

1. In-Sight Vision Suite 25.3.0 Enterprise Connectivity – MQTT

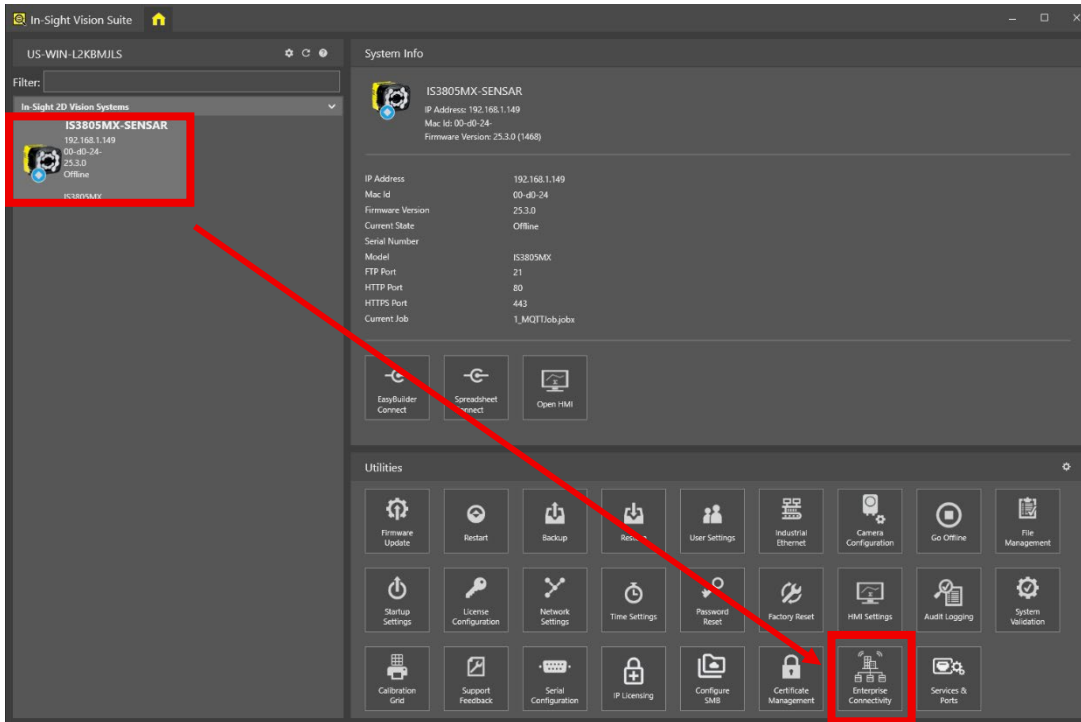
Starting with In-Sight Vision Suite (ISVS) 25.3.0, the utilities set now includes Enterprise Connectivity and supports MQTT. With this new feature, you can send specified results through the metrics builder option and use the native mode commands to perform options, such as remote triggering or specific cell value retrieval and setting.

This guide will walk you through MQTT communication in ISVS with both Spreadsheet and EZBuilder options.

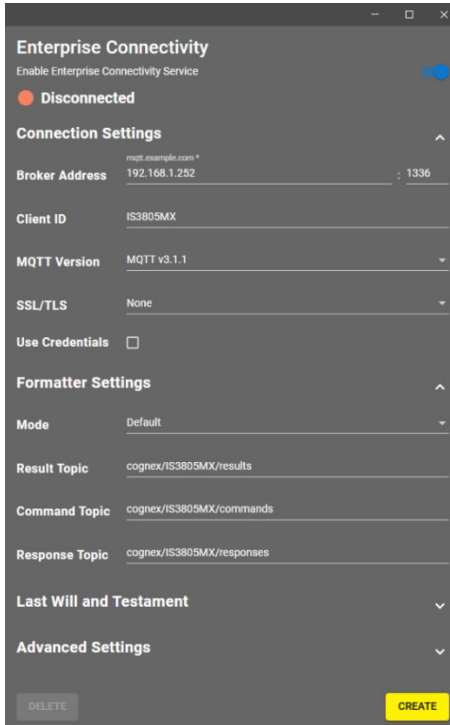
This guide assumes that you already have an MQTT Broker running v3.1.1 or v5 on an accessible server on the same network that the ISVS camera is on. Please ensure that you have the proper server access, and necessary credentials to communicate with your MQTT broker prior to starting this guide.

1.1 Enterprise Connectivity Utility

In ISVS 25.3.0 and up, you can use the Enterprise Connectivity utility to enable MQTT communication on the camera.



Once selected, you can create your connection.



The screenshot shows the 'Enterprise Connectivity' window. At the top, there's a toggle for 'Enable Enterprise Connectivity Service' which is turned on. Below it, a status indicator shows 'Disconnected'. The 'Connection Settings' section is expanded, showing fields for 'Broker Address' (192.168.1.252), 'Client ID' (IS3805MX), 'MQTT Version' (MQTT v3.1.1), and 'SSL/TLS' (None). There is a checkbox for 'Use Credentials' which is unchecked. Below this is the 'Formatter Settings' section, which is also expanded, showing 'Mode' (Default), 'Result Topic' (cognex/IS3805MX/results), 'Command Topic' (cognex/IS3805MX/commands), and 'Response Topic' (cognex/IS3805MX/responses). At the bottom, there are sections for 'Last Will and Testament' and 'Advanced Settings', both collapsed. At the very bottom, there are 'DELETE' and 'CREATE' buttons.

We will use the following options to configure the connection to talk to the broker.

