

# Using MQTT Modules to map ISA 95 Enterprise Structure in a UNS Architecture

How can we use Cirrus Link's Sparkplug MQTT Transmission and MQTT Engine modules along with the common ISA-95 equipment model of Enterprise > Site > Area > Line > Equipment to map the model data into a Unified NameSpace.

## Sparkplug Basics

Sparkplug defines a standard format for MQTT message topic paths of:

spBv1.0/group\_id/Message Type/edge\_node\_id/device\_id

Sparkplug defines a standard format for MQTT message payloads of:

```
{
  "timestamp": <timestamp>,
  "metrics": [{
    "name": <metric_name>,
    "alias": <alias>,
    "timestamp": <timestamp>,
    "dataType": <datatype>,
    "value": <value>
  }],
  "seq": <sequence_number>
}
```

where the 'name' of a metric can be hierarchical to build out proper folder structures for applications consuming the metric values.

For example, a simple Sparkplug™ B payload with a two metrics can be represented in JSON as follows:

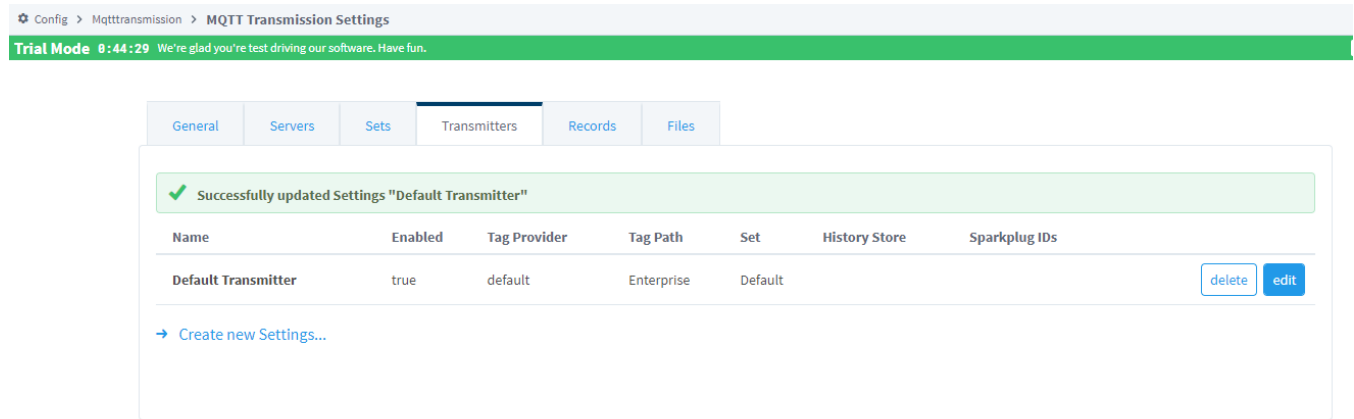
```
{
  "timestamp": 1486144502122,
  "metrics": [{
    "name": "Metric1",
    "alias": 1,
    "timestamp": 1479123452194,
    "dataType": "String",
    "value": "Test"
  },
  {
    "name": "Metric Level 1/Metric Level 2/Metric2",
    "alias": 2,
    "timestamp": 1479123452056,
    "dataType": "String",
    "value": "Test"
  }
],
  "seq": 2
}
```

Using the hierarchical functionality of the metric name, in conjunction with the group\_id, edge\_node\_id and device\_id already defined in the Topic Namespace, allows consuming Sparkplug applications to organize metrics in the same hierarchical fashion to construct the Unified NameSpace whilst taking advantage of the benefits afforded by using Sparkplug.

Let's take a look at a couple of examples using MQTT Transmission and MQTT Engine.

Review the [MQTT Transmission Transmitters and Tag Trees](#) to gain an understanding of how MQTT Transmission Transmitter configurations interact with Ignition tag trees to publish Sparkplug messages and the immense flexibility that can be realized.

We have configured an MQTT Transmission Transmitter to use the Ignition tag tree in the default tag provider at the edge to construct the Sparkplug Device Descriptor (group\_id, edge\_id and device\_id) starting at the folder level below Enterprise.

