

## TECHNICAL NOTE

**Model**

Graphic HMIs

**Title**

MQTT Set-up

TN5111

P/N: 0907-5111

Rev. 02 Date: 11/09/2022



### Summary

This tech note walks through the steps necessary to configure an HMI as an edge gateway for an MQTT network. There are 4 main tasks that will be addressed in this tech note:

- An EBPro project will be created that enables the HMI to act as an MQTT publisher
- HMI memory addresses will be mapped to MQTT topics
- The project will be run in the EBPro Online Simulation mode on a PC and publish data to a public test broker
- MQTT Explorer will be used to subscribe to topics and receive data from the HMI application

### What You Will Need

---

- The most recent version of EBPro:  
<https://www.maplesystems.com/supportcenter/softwaredownloads.htm>
- Basic knowledge of using EBPro
- MQTT client software. This guide will use MQTT Explorer, which is available at  
<http://www.mqtt-explorer.com>
- A windows PC

An MQTT sample project can be downloaded from:

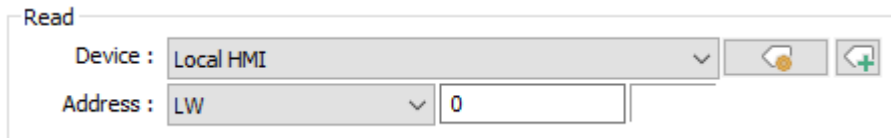
<https://www.maplesystems.com/supportcenter/sampleprojects.htm>

## Step 1 - Project Set-Up

---

This step will create an HMI screen with some numeric inputs to use later:

- Start EBPro and create a new project. Any model number can be selected.
- Select **Object...Numeric** from the menu bar.
- Under **Read address** select Local HMI for the PLC
- Select LW from the address type dropdown list, and enter **0** in the address field (this is the default).



- Click **OK** to place this object on the screen
- Make five copies of this numeric object and configure them with the following addresses:

Address
LW-0 (already done)
LW-1
LW-2
LW-10
LW-11
LW-12

In Step 2 the addresses used above are going to be assigned to MQTT topics. You will add some labels to the screen and organize the inputs under the topic they will be assigned to so that it is easy to tell which address is assigned to which topic:

- Create five text objects (Under **Draw...Text** from the menu bar) to create the following labels:

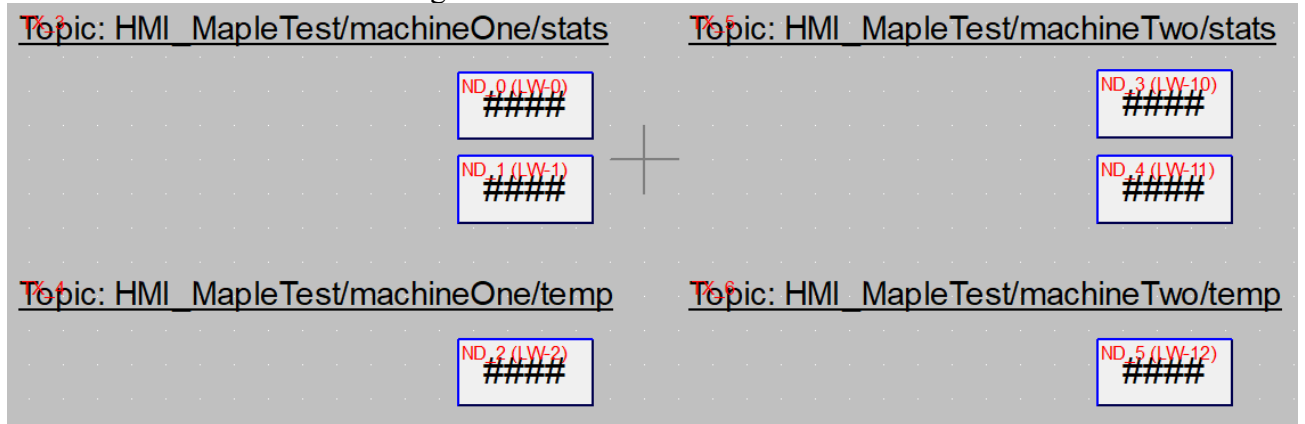
Label
Topic: <name>/machineOne/stats
Topic: <name>/machineOne/temp
Topic: <name>/machineTwo/stats
Topic: <name>/machineTwo/temp

Note: replace <name> with a unique name of your own choosing to use for all topics. Our sample uses **HMI\_MapleTest**.

- Arrange the labels on the screen so that the numeric inputs are placed under the label for the topic they will be assigned to as follows:

Label	Numeric Input Address
Topic: <name>/machineOne/stats	LW-0
	LW-1
Topic: <name>/machineOne/temp	LW-2
Topic: <name>/machineTwo/stats	LW-10
	LW-11
Topic: <name>/machineTwo/temp	LW-12

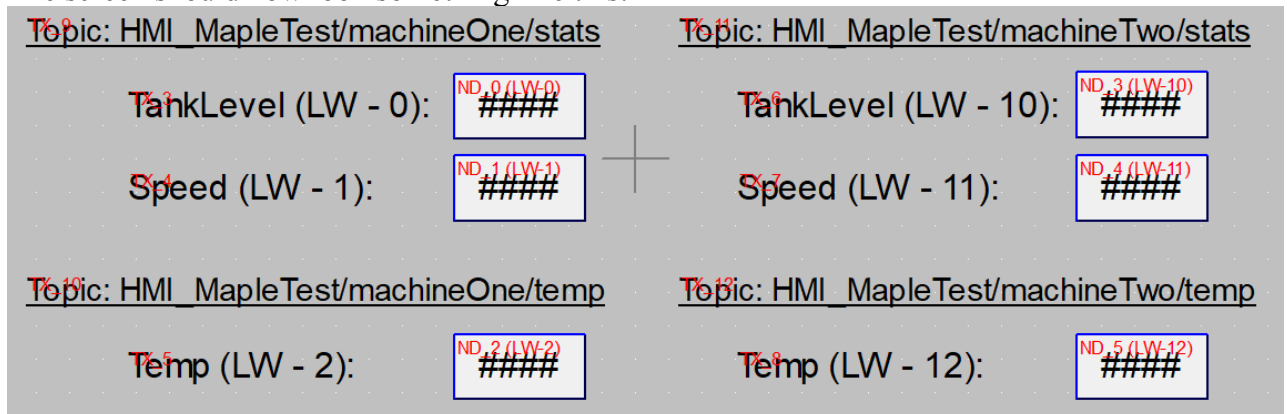
The screen should look something like this:



- Now add the following labels to the left of each numeric object so that it is clear what the number represents:

Numeric Input Address	Label
LW-0	TankLevel (LW - 0):
LW-1	Speed (LW - 1):
LW-2	Temp (LW - 2):
LW-10	TankLevel (LW - 10):
LW-11	Speed (LW - 11):
LW-12	Temp (LW - 12):

The screen should now look something like this:



- That completes the screen set up. The screen shot below shows some more optional labels and label formatting that can be added to better organize the screen:

Connecting to Mosquitto Test Broker

# MQTT Sample Project

## Simulated Data - Published

### Machine One

Topic: HMI\_MapleTest/machineOne/stats

TankLevel (LW - 0): ND\_0 (LW-0) #####

Speed (LW - 1): ND\_1 (LW-1) #####

Topic: HMI\_MapleTest/machineOne/temp

Temp (LW - 2): ND\_2 (LW-2) #####

Status: WL\_0 (LW-100) 00 - Stopped

+

### Machine Two

Topic: HMI\_MapleTest/machineTwo/stats

TankLevel (LW - 10): ND\_3 (LW-10) #####

Speed (LW - 11): ND\_4 (LW-11) #####

Topic: HMI\_MapleTest/machineTwo/temp

Temp (LW - 12): ND\_5 (LW-12) #####

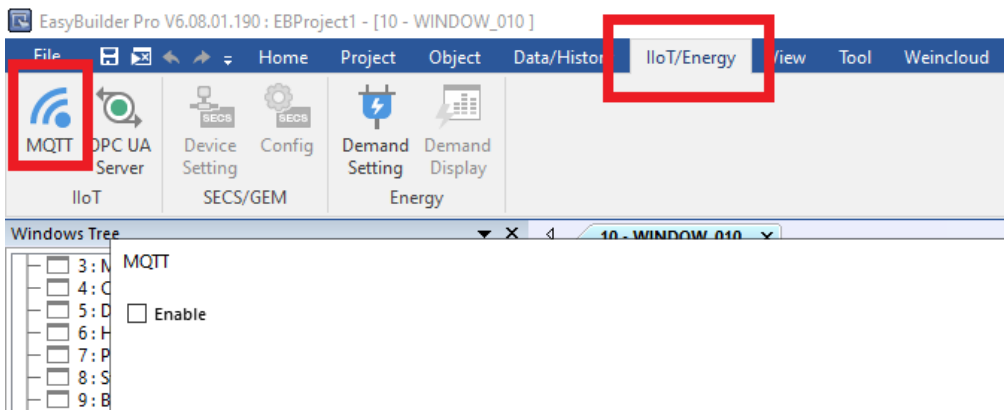
FK\_0 Subscribe

## Step 2 - MQTT Object Set-Up

Next you will add an MQTT object to the project, configure the broker settings and configure the topics.

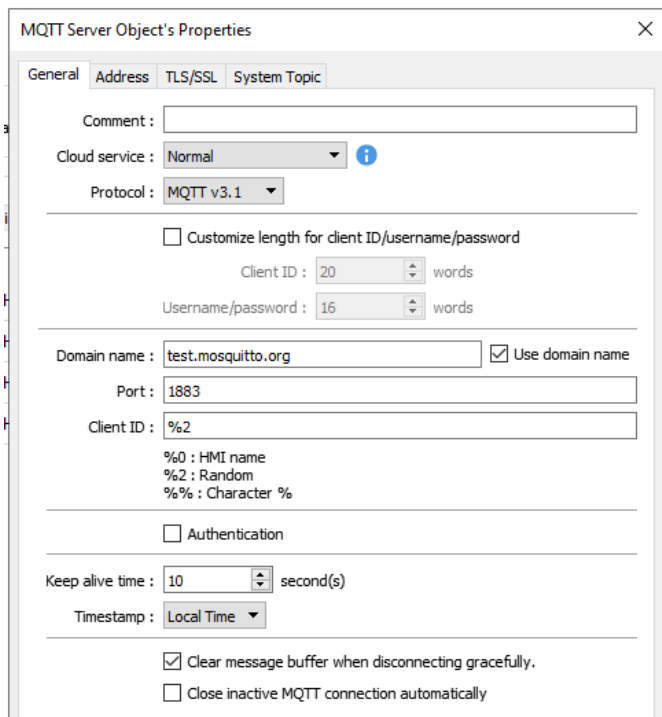
The settings below will configure the application to publish to a public broker at **broker.hivemq.com**. There are several public brokers that could be used to complete this project. Another, which is hosted by the Eclipse Mosquitto™ project, is **test.mosquitto.org**. These brokers are public and should only be used for testing purposes.

- Select **IIoT/Energy...MQTT** from the menu bar



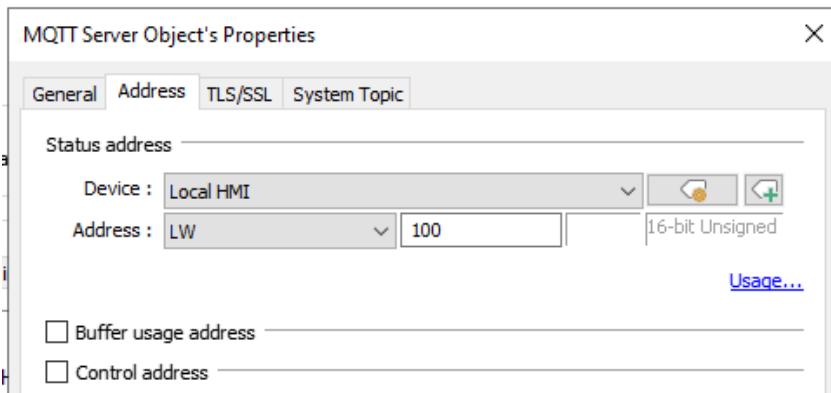
### Broker Configuration

- Click on the **Settings** button. This is where the settings for the MQTT broker are configured:



For more information on these settings refer to the MQTT chapter in the EBPro Programming Manual.

