



Sparkplug B MQTT Quick-Start Guide

Your Industrial Control Solutions Source
MAPLESYSTEMS.COM

Maple Systems



For use as the following:

- Installation and configuration of *Inductive Automation Ignition* as an MQTT Broker
- Installation and configuration of *MQTT.fx* as an MQTT Client
- Configuration of Maple Systems *cMT Devices* to send Sparkplug B payloads to *Ignition* via MQTT

Maple Systems, Inc. | 808 134th St. SW, Suite 120, Everett, WA 98204 | 425.745.3229

Sparkplug B MQTT Quick-Start Guide

Summary

This Quick-Start Guide is designed to help you accomplish the following tasks:

1. Install and set up Inductive Automation Ignition on a Windows PC
2. Connect a Maple Systems cMT Device to Ignition Gateway to start publishing MQTT data in Sparkplug B format
3. Connect MQTT.fx to Ignition Gateway; Subscribe to topics; Verify data being sent by the Maple cMT device

What is Sparkplug B?

Sparkplug B is a specification for MQTT enabled devices and applications to send and receive messages in a stateful way. While MQTT is stateful by nature it doesn't ensure that all data on a receiving MQTT application is current or valid. Sparkplug B provides a mechanism for ensuring that remote device or application data is current and valid.

Sparkplug B includes support for features such as:

- Complex data types using templates
- Datasets
- Rich metrics with the ability to add property metadata for each metric
- Metric alias support to maintain rich metric naming while keeping bandwidth usage to a minimum
- Historical data
- File data

To learn more, please see the full Sparkplug B specification documentation [here](#).

What is Ignition (Gateway)?

Ignition is a SCADA platform developed by Inductive Automation. It includes data historian, visualization and reporting, SQL database integration, OPC UA, and MQTT with Sparkplug B as some of its main features. Maple Systems cMT devices (cMT HMIs, cMT Servers, and cMT Gateways) are interoperable with Ignition via the Sparkplug B MQTT mode available in our EBPro programming software.

In order to pair your Maple Systems cMT Device with the Ignition Platform, you must have the following Cirrus Link MQTT Modules installed in the Ignition Gateway:

- Cirrus Link MQTT Distributor Module
- Cirrus Link MQTT Engine Module
- Cirrus Link MQTT Transmission Module

To learn more about Ignition and download a free trial version of the software and MQTT modules, visit the Inductive Automation home page [here](#).

Software Programs and Versions Used in this Quick-Start Guide

Maple Systems HMI Programming Software		
<i>Program</i>	<i>Version</i>	<i>Download Link</i>
EBPro	6.03.02.393	https://www.maplesystems.com/SupportCenter/SoftwareDownloads.htm
Inductive Automation Ignition Gateway SCADA Software		
<i>Program</i>	<i>Version</i>	<i>Download Link</i>
Ignition	8.0.6	https://inductiveautomation.com/downloads/archive/8.0.6
Cirrus Link Solutions MQTT Modules for Ignition		
<i>Module</i>	<i>Version</i>	<i>Download Link</i>
MQTT Distributor	4.0.2	https://inductiveautomation.com/downloads/third-party-modules/8.0.5
MQTT Engine	4.0.2	
MQTT Transmission	4.0.2	
MQTT.fx – MQTT Client Software		
<i>Program</i>	<i>Version</i>	<i>Download Link</i>
MQTT.fx	1.7.1	https://mqttfx.jensd.de

PC System Requirements for Ignition v8.0.6

Supported Operating Systems: *




- Windows Server 2008/2012/2016/2019
- Windows 7, 8, and 10

Requirements:

- Dual-core processor (or greater)
- Minimum: 4 GB RAM
- Minimum: 10 GB free HD space

* *Ignition is supported on additional Operating Systems not listed here. This guide focuses on the Windows OS.*

Description of Required Cirrus Link MQTT Modules for Ignition

	<p>MQTT Distributor Module</p> <p>Acts as an <u>MQTT v3.1.1 compliant MQTT Server</u>. Enables MQTT clients to securely connect, publish, and subscribe to data. Designed to serve up to 50 connecting clients at a time. Clients may include Maple cMT HMIs/Servers/Gateways, Ignition Edge/Edge Onboard nodes, or other third-party clients supporting Sparkplug B such as MQTT.fx.</p>
	<p>MQTT Engine Module</p> <p>Subscribes to any number of MQTT Distributors, whether these are hosted alongside Ignition Gateway, in the field, or the cloud. The MQTT Engine dynamically discovers and creates tags, UDTs/structured tags, and associated metadata via Sparkplug B payloads. MQTT Engine acts as an <u>MQTT to Ignition Tag Bridge</u>. Additionally, it listens for tag writes in Ignition and converts these to MQTT messages before sending them or updating data and I/O on remote MQTT devices.</p>
	<p>MQTT Transmission Module</p> <p>Acts as an <u>Ignition Tag to MQTT Bridge</u>. This module listens for tag change events in Ignition and converts these to outgoing Sparkplug B MQTT messages. Additionally, the MQTT Transmission module enables listeners to be attached to Ignition tags which then wait for tag values to change. When they do, MQTT Sparkplug B messages are generated to publish the data to MQTT Engine. MQTT Transmission also listens for commands sent in Sparkplug B format which allows Ignition tag values to be written remotely.</p>

Outline of Quick-Start Guide

Section	Section Heading	Page
1	Install and Activate <i>Ignition Gateway</i>	5
2	Install <i>Cirrus Link</i> MQTT Modules	13
3	Configure <i>Ignition Gateway</i> and <i>Cirrus Link</i> MQTT Modules	17
4	Install <i>Ignition Designer</i> and create an <i>Ignition</i> project	25
5	Configure <i>EBPro</i> Project for Communication with <i>Ignition Gateway</i>	30
6	Configure <i>MQTT.fx</i> as an MQTT Client	33
7	Test local connection using <i>EBPro</i> Simulation, <i>Ignition Gateway</i> , and <i>MQTT.fx</i>	37
8	Configure Firewall for Incoming/Outgoing MQTT Connections (Ports: 1883, 8883)	48
9	Perform Live Testing on a Maple Systems <i>cMT Server</i>	50
10	Generate SSL Certificates and Establish Secure Communications	52
11	Appendix and Additional Resources	52

1. Install and Activate Ignition Gateway Version 8.0.6

Reference: This section adapted from Inductive Automation’s [Ignition 8 User Manual](#).

Download the following software programs and modules if you have not already done so:

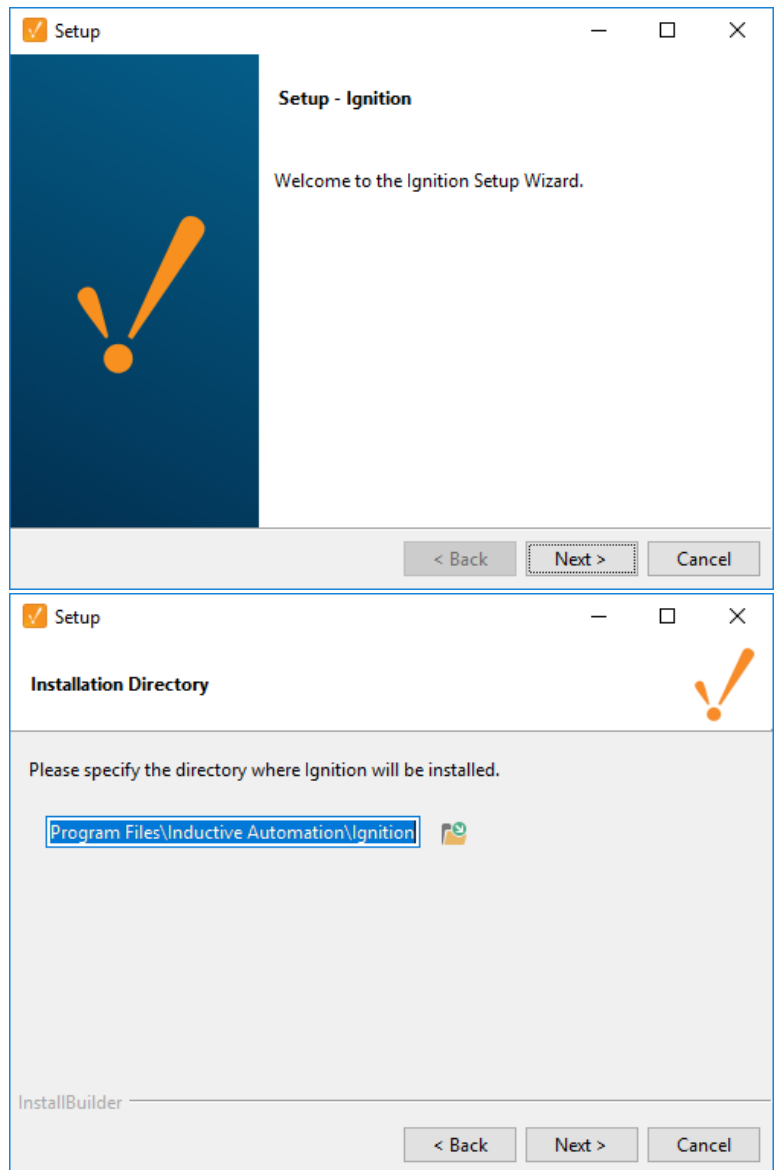
Inductive Automation Ignition Gateway SCADA Software		
Program	Version	Download Link
Ignition	8.0.6	https://inductiveautomation.com/downloads/archive/8.0.6 Select: Ignition - Windows Installer 64-bit File type: .exe; Size: 835MB; Version: 8.0.6.20191112-1641
Cirrus Link Solutions MQTT Modules for Ignition		
Module	Version	Download Link
MQTT Distributor	4.0.2	https://inductiveautomation.com/downloads/third-party-modules/8.0.5
MQTT Engine	4.0.2	MQTT Distributor Module; Size: 30.5 MB
MQTT Transmission	4.0.2	MQTT Engine Module; Size: 25.7 MB MQTT Transmission Module; Size: 21.8 MB

Installing Ignition

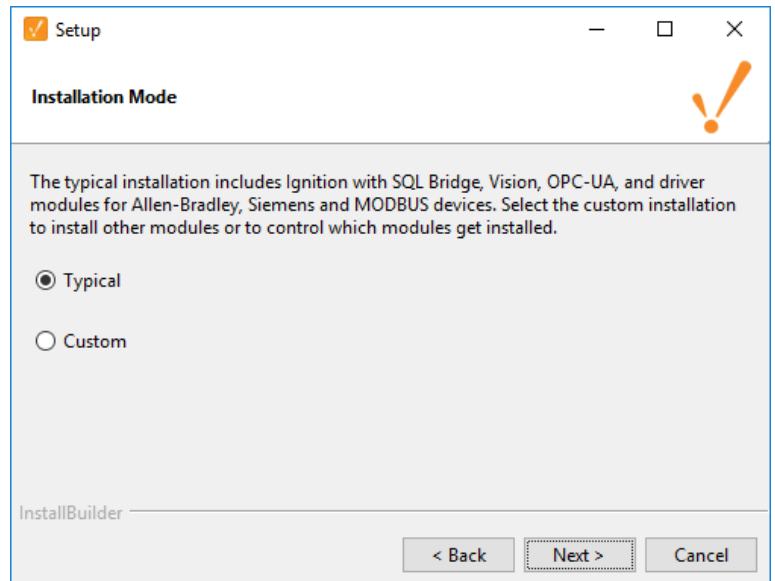
Installation and Setup Process

Double-click on the Ignition installer (“Ignition-8.0.6-windows-x64-installer.exe”) and click ‘Next’ to begin the guided installation process.

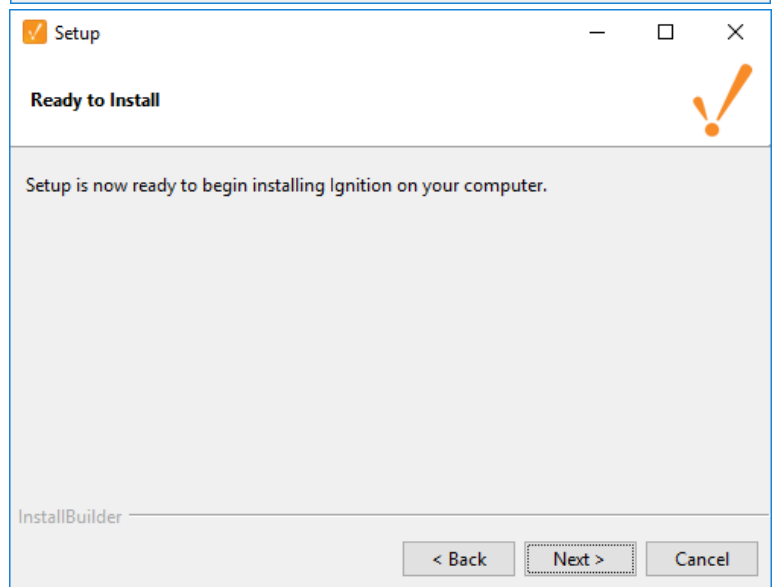
Choose a location for installation. (You may leave it set to the default path if you wish.)



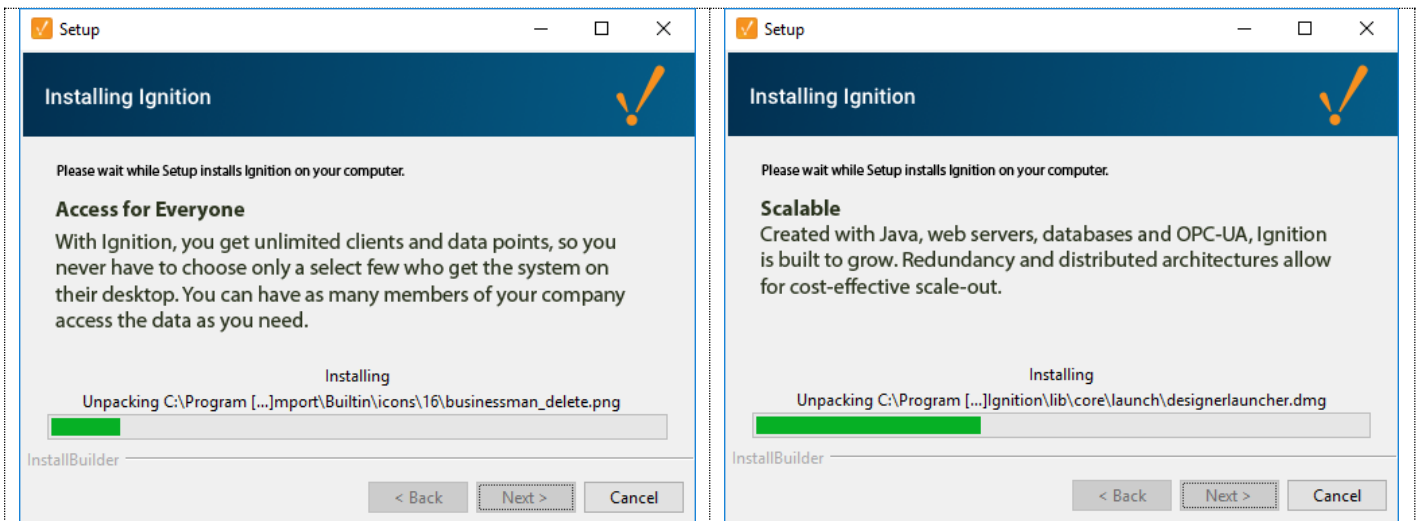
Set the installation mode (Default: Typical).

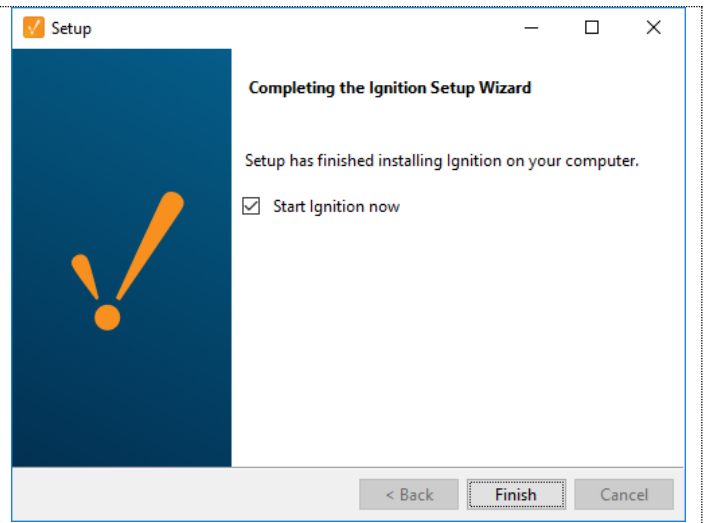
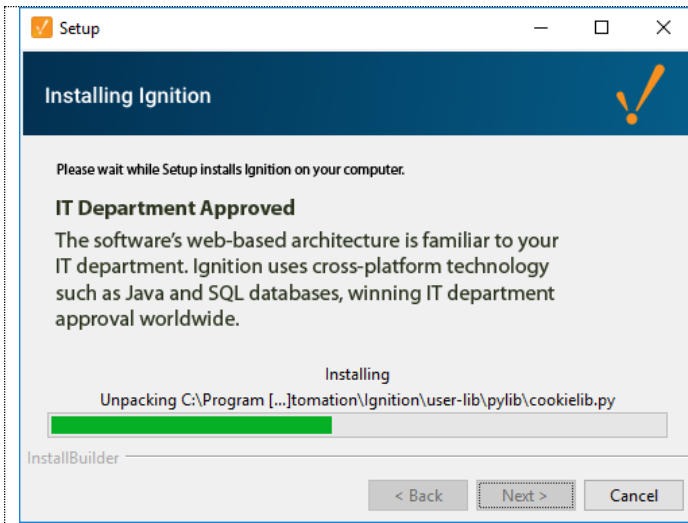


Click 'Next' to begin installation process.



Allow installation to proceed. Example screenshots from during installation:





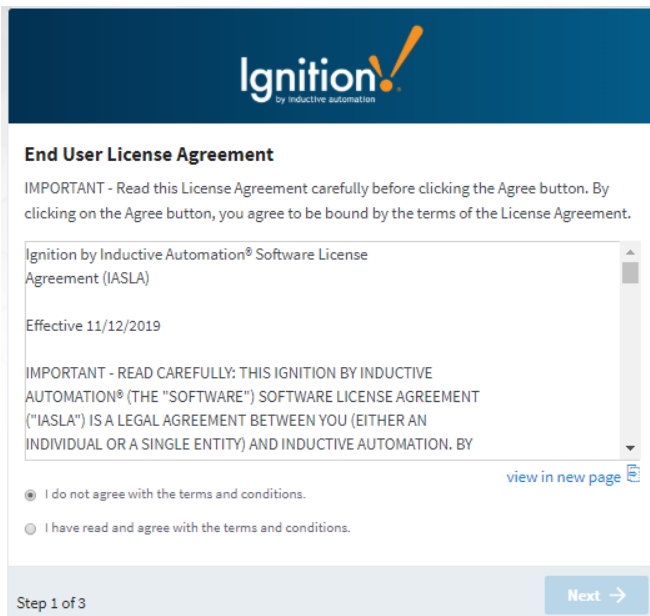
Upon completion, a Welcome Screen will be shown to you in your Browser. (We recommend using Google Chrome or Firefox with Ignition.)



Welcome to Ignition!
Version 8.0.6

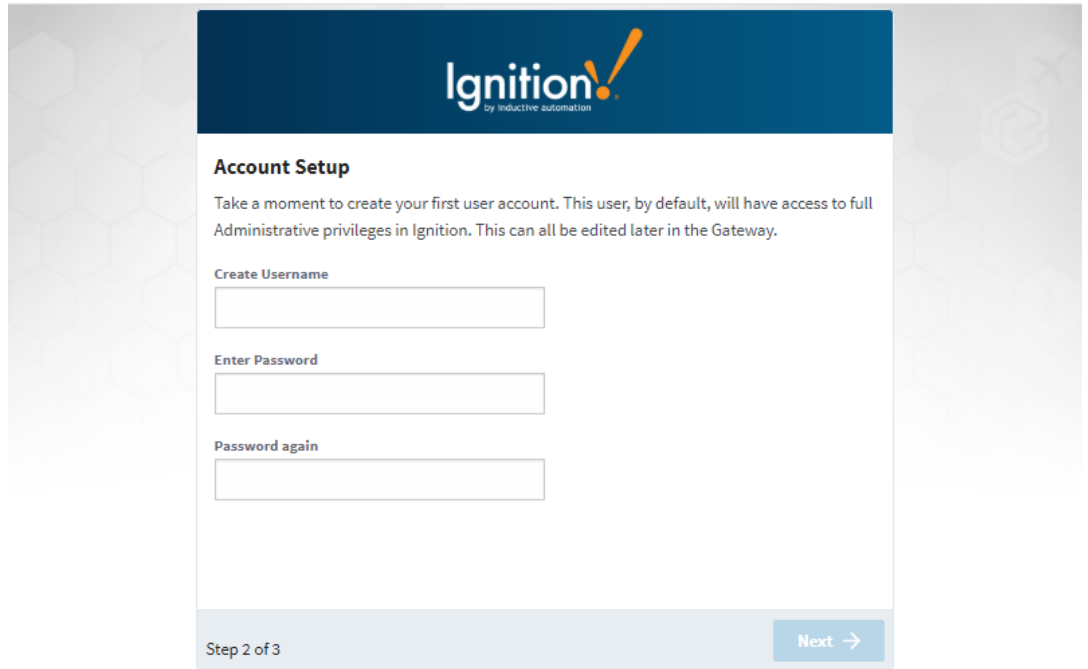
Thank you for installing Ignition! We need some quick information and agreements from you to get started. This will be quick.

