

Sending OPC Tag Data with Transmission

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
- A device that supports Modbus over TCP

Overview:

Transmission is an MQTT module for Ignition that can convert Ignition tag data and tag change events into MQTT messages to be consumed by MQTT Engine or other MQTT clients. This tutorial will show how to configure MQTT Transmission to send OPC tag data in Ignition as MQTT messages via MQTT Distributor to MQTT Engine where they will be displayed.

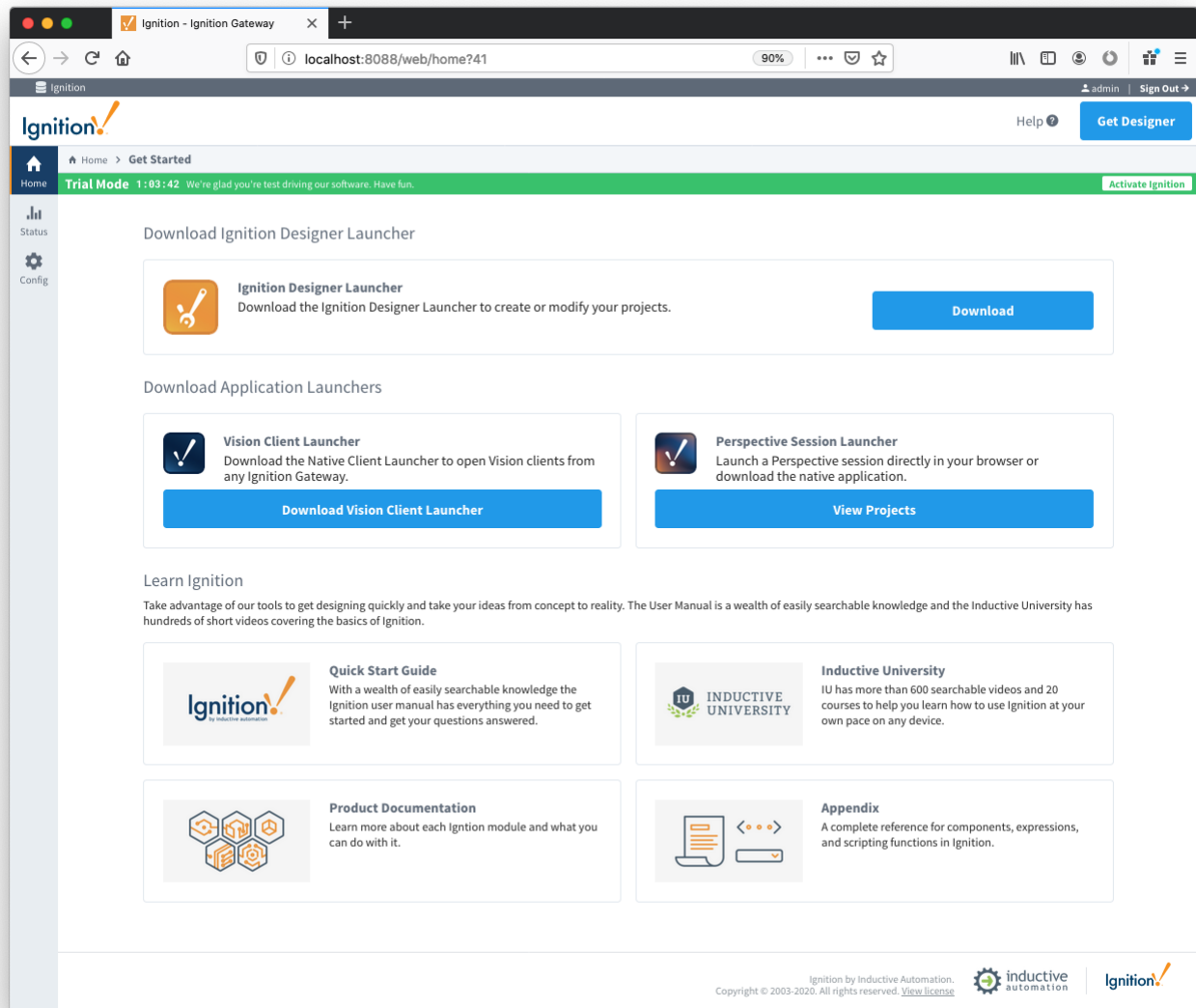
The topology of this example shows MQTT Distributor, MQTT Engine, and MQTT Transmission all running in the same Ignition instance. This is done for simplicity of the tutorial. But, this isn't required or even intended to be a real use case. In a more realistic scenario MQTT Transmission and MQTT Engine would be located on separate machines.

