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.

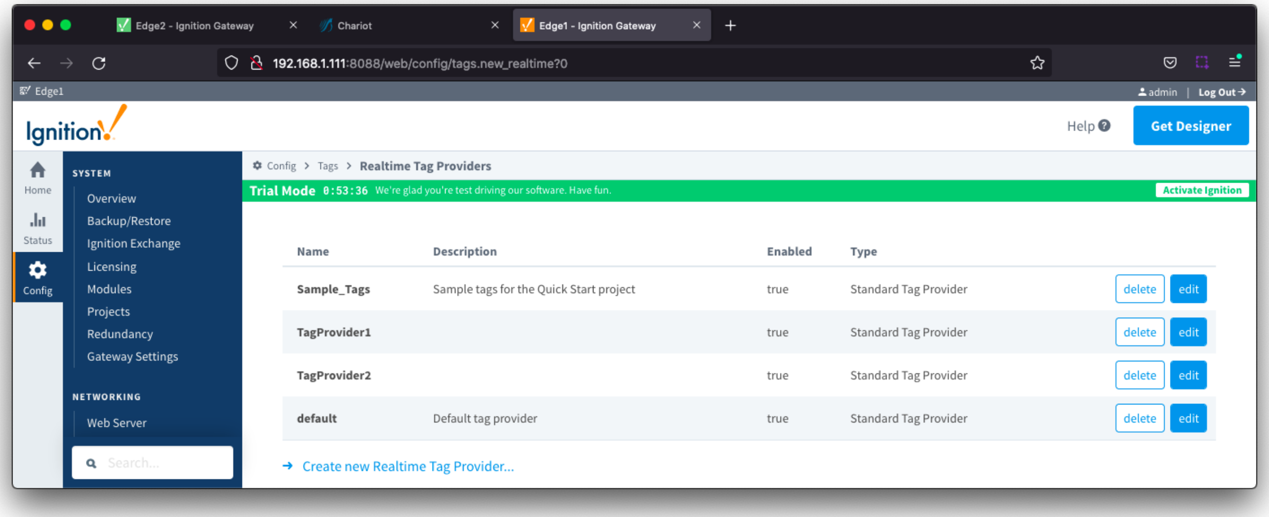
There are some basic rules to the multi-tag path configuration:

- 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.

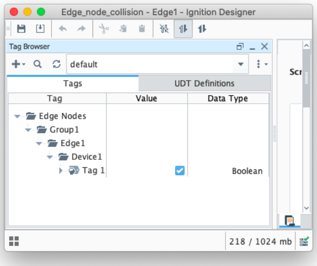


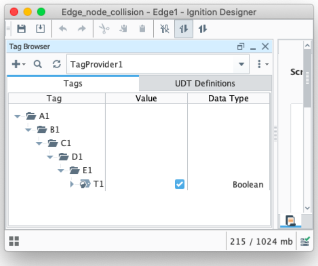
Trial Mode 0:53:36 We're glad you're test driving our software. Have fun.

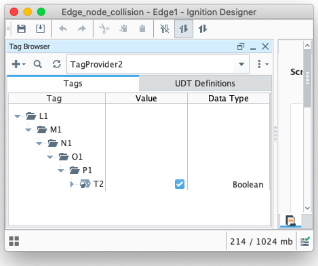
Name	Description	Enabled	Type	
Sample_Tags	Sample tags for the Quick Start project	true	Standard Tag Provider	delete edit
TagProvider1		true	Standard Tag Provider	delete edit
TagProvider2		true	Standard Tag Provider	delete edit
default	Default tag provider	true	Standard Tag Provider	delete edit

→ Create new Realtime Tag Provider...

