Rabbit MQ Transport 1.0

In this section:

About the Plugin

The Parasoft RabbitMQ Transport Extension adds support for the RabbitMQ transport to applicable messaging client tools in Parasoft SOAtest. RabbitMQ is a lightweight, reliable, scalable and portable message broker. Applications communicate with it via a platform-neutral, wire-level protocol: the Advanced Message Queuing Protocol (AMQP).

A related extension, the RabbitMQ Listener, enables a virtual asset to receive and respond to messages over RabbitMQ in Parasoft Virtualize.

Requirements

- SOAtest 9.9.0 or later

Installation

This tool can be installed from the UI or the command line.

UI Installation

1. Choose **Parasoft> Preferences**.
2. In the System Properties preferences page, click **Add JARs**.
3. Choose **rabbitmqtransport.jar** in the file browser.

Once this jar file is added to the SOAtest classpath, all of the required dependencies will be loaded.

Command Line Installation

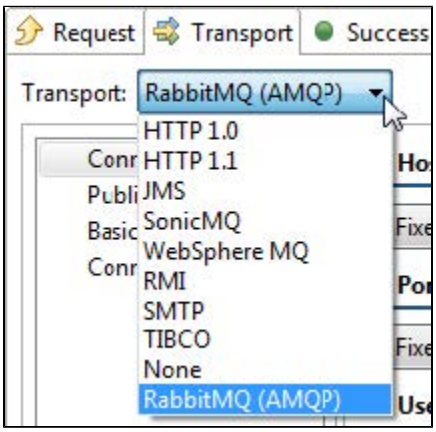
Add the rabbitmqtransport.jar file to the system.properties.classpath property in your localsettings properties file. For example:

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

Once the classpath is modified, all of the required dependencies will be loaded.

Usage

The RabbitMQ Transport is primarily used in message client tools (e.g., SOAP Client, EDI Client, and Messaging Client). The transport is configured in the Messaging Client's **Transport** tab. To use the RabbitMQ Transport in a Messaging Client, choose **RabbitMQ (AMQP)** from the Transport drop-down menu and configure the available options.



The following configuration options are available. Many settings are identical to RabbitMQ settings; see the Rabbit MQ documentation for more details.

Connection Settings

Host	Defines the hostname of the RabbitMQ server. The default is `localhost`.
Port	Defines the port of the RabbitMQ server. The default is `5672` (`5761` with SSL enabled).
Use SSL	Enables/disables SSL. The default is `false`.
Username	Defines the RabbitMQ username. The default is `guest`.
Password	Defines the RabbitMQ password. The default is `guest`.
Virtual Host	Defines the RabbitMQ Virtual Host to connect to. The default is `/`
Automatic Recovery	Set to `true` if the transport should attempt to automatically recover from connection failures. The default is `true`.
Topology Recovery	Set to `true` if the listener should attempt to automatically recover from topological failures (i.e., failures with exchanges or queues). Default is `true`.

Publish Settings

Exchange Name	Defines the exchange that the message is sent to. The default is `default exchange`.
Routing Key	Defines the routing key that will be used when sending the message.
Mandatory	Determines whether the message is published with the mandatory field set to true or false. The default is `false`.
Immediate	Not supported in RabbitMQ 3.0 and higher. Determines whether the message is published with the immediate field set to true or false. The default is false.

Basic Properties

App id	Defines the identifier of the application that produced the message.
Cluster id	Deprecated in AMQP 0.9.1. Defines the intra-cluster routing identifier.
Content Encoding	Defines the message content encoding. Enter a MIME content type (e.g., gzip).
Content Type	Defines the message content type. Enter a MIME content encoding (e.g., application/json). The default is to use the request message's content type.
Correlation Id	Defines the ID of the correlated message (e.g., the message that this is a reply to).
Delivery Mode	Determines if the message should be persisted to disk. Enter `1` for non-persistent, `2` for persistent.
Expiration	Defines the expiration time (in milliseconds) after which the message will be deleted.
Headers	Defines message headers. Enter a comma-separated key/value pairs. Example: `key1=value1,key2=value2`
Message Id	Defines the message identifier.
Priority	Defines the message priority. You can enter a number from `0` to `9`.

ReplyTo	Defines the name of the queue to which the response should be sent.
Timestamp	Defines how to set the message timestamp. You can enter `auto` to use the current time, or enter a fixed date/time. If specifying a fixed date/time, use fixed yyyy-MM-dd HH:mm:ss format (e.g., `2017-01-18 19:14:59`) or locale default format M/d/yy h:mm a (e.g., `1/17/17 7:15 pm` for the US). UNIX timestamp format is also accepted.
Type	Defines the message type (e.g., what type of event or command this message represents).
User Id	Defines the optional user ID.

Consume Settings

Wait for Reply	Set to `true` if the transport should wait for a reply message. The default is `true`. If set to `true`, you can also filter the reply messages by specifying response correlation parameters in the Response Correlation Settings.
Create Consumer Before Sending the Message	Set to `true` if a consumer should be created before sending the message. The default is `true`.
Queue Name	Defines the reply queue. The default is `temporary queue`.
Binding Exchange	Defines the exchange the temporary queue binds to.
Binding Routing Key	Defines the binding key used when binding the temporary queue to the exchange.
Timeout for Reply	Defines the timeout for the reply in milliseconds. The default is `30000`.

Response Correlation Settings

If the RabbitMQ transport is configured to wait for a reply message, these settings let you filter the reply messages by specifying response correlation parameters. If no correlation is configured (e.g., all fields are empty), no filtering will be performed and the transport will consume the first available reply message. If correlation is configured, the field entries will be evaluated from top to bottom; the first matching field will be used for correlation.

Match Response Correlation Id with Request Message Id	Set to `true` if the transport should ignore all responses except for those with Correlation ID matching the Message ID of the published message. The default is `false`.
Match Response Correlation Id with Request Correlation Id	Set to `true` if the transport should ignore all responses except for those with Correlation ID matching the Correlation ID of the published message. The default is `false`.
Match Correlation Id Value	Specify a value in this field and the transport will ignore all responses except for those with Correlation ID matching the specified value.

Connection Management Settings

Keep connection alive/ Close connection after test execution	The RabbitMQ transport allows multiple active connections to be used. If the connection settings match a connection that is already open, then it will reuse that connection. If any of the connection settings are different, then a new connection will be created. Marking a test as "keep alive" (with **Keep connection alive** enabled) will tell SOAtest not to call close() on the connection at the end of that test's execution. To close the connection, a test must be configured as "close connection" (with **Close connection after test execution** enabled). SOAtest will close the last connection used, which will be the connection used for that test. Unlike with the built-in HTTP transport, there is no final cleanup of open connections for custom transports when all execution is finished. Instead, it is the user's responsibility to mark the last test of a given connection as "close connection," otherwise the connection will stay open. For example, in the following scenario both connections will be kept open after the test run is over. Settings A (keep alive) Settings B (keep alive) Settings A (keep alive) Settings B (keep alive) To fix this, you would mark the last test for each unique set of settings as "close connection". Settings A (keep alive) Settings B (keep alive) Settings A (close connection) Settings B (close connection)

Third-Party Content

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

- RabbitMQ (Apache license)

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