Variations:

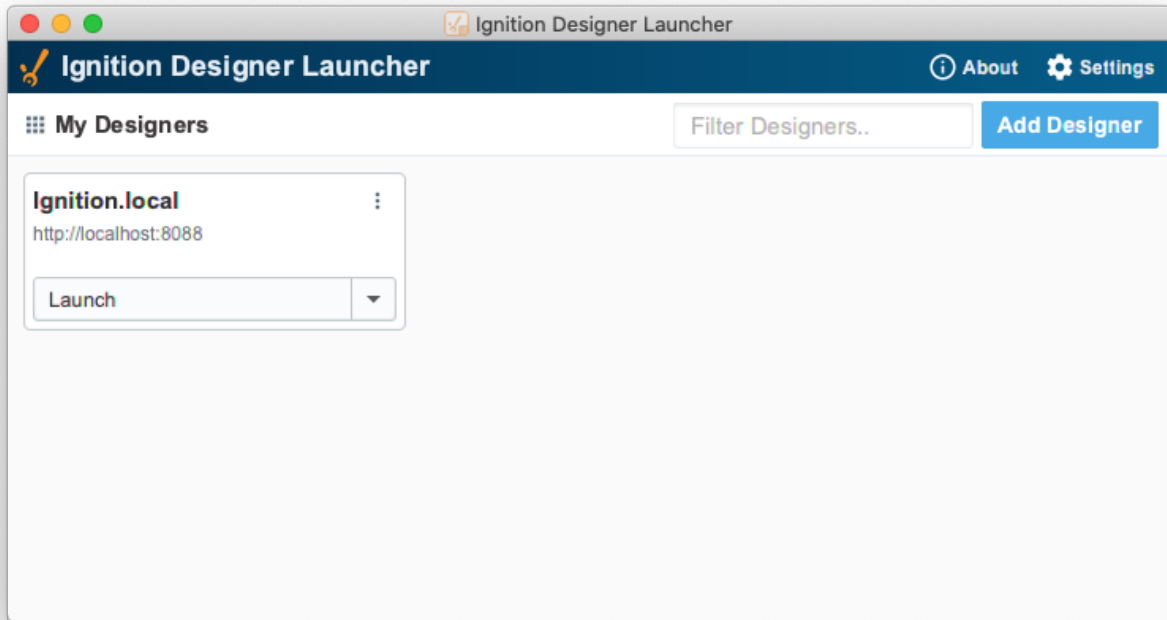
This tutorial shows how to use OPC tags and MQTT Transmission to generate MQTT messages based on tag change events. However, the tag source does not have to be an OPC tag. Instead, as long as the tag structure for MQTT Transmission is followed, any Ignition tag can be used to generate MQTT messages and/or be controlled via MQTT messages.

Sending OPC Tag Data with Transmission:

The first step is to configure the tag provider in Ignition in a way that MQTT Transmission understands. Start by configuring your OPC server, client, and tags. This can be done using the Inductive Automation documentation [here](#). Once this is done, the Tag Provider needs to be set up in Ignition via the Ignition Designer. Using a Web Browser, browse to the Ignition Gateway on your Ignition Gateway. If it is running on your development machine, that is: <http://localhost:8088>. You should see this:



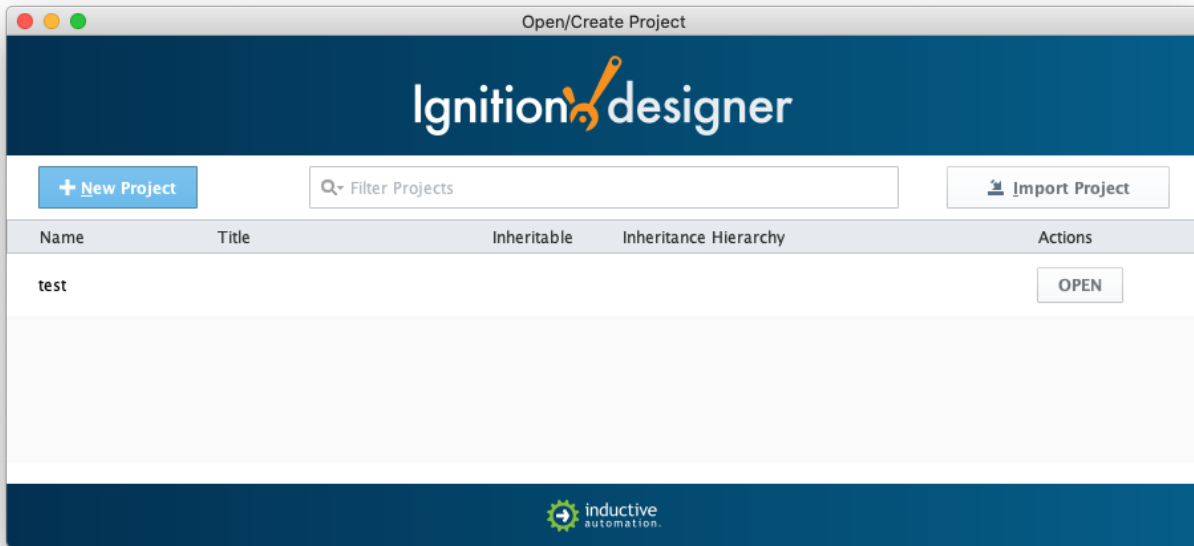
Near the upper right corner, click 'Get Designer'. This will take you to a screen to select the version of the Designer to download based on your operating system. Follow the instructions to download and install and then start the designer, showing a login screen like this:



