

Rational Integration Tester



# Reference Guide for IBM WebSphere MQ

*Version 8.0.0*



**Note**

Before using this information and the product it supports, read the information in “Notices” on page 51.

This edition applies to version 8.0.0 of Rational Integration Tester and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright IBM Corporation 2001, 2012.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>About this Publication</b> .....	<b>v</b>
Intended Audience .....	vi
Scope .....	vi
Typographical Conventions .....	vi
Contacting IBM Support .....	vi
<b>Software Requirements</b> .....	<b>1</b>
IBM WebSphere Libraries .....	2
JMS Interface Libraries (Optional) .....	4
<b>Setting Up IBM WebSphere Transports</b> .....	<b>6</b>
Creating an IBM WebSphere Transport .....	7
Configuring an IBM WebSphere MQ Transport .....	8
<b>Working with IBM WebSphere MQ Messages</b> .....	<b>25</b>
IBM WebSphere MQ Message Formats .....	26
Publishing Messages .....	29
Subscribing to Messages .....	32
<b>Using JMS to Access IBM WebSphere MQ</b> .....	<b>35</b>
Introduction .....	36
Creating JMS-Administered Objects .....	37
Sending Messages from Rational Integration Tester .....	39
<b>Deploying Rational Integration Tester's API Exit</b> .....	<b>41</b>
Introduction .....	42
Installing the API Exit File .....	42

---

Configuring a Queue Manager to Use the API Exit. ....	43
<b>Troubleshooting</b> .....	<b>46</b>
IBM MQ Error 2009 (MQRC Connection Broken) .....	47
IBM MQ Error 2035 (MQRC Not Authorized) .....	47
IBM MQ Error 2374 (MQRC API Exit Error) .....	47
Unable to Edit Transport or Formatter Unavailable. ....	48
Messages Not Recording When Using API Exits .....	48
Namelists for API Exit Not Created .....	48
<b>Glossary</b> .....	<b>50</b>
<b>Notices</b> .....	<b>51</b>
Trademarks and service marks .....	54

# About this Publication

## **Contents**

### **Intended Audience**

### **Scope**

### **Typographical Conventions**

### **Contacting IBM Support**

This guide describes how to configure and run IBM® Rational® Integration Tester using the IBM WebSphere® MQ transport, which provides support for queue messaging-types. Information is also provided about using the JMS interface to connect to IBM WebSphere MQ.

---

## Intended Audience

This document assumes that readers are familiar with software testing, and with installing and using Rational Integration Tester and IBM WebSphere MQ.

## Scope

This document discusses how to use Rational Integration Tester with IBM WebSphere MQ.

## Typographical Conventions

The following typographical conventions are observed throughout this document.

Type	Usage
Constant Width	Program output, listings of code examples, file names, commands, options, configuration file parameters, and literal programming elements in running text.
<i>Italic</i>	Document title names in statements that refer you to other documents. Also used to highlight concepts when first introduced.
<b>Bold</b>	Menu items in graphical user interface windows (such as Microsoft Windows-based or UNIX X Window applications) from which you select options or execute macros and functions.  Submenus and options of a menu item are indicated with a “greater than” sign, such as <b>Menu &gt; Submenu</b> or <b>Menu &gt; Option</b> .

## Contacting IBM Support

To contact IBM Support, see: [www.ibm.com/contact/us/en/](http://www.ibm.com/contact/us/en/)

# Software Requirements

## **Contents**

**IBM WebSphere Libraries**

**JMS Interface Libraries (Optional)**

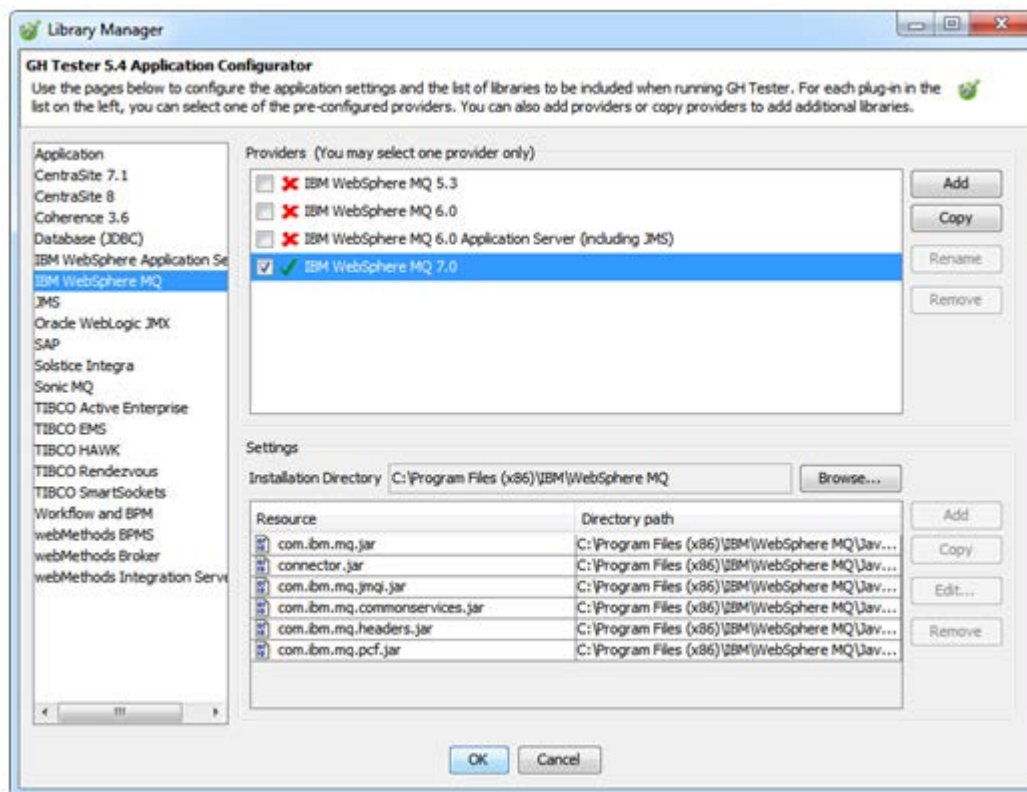
This chapter describes software requirements for using the IBM WebSphere MQ plugin for Rational Integration Tester.

## 1.1 IBM WebSphere Libraries

To enable Rational Integration Tester to work with IBM WebSphere, use Rational Integration Tester's Library Manager to specify and record the locations of the IBM libraries that are used for accessing IBM WebSphere.

Different versions of those libraries may be required depending on the version of IBM WebSphere being used. The following table outlines the libraries (and their default locations) that are required on a computer running Microsoft Windows.

WebSphere MQ Version	Library	Default Location
v5.3, v6.0	com.ibm.mq.jar connector.jar	C:\Program Files\IBM\WebSphere MQ\Java\lib
v7.0	com.ibm.mq.jar connector.jar com.ibm.mq.jmqi.jar com.ibm.mq.commonservices.jar com.ibm.mq.headers.jar com.ibm.mq.pcf.jar	C:\Program Files\IBM\WebSphere MQ\Java\lib





---

**NOTE:** Default locations for the specified libraries can be modified by using Rational Integration Tester's Library Manager. For more information, refer to *IBM Rational Integration Tester Installation Guide*.

---

## 1.2 JMS Interface Libraries (Optional)

The JMS interface can be used to interact with IBM WebSphere MQ. (For information about this, refer to [Using JMS to Access IBM WebSphere MQ](#).)

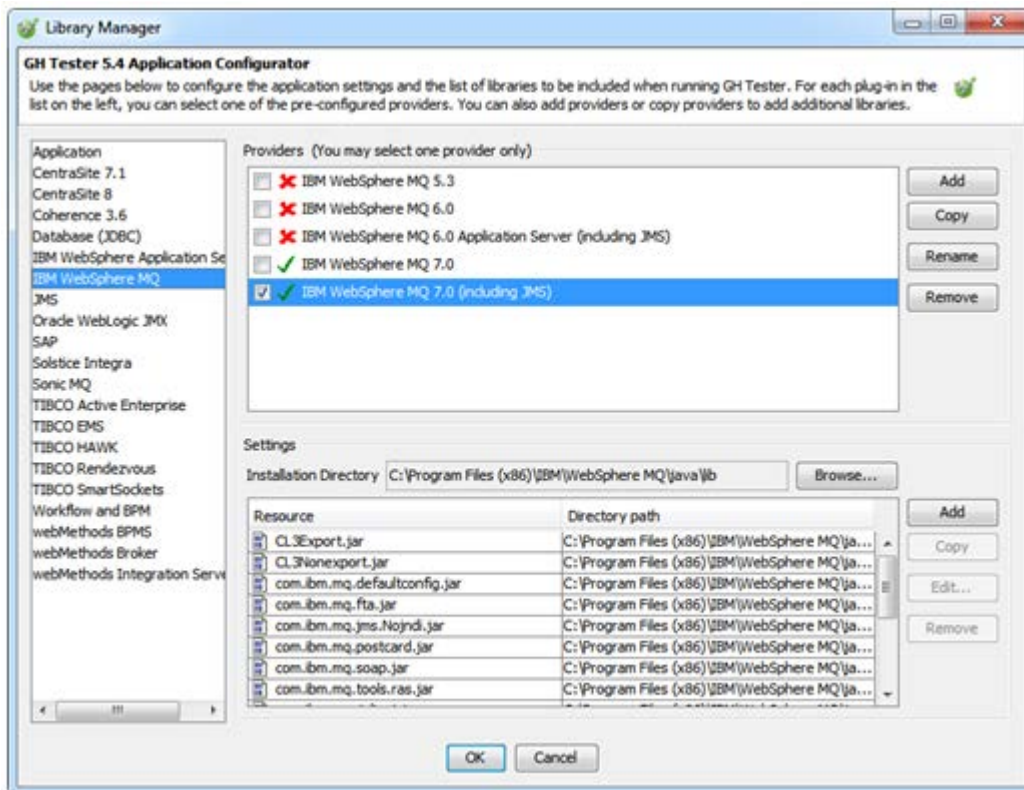
To use the JMS interface to access IBM WebSphere, create (if necessary) a new provider (under IBM WebSphere MQ) in Library Manager and record the locations of the relevant libraries.

Different versions of those libraries may be required depending on the version of IBM WebSphere being used. The following table outlines the libraries (and their default locations) that are required on a computer running Microsoft Windows.

