

Transmitters with Multi-Tag Paths

Prerequisites:

- Understand the concepts discussed in the [MQTT Transmission Transmitters and Tag Trees](#) which describes how MQTT Transmission Transmitter configurations interact with Ignition tag trees to publish MQTT messages and tags to an MQTT Server.

Overview:

MQTT Transmission allows for a multi-tag path to be defined for a Transmitter. This can be useful for when you have tags across multiple Edge Gateway tag providers so they can be consumed by a single Transmitter and contained within a single Edge Node and/or device by MQTT Engine rather than split across two different edge nodes.

There are some basic rules to the multi-tag path configuration:

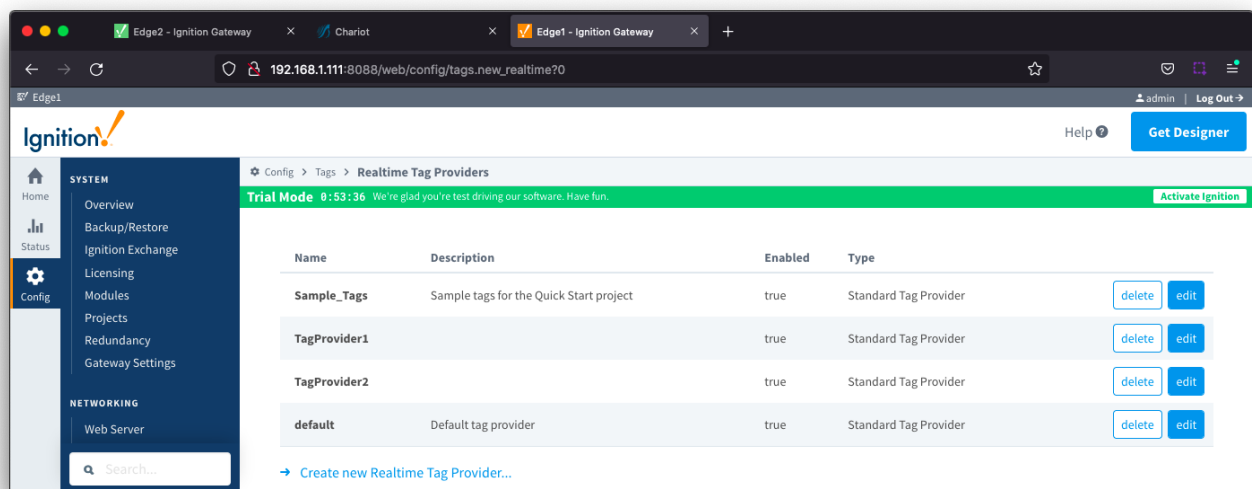
- Tag Provider and Tag Path are used in combination to define the start of each tag tree
- Specific Tag Providers can be specified in the Tag Path using the format `[TagProvider]`
 - If the Tag Provider is not explicitly defined in the Tag Path, Transmission will use the Tag Provider property
 - If the Tag Provider is explicitly defined in the Tag Path, Transmission will replace the Tag Provider property with the `[TagProvider]` configured
- Multiple tag paths are comma separated in the Tag Path property for each Transmitter

Let's put some examples together showing how the multi-tag path feature works.

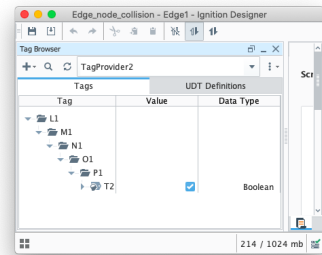
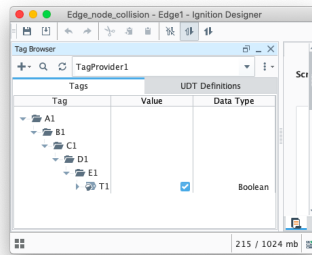
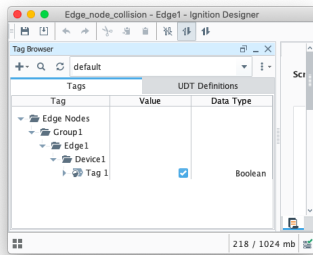
Following the [Ignition Tag Provider Guide](#), configure two new Realtime Tag Providers as type Standard Tag Provider for your Ignition instance where MQTT Transmission is running. Name them TagProvider1 and TagProvider2.