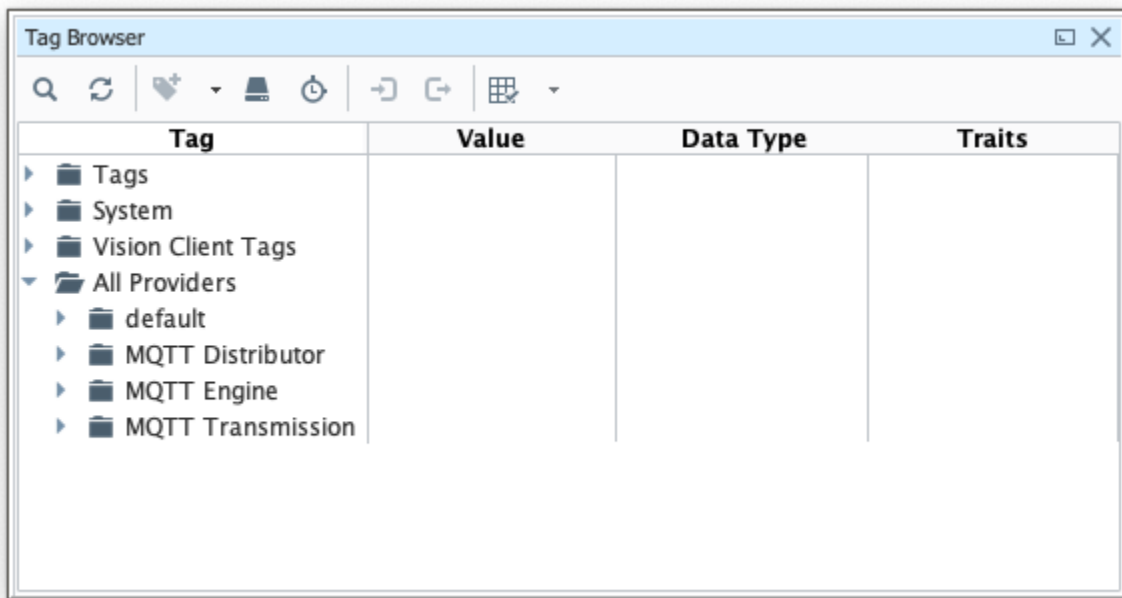
Press "Launch" to start the designer:



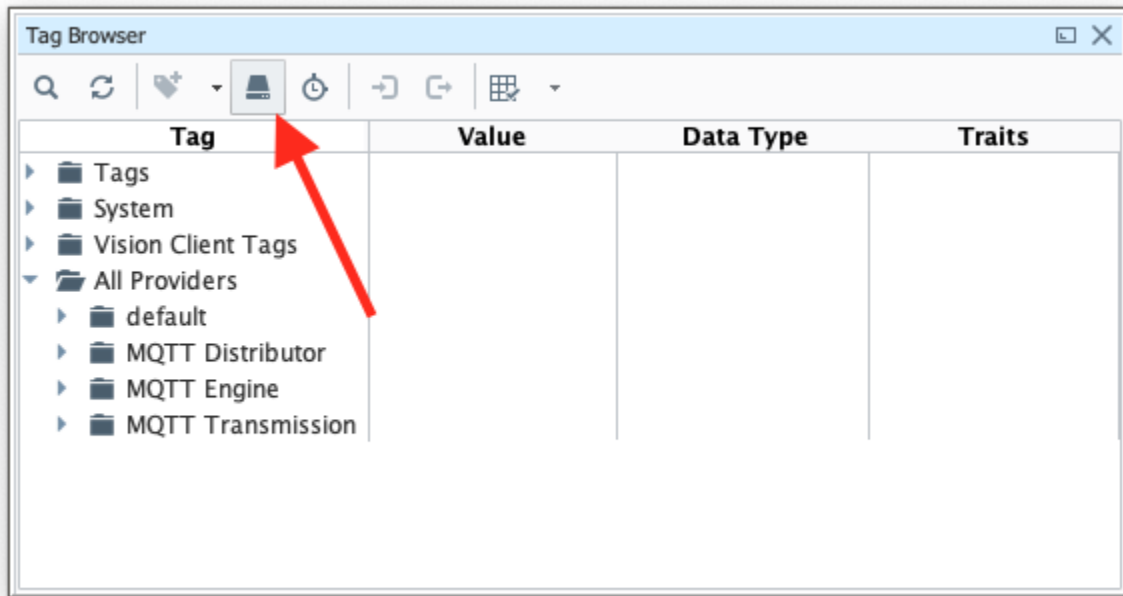
This will bring you to a new Window where you can select an Ignition Project or create a new one. Create a new project by giving it a name and clicking 'New Project'.