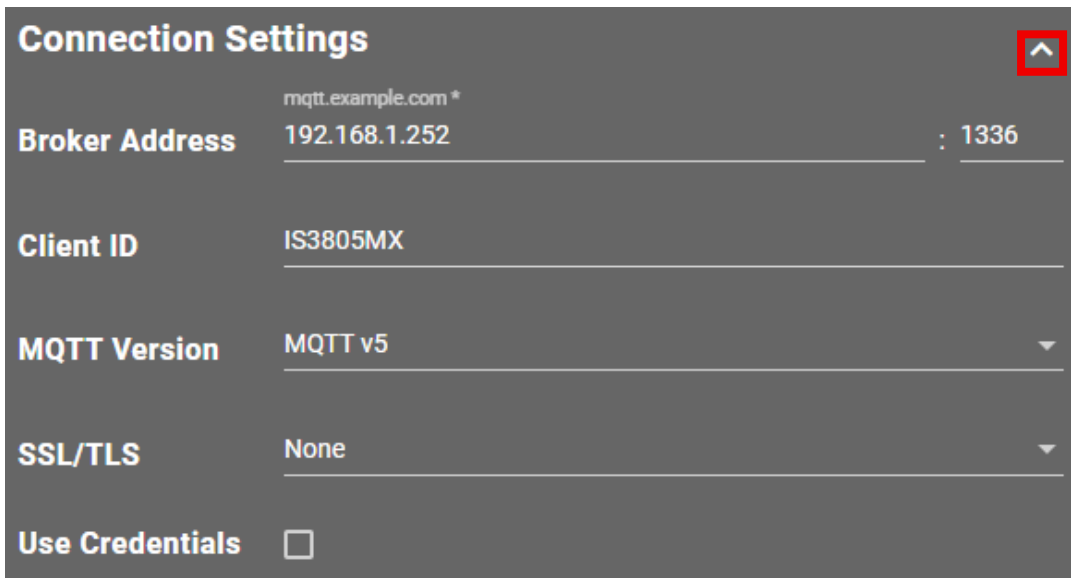
Broker Address – IP address or hostname of the broker (in this example the IP is 192.168.1.252) followed by the port number that the broker operates on.

Client ID – IS3805MX (default is the serial number but can be changed to what is needed).

MQTT Version – v5(v3.1.1 should also work but this will depend on your setup).

SSL/TSL – Use if your broker communicates with TLS/mTLS.

Use Credentials – Check this if your broker requires a username and password.



The screenshot shows the 'Connection Settings' form. It has a title bar with a red square icon containing a white upward arrow. The form has several fields: 'Broker Address' with a value of '192.168.1.252' and a port field with '1336'; 'Client ID' with 'IS3805MX'; 'MQTT Version' with a dropdown menu showing 'MQTT v5'; 'SSL/TLS' with a dropdown menu showing 'None'; and 'Use Credentials' with an unchecked checkbox. The form is dark-themed with white text.

Next, expand the Formatter Settings.



Note: In this example, we will use the default formatting. You can select a custom mode for formatting allowing full scripting control. Information on this method can be found in the [In-Sight Vision Suite Help](#) guide installed with ISVS as well as on Cognex.com.

Ensure that the following are set. By default, your camera's serial number is used in the topic name. You can set this to your camera's name like in this example.

Mode – Default.

Result Topic – cognex/IS3805MX/results (Metric results are published here).

Command Topic – cognex/IS3805MX/commands (Camera commands can be published to this topic).

Response Topic – cognex/IS3805MX/responses (Response from camera command inputs are published here).

Formatter Settings	
Mode	Default
Result Topic	cognex/IS3805MX/results
Command Topic	cognex/IS3805MX/commands
Response Topic	cognex/IS3805MX/responses

Last Will and Testament (LWT) set a message to send to the broker if the camera disconnects unexpectedly. If you are using MQTT v3, you can set LWT topic, QoS, retain client information, and message payload. If you are using MQTT V5 on your broker, you can enter values for content type, delay interval, and message expiry. In our example, we will leave this blank, but you can set them if your broker requires it.

Last Will and Testament	
Topic	
QoS	0 - At most once (fire and forget)
Retain	
Content Type	MIME type e.g. application/json
Delay Interval (Secs)	
Message Expiry (Secs)	
Payload	Message payload

In the **Advanced Settings**, you can set the following:

QoS – The level of the ISVS messages (0, 1, or 2).

Clean Session Start – Determines if the connection session should start cleanly and disregard the previous session state.

Max Timeout (sec) – Interval in seconds before abandoning non-responsive connection.

Inflight Limit – Maximum number of unacknowledged outgoing in-flight messages.

