

# MQTT Supplement

## Contents

Section 1: MQTT Licensing.....	2
Section 2: MQTT Configuration .....	7
2.1 - MQTT Profile.....	8
2.2 Controller Tab .....	9
2.3 - The Broker Connection .....	11
2.3.1 Encryption.....	13
2.4 Topic Configuration .....	14
2.4.1 Add Topic .....	15
2.4.2 Update Type .....	17
2.4.3 Status Registers .....	18
Change Log.....	19

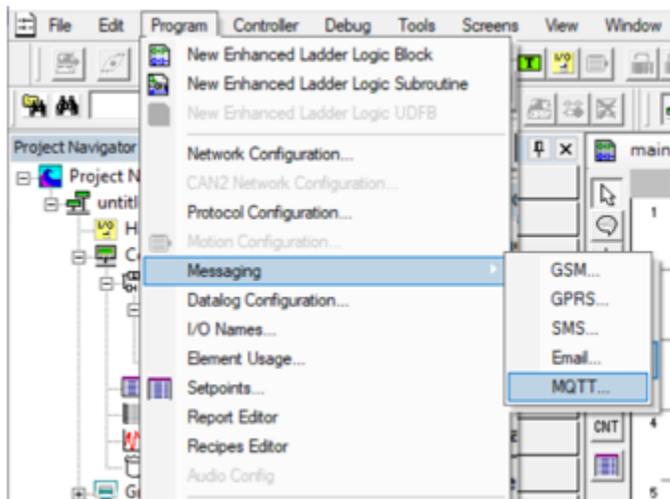
## Section 1: MQTT Licensing

From Cscape 9.90 SP3 and firmware 15.40 onwards, MQTT sparkplug has been supported in XL4, XL7, EXL6, EXL10, RCC1410, RCC2414, RCC8842, X5, EXLW and XL15+ models. MQTT is a licensed feature, and the user needs to purchase a permanent license to enable MQTT.

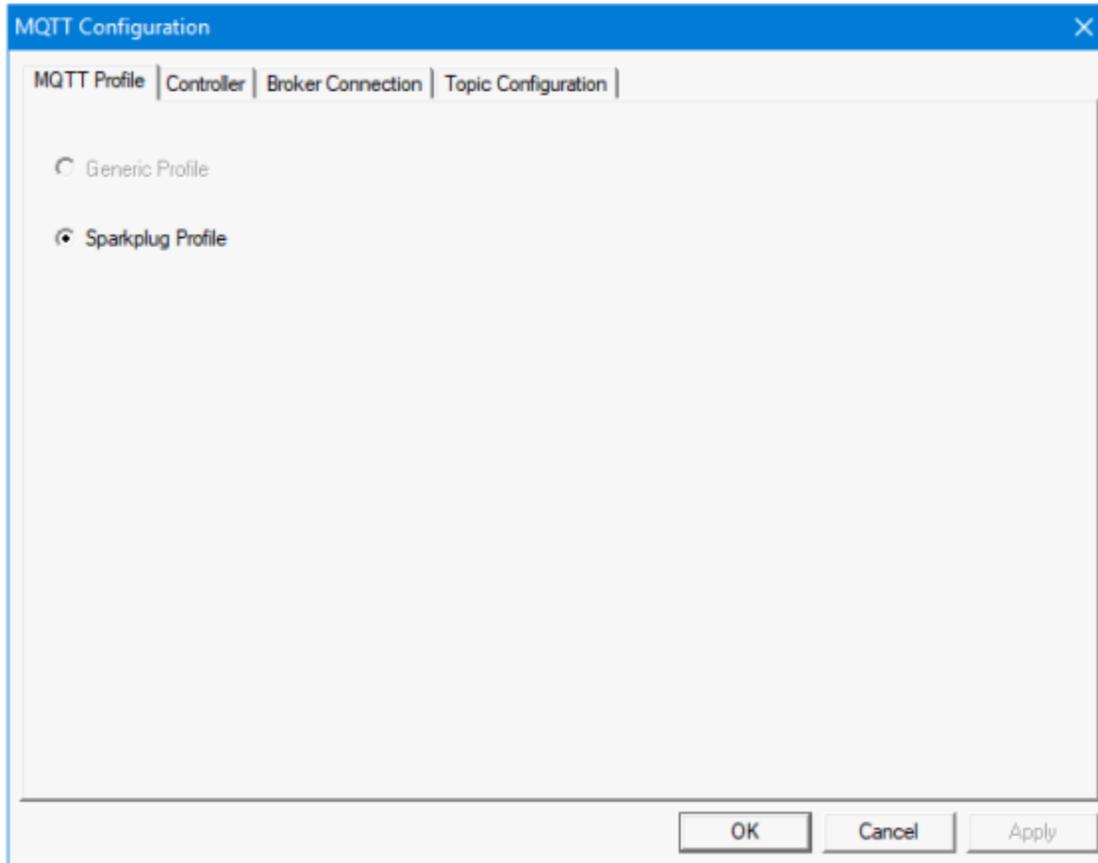
A removable media card needs to be present in the controller when the controller is being licensed.

NOTE: Once the controller is licensed the removable media card can be removed, if it is not being used for any other feature.

In Cscape, select **Program > Messaging > MQTT** option.

