MQTT Extensions 1.0

In this section:

About the Extensions

The MQTT Extensions are custom extensions for Parasoft SOAtest and Virtualize. You can use the extensions on the client side to perform many tasks:

- Publish to a topic
- Subscribe to a topic filter until the desired number of messages are received or a specified duration elapses
- Monitor a topic filter during test execution to perform validations on the events that occur

On the server side, you can use the listener extension to subscribe to a topic filter then publish to a topic based on the incoming subscription messages. The following custom extensions are provided for MQTT integration:

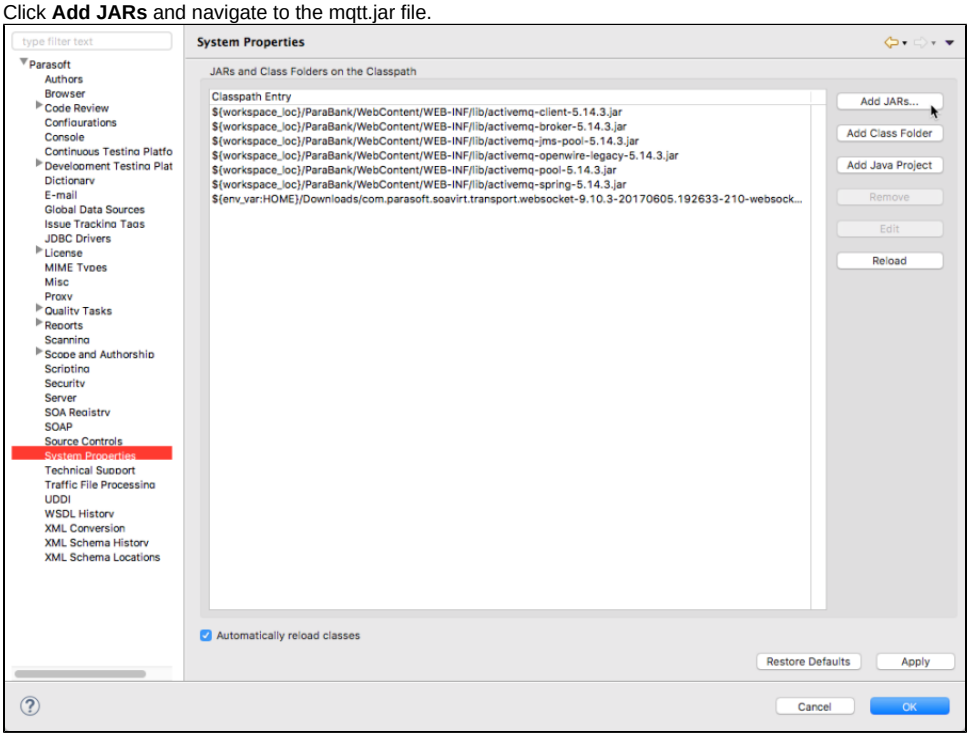
- MQTT Transport: A transport protocol for publishing MQTT messages.
- MQTT Subscriber: Use to subscribe to a topic filter until the specified duration elapses.
- MQTT Message Listener: A message listener used to subscribe to a topic filter and publish virtualized messages.
- MQTT Event Monitor: An entry point to the event monitoring interface, subscribes to a topic to report message events received during testing.
- MQTT Configure Tool: A simple tool that configures different MQTT properties so they do not have to be configured in every MQTT extension within a test.

Installation

This tool can be installed from the UI or command line. Download and unzip the compressed folder in a convenient location prior to starting the installation process.

UI Installation

1. Choose **Parasoft> Preferences** and click **System Properties** in the sidebar menu.
2. Click **Add JARs** and navigate to the mqtt.jar file.