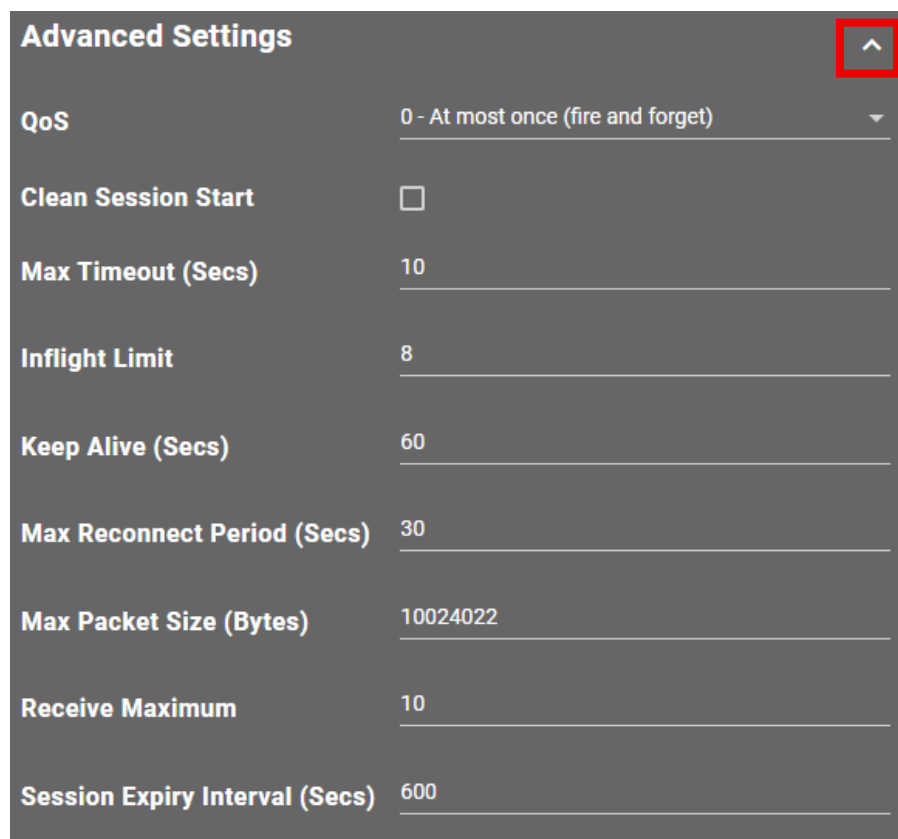
Keep Alive (Secs) – Interval in seconds between connection keep-alive messages.

Max Reconnect Period (Secs) – Maximum backoff interval in seconds before attempting reconnect.

Max Packet Size (Bytes) – Maximum allowed packet size in bytes.

Receive Maximum – Maximum number of QoS1/QoS2 messages the client can process concurrently.

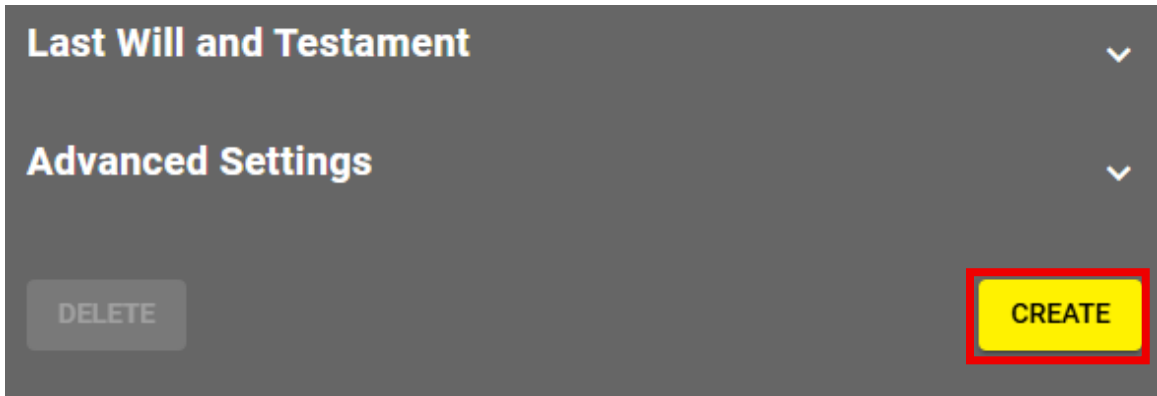
Session Expiry Interval (Secs) – Interval in seconds for broker to retain session after disconnecting.



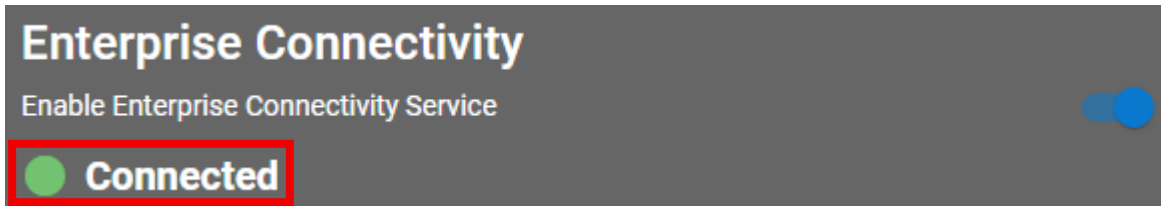
Advanced Settings	
QoS	0 - At most once (fire and forget) ▼
Clean Session Start	☐
Max Timeout (Secs)	10
Inflight Limit	8
Keep Alive (Secs)	60
Max Reconnect Period (Secs)	30
Max Packet Size (Bytes)	10024022
Receive Maximum	10
Session Expiry Interval (Secs)	600

For now, we will leave these at their default values. You can change them if your particular broker requires them.

Once you have finished setting up your connection parameters, click **Create**.

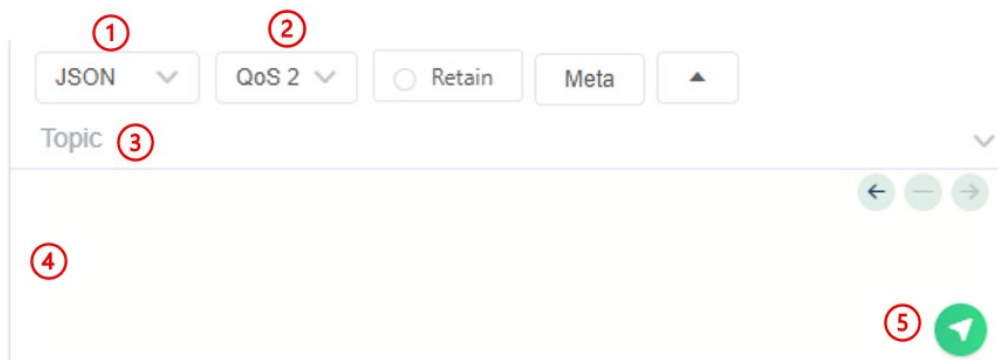


If successful, you will see the connection status displayed at the top as shown and can close the Enterprise Connectivity window.