Click 'Get Started' to continue.



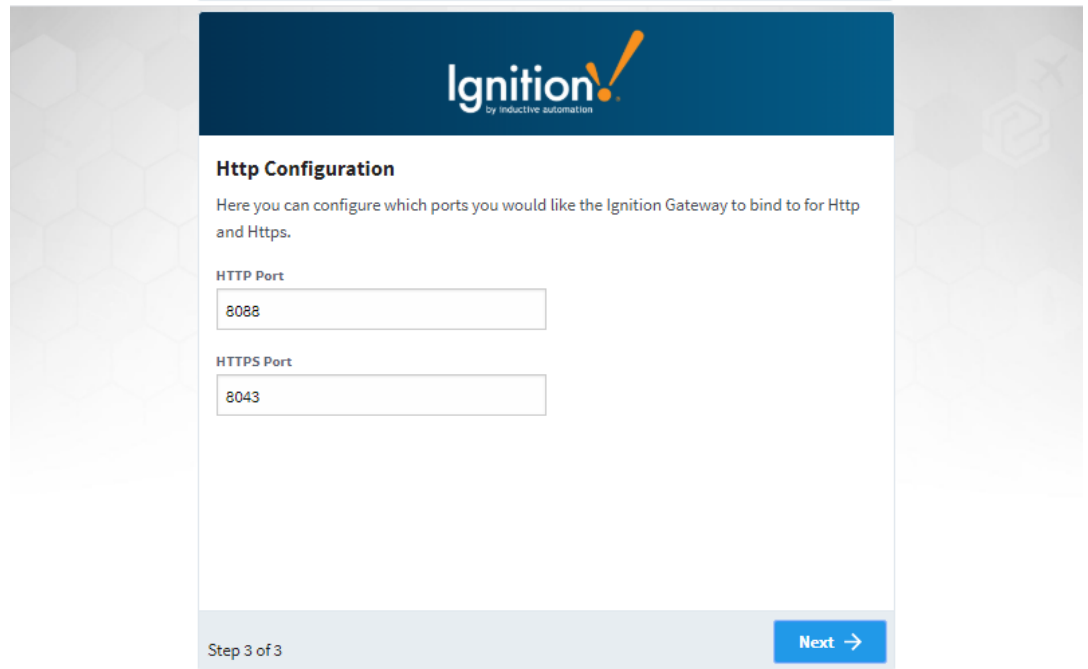
Accept the terms of the End-User License Agreement to continue.

Create an Account. Choose a Username and Password for the Ignition Gateway. (Record this information for later use.)



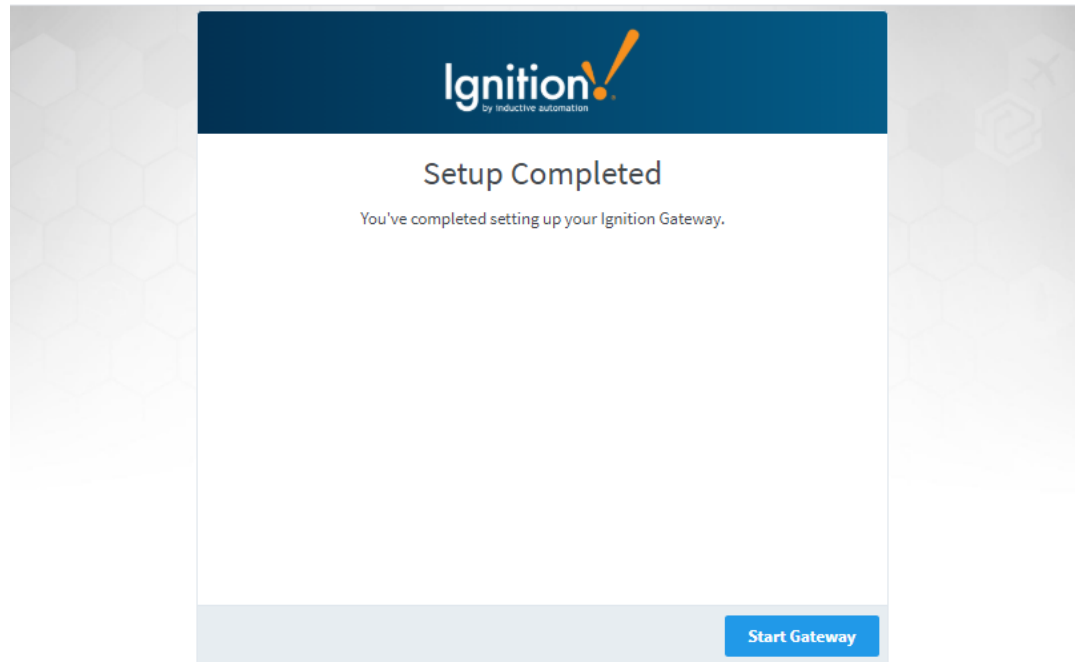
The screenshot shows the 'Account Setup' screen of the Ignition Gateway. At the top, there is a dark blue header with the Ignition logo and the text 'by inductive automation'. Below the header, the title 'Account Setup' is displayed. A paragraph of text explains that the user is creating their first account with full administrative privileges. There are three input fields: 'Create Username', 'Enter Password', and 'Password again'. At the bottom of the screen, it says 'Step 2 of 3' and has a 'Next →' button.

Leave the Gateway Network Ports defaults as they are. Click Next.

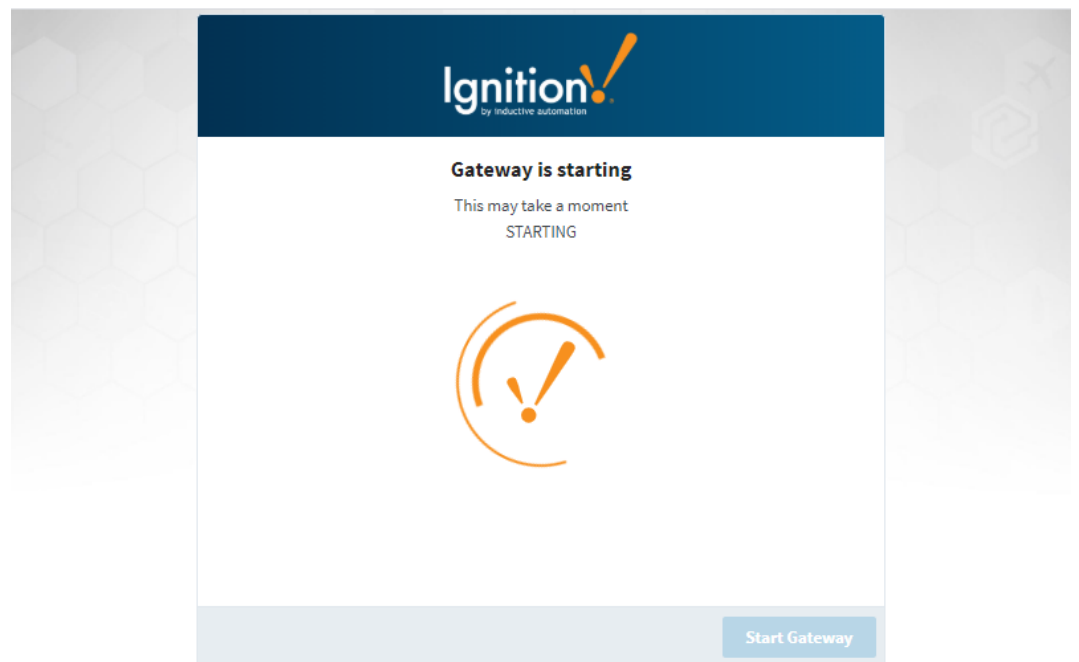


The screenshot shows the 'Http Configuration' screen of the Ignition Gateway. At the top, there is a dark blue header with the Ignition logo and the text 'by inductive automation'. Below the header, the title 'Http Configuration' is displayed. A paragraph of text explains that the user can configure which ports the Ignition Gateway should bind to for Http and Https. There are two input fields: 'HTTP Port' with the value '8088' and 'HTTPS Port' with the value '8043'. At the bottom of the screen, it says 'Step 3 of 3' and has a 'Next →' button.

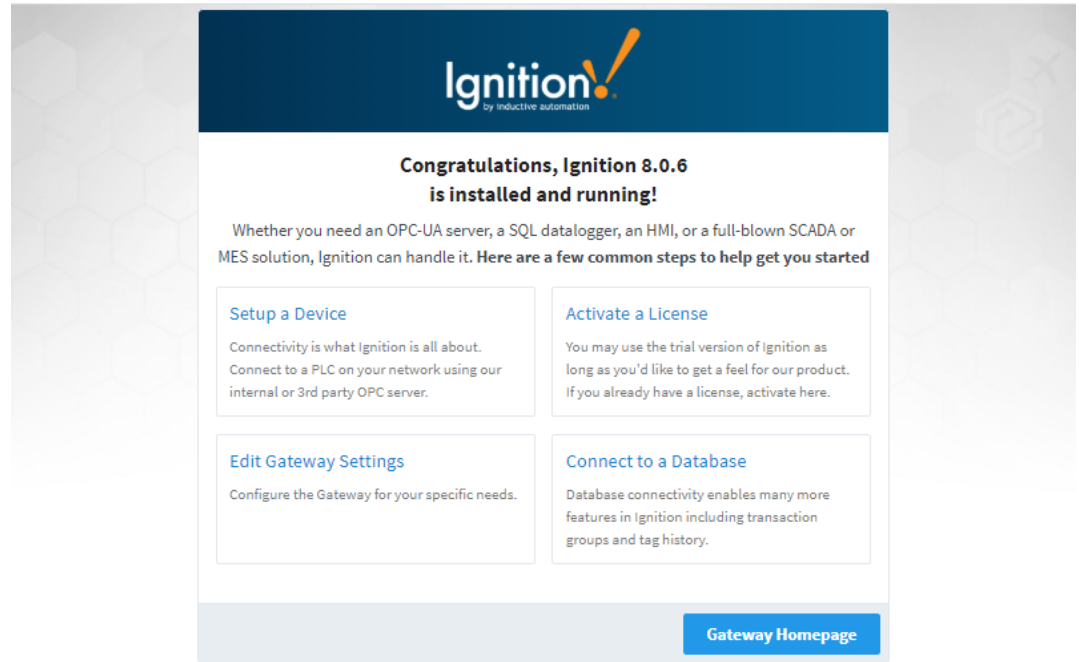
Message “Setup Complete” appears. Click on ‘Start Gateway’ to continue.



Message “Gateway Starting Up” appears.

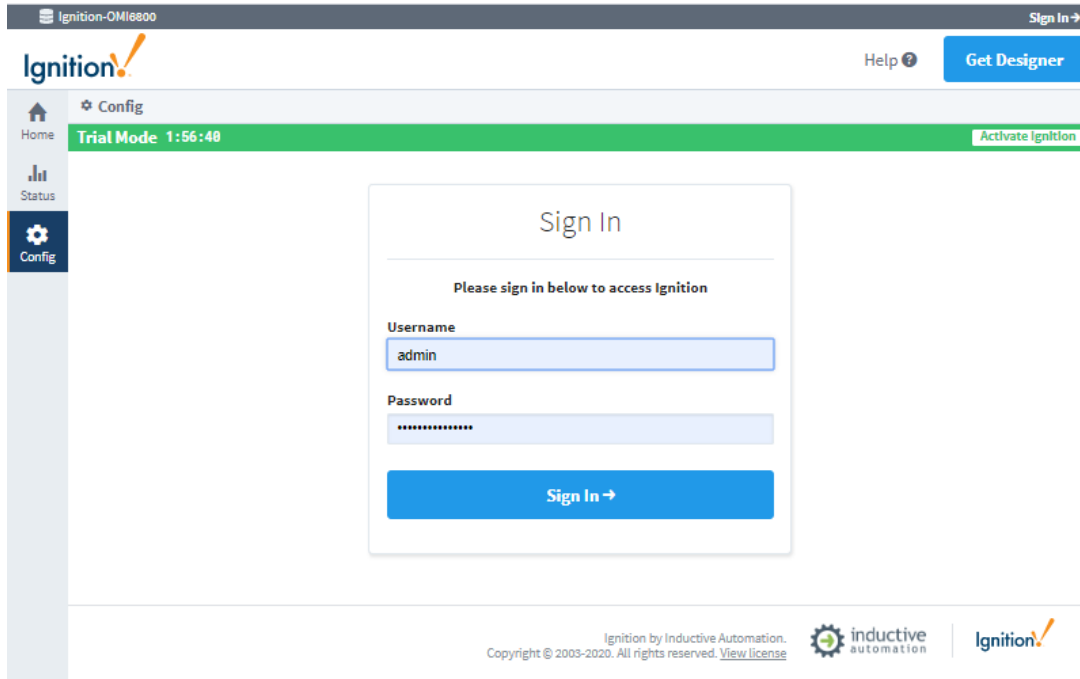


Message “Congratulations: Installed and Running” appears. Click on ‘Gateway Homepage’.



Login and Activate License

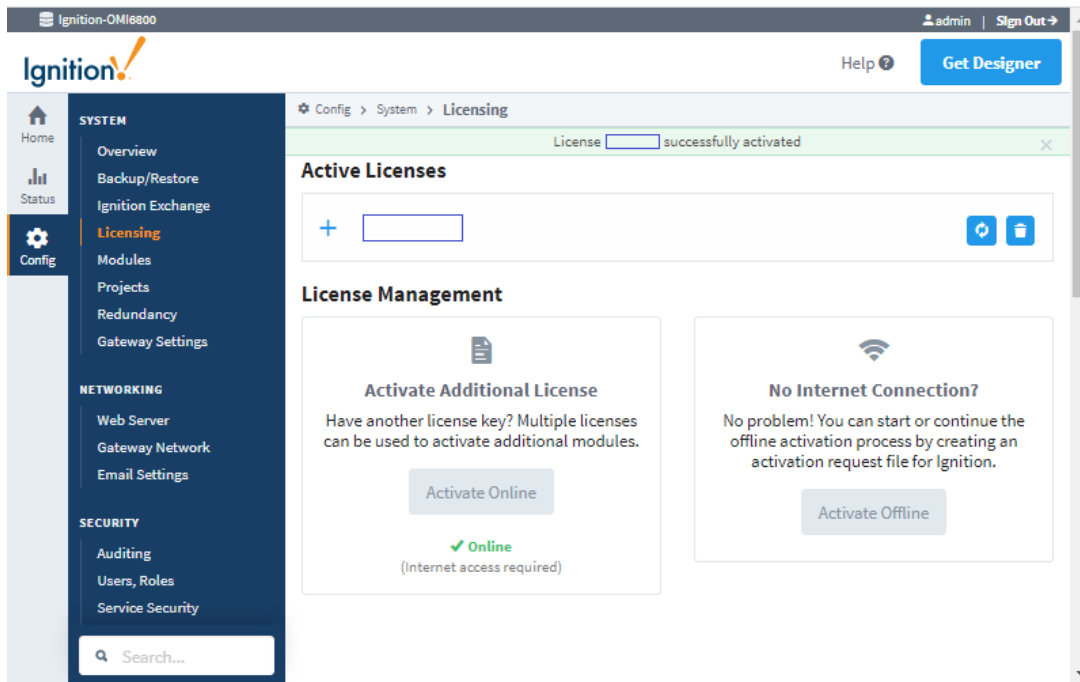
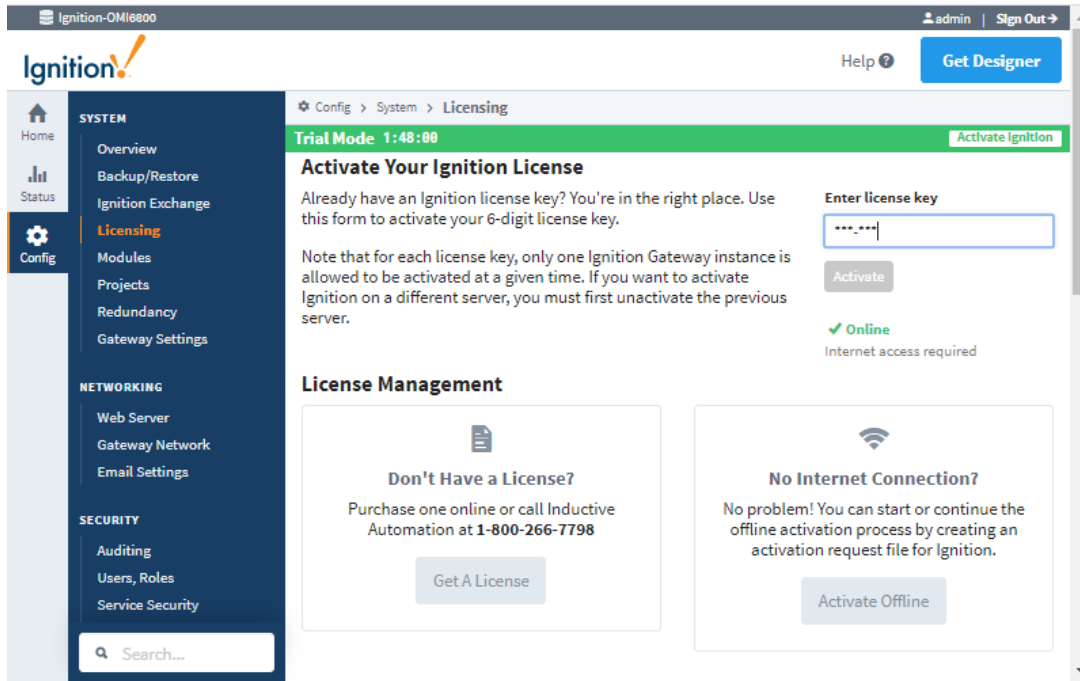
1. Enter your Username and Password. Then click ‘Sign In’.



2. Click on 'Activate Ignition'. Then select 'Activate Online'.

The image shows two screenshots of the Ignition-OMI6800 configuration interface. The top screenshot is the 'System' configuration page. The left sidebar contains navigation options: SYSTEM (Overview, Backup/Restore, Ignition Exchange, Licensing, Modules, Projects, Redundancy, Gateway Settings), NETWORKING (Web Server, Gateway Network, Email Settings), SECURITY (Auditing, Users, Roles, Service Security), and DATABASES. The main content area is titled 'Gateway Settings' and contains four sections: System Name (Ignition-OMI6800), System User Source (default), Gateway Config Role(s) (Administrator), and Status Page Role(s) (Administrator). A green banner at the top indicates 'Trial Mode 1:56:09' and 'Activate Ignition'. The bottom screenshot is the 'Licensing' configuration page. The left sidebar is similar, but 'Licensing' is highlighted. The main content area shows 'No License Installed' with a 'Get a License' button. Below this is the 'License Management' section with two options: 'Activate a License' (with an 'Activate Online' button and a green checkmark indicating 'Online (Internet access required)') and 'No Internet Connection?' (with an 'Activate Offline' button).

3. Enter your license key (Format: 6 characters with a dash in the middle). Then click 'Activate'.



2. Install Cirrus Link MQTT Modules: Distributor, Engine, Transmission

Click on 'Config' > 'Modules'.
The modules installed by default are shown.

