Rational Integration Tester

Reference Guide for JMS Messaging

Version 8.0.0

Note

Before using this information and the product it supports, read the information in “Notices” on page 22.

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.

Contents

About this Publication

Contents

This guide describes how to configure and run IBM® Rational® Integration Tester with the generic JMS plugin, which provides support for connectivity to a wide range of EAI platforms (any vendor providing an implementation of this Java standard).

Intended Audience

This document intended to be read by those with a fair understanding and exposure to the concepts involved in both testing and development and in enterprise integration.

Scope

This document is concerned only with the configuration and use of IBM Rational Integration Tester alongside JMS messaging technologies.

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.
Italic	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/

JMS Overview

Contents

This chapter provides a brief overview of the JMS, including descriptions of its elements and messaging models.

1.1 JMS Providers

In addition to native JMS support provided in Rational Integration Tester, the following JMS providers are supported:

- Fiorano MQ
- IBM WebSphere® MQ (refer to *IBM Rational Integration Tester Reference Guide for IBM WebSphere MQ*)
- Oracle OC4J (Application Server)
- Oracle OC4J (BPEL)
- Solace
- SonicMQ (refer to *IBM Rational Integration Tester Reference Guide* for *SonicMQ*)
- TIBCO EMS (refer to *IBM Rational Integration Tester Reference Guide for TIBCO*)
- WebLogic JMS Thick
- WebLogic JMS Thin
- webMethods 6.5 JMS
- webMethods 7.1 JMS
- webMethods 8.0 JMS
- webMethods 8.2 JMS

NOTE: The need for the WebLogic JMS thin or thick libraries is determined by your specific environment.

For more information about when to use which provider, see http://download.oracle.com/docs/cd/E12840_01/wls/docs103/client/basics.html#wp1069994.

For more information about generating `wlfullclient.jar` (used by the WebLogic JMS Thick provider), see http://download.oracle.com/docs/cd/E12840_01/wls/docs103/client/jarbuilder.html.

1.2 JMS Elements

The following are the basic elements of JMS, and some or all may be discussed later in this document.

JMS Provider

An implementation of the JMS interface for a Message Oriented Middleware (MOM). Providers are implemented as either a Java JMS implementation or an adapter to a non-Java MOM.

JMS Client

An application or process that produces and/or receives messages.

JMS Producer

A JMS client that creates and sends messages.

JMS Consumer

A JMS client that receives messages.

JMS Message

An object that contains the data being transferred between JMS clients.

JMS Queue

A staging area that contains messages that have been sent and are waiting to be read. As the name queue suggests, the messages are delivered in the order sent. A message is removed from the queue once it has been read.

JMS Topic

A distribution mechanism for publishing messages that are delivered to multiple subscribers.

1.3 JMS Models

The JMS API supports two models:

- point-to-point (queuing) model
- publish and subscribe model

Point-to-point Model

In the point-to-point or queuing model, a sender posts messages to a particular queue and a receiver reads messages from the queue. Here, the sender knows the destination of the message and posts the message directly to the receiver's queue. It is characterized by the following:

- Only one consumer gets the message.
- The producer does not have to be running at the time the consumer consumes the message, nor does the consumer need to be running at the time the message is sent.
- Every message successfully processed is acknowledged by the consumer

Publish and Subscribe Model.

The publish/subscribe model supports publishing messages to a particular message topic. Subscribers may register interest in receiving messages on a particular message topic. In this model, neither the publisher nor the subscriber know about each other. A good metaphor for it is anonymous bulletin board. The following are characteristics of this model:

- Multiple consumers can get the message.
- There is a timing dependency between publishers and subscribers. The publisher has to create a subscription in order for clients to be able to subscribe. The subscriber has to remain continuously active to receive messages, unless it has established a durable subscription. In that case, messages published while the subscriber is not connected will be redistributed whenever it reconnects.

Using Java, JMS provides a way of separating the application from the transport layer of providing data. The same Java classes can be used to communicate with different JMS providers by using the JNDI information for the desired provider. The classes first use a connection factory to connect to the queue or topic, and then use populate and send or publish the messages. On the receiving side, the clients then receive or subscribe to the messages.

JMS Transport

Contents

This chapter provides an overview of how to create and configure the JMS transport.

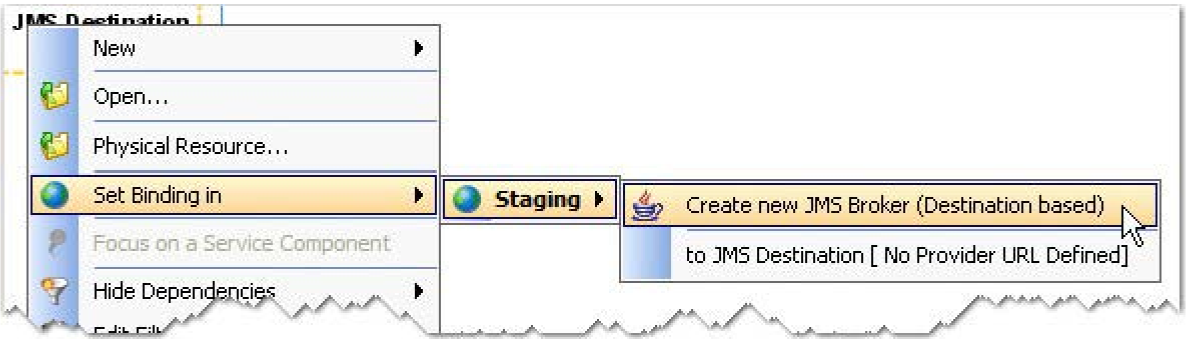
2.1 Creating the JMS Transport

JMS messaging in Rational Integration Tester supports publishing to destinations, topics, and queues, and each is managed by a separate transport. The configuration of all three, however, is identical. Only when selecting the different transports in a message editor will you see the option to publish/subscribe to a destinations, topics, or queues.