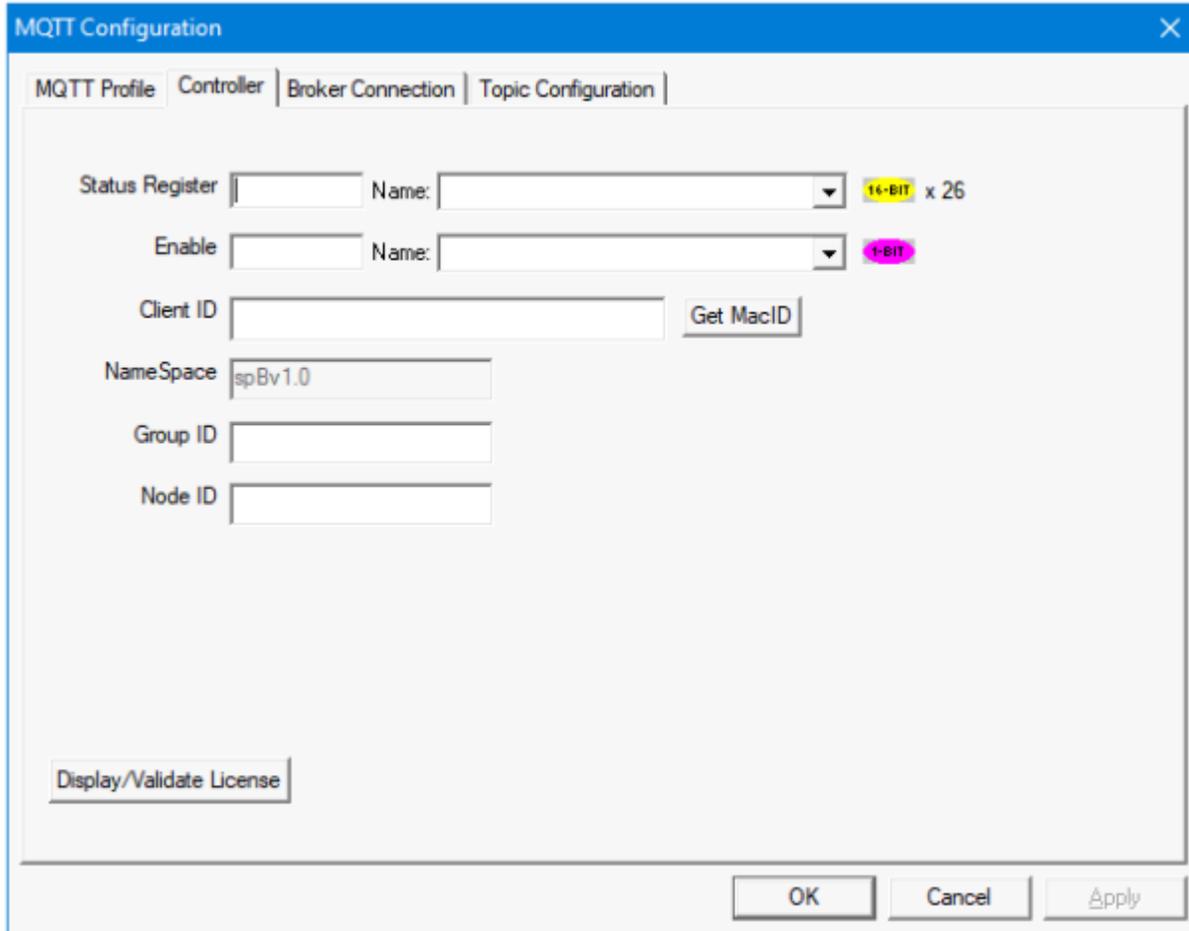


The following window is displayed:



Presently, Only Sparkplug Profile is supported in MQTT Profile tab.

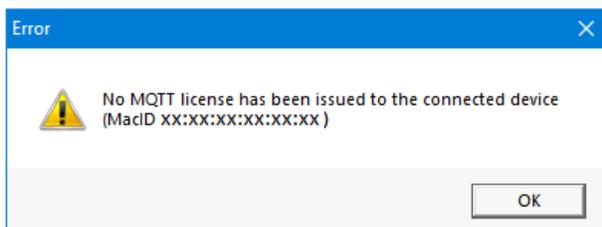
Select the **Controller** tab and the following fields will be displayed. Select **Display/Validate License**.



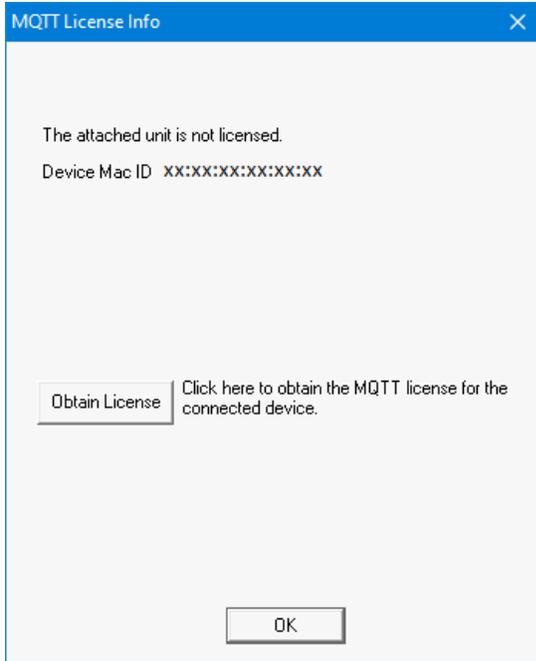
The image shows the MQTT Configuration dialog box with the Controller tab selected. The dialog has three tabs: MQTT Profile, Controller, Broker Connection, and Topic Configuration. The Controller tab is active. It contains several fields: Status Register (with a dropdown menu and a '16-BIT x 26' label), Enable (with a dropdown menu and a '1-BIT' label), Client ID (with a 'Get MacID' button), NameSpace (containing 'spBv 1.0'), Group ID, and Node ID. At the bottom left, there is a 'Display/Validate License' button. At the bottom right, there are 'OK', 'Cancel', and 'Apply' buttons.