---

<b>WebSphere MQ Version</b>	<b>Library</b>	<b>Default Location</b>
<hr/>		
v5.3	client.jar, com.ibm.mq.jar, com.ibm.mqjms.jar, connector.jar, fscontext.jar, jms.jar, jta.jar, messagingClient.jar, providerutil.jar, sibc.jms.jar, sibc.jndi.jar, sibc.orb.jar	C:\Program Files\IBM\WebSphere MQ\Java\lib
v6.0	CL3Export.jar, CL3Nonexport.jar, client.jar, com.ibm.mq.jar, com.ibm.mqjms.jar, connector.jar, dhbcore.jar, fscontext.jar, jms.jar, jta.jar, messagingClient.jar, providerutil.jar, rmm.jar, sibc.jms.jar, sibc.jndi.jar, sibc.orb.jar	C:\Program Files\IBM\WebSphere MQ\Java\lib
v7.0	CL3Export.jar, CL3Nonexport.jar, com.ibm.mq.defaultconfig.jar, com.ibm.mq.fta.jar, com.ibm.mq.jms.Nojndi.jar, com.ibm.mq.postcard.jar, com.ibm.mq.soap.jar, com.ibm.mq.tools.ras.jar, com.ibm.mqetclient.jar, com.ibm.mqjms.jar, dhbcore.jar, fscontext.jar, jms.jar, jndi.jar, jta.jar, ldap.jar	C:\Program Files\IBM\WebSphere MQ\Java\lib

---



**NOTE:** For information about Library Manager, refer to *IBM Rational Integration Tester Installation Guide*.

# Setting Up IBM WebSphere Transports

## **Contents**

**Creating an IBM WebSphere  
Transport**

**Configuring an IBM WebSphere  
MQ Transport**

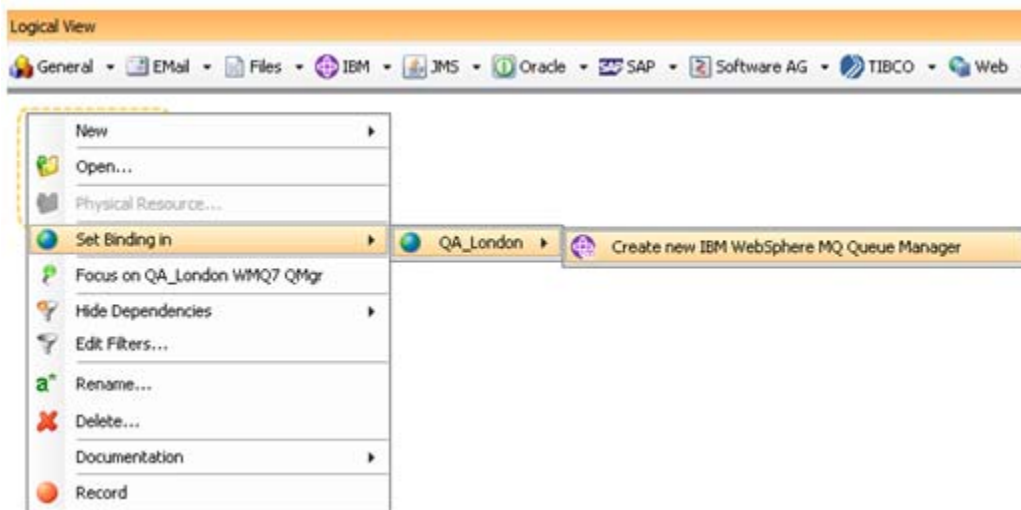
This chapter describes how to publish and subscribe to IBM WebSphere messages within Rational Integration Tester.

---

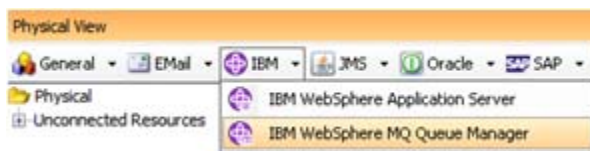
## 2.1 Creating an IBM WebSphere Transport

An IBM WebSphere transport is created when you create a physical IBM WebSphere MQ Queue Manager resource in Rational Integration Tester's Architecture School using either of the following methods:

- In the Logical View, right-click an IBM WebSphere MQ Queue Manager, and click **Set Binding in**, the relevant environment, and **Create new IBM WebSphere MQ Queue Manager** on the shortcut menus.



- On the Physical View's toolbar, click **IBM > IBM WebSphere MQ Queue Manager**.



Each physical resource will represent an IBM WebSphere MQ Queue Manager transport that can be selected and configured subsequently.

**NOTE:** The physical MQ resource must be bound by means of an environment to a logical IBM WebSphere MQ Queue Manager at run time.

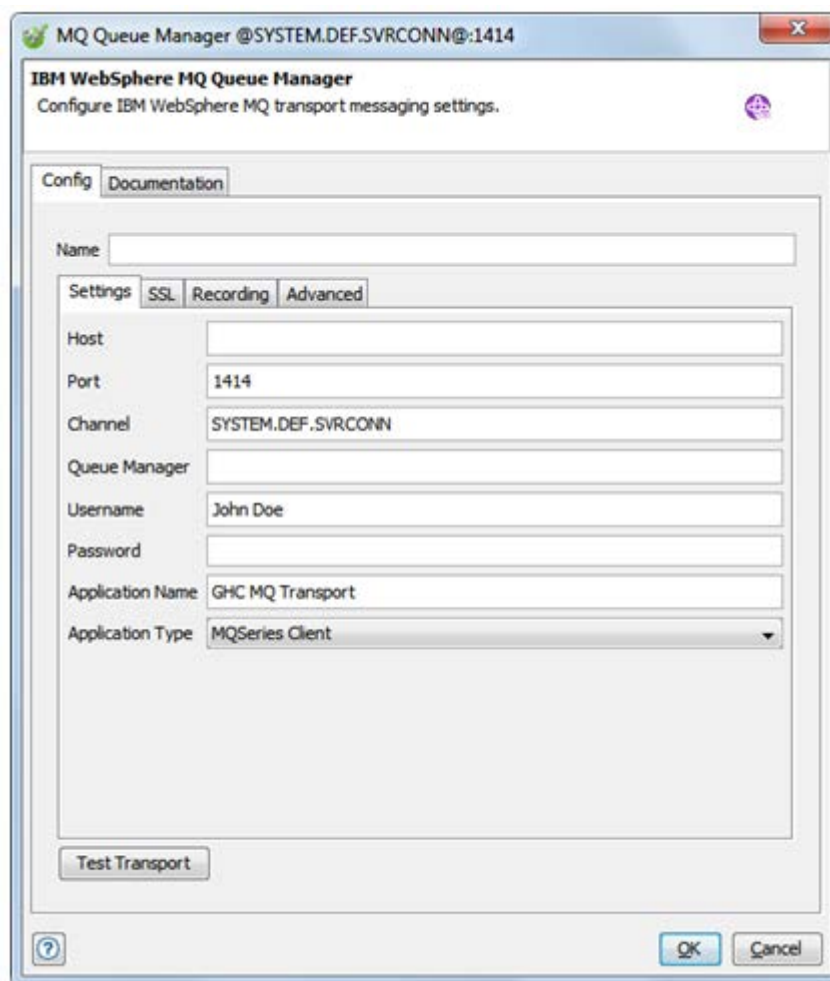
---

## 2.2 Configuring an IBM WebSphere MQ Transport

The IBM WebSphere MQ transport provides the ability to record, send, and receive IBM WebSphere MQ messages within Rational Integration Tester.

**NOTE:** Before recording, publishing to, and subscribing from queues, IBM WebSphere MQ and the relevant Queue Manager must be started, and the desired channels must be defined.

To configure a transport, double-click the appropriate IBM WebSphere MQ Queue Manager resource in Architecture School's Physical View.



If desired, enter a name for the transport in the **Name** field (to help identify it when multiple IBM WebSphere MQ Queue Manager transports are available).

---

The settings for the transport are separated into four tabs: **Settings**, **SSL**, **Recording**, and **Advanced**.

**NOTE:** All of the configuration fields support the use of tags. Tags can be entered manually or from the context menu, except in the **Password** field, where tag names must be entered directly (for example, %%pswd%%). Since this field is encrypted, any characters entered will be hidden.

**NOTE:** To test the transport settings at any time, click **Test Transport**. If the test is unsuccessful, verify the values in use.

---

## 2.2.1 Basic Settings

To configure basic transport settings, click the **Settings** tab.

The screenshot shows a web-based configuration interface with four tabs: 'Settings', 'SSL', 'Recording', and 'Advanced'. The 'Settings' tab is active. Below the tabs are several input fields: 'Host' (empty), 'Port' (1414), 'Channel' (SYSTEM.DEF.SVRCONN), 'Queue Manager' (empty), 'Username' (John Doe), 'Password' (empty), 'Application Name' (GHC MQ Transport), and 'Application Type' (MQSeries Client, shown as a dropdown menu).

Field	Description
Host	The host name or IP address of the IBM WebSphere MQ server.
Port	The port number where MQ is listening (the default is 1414).
Channel	The name of the MQ communications channel to use for sending and receiving.  <b>NOTE:</b> This must be a Server Connection type because Rational Integration Tester is a client application. <b>NOTE:</b> This field is case sensitive.
Queue Manager	The name of the MQ queue manager to use.  <b>NOTE:</b> This field is case sensitive.
Username	The ID of the user who is sending or receiving messages from Rational Integration Tester.
Password	The password (if required) associated with the user name specified above.
Application Name	The name of Rational Integration Tester MQ transport.
Application Type	Click the relevant application-type in the list.

