

*SeeBeyond ICAN Suite*

# eGate Integrator JMS Reference Guide

*Release 5.0.1*



The information contained in this document is subject to change and is updated periodically to reflect changes to the applicable software. Although every effort has been made to ensure the accuracy of this document, SeeBeyond Technology Corporation (SeeBeyond) assumes no responsibility for any errors that may appear herein. The software described in this document is furnished under a License Agreement and may be used or copied only in accordance with the terms of such License Agreement. Printing, copying, or reproducing this document in any fashion is prohibited except in accordance with the License Agreement. The contents of this document are designated as being confidential and proprietary; are considered to be trade secrets of SeeBeyond; and may be used only in accordance with the License Agreement, as protected and enforceable by law. SeeBeyond assumes no responsibility for the use or reliability of its software on platforms that are not supported by SeeBeyond.

SeeBeyond, e\*Gate, and e\*Way are the registered trademarks of SeeBeyond Technology Corporation in the United States and select foreign countries; the SeeBeyond logo, e\*Insight, and e\*Xchange are trademarks of SeeBeyond Technology Corporation. The absence of a trademark from this list does not constitute a waiver of SeeBeyond Technology Corporation's intellectual property rights concerning that trademark. This document may contain references to other company, brand, and product names. These company, brand, and product names are used herein for identification purposes only and may be the trademarks of their respective owners.

© 2003 by SeeBeyond Technology Corporation. All Rights Reserved. This work is protected as an unpublished work under the copyright laws.

**This work is confidential and proprietary information of SeeBeyond and must be maintained in strict confidence.**

Version 20031121151445.

# Contents

<b>List of Tables</b>	<b>8</b>
<b>List of Figures</b>	<b>9</b>
<hr/>	
<b>Chapter 1</b>	
<b>Introduction</b>	<b>11</b>
Contents of This Guide	11
Writing Conventions	12
Additional Conventions	12
Supporting Documents	13
SeeBeyond Web Site	13
<hr/>	
<b>Chapter 2</b>	
<b>About Message Servers in eGate Integrator</b>	<b>14</b>
About the JMS IQ Manager	14
Message Delivery Order	14
Message Priorities	15
Security	16
Distributed Transactions	16
Connection Consumer	16
About the WebLogic JMS Server	16
About the WebSphere JMS	17
Message Servers in the eGate Implementations	17
The eGate Integrator Model	17
Project Design Phase: Topics, Queues and JMS Properties	19
Environment Design Phase: Message Server Configurations	20
Deployment Profile Design Phase: Creating Mappings	21
Activating Deployment Profile Phase: Configuration File	22
Message Servers at Runtime	22
Quick Overview: Building eGate Projects with JMS	22
Message Server Configuration and Administration Tools	23

---

Chapter 3

<b>Functional Overview of the JMS IQ Manager</b>	<b>24</b>
<b>About the JMS IQ Manager Database</b>	<b>24</b>
About Segments	24
About the Database Location	25
The Default Database Configuration	25
<b>Message Processing Order</b>	<b>27</b>
JMS IQ Manager FIFO Delivery Modes	27
Fully Concurrent Processing	27
Protected Concurrent Processing	28
Fully Serialized Processing	29
JMS Client Concurrency Effect on FIFO Delivery Modes	30
Serial Processing Across a Destination Group	31
<b>About Performance</b>	<b>31</b>
Managing Resources	32
Throttling Producers	32

---

Chapter 4

<b>Installing Third-Party and 4.5.x Message Server Support</b>	<b>35</b>
Installing Message Server .sar Files	35
Additional Instructions for WebLogic	36
Additional Instructions for WebSphere	36

---

Chapter 5

<b>Configuring Collaborations for JMS Web Services</b>	<b>37</b>
Adding Message Destinations to Projects	37
Adding JMS Web Services to Collaboration Definitions	38
Using JMS Methods in Collaboration Definitions	41
JMS Methods Available for JMS OTDs	41
JMS Methods Available for JMS Messages	54
Examples of Building JMS Collaboration Definitions	59
Sending Text Messages with the <i>sendTo</i> Method	60
Populating Text Messages	60
Creating the Text Message To Be Sent	61
Sending the Message to a Specified Destination ( <i>sendTo</i> )	63
Receiving and Sending Messages	65
Defining the Message To Be Sent	65
Receiving the Message from a Specified Destination	67
Sending Messages to a Different Destination	68
Setting JMS Message Properties in Collaborations	69

Binding Collaboration Definitions to Services	72
Linking JMS Web Service Collaborations to Destinations	73

## Chapter 6

### Configuring JMS Clients 76

<b>Configuring JMS Clients</b>	<b>76</b>
Setting JMS Client Message Processing Order	79
Specifying Persistent or Non-Persistent Delivery Mode	79
Specifying Message Selectors	80
Specifying Message Priorities	80
Specifying the Transaction Mode	81
Specifying Security Properties	81
Specifying Security Name and Password	82
Specifying the Security Realm	82
Use for JMS Connection	82
Enabling Auditing	83
Enabling Authentication	83
Enabling Authorization	83
Setting Durability	84
Setting the Number of ServerSessionPool Threads	84

## Chapter 7

### Adding and Configuring Message Servers 86

<b>Adding Message Servers to Logical Hosts</b>	<b>86</b>
<b>Configuring Message Servers</b>	<b>88</b>
<b>JMS IQ Manager Configurations</b>	<b>88</b>
<b>Setting General Options</b>	<b>88</b>
Enabling Security	89
Specifying the Host Name	89
Specifying the Server Port Number	90
<b>Setting Performance Options</b>	<b>90</b>
Specifying Cache Size	91
Enabling Cache Control	91
Locking Cache Into RAM	92
Specifying the Maximum Number of Segments	92
Specifying the Maximum Amount of Payload Memory	93
Specifying Padding for the Payload Memory Threshold	93
Specifying the Minimum Number of Segments	94
Specifying the Maximum Amount of Physical RAM for Processes	94
Specifying Segment Size	95
Specifying the Throttling Threshold for the Total Number of Messages	96
Specifying the Throttling Threshold for Message Destinations	96
Specifying the Throttling Lag	96
<b>Setting Delivery Ordering Options</b>	<b>97</b>
Specifying Special FIFO Mode Expiration Time	97
Specifying Special FIFO Modes	98

Specifying Delivery Order by Time Dependency Group	98
<b>Setting Logical Host Options</b>	<b>99</b>
Specifying Heap Size	100
<b>Setting Diagnostic Options</b>	<b>100</b>
Specifying the Action Interval for Performance Statistics Logging	100
Enabling Logging	101
Enabling Object-level Logging	101
Enabling Performance Logging	102
Enabling Timestamp Logging	102
Enabling Verbose Logging	102
Specifying the Journal Logging Level	103
Specifying the Logging Level	103
Specifying the Message Interval for Performance Statistics Logging	104
Specifying the Interval for Performance Statistics Logging	104
<b>Setting Stable Storage and Journaling Options</b>	<b>104</b>
Specifying the Location of JMS IQ Manager Database Files	105
Specifying the Prefix for Database Filenames	106
Specifying the File Extension for Database Files	106
Enabling Journaling	107
Specifying the Location of the Journal Database	107
Specifying Time To Live for Journaled Messages	108
Specifying Time To Live for Live Messages	108
<b>4.5.x SeeBeyond Message Server Configurations</b>	<b>109</b>
<b>Deploying Topics and Queues to Message Servers</b>	<b>110</b>
<b>Reconfiguring Message Servers</b>	<b>112</b>

---

## Chapter 8

<b>Managing JMS IQ Managers</b>	<b>114</b>
<b>About Administration Tools</b>	<b>114</b>
<b>Managing JMS IQ Managers with the Enterprise Manager</b>	<b>114</b>
Starting Enterprise Manager	115
Starting and Stopping JMS IQ Managers	116
Monitoring Topics and Queues	117
Viewing Message Properties	119
Viewing Message Details	120
Viewing Message Destination Summaries	121
Viewing Message Payload	123
<b>Managing JMS IQ Managers with the MS Control Utility</b>	<b>123</b>
Syntax	124
Flags and Arguments	124
Displaying the Utility Version	126
Shutting Down the Server	126
Viewing Message Server Statistics	126
Viewing All Topics for a Server	127
Changing Topic Message Contents	127
Viewing Statistics for Topics	127
Viewing Properties of All Subscribers	128
Viewing Properties of All Subscribers to Topics	128

## Contents

Viewing All Queues for a Server	128
Displaying Statistics for Queues	129
Viewing Properties of All Receivers	129
Viewing Properties of All Receivers of Queues	129
Republishing Messages from Topics	129
Republishing Messages from Queues	130
Browsing Journalled Messages	130
Backing Up	131
Browsing Archives	131
Setting Timeout	132
stcmsutil.Properties	132

<b>Index</b>	<b>134</b>
--------------	------------

# List of Tables

Table 1	Writing Conventions	12
Table 2	JMS IQ Manager Administration Tools	23
Table 3	Benefits and Costs—Fully Concurrent Processing	28
Table 4	Benefits and Costs—Protected Concurrent Processing	29
Table 5	Benefits and Costs—Fully Serialized Processing	29
Table 6	JMS Client Concurrency Effect on FIFO modes—Topic	31
Table 7	JMS Client Concurrency Effect on FIFO modes—Queue	31
Table 8	Publisher Throttling	34
Table 9	JMS Message Properties	70
Table 10	Topic Properties	118
Table 11	Queue Properties	119
Table 12	Message Properties	120
Table 13	Message Details	121
Table 14	MS Control Utility Flags and Arguments	124
Table 15	stcmsutil.Properties - Message Service Utility Property Names and Values	133



# List of Figures

Figure 1	eGate Integrator Model	18
Figure 2	Message Server in the eGate Integrator Model	19
Figure 3	eGate Integrator Projects	20
Figure 4	Adding Message Servers to an Environment	21
Figure 5	Deployment Profile Example	21
Figure 6	JMS IQ Manager Database Files Location—Data Directory	25
Figure 7	JMS IQ Manager Database Structure	26
Figure 8	Fully Concurrent Processing	28
Figure 9	Protected Concurrent Processing	28
Figure 10	Fully Serialized Processing	29
Figure 11	Multiple Integration Server Configuration	30
Figure 12	Selecting a Project	38
Figure 13	Java Collaboration Definition Wizard—Step 2 Dialog Box	39
Figure 14	Java Collaboration Definition Wizard—Adding the JMS OTD	40
Figure 15	Java Collaboration Editor—Collaboration with JMS OTD	41
Figure 16	Populating Text Messages	61
Figure 17	Mapping Input and Output Components	61
Figure 18	Assigning the Marshal Result to Destination	62
Figure 19	Creating a Variable	63
Figure 20	Creating a Message in a Collaboration Definition	63
Figure 21	Sending Messages to a Different Destination	65
Figure 22	Creating a Variable	66
Figure 23	Finding Classes for Message Variables	67
Figure 24	Populating the Message	68
Figure 25	Sending Messages to a Different Destination	69
Figure 26	JMS Message Priorities	71
Figure 27	Setting JMS Message Properties	72
Figure 28	Binding Collaboration Definitions to Services	73
Figure 29	Changing a Service to a Collaboration	73
Figure 30	Collaboration Binding Window	74
Figure 31	Binding Services to eWays	74
Figure 32	Collaboration Connections to Destinations	75

## List of Figures

Figure 33	Finished Connectivity Map	75
Figure 34	JMS Client Properties Icon	77
Figure 35	Basic JMS Client Properties	77
Figure 36	Expanding the Environments Folder	87
Figure 37	JMS IQ Manager—General Configurations	89
Figure 38	JMS IQ Manager—Performance Configurations	90
Figure 39	JMS IQ Manager—Delivery Order Configurations	97
Figure 40	JMS IQ Manager—Logical Host Configurations	99
Figure 41	JMS IQ Manager—Diagnostic Configurations	100
Figure 42	JMS IQ Manager—Stable Storage and Journaling Configurations	105
Figure 43	JMS IQ Manager Database Files Location—Data Directory	106
Figure 44	JMS IQ Manager Database Files Location—Data Directory	108
Figure 45	4.5.x Message Server Properties Dialog Box	110
Figure 46	Initial Deployment Profile	111
Figure 47	Deployment Profile Example	112
Figure 48	Enterprise Manager Home Page	115
Figure 49	Enterprise Manager—Monitor Tab	116
Figure 50	Starting or Stopping Message Servers	117
Figure 51	Viewing Topics	118
Figure 52	Viewing Message Properties	120
Figure 53	Viewing Message Details	121
Figure 54	Summary Page	122
Figure 55	Using the Message Slider to Select a Message	122
Figure 56	Viewing Message Payload	123

# Introduction

This reference guide describes the eGate Integrator JMS implementation as well as the supported third-party JMS message servers. The JMS implementation provided with eGate Integrator is the JMS IQ Manager, also referred to as the STCMS. eGate Integrator also supports several third-party JMS applications such as the WebLogic JMS and the WebSphere JMS. This guide describes how you, as an administrator and user of eGate Integrator, configure and manage message servers. This reference guide assumes that you are familiar with Java and the JMS API.

## In This Chapter

- [Contents of This Guide](#) on page 11
- [Writing Conventions](#) on page 12
- [Supporting Documents](#) on page 13
- [SeeBeyond Web Site](#) on page 13

---

## 1.1 Contents of This Guide

This guide contains the following information:

- [Chapter 2, “About Message Servers in eGate Integrator” on page 14](#) provides an overview of the JMS IQ Manager and supported third-party message servers, as well as message server administration tools available in eGate Integrator.
- [Chapter 3, “Functional Overview of the JMS IQ Manager” on page 24](#) provides detailed information about how the JMS IQ Manager processes messages. This chapter discusses such concepts as the JMS IQ Manager database, message processing, security, performance, and optimization.
- [Chapter 4, “Installing Third-Party and 4.5.x Message Server Support” on page 35](#) describes how to install the .sar files needed for third-party and 4.5.x message server support. Without installing these files, the options to select different types of message servers will be unavailable in Enterprise Designer.
- [Chapter 5, “Configuring Collaborations for JMS Web Services” on page 37](#) describes how to use Enterprise Designer to add message destinations (topics and queues) and build the JMS Web Services.

- **Chapter 6, “Configuring JMS Clients” on page 76** describes how to configure JMS properties for JMS IQ Managers and third-party message servers such as the WebSphere and WebLogic JMS.
- **Chapter 7, “Adding and Configuring Message Servers” on page 86** describes how to use Enterprise Designer to configure JMS IQ Managers and third-party message servers such as the WebSphere and WebLogic JMS.
- **Chapter 8, “Managing JMS IQ Managers” on page 114** describes how you can use Enterprise Manager and the STC Message Server Control utility to manage JMS IQ Managers.

## 1.2 Writing Conventions

The following writing conventions are observed throughout this document.

**Table 1** Writing Conventions

Text	Convention	Example
Button, file, icon, parameter, variable, method, menu, and object names.	<b>Bold</b> text	<ul style="list-style-type: none"> <li>▪ Click <b>OK</b> to save and close.</li> <li>▪ From the <b>File</b> menu, select <b>Exit</b>.</li> <li>▪ Select the <b>logicalhost.exe</b> file.</li> <li>▪ Enter the <b>timeout</b> value.</li> <li>▪ Use the <b>getClassName()</b> method.</li> <li>▪ Configure the <b>Inbound File</b> eWay.</li> </ul>
Command line arguments and code samples	Fixed font. Variables are shown in <b><i>bold italic</i></b> .	<code>bootstrap -p <b><i>password</i></b></code>
Hypertext links	<b>Blue</b> text	For more information, see <b>“Writing Conventions” on page 12.</b>

### Additional Conventions

#### Windows Systems

For the purposes of this guide, references to “Windows” will apply to Microsoft Windows Server 2003, Windows XP, and Windows 2000.

#### Path Name Separator

This guide uses the backslash (“\”) as the separator within path names. If you are working on a UNIX system, please make the appropriate substitutions.

---

## 1.3 Supporting Documents

The following SeeBeyond documents provide additional information about the ICAN Suite:

- *SeeBeyond Integrated Composite Application Network Suite Primer*
- *eGate Installation Guide*
- *eGate Integrator User's Guide*
- *eGate Integrator Tutorial*
- *SeeBeyond ICAN Suite Deployment Guide*

---

## 1.4 SeeBeyond Web Site

The SeeBeyond Web site is a useful source for product news and technical support information at [www.seebeyond.com](http://www.seebeyond.com).

# About Message Servers in eGate Integrator

eGate Integrator supports several types of message servers. eGate's own JMS implementation, the JMS IQ Manager, is included with eGate Integrator. eGate Integrator also provides support for the following third-party message servers:

- WebLogic Server 8.1 JMS for Logical Hosts on the BEA WebLogic Platform™ 8.1
- Websphere® JMS for Logical Hosts on Websphere Application Servers V5.0.x

This chapter provides an overview of all message server types, and introduces the tools that eGate Integrator provides for message server configuration and administration.

## In This Chapter:

- [About the JMS IQ Manager](#) on page 14
- [About the WebLogic JMS Server](#) on page 16
- [About the WebSphere JMS](#) on page 17
- [Message Servers in the eGate Implementations](#) on page 17
- [Quick Overview: Building eGate Projects with JMS](#) on page 22
- [Message Server Configuration and Administration Tools](#) on page 23

---

## 2.1 About the JMS IQ Manager

eGate Integrator includes the JMS IQ Manager as its Java Messaging Service implementation, also called STCMS. The JMS IQ Manager conforms to the Java Message specification version 1.0.2b and includes the following features:

- [Message Delivery Order](#) on page 14
- [Message Priorities](#) on page 15
- [Security](#) on page 16
- [Distributed Transactions](#) on page 16

### 2.1.1 Message Delivery Order

The JMS IQ Manager provides the following special facilities to maintain message order in concurrent processing and across message destinations. These facilities are not mandated by the Java Message specification.

- Configuring the JMS IQ Manager for special first-in, first out (FIFO) ordering modes.
- Specifying a set of message destinations (a time order group) for which fully serialized processing occurs.
- Configuring JMS clients for concurrent or serial processing.

The sections below describe each method of processing order. For detailed information about processing order, refer to [“Message Processing Order” on page 27](#).

### Special FIFO Modes for All Destinations

For each topic or queue, you can specify the following FIFO ordering modes:

- Fully concurrent first-in, first out (FIFO) order
- Protected concurrent FIFO order
- Fully serialized FIFO order

For detailed information regarding these special FIFO ordering methods, refer to [“Specifying Special FIFO Modes” on page 98](#).