- Check **Use domain name** and enter the broker address.
  - The port is set to **1883**
  - The **Authentication** box is not checked.
- Click on the **Address** tab. Change the **Status address** from **LW – 0** to **LW – 100**

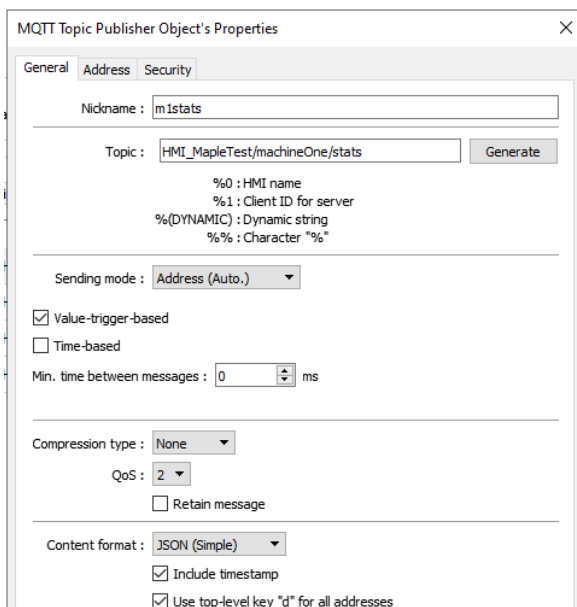


- Click **OK** to accept the settings and return to the MQTT dialog

## MQTT Topic Configuration

The next task is to create topics and assign HMI memory addresses to these topics. The process is the same for PLC addresses or tags.

- Click on **New...** from the MQTT dialog to create a new MQTT topic:



- Enter: **m1stats** for the **Nickname** of this topic
- Enter: **<name>/machineOne/stats** under **Topic**. Replace **<name>** with a unique identifier of your own choosing.
- Leave the Sending mode to **Address (Auto.)** and check the **Value-trigger-based** checkbox.

Now you will assign addresses to the topic:

- Click on the **Address** tab of the **MQTT Topic** window
- Click on **New** to assign a new address
- Enter: **TankLevel** for the **Name**
- Set the Type to **Word**
- Select **Local HMI, LW** and enter **0** for the address fields.

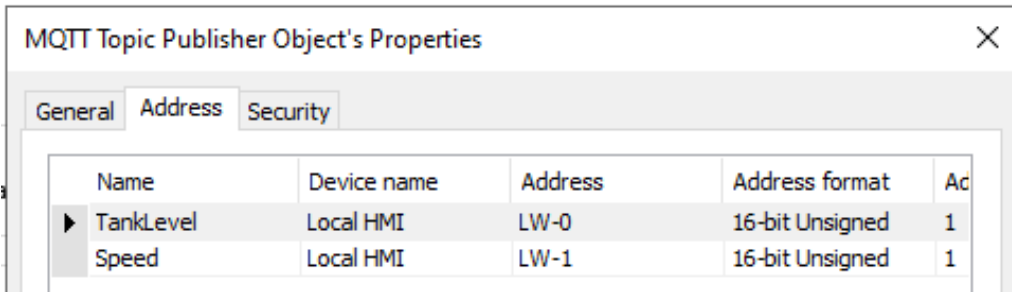
- Click **OK** to add the address to the topic.

Name	Device name	Address	Address format	Ac
TankLevel	Local HMI	LW-0	16-bit Unsigned	1

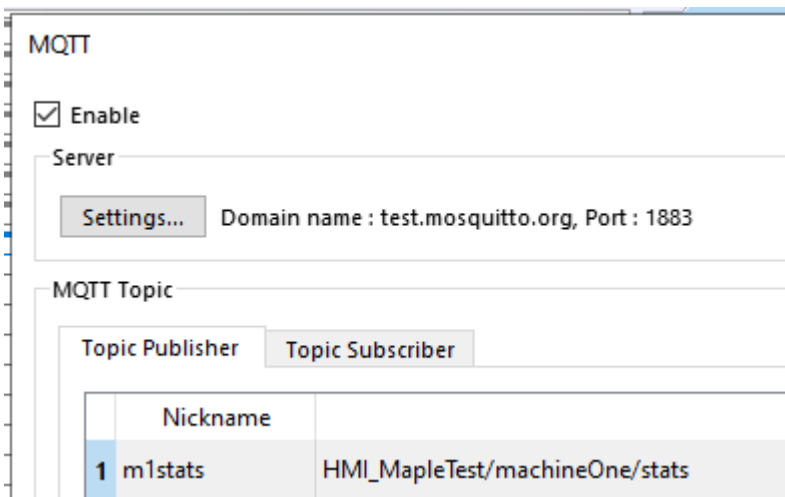
- This topic will have two addresses. Click new again and enter the following values to configure the second address:

Name: **Speed**  
 Type: **Word**  
 Address: **Local HMI, LW-1**

- The address tab for this topic should now look like:



- Click **OK** and the topic should appear in the **Topic:** list.

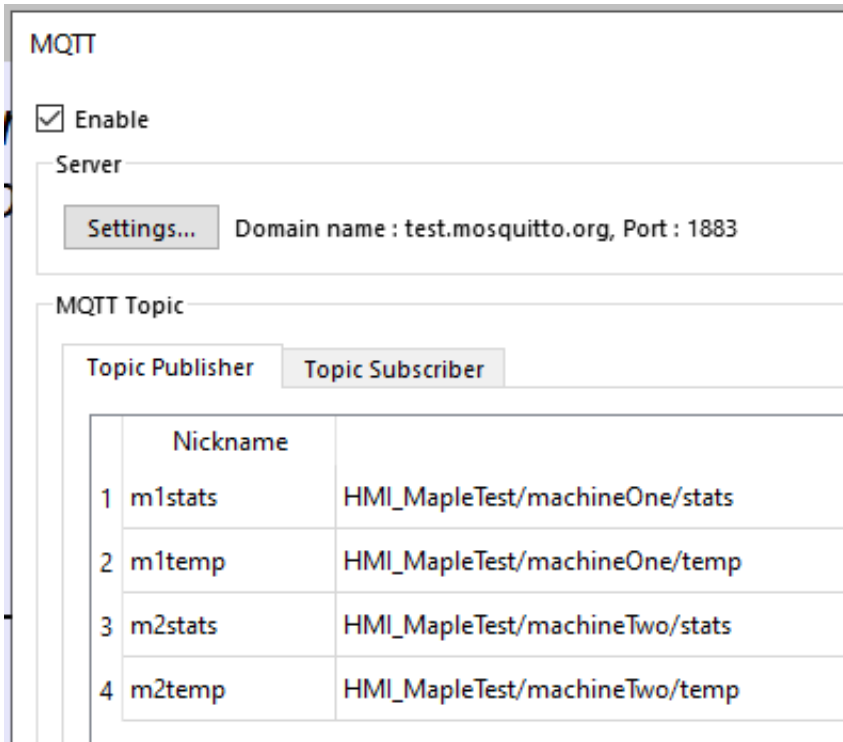


- Repeat this process to create the topics and addresses shown below (**Note:** Replace <name> with a unique name of your own choosing. The first topic shown was created in the preceding steps.):