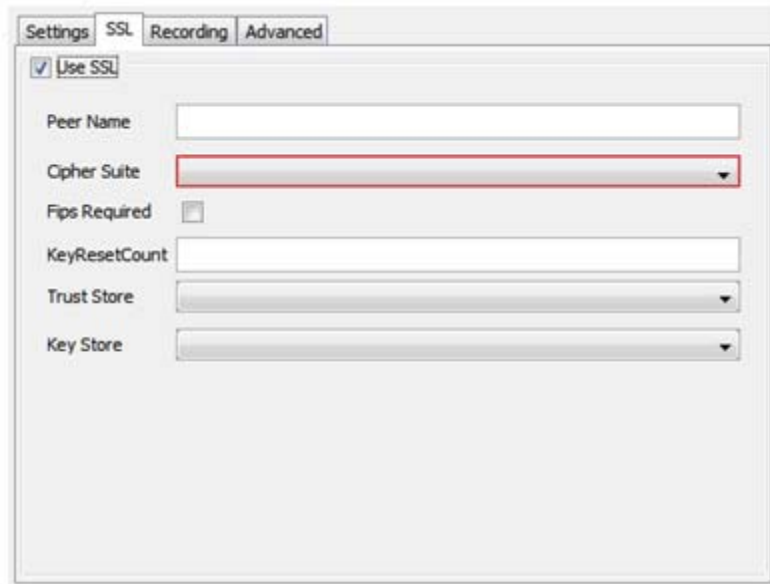
---



---

### 2.2.2 SSL Settings

To enable and configure SSL settings for the transport, click the **SSL** tab.



The screenshot shows a configuration window with four tabs: 'Settings', 'SSL', 'Recording', and 'Advanced'. The 'SSL' tab is active. Inside the tab, there is a checked checkbox labeled 'Use SSL'. Below this checkbox are several configuration fields: 'Peer Name' (text input), 'Cipher Suite' (dropdown menu, highlighted with a red border), 'Fips Required' (checkbox), 'KeyResetCount' (text input), 'Trust Store' (dropdown menu), and 'Key Store' (dropdown menu).

To enable SSL, select the **Use SSL** check box. The remaining configuration options, described below, will then become available.

---

Field	Description
Peer Name	The Distinguished Name (DN) of the queue manager to be used by SSL. The queue manager identifies itself using an SSL certificate, which contains a DN. Rational Integration Tester can use this DN to ensure that it is communicating with the correct queue manager.  In MQ, a DN pattern is specified using the <b>sslPeerName</b> variable of <b>MQEnvironment</b> . Connections will succeed only if <b>Peer Name</b> matches the pattern that is specified.
Cipher Suite	Select one of the available cipher suites to use for encrypting the transport communications.
Fips Required	This option specifies whether or not the requested cipher suite must use FIPS-certified cryptography in IBM WebSphere MQ.

---

---

Field	Description
KeyResetCount	The total number of non-encrypted bytes that can be sent and received within an SSL conversation before the secret key is renegotiated. If left blank or set to zero (default), the secret key is never renegotiated. This value is ignored if no cipher suite is specified. Valid values are integers between 0 and 999,999,999.  <b>NOTE:</b> KeyResetCount is not supported in IBM WebSphere MQ 5.3.s but is supported in IBM WebSphere MQ 6.0 (or later).
Trust Store	To enable server authentication, select the server identity store that has been configured in Rational Integration Tester's Physical View.
Key Store	To enable client authentication, select the client identity store that has been configured in Rational Integration Tester's Physical View.

---

**NOTE:** For more information about creating and configuring an identity store, refer to *IBM Rational Integration Tester Reference Guide*.

### 2.2.3 Recording Settings

Rational Integration Tester provides a number of ways to record IBM WebSphere MQ messages. Choosing the most appropriate method will depend on several factors. Some methods require the installation of additional software on the IBM WebSphere MQ queue manager server. (For information about this additional software, refer to [Deploying Rational Integration Tester's API Exit](#).)

**NOTE:** In Rational Integration Tester, recording methods are configured on a per-transport basis. If you wish to record different queues using different methods, you will have to create multiple transports or create just a single transport but change the setting on that transport each time you want to change the recording method.

---

The following table provides guidelines for selecting recording methods in Rational Integration Tester when using IBM WebSphere MQ.

---

<b>Recording Method</b>	<b>Description (Summary)</b>	<b>Advantages</b>	<b>Disadvantages</b>
Queue browsing (applicable for all versions of IBM WebSphere MQ)	<ul style="list-style-type: none"> <li>• Suitable for “recording” a snapshot of the messages in a queue because it does not affect the messages in the queue.</li> <li>• A quick and easy way to get a copy of messages that were posted into a queue thus facilitating the quick creation of tests.</li> </ul>	<ul style="list-style-type: none"> <li>• No additional software required on the queue manager.</li> <li>• No requirement to reconfigure queues or queue managers when recording starts or stops.</li> </ul>	<ul style="list-style-type: none"> <li>• The subscribing system must be stopped to ensure that Rational Integration Tester can view the messages.</li> <li>• Messages are not removed, so if you record again, you will get all the events again.</li> </ul>
Proxy queues (applicable for all versions of IBM WebSphere MQ)	<ul style="list-style-type: none"> <li>• The client is changed to publish into a queue that Rational Integration Tester monitors.</li> <li>• When Rational Integration Tester sees an event it “records” it and then publishes a duplicate message to the original queue.</li> <li>• A proxy queue is named the same as the request or reply queue, except for the addition of the Queue Suffix (specified in the IBM WebSphere MQ transport).</li> <li>• Message initiators should be configured to send their requests to the proxy queue.</li> </ul>	<ul style="list-style-type: none"> <li>• No additional software required on the queue manager.</li> <li>• The subscriber does not have to be reconfigured.</li> </ul>	<ul style="list-style-type: none"> <li>• This requires the client application to be changed to put into the Proxy Queue before recording starts.</li> <li>• When recording stops, the client application must be re-configured to put into the original queue.</li> <li>• If recording studio is not running, the end-system will not see the messages.</li> </ul>

---

---

---

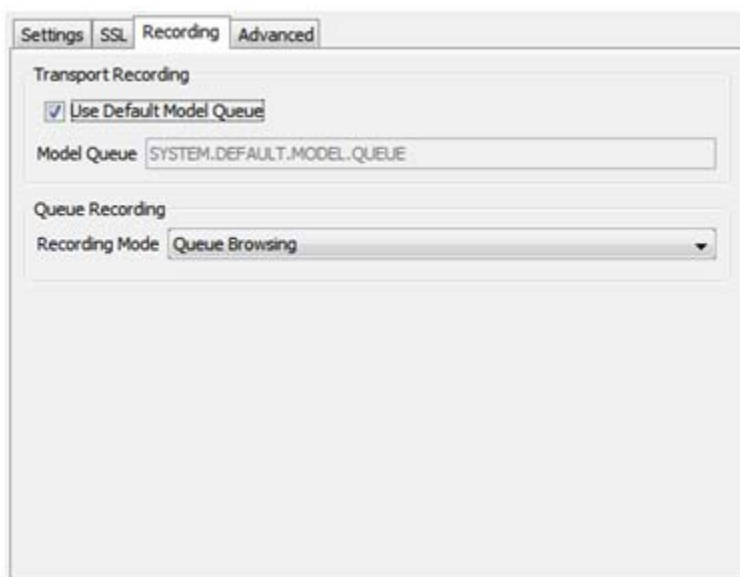
Recording Method	Description (Summary)	Advantages	Disadvantages
Mirror queues (applicable only to IBM WebSphere MQ 7.0 but not 7.1 or later)	<ul style="list-style-type: none"><li>• Record messages from a single queue (or request/reply queue pair) with no re-configuration of the system under test.</li><li>• With this technique, Rational Integration Tester configures the queue manager to create a “mirrored queue”.</li><li>• Every message put into the queue being recorded is copied into the mirrored queue.</li><li>• Rational Integration Tester subscribes to the mirrored queue.</li></ul>	<ul style="list-style-type: none"><li>• Transparent to the system under test.</li><li>• No requirement to reconfigure the system under test.</li></ul>	<ul style="list-style-type: none"><li>• The performance of the queue manager is affected adversely.</li><li>• Software must be installed on the queue manager.</li></ul> <p><b>NOTE:</b> Changes to the queue manager may require authority to administer IBM WebSphere MQ.</p>
Dynamic mirror queues	<ul style="list-style-type: none"><li>• Records messages to a temporary queue instead of a fixed queue.</li></ul>	<ul style="list-style-type: none"><li>• Transparent to the system under test.</li><li>• No requirement to reconfigure the system under test.</li><li>• IBM WebSphere MQ clears queues automatically.</li></ul>	<ul style="list-style-type: none"><li>• The performance of the queue manager is affected adversely.</li><li>• Software must be installed on the queue manager.</li></ul> <p><b>NOTE:</b> Changes to the queue manager may require authority to administer IBM WebSphere MQ.</p>

---

<b>Recording Method</b>	<b>Description (Summary)</b>	<b>Advantages</b>	<b>Disadvantages</b>
Queue aliasing (applicable only to IBM WebSphere MQ 7.0 or later)	<ul style="list-style-type: none"> <li>If the system under test publishes by means of a Queue Alias rather than directly to a queue, Rational Integration Tester can reconfigure the queues automatically so that both Rational Integration Tester and the original subscriber receive a copy of the message.</li> </ul>	<ul style="list-style-type: none"> <li>No additional software required on the queue manager.</li> <li>If the system under test already uses alias queues, there is no requirement to reconfigure it.</li> </ul>	<ul style="list-style-type: none"> <li>There must be no connections to the queue alias when recording starts; otherwise, Rational Integration Tester will not be able to perform the necessary automatic reconfiguration.</li> <li>This recording mode works only if the client application is using a queue alias when publishing.</li> <li>Because the queue alias will be configured to point to a topic during recording, the client must not be using any queue-specific API calls on the alias, such as queue depth.</li> <li>There may be an overhead during performance testing.</li> <li>Rational Integration Tester will attempt to reconfigure the system under test back to the original settings when recording ends; however, this will fail if applications are still connected to the queue alias.</li> </ul>

Recording Method	Description (Summary)	Advantages	Disadvantages
Record-The-Transport (applicable only to IBM WebSphere MQ 7.0 but not 7.1 or later)	<ul style="list-style-type: none"> <li>Record messages from multiple queues simultaneously.</li> <li>Record when you do not know the queue being used.</li> <li>Record many queues without having to set up individual operations first.</li> </ul> <p><b>NOTE:</b> This is especially useful when dynamic reply queues are being used and the queue name is not known in advance.</p>	<ul style="list-style-type: none"> <li>Transparent to the system under test.</li> <li>No requirement to reconfigure of the system under test.</li> <li>There is no requirement to know the exact queue names in use.</li> </ul>	<ul style="list-style-type: none"> <li>A very large number of recorded messages can be produced.</li> <li>The performance of the queue manager is affected adversely.</li> <li>Software must be installed on the queue manager.</li> </ul> <p><b>NOTE:</b> Changes to the queue manager may require authority to administer IBM WebSphere MQ.</p>

The recording method and options (used when capturing messages in the Recording Studio perspective) can be set under the **Recording** tab on the physical transport.