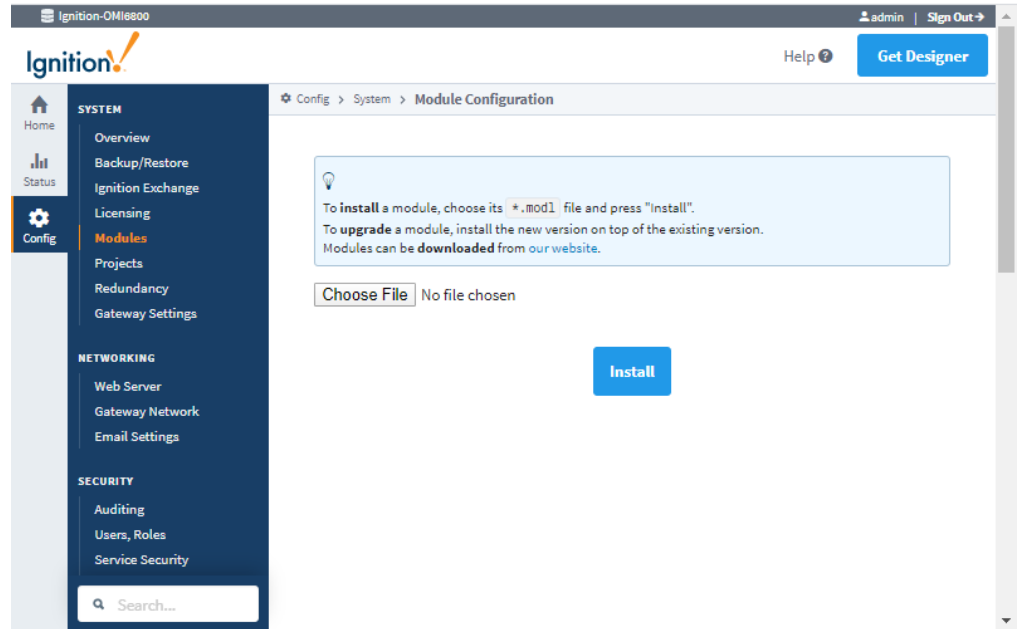
The screenshot shows the Ignition-OM16800 web interface. The top navigation bar includes 'admin' and 'Sign Out'. The main content area is titled 'Module Configuration' and displays a list of installed modules under the 'Inductive Automation' section. The modules listed are:

Name	Version	Description	License	State	More	restart
Alarm Notification	5.0.6 (b2019111216)	Provides alarm notifications via email	Activated	Running	More	restart
Allen-Bradley Driver	5.0.6 (b2019111216)	Allen-Bradley driver suite for the OPC UA module.	Activated	Running	More	restart
DNP3 Driver	3.0.6 (b2019111216)	A driver supporting DNP3 (Distributed Network Protocol) device.	Activated	Running	More	restart
Enterprise Administration	3.0.6 (b2019111216)	A remote Gateway administration system, allowing you to manage Gateways and automate tasks from a single controller.	Activated	Running	More	restart
SQL Bridge	9.0.6 (b2019111216)	An OPC-to-SQL data logger and transaction manager.	Activated	Running	More	restart
Symbol Factory	6.0.6 (b2019111216)	Vector graphics clipart library for the Vision module.	Activated	Loaded	More	restart
Tag Historian	3.0.6 (b2019111216)	Turns any database into a powerful historian that can store and drive data in Ignition.	Activated	Running	More	restart
UDP and TCP Drivers	5.0.6 (b2019111216)	Drivers for receiving and parsing UDP or TCP packets.	Activated	Running	More	restart
Vision	10.0.6 (b2019111216)	A module that provides web-launched HMI/SCADA clients.	Activated	Running	More	restart

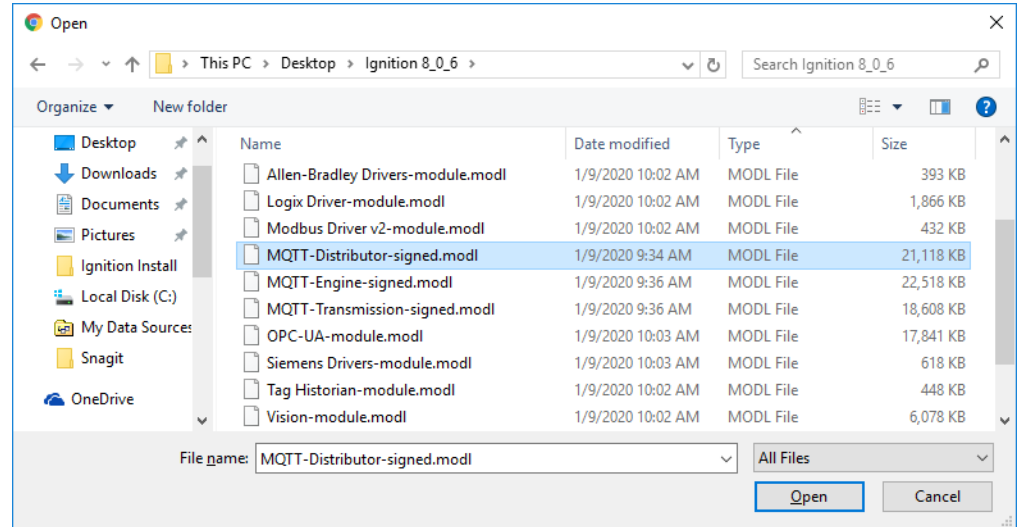
At the bottom of the page, there is a link to 'Install or Upgrade a Module...' and a note: 'Note: For details about a module's status, see the Module Status page.'

Scroll down and click on 'Install or Upgrade a Module...'

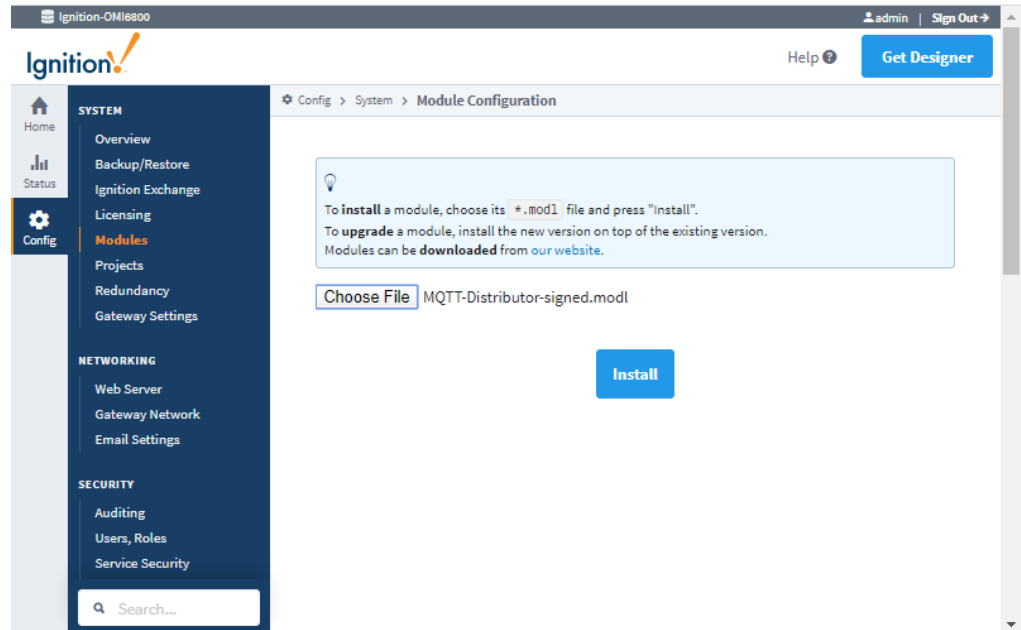
Click on 'Choose File'.



Select the 'MQTT-Distributor-signed.mod1' file download previously.



Click 'Install'.



Review the Cirrus Link End User License Agreement. Check the box 'I accept the terms...' and click 'Accept License'.

The screenshot shows the Ignition software interface. The top navigation bar includes the Ignition logo, user name 'admin', and 'Sign Out' button. The main content area is titled 'Module Configuration' and displays the 'Cirrus Link Solutions Software License Agreement (CLSSLA)'. The agreement text states: 'IMPORTANT - READ CAREFULLY: THIS SOFTWARE MODULE LICENSE IS IN ACCORDANCE WITH CIRRUS LINK SOLUTIONS SOFTWARE LICENSE AGREEMENT ("CLSSLA") AND IS A LEGAL AGREEMENT BETWEEN YOU (EITHER AN INDIVIDUAL OR A SINGLE ENTITY) AND CIRRUS LINK SOLUTIONS. BY INSTALLING, COPYING OR OTHERWISE USING THE SOFTWARE MODULE, YOU INDICATE YOUR ACCEPTANCE OF THE CLSSLA. IN ORDER TO USE THIS MODULE SOFTWARE, YOU MUST HAVE ALREADY INSTALLED THE IGNITION BY INDUCTIVE AUTOMATION® SOFTWARE. YOU AGREE THAT THE CIRRUS LINK SOLUTIONS SOFTWARE LICENSE AGREEMENT ("CLSSLA") CONTROLS ALL ASPECTS OF THE RELATIONSHIP BETWEEN THE PARTIES WITH REGARD TO THE LICENSING OF THE MODULE SOFTWARE AND SUPERCEDES ANY OTHER AGREEMENT BETWEEN YOU AND CIRRUS LINK SOLUTIONS, INCLUDING, BUT NOT LIMITED TO, LICENSEE PURCHASE ORDERS AND/OR TERMS AND CONDITIONS, AND WHETHER ENTERED INTO BEFORE OR AFTER YOUR ACCEPTANCE OF THIS AGREEMENT. IF YOU DO NOT ACCEPT THE TERMS OF THIS AGREEMENT, DON'T INSTALL, COPY OR OTHERWISE USE THIS SOFTWARE MODULE. BY DOWNLOADING THE MODULE SOFTWARE, YOU ARE ACCEPTING THE TERMS OF THIS AGREEMENT. Copyright © 2015-2018. Cirrus Link Solutions. All rights reserved. http://www.cirrus-link.com Cirrus Link Solutions module licensed through the CLSSLA. This agreement is available on'.

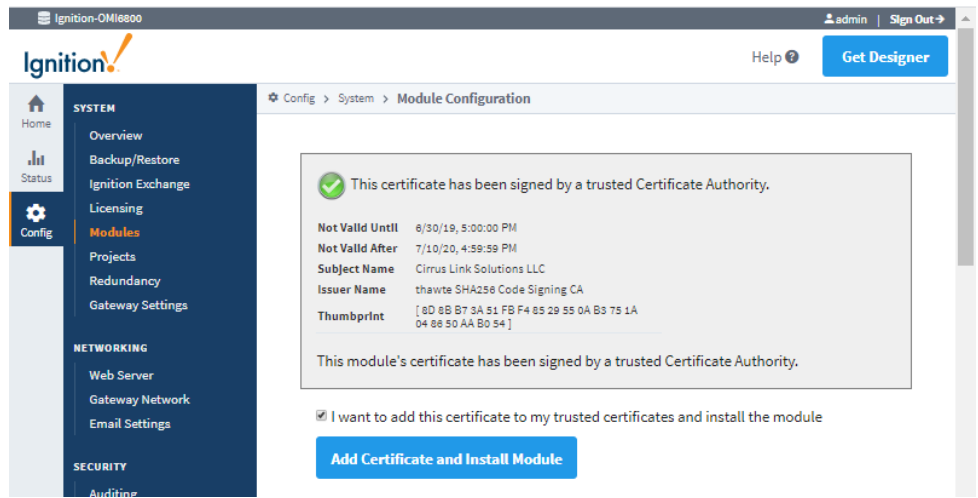
Below the main agreement, there are three sections of license information:

- G. HSQLDB**
Copyright © 2001-2010, The HSQL Development Group. All rights reserved.
Copyright © 1995-2000 by the Hypersonic SQL Group. All rights reserved.
The Software contains HSQLDB, which is made available under the HSQLDB License (based on BSD License). You can get the full source code HSQLDB at: <http://hsqldb.org>. A copy of the HSQLDB License is available at <http://hsqldb.org/web/hsqldbLicense.html>
- H. MapDB**
Copyright © 2012-2015 Jan Kotek
The Software contains MapDB, which is protected under the Apache License, Version 2.0. You can get the full source code for MapDB at: <https://github.com/jankotek/mapdb>. A copy of the Apache License, Version 2.0 is available at <http://www.apache.org/licenses/LICENSE-2.0>
- I. SLF4J**
Copyright © 2004-2015 QOS.ch
The Software contains SLF4J, which is protected under the MIT License. You can get the full source code for SLF4J at: <http://www.slf4j.org/download.html>. A copy of the MIT License is available at <http://www.slf4j.org/license.html>

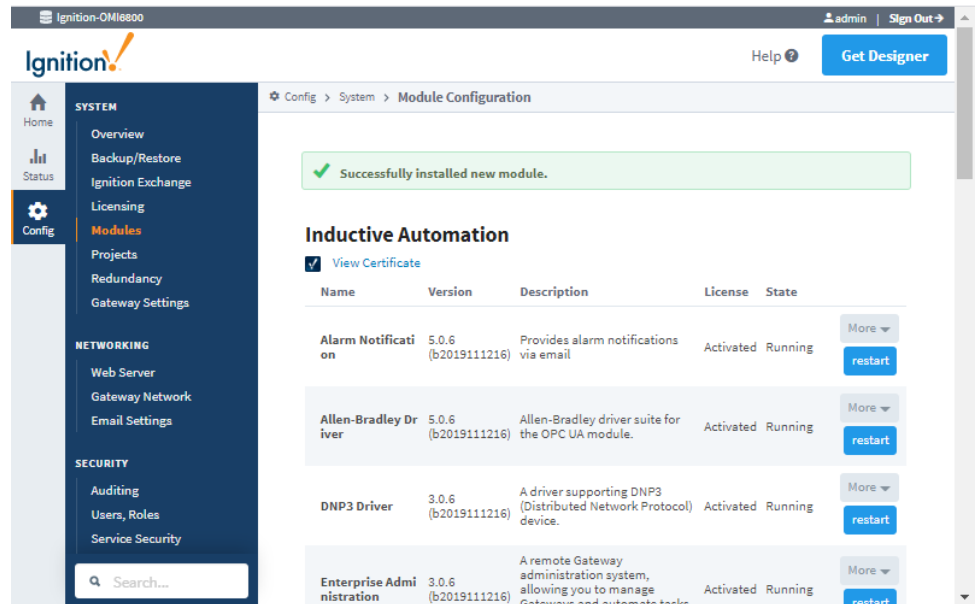
At the bottom of the license information, there is a checkbox labeled 'I accept the terms in the License Agreement' which is checked. Below the checkbox is a blue button labeled 'Accept License' and a '< Back' link.

The footer of the interface includes the text 'Ignition by Inductive Automation. Copyright © 2003-2020. All rights reserved. View license' and logos for 'inductive automation' and 'Ignition!'.

Check the box for “I want to add this certificate to my trusted certificates and install the module”. Then click ‘Add Certificate and Install Module’.



Message “Successfully installed new module” appears.



Repeat the process as shown above to install the remaining two Cirrus Link MQTT Modules:

- *MQTT Engine* – Filename: ‘MQTT-Engine-signed.modl’
- *MQTT Transmission* – Filename: ‘MQTT-Transmission-signed.modl’

Once complete, you should have three modules installed and listed under the *Cirrus Link Solutions LLC* section, as seen in the following screenshot:

Cirrus Link Solutions LLC

Name	Version	License	Status
MQTT Distributor	4.0.2 (b2019101100)	Activated Q	✓ RUNNING
MQTT Engine	4.0.2 (b2019101100)	Activated Q	✓ RUNNING
MQTT Transmission	4.0.2 (b2019101100)	Activated Q	✓ RUNNING

3. Ignition Gateway Settings: Set 'Platform Name'

- Click on 'Config' > 'Gateway Settings'
- Enter a name in the field provided for 'System Name'
 - For example: *Ignition-Gateway*
- Scroll down and click 'Save Changes'

The screenshot displays the Ignition Gateway configuration interface. The top navigation bar includes the Ignition logo, a user profile for 'admin', and a 'Sign Out' button. A 'Get Designer' button is also visible. The left sidebar contains a 'Config' menu with a search bar and a list of system settings categories: SYSTEM (Overview, Backup/Restore, Ignition Exchange, Licensing, Modules, Projects, Redundancy, Gateway Settings), NETWORKING (Web Server, Gateway Network, Email Settings), and SECURITY (Auditing, Users, Roles, Service Security, Identity Providers, Security Levels, Security Zones). The main content area shows the 'Gateway Settings' page, which is a table with the following rows:

Gateway Settings	
System Name	<input type="text" value="Ignition-Gateway"/> The name of this Ignition system, used to differentiate this system from others in a larger architecture. (default:)
System User Source	<input type="text" value="default"/> This user source controls access to the Gateway's web configuration interface and the Designer.
Gateway Config Role(s)	<input type="text" value="Administrator"/> Users must belong to one of these roles in order to log into the configuration section. Multiple roles can be specified by separating them with commas. (default: Administrator)
Status Page Role(s)	<input type="text" value="Administrator"/> Users must belong to one of these roles in order to log into the status section. Multiple roles can be specified by separating them with commas. (default: Administrator)
Home Page Role(s)	<input type="text"/> Users must belong to one of these roles in order to log into the home section. Multiple roles can be specified by separating them with commas. If blank, the home page will not be password-protected. (default:)

Configure Service Security

1. From 'Config' > 'Security', click on 'Service Security'. By default, 'Policy Defined?' will be set to 'false'.
2. Click on 'edit' to configure the policy
3. Under 'Tag Access', use the drop-down menu change 'Default Provider Access Level' to 'ReadWriteEdit'
4. Click 'Save' to finish setting up the policy

The screenshot shows the 'Service Security' configuration page. The left sidebar contains navigation options: Home, Status, Config, TAGS (History, Realtime), OPC CLIENT (OPC Connections, OPC Quick Client), OPC UA (Device Connections, Security, Server Settings), ENTERPRISE ADMINISTRATION (Setup), SEQUENTIAL FUNCTION CHARTS (Settings), and MQTT DISTRIBUTOR. The main content area is titled 'Config > Security > Service Security'. The 'Tag Access' section includes the following fields:

Service Access	Allow
Default Provider Access Level	ReadWriteEdit
Impersonation Role Name	<input type="text"/>
Access Level: 'default'	Inherited
Access Level: 'System'	Inherited
Access Level: 'MQTT Distributor'	Inherited
Access Level: 'MQTT Transmission'	Inherited
Access Level: 'MQTT Engine'	Inherited

A 'Save' button is located at the bottom right of the configuration area.

The 'Policy Defined?' field should now be set to 'true'.

The screenshot shows the 'Service Security' configuration page after the policy has been updated. The left sidebar is now under the 'SYSTEM' section, with 'Service Security' highlighted. The main content area shows a success message: 'Successfully updated Security Zone Policy "Default"'. Below this is a warning message: 'Security policies are defined based on Security Zones. The highest ranking policy (from the top down) for a connection's zones will be used. If no other policies match, the "Default" policy will be used.' A table displays the updated policy:

Security Zone	Policy Defined?
Default	true

'Edit' and 'Clear Policy' buttons are located to the right of the table.

Configure Cirrus Link MQTT Modules

Configure MQTT Distributor

1. Click on 'Config', then click on 'Settings' under MQTT Distributor.
2. On the 'General' tab, scroll down and click on 'Show advanced properties'.
3. Check the box for 'Enable Anonymous MQTT Connections'
4. Click 'Save Changes'

The screenshot displays the 'MQTT Distributor Settings' configuration page. The left sidebar contains navigation options: Home, Status, Config, Drivers, Store and Forward, ALARMING (General, Journal, Notification, On-Call Rosters, Schedules), TAGS (History, Realtime), OPC CLIENT (OPC Connections, OPC Quick Client), OPC UA (Device Connections, Security, Server Settings), ENTERPRISE ADMINISTRATION (Setup), SEQUENTIAL FUNCTION CHARTS (Settings), and MQTT DISTRIBUTOR (Settings). The main content area shows the following settings:

- Secure MQTT Port:** 8883 (TLS enabled MQTT Server port)
- Enable Secure Websocket:** Enable Secure Websocket connections for the MQTT Server
- Secure Websocket Port:** 9443 (TLS enabled MQTT Server Websocket port)
- Keystore Password:** [Redacted]
- Java Keystore File:** Choose File | No file chosen (Java Keystore File to upload for SSL enabled MQTT)

Below these settings, the **Show advanced properties** checkbox is checked. Under the **Advanced** section, the **Allow Anonymous MQTT Connections** checkbox is checked, with the text: **Enable Anonymous MQTT Connections (NOT RECOMMENDED)** (default: false). A blue **Save Changes** button is located below the advanced properties section. A note at the bottom states: **Note:** For additional details on configuring MQTT Distributor, see the documentation [here](#).

NOTE: Once you have completed testing of a Local Connection between your Maple cMT Device and Ignition gateway, we recommend that you uncheck the 'Allow Anonymous MQTT Connections' and use password-based authentication to control which devices can connect to Ignition.