### Fully Concurrent Processing Across a Time Order Group

You can also specify delivery order specifically for a set of topics and queues (a time order group). For this group, consumers can only receive messages when all other messages in the time order group have been received or are in the process of being received. For information, refer to [Setting Delivery Ordering Options](#) on page 97.

### Connection Consumer or Serial Processing for JMS Clients

The delivery order options above are configured for the JMS IQ Manager. The eGate Integrator JMS implementation enables you to configure JMS clients as connection consumers to improve message throughput through concurrent processing. You can set the JMS client configuration with the **Concurrency** property as described in [“Setting JMS Client Message Processing Order” on page 79](#).

The use of connection consumers increases message processing performance by enabling concurrent processing via multiple threads. You can specify the number of message driven beans (MDBs) or server session pool to assign to a JMS Collaboration to process messages concurrently. When you use connection consumer with fully concurrent or protected concurrent FIFO processing, this setting allows the integration server to assign multiple threads to execute the Collaboration on a particular message destination.

For queues, it is also possible to use connection consumer for concurrent processing on multiple CPUs (and integration servers) on a system.

This configuration does have an effect on FIFO processing. For information, refer to [“JMS Client Concurrency Effect on FIFO Delivery Modes” on page 30](#).

## 2.1.2 Message Priorities

eGate Integrator enables you to set message priorities for JMS clients as well as Collaborations. The default message priority for JMS clients is 4. The JMS client priority level causes all messages produced by the client to have that same priority level. For

example, if you set the priority level to 2, all messages sent by that client have message priority level 2. For information about setting the JMS client message priority level, refer to [“Specifying Message Priorities” on page 80](#).

You can also specify message priorities in Collaborations with the JMS OTD. Collaboration message priorities override JMS client message priorities. For more information, refer to [“Setting JMS Message Properties in Collaborations” on page 69](#).

### 2.1.3 Security

The eGate Integrator provides role-based security for the JMS IQ Manager. To implement security, you assign roles to users. Each role includes a series of permissions that identify what tasks a user is authorized to perform.

JMS IQ Manager security is disabled by default. To enable security, refer to [“Enabling Security” on page 89](#). You then must specify a username and password for each JMS client as described in [“Specifying Security Name and Password” on page 82](#).

### 2.1.4 Distributed Transactions

The JMS IQ Manager enables you to configure JMS properties for distributed transactions (XA). To enable a JMS producer or consumer for XA, set the **Transaction mode** property for the JMS consumer or producer to the XA option as described in [“Specifying the Transaction Mode” on page 81](#).

### 2.1.5 Connection Consumer

The eGate Integrator JMS implementation enables you to configure JMS clients as connection consumers to improve message throughput through concurrent processing. The use of connection consumers increases message processing performance by enabling concurrent processing via multiple threads. For queues, it is also possible to use connection consumer for concurrent processing on multiple CPUs (and integration servers) on a system. To use connection consumer, you enable the **Concurrency** property for the JMS producer or consumer. For information about configuring JMS properties, refer to [“Setting JMS Client Message Processing Order” on page 79](#).

For information about how this configuration affects message processing order, refer to [“JMS Client Concurrency Effect on FIFO Delivery Modes” on page 30](#).

---

## 2.2 About the WebLogic JMS Server

eGate Integrator enables you to set up eGate Projects so that they can be deployed to the BEA WebLogic Platform™ 8.1. To create a WebLogic deployment, you must do the following:

- 1 Upload the **weblogic.sar** and **weblogicjmsmessageserver.sar**.  
For information, refer to [Installing Message Server .sar Files](#) on page 35.



- 2 In your eGate Project, select a WebLogic application server and message server. For information, refer to [Adding Message Servers to Logical Hosts](#) on page 86.
- 3 Deploy the Project as described in the *eGate Integrator User Guide*.

---

## 2.3 About the WebSphere JMS

eGate Integrator enables you to set up eGate Projects so that they can be deployed to the WebSphere Application Servers V5.0.x. To create a WebSphere deployment, you must do the following:

- 1 Upload the `websphereintegserver.sar` and `webspherejmsmessageserver.sar`. For information, refer to [Installing Message Server .sar Files](#) on page 35.
- 2 In your eGate Project, select a WebSphere application server and message server. For information, refer to [Adding Message Servers to Logical Hosts](#) on page 86.
- 3 Deploy the Project as described in the *eGate Integrator User Guide*.

---

## 2.4 Message Servers in the eGate Implementations

eGate Integrator provides an efficient way of designing and deploying your eGate implementations by modeling the design and deployment phases after business workflows. This section describes the eGate Integrator model, and continues on to describe how message servers fit into this model.

### 2.4.1 The eGate Integrator Model

eGate Integrator models the design and deployment phases after business workflows. In broad outline, an eGate Integrator implementation includes the following steps:

- 1 **Designing an eGate Project.**  
The Project is where you create your business processes. For example, you can create different Projects for patient admittance, patient records, and insurance billing.
- 2 **Building an eGate Environment.**  
In the Environment, you specify the systems used, and the configurations for these systems. You can create multiple Environments, such as different Environments for development, testing, staging, and production.
- 3 **Building deployment profiles.**  
Deployment profiles map a business process to a physical environment where the process will be running; that is, deployment profiles map Projects to Environments. You can map a Project to multiple Environments, thus enabling you to run the same Project in different system environments. For example, in the deployment profile, you map a Project for patient admittance to a test Environment. You then deploy to the test systems and make fixes as necessary to either the Project or the

Environment. After that, you can map the Project to the staging Environment using another deployment profile, and then deploy to staging systems.

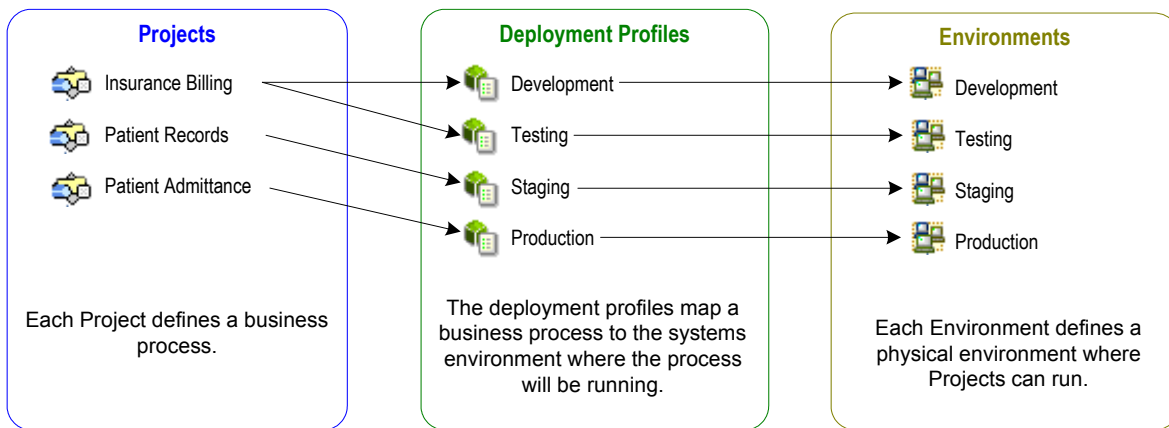
#### 4 Deploying the eGate implementation.

To deploy the eGate implementation, you activate the deployment profile. This creates configuration file that the Repository forwards to Logical Hosts.

This structure of Projects, Environments, and deployment profiles isolates each component into logical and physical components. This provides you with extensive flexibility and efficiency in designing eGate Integrator implementations. For example, once you build your Projects and Environments, you have the flexibility to change each component without having to make changes to the other component.

The figure below shows an example of the eGate Integrator model.

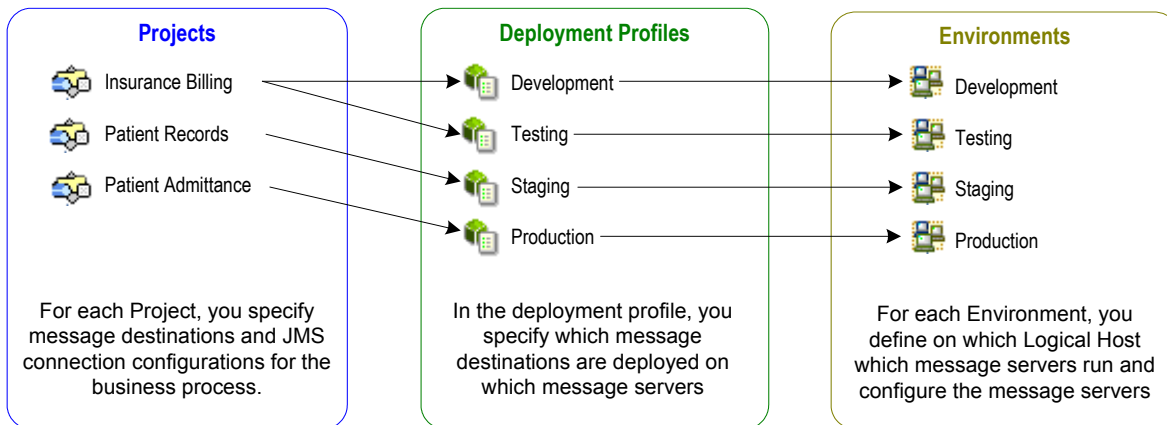
**Figure 1** eGate Integrator Model



In the figure above, any of the Projects can be deployed to any of the Environments via the mapping defined in the deployment profiles. The example in the figure above shows that the patient admittance Project is already in the production phase and therefore was deployed using the production deployment profile. The patient records Project is in the staging phase and was therefore deployed to the staging Environment using the staging deployment profile. The insurance billing Project is still being developed and tested, and therefore it is deployed to development and testing via the development and testing profiles.

The figure below shows how the message server design fits into the eGate Integrator model.

**Figure 2** Message Server in the eGate Integrator Model



The sections below describe in detail how the message server design fits into each step listed above.

## 2.4.2 Project Design Phase: Topics, Queues and JMS Properties

Your first step in incorporating Message Servers in eGate is designing an eGate Project in Enterprise Designer. In an eGate Project, you specify the business logic for the eGate implementation by defining the following items:

- Message destinations (topics or queues)
- Collaboration Definitions (using JMS OTDs)
- Collaborations
- JMS client properties
- External applications (such as eWays)

For each of these components you specify logical properties; these properties are independent from the physical implementation. For the message server-related components, Projects are where you add and name message destinations. You do this by dragging and dropping topics and queue buttons onto the Collaboration Map canvas.

After having added the Connectivity Map components for message destinations and other logical components, you then create the Object Type Definitions (OTDs) and Collaboration Definitions. For any Collaboration that reads or write to a message server you must add the JMS web service.

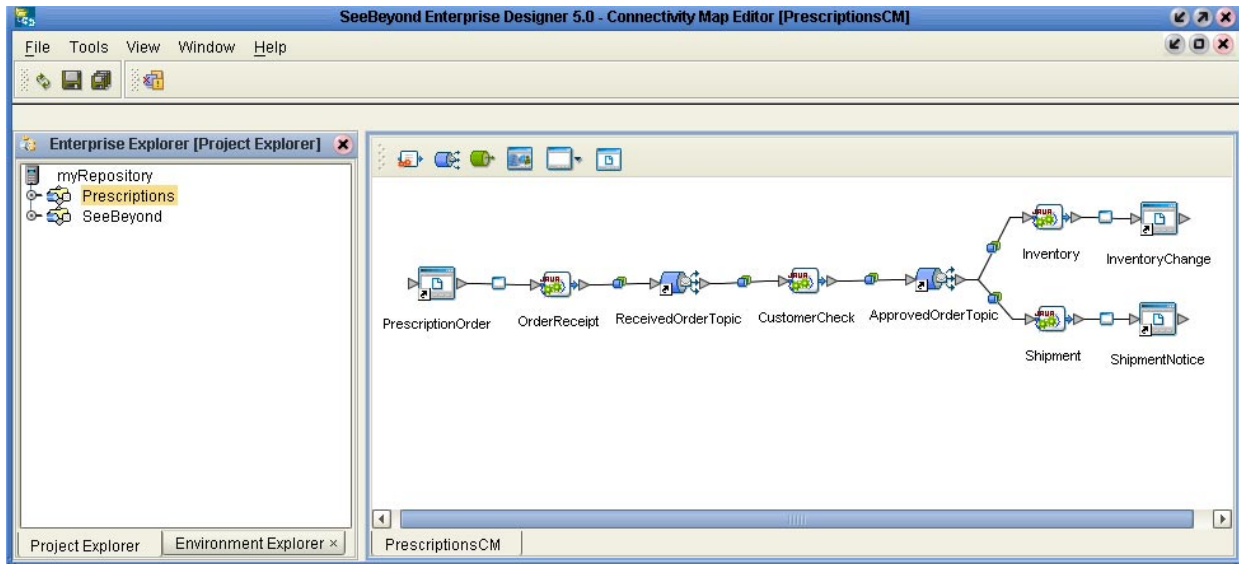
After defining the Collaboration Definitions, you create the relationships between the components by associating Collaboration Definitions with Collaborations. This creates links between the components. On links between message destinations and their subscribers and publishers, there is a JMS client properties icon.

By double-clicking the JMS client properties icon in the Connectivity Map, you can configure the connection for such items as persistent or non-persistent delivery mode, XA, and concurrent processing.

For procedures for this phase, refer to [Chapter 5, “Configuring Collaborations for JMS Web Services” on page 37](#) and [Chapter 6, “Configuring JMS Clients” on page 76](#).

The figure below shows a Project and its Connectivity Map in Enterprise Designer.

**Figure 3** eGate Integrator Projects



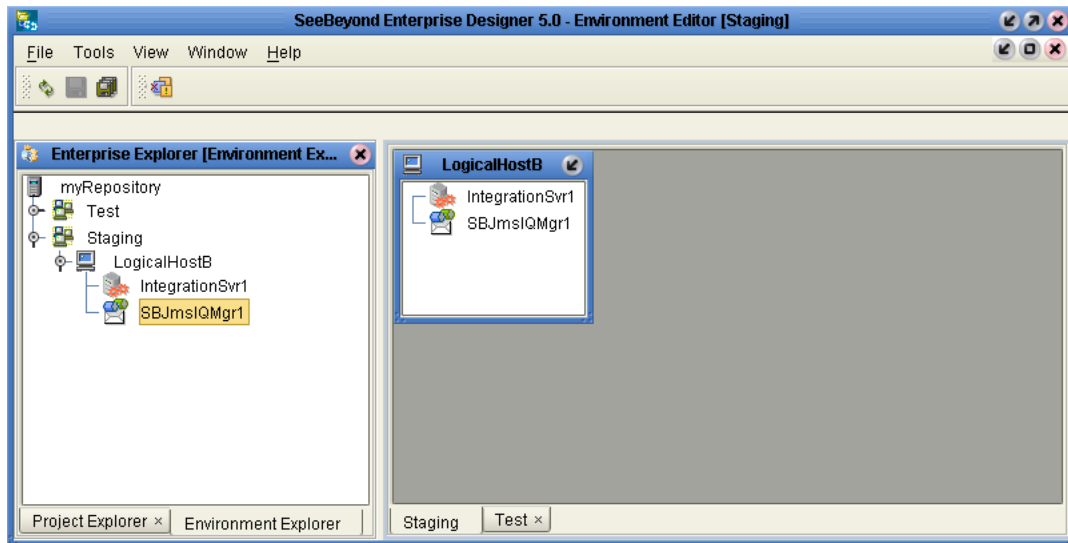
For more information about building Projects, refer to [“Project Design Phase: Topics, Queues and JMS Properties” on page 19](#).

### 2.4.3 Environment Design Phase: Message Server Configurations

In the eGate Environment, you specify which message servers run on which Logical Hosts. Once you add a message server to a Logical Host, you specify the physical configurations for the message server. You can specify many configurations, such as the port number, message delivery order, tuning configurations, journaling options, and diagnostic options. For procedures, refer to [“Adding and Configuring Message Servers” on page 86](#).

The figure below shows an Environment in the Enterprise Designer. The Environment is called **Staging**, under which there is one Logical Host, **LogicalHostB**. This Logical Host will run the JMS IQ Manager **SBjmsIQMgr1**.

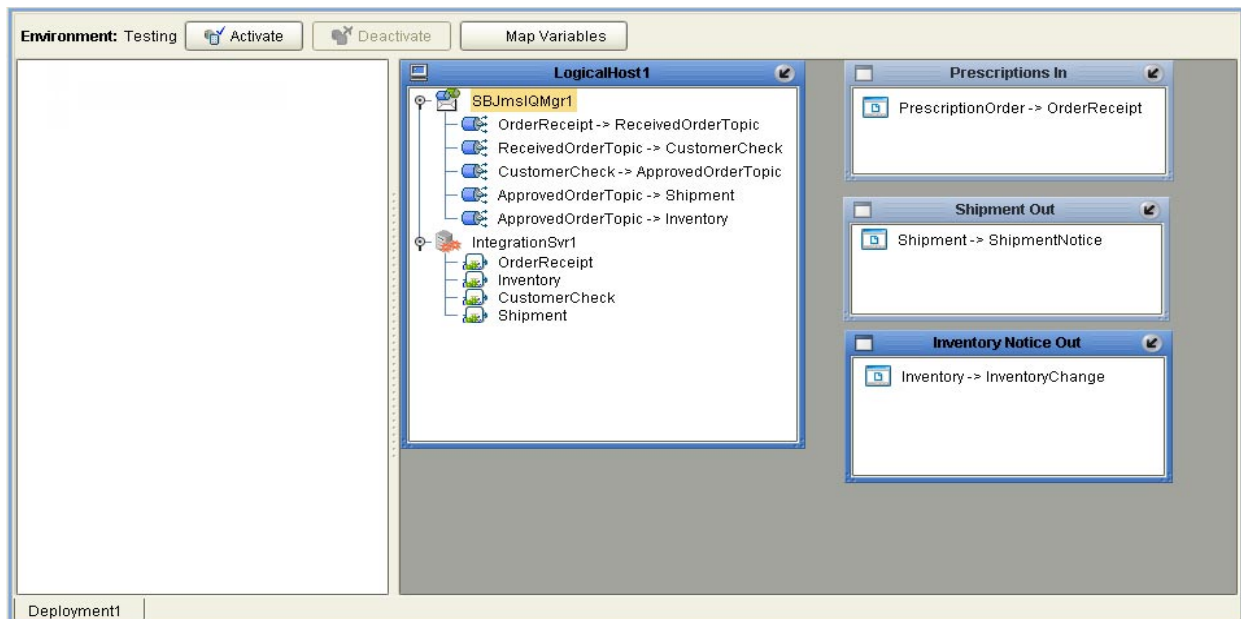
Figure 4 Adding Message Servers to an Environment



## 2.4.4 Deployment Profile Design Phase: Creating Mappings

During the deployment profile design phase, you create mappings between Projects and Environments. In the deployment profile, you specify which components of the business process are located on which systems in a specific Environment, such as which Collaborations run on which integration server, and which eWays run on which external systems. For the message server, you specify which message destinations run on a particular message server. For more details, refer to [“Deploying Topics and Queues to Message Servers” on page 110](#). The figure below shows an example of a deployment profile.

