MQTT Transmission Transmitters and Tag Trees

Abstract

This page describes how MQTT Transmission Transmitter configurations interact with Ignition tag trees to publish MQTT messages and tags to an MQTT Server. It explains how tags get identified to be published as well as the specific topics that data will be published on. It also goes over some example configurations to show how the system will behave in different scenarios.

> ⚠️ The combination of the Transmitter configuration and Ignition tag trees provides the ultimate flexibility in how published data is described. It is important to understand these concepts when defining your system data hierarchy.

Sparkplug Overview

All MQTT clients using the Sparkplug specification will publish MQTT messages consisting of a topic and payload component. For the Sparkplug B specification, which is used by MQTT Transmission, the Topic Namespace structure is defined as

spBv1.0/GROUP_ID/VERB/EDGE_NODE_ID/DEVICE_ID

where

- GROUP_ID is meant to represent a logical grouping of Edge Nodes. This can be a region, a facility name, or any meaningful grouping of Edge Nodes within your application.
- VERB provides an indication on how to handle the payload of the message. This includes whether the message is a birth certificate, death certificate, data message, command message, etc.
- EDGE_NODE_ID is the ID of the logical Edge Node. This is often the name of the system running Ignition with MQTT Transmission.
- DEVICE_ID is the ID of a device attached to the Ignition instance. This can be a PLC or logical grouping of tags that represent a physical or logical device connected to Ignition.

In the Sparkplug specification there are two Sparkplug Descriptors defined as:

- Edge Node Descriptor which is the GROUP_ID and EDGE_NODE_ID combination
- Device Descriptor which is the GROUP_ID, EDGE_NODE_ID and DEVICE_ID combination

The MQTT topics are essentially the 'addresses' of the different components in the distributed system and the Sparkplug Edge Node Descriptor must be unique within the distributed system. Because these are used as 'addresses' in the system you should never have two that conflict with each other - it is a bit like having two houses with the same postal address. It isn't possible for other MQTT clients in the system to tell which edge node sent the data and, when sending messages to them, they will both receive the messages.

Of course, you can have two or more Edge Nodes with the same GROUP_ID as long as each EDGE_NODE_ID is unique. To use the analogy above, it would be like having houses on the same street with different house numbers.

Transmitters and Tag Trees

MQTT Transmission is designed to dynamically pick up tags in Ignition or Ignition Edge and publish tag change events to an MQTT Server. How the tags are published to the MQTT Server is based on a combination of a 'Transmitter' configuration as well as the arrangement of tags in the Ignition tag tree.

A transmitter will use the tag folder hierarchy under the configured Tag Path in combination with the Sparkplug IDs to dynamically build the **topic namespace** used to represent the tags.

The following the rules below are used when building the topic namespace:

- If the Sparkplug IDs are empty, there must be at least 3 folder levels below the Tag Path to resolve the Group, Edge and Device IDs (if needed) where the Group and Edge combination is unique.
- If the Sparkplug Group ID only is configured, there must be at least 2 folder levels below the Tag Path to resolve to Edge and Device IDs (if needed) where the Group and Edge combination is unique.
- If the Sparkplug Group ID and Edge ID only are configured, there must be least 1 folder level or device level UDT below the Tag Path to resolve to a Device ID (if needed)
- If the Sparkplug IDs are all configured the tag folder hierarchy will not be used

MQTT Transmission will then use the remainder of the tag folder hierarchy not used in the topic namespace to identify the name of the tag included in the payload metrics.

⚠️

⚠️ Each Edge Node Descriptor will generate a single MQTT Client connection. If your tag tree has multiple folders that are identified as Edge Node IDs, a single Transmitter configuration will result in multiple Sparkplug MQTT clients publishing messages to an MQTT Server.