- EBPro: IloT/Energy > MQTT > Settings > General Tab: Enter your Authentication Details (Username & Password)
 - OR: Use the MQTT Control Address: Username (Ctrl Addr + 27); Password (Ctrl Addr + 43)
- See EBPro "Project Configuration" in Section #5 of this document for more details

Configure MQTT Engine

1. Click on 'Config', then click on 'Settings' under MQTT Engine.
2. From the 'General' tab, choose a 'Primary Host ID'. Type this into the given field under the 'Main' section.
 - a. For example: *maple-ignition*
3. Next, uncheck the following options under 'Miscellaneous':
 - a. Uncheck 'Block outbound edge node tag writes'
 - b. Uncheck 'Block outbound device tag writes'

The screenshot shows the Ignition web interface for configuring the MQTT Engine. The breadcrumb trail is 'Config > Mqttengine > MQTT Engine Settings'. The 'General' tab is selected. The 'Main' section contains the following settings:

- Enabled:** Enable the MQTT Engine
- Primary Host ID:** The Primary Host ID to allow connecting clients to ensure they remain connected to this application (optional)
- Group ID Filters:** A comma separated list of Group IDs to listen for (optional)

The 'Chariot Access' section contains the following settings:

- Chariot Cloud Access Key:** The optional Chariot Cloud Access Key used for Cirrus Link hosted Chariot MQTT Servers (optional)
- Chariot Cloud Secret Key:** The optional Chariot Cloud Secret Key used for Cirrus Link hosted Chariot MQTT Servers (optional)

The 'Miscellaneous' section contains the following settings:

- Block Node Commands:** Block outbound edge node tag writes
- Block Device Commands:** Block outbound device tag writes
- Block Property Changes:** Block incoming Tag property changes
- File Policy:**

The left sidebar shows the navigation menu with categories: SYSTEM, NETWORKING, SECURITY, DATABASES, and ALARMING. The 'Config' option is highlighted under SYSTEM. A search bar is located at the bottom of the sidebar.

4. Click 'Save Changes'
5. From the Servers tab, click 'edit' on the existing 'Chariot SCADA' server:
 - a. Enter the username ('admin') and password in the fields provided
6. Click 'Save Changes'

Configure MQTT Transmission

1. From 'Config' > MQTT Transmission 'Settings', go to the 'Sets' tab
2. Click on 'Create new MQTT Server Set...'
 - a. Enter a Name and Primary Host ID. For example:
 - i. Name: *maple-ignition*
 - ii. Primary Host ID: *maple-ignition*
 - b. Click 'Create New MQTT Server Set'

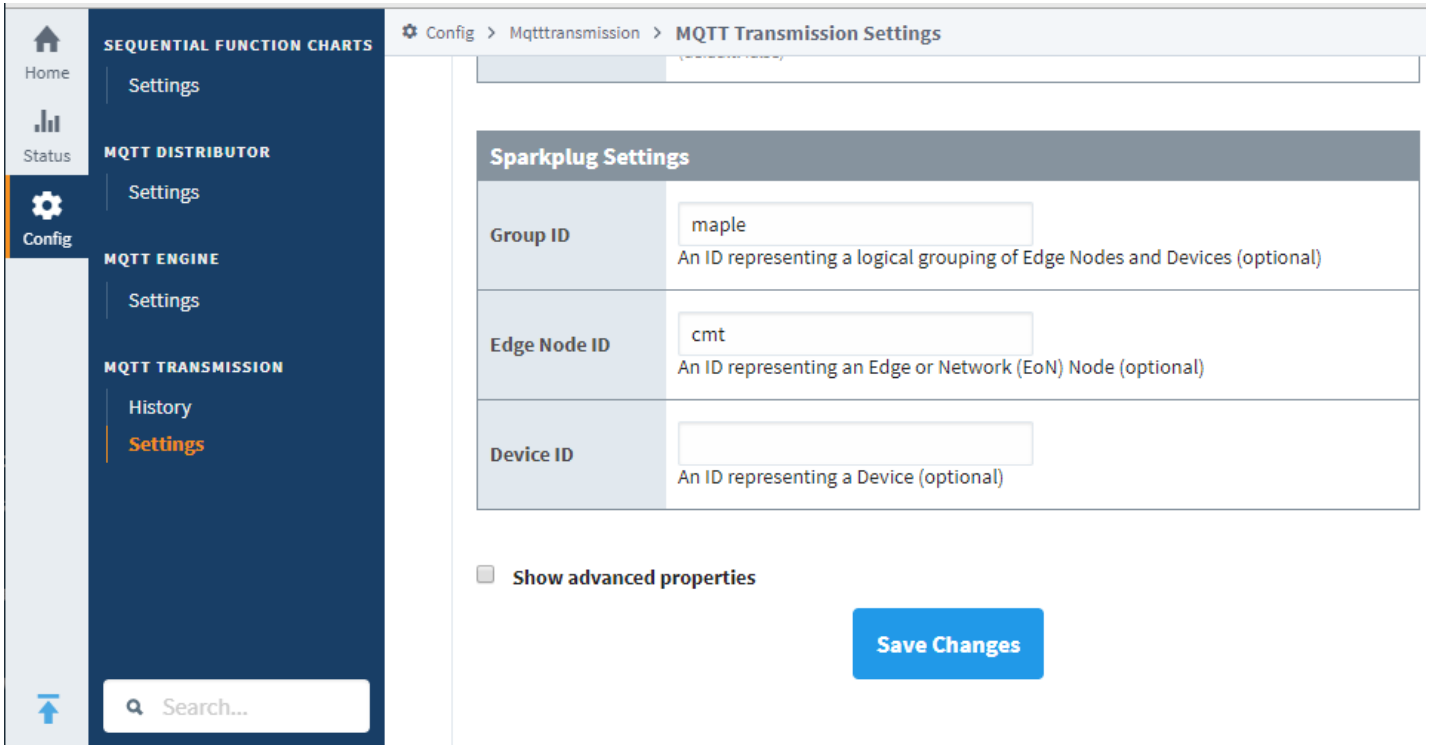
The screenshot displays the Ignition software interface. The top left corner features the Ignition logo and a navigation menu with icons for Home, Status, and Config. The main navigation pane on the left is divided into three sections: SYSTEM (Overview, Backup/Restore, Ignition Exchange, Licensing, Modules, Projects, Redundancy, Gateway Settings), NETWORKING (Web Server, Gateway Network, Email Settings), and SECURITY (Auditing, Users, Roles, Service Security, Identity Providers). The top right corner includes a Help icon and a 'Get Designer' button. The breadcrumb trail reads 'Config > Mqtttransmission > MQTT Transmission Settings'. The main content area has five tabs: General, Servers, Sets (selected), Transmitters, and Records. The 'Sets' tab is active, showing a 'Main' section with three fields: 'Name' (value: maple-ignition, description: The friendly name of this MQTT Server Set), 'Description' (empty field, description: Description of this MQTT Server Set), and 'Primary Host ID' (value: maple-ignition). A blue 'Save Changes' button is located at the bottom right of the form.

3. Next, from the 'Transmitters' tab, click 'Create new Settings...'
 - a. Enter a 'Name' for the Transmitter. For example: *maple-ignition*
 - b. Select the 'Set' created previously from the drop-down menu. (e.g. *maple-ignition*)

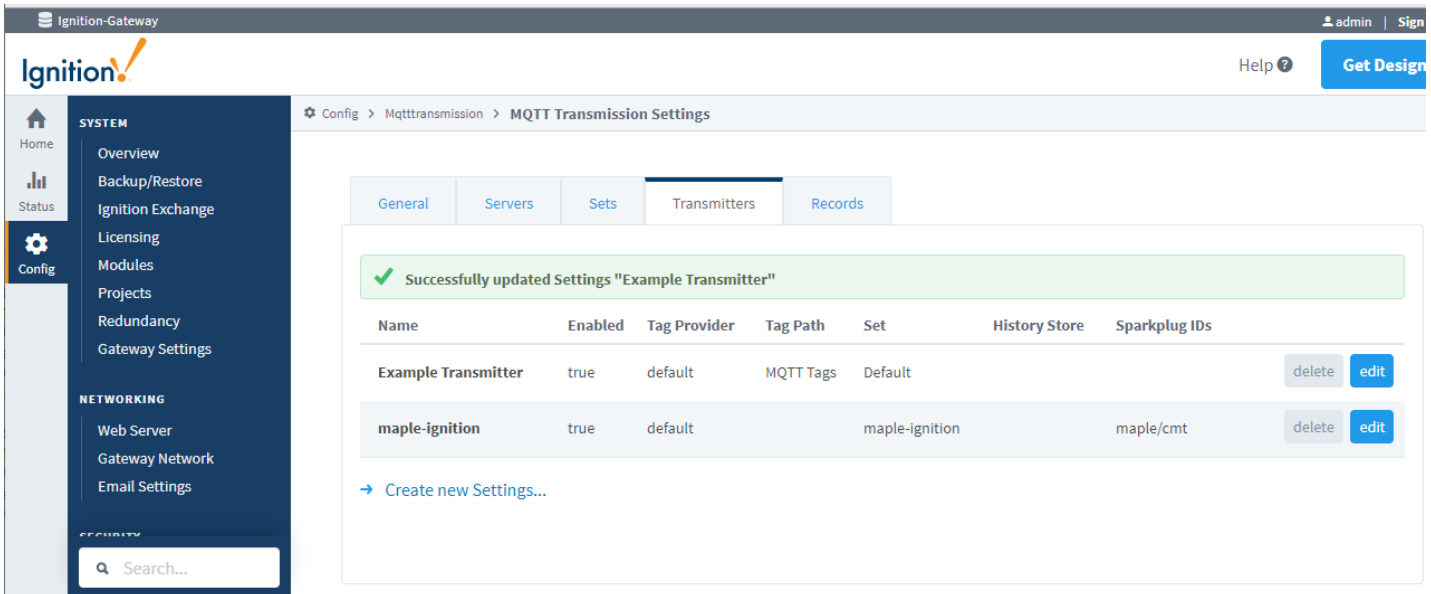
The screenshot displays the 'MQTT Transmission Settings' interface. The left sidebar contains navigation options: Home, Status, Config, NETWORKING (Web Server, Gateway Network, Email Settings), SECURITY (Auditing, Users, Roles, Service Security, Identity Providers, Security Levels, Security Zones), DATABASES (Connections, Drivers, Store and Forward), ALARMING (General, Journal, Notification, On-Call Rosters, Schedules), and TAGS (Search...). The main content area shows the 'Transmitters' tab with the following settings:

Tag Settings	
Name	maple-ignition A unique name for the Transmitter
Enabled	<input checked="" type="checkbox"/> Enable Transmitter
Tag Provider	default The Name of the tag provider
Tag Path	<input type="text"/> A path to the root folder where the tag tree starts (optional)
Tag Pacing Period	1000 The waiting period in milliseconds after an initial tag change event before publishing all changed tags (default: 1,000)
Set	maple-ignition The MQTT Server Set to use with this Transmitter
Discovery Delay	0 The Transmitter Discovery Delay in milliseconds. This is useful when using MQTT Engine as the tag provider (default: 0)
Aliased Tags	<input type="checkbox"/> Use aliases for tag names to optimize payload sizes when publishing data
Compression	NONE The algorithm to use for compressing payloads before publishing
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

- c. Scroll down to 'Sparkplug Settings'.
 - i. Enter a 'Group ID'. E.g. *maple*
 - ii. Enter a 'Edge Node ID'. E.g. *cmt*
4. Click 'Create New Settings' (Save Changes)



Under the Transmitters tab, you should now have two transmitters set up. For example:



5. From the 'Servers' tab, click on 'edit' for the existing server named 'Chariot SCADA'
6. From the 'Server Set' drop-down menu, select the 'maple-ignition' Set created previously
7. Enter the Ignition Gateway username ('admin') and password in the fields provided
8. Click 'Save Changes'

You should now have a single Server connected to 'maple-ignition' as shown below:

The screenshot shows the 'MQTT Transmission Settings' page in a web application. The left sidebar contains navigation options under 'SYSTEM' (Overview, Backup/Restore, Ignition Exchange, Licensing, Modules, Projects, Redundancy, Gateway Settings) and 'NETWORKING' (Web Server, Gateway Network, Email Settings). The main content area has tabs for 'General', 'Servers', 'Sets', 'Transmitters', and 'Records'. The 'Servers' tab is active, displaying a table with one server entry: 'Chariot SCADA' with URL 'tcp://localhost:1883', Server Set 'maple-ignition', Username 'admin', and '1 of 1' connected. Below the table is a link to 'Create new MQTT Server...' and a note box with a link to documentation.

Name	URL	Server Set	Username	Certificate Files	Connected
Chariot SCADA	tcp://localhost:1883	maple-ignition	admin		1 of 1

→ [Create new MQTT Server...](#)

Note: For additional details on configuring MQTT Transmission, see the [documentation here](#)

Ignition Gateway is now configured to work with Maple Systems cMT devices.

4. Install Ignition Designer


1. From the 'Home' page in Ignition Gateway, click on the 'Download' button next to 'Ignition Designer Launcher'.

Ignition-OMI6800 admin | Sign Out →


Ignition! Help ? Get Designer


Home > Get Started

Download Ignition Designer Launcher

 **Ignition Designer Launcher**
Download the Ignition Designer Launcher to create or modify your projects. [Download](#)

Download Application Launchers

 **Vision Client Launcher**
Download the Native Client Launcher to open Vision clients from any Ignition Gateway. [Download Vision Client Launcher](#)

 **Perspective Session Launcher**
Launch a Perspective session directly in your browser or download the native application. [View Projects](#)

Learn Ignition

Take advantage of our tools to get designing quickly and take your ideas from concept to reality. The User Manual is a wealth of easily searchable knowledge and the Inductive University has hundreds of short videos covering the basics of Ignition.

2. Click 'Download for Windows'

Ignition-Gateway admin | Sign Out →

Ignition! Help ? Get Designer

Home > Designer Launcher Download




Download the Designer Launcher

[Download for Windows](#)

We've detected you're on Windows. Download the Designer Launcher for Windows and follow these steps below to install.

Alternative Designer Launchers

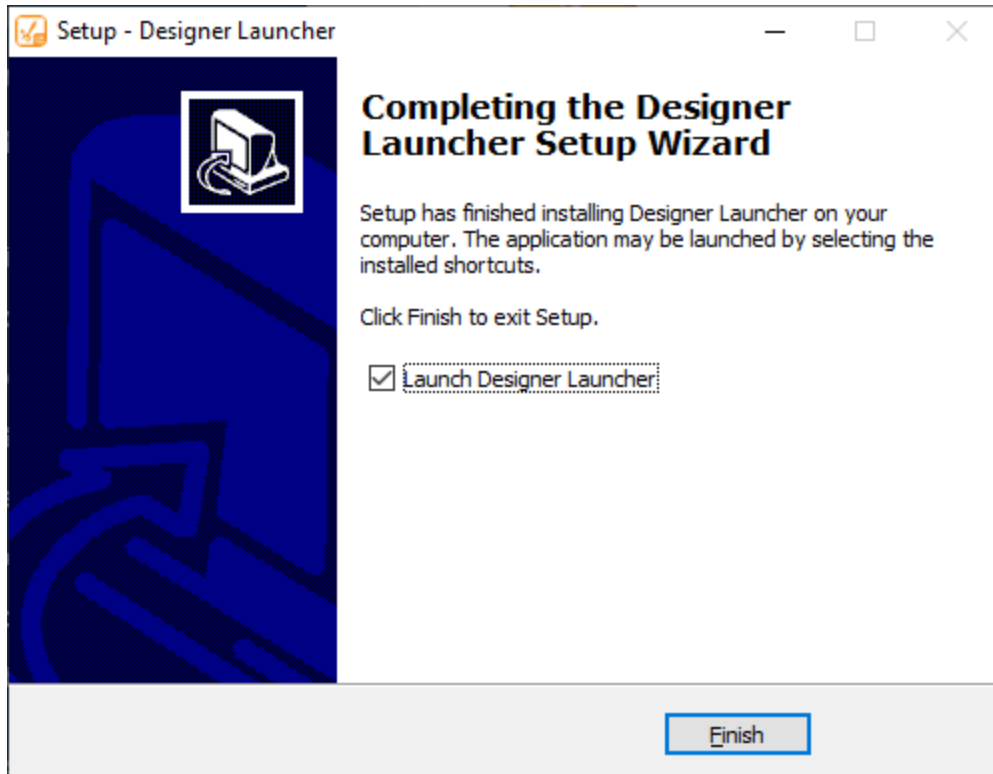
The Designer Launcher is also available for these operating systems.

-  **Windows** 45.9MB
-  **Mac** 42.8MB
-  **Linux** 55.3MB

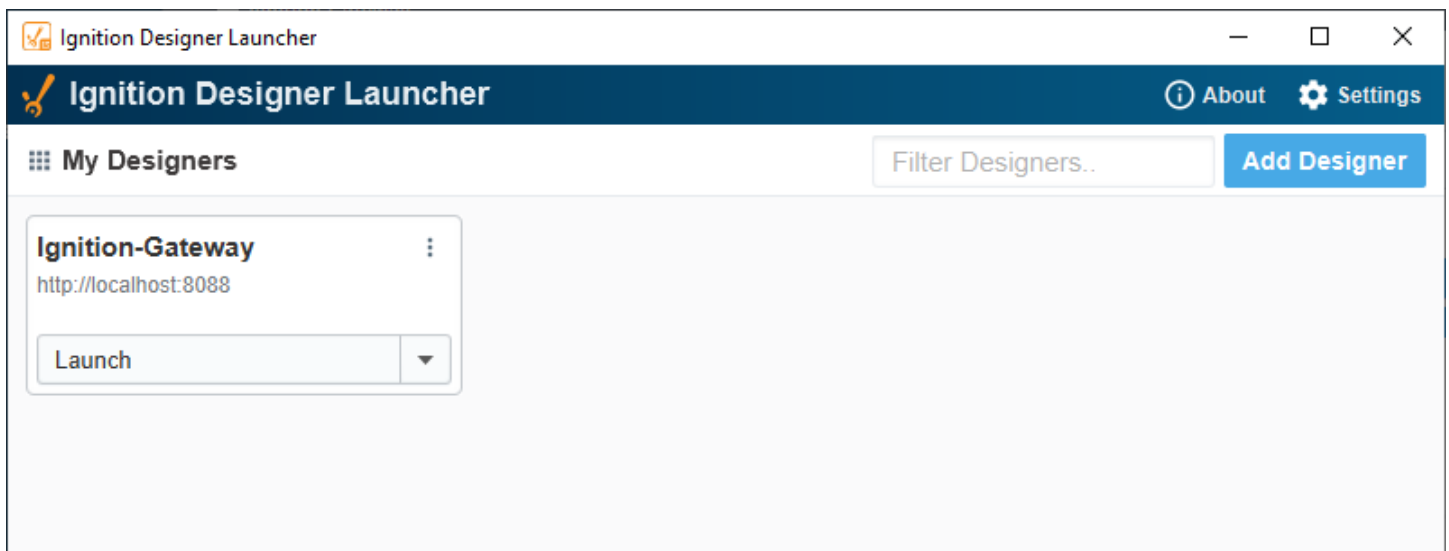
- 1. Run the Windows Designer Installer**
Click Run or Save to initialize the download.

*If you chose Save, **double-click** the download to start installing.
- 2. Follow Setup Instructions**
Follow the instructions to get the Launcher installed on your computer. After the installation is complete you will be asked if you want to launch now.
- 3. Provide Firewall Access**
On the majority of windows machines you will need to allow access to the firewall for the application to work properly. Windows Defender will prompt you the first time you run the application. **Click Allow Access.**

3. Open and run the Designer Launcher Installer. Follow the instructions to install and then launch the Designer Launcher once installation is complete.

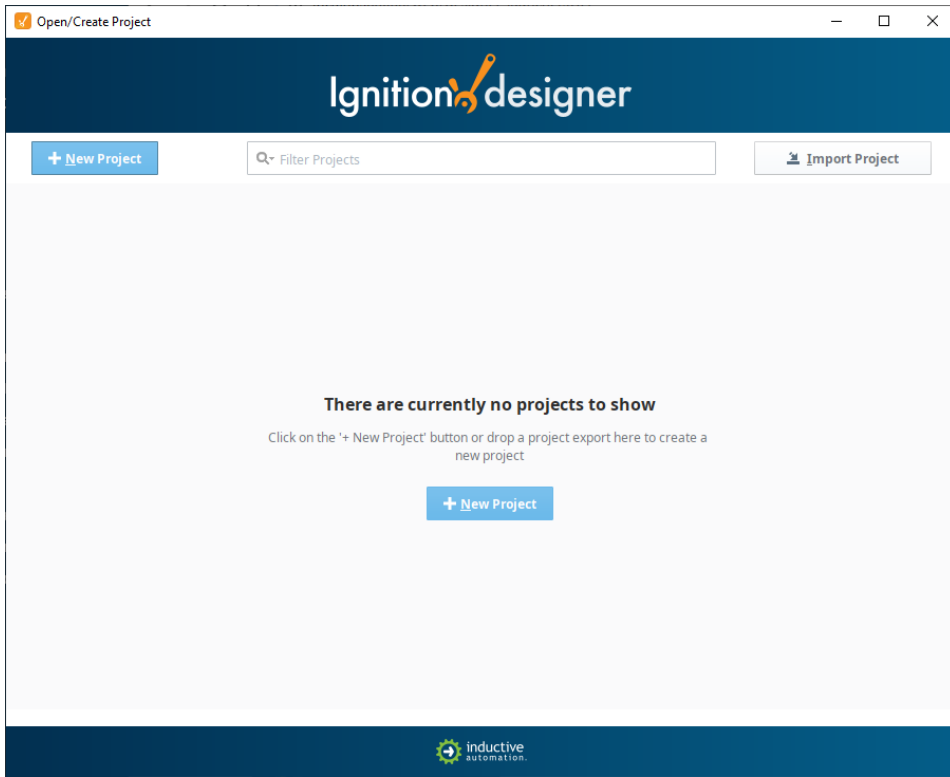


4. If you don't see your Ignition Gateway Designer in the Launcher window, click 'Add Designer' and select from the available Ignition Gateways.



