Configuring Server and Deployment Settings

This topic explains how to configure deployment settings for Virtualize servers—and for the virtual assets deployed upon them.

Sections include:

Configuring Virtualize Server Startup and Authentication Options

You can configure additional settings (for example, startup, authentication, and CTP notification options) from the Preferences panel as described in Server Settings.

Security Considerations

Be sure to deploy the Virtualize server in a secure manner. The Virtualize server hosts web services that can be used to manage virtual assets. This means that *any host with network access to the Virtualize server can add, modify, or remove virtual assets hosted by that Virtualize server*

.

We generally recommended that you deploy the Virtualize server on a trusted network. A proxy server or gateway could also be used in front of the Virtualize server to add layers of security other than what is provided by the Virtualize server.

The Virtualize server's HTTPS port (9443) should be used when sending login credentials.

Extra caution should be taken if deploying a Virtualize server on an untrusted network such as the Internet. We recommend that you use a firewall to block unacceptable access (such as restricting access by IP addresses). A proxy server or gateway could also be used in front of the Virtualize server to add authentication, filtering, and logging.

Configuring SSL (HTTPS) for the Virtualize Server

When the AUT within an environment uses SSL to connect to a dependency that needs to be virtualized, there are several options:

a. Configure Virtualize Server with a generated (possibly self-signed) certificate and add it to the trust store of the AUT.
b. Add your actual server certificate to Virtualize. This option assumes that access to the server certificate and keys is possible—and changes to the AUT are difficult or shouldbe avoided. However, this option may not be possible if the certificate was signed for a hostname other than the hostname where Virtualize is deployed.
c. (preferred) Create a certificate for the Virtualize server, sign it with a certificate authority that is trusted by the AUT, and issue it for the host where the Virtualize server is installed. With this option, you don't need to make any changes to the AUT.
d. Disable certificate trust in the AUT. The AUT would still connect over SSL but trust any server (such as the Virtualize server) without validating its certificate or its trust paths.

For the AUT to accept a certificate/private key pair, you generally need—at minimum—a self-signed certificate/private key pair whose common name (CN) parameter matches the fully-qualified name of the server. For example, if your Virtualize server URL is http://myserver.mycompany.com, the CN parameter should be "myserver.mycompany.com".

In any case, Virtualize can be configured to accept incoming HTTPS connections on port 9443 (default SSL connector) or another port.

Editing server.xml

SSL details can be configured by modifying the `SSL HTTP/1.1 Connector` entry in `server.xml`. For example:

```
<Connector port="9443" maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75" enableLookups="false" disableUploadTimeout="true"
acceptCount="100" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS"
keystoreFile="C:/Path To Keystore/virtualize.pfx"
keystorePass="security" keyAlias="virtualize" keystoreType="PKCS12" truststoreFile="../lib/cacerts"
truststorePass="changeit" truststoreType="JKS" />
```

Where is server.xml?

- **If you installed Virtualize but not SOAtest:** Launch Virtualize, ensure that at least one responder has been created, then modify the server.xml file at [Virtualize install dir]/eclipse/plugins/com.parasoft.xtest.libs.web_[Virtualize_ver]/root/tomcat/conf/server.xml
- **If you installed Virtualize and SOAtest together:** Launch Virtualize, ensure that at least one responder has been created, then modify the server.xml file at [SOAtest install dir]/eclipse/plugins/com.parasoft.xtest.libs.web_[SOAtest_ver]/root/tomcat/conf/server.xml

Updating keystoreFile

If option (b) is being followed, the keyStoreFile attribute should be changed to point to the same keystore file as the actual SSL-based service that you want to emulate. Use forward slashes (/) instead of backward slashes (\).For example, C:/Users/myUser/keystore.jks.

If option (a) or (c) is being followed, use the path to the generated server keystore. Although the keystore paths can be relative to the location of the server.xml file, it is best to provide absolute paths in order to ensure correct configuration.

With option (d), you do not need to modify the keyStoreFile attribute.

Updating keystorePass, keyAlias, and keystoreType

The keystorePass, keyAlias, and keystoreType attributes should be updated accordingly:

- Modify keystorePass to be the password to your keystore
- Modify keyAlias to point to the alias of the certificate/private key pair
- Modify keystoreType to PKCS12, JKS, BKS, UBER, or PEM—depending on the type of keystore you're using

For two-way SSL (mutual authentication), the clientAuth attribute must be set to `true` and the trust store used for validating client certificates should be specified using the truststoreFile, truststorePass, and truststoreType attributes:

- Modify truststoreFile to point to your keystore file. Use forward slashes (/) instead of backward slashes (\). For example, C:/Users/myUser/keystore.jks
- Modify truststorePass to be the password to your keystore
- Modify truststoreType to PKCS12, JKS, BKS, UBER, or PEM—depending on the type of keystore you're using

Additional Configuration Details

For more details on how to enable and configure the SSL connector, see the Apache Tomcat documentation (http://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html).

It is possible to configure more than one port number for SSL. This is typically the case if different keystore/certificate configurations need to be "virtualized."