3. Restart SOAtest.

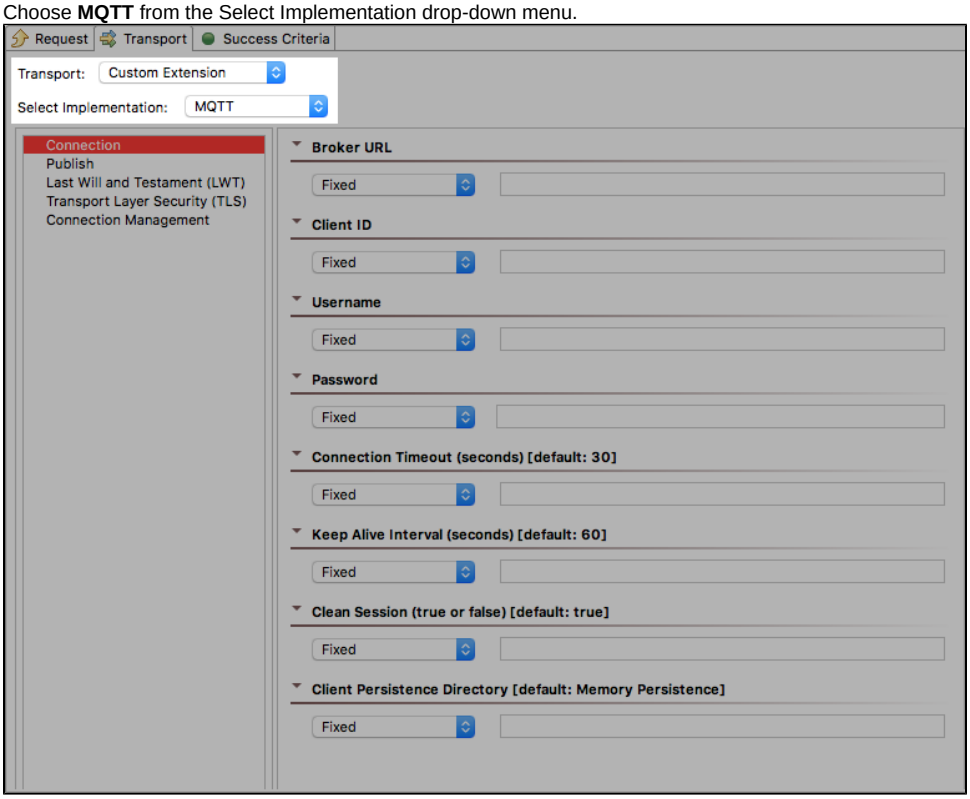
Command Line Installation

Add the mqtt.jar file to the **system.properties.classpath** property in your local settings properties file. For example:

```
system.properties.classpath=<path to jar>/mqtt.jar
```

MQTT Transport

1. Add a Message Client to a test and switch the transport to **Custom Extension**.
2. Choose **MQTT** from the Select Implementation drop-down menu.



Configuration

You can configure the following settings.

Connection Settings

Click the Connection tab to configure the connection settings.

Broker URL	Defines the URL of the MQTT broker in the following format: **protocol://host:port**. • **Protocol** is the connection protocol. The transport can communicate with MQTT brokers over TCP or WebSocket protocols. Acceptable values for the protocol are **tcp**, **ssl**, **ws**, or **wss**. • **Host** is the broker's host. • **Port** is the broker's port. **This field is required**. If the defined protocol is **ssl** or **wss**, the Transport Layer Security (TLS) settings must also be configured.
Client ID	Defines the client ID to use when connecting to the broker. If left blank, the client ID will be generated by the MQTT broker.
Username	Defines the username to use when connecting to the broker.
Password	Defines the password to use when connecting to the broker.

Connection Timeout	Defines the maximum seconds allowed to establish a connection to the MQTT broker. Default: `30`.
Keep Alive Interval	Defines the maximum interval before disconnecting an idle MQTT client. Default: `60`.
Clean Session	Enables/disables clean sessions. `True`: Establish a clean session. Any messages that have been persisted by the broker will not be solicited to the client. `False`: Don't establish a clean session. Any messages that have been persisted by the broker will be solicited to the client. Default: `true`.
Client Persistence Directory	Defines a directory to store client persistence files. Client persistence is useful if **Clean Session** is set to `false`. This is because persistence is used by the client to help the broker determine which messages were missed by the client in case the connection is lost with the broker. Default: `Memory Persistence`. Messages will not persist if you restart SOAtest/Virtualize. You should define a directory for the data if it must persist across restarts.