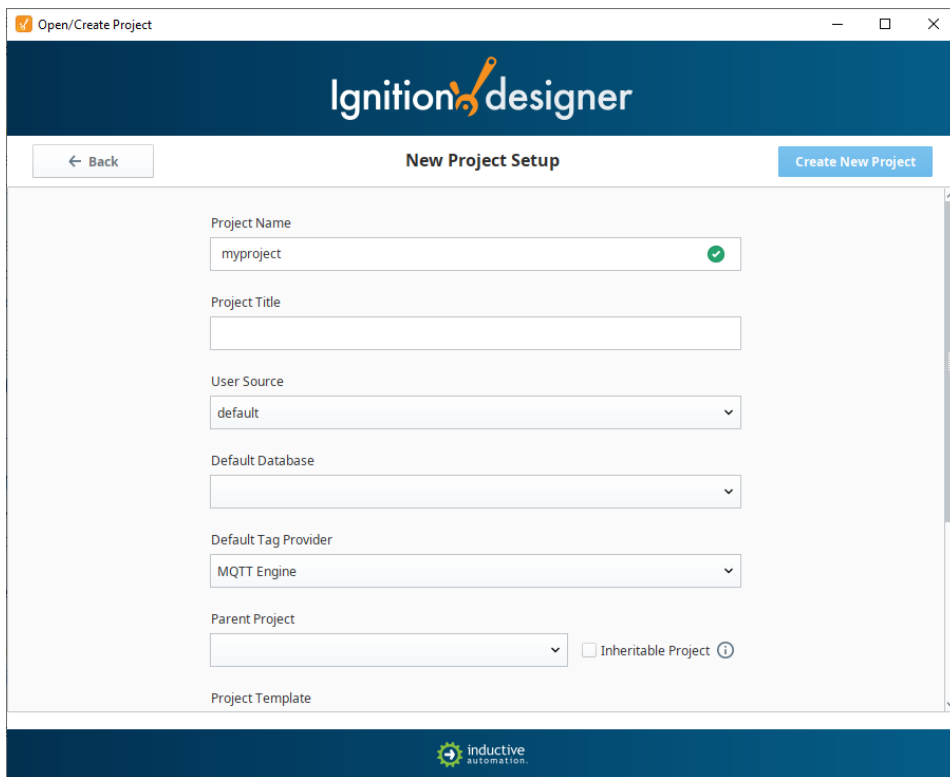
5. Click the 'Launch' button on the tile for your Designer.
6. Enter your Ignition Gateway Username and Password when prompted.

7. Once Designer opens, click 'New Project'.

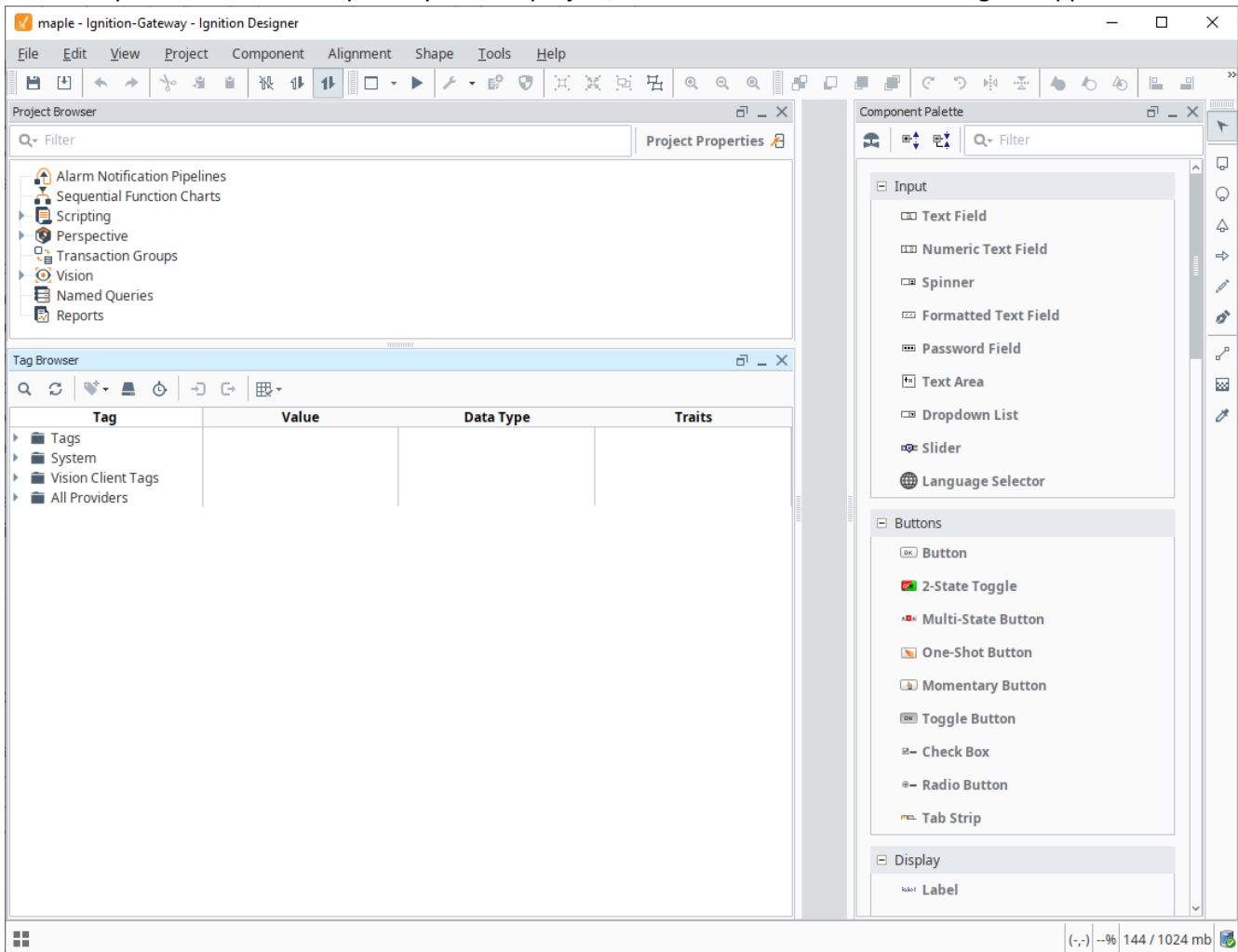


8. Give your project a name.

9. For Default Tag Provider, select 'MQTT Engine'.



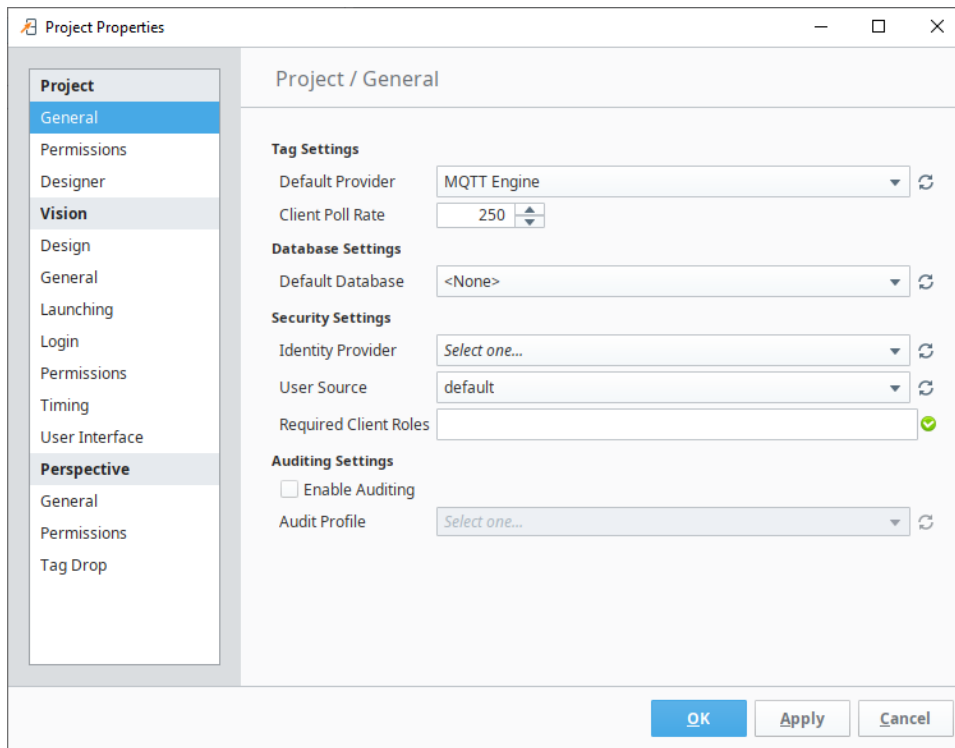
10. Once you've created and opened your first project, a window similar to the following will appear:



11. From the 'Project' menu, select 'Project Properties'. By default, an Identity Provider is not selected.

12. Click on the drop-down menu next to 'Identity Provider' and select 'default'.

13. Click OK to save these settings.

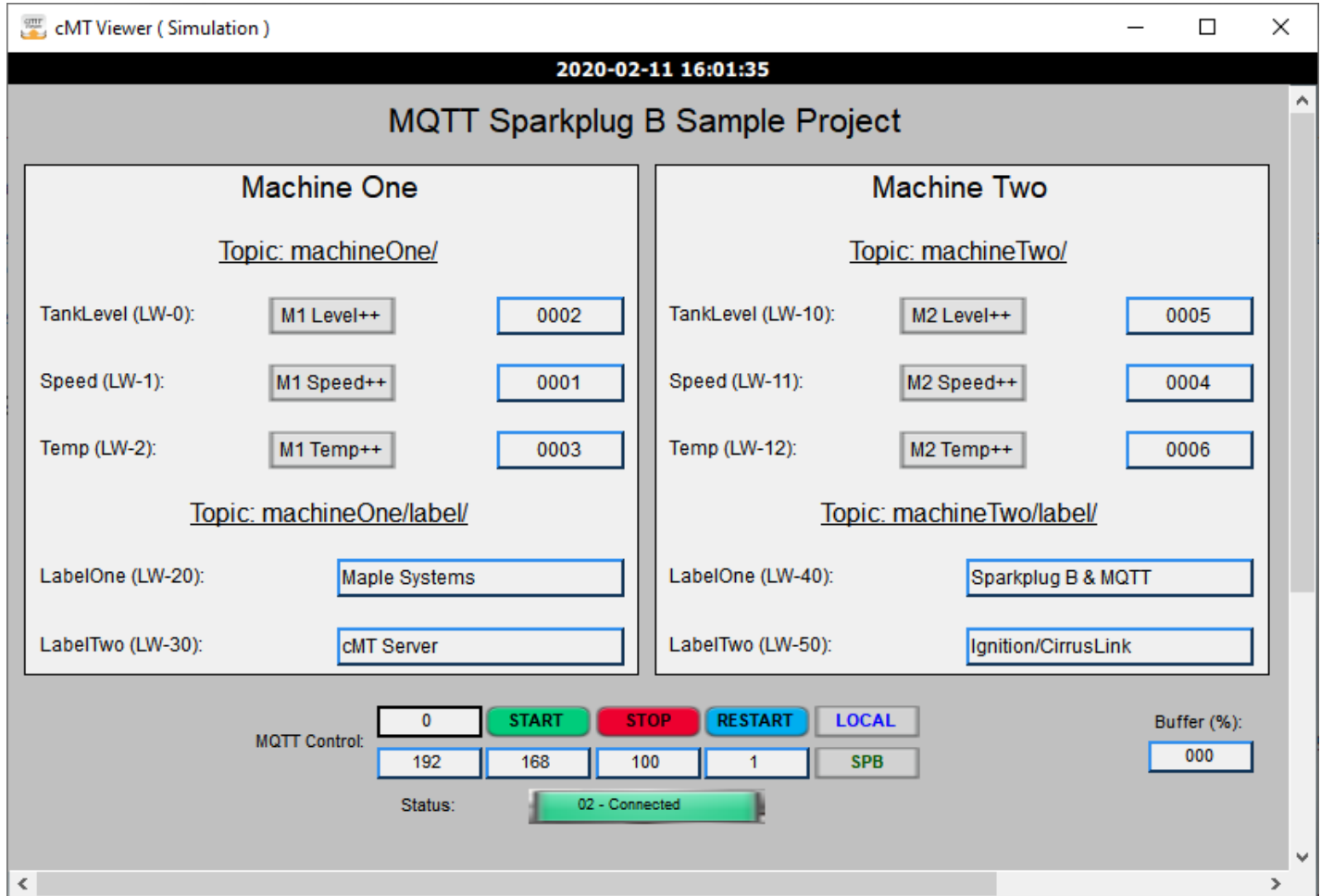


Your Ignition Designer Project is now configured to work with Maple Systems cMT devices.

5. Prepare a Maple Systems EBPro Sample Project and Connect to Ignition

Visit our [Sample Projects](#) page in order to download a free copy of our *Sparkplug B MQTT Sample Project*. You can use this for testing purposes or adapt it for your own application.

Sparkplug B MQTT Sample Project for EBPro – Click [here](#) to download this project

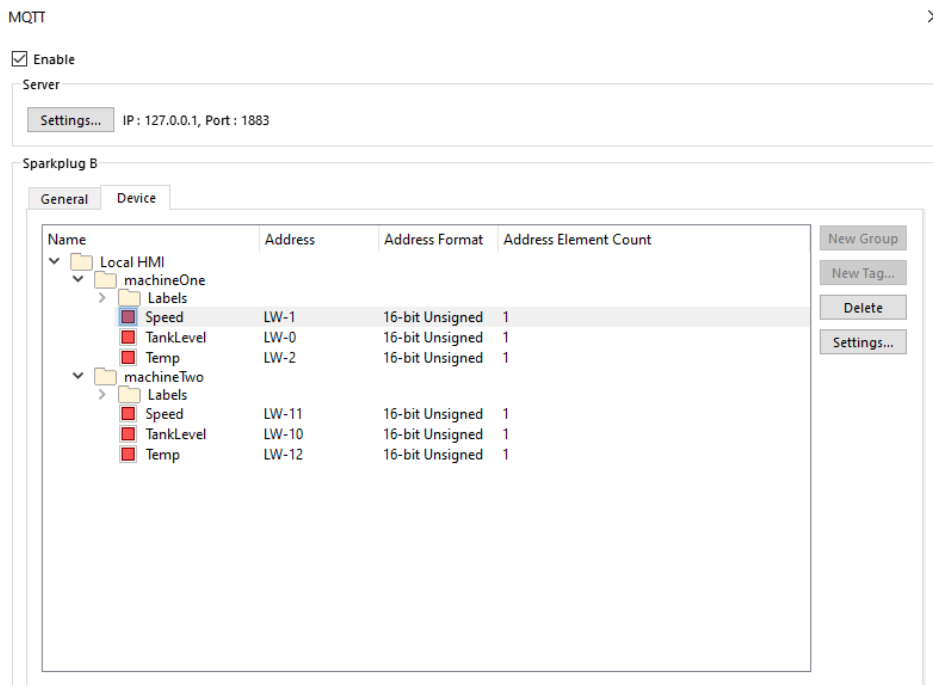


Software/Firmware Requirements for Connecting a Maple Systems cMT Device to Ignition

Maple Systems HMI Programming Software		
Program/FW	Version	Download Link
EBPro	6.01.01.192 or greater	https://www.maplesystems.com/SupportCenter/SoftwareDownloads.htm
Device OS, Firmware	20150923 or later	(Contact support@maplesystems.com for more information.)

Project Configuration for Connecting a Maple Systems cMT Device to Ignition

- From the 'IIoT/Energy' tab in EBPro, click on MQTT and check the box to *enable* MQTT
- Settings:
 - Set 'Cloud service' to 'Sparkplug B'
 - Set the IP address to that of the Ignition Gateway
 - The default is 127.0.0.1; this works only during simulation with Ignition running on the same PC
- Address:
 - Choose a 'Status address'. E.g. LW-100
 - 'Error address' → (Status Addr) +1
 - [OPTIONAL] Set a 'Buffer usage address'. E.g. LW-200
 - Choose a 'Control address'. E.g. LW-110
 - Command → (Ctrl Addr) +0
 - e.g. LW-110
 - Command values:
 - 0: none; 1: start; 2: stop; 3: update
 - IP of Broker (Ignition) → (Ctrl Addr) +1 to +4
 - e.g. LW-111 through LW-114
 - Port → (Control Address) +5
 - [OPTIONAL] Client ID, Authentication → (Ctrl Addr) +16 to +26
 - Username, Password → (Ctrl Addr) +27 to +69
- From the main 'MQTT' window, under 'Sparkplug B' section, 'General' tab:
 - Enter a Group ID as done in the Ignition Gateway previously. E.g. *maple*
 - Enter an Edge ID as done previously in Ignition. E.g. *cmt*
 - Leave at default values: DDATA min. time: 0 ms; QoS: 0 ms
- From the 'Device' tab on the main 'MQTT' window:
 - Create at least one 'Group' of tags
 - E.g. Local HMI > *machineOne*
 - Add at least one 'Tag' to your 'Groups'
 - E.g. Local HMI > *machineOne* > *Speed* [LW-1, 16-bit Unsigned, 1 Element]
 - E.g. Local HMI > *machineOne* > *Temp* [LW-2, 16-bit Unsigned, 1 Element]



- For each of your tags, add the appropriate object type to your project window to enable writing values. For example, for numeric tags, add a Numeric Object. For strings, add an ASCII Entry object. For Booleans, add a Toggle Switch or Bit Lamp to your project.
- In order to be able to start, stop, and restart the connection with the Ignition Gateway, add some or all of the following objects and functions to your project:
 - Status: Create a Word Lamp or Numeric Object to display the current status based on the designated MQTT Status Address
 - Commands – Use either a single Numeric Entry Object for writing to the MQTT Control Address, or:
 - Start: Set Word object writing a '1' into the MQTT Control Address register
 - Stop: Set Word object writing a '2' into the MQTT Control Address register
 - Update: Set Word object writing a '3' into the MQTT Control Address register
 - IP Address of Broker: Add Numeric Objects for each of the 4 octets of the IP Address for the MQTT Broker (e.g. Ignition Gateway). These use the Control Address +1 through +4, respectively.
 - MQTT Port Number: Either 1883 (unencrypted) or 8883 (encrypted with TLS/SSL). Optionally, add a Numeric Entry Object at Control Address +5.
 - Buffer Usage: Create a Numeric Object to display the Buffer utilization (percentage value) by pointing to the designated Buffer Usage Address. This is used to hold messages until the next reconnect if the HMI loses its connection to the Broker for one reason or another.