The "default" Tag Provider is automatically created when Ignition is installed on the gateway.

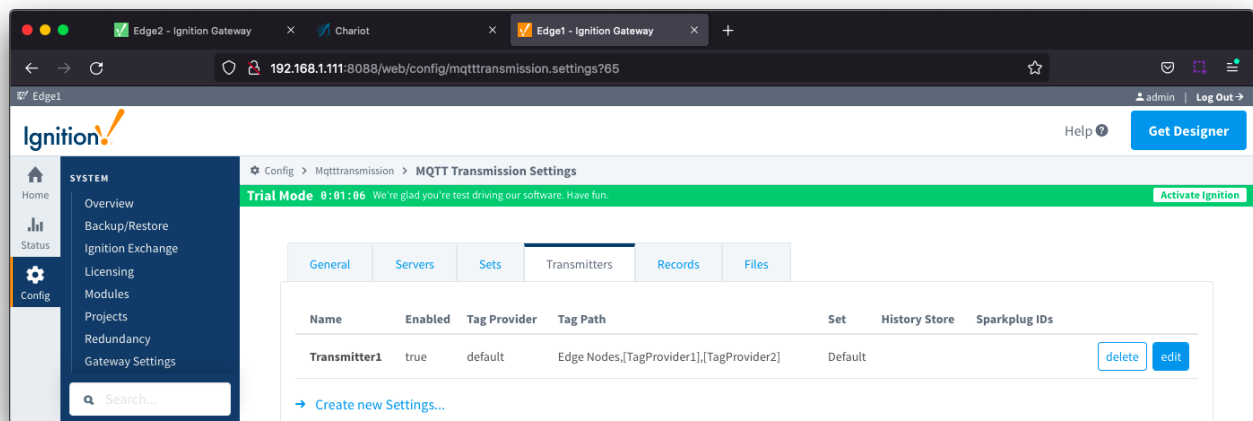


In Ignition Designer, create a tag tree under each Tag Provider as shown below:



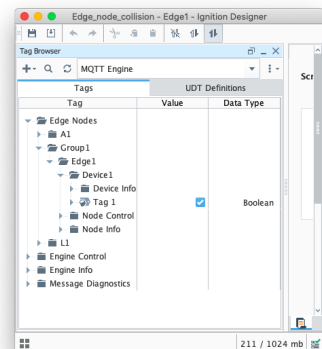
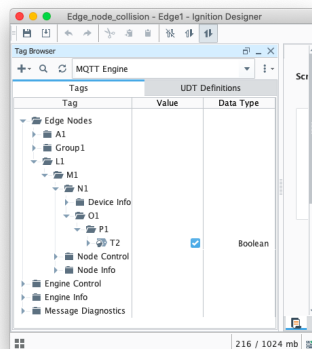
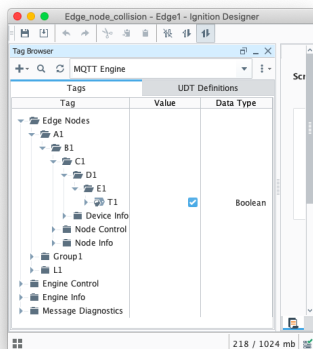
Let's create a transmitter using these properties and use [MQTT Transmission refresh mechanism](#) to publish the tags

Name	Tag Provider	Tag Path
Transmitter1	default	Edge Nodes,[TagProvider1],[TagProvider2]



