Sending ABB Totalflow Events to a Central Ignition Gateway

Prerequisites

- Installing the Java Runtime Environment
- Installing Ignition
- Installing the following MQTT Modules on two Ignition systems
  - Ignition System 1 (Central Ignition Gateway)
    - MQTT Distributor
    - MQTT Engine
    - MQTT Recorder
  - Ignition System 2 (Remote/Edge Ignition Gateway)
    - MQTT Transmission
    - EFM ABB Totalflow driver module

Overview

The EFM ABB Totalflow module is capable of polling events from an ABB Totalflow device based on a specified polling rate. With MQTT Transmission, these events can be published as Sparkplug records to an MQTT server. Any client subscribed on Sparkplug RECORD messages can receive these objects. In addition, MQTT Engine when combined with MQTT Recorder can also receive these messages and store these objects in a configured Ignition database. The following drawing shows the general architecture used to do this. This tutorial outlines the process of getting events to the central Ignition gateway.
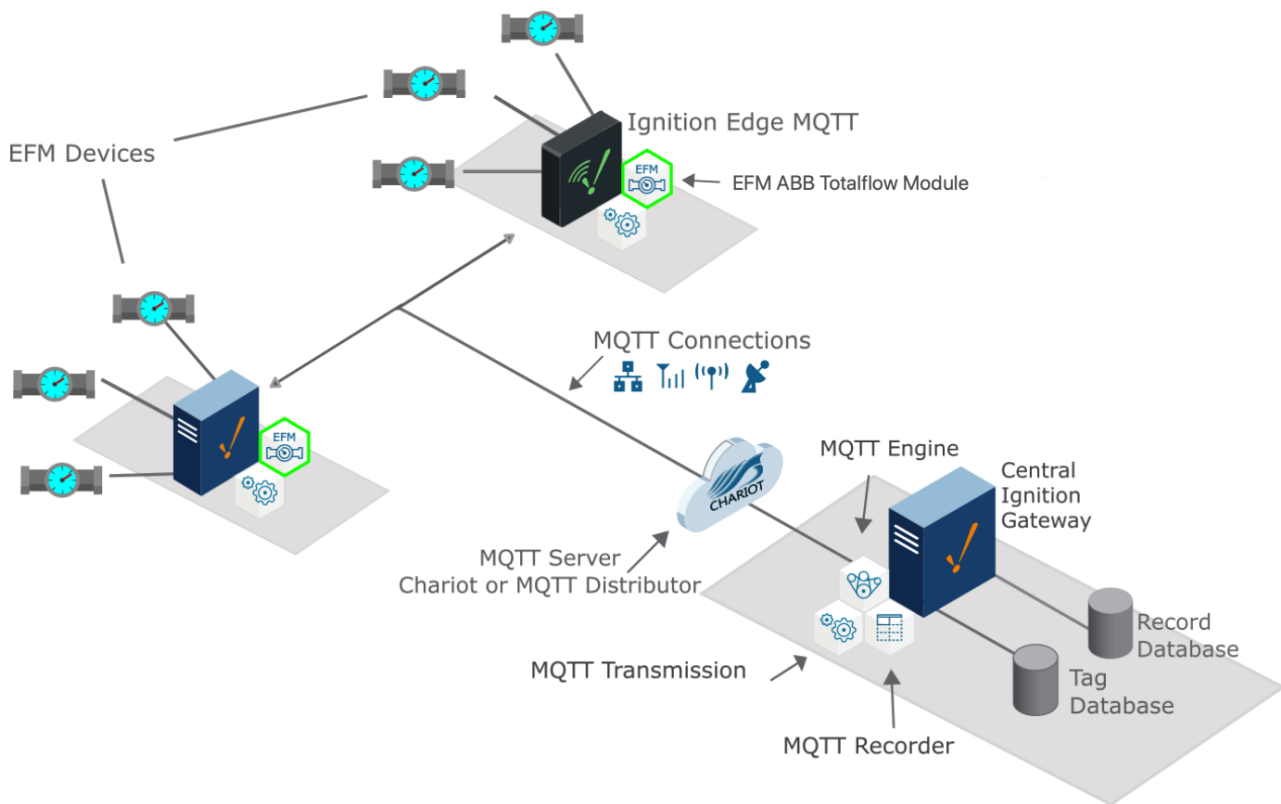
Sending ABB Totalflow Events to a Central Ignition Gateway