---

Rational Integration Tester uses temporary queues for recording. Selecting the **Use Default Model Queue** check box (default setting) enables you to accept the default model queue provided by IBM WebSphere MQ. If you want specify a different model queue, clear the **Use Default Model Queue** check box and enter a queue name in the **Model Queue** field.

In the **Recording Mode** list, click the desired recording method: **Queue Browsing**, **Proxy Queues**, **Mirror Queues**, **Dynamic Mirror Queues**, or **Queue Aliasing**.

These recording techniques are used when an operation referencing this transport is recorded. In contrast, Record-The-Transport is not selected within the transport but is chosen automatically when you set-up Recording Studio to record the transport rather than an operation.

**NOTE:** The **Mirror Queues**, **Dynamic Mirror Queues**, and **Queue Aliasing** recording modes are available only if IBM WebSphere MQ 7.0 (but not 7.1 or later) is installed and configured in Library Manager.

When you add an event monitor for an operation in Recording Studio, the recording approach is specified by the operation's transport while the queues that are actually recorded are specified by the operation itself.

If **Proxy Queues**, **Mirror Queues**, or **Queue Aliasing** is clicked, the **Queue Suffix** field becomes available for specifying the naming suffix that Rational Integration Tester will use to create (unique) queues and topics for the purposes of recording.

The following table describes the possible suffixes.

---

<b>Recording Mode</b>	<b>Suffix Description</b>
Mirror Queues	The name of the queue that Rational Integration Tester creates to receive copies of all messages posted to the queue you are recording will be called " <i>&lt;OriginalQueueName&gt;_&lt;Rational Integration Tester "Queue Suffix" Field Value&gt;</i> ".
Proxy Queues	The name of the queue to which the client application must be configured to publish messages is " <i>&lt;QueueName&gt;_&lt;Rational Integration Tester "Queue Suffix" Field Value&gt;</i> ".
Queue Aliasing	Rational Integration Tester creates a topic called " <i>TTTTT_&lt;Rational Integration Tester "Queue Suffix" Field Value&gt;</i> " and queue called " <i>QQQQQ_&lt;Rational Integration Tester "Queue Suffix" Field Value&gt;</i> ".

---

If reply queues are specified, suffixed versions of those queues will also be created.

---

---

If **Dynamic Mirror Queues** is clicked, an additional **Use Default Model Queue** check box and **Model Queue** field are displayed.

The following sections describe all of the recording modes in detail.

### 2.2.3.1 Queue Browsing Explained

In **Queue Browsing** mode, Rational Integration Tester uses IBM WebSphere MQ's browse facility to capture messages. To record IBM WebSphere MQ messages, Rational Integration Tester polls the queue at a predefined interval and copies messages to the Events View.

**NOTE:** In **Queue Browsing** mode, it is possible that an event could be placed on the queue and removed from the queue before it is seen by Rational Integration Tester. Such events would not be recorded, and they would not be listed in the Events View. Additionally, if you stop recording an operation and start again, all of the events on the queue will be placed in the Events View, even if they had been recorded previously.

### 2.2.3.2 Proxy Queues Explained

In **Proxy Queues** mode, Rational Integration Tester will monitor a “proxy” request queue (denoted by combining the actual queue name with the **Queue Suffix**), record and remove any messages that are received, display these in recording studio, then pass the message on to the original queue. Request-based messages that are passed to the proxy queue will be amended so that any reply queues that they specify are also converted to use the corresponding proxy queue.

**Proxy Queues** mode enables the Recording Studio perspective to show all of the original message details, and it allows tests and stubs to be created that use the original (non-proxy) queue names.

Before **Proxy Queues** mode can be used, the system administrator needs to set up the proxy queues. A proxy queue is named the same as the request or reply queue, except for the addition of the **Queue Suffix** (specified in the IBM WebSphere MQ transport). Additionally, message initiators should be configured to send their requests to the proxy queue.

**NOTE:** Proxy queues are required for both the request queue and for the reply queue.



---

Within the Recording Studio perspective and in transport's bindings (in the definition of the containing operation), the original queue names should be used. These will be "proxied" and "de-proxied" by Rational Integration Tester when recording.

A "working" example of the process is described in the following steps:

1. The client sends a request to Q1.RIT, specifying Q2 for the reply (note that the reply queue is not "proxied" by the initiator).
2. Rational Integration Tester (Recording Studio) picks up the request on Q1.RIT, adds it to the list of recorded events, "proxies" the reply queue from Q2 to Q2.RIT, and "de-proxies" the request message by placing it on Q1.
3. The server picks up the message on Q1, processes it, and sends the reply to Q2.RIT.
4. Rational Integration Tester (Recording Studio) picks up the reply on Q2.RIT, adds it to the list of recorded events, and "de-proxies" the reply message by placing it on Q2.
5. The client receives the reply on Q2.

**NOTE:** De-proxying occurs only while recording is taking place.

### 2.2.3.3 Mirror Queues Explained

**NOTE:** This recording mode is available only if IBM WebSphere MQ 7.0 (but not 7.1 or later) is installed and configured in Library Manager.

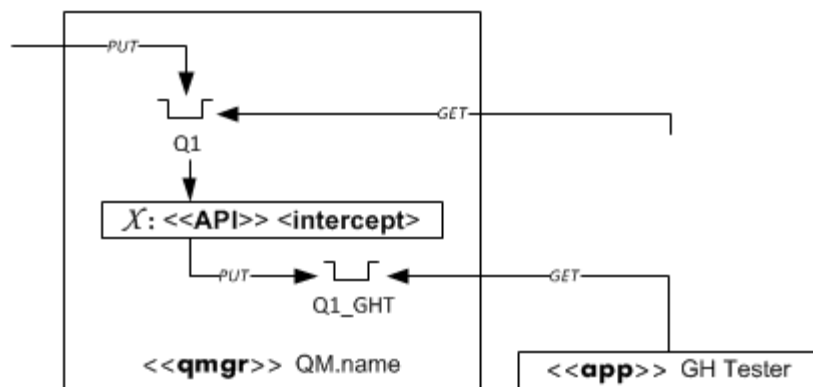
In **Mirror Queues** mode, Rational Integration Tester will use an IBM WebSphere MQ API Exit, which is installed on the queue manager, to take a copy of each message and put it into another queue. (For general information about IBM WebSphere MQ API Exits, refer to IBM WebSphere MQ documentation or go to [ibm.com](http://ibm.com)®.) Therefore, if you record multiple queues, multiple mirror queues will be set up.

For information about deploying the API Exit, refer to [Deploying Rational Integration Tester's API Exit](#).

A "working" example of the process is described in the following steps:

1. Start recording on Q1.
2. Rational Integration Tester (Recording Studio) configures the API Exit to mirror all messages put onto Q1 to Q1\_RIT.
3. The client puts message onto Q1 with the reply queue specified as Q1R.

- 
4. The queue manager triggers the API Exit, which puts a copy of the message onto Q1\_RIT.
  5. Rational Integration Tester (Recording Studio) subscribes to this queue and gets the message. Rational Integration Tester detects that message specified a reply of Q1R. At this point, Rational Integration Tester will configure the API Exit to mirror all messages from Q1R to Q1R\_RIT.
  6. The server picks up the message from Q1 and replies to Q1R.
  7. The queue manager triggers the API Exit, which puts a copy of the message onto Q1R\_RIT.
  8. Rational Integration Tester (Recording Studio) subscribes to this queue and gets the message.
  9. The client receives the reply from Q1R.



When recording is stopped, Rational Integration Tester will, by default, delete the mirrored queue (Q1\_RIT and Q1R\_RIT in the above example) and tell the API exit to stop mirroring. This ensures that the mirrored queues do not fill up with messages when you are not recording.

#### 2.2.3.4 Dynamic Mirror Queues Explained

**NOTE:** This recording mode is available only if IBM WebSphere MQ 7.0 (but not 7.1 or later) is installed and configured in Library Manager.

**Dynamic Mirror Queues** mode is similar to **Mirror Queues** mode. The main difference between the two modes is that if **Dynamic Mirror Queues** mode is selected, IBM WebSphere MQ will clear queues automatically.

---

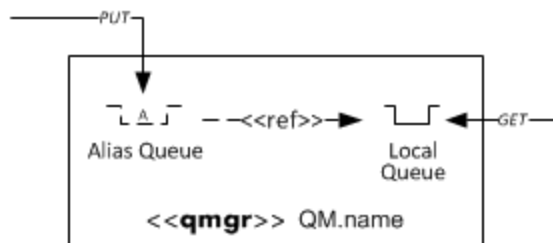
Clicking **Dynamic Mirror Queues** in the **Recording Mode** list displays an additional **Use Default Model Queue** check box and an additional **Model Queue** field. Selecting the **Use Default Model Queue** check box (default setting) enables you to accept the default model queue provided by IBM WebSphere MQ for this recording mode. If you want specify a different model queue, clear the **Use Default Model Queue** check box and enter a queue name in the **Model Queue** field.

### 2.2.3.5 Queue Aliasing Explained

**NOTE:** This recording mode is available only if IBM WebSphere MQ 7.0 (but not 7.1 or later) is installed and configured in Library Manager.

An alias queue is an IBM WebSphere MQ object that refers to another queue. The object to which the alias is pointing can be changed without affecting the client application, which continues to reference the alias.

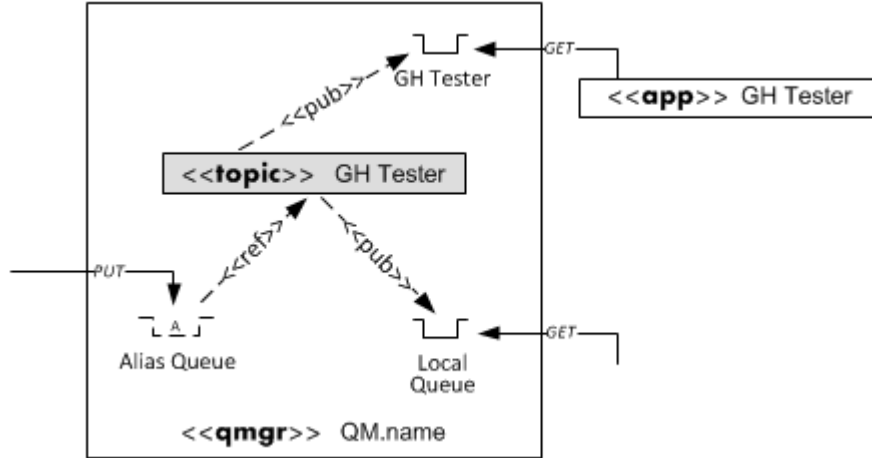
This recording approach can be used where the publishing application uses a queue alias name rather than an actual queue name.