Send Test Trigger

From here, we can now test that our camera is taking commands through MQTT by using the manual trigger native mode command. To do so, go to your MQTT viewer application of choice and publish to the **commands** topic. In this example, we are using the MQTTX application that is free to use and download online. Publishing can be done with the lower-right corner of the MQTTX application.

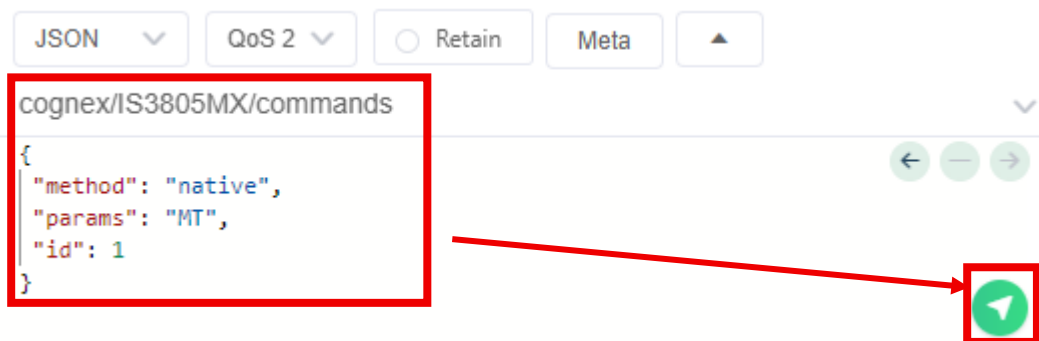


Location	Description
1	Payload/message format (leave this at JSON)
2	QoS setting (leave this at QoS 2)
3	Topic to publish to (set this to the commands topic, cognex/IS3805MX/commands)
4	Payload/message (we will set this to the following JSON) below
5	Publish button

For the message, enter the following:

```
{  
  "method": "native",  
  "params": "MT",  
  "id": 1  
}
```

The **"id": 1** portion of the message payload assigns an ID to the response we want sent back. If multiple commands are sent, we have a way to identify which response corresponds to which previously sent command.



From here, press the publish button to send the command. If successful, you should see your camera trigger. Inside MQTTX, you should also see a response tied to the **Response** topic we set up earlier, letting you know the command was acknowledged and valid with the status code of 1 and is tied to the command we sent with ID 1.



If your camera does not trigger, ensure that you have copied the command message correctly and that your camera's connection to the broker is active by going to the **Enterprise Connectivity** utility.

If your camera triggered correctly, you can move onto setting up your inspection to include MQTT results.

1.1.1 Inspection Results Set Up

Metrics

When looking to interact with the MQTT communication within ISVS, you will need to create **Metrics**. Metrics are specific values from your inspection tools that are set up to be sent into the MQTT **results topic** payload. Below is a summary of the types of metrics that can be used.

Name	Description
Measurement	A numerical value that can contain a high/low range value as well as unit type.
Count	A numerical value to specify a quantity.
Classification	An enumerable with a single string value selected from a set of possible values.
Identification	A string value that is indexed for faster lookup.
Text	A simple string value.

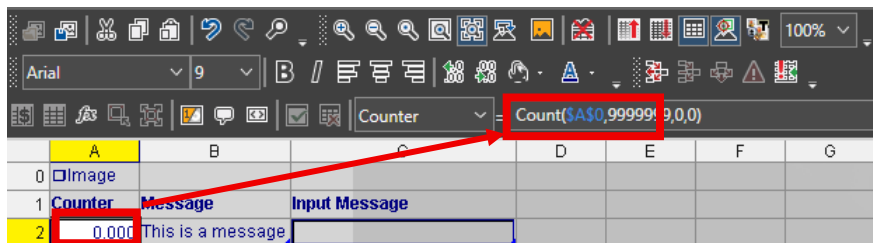
The next two sections will show how to do this in both Spreadsheet and EZBuilder.

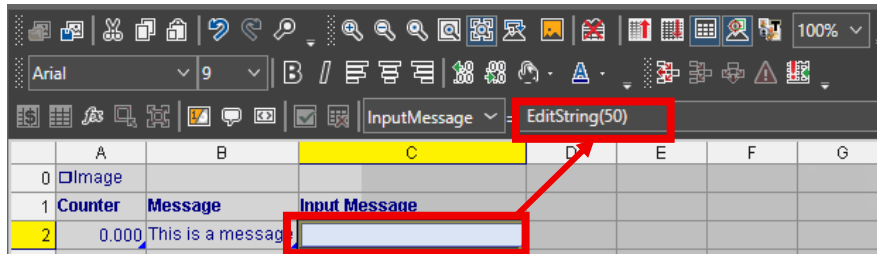
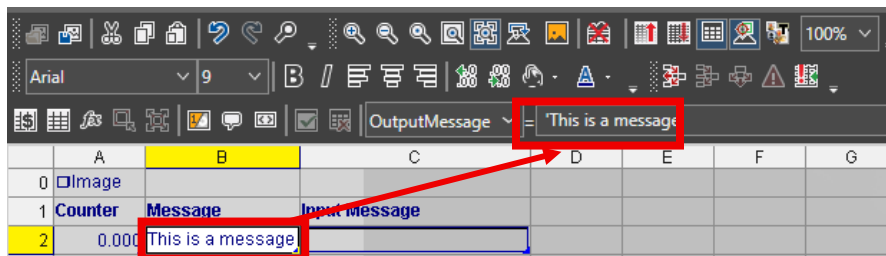
1.1.2 Spreadsheet Metrics Set Up

To set up the metrics for use in Spreadsheet, add cell names to the cells you wish to make into metrics. Select the cell you wish to add a name to and type in the name next to the cell equation editor at the top.