Example project showing:

- Sparkplug B Tags (blue area/text)
- Command related objects using the Control Address (green area/text)
- IP Address of Ignition Gateway (purple area/text)
- Status of MQTT Connection (red outline)
- Buffer Percentage (orange outline)

MQTT Sparkplug B Sample Project

Machine One	Machine Two
<u>Topic: machineOne/</u>	<u>Topic: machineTwo/</u>
TankLevel (LW-0): <input type="text" value="M1 Level++"/> <input type="text" value="####"/>	TankLevel (LW-10): <input type="text" value="M2 Level++"/> <input type="text" value="####"/>
Speed (LW-1): <input type="text" value="M1 Speed++"/> <input type="text" value="####"/>	Speed (LW-11): <input type="text" value="M2 Speed++"/> <input type="text" value="####"/>
Temp (LW-2): <input type="text" value="M1 Temp++"/> <input type="text" value="####"/>	Temp (LW-12): <input type="text" value="M2 Temp++"/> <input type="text" value="####"/>
<u>Topic: machineOne/label/</u>	<u>Topic: machineTwo/label/</u>
LabelOne (LW-20): <input type="text" value="AAAAAAAAAAAAAAAAAAAA"/>	LabelOne (LW-40): <input type="text" value="AAAAAAAAAAAAAAAAAAAA"/>
LabelTwo (LW-30): <input type="text" value="UNICODE_UNICODE_UNIC"/>	LabelTwo (LW-50): <input type="text" value="UNICODE_UNICODE_UNIC"/>

COMMAND (CONTROL REGISTER)

MQTT Control:

IP ADDRESS

Status:

Buffer (%):

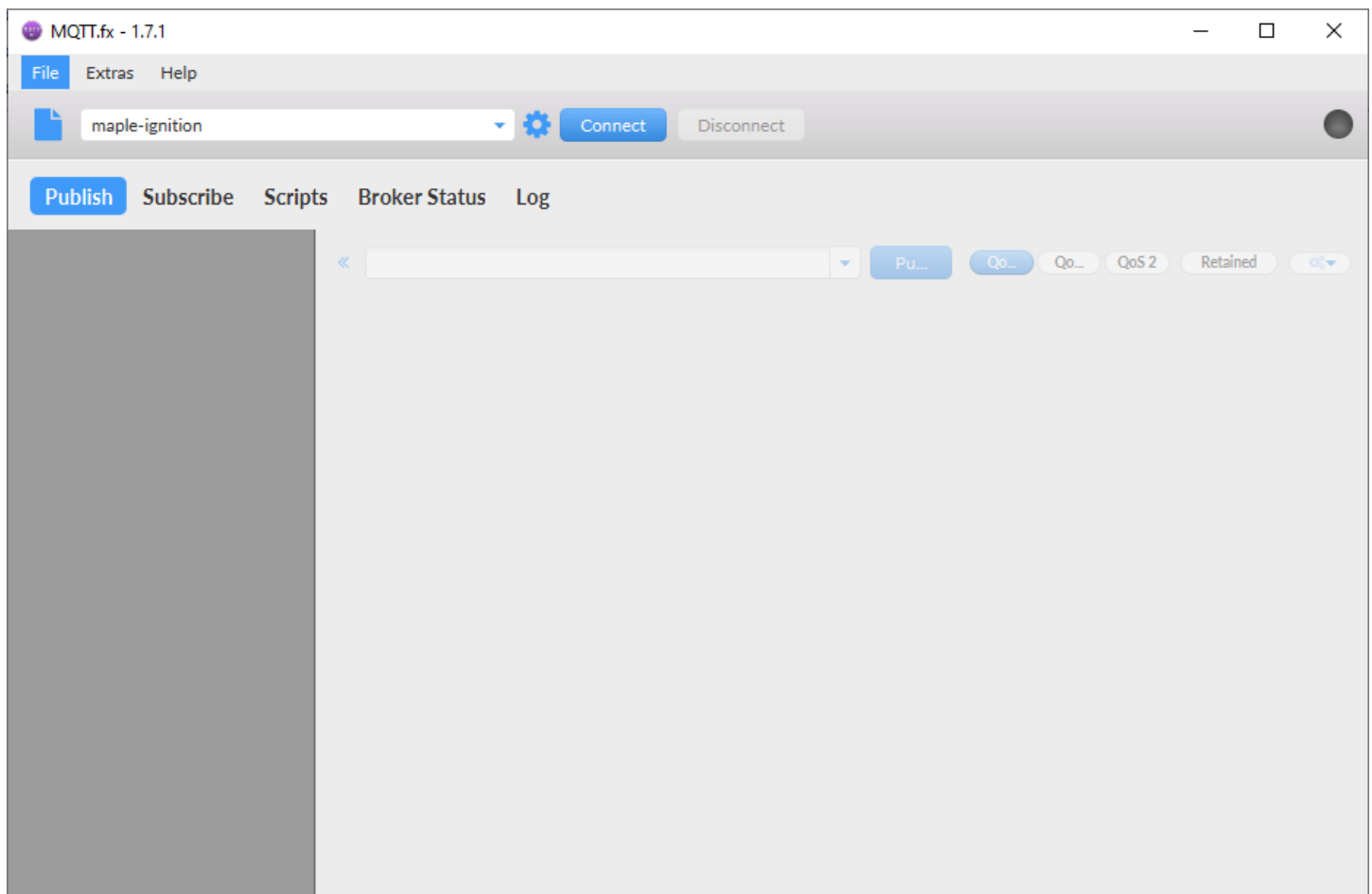
6. Install and Configure MQTT.fx

Download MQTT.fx from the link provided below. This software is used to verify the connection to the Ignition Gateway and inspect the payloads (messages) generated and published by Maple Systems cMT Devices and the Ignition Gateway itself.

MQTT.fx – MQTT Client Software		
Program	Version	Download Link
MQTT.fx	1.7.1 or greater	https://mqttfx.jensd.de Click download and select: <i>mqttfx-1.7.{x}-windows-x64.exe</i>

- Double-click to open and start the MQTT.fx installer once the download has completed
- Follow the instructions and install using the default options
- Once installation is complete, click Finish/OK and run the program

Once you open MQTT.fx, you will see a window as in that shown below:



- Click on the gear icon to configure a new connection
- Enter a name in the field provided for 'Profile Name'. E.g. 'maple-ignition'

- Next, enter the IP address of the Ignition Gateway in the 'Broker Address' field

Profile Name

Profile Type

MQTT Broker Profile Settings

Broker Address

Broker Port

Client ID

General User Credentials SSL/TLS Proxy LWT

Connection Timeout

Keep Alive Interval

Clean Session

Auto Reconnect

Max Inflight

MQTT Version Use Default

- From the 'User Credentials' tab, enter the 'User Name' and 'Password' configured in Ignition Gateway

Profile Name

Profile Type

MQTT Broker Profile Settings

Broker Address

Broker Port

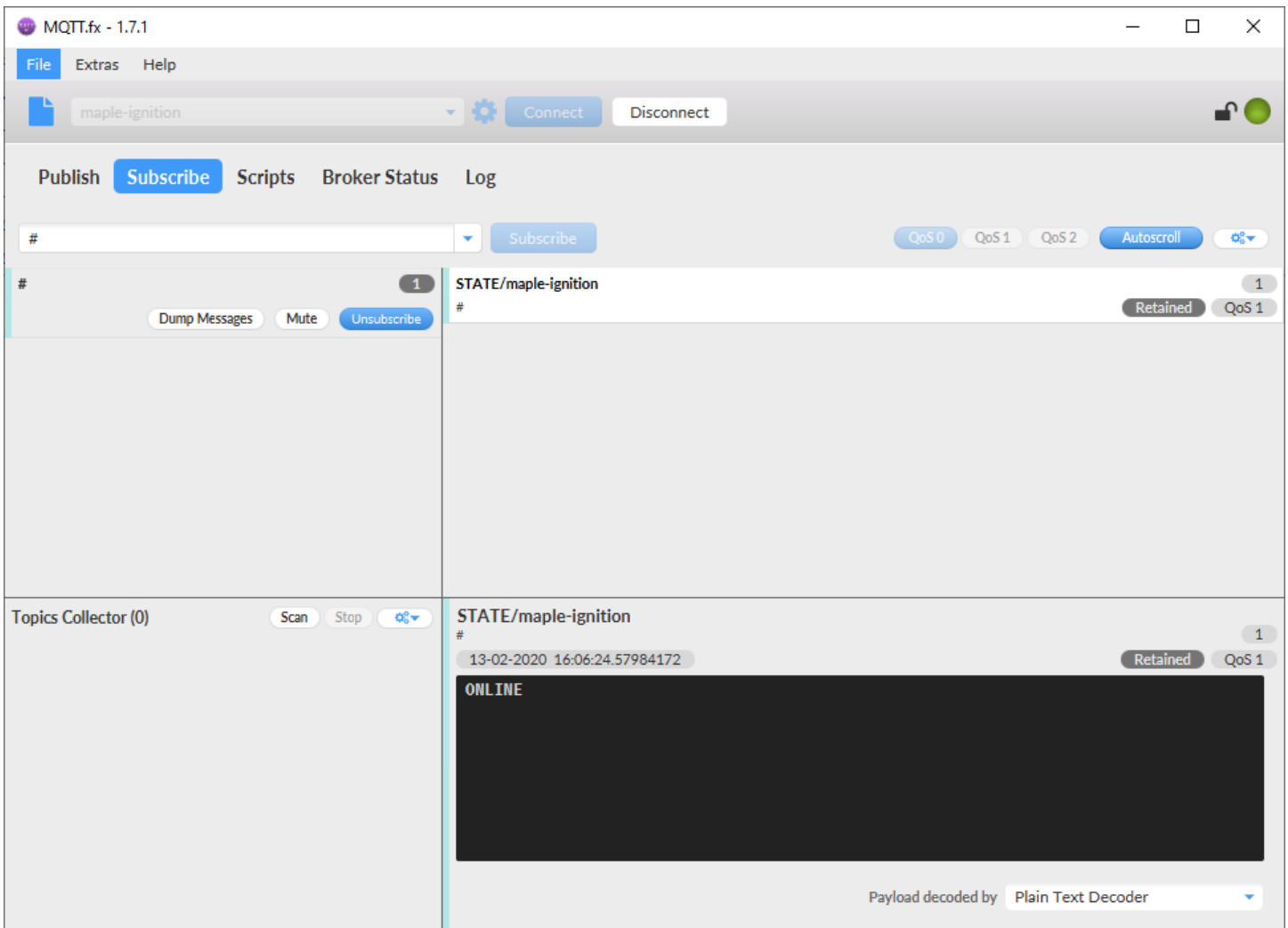
Client ID

General **User Credentials** SSL/TLS Proxy LWT

User Name

Password

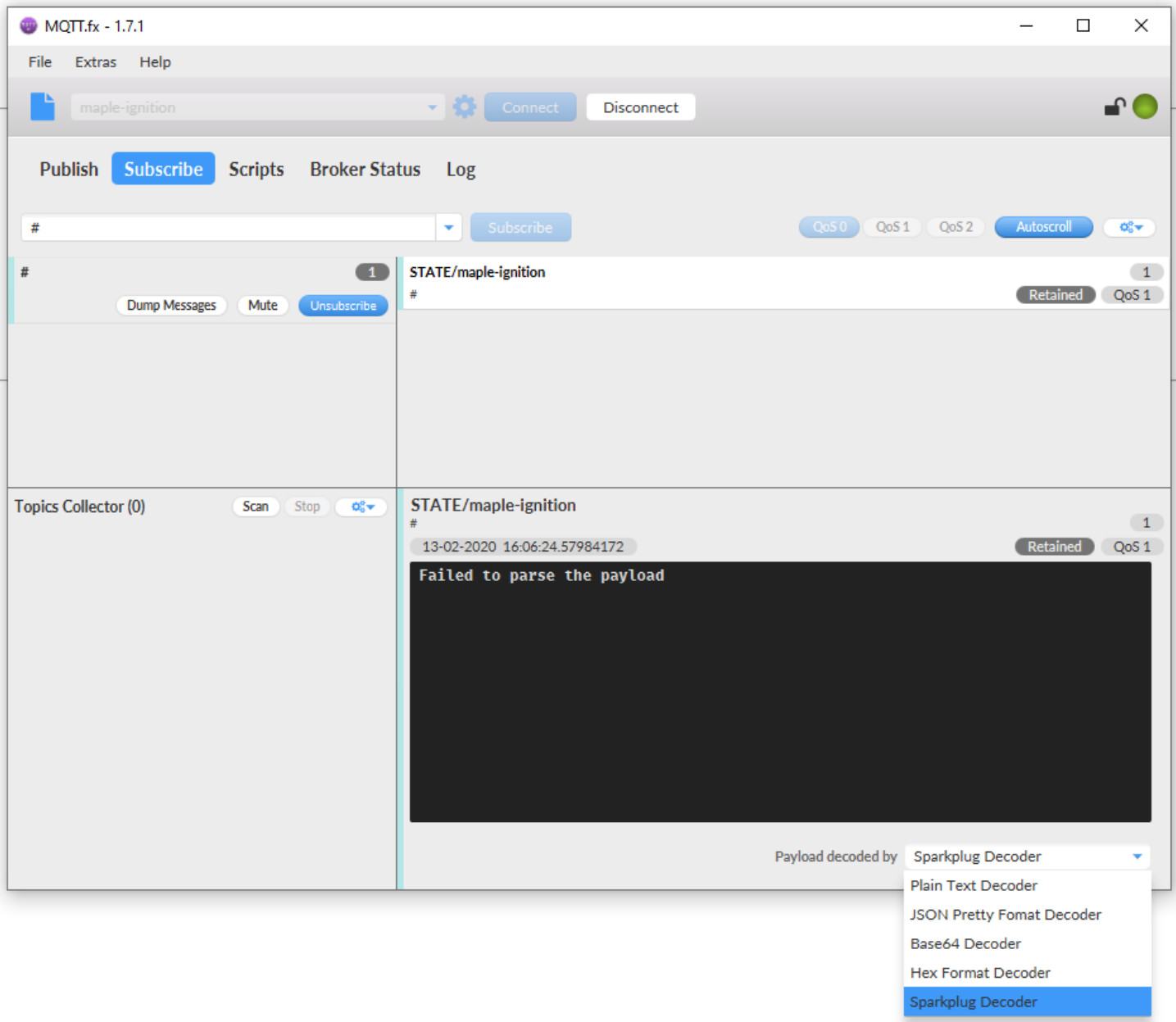
- Click OK when you are done to save the new connection profile
- Next, select the new connection profile and click the 'Connect' button
- Click on the 'Subscribe' tab after you've connected (icon turns green to indicate successful connection)
- Enter a '#' into the 'Subscribe' text field and click the 'Subscribe' button
 - If successfully connected, you should see an STATE: 'ONLINE' message from Ignition displayed



Your MQTT.fx client is now connected to the Ignition Gateway.

Before proceeding to test the EBPro project simulation, be sure to switch from 'Plain Text' to the 'Sparkplug Decoder':

- From the lower-right hand corner of the window, locate the 'Payload decoded by' drop-down menu
- Select 'Sparkplug Decoder'
- NOTE:
 - The original 'STATE: ONLINE' message from Ignition will now say "Failed to parse the payload".
 - This is expected behavior.
 - Messages published from the Maple cMT Device in subsequent steps must be decoded using the Sparkplug Decoder.



Your MQTT.fx client is now ready to decode Sparkplug B MQTT payloads.

7. Test Local Connection using EBPro Simulation Mode and MQTT.fx

Now that you have configured Ignition, your EBPro project, and MQTT.fx, you can proceed to test the connection and send values back and forth using MQTT.

From the 'Project' tab in EBPro, launch a Simulation (click either 'Online Simulation' or 'Offline Simulation').

Assuming you are running the simulation on the same PC hosting the Ignition Gateway, then you should see the simulated HMI connect as shown in this sample project screenshot:

The screenshot shows the cMT Viewer (Simulation) window titled "MQTT Sparkplug B Sample Project" with a timestamp of "2020-02-18 08:56:44". The interface is split into two panels: "Machine One" and "Machine Two".

Machine One:

- Topic: machineOne/
- TankLevel (LW-0): M1 Level++ button, 0000 display
- Speed (LW-1): M1 Speed++ button, 0000 display
- Temp (LW-2): M1 Temp++ button, 0000 display
- Topic: machineOne/label/
- LabelOne (LW-20): empty text box
- LabelTwo (LW-30): empty text box

Machine Two:

- Topic: machineTwo/
- TankLevel (LW-10): M2 Level++ button, 0000 display
- Speed (LW-11): M2 Speed++ button, 0000 display
- Temp (LW-12): M2 Temp++ button, 0000 display
- Topic: machineTwo/label/
- LabelOne (LW-40): empty text box
- LabelTwo (LW-50): empty text box

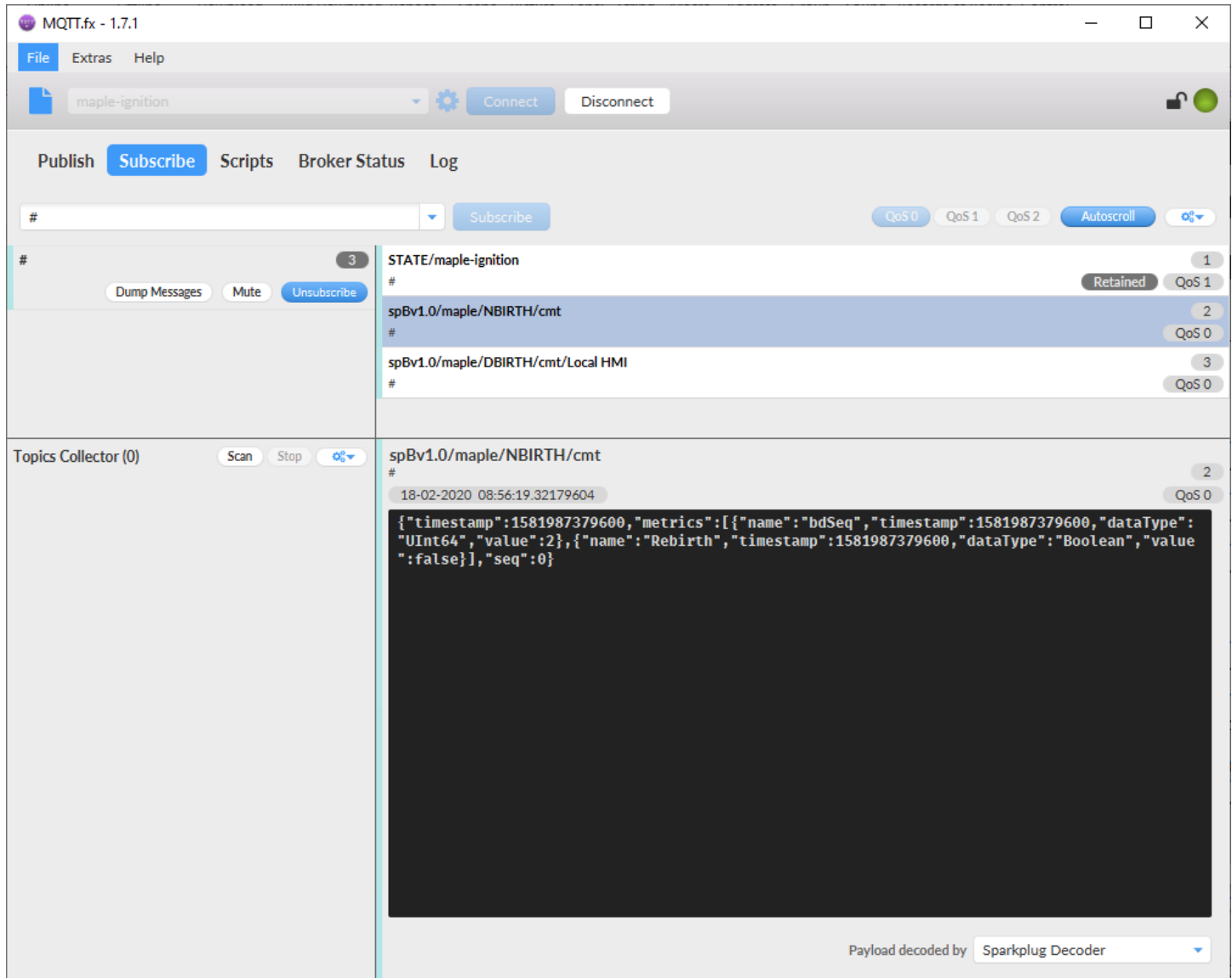
MQTT Control:

- 0 (display)
- START (green button)
- STOP (red button)
- RESTART (blue button)
- LOCAL (grey button)
- 127 (display)
- 0 (display)
- 0 (display)
- 1 (display)
- SPB (green button)
- 192.168.100.3 (IP address)
- Buffer (%): 000 (display)
- Status: 02 - Connected (green bar)

While still connected to the Ignition Gateway using MQTT.fx, you should see a few messages published as soon as the simulated HMI connects to Ignition:

The first is a Node Birth (NBIRTH) certificate:

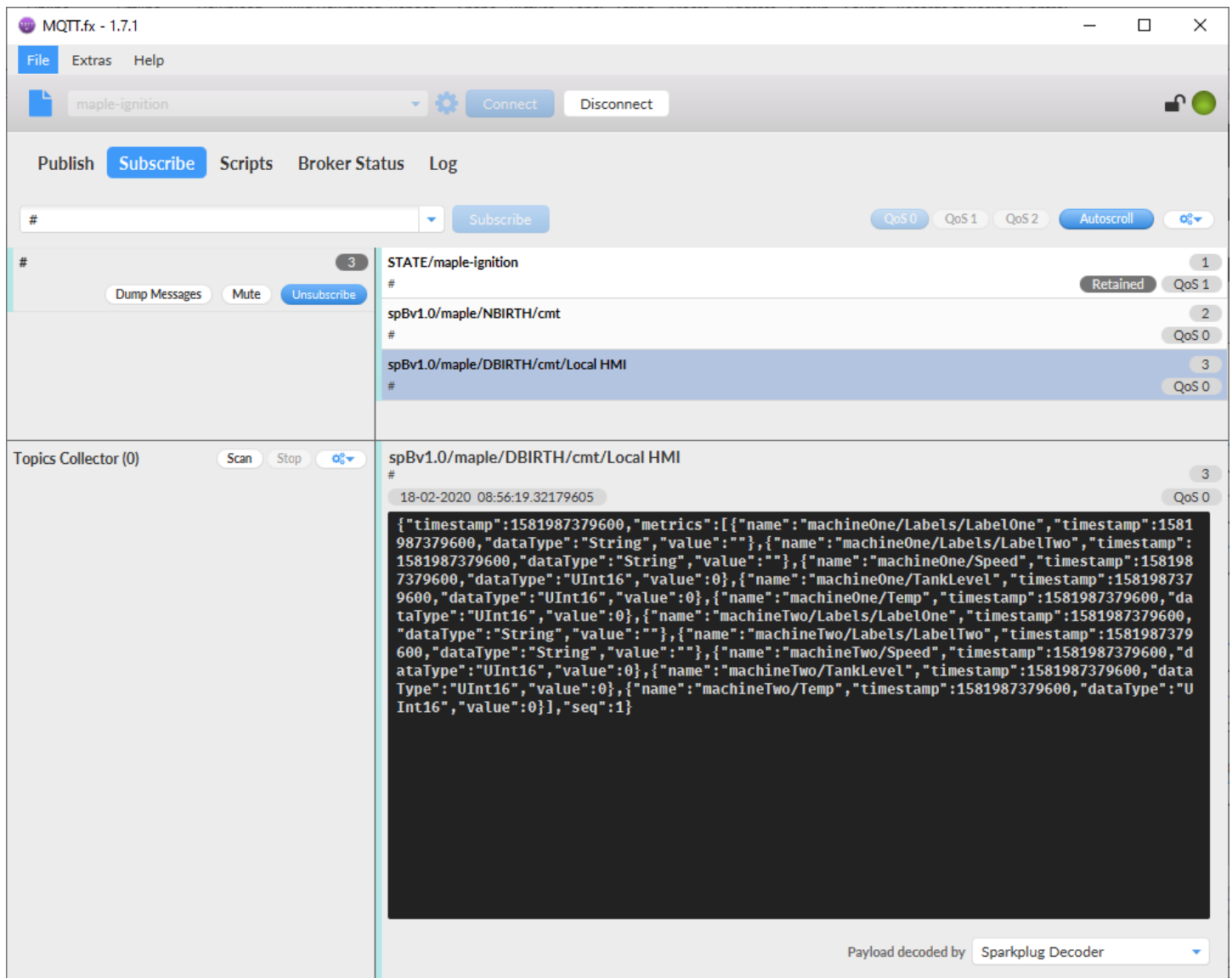
```
{"timestamp":1581987379600,"metrics":[{"name":"bdSeq","timestamp":1581987379600,"dataType":"UInt64","value":2},{name":"Rebirth","timestamp":1581987379600,"dataType":"Boolean","value":false}], "seq":0}
```



The second is a Device Birth (DBIRTH) certificate, containing all the most current values for the tags that have been added to the Sparkplug B configuration in the EBPro project:

```
{
  "timestamp":1581987379600,"metrics":[{"name":"machineOne/Labels/LabelOne","timestamp":1581987379600,"dataType":"String","value":""}, {"name":"machineOne/Labels/LabelTwo","timestamp":1581987379600,"dataType":"String","value":""}, {"name":"machineOne/Speed","timestamp":1581987379600,"dataType":"UInt16","value":0}, {"name":"machineOne/TankLevel","timestamp":1581987379600,"dataType":"UInt16","value":0}, {"name":"machineOne/Temp","timestamp":1581987379600,"dataType":"UInt16","value":0}, {"name":"machineTwo/Labels/LabelOne","timestamp":1581987379600,"dataType":"String","value":""}, {"name":"machineTwo/Labels/LabelTwo","timestamp":1581987379600,"dataType":"String","value":""}, {"name":"machineTwo/Speed","timestamp":1581987379600,"dataType":"UInt16","value":0}, {"name":"machineTwo/TankLevel","timestamp":1581987379600,"dataType":"UInt16","value":0}, {"name":"machineTwo/Temp","timestamp":1581987379600,"dataType":"UInt16","value":0}], "seq":1}

```



If you now enter a string for LabelOne into the accompanying ASCII Entry Object, on the HMI end you will see:

The image shows the MQTT Sparkplug B Sample Project HMI interface. It is divided into two main sections: Machine One and Machine Two. Machine One has three data points: TankLevel (LW-0) with a value of 0000, Speed (LW-1) with a value of 0000, and Temp (LW-2) with a value of 0000. Machine Two has three data points: TankLevel (LW-10) with a value of 0000, Speed (LW-11) with a value of 0000, and Temp (LW-12) with a value of 0000. Below these are LabelOne and LabelTwo fields. LabelOne (LW-20) contains the text 'Maple Systems', and LabelTwo (LW-30) is empty. At the bottom, there is an MQTT Control section with buttons for START, STOP, RESTART, and LOCAL, and a Buffer (%) field showing 000. The status is '02 - Connected' and the IP address is 192.168.100.3.

In MQTT.fx, you should now see the first Device Data (DDATA) message has been published:

```
{"timestamp":1581987826620,"metrics":[{"name":"machineOne/Labels/LabelOne","timestamp":1581987826620,"dataType":"String","value":"Maple Systems"}],"seq":2}
```

The image shows the MQTT.fx interface. The main window displays a list of topics and their QoS values. The selected topic is 'spBv1.0/maple/DDATA/cmt/Local HMI'. The message content is displayed in a text area, showing the JSON payload: {"timestamp":1581987826620,"metrics":[{"name":"machineOne/Labels/LabelOne","timestamp":1581987826620,"dataType":"String","value":"Maple Systems"}],"seq":2}. The interface also shows buttons for Publish, Subscribe, Scripts, Broker Status, and Log.

You may proceed to try entering different values in the ASCII Entry or Numeric Entry Objects.

For example, a Unicode string in EBPro is represented the same way as an ASCII string from Ignition's point of view. If you enter "UNICODE STRING" into the LabelTwo field, you will see:

```
{"timestamp":1581988072485,"metrics":[{"name":"machineOne/Labels/LabelTwo","timestamp":1581988072485,"dataType":"String","value":"UNICODE STRING"}],"seq":3}
```

The screenshot shows the MQTT.fx 1.7.1 application window. The main interface includes a menu bar (File, Extras, Help), a toolbar with 'Connect' and 'Disconnect' buttons, and a central panel for topic management and message viewing. The 'Publish' and 'Subscribe' buttons are visible. The 'Subscribe' field is empty. The 'Topics Collector (0)' panel shows a list of topics, with 'spBv1.0/maple/DDATA/cmt/Local HMI' selected. The message view shows a timestamp of '18-02-2020 09:07:52.32872487' and a JSON payload:

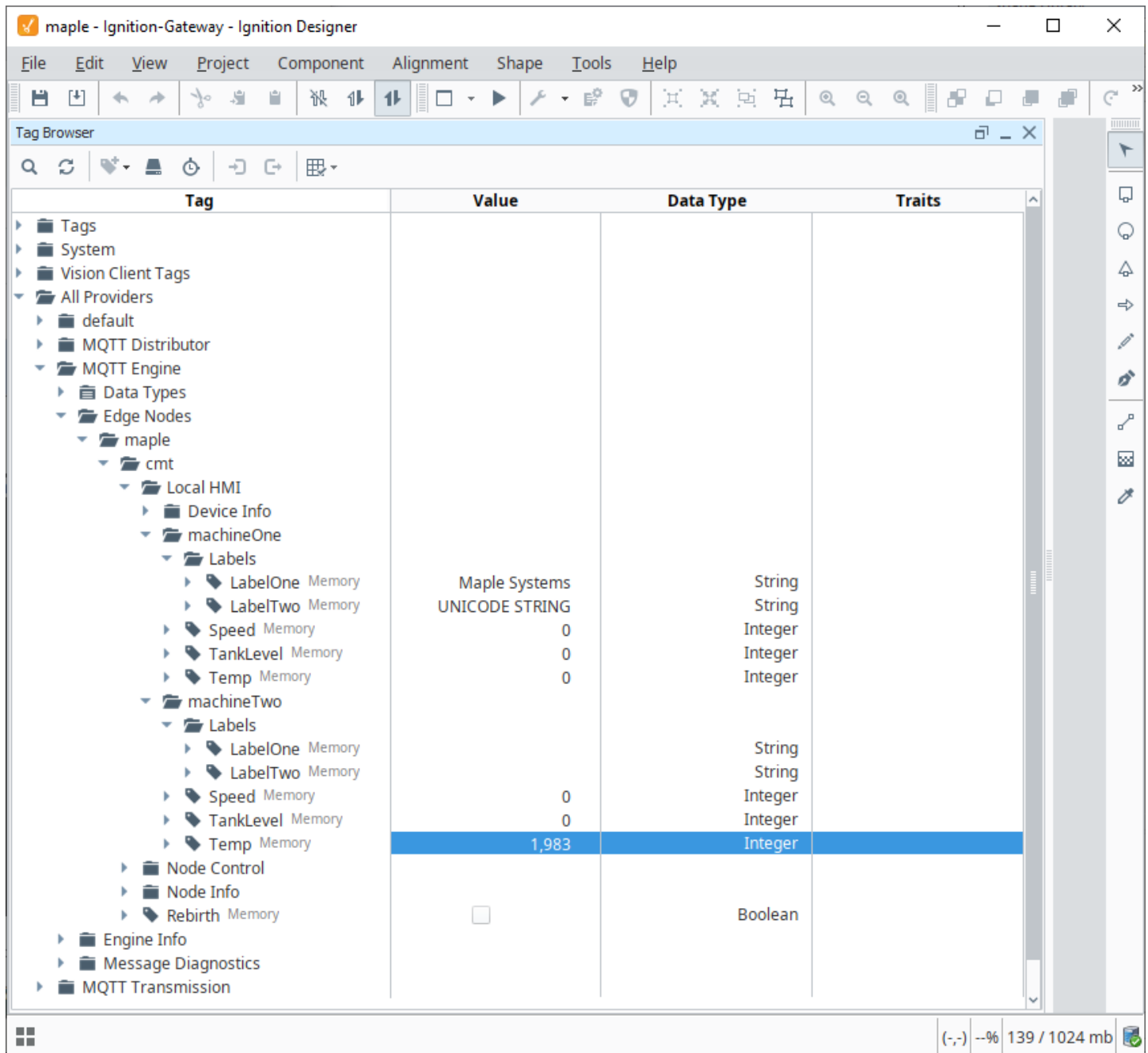
```
{"timestamp":1581988072485,"metrics":[{"name":"machineOne/Labels/LabelTwo","timestamp":1581988072485,"dataType":"String","value":"UNICODE STRING"}],"seq":3}
```

 The payload is decoded by the 'Sparkplug Decoder'.

If you enter a numeric value, such as '1983' into machineTwo > Temp (LW-12), you will see the Sparkplug B payload is formatted as follows:

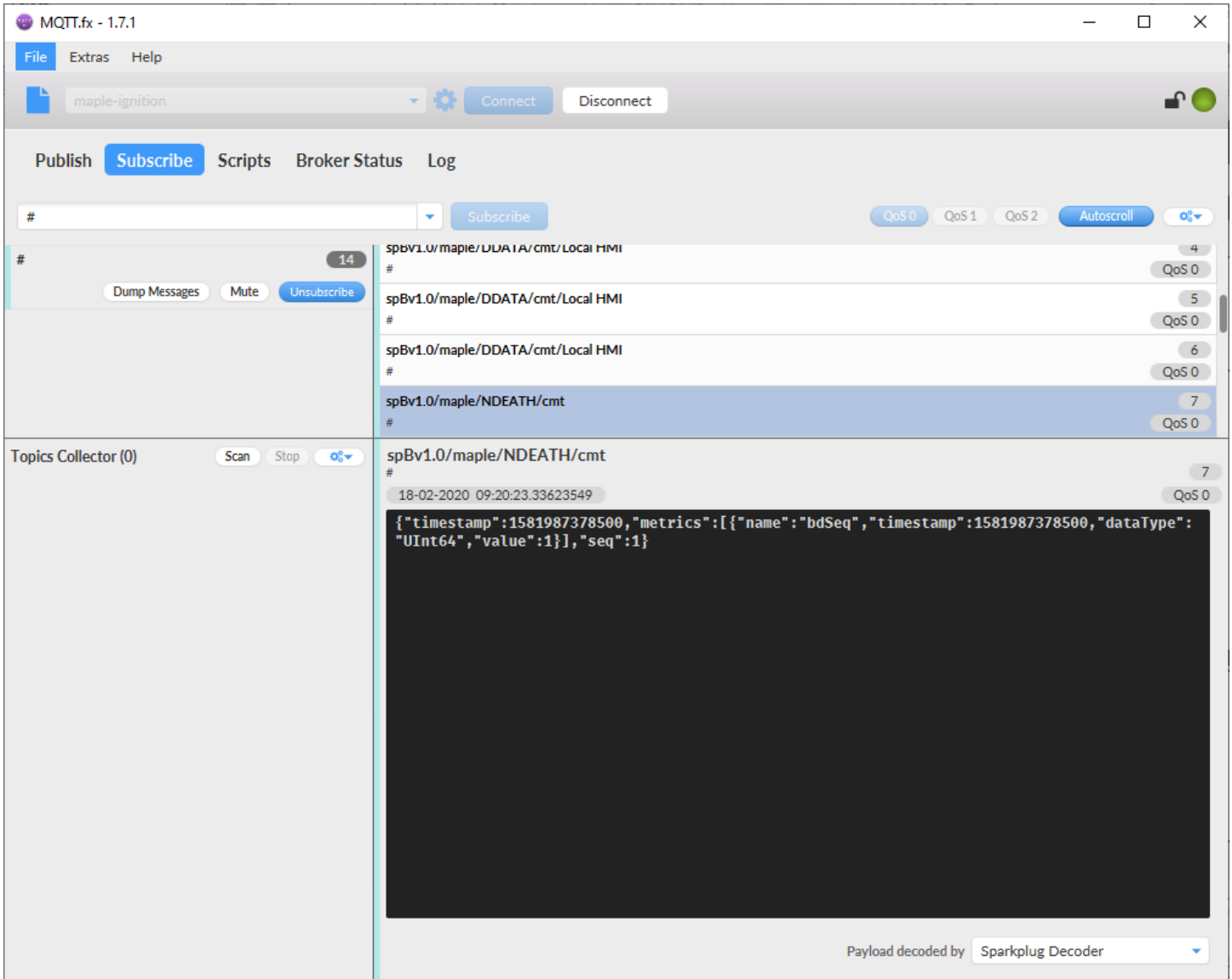
```
{"timestamp":1581988581481,"metrics":[{"name":"machineTwo/Temp","timestamp":1581988581481,"dataType":"UInt16","value":1983}], "seq":4}
```

And in Ignition Designer, you would see:



If you now click on 'Stop' or close the HMI Simulation, you will see the following Node Death (NDEATH) message published:

```
{"timestamp":1581987378500,"metrics":[{"name":"bdSeq","timestamp":1581987378500,"dataType":"UInt64","value":1}], "seq":1}
```



Once you disconnect, within the Ignition Designer project you will see the 'Bad_Stale' flag (red exclamation point icon) next to each of the tags.

As soon as the connection is reestablished, the 'Bad_Stale' flags will disappear, and the tag quality will be marked as 'Good' again.

The screenshot shows the Ignition Designer interface with the Tag Browser window open. The Tag Browser displays a tree view on the left and a list of tags on the right. The tree view shows the following structure:

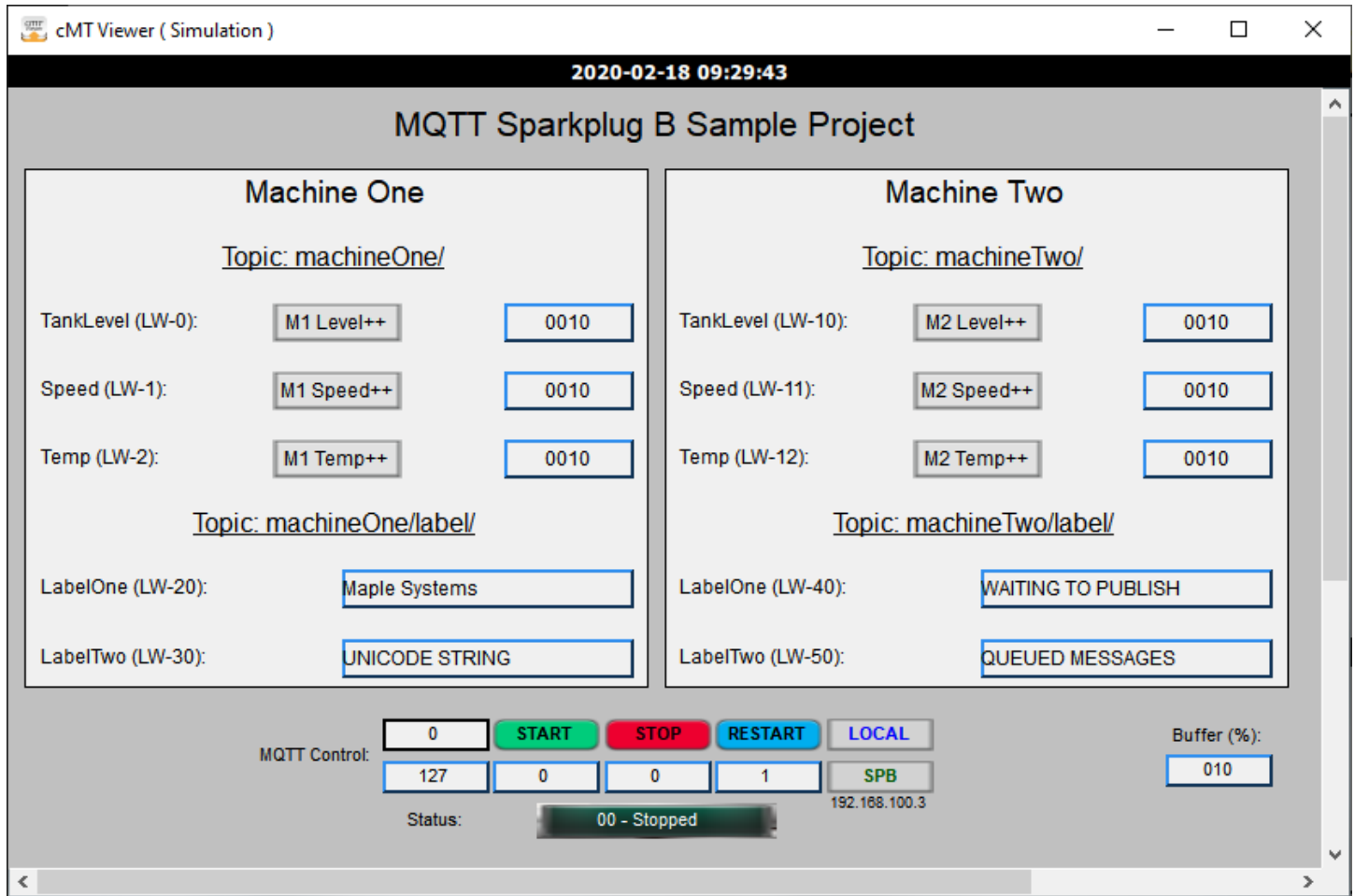
- MQTT Engine
 - Data Types
 - Edge Nodes
 - maple
 - cmt
 - Local HMI
 - Device Info
 - machineOne
 - Labels
 - LabelOne Memory
 - LabelTwo Memory
 - Speed Memory
 - TankLevel Memory
 - Temp Memory
 - machineTwo
 - Labels
 - LabelOne Memory
 - LabelTwo Memory
 - Speed Memory
 - TankLevel Memory
 - Temp Memory
 - MQTT Tags
 - Node Control
 - Node Info
 - Rebirth Memory
 - AlarmEvalEnabled
 - Deadband
 - Documentation
 - EngHigh
 - EngLow
 - EngUnit
 - FormatString
 - HistoryEnabled
 - Quality
 - Timestamp
 - Tooltip
 - value

The tag list on the right shows the following data:

Tag Name	Value	Quality	DataType
Maple Systems		!	String
UNICODE STRING		!	String
0	0	!	Integer
0	0	!	Integer
0	0	!	Integer
LabelOne		!	String
LabelTwo		!	String
Speed	0	!	Integer
TankLevel	0	!	Integer
Temp	1,983	!	Integer
Rebirth	<input type="checkbox"/>	!	Boolean
AlarmEvalEnabled	<input checked="" type="checkbox"/>		Boolean
Deadband	0		Double
Documentation			String
EngHigh	100		Double
EngLow	0		Double
EngUnit			String
FormatString	#,##0.##		String
HistoryEnabled	<input type="checkbox"/>		Boolean
Quality			String
Timestamp	2020-02-21 11:42:10 ...		DateTime
Tooltip			String
value	<input type="checkbox"/>		Boolean

When you reconnect, you will see a new Node Birth (NBIRTH) and Device Birth (DBIRTH) message are published.