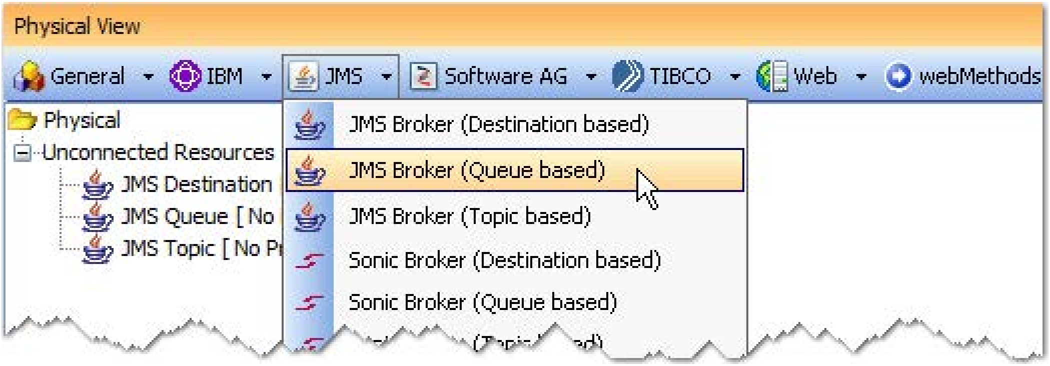
The JMS transport is created when you create a physical JMS Broker resource in Rational Integration Tester's Architecture School.

In Architecture School, you can create a new resource in two ways:

- In the Logical View, right-click on a JMS Topic, Queue, or Destination and select the **Set Binding in > [environment] > Create new JMS Broker** option.


JMS Destination
New
Open...
Physical Resource...
Set Binding in
Focus on a Service Component
Hide Dependencies
Staging
Create new JMS Broker (Destination based)
to JMS Destination [No Provider URL Defined]

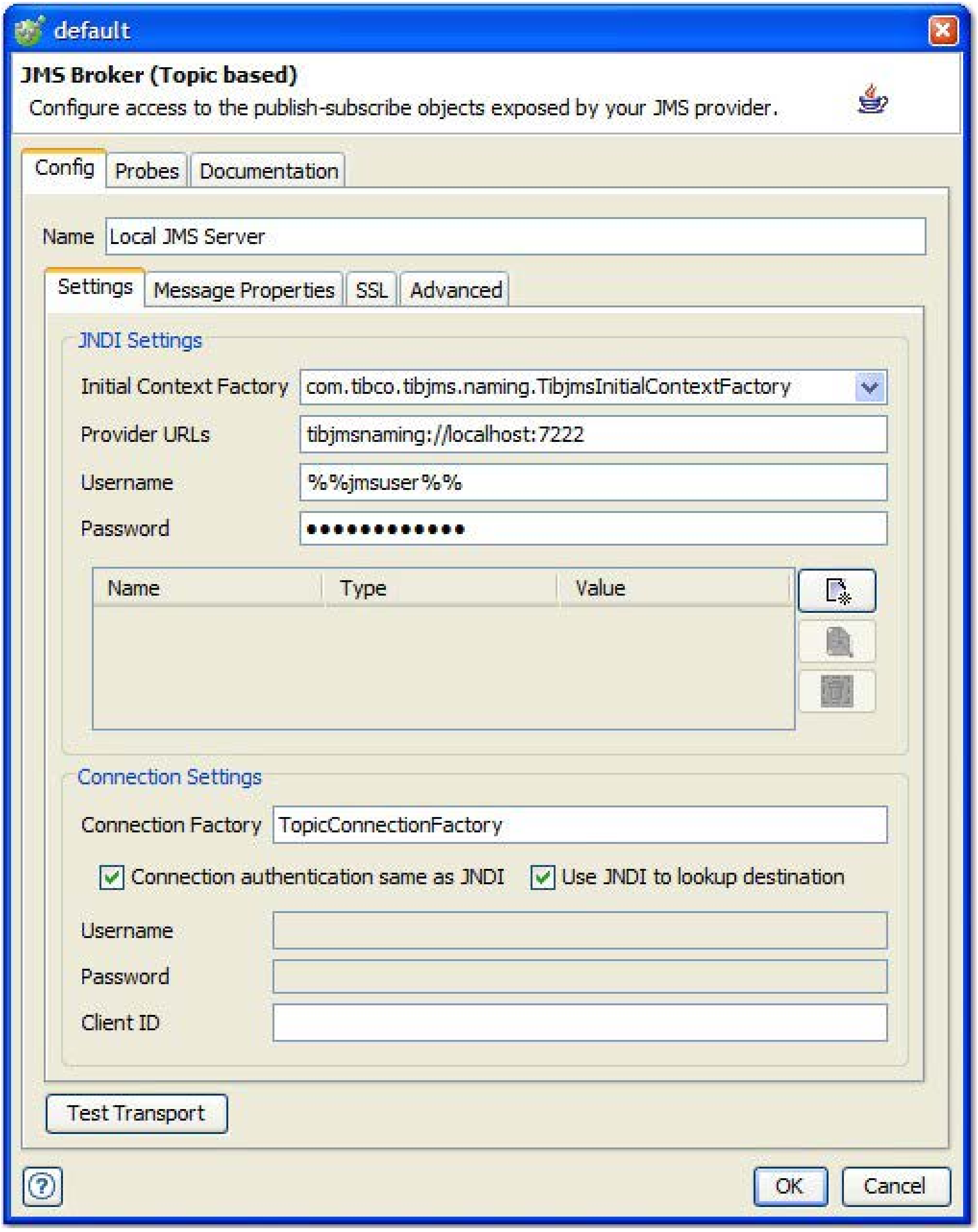
- In the Physical View, select the **JMS > JMS Broker** option.