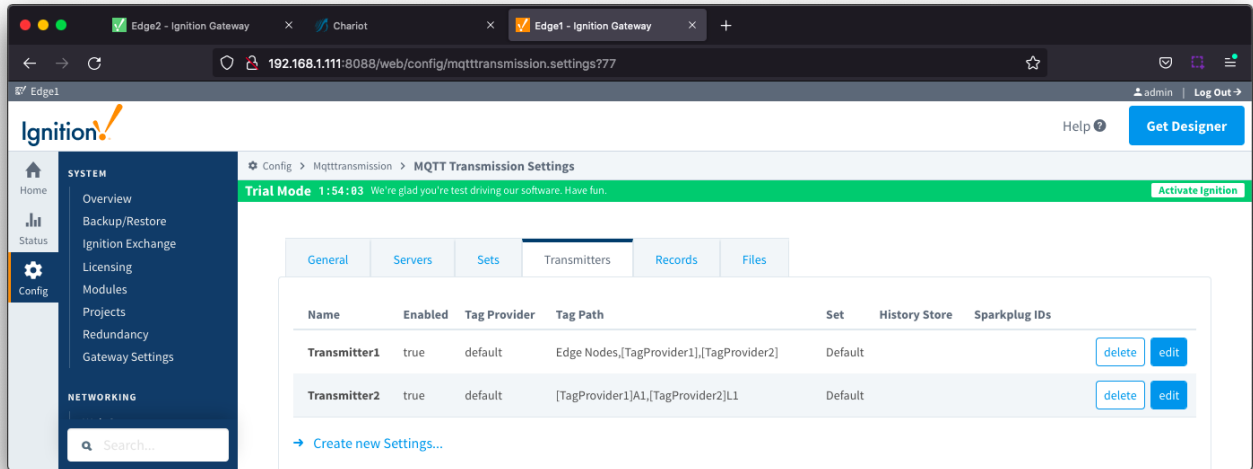
At MQTT Engine, we will see the following three tag trees each starting at one of the three Tag Paths specified:

- Edge Nodes
- TagProvider1
- TagProvider2



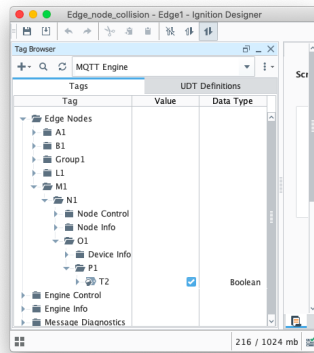
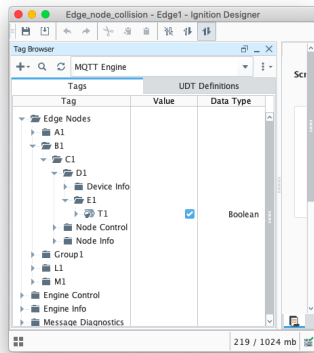
Let's create a second transmitter using these properties and use [MQTT Transmission refresh mechanism](#) to publish the tags

Name	Tag Provider	Tag Path
Transmitter2	default	[TagProvider1]A1,[TagProvider2]L1



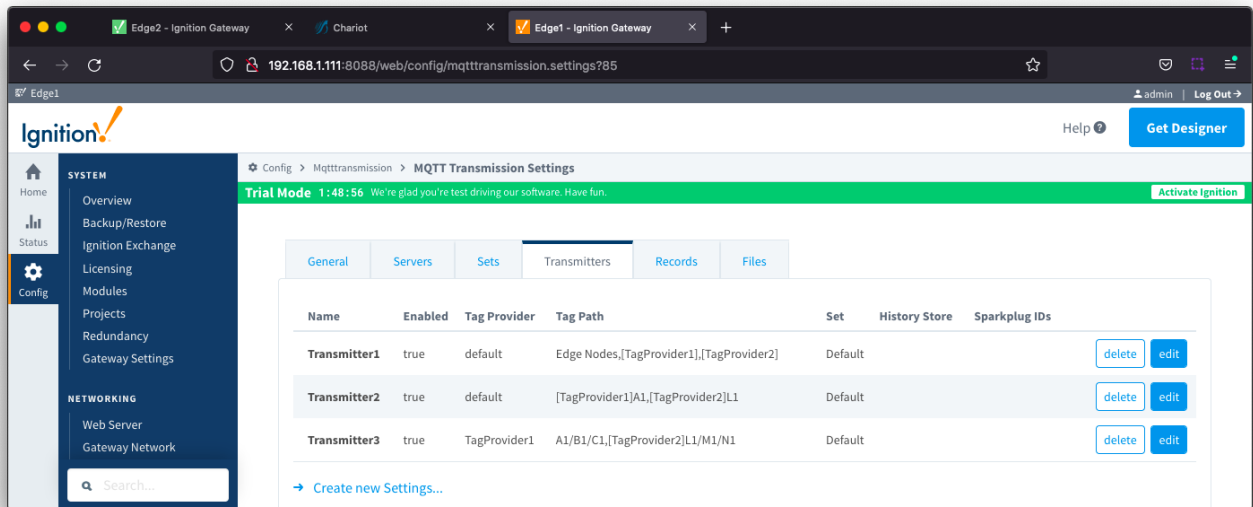
At MQTT Engine, we will see two additional tag trees created each starting at one of the two Tag Paths specified