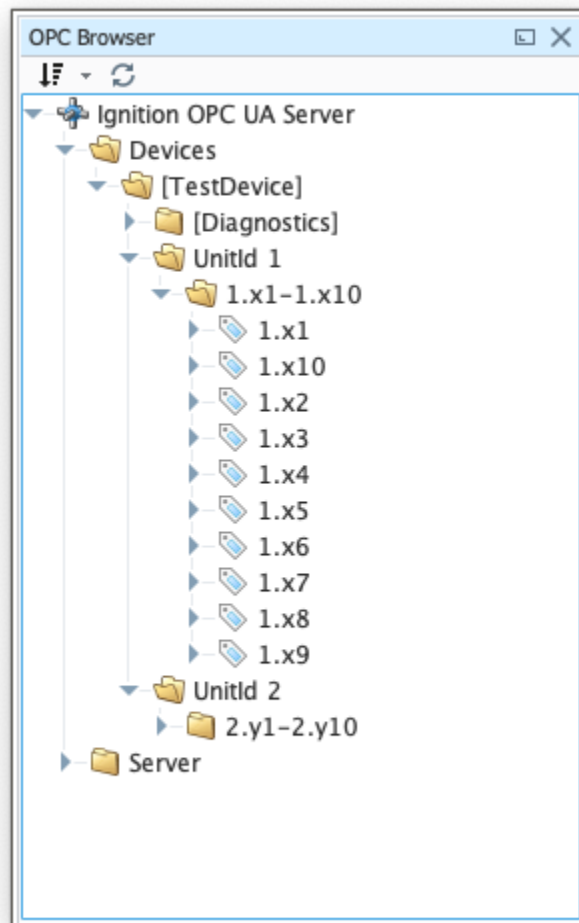
This tutorial will use the "default" Tag Provider. Expand 'All Providers' in the Tag Browser and select 'default'. Note that if Ignition Edge is being used, the tag provider will be named 'edge'.



With 'default' (or 'edge') selected, click the 'OPC' icon in the Tag Browser icon list:



This will open a new window as shown below. If the OPC server and client were set up and configured properly, you should see something similar to the following:



Note there is a device with an attached PLC and two sets of registers. Yours will look different based on the device you are using and how it is configured. At this point, we can do the following:

1. Select the device (TestDevice in this case) and drag it into the Tag Browser under the 'default' Tag Provider.
2. Rename the "_TestDevice_" folder to "Tutorial Tags".
3. Rename the "UnitId 1" and "UnitId 2" folders to "PLC1" and "PLC2".
4. Delete the "_Diagnostics_" folder as it is not used in this tutorial.

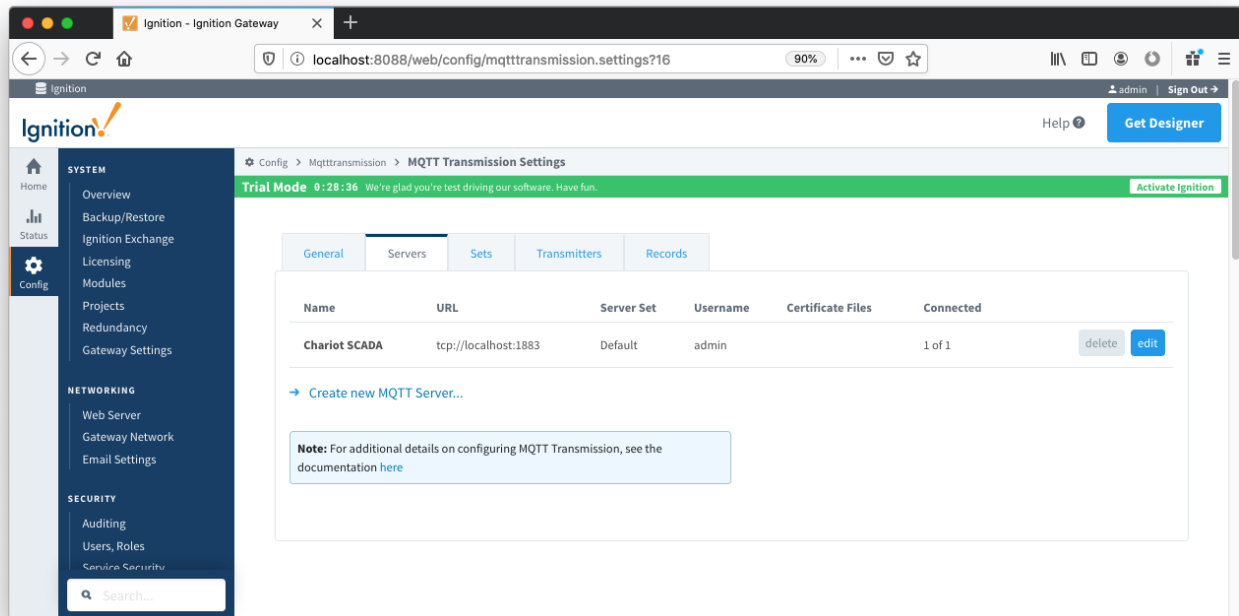
This is shown below:

Tag	Value	Data Type	Traits
Tags			
System			
Vision Client Tags			
All Providers			
default			
Data Types			
MQTT Tags			
Tutorial Tags			
PLC1			
1_x1-1_x10			
1_x1 OPC	<input type="checkbox"/>	Boolean	
1_x2 OPC	<input type="checkbox"/>	Boolean	
1_x3 OPC	<input type="checkbox"/>	Boolean	
1_x4 OPC	<input type="checkbox"/>	Boolean	
1_x5 OPC	<input type="checkbox"/>	Boolean	
1_x6 OPC	<input type="checkbox"/>	Boolean	
1_x7 OPC	<input type="checkbox"/>	Boolean	
1_x8 OPC	<input type="checkbox"/>	Boolean	
1_x9 OPC	<input type="checkbox"/>	Boolean	
1_x10 OPC	<input type="checkbox"/>	Boolean	
PLC2			
2_y1-2_y10			
2_y1 OPC	0	Short	
2_y2 OPC	0	Short	
2_y3 OPC	0	Short	
2_y4 OPC	0	Short	
2_y5 OPC	0	Short	
2_y6 OPC	0	Short	
2_y7 OPC	0	Short	
2_y8 OPC	0	Short	
2_y9 OPC	0	Short	
2_y10 OPC	0	Short	
MQTT Distributor			
MQTT Engine			
MQTT Transmission			

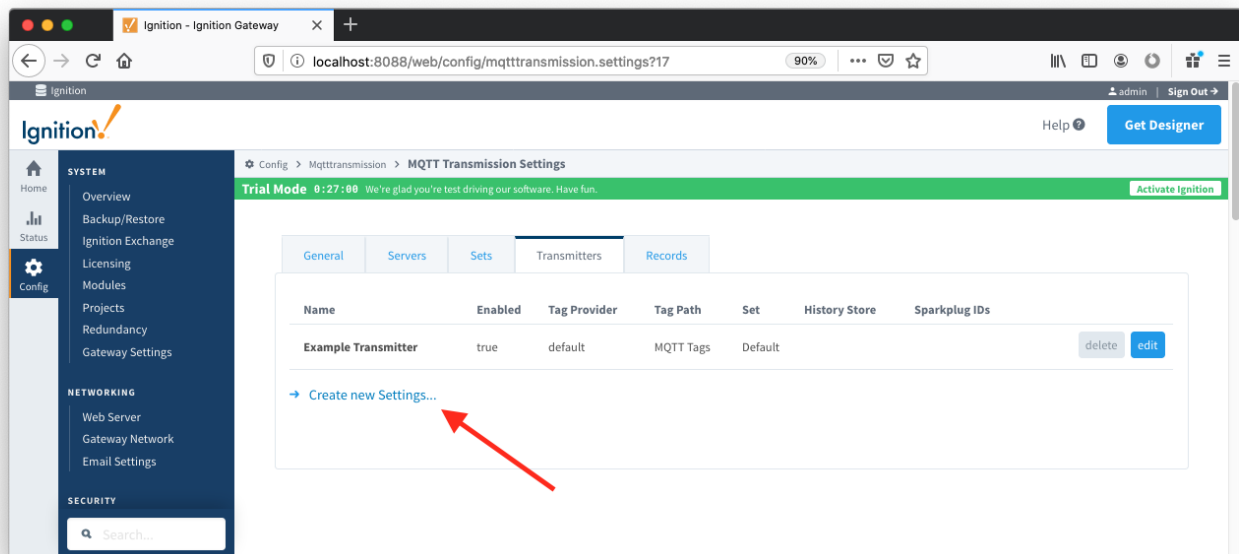
The folder structure of these Tags should be carefully noted so that MQTT Transmission can be configured correctly to monitor the Tags.

With the Tags set up, MQTT Transmission can now be configured. Do so by browsing to the the Configure section of the Ignition Gateway web UI and selecting 'MQTT Transmission -> Settings' on the left:

This tutorial uses the default MQTT Server of MQTT Distributor:



Under the Transmitters tab, a new Transmitter will need to be created. Click on the "Create new Setting..." link as shown below:



Modify the following settings:

Tag Settings

- **Name:** Tutorial Transmitter
- **Tag Provider:** default
- **Tag Path:** Tutorial Tags