Each physical JMS Broker resource will represent a JMS transport that can be selected and configured later on.

2.2 Configuring the JMS Transport

To configure a JMS transport, double-click the appropriate JMS Broker resource in Architecture School's Physical View. If desired, enter a name for the transport in the **Name** field (for example, to help identify it when more than one JMS transports are available).



default

JMS Broker (Topic based)

Configure access to the publish-subscribe objects exposed by your JMS provider.

Name: Local JMS Server

JNDI Settings

Initial Context Factory: com.tibco.tibjms.naming.TibjmsInitialContextFactory

Provider URLs: tibjmsnaming://localhost:7222

Username: %%jmsuser%%

Password: ••••••••••••

Name	Type	Value

Connection Settings

Connection Factory: TopicConnectionFactory

☑ Connection authentication same as JNDI ☑ Use JNDI to lookup destination

Username:

Password:

Client ID:

The transport settings are broken into the JNDI and Connection Settings (**Settings** tab), Message Properties, SSL Settings, and Advanced Settings.

2.2.1 JNDI and Connection Settings

JNDI and connection settings are configured under the **Settings** tab (shown in the previous section). The available settings are described in the following sections.

NOTE: All of the configuration fields allow the use of tags, from the context menu or entered manually. The **Password** field does not support the context menu, so you will need to enter tags manually (for example, %%JMS_password%%). Since this field is encrypted, characters will be masked when entered.

JNDI Settings

JNDI settings are described in the following table:

Initial Context Factory	The Java class used to obtain context information to perform naming and directory service functions through JNDI. Default values are provided for the JMS implementations supported by Rational Integration Tester.
Provider URLs	URL of the JMS server's JNDI tree, specific to the selected JMS implementation. The required format of the URL is provided when one of the supplied context factory entries is selected.
Username/Password	The default user name and password to send when connecting to JNDI.
Context Properties	In the table at the bottom of the JNDI Settings you can enter additional JNDI properties (name-value pairs) that should be set. These properties are specific to the server you are connecting to, and you should refer to the server's documentation if you are unsure about any of these settings. • Click 📄 to create a new property, then enter the property name, type, and value in the New Message Property dialog. • Select an existing property and click ✎ to edit it. • Select an existing property and click 🗑 to delete it.

Connection Settings

Connection settings are described in the following table:

Connection Factory	The default JNDI lookup name of a JMS connection factory that exists on the broker. If specified, indicates the location in a JNDI tree to find a Connection Factory object.
Connection authentication same as JNDI	Enable this option to send the user name/password specified under JNDI settings when obtaining a connection from the connection factory.
Use JNDI to lookup destination	Enable this option to use the specified JNDI settings to look up destinations.
Username/Password	The user name and password to use when obtaining a connection from the connection factory.
Client ID	The JMS client identifier needed for durable topic subscriptions on all connections created using this connection factory.

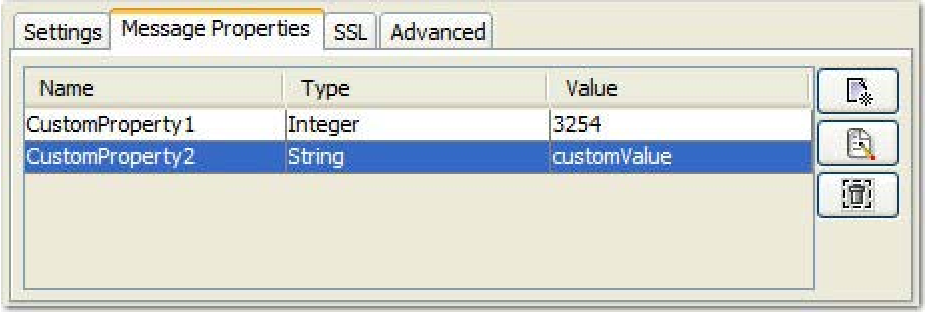
Testing JNDI and Connection Settings

After configuring the JNDI and connection settings for the transport, click **Test Transport** to verify them. If the connection is successful, you may proceed. Otherwise, verify the settings and try again.

2.2.2 Message Properties

It is possible to send additional properties in the header of JMS messages. These properties can be used, for example, to let recipients make decisions about which messages should be presented to the receiving application.

The **Message Properties** tab allows you to specify additional properties that can be set on each message sent using the specified transport.



Name	Type	Value
CustomProperty1	Integer	3254
CustomProperty2	String	customValue

- Click to create a new property, then enter the property name, type, and value in the **New Message Property** dialog.
- Select an existing property and click to edit it.
- Select an existing property and click to delete it.

NOTE: For Solace JMS, it may be necessary to add a message property to specify a VPN, as follows:

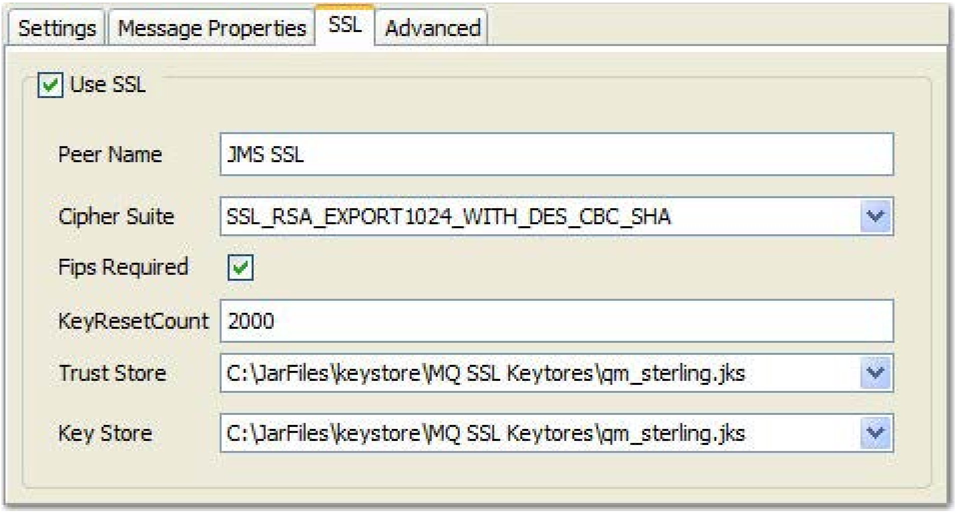
Name: Solace_JMS_VPN

Type: String

Value: Name of the pre-defined VPN connection.

