

Rational Integration Tester



Reference Guide for TIBCO

Version 8.0.0



Note

Before using this information and the product it supports, read the information in “Notices” on page 103.

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

About this Publication	v
Intended Audience	vi
Scope	vi
Typographical Conventions	vi
Contacting IBM Support	vi
Requirements	1
TIBCO Rendezvous	2
TIBCO Active Enterprise	2
TIBCO EMS	2
TIBCO SmartSockets	2
Rendezvous Transport	3
Libraries	4
Creating the Rendezvous Transport	5
Configuring the Rendezvous Transport	6
Sending RV Messages	10
Receiving RV Messages	12
EMS Transport	15
Libraries	16
Creating the EMS Transport	17
Configuring the EMS Transport	18
Sending EMS Messages	25
Receiving EMS Messages	29
Testing SSL in TIBCO EMS	35

Configuration Files	38
Testing TIBCO BusinessWorks and Design Time Libraries	41
Overview	42
Add a BusinessWorks Project	43
Add a Design Time Library	49
Monitor TIBCO BusinessWorks	52
Example: The Hotair Scenario	56
Testing BusinessWorks Private Processes	79
SmartSockets Transport	85
Libraries	86
Creating the SmartSockets Transport	87
Configuring the SmartSockets Transport	88
Sending SmartSockets Messages	90
Receiving SmartSockets Messages	93
Troubleshooting	97
No Formatter Available	98
Unable to Connect to Daemon	98
Overlapping Subjects Error	98
Class Definitions Not Visible	98
AE Transport Unable to Use BusinessWorks 5 Projects	98
Unable to Watch EMS Queues	99
Unable to Access Topics/Queues on Remote Servers	99
“Not Permitted” Error When Watching	99
Unable to Tag Generated Tracking ID	100
Failed to Rebuild Schema	100
Glossary	101
Notices	103
Trademarks and service marks	106

About this Publication

Contents

Intended Audience

Scope

Typographical Conventions

Contacting IBM Support

This guide describes how to configure and run IBM® Rational® Integration Tester with the TIBCO messaging plugin, supporting TIBCO Rendezvous messaging (“plain” TIBCO Rendezvous messaging and TIBCO ActiveEnterprise (AE) formats), TIBCO EMS messaging, and sending and receiving AE messages using TIBCO EMS.

This guide also shows you how to create a Rational Integration Tester project from a BusinessWorks 5 project.

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 discusses the configuration and use of IBM Rational Integration Tester alongside TIBCO technologies. If you wish to familiarize yourself with these technologies, please refer to documents provided by the relevant companies or individuals.

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

TIBCO Rendezvous

TIBCO Active Enterprise

TIBCO EMS

TIBCO SmartSockets

The system requirements for using the TIBCO messaging plugin are the same as those for installing and using Rational Integration Tester. For information about this, refer to *IBM Rational Integration Tester Installation Guide*.

To enable support for all of the supported Rational Integration Tester transports, certain TIBCO applications and packages are required. This chapter provides more information about these requirements.

1.1 TIBCO Rendezvous

To use TIBCO Rendezvous messaging, Rendezvous v6 or later must be installed.

1.2 TIBCO Active Enterprise

To use TIBCO Active Enterprise messaging, one or more of the following product sets should be installed and configured:

- TIBCO Integration Manager 4.0 (or later)
- TIBCO BusinessWorks 5.1 (or later)
- TIBCO TRA 5.x
- TIBCO EMS 4.x

1.3 TIBCO EMS

To use TIBCO EMS messaging, TIBCO EMS v4.x or higher should be installed and configured.

1.4 TIBCO SmartSockets

To use TIBCO SmartSockets, TIBCO SmartSockets v6.x should be installed and configured.

Rendezvous Transport

Contents

Libraries

Creating the Rendezvous Transport

Configuring the Rendezvous Transport

Sending RV Messages

Receiving RV Messages

This chapter describes the libraries that Rational Integration Tester needs to work with TIBCO Rendezvous. It also describes how to configure the Rendezvous transport.

2.1 Libraries

Depending on the version of TIBCO Rendezvous in use, specific product libraries are required. The following table describes the libraries that are required along with their default location.

Rendezvous Version	Library	Default Location
v6 and 7	tibrvj.jar	c:\tibco\tibrv\lib
v8.1	tibrvj.jar	c:\tibco\tibrv\8.1\lib
v8.3	tibrvj.jar	c:\tibco\tibrv\8.3\lib

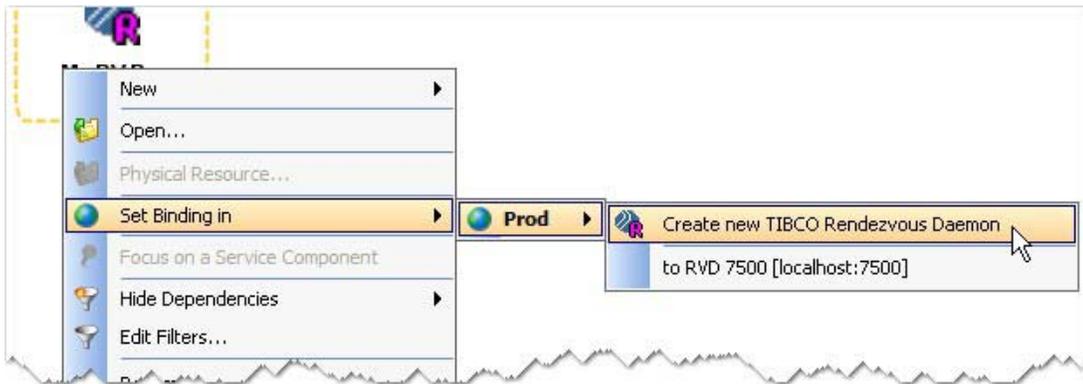
NOTE: Default locations for the specified libraries can be modified using Library Manager. For information about this, refer to *IBM Rational Integration Tester Installation Guide* for more information.

2.2 Creating the Rendezvous Transport

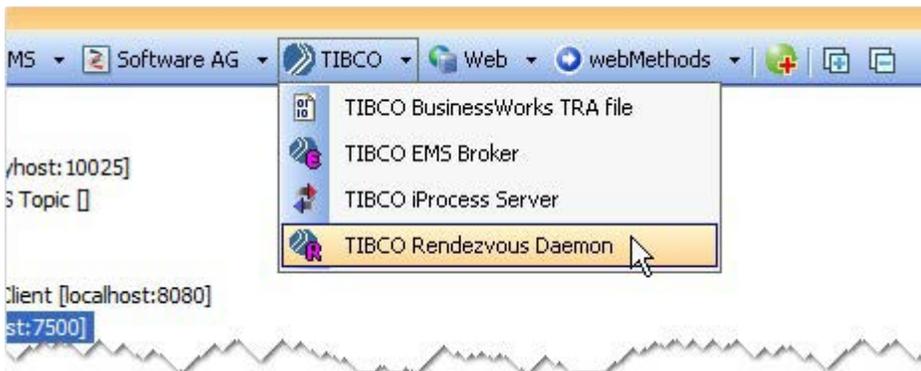
The Rendezvous transport is created when you create a physical TIBCO Rendezvous Daemon resource in Rational Integration Tester's Architecture School.

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

- In the Logical View, right-click on a TIBCO Rendezvous Bus and select the **Set Binding in > [environment] > Create new TIBCO Rendezvous Daemon** option.



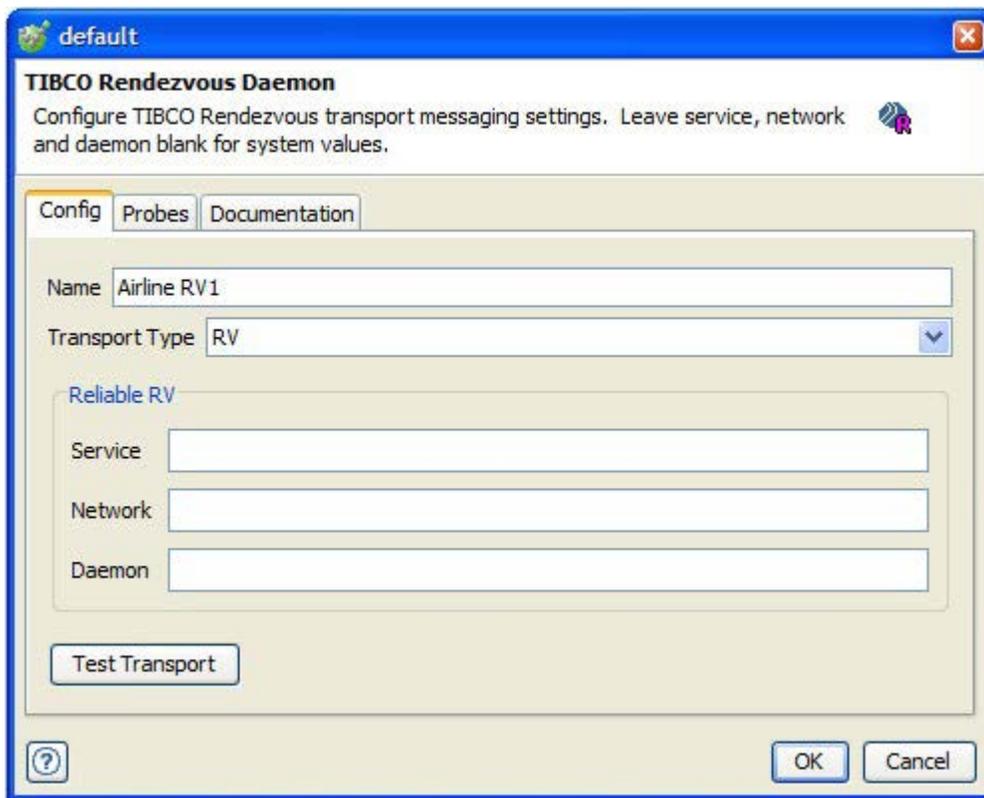
- In the Physical View, select the **TIBCO > TIBCO Rendezvous Daemon** option.



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

2.3 Configuring the Rendezvous Transport

To configure a Rendezvous transport, double-click the appropriate TIBCO Rendezvous Daemon 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 Rendezvous transports are available).

Next, select the connection type from the **Transport Type** menu. The Rendezvous transport provides support for plain (reliable) RV, Certified, and Distributed Queue connections to the daemon process. See [Plain RV](#), [Certified Messaging](#), or [Distributed Queues](#) for more information.

NOTE: All of the configuration fields support the use of tags, which can be entered manually or by selecting them from the context menu.

NOTE: You can test the connection parameters at any time by clicking the **Test Transport** button.

2.3.1 Plain RV

The plain RV transport must be supplied with the basic connection parameters to the Rendezvous daemon.



The image shows a dialog box titled "Reliable RV" with a light beige background and a thin border. Inside the dialog, there are three vertically stacked input fields. The first field is labeled "Service", the second is labeled "Network", and the third is labeled "Daemon". Each label is positioned to the left of its corresponding empty text box.

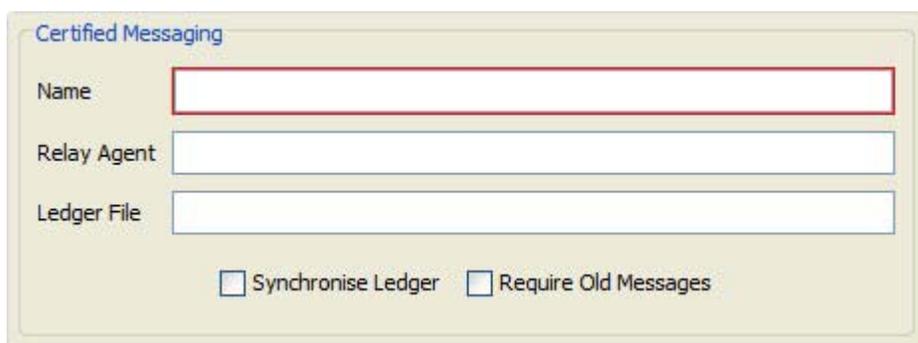
Service	The service name, number or default entry that the daemon should listen on.
---------	---

Network	Specifies the network interface that the daemon should connect on.
---------	--

Daemon	Tells Rational Integration Tester how to connect to the Rendezvous daemon.
--------	--

2.3.2 Certified Messaging

To send and receive messages using the certified delivery features of Rendezvous, Rational Integration Tester utilizes a Certified Messaging (RVCM) transport. Certified Messaging requires the same RV daemon connection settings (see [Plain RV](#)), as well several additional options (for example, a ledger file for saving tracking details).



The image shows a configuration dialog box titled "Certified Messaging". It contains three text input fields: "Name", "Relay Agent", and "Ledger File". Below these fields are two checkboxes: "Synchronise Ledger" and "Require Old Messages".

Name	The correspondence name used to identify this RVCM transport to other RVCM transports (mandatory).
Relay Agent	Specifies a Relay Agent that is used for certified delivery when persistent correspondence is only connecting to the network intermittently.
Ledger File	Specifies the location of the optional file where all the correspondence between RVCM transports is stored.
Synchronise Ledger	Forces any operations in the transport daemon that update the ledger file to block until all data has been written to the file. Otherwise, all changes are written asynchronously.
Require Old Messages	Saves all messages sent to the ledger file. In this case, an RVCM transport could use the message history to continue interrupted operations (for example, in case of a network failure).

NOTE: Additional information about these parameters can be found in the TIBCO Rendezvous documentation.

2.3.3 Distributed Queues

To configure a transport as a member of an RVCN distributed queue, additional parameters are required in addition to the same RV daemon connection settings (see [Plain RV](#)).

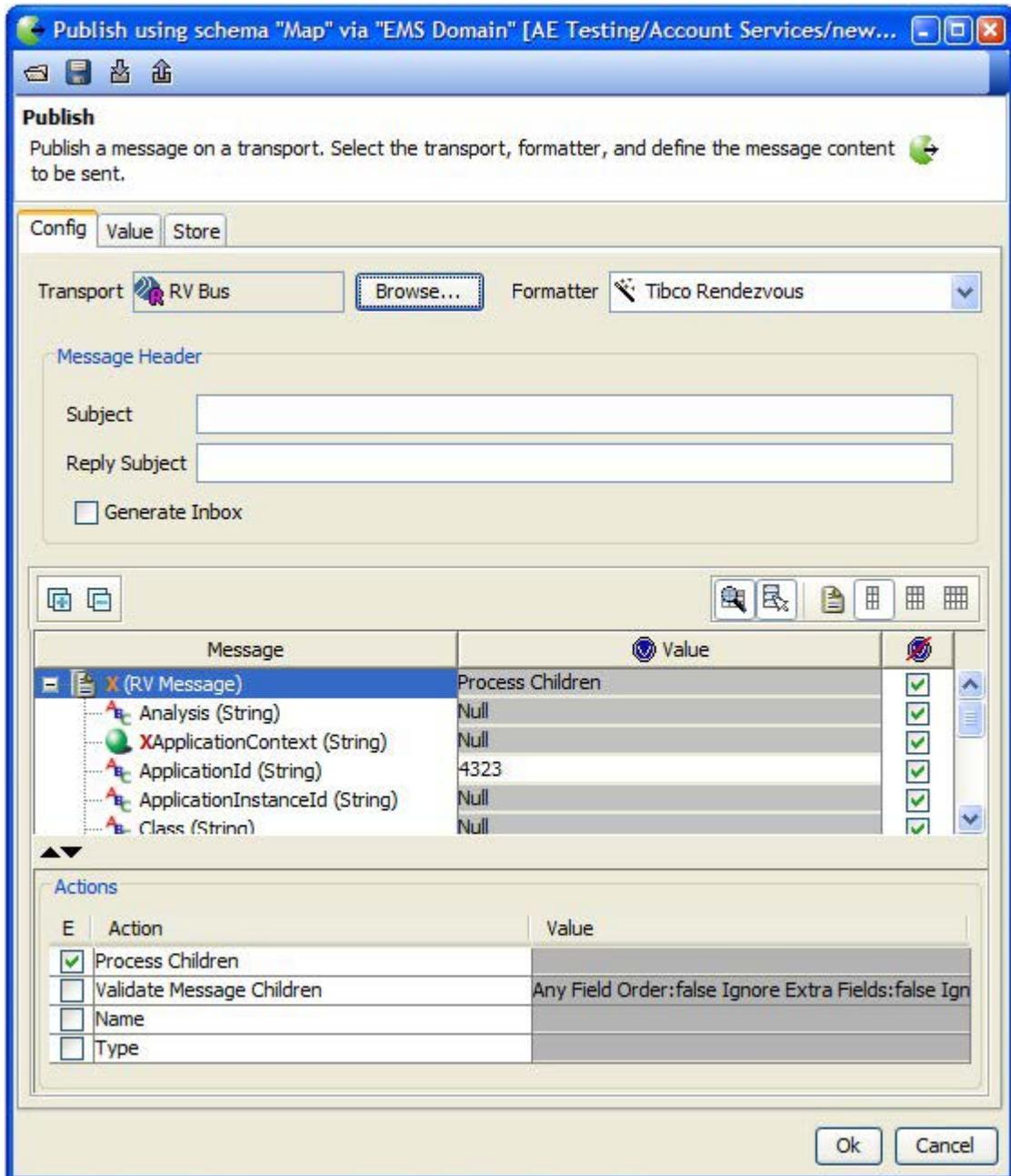
The image shows a configuration window titled "Distributed Queue". It contains six input fields, each with a label to its left: "Queue Name", "Worker Weight", "Worker Tasks", "Scheduler Weight", "Scheduler Heartbeat (sec)", and "Scheduler Activation (sec)". The "Queue Name" field is highlighted with a red border.

Queue Name	The unique identifier of the queue with which this transport is associated (mandatory).
Worker Weight	In a queue, tasks are assigned to transports according to this value. Setting a higher number (relative to other weights) will result in more tasks being assigned to this transport.
Worker Tasks	Specifies the maximum amount of tasks this transport can accept before delegating to other workers.
Scheduler Weight	If a new scheduler is required, the one with the greatest weight will be assigned next.
Scheduler Heartbeat	The amount of delay (in seconds) the scheduler should use between the heartbeat signals it broadcasts.
Scheduler Activation	The amount of time (in seconds) that an inactive scheduler should wait before becoming active when a heartbeat signal is lost.

NOTE: When using a distributed queue transport in Rational Integration Tester, all publish operations will be performed on the underlying plain RV transport.

2.4 Sending RV Messages

When publishing by means of RV, you must configure the message header (see [Configure the Message Header](#)) and the message body ([Configure the Message Body](#)).



2.4.1 Configure the Message Header

RV header information is configured under **Message Header**.



The screenshot shows a dialog box titled "Message Header" with a light beige background. It contains two text input fields: "Subject" and "Reply Subject". Below these fields is a checkbox labeled "Generate Inbox".

The RV header options are described in the following table:

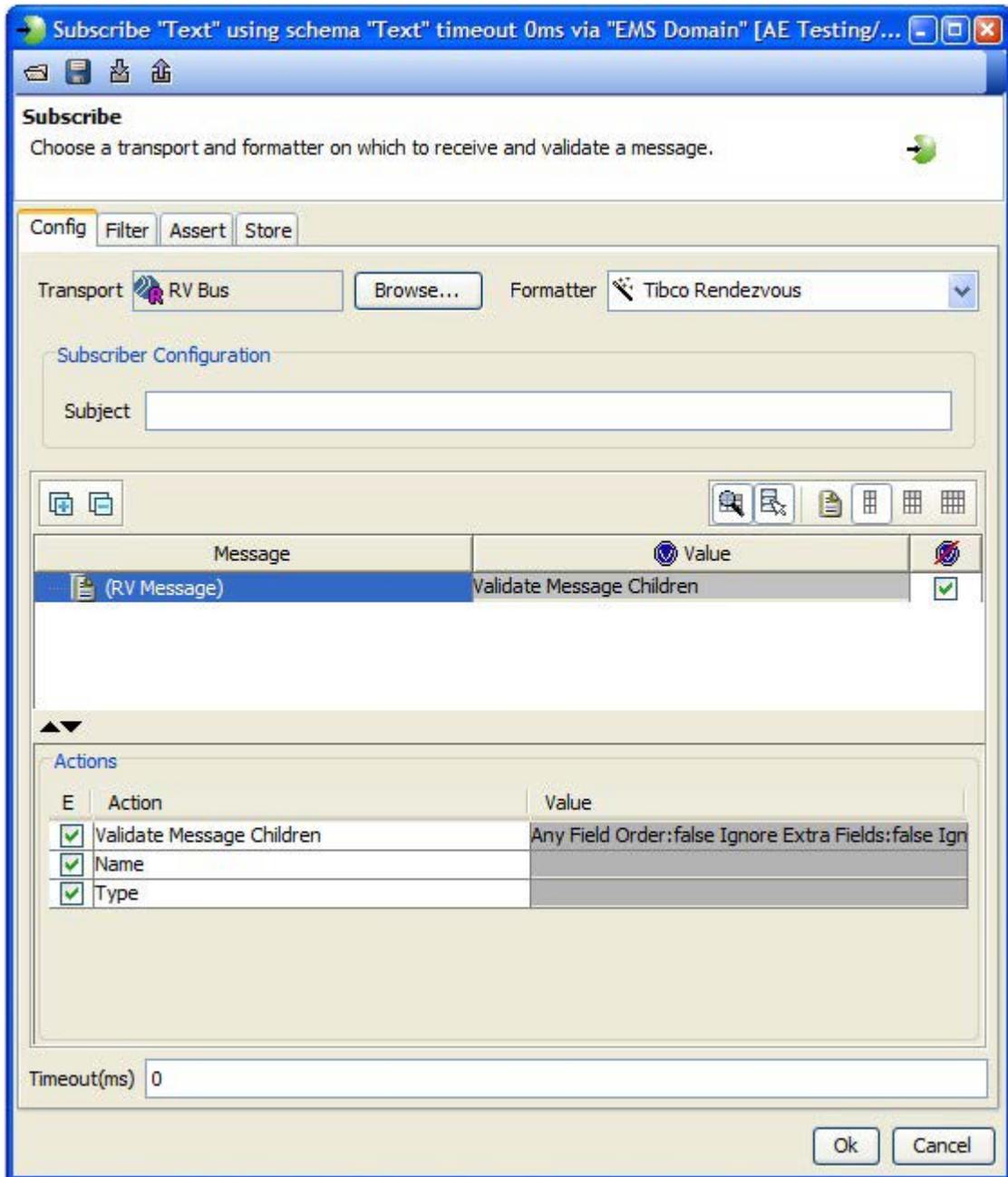
Option	Description
Subject	The subject (destination) of the outgoing message.
Reply Subject	The optional return address to which replies should be sent.
Generate Inbox	Generates an inbox, which is a specific destination for messages in point-to-point communications.

2.4.2 Configure the Message Body

The contents of an RV message body are configured below the header. For more information about configuring messages, refer to *IBM Rational Integration Tester Reference Guide*.

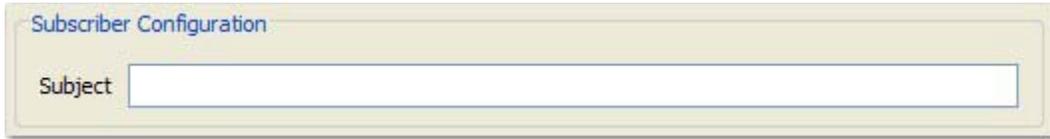
2.5 Receiving RV Messages

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



2.5.1 Configure Subscriber Options

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



The image shows a screenshot of a software dialog box titled "Subscriber Configuration". Inside the dialog, there is a single text input field with the label "Subject" positioned to its left. The input field is currently empty.

The only option to configure is the **Subject**, which is the message destination to listen to for incoming messages.

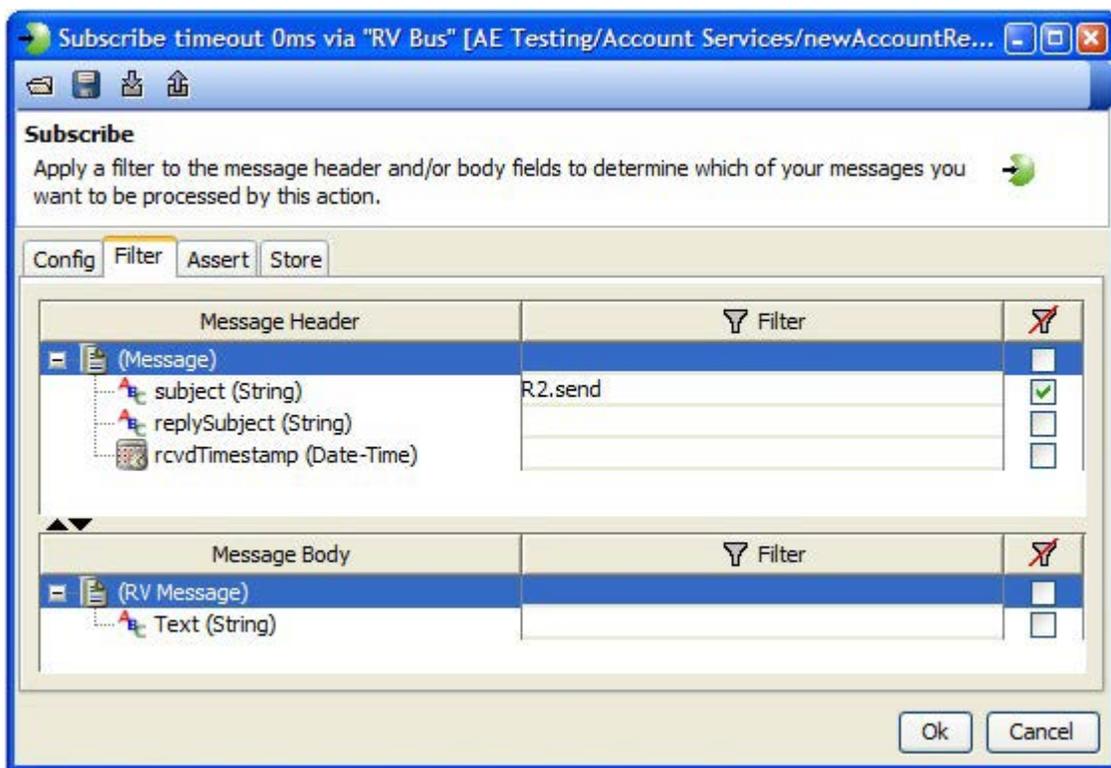
2.5.2 Configure Message Content

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

2.5.3 Message Filtering

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

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.



