



SUPPORTING DOCUMENT

# Third-Party Test Results

Report For: [NAME OF THIRD PARTY]

Version: 00.01.00  
Date: August 2018

## Version History

Date	Version	Author(s)	Notes
15/08/2018	00.00.01	Caitlin Randles	Initial Creation

## Final Approval

Approval Type	Date	Version	EA Technology Issue Authority

CONFIDENTIAL - This document may not be disclosed to any person other than the addressee or any duly authorised person within the addressee's company or organisation and may only be disclosed so far as is strictly necessary for the proper purposes of the addressee which may be limited by contract. Any person to whom the document or any part of it is disclosed must comply with this notice. A failure to comply with it may result in loss or damage to EA Technology Ltd or to others with whom it may have contracted and the addressee will be held fully liable therefor.

Care has been taken in the preparation of this Report, but all advice, analysis, calculations, information, forecasts and recommendations are supplied for the assistance of the relevant client and are not to be relied on as authoritative or as in substitution for the exercise of judgement by that client or any other reader. EA Technology Ltd. nor any of its personnel engaged in the preparation of this Report shall have any liability whatsoever for any direct or consequential loss arising from use of this Report or its contents and give no warranty or representation (express or implied) as to the quality or fitness for the purpose of any process, material, product or system referred to in the report.

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means electronic, mechanical, photocopied, recorded or otherwise, or stored in any retrieval system of any nature without the written permission of the copyright holder.

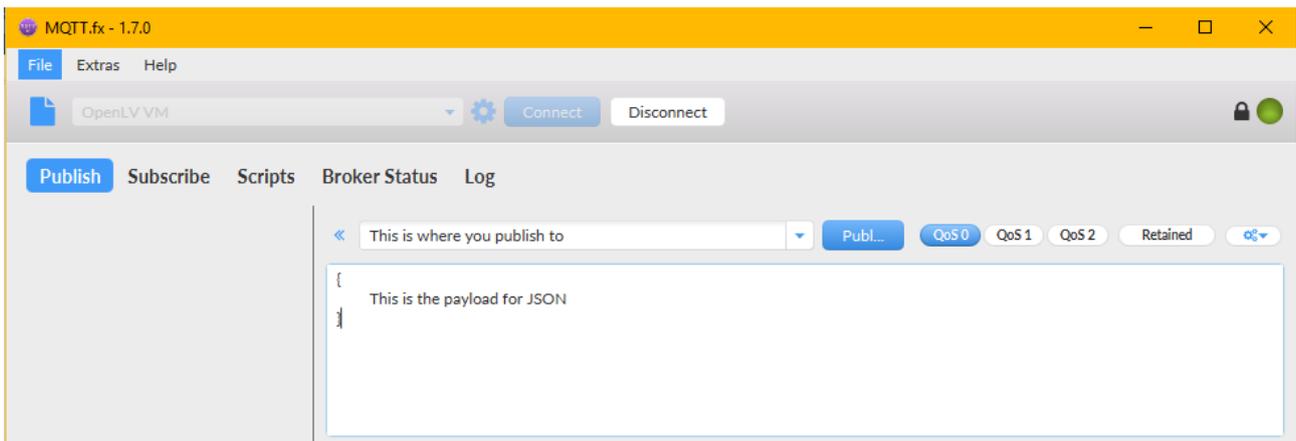
© EA Technology Ltd August 2018

## Summary

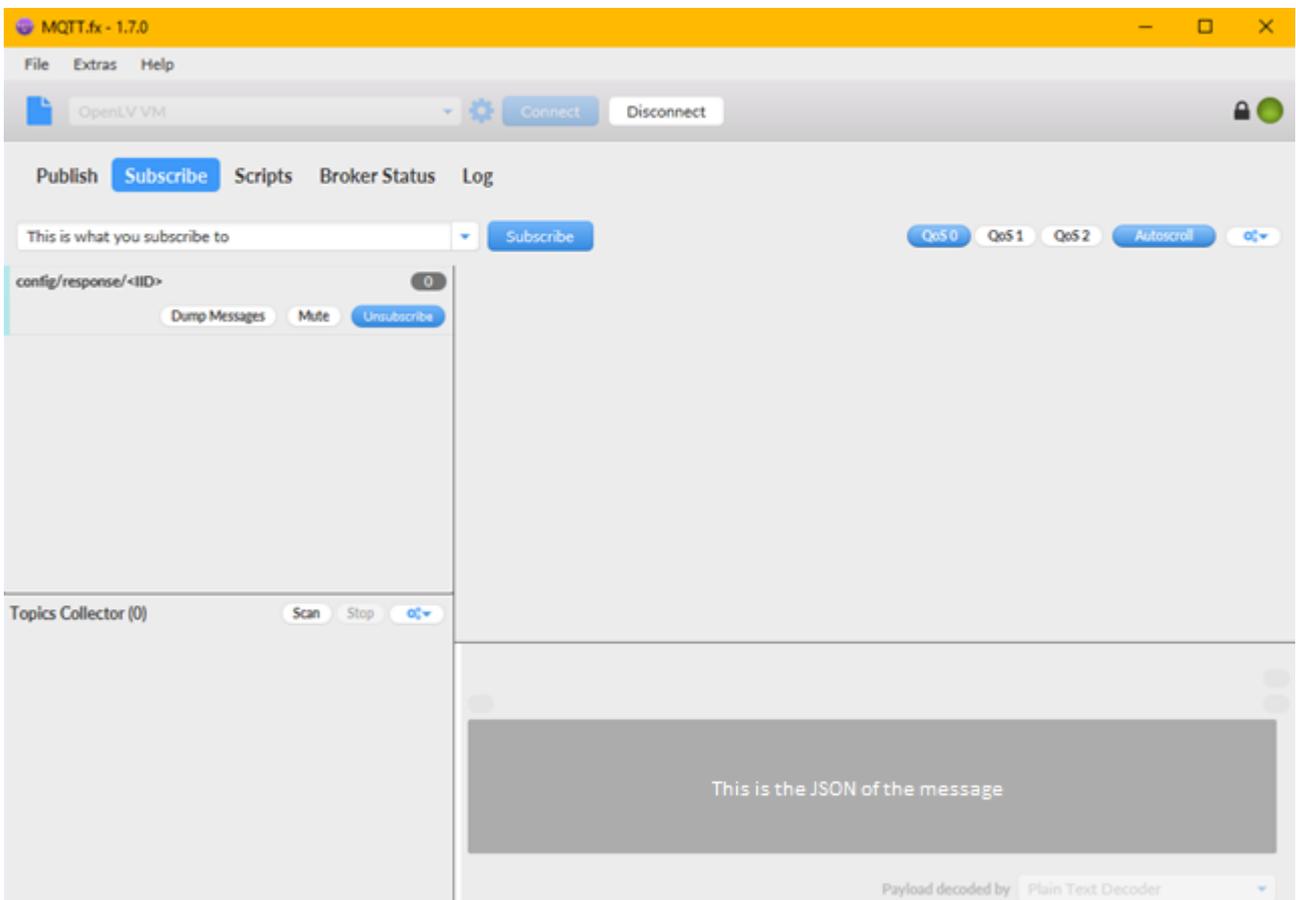
This document is to be used in conjunction with *E717-TSTDC-S004-V00.00.01 OpenLV Third Party Application Testing Document*. The afore mentioned document is primarily for internal testing and has been provided by EA Technology as it offers more detail and transparency into the failures and successes of the application.