We must configure a total of five Cirrus Link modules on two different Ignition gateways to get event data flowing from an ABB Totalflow device to a central database. These are:

- Central Ignition Gateway
  - MQTT Distributor
  - MQTT Engine
  - MQTT Recorder
- Remote/Edge Ignition Gateway

- MQTT Transmission
- EFM ABB Totalflow

The configuration of each of these modules is covered below based on the Ignition gateway they're installed on.
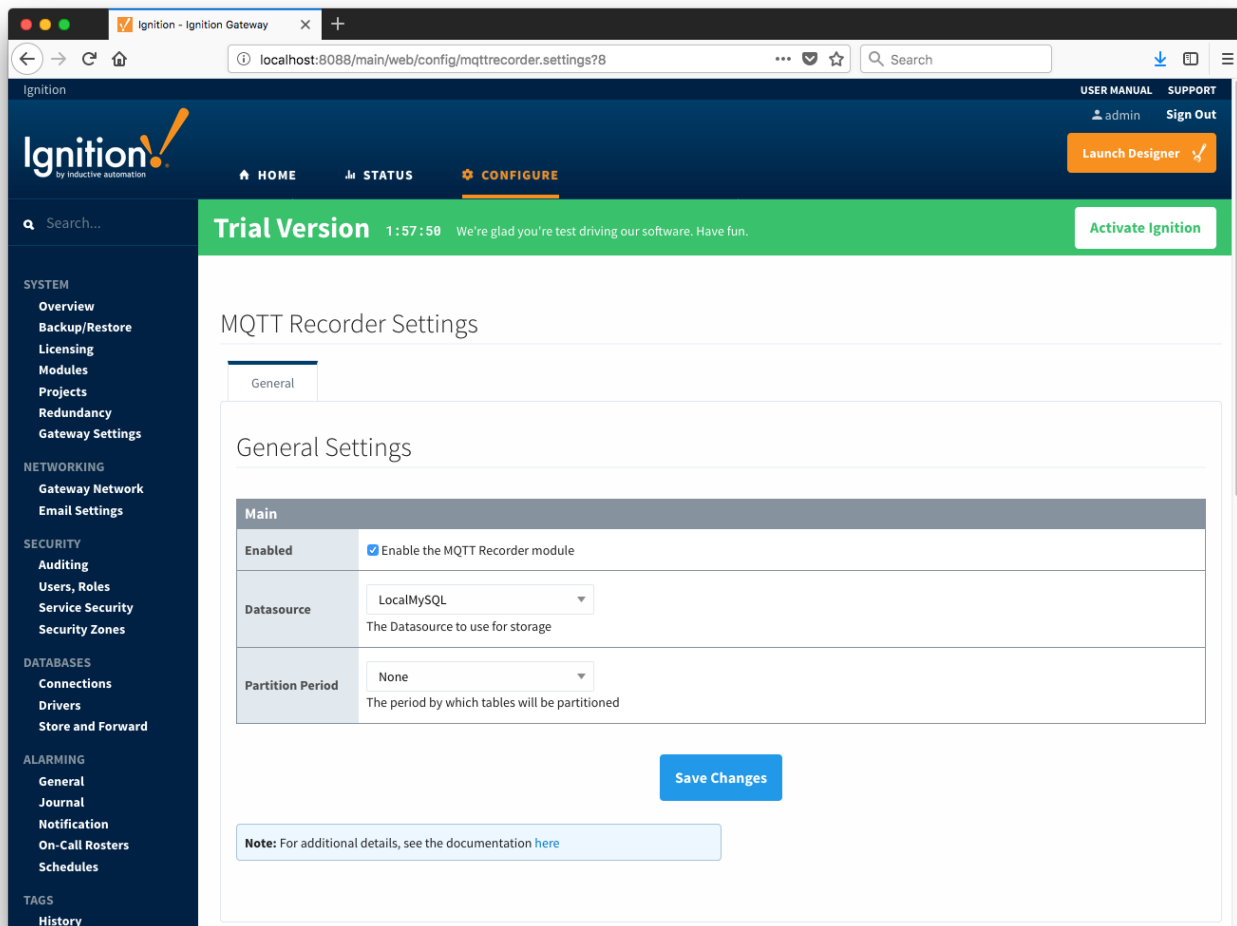
Central Ignition Gateway Setup

MQTT Distributor can be left in its default configuration. Note in a production system you would likely want to set up TLS especially if MQTT connections are using the Internet.

