

Rational Integration Tester



Reference Guide for IBM z/OS

Version 8.0.1



Note

Before using this information and the product it supports, read the information in “Notices” on page 53.

This edition applies to version 8.0.1 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

About this Publication	v
Intended Audience	vi
Scope	vi
Typographical Conventions	vi
Contacting IBM Support	vi
Requirements	1
Introduction	2
Hardware Requirements	3
Software Requirements	4
Configuring Rational Integration Tester to Use the z/OS System	5
Configuring IBM WebSphere MQ	6
Transferring Files to z/OS	6
Unpacking the Files	7
Authorizing APF for the Load Library	7
Adding RITMQF to the LPA	7
Editing the JCL for Local Customization	8
Uninstalling the Rational Integration Tester MQF Exit Module	8
Configuring IBM WebSphere MQ on z/OS	9
Configuring Web Services Support	13
Configuring Rational Integration Tester for IBM IMS Connect	14
Configuring Recording for CICS Transaction Gateways	17
Setting Up Transports	19
Before Creating Transports	20
IBM WebSphere MQ	20
CICS Transaction Gateway	20

IMS Connect	20
Creating Transports	21
Creating an IBM WebSphere MQ Transport	21
Creating a CICS Transport	21
Configuring CICS Transaction Gateway Messages	24
Creating the IMS Connect Transport	26
Configuring IMS Connect Messages	29
Troubleshooting	31
Error Messages	32
Common Errors	52
Notices	53
Trademarks and service marks	56

About this Publication

Contents

Intended Audience

Scope

Typographical Conventions

Contacting IBM Support

This guide describes how to configure and run IBM® Rational® Integration Tester for testing SOA and IBM WebSphere® MQ applications on IBM z/OS® systems.

Intended Audience

This document is primarily intended for software testers and for anyone who sets up software test environments.

This document assumes that readers are familiar with software testing, IBM Rational Integration Tester, and the IBM z/OS operating system for testing IBM CICS® (Customer Information Control System) and IBM IMS™ (Information Management System) transactions.

Scope

This document discusses how to use Rational Integration Tester with IBM z/OS operating systems, using CICS and 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.
Bold	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 Menu > Submenu or Menu > Option .

Contacting IBM Support

To contact IBM Support, see: www.ibm.com/contact/us/en/

Requirements

Contents

Introduction

Hardware Requirements

Software Requirements

This chapter describes hardware and software requirements for using IBM Rational Integration Tester to test applications on z/OS systems.

1.1 Introduction

You can test applications on z/OS systems using the mainframe support capabilities of Rational Integration Tester.

Rational Integration Tester 8.0.1 (or later) supports the following features for z/OS:

- CICS Transaction Gateway transport
- CICS Transaction Gateway recording
- CICS web services support
- IMS Connect recording
- IMS Connect transport
- WebSphere MQ queue manager message recording

For more information about z/OS, CICS, and IMS, go to <http://publib.boulder.ibm.com/infocenter/zos/basics/index.jsp>

1.2 Hardware Requirements

There are no special hardware requirements for any of the z/OS features supported by Rational Integration Tester 8.0.1 (or later).

For general information about Rational Integration Tester's hardware requirements, refer to *IBM Rational Integration Tester Installation Guide*.

For information about z/OS hardware requirements, go to <http://publib.boulder.ibm.com/infocenter/zos/basics/index.jsp>.

1.3 Software Requirements

IBM z/OS must be installed and configured. For more information, see

<http://publib.boulder.ibm.com/infocenter/zos/basics/index.jsp>.

IBM Rational Integration Tester 8.0.1 (or later) and IBM Rational Test Control Panel 8.0.1 (or later) must be installed.

The following software versions are supported:

- WebSphere MQ 7.0.1 and 7.1
- CICS Transaction Gateway 8.0 and 8.1
- IMS 12 (includes IMS Connect)

The following table describes additional requirements for performing specific testing activities.

To perform these actions...	Install...
Record IMS Connect traffic using the HTTP/TCP proxy	IBM Rational Integration Tester Platform Pack HTTP/TCP proxies
Record inbound or outbound web service calls to or from CICS programs	
Run tests and stubs on non-local systems	IBM Rational Integration Tester Agent

For information, refer to *IBM Rational Integration Tester Installation Guide*, *IBM Rational Test Control Panel Installation Guide*, *IBM Rational Integration Tester Agent Installation Guide*, and *IBM Rational Integration Tester Platform Pack Installation Guide*.

Configuring Rational Integration Tester to Use the z/OS System

Contents

[Configuring IBM WebSphere MQ](#)

[Configuring Web Services Support](#)

[Configuring Rational Integration Tester for IBM IMS Connect](#)

[Configuring Recording for CICS Transaction Gateways](#)

This chapter describes the process of setting up Rational Integration Tester to work with a z/OS system.

Before you begin, the z/OS system must be installed and configured (for information about this refer to [Software Requirements](#)).

2.1 Configuring IBM WebSphere MQ

This section describes the steps for installing the Rational Integration Tester MQ exit module and configuring it for use with WebSphere MQ on a z/OS system.

This installation process involves the following high-level tasks:

1. Transfer JCL and executable files to z/OS.
2. Unpack the files.
3. APF authorize the load library.
4. Add the RITMQF exit module to the Link Pack Area (LPA).
5. Edit the Job Control Language (JCL) for local customization.
6. Uninstall the RITMQF exit module.
7. Configure IBM WebSphere MQ on z/OS.
8. Set up IBM WebSphere MQ.
9. Edit the Rational Integration Tester JCL file.
10. Run the Rational Integration Tester job.
11. Set up security for the Rational Integration Tester MQ exit.

The following sections describe these steps.

2.1.1 Transferring Files to z/OS

Use FTP to transfer the following files to z/OS in binary format.

- RIT.PROC.XMIT.BIN
- RIT.AUTH.XMIT.BIN
- RIT.LINK.XMIT.BIN

For example, to transfer the files to z/OS:

1. FTP <hostname>.
2. Enter your user ID and password.
3. Bin.
4. Quote site CYL PRI=1 SEC=1 recfm=fb lrecl=blksize=3200
5. Put RIT.PROC.XMIT.BIN 'userid.RIT.PROC.XMIT'
6. Put RIT.AUTH.XMIT.BIN 'userid.RIT.AUTH.XMIT'

-
7. Put `RIT.LINK.XMIT.BIN 'userid.RIT.LINK.XMIT'`
 8. Exit FTP.

2.1.2 Unpacking the Files

Use the TSO receive command to unpack the files, for example:

- `tso receive indsn(RIT.PROC.XMIT)`
`dsn(RIT.PROC)`
- `tso receive indsn(RIT.AUTH.XMIT)`
`dsn(RIT.AUTH)`
- `tso receive indsn(RIT.LINK.XMIT)`
`dsn(RIT.LINK)`

This creates three datasets with the following formats:

- `userid.RIT.PROC`
- `userid.RIT.AUTH`
- `userid.RIT.LINK`

2.1.3 Authorizing APF for the Load Library

An authorized person must APF-authorize the `userid.RIT.AUTH` library.

For example, you could update a `PROGxx` member in `SYS1.PARMLIB` concatenation, or temporarily define it by using the `SETPROG` `APF,ADD,DSN=userid.RIT.AUTH,SMS` operator command.

2.1.4 Adding RITMQF to the LPA

You can dynamically add the `RITMQF` module to the Link Pack Area (LPA) by using the `SETPROG LPA` command, for example:

```
setprog lpa,add,module=(RITMQF),dsname=userid.RIT.LINK
```

2.1.5 Editing the JCL for Local Customization

Customize the JCL member in `RIT.SOURCE`:

1. Change `++QMGR++` to the queue manager name.
2. Change `++INSTLIB++` the library name containing the `RIT.AUTH` code.
3. Specify the MQ libraries.
4. Change `++SYSMDUMP++` to the name of a data set, for example, `user.SYSMDUMP`.

You can run the JCL by submitting it from a library, or by creating a started task procedure. If you use a started task, you must:

1. Copy the JCL to a member in the `SYS1.PROCLIB` concatenation.
2. Define the name of the started task to your security subsystem, and specify the userid that the started task runs with.

The parameters for the `RITSTART` program are:

- `QM(value)`

Where the value is a 1 to 4 character queue manager name. This parameter is required.

- `TRACE(n)`

Where `n` is between 0 and 9. The value of 0 means that no trace is displayed, for example `PARM='QM(&QM.) TRACE(0)'`.

Use this parameter under the direction of IBM Support. This parameter is optional.