In Queue Aliasing mode, when recording starts, Rational Integration Tester will create a topic automatically and reconfigure the queue alias to publish to the topic. Rational Integration Tester then sets up two subscribers to the topic:

- One subscriber is the original queue (so that the subscriber still receives the message).
- The other subscriber is a queue used for Recording Studio.

**NOTE:** Recording Studio will warn you if it is unable to perform the required re-configuration.



When recording is stopped, Rational Integration Tester will reconfigure the queues back to the original configuration.

If this is not possible (for example, if there is an open connection to the alias), Rational Integration Tester will display a warning message that it could not perform the reconfiguration. In such cases, manual reconfiguration using the queue manager may be required.

Alternatively, starting and stopping the recording again will force Rational Integration Tester to reattempt the reconfiguration; however, this should be attempted only after disconnecting the applications that are holding open any connections to the queue alias.

## 2.2.4 Record-The-Transport

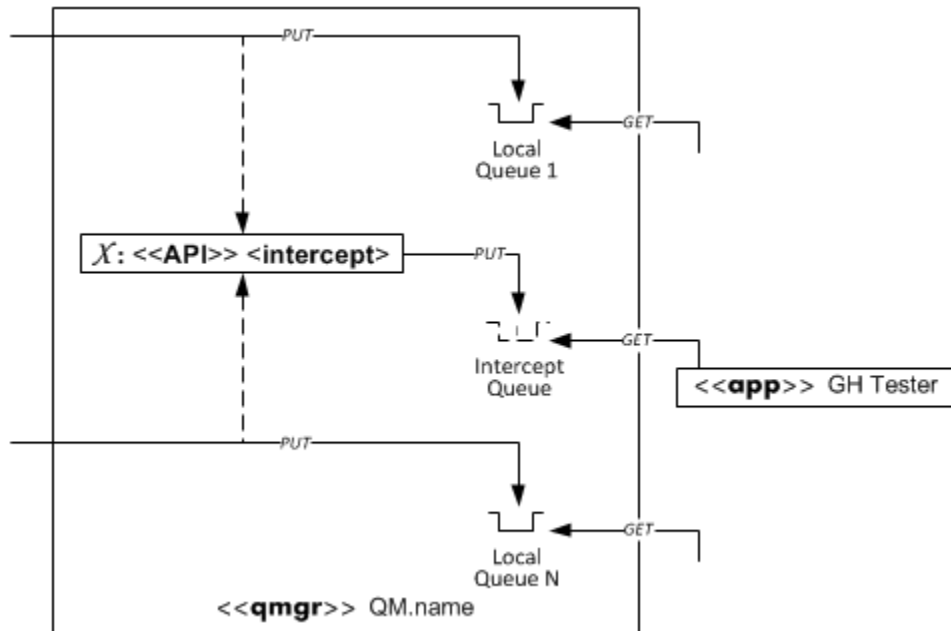
**NOTE:** This is available only if IBM WebSphere MQ 7.0 (but not 7.1 or later) is installed and configured in Library Manager.

To use this recording approach, you must do either of the following:

- Select the queue manager in Architecture School's Logical View and choose record (from top or context menu).
- Add an event monitor in Recording Studio and choose the MQ transport.

When you use this approach, the "Record" setting on the Recording tab of the physical item is ignored by Rational Integration Tester and the "Record-The-Transport" approach is used instead.

This approach requires using Rational Integration Tester's API Exit, which enables Rational Integration Tester to receive copies of messages posted to large number (possibly all) queues hosted on a queue manager at once. (For general information about IBM WebSphere MQ API Exits, refer to IBM WebSphere MQ documentation or go to [ibm.com](http://ibm.com).)

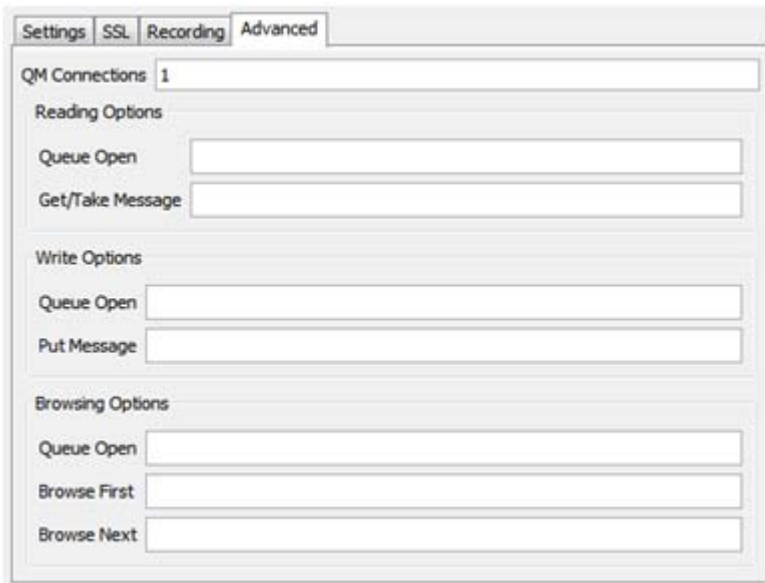


For information about deploying the API Exit, refer to [Deploying Rational Integration Tester's API Exit](#).

---

## 2.2.5 Advanced Settings

The number of queue manager connections to use, as well as Read, Write, and Browsing options can be set under the **Advanced** tab.



The screenshot shows a configuration window with four tabs: Settings, SSL, Recording, and Advanced. The Advanced tab is selected. It contains several input fields organized into sections: 'QM Connections' with a value of 1; 'Reading Options' with 'Queue Open' and 'Get/Take Message' fields; 'Write Options' with 'Queue Open' and 'Put Message' fields; and 'Browsing Options' with 'Queue Open', 'Browse First', and 'Browse Next' fields.

The default number of queue managers is set to 1. To use a pool of connection managers, enter the desired value in the **QM Connections** field.

**NOTE:** If you are running stubs that subscribe to IBM WebSphere MQ queues, the number of queue manager connections should be at least the number of stubs subscribing to this operations on this transport that are required to run simultaneously plus one.

To use any advanced read/write/browse options, enter the applicable details under **Reading Options**, **Write Options**, and **Browsing Options**.

# Working with IBM WebSphere MQ Messages

## **Contents**

**IBM WebSphere MQ Message  
Formats**

**Publishing Messages**

**Subscribing to Messages**

This chapter describes how to create publishers and subscribers to send and receive IBM WebSphere MQ messages in Rational Integration Tester.

---

## 3.1 IBM WebSphere MQ Message Formats

In order that a receiving system can best interpret the received payload, IBM WebSphere MQ provides the ability to indicate the format of user data. User data may contain one or more blocks that can be chained together. Each block includes the format of the next item in the chain. The last block is the user data.

**NOTE:** IBM WebSphere MQ 7.0 (but not 7.1 or later) supports message properties, which are used to add optional fields to a message. Categories of optional fields are as follows: JMS optional header fields, application-specific properties, and provider specific properties. When used with IBM WebSphere MQ 7.0 (but not 7.1 or later), Rational Integration Tester also supports message properties and exposes them as header fields in messaging actions. (For more information about message properties, refer to IBM WebSphere MQ documentation or go to [ibm.com](http://ibm.com).)

For example, most publish and subscribe messages generated by the JMS support in IBM WebSphere MQ contain a version one and version two Rules and Formatting Header, followed by the message payload.

In Rational Integration Tester, the chaining together of blocks is achieved by including the optional blocks as part of the message payload, meaning that they will appear as optional fields on the root of the message when the **WebSphere MQ** formatter is selected.

Similar to the format, the code page and encoding settings for the header and RF headers should follow a similar pattern (that is, the setting in the message header should match the first header, and the setting in each RFH should match the settings of the data that immediately follows the particular RFH), but this is left to the control of the user.

### 3.1.1 RFH1 and RFH2 Content Blocks

A Rules and Formatting Header is an optional data block that gives the receiving application information about the format of the message data. There are two versions of the header:

- Version 1 (RFH1)
- Version 2 (RFH2)

**NOTE:** In messages, an RFH2 content block may not be used after an RFH1 content block.



---

**NOTE:** In IBM WebSphere MQ 6.0 (or earlier), RFH2 headers are used to store MQ JMS properties. However, if IBM WebSphere MQ 7.0 (but not 7.1 or later) is being used, properties are added as MQ properties to each message instead of adding an RFH2 block. In addition, all MQ JMS properties are stored as a property. (For more information about how IBM WebSphere MQ 6.0 (or earlier) and 7.0 (but not 7.1 or later) handle RFH2 blocks, refer to IBM WebSphere MQ documentation or go to [ibm.com](http://ibm.com).)

Each block contains a fixed portion of data followed by an optional, variable portion. The fixed data identifies and describes the Rules and Formatting Header itself, while the variable portion contains optional data fields.

In the fixed portion of the block, the version, length and format fields will be set automatically. The CCSID field, which is present only in the version 2 format, denotes the code page used for the variable portion of the header. CCSID may be one of four values (1200, 1208, 13488, 17584), and should be set to code page 1208 in most cases.

The variable portion of a version one header contains name-value pairs that can define the format of the user data (the application group and format name) and fields that are related to publish and subscribe requests. These are interpreted by Rational Integration Tester as a plain string of data.

The variable portion of a version two header is divided into a number of XML folders:

- The **mcd** folder contains up to four fields that define the data type and format (that is, message domain, set, type, and format). The domain indicates the parser that will process the message data, while the parser determines the meaning of the other three fields.
- The **jms** folder contains several fields that are populated by the IBM WebSphere MQ implementation of JMS.
- The variable portion can also contain folders related to publish and subscribe, user data, and other miscellaneous, user-defined folders.
- All folders are optional, and more than one folder may be present in the same RF Header.