MQTT Engine can also be left in its default configuration.

MQTT Recorder requires that a database be set up in Ignition. That can be done as described in the 'Connect to a Database' section here. Note Ignition supports additional database types. For more detailed information about supported types, take a look at the information provided here. Once a database is set up, MQTT Recorder can be configured. Do so by opening the Ignition Gateway Web UI and browsing to the Configure tab at the top of the screen and then selecting 'MQTT Recorder Settings' as shown in the lower left below.

Once there, select a Datasource as shown in the image below. This drop-down will be populated with any database connections set up in Ignition. Optionally, a Partition Period can be selected to segregated tables by time periods.

At this point, the Central Ignition Gateway with MQTT Distributor, MQTT Engine, and MQTT Recorder is fully configured and ready to receive MQTT Sparkplug messages from the Remote/Edge Ignition Gateway. MQTT Distributor listens on TCP port 1883 by default for inbound MQTT connections. Make sure the Operating System's Firewall, Antivirus, and Malware protection services allow inbound connections on port 1883/TCP before proceeding.

Remote/Edge Ignition Gateway Setup

With the Central Ignition Gateway ready to receive MQTT/Sparkplug RECORD objects, the EFM ABB Totalflow and MQTT Transmission modules can be configured on the Remote/Edge Ignition Gateway.

Start by configuring the MQTT Transmission module. Do so by opening Ignition Designer and creating a tag structure similar to what is shown below.
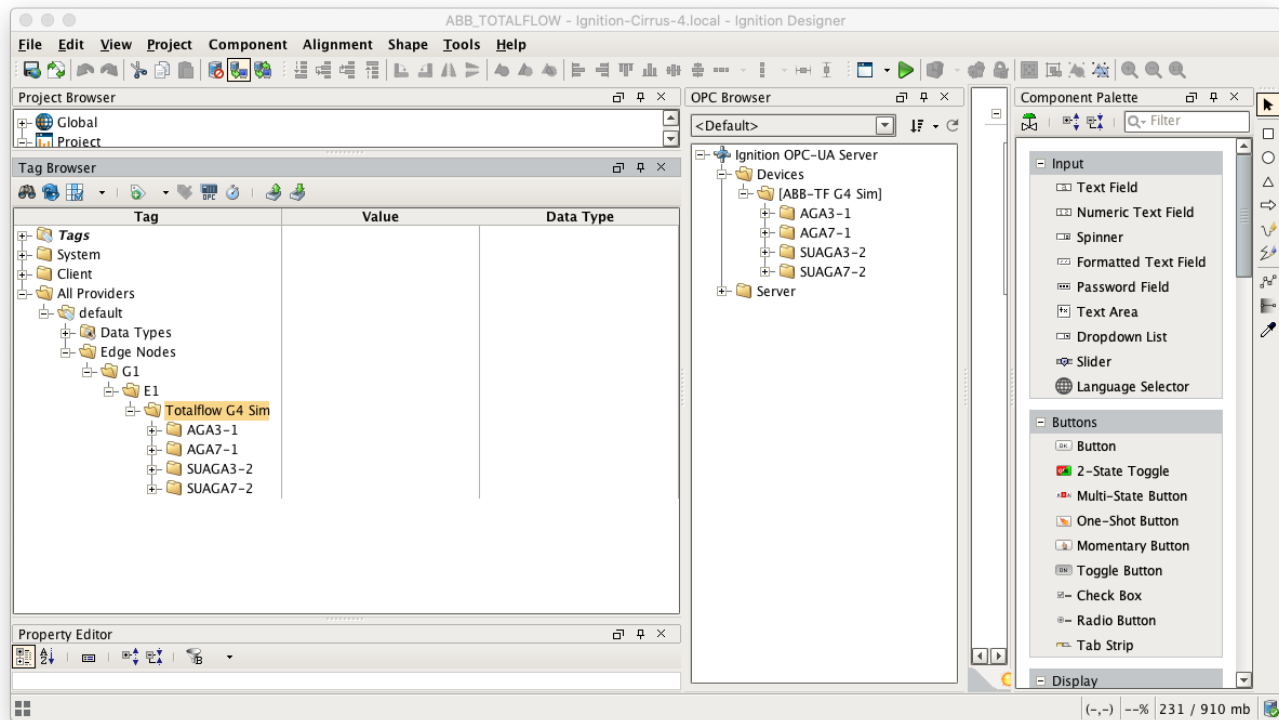
Note this structure is based on usage of the 'Default Transmitter' in MQTT Transmission. So, the directory structure is very important. Note the structure.