2.1.6 Uninstalling the Rational Integration Tester MQF Exit Module

To uninstall `RITMQF`:

1. Stop and restart any queue managers that have used the `RITMQF` exit module. This is because the queue managers refer to the LPA module. The `RIT JOB` will stop.
2. Remove the APF authorization from the `PROG` member in the `SYS1.PARMLIB` concatenation.
3. Delete the `RIT` libraries

If you change the contents of `RITMQF` in the LPA, you must stop any queue managers that are using it, change the LPA, and restart the queue managers. Failure to do so can cause abends in the queue managers or the application.

2.1.7 Configuring IBM WebSphere MQ on z/OS

Set up a local queue `COM.GREENHAT.COMMAND.QUEUE`. This has low throughput, and can go into page set 1.

For information about setting up security, refer to [Setting up Security for the Rational Integration Tester MQ Exit](#). The Rational Integration Tester userid needs to get from this queue, so ensure that the userid has the appropriate permission.

The Rational Integration Tester tooling will internally create and use two name lists (the names are lower-case):

- `com.greenhat.intercept`
- `com.greenhat.intercept_lck`

2.1.7.1 Intercepting Messages

Every message that is intercepted has an `MQPUT1` to the `RIT` queue. This doubles the CPU cost of a `MQPUT` or `MQPUT1`. To reduce the overhead, select only the messages that you are interested in. Avoid intercepting all messages.

To avoid problems with the `RIT` queue filling up, you should have a low message rate of intercepted messages, for example 10's of messages per second.

Messages are not intercepted when:

- The queue name is `SYSTEM.*` and `AMQ.MQEXPLORER.*`.
- Messages put to a `QALIAS` of a topic
- Puts are the result of an integrated publish/subscribe action.

Messages put to a `QALIAS` queue are recorded against the base queue.

2.1.7.2 Running the Rational Integration Tester Job

To start the Rational Integration Tester job, you can either submit a job using the procedure customized above, or start the started task you defined above.

When the queue manager stops, the Rational Integration Tester job ends.

To stop the Rational Integration Tester job, use MVS operator command `P jobname` or `F jobname,stop`.

Other commands are as follows:

- `F jobname,DIS`

-
- Displays queues that have been used, or have errors.
 - F jobname,DIS ERRORS
Displays only those queues that had an error when putting to the destination queue.
 - F jobname,DIS ALL
Displays information about all queues being intercepted.
 - F jobname, refresh
Reads the definitions and updates the queues being intercepted. It also resets any statistics or errors and restarts interception.

2.1.7.3 Setting up an Application

When a message is intercepted, it is put to a RIT queue. The userid running the application must have permission to set the context of the RIT queue.

When there are two userids involved, for example, the CICS transaction userid, and the userid of the CICS region, both userids must have permission to set the context.

2.1.7.4 Setting up Security for the Rational Integration Tester MQ Exit

If you have security enabled for your queue manager, you must set up security for the Rational Integration Tester MQF exit resources, and give userids access to the RIT intercept queue.

When using mixed-case security, use the following definitions.

Class	Resource	RIT Job Userid	CHINIT	RIT User	Application Userid
MXNLIST	MQPG.com.greenhat.i ntercept	READ	ALTER		
MXADMIN	MQPG.CONTEXT.appli cation.queueName				CONTROL
	MQPG.CONTEXT.AMQ * where AMQ* is the intercept queue name		ALTER		ALTER
MQCMDS	MQPG.ALTER.NAMELI ST		ALTER	ALTER	
	MQPG.DEFINE.NAME LIST		ALTER	ALTER	
	MQPG.DELETE.NAME LIST		ALTER	ALTER	
MXQUEUE	MQPG.COM.GREENHA T.COMMAND.QUEUE	ALTER	ALTER	ALTER	

When using security with upper case classes, use the following definitions.

Class	Resource	RIT Job Userid	CHINIT	RIT User	Application Userid
MQNLIST	MQPG**	ALTER	ALTER		
MQADMIN	MQPG.CONTEXT.appli cation.queueName				CONTROL
	MQPG.CONTEXT.AMQ * where AMQ* is the intercept queue name				
MQCMDS	MQPG.ALTER.NAMELI ST		ALTER	ALTER	

Class	Resource	RIT Job Userid	CHINIT	RIT User	Application Userid
	MQPG.DEFINE.NAME LIST		ALTER	ALTER	
	MQPG.DELETE.NAME LIST		ALTER	ALTER	
MXQUEUE	MQPG.COM.GREENHA T.COMMAND.QUEUE	ALTER	ALTER	ALTER	

NOTE: If you are using RACF, you can use the RACF RLIST command to display which RACF profile will be used to protect a particular resource, for example:

```
rlist mxnlist MQ7A.com.greenhat.intercept.
```

Because Rational Integration Tester uses lower-case for resource names, you must use generic profiles. An upper-case profile name will not match a lower-case resource name.

For example, to set up the RACF(R) commands to give application users in group (SCENU) permission to put intercept messages into queue RITI, you could use the following commands:

```
RDEFINE MXADMIN MQPG.CONTEXT.RITI -
    OWNER(SCENU) UACC(NONE ) warning
PERMIT MQPG.CONTEXT.RITI -
    CLASS(MXADMIN) ACCESS(ALTER) ID(SCENU)
```

2.2 Configuring Web Services Support

You can use CICS web services to record, redirect, or create stubs for HTTP traffic. CICS programs can expose or consume HTTP-based web services.

Rational Integration Tester can be used to:

- Test a CICS program exposed as an HTTP web service.
- Virtualize an HTTP-based web service called from a CICS program.
- Use the HTTP client open exit XWBOPEN to specify an HTTP proxy for the CICS programs to use.

You can then use the HTTP/TCP proxy from the Rational Integration Tester Platform Pack to test HTTP traffic.

To record or virtualize a web service called from a CICS transaction, ensure that CICS uses the HTTP proxy included in the Rational Integration Tester Platform Pack. You can configure CICS to use an HTTP proxy by using the HTTP client open exit XWBOPEN.

For information about how to specify an HTTP proxy for CICS, go to http://publib.boulder.ibm.com/infocenter/cicsts/v3r1/topic/com.ibm.cics.ts31.doc/dfhtl/topics/dfhtl_xwbproxy.htm

2.3 Configuring Rational Integration Tester for IBM IMS Connect

To record IMS Connect traffic, you must set up port-forwarding in the HTTP/TCP proxy's `registration.xml` file to forward IMS Connect traffic into the proxy, and then to the actual IMS Connect server. This file is located in the `httptcp` folder in the Rational Integration Tester Platform Pack installation directory.

Edit the forward bind lines in the `registration.xml` file as shown:

```
<!-- The list of base port forwarding rules, the proxy will
listen on the address specified by the bind attribute and by
default will forward traffic on to thedestination address. The
type attribute indicates how the contents should be treated in
order to make recording and routing decisions.-->

<!--

<forward bind="localhost:2000" destination="localhost:3000" />

<forward bind="localhost:2001" destination="localhost:3001"
type="fix" />-->

<forward bind="<ProxyServer1:portNumber>"
destination="<ImsConnectServer1:portNumber>" type="imsconnect"
/>

<forward bind="<ProxyServer2:portNumber>"
destination="<ImsConnectServer2:portNumber>" type="imsconnect"
/>
```

This is an example of a `registration.xml` file that has been edited to redirect IMS traffic from proxies to IMS Connect servers:

```
<?xml version="1.0" encoding="UTF-8"?><registration>
<!-- Connection details to the Rational Test Control Panel
instance that <this proxy will register with -->
<server base-url="http://<ritserver08:1234/RTCP"/>
<logger level="info"/>
<!-- HTTP proxy settings where: -->
<!-- port Port number, client applications must be configured to
use this -->
<!-- bind-addressNIC to bind to, blank for all -->
<!-- log-level Default level of events to present to Rational Test
Control Panel instance -->
<!-- proxy/hostOnward proxy server host to chain requests through
-->
```

```

<!-- proxy/port Onward proxy server port to chain requests through
-->

<http-proxy port="3128" bind-address="">
<!-- proxy host="localhost" port="7128"/ -->
</http-proxy>

<!--HTTPS proxy settings where: -->

<!--    portPort number, client applications must be configured to
use this -->

<!--    bind-address NIC to bind to, blank for all -->

<!--    log-level    Default level of events to present to Rational
Test

Control Panel instance -->

<!--    keyStoreFileFile containing public and private keys used to
<!--accept client connections -->

<!--    keyStoreAliasAlias of KeyStore entry -->

<!--    keyStoreType    KeyStore type -->

<!--    keyStorePasswordKeyStore password -->

<!--    signingAlgorithmAlgorithm used to sign generated
certificates -->

<!--    serverProtocol The protocol that the secure server socket
will use -->

<!--    plainCommsPort  The port used to handle plain text to SSL
comms -->

<!--    outboundKeyStoreFile File containing public and private
keys used by the

<!--proxy to identify itself during mutual -->

<!--authentication -->

<!--    outboundKeyStorePasswordPassword to access the keystore
used during mutual

<!--authentication -->

<!--    proxy/host                Onward proxy server host to chain
requests through -->

<!--    proxy/port                Onward proxy server port to chain
requests through -->

<https-proxy port="3129" bind-address=""
keyStoreFile="greenhat.jks" keyStoreType="jks"
keyStoreAlias="mykey" keyStorePassword="passphrase"
signingAlgorithm="SHA1withRSA" serverProtocol="TLS"
plainCommsPort="3131" outboundKeyStoreFile="greenhat.jks"
outboundKeyStorePassword="passphrase">

