# Using MQTT Transmissions 'Refresh' Mechanism

**Prerequisites:**

- Knowledge of Ignition and Module installation process: Cirrus Link Module Installation
- Install the following MQTT Modules
  - MQTT Distributor
    - v4.0.X if using Ignition 8.0.x
  - MQTT Engine
    - v4.0.X if using Ignition 8.0.x
  - MQTT Transmission
    - v4.0.X if using Ignition 8.0.x
- Ensure you are familiar with Transmitter configurations: MQTT Transmission Transmitters and Tag Trees

**Overview:**

MQTT Transmission provides a mechanism for publishing newly added tags to the tag tree for which a Transmitter publishes on. This must be done any time new tags are added to the tag tree or tags are moved within the tag tree. It must also be done any time properties of tags are modified.

Since MQTT Transmission only publishes tag updates on Value or Quality changes, this mechanism can also be useful in refreshing slow changing tag values so that downstream processes can confirm that the system is operating correctly.

This can be done manually or via a script.

**Refreshing the Tag Tree for all Transmitters and Edge Nodes:**

Consider the following Transmitter configuration in Transmission.
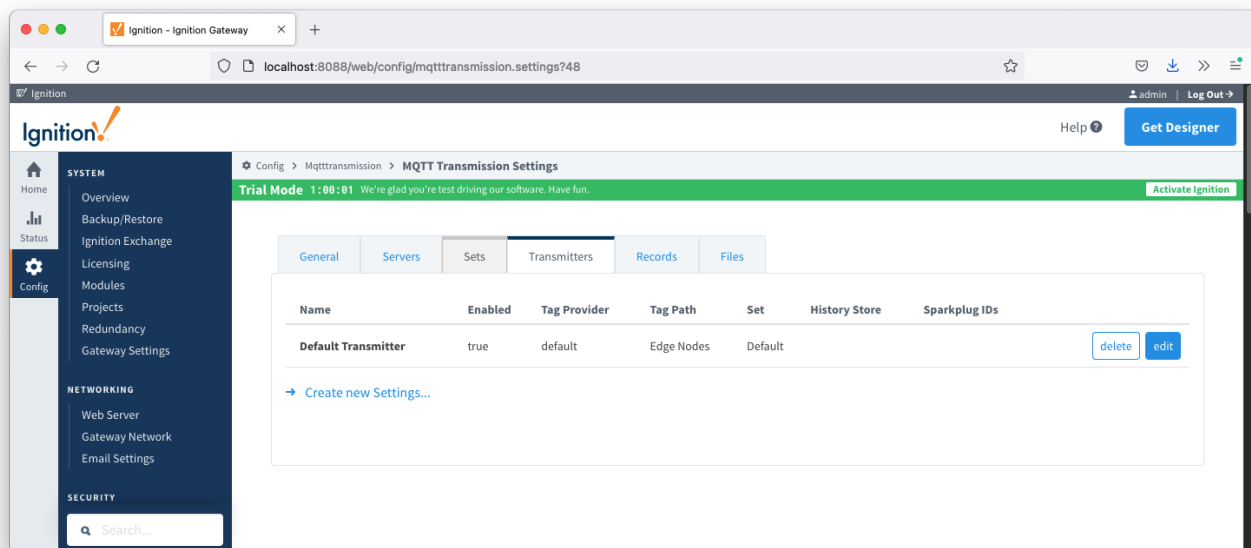