Nickname	Topic	Address Name	Address
m1stats	<name>/machineOne/stats	TankLevel	LW – 0
		Speed	LW – 1
m1temp	<name>/machineOne/temp	Temp	LW – 2
m2stats	<name>/machineTwo/stats	TankLevel	LW – 10
		Speed	LW – 11
m2temp	<name>/machineTwo/temp	Temp	LW – 12

- The Topic list should now contain four topics:





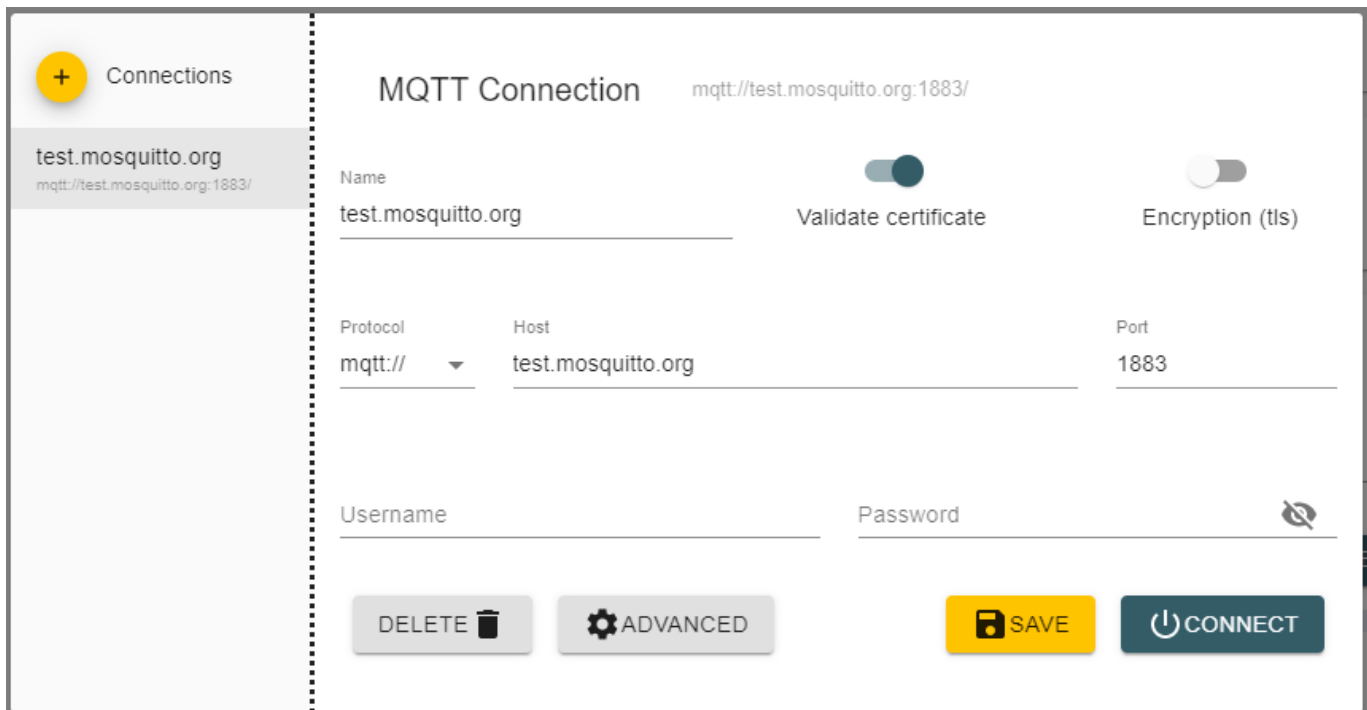
- When everything looks correct click **Exit** to return to the development screen.
- Select **file > Save** from the menu bar then **Tools > Compile** to build an executable version of the project that can be run in the simulator.

You have completed the HMI project set up. Now you can run the project in the simulator and start receiving data with an MQTT subscriber application.

## Step 3 - Testing

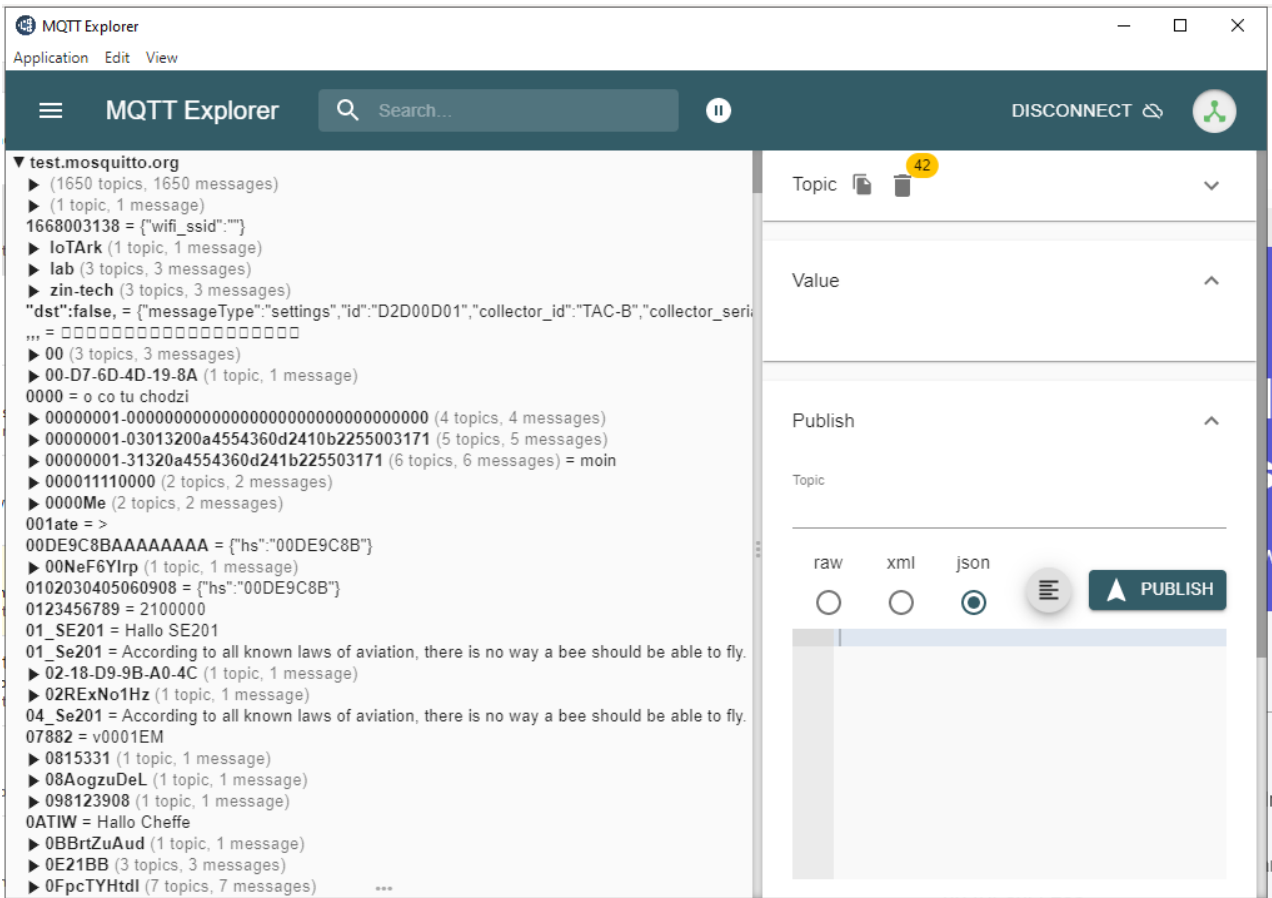
This step will use MQTT Explorer for the subscriber, if you don't have an MQTT client download and install MQTT Explorer before proceeding.