Figure 5 Deployment Profile Example



## 2.4.5 Activating Deployment Profile Phase: Configuration File

When you activate a deployment profile, eGate sends a configuration file from the Repository to the Logical Host.

## 2.4.6 Message Servers at Runtime

Once the eGate Project is deployed, you can manage the JMS IQ Manager and NSJMS with the Enterprise Manager as described in [“Managing JMS IQ Managers with the Enterprise Manager” on page 114](#).

eGate Integrator also provides the STC MS Control utility to manage the JMS IQ Manager. For information, refer to [“Managing JMS IQ Managers with the MS Control Utility” on page 123](#).

To reconfigure an eGate Project after it has been deployed, refer to [“Configuring JMS Clients” on page 76](#) and [“Reconfiguring Message Servers” on page 112](#).

---

## 2.5 Quick Overview: Building eGate Projects with JMS

This section provides a quick overview of the overall process of building and deploying an eGate Project and how JMS fits into this process.

- 1 Install the WebLogic JMS, the WebSphere JMS, or the 4.5.x Message Server if necessary. For information, refer to [“Installing Third-Party and 4.5.x Message Server Support” on page 35](#). The JMS IQ Manager 5.0 is included with eGate Integrator and does not require separate installation.
- 2 Create a Project and Connectivity Map as described in the *eGate Integrator User Guide*.
- 3 Add Project components to the Connectivity Map; this includes adding message destinations as described in [“Adding Message Destinations to Projects” on page 37](#).
- 4 Create OTDs as described in the *eGate Integrator User Guide*.
- 5 Create Collaboration Definitions. For Collaborations that will read or write to the JMS IQ Manager, you must add the JMS Web Service to the Collaboration Definition as described in [“Adding JMS Web Services to Collaboration Definitions” on page 38](#).
- 6 Build Collaboration Definitions. For Collaborations with the JMS Web Service, you use JMS methods to build the JMS business rules as described in [“Using JMS Methods in Collaboration Definitions” on page 41](#).
- 7 Bind Collaboration Definitions to Services as described in [“Binding Collaboration Definitions to Services” on page 72](#).
- 8 Link the Connectivity Map components; this includes connecting JMS Web Service Collaborations to their destinations as described in [“Linking JMS Web Service Collaborations to Destinations” on page 73](#).

- 9 Configure eWays and JMS clients. For JMS client configuration, refer to [“Configuring JMS Clients” on page 76](#).
- 10 Create an eGate Environment and add message servers as described in [“Adding Message Servers to Logical Hosts” on page 86](#).
- 11 Configure the message server as described in [“Configuring Message Servers” on page 88](#).
- 12 Deploy the eGate Project as described in the *eGate Integrator User Guide*.

---

## 2.6 Message Server Configuration and Administration Tools

eGate Integrator includes several tools for you to configure and manage JMS IQ Managers and NSJMS servers:

- Enterprise Designer
- Enterprise Manager
- STC MS Control Utility (JMS IQ Manager only)

The table below describes each tool, its functionality, and where you can find more information about the tool.

**Table 2** JMS IQ Manager Administration Tools

Tool	Description	For More Information
Enterprise Designer	Application that allows you to add message destinations, and add and configure message servers and JMS clients	<a href="#">“Configuring JMS Clients” on page 76</a> and <a href="#">“Adding and Configuring Message Servers” on page 86</a>
Enterprise Manager	Application that allows you to start and stop message servers, view alerts, monitor, view, and edit live messages, for JMS IQ Managers	<a href="#">“Managing JMS IQ Managers with the Enterprise Manager” on page 114</a>
STC MS Control Utility	Command-line tool that allows you to configure and manage JMS IQ Managers	<a href="#">“Managing JMS IQ Managers with the MS Control Utility” on page 123</a>



# Functional Overview of the JMS IQ Manager

This chapter describes in detail the internal workings of the JMS IQ Manager; such as it processes messages, where it stores persistent messages, journaled messages, and for how long.

## In This Chapter:

- [About the JMS IQ Manager Database](#) on page 24
- [Message Processing Order](#) on page 27
- [About Performance](#) on page 31

---

## 3.1 About the JMS IQ Manager Database

The JMS IQ Manager uses the JMS IQ Manager database to store persistent messages. The database is also used to store messages that are larger than can be kept in the JMS IQ Manager memory, which is determined by the cache size setting (by default 0.5 MB for Windows and 1 MB for UNIX). By default, JMS clients are configured for persistent messaging; therefore, in a default configuration, the database is used to store messages. The messages are stored until they are consumed or until duration set for the maximum time to live for a live message expires, which is 30 days by default. Therefore these messages are sometimes referred to as “live” messages.

### 3.1.1 About Segments

The database resides in the message server folder on the Logical Host. The database consists of a number of segments (database files). A segment is a disk space store that is memory-mapped on the server. The segments act together to form the equivalent of a sequential database. By default, these files are named **stcms\*.dbs**.

The JMS IQ Manager creates four segments in the database when it starts up initially. The size of a segment is 8 MB on Windows and 16 MB on UNIX. The JMS IQ Manager creates as many segments as necessary. Before running your eGate Project, *it is important to set the segment size higher than the largest transaction the JMS IQ Manager may need to process*. The JMS IQ Manager cannot accommodate a transaction that is larger than the segment size.



Configurations such as the database filenames, segments size, the maximum and minimum number of segments created can all be specified. For information, refer to [“Setting Stable Storage and Journaling Options” on page 104](#) and [“Setting Performance Options” on page 90](#).

### 3.1.2 About the Database Location

The JMS IQ Manager database resides in the directory specified with the JMS IQ Manager **Data directory** property as described in [“Specifying the Location of JMS IQ Manager Database Files” on page 105](#). By default, the database resides in the following directory:

*logicalhostname\logicalhost\stcms\JMSIQMgrname*

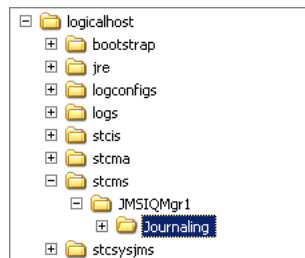
where *logicalhostname* is the name of the Logical Host and where *JMSIQMgrname* is the name of the JMS IQ Manager.

The JMS IQ Manager database consists of database files called segments. For more information about segments, refer to the previous section.

If journaling is enabled, the data directory contains a **Journaling** directory, unless another location has been specified for the **Journal directory** property. The Journaling directory holds the journaling database files. For information, refer to [“Specifying the Location of the Journal Database” on page 107](#). Journaling is disabled by default.

The figure below shows the directory structure for a data root defined as .. as in the default setting (one directory up).

**Figure 6** JMS IQ Manager Database Files Location—Data Directory



### 3.1.3 The Default Database Configuration

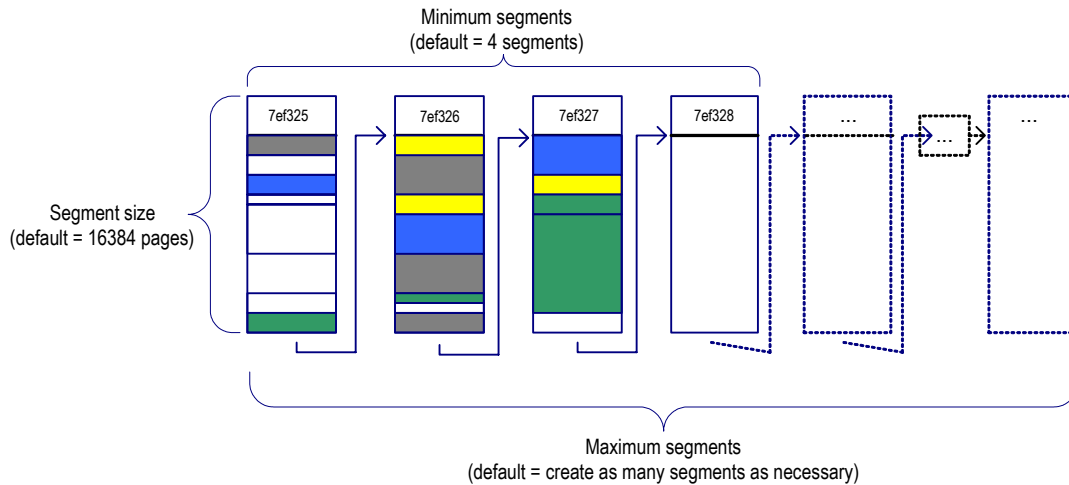
The default configuration for the JMS IQ Manager database is as follows:

- The database resides in the following directory:  
*logicalhostname\logicalhost\stcms\JMSIQMgrname*
- The segment filenames are **stcms\*.dbs**.
- The number of segments created initially for the database is four.
- The size of each segment is 8 MB on Windows and 16 MB on UNIX.
- The maximum number of segments that can be created is infinite.

This means that when the JMS IQ Manager starts up, the database consists of four segments; the total size of the database is 24 MB on Windows and 64 MB on UNIX.

The figure below shows a sample JMS IQ Manager database:

**Figure 7** JMS IQ Manager Database Structure



On startup, the JMS IQ Manager performs the following operations:

- 1 It allocates sufficient disk space to hold the minimum number of segments.  
Figure 7 shows a JMS IQ Manager allocation of four segments, numbered 7ef325 through 7ef328.
- 2 As messages arrive, they are appended to the first segment until the segment is full. When a segment is full, the JMS IQ Manager stores subsequent messages in the first free segment.  
Figure 7 shows that the third segment, 7ef327 in file `stcms7ef327.dbs`, is almost full.
- 3 If there is no free segment, the JMS IQ Manager allocates a new segment if possible.
- 4 When a message has been delivered to all consumers:
  - ♦ For a transacted message, the message is dequeued immediately and its slot is marked eligible after acknowledgement; but
  - ♦ For an XA message, the message is dequeued (and its slot marked eligible) only after all consumers have committed it.
- 5 When all messages in a segment have expired or been dequeued, the JMS IQ Manager cleans up the segment, freeing it for re-use.

In Figure 7, the first segment (7ef325) has several segments that are white, indicating the slot is marked eligible. The segment is therefore almost ready for cleanup.

## 3.2 Message Processing Order

There are several ways to control the message processing order in eGate Projects:

- You can specify first-in, first out (FIFO) ordering modes when you configure the JMS IQ Manager.  
*For information see the section below.*
- You can also specify message processing at the JMS client level (connection consumer or serial mode). This affects FIFO processing in several ways.  
*For information, see “[JMS Client Concurrency Effect on FIFO Delivery Modes](#)” on [page 30](#).*
- You can specify a set of message destinations with a specific processing order.  
*For information, see “[Serial Processing Across a Destination Group](#)” on [page 31](#).*

### 3.2.1 JMS IQ Manager FIFO Delivery Modes

You can specify several FIFO delivery modes for eGate Projects; these ordering modes apply to all message destinations in the eGate Project. The FIFO modes are:

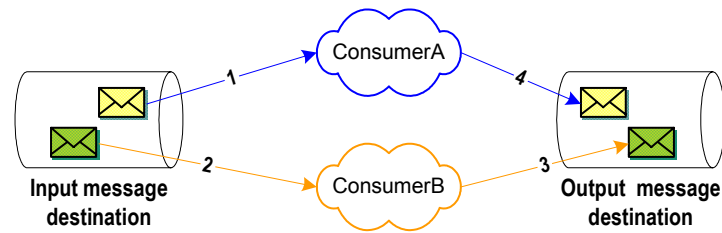
- *Fully concurrent first-in, first out (FIFO) order*  
Consumers can retrieve messages when all older messages have been received or are being received and can commit messages in any order (without using time sequence).
- *Protected concurrent FIFO order*  
Consumers can retrieve messages when all older messages have been received or are being received, but must commit using time sequence.
- *Fully serialized FIFO order*  
Consumers read a messages only after all messages have been received and commit messages using time sequence.

Setting the FIFO delivery mode only comes into play when multiple consumers or multiple processes within a single consumer subscribe to the same message destination. For a single consumer with a single process, processing is by default fully serialized.

### Fully Concurrent Processing

In fully concurrent mode, consumers can retrieve messages from a destination only when all older messages have been received or are in the process of being received. Consumers can then commit messages without restrictions. By default, JMS IQ Managers use fully concurrent processing.

**Figure 8** Fully Concurrent Processing



The figure above shows a sample delivery sequence for fully concurrent processing. In step 1 and 2, the consumers retrieve their messages from the input queue. Both consumers must wait until each consumer has retrieved its messages (or is in the process of retrieving) before being able to commit messages to the output destination. As step 3 and step 4 indicate, consumers can commit messages in any order.

This means that the messages can be committed out of sequence, for example, a cancellation may be committed before the order is committed. This type of delivery order is not always desirable. The table below shows the benefits and costs of fully concurrent processing.

**Table 3** Benefits and Costs—Fully Concurrent Processing

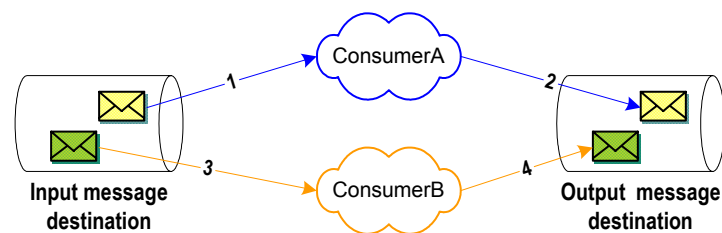
Benefits	Costs
Provides the best performance	Delivery not time sequenced
Consumers are not hampered by other consumers	

You specify fully concurrent processing for JMS IQ Managers with the **Special FIFO modes** property as described in [“Specifying Special FIFO Modes” on page 98](#).

## Protected Concurrent Processing

In protected concurrent mode, consumers retrieve messages just as in fully concurrent mode (after all messages have been received or are being received), but messages can only be committed if all older messages have been committed.

**Figure 9** Protected Concurrent Processing



The figure above shows a sample delivery sequence for protected concurrent processing. In step 1 and 2, the consumers retrieve their messages from the input queue. Both consumers must wait until each consumer has retrieved its messages (or is in the process of retrieving) before being able to commit messages to the output

destination. ConsumerB might be ready to commit its message before ConsumerA, but must wait until ConsumerA commits its message (step 3). Only when ConsumerA’s message has been committed, can ConsumerB commit its message (step 4).

Protected concurrent processing thus is a more workable solution in a scenario where a Project deals with messages such as orders and cancellations, where the order must be committed before the cancellation.

The table below shows the benefits and costs of protected concurrent processing.

**Table 4** Benefits and Costs—Protected Concurrent Processing

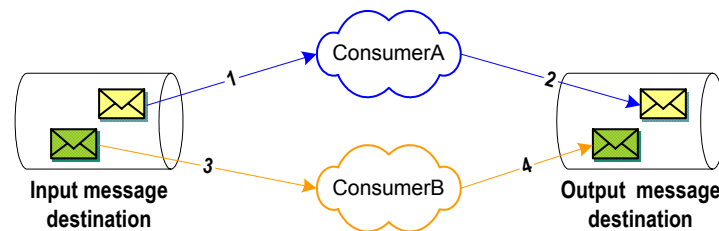
Benefits	Costs
Provides better performance than serialized processing	Provides less performance than fully concurrent processing
Messages are delivered by time sequence	

You specify protected concurrent processing for JMS IQ Managers with the **Special FIFO modes** property as described in **“Specifying Special FIFO Modes” on page 98**.

## Fully Serialized Processing

In fully serialized mode, consumers can only retrieve messages after all older messages have been received *and* committed.

**Figure 10** Fully Serialized Processing



The figure above shows a sample delivery sequence for serialized processing. In step 1, ConsumerA retrieves its message. ConsumerB might at this point be ready to receive its message, but must wait until ConsumerA has committed its message. After ConsumerA commits the message in step 2, ConsumerB can then retrieve and commit its message (steps 3 and 4).

The table below shows the benefits and costs of protected concurrent processing.

**Table 5** Benefits and Costs—Fully Serialized Processing

Benefits	Costs
Guaranteed delivery by time sequence	Provides the least performance of all FIFO modes

You specify fully serialized processing for JMS IQ Managers with the **Special FIFO modes** property as described in **“Specifying Special FIFO Modes” on page 98**.

### 3.2.2 JMS Client Concurrency Effect on FIFO Delivery Modes

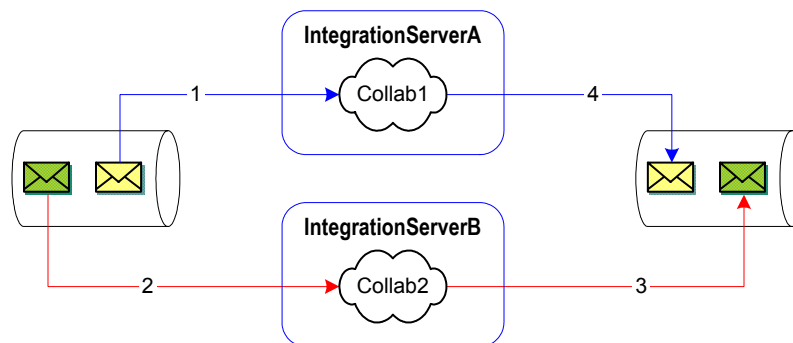
Apart from specifying FIFO delivery modes for JMS IQ Managers as described above, you can also configure JMS clients for concurrency: connection consumer or serial mode. To do this, you use the JMS client **Concurrency** property as described in **“Setting JMS Client Message Processing Order” on page 79**.

The use of connection consumers increases message processing performance by enabling concurrent processing via multiple threads. You can specify the number of message driven beans (MDBs) or server session pool to assign to a JMS Collaboration to process messages concurrently. When you use connection consumer with fully concurrent or protected concurrent FIFO processing, this setting allows the integration server to assign multiple threads to execute the Collaboration on a particular message destination.

For queues, it is also possible to use connection consumer for concurrent processing on multiple CPUs (and integration servers) on a system.

Using JMS client connection consumer does affect on message processing order. For example, consider the scenario shown below. The JMS IQ Manager is set to fully concurrent FIFO processing. However, each Collaboration on each integration server retrieves messages as they come in, and is able to commit them unrestricted to the queue. Therefore, although the JMS IQ Manager is configured for fully concurrent FIFO processing, message order cannot be guaranteed.