Publish Settings

Click the **Publish** tab to configure the publish settings:

Topic	Defines which topic to publish the message to.
Quality of Service	Defines the quality of service metric that's tied to the published message. Possible values are `0`,`1`, and `2`. Default: `1`.
Retain Message on Server	Enables/disables retaining the message on the server. Setting to `true` retains the message. Retained messages are solicited to clients upon subscription to a topic regardless of their **Clean Session** settings. Default: `false`.

Last Will and Testament (LWT) Settings

The Last Will and Testament (LWT) is defined when a client connects to a broker. If the client loses connection from the broker for any reason other than a clean disconnect, then the broker will publish the clients LWT automatically.

Topic	Defines which topic to publish the LWT to.
Will Payload	Defines the payload of the LWT. Default: `none`. If this field is not set, the client will not establish a LWT with the broker.
Quality of Service	Defines the quality of service metric that's tied to the LWT. Possible values are `0`,`1`, and `2`. Default: `1`.
Retain Message on Server	Enables/disables retaining the LWT on the server. Setting to true retains the LWT. Retained messages are solicited to clients upon subscription to a topic regardless of their **Clean Session** settings. Default: `false`.

Transport Layer Security (TLS) Settings

The TLS settings must be configured when the protocol is set to **ssl or wss** in the **Broker URL** so that secure connections can be established properly.

TLS Protocol	Defines the Transport Layer Protocol to use when establishing a secure connection. Default: `TLSv1.2`.

Key Manager Factory	Defines the Key Manager Factory used to manage the keys provided in the **Key Store File**. Different Key Managers are available depending on the security providers that have been loaded into the JVM. If additional security providers have been added to the JVM, check the security provider's documentation to gather the available factory names. Default: `SunX509`**.**
Key Store File Location	Defines an absolute or relative path to the asset (.tst, .pva, .pvn) key store file.
Key Store Type	Defines the type of Key Store File provided. Different Key Store Types can be handled depending on the security providers that have been loaded into the JVM. If additional security providers have been added to the JVM, check the security provider's documentation to gather the available Key Store Type names.
Key Store Password	Defines the password to be used to extract the keys/certificates from the Key Store File. If left blank, an attempt will be made to extract the keys without a password.
Trust Manager Factory	Defines the Trust Manager Factory used to manage the keys provided in the **Trust Store File**. Different Trust Managers are available depending on the security providers that have been loaded into the JVM. If additional security providers have been added to the JVM, check the security provider's documentation to gather the available factory names. Default: `SunX509`.
Trust Store File Location	Defines an absolute or relative path to the asset (.tst, .pva, .pvn) Trust Store file. Default: `Trust All Certificates`.
Trust Store Type	Defines the type of Trust Store File provided. Different Trust Store Types can be handled depending on the security providers that have been loaded into the JVM. If additional security providers have been added to the JVM, check the security provider's documentation to gather the available Trust Store Type names.
Trust Store Password	Defines the password to be used to extract the keys/certificates from the Trust Store File. If left blank, an attempt will be made to extract the keys without a password.

Connection Management Settings

Click the Connection Management tab to configure the connection management settings.

Keep connection alive	Enable this option to keep the client connection alive and reused for subsequent publishing.
Close connection after test execution	Enable this option to close the client connection directly after publishing.

MQTT Subscriber

The MQTT Subscriber listens on the topic until the specified duration is reached, the maximum number of messages is reached, or the connection is terminated. If the maximum number of messages is zero (unlimited) then the Subscriber will attempt to listen for the full subscription duration.

Configuration

You can configure the following settings.

Connection Settings

See Connection Settings for details on how to configure this page.

Subscribe Settings

Click the **Subscribe** tab to configure the subscribe settings.

Topic Filter	Defines the topic filter to use when establishing a subscription. You can use the special wildcards + and # to filter multiple topics at the same time. The + establishes a wildcard for a single level of the topic hierarchy, while the # wildcard can handle multiple levels of a topic hierarchy. Example: For the topics *parasoft/example/topic* and *parasoft/example/soavirt/topic* *parasoft/+/topic* would filter the first topic. *parasoft/#/topic* would filter both topics. **This field is required**.
Quality of Service	Defines the quality of service (QoS) level of the subscription. Messages published at a lower QoS will be received at the published QoS. Messages published at a higher QoS will be received using the QoS specified in this setting. Default: `1`.
Max Messages	Defines the maximum number of messages to wait for before ending the subscription. Default: `0`. A value of `0` specifies an unlimited number of messages.
Subscription Duration.	Defines the maximum number of milliseconds that a topic filter can be subscribed to when the Max Messages is set to `0`. Default: `30000`.
Timestamp Format	Defines the formatting of the timestamps that are generated with each message that arrives. Default: `yyyy-MM-dd'T'hh:mm:ss.SSSX`. To define custom timestamp formats please see the SimpleDateFormat Javadocs: http://docs.oracle.com/javase/8/docs/api/java/text/SimpleDateFormat.html