- Start the EBPro Online Simulation by selecting **Project...Online Simulation** from the menu bar. This will start the publisher on the PC. Leave this simulator running.
- Start MQTT Explorer. Click the yellow “+” sign to add a connection. We are going to use the Mosquitto Test broker:



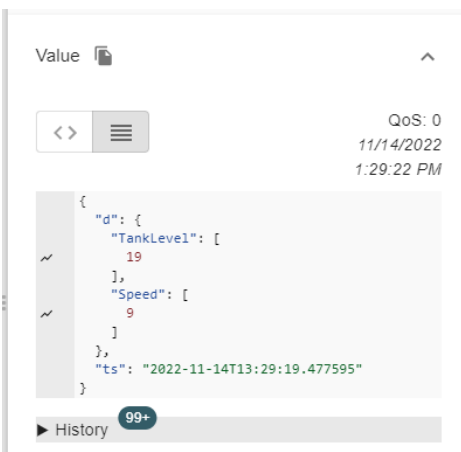
The screenshot shows the MQTT Explorer interface. On the left, there is a sidebar with a yellow plus icon and the text "Connections". Below it, a connection named "test.mosquitto.org" is listed with the URL "mqtt://test.mosquitto.org:1883/". The main area is titled "MQTT Connection" and shows the URL "mqtt://test.mosquitto.org:1883/". The "Name" field is "test.mosquitto.org". There are two toggle switches: "Validate certificate" (checked) and "Encryption (tls)" (unchecked). Below these, there are fields for "Protocol" (mqtt://), "Host" (test.mosquitto.org), and "Port" (1883). There are also fields for "Username" and "Password" with a clear icon. At the bottom, there are four buttons: "DELETE" (trash icon), "ADVANCED" (gear icon), "SAVE" (yellow button with floppy disk icon), and "CONNECT" (power icon).

- Enter a name for the connection.
- Leave **Validate certificate** active and **Encryption (tls)** inactive.
- Enter **test.mosquitto.org** as the **Host** address. Use **1883** for the port number.
- Click Save and then click connect.
- A list of topics will begin to populate from all of the users currently publishing to this test broker.



- To narrow down the view enter the unique topic identifier currently being used in your HMI project. For our sample, this would be **HMI\_MapleTest**.
- The list should now only display the topics that were created in the HMI project that is currently running in Online Simulation mode.

Since the **Sending mode** in the HMI application is set to **Value-trigger-based** you will see the data updated every time it changes in the HMI. In our sample project, we created six different Set Word objects that are jogging values to simulate data.



The packet payload delivered to the broker is in Java Script Object Notation (JSON).

The data in the “**d**” object are the data points added to the topic in EBPro, based on the names given in the address tab of the MQTT configuration window.

The data in the “**ts**” object is a combined date and time timestamp based on the HMI local time when the message was sent.

Congratulations, the HMI project is configured as a MQTT edge gateway.

Once your data is in the broker, you can subscribe to it from several software packages, or your own custom application. This data can be delivered to your entire team without overloading the control network with communications requests.

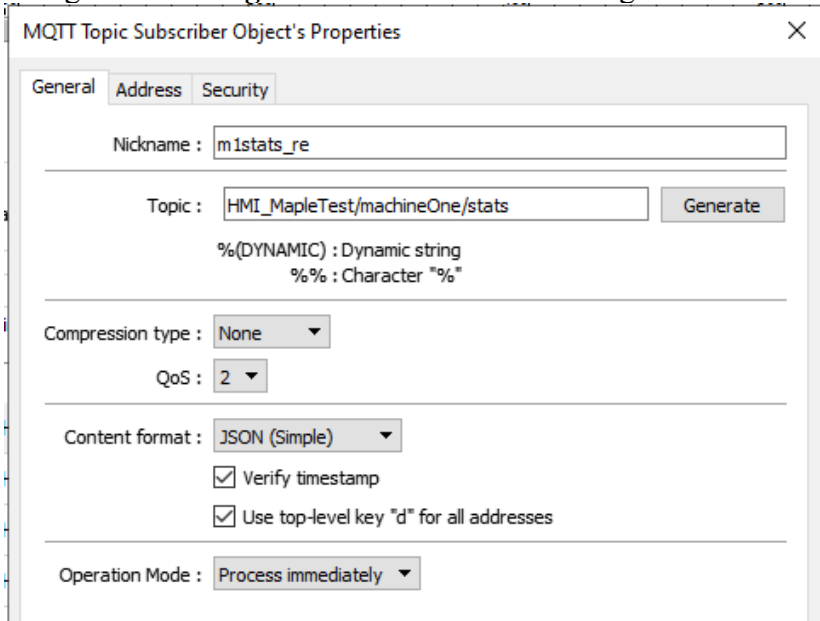
**Trouble Shooting Note:** If you don't get any messages from the HMI project try:

1. Make sure the Client ID is **%2** (for random), or something unique, in the MQTT object settings in EBPro. Recompile the project and relaunch the simulator.
2. Double check that the MQTT server is active and that the current Domain Name/IP address for the server was entered in the MQTT object.