This Transmitter will look in the '[default]Edge Nodes' folder for tags to publish via MQTT. Because no Sparkplug IDs are defined in the Transmitter configuration, the first folder under the 'Edge Nodes' folder will be the Sparkplug Group ID. The second level folders will be the Sparkplug Edge Nodes IDs. So, given the following tag tree:
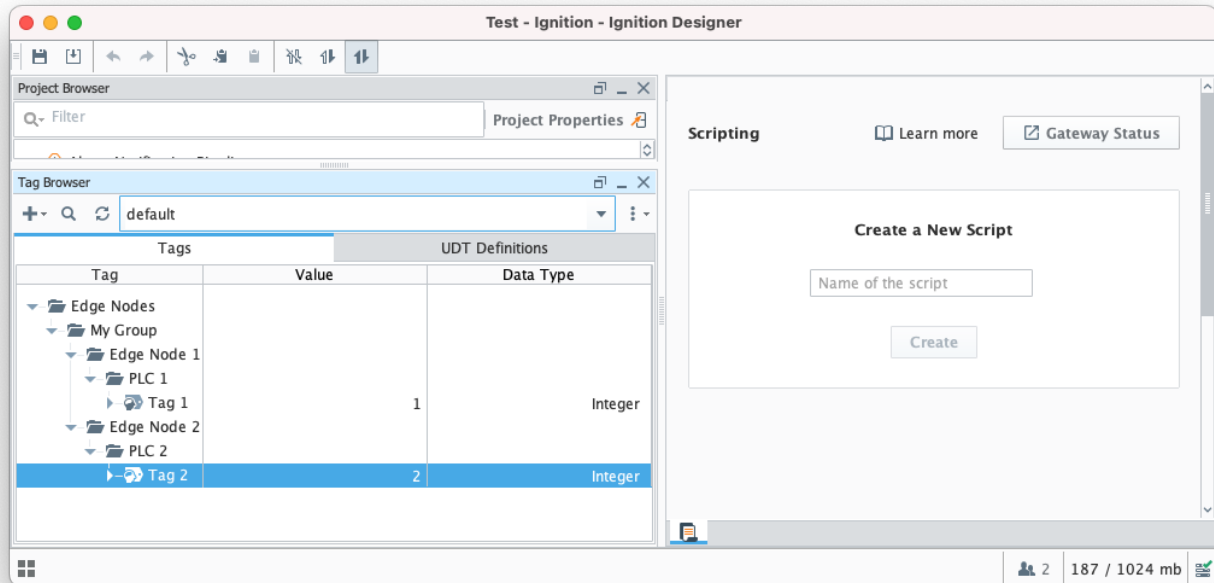
There are two Edge Nodes (each with their own MQTT client) with the Sparkplug Group and Edge Node IDs of:

- My Group/Edge Node 1
    - This Edge Node has a single Sparkplug Device called 'PLC 1'
- My Group/Edge Node 2
    - This Edge Node has a single Sparkplug Device called 'PLC 2'

Browse to the MQTT Transmission '[MQTT Transmission]Transmission Control/Refresh' tag as shown below and write a value of true to it by clicking the boolean checkbox:
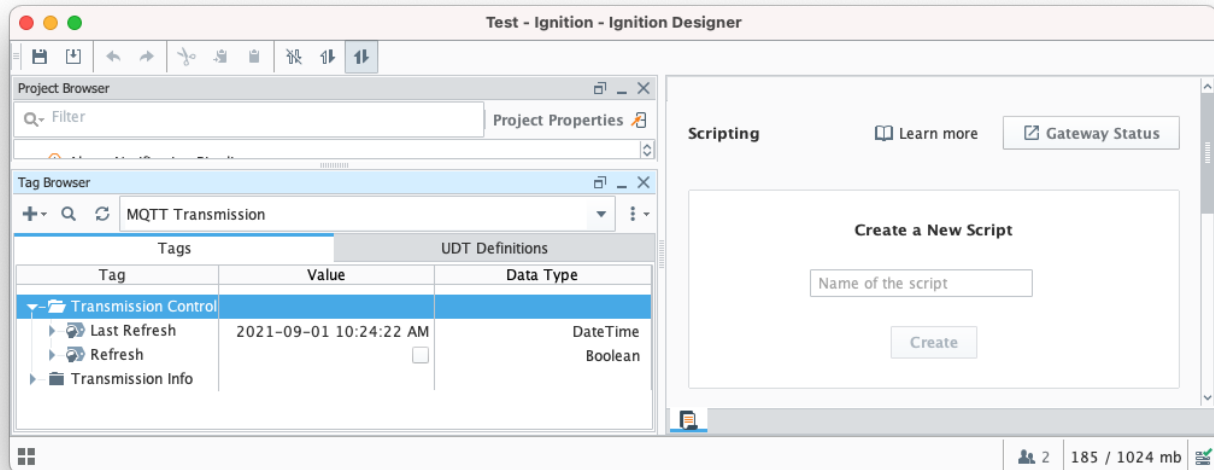