```

```

<!-- proxy host="localhost" port="7128" / -->
</https-proxy>
<!--
The list of base port forwarding rules, the proxy will listen on
the address
specified by the bind attribute and by default will forward traffic
on to the
destination address. The type attribute indicates how the
contents should be
treated in order to make recording and routing decisions.
-->
<!--
<forward bind="localhost:2000" destination="localhost:3000" />
<forward bind="localhost:2001" destination="localhost:3001"
type="fix"/>
-->
<forward bind="ritserver01.com:1234"
destination="ImsConnectServer.com:5678" type="imsconnect"/>

<!-- A proxy does not need to register against a domain or
environment.
By default it will proxy for all domains/environments. If you wish
to restrict
its use then add <domain> entries based on the example below. Each
domain
may have 0 or more environments -->
    <!--
    <domains>
    <domain name="testDomain">
    <environment name="testEnvironment" />
    </domain>
    </domains>
    -->
</registration>

```

For more information about how to edit the registration.xml file, refer to *IBM Rational Integration Tester Platform Pack Installation Guide*.

NOTE: You cannot create stubs for IMS Connect services.

2.4 Configuring Recording for CICS Transaction Gateways

Before you can use CICS Transaction Gateway to record:

- IBM Rational Test Control Panel must be installed and configured. (For information about installing Rational Test Control Panel, refer to *IBM Rational Test Control Panel Installation Guide*.)
- The class path of the CICS Transaction Gateway daemon must be edited to include the Rational Integration Tester CTG Recording JAR file.

To configure recording for CICS Transaction Gateway:

1. Copy the Rational Integration Tester CTG Recording JAR file and supporting files to the system that is running the CICS Transaction Gateway daemon. The files are as follows:
 - `com.ibm.rational.rit.ctg.jar`
 - `registration.xml`
2. Edit the `registration.xml` file so that the server base-URL attribute targets your Rational Test Control Panel instance.
3. Configure the CTG Daemon classpath to include the Rational Integration Tester CTG Recording JAR file:

On Microsoft Windows:

- Use the `ctgservice` command to add the JAR file to the classpath.
- Use the `ctgservice -T` command to list the current configuration.
- If there is no existing classpath setting, use the following command to add one:

```
ctgservice -R -A-classpath=<gateway jar location>
```
- If there is already a classpath setting, add the gateway JAR location into the classpath value, using ‘;’ as the path separator, for example:

```
ctgservice -R -A-classpath=<existing value>;<gateway jar location>
```

On Linux/Unix:

- Add the Rational Integration Tester CTG Recording JAR file to the CTG daemon classpath.
- For more information, refer to “Configuring Gateway daemon settings” at <http://publib.boulder.ibm.com/infocenter/cicstgmp/v7r2/topic/com.ibm.cics.tg.doc/ctgunx/cclaha2.html>.

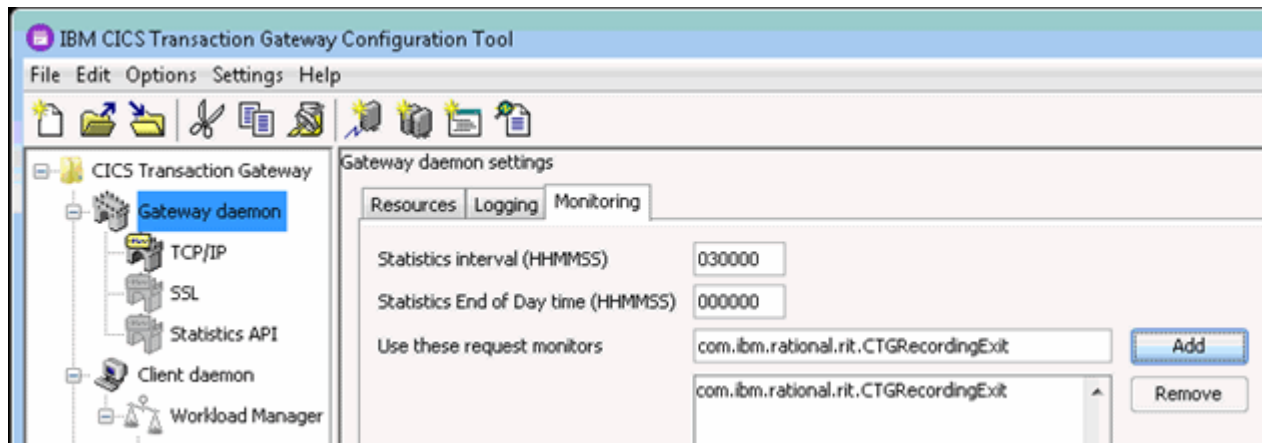
-
- Add the Rational Integration Tester recording exit into the CTG daemon configuration. You can use the configuration tool (ctgcfg) or edit the `cgt.ini` file to add `com.ibm.rational.rit.CTGRecordingExit` to the `requestexits` parameter for the gateway instance.

The configuration tool is part of the CICS Transaction Gateway.

To use the configuration tool:

1. Choose the gateway daemon.
2. Select the monitoring tab.
3. Add the exit.
4. Save and then exit the configuration tool.

The following graphic shows the CICS Transaction Gateway Configuration tool.



Setting Up Transports

Contents

Before Creating Transports

Creating Transports

This chapter describes how to create and configure transports for IBM Websphere MQ, CICS Transaction Gateway, and IMS Connect.

3.1 Before Creating Transports

On your Windows or LINUX system, you must configure JAR files to support transports for WebSphere MQ, CICS Transaction Gateway (CTG), and IMS Connect. The JAR files are located in the installation directories of the installed products.

Rational Integration Tester uses a decoupled, plug-in architecture to provide maximum flexibility with regard to the messaging transport software in use by a system under test. A messaging transport provides the information that Rational Integration Tester needs to communicate with applications on system Z.

The program code for each transport is contained in one or more JAR files (libraries), which are stored in Rational Integration Tester's `plugins` directory. Any libraries upon which these plug-ins depend must be accessible to Rational Integration Tester.

Library Manager is used to configure these third-party libraries. For information about using Library Manager, refer to *IBM Rational Integration Tester Installation Guide*.

3.1.1 IBM WebSphere MQ

System Z uses the same JAR files as IBM WebSphere MQ. If you have WebSphere MQ installed on your Windows or LINUX system, you do not need additional configuration. If you do not have MQ installed, refer to *IBM Rational Integration Tester Reference Guide* for IBM WebSphere MQ for configuration instructions.

3.1.2 CICS Transaction Gateway

The following JAR files must be set up in the Library Manager:

- `ctgclient.jar`
- `ctgserver.jar`

These files are located in the CTG installation folder, in the `classes` subfolder.

3.1.3 IMS Connect

You must configure the JAR file, `ImESConnectApiJavaV2R1.jar`, in Library Manager. This JAR file is located in the IMS installation folder, in the `IMS Connect for Java API` installation folder.

3.2 Creating Transports

To record interactions with WebSphere MQ, CICS, and IMS systems, set up transports to provide information about the infrastructure of these systems.

Before you can configure the transports in Rational Integration Tester, set up the following components in the Architecture School perspective's Logical view:

- Service component
- Operations for each service component

For information about setting up these components, refer to *IBM Rational Integration Tester Reference Guide*.

3.2.1 Creating an IBM WebSphere MQ Transport

For information about setting up WebSphere MQ transports in Rational Integration Tester, refer to *IBM Rational Integration Tester Reference Guide for IBM WebSphere MQ*.