The default SSL connector (the one with attribute name="default") port number should be changed in the Virtualize server preferences (see Server Settings for details). When Virtualize starts, the preferences settings will take precedence over the server.xml settings.

You may add additional Connector elements to server.xml with distinct names and distinct SSL/certificate configurations as needed. Virtual assets and proxies cannot be mapped to specific Connector ports. A message received on a particular connector/port could be processed by any virtual asset based on header, URL and message content correlation criteria within the virtual asset path, proxy path or responder correlation rules, but not based on the port.

Reviewing and Configuring Server Settings in the Server Configuration Panel

You can configure various preferences and settings for a Virtualize server in the server configuration panel. To open this panel:

1. Start Virtualize Server in GUI mode.
2. In that GUI, open the configuration panel for the server whose settings you want to review.

From this panel, you can review and modify settings related to monitoring, server statistics, global JMS and MQ connections, and user authentication (for remote servers only).

Monitoring Tab

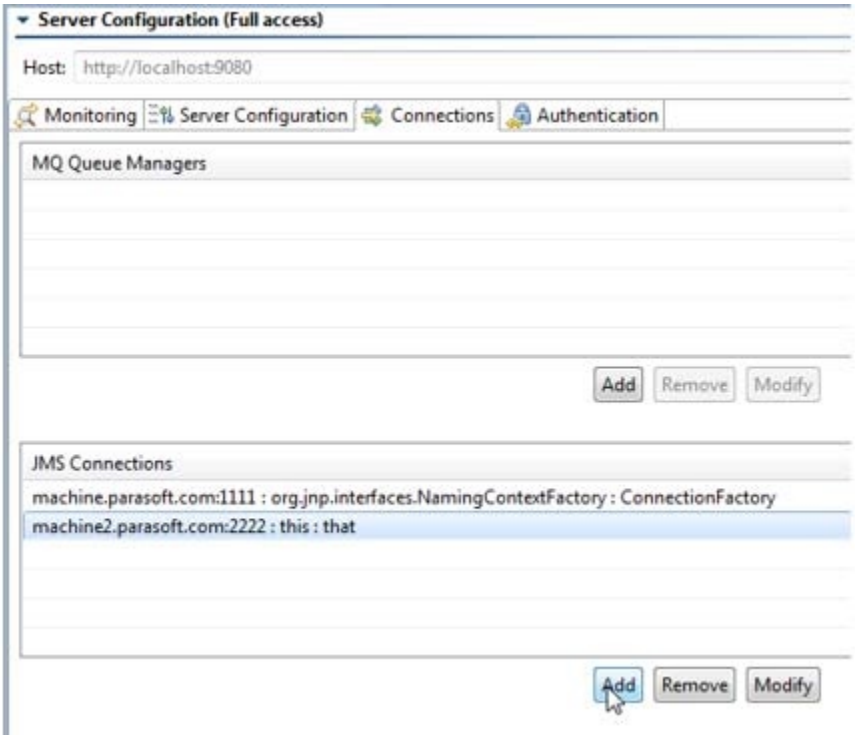
This tab displays a summary of Virtualize server statistics. See Collecting Server Statistics for details.

Server Configuration Tab

This tab allows you to configure event monitoring and statistics collection settings. See Gaining Visibility into Server Events and Collecting Server Statistics for details.