**Display / Validate License** - This option is either used to download the license file to the controller or it connects to the licensing website to get a license if the controller is not licensed.

Click on the Display/Validate License option, if the controller has not been assigned a license the below window pops up.



After clicking on **OK** the following window will be displayed, select the Obtain License option.



Selecting Obtain License displays the following window and the licensing website will be displayed:

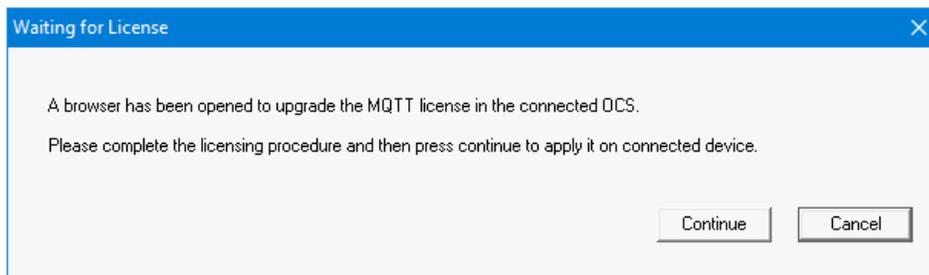


The user needs to login and apply the voucher code provided by Horner APG.



Once the voucher code is applied, click on the Redeem Voucher option to activate the license.

Once the voucher has been redeemed click on the Continue option to download the license file to the controller.



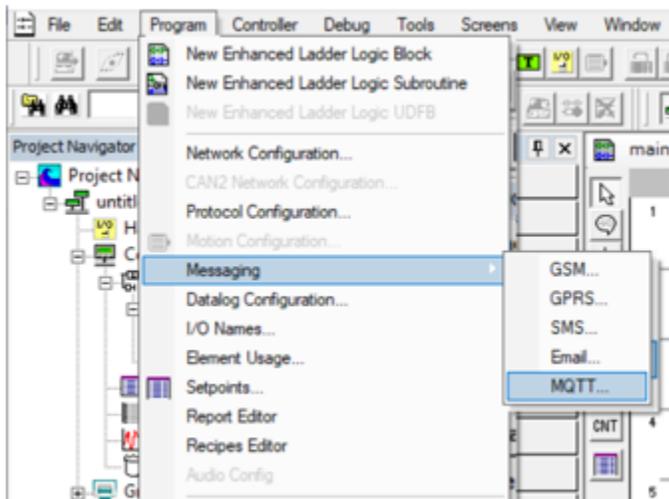
Check the Removable media for MQTTFirmware.lic file.

## Section 2: MQTT Configuration

From Cscape 9.90 SP3 and firmware 15.40 onwards, MQTT sparkplug has been supported in the XL4, XL7, EXL6, EXL10, RCC1410, RCC2414, RCC8842, X5, EXLW, and XL15+ models.

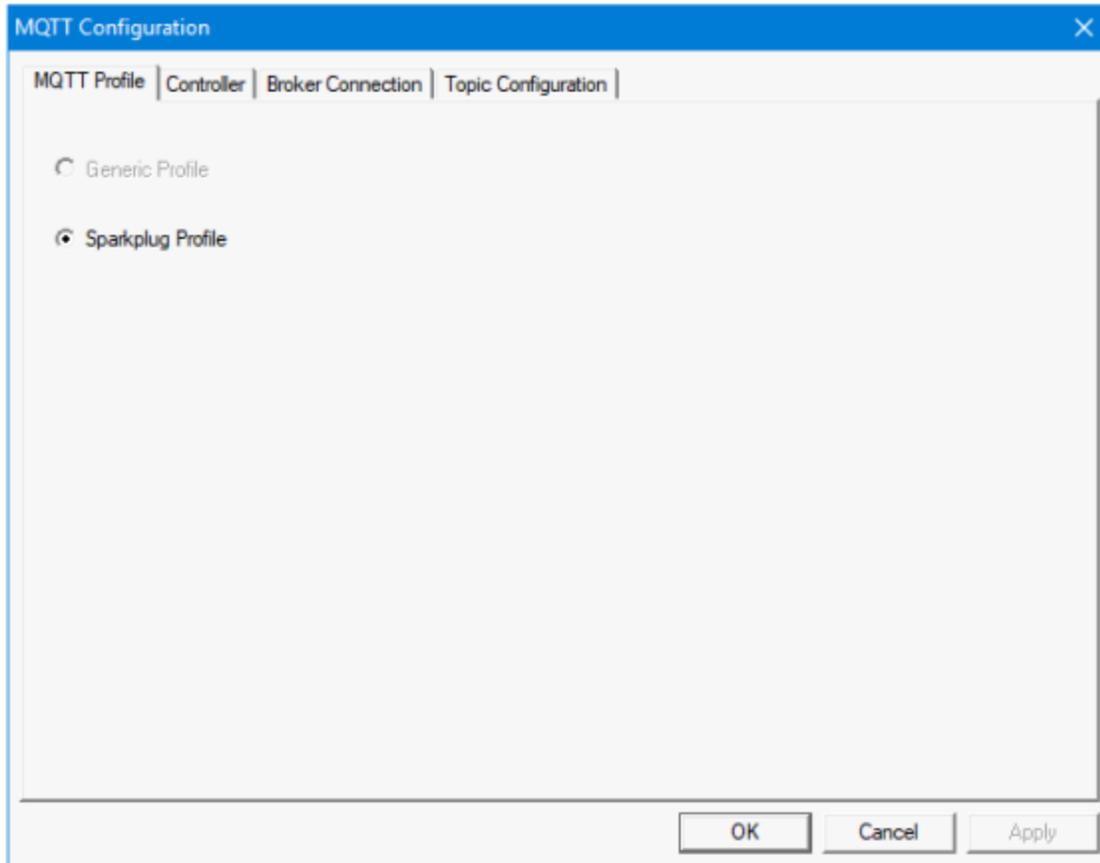
MQTT stands for Message Queuing Telemetry Transport. MQTT is a simple messaging protocol. It is a lightweight publish and subscribe system where a user can publish and receive messages as a client. MQTT allows users to send commands to control outputs, read and publish data.