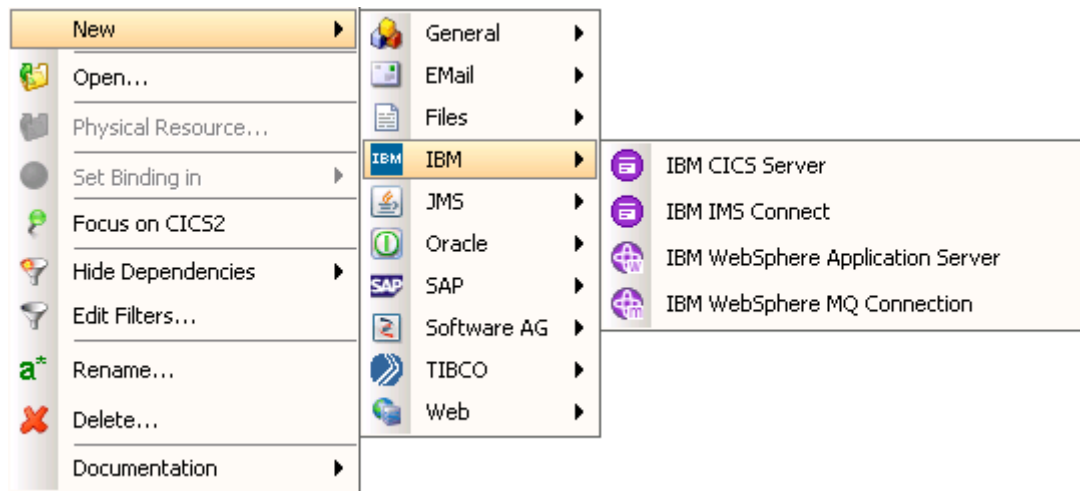
3.2.2 Creating a CICS Transport

You create a CICS transport when you create a physical CICS resource in Rational Integration Tester's Architecture School perspective.

Create a CICS physical resource, and then create a CICS Transaction Gateway for that resource.

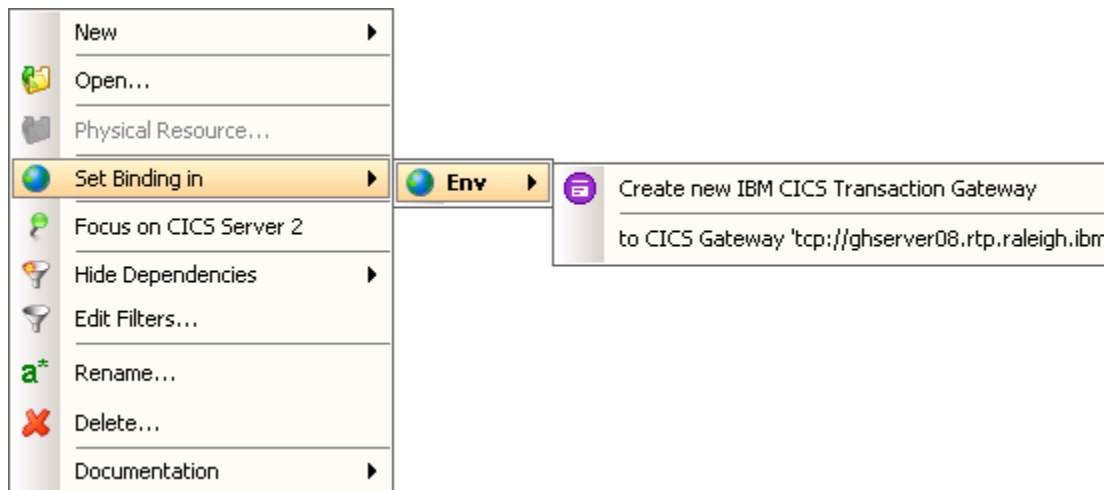
To create a logical CICS resource:

1. In Architecture School, in the Logical view, select a component in which to create the resource, or create it separately. Right-click and select **New > IBM > IBM CICS Server**.

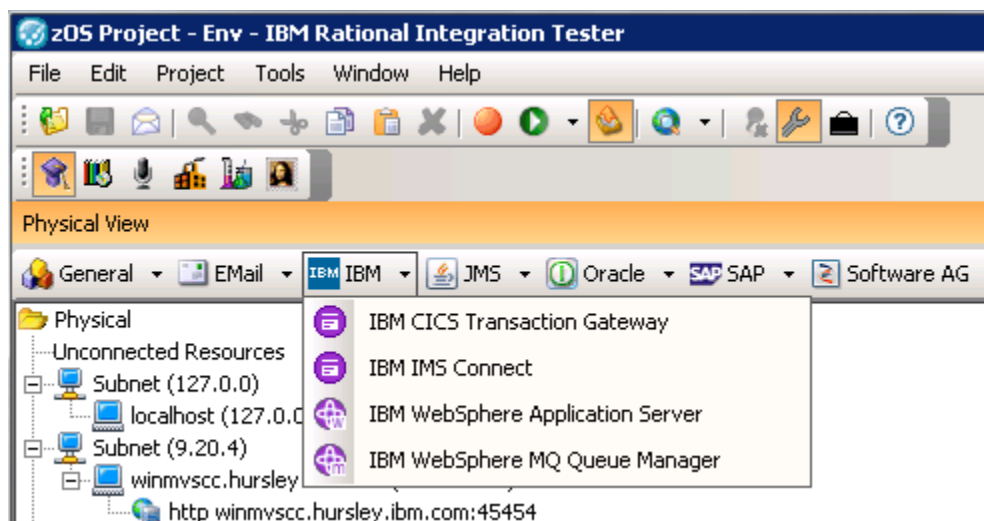


To create a CICS Transaction Gateway transport physical resource:

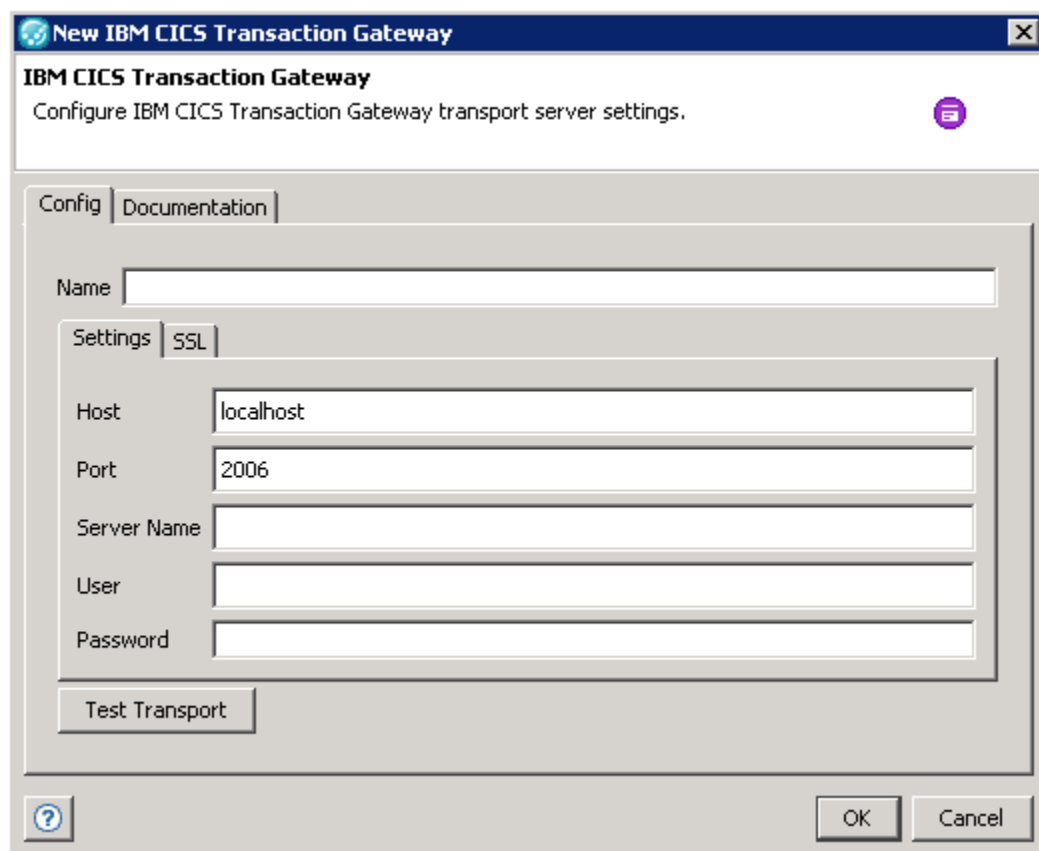
1. Set the binding for the CICS physical resource to the CICS Transaction Gateway using one of these steps:
 - In the Logical view, right click the CICS resource and select **Set Binding in > Env > Create new IBM CICS Transaction Gateway**.



- In the Physical view, click **IBM > IBM CICS Transaction Gateway**.



The New IBM CICS Transaction Gateway window is displayed.