In this case, **subject** must equal "R2.send." Otherwise, Rational Integration Tester will discard the message.

EMS Transport

Contents

Libraries

Creating the EMS Transport

Configuring the EMS Transport

Sending EMS Messages

Receiving EMS Messages

Testing SSL in TIBCO EMS

This chapter describes the libraries that Rational Integration Tester needs to work with TIBCO EMS. It also describes how to configure the EMS transport.

3.1 Libraries

Depending on the version of TIBCO EMS in use, specific product libraries are required. The following table describes the libraries that are required along with their default location.

EMS Version	Library	Default Location
v4.x	tibjms.jar	c:\tibco\ems\clients\java
	tibjmsadmin.jar	
	tibrvjms.jar	
v5.0	tibjms.jar	c:\tibco\ems\5.0\lib
	tibjmsadmin.jar	
	tibrvjms.jar	
v6.0	tibjms.jar	c:\tibco\ems\6.0\lib
	tibjmsadmin.jar	
	tibrvjms.jar	

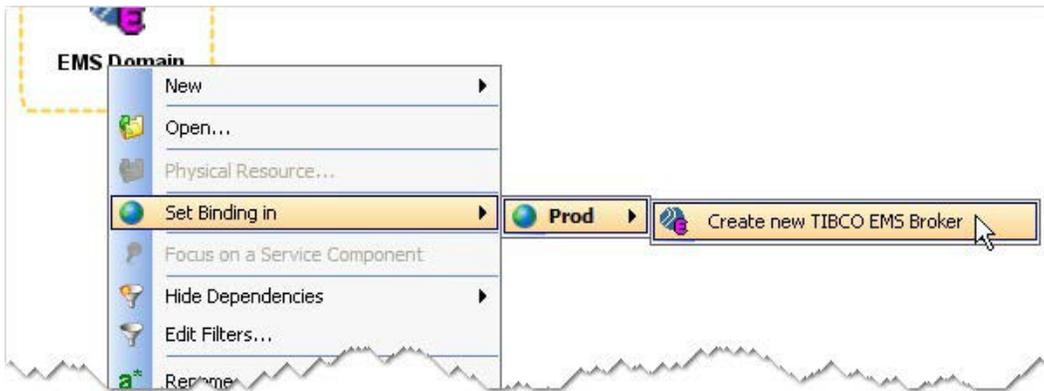
NOTE: Default locations for the specified libraries can be modified using Library Manager. For more information, refer to *IBM Rational Integration Tester Installation Guide*.

3.2 Creating the EMS Transport

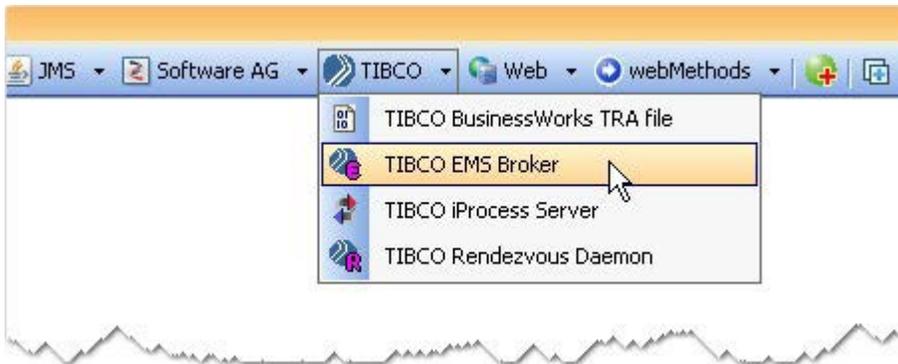
The TIBCO EMS transport is created when you create a physical TIBCO EMS Broker resource in Rational Integration Tester's Architecture School.

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

- In the Logical View, right-click on a TIBCO EMS Domain and select the **Set Binding in > [environment] > Create new TIBCO EMS Broker** option.



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

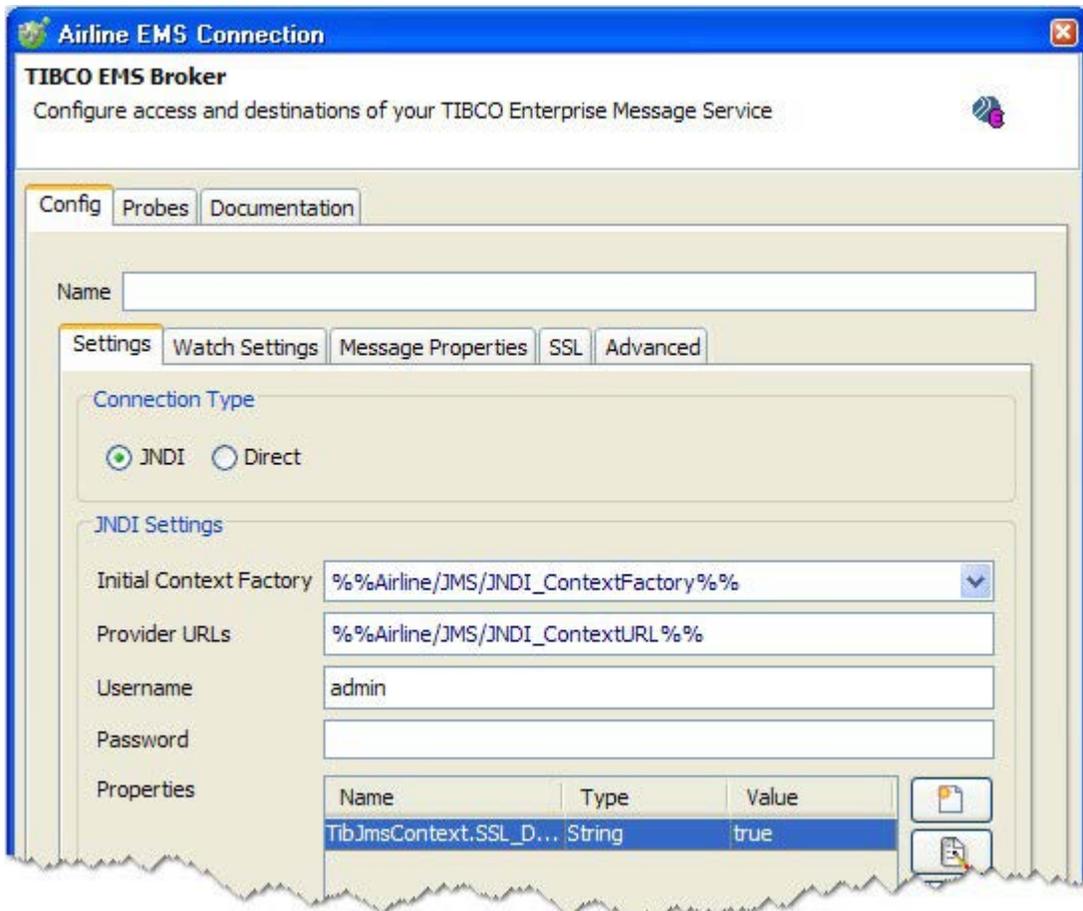


Each physical EMS Broker resource will represent a TIBCO EMS Broker transport that can be selected and configured later on.

3.3 Configuring the EMS Transport

To configure an EMS transport, double-click the appropriate TIBCO EMS Broker resource in Architecture School's Physical View.

The configuration of the EMS transport is divided among several different areas, including [JNDI Settings](#), [Connection Settings](#), [Broker Connection Settings](#), [Watch Settings](#), [Message Properties](#), [SSL Settings](#), and [Advanced Settings](#).



If desired, enter a name for the transport in the **Name** field (to help identify it when multiple EMS transports are available). Additionally, select the connection type (**JNDI** or **Direct**) under **Connection Type**.

NOTE: When configuring the transport, 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, `%%JMS_password%%`). Since this field is encrypted, any characters entered will be hidden.

3.3.1 JNDI Settings

If the Connection Type is **JNDI**, the **JNDI Settings** are available under the **Settings** tab for configuring how to look up EMS topics and queues.

The screenshot shows the 'JNDI Settings' dialog box. It has the following fields and content:

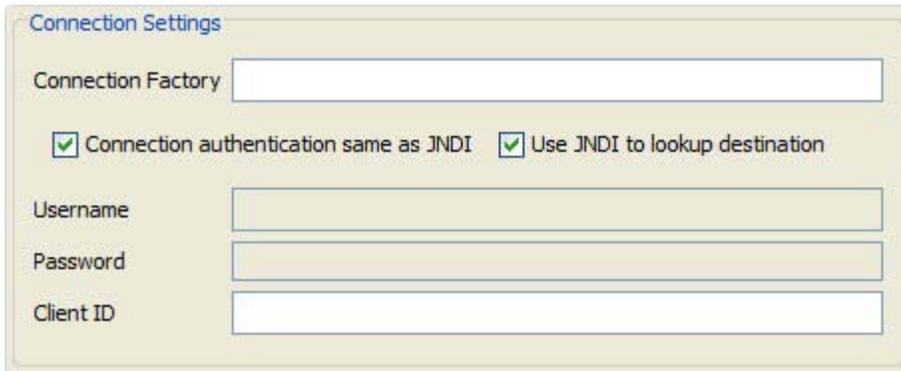
- Initial Context Factory:** A dropdown menu with the value '%%Airline/JMS/JNDI_ContextFactory%%'.
- Provider URLs:** A text field with the value '%%Airline/JMS/JNDI_ContextURL%%'.
- Username:** A text field with the value 'admin'.
- Password:** An empty text field.
- Properties:** A table with three columns: Name, Type, and Value. It contains one row: 'TibJmsContext.SSL_DE...' (Name), 'String' (Type), and 'true' (Value). To the right of the table are three icons: a plus sign, a pencil, and a trash can.

The available configuration options are described in the following table:

Field	Description
Initial Context Factory	Set to com.tibco.tibjms.naming.TibjmsInitialContextFactory for the EMS transport. You should not need to change this option.
Provider URLs	The server and port of the EMS server to use for lookup, in the form of tibjmsnaming://<host name>:<port>. The default port number is 7222. For example, tibjmsnaming://myserver:7272.
Username & Password	If security is enabled on the EMS server, these are the user name and password that have been configured to access the topics and queues (contact your system or TIBCO administrator if you don't know these values). If security is not in use, these fields may be left blank.
Properties	Additional JNDI properties to use (for example, when secure connections are made to the JNDI). Each property has a name, type, and value. <ul style="list-style-type: none">- To add a new property, click .- To edit an existing property, select it and click .- To delete a property, select it and click .

3.3.2 Connection Settings

If the Connection Type is **JNDI**, the **Connection Settings** are available under the **Settings** tab.



The screenshot shows a dialog box titled "Connection Settings". It contains the following elements:

- A text input field labeled "Connection Factory".
- Two checked checkboxes: "Connection authentication same as JNDI" and "Use JNDI to lookup destination".
- A text input field labeled "Username".
- A text input field labeled "Password".
- A text input field labeled "Client ID".

The available configuration options are described in the following table:

Field	Description								
Connection Factory	<p>The connection factory used to locate topics and queues. To see the list of available factories, use the show factories command in the TIBCO EMS Admin (tibemsadmin) utility:</p> <pre>tcp://myserver:7272> show factories</pre> <table border="1"><thead><tr><th>Factory Type</th><th>JNDI Names</th></tr></thead><tbody><tr><td>TopicConnectionFactory</td><td>"TopicFactory"</td></tr><tr><td>QueueConnectionFactory</td><td>"QAqueues"</td></tr><tr><td>TopicConnectionFactory</td><td>"TopicConnectionFactory"</td></tr></tbody></table> <p>To see settings for a specific factory use the following command:</p> <pre>show factory <factoryname></pre>	Factory Type	JNDI Names	TopicConnectionFactory	"TopicFactory"	QueueConnectionFactory	"QAqueues"	TopicConnectionFactory	"TopicConnectionFactory"
Factory Type	JNDI Names								
TopicConnectionFactory	"TopicFactory"								
QueueConnectionFactory	"QAqueues"								
TopicConnectionFactory	"TopicConnectionFactory"								
Connection authentication same as JNDI	<p>Enable this option to use the same user name and password for the connection factory as those for JNDI. If disabled, the Username and Password fields become active.</p>								
Use JNDI to lookup destination	<p>If you are using Rational Integration Tester to watch queue messages, this option should be disabled.</p>								
Username & Password	<p>If security is enabled on the EMS server, these are the user name and password that have been configured to access the topics and queues (contact your system or TIBCO administrator if you don't know these values). If security is not in use, these fields may be left blank.</p>								
Client ID	<p>A unique ID that identifies Rational Integration Tester to the EMS server. Typically, this ID is used to identify durable subscribers.</p>								

3.3.3 Broker Connection Settings

If the Connection Type is **Direct**, the **Broker Connection Settings** are available under the **Settings** tab.

The screenshot shows a dialog box titled "Broker Connection Settings". It contains several input fields and a table. The "Connection Factory" is a dropdown menu showing "com.tibco.tibjms.TibjmsQueueConnectionFactory". The "Broker URL" is a text box with "tcp://localhost:7222". The "Username" is a text box with "ghuser". The "Password" is a text box with seven dots. The "Client ID" is an empty text box. Below these is a "Properties" section with a table:

Name	Type	Value
TibJmsContext.SSL...	String	true

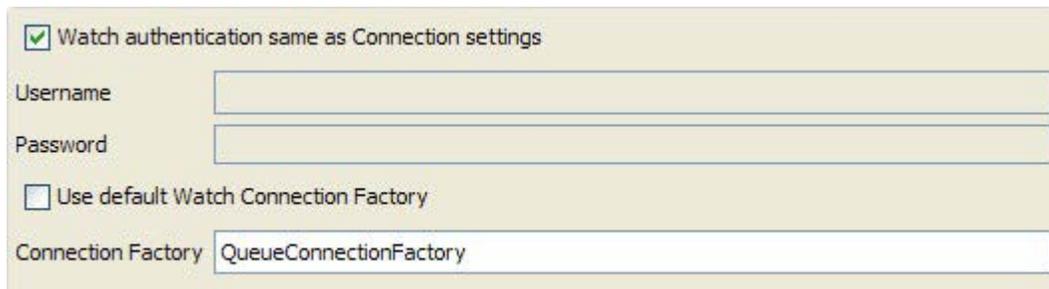
There are three icons on the right side of the Properties table: a document with a plus sign, a document with a magnifying glass, and a trash can.

The available configuration options are described in the following table:

Field	Description
Connection Factory	See <i>Initial Context Factory</i> under JNDI Settings .
Broker URL	The server and port URL of the broker to use for lookup, in the form of tcp://<host name>:<port>. The default port number is 7222. For example, tcp://myserver:7272.
Username & Password	If security is enabled on the EMS server, these are the user name and password that have been configured to access the topics and queues (contact your system or TIBCO administrator if you don't know these values). If security is not in use, these fields may be left blank.
Client ID	A unique ID that identifies Rational Integration Tester to the EMS server. Typically, this ID is used to identify durable subscribers.
Properties	See <i>Properties</i> under JNDI Settings .

3.3.4 Watch Settings

Watch settings, available under the **Watch Settings** tab, tell Rational Integration Tester how to connect to the EMS server when watching queue messages.



Watch authentication same as Connection settings

Username

Password

Use default Watch Connection Factory

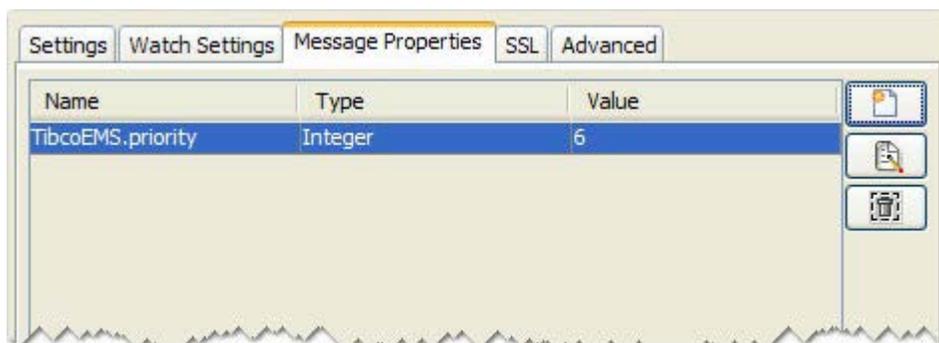
Connection Factory

NOTE: The transport login or user name/password must provide administration rights on the EMS server in question to watch messages. If this is not permitted, you may be able to request specific permissions that allow only specific queues to be accessed.

Field	Description
Watch authentication same as Connection settings	Enable this option to use the same user name and password that are used under Connection Settings . If disabled, the Username and Password fields become active.
Use default Watch Connection Factory	If enabled, the default watch connection factory will be used. If disabled, users can specify the name of another connection factory to use.

3.3.5 Message Properties

Under the **Message Properties** tab you can configure additional message properties to be used with the EMS transport. Each property has a name, type, and value. These properties will be added automatically to any messages that are created using the transport.



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

3.3.6 SSL Settings

SSL is configured using the additional JNDI properties (under [JNDI Settings](#)) in conjunction with modifications made on the EMS Server. See [Testing SSL in TIBCO EMS](#) for more information.

3.3.7 Advanced Settings

When using temporary destinations with multiple parallel test instances (for example, performance testing), temporary reply destinations may be shared by more than one publisher. Using temporary destinations can reduce the load on the EMS server.

Under the **Advanced** tab, Rational Integration Tester can be configured in a number of ways to enable the matching of requests with their replies.

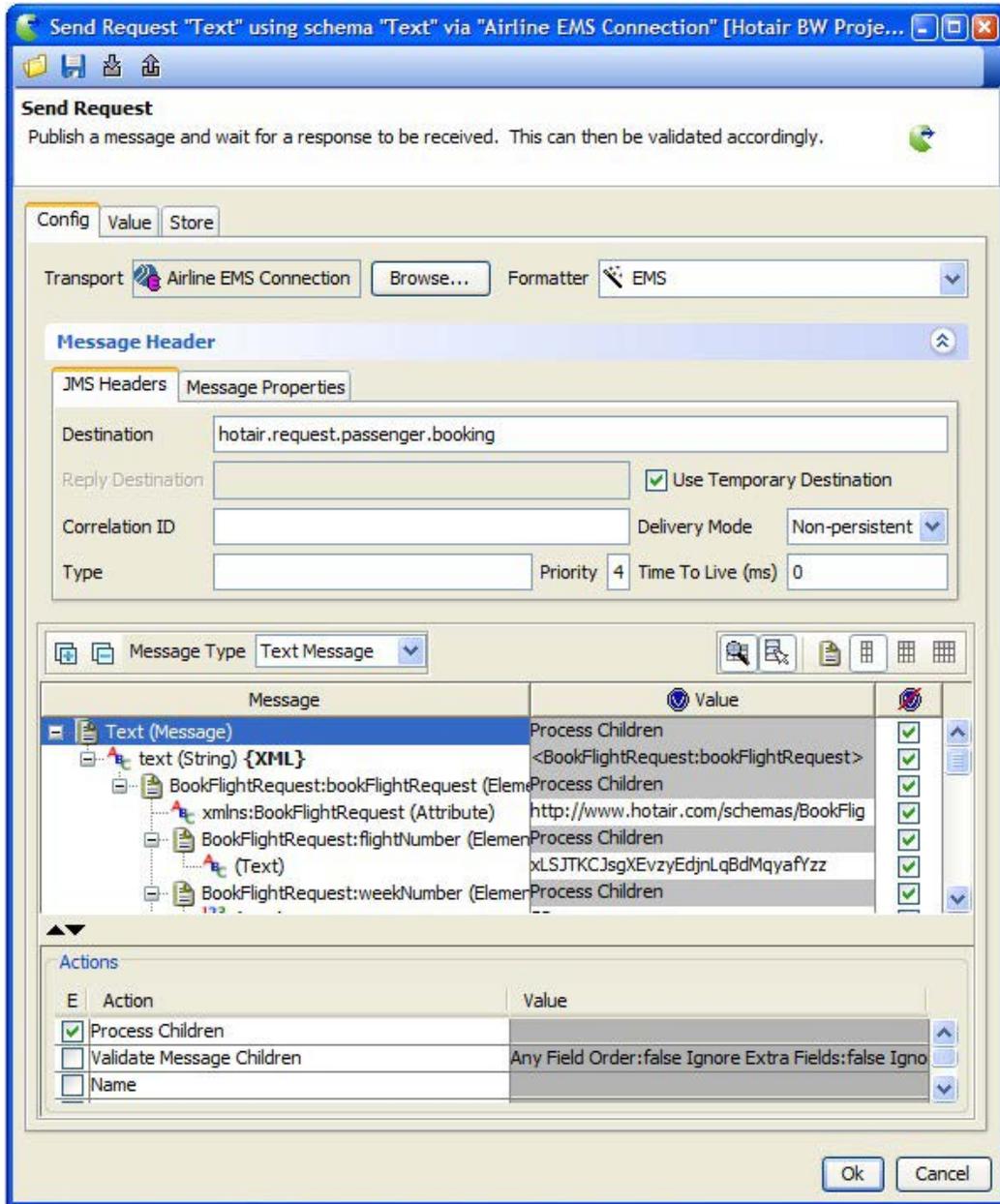


Option	Description
Use Destination...	For each request, create a unique temporary destination for receiving responses. Using this configuration, a separate reply destination is created for each published request.
Use JMSCorrelationID...	The JMSCorrelationID is assumed to be returned in the response. This message field is found within the message header and is used by applications to identify messages. You can set this string value in Rational Integration Tester under the Value tab in the field editor.
Use JMSMessageID mapped to JMSCorrelationID...	Using this option, the JMSMessageID is assumed to be returned in the response as the JMSCorrelationID (this will be handled by the application receiving and responding to the request). The JMSMessageID field uniquely identifies an individual message and is automatically generated by the sending JMS server; it is a string prefixed with 'ID:'. An example from a TIBCO EMS server is 'ID:EMS-SERVER-GHC-PC048.B184766342760:3', and it is only visible in a subscriber message or configuration.

Under **Session Settings**, you can use a single session for all subscribers that use the transport. This setting (to be used only when running performance tests) can improve throughput in the EMS server.

3.4 Sending EMS Messages

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



3.4.1 Configure JMS Headers

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

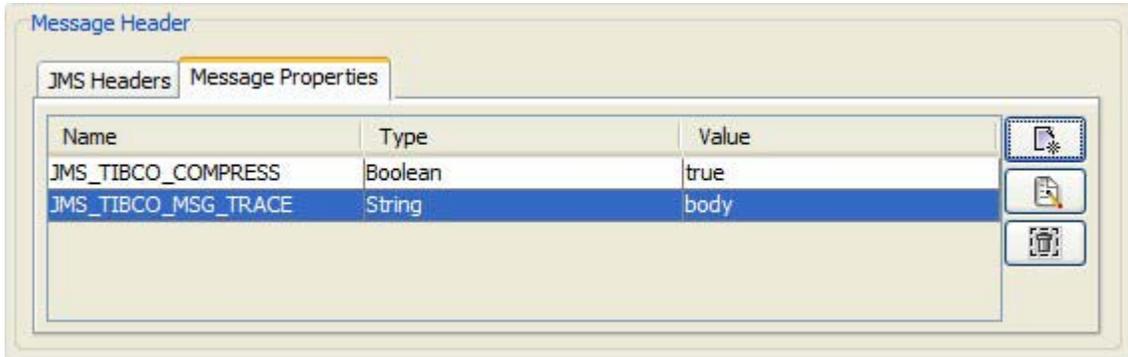
The screenshot shows a configuration window with two tabs: "JMS Headers" and "Message Properties". The "JMS Headers" tab is active. It contains several input fields and controls: "Destination" is set to "hotair.request.passenger.booking"; "Reply Destination" is empty; "Correlation ID" is empty; "Type" is empty; "Priority" is set to "4"; "Time To Live (ms)" is set to "0"; "Delivery Mode" is a dropdown menu set to "Non-persistent"; and a checkbox labeled "Use Temporary Destination" is checked.

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.