In Cscape, select **Program > Messaging > MQTT** option.



## 2.1 - MQTT Profile

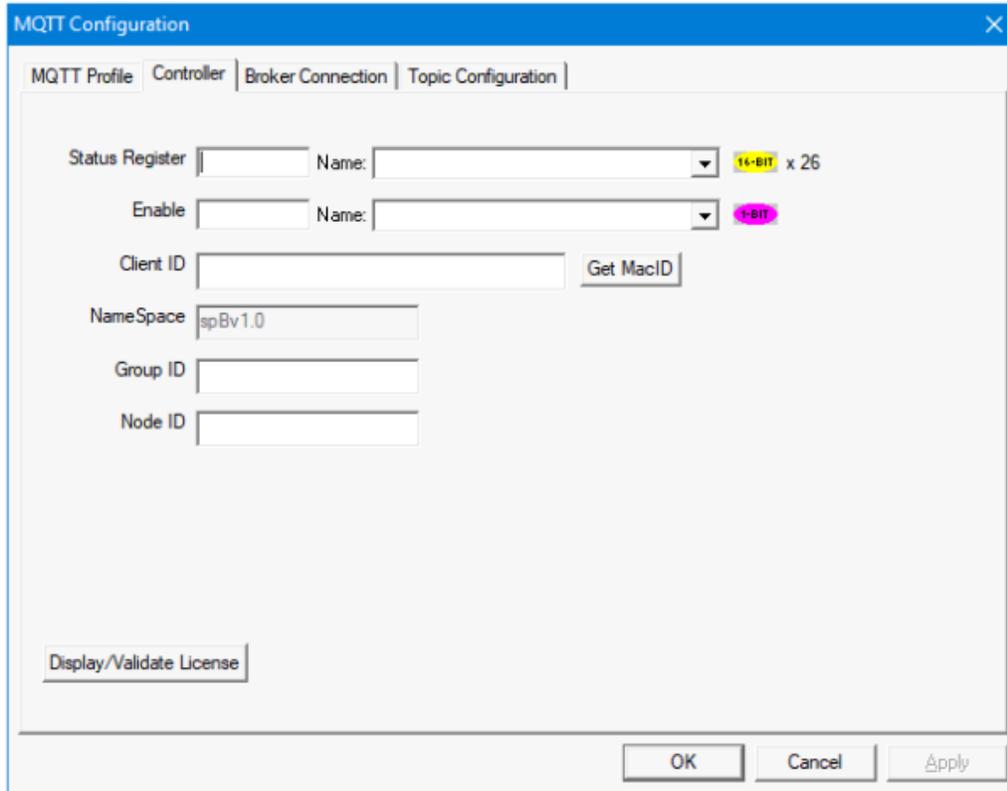
This opens the MQTT Profile tab:



Presently, Only Sparkplug Profile is supported in MQTT Profile tab.

## 2.2 Controller Tab

Select the **Controller** tab and the following fields will be displayed. Configure all the fields.



**Status Register** - It is 16-bit register which occupies 36 consecutive registers.

### Status Register Details

#### Base Register (16 Bit)

Bit 1 0 - MQTT main engine start failed  
1 - MQTT main engine start success.

Bit 2 0 - Invalid MQTT configuration downloaded from Cscape.  
1 - MQTT configuration OK.

Bit 3 0 - MQTT License Invalid1  
1 - MQTT License valid

Bit 4 to Bit 11 (1 bit for each client's connection status)  
0 - MQTT connection failed  
1 - MQTT connection success

Base Register + 1 - Total count of Published messages  
Base Register + 3 - Total count of Subscribed messages  
Base Register + 5 - Count of Broker 1 Published messages  
Base Register + 7 - Count of Broker 1 Subscribed messages  
Base Register + 9 - Count of Broker 2 Published messages  
Base Register + 11 - Count of Broker 2 Subscribed messages  
Base Register + 13 - Count of Broker 3 Published messages  
Base Register + 15 - Count of Broker 3 Subscribed messages  
Base Register + 17 - Count of Broker 4 Published messages  
Base Register + 19 - Count of Broker 4 Subscribed messages  
Base Register + 21 - Count of Broker 5 Published messages  
Base Register + 23 - Count of Broker 5 Subscribed messages  
Base Register + 25 - Count of Broker 6 Published messages  
Base Register + 27 - Count of Broker 6 Subscribed messages  
Base Register + 29 - Count of Broker 7 Published messages  
Base Register + 31 - Count of Broker 7 Subscribed messages  
Base Register + 33 - Count of Broker 8 Published messages  
Base Register + 35 - Count of Broker 8 Subscribed messages

**Enable** - It is a 1 bit register used to run MQTT.

**Client ID** - This field is used to enter client MAC ID manually.

**Get MacID** - Selecting this button will autofill the MACID of the controller into Client ID field.

**Group ID** - Logical grouping of MQTT nodes (String)

**Node ID** - Identifies the MQTT node within the broker (String). **NOTE:** The group ID combined with Node ID needs to be unique within the broker.