---

### 3.1.2 Other Content Blocks

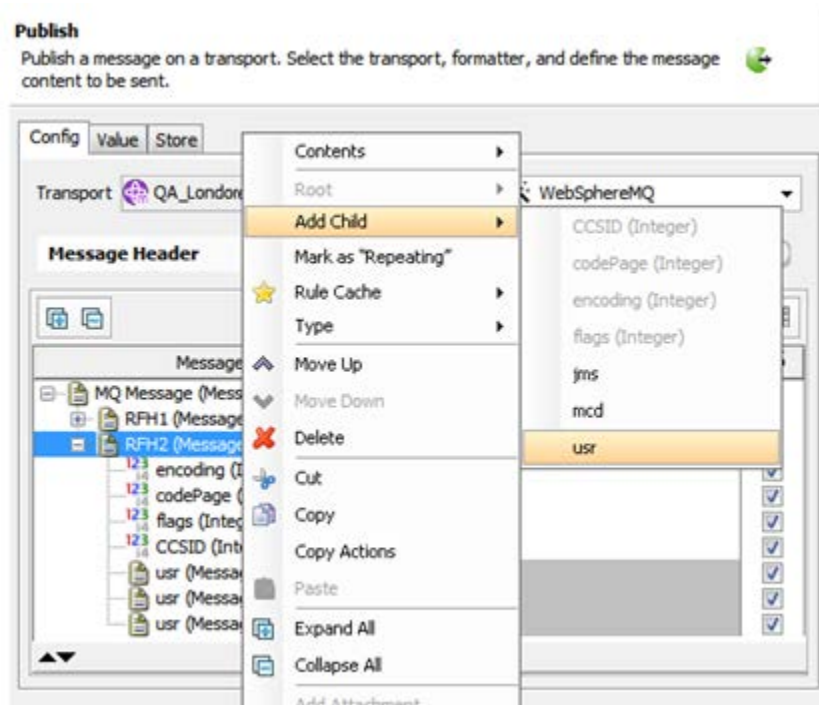
Other folder types, such as CICS, IMS, Dead Letter, and so on are not supported explicitly by Rational Integration Tester, but they will be processed as a content block containing a byte array. Since Rational Integration Tester is unaware of the length of these blocks, all of the remaining data from the body of the message will be included as part of this header.

The **Unknown header** element from the message schema can be used to send these blocks. In such a case, the name of the container will be used to populate the format field in the preceding block, or in the message header if this is the first.

### 3.1.3 Creating RF Headers

To add headers to an IBM WebSphere MQ message, click **WebSphereMQ** in the **Formatter** list and add the headers to the **MQ Message** root using the context menu.

**NOTE:** An RFH2 content block may not be used after an RFH1 content block in Rational Integration Tester messages.



**NOTE:** The **jms**, **mcd**, and **usr** child elements can be added to RFH2 blocks.

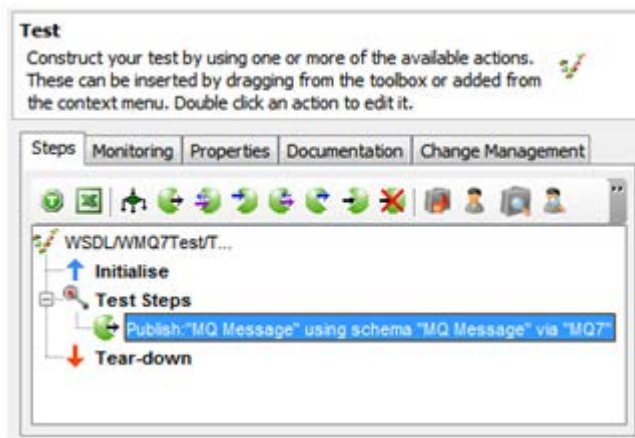
---

## 3.2 Publishing Messages

MQ messages are put on to queues using the Publish action in Rational Integration Tester tests. This section provides information about how to configure a publisher that uses the MQ transport.

### 3.2.1 Creating a Publish Action

The first step in publishing MQ messages is to create a test in Rational Integration Tester's Test Factory perspective and add a Publish action to it.



**NOTE:** A test can be created only within an existing operation.

**NOTE:** For information about creating operations and tests, and adding actions to tests, refer to *IBM Rational Integration Tester Reference Guide*.

---

### 3.2.2 Selecting the Transport and Formatter

1. In the Test Factory, open the test that contains the Publish action.
2. Double-click the Publish action to open it for editing.
3. If necessary, click the **Config** tab.

The **Transport** field and **Formatter** list are displayed at the top of the tab.



4. Click **Browse** to select an MQ transport in the Select a Resource dialog box.

Select the option for obeying references. If you select **No**, you can select any transport in the project. If you select Yes, you can select only transports that are referenced (directly or indirectly) by the operation containing the test. If you select Only Direct, you can select only transports that are referenced directly by the operation containing the test.

5. Locate and select the desired transport in the resource tree.
6. Click **OK** when finished.
7. In the **Formatter** list, click the format of the published message.

---

### 3.2.3 Configuring the Message Header

MQ message headers are configured using five tabs. For the purpose of publishing a simple message, only a few of the options under the **General** tab need to be set.

1. Enter the name of the local MQ queue (for example, `default`). A queue called “default” should be installed on the queue manager following installation.

**Publish**  
Publish a message on a transport. Select the transport, formatter, and define the message content to be sent.

Config Value Store

Transport QA\_London ... Browse... Formatter WebSphereMQ

**Message Header**

General Report Context Identifiers Segmentation

Queue  Reply Queue  Subscribe C

Message Type Datagram ▼ Reply Manager  Encoding

Message Persistence Persistent ▼ Priority 0 Expiry Interval -1

Message	Value
MQ Message (Message)	Process Children
RFH1 (Message)	Process Children
RFH2 (Message)	Process Children

▲▼

2. In the **Message Type** list, click **Request**.
3. Double-click the value field next to the **text (String)** field to enable editing.
4. Enter a message (for example, This is a Rational Integration Tester message).
5. Press ENTER or click somewhere else in the dialog box to save the message.
6. Click **OK** to close the message editor.

**NOTE:** If desired, the MQMD (Message Queue Message Descriptor) can be configured using the additional header tabs, including details such as the message and correlation IDs, reporting, segmentation, and context details.

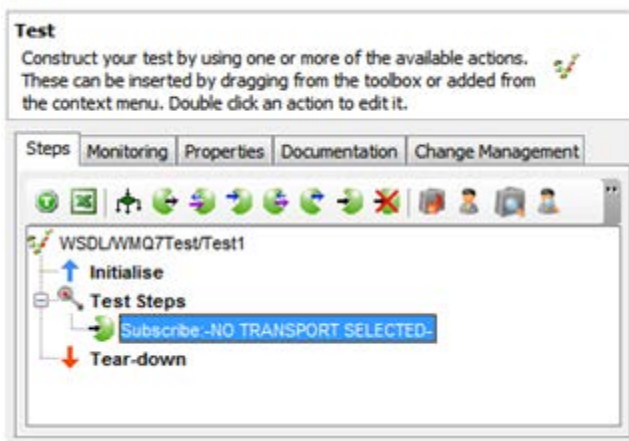
---

## 3.3 Subscribing to Messages

MQ messages can be retrieved from a queue using the Subscribe action in Rational Integration Tester tests. This section provides information about how to configure a subscriber that uses the MQ transport.

### 3.3.1 Creating a Subscribe Action

The first step in subscribing to MQ messages is to create a test in Rational Integration Tester's Test Factory perspective and add a Subscribe action to it.



**NOTE:** A test can be created only within an existing operation.

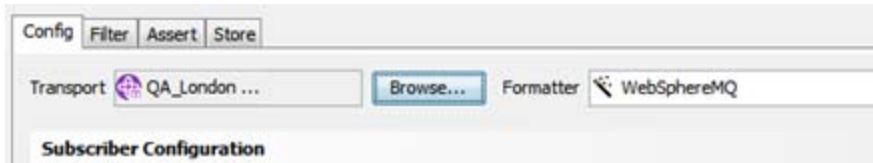
**NOTE:** For information about creating operations and tests, and adding actions to tests, refer to *IBM Rational Integration Tester Reference Guide*.

---

### 3.3.2 Selecting the Transport and Formatter

1. In the Test Factory, open the test that contains the Subscribe action.
2. Double-click the Subscribe action to open it for editing.
3. If necessary, click the **Config** tab.

The **Transport** field and **Formatter** list are displayed at the top of the tab.



4. Click **Browse** to select an MQ transport in the Select a Resource dialog box.

Select the option for obeying references. If you select **No**, you can select any transport in the project. If you select **Yes**, you can select only transports that are referenced (directly or indirectly) by the operation containing the test. If you select **Only Direct**, you can select only transports that are referenced directly by the operation containing the test.

5. Locate and select the desired transport in the resource tree.
6. Click **OK** when finished.
7. In the **Formatter** list, click the format of the message.

---

### 3.3.3 Configuring the Subscriber Details

Subscribing to MQ message requires only the name of the local queue from which messages should be reviewed or retrieved. Additional filtering options, however, are available if needed.

1. Enter the name of the local MQ queue (for example, `default`). This should be the same queue to which the message was previously published.

**Subscribe**  
Choose a transport and formatter on which to receive and validate a message.

Config Filter Assert Store

Transport QA\_London ...  Formatter WebSphereMQ

**Subscriber Configuration**

Queue

Message Selector

Message ID

Correlation ID

Group ID

Logical Sequence Number  Offset

Message	Value
MQ Message (Message)	Validate Message Children

Timeout (ms)  Tolerance (ms)

2. Select the **Browse** check box to leave messages on the queue. Clear the **Browse** check box to remove messages from the queue.
3. If desired, enter filter criteria for messages under the **Message Selector** section. If no filtering is required, these fields can be left blank.
4. Enter the same message (for example, This is a Rational Integration Tester message) in the **text (String)** field.
5. Enter a time-out value, if desired, to set the number of milliseconds after which the subscriber will stop waiting for the message (the default, 0, indicates that the subscriber will wait indefinitely).
6. Click **OK** to close the message editor.