Now browse to the MQTT Engine tag provider and verify the tags for both Edge Nodes have arrived as shown below.
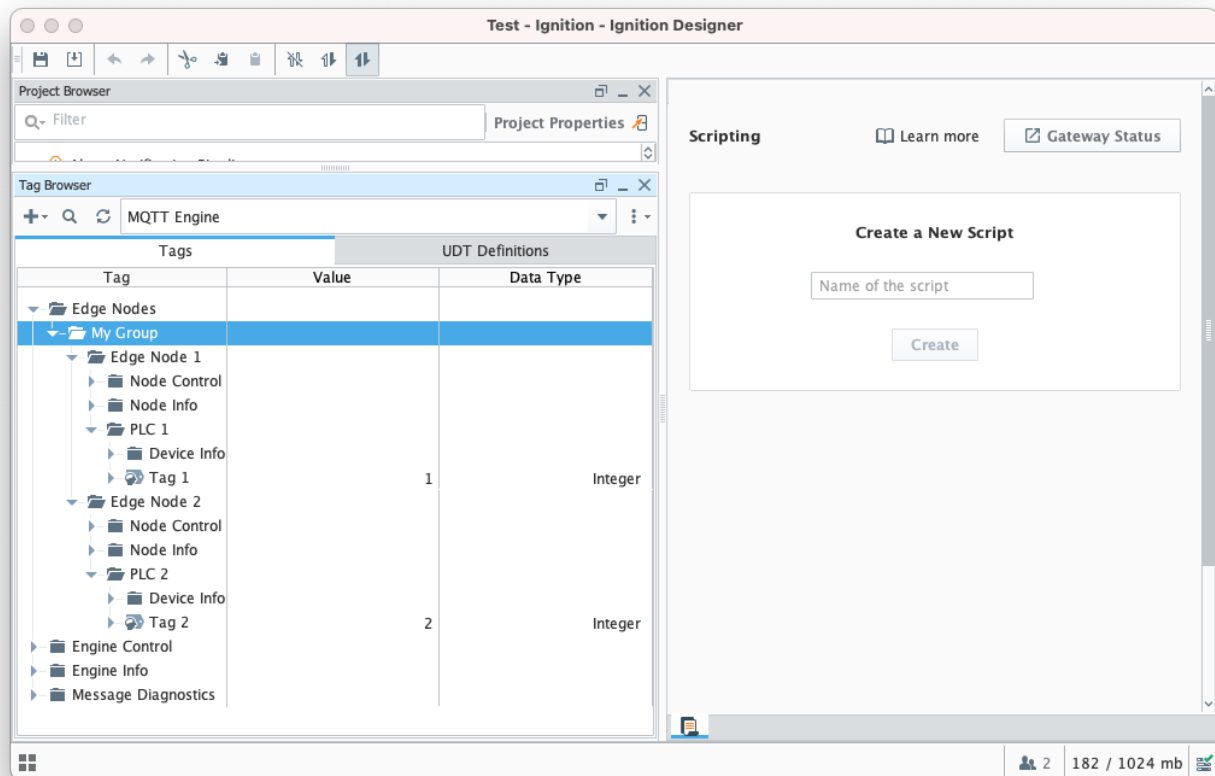
Browse back to the default tag provider and create a new memory tag under 'PLC 1' called 'Tag 3' as shown below:
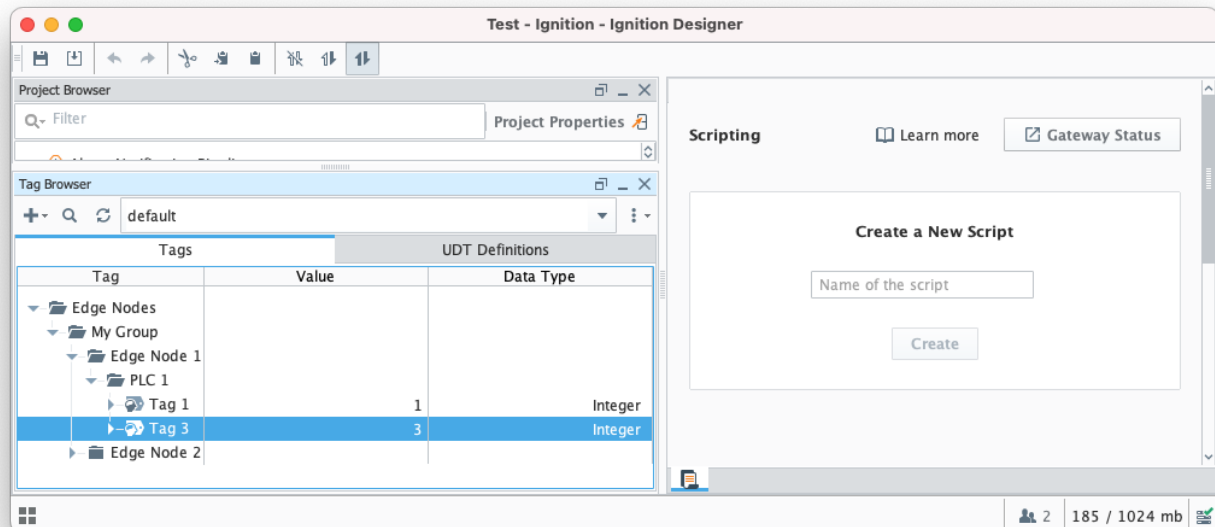