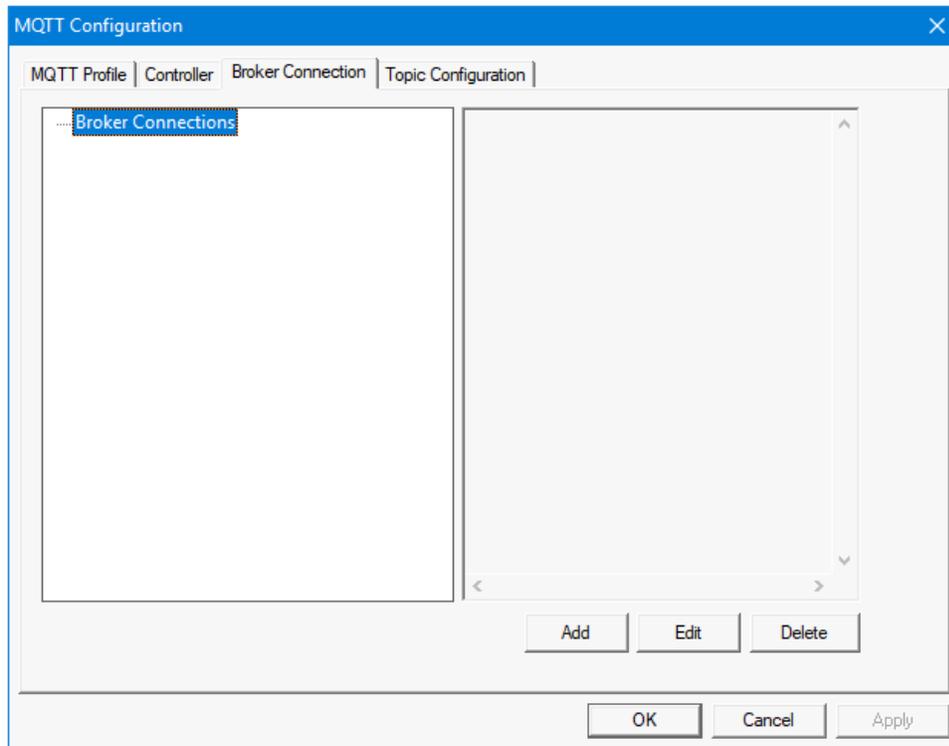
**NameSpace** - Defines the encoding used for payload data.

**Display / Validate License** - This option is either used to download the license file to the controller or it connects to licensing website to get a license, if the controller is not licensed.

See also - [MQTT Licensing](#)

## 2.3 - The Broker Connection

Selecting the **Broker Connection** tab displays the following window:

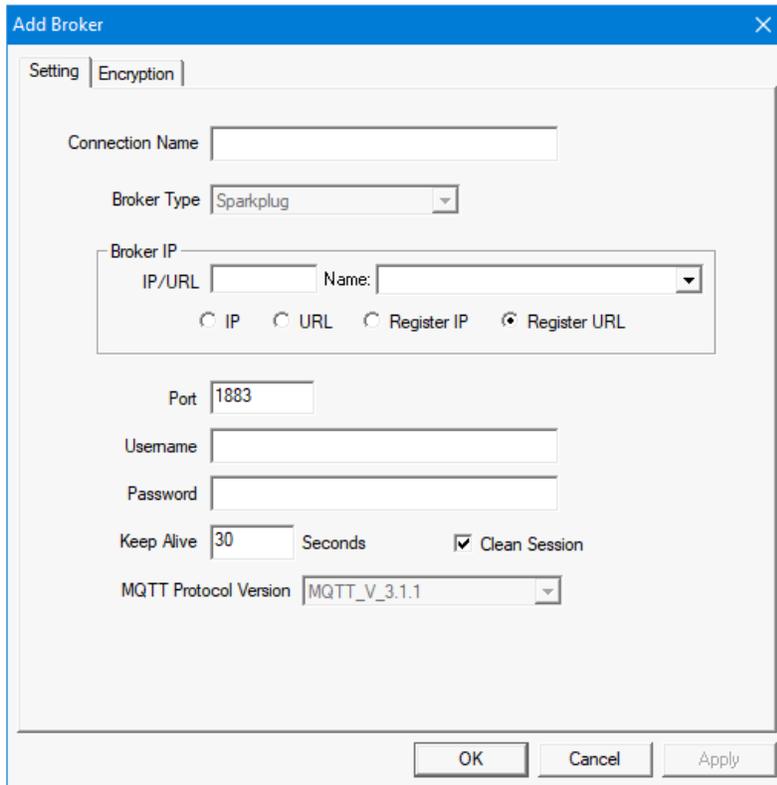


**Add** - Selecting this option allows user to add a new broker connection.

**Edit** - Selecting this option allows user to edit an existing broker connection.

**Delete** - Selecting this option allows user to delete an existing broker connection.

Selecting the **Add** option displays the following window:



**Connection Name** - User defined name (string).

**Broker Type** - SparkPlug is selected as default.

**Broker IP** - This is the broker IP field which can be configured as IP / URL / via Register for IP /Via Register for URL

