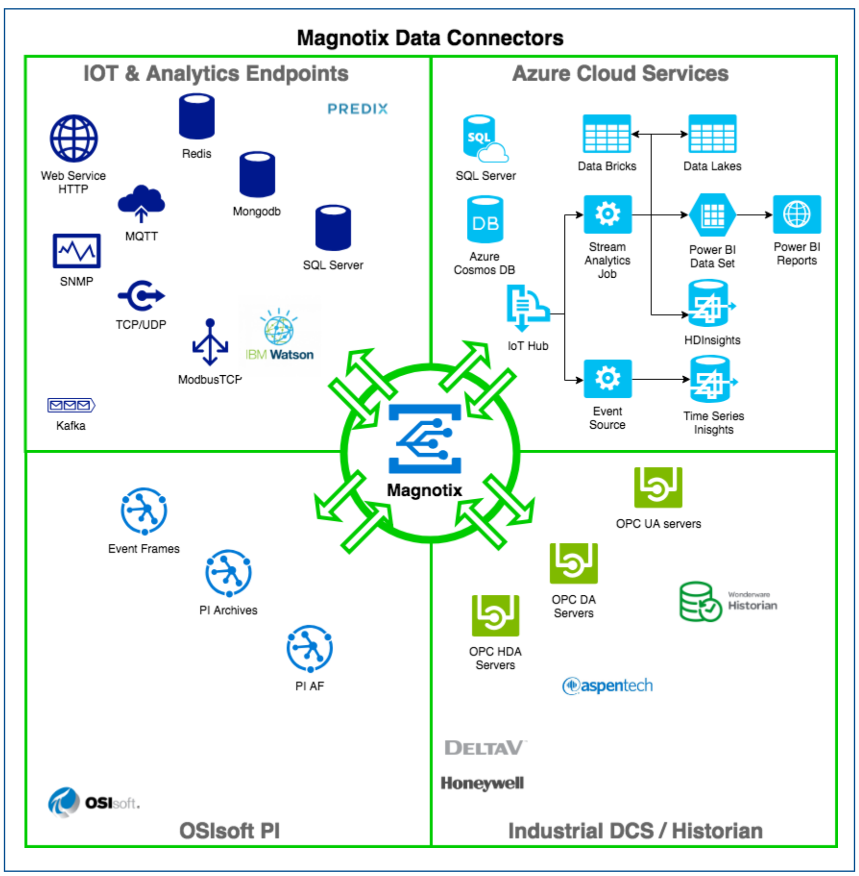

Figure 1: High-Level Overview

Table of Contents

Currently available data connector details are shown in the table below:

Node	Description
Core Industrial Connectors	
PI AF	Read historical and instantaneous information
OPC DA	OPC DA synchronous read/write
OPC HDA	OPC HDA historical read/insert
OPC UA	OPC UA client/server
Modbus TCP	Modbus TCP client/server
Relational DB Connectors	
SQL Server	Select, insert, update stored procedures (with prepared statements)
Oracle	Select, insert, update stored procedures (with prepared statements)
None Relational Connectors	
MongoDB	MongoDB No SQL database connectivity
Redis	In memory database connectivity
IOT	
MQTT	MQTT message queue client/server
Azure IoT Hub	Send or retrieve information from Azure IoT hub
Kafka	Kafka message queue client
Cloud	
Azure SQL Server	Send or retrieve information from Azure SQL Server
Azure Cosmos	Send or retrieve information from Azure Cosmos Database
Other	
HTTP (REST)	HTTP client/server (connectivity to restful API services)
TCP	TCP client/server
UDP	TCP client/server
SNMP	SNMP client
Files	Perform file operations (read and/or write files); monitor file updates & trigger workflow actions on change

The connectors above have been tested by iSolutions against industrial data sources. In addition to the connectors above, the Node-Red core library has hundreds of additional open-source node types that can be added to Magnotix workflows.