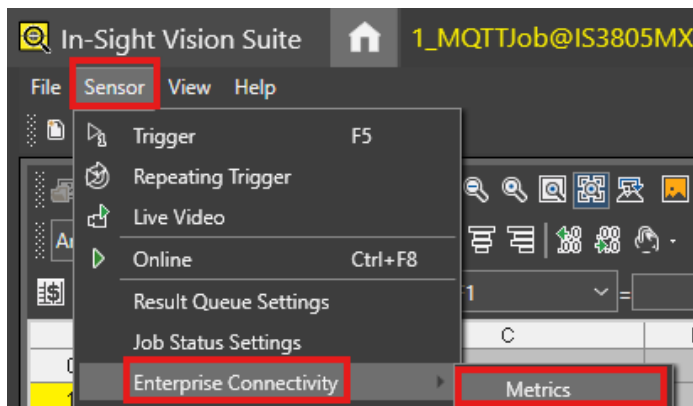
In the example spreadsheet below, we will set up three cells and give each cell a name so we can reference them.

Cell Location	Cell Name	Cell Type
A2	Counter	Count
B2	OutputMessage	String
C3	InputMessage	EditString

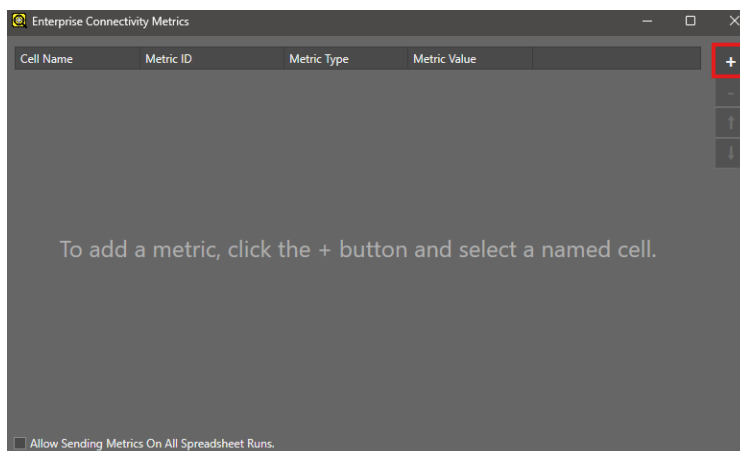




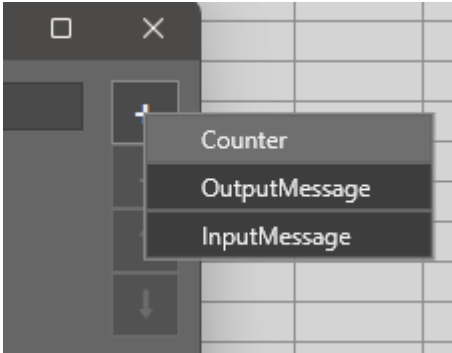
Once we have these cells named, we can go to the following menu: **Sensor -> Enterprise Connectivity -> Metrics.**



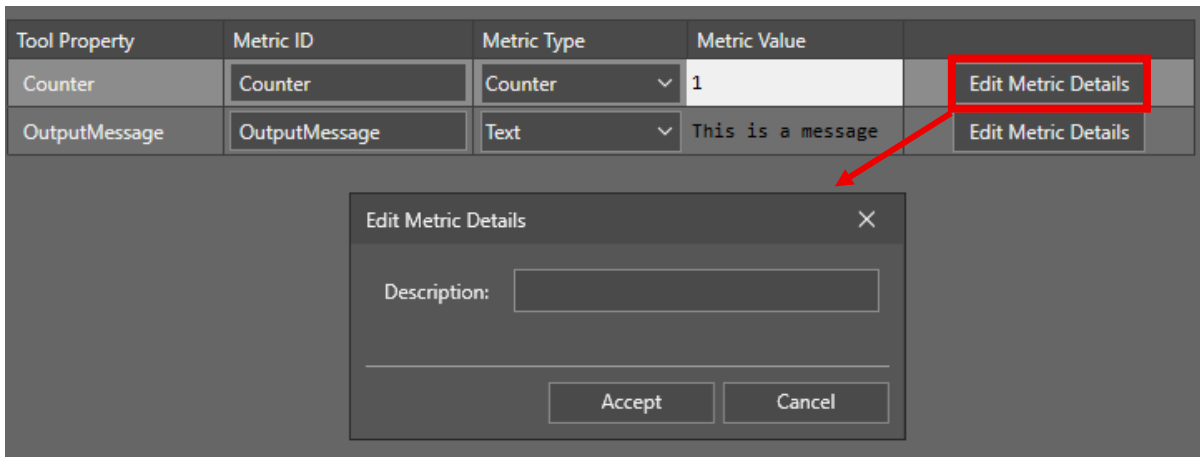
This will bring up the metrics window. In this window, we can add the metrics that we wish to publish in our results topic by clicking +.



When you click **+**, any cells that have a valid name will appear and can be selected. We will select the **Counter** and **OutputMessage**.