**Port** - Port number to connect to the broker

**Username** - Username to connect to the broker

**Password** - Password to connect to the broker

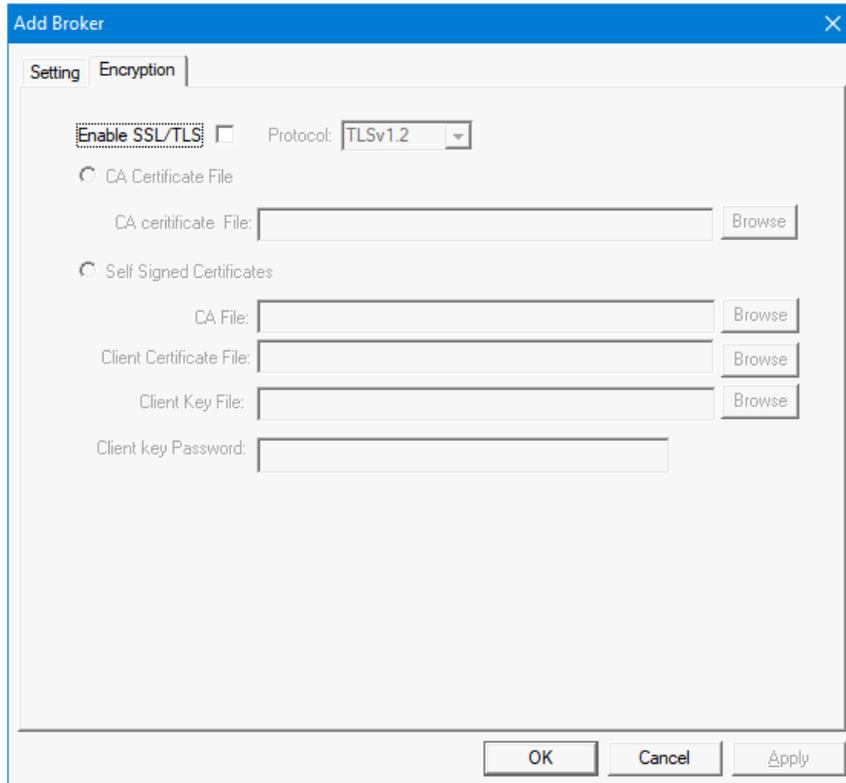
**Keep Alive** - The client and broker exchange a Heartbeat message within this time, to ensure both are online.

**Clean Session** - If unchecked saves all the data relevant to the client on the broker.

**MQTT Protocol Version** - Displays the version of MQTT protocol

## 2.3.1 Encryption

Selecting the **Encryption** tab displays the following window:



The screenshot shows a dialog box titled "Add Broker" with a close button (X) in the top right corner. The dialog has two tabs: "Setting" and "Encryption", with "Encryption" selected. The "Encryption" tab contains the following controls:

- An "Enable SSL/TLS" checkbox, which is currently unchecked.
- A "Protocol" dropdown menu set to "TLSv1.2".
- A radio button for "CA Certificate File", which is selected.
- A text input field for "CA certificate File:" with a "Browse" button to its right.
- A radio button for "Self Signed Certificates", which is unselected.
- Three text input fields for "CA File:", "Client Certificate File:", and "Client Key File:", each with a "Browse" button to its right.
- A text input field for "Client key Password:".

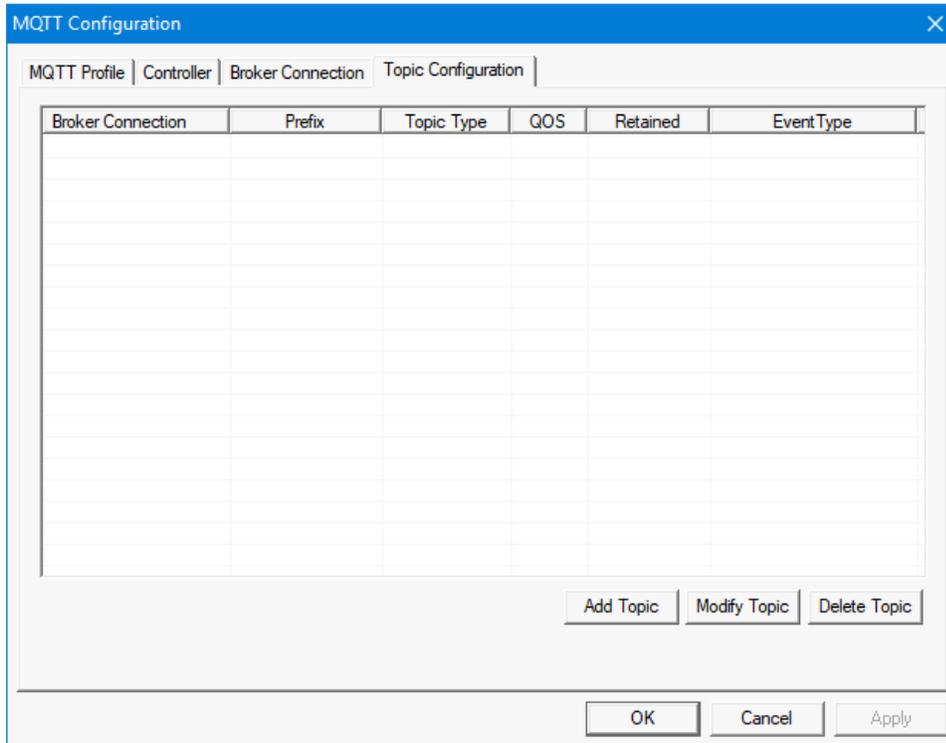
At the bottom of the dialog are three buttons: "OK", "Cancel", and "Apply".