Last Will and Testament (LWT) Settings

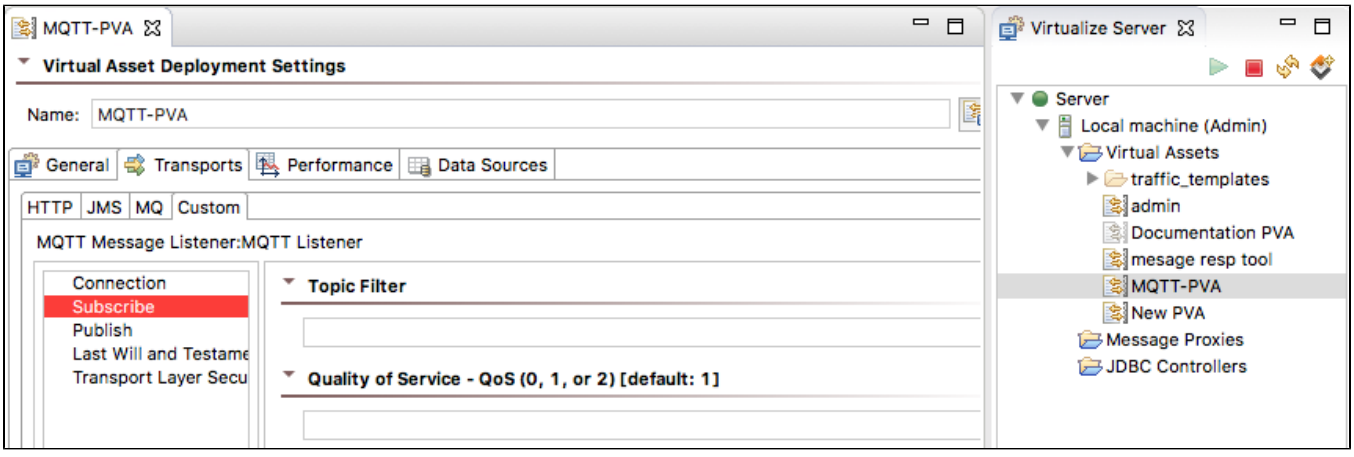
See Last Will and Testament (LWT) Settings for details on how to configure these settings.

Transport Layer Security (TLS) Settings

See Transport Layer Security (TLS) for details on how to configure these settings.

MQTT Message Listener

To use this message listener, add a Parasoft Virtual Asset (PVA) to a Virtualize server and configure the MQTT settings in the custom tab of the Virtual Asset Deployment Settings.



Configuration

You can configure the following settings.

Connection Settings

See Connection Settings for details on how to configure this page.

Subscribe Settings

Topic Filter	See Topic Filter for a description of how to configure this field. **This field is required.**
Quality of Service	See Quality of Service for a description on how to configure this field. Default: 1.

Publish Settings

See Publish Settings for details on how to configure this page.

Last Will and Testament (LWT) Settings

See Last Will and Testament (LWT) Settings for details on how to configure this page.