## Definitions

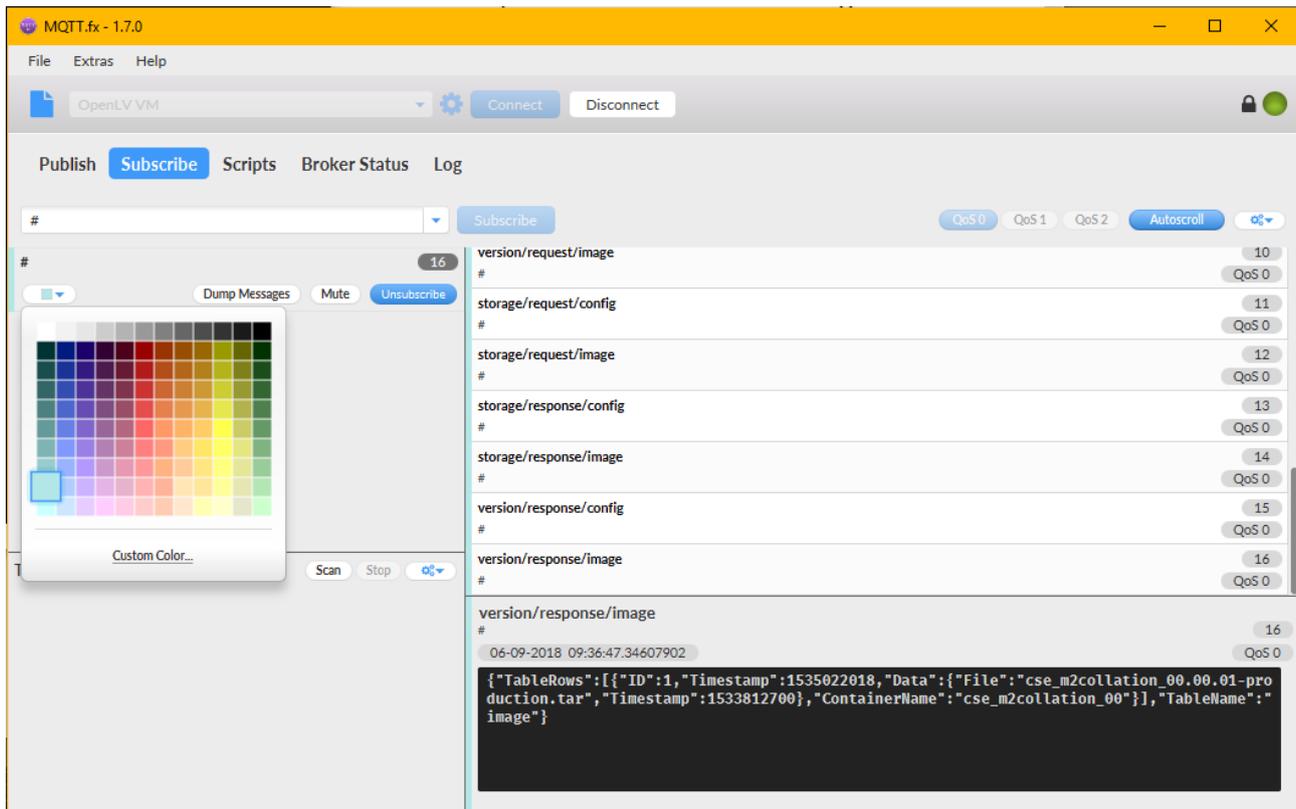
### Publish tab:



### Subscribe tab:



## Colour Coding



## Additional Testing Info

Access the acl but using the command `sudoedit $(docker inspect dd7e7e53-7de2-456e-9e9b-01a136f1cd84 | jq -r '.[0].Mounts[] | select(.Destination == "/etc/mosquitto") .Source')/containers.acl` and scroll down to the second bottom set of rules. Add to the list "topic read #" and "topic write #" as shown below:

```
# Rules for development tools certificate set
# These rules give limited read access for testing, but no write access
user eatl_tlsdevtools
topic read  algorithm/data/#
topic read  sensor/data/#
topic read  status/response/#
topic read  #
topic write #
```

## Contents

1.	Environment .....	4
2.	Config .....	5
3.	Images.....	8
4.	Status.....	9
5.	Configuration.....	11
6.	Command .....	13
7.	Errors.....	14
8.	Publish and Subscribe .....	15
9.	Documentation .....	17
10.	References .....	18

# 1. Environment

Please note that the Environment test will be performed without exception. The restrictions have been imposed due to a variety of constraints and it is vital that Third Party applications conform to this. At this stage in the process there is little expectation of applications requiring more than the already allocated resources for individual projects.

The checks for the following steps can be done as followed:

- Compressed image file size <100Mbytes – The original file size we will receive. In Windows you may observe the size column in the UI. In Linux you may run the command **ls -l filename** . This is advised to be done in Windows as that is what will be performed in EA Technology testing.
- Configuration file size <1MB – From the home/development directory, type the command **ls -hl** and look for the .json file. This will give you the configuration file size. (Please see the highlighted line below). The JSON file you want is the JSON file name with your IID. The unit of measurement if there is no indicator is in bytes.

```
development@openlv_20180815p:~$ ls -hl
total 11M
-rw-r--r-- 1 development development 7.7M Aug 15 12:06 2358-SWREL-S012-V01.06.02_LV-CAP_core_production_dummy_comms.tar.xz
drwxrwxr-x 4 development development 4.0K Aug 23 12:24 2746-SWREL-S011-V01.04.00-production_eatl-tlsdevtools
-rw-r--r-- 1 development development 115K Aug 15 15:22 2746-SWREL-S011-V01.04.00-production_eatl-tlsdevtools.tar.xz
-rw-rw-r-- 1 development development 516 Aug 23 11:59 75e81145-e85f-42ff-b992-d9d12c865c0e.json
-rw-rw-r-- 1 development development 2.4M Aug 23 11:50 cse_m2collation_00.00.01-production.tar
-rw-rw-r-- 1 development development 639 Aug 23 11:50 cse_m2collation_00.json
```