2.2.3 SSL Settings

SSL connections can be enabled by configuring the options available under the **SSL** tab.



☑ Use SSL

Peer Name: JMS SSL

Cipher Suite: SSL_RSA_EXPORT1024_WITH_DES_CBC_SHA

Fips Required: ☑

KeyResetCount: 2000

Trust Store: C:\JarFiles\keystore\MQ SSL Keytores\qm_sterling.jks

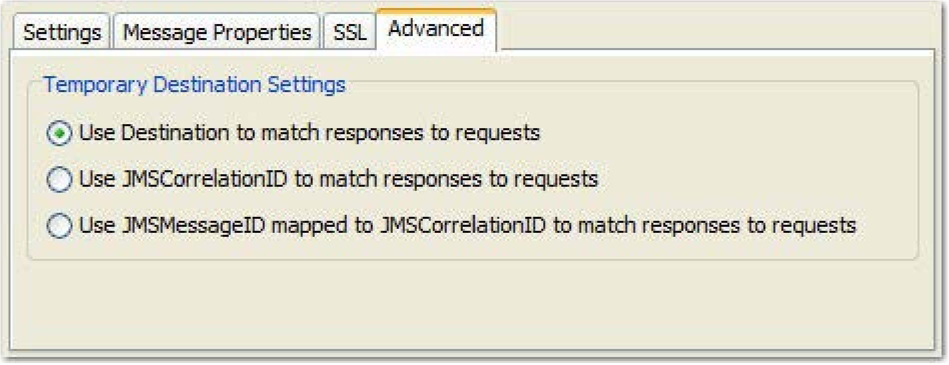
Key Store: C:\JarFiles\keystore\MQ SSL Keytores\qm_sterling.jks

To enable SSL, tick the box next to **Use SSL**. The configuration options are described in the following table:

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.
Cipher Suite	The cipher suite to use for the connection. Select one of the available suites from the dropdown menu.
Fips Required	Specifies whether the requested cipher suites must use FIPS-certified cryptography.
KeyResetCount	The total number of non-encrypted bytes that are sent and received within an SSL conversation before the secret key is renegotiated. If set to zero (default), the secret key is never renegotiated.
Trust Store	Specifies the trust store to use for the connection. Select the previously configured Identity Store that contains the desired trust store.
Key Store	Specifies the key store to use for the connection. Select the previously configured Identity Store that contains the desired key store.

2.2.4 Advanced Settings

The use of temporary destinations for receiving replies can be configured under the **Advanced** tab.



Temporary Destination Settings

● Use Destination to match responses to requests

○ Use JMSCorrelationID to match responses to requests

○ Use JMSMessageID mapped to JMSCorrelationID to match responses to requests

The details of each option are described below:

Use Destination to match responses to requests

The default option creates a unique temporary destination for receiving responses. Using this configuration, a separate reply destination is created for each published request.

Use JMSCorrelationID to match responses to requests

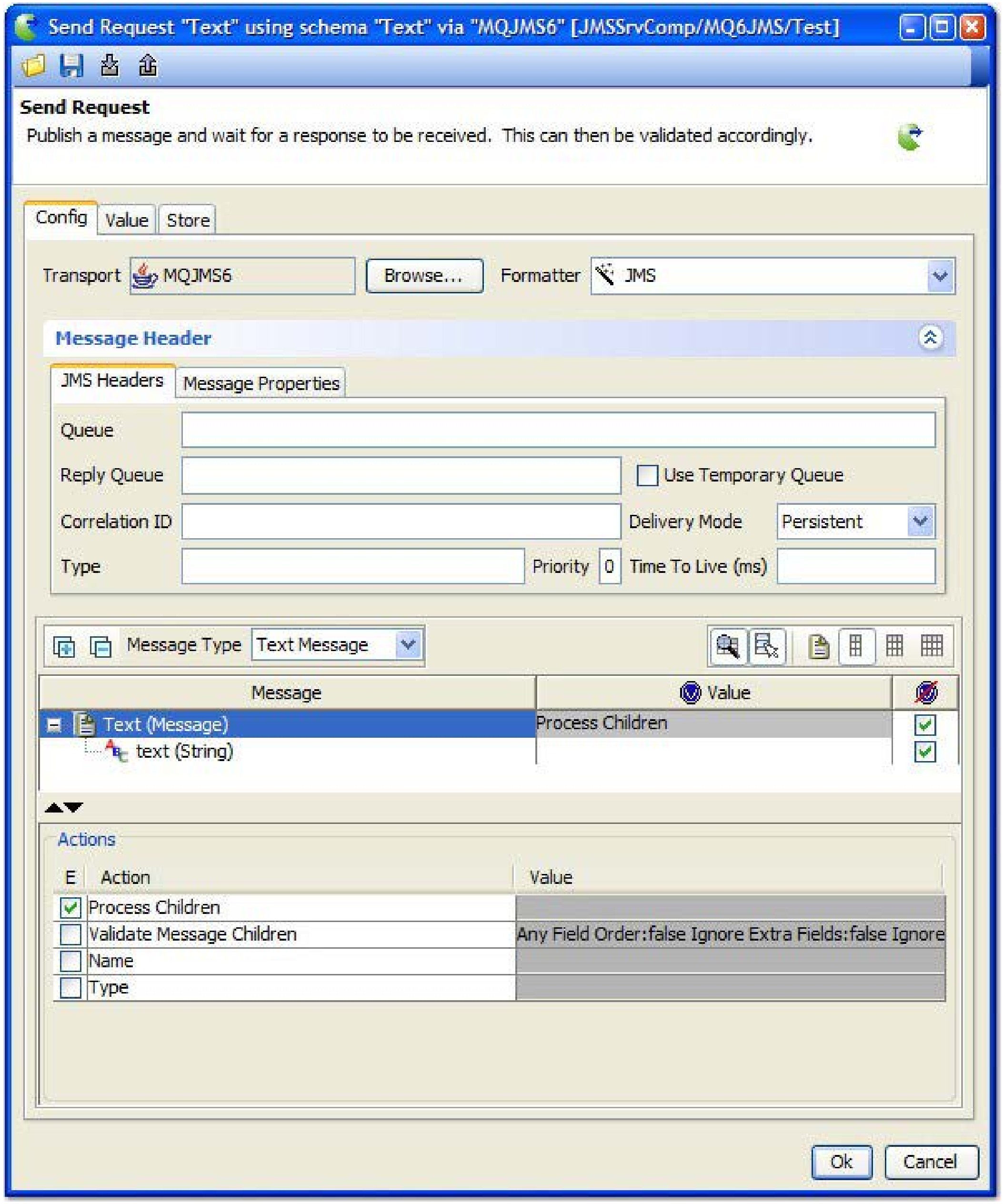
This option uses a single temporary reply destination, using the JMSCorrelationID (set as an additional message property) to correlate responses with requests.