3.4.2 Configure Message Properties

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



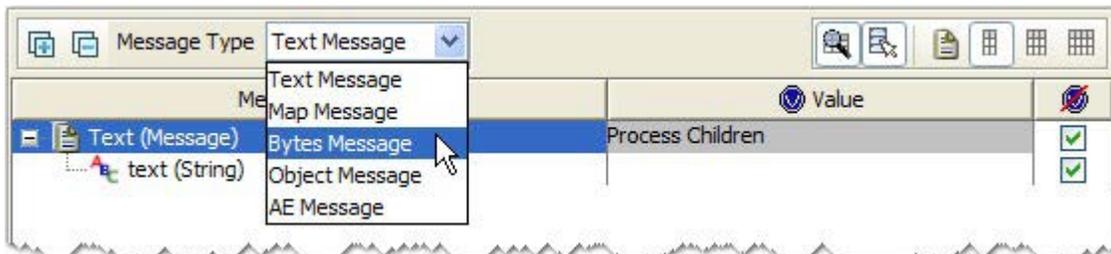
Each property has a name, type, and value.

- 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 TIBCO documentation for more information about additional message properties.

3.4.3 Configure the EMS Message Body

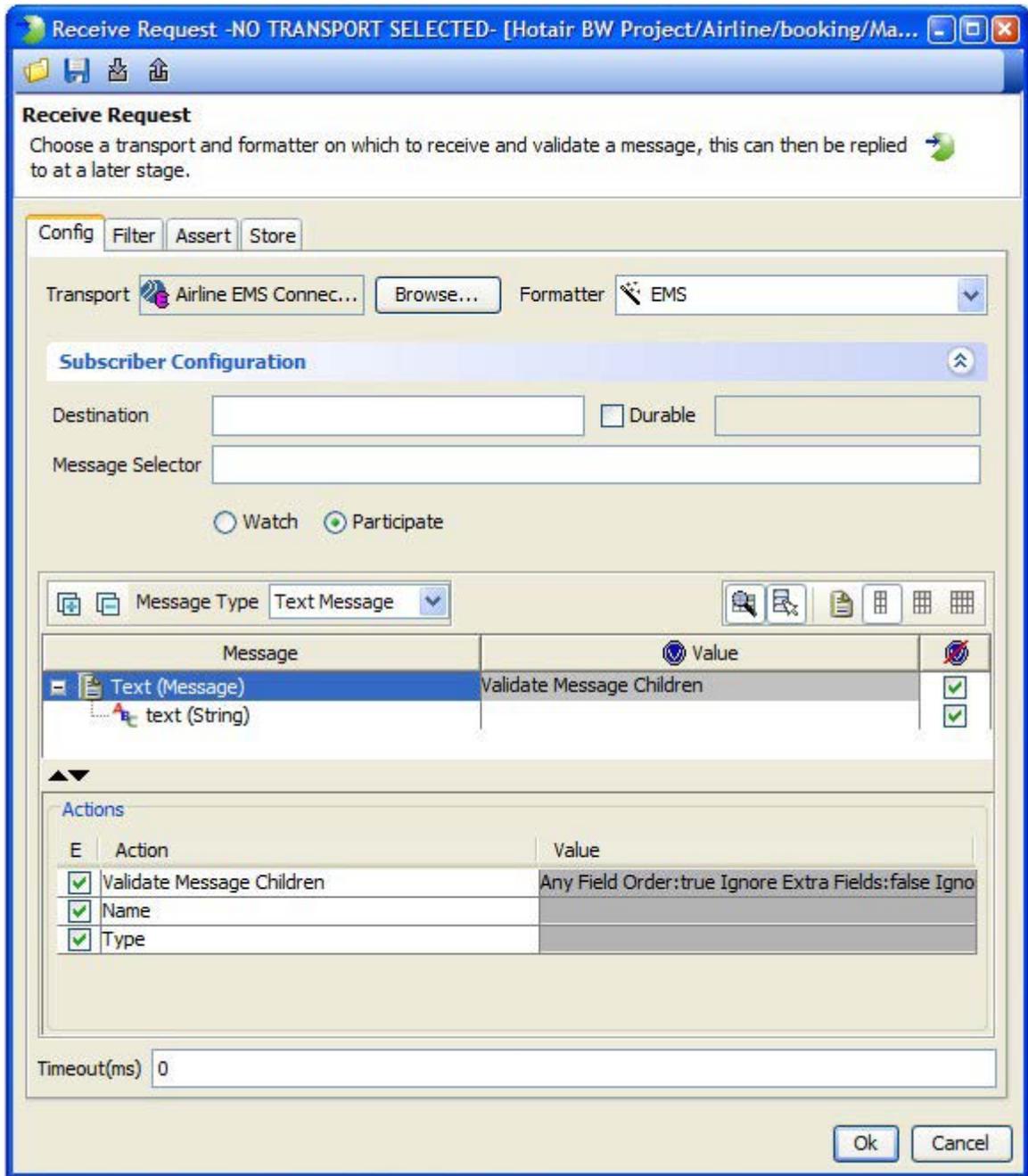
The contents and structure of an EMS message body will vary according to the message type. To set the message type, select one of the options available from the **Message Type** dropdown menu.



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

3.5 Receiving EMS Messages

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



NOTE: You can record all traffic from a TIBCO EMS server thus removing the requirement to specify in advance which server operations are to be recorded.

3.5.1 Configure Subscriber Options

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

The subscriber options are described in the following table:

Option	Description
Destination	The destination queue or topic to monitor for incoming messages.
Durable	Enable this option and enter a name to create a durable subscriber, which ensures that messages will be received even if the message consumer is not currently running.
Message Selector	Filters incoming messages according to message header properties (see Message Filtering for more information).
Watch or Participate	See Watch or Participate .

3.5.2 Watch or Participate

The EMS transport allows you to either watch or participate in queue messaging subscriptions.

NOTE: The watch/participate option is not available when using the generic JMS transport.

Participate

In **Participate** mode (default), messages are pulled off of the queue. This may be undesirable in a testing scenario since a JMS queue message may go to only one connected client. In this case, the Rational Integration Tester connection will prevent other systems from receiving the messages. If this is a problem, use the **Watch** option.

Watch

In **Watch** mode, Rational Integration Tester receives new messages without removing them from the queue (that is, you can passively access messages without affecting the queue itself). To “watch” a queue, Rational Integration Tester subscribes to system monitor topics (that is, the transport login configuration must have administrative rights on the EMS server it is watching).

NOTE: If security policies prohibit access to the server with full administrative privileges, you may need to request access to a limited number of queues (that is, only the specific queues needed for testing).

NOTE: The “Use JNDI to lookup destination” option should be disabled when in “watch” mode (see [Connection Settings](#)).

Individual Queue Permissions

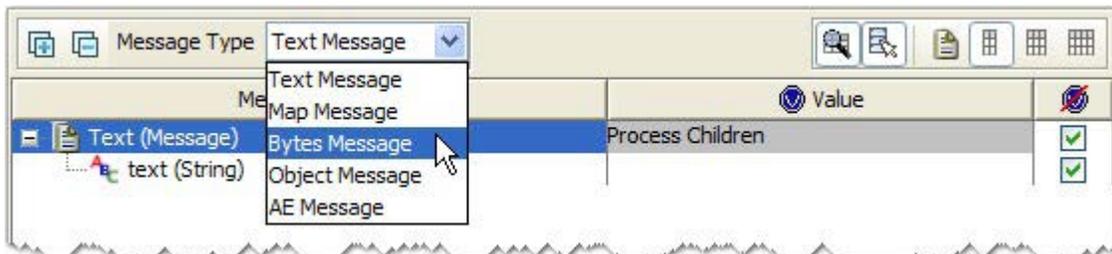
The EMS system administrator may not allow you to connect to the server with full administrative rights as this could be seen as a security risk since you would be able to access any message. If necessary, you may be able to request that permissions be added to your user account to allow access to specific queues. For example, the **tibemsadmin** command to grant permissions to watch queue.sample would be:

```
grant topic $sys.monitor.Q.*.queue.sample user=testuser
```

The wildcards ***** and **>** may be used to give permissions across a number of queues, for example: `grant topic $sys.monitor.Q.*.queue.* user=testuser`

3.5.3 Configure Message Content

The contents and structure of an EMS 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.



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

3.5.4 Message Filtering

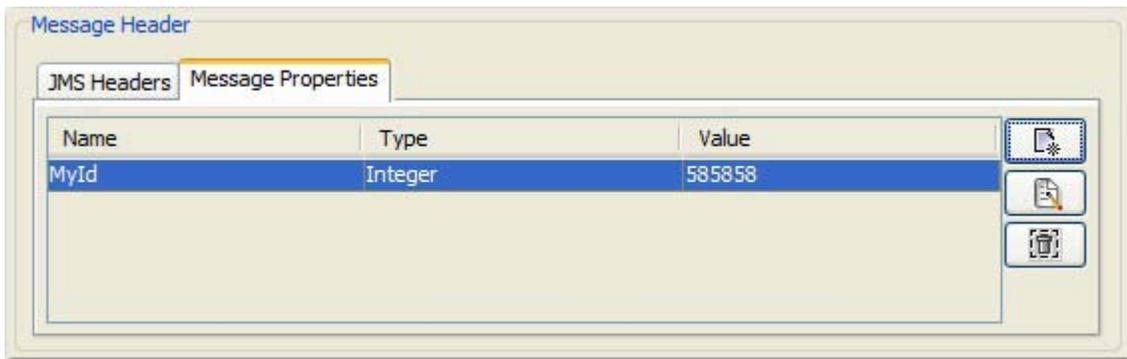
You can filter the way messages are received in two different ways: [JMS Filtering](#) and [Rational Integration Tester Filtering](#).

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** accepts standard JMS filtering expressions (message selectors). When utilized, Rational Integration Tester only receives messages that match the specified filter.

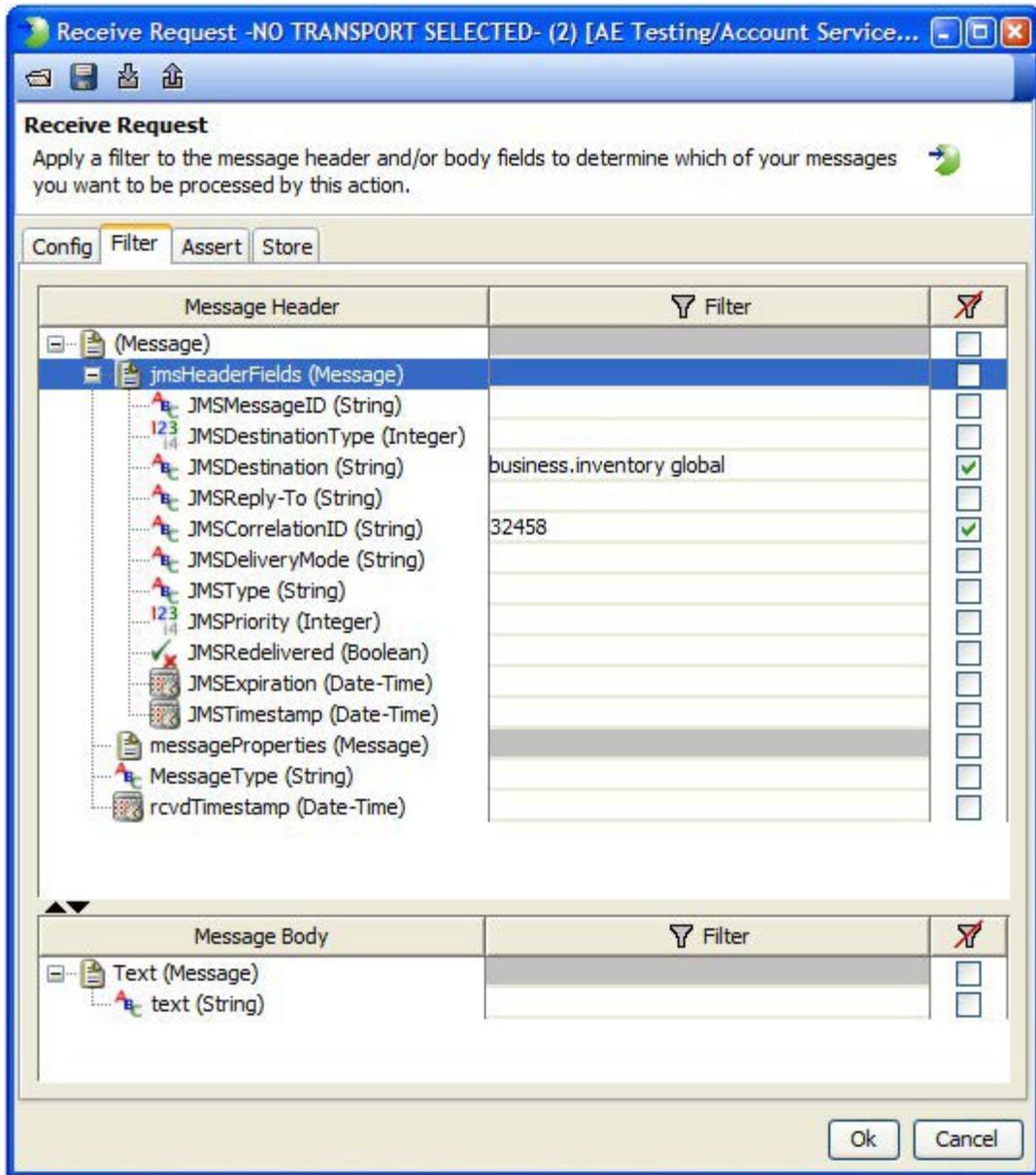
For example, if `MyId= '585858'` is entered, only messages with the header field “MyId” having a value of 585858 will be received. To send matching messages in a Rational Integration Tester publisher, you would configure a message property as shown below:



NOTE: Filtering is performed on message header properties, and only matching messages are passed to Rational Integration Tester (that is, the EMS server filters the messages, not Rational Integration Tester). This is useful in environments that share a single EMS 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.



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

3.6 Testing SSL in TIBCO EMS

This section provides an example of how to use Rational Integration Tester to send and receive messages using an EMS server over an SSL connection.

Enable SSL on the EMS Transport

1. Create a backup copy of the `factories.conf` and `tibemsd.conf` files in the `bin` directory of your EMS installation (for example, `C:\tibco\ems\5.0\bin`).
2. Ensure the contents of `factories.conf` and `tibemsd.conf` match what is shown in the files found [Configuration Files](#).
3. Restart the **Tibco EMS Server** service.

Load Required Libraries

To make use of the SSL functionality, you will need to make certain JAR files available to Rational Integration Tester by means of the Library Manager.

1. Launch the Library Manager application.
2. Select the **JMS** tab.
3. Click **Add** to create a new provider and call it “JMS SSL EMS 5.0”.
4. Select the new provider and add the following resources:
 - `tibjms.jar`
 - `tibjmsadmin.jar`
 - `tibcrypt.jar`
 - `sl4f-simple-1.4.2.jar`
 - `sl4f-api-1.4.2.jar`
5. Click **OK** to save the changes and close the Library Manager.

Configure Rational Integration Tester

1. Launch Rational Integration Tester or restart it if already running.
2. Switch to the Logical View in Architecture School (**F7**) and create a new TIBCO EMS Domain (right-click and select **New > TIBCO > TIBCO EMS Domain**).
3. Enter “ems ssl” as the EMS Domain name.
4. Switch to the Physical View and add a new TIBCO EMS Broker (right-click the **Physical** folder and select **New > TIBCO > TIBCO EMS Broker**).

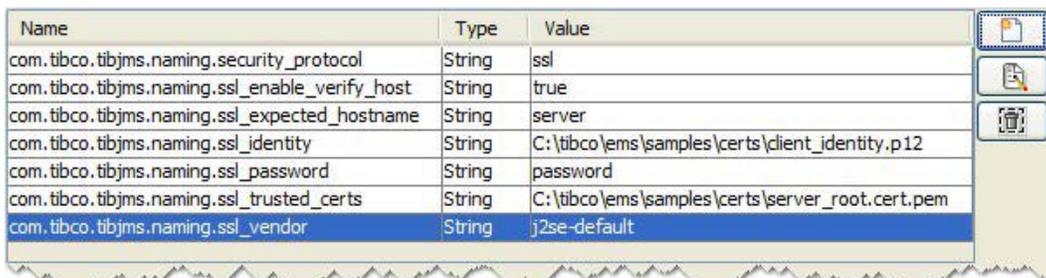
Edit the new broker (double-click) and enter the following JNDI details:

Field	Value
Initial Context Factory	com.tibco.tibjms.naming.TibjmsInitialContextFactory
Provider URLs	tibjmsnaming://localhost:7243
Username	SSHTest
Password	password

Add the following custom properties to the transport (see [Configure Message Properties](#)):

Name	Value
com.tibco.tibjms.naming.security_protocol	ssl
com.tibco.tibjms.naming.ssl_enable_verify_host	true
com.tibco.tibjms.naming.ssl_expected_hostname	server
com.tibco.tibjms.naming.ssl_identity	C:\tibco\ems\samples\certs\client_identity.p12
com.tibco.tibjms.naming.ssl_password	password
com.tibco.tibjms.naming.ssl_trusted_certs	c:\tibco\ems\samples\certs\server_root.cert.pem
com.tibco.tibjms.naming.ssl_vendor	j2se-default

The message properties should look like those shown below:



The screenshot shows a table with three columns: Name, Type, and Value. The table contains the same seven rows of properties as the table above. The last row, 'com.tibco.tibjms.naming.ssl_vendor' with value 'j2se-default', is highlighted in blue. To the right of the table are three icons: a document with a plus sign, a document with a minus sign, and a trash can.

Name	Type	Value
com.tibco.tibjms.naming.security_protocol	String	ssl
com.tibco.tibjms.naming.ssl_enable_verify_host	String	true
com.tibco.tibjms.naming.ssl_expected_hostname	String	server
com.tibco.tibjms.naming.ssl_identity	String	C:\tibco\ems\samples\certs\client_identity.p12
com.tibco.tibjms.naming.ssl_password	String	password
com.tibco.tibjms.naming.ssl_trusted_certs	String	C:\tibco\ems\samples\certs\server_root.cert.pem
com.tibco.tibjms.naming.ssl_vendor	String	j2se-default

5. Click the **Test Transport** button to ensure the configuration is valid.
6. Create a new test in the Test Factory (F10) and add a Publish action that uses the **ems ssl** transport, and ensure that the test passes successfully.

3.7 Configuration Files

The following sections illustrate the contents of the configuration files used in setting up an SSL connection for TIBCO EMS.

factories.conf

```
[GenericConnectionFactory]
  type          = generic
  url           = tcp://7222

[TopicConnectionFactory]
  type          = topic
  url           = tcp://7222

[QueueConnectionFactory]
  type          = queue
  url           = tcp://7222

[FTTopicConnectionFactory]
  type          = topic
  url           = tcp://localhost:7222,tcp://localhost:7224

[FTQueueConnectionFactory]
  type          = queue
  url           = tcp://localhost:7222,tcp://localhost:7224

[SSLQueueConnectionFactory]
  type          = queue
  url           = ssl://localhost:7243
  ssl_verify_host = true
  ssl_expected_hostname = server
  ssl_trusted     = certs\server_root.cert.pem
  ssl_identity    = certs\client_identity.p12

[SSLTopicConnectionFactory]
  type          = topic
  url           = ssl://localhost:7243
  ssl_verify_host = true
  ssl_expected_hostname = server
  ssl_trusted     = certs\server_root.cert.pem
  ssl_identity    = certs\client_identity.p12
```

tibemsd.conf

```
server                = EMS-SERVER
password              =
#####
users                 = users.conf
groups                = groups.conf
topics                = topics.conf
queues                = queues.conf
acl_list              = acl.conf
factories              = factories.conf
routes                = routes.conf
bridges                = bridges.conf
transports            = transports.conf
tibrvcm                = tibrvcm.conf
durables              = durables.conf
channels              = channels.conf
stores                 = stores.conf
#####
store                  = datastore
#####
max_connections        = 0
#####
max_msg_memory         = 512MB
msg_swapping           = enabled
#####
listen                 = tcp://7222
listen                 = ssl://localhost:7243
#####
authorization          = disabled
#####
routing                = disabled
#####
flow_control           = disabled
#####
multicast              = disabled
#####
```

```
server_rate_interval      = 1
statistics                = enabled
rate_interval            = 3
detailed_statistics      = NONE
statistics_cleanup_interval = 30
max_stat_memory          = 64MB
#####
ssl_require_client_cert = yes
#####
ssl_cert_user_specname   = CERTIFICATE_USER
#####
ssl_server_identity      = certs/server.cert.pem
ssl_server_key           = certs/server.key.pem
ssl_password             = password
#####
ssl_server_trusted       = certs/client_root.cert.pem
```

Testing TIBCO BusinessWorks and Design Time Libraries

Contents

Overview

Add a BusinessWorks Project

Add a Design Time Library

Monitor TIBCO BusinessWorks

Example: The Hotair Scenario

Testing BusinessWorks Private Processes

Rational Integration Tester enables you to synchronize with a TIBCO BusinessWorks project or Design Time Library. The synchronisation process analyses the project or project library being tested and generates all the necessary artifacts in the Rational Integration Tester project.

This chapter describes how to add a BusinessWorks project or Design Time Library to a Rational Integration Tester project, and provides information about how to produce proper BusinessWorks coverage reports.

4.1 Overview

To simplify the process of testing TIBCO BusinessWorks projects and Design Time Libraries, Rational Integration Tester provides a synchronization feature that examines the source project or library and its configuration, then creates the equivalent test artifacts in Rational Integration Tester.

In Rational Integration Tester's Architecture School, you can add a BusinessWorks project or Design Time Library as an external resource. A wizard guides you through the process and finishes with the synchronization phase. When the synchronization is complete, all of the required components, operations, dependencies, and so on will be available in your Rational Integration Tester project.

NOTE: For more information about Architecture School and synchronization, refer to *IBM Rational Integration Tester Reference Guide*.

4.2 Add a BusinessWorks Project

Follow the steps below to synchronize a TIBCO BusinessWorks 5 project with Rational Integration Tester.

1. Create a new project or open an existing one.
2. Open the Architecture School perspective (F7) and select the Logical or Synchronisation view.
3. Add a reference to the project by clicking the **Add Item** button . Alternatively, you can select **TIBCO BusinessWorks Project** from the **TIBCO** menu, or drag and drop your BusinessWorks project folder on the Logical view of Architecture School.



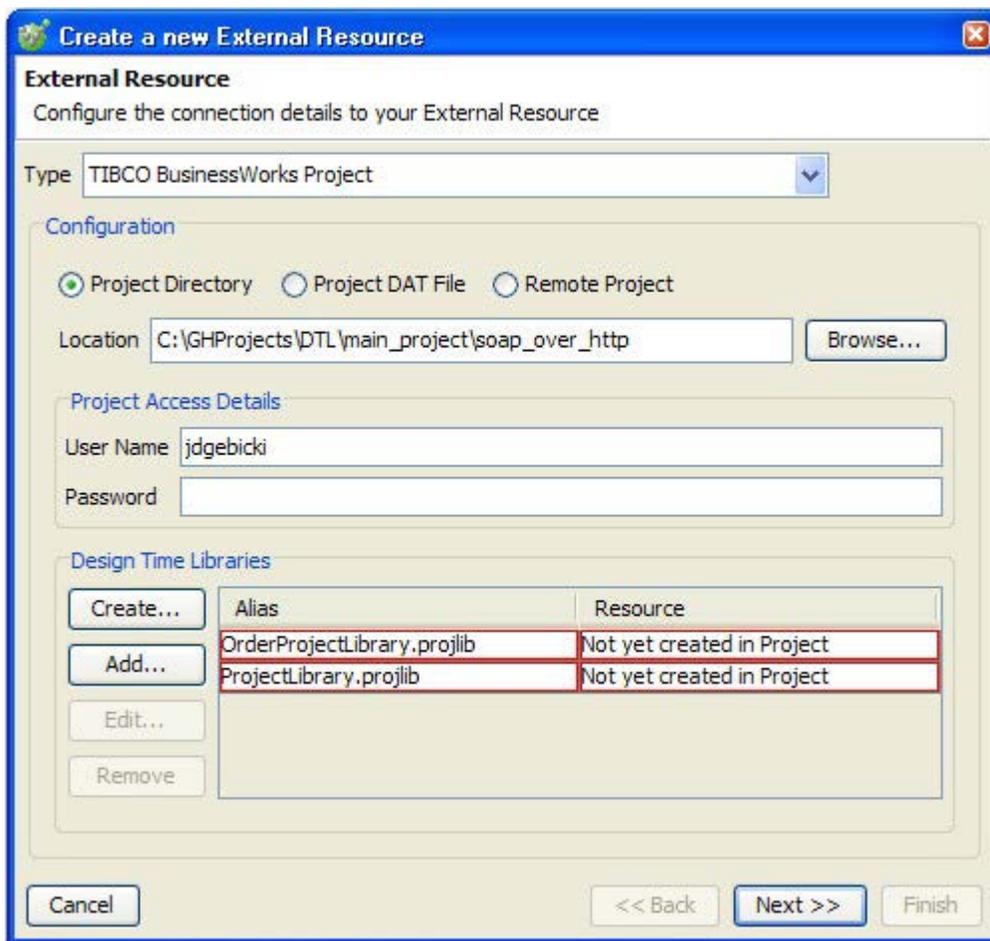
The **Create a new External Resource** wizard is displayed.