Transport Layer Security (TLS) Settings

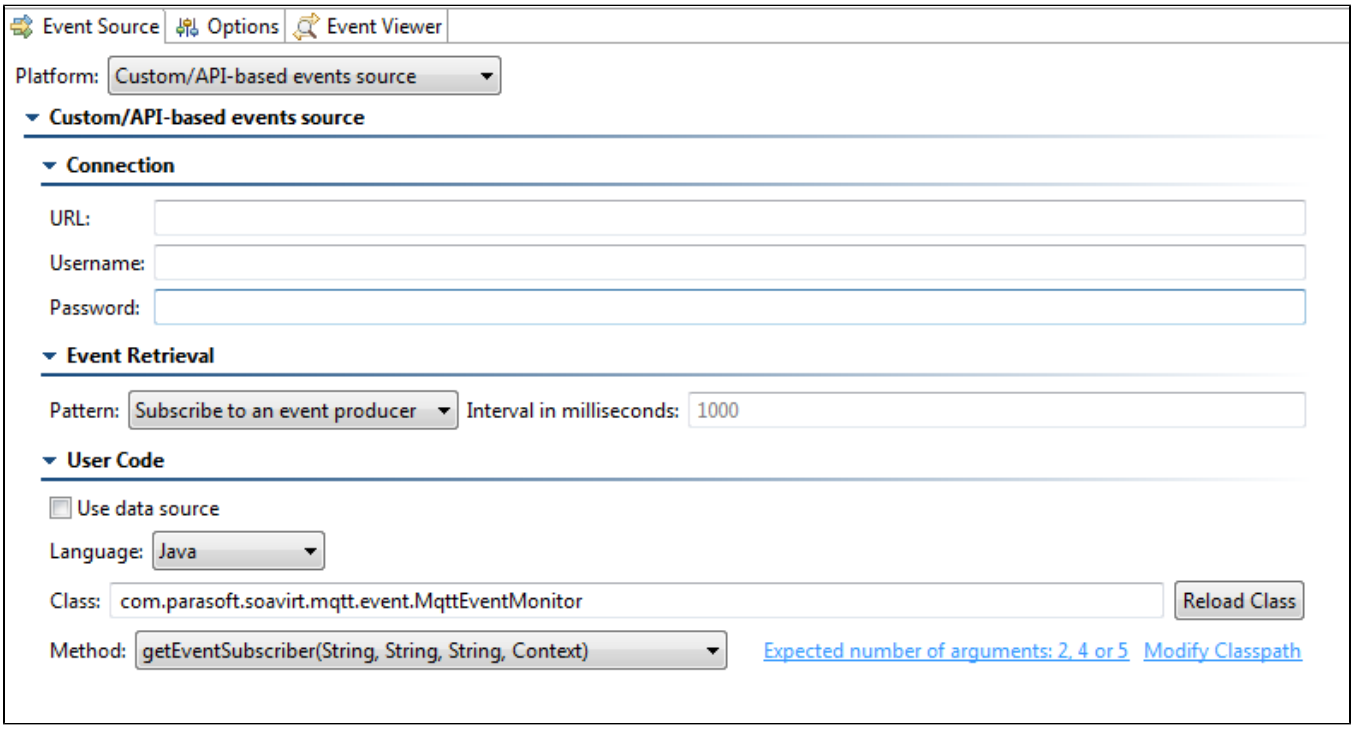
See Transport Layer Security (TLS) Settings for details on how to configure this page.

MQTT Event Monitor

The event monitor is used to subscribe to a topic filter and monitor the message arrivals during the test execution. This allows for validations to be performed on the messages that arrive. Validations include the order in which messages arrive, the time of message arrival, the topic in which the messages were published, and the message payload.

Setting up the MQTT Event Monitor

- Add an event monitor to a test.
- In the Event Source tab, choose **Custom/API-based events source** from the Platform dropdown menu.
- In the Event Retrieval section, choose **Subscribe to an event producer** from the Pattern drop-down menu.
- In the User Code section, choose **Java** from the Language drop-down menu.
- Enter **com.parasoft.soavirt.mqtt.event.MqttEventMonitor** in the Class field of the User Code section.
- Choose **getEventSubscriber(String, String, String, Context)** from the Method drop-down menu of the User Code section.



You can configure the following settings.

URL	Defines the broker URL. The broker URL is defined differently in the Event Monitor than in the other tooling. Use the following pattern to configure the broker URL: **protocol://client_id@host:port/topic_filter** • **protocol** is either tcp or SSL • **client_id** is a unique client ID • **host** is the broker's host • **port** is the broker's port • **topic_filter** is the subscription topic filter. **This field is required**. If the **SSL** protocol is defined, the **MQTT Configure Tool** must be used to configure the **Transport Layer Security (TLS)** settings for the client. If **client_id** is not provided, a client ID will be generated by the server. You cannot use wildcards in the URL to define the topic filter or else the URL will not parse properly. To use wildcards in your topic filter, omit the topic filter from the URL declaration and define the topic filter using the **MQTT Configure Tool**. You can define a custom timestamp format of the message arrival events with the **MQTT Configure Tool**.
Username	Defines the username to use when connecting to the broker.
Password	Defines the password to use when connecting to the broker.