## 2. Config

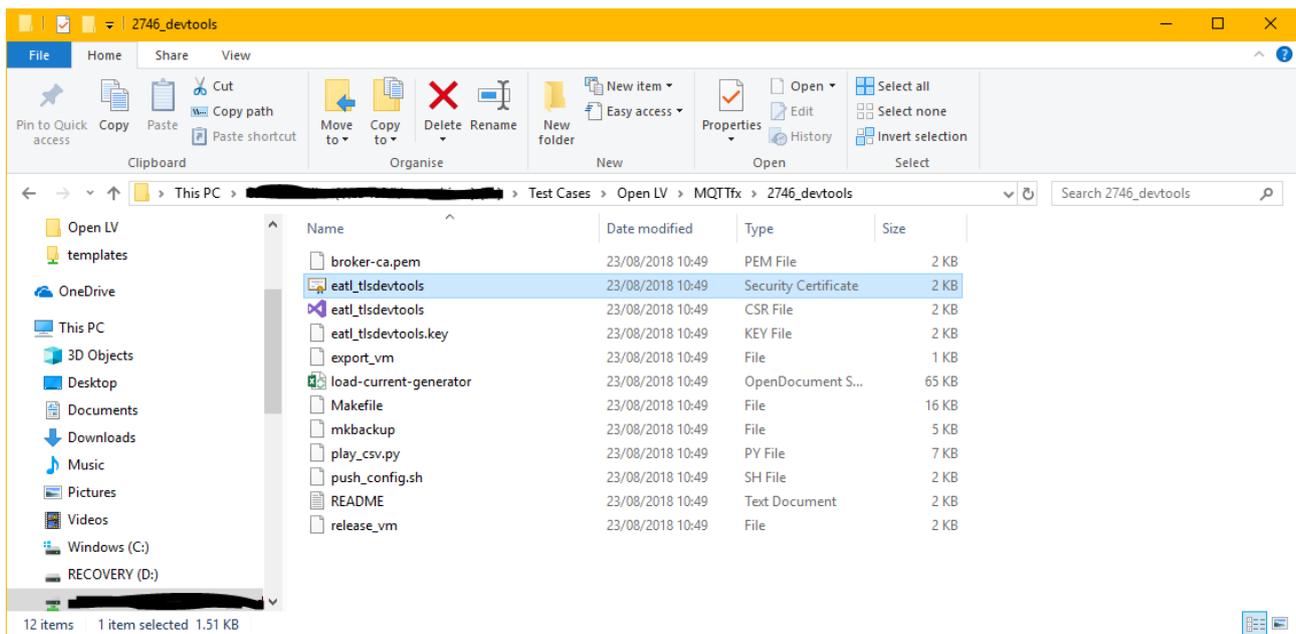
Please note that the config test is mainly for your internal testers if you wish to perform these tests before we do. This test contains instructions for your testers in simple terms confined to one document, as is the main methodology we will be using ourselves. It is important however that you pass the following from Config:

- Application runs
- Certificate is included (see below)
- Ensure all documents required to be created have been completed and correctly
- Test data is supplied
- Ensure playback is configured

If you require any more information regarding the configuration for testing, please refer to the *2622-MANUL-S001-V02.03.04 Developing for LV-CAP using the Virtual Machine*. If the image is being put on the machine for the first time, follow sections 1, 2, 3, 4, 5, 8, 9. A certificate should already have been provided for testing.

Ensure all environment and config has been completed before attempting the following tests.

### Checking the Certificate



### Summary of running the Virtual Machine

1. Download [Oracle's Virtual Machine](#). Please note, VirtualBox only requires to be downloaded once
2. Download the LV\_CAP\_xxx\_yyy\_zzz.ova file
3. Launch the Virtual Box
4. Select File > Import Appliance
5. Select the .ova file listed in step 2
6. Select "Next" button
7. Check the "Reinitialize the MAC address of all network cards" text box
8. Select Import
9. User is taken to the initial VirtualBox manager, where the environment will be listed on the left hand side of the application widget
10. Right-click on the environment and select settings
11. Select "OK"

*For more detailed information see Section 3, "Developing with the LV-CAP Virtual Machine"*

### Summary of accessing the Virtual Machine

1. To start the VirtualBox, select the environment and then "Start"
2. If necessary to close the environment, try to avoid "Power Off the Machine", as this may cause data loss
3. Command line is displayed. Please note there is no GUI for this environment
4. User is prompted for username, the username is "development"
5. User is prompted for password, the password is "OpenLVdevelopment"
6. User will be asked to change the password if this is the user's first time accessing the VirtualBox. Enter the password "OpenLVdevelopment" again and then enter your new password, twice
7. User's password has been changed successfully
8. Run the command " ip addr sh enp0s3 "
9. Take note of the IP address (value following inet)
10. Access MobaXTerm and connect using the IP address discovered to the SSH option and log in using the created credentials

*For more detailed information see Section 4, "Developing with the LV-CAP Virtual Machine"*

**Please include the version (file name) of the Virtual Machine used for testing:** \_\_\_\_\_

### Summary of running an Application

1. Using MobaXTerm, copy the Third Party files onto the VirtualBox by selecting "Upload" from the icons above the files (e.g. boot)
2. Unpack the .tar files by using the command " tar -xf file.tar.xz "
3. To authorise the application, run the following commands:
4. Inspection of the Data Marketplace to determine where configuration files are stored, please note file paths will vary from machine to machine: `docker inspect dd7e7e53-7de2-456e-9e9b-01a136f1cd84 | jq '[0].Mounts[] | select(.Destination == "/etc/mosquitto") .Source" '`
5. Edit the container.acl, located in the directory obtained by 3a. by running the command: `sudoedit $(docker inspect dd7e7e53-7de2-456e-9e9b-01a136f1cd84 | jq -r '[0].Mounts[] | select(.Destination == "/etc/mosquitto") .Source" ')/containers.acl`

The application should contain all essential properties by this point to run.

*For more detailed information see Section 5, "Developing with the LV-CAP Virtual Machine"*

## Summary of accessing MQTT.fx

1. Download [MQTT.fx](#). This application must be used on a PC or laptop (it is not possible to run it on the virtual machine, you access the VM with the IP)
2. Ensure the devtools.tar.xz is unpacked by using " tar -xf ...devtools.tar.xz "
3. Publish is sending the requests and Subscribe is viewing any requests or responses
4. Ensure the IP found in Accessing the VirtualBox is used to connect to MQTT.fx by entering the IP into Broker Address
5. Give a suitable Profile name, Broker Port 8883 and use eatl\_tlsdevtools as the client ID
6. Access the General Tab and set the values as follows: 30 (connection timeout), 60 (keep alive interval), ticked (clean session) and checked (Use Default MQTT.fx version)
7. Navigate to the user credentials tab and set the user name as eatl\_tlsdevtools
8. Navigate to the SSL tab and enable SSL
9. For the CA file select the broker-ca.pem file, for Client certificate file select eatl\_tlsdevtools.crt, for client key file select eatl\_tlsdevtools.key, leave password black and tick the PEM formatted box
10. Select OK and connect
11. In the command line run the following command:

*For more detailed information see Section 8.2, "Developing with the LV-CAP Virtual Machine"*

Please note that MQTT.fx is not required to run OpenLV solutions. This is a GUI for those who would rather use an interface of some kind as opposed to the command line. These instructions are specific to those who wish to use MQTT.fx only.