Use JMSMessageID mapped to JMSCorrelationID to match responses to requests

This option uses a single temporary reply destination, mapping the JMSMessageID (set as an additional message property) to the JMSCorrelationID to correlate responses with requests.

2.3 Sending JMS Messages

When publishing by means of JMS, you must configure the message header (see Configure JMS Headers and Configure Message Properties) and the message body (Configure the JMS Message Body).



Send Request "Text" using schema "Text" via "MQJMS6" [JMSSrvComp/MQ6JMS/Test]

Send Request

Publish a message and wait for a response to be received. This can then be validated accordingly.

Transport: MQJMS6

Formatter: JMS

Queue:

Reply Queue: □ Use Temporary Queue

Correlation ID: Delivery Mode: Persistent

Type: Priority: 0 Time To Live (ms):

Message Type: Text Message

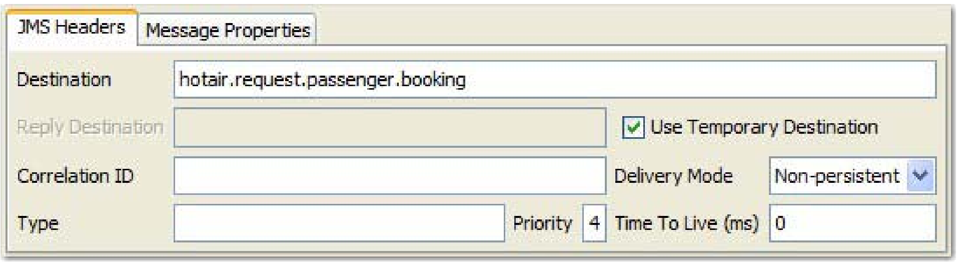
Message	Value	
Text (Message)	Process Children	☑
text (String)		☑

Actions

E	Action	Value
☑	Process Children	
□	Validate Message Children	Any Field Order:false Ignore Extra Fields:false Ignore
□	Name	
□	Type	

2.3.1 Configure JMS Headers

JMS header information is configured under the **JMS Headers** tab.



JMS Headers | Message Properties

Destination: hotair.request.passenger.booking

Reply Destination:

☑ Use Temporary Destination

Correlation ID:

Delivery Mode: Non-persistent

Type:

Priority: 4

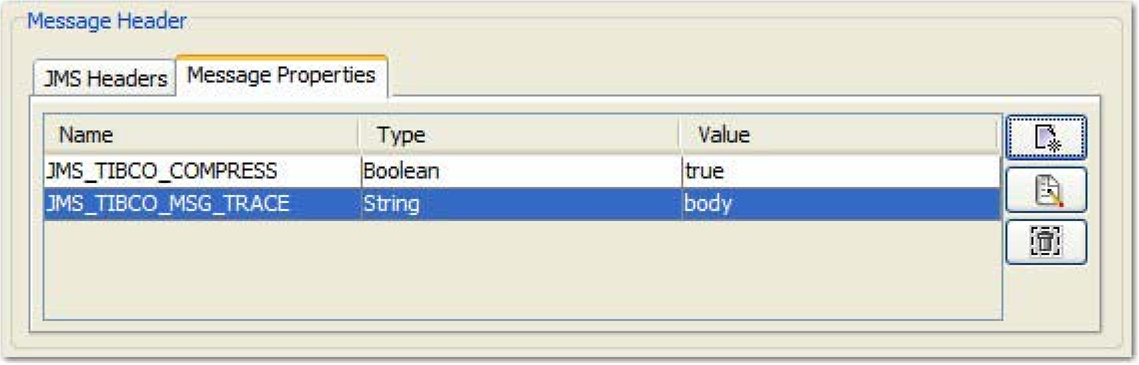
Time To Live (ms): 0

The JMS header options are described in the following table:

Option	Description
Destination	The destination to which the message will be sent.
Reply Destination	An optional destination to which a message reply should be sent.
Use Temporary Destination	Enable this option to create and use a temporary destination.
Correlation ID	An optional ID that can be used to link messages, such as linking a response message to a request message.
Delivery Mode	The delivery mode to use, Persistent or Non-persistent.
Type	The message type identifier.
Priority	A numerical ranking of the message priority, between 0 and 9. Larger numbers represent higher priority.
Time to Live (ms)	Length of time (in milliseconds) the message will live before it expires. If set to 0, the message will not expire. If the server expiration property is set for a destination, it will override this value.