Now browse back to the MQTT Transmission tag provider and you will see the 'Refresh Required' boolean tag now shows 'true'. This is because a new tag was added which means a new BIRTH sequence must be published for consumers of the messages (such as MQTT Engine) so they can pick up the tags.
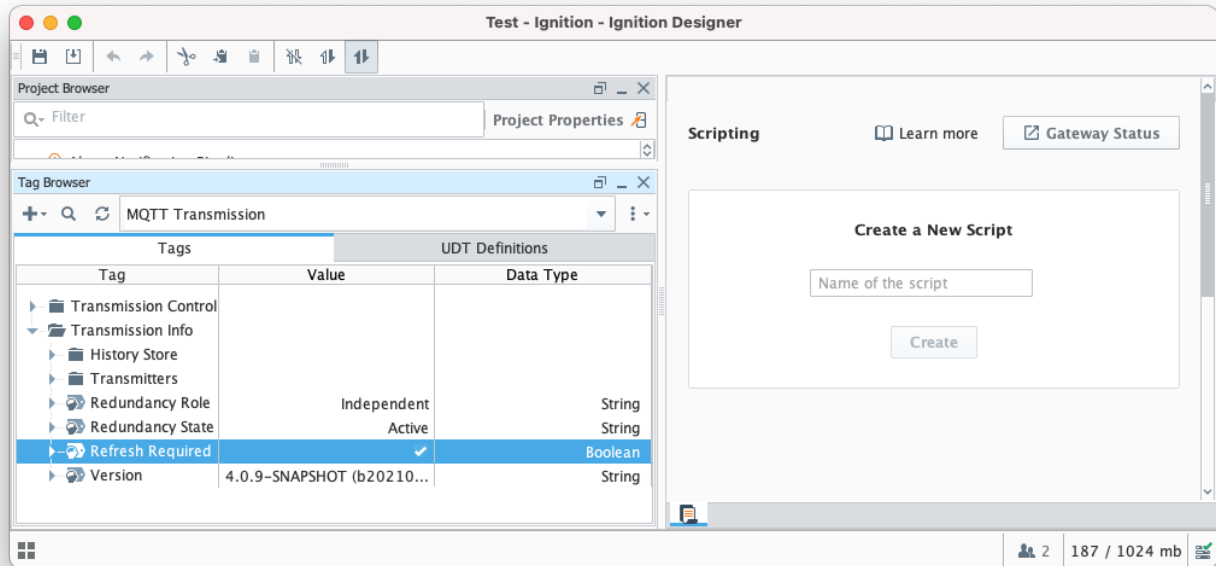
At this point writing a true to the tag '[MQTT Transmission]Transmission Control/Refresh' must be done before the new tag will be picked up by MQTT Engine.
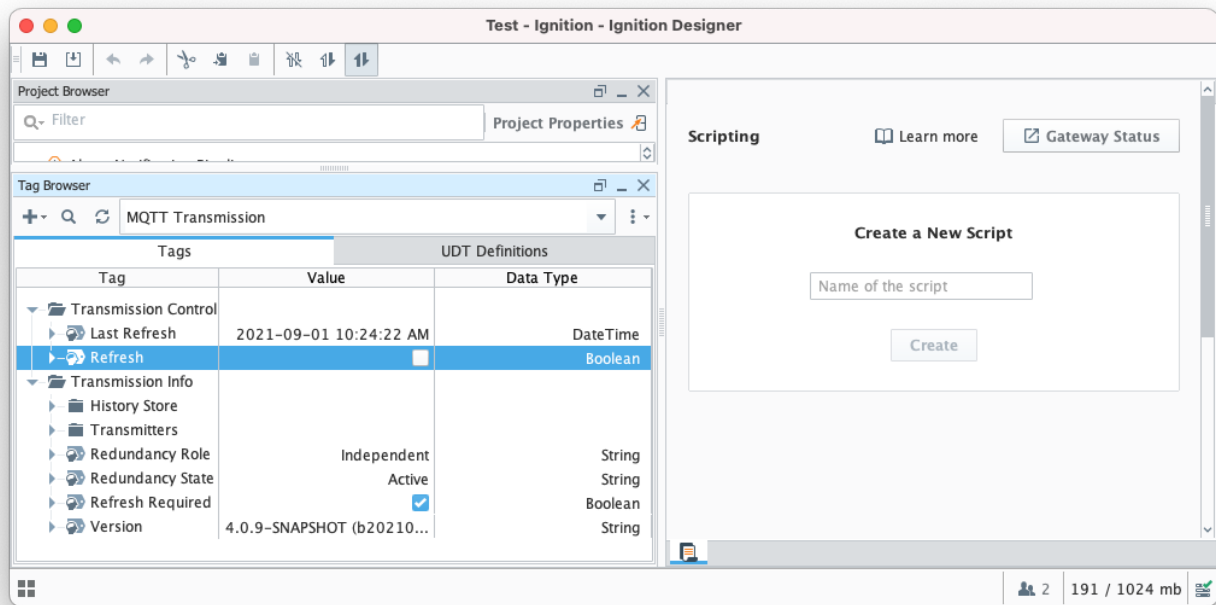