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

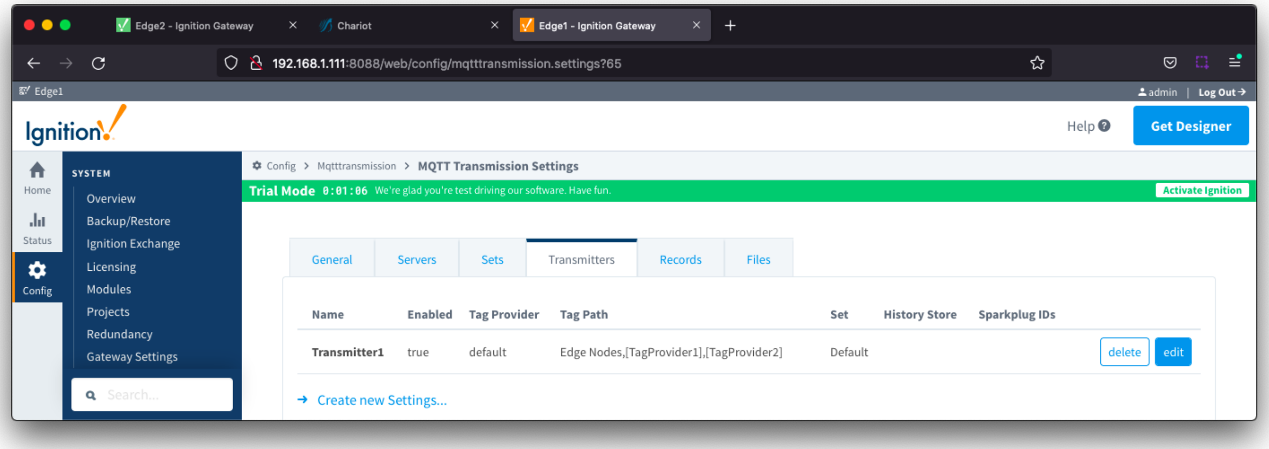






Let's create a transmitter using these properties and use MQTT Transmission refresh mechanism to publish the tags

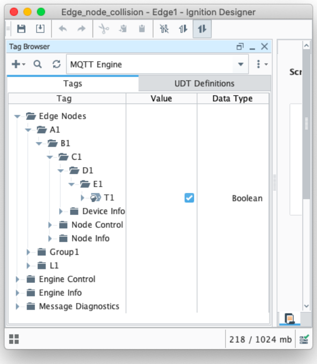
Name	Tag Provider	Tag Path
Transmitter1	default	Edge Nodes,[TagProvider1],[TagProvider2]

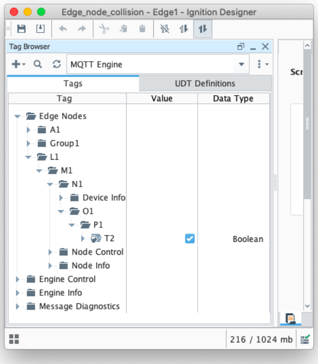


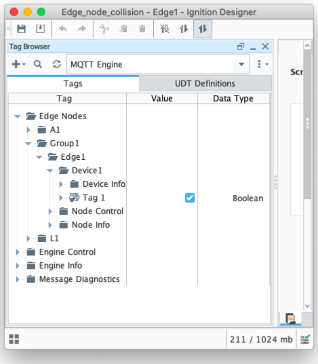
Name	Enabled	Tag Provider	Tag Path	Set	History Store	Sparkplug IDs
Transmitter1	true	default	Edge Nodes,[TagProvider1],[TagProvider2]	Default		

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

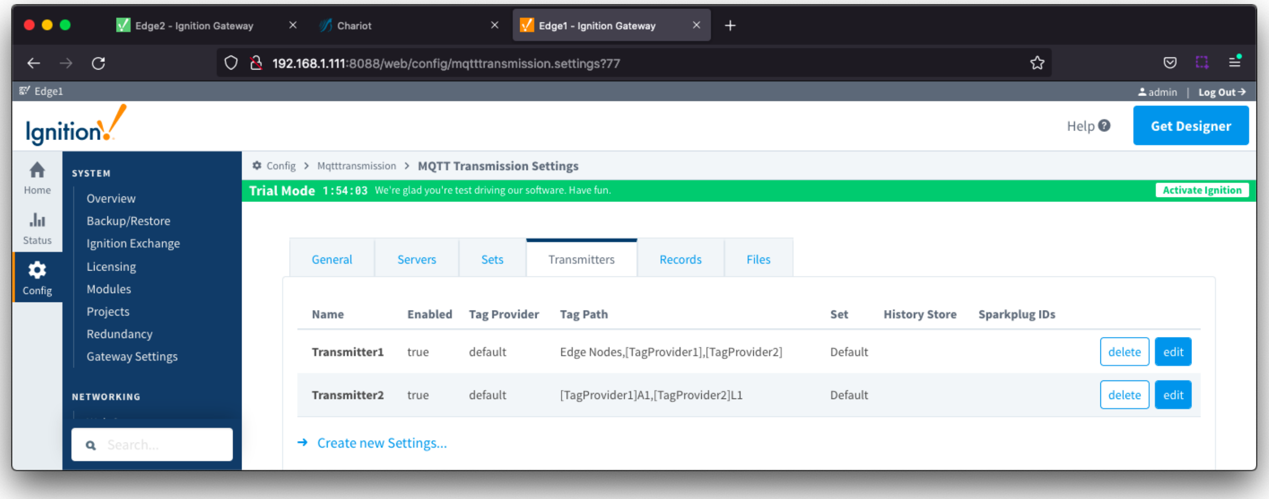






Let's create a second transmitter using these properties and use MQTT Transmission refresh mechanism to publish the tags