MQTT Configure Tool

The configure tool enables you to configure certain MQTT properties for an entire test, reducing the amount of configuration required in the individual test steps if the same properties are required. Any settings defined in the configure tool can be overridden by the individual test steps. The configure tool is especially useful when using the Event Monitoring interface because there is no way to configure TLS properties in the Event Monitor tool. As a best practice, this tool should only be used as a Set-Up test.

Configuration

You can configure the following settings.

Connection Settings

Connection Timeout	See Connection Timeout for a description of how to configure this field.
Keep Alive Interval	See Keep Alive Interval for a description on how to configure this field.
Clean Session	See Clean Session for a description on how to configure this field.
Client Persistence Directory	See Client Persistence Directory for a description on how to configure this field.

Subscribe Settings

This page should only be configured when using the Event Monitor Tool. This page allows you to use wildcards in the topic filter and allows you to customize Quality of Services levels and message arrival timestamp formats.

Topic Filter	See Topic Filter for a description of how to configure this field.
Quality of Service	See Quality of Service for a description on how to configure this field.
Timestamp Format	See Timestamp Format for a description on how to configure this field.

Last Will and Testament (LWT) Settings

See Last Will and Testament (LWT) Settings for details on how to configure this page.

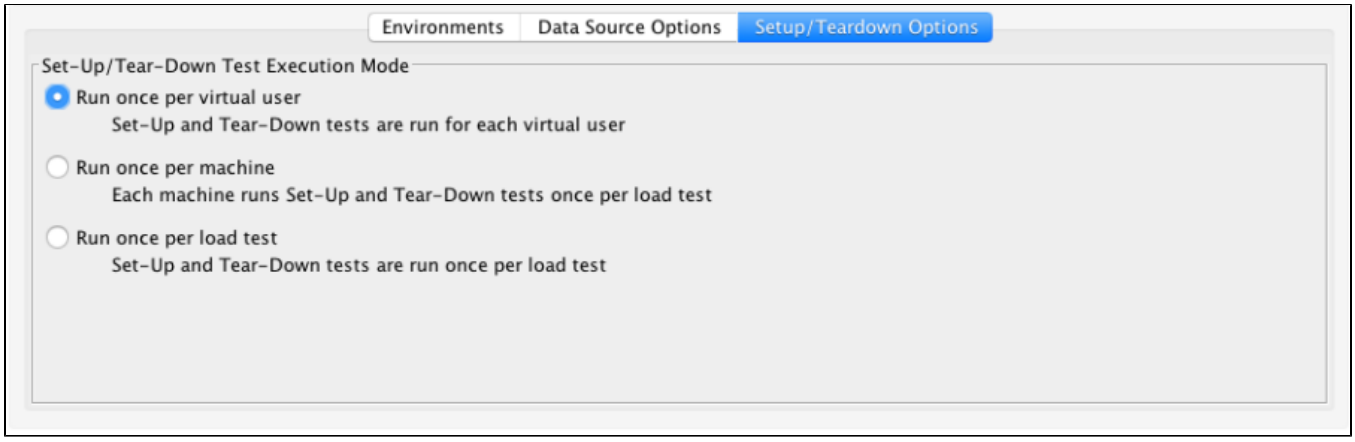
Transport Layer Security (TLS) Settings

See Transport Layer Security (TLS) Settings for details on how to configure this page.

Running Load Tests with Set-Up MQTT Configure Tests

If you want to run a load test with a .tst that uses MQTT Configure tools as Set-Up tests, you must run the tools in the virtual user context.

1. Click the **Profiles** node under Load Test Configuration.
2. Click the Setup/Teardown Options tab in the Project Configuration panel and enable the **Run once per virtual user** option.



Third-party Content

This extension includes items that have been sourced from third parties as outlined below.

- Eclipse Paho (EPL 1.0)
- SLF4J (MIT License)

Additional license details are available in this plug-in's licenses folder.