After doing so, the new tag appears in MQTT Engine as shown below.
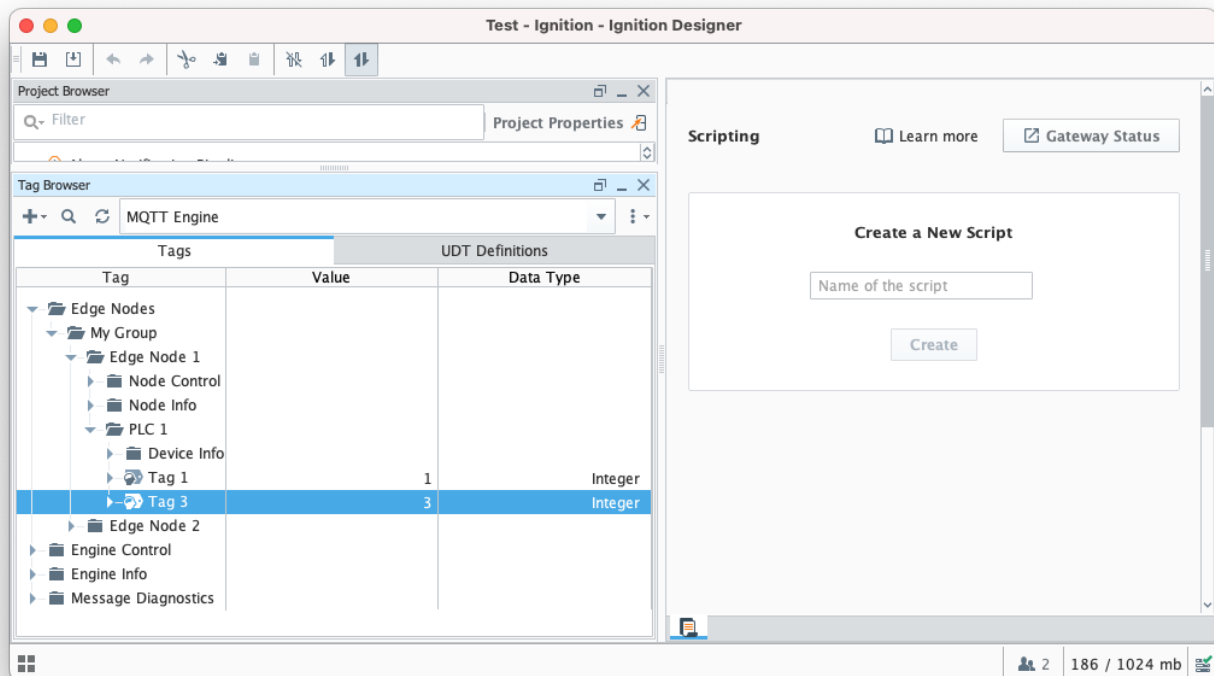
ⓘ In performing this type of 'Global Refresh' using the '[MQTT Transmission]Transmission Control/Refresh' means that all Transmitter tag trees in Transmission will be rescanned. So, using this mechanism will also publish new BIRTH sequences for all Edge Nodes in MQTT Transmission. In this example both 'My Group/Edge Node 1' as well as 'My Group/Edge Node 2' would publish their BIRTH sequences. In addition, if tags had been added to any Edge Nodes the changes would also be picked up for all and published in their respective BIRTH messages.