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

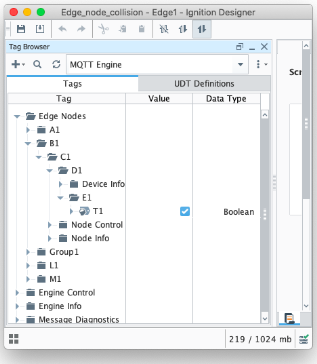


Name	Enabled	Tag Provider	Tag Path	Set	History Store	Sparkplug IDs	
Transmitter1	true	default	Edge Nodes,[TagProvider1],[TagProvider2]	Default			delete edit
Transmitter2	true	default	[TagProvider1]A1,[TagProvider2]L1	Default			delete edit

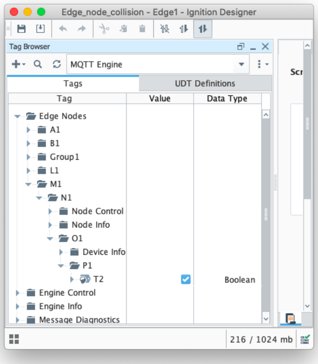
→ Create new Settings...

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



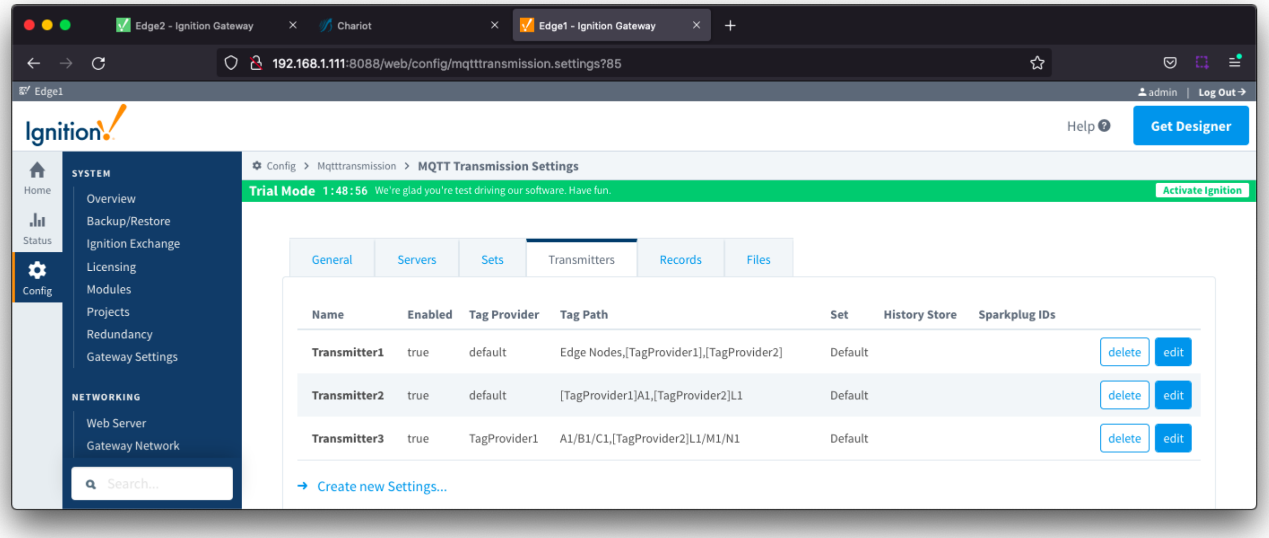
Tag	Value	Data Type
Edge Nodes		
A1		
B1		
C1		
D1		
Device Info		
E1		
T1	☑	Boolean
Node Control		
Node Info		
Group1		
L1		
M1		
Engine Control		
Engine Info		
Message Diagnostics		



Tag	Value	Data Type
Edge Nodes		
A1		
B1		
Group1		
L1		
M1		
N1		
Node Control		
Node Info		
O1		
Device Info		
P1		
T2	☑	Boolean
Engine Control		
Engine Info		
Message Diagnostics		

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

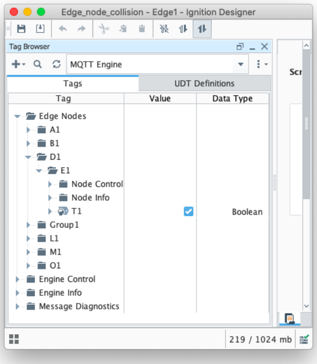


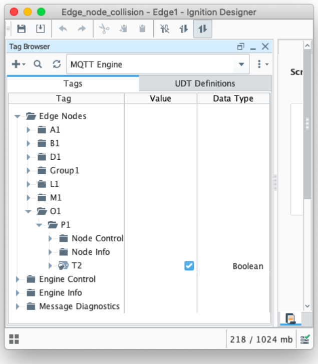
Name	Enabled	Tag Provider	Tag Path	Set	History Store	Sparkplug IDs	
Transmitter1	true	default	Edge Nodes,[TagProvider1],[TagProvider2]	Default			delete edit
Transmitter2	true	default	[TagProvider1]A1,[TagProvider2]L1	Default			delete edit
Transmitter3	true	TagProvider1	A1/B1/C1,[TagProvider2]L1/M1/N1	Default			delete edit

→ Create new Settings...