Sparkplug Settings

- **Group ID:** Tutorial Group

- **Edge node ID:** Tutorial Edge

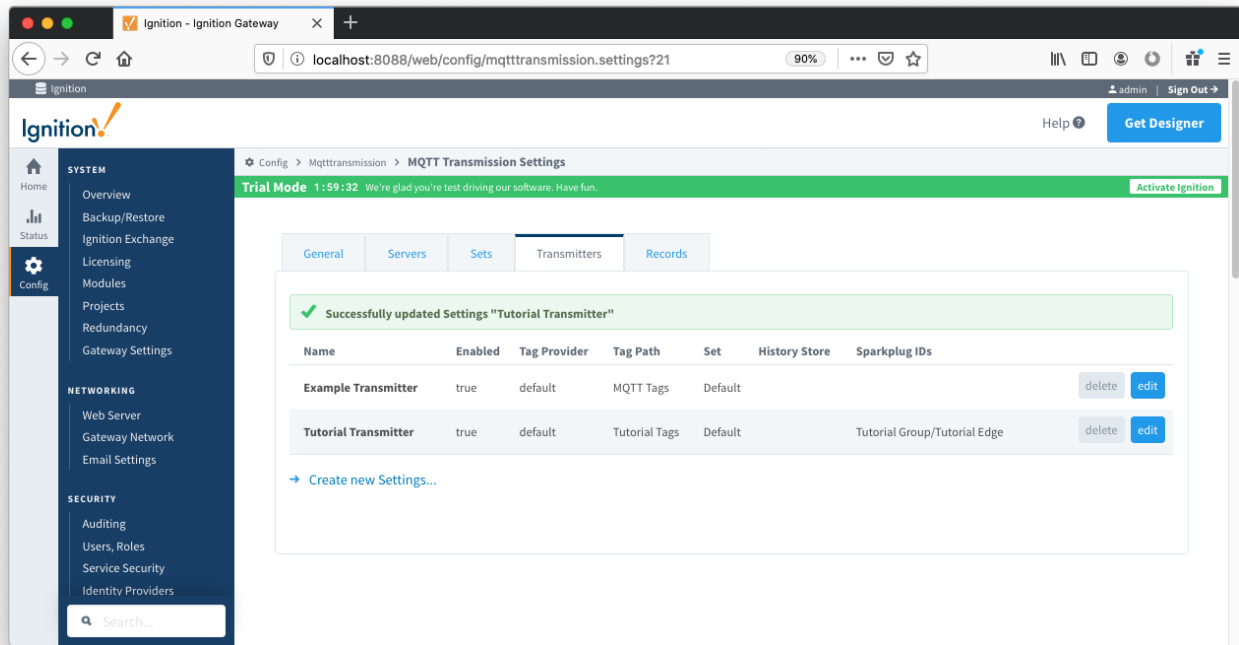
(leave all other setting as defaults)

The screenshot shows the Ignition Gateway web interface in a browser window. The address bar shows the URL `localhost:8088/web/config/mqtttransmission.settings?20`. The page title is "MQTT Transmission Settings". A green banner at the top indicates "Trial Mode 0:12:24". The left sidebar contains a navigation menu with categories: SYSTEM, NETWORKING, SECURITY, DATABASES, ALARMING, TAGS, and OPC CLIENT. The main content area has tabs for General, Servers, Sets, Transmitters, and Records. The "Transmitters" tab is active, showing the "Tag Settings" section. The settings are as follows:

Tag Settings	
Name	Tutorial Transmitter <small>A unique name for the Transmitter</small>
Enabled	<input checked="" type="checkbox"/> Enable Transmitter
Tag Provider	default <small>The Name of the tag provider</small>
Tag Path	Tutorial Tags <small>A path to the root folder where the tag tree starts (optional)</small>
Tag Pacing Period	1000 <small>The waiting period in milliseconds after an initial tag change event before publishing all changed tags (default: 1,000)</small>
Set	Default <small>The MQTT Server Set to use with this Transmitter</small>
Discovery Delay	0 <small>The Transmitter Discovery Delay in milliseconds. This is useful when using MQTT Engine as the tag provider (default: 0)</small>
Aliased Tags	<input type="checkbox"/> Use aliases for tag names to optimize payload sizes when publishing data
Compression	NONE <small>The algorithm to use for compressing payloads before publishing</small>
Block Commands	<input type="checkbox"/> Block incoming commands (writes) to Edge Node and Device Tags
Convert UDTs	<input checked="" type="checkbox"/> Converts UDT members to normal Tags before publishing
Publish UDT Definitions	<input checked="" type="checkbox"/> Publish UDT Definitions in BIRTH

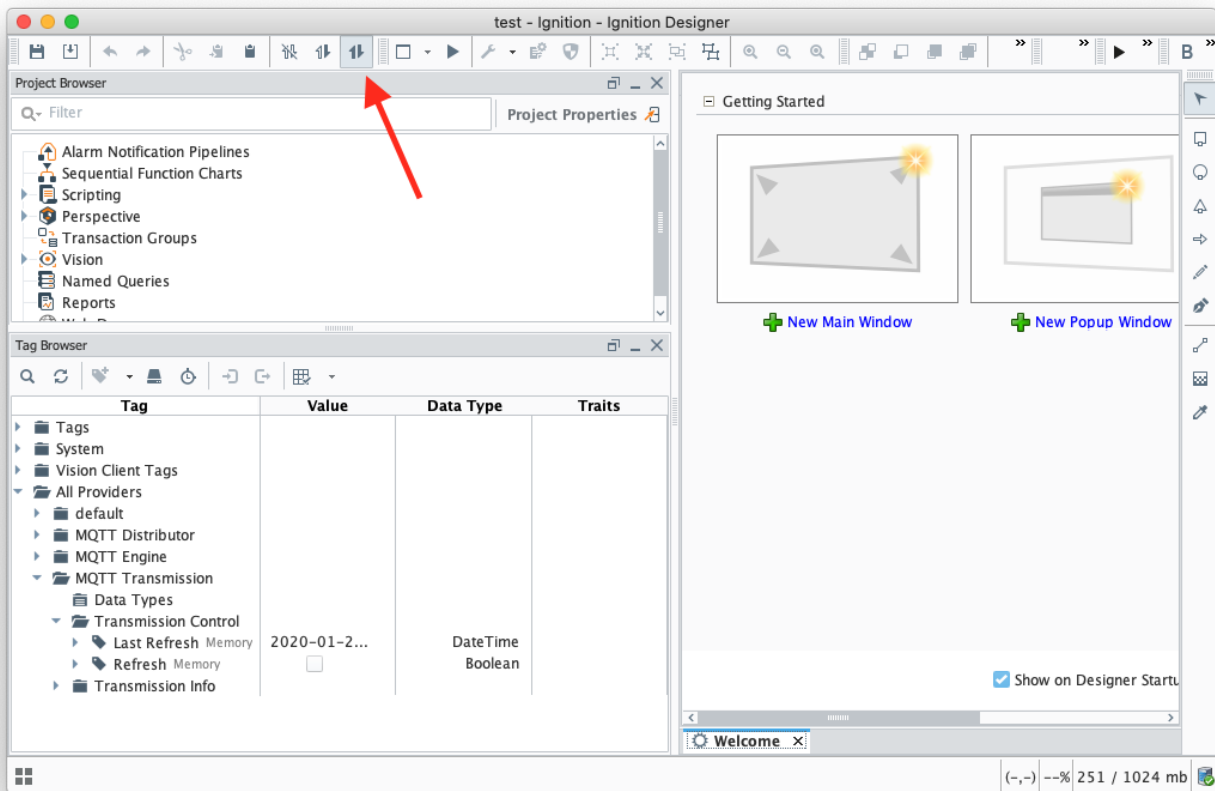
Below the Tag Settings is a section for "History Settings" which is currently empty.