2.3.2 Configure Message Properties

JMS-specific properties can be managed under the **Message Properties** tab.



Name	Type	Value
JMS_TIBCO_COMPRESS	Boolean	true
JMS_TIBCO_MSG_TRACE	String	body

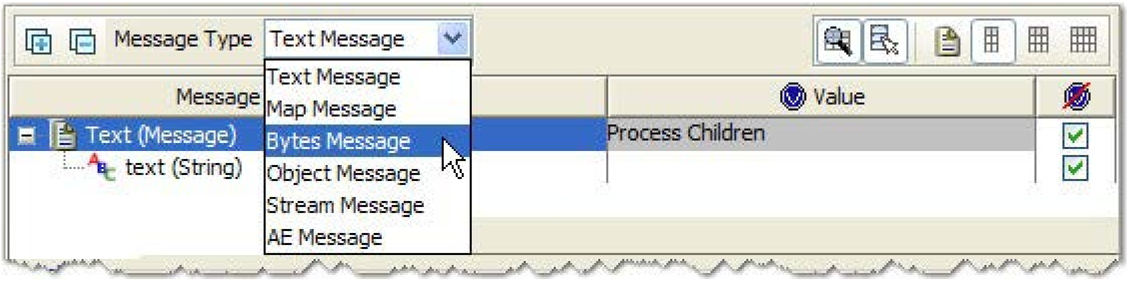
Each property has a name, type, and value.

- To add a new property, click 📄.
- To edit an existing property, select it and click ✎.
- To delete a property, select it and click 🗑.

NOTE: See your JMS provider documentation for more information about additional message properties.

2.3.3 Configure the JMS Message Body

The contents and structure of an JMS message body will vary according to the message type. To set the message type, select one of the options available from the **Message Type** dropdown menu.

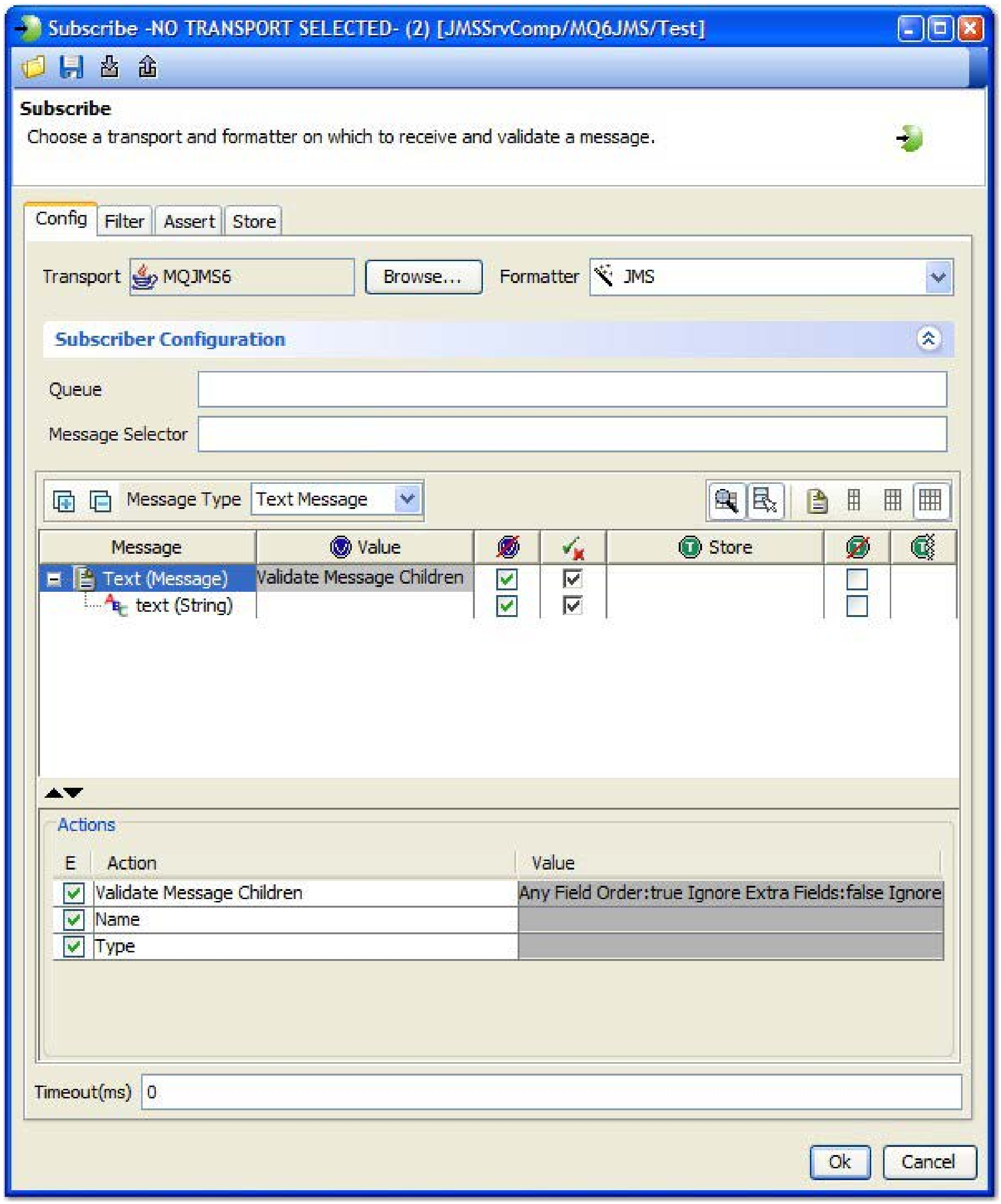

Message Type Text Message

Message	Value	
Text (Message)	Process Children	☑
text (String)		☑

For more information about configuring messages, refer to *IBM Rational Integration Tester Reference Guide.*

2.4 Receiving JMS Messages

When receiving messages by means of JMS, you must configure the subscriber options (Configure Subscriber Options), message content (Configure Message Content), and optional filtering (Message Filtering).