1. The following command will only work when run from inside the directory created by unpacking the devtools archive 2746-SWREL-S011-VXX.YY.ZZ-production\_LVCAP\_TLS\_devtools/2746\_devtools/ (add detail how to unpack, linux command etc)

2. Run the following command into the container:

```
./play_csv.py --host marketplace --port 8883 --cafile broker-ca.pem --cert eatl_tlsdevtools.crt --key eatl_tlsdevtools.key -i input-1day-w5d1.csv -f [maximum real-time playback value defined by third parties] (
```

3. Run the application for the maximum length of time possible which will be defined by the Third-Party [INSERT HERE]

if run from inside the directory created by unpacking the devtools archive 2746-SWREL-S011-VXX.YY.ZZ-production\_LVCAP\_TLS\_devtools/2746\_devtools/ is the name of the file to play back, and it must be located inside the directory created by unpacking the devtools archive 2746-SWREL-S011-VXX.YY.ZZ-production\_LVCAP\_TLS\_devtools/2746\_devtools/

### 3. Images

Type in the following command to view the size of the image: ***docker images <Docker ID>***

There is no need to include your version number for this. Please note that underscores will be replaced by forward slashes. For example: eatl\_utility will become eatl/utility. Here is an example:

```
development@openlv_20180815p:~$ docker images cse/m2collation
REPOSITORY          TAG                IMAGE ID           CREATED           SIZE
cse/m2collation     00.00.01-production f5c961dd02e5     3 weeks ago     8.32 MB
```

If you are unable to do this then you are welcome to use the command ***docker images*** and manually search for your image, however this may lead to human error. Here is an example of what will be shown if you do not filter the results, however please note there can be many more lines than listed:

```
development@openlv_20180815p:~$ docker images
REPOSITORY          TAG                IMAGE ID           CREATED           SIZE
lucapcore/datastorage 04.01.00-production 155f735289ab     2 weeks ago     9.4 MB
nortech/commscontainer 01.02.00-production 9c55804950ba     2 weeks ago     8.23 MB
lucapcore/marketplace 01.06.00-production 324d0848fb19     2 weeks ago     4.33 MB
cse/m2collation     00.00.01-production f5c961dd02e5     3 weeks ago     8.32 MB
<none>              <none>            4b585a899765     2 months ago     8.23 MB
<none>              <none>            73530b62793e     2 months ago     9.4 MB
lucapcore/marketplace 01.05.00-production 389221734396     2 months ago     4.33 MB
nortech/commscontainer 1.1.0-production 7374c38389a3     9 months ago     9.64 MB
```

As you can see, there are many more results and it is better to filter them. Regardless, the value you want will always be in far right-hand side column labelled "SIZE". This will indicate the image size. The size must be <500Mbytes.

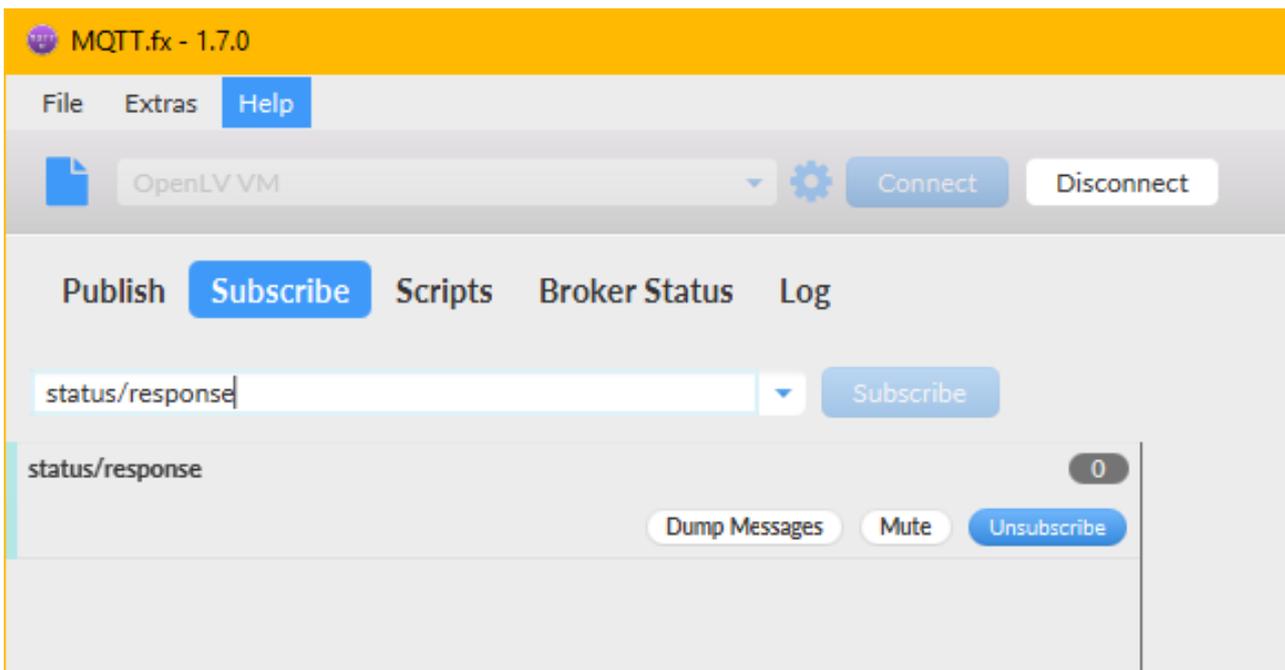
If you were unable to filter the command, a check will undergo that the Docker ID matches the file name. A check will still be performed if otherwise, however it is unlikely in this instance.

Then you need to observe the second column titled TAG, which should match your image version.