As you add properties, you will see them added to the list. Note that the order they appear in the list will be the order that they are outputted to the result topic. You have the option to alter the **Metric ID** and **Metric Type**. There is a button labeled **Edit Metric Details**. This allows you to add a description of the metric if needed. If the metric type is a measurement, you can also set the **Range Low** value, **Range High** value, and **Unit** type in this window as well.

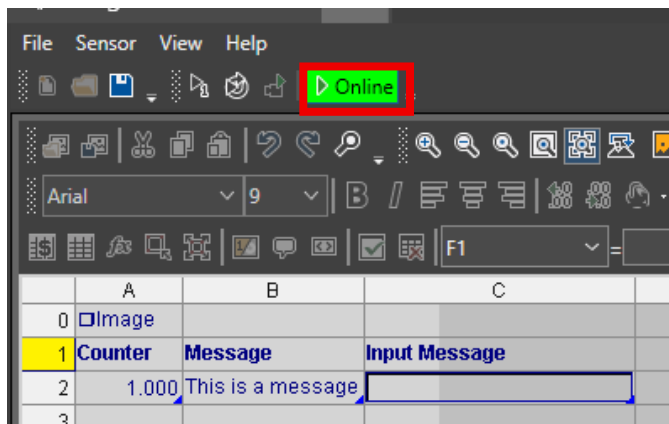


At the bottom of the window, you have the options to publish the metrics without requiring an acquisition from the camera. This can be useful if you need to publish metrics without triggering the camera.

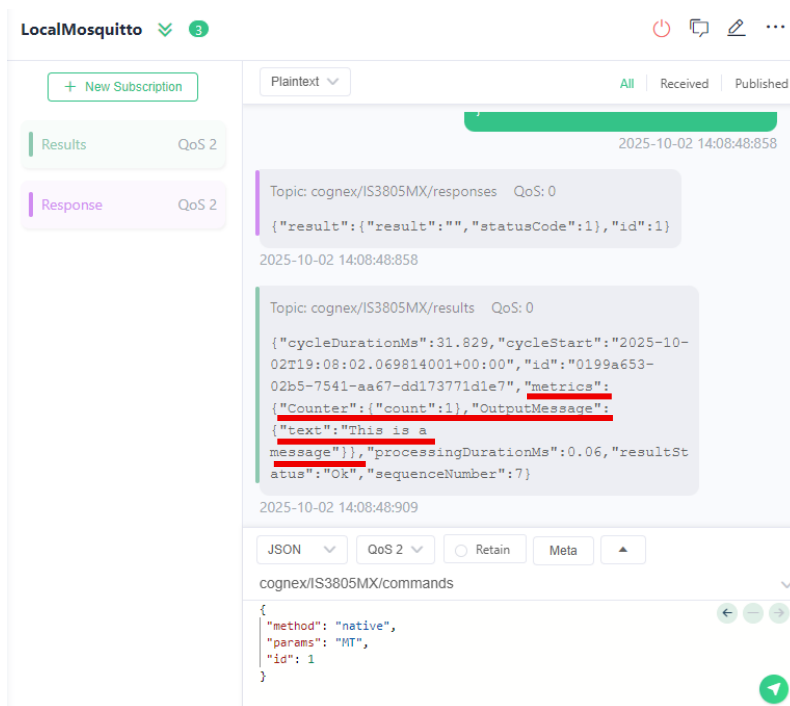
☐ Allow Sending Metrics On All Spreadsheet Runs.

Once you have entered your metrics, you can close this window.

The camera will need to be online for the MQTT communications to work.

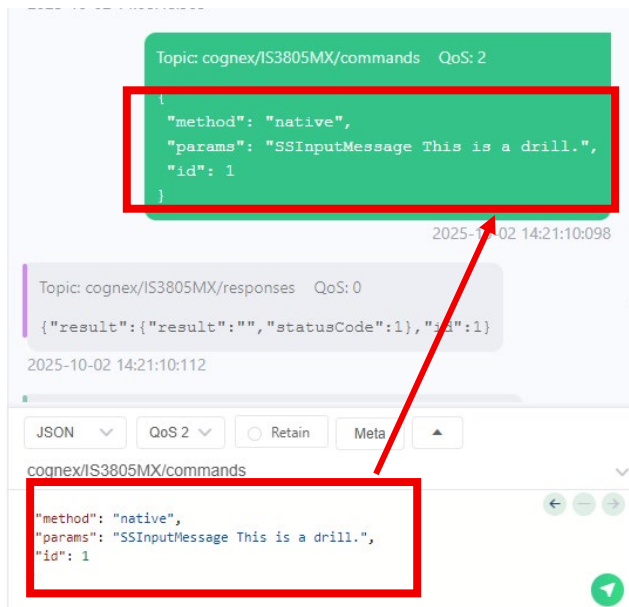


From here, you can trigger the camera and see the response in the results topic get published by the camera. This is formatted as a JSON string.