4. Ensure that **TIBCO BusinessWorks Project** is selected in the **Type** field.
5. Select the import type (**Project Directory**, **Project DAT File**, or **Remote Project**).

NOTE: If using the **Remote Project** option, please note that newer versions of TIBCO software have an option for 'Local Application Data' when creating a new Administration Domain. If this (default) option was selected during installation, TIBCO Administrator does not make projects available as 'repositories' and, as such, Rational Integration Tester will not be able to access them over the network.

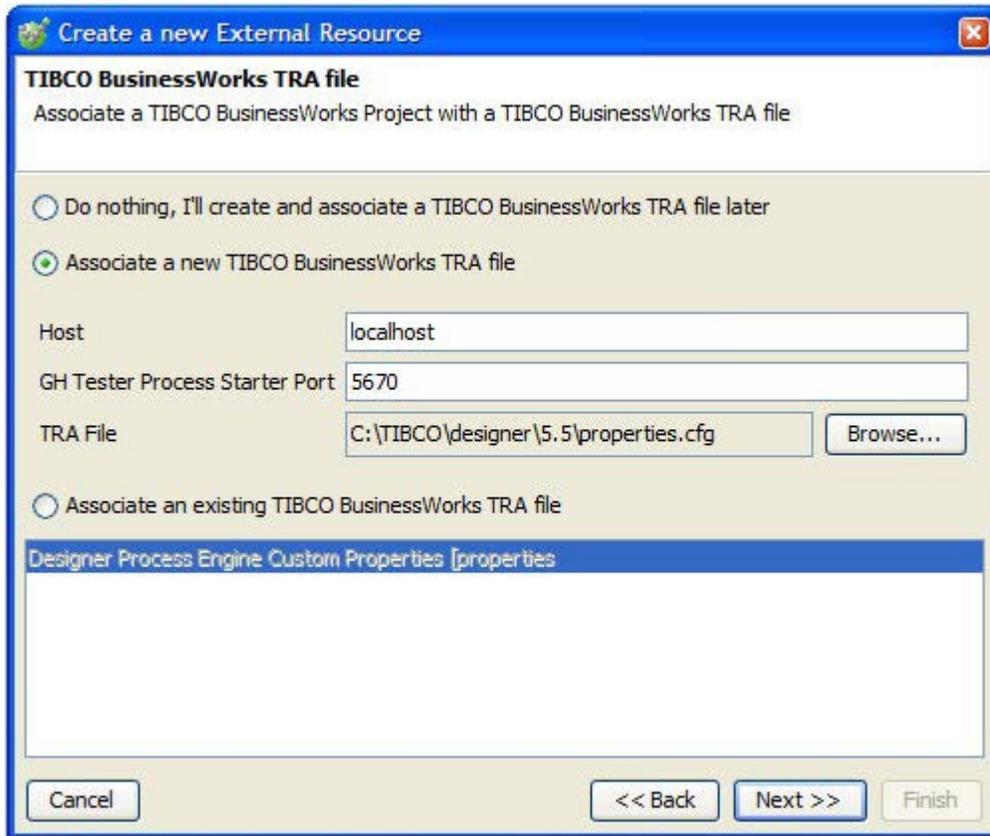
In this case you would need to use one of the other methods to connect to the project files.

-
6. Enter the full path to the project or click **Browse** to locate and select the project directory or DAT file, or to enter the details of a remote project.



7. Under Design Time Libraries, you can manage references to existing Design Time Libraries in the project, or add new libraries to the project.
 - Click **Create** to add a new library to the project or link the selected alias to a library that is not yet part of the current Rational Integration Tester project.
 - Click **Add** to add a reference to a library that already exists in the current Rational Integration Tester project.
 - Select an existing alias and click **Edit** to modify the reference to a library that is already part of the current Rational Integration Tester project.
 - To remove a reference, select an existing alias and click **Remove**.
8. When finished, click **Next** to proceed.

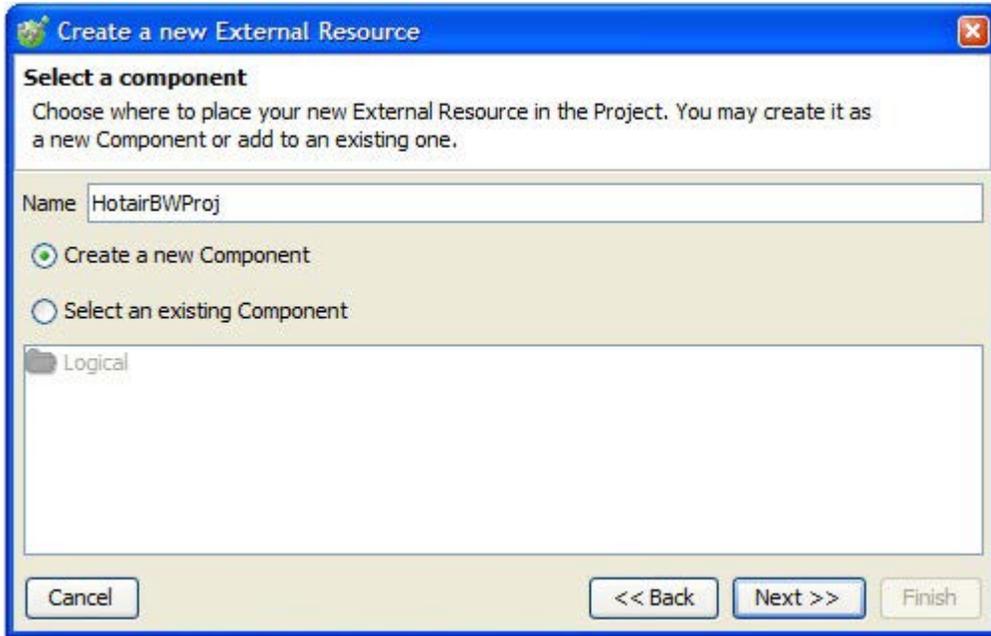
The **TIBCO BusinessWorks TRA File** dialog is displayed, which relays the location of the deployed project by means of a BusinessWorks TRA file or a local TIBCO Designer configuration file. The BusinessWorks project will be created in the Logical view and the TRA file in the Physical view, and they will be associated with one another by means of an entry in the Environment.



9. Select the second option and enter the name of the host from which the TRA file was taken (that is, the host name where the instance of TIBCO BusinessWorks has been deployed, “localhost” in this case) in the **Host** field and the port to use for Rational Integration Tester in the **Rational Integration Tester Process Starter Port** field.
10. Click **Browse** to locate and select the TRA or configuration file and click **Next**.

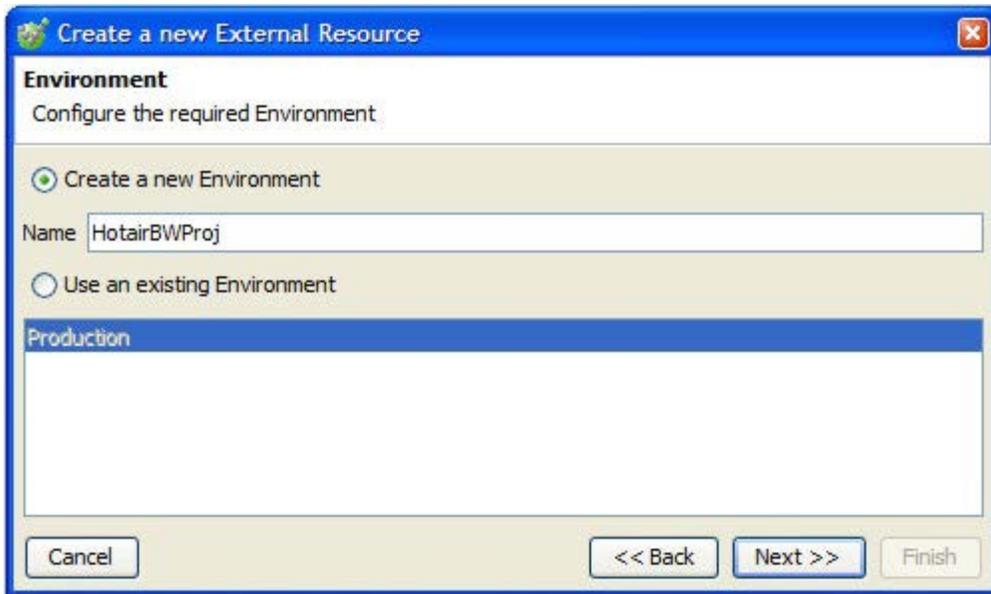
NOTE: If a configuration has already been declared in the project, then a third option would be available to select that configuration.

The **Select a Component** dialog is displayed, letting you specify where to create the BusinessWorks artifact in your project (that is, in a new or existing Service Component).



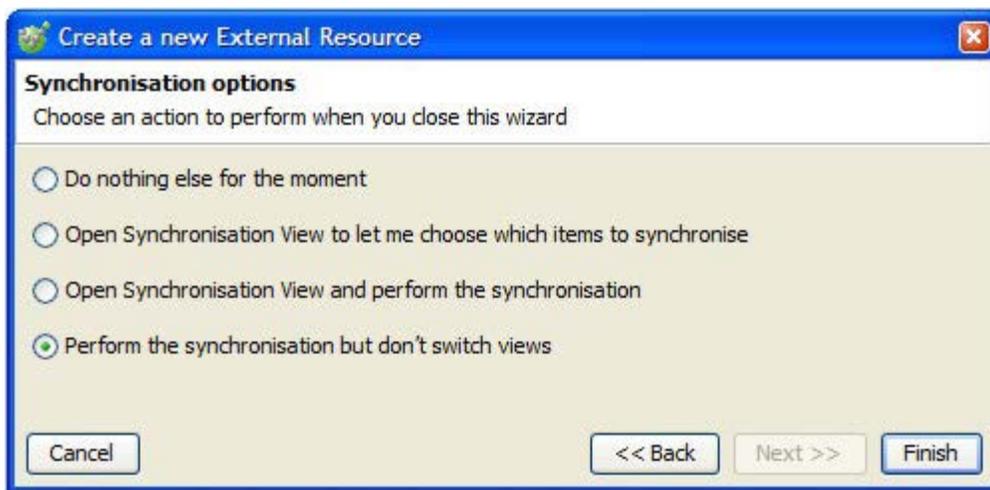
11. Select the desired option (providing a name for a new component or selecting the desired existing component), then click **Next**.

If any environments exist in the current project, the **Environment** dialog is displayed.



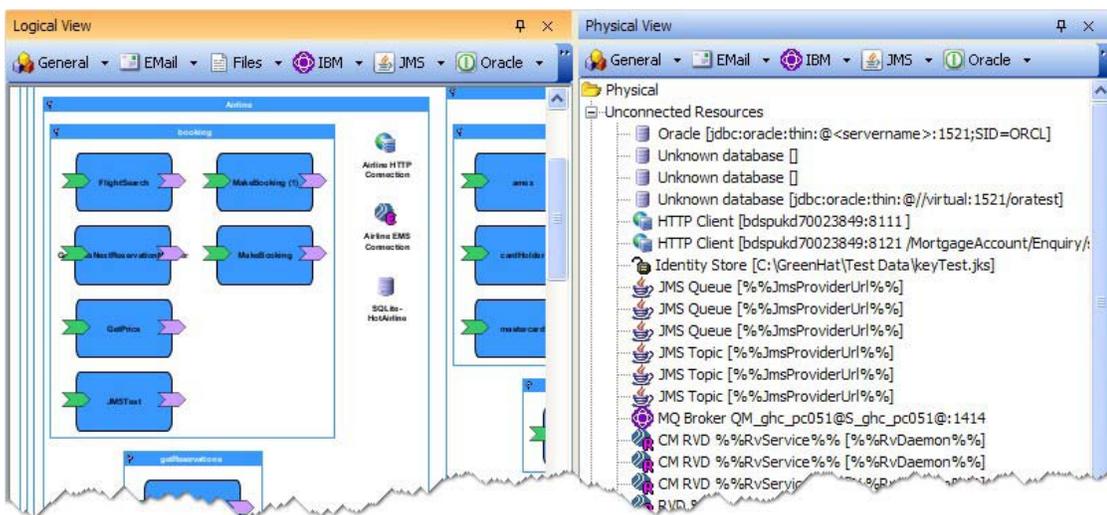
-
12. If you want to create a new environment, select the first option and enter the desired name for the new environment. Otherwise, select the second option and select the existing environment to use, then click **Next**.

The **Synchronisation Options** dialog is displayed.

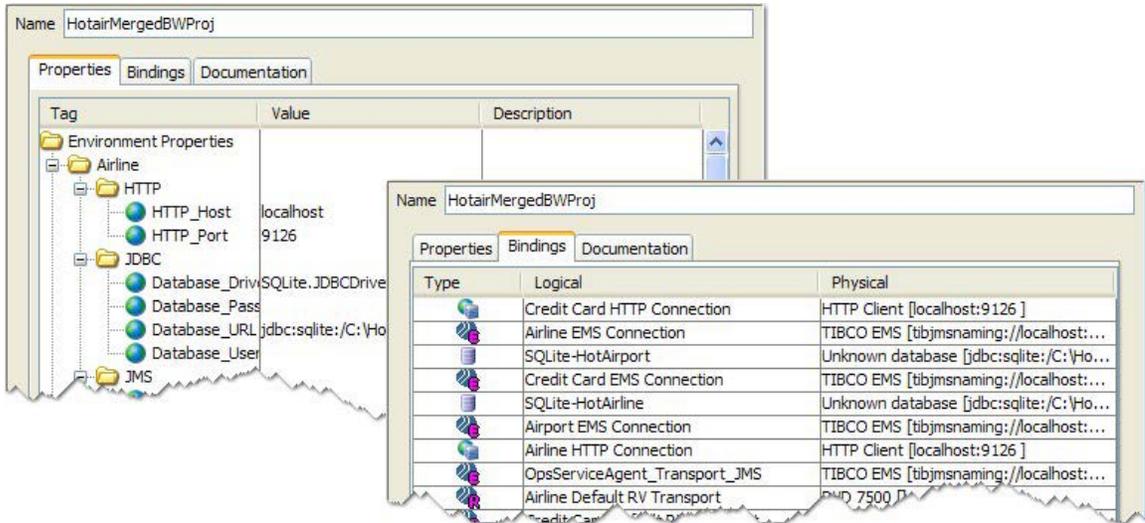


13. Select the last option (“Perform the synchronisation but don’t switch views”) and click **Finish**.

The required artifacts will be created in the project, and the results can be seen in the Logical and Physical views.



Additionally, the Environment will be populated with the required variables and bindings that have been converted from the TIBCO global variables.



NOTE: If any BusinessWorks services contain multiple endpoints, a service component containing the required operations will be created for each endpoint.

4.3 Add a Design Time Library

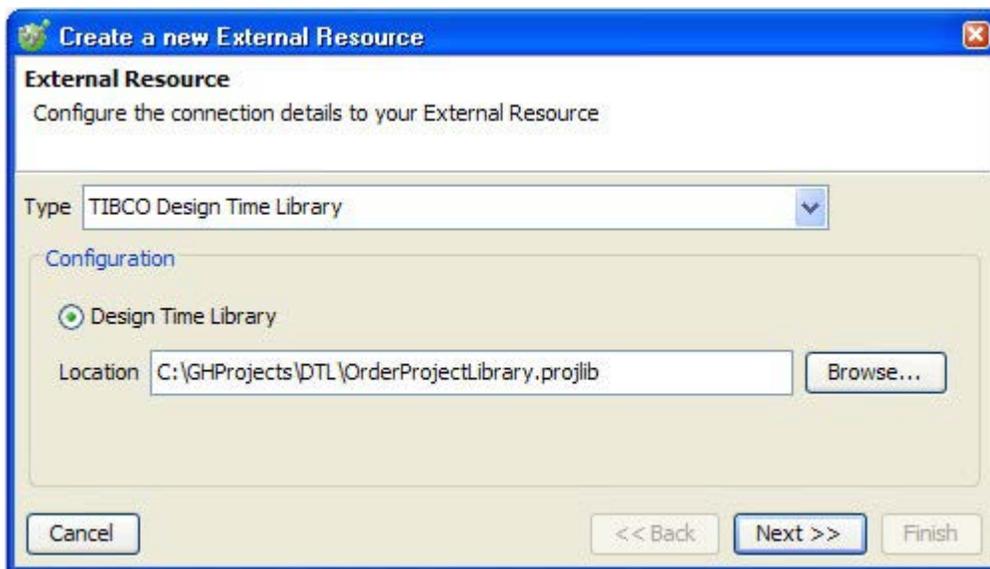
Follow the steps below to synchronize a TIBCO BusinessWorks Design Time Library with Rational Integration Tester.

1. Create a new project or open an existing one.
2. Open the Architecture School perspective (F7) and select the Logical or Synchronisation view.
3. Add a reference to the project by clicking the **Add Item** button . Alternatively, you can select **TIBCO Design Time Library** from the **TIBCO** menu, or drag and drop your project library file (*.projlib) on the Logical view of Architecture School.



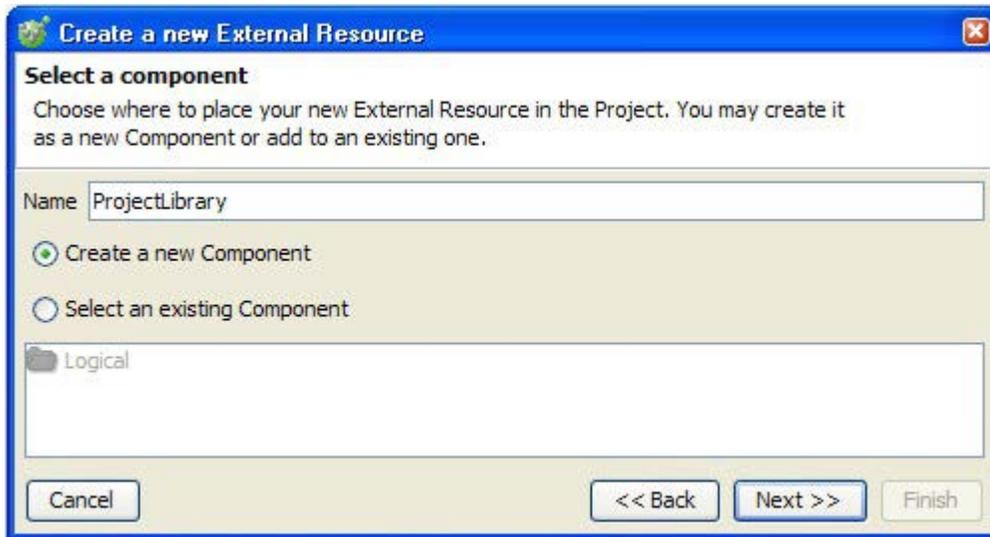
The **Create a new External Resource** wizard is displayed.

4. Ensure that **TIBCO Design Time Library** is selected in the **Type** field.
5. Enter the full path to the library or click **Browse** to locate and select it.



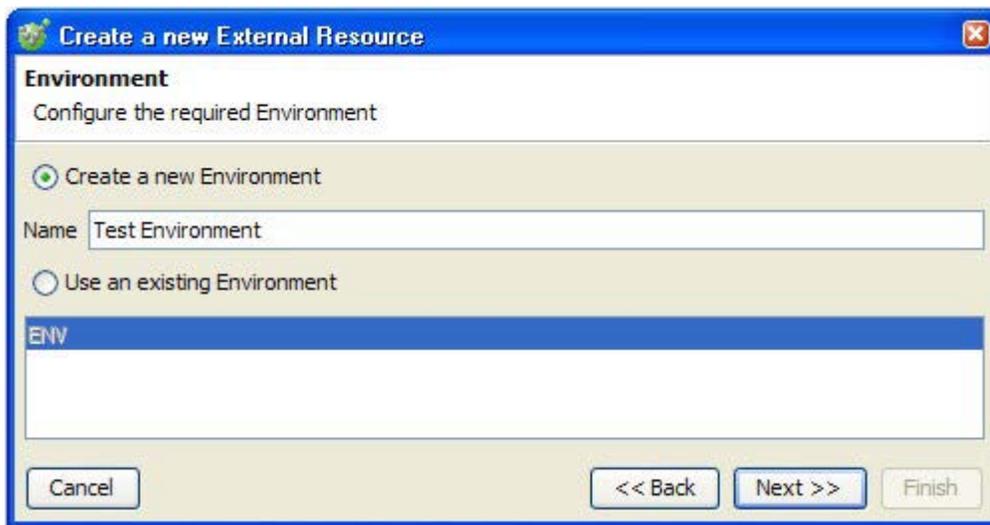
6. When finished, click **Next** to proceed.

The **Select a Component** dialog is displayed, letting you specify where to create the BusinessWorks project artefacts in your Rational Integration Tester project (that is, in a new or existing Service Component).



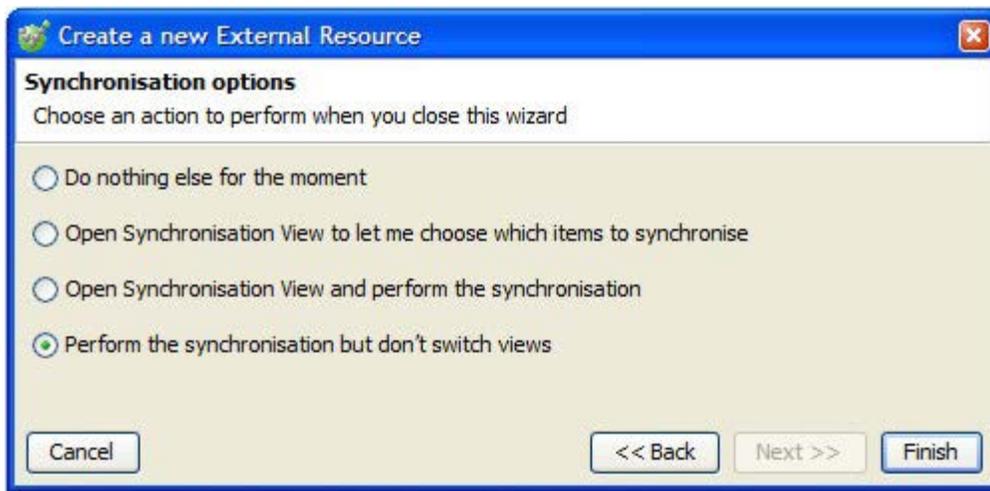
7. Select the desired option (providing a name for a new component or selecting the desired existing component), then click **Next**.

If any environments exist in the current project, the **Environment** dialog is displayed.



8. If you want to create a new environment, select the first option and enter the desired name for the new environment. Otherwise, select the second option and select the existing environment to use, then click **Next**.

The **Synchronisation Options** dialog is displayed.



9. Select the desired option and click **Finish**.

The required artifacts will be created in the project, and the results can be seen in the Logical and Physical views.

Additionally, the Environment will be populated with the required variables and bindings that have been converted from the TIBCO global variables.

4.4 Monitor TIBCO BusinessWorks

BusinessWorks (BW) is configured to publish tracing messages using a Rendezvous transport. These tracing messages record the progress of the various steps in the business processes under test. Rational Integration Tester can be configured to listen to these messages in order to produce fully detailed reports, including timings and coverage (that is, how much of the BW project has been tested).

NOTE: Since the tracing uses Rendezvous messaging, you should understand the Rendezvous configuration options or seek advice from your TIBCO administrator.

NOTE: The configuration of BW engine tracing may result in a lot of messages being published – messages will be published for all processes, even those not started by Rational Integration Tester. A multicast service can be used to avoid any unnecessary burden on the network or other downstream systems.

Please refer to the TIBCO Designer and TIBCO BusinessWorks documentation for full details of the tracing configuration.

To produce full reports, the following are required:

- You must have a TIBCO BusinessWorks project in the Logical view of Architecture School
- You must have a TIBCO BusinessWorks TRA file in the Physical View of architecture School

Rational Integration Tester will create a coverage report when a suite is run. The suite should contain a scenario where the execution environment binds the TIBCO BW project to the TIBCO BW TRA file – this happens automatically when the BW project is imported into Rational Integration Tester. A copy of the TRA file should be copied somewhere within the Rational Integration Tester project area (on disk) to make the project more portable when sharing it with other users.

Information about configuring TIBCO and Rational Integration Tester can be found in the following sections.

- [TIBCO Configuration](#)
- [Rational Integration Tester Configuration](#)

4.4.1 TIBCO Configuration

The TIBCO configuration depends on which of two distinct testing environments is in use: **Testing with TIBCO Designer** or **Testing Against a Deployed BW Engine**.

Testing with TIBCO Designer

When testing with TIBCO Designer, which starts a BW engine internally, you need to create or edit **properties.cfg** and place it in the top level of the Designer installation (for example, C:\tibco\designer\5.6). Alternatively, you can create the file in a different location and refer to it. See “Setting Custom Engine Properties for the Testing Environment” in your TIBCO BusinessWorks documentation for additional details.

The content of a sample file is shown below:

```
Trace.Task.*=true
Trace.Info.Publish=true
Trace.Warn.Publish=true
Trace.Error.Publish=true
Trace.Debug.Publish=true

Trace.Info.Publish.Subject=<mysubjectpart>.Info
Trace.Warn.Publish.Subject=<mysubjectpart>.Warn
Trace.Error.Publish.Subject=<mysubjectpart>.Error
Trace.Debug.Publish.Subject=<mysubjectpart>.Debug
```

You may find that 'Trace.Task.*=false' is present in the TRA file. If so, this would need to be commented out or set to true.