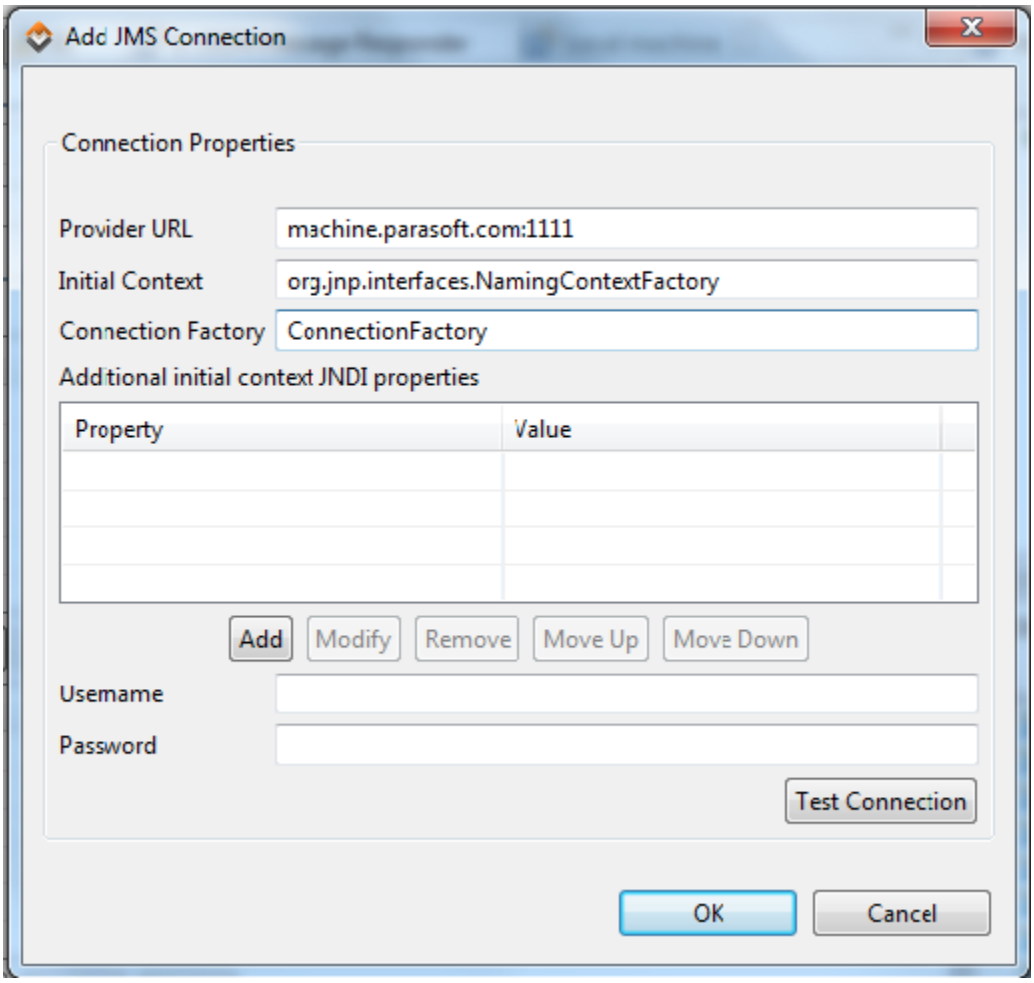
Connections Tab

This tab allows you to define multiple JMS connections and MQ queue managers that you plan to use across proxies and virtual assets on this server.



Defining Global Connections

From this tab, click the appropriate **Add** button to add a JMS connection or MQ queue manager. The fields to complete here are also used in the virtual asset or proxy configuration panels. See Configuring JMS Settings and Configuring MQ Settings for details on the applicable fields.

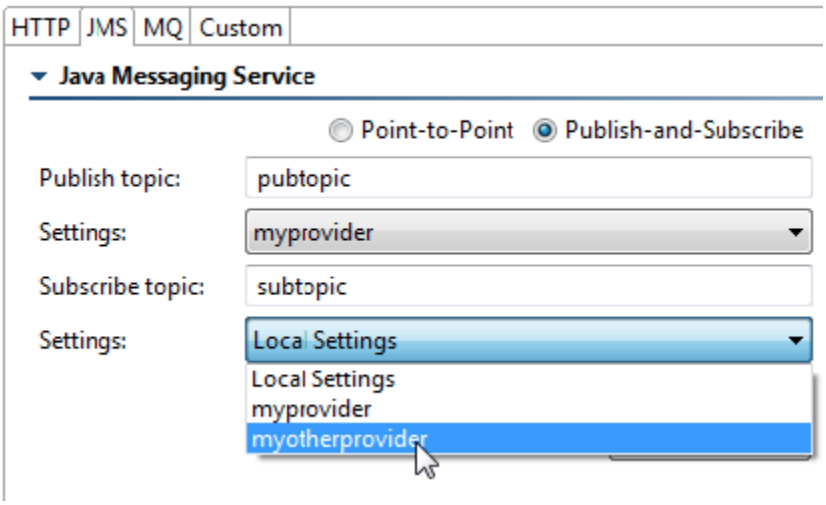


Using Global Connections

After you add a connection, you will be able to select it from the proxy and virtual asset configuration panel.

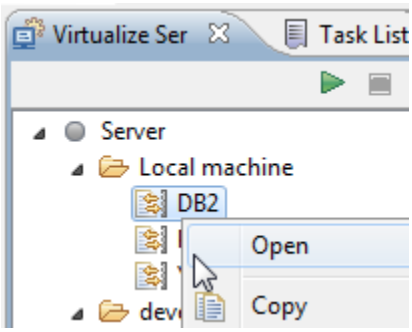
For example, this shows how to select a global connection for a proxy using MQ:

For example, this shows how to select a global connection for a virtual asset using JMS:



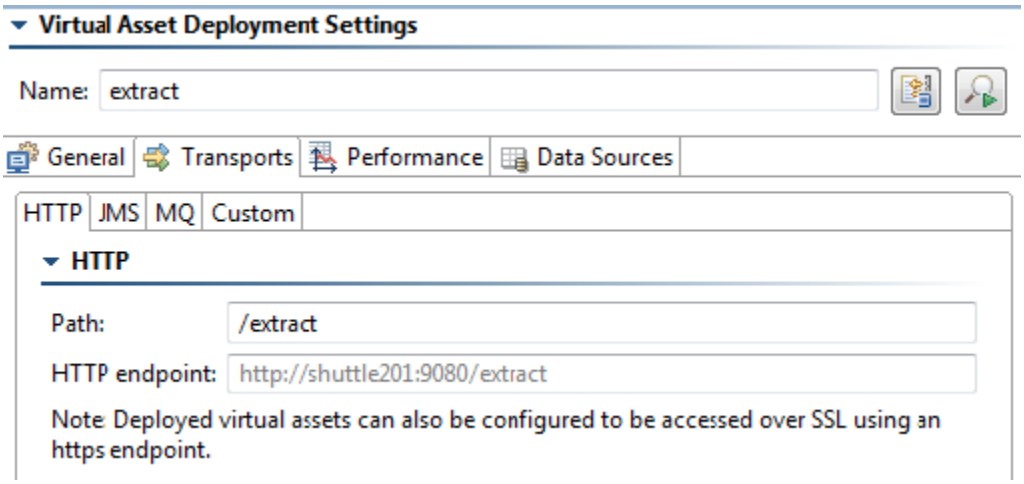
Configuring Individual Virtual Asset Deployment Settings