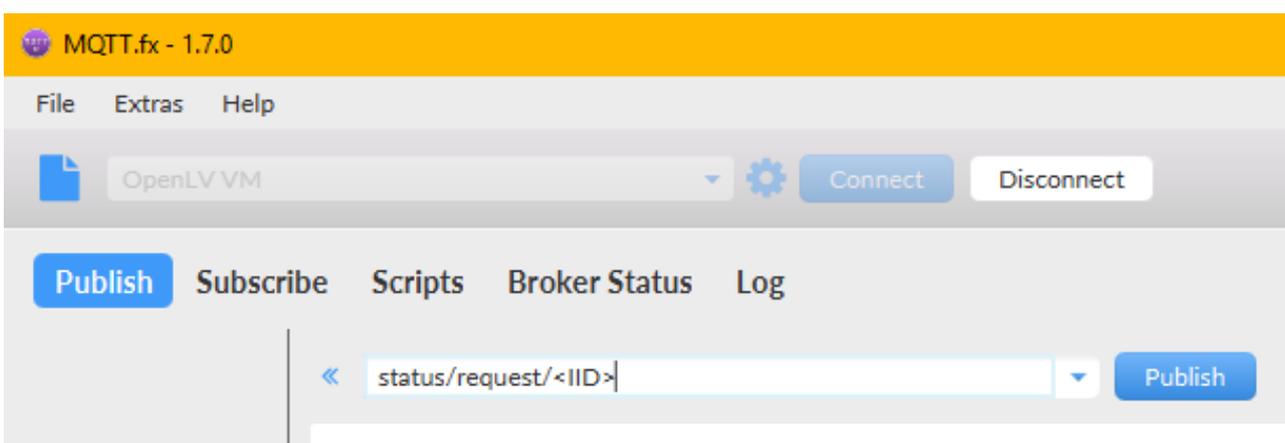
## 4. Status

The application should always respond to status requests. The application should also not arbitrarily send status responses. If the container does not respond to two status messages in a row, the application will be automatically restarted. If this continually happens the application will never run, however this should not happen even as an occasional one off. If the occasional first message is not responded to, but the second is, that is fine. This, however, should not be consistently so.

Ensure you are following these steps in the order they are presented. Navigate to the Subscribe tab and subscribe to status/response like the following:



Publish to status/request/<IID> as demonstrated below and select publish, keep the payload empty:



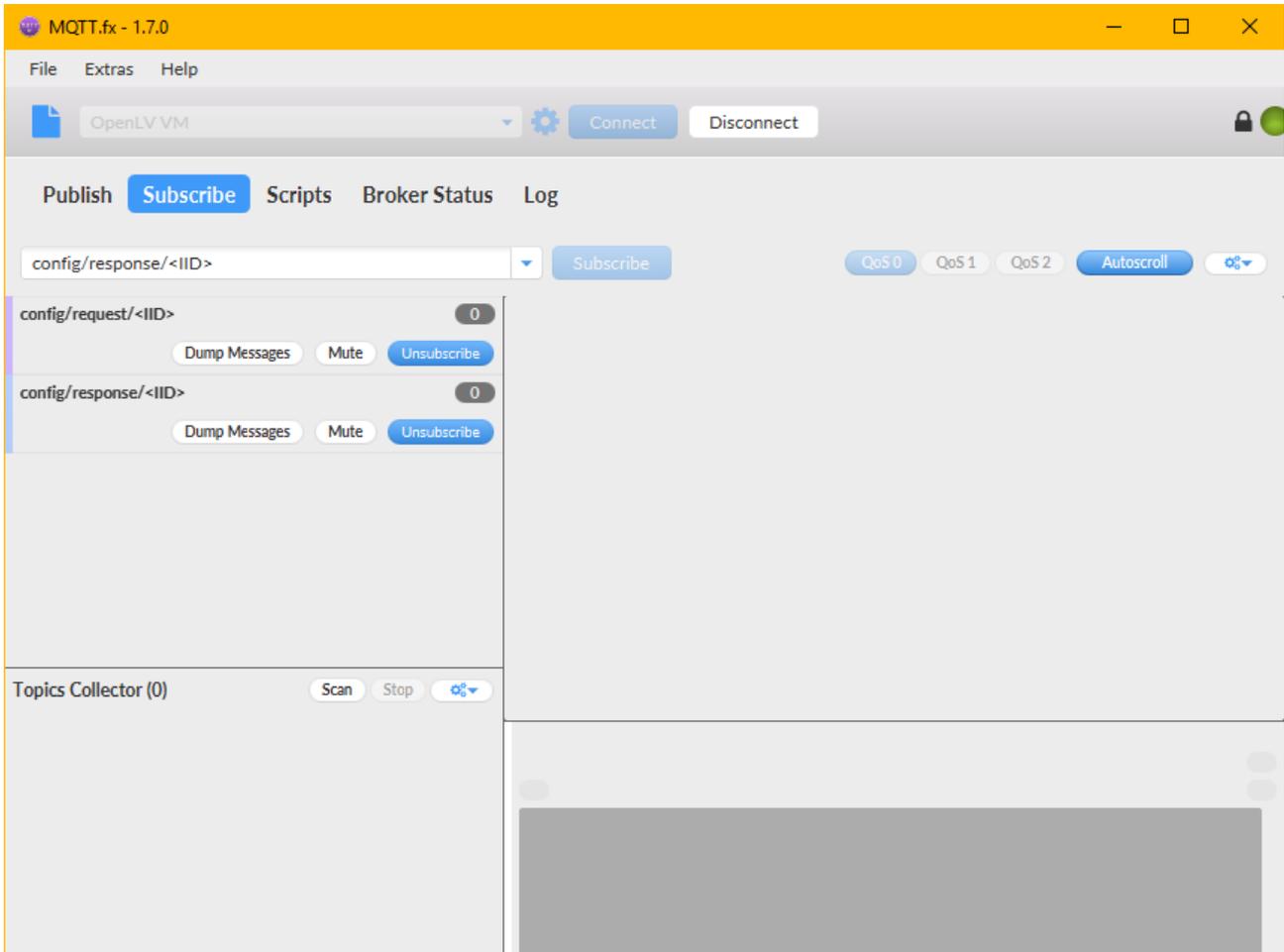
The newly subscribed topic will display in the subscribe tab, and you can subscribe to multiple topics at once. If you do, it is advised to colour code the subscriptions to tell them apart more efficiently. You can subscribe to both the request at response, however only publish to one. This feature is not necessary for your testing but is useful information if you are comfortable block testing. We do, however, advise against this.

This is how the response will look:

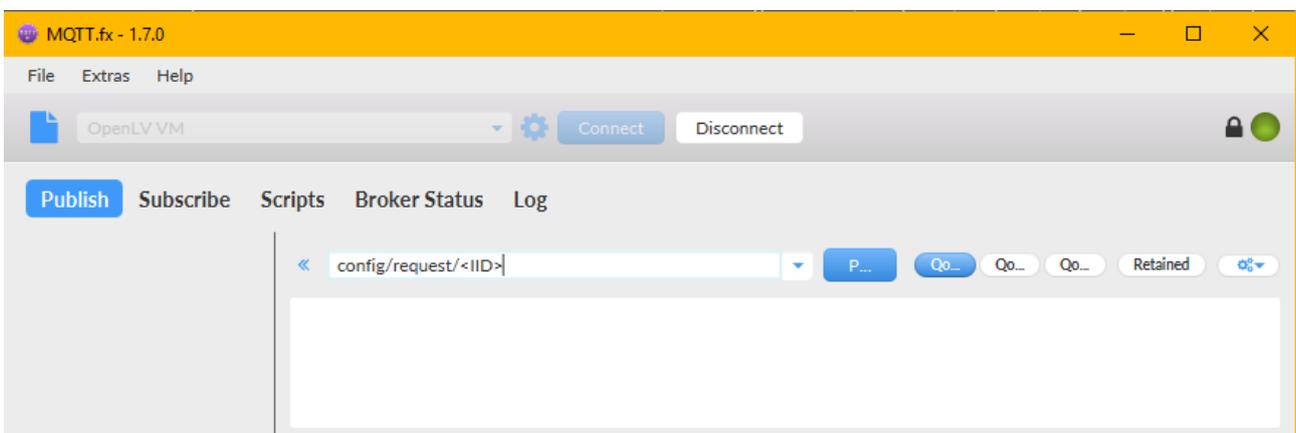
The screenshot shows a MQTT client interface with a top navigation bar containing 'Publish', 'Subscribe', 'Scripts', 'Broker Status', and 'Log'. Below this is a search bar with a '#' icon and a 'Subscribe' button. On the right side, there are controls for 'QoS 0', 'QoS 1', 'QoS 2', 'Autoscroll', and a refresh icon. The main area is divided into two panes. The left pane shows a list of subscribed topics, including 'status/response', with controls for 'Dump Messages', 'Mute', and 'Unsubscribe'. The right pane shows a list of messages with their respective topics and QoS values. The selected message is from the topic 'status/response/cse\_m2collation\_00' with a QoS of 0. The message payload is shown in a dark box with the JSON: `{"Status":1,"Message":""}`. At the bottom right, there is a 'Payload decoded by' dropdown menu set to 'Plain Text Decoder'.

## 5. Configuration

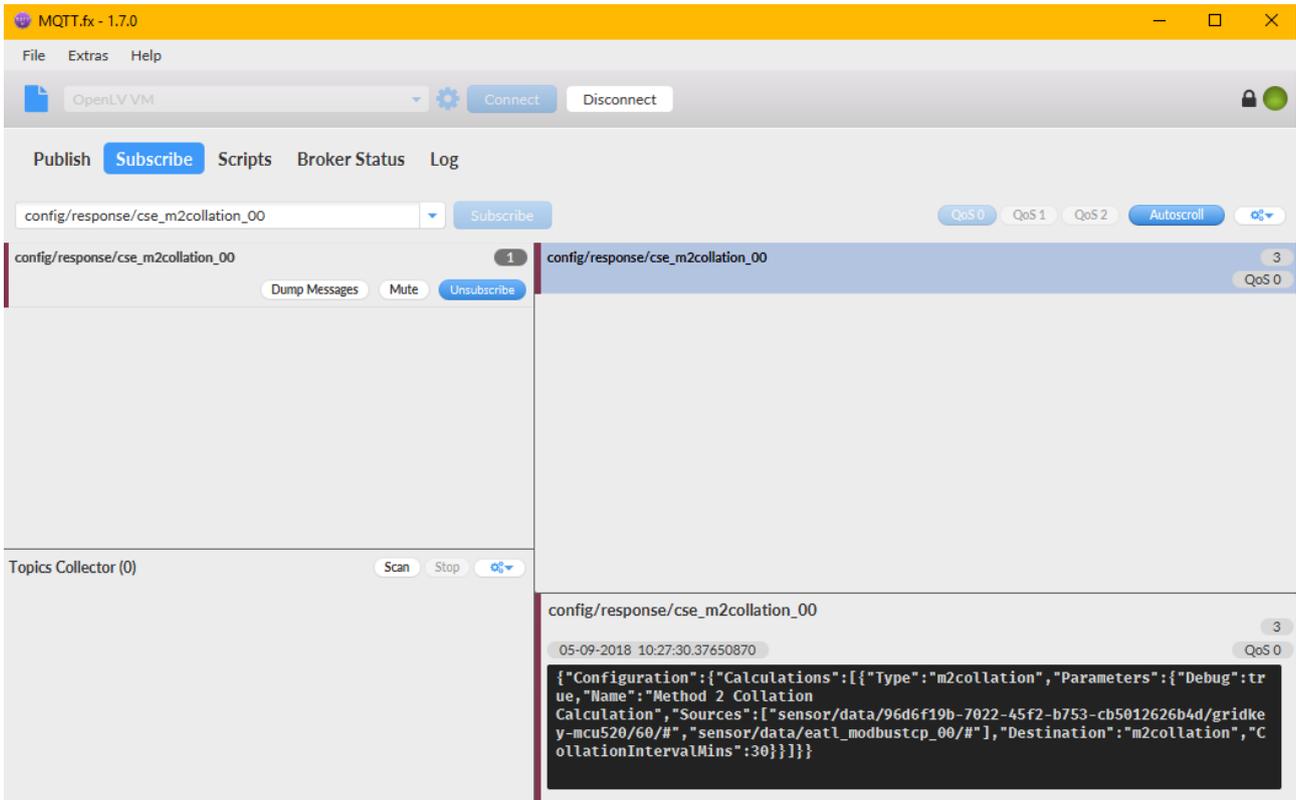
Ensure you are following these steps in the order they are presented. Here is how to subscribe to configuration messages:



Leave the payload empty and publish to the following:



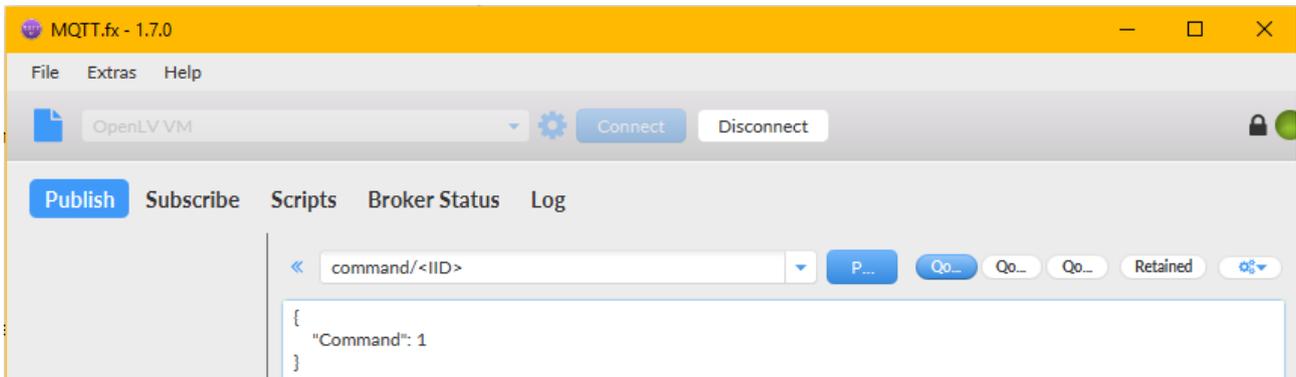
Here are what the results from the earlier config steps should look like with varying payloads depending on your configuration.