## Step 4 – Set-up Topic Subscriber on HMI

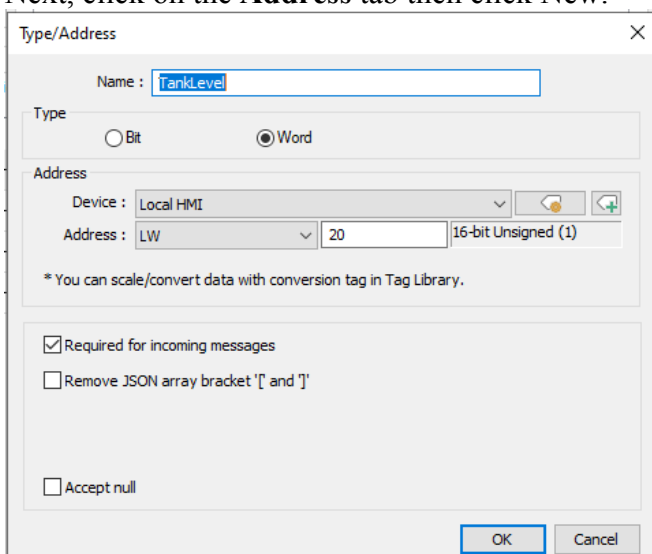
Now we will subscribe back to the data we are publishing to the test broker and place it into Local HMI memory.

- Begin by opening the **MQTT** object (**IIoT/Energy... MQTT**) and opening the **Topic Subscriber** tab.
- Topics are created the same way as when the Published topics were created, without the option to change the **Sending mode** since data is not being sent out.



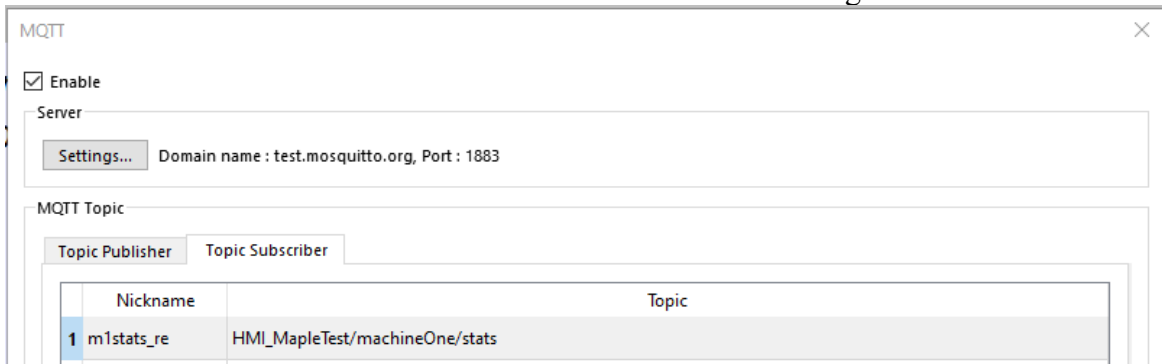
The screenshot shows the 'MQTT Topic Subscriber Object's Properties' dialog box. The 'General' tab is active. The 'Nickname' field contains 'm1stats\_re'. The 'Topic' field contains 'HMI\_MapleTest/machineOne/stats' and has a 'Generate' button next to it. Below the topic field, there is a legend: '%(DYNAMIC) : Dynamic string' and '%% : Character "%"'. The 'Compression type' is set to 'None'. The 'QoS' is set to '2'. The 'Content format' is set to 'JSON (Simple)'. There are two checked checkboxes: 'Verify timestamp' and 'Use top-level key "d" for all addresses'. The 'Operation Mode' is set to 'Process immediately'.

- Add a nickname for this topic, in our sample the same nickname is used as the published topic but with an added suffix “\_re”.
- The topic must match exactly what was created in the Publisher Object, <name>/machineOne/stats. Replace <name> with the unique topic identifier that was created previously
- Leave the rest of the settings as their defaults.
- Next, click on the **Address** tab then click New.



The screenshot shows the 'Type/Address' dialog box. The 'Name' field contains 'TankLevel'. The 'Type' section has 'Word' selected. The 'Address' section has 'Device' set to 'Local HMI', 'Address' set to 'LW', and '20' in the address field. The data type is '16-bit Unsigned (1)'. There is a note: '\* You can scale/convert data with conversion tag in Tag Library.' There are three checkboxes: 'Required for incoming messages' (checked), 'Remove JSON array bracket '[' and ']' (unchecked), and 'Accept null' (unchecked). 'OK' and 'Cancel' buttons are at the bottom.

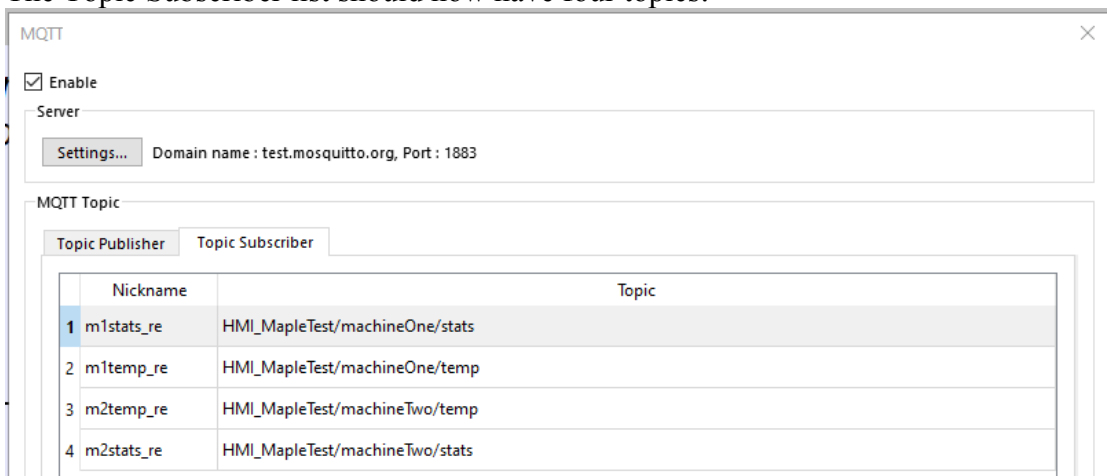
- Change the name to match exactly what the name of the published data is.  
**Note:** The name is case sensitive and if it does not match exactly then data will not come through.
- For the address, click the drop-down menu and select LW and type in 20 next to it. Click Ok to accept and close this window.
- Select **New...** and add another address for the below information:  
Name: **Speed**  
Type: **Word**  
Address: **Local HMI, LW-21**
- Click Ok to close and accept the addresses created for this topic. Our first topic subscription has now been created and the window should look like the below image:



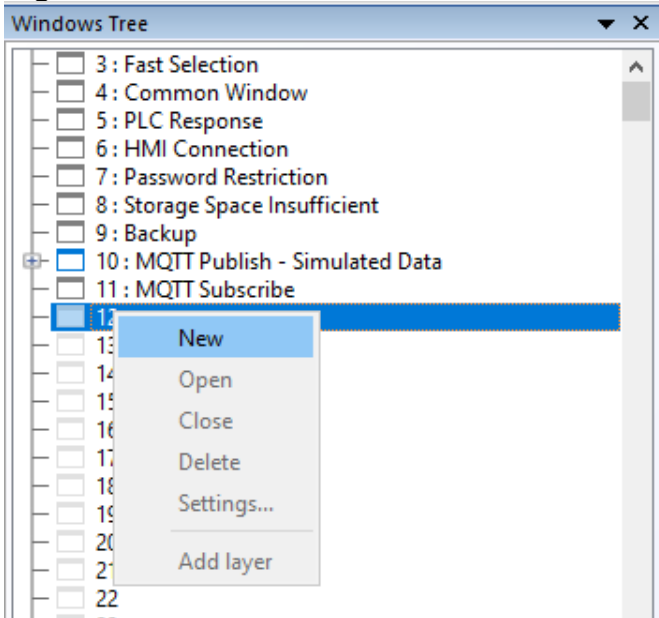
- Repeat this process to create the topics and addresses shown below (**Note:** Replace <name> with a unique name of your own choosing. The first topic shown was created in the preceding steps.):

Nickname	Topic	Address Name	Address
m1stats_re	<name>/machineOne/stats	TankLevel	LW – 20
		Speed	LW – 21
m1temp_re	<name>/machineOne/temp	Temp	LW – 22
m2stats_re	<name>/machineTwo/stats	TankLevel	LW – 30
		Speed	LW – 31
m2temp_re	<name>/machineTwo/temp	Temp	LW – 32

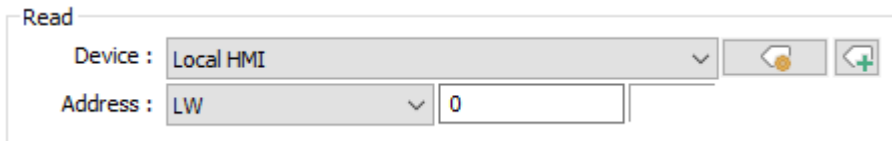
- The Topic Subscriber list should now have four topics:



- When everything looks correct, click **Exit** to close the MQTT Object.
- Select **File... Save** to save the project before proceeding to the next steps.
- Next, a window to display the data we are subscribing to will be created along with the associated Numeric Displays.
- Right-click on a blank window in the Windows Tree and select **New**.



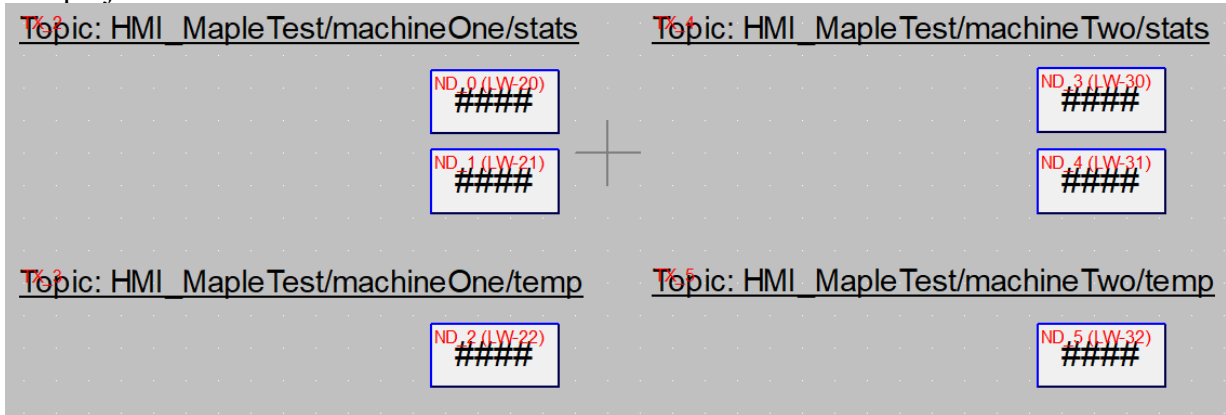
- Select **Object...Numeric** from the menu bar.
- Under **Read address** select Local HMI for the Device
- Select LW from the address type dropdown list, and enter **20** in the address field.



- Click **OK** to place this object on the screen
- Make five copies of this numeric object and configure them with the following addresses:

Address
LW-20 (already done)
LW-21
LW-22
LW-30
LW-31
LW-32

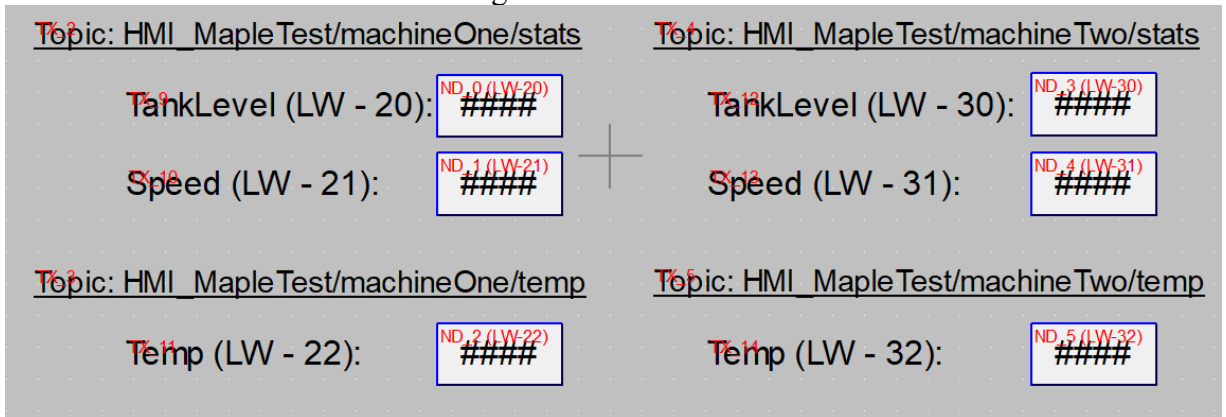
- The project should now look similar to what is shown below:



- Now add the following labels to the left of each numeric object so that it is clear what the number represents:

Numeric Input Address	Label
LW-20	TankLevel (LW -2 0):
LW-21	Speed (LW - 21):
LW-22	Temp (LW - 22):
LW-30	TankLevel (LW - 30):
LW-31	Speed (LW - 31):
LW-32	Temp (LW - 32):

The screen should now look something like this:



- Add a function key to take you between the Subscribe and Publish windows of the project.
- That completes the set up set up for the Subscription screen. The screen shot below shows some more optional labels and label formatting that can be added to better organize the screen:



## Machine One

Topic: HMI\_MapleTest/machineOne/stats

TankLevel (LW - 20):

ND\_0 (LW-20)  
#####

Speed (LW - 21):

ND\_1 (LW-21)  
#####

Topic: HMI\_MapleTest/machineOne/temp

Temp (LW - 22):

ND\_2 (LW-22)  
#####

## Machine Two

Topic: HMI\_MapleTest/machineTwo/stats

TankLevel (LW - 30):

ND\_3 (LW-30)  
#####

Speed (LW - 31):

ND\_4 (LW-31)  
#####

Topic: HMI\_MapleTest/machineTwo/temp

Temp (LW - 32):

ND\_5 (LW-32)  
#####

- To test the project, run it in Online Simulation mode again (**Project... Online Simulation**).
- Open MQTT Explorer and connect to the Mosquitto Test Broker, the connection should be saved from the previous test.
- Check that the data is being published using the MQTT Explorer client then change to the Subscribe window of the test project to see the data being written to the Local HMIs set-up in the previous steps.

If the data is not being written back to the HMI on the subscribe window, then try the below troubleshooting steps.

1. Check that the topics are written exactly in the Topic Subscribe set-up tab as they are in the Topic Publish tab.
2. Open up the address tab and make sure that the names of the data match **exactly** what is written in the publish tab. These names are case sensitive and if they do not match then data will not be shown.
3. If the option to “Use top-level key ‘d’ for all addresses” is checked in the Topic Publisher set-up, then it must be checked for the Topic Subscriber as well.

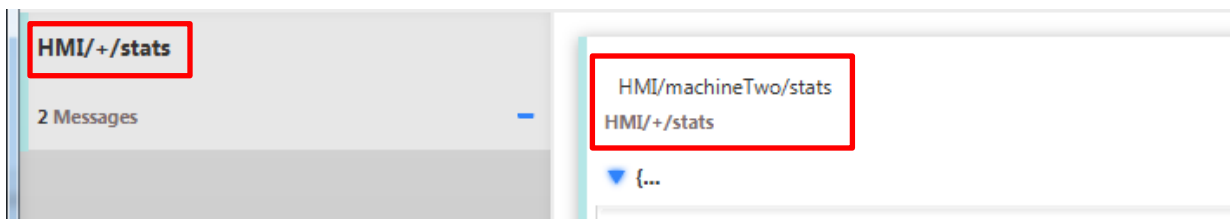
## Additional Information:

The following sections will give you more information on what can be done with the MQTT object and how to use it in your IIoT solution.

### Using Wildcards

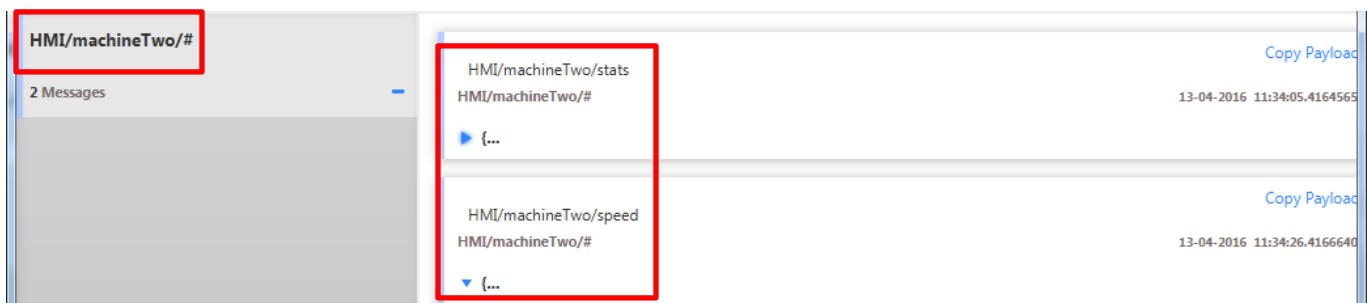
Subscribers can use wildcards to get data from multiple topics at once.

+ is the single level wildcard. As you saw above, we created the `<name>/machineOne/stats` and `<name>/machineTwo/stats` topics. If you subscribe to the topic `<name>/+/stats`, the topics of `<name>/machineOne/stats` and `<name>/machineTwo/stats` are delivered to the subscriber. Any time the variables in either of the `stats` topics changes a message will be received in MQTT Explorer.



In this example the two topics are on the same physical device but they do not have to be. The client would be subscribed to any device with a `stats` topic that was publishing to this broker.

The pound, hashtag or number sign (#) is the multi-level wildcard. Similar to the single level wildcard, you can subscribe to multiple topics at once. If subscribed to `<name>/machineOne/#`, you will receive ALL topics that start with `<name>/machineOne/`. This will include `<name>/machineOne/stats`, `<name>/machineOne/temp`, etc.



Subscribing to the topic `<name>/#` will subscribe to ALL of the topics on the HMI.

For more information on wildcards and MQTT best practices, go to: <http://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt-topics-best-practices>

## Other Broker Options

### Internal Broker

Selecting the local host option in the server settings dialog of the MQTT object will cause the HMI to use an internal broker. Other HMIs can publish through this HMI's broker using the IP address assigned to the HMI in the system settings dialog.

### Third Party Broker

There are several organizations that provide MQTT broker software. Generally these can be run on any PC and are pretty straightforward to configure. Once the server is up and running the HMI should be configured with the IP address of the computer where the broker is running.

### Mosquitto

Mosquitto ([www.mosquitto.org](http://www.mosquitto.org)) is an open source MQTT broker software that can be installed and hosted on several platforms. Mosquitto can be installed on a PC, a laptop, or even on a device like a Raspberry Pi.

### HiveMQ

HiveMQ ([www.hivemq.com](http://www.hivemq.com)) is a paid MQTT broker that offers additional features over Mosquitto.

### Cloud Broker

Several companies offer MQTT brokers hosted on their cloud services. Instead of creating and maintaining a server and software to distribute your data, the data can be transmitted to an offsite broker.

### Online Testing Broker

[test.mosquitto.org](http://test.mosquitto.org), and [broker.hivemq.com](http://broker.hivemq.com) are online test brokers that are open to the public. However, these broker is not private, so your data can be exposed. With that being said, it's a great place to start transmitting data from your HMI and subscribing with a testing client.

## Custom Client Applications

The JSON format used to publish messages from the HMI can easily be incorporated into website application, smart phone applications and much more.

A sample Android application can be found here:

<http://www.eclipse.org/paho/clients/android/sample/>

Information on using JSON objects in many programming languages can be found here:

<http://json.org/>