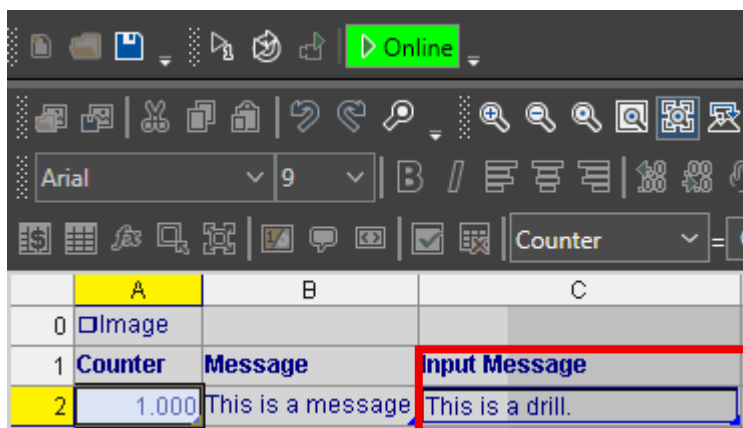


You can write to certain cells using the command topic and using the set value commands. For the **InputMessage**, we will use the SetString command. Enter the following to be published to the commands topic:

```
{
  "method": "native",
  "params": "SSInputMessage This is a drill.",
  "id": 1
}
```



You should now see your cell value update with the message you sent.



If you want to retrieve the value of a particular cell without getting the full results after an inspection, you can use the **Get Value** native mode command and reference the cell name or the cell address. For example, if we want to get the info from the **OutputMessage** cell, publish the following to the command topic.

```
{
  "method": "native",
  "params": "GVOutputMessage",
  "id": 1
}
```



If you use the cell address, type the following in the command topic.

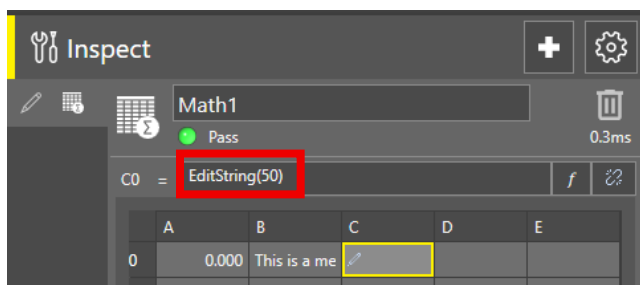
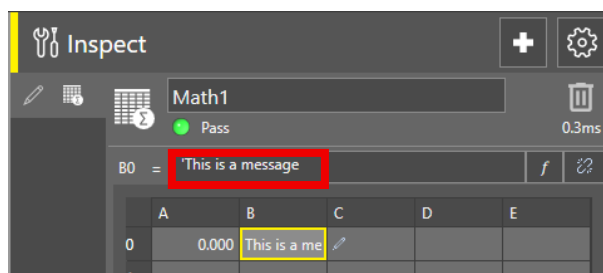
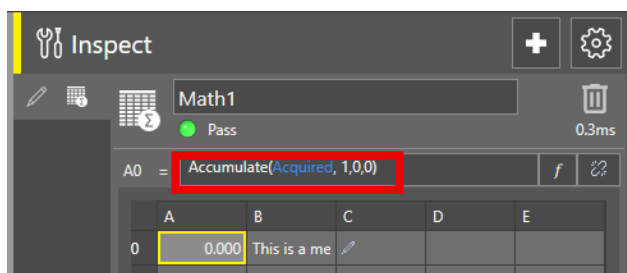
```
{
  "method": "native",
  "params": "GVB2",
  "id": 1
}
```



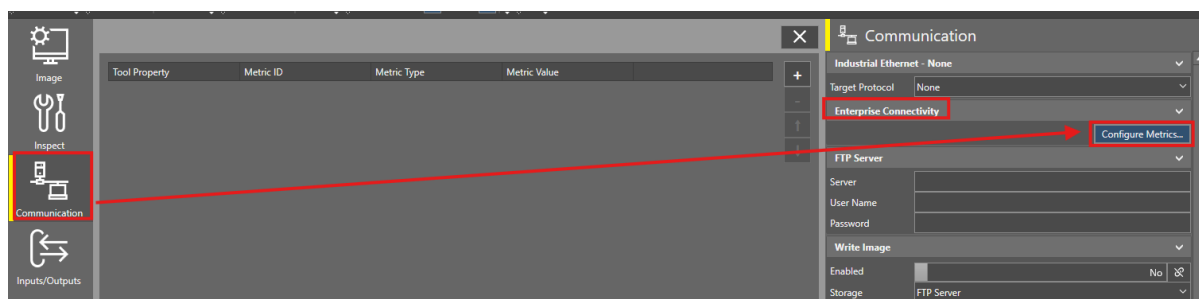
1.1.3 EZBuilder Metrics Set Up

To set up metrics in EZBuilder, set up your inspection tools. In this example we are using a Math and Logic tool with the following cells.

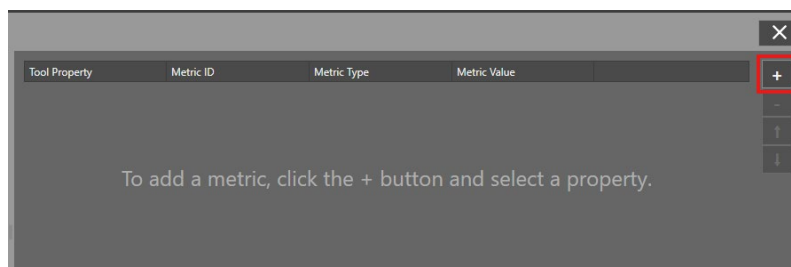
Location	Value	Description
A0	Accumulat(acquired, 1,0,0)	Counts for every trigger
B0	'This is a message	String comment
C0	EditString(50)	User editable comment up to 50 characters