Here, the user can enable SSL/TLS support.

If SSL/TLS is enabled, then the user is required to browse and attach the certificate file to be downloaded to the controller.

## 2.4 Topic Configuration

Selecting **Topic Configuration** tab displays the following window:



The configured topics are displayed in the above table.

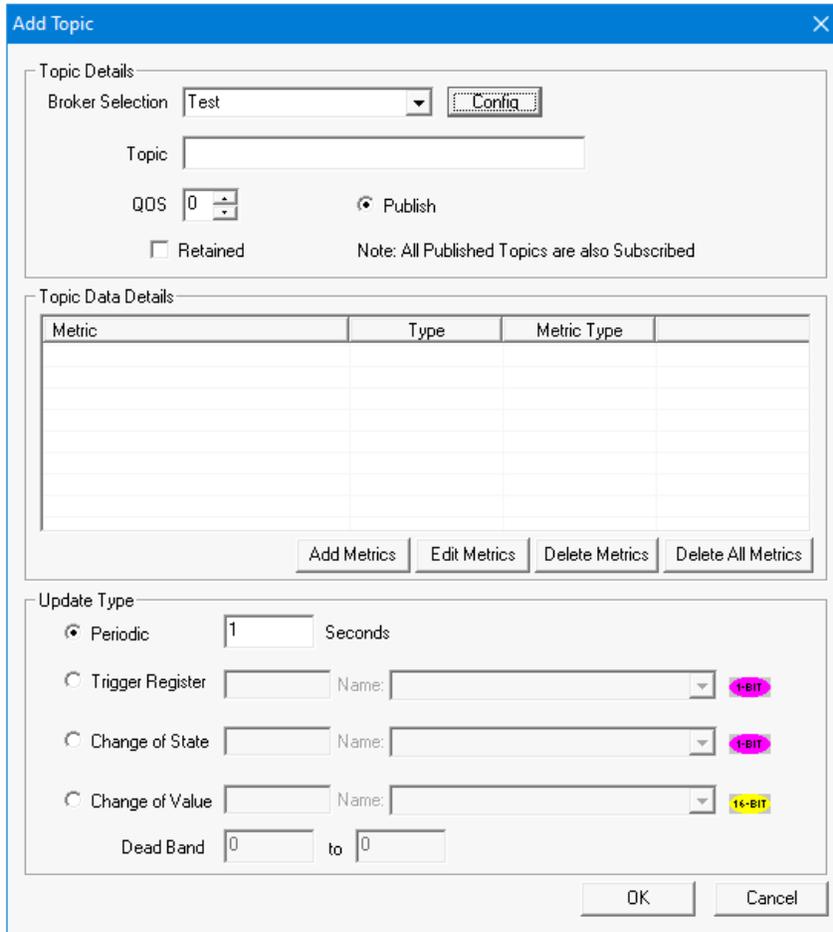
**Add Topic** - This option is used to add a new topic.

**Modify Topic** - This option is used to modify the existing topic.

**Delete Topic** - This option is used to delete the existing topic.

## 2.4.1 Add Topic

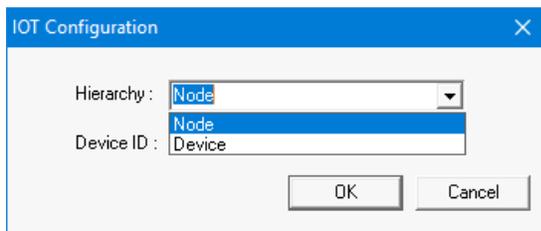
Selecting **Add Topic** displays the following window:



### Topic Details

**Broker Selection** - The configured brokers are listed in this drop down.

**Config** - Selecting this option displays the following window where user can configure the hierarchy of the topic.



Two hierarchies are available: **Node** and **Device**. To segregate the data being sent, it can be sent either as node data or device data.

If user selects **Device** option, then Device ID option is enabled for publishing topics.

**Prefix** - This is the Data folder (String)

**QOS (Quality of Service)** - QOS level is an agreement between the sender of a message and the receiver of a message that defines the guarantee of delivery for a specific message.

3 QoS levels in MQTT:

0 - At most once

1 - At least once

2 - Exactly once

When discussing QoS in MQTT, the two sides of message delivery need to be considered:

- Message delivery from the publishing client to the broker.
- Message delivery from the broker to the subscribing client.

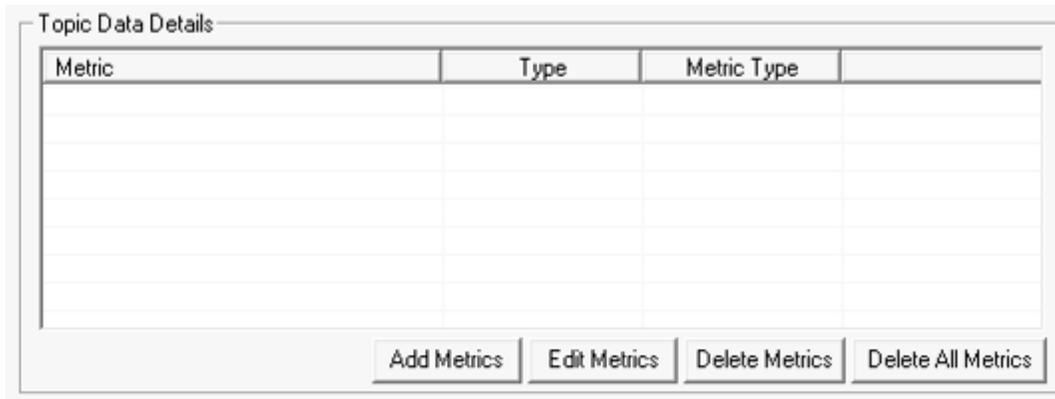
We will look at the two sides of the message delivery separately because there are subtle differences between the two.