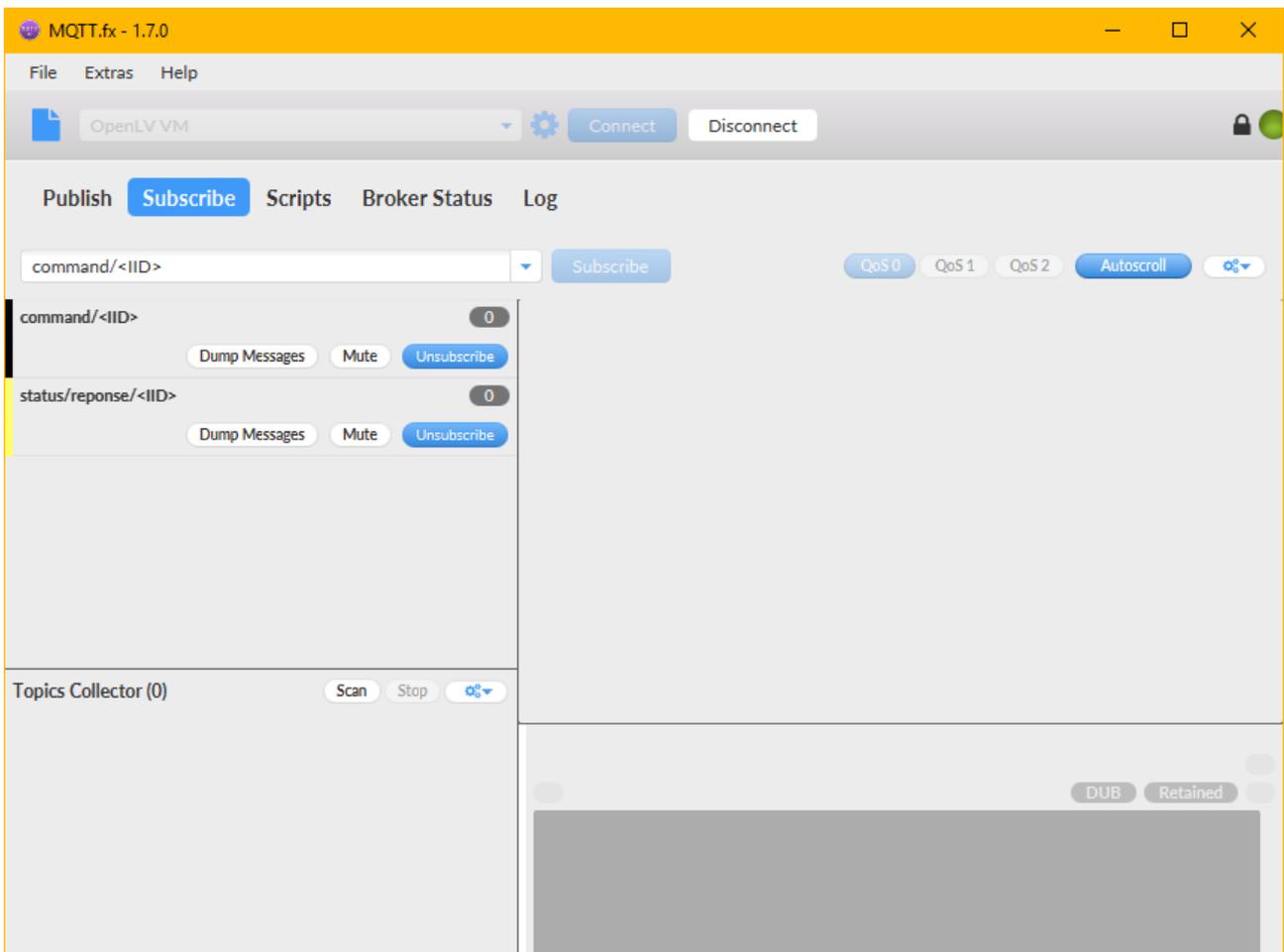
## 6. Command

Use the steps from section 4 and make note of the JSON output message.