Common use cases are as follows:

1) Transfer data between systems such as the following:

- PI to PI
- OPC to PI
- PI or OPC data to cloud services

2) Transfer data to and/or from machine learning services in the cloud

3) Move data to the cloud for reporting and analysis

4) Data transformation and/or validation during transit

5) Backfill historical data with validation

6) Asynchronous / event-based signals exchange between corporate and industrial systems

7) Events & alerting frameworks between industrial, corporate & cloud systems

Magnotix Software Architecture

Magnotix Software Architecture consists of the runtime Windows service, Web Browser Editor, data manager ASP.Net API service and Windows service console. The software architecture details are as follows:

1) Runtime Windows service – the core service functionality use to execute the workflows

2) Web Browser Editor – used to build the flows and deploy changes

3) Data Managers – used to manage OPC and PI information

4) Windows service console – Windows program used to control, configure and monitor the windows service and view log files

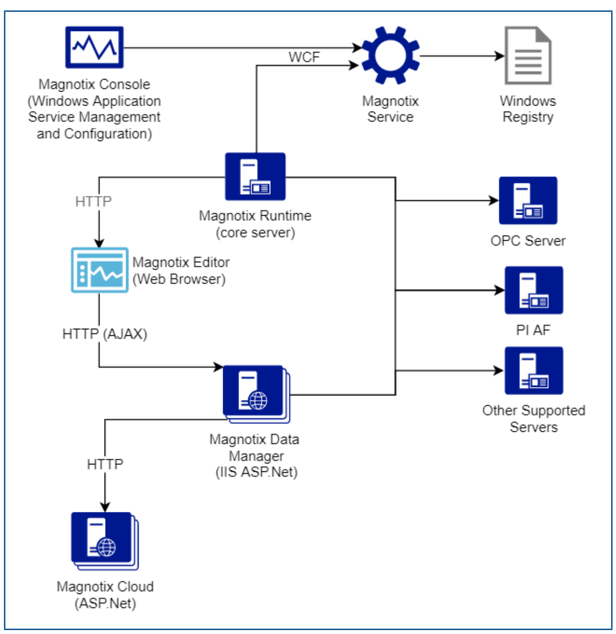

Figure 2: System Architecture

System Performance

Magnotix is built on the core Node Red framework, and supports asynchronous data transfer.. the Framework effectively leverages multiple CPU cores, available system RAM and disk for caching of datasets. In most cases, overall performance is constrained by the throughput of the source industrial system (OPC servers for instance), and to a lesser extent, the destination server's available processing capacity. Source and destinations available CPU and Disk IOPs will affect performance depending on how often information is retrieved. Magnotix performance is dependent on the operations performed on the data as wells as the frequency of the data. The higher the frequency the more CPU processing is required.

Typical performance on a desktop-based server with client and server on the same machine are is follows:

1) PI AF Reads – 20,000 per second

2) OPC DA Reads – 20,000 per second

3) OPC HDA Reads – 4000 samples per second depending on the amount of data per read

Note: the throughput scenarios above are constrained by the available source system throughput. If additional source OPC / AF servers are available Magnotix can achieve higher throughput metrics.

System Requirements

Sample machine specification for high tag count / high-throughput scenarios:

- Windows OS: 64 bit Windows 2016 or later
- CPU 16 cores
- RAM 32 GB
- Hard Drive Space: 20 GB

Sample machine specification for low tag count / low-throughput scenarios:

- Windows OS: 64 bit Windows 2016 or later
- CPU 4 cores
- RAM 8 GB
- Hard Drive Space: 20 GB

Drive space reservation is to support Magnotix core install, storage of workflows / configuration, and to allow for local buffering of data in the case that connectivity to cloud / corporate networks should be interrupted.

Software Requirements

Software requirements are as follows:

- IIS 7.0 (Web Service)
- ASP.Net 4.6
- PI AF connectors requires the PI/AF SDK
- DotNet framework 4.

Installation Overview

Installation requires IIS and ASP.Net to be installed before installing Magnotix.

Installation should be done in the following order:

1) IIS

2) ASP.Net

3) Install Magnotix (using Magnotix installation program)

If OPC HDA Nodes are required:

1) Configure DCOM

2) Configure Local Security Policies

If PI Data Archive Nodes are required:

1) Install PI SDK 2018

If PI AF Nodes are required:

1) Install PI AF SDK 2018

Sample Magnotix Use Case

Magnontix can be used to read historical data from an OPC HDA server and transfer the contents to an Azure IoT Hub. A flow is configured using nodes (such as the OPCHDA now shown in the picture below). On the left is the node list which categories the nodes. On the right is the node editor worksheet. Nodes are dragged and dropped from the node list onto the editor to the right of the node list:

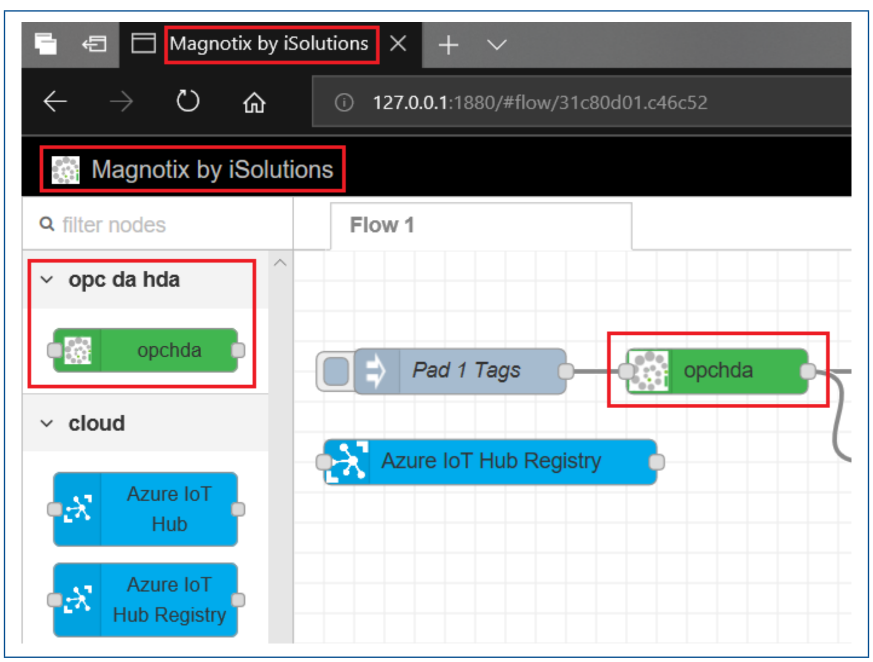

Figure 3: Magnotix UI 1

The Data Flow below is used to transfer information to Azure.

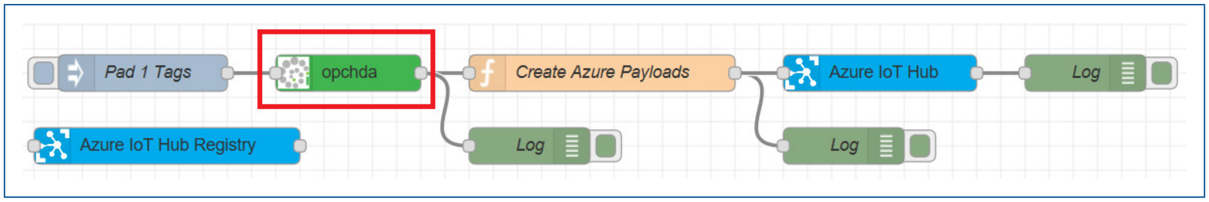

Figure 4: Magnotix UI 2

The Data Flow consists of the following nodes:

1) Tag definition trigger (Pad 1 Tags node) – this is a manual or scheduled trigger

2) OPCHDA client node – used to read historical data via OPC HDA

3) Azure payload preparation function – add Azure required information to the data

4) Azure IoT Hub node - transfers the data to Azure IoT Hub

Messages containing values and configuration information are based from node to node. Nodes are used to read, transform or send to the data to an endpoint. Once the data is sent to Azure, other Azure components can be utilized as shown in the picture below:

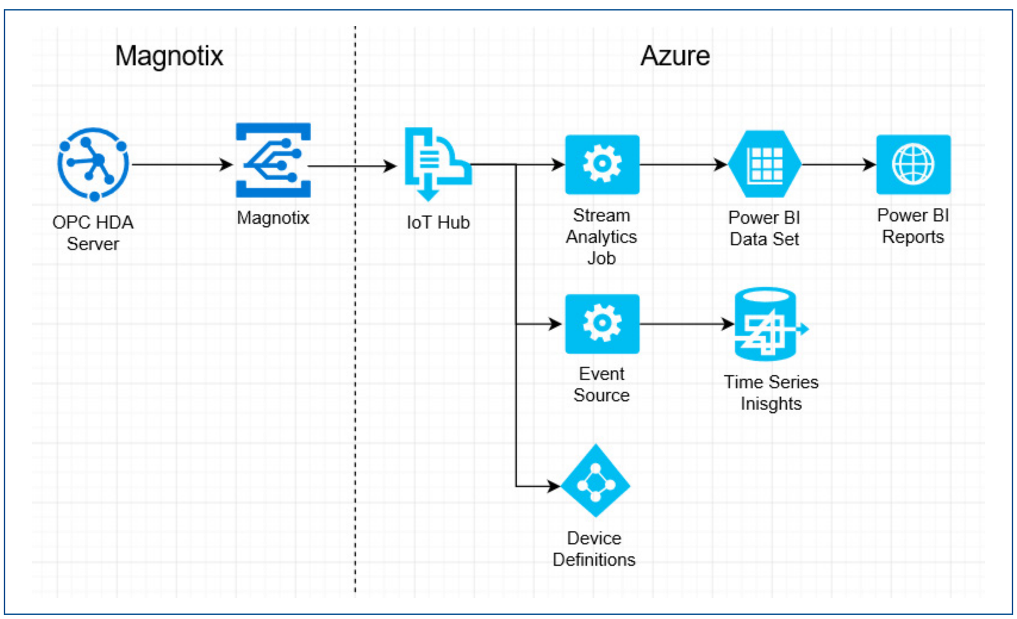

Figure 5: Magnotix-Azure Orchestration

The Azure components consist of the following components:

1) IoT Hub

2) Steam Analytics Job – used to transfer information between Azure components

3) Power BI Data Set – store data to be utilized by PowerBI

4) Power BI Reports – Visualize data or create reports

5) Event Source – used to send data to TSI

6) Time Series Insights – visualize time series data

7) Device Definitions

The Time Series Insights can be used to view the historical data as shown in the picture below.

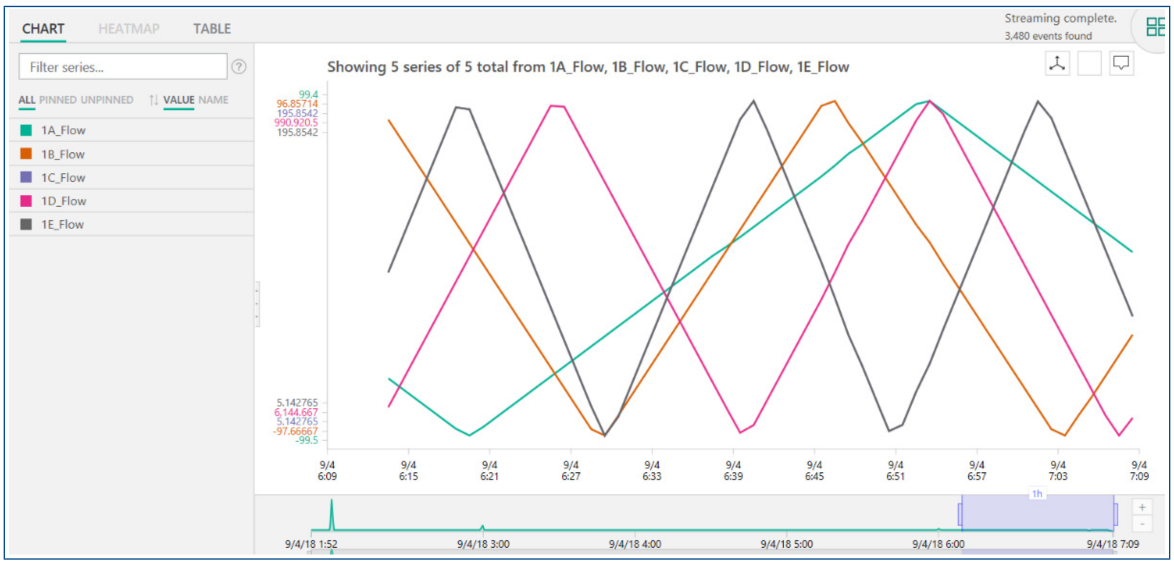

Figure 6: Microsoft TSI Front-end Display

Power BI can be used to view the OPC HDA Magnotix data as shown below.