Individual settings can be configured by double-clicking a specific virtual asset listed in the Virtualize Server view—or by right-clicking it and choosing **Open**
.



You can then configure options as described in:

- Configuring General Virtual Asset Deployment Settings
- Configuring HTTP Settings
- Configuring JMS Settings
- Configuring MQ Settings
- Configuring Custom Transport Virtual Asset Settings
- Configuring Performance Settings
- Configuring Data Source Settings
- Configuring Virtual Database Deployment Settings



Configuring Virtual Asset Behavior

For details on customizing virtual asset behavior (e.g., how to customize virtual assets with different request/response use cases, error conditions, and so forth), you customize the related Message Responder tools as described in Message Responder Overview.

Configuring General Virtual Asset Deployment Settings

In the **General** tab, you can specify:

Option	Description
Responder Suite	Specifies the Responder Suite in which Message Responders are configured, which is essentially a .pva file. To change the file that is used for this virtual asset deployment definition, browse to a .pva file within the local workspace or the file system in general. The file paths are relative to the VirtualAssets project folder.
Description	Provides a description of the virtual asset.

Jar Files Must Be Added to Classpath

Before deploying virtual assets over JMS and/or MQ, be sure to add the appropriate jar files to the Virtualize classpath. For details on how to do this, see System Properties Settings.

Configuring HTTP Settings

You can specify the following HTTP settings:

Option	Description

Path	If you specify a value here, the virtual asset will listen for incoming messages on the HTTP path specified. If no value is specified, then only JMS, MQ or custom transport listener settings will be applied for receiving messages. To change the port number, see Using an Alternative Port for the Virtualize HTTP Server. To enable SSL (through HTTPS), see Configuring SSL (HTTPS) for the Virtualize Server. We recommend giving each virtual asset a unique HTTP path unless you want Virtualize to look into multiple virtual assets when searching for a responder that is configured to respond to the incoming message. If you deploy more than one virtual asset on the same HTTP path, Virtualize will look at each virtual asset until it finds a responder that matches the message (based on the responder correlation criteria) and use that responder and virtual asset for the response. The order in which virtual assets are evaluated in this case is not fixed; Virtualize will use the first virtual asset it finds during the evaluation. Wild cards can be used to allow segments of the path to be dynamic. For example, assume two requests were sent to Virtualize on the `paths /path/1/service` and `/path/2/service`. To have both requests go to the same virtual asset, configure the path as `/path/*/service`. Wild cards are supported for replacing an entire segment of a path. For example: • `/path/*/service` — valid • `/path/1*2/service` — not valid
HTTP Endpoint	If you are using HTTP (not JMS or MQ), this is where the virtual asset can be accessed. To exercise the virtual asset, you can configure your application to use this URL instead of the URL for the actual resource. Any machine that can access this endpoint can access and use your virtual asset.

Configuring JMS Settings

A virtual asset can be configured to receive messages from (and send messages to) a queue or a topic.

- To configure global JMS settings that apply across a specific Virtualize server, double-click the appropriate server machine node in the Virtualize Server view.
- To configure JMS settings for a specific virtual asset, double-click the appropriate virtual asset node in the Virtualize Server view.

You can specify the following JMS settings:

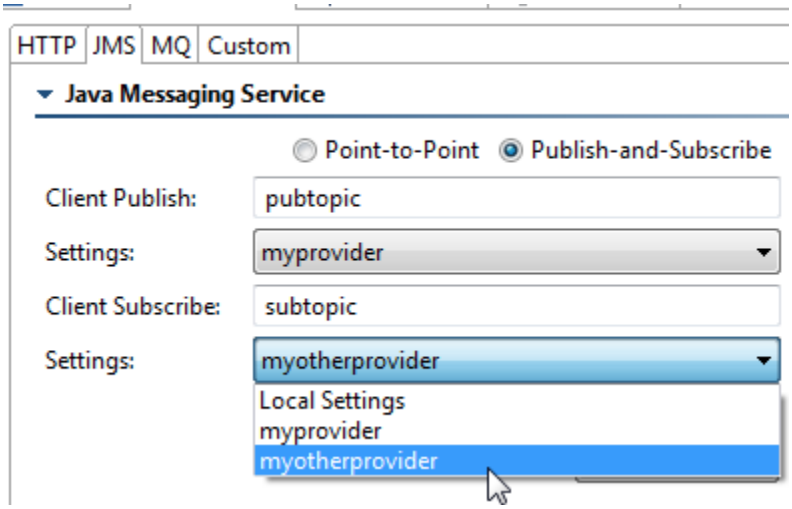
Option	Description
Messaging Mode (Point-to-Point or Publish and Subscribe)	Messaging Model options specify how messages are sent between applications. Select either **Point to Point** or **Publish and Subscribe**, then specify the settings in the appropriate area. For example, for point-to-point, you will specify Destination and replyTo queues. For publish and subscribe, you will specify Publish and Subscribe topics. You can then either enter the settings for each connection in this panel (via the Local Settings option) or reference a global JMS connection defined at the Virtualize server level (see Reviewing and Configuring Server Settings in the Server Configuration Panel for details)
Use JMSReplyTo for response	This option specifies whether to use the message's JMSReplyToQueueName header to determine where the virtual asset sends the response. If **Use JMSReplyTo for response** is enabled, values from the incoming request will be used to determine where to send the response. If it is not enabled, the response will be sent to the queue specified in the UI; the values in the JMS message header will be ignored.
Message selector	(Optional) When the same queue is being used by multiple services, it is helpful to specify a message selector expression. For example, if the message selector expression is `"product = 'virtualize'"`, then the virtual assets will select only messages in the queues/topics that have a JMS Message Property `product` with the value `virtualize`. See Using Message Selector Filters for tips.
Worker count	Worker count impacts the number of listener worker threads that get created. See Adjusting the Worker Count for details.
Provider URL	Specifies the value of the property named javax.naming.Context.PROVIDER_URL passed to the JNDI javax.naming.InitialContext constructor.
Initial context class	Specifies a fully-qualified class name string, passed to the JNDI javax.naming.InitialContext constructor as a string value for the property named javax.naming.Context.INITIAL_CONTEXT_FACTORY.
Connection factory	Specifies the key used to look up the MOM-specific factory from the initial context. This can be either a Queue Connection Factory or a Topic Connection Factory. Be sure to add the related jars to the Virtualize classpath. See JMS Provider Configuration for tips on which factory classes to add for your specific JMS provider.
Username /Password	Enter if needed.

JNDI Properties table	Specifies any additional JNDI properties you want applied to this deployment.

Using Global JMS Connections

Global JMS settings that apply across a specific Virtualize server can be defined at the server level, then referenced here. See Connections Tab for details.

To use a global JMS connection, select it from the **Settings** box.



To review the details of a predefined global connection, click **View settings**.

Message Type Support

The built-in JMS message listener supports receiving and responding with messages of type javax.jms.TextMessage.

Behavior of Virtual Assets Deployed Over JMS

The `JMSMessageID` of the request message will be sent as the `JMSCorrelationID` of the response message.

Virtual assets deployed over JMS can be invoked simply by having the application send or publish the messages to the specified destination as usual. Virtualize will consume messages on that destination. If a value is specified in the **Message Selector Expression** field, it will consume any message that matches the specified expression.

Browsing Queue Contents - Debugging and Testing of Virtual Assets Deployed Over JMS

When debugging the environment and testing virtual asset configuration, you might want to use the Queue browser to review queue contents. This visibility can be very helpful when you are trying to understand and resolve unexpected behavior.

For details, see Browsing Queues (SOAtest) or Browsing Queues (Virtualize).

Configuring MQ Settings

Virtual assets can emulate services that communicate over IBM WebSphere MQ queues if you configure the necessary MQ settings.

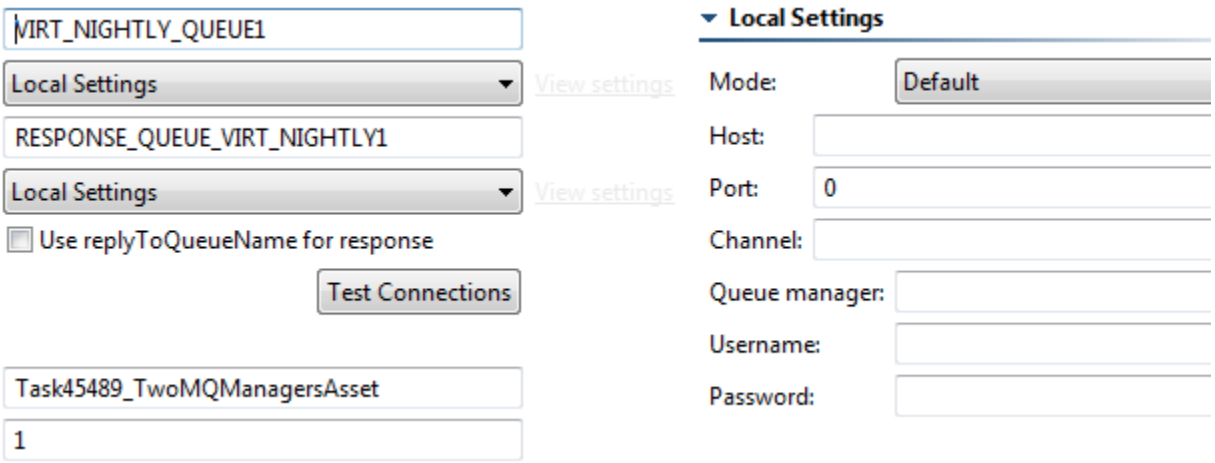
- To configure global MQ settings that apply across a specific Virtualize server, double-click the appropriate server machine node in the Virtualize Server view.
- To configure MQ settings for a specific virtual asset, double-click the appropriate virtual asset node in the Virtualize Server view.