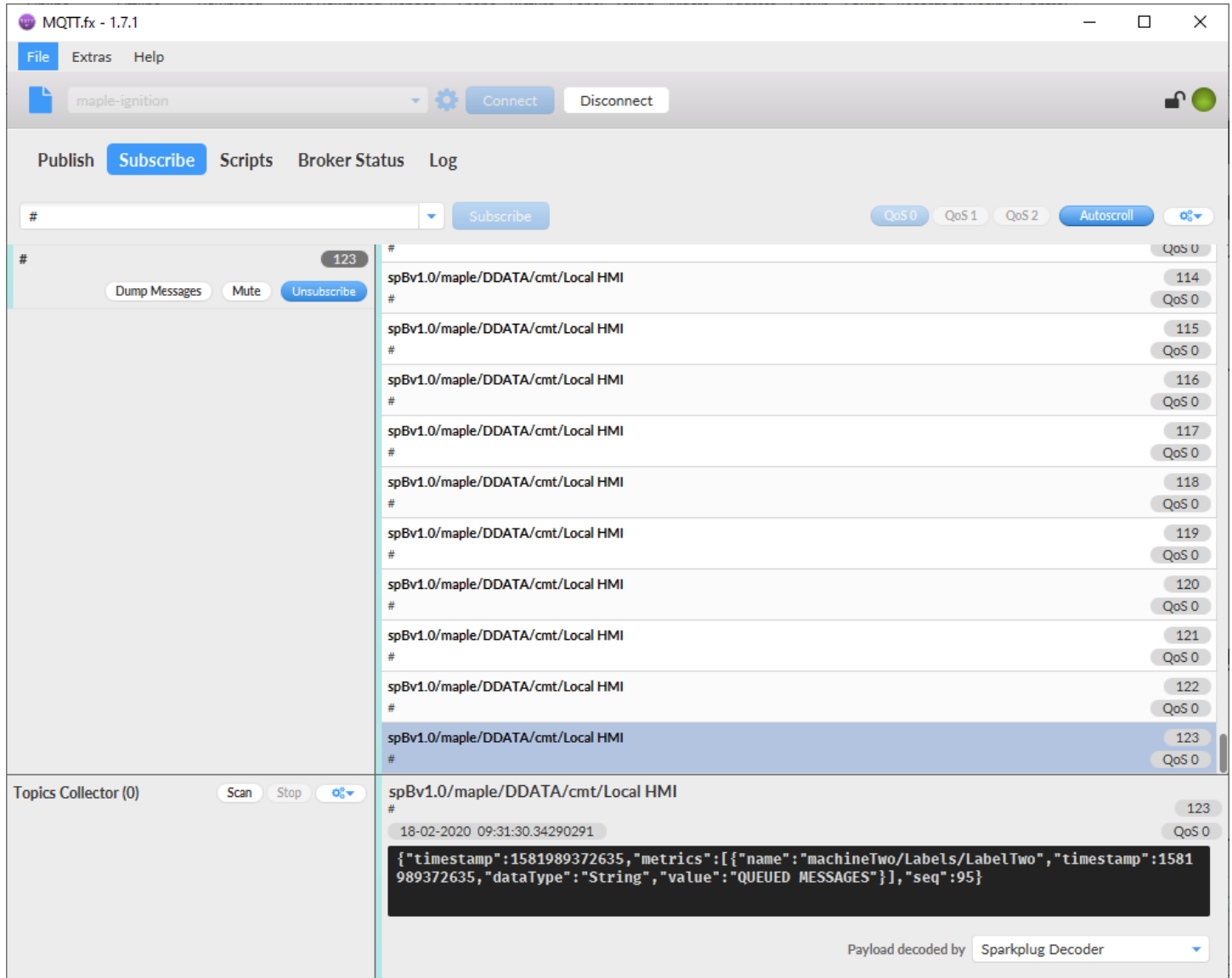
In the meantime, if you have opted to enable Buffering, any values that haven't been published to Ignition while the Simulated HMI is disconnected are held in HMI memory. The Buffering percentage indicator reflects the amount of such messages waiting to be published when the connection is reestablished:



In the above screenshot, you can see the buffer is filled to 10% of capacity. This was achieved in Simulation mode by writing a large number of different values to each of the tags while in a 'Stopped' or disconnected state.

Once you click on 'Restart', the messages in the Buffer will be published and the Buffer percentage will go back to zero.

Upon reconnection, in MQTT.fx you will see the new NBIRTH and DBIRTH messages published, followed by all the DDATA messages that had been queued up while disconnected.



You have now configured your EBPro project to connect in Simulation Mode to the Ignition Gateway.

Next, you may perform a Live Test, downloading the EBPro project to a cMT Device.

8. Perform Live Test from cMT Device

[OPTIONAL] Open Network Firewall Ports on PC running Ignition Gateway (MQTT Ports)

Prior to downloading your EBPro project to a cMT device, it is a good idea to verify that incoming MQTT messages are not blocked by a Windows or Network-based firewall.

One tool that can be used to check for open ports on the host PC is Microsoft's 'PortQry' Command Line Port Scanner.

You may download a free copy of *PortQry* from the following link:

<https://www.microsoft.com/en-us/download/details.aspx?id=17148>

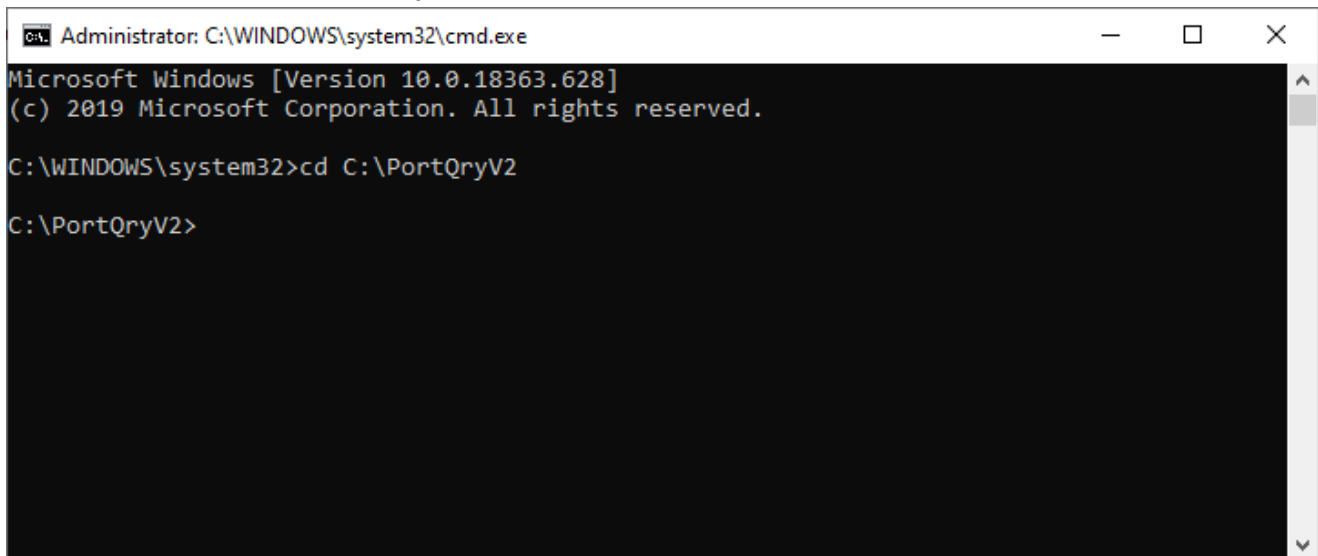
- Once you have downloaded *PortQry*, double-click to open the installer titled "PortQryV2.exe"
- Review and accept the terms of the End User License Agreement, and then proceed to install *PortQry*
- The default installation directory is C:\PortQryV2. Take note of where you choose to install it.

Open your command prompt as an Administrator:

1. Hit CTRL + R
2. Type 'cmd'
3. Hit 'CTRL + SHIFT + ENTER' to launch 'cmd' as an Administrator

Navigate to the location of the *PortQry* executable:

1. Type 'cd <PATH\TO\PortQryV2>' and hit 'ENTER'

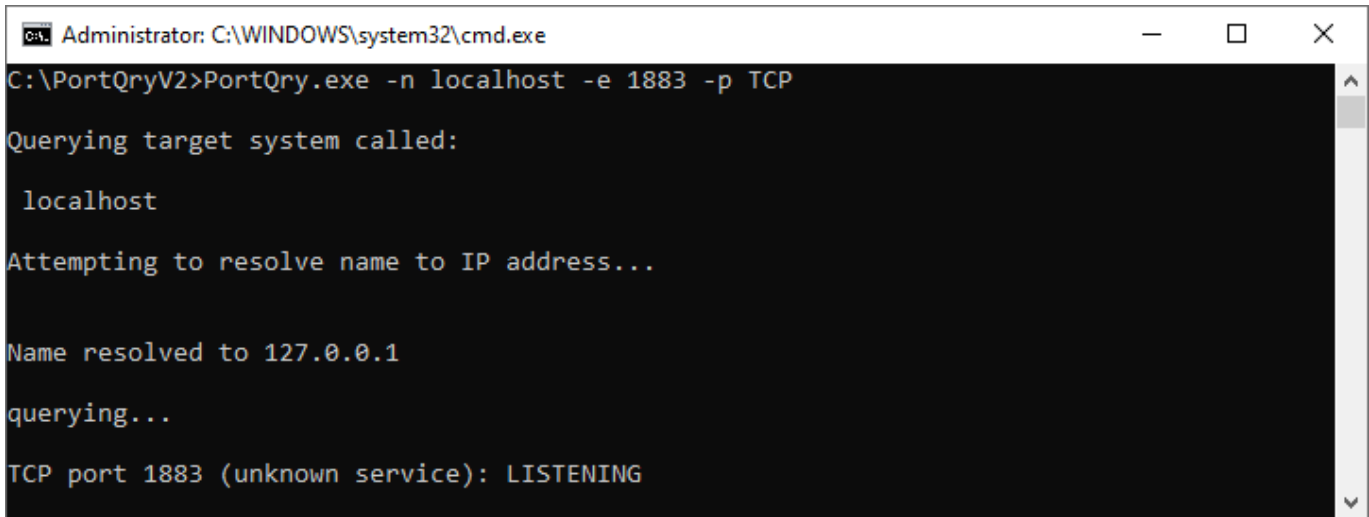


```
Administrator: C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.18363.628]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\PortQryV2

C:\PortQryV2>
```


2. Run the following command to check if the Unencrypted MQTT Port (1883) is open (i.e. 'Listening'):
 - a. PortQry.exe -n localhost -e 1883 -p TCP



```
Administrator: C:\WINDOWS\system32\cmd.exe
C:\PortQryV2>PortQry.exe -n localhost -e 1883 -p TCP
Querying target system called:
localhost
Attempting to resolve name to IP address...
Name resolved to 127.0.0.1
querying...
TCP port 1883 (unknown service): LISTENING
```

If the port is not currently set to 'Listening' (Open), then follow the steps below to create a rule for Windows Firewall:

Syntax for adding a Windows Firewall rule (as Administrator from CMD):

- netsh advfirewall firewall add rule name="{name}" dir=[in/out] action=allow protocol=TCP localport={####}

Example Commands for Opening MQTT Port 1883:

- netsh advfirewall firewall add rule name="MQTT TCP 1883 IN" dir=in action=allow protocol=TCP localport=1883
- netsh advfirewall firewall add rule name="MQTT TCP 1883 OUT" dir=out action=allow protocol=TCP localport=1883

If you plan to use SSL/TLS to encrypt your MQTT traffic, then please add the following additional firewall rules:

- netsh advfirewall firewall add rule name="MQTT TCP 8883 IN" dir=in action=allow protocol=TCP localport=8883
- netsh advfirewall firewall add rule name="MQTT TCP 8883 OUT" dir=out action=allow protocol=TCP localport=8883

Run *PortQry* again to confirm both ports are now open to incoming connections:

- PortQry.exe -n localhost -e 1883 -p TCP
- PortQry.exe -n localhost -e 8883 -p TCP

If you get inconsistent results on the local PC, try running PortQry from another PC and pointing it at the SCADA host system. Alternatively, try the 'netstat' command to check port status, from the Admin command prompt:

- netstat -ano | findstr 883

(This will show the status of both port 1883 and port 8883.)

Your PC can now accept incoming MQTT connections. Proceed to download the EBPro project to your cMT Device.

9. Connecting to Ignition from a cMT Device

Whether you are using the provided Sample Project, or a project you have created yourself in EBPro, it is important to note that **Ignition expects each device to be given a different Edge Node ID**.

MQTT

Enable

Server

Settings... IP: 127.0.0.1, Port: 1883

Sparkplug B

General Device

Group ID : maple

Edge node ID : cmt

DDATA min. time : 0 ms

* Minimal waiting time before sending a new DDATA (Device DATA) message (if data changes are detected)

QoS : 0

* Supported OS version : 20150923 or later.

Exit

*Always set a different **Edge node ID** for each device, simulated or otherwise.*

For each cMT Device or Simulated cMT Device that has identically named tags, if you do not use a different 'Edge node ID', then Ignition will not know how to interpret the tag data correctly, and tag values may be overwritten due to a race condition.

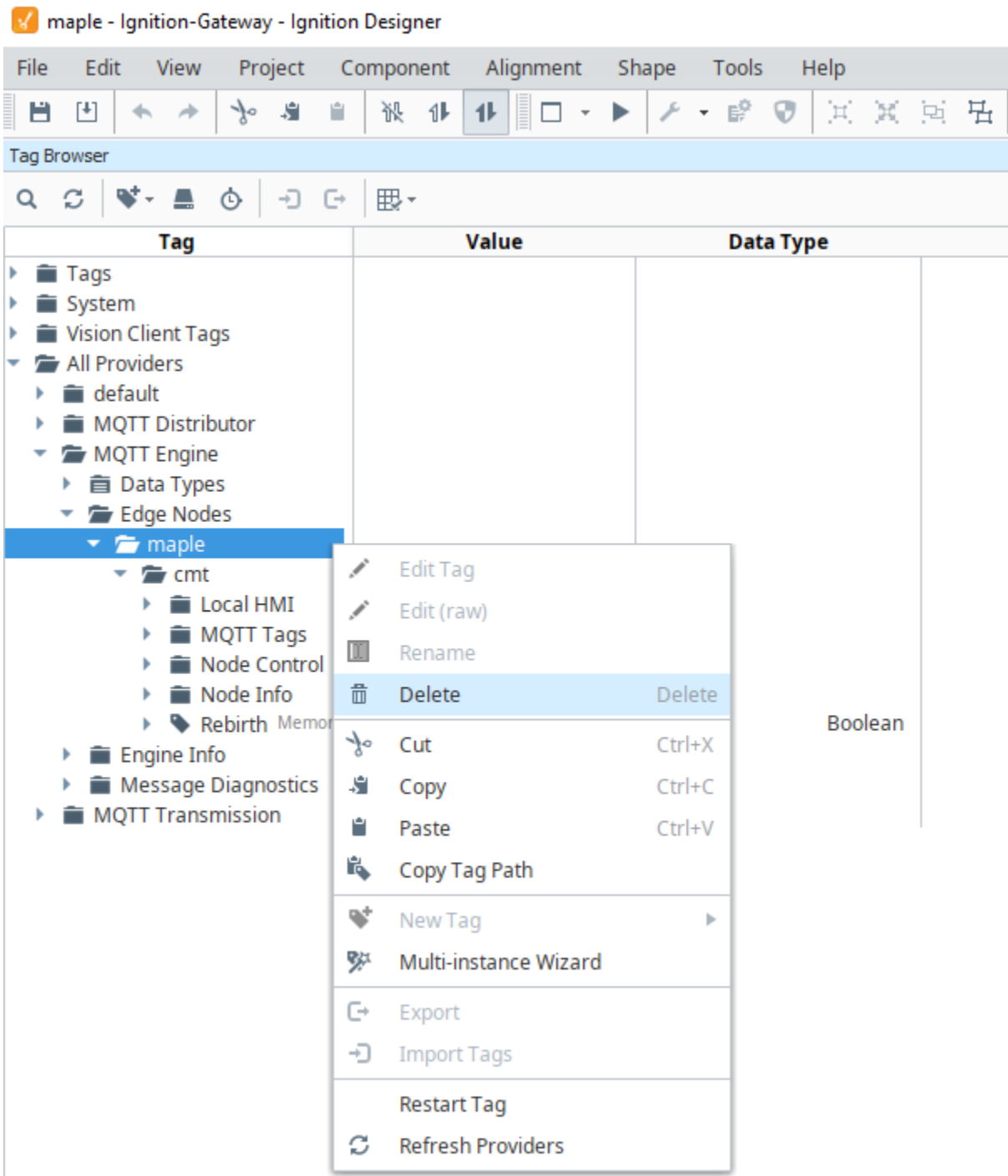
Example of this issue:

From the Ignition Gateway > Status > Logs, you may see errors such as "SparkplugBPayloadHandler cmt Message Sequence number ERROR: 1 :: 3" that indicate there is a conflict relating to duplicate Edge Node ID and/or Tag Names.

```
E SparkplugBPayloadHandler 21Feb2020 12:26:59 cmt Message Sequence number ERROR: 1 :: 3
```

If you run into this issue, try the following troubleshooting steps:

- Stop all connections to Ignition (disconnect) from each of your cMT Devices (simulated or real)
- Double-check the 'Edge node ID' setting in each of your projects connecting to the Ignition Gateway
 - Make sure that all Edge Node IDs are unique (in the case that Tag Names are shared)
- Delete the associated Tag Folder(s) listed under 'Edge Nodes' within your Ignition Designer project
 - Upon reconnection from each cMT Device, the tags will automatically be discovered or rediscovered by Ignition



You may now proceed to connect as many cMT Devices as needed to Ignition Gateway.

10. Enable Encrypted Connections using SSL/TLS

For a production environment, it is highly recommended to encrypt all data sent between Edge Nodes (cMT Devices) and the MQTT Broker (Ignition Gateway) using SSL/TLS certificates.

- The default for MQTT is to send all data in unencrypted payloads over TCP port 1883.
- Encrypted MQTT payloads rely on SSL certificates for security and are transmitted over TCP port 8883 instead.

Maple Systems has published a free Technical Note describing the SSL certificate configuration process. The document is titled:

- Technical Note: **Secure MQTT Connections between Maple Systems HMIs and Ignition Gateway**

Please visit our Support Center > [Technical Notes](#) to download and review this documentation.

11. Appendix

EBPro MQTT Server Object – Status Codes:

- 0: Stopped
- 1: Disconnected
- 2: Connected

EBPro MQTT Server Object – Error Codes:

- 0: Success
- 1: Unknown Error
- 2: Failed to Connect
- 3: Access Denied
- 4: Designated MQTT Port is Blocked or Unavailable
- 5: Domain Name Resolution Error
- 6: Buffer Overflow
- 32: Incorrect Client ID
- 48: Failed to Verify SSL Certificate
- 256: Still Connecting

Tutorial Videos

Visit our YouTube channel [here](#) to watch our Sparkplug B MQTT Quick-Start Video Series.

Additional Resources

- Sparkplug B MQTT Sample Project:
<https://www.maplesystems.com/SupportCenter/SampleProjects.htm>
- Technical Note: “Secure MQTT Connections between Maple Systems HMIs and Ignition Gateway”:
<https://www.maplesystems.com/SupportCenter/TechnicalNotes.htm>
- Learn more about Sparkplug B:
<https://sparkplug.eclipse.org/>
- Read the Sparkplug B Specification:
<https://www.eclipse.org/tahu/spec/Sparkplug%20Topic%20Namespace%20and%20State%20ManagementV2.2-with%20appendix%20B%20format%20-%20Eclipse.pdf>

Your Industrial Control Solutions Source
MAPLESYSTEMS.COM



Maple Systems, Inc. | 808 134th St. SW, Suite 120, Everett, WA 98204 | 425.745.3229