**Figure 11** Multiple Integration Server Configuration



The table below lists how the connection consumer and serial mode setting affect the JMS IQ Manager FIFO selection for topics. For topics, only one integration server per subscriber can be used.

**Table 6** JMS Client Concurrency Effect on FIFO modes—Topic

FIFO mode	Fully serialized	Protected concurrent	Fully concurrent
<b>Concurrency mode</b>			
serial mode	fully serialized no concurrency	protected concurrent no concurrency	fully serialized
connection consumer	fully serialized no concurrency	protected concurrent concurrent	no strict order maintained

The table below lists the effect of JMS client concurrency on queues for configurations with single or multiple integration servers.

**Table 7** JMS Client Concurrency Effect on FIFO modes—Queue

FIFO mode	Fully serialized	Protected concurrent	Fully concurrent
<b>Concurrency mode</b>			
<i>1 integration server</i>			
serial mode	fully serialized no concurrency	protected concurrent no concurrency	fully serialized no concurrency
connection consumer	fully serialized no concurrency	protected concurrent concurrent	no strict order maintained concurrent
<i>&gt;1 integration servers</i>			
serial mode	fully serialized no concurrency	protected concurrent concurrent	no strict order maintained concurrent
connection consumer	fully serialized no concurrency	protected concurrent concurrent	no strict order maintained concurrent

### 3.2.3 Serial Processing Across a Destination Group

You can also specify delivery order specifically for a set of topics and queues (a time order group). For this group, consumers can only receive messages when all other messages in the time order group have been received or are in the process of being received. For information, refer to [Setting Delivery Ordering Options](#) on page 97.

---

## 3.3 About Performance

Because of the large assortment of configuration parameters, you have a high degree of control over speed, memory, and space. The configuration parameters work together to allow you to fine-tune your system according to load and hardware constraints.

Because every message is written to disk, file input/output (I/O) is usually the hardware factor with the largest performance impact. For a disk with adequate I/O speed, fastest performance is achieved by holding all messages in server memory continuously until the corresponding segment is cleaned up.

Because available server memory can easily be exceeded for systems handling very large messages, there are several configuration parameters to help you manage a memory-bound server; see [“Managing Resources” on page 32](#).

The segment size must always be larger than the largest expected message—preferably by an order of magnitude. The segment size is expressed in bytes. For Windows, the size of a segment is 512 bytes per page; for UNIX, it is 1024. The size of a segment is calculated as the number of pages multiplied by the bytes per page.

To maximize performance:

- Use the fastest disk possible.
- Keep as many messages in server memory as possible.
- Allocating a new segment requires more time than freeing a cleaned-up segment.
- Smaller segments turn over more rapidly and thus provide more effective use of server memory. However, because cleaning up two small segments requires more time than cleaning up one large segment, you can use very large segments to increase performance on systems that are constrained by disk I/O speed rather than memory or space.

### 3.3.1 Managing Resources

As for any server, there are configurations purely for managing disk access memory (such as the **Max payload memory** and **Payload memory recovery amount** properties) and memory/disk trade-offs (such as the **Cache size** property for controlling read memory, the **Enabling cache control** property on UNIX for increasing speed over reliability, and the **Lock cache into RAM** property on Windows for reducing page faults). In addition, there are special configuration properties that specifically deal with messages, message destinations, and producers. Three of these are discussed in detail below.

#### Throttling Producers

When the amount of JMS IQ Manager memory allocated to messages reaches a certain limit, the JMS IQ Manager can be instructed to stop reading all messages from one or more publishers until a certain criterion is met. This process is called “throttling” the producer.

Producer throttling is done on a per-message destination basis. This caters to the most common reasons for approaching JMS IQ Manager memory limit in an otherwise well-tuned system:

- A particular message destination has a period of abnormally heavy traffic. Throttling all producers of such a message destination gives the destination’s consumer a chance to catch up while maintaining normal throughput for other message destinations.



- A particular consumer fails, causing a backup of all message destinations it subscribes to. If the consumer problem is transient and normal, then throttling all its producers gives it “breathing room” to catch up on the backlog. And, if the consumer problem is serious, then throttling its producers allows unaffected message destinations to flow freely while the problem can be diagnosed and repaired without taking the server offline.

Three configuration properties govern producer throttling:

- **Throttling threshold (total messages)** sets the JMS IQ Manager limit. When the JMS IQ Manager is below this threshold, it does not throttle any producers even if they are eligible for throttling.
- When producer throttling is in effect, **Throttling threshold (per queue/topic)** sets the per-topic limit. The JMS IQ Manager stops reading messages from producers of any message destination that has exceeded this limit. It resumes reading messages for the message destination only when one or both of the following criteria are met:
  - ♦ The JMS IQ Manager falls below the **Throttling threshold (total messages)** threshold; or
  - ♦ The message destination has so few messages that the JMS IQ Manager can stop throttling its producers.
- **Throttling lag** determines how many messages for this topic must be dequeued before throttling can stop.

**Note:** *Each message in a message destination counts against the message destination’s **Throttling threshold (per queue/topic)** limit until the message is dequeued. In particular: A non-transactional message is counted until it has been delivered to all its subscribers; a transactional or XA-compliant message is counted until it has been committed by all consumers.*

**Note:** *When a Collaboration sends messages to a JMS IQ Manager faster than the Collaboration receives messages from the same JMS IQ Manager, a deadlock occurs. To avoid this situation, do not use the JMS IQ Manager for long-term storage and change the **Throttling threshold (total messages)** property.*

### Example of Producer Throttling and Unthrottling

Table 8 illustrates a scenario where a JMS IQ Manager becomes loaded and starts to compensate by using producer throttling. The JMS IQ Manager uses default values for throttling properties:

- **Throttling threshold (total messages)**=10,000
- **Throttling threshold (per queue/topic)**=1,000
- **Throttling lag**=100

Two minutes later, this affects Topic\_A, which has two subscribers and one publisher: Its publisher is throttled for three minutes, until the number of undelivered messages can drop below 900. Later, because the JMS IQ Manager is no longer loaded, the same topic is allowed build up an even greater backlog without having its publisher throttled.

**Table 8** Publisher Throttling

Time	For Server: Total messages on all topics	For messages in Topic_A (only): The highest sequence number			Comment
		read from Pub1:	sent to Sub1:	sent to Sub2:	
11:37	98604	500	200	75	Server is not yet loaded.
11:38	100307	800	500	150	Server is loaded, but Topic_A is unaffected—its subscribers are keeping up well enough.
11:39	101283	1100	800	225	Server still loaded, Topic_A still unaffected—only 875 undelivered messages.
11:40	103429	1350	1050	300	Topic_A has crossed the limit now that it has 1050 undelivered messages; while the server remains loaded, Pub1 will stay throttled until the number of undelivered messages falls below 900.
11:41	104031	1350	1300	375	Pub1 is throttled; Sub1 is nearly caught up; Sub2 is catching up, but has 975 undelivered messages.
11:42	103204	1350	1350	449	Pub1 is throttled; Sub1 has caught up; Sub2 has 901 undelivered messages — still too many.
11:43	102762	1350	1350	451	Although server is still loaded, it unthrottles Pub1 now that the undelivered message count for Topic_A has fallen below 900.
11:44	101095	1375	1370	525	Server is loaded, but Topic_A is unaffected — it has only 850 undelivered messages.
11:45	100028	1575	1500	600	Server is loaded, but Topic_A is unaffected — it has only 975 undelivered messages.
11:46	99248	1900	1700	675	Server is no longer loaded; no publishers are throttled even though Sub2 has more than 1000 undelivered messages.

# Installing Third-Party and 4.5.x Message Server Support

To be able to design eGate Projects with third-party message servers such as WebSphere JMS and WebLogic JMS, you must install separate .sar files. If you do not install these files, the options to create third-party message servers will be unavailable in the Enterprise Designer.

If you need to connect to a 4.5.x message server, you must also upload a separate sar file to be able to design your Project in Enterprise Designer with the 4.5.x message server.

This chapter describes how you install message server .sar files.

## In This Chapter

- [“Installing Message Server .sar Files” on page 35](#)
- [“Additional Instructions for WebLogic” on page 36](#)

---

## 4.1 Installing Message Server .sar Files

During the eGate Integrator installation process, the Enterprise Manager, a web-based application, is used to select and upload products as .sar files from the eGate installation CD-ROM to the Repository.

### To install message server .sar files

For details for the steps below, refer to the *SeeBeyond ICAN Suite Installation Guide*.

- 1 Install and start the Repository.
- 2 Upload **license.sar** to the Repository.
- 3 Upload **eGate.sar** and other product .sar files to the Repository.
- 4 Download components (such as Enterprise Designer and Logical Host).
- 5 Click any **Browse** button on the Enterprise Manager **Upload** page and upload the following files:
  - ♦ To use WebLogic JMS, upload **weblogic.sar** and **weblogicjmsmessageserver.sar**.
  - ♦ To use WebSphere JMS, upload **websphereintegserver.sar** and **webspherejmsmessageserver.sar**.

- ♦ To use the 4.5.x message server, upload **stcmessageserver45x.sar**.

---

## 4.2 Additional Instructions for WebLogic

If you are using the WebLogic JMS, you must download additional .jar files and change the classpath as described below. To use the WebLogic integration server, you must download additional files as described in the *eGate Integrator User Guide*.

### To download additional .jar files

- 1 Download **log4j.jar** to the `\weblogic81\server\lib` directory. You can acquire **log4j.jar** from the location below. This location may change.

<http://jakarta.apache.org/log4j/docs>

- 2 Download **xerces.jar** to the `\weblogic81\server\lib` directory. You can acquire **xerces.jar** from the location below. This location may change.

<http://xml.apache.org/dist/xerces-j>

### To change the classpath

- 1 Open the following file in an editor:

`\weblogic81\server\bin\startWLS.cmd`

- 2 Change the `set CLASSPATH` segment as follows:

```
set CLASSPATH=%JAVA_HOME%\lib\tools.jar;%WL_HOME%\server\lib
\weblogic_sp.jar;%WL_HOME%\server\lib\weblogic.jar;
%WL_HOME%\server\lib\log4j.jar;%WL_HOME%\server\lib\xerces.jar;
%CLASSPATH%
```

Where you add

`%WL_HOME%\server\lib\log4j.jar;%WL_HOME%\server\lib\xerces.jar` to the classpath.

---

## 4.3 Additional Instructions for WebSphere

If you are using the WebSphere JMS, you must download the **log4j.jar** file. To use the WebSphere application server, you must add additional files as described in the *eGate Integrator User Guide*.

### To download log4j.jar

- Download **log4j.jar** to the `\WebSphere\AppServer\lib` directory. You can acquire **log4j.jar** from the location below. This location may change.

<http://jakarta.apache.org/log4j/docs>

# Configuring Collaborations for JMS Web Services

This chapter describes how to configure Collaborations for JMS Web services, and how to add message destinations (topics and queues) to Projects in the Enterprise Designer.

### In This Chapter

- [Adding Message Destinations to Projects](#) on page 37
- [Adding JMS Web Services to Collaboration Definitions](#) on page 38
- [Using JMS Methods in Collaboration Definitions](#) on page 41
- [Examples of Building JMS Collaboration Definitions](#) on page 59
- [Setting JMS Message Properties in Collaborations](#) on page 69
- [Binding Collaboration Definitions to Services](#) on page 72
- [Linking JMS Web Service Collaborations to Destinations](#) on page 73

---

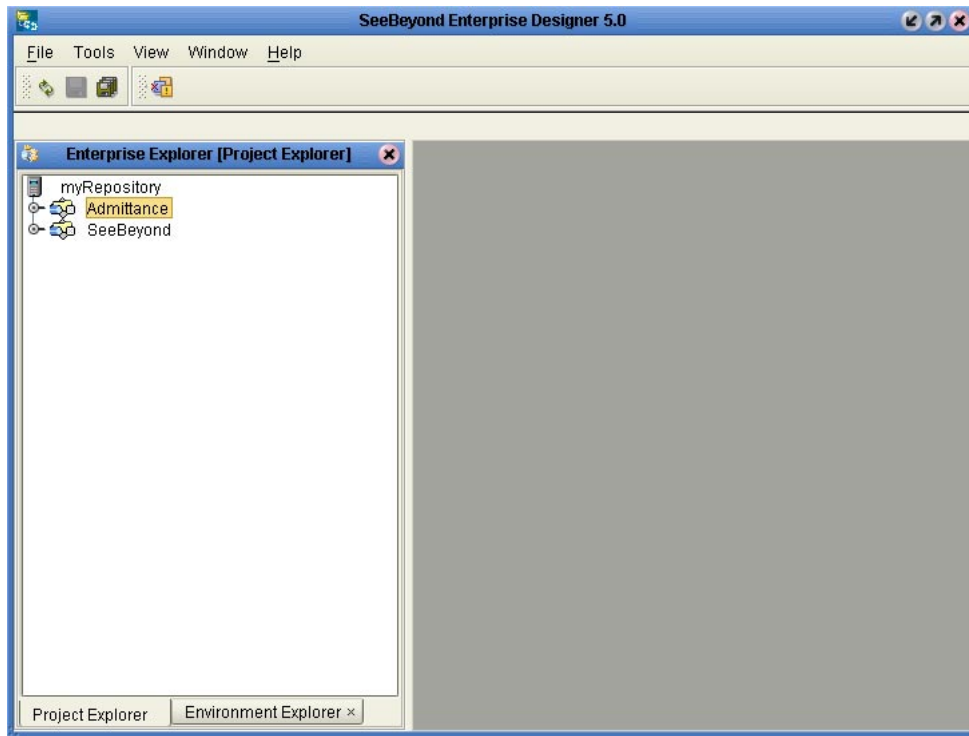
## 5.1 Adding Message Destinations to Projects



To start implementing JMS in an eGate Project, you add topics and queues to the Connectivity Map as described below.

### To add message destinations to Projects

- 1 In the **Project Explorer** tab of the Enterprise Designer, create or click the Project to which you want to add topics or queues.

Figure 12 Selecting a Project



- 2 Create or click a Connectivity Map and drag the **Topic**  or **Queue**  buttons from the Connectivity Map toolbar onto the canvas.
- 3 To rename message destinations, right-click the **Topic** or **Queue** button, click **Rename**, and enter a new name. Do not use colons or semicolons in the message destination names.

You can also create topics and queues in the Enterprise Explorer window by right-clicking the Project for which you want to create topics or queues, clicking **New**, and clicking **Topic** or **Queue**. You can drag these icons to the Connectivity Map.

Now that you have created the Project components, you can create the Collaboration Definitions (the business logic for Collaborations) as described in the *eGate Integrator User Guide*. For Collaborations that read or write to JMS, you must add JMS Web Services to the Collaboration as described in the next section.

---

## 5.2 Adding JMS Web Services to Collaboration Definitions

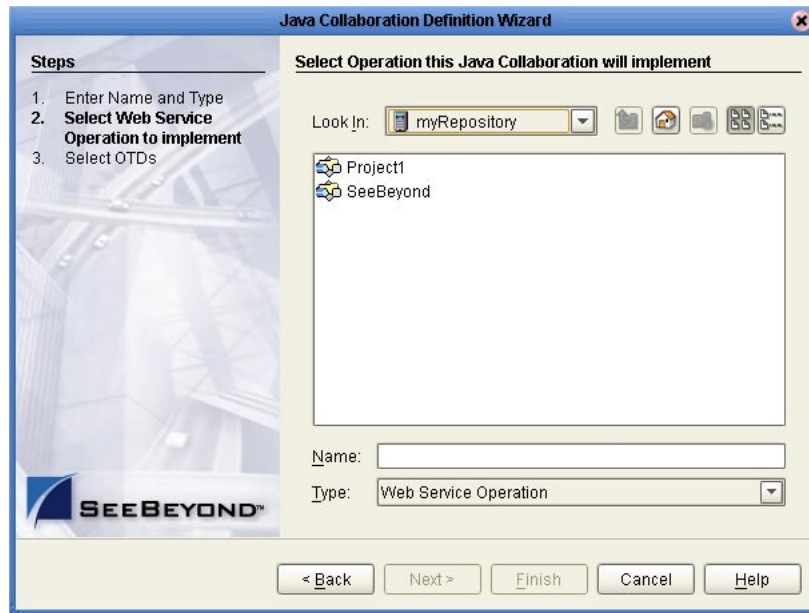
For Collaborations that read from and write to topics or queues, you must add the JMS Web Service to the Collaboration Definition. This indicates to the Collaboration from which message destination it expect to receive messages, and allows you to build the JMS business rules.

The procedure below describes how to add JMS Web Services to Collaboration Definitions.

### To add JMS Web Services to Collaboration Definitions

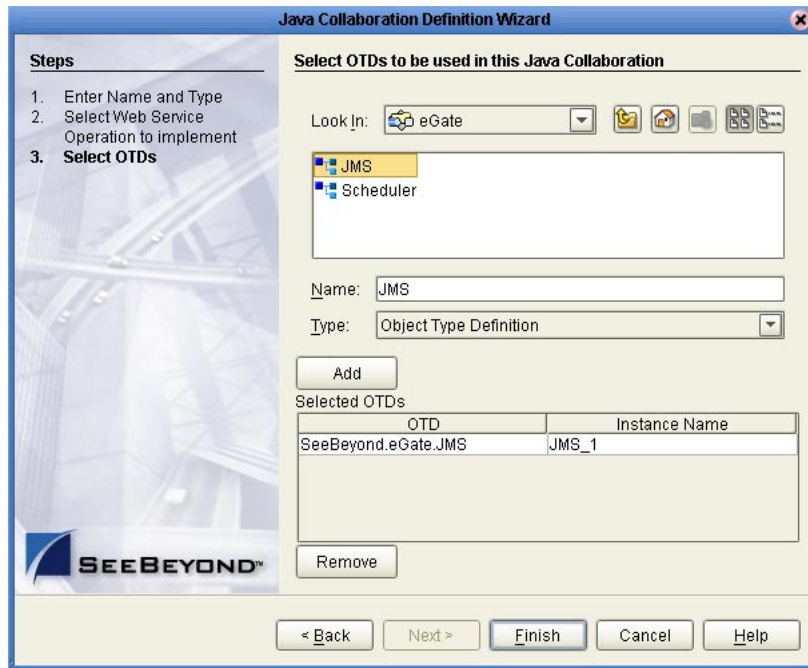
- 1 In the **Project Explorer** tab of the Enterprise Designer, right-click the Project to which you want to add a Collaboration Definition.
- 2 Click **New** and click **Java Collaboration Definition**. The **Java Collaboration Definition Wizard—Step 1** dialog box appears.
- 3 Enter the name for the Collaboration in the **Collaboration Name** box and click **Next**. The following dialog box appears.