You can specify the following MQ settings:

Option	Description

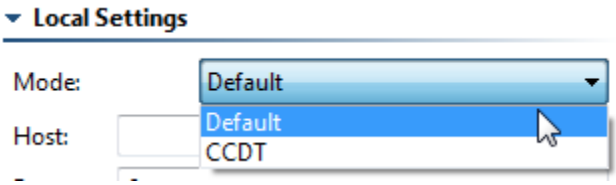
Get queue	Specifies the queue that Virtualize retrieves the request message from (also referred to as get or pull). Note that a single virtual asset can use queues that are deployed on different MQ servers; for details, see Using Global Queue Managers.
Put queue	Specifies the queue to which Virtualize sends (put or push) the response message. Note that a single virtual asset can use queues that are deployed on different MQ servers; for details, see Using Global Queue Managers.
Use replyToQueueName for Response	This option specifies whether to use the message's replyToQueueName header to determine where the virtual asset sends the response. It impacts responses to MQ messages of type MQMT_REQUEST. If **Use replyToQueueName for Response** is enabled, values from the incoming request will be used to determine where to send the response. If it is not enabled, the response will be sent to the queue specified in the UI. More specifically: • If **Use replyToQueueName for Response** is enabled and values for both MQMD.replyToQueueManagerName and MQMD.replyToQueueName have been specified, those values will determine both the queue manager and the queue name to send the response to. • If **Use replyToQueueName for Response** is enabled and either MQMD.replyToQueueManagerName or MQMD.replyToQueueName is missing from the request, the value specified in the UI will be used in place of the missing value. If **Use replyToQueueName for Response** is disabled, the replyToQueueName and replyToQueueManagerName fields in the MQ request message will be ignored. The UI settings will determine where the message is sent.
Message selector	(Optional) When the same queue is being used by multiple applications (or there are multiple types of messages exchanged over the queue), it might be necessary to filter the messages that are picked up by Virtualize. The value in this field (if a value is provided) is matched against the MQMD applicationIdData field in the messages on the queue. In this case, the MQ API MQC. MQGMO_BROWSE_NEXT flag is used to get the messages from the queue. See Using Message Selector Filters for tips.
Worker count	Worker count impacts the number of listener worker threads that get created. See Adjusting the Worker Count for details.

If the **Settings** option for the get queue and/or put queue is set to Local Settings, you will see additional options in the **Local Settings** section on the right.

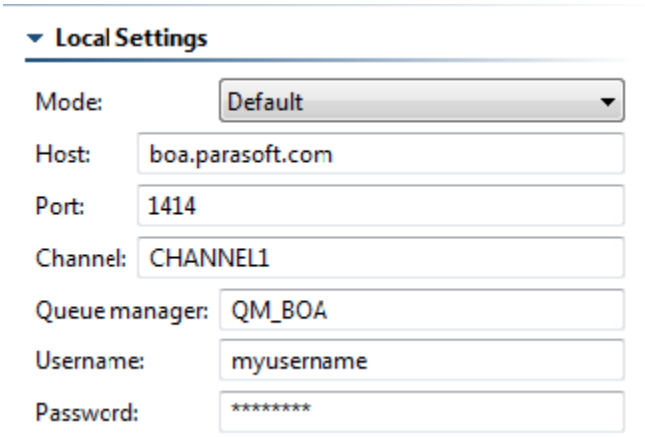


This is where you specify queue connection details using one of two modes:

- **Default mode:** Lets you enter connection details (e.g., host, port, channel, etc.) manually.
- **CCDT mode:** Lets you specify a client channel definition table (CCDT) file that provides connection details.



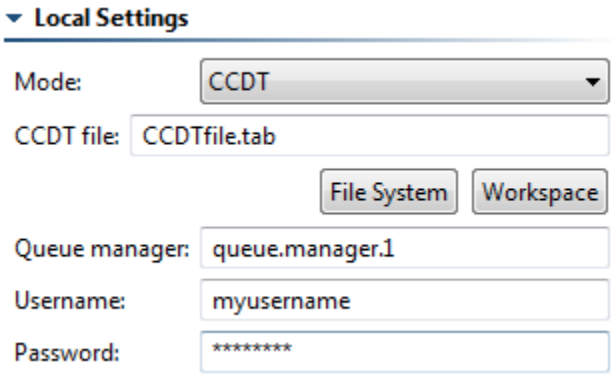
If you are using Default mode, complete the following fields:



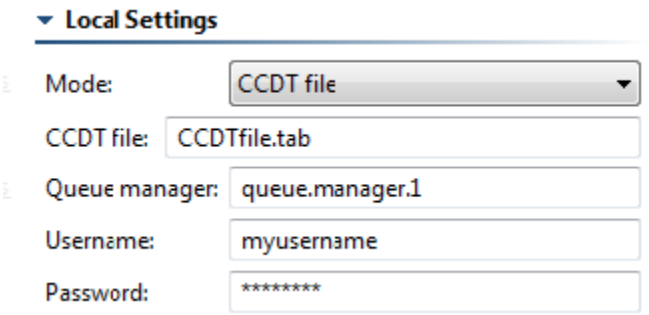
Option	Description
Host	Specifies the name of the host running IBM MQ.
Port	Specifies the port where IBM MQ is running.
Queue manager	Specifies the queue manager's name.
Channel	Specifies the name of the server-defined channel.
Username/Password	Enter if needed.

If you are using CCDT mode, complete the following fields...

For a local server:



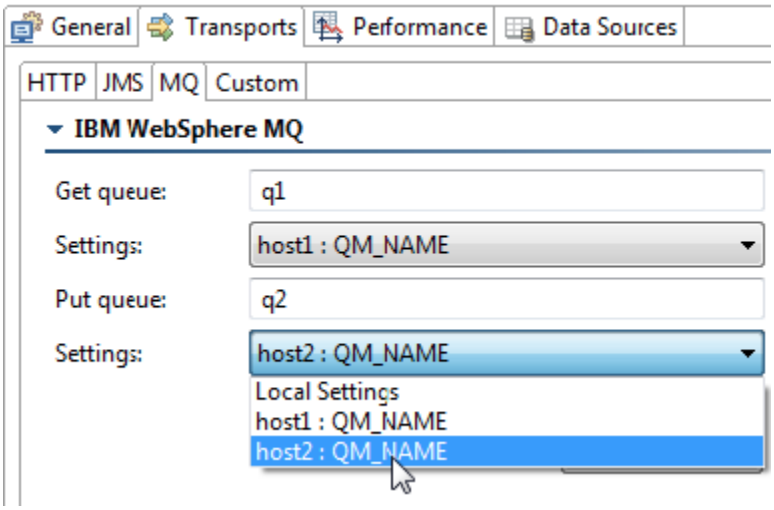
For a remote server:



Option	Description
CCDT file	Specifies the location of the CCDT file (with a .tab extension). If the virtual asset is deployed on a remote server, use the text field to specify the relative path to the CCDT file *as it would appear under the "Files" node in the Virtualize Server tree*. If the virtual asset is deployed on a local server, you can use the File System or Workspace buttons to browse to the file's location. If you are configuring a virtual asset that is currently deployed on a local server but will later be deployed on a remote server, we recommend keeping the CCDT file alongside the .pva file. Note that you will need to deploy the CCDT file to the remote server before deploying the virtual asset to that remote server. See Transferring Files Between the Remote Server and the Local Machine for details.
Queue manager	Specifies the queue manager's name.
Username /Password	Enter if needed.

Using Global MQ Queues

If you want to configure queues deployed on different MQ servers within a single virtual asset(e.g., you want a specific virtual asset to use two queues that are deployed on two different MQ servers), you can define them globally at the Virtualize server level, then reference them here. See Connections Tab for details.



Browsing Queue Contents - Debugging and Testing of Virtual Assets Deployed Over MQ

When debugging the environment and testing the virtual asset configuration, you might want to use the Queue browser to review queue contents. This visibility can be very helpful when you are trying to understand and resolve unexpected behavior.

For details, see Browsing Queues.

Adjusting the Worker Count

Each worker creates its own connection to the MQ/JMS provider. For example, for MQ, if you have 20 workers, your WebSphere MQ Explorer should show the value 20 in the Open input count column for the request (get) queue that the virtual asset is listening on. Whenever an asset is deployed /redeployed with a worker count that is higher than the default value of 1, you should see messages like "Started x listener(s)" in the Console (where x is the number of workers configured).

Increasing the worker count can help performance under concurrency. The entire message processing chain of the virtual asset is parallelized, so each worker thread will do message correlations, response message generation, etc. in parallel with other threads. However, beware that if you provide a high worker count, deploying/undeploying/redeploying an asset will take longer because there are more connections to create/destroy. Also, it is possible that the WebSphere MQ infrastructure or the JMS provider has a limit on how many concurrent connections to allow. You should not exceed what is configured /allowed by your infrastructure.