# Using JMS to Access IBM WebSphere MQ

## **Contents**

### **Introduction**

### **Creating JMS-Administered Objects**

### **Sending Messages from Rational Integration Tester**

This chapter describes how to use the JMS interface to interact with IBM WebSphere MQ.

---

## 4.1 Introduction

IBM WebSphere MQ classes for Java Message Service (also referred to as IBM WebSphere MQ JMS), which is available as part of IBM WebSphere MQ client support, is a set of Java classes that:

- Enables JMS applications to access IBM WebSphere MQ systems.
- Supports the point-to-point and publish/subscribe models of JMS.

A JMS application comprises the following:

- **A JMS provider:** A messaging system that implements the JMS specification.
- **JMS clients:** Java applications that send and receive messages.
- **Messages:** Objects that are used to communicate information between JMS clients.
- **Administered objects:** Preconfigured JMS objects that are created by an administrator for the use of JMS clients.

A JMS application is written to use only references to the interfaces in the `javax.jms` package. All vendor-specific information is encapsulated in implementations of the following JMS-administered objects:

- `QueueConnectionFactory`
- `TopicConnectionFactory`
- `Queue`
- `Topic`

These JMS-administered objects are built using a vendor-supplied administration tool and stored on a JNDI namespace. A JMS application can retrieve these objects from the namespace and use them without needing to know which vendor provided the implementation, which, in this case, is the IBM WebSphere MQ JMS provider.

JMSAdmin is the IBM WebSphere MQ JMS administration program. You can use JMSAdmin to create administered objects that map to the IBM WebSphere MQ objects, such as a queue manager, queue, topic, and so on.

The Java classes of the IBM WebSphere MQ JMS provider communicate with the queue manager using one of the following transport options:

- As an IBM WebSphere MQ client using Transmission Control Protocol/Internet Protocol (TCP/IP).
- In bindings mode, connecting directly to IBM WebSphere MQ.

---

For more information about JMS, visit the Oracle website. For more information about IBM WebSphere MQ JMS, refer to IBM WebSphere MQ documentation or go to [ibm.com](http://ibm.com).

## 4.2 Creating JMS-Administered Objects

IBM WebSphere MQ includes several administration tools that can be used to create the JMS-administered objects. These hold the configuration settings to be used to connect to new or existing queues on the queue managers.

The method that can be used to make all necessary resources available to IBM WebSphere MQ JMS programs will depend on the version of IBM WebSphere MQ that is being used:

WebSphere MQ Version	Manual Configuration Method	WebSphere MQ Explorer Method
5.x (or earlier)	Yes	No
6.0 (or later)	Yes	Yes

Both methods are easy to use; however, the IBM WebSphere MQ Explorer method has an advantage over the Manual Configuration method because IBM WebSphere MQ Explorer enables users to configure connection factories.

The following table outlines the configuration settings that must be changed.

Configurable Item	Settings
JMSAdmin.config file	<p>1. Edit the INITIAL_CONTEXT_FACTORY lines as follows:</p> <pre>#INITIAL_CONTEXT_FACTORY=com.sun.jndi.ldap.LdapCtxFactory INITIAL_CONTEXT_FACTORY=com.sun.jndi.fscontext.RefFSContextFactory #INITIAL_CONTEXT_FACTORY=com.ibm.ejs.ns.jndi.CNInitialContextFactory #INITIAL_CONTEXT_FACTORY=com.ibm.websphere.naming.WsnInitialContextFactory</pre> <p>2. Edit the PROVIDER_URL lines as follows:</p> <pre>#PROVIDER_URL=ldap://polaris/o=ibm,c=us PROVIDER_URL=file:/C:/JNDI-Directory #PROVIDER_URL=iiop://localhost/</pre>

---

---

Configurable Item	Settings
JMS Admin tool	<p>The command prompt enables you to configure the following objects:</p> <ul style="list-style-type: none"><li>• MQQueueConnectionFactory</li><li>• MQTopicConnectionFactory</li><li>• MQQueue</li><li>• MQTopic</li></ul> <p><b>Example:</b> Modify the MQQueueConnectionFactory object as follows:</p> <ul style="list-style-type: none"><li>• QueueConnectionFactory=ConnectionFacto ry</li><li>• Description=JMSTest Connection</li><li>• Transport=CLIENT</li><li>• MQ Queue Manager=JMSTest</li><li>• Port for MQSeries Server=1415</li><li>• MQSeries Server Host=&lt;IP address / Computer name&gt;</li><li>• Queue Manager Channel=JMSChannel</li></ul> <p><b>Example:</b> Modify the MQQueue object as follows:</p> <ul style="list-style-type: none"><li>• Queue Object=QueueRef</li><li>• Description=Queue reference</li><li>• MQ Queue Manager=JMSTest</li><li>• Message Expiry=UNLIM</li><li>• Message Priority=QDEF</li><li>• Target Client=JMS or MQ</li><li>• Actual Physical Queue=JMSQueue</li></ul> <p><b>NOTE:</b> Using MQ for the Target Client will result in the removal of RFH1 and RFH2 headers from the message when sent by means of JMS.</p>

---

For more information about these two methods of creating JMS-administered objects, refer to IBM WebSphere MQ documentation or go to [ibm.com](http://ibm.com).

---

## 4.3 Sending Messages from Rational Integration Tester

It is possible to both send and receive IBM WebSphere MQ messages within Rational Integration Tester. For this example, the `MQQueueConnectionFactory` and `MQQueue` definitions configured previously will be used to send and receive a message to the `JMSQueue` queue located on the `JMSTest` queue manager.

### 4.3.1 Creating a JMS Transport

This section describes how to create and configure a JMS transport. (For general information about creating a transport, refer to [Setting Up IBM WebSphere Transports](#).)

1. In Architecture School's Logical View, create a queue-based JMS domain and assign a meaningful name to it.
2. In the Physical View, create a queue-based JMS Broker and call it **JMSTransport** (or an equivalent name).
3. Bind the logical and physical resources in a new or current environment.
4. Open the JMS transport (in the Physical view) for editing.
5. Enter the following properties under **JNDI Settings** and **Connection Settings**:
  - Initial Context Factory: `com.sun.jndi.fscontext.RefFSContextFactory`
  - Provider URLs: `file:/C:/JNDI-Directory`
  - ConnectionFactory: `ConnectionFactory`
6. When finished, click **OK** to save the transport.

---

### 4.3.2 Publishing a Message

This section describes how to create and configure Publish action. (For more information, refer to [Working with IBM WebSphere MQ Messages](#).)

1. In a new or existing test, create a Publish action and open it for editing.
2. Select the transport that was created in [Creating a JMS Transport](#) (“JMSTransport”).
3. In the **Formatter** list, click **Text**.
4. In the **Queue** field, enter QueueRef.

**NOTE:** This is the JMS reference to the queue, not the name of the local MQ queue.

5. Double-click the **Value** field next to the text (String) field and enter Test message from Rational Integration Tester.
6. Click **OK** to close the message editor.

When the test is run, a message will be persisted to the MQ queue by means of the JMS API. This can be repeated to publish information to any queue that has been configured for access by means of the JMS-administered objects.

# Deploying Rational Integration Tester's API Exit

## **Contents**

### **Introduction**

### **Installing the API Exit File**

### **Configuring a Queue Manager to Use the API Exit**

This chapter describes how to install and configure Rational Integration Tester's API Exit for IBM WebSphere MQ.

---

## 5.1 Introduction

The Mirror Queue recording mode and Record-The-Transport in Rational Integration Tester require deploying the API Exit that is supplied with Rational Integration Tester. (For information about the Mirror Queue Recording mode and Record-The-Transport, refer to [Recording Settings](#). For general information about WebSphere MQ API Exits, refer to IBM WebSphere MQ documentation or go to [ibm.com](http://ibm.com).)

**NOTE:** This installation process requires administrative access to the queue manager and to the server where the queue manager is running.

To deploy Rational Integration Tester's API Exit:

1. Install the relevant API Exit file on the server hosting the queue manager that is to be configured.
2. Configure the relevant queue manager to use the API Exit.

The following sections describe these steps in detail.

## 5.2 Installing the API Exit File

To install Rational Integration Tester's API Exit on the server hosting the queue manager that is to be configured:

1. On the computer where Rational Integration Tester is installed, find the following directory:

```
<Rational Integration Tester Installation  
Directory>\tools\IBM\MQ\7.0
```

**NOTE:** Rational Integration Tester does not have to be installed on the server that hosts the queue manager that is to be configured.

The directory contains intercept exit modules for supported platforms. Intercepts are named as follows:

```
intercept_<Operating System>_<Processor>_<IBM WebSphere MQ  
Version>
```

Modules for Microsoft Windows have a .dll extension. For some platforms, there is a variant with an \_r extension, which means that both modules should be copied into the appropriate exits or exits64 directory. However, for configuration purposes, only the module without the \_r extension is referenced.



- 
2. Copy the relevant file(s) to the appropriate directory on the server hosting the queue manager that is to be configured.

For example:

- Windows 32-bit API Exit: <IBM WebSphere MQ Installation Directory>\exits\  
Directory>\exits\  
Directory>\exits64\  
Directory>\exits64\
- Windows 64-bit API Exit: <IBM WebSphere MQ Installation Directory>\exits64\
- Linux API Exit: /var/mqm/exits

**NOTE:** The file(s) must be copied to the correct location. Otherwise, IBM WebSphere MQ will display error messages and/or the API Exit will not work (correctly).

## 5.3 Configuring a Queue Manager to Use the API Exit

**NOTE:** You need to have read/write permissions to the IBM WebSphere MQ installation on the server that is hosting the queue manager that is to be configured. In addition, the queue manager will need to be restarted during this procedure.