Subscribe -NO TRANSPORT SELECTED- (2) [JMSSrvComp/MQ6JMS/Test]

Subscribe

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

Transport: MQJMS6

Formatter: JMS

Subscriber Configuration

Queue:

Message Selector:

Message Type: Text Message

Message	Value			Store		
Text (Message)	Validate Message Children	☑	☑		□	
text (String)		☑	☑		□	

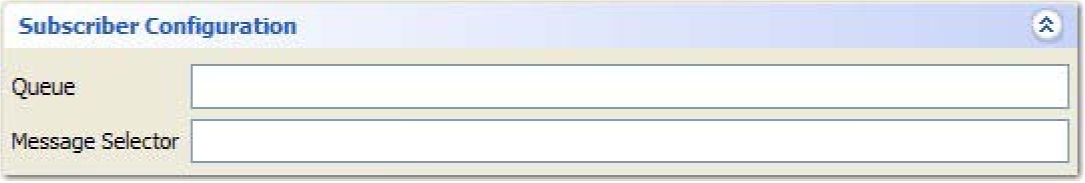
Actions

E	Action	Value
☑	Validate Message Children	Any Field Order:true Ignore Extra Fields:false Ignore
☑	Name	
☑	Type	

Timeout(ms): 0

2.4.1 Configure Subscriber Options

Subscriber options for receiving messages by means of the JMS transport are managed under **Subscriber Configuration**.

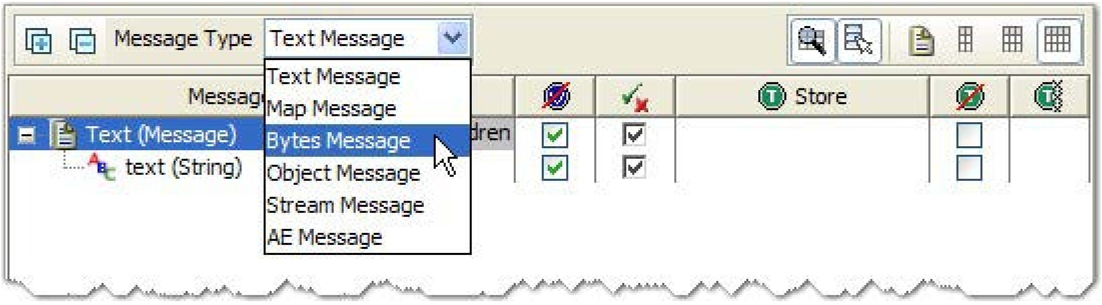


The subscriber options are described in the following table:

Option	Description
Queue	The destination, queue, or topic to monitor for incoming messages.
Message Selector	Filters incoming messages according to message header properties (see Message Filtering for more information).

2.4.2 Configure Message Content

The contents and structure of an JMS message body will vary according to the message type. To set the type of messages to receive, select one of the options available from the **Message Type** dropdown menu.


Message Type: Text Message

For more information about configuring message content, refer to *IBM Rational Integration Tester Reference Guide.*

2.4.3 Message Filtering

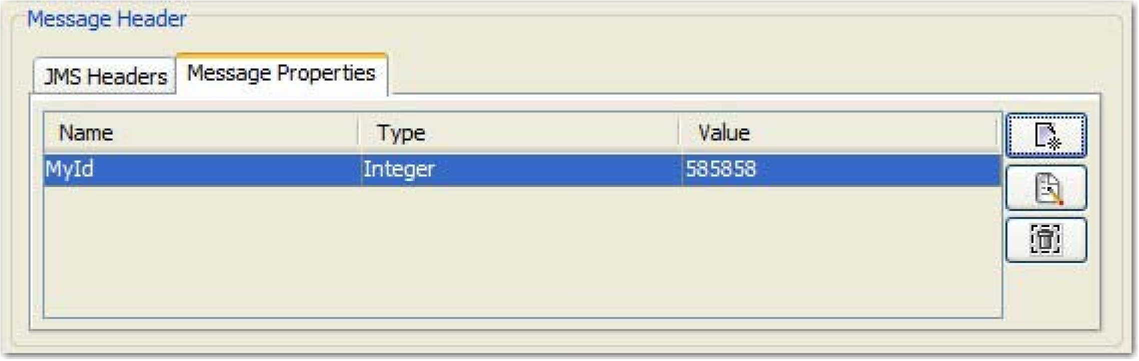
You can filter the way messages are received in two different ways: JMS Filtering and Rational Integration Tester Filtering.

NOTE: When using filtering, the system under test must be correctly configured. Messages that are filtered out are silently discarded – Rational Integration Tester will not produce any warnings or errors.

JMS Filtering

The **Message Selector** field under **Subscriber Configuration** (see Configure Subscriber Options) accepts standard JMS filtering expressions (message selectors). When utilized, Rational Integration Tester only receives messages that match the specified selector.