- The client that publishes the message to the broker defines the QoS level of the message when it sends the message to the broker.
- The broker transmits this message to subscribing clients using the QoS level that each subscribing client defines during the subscription process. If the subscribing client defines a lower QoS than the publishing client, the broker transmits the message with the lower quality of service.

**Retained** - A retained message is a normal MQTT message with the retained flag set to true. The broker stores the last retained message and the corresponding QoS for that topic. Each client that subscribes to a topic pattern that matches the topic of the retained message receives the retained message immediately after they subscribe. The broker stores only one retained message per topic.

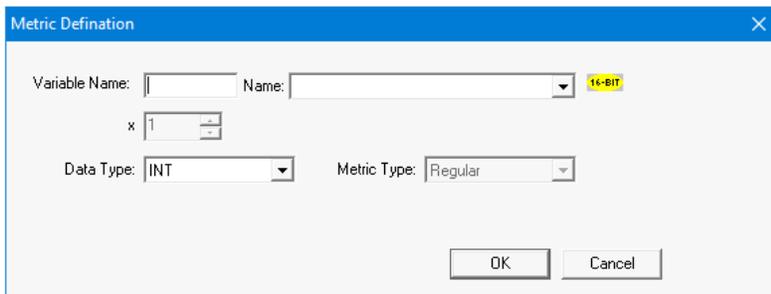
**Publish** - Selecting this publishes the data.

## Topic Data Details



Metric	Type	Metric Type

Selecting **Add Metrics** displays the following window:



Metric Definition

Variable Name:  Name:  16-BIT

x

Data Type:  Metric Type:

**Variable Name** - The register data to be sent.

**Data Type** - Type of data to be sent. All Data types are supported.

**Edit Metrics** - This option is used to edit an existing metric data.

**Delete Metrics** - This option is used to delete the selected metric data.

**Delete All Metrics** - This option is used to delete all metric data configured.

### 2.4.2 Update Type

There are 4 update types supported. User can select any option as needed.

- **Periodic** - Triggers an update for every configured time.
- **Trigger Register** - Triggers an update whenever the trigger register goes high.
- **Change of State** - Triggers an update whenever the configured bit changes state from OFF to ON or ON to OFF.
- **Change of Value** - Triggers an update whenever the value in the configured register changes from the last present value.
- **Dead Band** - If there is a value change within the values set in the dead band then, there is no publish triggered.

### 2.4.3 Status Registers

#### Register Address: Base Register

##### General Status 1 (16 bit):

- Bit 1 0 - MQTT Main engine start failed  
1 - MQTT Main engine start success
- Bit 2 0 - Invalid MQTT configuration downloaded from Cscape  
1 - MQTT configuration OK
- Bit 3 0 - MQTT License Invalid  
1 - MQTT License valid
- Bit 4 to 11 - 1 bit for each client's connection status
  - 0 - MQTT connection failed
  - 1 - MQTT connection success

- Base Register + 1 - Total count of Published messages
- Base Register + 3 - Total count of Subscribed messages
- Base Register + 5 - Count of Broker 1 Published messages
- Base Register + 7 - Count of Broker 1 Subscribed messages
- Base Register + 9 - Count of Broker 1 Buffer messages
- Base Register + 11 - Count of Broker 2 Published messages
- Base Register + 13 - Count of Broker 2 Subscribed messages
- Base Register + 15 - Count of Broker 2 Buffer messages
- Base Register + 17 - Count of Broker 3 Published messages
- Base Register + 19 - Count of Broker 3 Subscribed messages
- Base Register + 21 - Count of Broker 3 Buffer messages
- Base Register + 23 - Count of Broker 4 Published messages
- Base Register + 25 - Count of Broker 4 Subscribed messages
- Base Register + 27 - Count of Broker 4 Buffer messages
- Base Register + 29 - Count of Broker 5 Published messages
- Base Register + 31 - Count of Broker 5 Subscribed messages
- Base Register + 33 - Count of Broker 5 Buffer messages
- Base Register + 35 - Count of Broker 6 Published messages
- Base Register + 37 - Count of Broker 6 Subscribed messages
- Base Register + 39 - Count of Broker 6 Buffer messages
- Base Register + 41 - Count of Broker 7 Published messages
- Base Register + 43 - Count of Broker 7 Subscribed messages
- Base Register + 45 - Count of Broker 7 Buffer messages
- Base Register + 47 - Count of Broker 8 Published messages
- Base Register + 49 - Count of Broker 8 Subscribed messages
- Base Register + 51 - Count of Broker 8 Buffer messages

##### Limitations:

- Maximum of 8 broker connections are possible.
- Maximum of 64 topic per broker can be published.
- Maximum of 64 topic per broker can be subscribed.
- Maximum of 64 metrics / registers can be configured per topic.

## Change Log

Date	Rev#	Description of Revision	Location in Doc
12/13/21	1.1	Specified which RCC units support MQTT.	Section 1 and 2 overviews.