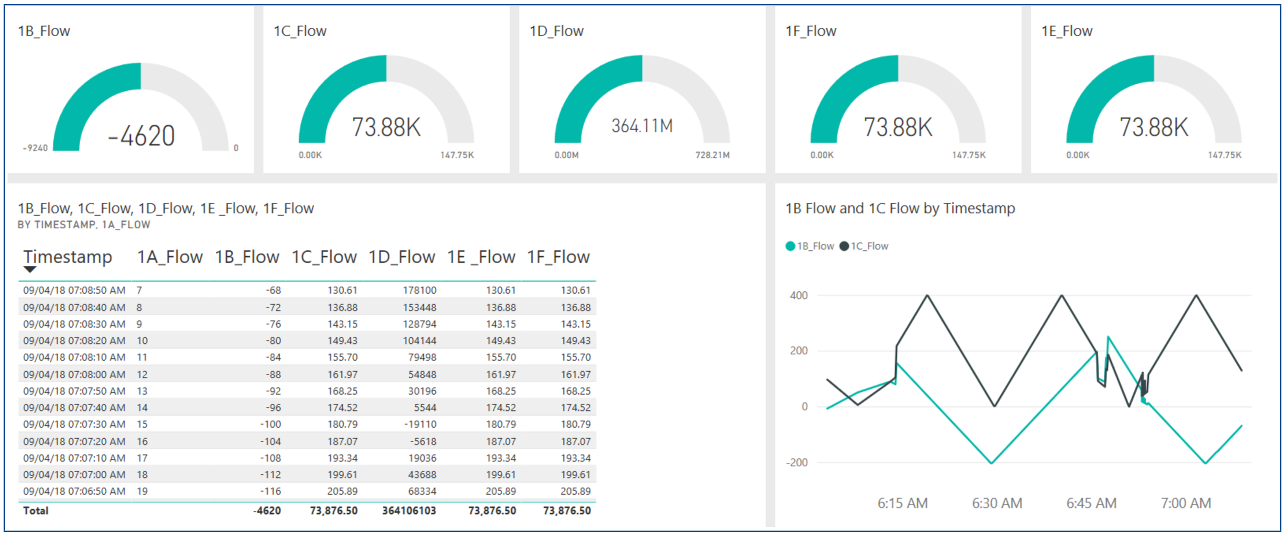

Figure 7: MS PowerBI Front End

Workflow Builder

The Magnotix Editor is accessed through the web browser. The Editor consists for five main parts the node list, flow, information panel, main menu and deploy menu (as shown in the picture below).

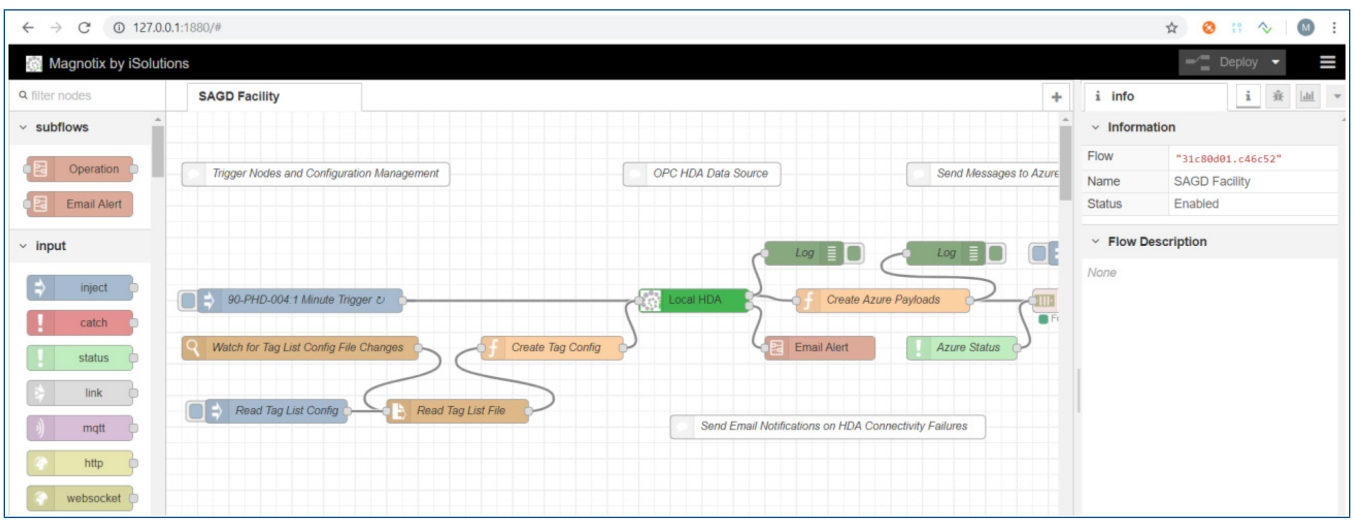

Figure 8: Sample Workflow

Node List

The node list is on the left side of the screen as shown below. Nodes can be dragged and dropped onto flows.