2. On the **Config** tab, type a name for CICS Transaction Gateway.

-
3. On the **Settings** tab, type the following information for the CICS Transaction Gateway transport server:
 - Host name
 - Port
 - Server Name
 - User ID
 - Password
 4. To set up the transaction gateway to use SSL, click the **SSL** tab, select the **Use SSL** check box, and enter the key store.
 5. Click **Test Transport** to test the configuration.

3.2.3 Configuring CICS Transaction Gateway Messages

In CICS, you can group multiple program executions into a single Logical Unit of Work (LUW) that is committed as a transactional, or atomic, operation. All executions (and their side effects) succeed or fail. For information about how LUW works, refer to “Transactions in CICS” in the TXSeries information center: <http://publib.boulder.ibm.com/infocenter/txformp/v5r1/index.jsp?topic=%2Fcom.ibm.txseries510.doc%2Fatshak0029.htm>

Rational Integration Tester supports this functionality with fields in the Send Request message window for an LUW control flag and ID that can be passed as part of a Send Request messaging operation over a CICS Transaction Gateway Transport.

In the Send Request window, there are specific message fields for CICS Transaction Gateway, which are described here. For detailed instructions about defining messages, refer to *IBM Rational Integration Tester Reference Guide*.

To join two Request/Reply operations together as a single LUW, set the first interaction to use **LUW Control: Extended**, and store the resulting LUW token from the header in a tag. For the second Request/Reply operation, set the interaction to use LUW Control **Commit and populate the ID** with the tag from the first interaction. To have more than two operations in an LUW, use the same tagged value in the intermediate interactions, using LUW Control: **Extended**.

To roll back all the work done under that LUW, use the **Backout** option specifying an LUW ID.

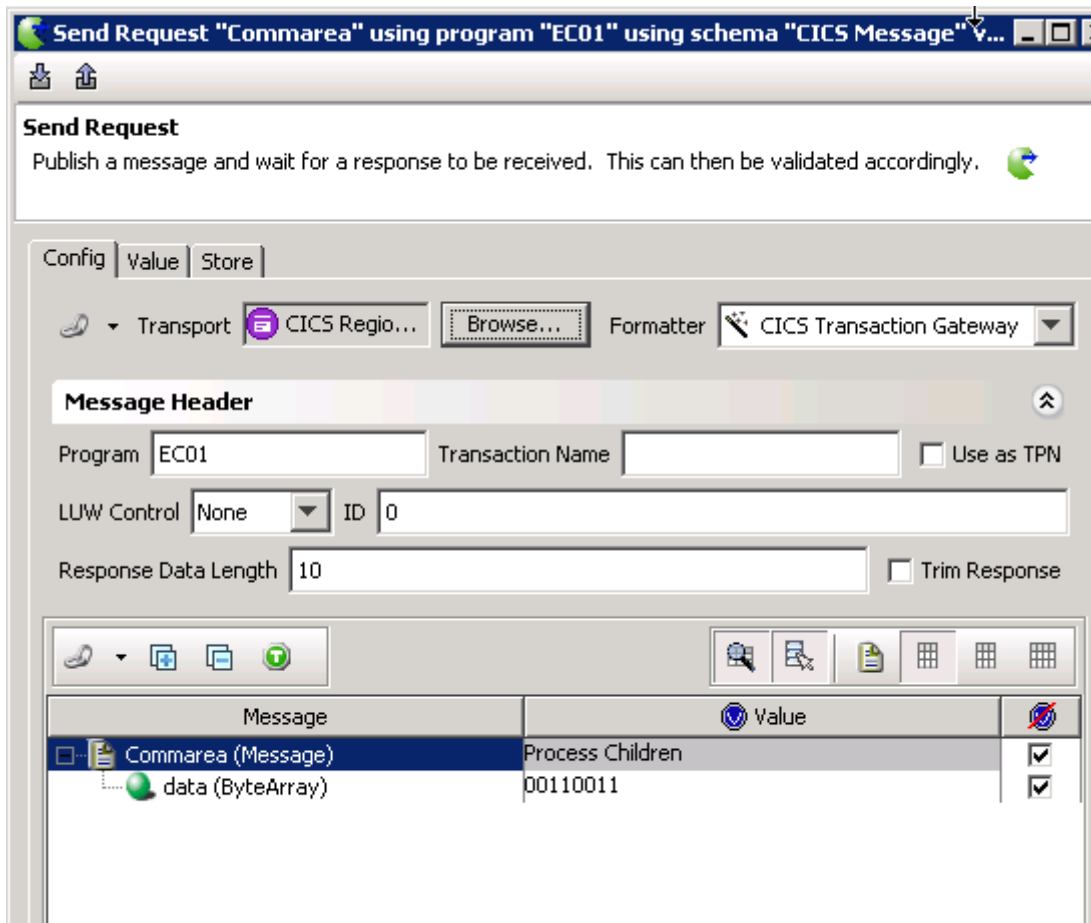
Prerequisites

You must create a test for an operation that is bound to a logical CICS server.

Procedure

To configure CICS Transaction Gateway Send messages:

1. In the Test Factory, open a CTG test and double-click the send request. The following graphic shows an example of a send request.



2. Click **Browse** to select the CICS transport to use for sending messages.
3. Using the **Formatter** list, ensure that CICS Transaction Gateway is selected.
4. In the **Program** field, type the name of the program to be executed on the CICS system.
5. In the **Transaction Name** field, type the name of the CICS Transaction Gateway transaction.

-
6. Select the **Use as TPN** check box to use the transaction name from previous step as the Transaction Program Name (TPN).
 7. Using the **LUW Control** list (Linux/Unix/Windows), select the LUW control to use. The options are **None**, **Extended**, **Commit**, or **Backout**.
 8. In **ID**, type the LUW ID for the message.
 9. In the **Response Data Length** field, type the number of characters allowed for response data.
 10. Select the **Trim Response** check box to limit the length of the response.
 11. Click **Ok** to save the Send message formatting.

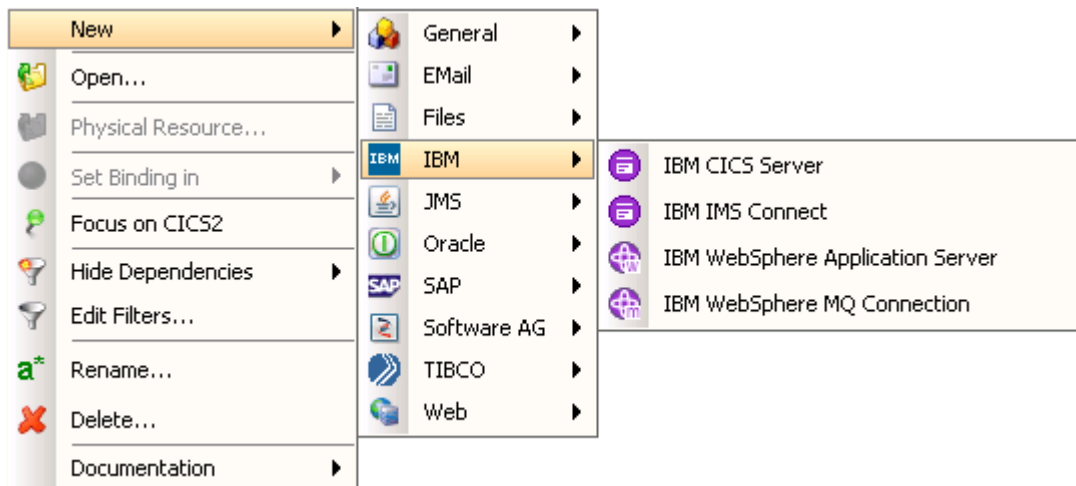
When you configure CICS Transaction Gateway to receive messages, you have the option of using the **CICS Transaction Gateway** formatter. This option is selected by default.

3.2.4 Creating the IMS Connect Transport

You create an IMS Connect transport when you create a logical IMS Connect resource in Rational Integration Tester's Architecture School.