**Figure 13** Java Collaboration Definition Wizard—Step 2 Dialog Box



- 4 Double-click **SeeBeyond**, **eGate**, and **JMS** to open the JMS folder. This folder contains the *Receive* method.
- 5 Click **Receive** and click **Next**. The **Java Collaboration Definition Wizard—Step 3** dialog box appears.
- 6 Double-click **SeeBeyond**, **eGate**, and **JMS**. This adds the JMS OTD to the Collaboration Definition as shown below.

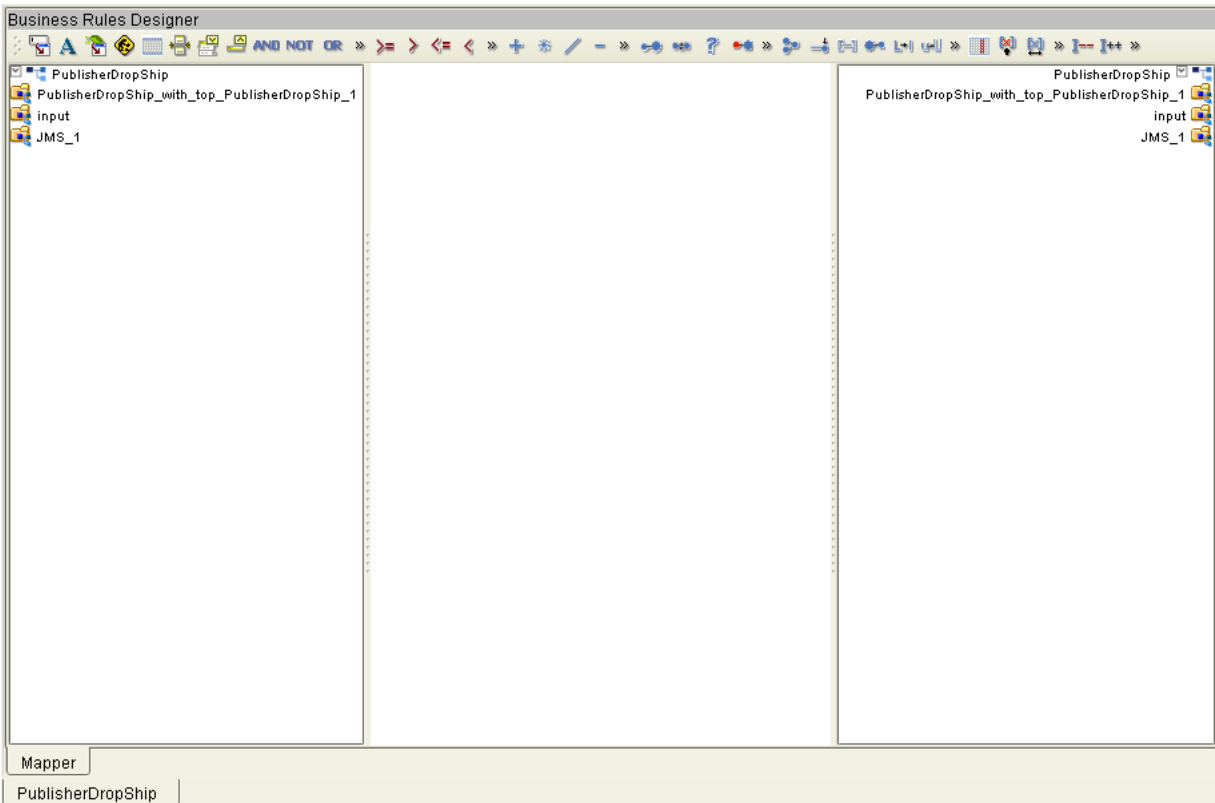
**Figure 14** Java Collaboration Definition Wizard—Adding the JMS OTD



- 7 Add any other OTDs necessary for this Collaboration Definition.
- 8 Click **Finish**. The Java Collaboration Editor displays the Collaboration similar to the figure below.



**Figure 15** Java Collaboration Editor—Collaboration with JMS OTD



The JMS\_1 OTD provides several JMS methods that enable you to build the JMS functionality for the Collaboration. For information about using the JMS methods, refer to the section below. For an example of how to create, define, and send messages, refer to [“Sending Text Messages with the sendTo Method” on page 60](#).

## 5.3 Using JMS Methods in Collaboration Definitions

When the JMS OTD is selected for a Collaboration Definition as described in the section above, several JMS methods are provided in the Java Collaboration Editor. This section provides a list of the JMS methods and some examples of how to build business rules with the JMS methods. There are two types of JMS methods available:

- [JMS Methods Available for JMS OTDs](#) on page 41
- [JMS Methods Available for JMS Messages](#) on page 54

### 5.3.1 JMS Methods Available for JMS OTDs

Collaboration Definitions that use the JMS OTD as described in [“Adding JMS Web Services to Collaboration Definitions” on page 38](#) have the several JMS methods available. The table below lists the JMS methods available for the JMS OTD. For all

other OTDs, the JMS methods are listed in [“JMS Methods Available for JMS Messages” on page 54](#).

---

## createBytesMessage()

### Syntax

```
createBytesMessage()
```

### Description

Creates a byte message.

### Parameters

None.

### Return Value

Returns	Type
The byte message object.	com.stc.connectors.jms.Message

### Exceptions

None.

---

## createBytesMessage(msg)

### Syntax

```
createBytesMessage(msg)
```

### Description

Creates a byte message with the specified byte array value.

### Parameters

Name	Type	Description
<i>msg</i>	byte[]	byte array value for the bytes message

### Return Value

Returns	Type
The message object with data <i>msg</i> .	com.stc.connectors.jms.Message

### Exceptions

None.

## createMapMessage()

### Syntax

```
createMapMessage()
```

### Description

Creates a map message.

### Parameters

None.

### Return Value

Returns	Type
The map message object.	com.stc.connectors.jms.Message

### Exceptions

None.

## createMessage(msg)

### Syntax

```
createMessage(msg)
```

### Description

Creates a message of this type of variable.

### Parameters

Name	Type	Description
<i>msg</i>	javax.jms.Message	The message variable type to be created.

### Return Value

Returns	Type
The message object, with data from JMS message object.	com.stc.connectors.jms.Message

Return t (Message).

### Exceptions

Throws JMS Exception, IOException.

---

## createStreamMessage()

### Syntax

```
createStreamMessage()
```

### Description

Creates a stream message.

### Parameters

None.

### Return Value

Returns	Type
The stream message object.	com.stc.connectors.jms.Message

### Exceptions

None.

---

## createTextMessage(msg)

### Syntax

```
createTextMessage(msg)
```

### Description

Creates a text message that includes the specified text.

### Parameters

Name	Type	Description
<i>msg</i>	String	The string to populate the <i>msg</i> object with.

### Return Value

Returns	Type
The message object with data <i>msg</i> .	com.stc.connectors.jms.Message

### Exceptions

None.

---

## createTextMessage()

### Syntax

```
createTextMessage()
```

### Description

Creates a text message.

### Parameters

None.

### Return Value

Returns	Type
The text message object.	com.stc.connectors.jms.Message

### Exceptions

None.

## receive(*timeout*, *destination*)

### Syntax

```
receive(timeout, destination)
```

### Description

Receives the next message that arrives within the specified timeout interval from the message destination specified.

### Parameters

Name	Type	Description
<i>timeout</i>	long	The number of milliseconds before the <i>receive</i> method times out.
<i>destination</i>	java.lang.String	The name of the topic or queue from which this method receives messages.

### Return Value

Returns	Type
The next message produced, or null if there is no message available.	com.stc.connectors.jms.Message

### Exceptions

Throws JMSEException.

## receive(*timeout*)

### Syntax

```
receive(timeout)
```

### Description

Receives the next message that arrives within the specified timeout interval.

### Parameters

Name	Type	Description
<i>timeout</i>	long	The number of milliseconds before the <i>receive</i> method times out.

### Return Value

Returns	Type
The next message produced, or null if the timeout expires.	com.stc.connectors.jms.Message

### Exceptions

Throws JMSEException.

## receiveNoWait(*destination*)

### Syntax

```
receiveNoWait(destination)
```

### Description

Receives the next message from the specified message destination if a message is immediately available.

### Parameters

Name	Type	Description
<i>destination</i>	java.lang.String	The name of the topic or queue from which this method receives messages.

### Return Value

Returns	Type
The next message produced, or null if the timeout expires.	com.stc.connectors.jms.Message

### Exceptions

Throws JMSEException.

## receiveNoWait()

### Syntax

```
receiveNoWait()
```

### Description

Receives the next message if one is immediately available.

### Parameters

None.

### Return Value

Returns	Type
The next message produced, or null if the timeout expires.	com.stc.connectors.jms.Message

### Exceptions

Throws JMSEException.

## send(message, persistence, priority, timetolive)

### Syntax

```
send(message, persistence, priority, timetolive)
```

### Description

Sends a persistent or nonpersistent message with the message priority and time to live specified.

### Parameters

Name	Type	Description
<i>message</i>	com.stc.connectors.jms.Message	The message variable to be created.
<i>persistence</i>	int	The delivery mode; 2 indicates persistent messages or 1 indicates nonpersistent messages.
<i>priority</i>	int	The message priority (0 through 9, with 9 being the highest priority).
<i>timetolive</i>	long	The amount in milliseconds before the message expires.

### Return Value

None.

### Exceptions

Throws JMSEException if a message is null or if the JMS provider fails to send the message due to an internal error.

---

## send(*message*)

### Syntax

`send(message)`

### Description

Sends a message using the JMS session's default settings for message priority, time to live specified, and delivery mode (persistent/nonpersistent).

### Parameters

Name	Type	Description
<i>message</i>	com.stc.connectors.jms. Message	The message variable type to be sent.

### Return Value

None.

### Exceptions

Throws JMSEException if a message is null or if the JMS provider fails to send the message due to an internal error.

---

## sendBytes(*payload*)

### Syntax

`sendBytes(payload)`

### Description

Sends a byte message of the specified byte array value.

### Parameters

Name	Type	Description
<i>payload</i>	<i>byte[]</i>	The message byte array value.

### Return Value

None.

### Exceptions

Throws JMSEException if a message is null or if the JMS provider fails to send the message due to an internal error.



---

## **sendBytes(payload, persistence, priority, timetolive)**

### Syntax

```
sendBytes(payload, persistence, priority, timetolive)
```

### Description

Sends a persistent or nonpersistent byte message of the byte array value, message priority, and time to live specified.

### Parameters

Name	Type	Description
<i>payload</i>	<i>byte[]</i>	The byte array value.
<i>persistence</i>	int	The delivery mode; 2 indicates persistent messages or 1 indicates nonpersistent messages.
<i>priority</i>	int	The message priority (0 through 9, with 9 being the highest priority).
<i>timetolive</i>	long	The amount in milliseconds before the message expires.

### Return Value

None.

### Exceptions

Throws JMSEException if a message is null or if the JMS provider fails to send the message due to an internal error.

---

## **sendBytesTo(payload, destination)**

### Syntax

```
sendBytesTo(payload, destination)
```

### Description

Sends a byte message of the specified byte array value to the specified topic or queue.

### Parameters

Name	Type	Description
<i>payload</i>	<i>byte[]</i>	The byte array value.
<i>destination</i>	java.lang.String	The name of the topic or queue from which this method receives messages.

### Return Value

None.

### Exceptions

Throws JMSEException if a message is null or if the JMS provider fails to send the message due to an internal error.

---

## **sendBytesTo(*payload, destination, persistence, priority, timetolive*)**

### Syntax

```
sendBytesTo(payload, destination, persistence, priority, timetolive)
```

### Description

Sends a persistent or nonpersistent byte message of the specified byte array value, message priority, and time to live to the specified topic or queue.

### Parameters

Name	Type	Description
<i>payload</i>	<i>byte[]</i>	The byte array value.
<i>persistence</i>	int	The delivery mode; 2 indicates persistent messages or 1 indicates nonpersistent messages.
<i>priority</i>	int	The message priority (0 through 9, with 9 being the highest priority).
<i>timetolive</i>	long	The amount in milliseconds before the message expires.

### Return Value

None.

### Exceptions

Throws JMSEException if a message is null or if the JMS provider fails to send the message due to an internal error.

---

## **sendText(*payload, persistence, priority, timetolive*)**

### Syntax

```
sendText(payload, persistence, priority, timetolive)
```

### Description

Sends a persistent or nonpersistent text message with the text, message priority, and time to live specified.

### Parameters

Name	Type	Description
<i>payload</i>	<i>byte[]</i>	The byte array value.

Name	Type	Description
<i>persistence</i>	int	The delivery mode; 2 indicates persistent messages or 1 indicates nonpersistent messages.
<i>priority</i>	int	The message priority (0 through 9, with 9 being the highest priority).
<i>timetolive</i>	long	The amount in milliseconds before the message expires.

#### Return Value

None.

#### Exceptions

Throws JMSEException if a message is null or if the JMS provider fails to send the message due to an internal error.

### **sendText(*payload*)**

#### Syntax

```
sendText (payload)
```

#### Description

Sends a text message with the specified text.

#### Parameters

Name	Type	Description
<i>payload</i>	java.lang.String	The text in the message.

#### Return Value

None.

#### Exceptions

Throws JMSEException if a message is null or if the JMS provider fails to send the message due to an internal error.

### **sendTextTo(*payload*, *destination*)**

#### Syntax

```
sendTextTo (payload, destination)
```

#### Description

Sends a text message with the specified text to the specified topic or queue.

### Parameters

Name	Type	Description
<i>payload</i>	java.lang.String	The text in the message.
<i>destination</i>	java.lang.String	The name of the topic or queue from which this method receives messages.

### Return Value

None.

### Exceptions

Throws JMSEException if a message is null or if the JMS provider fails to send the message due to an internal error.

## ***sendTextTo(payload, destination, persistence, priority, timetolive)***

### Syntax

```
sendTextTo(payload, destination, persistence, priority, timetolive)
```

### Description

Sends a persistent or nonpersistent text message with the specified text, message priority, and time to live to the specified topic or queue.

### Parameters

Name	Type	Description
<i>payload</i>	java.lang.String	The text in the message.
<i>destination</i>	java.lang.String	The name of the topic or queue from which this method receives messages.
<i>persistence</i>	int	The delivery mode; 2 indicates persistent messages or 1 indicates nonpersistent messages.
<i>priority</i>	int	The message priority (0 through 9, with 9 being the highest priority).
<i>timetolive</i>	long	The amount in milliseconds before the message expires.

### Return Value

None.

### Exceptions

Throws JMSEException if a message is null or if the JMS provider fails to send the message due to an internal error.

## ***sendTo(message, destination, persistence, priority, timetolive)***

### Syntax

```
sendTo(message, destination, persistence, priority, timetolive)
```

### Description

Sends a persistent or nonpersistent message with a specified message priority and time to live to the specified topic or queue.

### Parameters

Name	Type	Description
<i>message</i>	com.stc.connectors.jms.Message	The message variable type.
<i>destination</i>	java.lang.String	The name of the topic or queue from which this method receives messages.
<i>persistence</i>	int	The delivery mode; 2 indicates persistent messages or 1 indicates nonpersistent messages.
<i>priority</i>	int	The message priority (0 through 9, with 9 being the highest priority).
<i>timetolive</i>	long	The amount in milliseconds before the message expires.

### Return Value

None.

### Exceptions

Throws JMSEException if a message is null or if the JMS provider fails to send the message due to an internal error.

## ***sendTo(message, destination)***

### Syntax

```
sendTo(message, destination)
```

### Description

Sends a message to the specified topic or queue.

### Parameters

Name	Type	Description
<i>msg</i>	com.stc.connectors.jms.Message	The message variable to be sent.
<i>destination</i>	java.lang.String	The name of the topic or queue from which this method receives messages.

### Return Value

None.

### Exceptions

Throws JMSEException if a message is null or if the JMS provider fails to send the message due to an internal error.

For information about using the JMS methods to build Collaboration Definitions, refer to [“Examples of Building JMS Collaboration Definitions” on page 59](#).

## 5.3.2 JMS Methods Available for JMS Messages

Collaboration Definitions that use the JMS OTD as described in [“Adding JMS Web Services to Collaboration Definitions” on page 38](#) have the several JMS methods available. One set of JMS methods is available for JMS OTDs and are described in the section above. The second set of JMS OTDs is available for JMS messages received in the Collaboration—messages of the type `com.stc.connectors.jms.Message`. This section describes these JMS methods.

---

### countMapMessage()

#### Syntax

```
countMapMessage()
```

#### Description

Counts the number of keys in the map.

#### Parameters

None.

#### Return Value

Returns	Type
The number of keys.	int

#### Exceptions

None.

---

### countStreamMessage()

#### Syntax

```
countStreamMessage()
```

#### Description

Count the number of items in the stream message.

**Parameters**

None.

**Return Value**

Returns	Type
The number of items in the stream message.	int

**Exceptions**

None.

## countUserProperty()

**Syntax**

```
countUserProperty()
```

**Description**

Counts the number of user properties.

**Parameters**

None.

**Return Value**

Returns	Type
The number of user properties.	int

**Exceptions**

None.

## retrieveBytesFromMessage(*encoding*)

**Syntax**

```
retrieveBytesFromMessage(encoding)
```

**Description**

Returns the byte array of the message object, if possible, with the specified encoding.

**Parameters**

Name	Type	Description
<i>encoding</i>	java.lang.String	The encoding to use when converting to a byte array.

### Return Value

Returns	Type
The byte array corresponding with the message payload.	byte[]

### Exceptions

Throws JMSEException, UnsupportedEnclodingException.

---

## retrieveBytesFromMessage()

### Syntax

```
retrieveBytesFromMessage()
```

### Description

Returns the byte array of the message object.

### Parameters

None.

### Return Value

Returns	Type
The byte array corresponding with the message payload.	byte[]

### Exceptions

Throws JMSEException.

---

## retrieveMapMessage(*name*)

### Syntax

```
retrieveMapMessage(name)
```

### Description

Retrieve the specified message from the map message list, if the message exists.

### Parameters

Name	Type	Description
<i>name</i>	java.lang.String	The name of the map message.



### Return Value

Returns	Type
The value of the message, or null if the message does not exist.	Object

### Exceptions

None.

---

## retrieveStringFromMessage(*encoding*)

### Syntax

```
retrieveStringFromMessage(encoding)
```

### Description

Returns the string value of the message with the specified encoding.