Typically you would set <mysubjectpart> to be the machine name, but the full subject can be anything valid for Rendezvous as long as it matches the Rational Integration Tester configuration.

Testing Against a Deployed BW Engine

The tracing configuration shown above (in [Testing with TIBCO Designer](#)) must be added to the deployed BW TRA, and the BW engine must be re-started. On Windows, this file might typically be found in the following location:

```
C:\tibco\tra\domain\MYDOMAIN\application\hotair\hotair_archive.tra
```

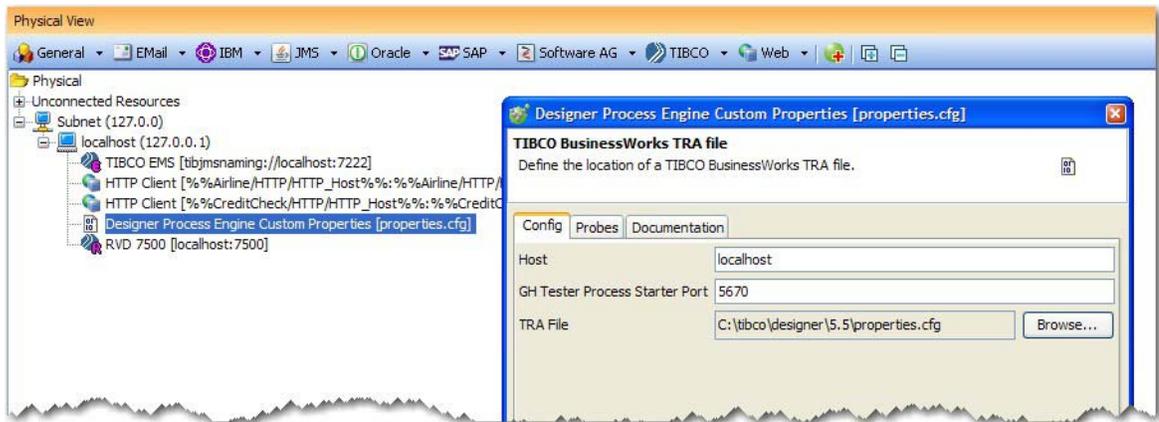
Trace messages are sent on the BW default RV transport, but it is possible to specify alternate Rendezvous transport settings, as illustrated below:

```
Trace.Info.Publish.Service=7878
Trace.Info.Publish.Network=eth0;239.1.1.1
Trace.Info.Publish.Daemon=tcp:myserver:7575
Trace.Warn.Publish.Service=7878
Trace.Warn.Publish.Network=eth0;239.1.1.1
Trace.Warn.Publish.Daemon=tcp:myserver:7575
Trace.Error.Publish.Service=7878
Trace.Error.Publish.Network=eth0;239.1.1.1
Trace.Error.Publish.Daemon=tcp:myserver:7575
Trace.Debug.Publish.Service=7878
Trace.Debug.Publish.Network=eth0;239.1.1.1
Trace.Debug.Publish.Daemon=tcp:myserver:7575
```

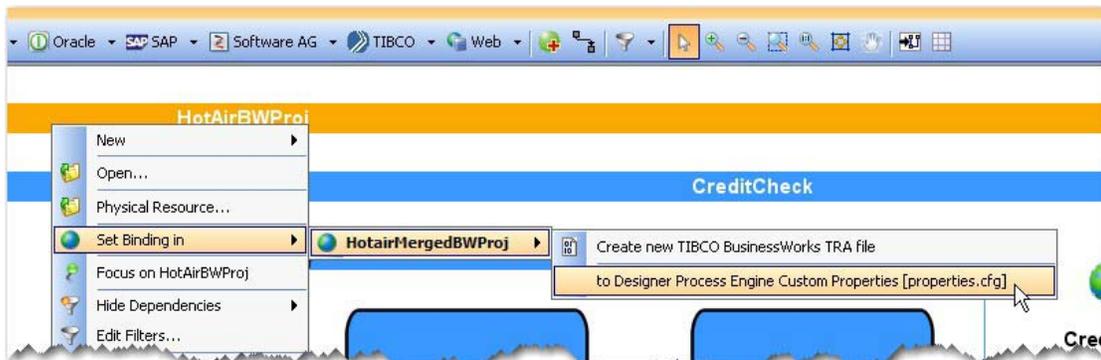
NOTE: Manual changes to the BW TRA file will be lost if the project is redeployed. It is possible to configure TIBCO Administrator to include these engine properties on each deployment by editing `bwengine.xml`. Please refer to “Setting Custom Engine Properties in Deployed Projects” in your TIBCO BusinessWorks documentation for additional details.

4.4.2 Rational Integration Tester Configuration

In the Physical View of Architecture School, ensure that you have added a TRA file (for example, `properties.cfg` from TIBCO Designer or a BW TRA file). You may have already done this when importing the BW project, but you can add one later as well.



In the Logical View of Architecture School, you can set the binding per environment to the required TRA by right-clicking as shown.



Rational Integration Tester's TRA file must be configured with the same tracing parameters in order for it to be able to generate full BW test reports. In most instances, Rational Integration Tester will be running on a different server from BW and you will need to make sure that the RV trace messages can be received by Rational Integration Tester. If the two machines are not on the same broadcast or multicast network it will be necessary to configure the Rational Integration Tester RV daemon parameter to connect to the remote RVD. You would need a setting for each type of trace message (Info/Warn/Error/Debug) but a single example follows:

```
Trace.Debug.Publish.Daemon=tcp:remoteserver:7575
```

4.5 Example: The Hotair Scenario

This section shows how Rational Integration Tester can be used to test TIBCO EMS and TIBCO BusinessWorks by means of a small but reasonably complex BusinessWorks project. The project, which is called “Hotair”, comprises a number of BusinessWorks processes that bring together several systems and make use of EMS, Rendezvous, and SOAP protocols.

Specific examples will be used to demonstrate how Rational Integration Tester can work with EMS and BusinessWorks.

This section contains the following information:

- [Scenario Overview](#)
- [Using Rational Integration Tester and Hotair to Test a TIBCO BusinessWorks Project](#)

4.5.1 Scenario Overview

The Hotair scenario features the following sub-systems:

- The Hotair booking web application (a website for airline customers), which is a J2EE webapp (JSPs) running on a web server. The application communicates with the BusinessWorks processes by means of JMS request/reply interactions with XML formatted payloads.
- The Hotair check-in web application (a website for airport check-in clerks) uses the same technology as the booking application, but with a more complex internal servlet design).
- Hotair booking processes, which provide the business logic powering the airline booking system.
- Hotair checkin processes, which provide the business logic powering the airport check-in system.
- A Flight Search Server, which is a simple Java program providing flight search functionality over a request/reply interface.
- Reservations Database, which is a database for the passenger reservations table.
- Airport Check-in database, which powers the airport check-in system. The database includes passenger details, which initially are fetched from the hotair booking system by means of a SOAP web service.

NOTE: Not all of the components described above are utilized in the examples that are illustrated in the following sections.

Queues

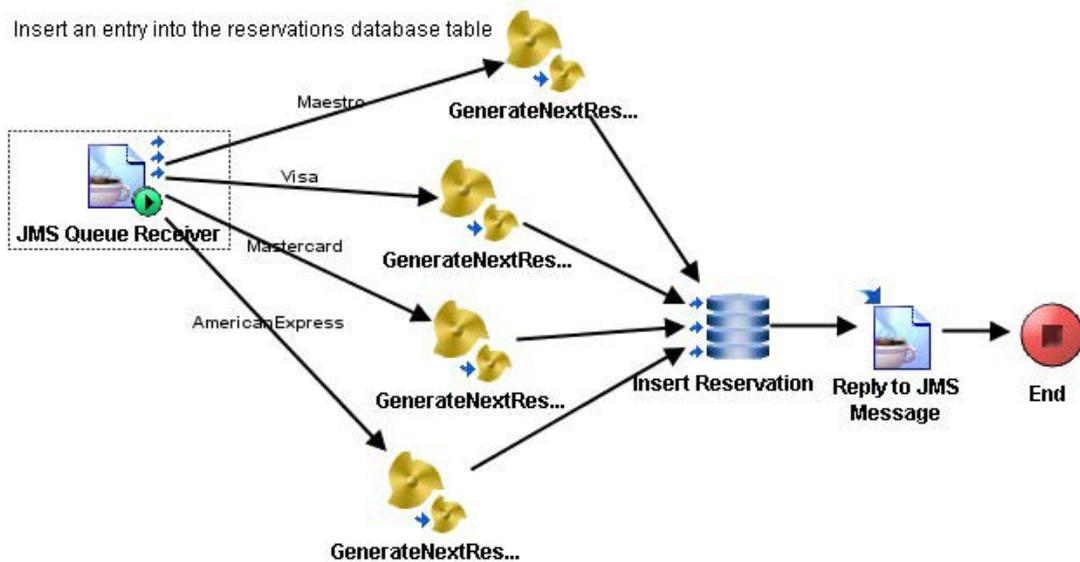
The following TIBCO EMS queues are utilized in the Hotair scenario:

- hotair.booking.simpletest
- hotair.request.passenger.booking
- hotair.request.flightsearch
- hotair.request.user.validate
- hotair.request.flight.getForCheckin
- hotair.request.passenger.getForCheckin
- hotair.request.passenger.checkin
- hotair.response

MakeBooking Process

The TIBCO BusinessWorks process “MakeBooking” is one of the most important processes in the Hotair scenario.

The following graphic shows how the MakeBooking process is displayed in TIBCO Designer.



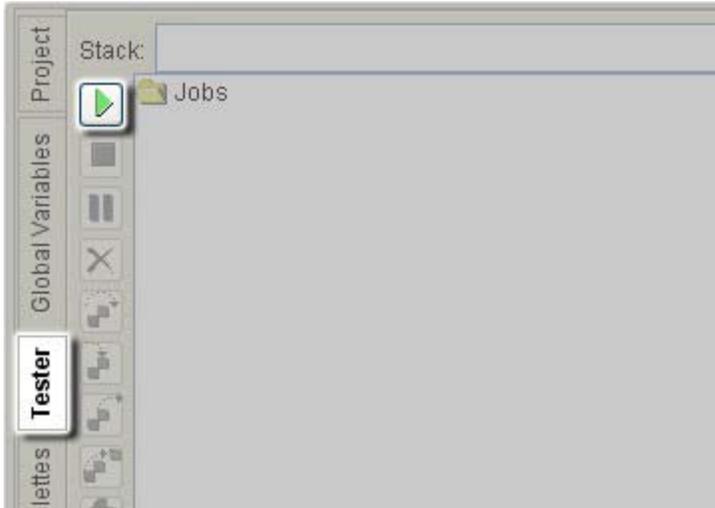
4.5.2 Using Rational Integration Tester and Hotair to Test a TIBCO BusinessWorks Project

The following sections describe how to use Rational Integration Tester and Hotair to test a TIBCO BusinessWorks project.

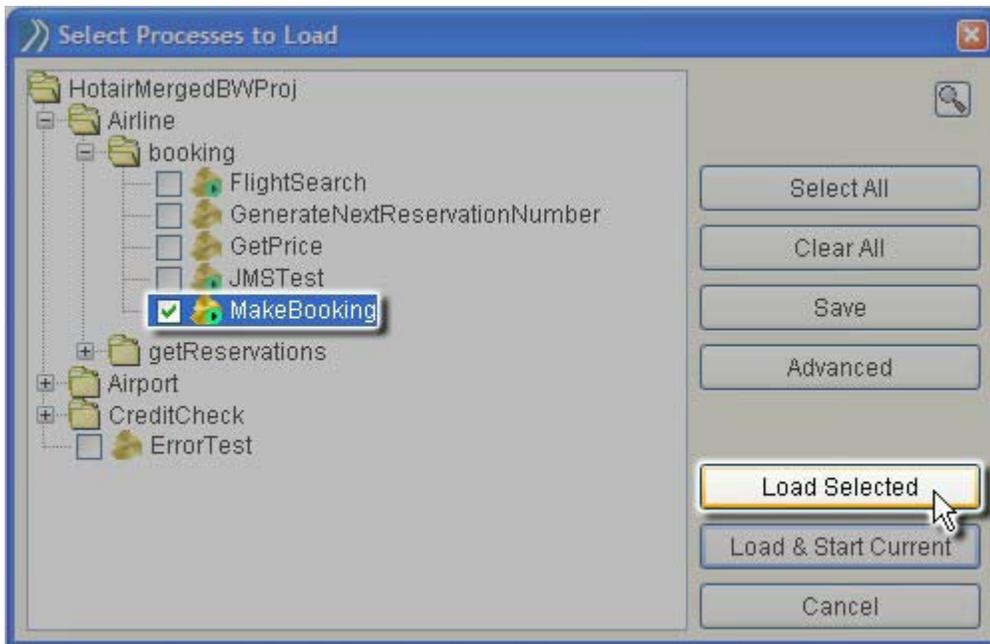
- [Load the Operation to Test in TIBCO Designer](#)
- [Build a Regression Test – Record BusinessWorks Events](#)
- [Create a Test from Recorded Events](#)
- [Create Tests from an Operation](#)
- [Create a Test Suite](#)
- [Run and Repair a Test Suite](#)
- [View the BusinessWorks Coverage Report](#)
- [Create and Run a Stub](#)
- [Create and Run a Trigger](#)

Load the Operation to Test in TIBCO Designer

1. Start the web server that hosts the Hotair booking web application.
2. Start TIBCO Designer and open the Hotair scenario project.
3. Open the **Tester** view and click the play button to test a specific process.



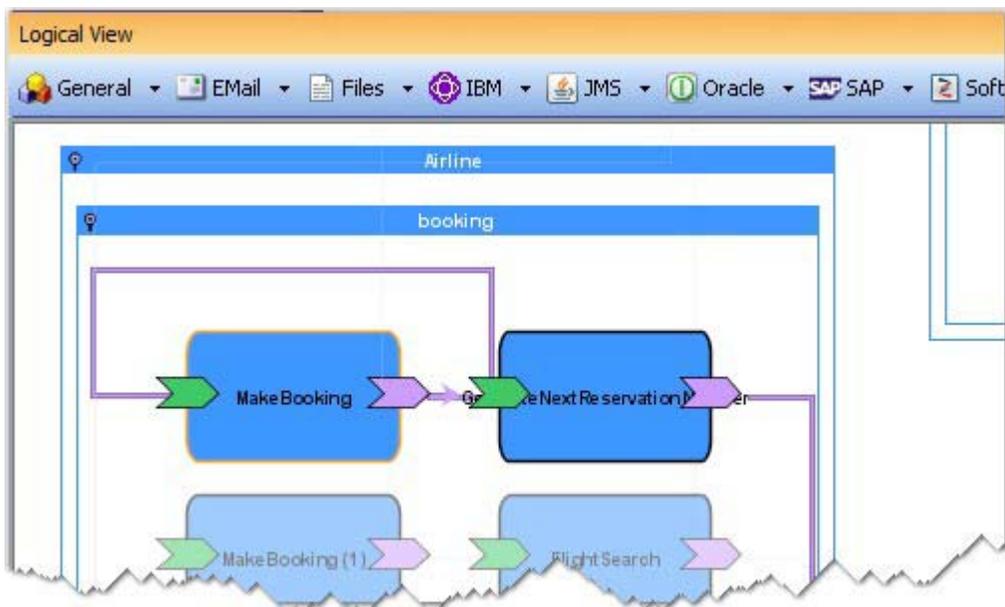
4. Load the **Airline/booking/MakeBooking** process.



Build a Regression Test – Record BusinessWorks Events

By capturing events in Rational Integration Tester, you can generate regression tests very quickly.

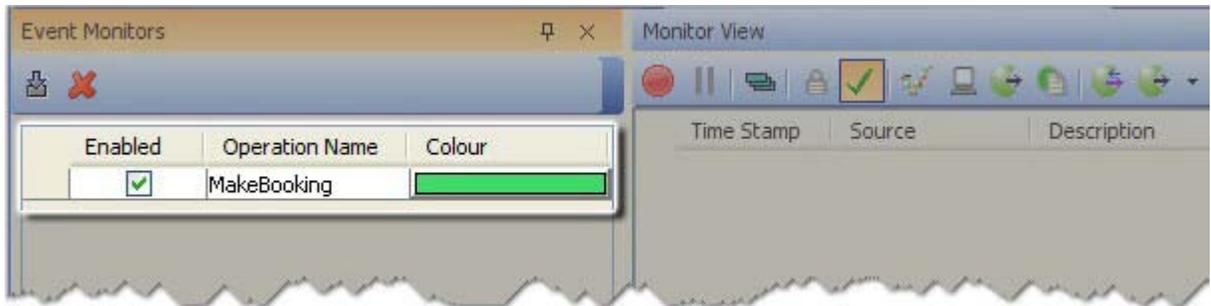
1. Launch Rational Integration Tester.
2. Create a new project in Rational Integration Tester and synchronize the BusinessWorks project (see [Add a BusinessWorks Project](#)).
3. Open the Logical View in Architecture School and select the **MakeBooking** operation.



4. Click the **Record** button in Rational Integration Tester's main toolbar.



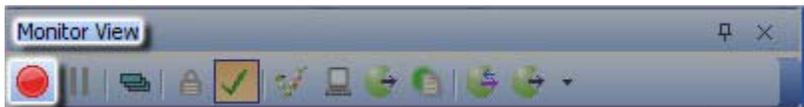
The Recording Studio is displayed, ready to record events for the **MakeBooking** operation.



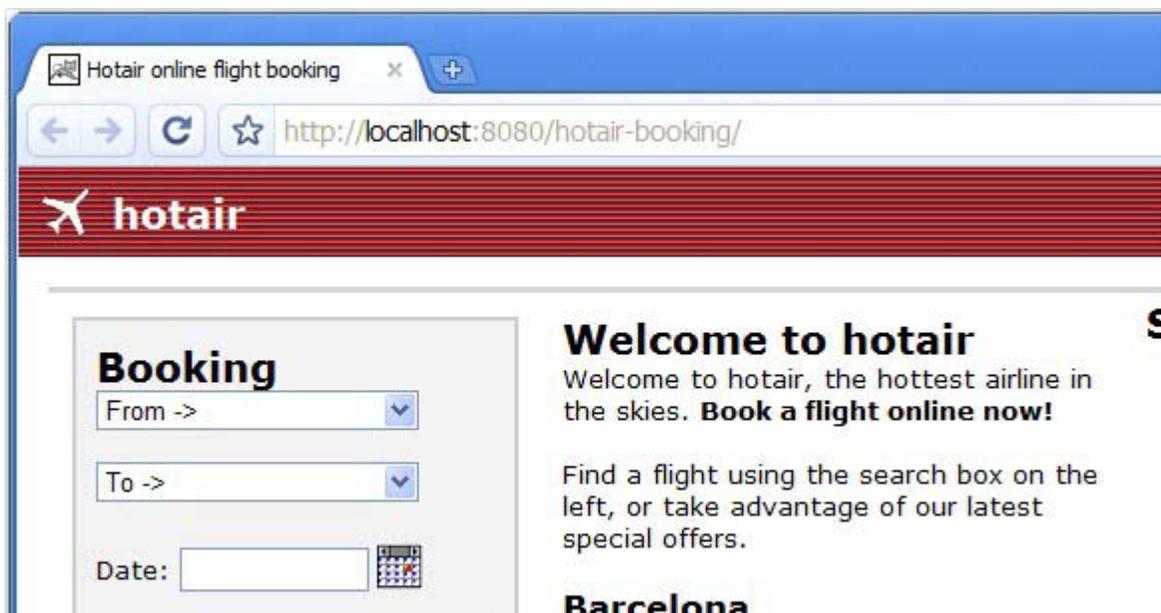
5. Select the desired environment by clicking the small arrow on the **Environments** button.

NOTE: In this example only one environment exists, so it can not be changed without creating a new environment.

6. Click the **Record** button in the Monitor View window to start recording.



7. Launch the Hotair booking site.



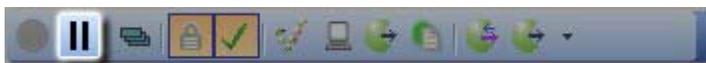
-
- Click either of the **Book Now** buttons and enter passenger/payment details on the next page.
 - Click **Proceed** when finished and you should see a confirmation page with a reservation number.



- Return to Rational Integration Tester and you should see that the request and response have been recorded in the Monitor View.

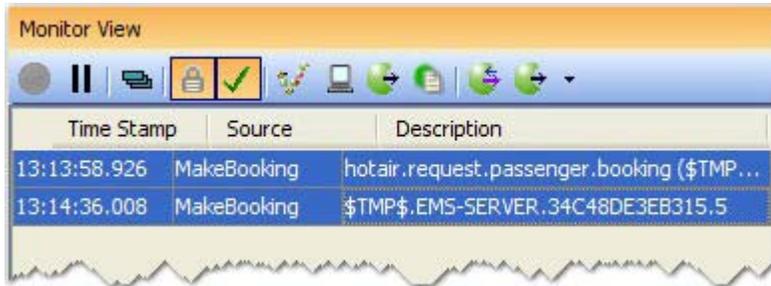


- Click **Pause** in the Monitor View to stop recording.



Create a Test from Recorded Events

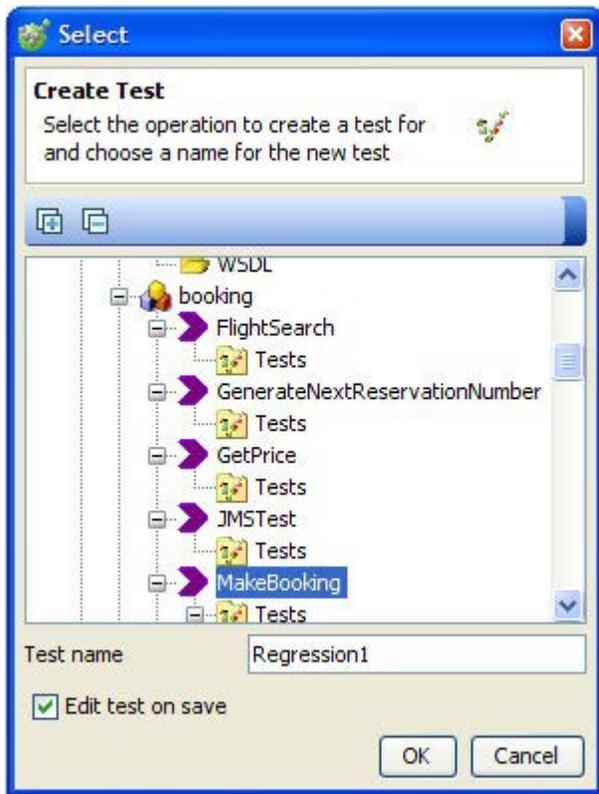
1. In the Monitor View, select both events that were recorded in the previous section ([Build a Regression Test – Record BusinessWorks Events](#)).
2. To select both events, you can click and drag while holding the left mouse button, select a range of events using the mouse and the **Shift** key, or select one event and press **Ctrl + A** (to select all). Selected events are highlighted in bright blue, as shown below.



3. Click the **Create Test** button in the Monitor View.

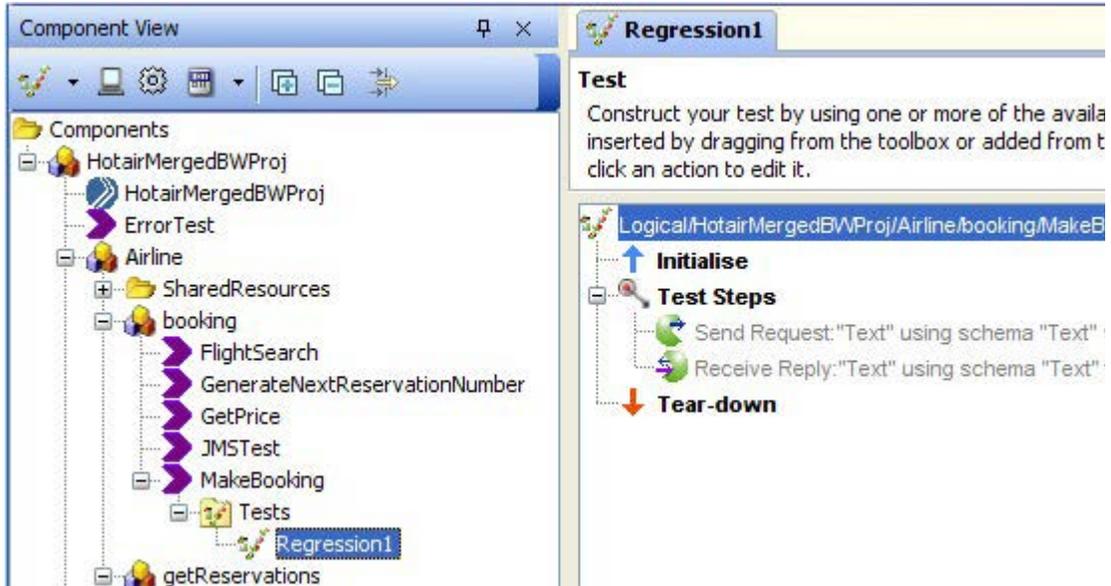


-
4. Select the operation for which you are creating the test (for example, MakeBooking) and provide a name for the new test, then click **OK**.