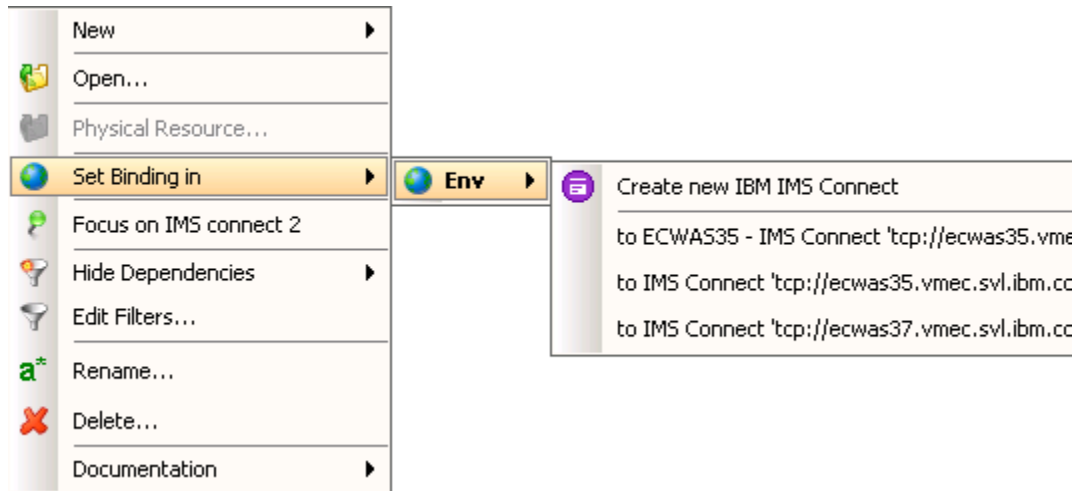
To create a logical IMS Connect resource:

In Architecture School, in the Logical view, you can select a component in which to create the resource, or you can create it separately. Right-click and select **New > IBM > IBM IMS Connect**.

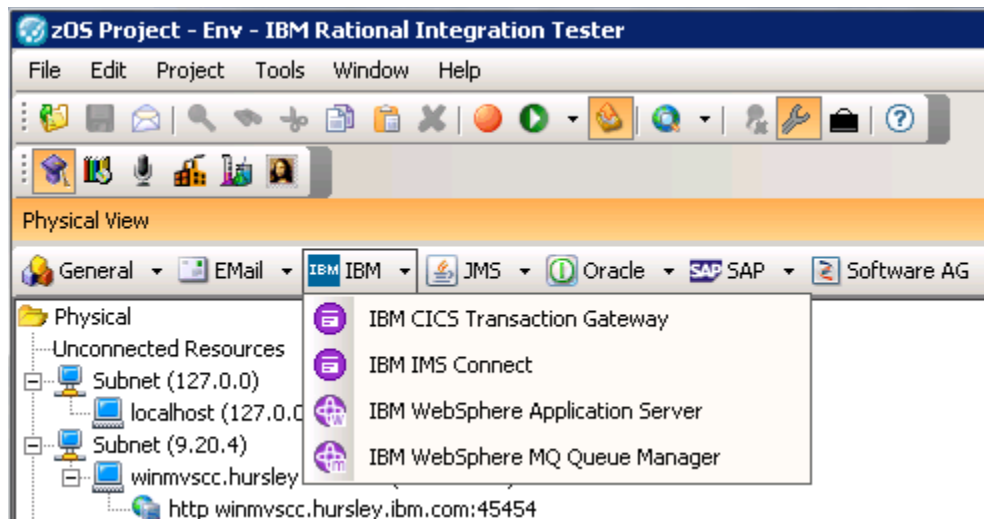


To create an IMS Connection transport:

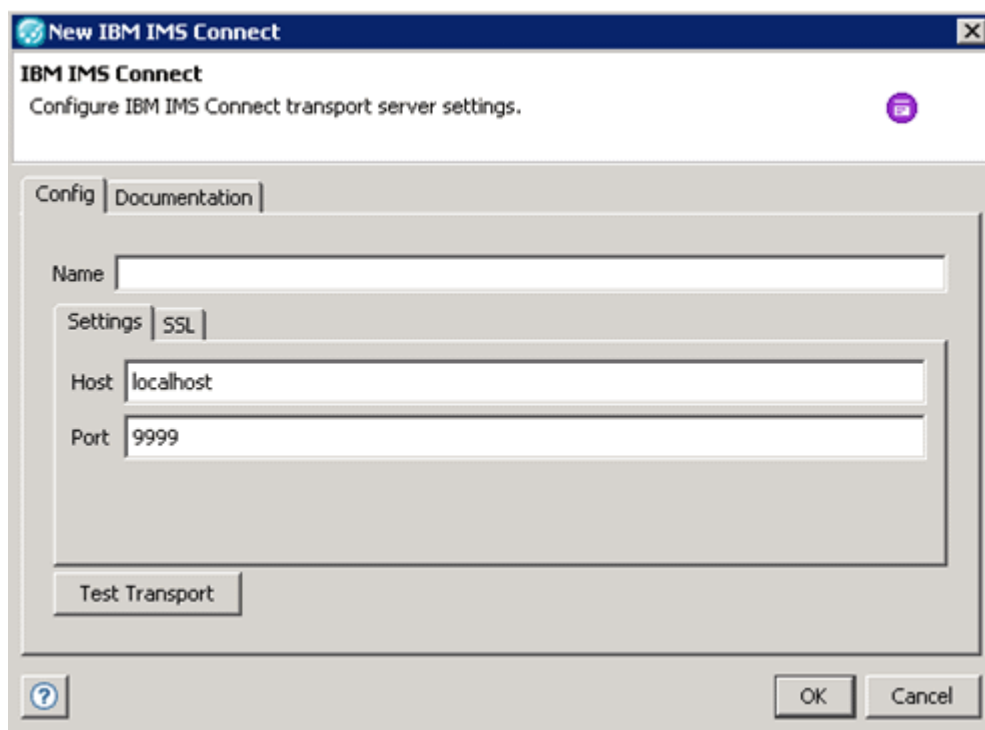
1. Set the binding for the IMS Connection physical resource using one of these steps:
 - In the Logical view, right click the IMS Connect resource and select **Set Binding in > Env > Create new IBM IMS Connect**.



- In the Physical view, click **IBM > IBM IMS Connect**.



The New IBM IMS Connect window is displayed.



The image shows a Windows-style dialog box titled "New IBM IMS Connect". The title bar is blue with a green icon on the left and a close button (X) on the right. Below the title bar, the text "IBM IMS Connect" is displayed in bold, followed by the instruction "Configure IBM IMS Connect transport server settings." in a smaller font. To the right of this text is a small purple circular icon with a white 'i' inside. Below the instruction, there are two tabs: "Config" (which is selected) and "Documentation". Under the "Config" tab, there is a "Name" label followed by a text input field. Below that, there are two sub-tabs: "Settings" (selected) and "SSL". Under the "Settings" sub-tab, there are two labels: "Host" and "Port". The "Host" label is followed by a text input field containing the text "localhost". The "Port" label is followed by a text input field containing the text "9999". Below these input fields is a button labeled "Test Transport". At the bottom left of the dialog box is a help icon (a question mark inside a circle). At the bottom right are two buttons: "OK" and "Cancel".

New IBM IMS Connect

IBM IMS Connect
Configure IBM IMS Connect transport server settings.

Config | Documentation

Name

Settings | SSL

Host

Port

Test Transport

?

OK Cancel

3.2.5 Configuring IMS Connect Messages

There are specific message fields for IMS Connect, which are described here. For detailed instructions about defining messages, refer to *IBM Rational Integration Tester Reference Guide*.