- tag provider/Edge Nodes/[Group ID]/[Edge Node ID]/[Device ID]/...

In the example below this implies the following definitions:

- [Group ID] = G1
- [Edge Node ID] = E1
- [Device ID] = Totalflow G4 Sim

These exact values will be used for the EFM ABB Totalflow connection Sparkplug parameters to tell the EFM ABB Totalflow which MQTT Transmission Transmitter configuration to use and, in turn, which MQTT connection to use to send the history data on.

Next the MQTT Transmission server configuration must be modified to point to the Central Ignition Gateway we set up earlier. To do so, in the Ignition Gateway Web UI browse to the Configure tab on the top and then to MQTT Transmission Settings in the lower left as shown below.

In the MQTT Transmission Settings configuration, click the Servers tab. Then click 'edit' on the Chariot SCADA MQTT Server definition. Modify the URL to match the URL of the Central Ignition Gateway. In this example, MQTT Distributor is installed on a Central Ignition Gateway at the IP address of 192.168.1.2. Once the URL is modified to match the configuration, there should be a '1 of 1' in the Connected column as shown below.
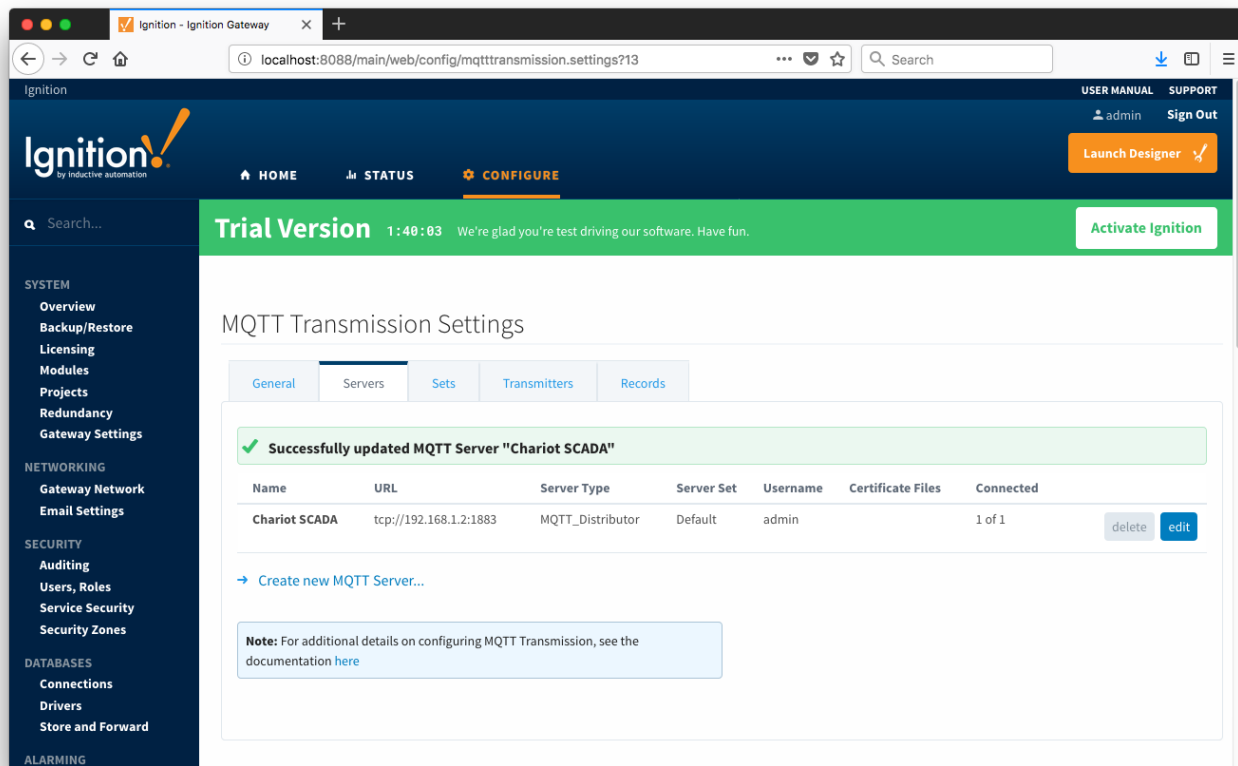
The next step is configuring the ABB Totalflow module. This is done as described in the ABB Totalflow Configuration manual. In going through the basic setup and configuration for Events configuration the following steps must be performed:

- Define the global Array-Registers definitions available for all ABB Totalflow devices in this Ignition instance.
  - This step can be skipped if not configuring the driver to poll for AAR data.
- Upload the Periodic Mappings for all ABB Totalflow devices in this Ignition instance.
  - This step can be skipped if default mapping provided by the driver is ok.
- Create the base device connection to the ABB Totalflow device.
- Specify the subset of global Array-Register definitions that this specific ABB Totalflow device uses.
  - This step can be skipped if not configuring the driver to poll for AAR data.
- Reconfigure device connection to enable polling for events.

As en example, let's configure the driver to poll for Event records and disable polling for Alarms and History. This can be done in two ways:

The first way to do it is to configure scan rates as shown below:

- **Alarm Scan Rate**
  - Set to -1 to disable polling for Alarms. If Alarms are not needed, it is the best practice to set the 'Alarm Scan Rate' to -1. With this setup, alarm records are not going to be pushed to the rs_efm_meter_alarm database table regardless of the alarm source setting (i.e. PERIODIC_HISTORY or ALARM_LOG_RECORDS).
  - If the 'Alarm Source' is set to the 'ALARM_LOG_RECORDS' and the 'Record Info Scan Rate' is set to -1, the 'Alarm Scan Rate' can be set to 0. With this setup, the AlarmPoller will not run on it own nor will it poll on notifications from the RecordInfoPoller.
- **Event Scan rate**
  - Set to a positive number to launch the EventPoller with specified poll rate.
- **Periodic History Scan Rate**
  - Set to -1 to disable polling for Periodic History records.
  - If the 'Record Info Scan Rate' is set to -1, the 'Periodic History Scan Rate' can be set to 0. With this setup, the PeriodicHistoryPoller will not run on it own nor will it poll on notifications from the RecordInfoPoller.
- **Daily History Scan Rate**
  - Set to -1 to disable polling for Daily History records.
  - If the 'Record Info Scan Rate' is set to -1, the 'Daily History Scan Rate' can be set to 0. With this setup, the DailyHistoryPoller will not run on it own nor will it poll on notifications from the RecordInfoPoller.
- **Record Info Scan Rate**
  - Set to -1 to disable the RecordInfoPoller so that there will be no notifications to Alarm, Events and History pollers on any 'RecordInfo' change such as 'Last Record Sequence number Used', etc.

The second way to do it is to configure scan rates as shown below:

- **Alarm Scan Rate**
  - Set to -1 to disable polling for Alarms.
- **Event Scan rate**
  - 0 Scan on notification form the 'Record Info' poller.
- **Periodic History Scan Rate**
  - Set to -1 to disable polling for Periodic History records.
- **Daily History Scan Rate**
  - Set to -1 to disable polling for Daily History records.
- **Record Info Scan Rate**
  - Set to a positive number to launch the RecordInfoPoller with specified poll rate. With this setup, EventPoller will be notified on any 'RecordInfo' change such as 'Last Record Sequence number Used', etc.

At this point, the EFM ABB Totalflow driver is configured and is polling for Events at the rate specified in the EFM ABB Totalflow device configuration.

MQTT Transmission is connected to the MQTT Server and as a result MQTT Engine is receiving tag change events. In addition, because an EFM ABB Totalflow device has been created and configured with the same Sparkplug Group ID, Edge Node ID, and Device ID, history data will also be pushed to the MQTT server as Sparkplug RECORD objects. When new history data is polled by the EFM ABB Totalflow driver, they will be published to the MQTT server, consumed by MQTT Engine, passed on to MQTT Recorder, and then inserted into the specified database. Below is a view of some events records using a third party database viewing tool.

Additional Resources

- Inductive Automation's Ignition download with free trial
  - https://inductiveautomation.com/downloads/
- Azure Injector download with free trial
  - https://inductiveautomation.com/downloads/third-party-modules
- Questions about this tutorial?
  - Check out the Cirrus Link Forum: https://forum.cirrus-link.com/
  - Contact support: support@cirrus-link.com
- Sales questions
  - Email: sales@cirrus-link.com
  - Phone: +1 (844) 924-7787
- About Cirrus Link
  - https://www.cirrus-link.com/about-us/