Click "Save Changes" to see the new Transmitter was successfully created.



This new Transmitter will scan the Tags in the "Tutorial Tags" folder and publish them to MQTT Engine using the hard coded Group and Edge Node IDs that were configured in the Transmitters settings. Since the Device ID was left blank in the Transmitter settings, they will be scanned from the Tag Tree. Each Folder in the "Tutorial Tags" folder will be considered a device. For this tutorial the devices will be "PLC1" and "PLC2".

In Designer force MQTT Transmission to update by toggling the "Refresh" Tag shown below:

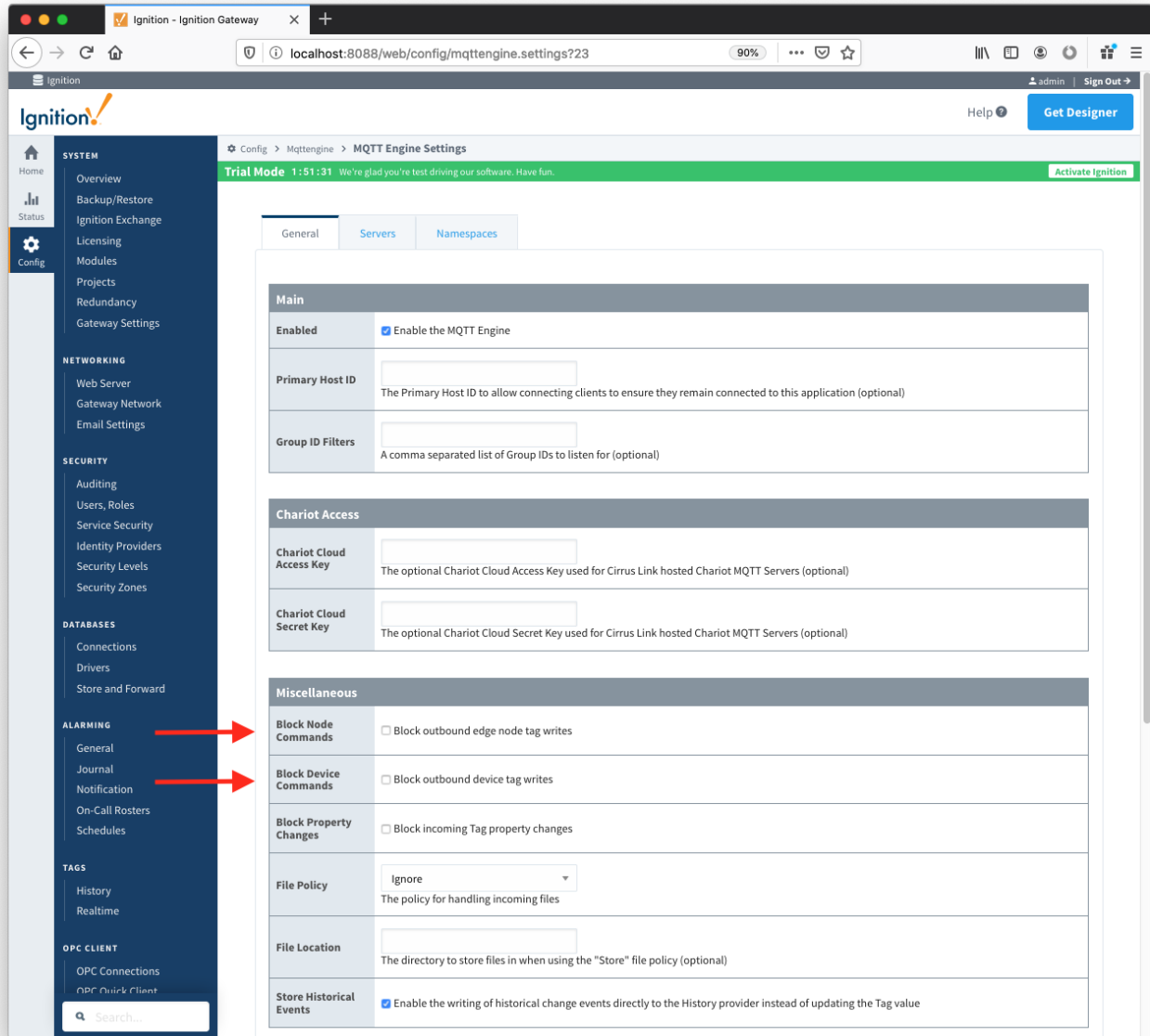


At this point, you should be able to expand the "MQTT Engine" Tag provider and see all of the tags in MQTT Engine:

Tag	Value	Data Type	Traits
Tags			
System			
Vision Client Tags			
All Providers			
default			
MQTT Distributor			
MQTT Engine			
Data Types			
Edge Nodes			
My MQTT Group			
Tutorial Group			
Tutorial Edge			
Node Control			
Node Info			
PLC1			
1_x1-1_x10			
1_x1 Memory	<input type="checkbox"/>	Boolean	
1_x2 Memory	<input type="checkbox"/>	Boolean	
1_x3 Memory	<input type="checkbox"/>	Boolean	
1_x4 Memory	<input type="checkbox"/>	Boolean	
1_x5 Memory	<input type="checkbox"/>	Boolean	
1_x6 Memory	<input type="checkbox"/>	Boolean	
1_x7 Memory	<input type="checkbox"/>	Boolean	
1_x8 Memory	<input type="checkbox"/>	Boolean	
1_x9 Memory	<input type="checkbox"/>	Boolean	
1_x10 Memory	<input type="checkbox"/>	Boolean	
Device Info			
PLC2			
2_y1-2_y10			
2_y1 Memory	0	Short	
2_y2 Memory	0	Short	
2_y3 Memory	0	Short	
2_y4 Memory	0	Short	
2_y5 Memory	0	Short	
2_y6 Memory	0	Short	
2_y7 Memory	0	Short	
2_y8 Memory	0	Short	
2_y9 Memory	0	Short	
2_y10 Memory	0	Short	
Device Info			
Engine Info			
Message Diagnostics			
MQTT Transmission			

In addition to the tags being displayed in Engine, they are also writable if this is enabled in MQTT Engine. By default, MQTT Engine blocks command messages from being sent to devices. To enable this feature, in the Ignition web console browse to the MQTT Engine Module Settings.