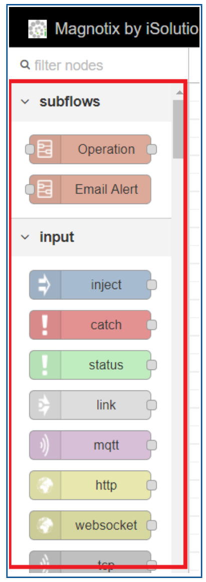

Figure 9: Node Selector

Node Configuration

Double clicking on a node opens the node's configuration. For example, if you double click the "90-PHD-004: 1 Minute Trigger" node (this represents an OPCHDA server for Honeywell Uniformance PHD), you will see the node configuration shown in the picture below.

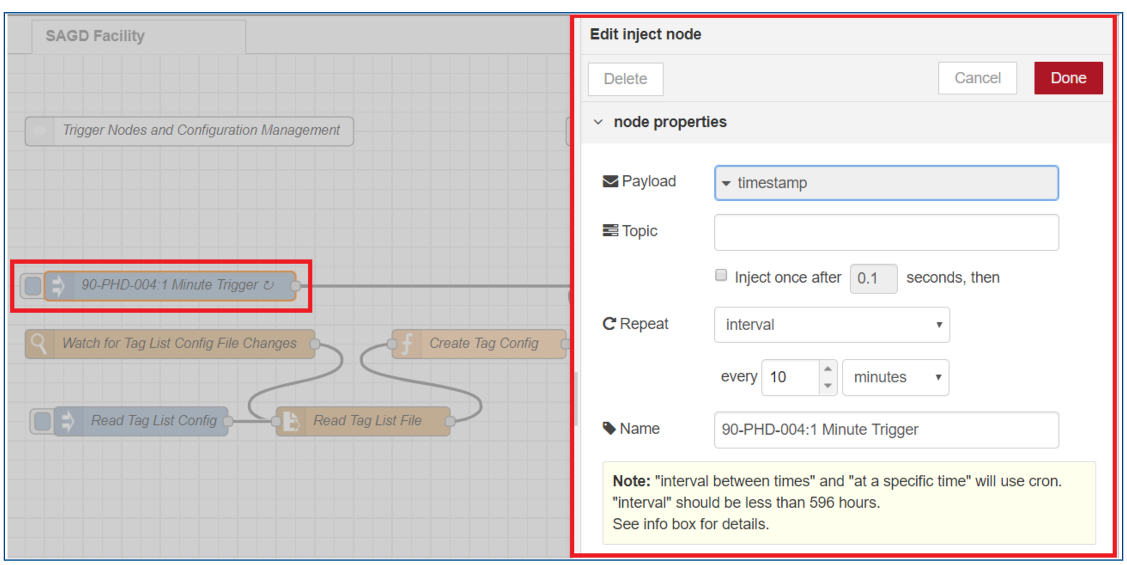

Figure 10: Node Configurator

OPC HDA Node Configuration Example

The OPCHDA node takes an array of tags and read the tag values. Double click the OPCHDA node to see its configuration.

Below is a picture of the OPCHDA node configuration:

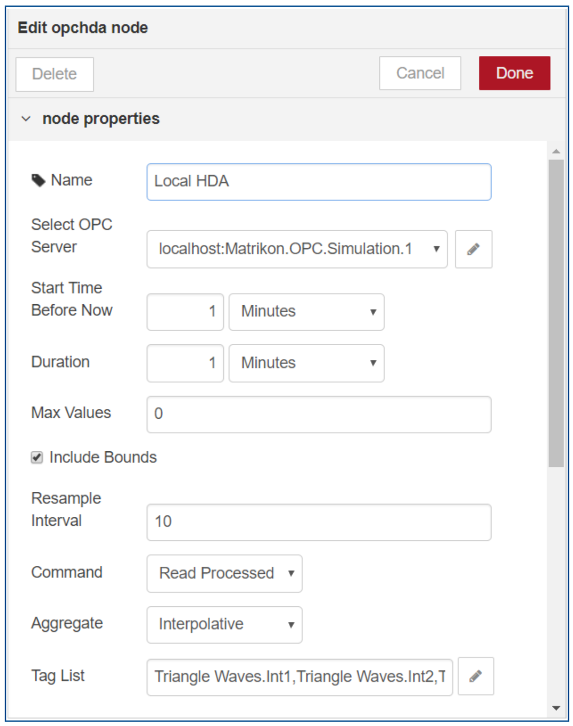

Figure 11: OPCHDA Node Config

SQL Server Node Configuration Example

The SQL Server Prepared Statement node used to execute a stored procedure or other SQL statement is shown in the picture below.

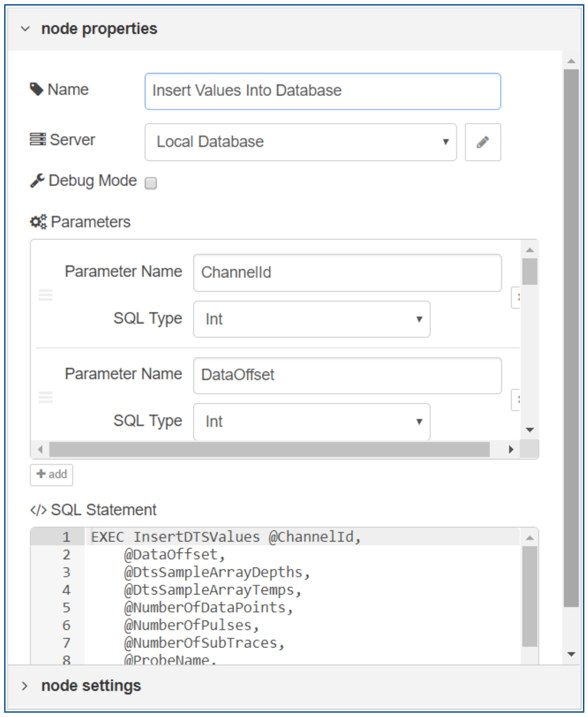

Figure 12: SQL Server Node Config

Sample SQL Server Endpoint Flow

Below is an example SQL Server endpoint flow which reads OPC DA values and inserts them into SQL Server:

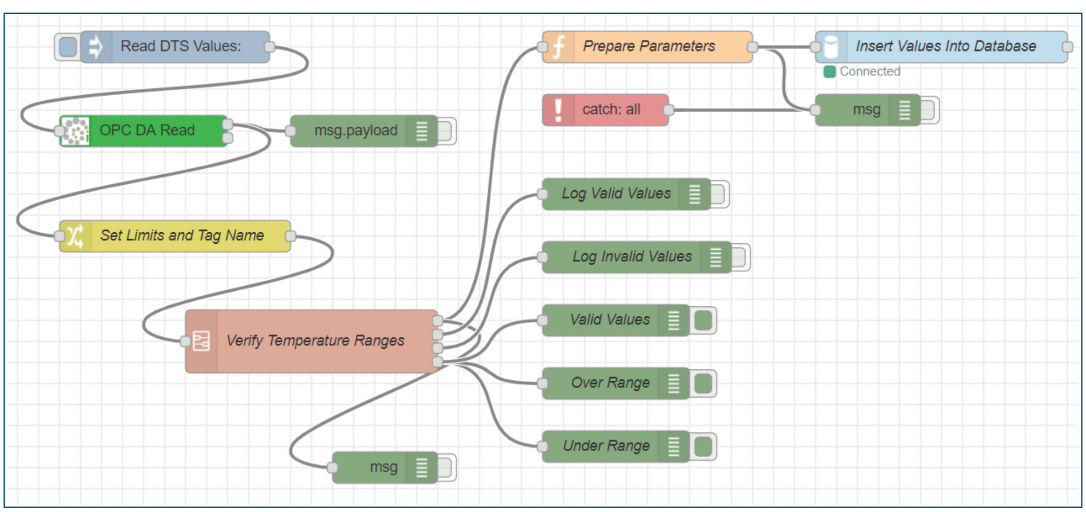

Figure 13: SQL Server Endpoint Config

Emerson
Operational Certainty Consulting
1100 W. Louis Henna Blvd.
Round Rock, TX 78681-7430
+1 888 889 9170

www.emerson.com/opcertcon