### Parameters

Name	Type	Description
<i>encoding</i>	java.lang.String	The encoding to use when converting to a string object.

### Return Value

Returns	Type
The string corresponding with the message payload.	String

### Exceptions

Throws JMSEException if it cannot convert message to a String, UnsupportedEncodingException.

---

## retrieveStringFromMessage()

### Syntax

```
retrieveStringFromMessage()
```

### Description

Returns the string value of a message.

### Parameters

None.

### Return Value

Returns	Type
The string corresponding with the message payload.	String

### Exceptions

Throws JMSEException if it cannot convert message to a String

## retrieveUserProperty(name)

### Syntax

```
retrieveUserProperty(string)
```

### Description

Returns the user property with the specified name.

### Parameters

Name	Type	Description
name	java.lang.String	The name of the user property.

### Return Value

Returns	Type
The value of the property, or null if the property does not exist.	String

### Exceptions

None.

## storeMapMessage(name, value)

### Syntax

```
storeMapMessage(name, value)
```

### Description

Writes the name and value of a map message to the map message object.

### Parameters

Name	Type	Description
<i>name</i>	java.lang.String	The name of the map message.
<i>value</i>	Object	The value of the map message.

### Return Value

None.

### Exceptions

None.

---

## storeUserProperty(*name, value*)

### Syntax

```
storeUserProperty(name, value)
```

### Description

Writes the name and value of a user property to the user property object.

### Parameters

Name	Type	Description
<i>name</i>	java.lang.String	The name of the user property.
<i>value</i>	java.lang.String	The value of the user property.

### Return Value

None.

### Exceptions

None.

The section below describes examples of how to build Collaboration Definitions with the JMS methods.

---

## 5.4 Examples of Building JMS Collaboration Definitions

This section provides examples of building Collaboration Definitions with the JMS methods. To be able to use JMS methods in a Collaboration Definition, you must have created the definition with the JMS OTD as described in [Adding JMS Web Services to Collaboration Definitions](#) on page 38. For an overview of all JMS methods provided with the JMS OTD, refer to [Using JMS Methods in Collaboration Definitions](#) on page 41.

This section provides the following examples of using JMS methods:

- [Sending Text Messages with the sendTo Method](#) on page 60
- [Receiving and Sending Messages](#) on page 65

For any methods that include a variable (such as `com.stc.connectors.jms.Message`), refer [Defining the Message To Be Sent](#) on page 65 for a description of defining variables.

### 5.4.1 Sending Text Messages with the *sendTo* Method

This section describes how to create the logic for a Collaboration to send text messages. To do this, you use the *sendTo* method in the Java Collaboration Rules Editor.

To be able to use this JMS method, the Collaboration Definition must have been created with the JMS OTD as described in [“Adding JMS Web Services to Collaboration Definitions” on page 38](#). Although you are sending message to a different destination than the one in the Connectivity Map, you must have a destination connected in the Connectivity Map. The *sendTo* method than overrides that message destination.

The process of creating a Collaboration that sends text messages consists of the following steps:

- 1 Populating the text message (see [“Populating Text Messages” on page 60](#)).
- 2 Creating the text message (see [“Creating the Text Message To Be Sent” on page 61](#)).
- 3 Sending the message to a specified destination (see [“Sending the Message to a Specified Destination \(sendTo\)” on page 63](#)).

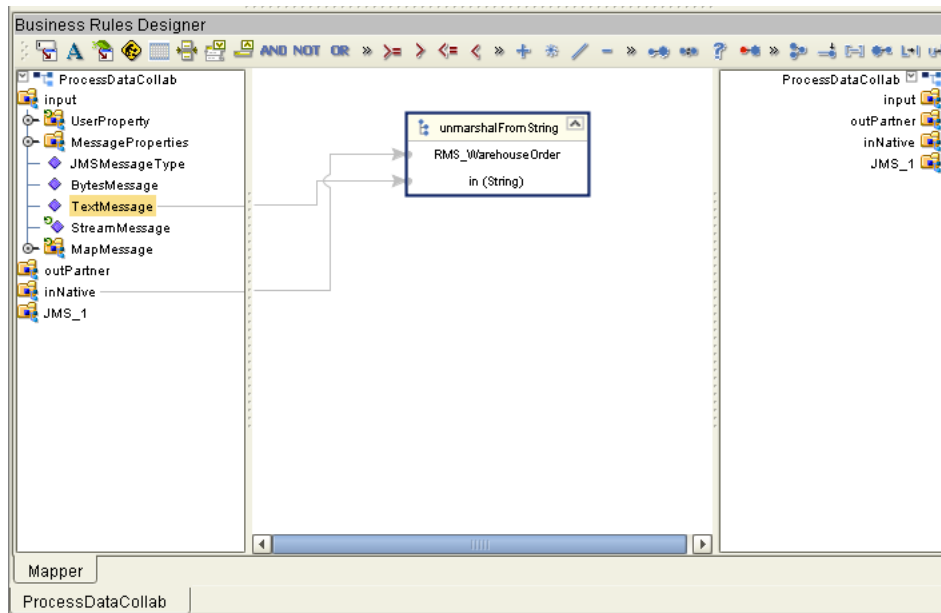
### Populating Text Messages

The procedure below describes how to unmarshal the text message and create the mappings for the input and output fields.

To populate text messages

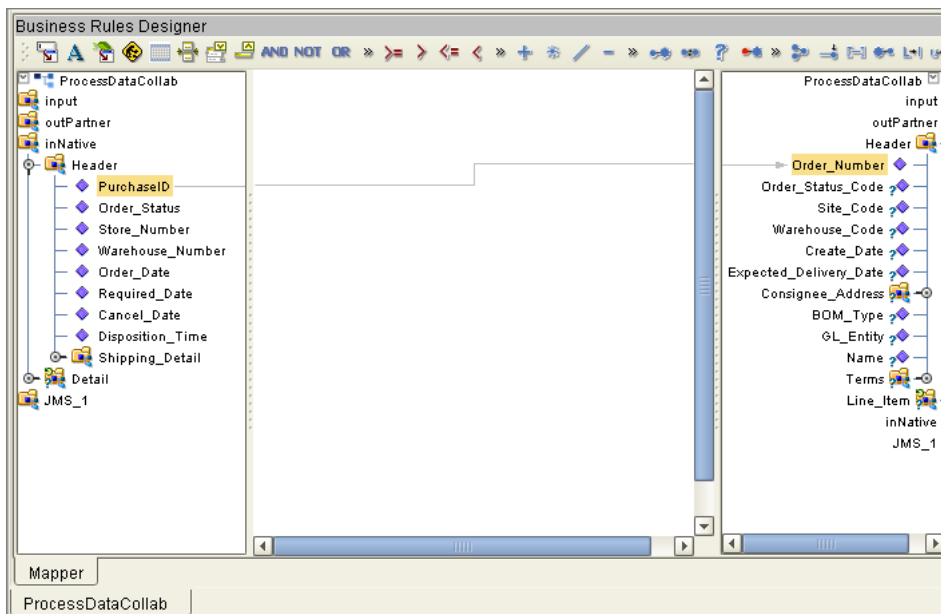
- 1 In the Business Rules toolbar of the Java Collaboration Rules Editor, right-click the input OTD and click **Select a method to call**. A list of methods appears.
- 2 Click **unmarshalFromString()**. The **unmarshalFromString** box appears.
- 3 Expand **MessageProperties** and drag **TextMessage** to **in (String)** in the **unmarshalFromString** box as shown below.

Figure 16 Populating Text Messages



- 4 Map the input OTD components to the output components as shown below.

Figure 17 Mapping Input and Output Components



Now that the message is populated, you can create the text message as described in the next section.

## Creating the Text Message To Be Sent

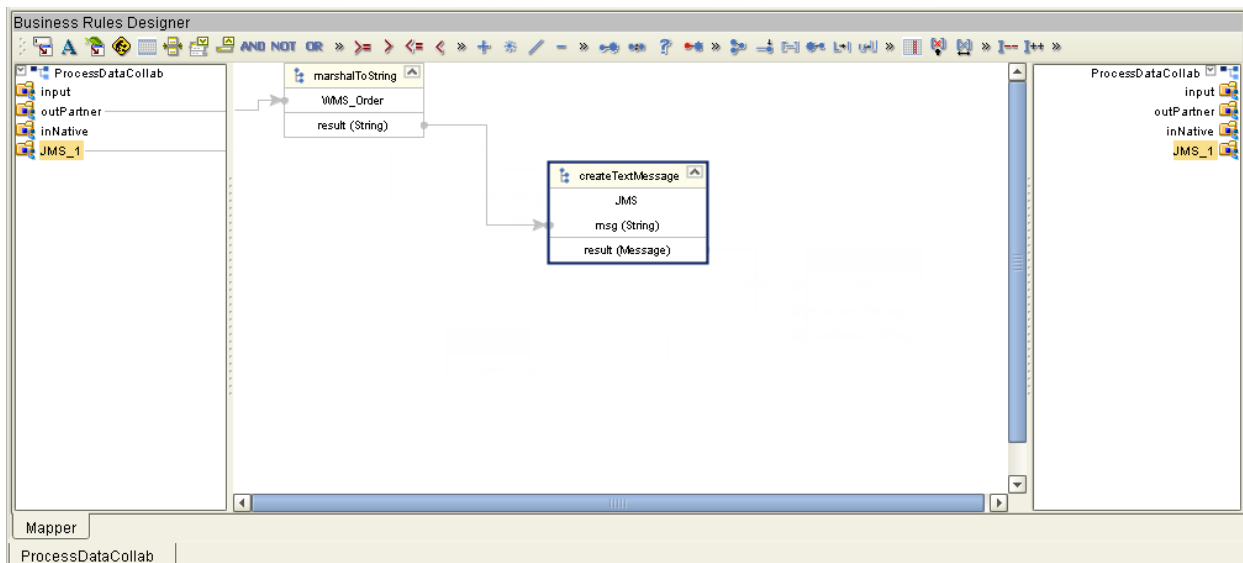
The procedure below describes how you create the text message that you populated in the previous section.

This example uses text as a JMS message object, which allows for advanced JMS OTD functionality such as setting message properties. Using *sendText* and other similar methods allows you to send JMS without the need for specifying the message type.

**To create the text message to be sent**

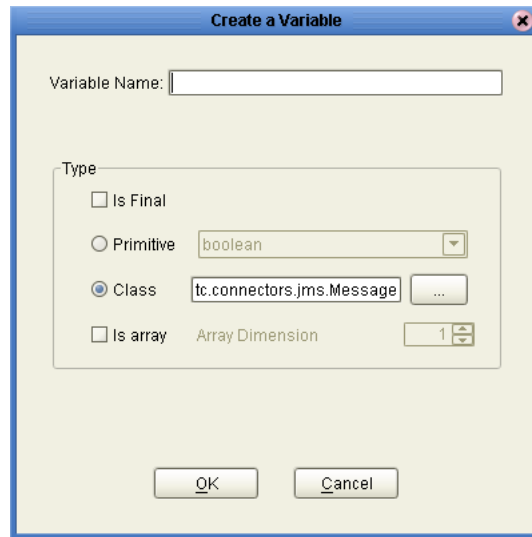
- 1 In the Java Collaboration Rules Editor, right-click the output OTD and click **Select a method to call**. A list of JMS methods appears.
- 2 Click **marshalToString()**. The **marshalToString** box appears.
- 3 Right-click **JMS\_1** and click **Select a method to call**. A list of JMS methods appears.
- 4 Click **createTextMessage()**. The **createTextMessage** box appears.
- 5 Drag **result (String)** from the **marshalToString** box to **Msg (String)** in the **createTextMessage** box as shown below.

**Figure 18** Assigning the Marshal Result to Destination



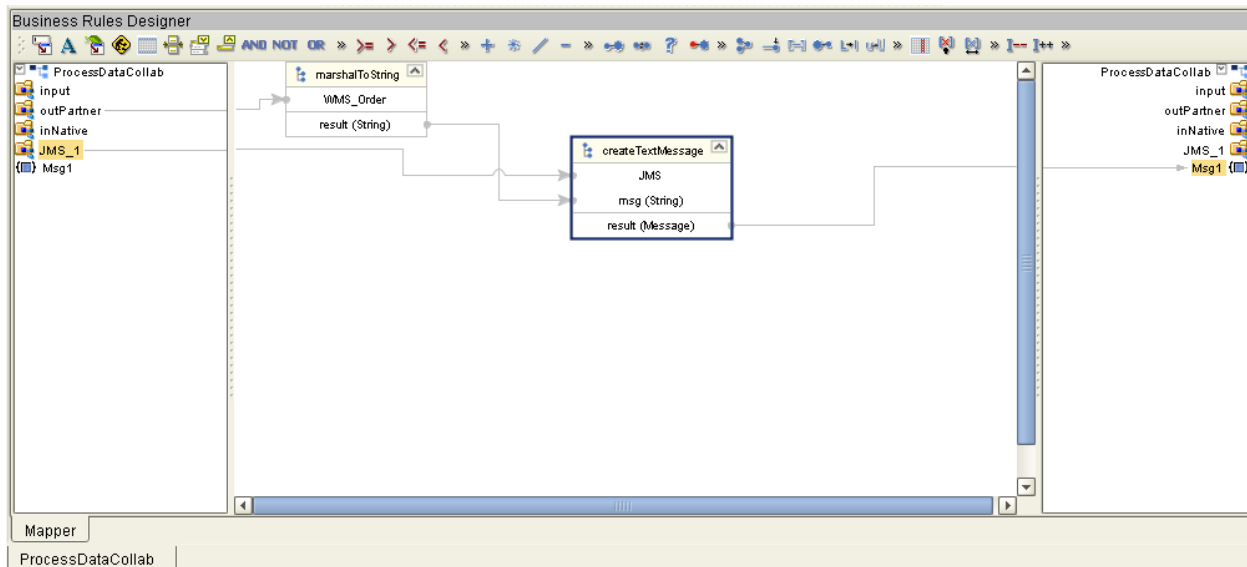
- 6 Right-click the **result (Message)** in the **createTextMessage** box and select **Create New Variable of this type**. The **Create a Variable** dialog box appears.

**Figure 19** Creating a Variable



- 7 Enter the variable name.
- 8 Click **OK**. This adds the message definition to the input and output columns.
- 9 Drag **result (Message)** to the newly defined message in the right-hand output column as shown below.

**Figure 20** Creating a Message in a Collaboration Definition




You have now created the text message. Continue with the next section to send the message.

### **Sending the Message to a Specified Destination (*sendTo*)**

The procedure below describes how you send the message that was marshaled to text in the previous section to a specified destination. With the regular *send* method,

messages are sent to the message destination that connects to the Collaboration in the Connectivity Map. With the *sendTo* method, you can specify a different destination. The sample below uses the complex *sendTo* method to show how you can set message priority and time to live.

#### To send the message to a specified destination

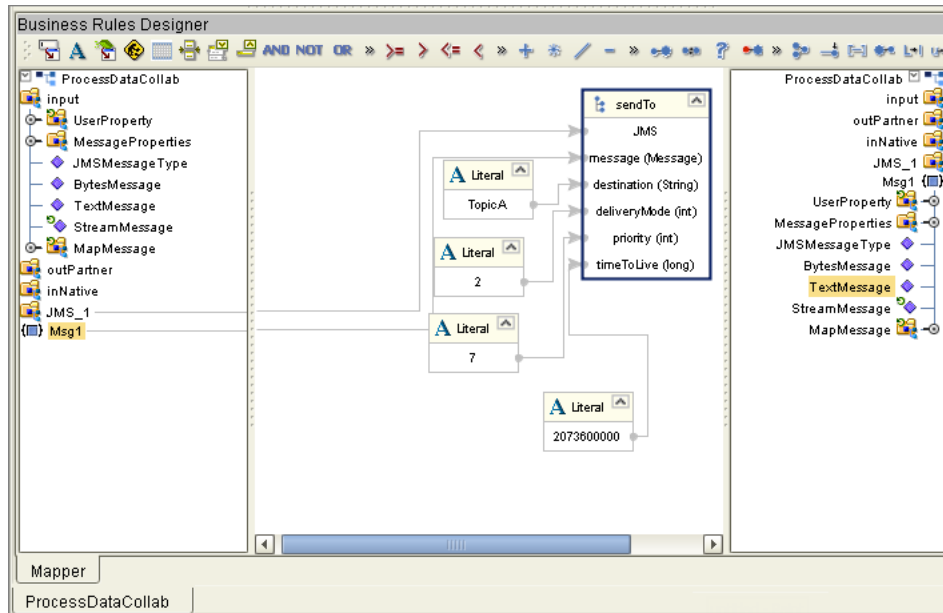
- 1 In the Java Collaboration Rules Editor, right-click **JMS\_1** and click **Select a method to call**. A list of JMS methods appears.
- 2 Click **sendTo(com.stc.connectors.jms.Message message, java.lang.String destination, int deliveryMode, int priority, long timeToLive)**. The **sendTo** box appears.
- 3 Drag the newly created input message into **message (Message)** in the **sendTo** box.
- 4 To specify the topic or queue where this message is sent, click **Create Literal** . The **Create Literal** dialog box appears.
- 5 In the **Type** box, click **string**.
- 6 In the **Value** box, enter the name of the topic or queue where you want the message to be sent, and click **OK**. This adds the literal to the Business Rules Designer.
- 7 Drag the literal to **destination (String)** in the **sendTo** box.
- 8 To specify the delivery mode, do the following:
  - ♦ Click **Create Literal**.
  - ♦ Click **int** in the **Type** box.
  - ♦ In the **Value** box, click 1 for persistent delivery mode, and 2 for nonpersistent mode.
  - ♦ Click **OK**.
- 9 Drag the literal to **deliveryMode (int)** in the **sendTo** box as shown below.
- 10 To specify the message priority for this message, do the following:
  - ♦ Click **Create Literal**.
  - ♦ Click **int** in the **Type** box.
  - ♦ In the **Value** box, click a number between 0 and 9 to indicate message priority, where 0 is the lowest priority.
  - ♦ Click **OK**.
- 11 Drag the literal to **priority (int)** in the **sendTo** box.
- 12 To specify the amount of time before this message expires, do the following:
  - ♦ Click **Create Literal**.
  - ♦ Click **long** in the **Type** box.
  - ♦ In the **Value** box, enter the amount of time in milliseconds for this message to live before it expires.
  - ♦ Click **OK**.



- 13 Drag the literal to **timeToLive (long)** in the **sendTo** box.
- 14 Click **Save**.

This completes the configuration for the *sendTo* method as shown below.