Make sure the "Block Node Commands" and "Block Device Commands" settings are disabled, as shown below.




Note that after updating the MQTT Engine settings, the Tags in the Tag Browser under the MQTT Engine Tag Provider may become Stale. This is because the primary host ID is not set up on both MQTT Engine and MQTT Transmission which is outside of the scope of this Tutorial. To get around this, manually toggle the MQTT Transmission "Refresh" Tag as shown above.

Now attempt to write to any output Tags that are available on the OPC device, via the corresponding Tag in the MQTT Engine Tag Provider.

Tag Browser

Search | Refresh | Add | Remove | Copy | Paste | Grid View | Filter

Tag	Value	Data Type	Traits
▸ Tags			
▸ System			
▸ Vision Client Tags			
▾ All Providers			
▸ default			
▸ MQTT Distributor			
▾ MQTT Engine			
Data Types			
▾ Edge Nodes			
▸ My MQTT Group			
▾ Tutorial Group			
▾ Tutorial Edge			
▸ Node Control			
▸ Node Info			
▾ PLC1			
▾ 1_x1-1_x10			
▸ 1_x1 Memory	<input checked="" type="checkbox"/>	Boolean	
▸ 1_x2 Memory	<input type="checkbox"/>	Boolean	
▸ 1_x3 Memory	<input type="checkbox"/>	Boolean	
▸ 1_x4 Memory	<input type="checkbox"/>	Boolean	
▸ 1_x5 Memory	<input type="checkbox"/>	Boolean	
▸ 1_x6 Memory	<input type="checkbox"/>	Boolean	
▸ 1_x7 Memory	<input type="checkbox"/>	Boolean	
▸ 1_x8 Memory	<input type="checkbox"/>	Boolean	
▸ 1_x9 Memory	<input type="checkbox"/>	Boolean	
▸ 1_x10 Memory	<input type="checkbox"/>	Boolean	
▸ Device Info			
▸ PLC2			
▸ Engine Info			
▸ Message Diagnostics			
▸ MQTT Transmission			



Note there is some delay in the response. This is due in part to MQTT Transmissions 'Tag Pacing Period'. This is the delay for MQTT messages to wait before being sent to allow multiple change events to buffer before putting them into a single MQTT message. This can be changed in the MQTT Transmission module's Transmitter configuration in the Ignition web console.

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/>