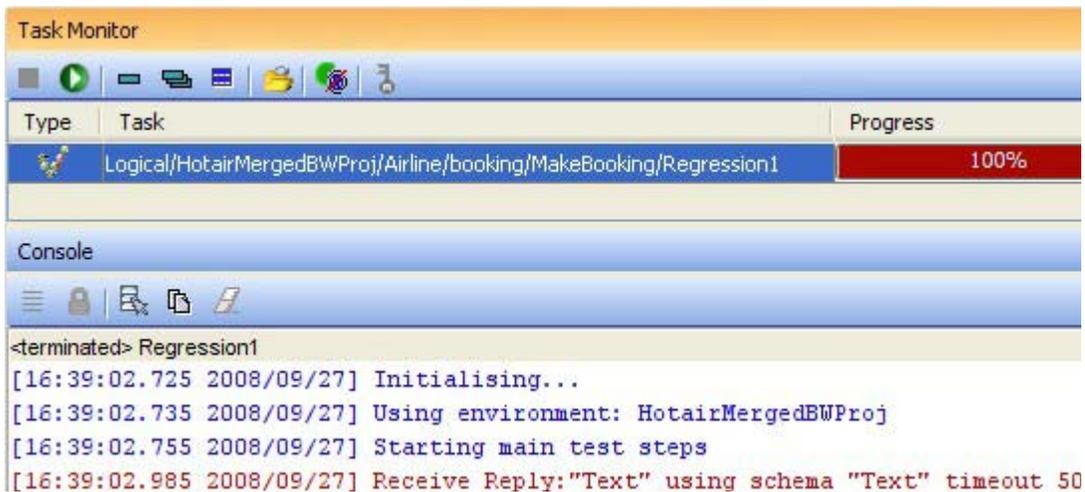
NOTE: For the sake of simplicity, enable the **Edit test on save** option to open the test in the Test Factory after saving.

In the Test Factory, you can see that the new test is created in a directory called **Tests**, contained by the operation that was selected when creating the test. The panel to the right displays the new test and the steps it contains.



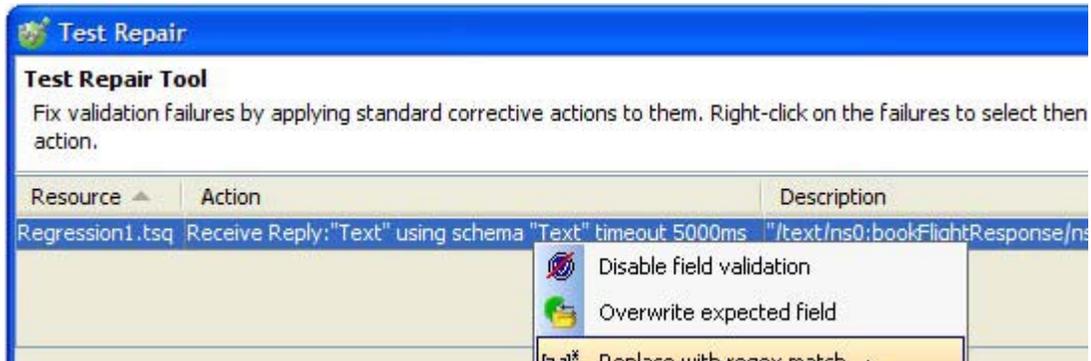
5. Run the test by right-clicking on it and selecting **Run** from the context menu, or select the test and click the **Run** icon  in Rational Integration Tester's main toolbar.

The test is opened and executed in the Test Lab. A graphical summary of the execution is displayed in the Task Monitor, and a detailed report of the test results can be found in the Console, in the lower portion of the window. In this case, the test should fail because the reservation number does not match what is expected.



-
6. Select the test in the Task Monitor and click the **Test Repair** icon  in the toolbar.

The Test Repair tool lists all validation failures that were found in the selected tests. You can right-click on any of the failures and apply an action to fix it. In this case, right-click the failure that is listed and select **Replace with regex match**, which will generate a regular expression to use in place of the differing message content.



If desired, you can right-click the failure entry in the Console and apply the same fix.

7. Select the test and click the **Run** icon  in the Task Monitor to run the test again.

This time, since the validation failure has been replaced, the test will pass.

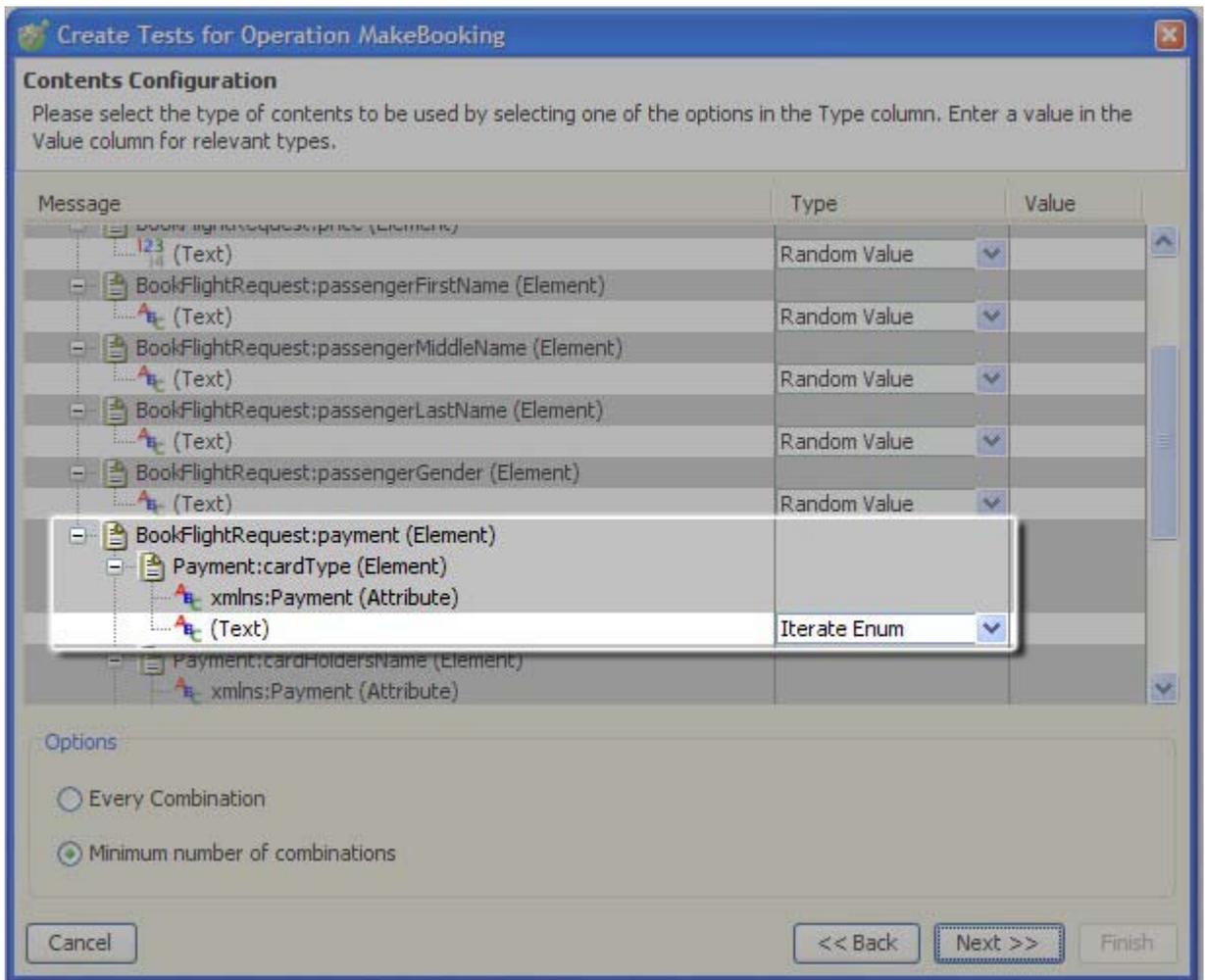
Create Tests from an Operation

1. Open the Test Factory perspective.
2. Right-click on the Tests virtual folder under the MakeBooking operation and select **New > Tests using MEP...**

A wizard is launched that will help you create multiple tests to validate all possibilities from the operation's message exchange pattern (MEP).

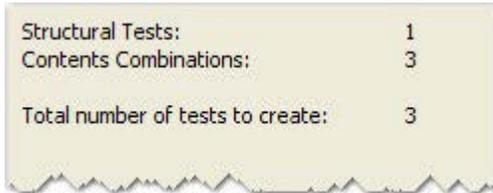
For the purpose of this example, you can leave the default settings in the first page of the wizard, This page lets you control the structure of the message by selecting how many occurrences of each element should be included.

In the wizard's second page, you can control the contents of the message by selecting different content types for each element and values for those elements that do not have random content generated for them.



-
3. Ensure that **Iterate Enum** is selected for the “Payment:cardType” element.

This will cause the wizard to generate a unique test for each type of credit card that the operation can process (three in this case), which can be verified in the next page of the wizard (Summary).

A summary window with a light beige background and a decorative, jagged bottom edge. It contains three lines of text: 'Structural Tests: 1', 'Contents Combinations: 3', and 'Total number of tests to create: 3'.

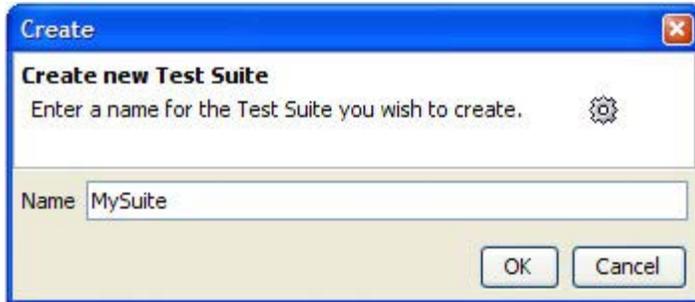
Structural Tests:	1
Contents Combinations:	3
Total number of tests to create:	3

4. Click **Finish** to complete the operation and close the wizard.

The tests that were specified by the wizard will be created in the **Tests** virtual folder under the MakeBooking operation.

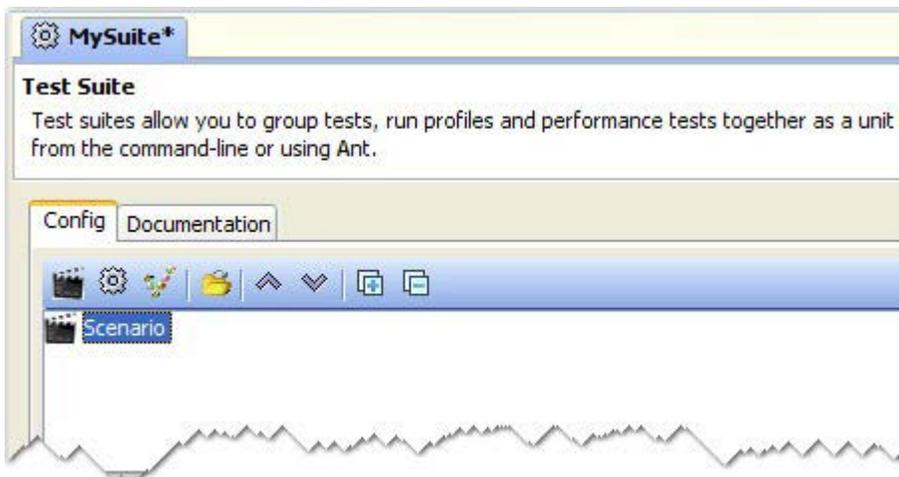
Create a Test Suite

1. Create a test suite by right-clicking on the MakeBooking operation and selecting **New > Tests > Test Suite**.



2. Provide a name for the suite and click **OK** to proceed.

The new suite appears in a new scenario, which is opened in its own tab in the Test Factory.



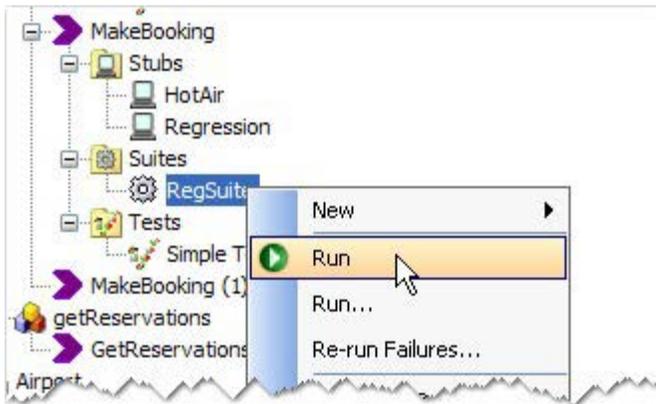
3. Add the tests that were just created by dragging them from the Tests virtual folder under MakeBooking and dropping them under the new scenario, or click the **Add Tests** button on the toolbar above the new scenario and select the tests that were just created.



4. Save the new suite by selecting **File > Save Resource**, or click the **Save** button in Rational Integration Tester's main toolbar.

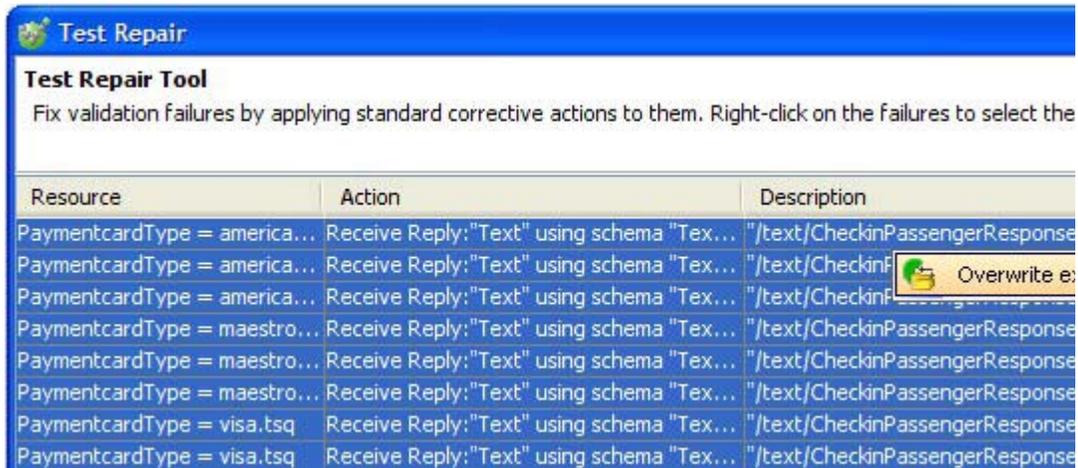
Run and Repair a Test Suite

1. Select the suite you created in the **Suites** virtual folder under the MakeBooking operation.
2. Run the suite by clicking **Run** in Rational Integration Tester's main toolbar or by right-clicking the suite and selecting **Run** from the context menu.



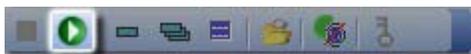
The suite is opened and executed in the Test Lab. For this first run, the suite should fail because the reply messages contain fields that are not expected.

3. Use the Test Repair tool to overwrite the expected message, ensuring that the same errors will not cause the tests in the suite to fail. Select the failed tests in the Task Monitor and click the **Test Repair** icon  in the Task Monitor toolbar.
4. Select all of the failures, then right-click one of them and select **Overwrite expected message**.



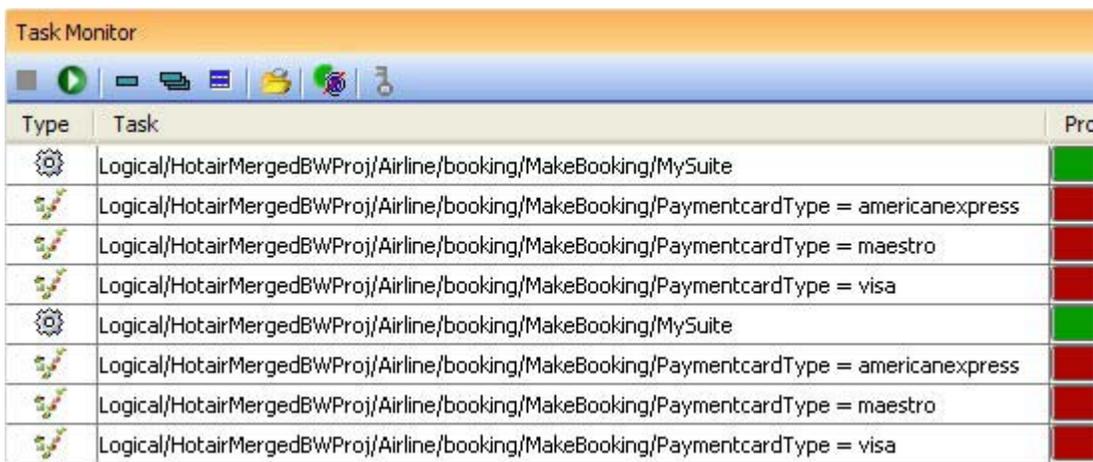
-
5. Click **Close** when finished.

To run the suite after using the repair tool, select the suite and click **Run** in the Task Monitor.



The tests should fail again, but for a different reason. Instead of the unexpected fields in the reply message, the reservation numbers that are returned do not match what is expected. These can be fixed with a regular expression match.

6. Select the failed test entries in the Task Monitor and repair them as before, except this time you will select the **Replace with regex match** option.
7. Run the suite from the Task Monitor again and this time the tests will pass.



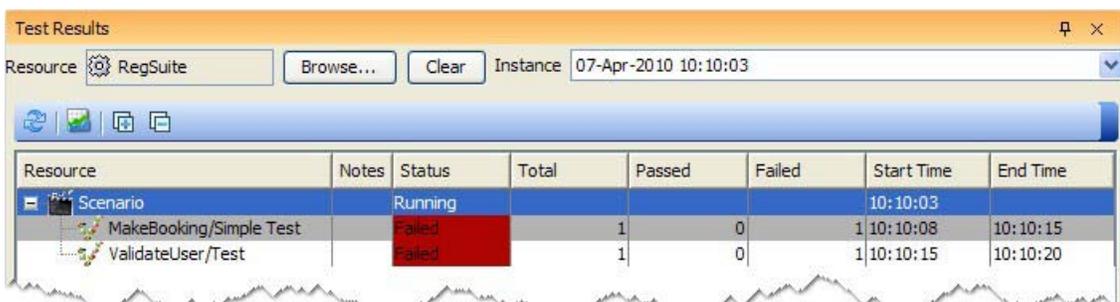
Type	Task	Pro
	Logical/HotairMergedBWProj/Airline/booking/MakeBooking/MySuite	
	Logical/HotairMergedBWProj/Airline/booking/MakeBooking/PaymentcardType = americanexpress	
	Logical/HotairMergedBWProj/Airline/booking/MakeBooking/PaymentcardType = maestro	
	Logical/HotairMergedBWProj/Airline/booking/MakeBooking/PaymentcardType = visa	
	Logical/HotairMergedBWProj/Airline/booking/MakeBooking/MySuite	
	Logical/HotairMergedBWProj/Airline/booking/MakeBooking/PaymentcardType = americanexpress	
	Logical/HotairMergedBWProj/Airline/booking/MakeBooking/PaymentcardType = maestro	
	Logical/HotairMergedBWProj/Airline/booking/MakeBooking/PaymentcardType = visa	

View the BusinessWorks Coverage Report

The Results Gallery contains all of the historical data for the test suites in your project, letting you view the results of any suite and any instance of that suite.

NOTE: For more information about available TIBCO BusinessWorks reports, refer to *IBM Rational Integration Tester Reference Guide*.

1. Open the Results Gallery.
2. Click **Browse** to select the desired test resource, then select the desired test instance.



The scenario is selected by default, but you can select any of the individual tests it contains to view applicable reports.

3. Select the **TIBCO BusinessWorks Coverage Report** tab.

The screenshot shows the "TIBCO BusinessWorks Coverage Report" tab selected in a window. The report title is "TIBCO BusinessWorks Coverage Report" and the section is "Summary". The table below shows the coverage for various processes:

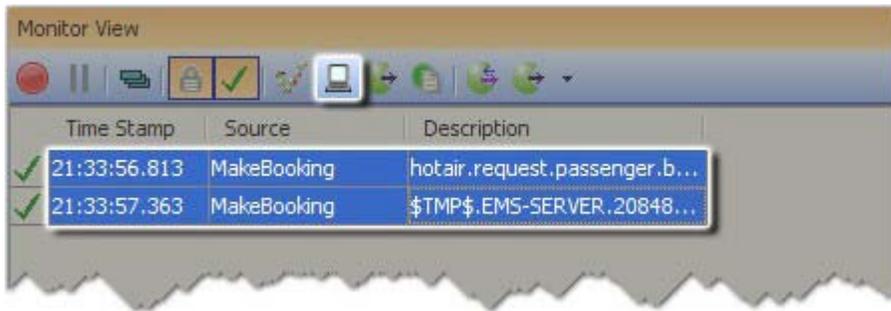
Process	Activities Run	Activities Missed	Coverage
Airline/booking/FlightSearch	0	8	0%
Airline/booking/GenerateNextReservationNumber	4	0	100%
Airline/booking/GetPrice	0	7	0%
Airline/booking/JMSTest	0	4	0%
Airline/booking/MakeBooking	7	1	88%

NOTE: Viewing the report relies upon the proper selection/configuration of the TRA file in the project (see [Add a BusinessWorks Project](#)).

Create and Run a Stub

A stub can be used to simulate a system or process that is unavailable. In the example below, we will create a stub from the request/reply events that were recorded earlier. This stub can then be used to simulate the BusinessWorks process that has been turned off.

1. Open the Recording Studio.
2. Select the previously recorded events in the Monitor View and click the **Generate Stub** button.



3. Provide a name for the new stub and click **OK**. The stub is created and displayed in the Test Factory.

NOTE: The operation under which you are creating the stub should already be selected, based on the events that were selected. If not, however, ensure that you select the proper operation.

4. Close TIBCO Designer (that is, stop the MakeBooking process).
5. Run the stub – right-click on it in the Component View and select **Run** from the context menu.

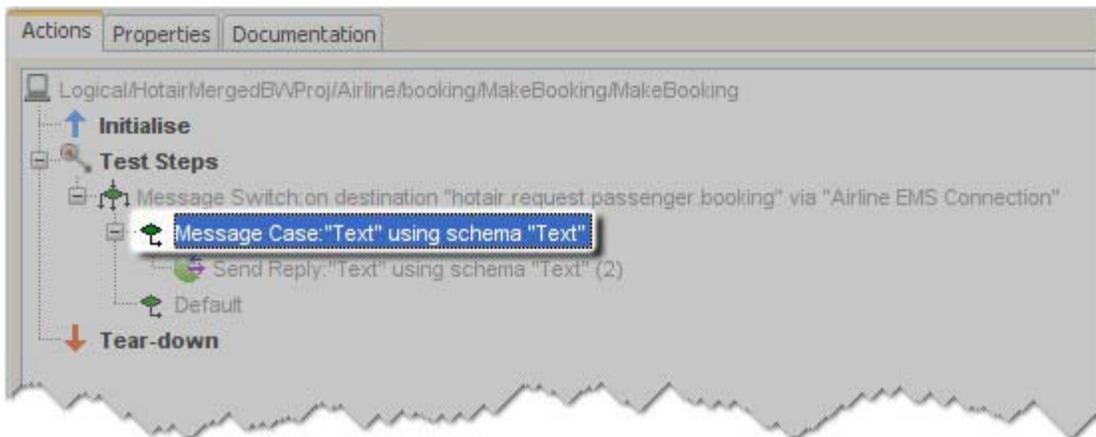
The Test Lab is displayed with the selected stub running.

6. Select the test suite in the Component View and select **Run** from the context menu.

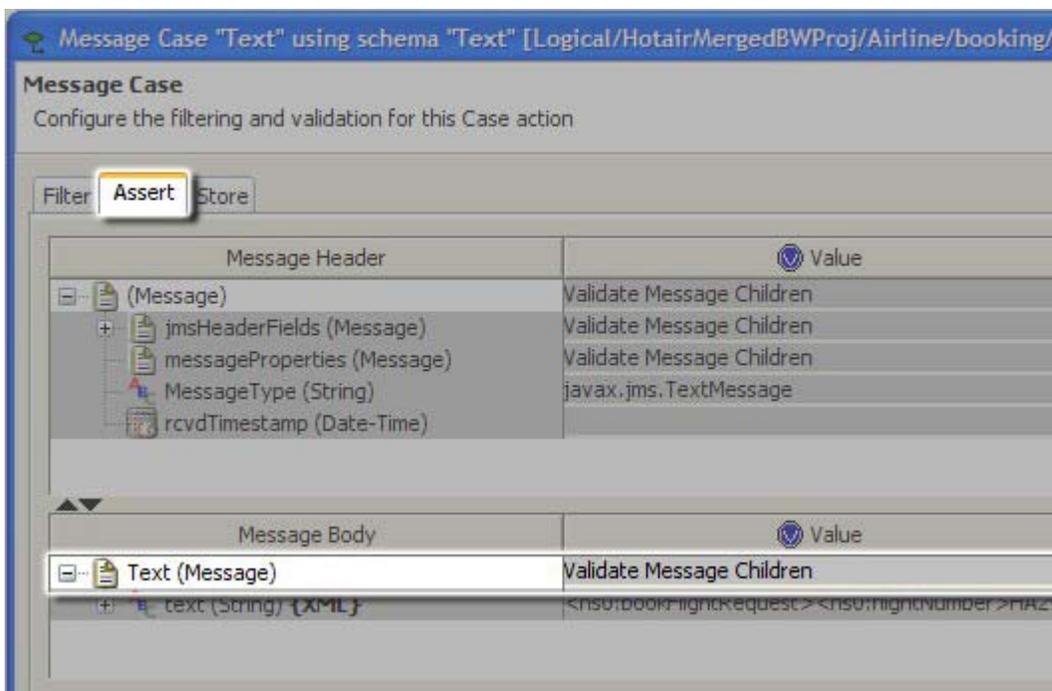
The test cases should fail due to a time-out error. To ignore this error for the purpose of completing the execution of the test suite, we will edit the stub and disable validation on the received message.