The MQTT Transmission Servers tab will show the number of MQTT Clients - in the image below, the Connected value 3 of 3 indicates 3 MQTT Clients have been created and 3 MQTT Clients are connected to the MQTT Server
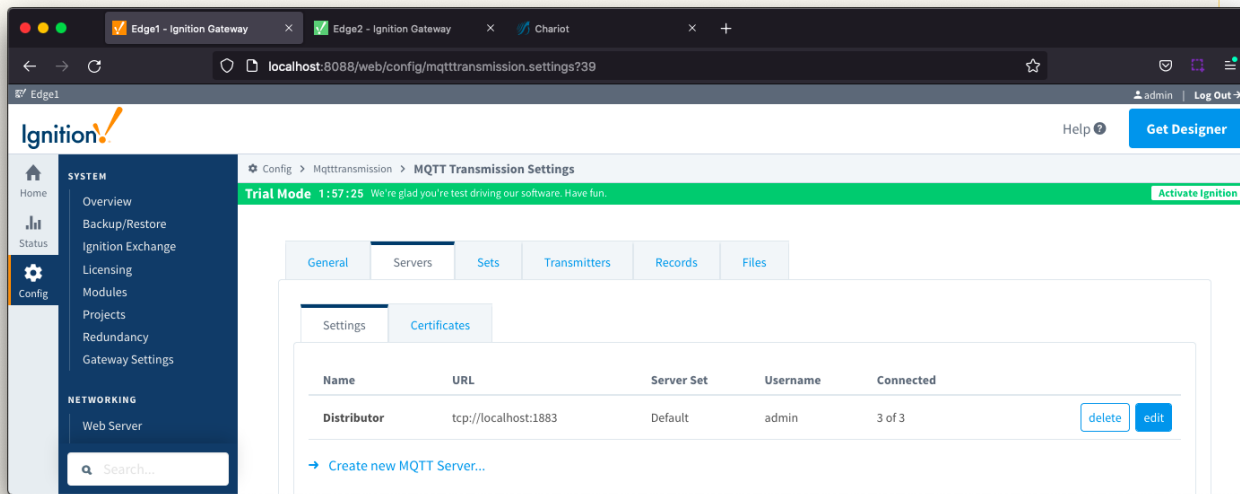
Use the How do I know how many MQTT clients are connected from MQTT Transmission to see the MQTT Client IDs and associated Sparkplug EdgeNode Descriptors

Let's take a look at some examples that demonstrate how the message topic and payload are affected by the tag tree and Transmitter configurations starting with a basic Ignition tag folder hierarchy conforming to the GroupID, EdgeNodeID and DeviceID structure and the default Transmitter configuration.
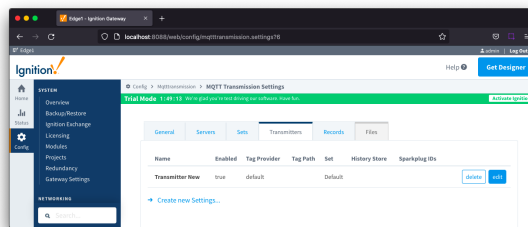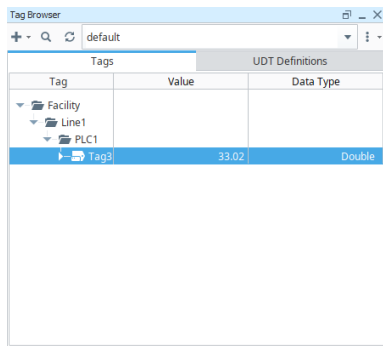
With this configuration of an empty Tag Path and empty Sparkplug IDs, Transmission will dynamically pickup the folders from the root of the tag provider building the Sparkplug Descriptors as:

- Edge Node Descriptor = Facility/Line1
- Device Descriptor = Facility/Line1/PLC1

Since there is only one Edge Node Descriptor identified, only one MQTT Client will be created.

When tag data changes, there is only one publish topic from this MQTT Client with the payload metrics containing the tag name Tag3 :

- Topic: spBv1.0/Facility/DDATA/Line1/PLC1
  - Payload: {"timestamp":1643823757750,"metrics":[{"name":"Tag3","timestamp":1643823756467,"dataType":"Double","value":33.02}],"seq":23}