Publish

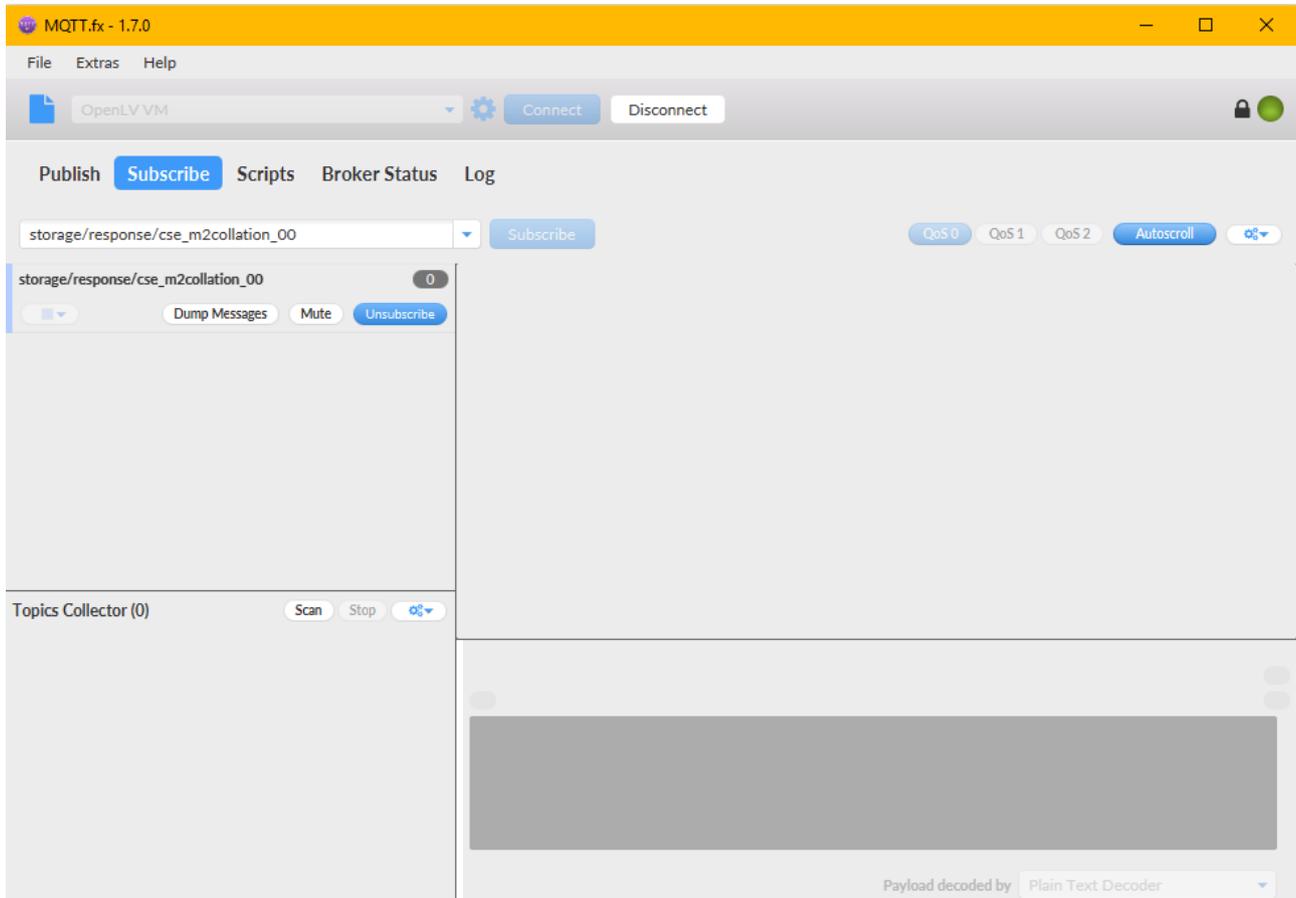


Subscribe



## 7. Errors

There Are two methods to Error checking. The first can be performed by anyone, however, must be completed throughout the entirety of communications with MQTT.fx. This is a fair test as no previous errors will be loaded and timestamps will not have to be checked. However, you want to ensure that all possible tests will be performed whilst subscribing to this.



As you can see there are no errors published here and this would be correct. For accurate results, you should test your storage data, algorithm data, status response, config data, command response etc. to ensure no aspect of your application throws any errors. The colour has not been changed here, however it is advised to make the colour for this a black block or bright red, so it stands out.

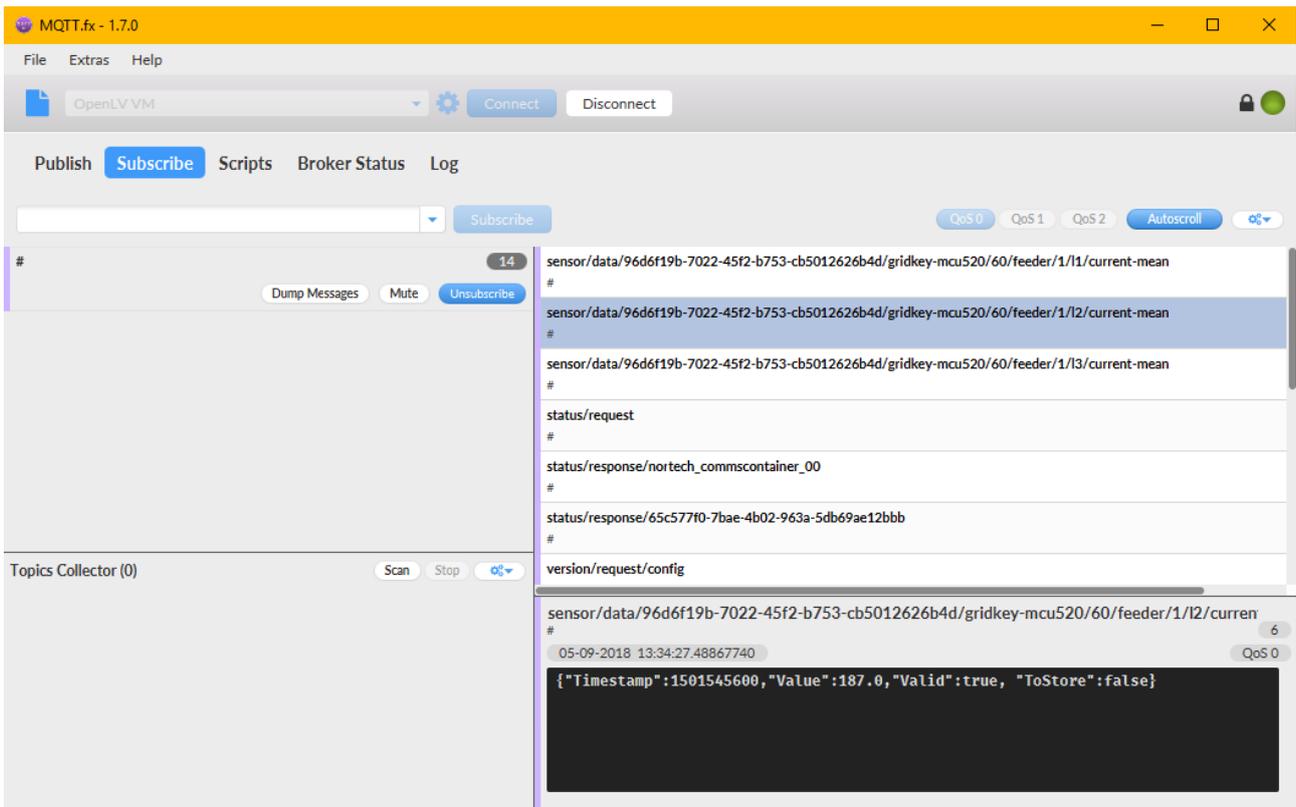
## 8. Publish and Subscribe

This section merely ensures that all the data provided to EA Technology is correct and the application does as was described in the Application Form. These will vary from application to application as the output will be different. Regardless, it cannot be too frequent. We will select a total of five algorithm and sensor topics to analyse and determine if they are suitable, both in frequency and accuracy.

This example of algorithm data, the JSON payload and brief description of what the function does.

algorithm/data/cse_m2collation_00/m2collation/feeder/1/l1/current-mean algorithm/#	14 QoS 0
algorithm/data/cse_m2collation_00/m2collation/feeder/1/l2/current-mean algorithm/#	15 QoS 0
algorithm/data/cse_m2collation_00/m2collation/feeder/1/l3/current-mean algorithm/#	16 QoS 0
algorithm/data/cse_m2collation_00/m2collation/feeder/1/l1/current-mean algorithm/#	14 QoS 0
07-09-2018 09:46:16.35176074	QoS 0
<pre>{"Timestamp":1501545600,"Interval":1800,"StartPoint":1501545600,"Value":[100,100,100,0,1], "Valid":true,"ToStore":true}</pre>	

### Example of sensor data



[Additional specific to third party info here]

## 9. Documentation

All applications submitted to EA Technology should come with the following:

- A user guide, containing any configuration requirements, input requirements, expected outcomes, run duration and a JSON Configuration example. This can be in any format you see fit if it contains this information and is presented well
- Real test data in the correct csv format
- Completed application form

Any application lacking this information will be required to submit it as we expect to have this information by the point of submission.

Please attach this separately from the application.

## 10. References

- [1] E717-TSTDC-S004-V00.00.01 OpenLV Third Party Application Testing Document;
- [2] 2622-MANUL-S001-V02.03.04 Developing for LV-CAP using the Virtual Machine;
- [3] 2383-MANUL-V04.03.04 LV Common Application Platform Public API