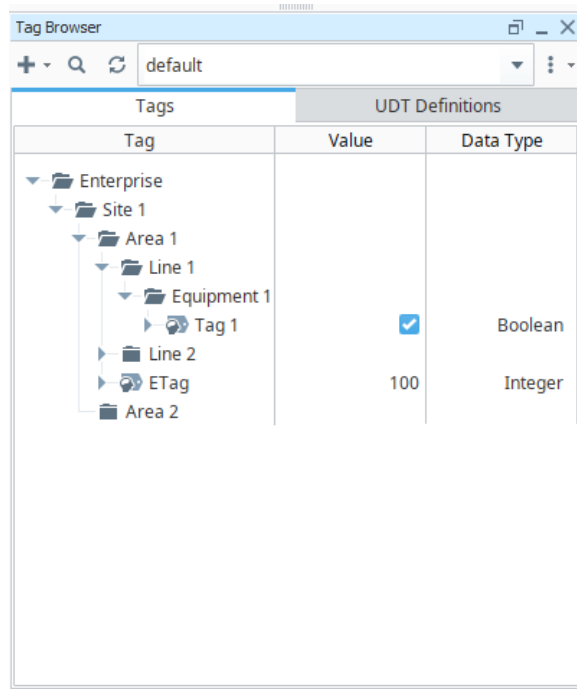


For the Ignition tag tree below, the topic and payload for a change to Tag 1 will be:

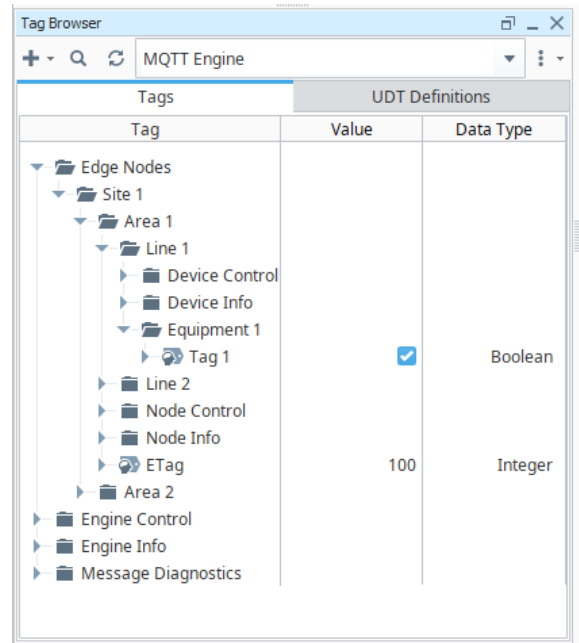
- Topic: spBv1.0/Site 1/DDATA/Area 1/Line 1
  - Payload: {"timestamp":1643823757750,"metrics":[{"name":"Equipment 1/Tag 1","timestamp":1643823756467,"dataType":"Boolean","value":True}],"seq":23}

At MQTT Engine, the topic and hierarchal metric name are parsed and the edge folder structure recreated

## Edge



## MQTT Engine



Let's look at another configuration to show the flexibility of using the Ignition tag tree along with the ability to configure the Sparkplug IDs directly.

Here we have configured the MQTT Transmitter to use the the Sparkplug Edge Node Descriptor (group\_id and edge\_node\_id) from values configured in the Transmitter with the device\_id and metric name to be picked up from the Ignition tag tree in the default provider.

Act

General
Servers
Sets
Transmitters
Records
Files

✓ Successfully updated Settings "Default Transmitter"

Name	Enabled	Tag Provider	Tag Path	Set	History Store	Sparkplug IDs
Default Transmitter	true	default		Default		Site 1/Area 1

[Create new Settings...](#)

For the Ignition tag tree below, the topic and payload for a change to Equipment 1/Tag 1 will be:

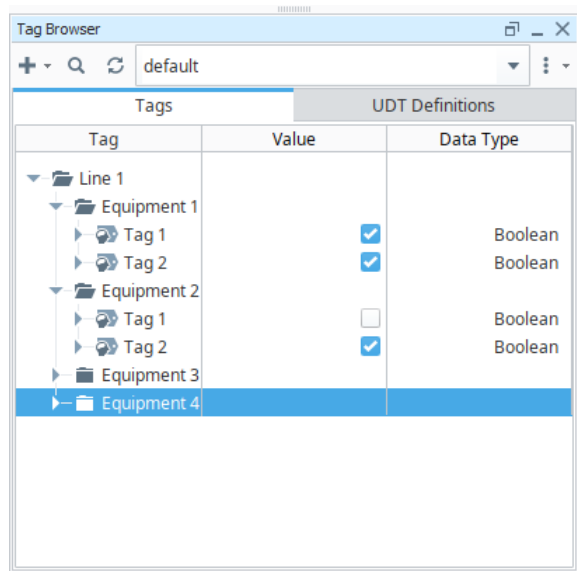
- Topic: spBv1.0/Site 1/DDATA/Area 1/Line 1
  - Payload: {"timestamp":1643823757750,"metrics":[{"name":"Equipment 1/Tag 1","timestamp":1643823756467,"dataType":"Boolean","value":True}],"seq":23}

For a change to Equipment 2/ Tag 1, the topic and payload will be:

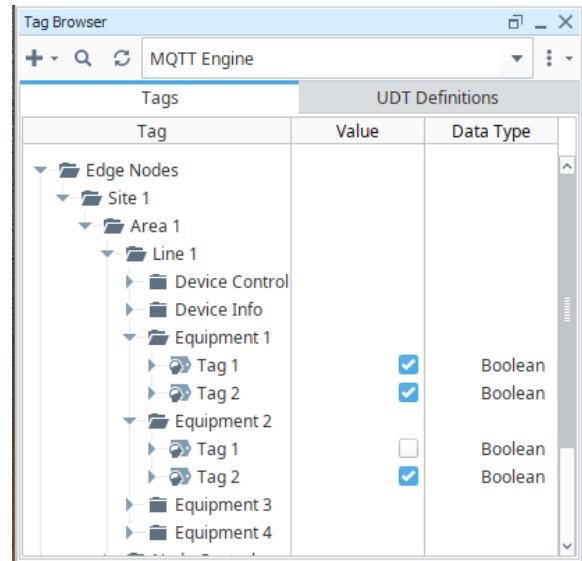
- Topic: spBv1.0/Site 1/DDATA/Area 1/Line 1
  - Payload: {"timestamp":1643823757750,"metrics":[{"name":"Equipment 2/Tag 1","timestamp":1643823756467,"dataType":"Boolean","value":FalseTrue}], "seq":23}

At MQTT Engine the topic and hierarchal metric name is parsed and the UNS structure created

## Edge



## MQTT Engine



This approach makes it simple to replicate an edge node configuration for a Line process across the plant.

If we added a couple more edge nodes with Transmitters configured as Site 1/Area 2 and Site 2/Area 1 whilst leaving the Ignition tag tree the same at each location

Config > Mqtttransmission > MQTT Transmission Settings

Trial Mode 9:02:14 We're glad you're test driving our software. Have fun.

GeneralServersSetsTransmittersRecordsFiles

✓ Successfully updated Settings "Default Transmitter"

Name	Enabled	Tag Provider	Tag Path	Set	History Store	Sparkplug IDs	
Default Transmitter	true	default		Default		Site 1/Area 2	<div>deleteedit</div>

→ Create new Settings...

Config > Mqtttransmission > MQTT Transmission Settings

Trial Mode 9:01:25 We're glad you're test driving our software. Have fun.

GeneralServersSetsTransmittersRecordsFiles

✓ Successfully updated Settings "Default Transmitter"

Name	Enabled	Tag Provider	Tag Path	Set	History Store	Sparkplug IDs	
Default Transmitter	true	default		Default		Site 2/Area 1	<div>deleteedit</div>

→ Create new Settings...

Data would be published on the following topics:

- spBv1.0/Site 1/DDATA/Area 1/Line 1
- spBv1.0/Site 1/DDATA/Area 1/Line 2
- spBv1.0/Site 1/DDATA/Area 2/Line 1
- spBv1.0/Site 1/DDATA/Area 2/Line 1
- spBv1.0/Site 2/DDATA/Area 1/Line 1
- spBv1.0/Site 2/DDATA/Area 1/Line 2

and MQTT Engine will parse the topics and payloads to create the UNS structure

Tag Browser		
<div> <div>+</div> <div>Q</div> <div>↺</div> <div>MQTT Engine</div> <div>▼</div> <div>⋮</div> </div>		
Tags	UDT Definitions	
Tag	Value	Data Type
<div> <div>▼</div> <div>Edge Nodes</div> <div> <div>▼</div> <div>Site 1</div> <div> <div>▼</div> <div>Area 1</div> <div> <div>▼</div> <div>Line 1</div> <div> <div>▶</div>Device Control <div>▶</div>Device Info <div>▶</div>Equipment 1 <div>▶</div>Equipment 2 <div>▶</div>Equipment 3 <div>▶</div>Equipment 4 <div>▶</div>Node Control <div>▶</div>Node Info </div> <div>▼</div> <div>Area 2</div> <div> <div>▼</div> <div>Line 1</div> <div> <div>▶</div>Device Control <div>▶</div>Device Info <div>▶</div>Equipment 1 <div>▶</div>Equipment 2 <div>▶</div>Equipment 3 <div>▶</div>Equipment 4 <div>▶</div>Node Control <div>▶</div>Node Info </div> </div> <div>▼</div> <div>Site 2</div> <div> <div>▼</div> <div>Area 1</div> <div> <div>▼</div> <div>Line 1</div> <div> <div>▶</div>Device Control <div>▶</div>Device Info <div>▶</div>Equipment 1 <div>▶</div>Equipment 2 <div>▶</div>Equipment 3 <div>▶</div>Equipment 4 <div>▶</div>Node Control </div> </div> </div> </div> </div> </div></div>		