- TagProvider1/A1
- TagProvider2/L1



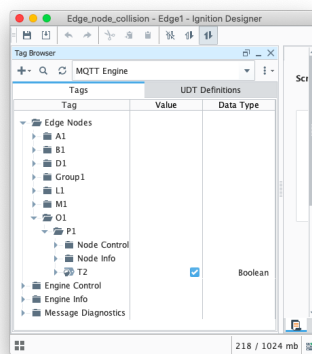
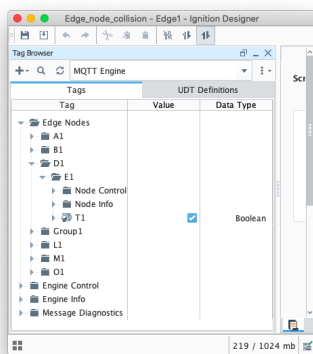
Let's create a third transmitter using these properties and use [MQTT Transmission refresh mechanism](#) to publish the tags

Name	Tag Provider	Tag Path
Transmitter3	TagProvider1	A1/B1/C1,[TagProvider2]L1/M1/N1



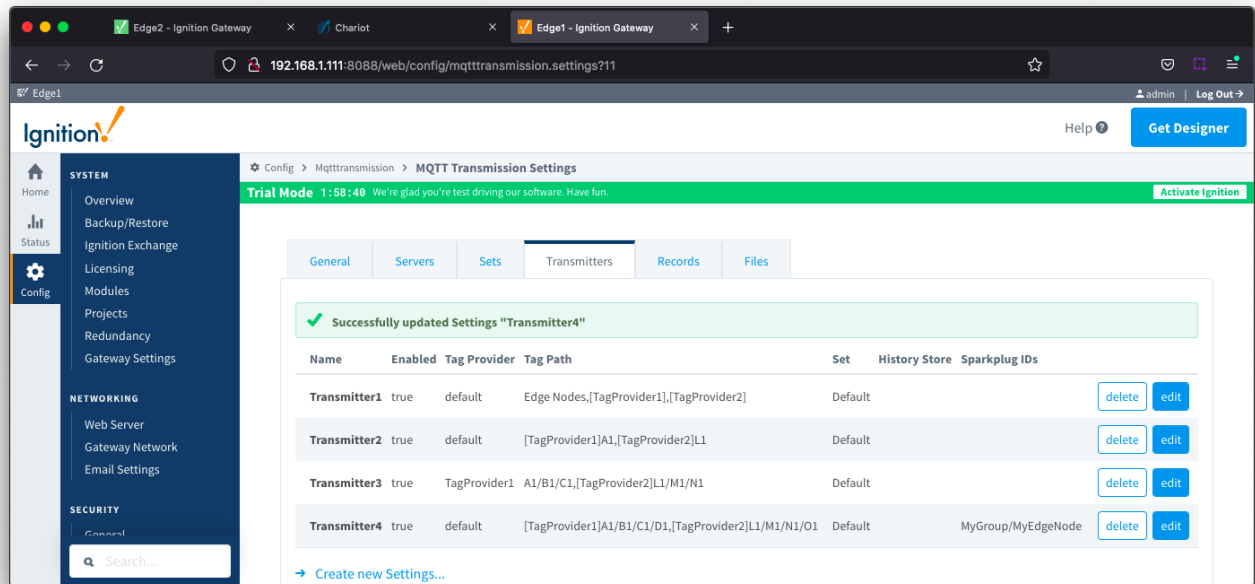
At MQTT Engine, we will see two additional two tag trees created each starting at one of the Tag Paths specified

- TagProvider1/A1/B1/C1
- TagProvider2/L1/M1/N1



All the previous examples create unique Edge Nodes with a single tag under each. Lets add a fourth transmitter but set the Group ID and Edge ID under the Sparkplug IDs to create a single Edge Node ID.

Name	Tag Provider	Tag Path	Sparkplug IDs
Transmitter4	default	[TagProvider1]A1/B1/C1/D1,[TagProvider2]L1/M1/N1/O1	MyGroup/MyEdgeNode



At MQTT Engine, we will only see one additional tag tree with two tags using the paths MyGroup/MyEdgeNode/E1 and MyGroup/MyEdgeNode/P1