7. Open the Test Factory and double-click the stub in the Component View.

-
8. Double-click the **Message Case** step to edit it.



9. In the Message Case dialog, click the **Assert** tab, then disable the option to Validate Message Children for the message body.



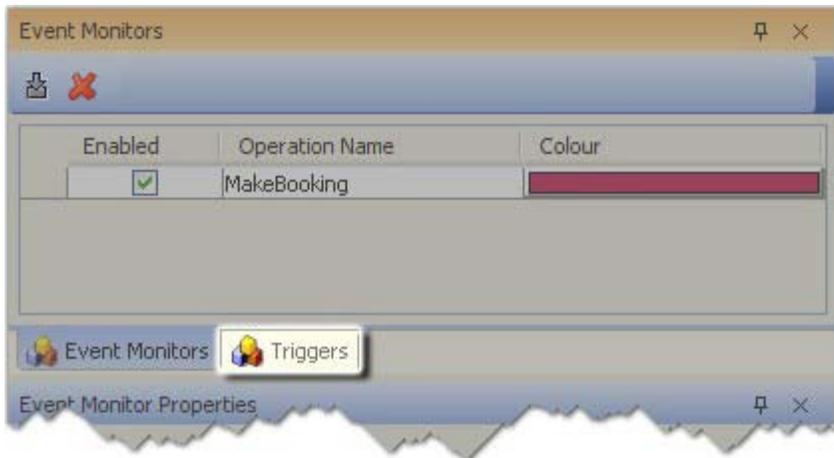
10. Stop and start the stub in the Test Lab, then run the suite again – all tests should pass.

NOTE: When finished, you can stop the stub again.

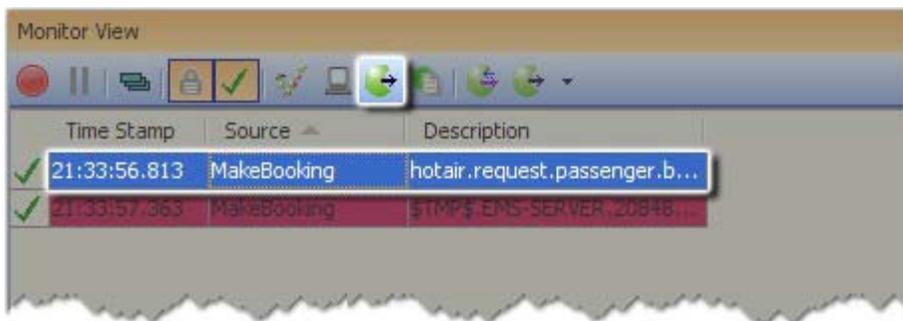
Create and Run a Trigger

A trigger is used to send an event to the system under test. A trigger can be created manually, or from a previously recorded event. In our example, we will use an existing event.

1. Open the Recording Studio.
2. Select the **Triggers** tab, below the Event Monitors window.



3. Select the first recorded event (the request) in the Monitor View and click the **Create Trigger** button in the Monitor View toolbar.

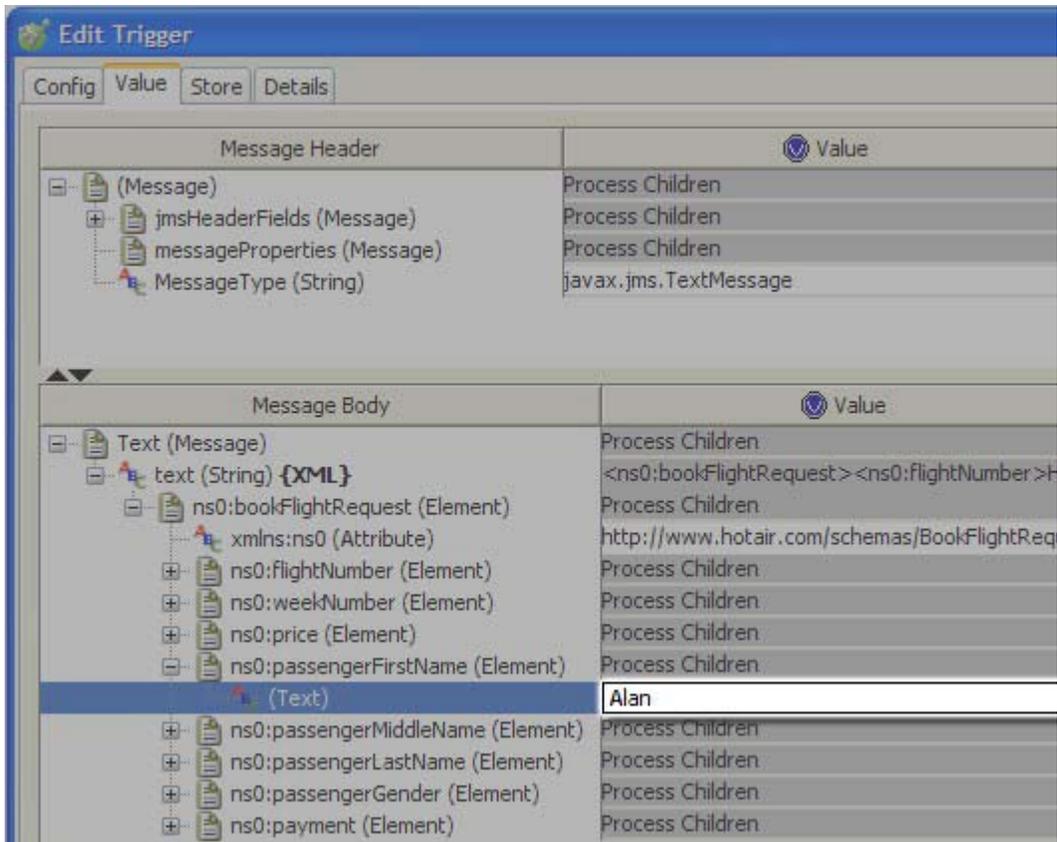


A new trigger is created under the Components tree in the Triggers view.

4. Expand the tree to show the new trigger (expand each node, or click  to expand all nodes).
5. Double-click the trigger to edit it.

The **Edit Trigger** dialog is displayed.

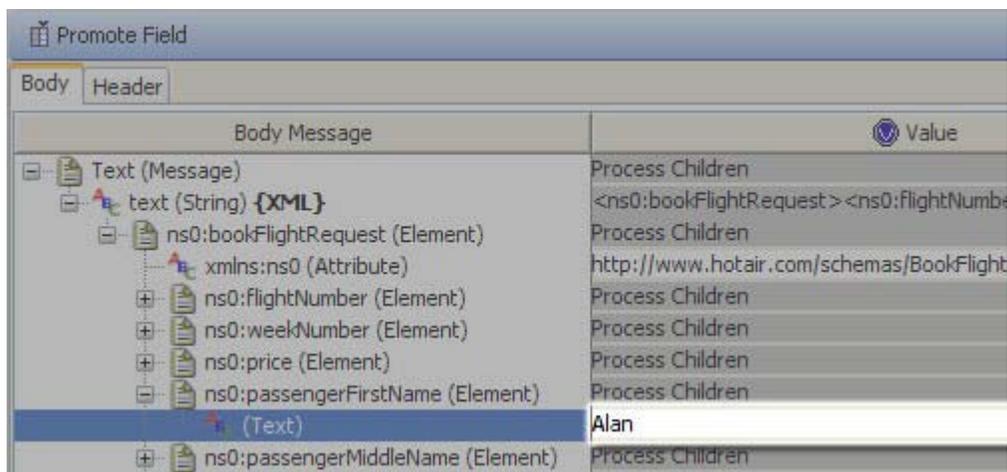
-
6. Select the **Values** tab and update one of the fields (for example, passengerFirstName) in the body.



7. Click **OK** to save the change and close the dialog.
8. Click **Record** in the Monitor View.
9. Run the trigger by selecting it and clicking the **Run** button in the Triggers view.

A new event will be recorded in the Monitor View.

-
10. To verify that the request contains the updated field, select the event and expand the request element to view the updated field (for example, passengerFirstName).



4.6 Testing BusinessWorks Private Processes

Some processes in a TIBCO BusinessWorks project are private (that is, they are only used by other processes). To enable the testing of these processes without having to test their parents, you can add a design time library to your BusinessWorks project by means of TIBCO Designer. The library is copied to *<Rational Integration Tester Installation Directroy>\tools\TIBCO* when Rational Integration Tester is installed.

The following sections demonstrate how to add the library to your project to enable the testing of private processes. Additionally, an example is provided that illustrates how to test such a process in Rational Integration Tester.

4.6.1 Enable Private Process Testing in TIBCO Designer

Follow the steps below to enable the testing of private processes in a TIBCO BusinessWorks project.

1. Open the BW project in TIBCO Designer.
2. Select the Global Variables tab and ensure that the variables for **GH_TESTER_HOST** and **GH_TESTER_PORT** match the values used for **Host** and **GH Tester Process Starter Port** for the TRA file when the BusinessWorks project was imported.

NOTE: If necessary, the values can be changed in the TRA file, located in the Physical View of Rational Integration Tester's Architecture School perspective.

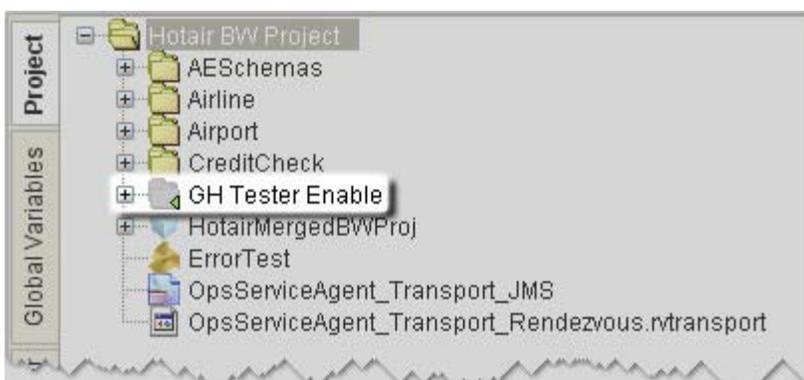
3. Select the root of the project in the **Project** tab.



-
- In the configuration panel, select the **Design Time Libraries** tab.



- Click **+** to locate and select a new design time library.
- Browse to the tools\TIBCO directory in your Rational Integration Tester installation (for example, C:\Program Files\IBM\RationalIntegrationTester\tools\TIBCO).
- Select **GHTesterEnable.projlib** and click **Open**.
- Click **Apply** in the configuration panel and let Designer save the project if necessary.
- A new folder, **GH Tester Enable**, will now appear in the project tree.



- Save the project in Designer.

4.6.2 Private Process Example

Once the design time library provided by Green Hat has been added to your BW project, you can begin testing private processes in Rational Integration Tester. The following example illustrates how to test one such process in the Hotair project.

NOTE: The following example illustrates a manual method of creating tests, but users will most likely be creating one or more tests using the Message Exchange Pattern (MEP) that is defined for the private process operation in Rational Integration Tester. Additionally, this example creates a test using a process that has no input. This should not be considered to be the standard when testing private processes as this is only an example.

1. Follow the steps in [Enable Private Process Testing in TIBCO Designer](#) using the example Hotair project (see [Example: The Hotair Scenario](#)).
2. If necessary, import and synchronize the project with Rational Integration Tester. If you have already imported the project, you must synchronize again after enabling private processes.
3. Open the Test Factory perspective (**F10**) in Rational Integration Tester.
4. Create a new test under the **GenerateNextReservationNumber** operation (under Hotair/Airline/booking) and call it **PrivateProcess**.

NOTE: This operation is a private process in the BusinessWorks project.

5. Add a **Send Request** action to the Test Steps phase, which will also add a **Receive Reply** action.

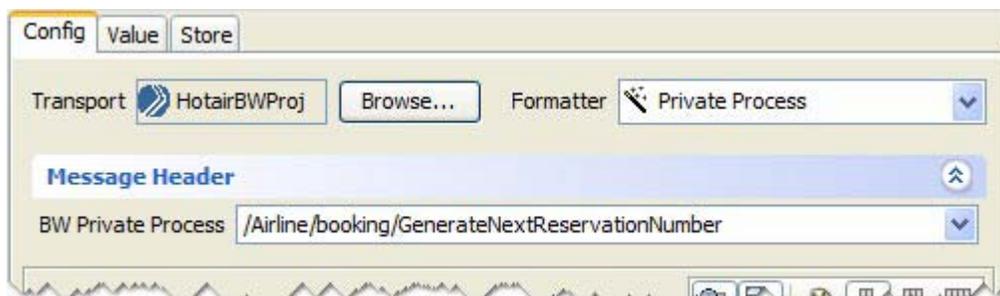
NOTE: Private processes can only be tested using the Send Request/Receive Reply actions.

6. Double-click the **Send Request** action to edit it.
7. Ensure that the root of your Hotair project is selected as the transport. If not, click **Browse** to select it. The formatter will be **Private Process**.

NOTE: The BusinessWorks project is used as the transport when testing private processes.

In the **Message Header** section, the **BW Private Process** combo box lists all of the private processes that are available in the project.

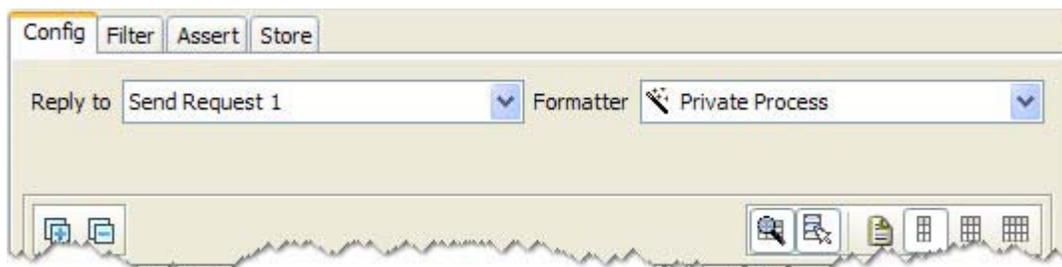
-
8. If necessary, select the private process that you want to test (in this case, **Airline/booking/GenerateNextReservationNumber**).



9. Click **OK** to save the test action and close it.

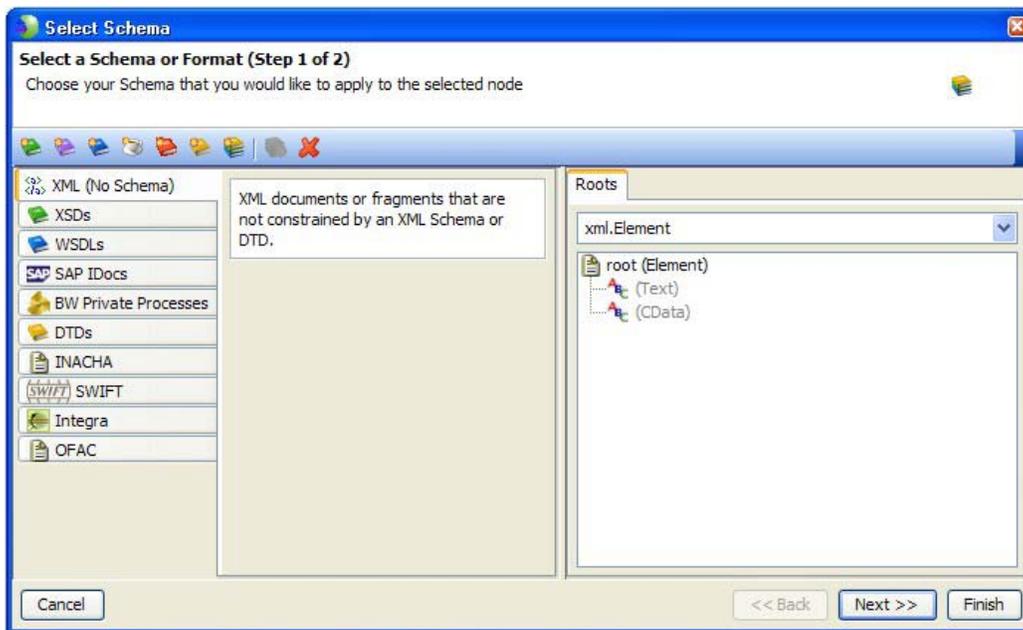
NOTE: There is no need to create a message for the request because the selected process has no input. You can select the Value tab to verify this, noting that the **In** field is false and the **Out** field is true. This means that the process has no input but does have an output.

10. Double-click the Receive Reply action to edit it.
11. Ensure that **Send Request 1** is in the **Reply to** combo box and **Private Process** is the formatter.

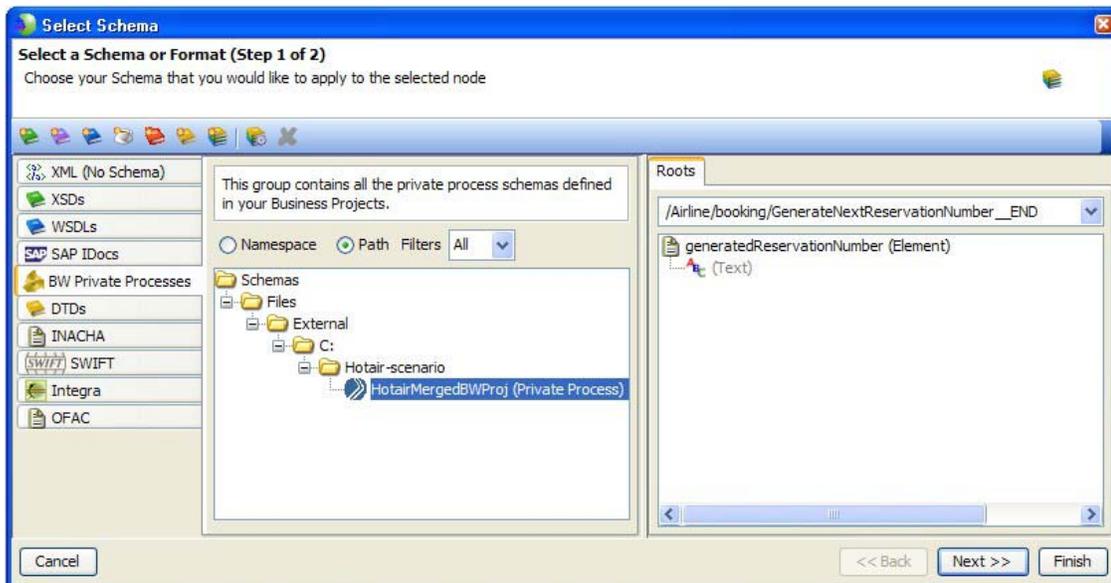


12. Right-click the **text(String)** node in the message and select **Schema** from the context menu.

The **Select Schema** wizard is displayed.



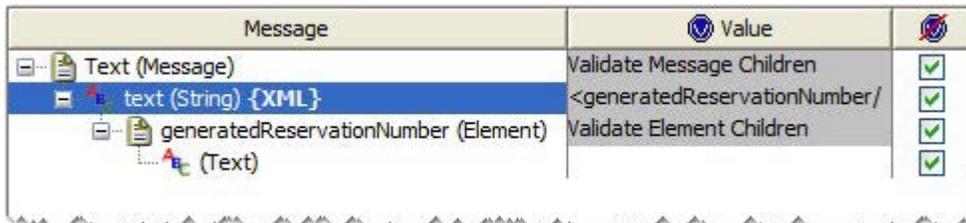
13. Select the **Private Process** tab and select the Hotair project in the tree to the right.
14. Select the **/Airline/booking/GenerateNextReservationNumber_END** root (to the right, under the **Roots** tab).



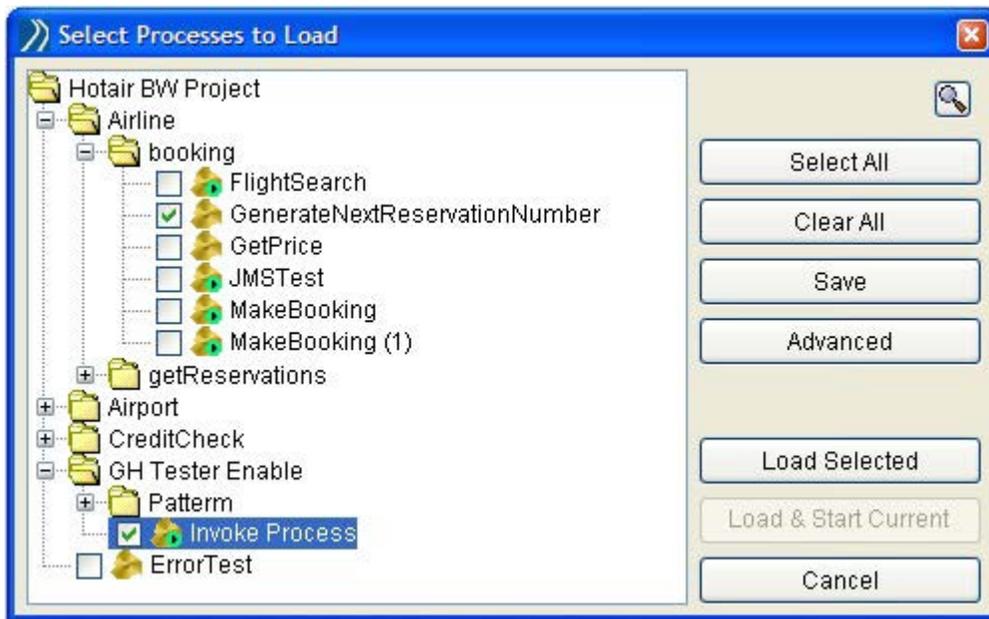
15. Click **Finish** to close the wizard.

NOTE: After selecting a root in the schema wizard, the process selected in the **BW Private Process** combo box will change accordingly (that is, to the process to which the selected root belongs).

16. The selected root be applied to the original (empty) message.



17. Return to TIBCO Designer and select the **Tester** tab, then click .
18. Locate and select the **Invoke Process** process in the **GH Tester Enable** folder and the GenerateNextReservationNumber process under **/Airline/booking**, then click **Load Selected**. If required, let Designer save the project before continuing.



19. Return to Rational Integration Tester and run the PrivateProcess test.
20. The test should fail since the expected message field was left blank. You can repair the failure by overwriting the expected field and run the test again.

SmartSockets Transport

Contents

Libraries

Creating the SmartSockets Transport

Configuring the SmartSockets Transport

Sending SmartSockets Messages

Receiving SmartSockets Messages

This chapter describes the libraries that Rational Integration Tester needs to work with TIBCO SmartSockets. It also describes how to configure the SmartSockets transport.

5.1 Libraries

Depending on the version of TIBCO SmartSockets in use, specific product libraries are required. The following table describes the libraries that are required along with their default location.

SmartSockets Version	Library	Default Location
v6.8	ss.jar ss-flavor.jar	c:\Program Files\tibco\ss68\java\lib

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

NOTE: To use Rational Integration Tester with an earlier version of SmartSockets, create a custom provider and add all of the libraries found in the \java\lib directory of the installation. For more information, refer to *IBM Rational Integration Tester Installation Guide*.

5.2 Creating the SmartSockets Transport

The TIBCO SmartSockets transport is created when you create a physical TIBCO SmartSockets Connection/Broker resource in Rational Integration Tester's Architecture School. The physical resource can only be used once it has been bound to a logical SmartSockets Cloud component.

To create the logical SmartSockets Cloud:

1. Open the Logical View of Architecture School (**F7**).
2. Right-click the operation that should contain the SS Cloud, or right-click the drawing palette to create the component at the top level, then select **New > TIBCO > TIBCO SmartSockets Cloud**.

NOTE: You may also select the destination for the logical resource and then select **TIBCO SmartSockets Cloud** from the **TIBCO** menu.