Let's extend the Ignition tag folder with this example:
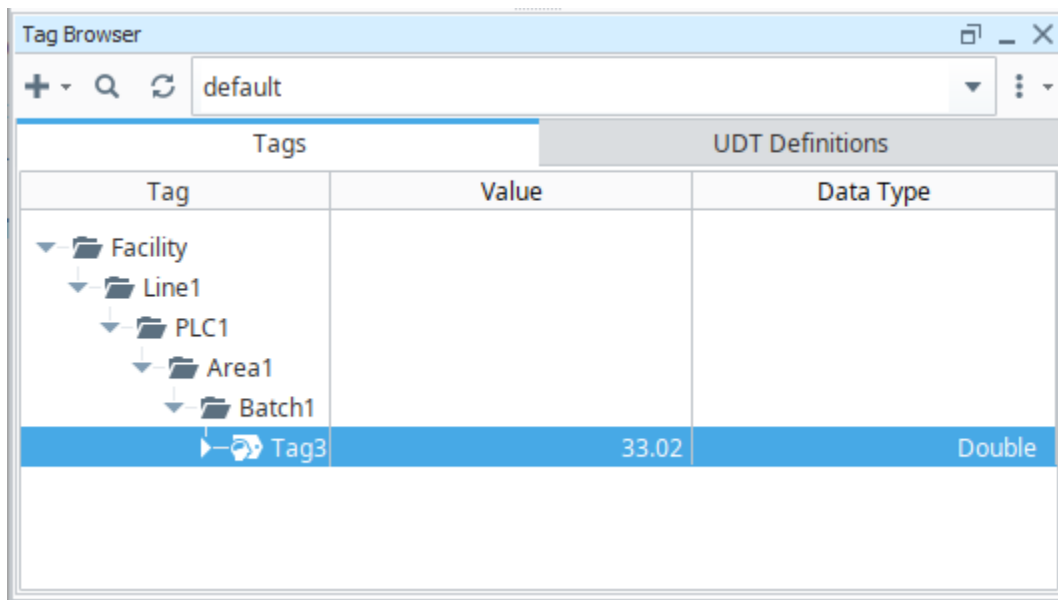
With the same configuration, there is no change to the message topic of spBv1.0/Facility/DDATA/Line1/PLC1 but the payload metrics will use the remainder of the Ignition tag path for the tag name:

- Payload: {"timestamp":1643823757750,"metrics":[{"name":"Area1/Batch1/Tag3","timestamp":1643823756467,"dataType":"Double","value": 33.02}],"seq":23}

Now let's see how changing the Tag Path and Sparkplug IDs in the Transmitter configuration affects the message topic and payload name metric for the Ignition tag based on the topic namespace rules list above:

TagPath	Sparkplug Group ID	Sparkplug Edge Node ID	Sparkplug Device ID	Sparkplug Topic	Tag Name in payload metrics
				spBv1.0/Facility/DDATA/Line1/PLC1	Area1/Batch1/Tag3
	My Group			spBv1.0/My Group/DDATA/Facility/Line1	PLC1/Area1/Batch1/Tag3
Facility	My Group			spBv1.0/My Group/DDATA/Line1/PLC1	Area1/Batch1/Tag3
Facility/Line1				spBv1.0/PLC1/DDATA/Area1/Batch1	Tag3
Facility/Line1/PLC1 /Area1/Batch1	My Group	My Edge Node	My Device	spBv1.0/My Group/DDATA/My Edge Node/My Device	Tag3
Facility/Line1/PLC1	My Group			spBv1.0/My Group/DDATA/Area1/Batch1	Tag3
Facility1/Line1/PLC1	My Group	My Edge Node		spBv1.0/My Group/DDATA/MyEdgeNode /Area1	Batch1/Tag3
	My Group	My Edge Node		spBv1.0/My Group/DDATA/MyEdgeNode /Facility1	Line1/PLC1/Area1/Batch1 /Tag3

As you can see the combination of Tag Path and Sparkplug IDs provides immense flexibility in your Sparkplug namespace configuration.

Let's add an additional tag and tag folder with the configuration of an empty Tag Path and empty Sparkplug IDs:

Transmission will build the following Sparkplug Descriptors:

- Edge Node Descriptor
    1. Facility/Line1
- Device Descriptors
    1. Facility/Line1/PLC1
    2. Facility/Line1/PLC2

As there is still only one Edge Node Descriptor, only one MQTT Client will be created.

When tag data changes, there are two publish topics from this MQTT Client with the payload metrics containing the tag names Tag3 and Tag1.