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

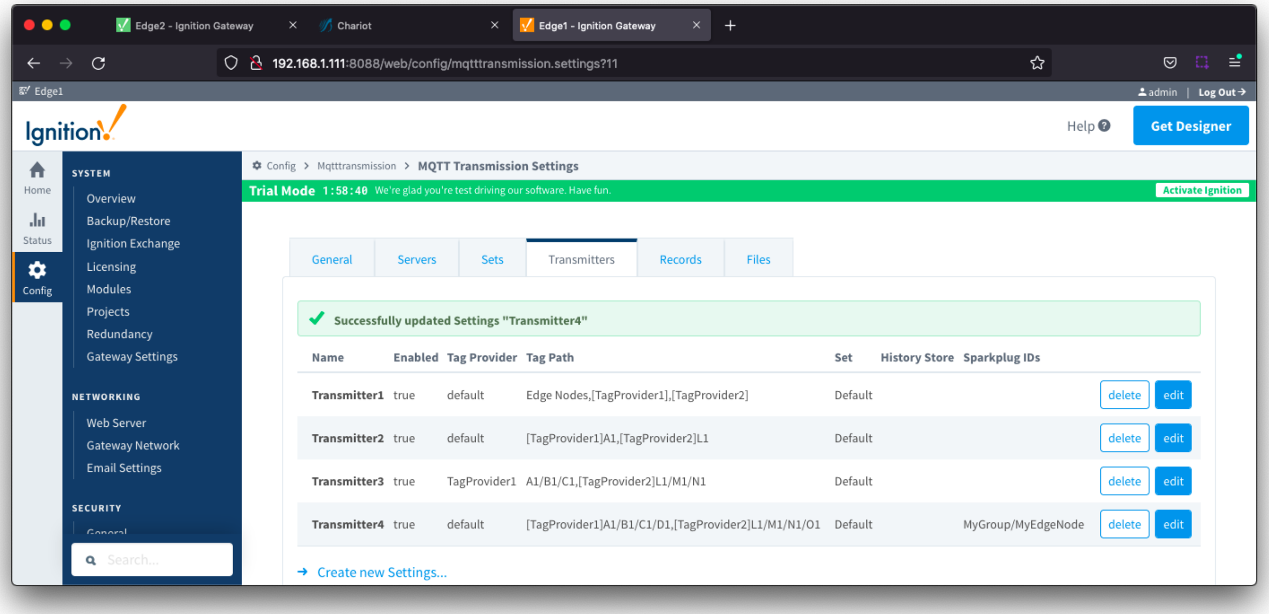
- TagProvider1/A1/B1/C1
- TagProvider2/L1/M1/N1





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 new Edge Node ID.

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



Config > Mqtttransmission > **MQTT Transmission Settings**

Trial Mode 1:58:40 We're glad you're test driving our software. Have fun.

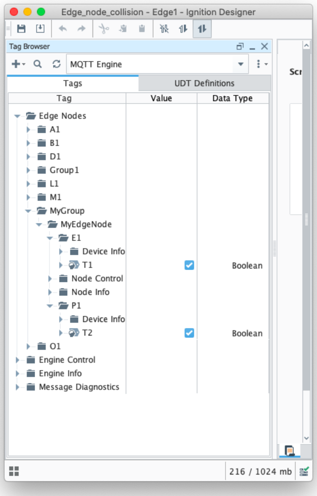
General | Servers | Sets | Transmitters | Records | Files

✓ **Successfully updated Settings "Transmitter4"**

Name	Enabled	Tag Provider	Tag Path	Set	History Store	Sparkplug IDs	
Transmitter1	true	default	Edge Nodes,[TagProvider1],[TagProvider2]	Default			delete edit
Transmitter2	true	default	[TagProvider1]A1,[TagProvider2]L1	Default			delete edit
Transmitter3	true	TagProvider1	A1/B1/C1,[TagProvider2]L1/M1/N1	Default			delete edit
Transmitter4	true	default	[TagProvider1]A1/B1/C1/D1,[TagProvider2]L1/M1/N1/O1	Default		MyGroup/MyEdgeNode	delete edit

→ Create new Settings...

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



Edge_node_collision - Edge1 - Ignition Designer

```
Tag                      Value   Data Type
Edge Nodes
  A1
  B1
  D1
  Group1
  L1
  M1
  MyGroup
    MyEdgeNode
      E1
        Device Info
        T1               ☑       Boolean
      Node Control
      Node Info
      P1
        Device Info
        T2               ☑       Boolean
  O1
Engine Control
Engine Info
Message Diagnostics
```