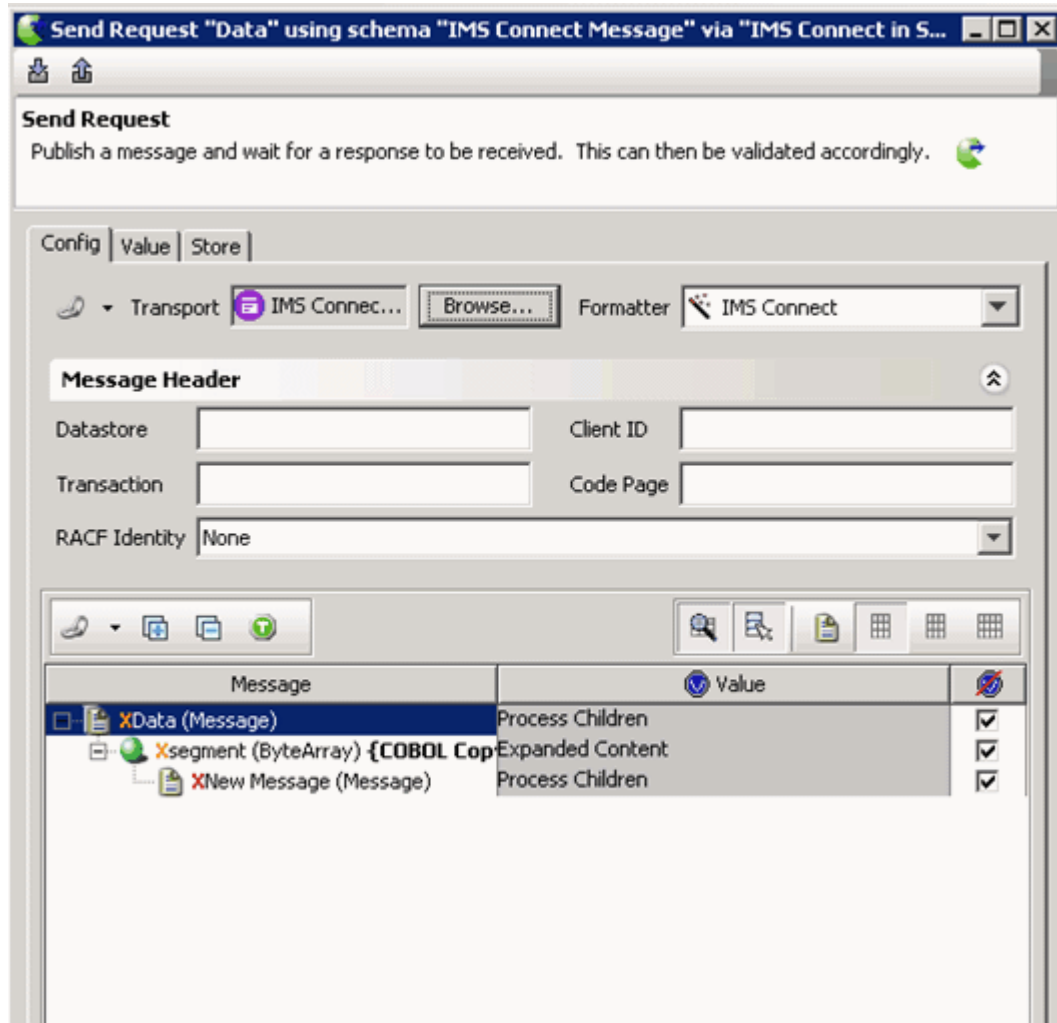
Prerequisites

You must create a test for an operation that is bound to a logical IMS Connect server.

Procedure

To configure IMS Connect Send messages:

1. In the Test Factory, open an IMS Connect test and double-click the send request. The following graphic shows an example of a send request.



-
2. Click **Browse** to select the IMS Connect transport to use for sending messages.
 3. Using the **Formatter** list, ensure that IMS Connect is selected.
 4. In the **Datastore** field, type the logical name that IMS Connect has for an IMS instance. The datastore is used to route the transaction execution to an IMS instance known to IMS Connect. There is a many-to-many relationship between IMS Connect and IMS instances. For example, in a simple configuration such as 1 IMS, 1 IMS Connect, the datastore name will be IMS1 by default.
 5. In the **Client ID** field, type the ID for IMS Connect and IMS to identify IMS Connect clients. The clientId is also used for the names of asynchronous message queues (OTMA Tpipes) and other internal data structures in IMS Connect and OTMA.

The clientId value is a string of 1 to 8 uppercase alphanumeric (A through Z, 0 to 9) or special (@, #, \$) characters.

If you do not enter a value for Client ID, a default value is assigned.

If a user is interacting asynchronously (for example, a Publish message action, followed by a Subscribe later in the test to pick up the result), use the same client ID for both steps.

6. In the **Transaction** field, type the IMS transaction code, which is a string of uppercase alphanumeric (A through Z, 0 to 9) or special (@, #, \$) characters. The trancode property identifies the IMS application (transaction) to be executed.
7. In the **Code Page** field, type the code page number.
8. In the **RACF Identity** field, type the RACF ID. RACF can be enabled on a per-transaction basis on the server. To support RACF credentials, a RACF identity can be created (in the Physical View of Architecture School) and referenced here. The default is **None**.
9. Click **Ok** to save the send message formatting.

When you configure IMS Connect receive messages, you have the option of using the **IMS Connect** formatter. This option is selected by default.

Troubleshooting

Contents

Error Messages

Common Errors

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

4.1 Error Messages

This section describes error messages that might be displayed while using IBM WebSphere MQ on z/OS systems.

Message title

RIT100I Rational Integration Tester for z/OS

Explanation

This message is issued when the Rational Integration Tester job starts.

Message title

RIT001S load of message module RITMSG failed

Explanation

The Rational Integration Tester job was unable to load the module RITMSG using the C runtime FETCH function. The SYSOUT dataset will contain a message giving the reason.

For example: Load of RITMSG: EDC5239S Fetched module not found.
(errno2=0xC4070044)

Severity

12

System action

The Rational Integration Tester job ends.

Response

Determine why the module failed to load. Resolve the problem and restart the Rational Integration Tester job.

Message title

RIT002I Abnormal end detected

Explanation

The Rational Integration Tester job detected that it was cancelled, or an ABEND was detected.

Severity

0

System action

The Rational Integration Tester job ends.

Response

Resolve the problem and restart the Rational Integration Tester job.

Message title

RIT003I Clean up successful

Explanation

After message RIT002I, the Rational Integration Tester job was able to recover and shut down successfully.

Severity

0

System action

The Rational Integration Tester job ends.

Response

None

Message title

RIT004I Recovery routine entered

Explanation

The Rational Integration Tester job detected an error had occurred and entered a recovery routine.

Severity

0

System action

The Rational Integration Tester job retries the operation. If it is unable to recover, the Rational Integration Tester job ends.

Response

Restart the Rational Integration Tester job if required.

Message title

RIT005S RIT Error recovery routine: RIT now inactive

Explanation

The Rational Integration Tester intercept exit detected that an error had occurred and entered a recovery routine.

Severity

0

System action

The Rational Integration Tester exit is set to inactive and stops intercepting messages.

Response

A dump is produced. Contact your IBM Service representative if you need assistance with this.

Message title

RIT010I LPA module information (1)

Explanation

The Rational Integration Tester job reports information about the RITMQF module in the LPA.

Severity

0

Response

You can use this to verify that you are using the correct version.

Message title

RIT100E Required parameter is missing QM(aaaa)

Explanation

The required queue manager name was not specified at the Rational Integration Tester job start.

Severity

8

System action

The Rational Integration Tester job ends.

Response

Specify the queue manager name using the format QM(name) and restart the Rational Integration Tester job.

Message title

RIT101E Invalid QM(value) specified. Where value should be a string of length 1-4 characters.

Explanation

The QM parameter was specified, but had an invalid value. The value is a string of characters between 1 and 4 characters in length.

Severity

8

System action

The Rational Integration Tester job ends.

Response

Specify the queue manager name correctly and restart the Rational Integration Tester job.

Message title

RIT102E Invalid TRACE(value) specified. Value should be a number in range 0-9.

Explanation

The optional TRACE parameter was specified, but had an invalid value. The value should be a number in the range 0-9.

Severity

8

System action

The Rational Integration Tester job ends.

Response

Specify the correct value and restart the Rational Integration Tester job.

Message title

RIT103E Unknown parameter (1) ignored

Explanation

An unknown parameter (1) was specified at startup.

Severity

8

System action

The Rational Integration Tester job ends.

Response

Remove the invalid parameter and restart the Rational Integration Tester job.

Message title

RIT104I Queue Manager: (1)

Explanation

The Rational Integration Tester job reports the value (1) of the specified queue manager to be used.

Severity

0

System action

None

Message title

RIT105I Trace: (1)

Explanation

The Rational Integration Tester job reports the value (1) of the specified TRACE parameter to be used.

Severity

0

System action

None

Message title

RIT110E MQCONN to (1) cc (2) rc (3) (4)

Explanation

The Rational Integration Tester job was unable to connect to the specified queue manager.

- Queue manager name (1)
- MQ Condition code (2)
- Reason code value (3)
- Reason code description(4)

For example

RIT110E MQCONN to MQPA cc 2 rc 2059 MQRC_Q_MGR_NOT_AVAILABLE

Severity

8

System action

The Rational Integration Tester job ends.

Response

Resolve the problem and restart the Rational Integration Tester job.

Message title

RIT111E MQOPEN for (1) cc (2) rc (3) (4)

Explanation

The Rational Integration Tester job was unable to open the queue.

-
- Queue name (1)
 - MQ Condition code (2)
 - Reason code value (3)
 - Reason code description(4)

For example

```
RIT111E MQOPEN for COM.GREENHAT.COMMAND.QUEUE cc 2 rc 2085  
MQRC_UNKNOWN_ OBJECT_NAME
```

Severity

8

System action

The Rational Integration Tester job ends.

Response

Resolve the problem and restart the Rational Integration Tester job.

Message title

RIT112E MQGET from (1) cc (2) rc (3) (4)

Explanation

RIT was unable to get from the queue,

- Queue name (1)
- MQ Condition code (2)
- Reason code value (3)
- Reason code description (4)

Severity

8

System action

The Rational Integration Tester job ends.

Response

Resolve the problem and restart the Rational Integration Tester job.