**Figure 21** Sending Messages to a Different Destination



## 5.4.2 Receiving and Sending Messages

This section describes how to create the logic for a Collaboration to receive and send messages. To do this, you use the *receive* and *sendTo* methods in the Java Collaboration Rules Editor.

To be able to use this JMS method, the Collaboration Definition must have been built with the JMS OTD as described in [“Adding JMS Web Services to Collaboration Definitions” on page 38](#).


The process of creating a Collaboration that receives and sends messages consists of the following steps:

- 1 Defining the text message (see [“Defining the Message To Be Sent” on page 65](#)).
- 2 Populating the message to a specified destination (see [“Receiving the Message from a Specified Destination” on page 67](#)).
- 3 Populating the message to a specified destination (see [“Sending Messages to a Different Destination” on page 68](#)).

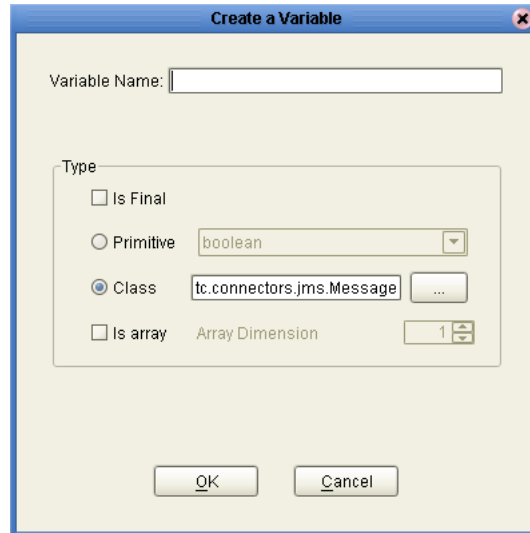
### Defining the Message To Be Sent

The procedure below describes how you define the message to be sent using the *sendTo* method.

### To define the message to be sent

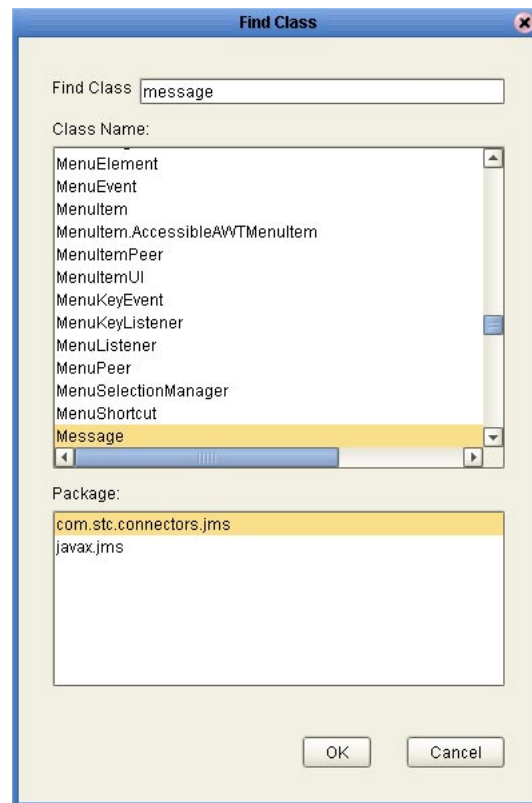
- 1 In the Business Rules toolbar of the Java Collaboration Rules Editor, click **Local Variable** . The **Create a Variable** dialog box appears.

**Figure 22** Creating a Variable



- 2 Enter the variable name.
- 3 Click the ellipsis to specify the class. The **Find Class** dialog box appears.
- 4 In the **Find class** box, type **message** and press **ENTER**.

**Figure 23** Finding Classes for Message Variables



- 5 Click **com.stc.connectors.jms.Message** and click **OK**. This adds the message definition to the input and output columns.

You have now defined the message. Continue with the next section to receive the message that the Collaboration will send.

## Receiving the Message from a Specified Destination


The procedure below describes how to add logic to a Collaboration Definition to receive messages from a specified destination using the *receive (timeout, destination)* method. This destination is one other than the message destination to which the Collaboration connects in the Connectivity Map. The timeout specifies how long the Collaboration waits for a message. If no message arrives, the method returns null.

The receive method uses a nondurable subscriber to perform the JMS receive; a subscriber to the destination must be established before the receive method is called.

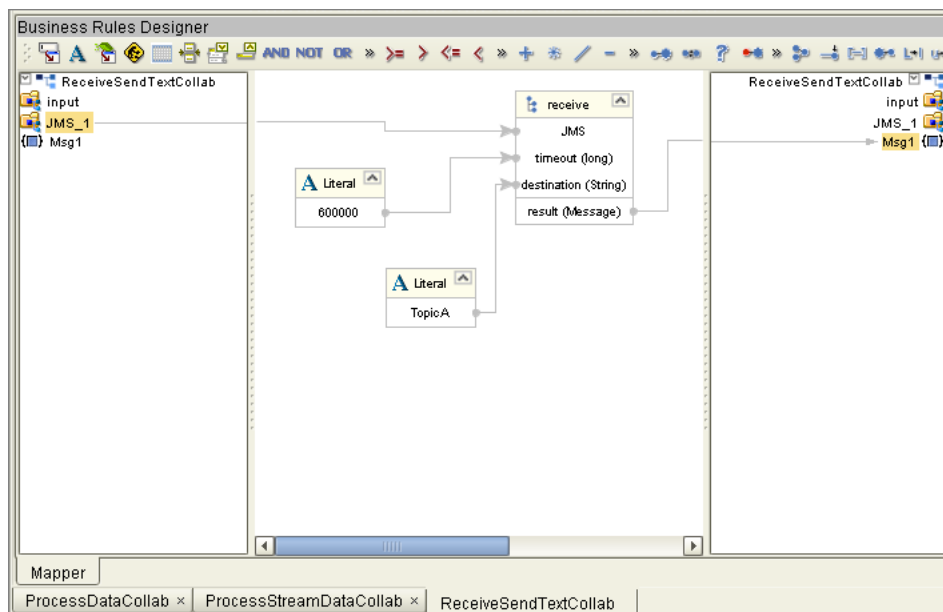
To start this procedure, you must have defined the message as described in the previous section.

### To receive messages with the *receive* method

- 1 In the Java Collaboration Rules Editor, right-click **JMS\_1** and click **Select a method to call**. A list of JMS methods appears.
- 2 Click **Receive (long timeout, java.lang.String destination)**. The receive box appears.

- 3 Click **Create Literal** . The **Create Literal** dialog box appears.
- 4 In the **Type** box, click **long**.
- 5 In the **Value** box, enter amount of time the Collaboration listens for a message to arrive in milliseconds, and click **OK**. This adds the literal to the Business Rules Designer.
- 6 Drag the literal to **timeout (long)** in the **receive** box.
- 7 To specify the destination from which the Collaboration receives messages, do the following:
  - ♦ Click **Create Literal**.
  - ♦ Click **string** in the **Type** box.
  - ♦ In the **Value** box, enter the name of the destination from which the Collaboration receives messages.
  - ♦ Click **OK**.
- 8 Drag the literal to **destination (String)** in the **receive** box.
- 9 Drag **result (String)** from the receive box to the newly created message in the output column as shown below.


**Figure 24** Populating the Message



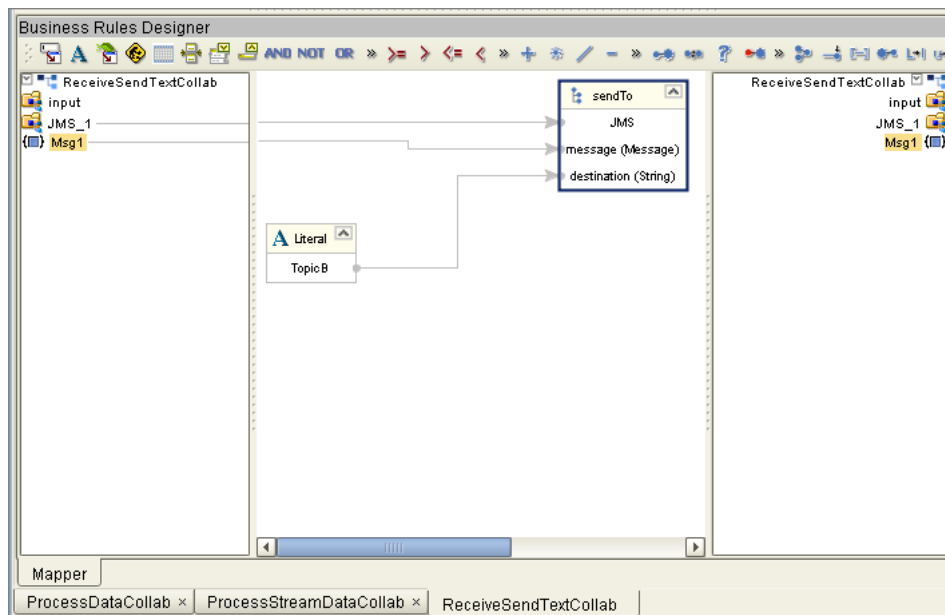
## Sending Messages to a Different Destination

The procedure below describes how to add logic to a Collaboration Definition to send messages to a specified destination using the *sendTo* method. This destination is one other than the message destination to which the Collaboration connects in the Connectivity Map. To start this procedure you must have built the logic to receive messages as described in the previous section.

### To send messages to a different destination

- 1 In the Java Collaboration Rules Editor, right-click **JMS\_1** and click **Select a method to call**. A list of JMS methods appears.
- 2 Click **sendTo(com.stc.connectors.jms.Message message, java.lang.String destination)**. The **sendTo** box appears.
- 3 Drag the newly created input message into **message (Message)** in the **sendTo** box.
- 4 Click **Create Literal** . The **Create Literal** dialog box appears.
- 5 In the **Type** box, click **string**.
- 6 In the **Value** box, enter the name of the topic or queue where you want the message to be sent, and click **OK**. This adds the literal to the Business Rules Designer.
- 7 Drag the literal to **destination (String)** in the **sendTo** box as shown below.

**Figure 25** Sending Messages to a Different Destination



- 8 Click **Save**.

## 5.5 Setting JMS Message Properties in Collaborations

For Collaboration Definitions built with the JMS OTD as described in [“Adding JMS Web Services to Collaboration Definitions” on page 38](#), you can set the message properties listed in the table below:

When you set these priorities in the Collaboration Definition, the properties are only used for the Collaboration that uses the definition and determine how the message is sent. The table below shows the defaults and the required values for the properties. It also shows the JMS method used when you set a message property in a Collaboration.

**Table 9** JMS Message Properties

Property	Values	Default	Equivalent JMS Methods
CorrelationID	correlation ID		getJMSCorrelationID() setJMSCorrelationID(string)
CorrelationIDAsBytes	correlation ID		getCorrelationIDAsBytes setCorrelationIDAsBytes(byte[])
DeliveryMode	persistent, nonpersistent	persistent	getJMSDeliveryMode() setJMSDeliveryMode(int)
Destination	destination	Connectivity Map destination	getJMSDestination() setJMSDestination(destination)
Expiration	number in milliseconds		getJMSExpiration() setJMSExpiration(long)
MessageID	message ID		getJMSMessageID() setJMSMessageID(string)
Priority	0 - 9 where 9 is the highest priority	4	getJMSPriority() setJMSPriority(int)
Redelivered	true, false		getJMSRedelivered() setJMSRedelivered(boolean)
ReplyTo	destination		getJMSReplyTo() setJMSReplyTo(destination)
Timestamp	number in milliseconds		getJMSTimestamp() setJMSTimestamp(long)
Timetolive	number in milliseconds	24 hours	getTimetolive() setTimetolive(long)
Timeout	number in milliseconds	infinite	n/a
Type	message type		getJMSType() setJMSType(string)

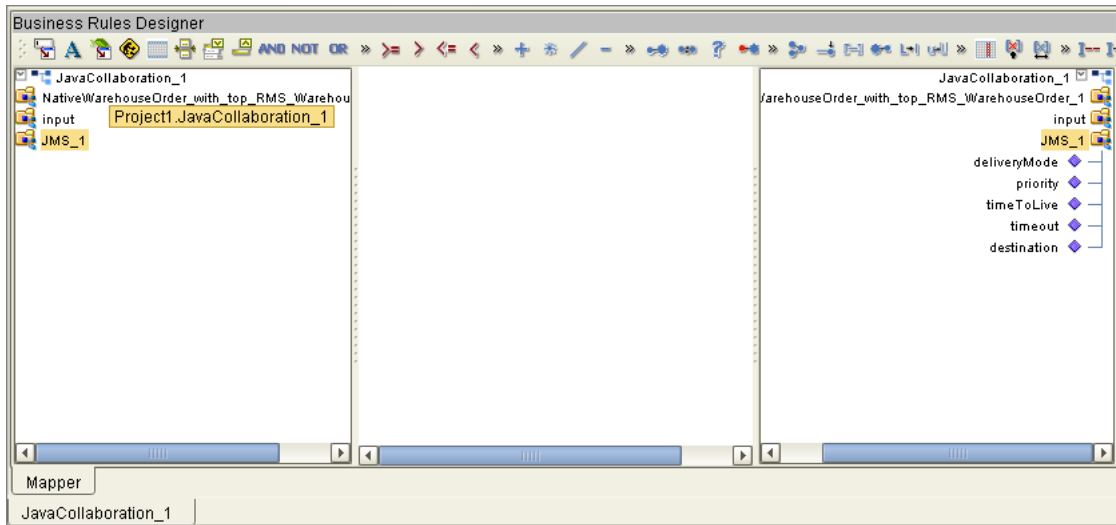
For more information about:


- Delivery mode, see [“Specifying Persistent or Non-Persistent Delivery Mode” on page 79](#).
- Message priority, see [“Specifying Message Priorities” on page 80](#).
- Time to live, see [“Specifying Time To Live for Live Messages” on page 108](#).

To set JMS message properties in Collaborations

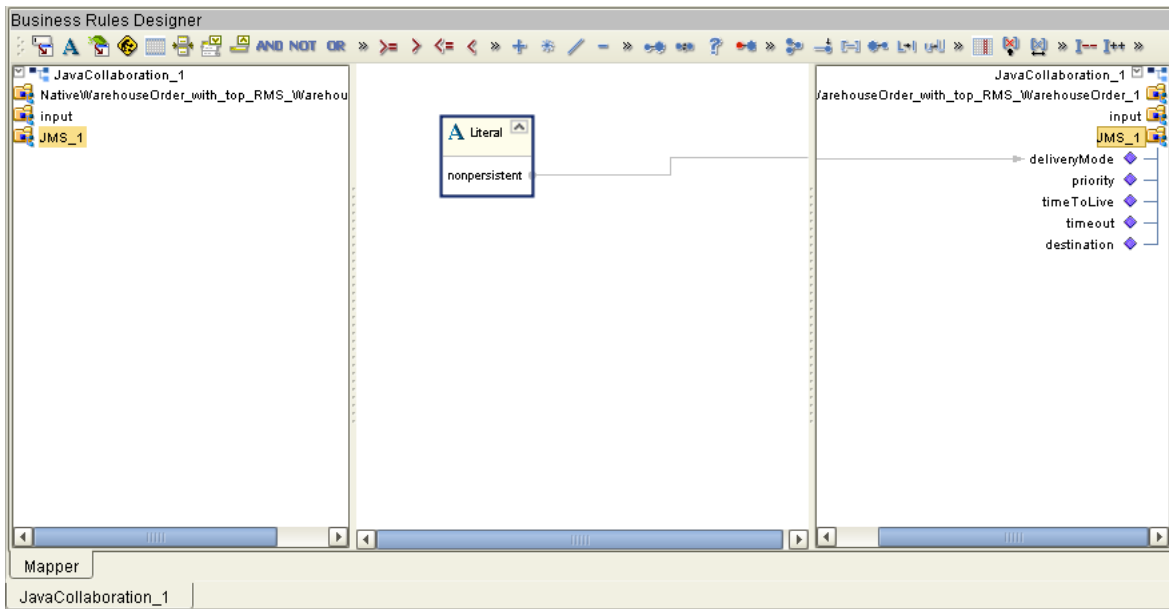
- 1 In the Java Collaboration Rules Editor, expand **JMS\_1**. The message priorities display as follows:

Figure 26 JMS Message Priorities



- 2 Click **Create Literal** . The **Create Literal** dialog box appears.
- 3 In the **Type** box, click the following option depending on the type of message property to be defined:
  - ♦ Delivery mode: string
  - ♦ Message priority: int
  - ♦ Time to live: long
  - ♦ Timeout: long
  - ♦ Destination: string
- 4 In the **Value** box, enter the value for the message property and click **OK**. For example:
  - ♦ Delivery mode: nonpersistent
  - ♦ Message priority: 7
  - ♦ Time to live: 72000000
  - ♦ Timeout: 86400000
  - ♦ Destination: topicB
- 5 Drag the literal to message property. The figure below shows an example of setting the outbound JMS session's delivery mode to nonpersistent.

**Figure 27** Setting JMS Message Properties



6 Click **Save**.

---

## 5.6 Binding Collaboration Definitions to Services

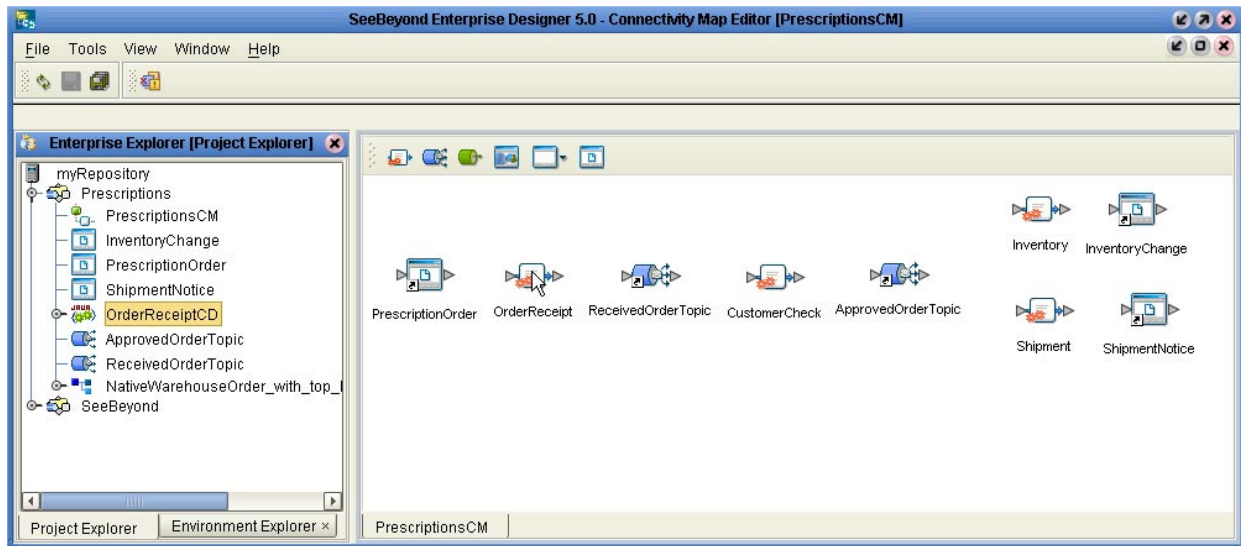
Once you have created a Collaboration Definition for the JMS Collaboration, you bind the definition to the service in the Connectivity Map as described below.