To configure a queue manager to use the API Exit:

1. Open IBM WebSphere MQ Explorer.

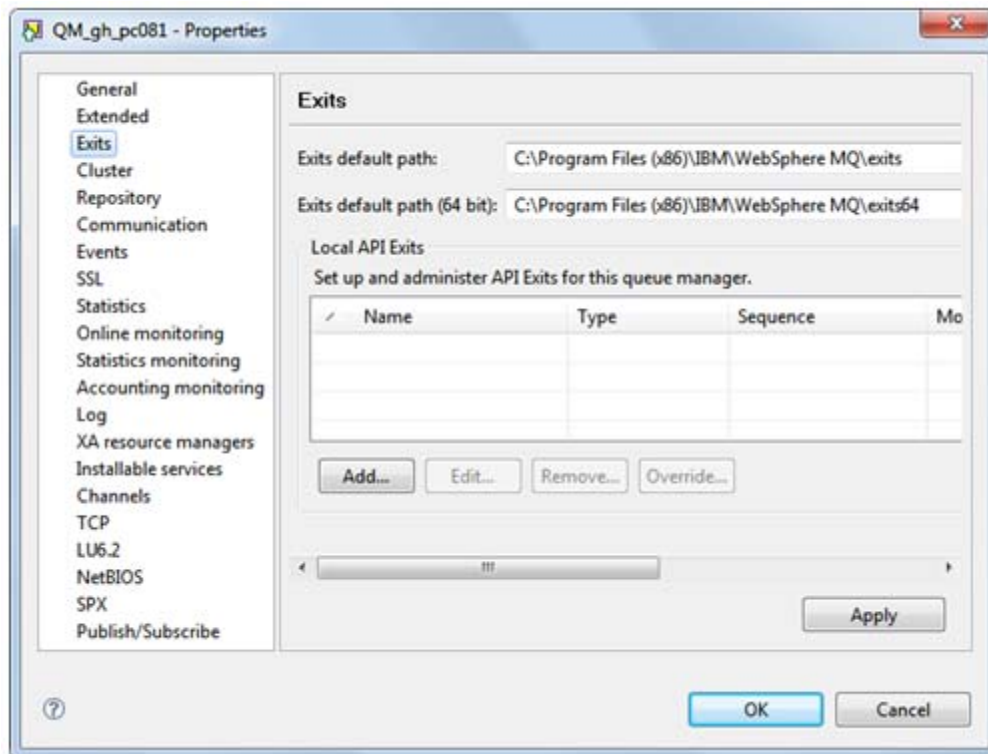
**NOTE:** IBM WebSphere MQ Explorer must be running on the server hosting the queue manager that is to be configured. This is because it is not possible to configure API Exits remotely.

2. On the MQ Explorer Navigation window, right-click the relevant queue manager and click **Properties** on the shortcut menu.

A Properties dialog box is displayed.

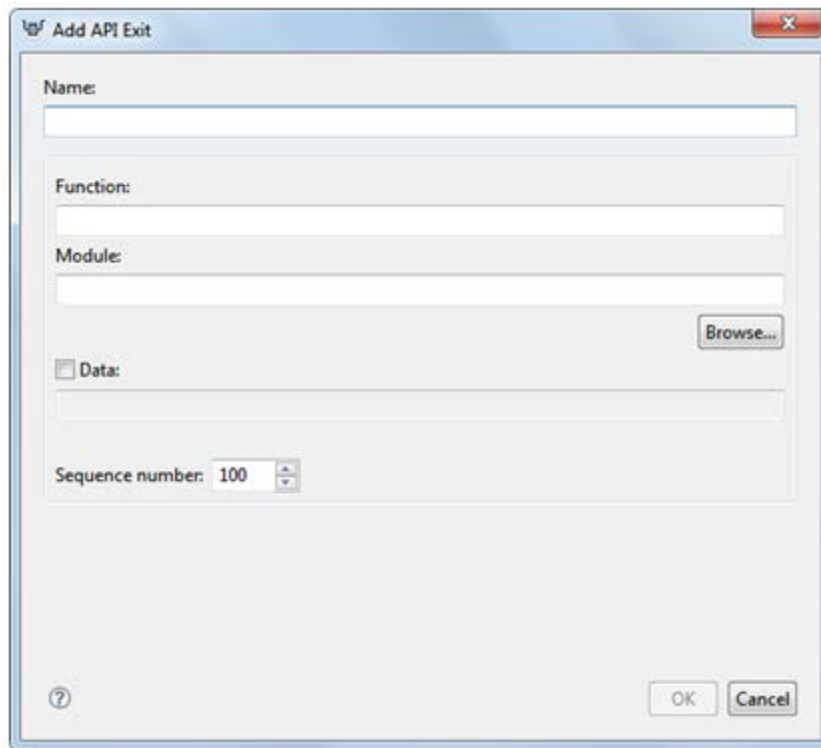
3. Click **Exits** on the navigation pane of the Properties dialog box.

The Exits page is displayed.



4. On the Exits page, click **Add**.

The Add API Exit dialog box is displayed.



5. In the **Name** field, enter Intercept.
6. In the **Function** field, enter EntryPoint.
7. Under the **Module** field, click **Browse** and navigate to the appropriate directory (refer to [Installing the API Exit File](#)).
8. Select the appropriate file (refer to [Installing the API Exit File](#)) and click **Open**.
9. Select the **Data** check box.
10. In the **Data** field, enter `com.greenhat.intercept`.

**NOTE:** The **Sequence number** box is not relevant to Rational Integration Tester.

11. Click **OK** to close the Add API Exit dialog box.
12. On the Exits page, click **Apply**.
13. Click **OK** to close the Properties dialog box.
14. On the MQ Explorer Navigation window, restart the queue manager.

The specified API Exit file is deployed on the specified queue manager.

# Troubleshooting

## **Contents**

**IBM MQ Error 2009 (MQRC Connection Broken)**

**IBM MQ Error 2035 (MQRC Not Authorized)**

**IBM MQ Error 2374 (MQRC API Exit Error)**

**Unable to Edit Transport or Formatter Unavailable**

**Messages Not Recording When Using API Exits**

**Namelists for API Exit Not Created**

This chapter provides answers to common questions and issues that may arise when using IBM WebSphere MQ with Rational Integration Tester.

---

## 6.1 IBM MQ Error 2009 (MQRC Connection Broken)

When using a publisher, if the error “Publish Error putting (publishing) message - An error occurred while putting the message on the queue “default”. Cause: MQJE001: Completion Code 2, Reason 2009” is displayed, verify that all the transport details are correct and that the correct queue manager is connected.

A common reason for this is that the channel on the queue manager has not been setup correctly. This channel needs to be configured as a Server Connection channel and have suitable access rights applied to it.

## 6.2 IBM MQ Error 2035 (MQRC Not Authorized)

When using any of the messaging operations, access to the queue will be made; if the settings on the queue differ from the defaults within the application, you may need to override the values. The actual values can be obtained from the IBM documentation or by contacting IBM Support.

A quick way to overcome most permission errors is to configure the server connection channel to use the appropriate IBM WebSphere MQ administrator user account (or mqm, if using Unix), which has privileges to access most objects.

## 6.3 IBM MQ Error 2374 (MQRC API Exit Error)

This error indicates that an API Exit function returned an invalid response code, or failed in some other way. The most likely cause of this error is that an API Exit's file(s) has (have) been copied to the incorrect directory of the server running the queue manager that is supposed to use the API Exit.

For more information about this, refer to [Deploying Rational Integration Tester's API Exit](#).

---

## 6.4 Unable to Edit Transport or Formatter Unavailable

If you are unable to edit a transport or if the formatter combo box is blank, it is likely that the requisite JAR files have not been configured using the Library Manager.

## 6.5 Messages Not Recording When Using API Exits

When you start recording queues and you are using Rational Integration Tester's API Exit, Rational Integration Tester will configure the queue manager automatically so that it puts copies of messages into the queues used by Rational Integration Tester. However, this configuration will apply only to connections made to the queue manager after you have started the recording.

If any process maintains a connection to the queue manager before recording started, you will not see events put into these queues. Depending on how your publishing applications are written, you may need to restart them after recording starts.

Rational Integration Tester adds reply queues to those being recorded automatically when it sees a message being sent with a reply queue. However, if the subscriber already has an open connection to the queue manager, you will not see it in Recording Studio when it puts a response to the reply queue. Restarting the subscriber after processing one message may work around this problem. Alternatively, use the Record-the-Transport option to record all queues.

(For more information about the API Exits supplied with Rational Integration Tester, refer to [Deploying Rational Integration Tester's API Exit](#).)

## 6.6 Namelists for API Exit Not Created

A namelist is an IBM WebSphere MQ object that contains a list of names of other objects.

To create a namelist for Rational Integration Tester's API Exit:

1. On the MQ Explorer Navigation window, click the relevant queue manager.

**NOTE:** There is no need to stop the queue manager.

2. Click **Advanced**.
3. Right-click **Namelists**, and click **New** and **Namelist** on the shortcut menus.

The New Namelist wizard is displayed.

4. In the **Name** field, enter `com.greenhat.intercept`.

- 
5. Click **Next**.

The General page is displayed.

6. Click **Edit**.

The Edit Names dialog box is displayed.

7. Click **Add**.

The Add to Names dialog box is displayed.

8. Enter the following:

- The name of the queue that is to be mirrored.
- The name of the mirror queue.
- The name of the queue manager where the destination is located (usually this will be the queue manager that is being configured).

**NOTE:** If there is more than one queue to be mirrored, these three values must be entered for each queue.

9. Click **OK** on the Edit Names dialog box.

10. Click **Finish** on the New Namelist wizard screen.

The specified namelist is created for the API Exit.

---

# Glossary

The following table below lists some of the key terms used in this document, and provides a description of each.

Term	Description
Field	A bit of data constituent to a message. Most fields are scalar and therefore unitary, equivalent to data attributes. Vector fields are an aggregation of fields both scalar and vector, and are usually referred to as Messages.
Message	A unit of information made up of a header consisting of meta-information and a body consisting of the message data.
Host	The computer where a software process runs.
Transport	Informally, the messaging software in use. For instance, TIBCO Rendezvous, TIBCO Active Enterprise, IBM WebSphere MQ and HTTP.
Publishing	Making a message (data) available on a message channel.
Subscribing	Receiving a stream of messages (data) on a given message channel.
Server	A host computer on a network shared by more than one user.
Queue Manager	The queue manager handles all functions not directly related to actual movement of data, for example, storing the messages.
Channel	IBM WebSphere MQ component to handle sending and receiving data over the network
MQMD	Message Queue Message Descriptor (MQMD). Contains all the header information of the message.



---

# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
1623-14, Shimotsuruma, Yamato-shi  
Kanagawa 242-8502 Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT,

---

MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM United Kingdom Limited  
Intellectual Property Law  
Hursley Park  
Winchester  
SO21 2JN  
Hampshire  
United Kingdom

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the

---

capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corporation 2001, 2012.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

---

## Trademarks and service marks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