**Refreshing the Tag Tree for a Single Edge Node:**

ⓘ This is only available in MQTT Transmission 4.0.9 and later.

Building off of the example from the previous section add another tag to 'Edge Node 1' called 'Tag 4' as shown below:
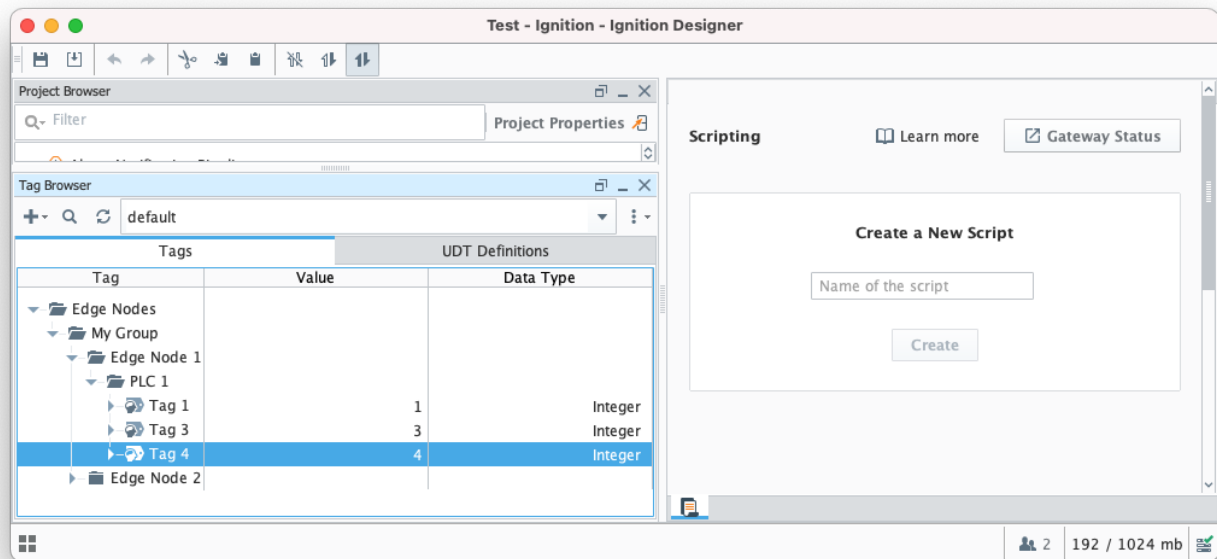
Now expand the Transmission Info folder under the MQTT Transmission tag provider as shown below.