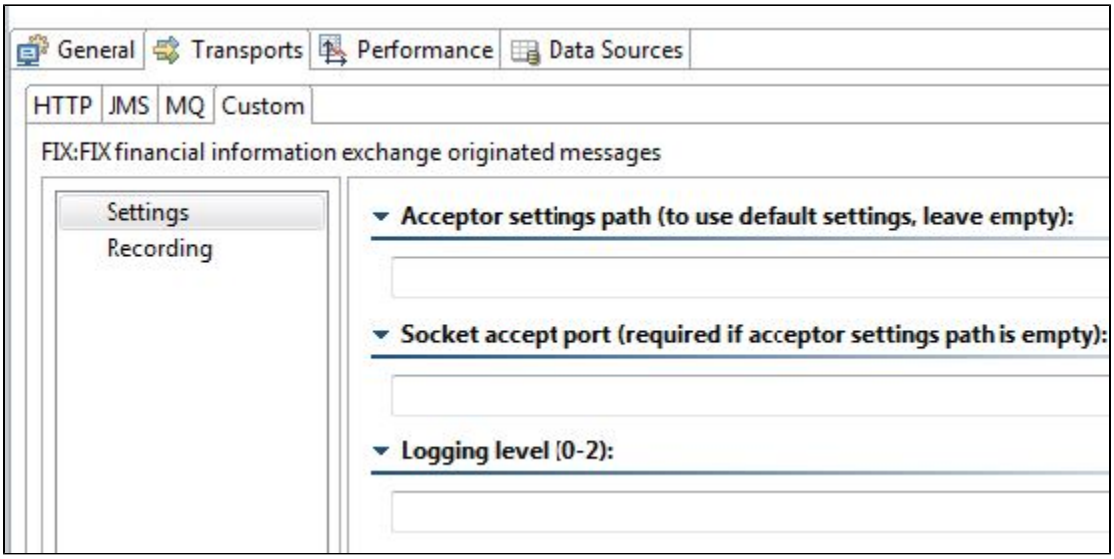
The worker count feature is equivalent to the “maxThreads” attribute in Tomcat server.xml; see Configuring SSL (HTTPS) for the Virtualize Server for details on where to find the server.xml for Virtualize’s Tomcat.

When subjecting Virtualize to a high number of concurrent virtual users (i.e. during a load test), you can usually expect better performance by increasing these values.

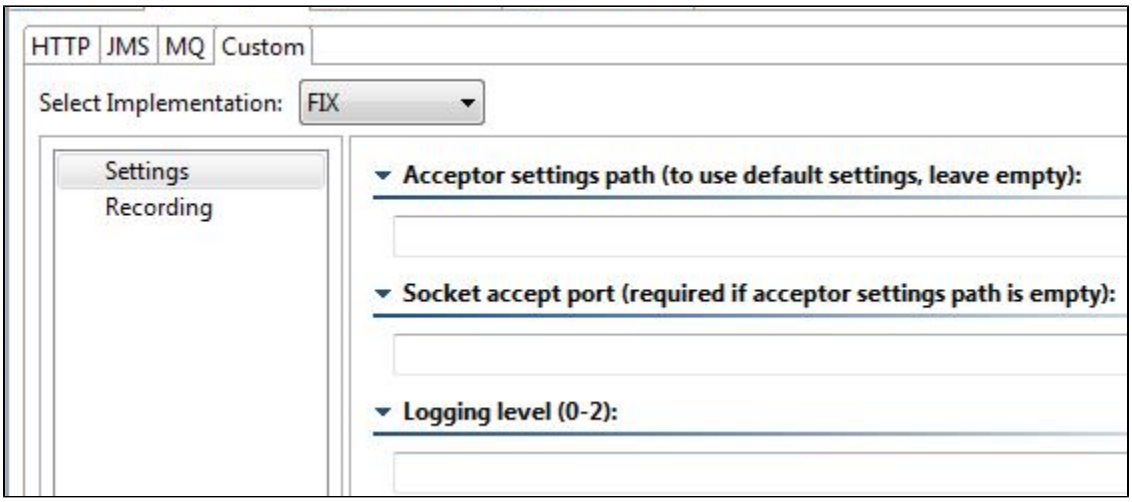
Configuring Custom Transport Virtual Asset Settings

You can configure settings for custom listeners via the **Transports> Custom** tab.

If only one custom listener is available, this tab will be dedicated to configuration for that custom format:



If multiple custom listeners are available, you can select the one you want to use from the **Select Implementation** box, then use the available controls to configure it as needed.



Configuring Performance Settings

See Working with Performance Profiles.

Configuring Data Source Settings

See Working with Data Groups.

Configuring Virtual Database Deployment Settings

There are no special restrictions for .pvas that include SQL Responders; any path is acceptable. By default, the Parasoft JDBC Driver calls the virtual asset deployed at /virtualDb (when in Virtualize or Hybrid mode). If you want to configure it to call a virtual asset that is deployed at a different endpoint, add the appropriate property as described in Specifying which Virtual Asset the Parasoft JDBC Driver Calls.

It is possible to have more than one virtual asset with that same path. In this case, Virtualize will find the virtual asset and SQL Responder that corresponds to the incoming query.

It is also possible to have a single database recording used by multiple virtual assets, each of which is deployed at a different path/endpoint. This way, if you have three different AUTs—all of which need to access the same virtualized database behavior—you can create three different virtual assets from that same recording, then point each AUT to a separate virtual asset (deployed at a separate endpoint).

Using an Alternative Port for the Virtualize HTTP Server

By default, the local Virtualize server uses port 9080. To change this:

1. Choose **Parasoft> Preferences**.
2. Open **Parasoft> Server**.
3. Change the port settings.
4. Restart the server.