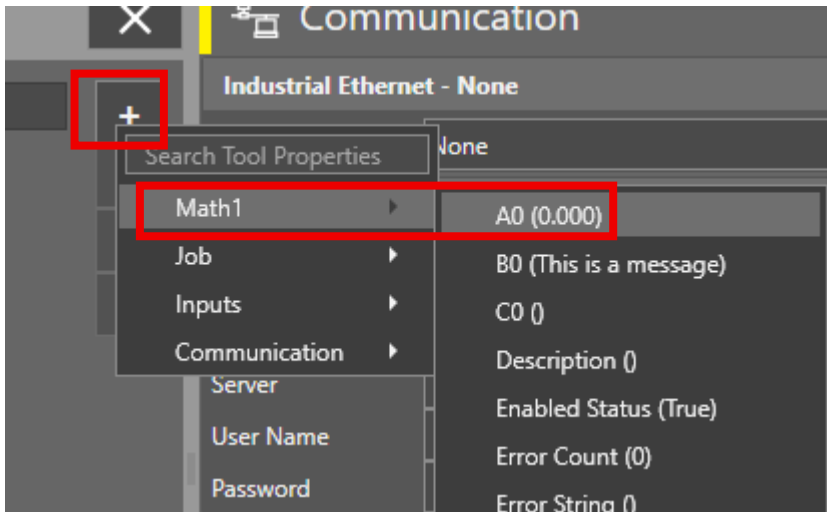
Now that we have these values set up, we can proceed to add them as metrics. To do this, go to the **Communication** section and under **Enterprise Connectivity**, click **Configure Metrics**.



This will load the following page. To add a metric, press the + icon at the top-right.



From here, we can see the Math tool we setup earlier as well as other inspection properties. For now, we will add the three cells we setup earlier in the Math and Logic tool (**Math1**) by selecting them in the list one at a time.



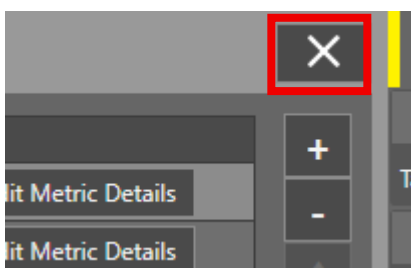
Tool Property	Metric ID	Metric Type	Metric Value	
Math1.A0	Math1.A0	Text	0	Edit Metric Details
Math1.B0	Math1.B0	Text	This is a message	Edit Metric Details
Math1.C0	Math1.C0	Text		Edit Metric Details

For each metric added, you will have a **Metric ID**, **Metric Type**, and **Metric Value**. By default, the Metric ID will be the ToolName.PropertyName. You can change this as needed. The **Metric Value** will display the current value of the property. There is a button labeled **Edit Metric Details**. This allows you to add a description of the metric if needed. If the metric type is a measurement, you can also set the **Range Low** value, **Range High** value, and **Unit** type in this window as well.

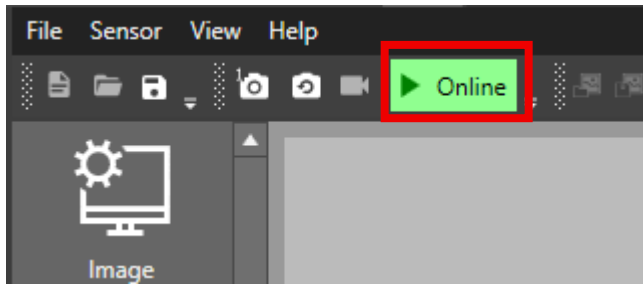
We will configure the metric IDs to have more friendly names and set the Count property to have a counter metric type.

Tool Property	Metric ID	Metric Type	Metric Value	
Math1.A0	Count	Counter	0	Edit Metric Details
Math1.B0	OutputMessage	Text	This is a message	Edit Metric Details
Math1.C0	Input Message	Text		Edit Metric Details

With these set up we can close this window and set the camera online.



The camera will need to be online for the MQTT communications to work.

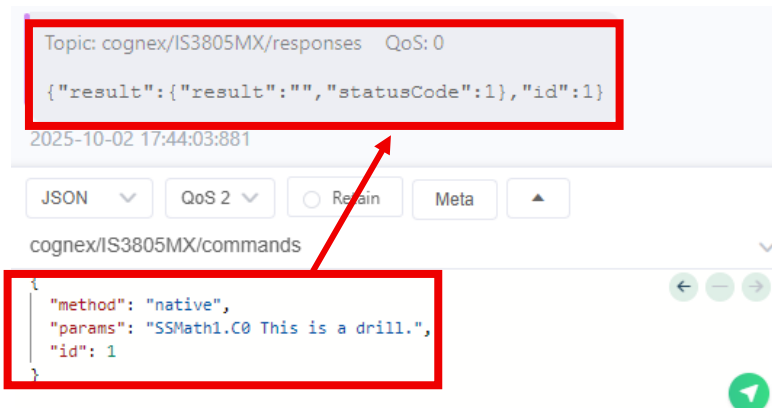


From here, you can trigger the camera and see the response in the results topic get published by the camera. This is formatted as a JSON string.



You can write to certain cells using the command topic and using the set value commands. For the **InputMessage**, we will use the SetString command. Unlike Spreadsheet, we will have to reference particular cells/properties by their ToolName.PropertyName. For instance, for the InputMessage that we setup in our Math and Logic tool, we'll use the following for the property name, "**Math1.C0**" followed by the string we want to set. Enter the following to be published to the commands topic:

```
{
  "method": "native",
  "params": "SSMath1.C0 This is a drill.",
  "id": 1
}
```



If you want to retrieve the value of a particular property without getting the full results after an inspection, you can use the **Get Value** native mode command and reference the ToolName.PropertyName. For example, if we want to get the info from the **OutputMessage** cell, publish the following to the command topic.

```
{
  "method": "native",
  "params": "GVMath1.B0",
  "id": 1
}
```