Message title

RIT201E (1) is not APF authorized

Explanation

The RIT STEPLIB libraries need to be APF authorized. Check that all datasets in STEPLIB are APF authorized. (1) is the name of the program that needs to be APF authorized

Severity

8

System action

The Rational Integration Tester job ends.

Response

An authorized person can use the MVS operator command.

D PROG APF,DSN=

to display the APF status of the specified dataset.

Resolve the problem and restart the Rational Integration Tester job.

Message title

RIT202E Unable to get lock - is there an instance running?

Explanation

The Rational Integration Tester job obtains a lock based on the queue manager name to ensure there is only one instance of the Rational Integration Tester job running. It was unable to get the lock.

Severity

8

System action

The Rational Integration Tester job ends.

Response

Check to see if another instance of the Rational Integration Tester job is running with same queue manager specified.

Message title

RIT203E Unable to locate RITMQF in LPA

Explanation

The Rational Integration Tester job requires module RITMQF to be located in the LPA. The Rational Integration Tester job was not able to locate it.

Severity

8

System action

The Rational Integration Tester job ends.

Response

Check to see if RITLINK library has been added to the LPA. An authorized person can issue the MVS operator command `D PROG,LPA,MODNAME=RITMQF` to display information about the module in the LPA.

Message title

RIT204E CSVQUERY return code (1) locating RITMQF in LPA.

Explanation

The MVS CSVQUERY macro was used to locate the module in the LPA. The request was unsuccessful, and the return code was (1).

Severity

8

System action

The Rational Integration Tester job ends.

Response

Resolve the problem and restart the Rational Integration Tester job.

Message title

RIT205S Internal error function (1) (2) code (3) trace (4)

Explanation

The Rational Integration Tester job was unable to start successfully due to an internal error.

Severity

8

System action

The Rational Integration Tester job ends.

Response

Contact IBM Support with the job log.

Message title

RIT205E Unable to start RIT job. Return code (1) trace (2)

Explanation

The Rational Integration Tester job was unable to start successfully due to an internal error.

Severity

8

System action

The Rational Integration Tester job ends.

Response

Contact IBM Support with the job log.

Message title

RIT300I No queues being monitored

Explanation

A display command was issued, but there are no queues being monitored.

Message title

RIT301I Output from Display command

Explanation

The display command produces several lines of output for each queue being monitored. There are two or three header lines and multiple sections of two or three lines each. With each intercepted queue, the output includes the queue name itself, the destination queue name, and the queue manager name.

If a destination queue is on the local queue manager, the queue manager name is not displayed.

RIT301I Output from Display command

Queue Name	successful	errors
Target Queue Name	successful	errors
CP0000	2	0
XCP0000	2	0
Last successful put at 2012/10/17,14:50:23		
CP0001	2	0
XCP0001	0	2
MQ Reason code 2085, MQRC_UNKNOWN_OBJECT_NAME at 2012/09/06,16:47:45		

This shows that two queues were intercepted, CP0000 and CP0001. One message was successfully put to queue CP0000. This was successfully put to the destination queue XCP0000.

One message was successfully put to queue CP0001, but there was an error putting this to destination queue XCP0001.

The last error detected is MQ reason code 2085 which is MQRC_UNKNOWN_OBJECT_NAME. This occurred on 2012/09/06 at 16:47:45.

If there is at least one instance where the queue manager for the destination queue is not the name of the local queue manager, the output is as follows:

RIT301I Output from Display command

Queue Name	successful	errors
Target Queue Name	successful	errors
Target Queue Manager Name		
CP0000	1	0
XCP0000	1	0
Last successful put at 2012/11/05, 09:59:22		
INPUTQ*	0	0
XCP00002	0	0
MQ004		

MQ004 Where MQ004 is the remote queue manager.

If there has been a problem putting the message to the destination queue, information about the error is displayed, for example:

RIT301I Output from Display command

Queue Name	successful	errors
Target Queue Name	successful	errors
CP0000	2	0
XCP0000	0	2

MQ Reason code 2051, MQRC_PUT_INHIBITED at 2012/09/12,08:31:25

Where the time (08:31:25) is local time. If the problem is resolved and puts to this queue are successful, this message will continue to be displayed.

Check the last successful put at... to see if the problem has been resolved.

If the refresh command is issued these errors are reset.

Severity

0

Message title

RIT307E Unable to allocate storage - display command failed

Explanation

The Rational Integration Tester job was unable to allocate storage to process the display command.

Severity

8

System action

No data is displayed. The Rational Integration Tester job continues to run.

Response

Increase the region size on the EXEC PGM statement in the Rational Integration Tester JCL and restart the Rational Integration Tester job.

Message title

RIT320I Invalid DIS command, syntax is DIS, DIS ALL, or DIS ERROR.

Explanation

An invalid DIS command was issued. The following commands are valid:

DIS ALL	Displays information about all queues being monitored.
DIS ERROR	Displays information about those queues which have had an error.
DIS	Displays queues which have processed messages.

Severity

0

Response

Enter the correct syntax.

Message title

RIT321E Problem allocating storage during intercept request.

Explanation

When the RIT exit was intercepting a message it was unable to allocate storage for a copy of the message.

Severity

8

System action

No data is displayed. The Rational Integration Tester job continues to run.

Response

Increase the region size of the job putting the message and restart this job.

Message title

RIT322I No records found matching request

Explanation

A display DIS or DIS ERRORs command was issued, but no queues had been used, so there were none to display.

Severity

0

Response

To display information about all queues, use the DIS ALL command.

Message title

RIT331I RIT interception is active.

Explanation

The Rational Integration Tester job is active and able to intercept messages.

Severity

0

Response

None

Message title

RIT332W RIT interception is inactive

Explanation

The Rational Integration Tester job is active, but the exit is not intercepting messages. The exit detected a problem and deactivated further interceptions.

Severity

4

Response

Look for RIT005S messages and resolve any problems.

Issue the F ritjob,REFRESH command to reload the current list of queues and reactivate the interception.

Message title

RIT400S MQOPEN for Namelist (1) cc (2) rc (3) (4)

Explanation

The Rational Integration Tester job was unable to open the following:

- Name list (1)
- MQ Condition code (2)
- Reason code value (3)
- Reason code description(4)

Severity

8

System action

The Rational Integration Tester job shuts down.

Response

Resolve the problem and restart the Rational Integration Tester job.

Message title

RIT401S MQINQ for Namelist (1) cc (2) rc(3) (4)

Explanation

The Rational Integration Tester job was unable to INQ on the following:

- Name list (1)
- MQ Condition code (2)
- Reason code value (3)
- Reason code description (4)

Severity

8

System action

The Rational Integration Tester job shuts down.

Response

Resolve the problem and restart Rational Integration Tester.

Message title

RIT402W No entries in name list (1)

Explanation

The Rational Integration Tester job name list existed, but had no entries.

Severity

4

Response

Configure queues to be monitored.

Message title

RIT403E Namelist (1) has invalid number of entries(2)

Explanation

The Namelist (1) has entries for each queue monitoring. The number of entries in the name list is not a multiple of 3.

Severity

8

System action

The name list is ignored.

Response

Resolve the problem.

Message title

RIT404I Refresh queue list successful using list from (1) at (2)

Explanation

The monitored queue list has been refreshed, using the name list which was last altered at time (2) on date (1).

For example: RIT404I Refresh queue list successful using list from 2012-09-01 at 13.52.34

Severity

0

Message title

RIT405E Refresh queue list unsuccessful rc (1)

Explanation

The Rational Integration Tester job was unable to replace the list of queues to be monitored. This is an internal error.

Severity

8

Response

Contact IBM Support with the Rational Integration Tester job log.

Message title

RIT410E Unable to decode character (1) in string (2) in Namelist

Explanation

There is an invalid character (1) in the entry (2) in the name list.

Severity

8

System action

The name list is ignored.

Response

Recreate the name list.

Message title

RIT450E Internal error processing command (1) (2)

Explanation

An internal error occurred when processing a command

Severity

8

System action

The Rational Integration Tester job ends.

Response

Report this problem to IBM Support and restart the Rational Integration Tester job.

4.2 Common Errors

Problem

```
ICH408I  USER(IBMUSER  ) GROUP(TSOUSER  ) NAME(IBM user      )  
        MQPG.CONTEXT.AMQ.CA40EF5E3FD3A36B CL(MQADMIN  )  
        INSUFFICIENT ACCESS AUTHORITY  
        FROM MQPG.CONTEXT.* (G)  
        ACCESS INTENT(CONTROL)  ACCESS ALLOWED(NONE    )
```

Solution

Give IBMUSER permission to access the context for the queue.

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.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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.

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.