- Topic: spBv1.0/Facility/DDATA/Line1/PLC1
    - Payload: {"timestamp":1643823757750,"metrics":[{"name":"Tag3","timestamp":1643823756467,"dataType":"Double","value":1.88}]," seq":23}
- Topic: spBv1.0/Facility/DDATA/Line1/PLC2
    - Payload: {"timestamp":1643823757750,"metrics":[{"name":"Tag1","timestamp":1643823722222,"dataType":"Double","value":85.16}]," seq":24}

Let's add an additional tags and tag folders:

Transmission will build the following Sparkplug Descriptors:

- Edge Node Descriptors
  1. Facility/Line1
  2. Facility/Line2
- Device Descriptors
  1. Facility/Line1/PLC1
  2. Facility/Line1/PLC2
  3. Facility/Line2/PLC1
  4. Facility/Line2/PLC2

Now there are two Edge Node Descriptors which results in the creation of two MQTT Clients.

When tag data changes, there are two publish topics from each MQTT Client with the payload metrics containing the tag names Tag1, Tag2 and Tag3.

- Topic: spBv1.0/Facility/DDATA/Line1/PLC1
  - Payload: {"timestamp":1643823757750,"metrics":[{"name":"Tag3","timestamp":1643823756467,"dataType":"Double","value":60.22}]" seq":23}
- cspBv1.0/Facility/DDATA/Line1/PLC2
  - Payload: {"timestamp":1643823757750,"metrics":[{"name":"Tag1","timestamp":1643823722222,"dataType":"Double","value":53.79}]," seq":17}
- Topic: spBv1.0/Facility/DDATA/Line2/PLC1
  - Payload: {"timestamp":1643823757750,"metrics":[{"name":"Tag1","timestamp":1643823756467,"dataType":"Double","value":60.22}]," seq":19}
- Topic: spBv1.0/Facility/DDATA/Line2/PLC2
  - Payload: {"timestamp":1643823757750,"metrics":[{"name":"Tag2","timestamp":1643823722222,"dataType":"Double","value":60.22}]," seq":57}

As you can see, we haven't made any changes to the Transmitter configuration and MQTT Transmission has dynamically created the MQTT Clients, topic and payloads from the Ignition tag tree.

Let's extend our infrastructure to add a second facility with a Transmitter configured with an empty Tag Path and Sparkplug IDs.

At this second facility, Transmission will build the following Sparkplug Descriptors:

- Edge Node Descriptor
    1. Facility2/Line1
- Device Descriptor
    1. Facility2/Line1/PLC1
    2. Facility2/Line1/PLC2

Data published from this second facility will use the topics spBv1.0/Facility2/DDATA/Line1/PLC1 and spBv1.0/Facility2/DDATA/Line1/PLC2.

Since the Edge Node Descriptors across the infrastructure are unique, there will be no data conflict at any MQTT Clients subscribing to the data messages.

Now let's consider extending the original configuration to 100 locations without having to make changes to the tag tree. How would we avoid Edge Node collisions in this scenario?

Here is where we can leverage the Transmitter Tag Path and Sparkplug ID configurations.

At each facility we can build a transmitter configuration with a Tag Path of Facility and set a specific GROUP_ID corresponding to the location such as :

With this configuration, MQTT Transmission will pickup the next two folder levels under the Tag Path to use as the EDGE_NODE_ID and DEVICE_ID resulting in the following:

- Edge Node Descriptors
    - Plant_1/Line1
    - Plant_1/Line2
    - Plant_2/Line1
    - Plant_2/Line2

through to

    - Plant_100/Line1
    - Plant_100/Line2

Topics for data published would then become:

- spBv1.0/Plant_1/DDATA/Line1/PLC1
- spBv1.0/Plant_1/DDATA/Line1/PLC1
- spBv1.0/Plant_2/DDATA/Line1/PLC1
- spBv1.0/Plant_2/DDATA/Line1/PLC1

through to

- spBv1.0/Plant_100/DDATA/Line1/PLC1
- spBv1.0/Plant_100/DDATA/Line1/PLC1

The infrastructure would have 200 MQTT Clients with unique Edge Node Descriptors across the infrastructure.

Additional Resources

- Inductive Automation's Ignition download with free trial
    - Current Ignition Release
- Cirrus Link Solutions Modules for Ignition
    - Ignition Strategic Partner Modules
- Support questions
    - 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/