For example, if `MyId='585858'` is entered, only messages with the header field "MyId" having a value of 585858 will be received. To send matching messages in a Rational Integration Tester publisher, you would configure a message property as shown below:

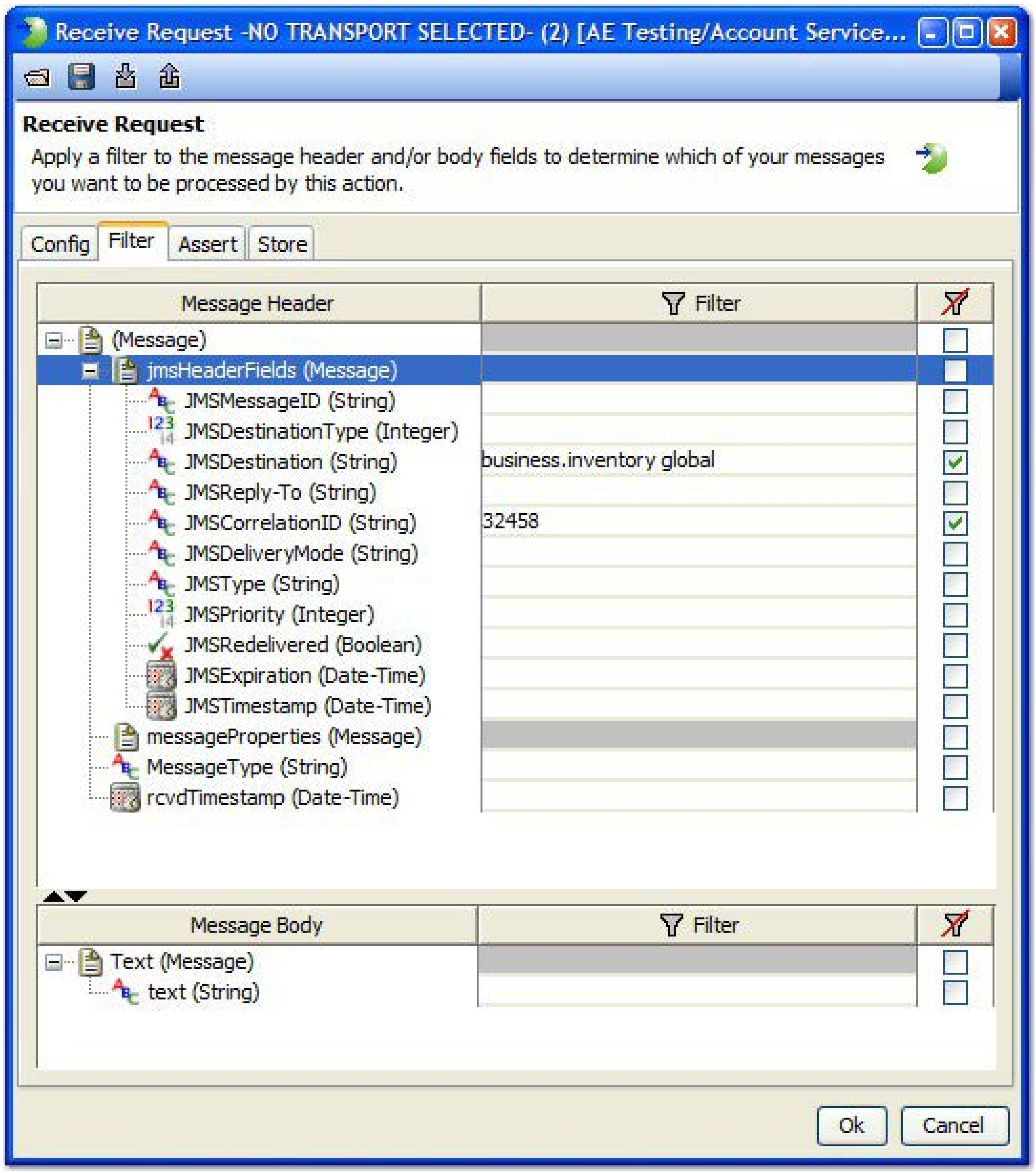


Name	Type	Value
MyId	Integer	585858

NOTE: Filtering is performed on message header properties, and only matching messages are passed to Rational Integration Tester (that is, the JMS server filters the messages, not Rational Integration Tester). This is useful in environments that share a single JMS queue or topic. By getting the applications to set header properties, you can separate messages logically. Rational Integration Tester can interact with messages from Application A without affecting those of Application B.

Rational Integration Tester Filtering

After messages have been passed to Rational Integration Tester, they may be further filtered (using header and body fields) with the configuration in the **Filter** tab.



Receive Request -NO TRANSPORT SELECTED- (2) [AE Testing/Account Service...
Config
Filter
Assert
Store
Ok
Cancel

Receive Request

Apply a filter to the message header and/or body fields to determine which of your messages you want to be processed by this action.

Message Header	Filter	
(Message)		□
jmsHeaderFields (Message)		□
JMSMessageID (String)		□
JMSDestinationType (Integer)		□
JMSDestination (String)	business.inventory global	☑
JMSReply-To (String)		□
JMSCorrelationID (String)	32458	☑
JMSDeliveryMode (String)		□
JMSType (String)		□
JMSPriority (Integer)		□
JMSRedelivered (Boolean)		□
JMSExpiration (Date-Time)		□
JMSTimestamp (Date-Time)		□
messageProperties (Message)		□
MessageType (String)		□
rcvdTimestamp (Date-Time)		□

Message Body	Filter	
Text (Message)		□
text (String)		□

In this case, **JMSDestination** must equal “business.inventory global” and **JMSCorrelationID** must equal “32458”. Otherwise, Rational Integration Tester will discard the message.

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. See also Message.
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 on which a software process runs.
Publisher-Subscriber	A messaging paradigm whereby a messaging network consists of Publishers and Subscribers.
Transport	Informally, the messaging software in use. For instance, TIBCO Rendezvous, TIBCO ActiveEnterprise, IBM WebSphere® MQ (JMS).
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.

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.

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.

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.

IBM®