3. Provide a name for the new SmartSockets Cloud when prompted, then click **OK**.

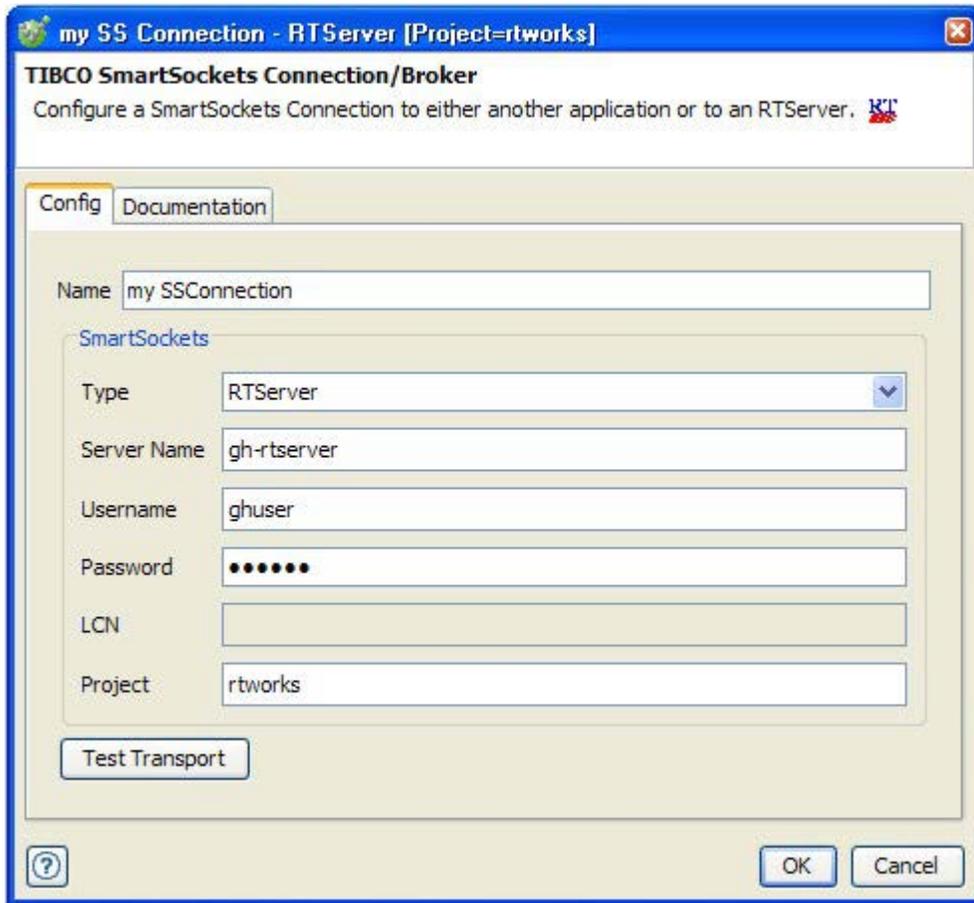
To create the physical SmartSockets Connection/Broker:

- In the Logical View, right-click on a TIBCO SmartSockets Cloud and select the **Set Binding in > [environment] > Create new TIBCO SmartSockets Connection/Broker** option. You can then open the Physical View and double-click the new resource to configure it (see [Configuring the SmartSockets Transport](#)).
- In the Physical View, select the **TIBCO > TIBCO SmartSockets Connection/Broker** option (or right-click the Physical folder and select **New > TIBCO > TIBCO SmartSockets Connection/Broker**). The new resource will be opened for editing (see [Configuring the SmartSockets Transport](#)).

Each physical resource will represent a TIBCO SmartSockets Connection/Broker transport that can be selected and configured as described in [Configuring the SmartSockets Transport](#).

5.3 Configuring the SmartSockets Transport

To configure a SmartSockets transport, double-click the appropriate TIBCO SmartSockets Connection/Broker 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 SmartSockets transports are available).

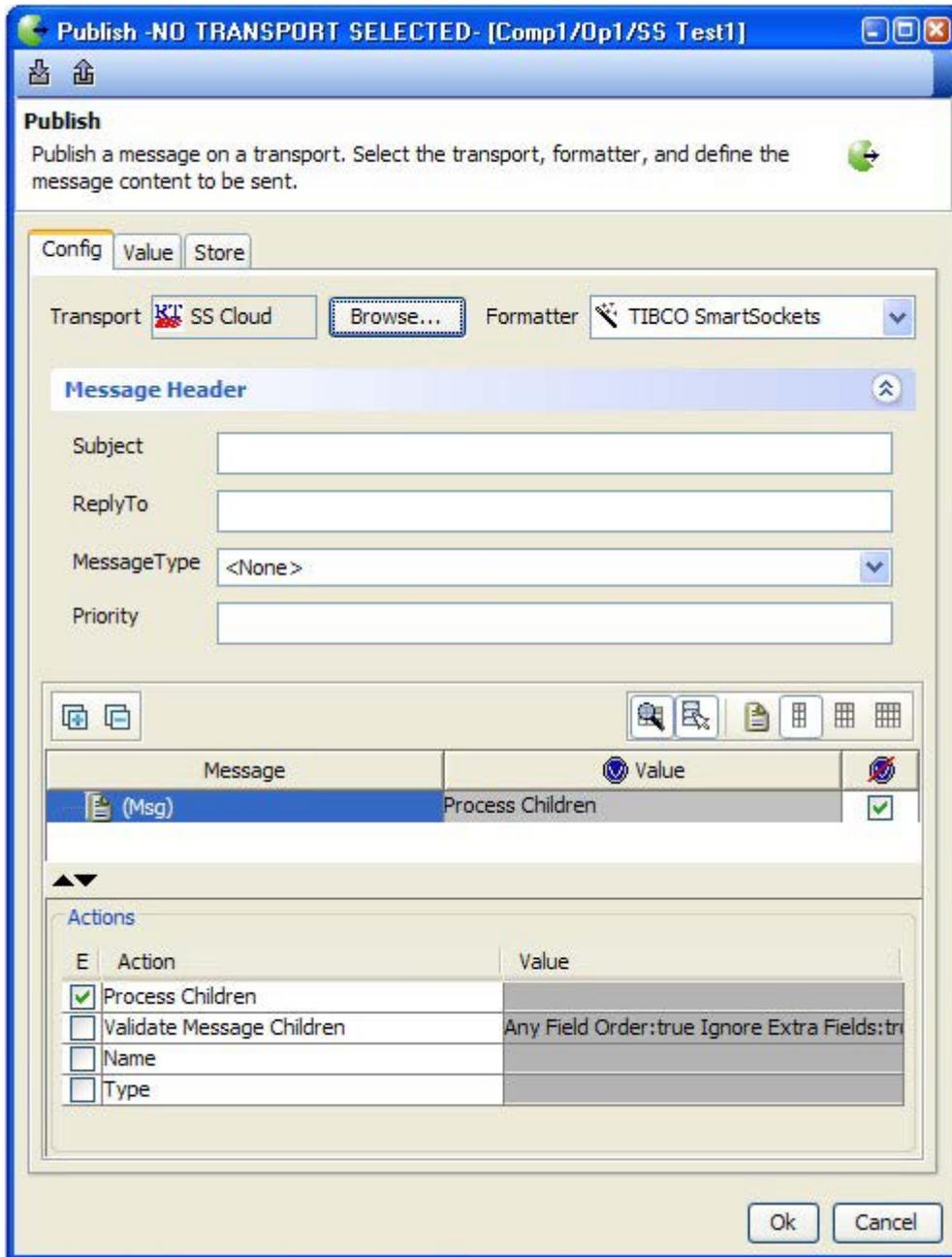
NOTE: When configuring the transport, all of the configuration fields support the use of tags. Tags can be entered manually or from the context menu, except in the **Password** field, where tag names must be entered directly (for example, %%JMS_password%%). Since this field is encrypted, any characters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

The transport settings are configured in the **SmartSockets** panel. The available configuration options are described in the following table:

Field	Description
Type	Select the desired connection type (as described in TIBCO SmartSockets User's Guide) from RTServer , Client , or Server .
Server Name	For RTServer connections, enter the host name or IP address of the RTServer to which you want to connect.
Username	For RTServer connections, enter the user name to send when connecting to the server.
Password	For RTServer connections, enter the password required for the specified user name (above).
LCN	For Client and Server connections, specify the logical connection name to use.
Project	For all connections, specify the existing SmartSockets project to use for the connection.

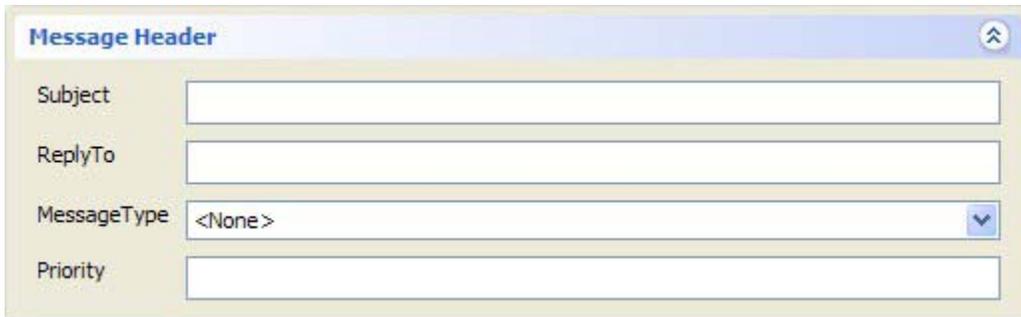
5.4 Sending SmartSockets Messages

When publishing by means of SmartSockets, you must configure the message header (see [Configure SmartSockets Headers](#)) and the message body ([Configure the SmartSockets Message Body](#)).



5.4.1 Configure SmartSockets Headers

SmartSockets header information is configured under the **Message Header** area.



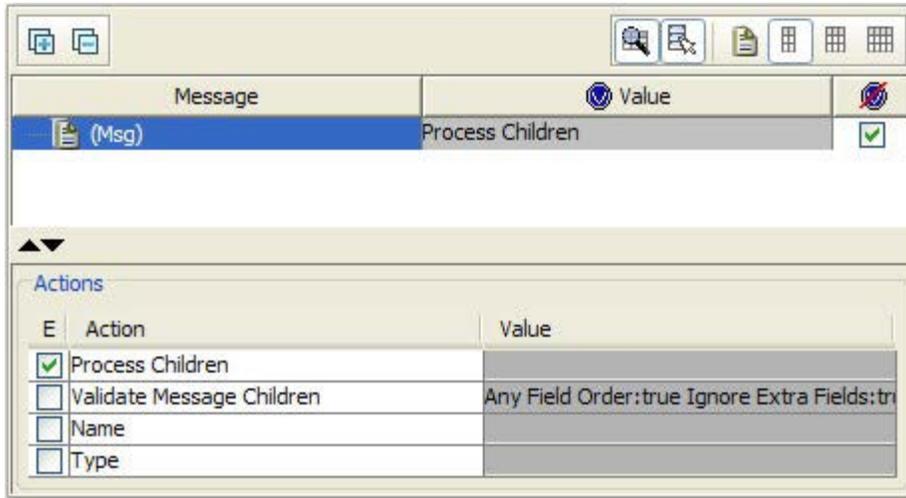
The SmartSockets header options are described in the following table:

Option	Description
Subject	The subject of the message being published.
ReplyTo	The destination or subject where a reply to the messages should be sent.
MessageType	The Message Type of the message.
Priority	Any integer from 0 to 65535, the priority identifies the level of importance of a message. The greater the number, the higher the importance. Priority determines the order in which a message is processed after it has been received. It does not affect the order in which messages are published. When a message is sent to a process through a connection, it is placed in the connection message queue of the receiving process in priority order.

NOTE: See your TIBCO SmartSockets documentation for more information about message header properties.

5.4.2 Configure the SmartSockets Message Body

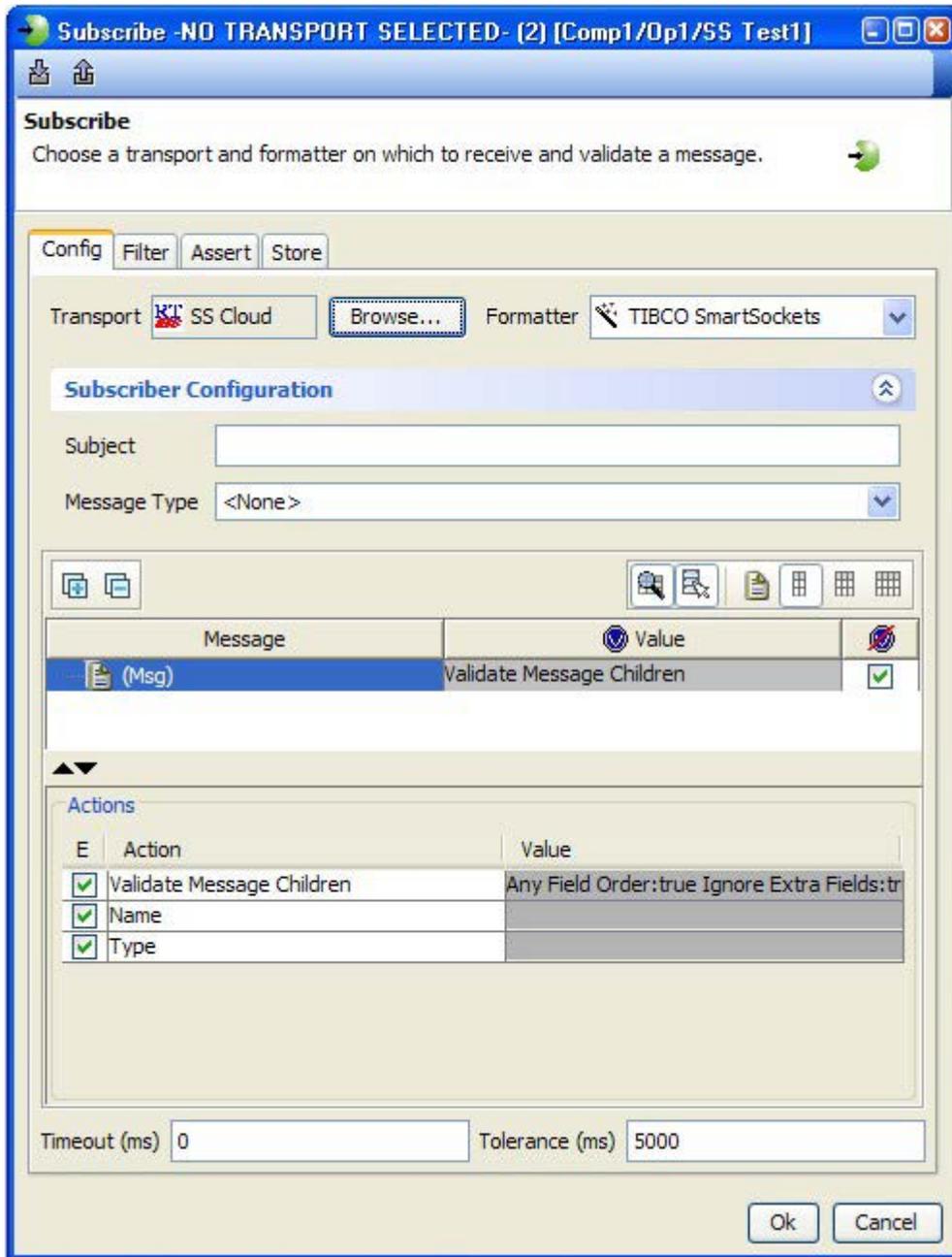
The contents and structure of a SmartSockets message body can be configured like any other message in Rational Integration Tester.



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

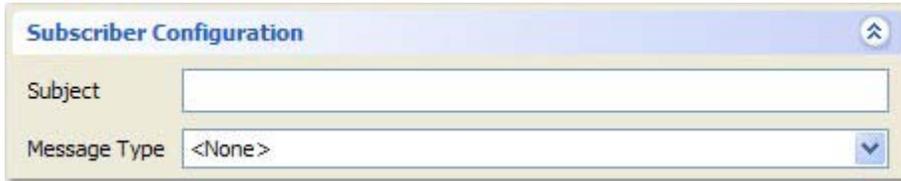
5.5 Receiving SmartSockets Messages

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



5.5.1 Configure Subscriber Options

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



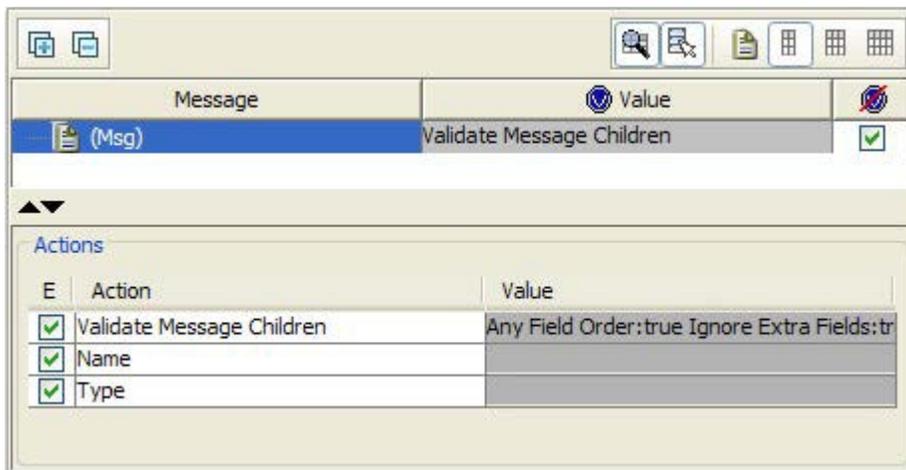
The subscriber options are described in the following table:

Option	Description
Subject	The subject of messages to receive.
Message Type	The Message Type of published messages to receive.

NOTE: See your TIBCO SmartSockets documentation for more information about message properties.

5.5.2 Configure Message Content

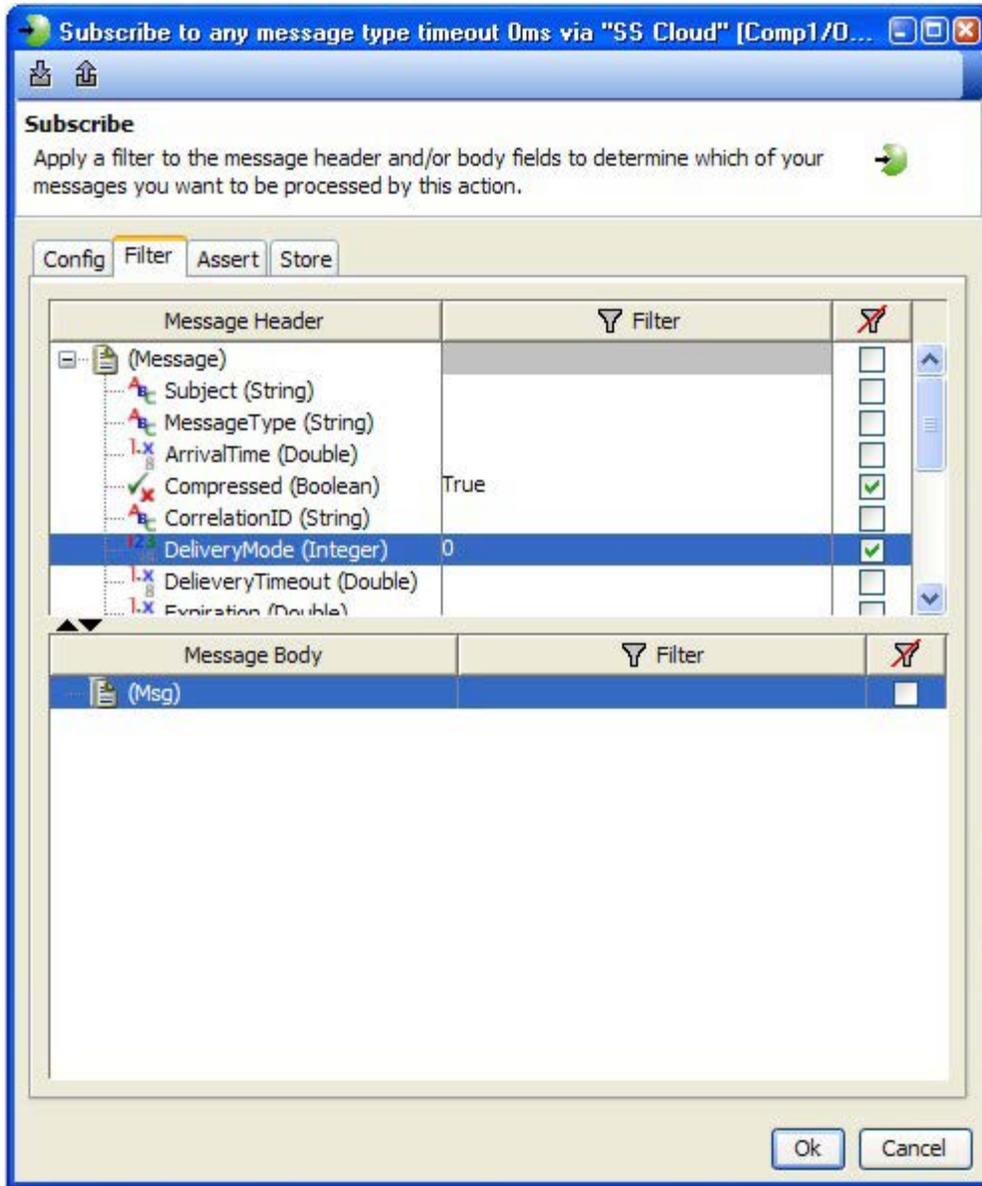
The content, structure, and validation actions of a SmartSockets message body can be configured like any other message in Rational Integration Tester.



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

5.5.3 Message 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.



In this case, **Compression** must be true and **DeliveryMode** must equal "0". Otherwise, Rational Integration Tester will discard the message.

Troubleshooting

Contents

No Formatter Available

Unable to Connect to Daemon

Overlapping Subjects Error

Class Definitions Not Visible

**AE Transport Unable to Use
BusinessWorks 5 Projects**

Unable to Watch EMS Queues

**Unable to Access Topics/Queues
on Remote Servers**

**“Not Permitted” Error When
Watching**

**Unable to Tag Generated Tracking
ID**

Failed to Rebuild Schema

This chapter provides answers to some of the common questions and issues that may be encountered when testing TIBCO products in Rational Integration Tester.

6.1 No Formatter Available

If the **Formatter** combo box is blank in a message editor, it is likely that the requisite jar files were not included on the classpath when Rational Integration Tester started. Note that the Library Manager must be configured and enabled for the TIBCO AE provider.

6.2 Unable to Connect to Daemon

This can occur when either the application is unable to start a suitable local daemon process or if the daemon settings in the transport configuration are invalid/reference an unreachable remote transport. In these cases, check that the directory in which the **rvd** executable is present is located on the system PATH environment variable.

6.3 Overlapping Subjects Error

When using an RVC-based transport, there are restrictions to the subjects to which you can subscribe at any given point. For example, it is not possible to have two tests using the same RVC transport with subscriptions to the subjects 'test.>' and 'test.application'. This would result in an error as you could potentially have two subscribers to the same subject and Rendezvous would not know how to handle the certified acknowledgements.

6.4 Class Definitions Not Visible

Changes made to the schema definitions inside of TIBCO Designer will not be visible in Rational Integration Tester until you synchronize the artifacts again. This is done in the Synchronise view of Architecture School.

6.5 AE Transport Unable to Use BusinessWorks 5 Projects

BusinessWorks 5 uses TIBCO AE version 5 libraries. You must run Rational Integration Tester with AE5 JAR files (v4 will not work). These settings can be confirmed in the Library Manager.

6.6 Unable to Watch EMS Queues

Ensure that the transport connects with administration permissions as watching involves accessing system topics instead of the target queues. You also need to uncheck the “Use JNDI to lookup destination” connection option.

6.7 Unable to Access Topics/Queues on Remote Servers

This is normally due to a problem with how the connection factory is configured. When you make a remote connection to an EMS server, the initial connection to the connection factory returns the location of the queue or topic. If the connection factory is configured incorrectly for a default port number or localhost, it can tell Rational Integration Tester to connect to its local machine or use the wrong port number. Please refer to the TIBCO EMS documentation and check the connection factory settings using the TIBCO tool **tibemsadmin**. An example follows where the connection factory has a correct URL (that is, no host name):

```
tcp://ghc-sun4:7222> show factory QueueConnectionFactory
Factory                = QueueConnectionFactory
JNDI Names             = "QueueConnectionFactory"
URL                   = tcp://7222
ClientID              =
Load Balanced         = no
```

6.8 “Not Permitted” Error When Watching

If the admin credentials are edited after an action has been performed using the EMS transport, Rational Integration Tester will need to be restarted to fully apply the changes.

6.9 Unable to Tag Generated Tracking ID

When publishing messages using the AE transport, the TIBCO APIs will add a tracking ID. Store actions within a publisher are currently unable to process this value.

6.10 Failed to Rebuild Schema

If transports are created on the Windows platform and used on a UNIX platform, the path separators may prevent the transport's schema from being able to refresh. In this case, the reference to the DAT file needs to be updated within the UNIX environment.

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.
Subject	A user-meaningful name for identifying messages on TIBCO transports. For example, the subject EQ.IBM might identify all pricing data about IBM stocks, while EQ.IBM.N might identify price data from the New York Stock Exchange only. See also: JMS queue, JMS topic.
Server	A host computer on a network shared by more than one user.
Subject	A user-meaningful name for identifying data objects. For example, the subject EQ.IBM might identify all pricing data about IBM stocks, while EQ.IBM.N might identify price data from the New York Stock Exchange only.

Term	Description
TIBCO Rendezvous	A software toolkit for creating distributed applications that can inter-operate with TIBCO servers and applications on a TIBCO network. The product includes a communications daemon and APIs that define protocols for publish/subscribe and request/reply data distribution and exchange. It uses the subject-based addressing™ messaging technique for data delivery, and defines rules for supported subject naming formats.
TIBCO AE Message	A TIBCO proprietary format for messages contained within the Rendezvous
TIBCO EMS	TIBCO Enterprise Messaging Service - JMS transport.

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.

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[®]