Now that the Global Refresh Required boolean is true as is the 'Edge Node 1' Refresh Required is also true. This is because we added a tag under the 'Edge Node 1' Edge Node. As a result, 'Edge Node 2' doesn't actually need to be refreshed. So, at this point we can write a true to the '[MQTT Transmission]Transmission Info/Transmitters/Default Transmitter/Edge Nodes/My Group/Edge Node 1/Refresh Edge Node' tag. In doing so, only the tag tree associated with 'Edge Node 1' is rescanned. Also as a result, only the BIRTH sequence associated with 'Edge Node 1' will be published. After performing this refresh, the Global Refresh Required tag as well as the 'Edge Node 1' specific Refresh Required tag should go back to false denoting that everything is up to date. This is shown below.

Finally, MQTT Engine should also now show 'Tag 4' in the tag tree.
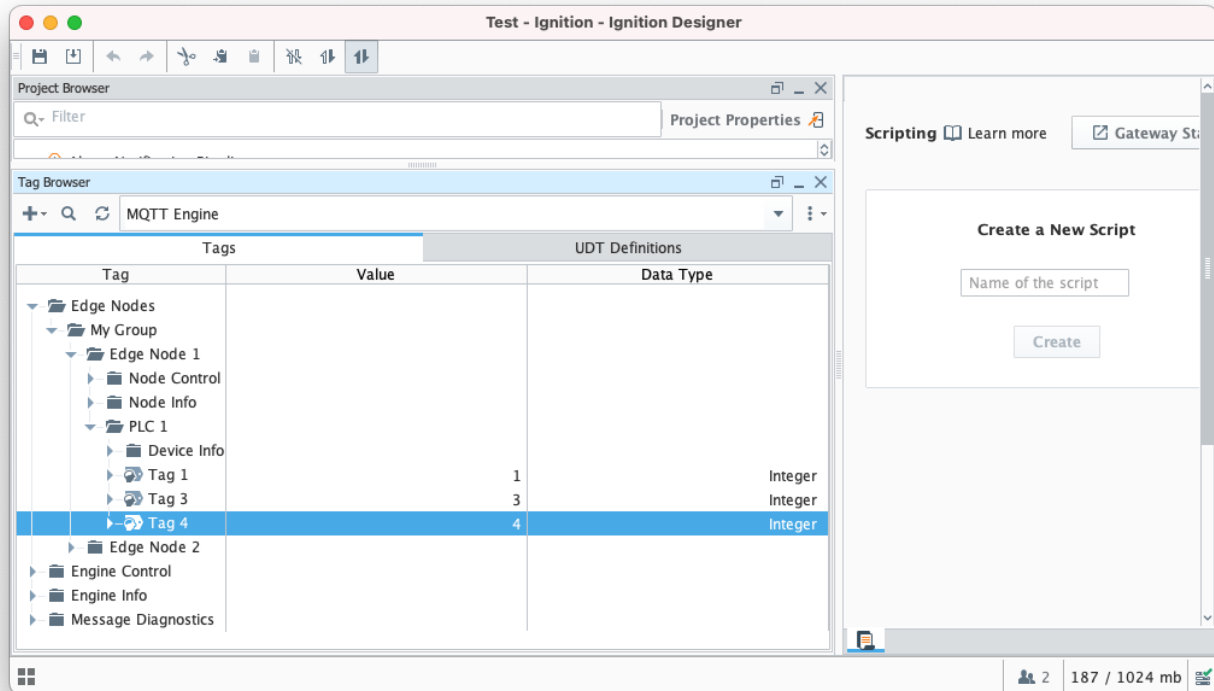
In performing this type of 'Edge Node specific Refresh' using the '[MQTT Transmission]Transmission Info/Transmitters/TRANSMITTER_NAME /Edge Nodes/GROUP_ID/EDGE_NODE_ID/Refresh Edge Node' means that only the sub-tag tree associated with this specific Edge Node will be rescanned. So, using this mechanism will only publish a new BIRTH sequence for this one Edge Node. In this example only 'My Group/Edge Node 1' would publish it's BIRTH sequence. In addition, if tags had been added to any other Edge Node(s) the changes would not be picked up because the refresh scope is to this one single Edge Node.

## 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/