### To bind Collaboration Definitions to services

- 1 In the Project Explorer tab of the Enterprise Designer, expand the Project and open the Connectivity Map for the Project if necessary.
- 2 Click and hold the Collaboration Definition to be bound to a service.
- 3 Drag the definition over the service icon in the Connectivity Map as shown below.

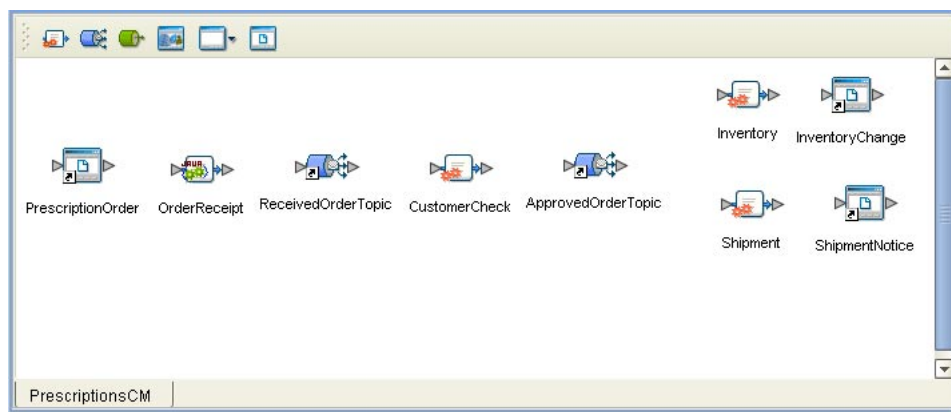


**Figure 28** Binding Collaboration Definitions to Services



- 4 Release the definition onto the service icon. The icon now changes to indicate that the service is now a Collaboration (a service bound to a Collaboration Definition) as shown below.

**Figure 29** Changing a Service to a Collaboration



You have now created a Collaboration that contains the services specified in its Collaboration Definition. To continue with the Project, the next step is to link the Collaboration's services to their in- and outbound destinations as described below.

---

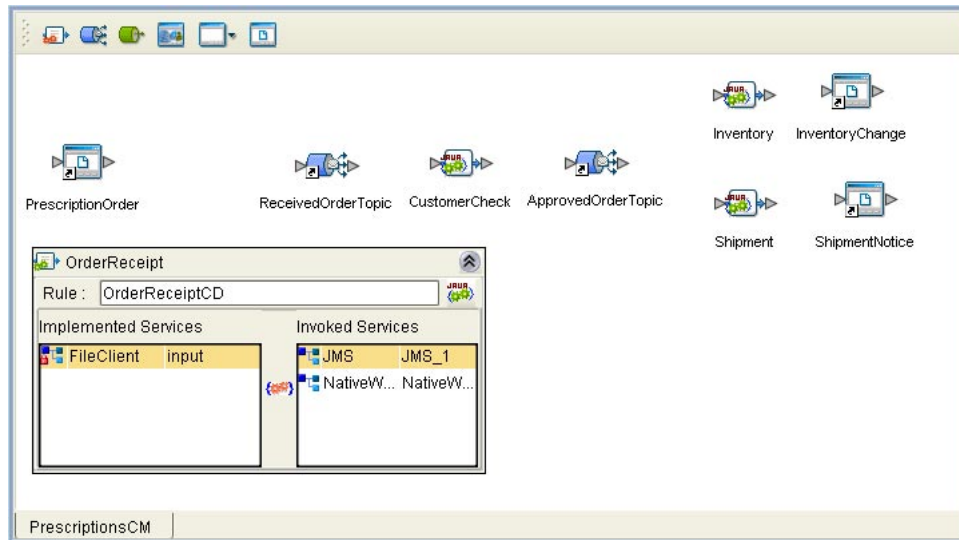
## 5.7 Linking JMS Web Service Collaborations to Destinations

When you have bound the Collaboration Definition to a service to create a Collaboration as described in the previous section, you bind the Collaboration's services to the inbound and outbound destinations as shown below.

### To link JMS Web Service Collaborations to destinations

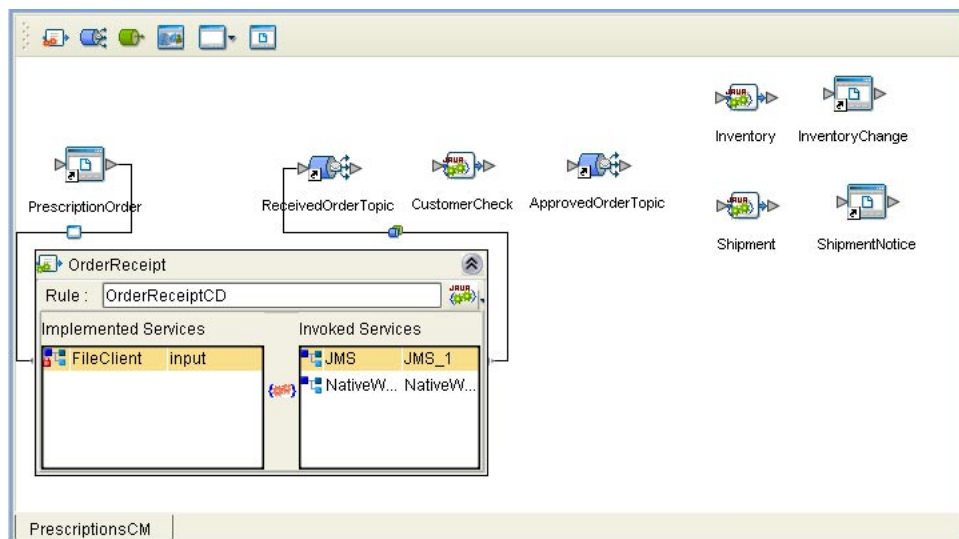
- 1 Double-click the Collaboration. The Collaboration binding windows appears as shown below.

**Figure 30** Collaboration Binding Window



- 2 Drag the services from the **Implemented Services** pane to their inbound destinations.
- 3 Drag the services from the **Invoked Services** pane to the outbound destinations as shown below.

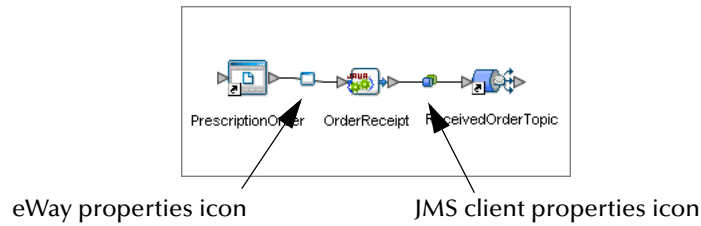
**Figure 31** Binding Services to eWays



- 4 Close the Collaboration binding window. This creates the connections between the Collaborations and its destinations.

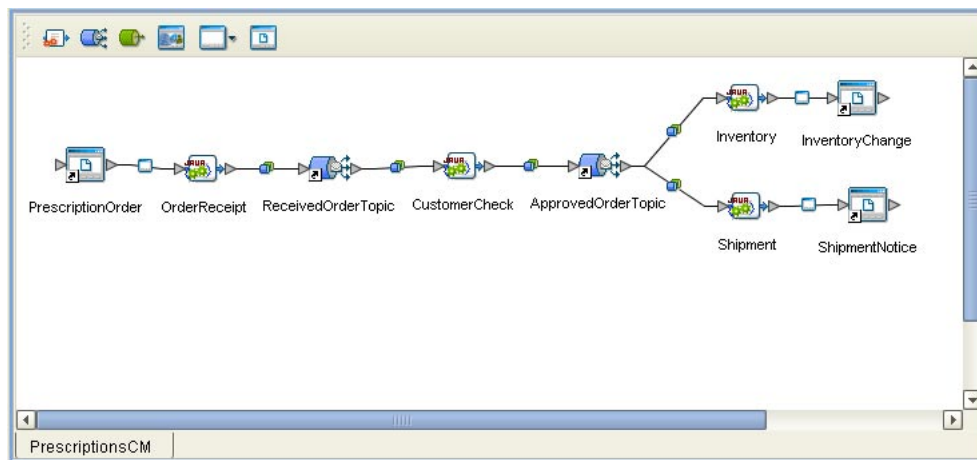
As you can see in the figure below, each connection contains an icon—in this case an eWay properties icon and a JMS client properties icon.

**Figure 32** Collaboration Connections to Destinations



- 5 Continue to bind other services to Collaboration Definitions and link the services to destinations.
- 6 Double-click the properties icons of the eWays and JMS clients to configure each eWay and client. For information about configuring eWays, refer to the eWay documentation. For information about configuring JMS clients, refer to the next chapter.
- 7 Click **Save**. A sample Connectivity Map with all the Collaborations connected to destinations is shown below.

**Figure 33** Finished Connectivity Map



# Configuring JMS Clients

This chapter describes how to configure JMS clients in the Enterprise Designer Connectivity Map. Once you add message destinations and Collaborations or eWays to Projects, and you set up the OTDs and bind them, you configure the JMS clients.

## In This Chapter

- [Configuring JMS Clients](#) on page 76

---

## 6.1 Configuring JMS Clients

After specifying the Project components on the Connectivity Map and defining and binding the Collaborations as described in the previous chapter, you can configure the newly created JMS clients

On the links between message destinations and other components there will be green/blue icons as shown in [Figure 34 on page 77](#). You click these JMS client properties icons to configure JMS clients.

You can specify JMS client properties for JMS IQ Managers. For WebLogic and WebSphere, you configure JMS clients with their management tools.

For topic message destinations, the following JMS client properties are available. Only a subset of these properties is available for queue message destinations.

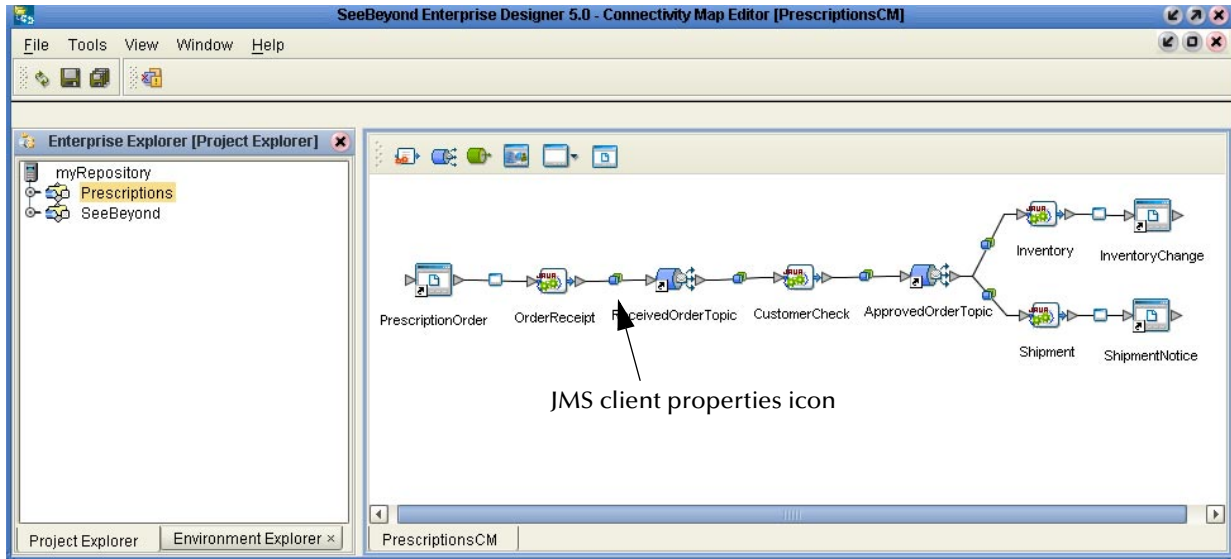
- Transaction mode (transacted or XA)
- Persistent or non-persistent message delivery mode
- Durability
- Concurrency (serial execution or ConnectionConsumer)
- Number of threads per ServerSessionPool
- Message selectors
- Message priority
- Security

### To configure JMS properties

- 1 In the **Project Explorer** tab of the eGate Enterprise Designer, expand the Project which contains the Connectivity Map where you want to configure JMS properties.

- 2 Click the Connectivity Map. The Connectivity Map appears, showing the JMS client properties icon.

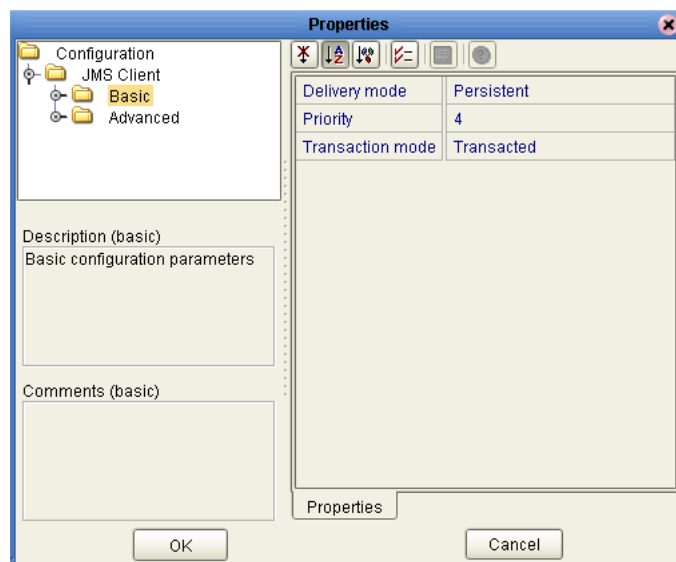
**Figure 34** JMS Client Properties Icon



- 3 Double-click the JMS client properties icon. The **Properties** dialog box appears.
- 4 Expand the **JMS Client** folder and click **Basic**.

The **Properties** dialog box shows the basic JMS client properties as shown below. Which properties appear depends on whether this is a producer, a consumer, a topic, or a queue.

**Figure 35** Basic JMS Client Properties



- 5 Specify the property settings. The list below shows the basic properties and the relevant page for more information:

Property	Available for	Page reference
Concurrency	Topic subscribers Queue receivers	page 79
Delivery mode	Topic publishers Queue senders	page 79
Message selector	Topic subscribers Queue receivers	page 80
Priority	Topic publishers Queue senders	page 80
Transaction mode	Topic subscribers Topic publishers Queue receivers Queue senders	page 81

- 6 To specify basic security settings, expand **Basic** and click **Run as Principal** or **Security** in the left pane and set the security properties. The list below shows the properties and the relevant page for more information:

Property	Available for	Page reference
Connection password and user ID	Topic publishers Queue senders	page 82
Name and password	Topic subscribers Queue receivers	page 82
SecurityRealm	Topic subscribers Queue receivers	page 82
Use for JMS connection	Topic subscribers Topic publishers Queue receivers Queue senders	page 82

- 7 Click **Advanced** in the left pane and specify the advanced property settings. The list below shows the advanced properties and the relevant page for more information:

Property	Available for	Page reference
Durability	Topic publishers	page 84
Server session pool	Topic subscribers Queue receivers	page 84

- 8 To set up advanced security, click **Security** in the left pane, specify the property settings, and click **OK**. For information about security properties, refer to [“Specifying Security Properties” on page 81](#).
- 9 If you reconfigured an existing JMS client, you must reactivate the Deployment Profile for the changes to be implemented.

The sections below describe the JMS client properties in detail.

## 6.1.1 Setting JMS Client Message Processing Order

You can set a consumer's message processing order with the **Concurrency** property. To display JMS client properties, refer to [“Configuring JMS Clients” on page 76](#).

The **Concurrency** property specifies whether the message consumers uses connection consumer or serialized processing. To use concurrent processing for a connection, select the **Connection consumer** setting. To use serial execution, select the **Serial mode** setting.

JMS clients can be configured to use connection consumers to improve message throughput through concurrent processing. Connection consumers consume messages that are specified by a destination and an optional message selector (see [“Specifying Message Selectors” on page 80](#)).

To start processing, a connection consumer gets a server session from its pool and loads the session with a message. Server sessions associate a JMS session with a thread. The server session pool is a set of server sessions provided to a connection consumer to process its messages.

The use of connection consumers increases message processing performance by enabling concurrent processing via multiple threads. You can specify the number of message driven beans (MDBs) or server session pool to assign to a JMS Collaboration to process messages concurrently. When you use connection consumer with fully concurrent or protected concurrent FIFO processing, this setting allows the integration server to assign multiple threads to execute the Collaboration on a particular message destination.

For queues, it is also possible to use connection consumer for concurrent processing on multiple CPUs (and integration servers) on a system. This configuration does have an effect on FIFO processing. For information, refer to [“Message Processing Order” on page 27](#).

You specify the maximum number of threads per server session pool as described in [“Setting the Number of ServerSessionPool Threads” on page 84](#). By default, the maximum number of threads is 5.

The maximum number of messages that a connection consumer can load into a server session at one time is set by the **Server session batch size** property. This property is set at 1 and cannot be changed.

### Default

The default JMS client concurrency mode is **Serial mode**.

## 6.1.2 Specifying Persistent or Non-Persistent Delivery Mode

You can set the delivery mode for topic publishers and queue senders with the **Delivery mode** property. To display JMS client properties, refer to [“Configuring JMS Clients” on page 76](#).

The **Delivery mode** property specifies whether the messages for this JMS connection are persistent or non-persistent.

Non-persistent delivery mode is the most efficient delivery mode; it does not require that messages are saved to permanent storage. Per JMS specification, the message destination delivers non-persistent messages with an at-most-once guarantee (the message is only delivered once, even if it is lost). There is a trade-off between performance and reliability; non-persistence offers better performance, but if a message server fails, non-persistent messages may be lost due to a power outage.

When messages are persistent, the message server places the message in permanent storage to ensure the message is not lost in transit if the message server fails. Persistent messages are delivered once, and only once.

For the JMS IQ Manager, persistent messages are stored in the message server database files. For information about the JMS IQ Manager database and tuning options, refer to [“Setting Performance Options” on page 90](#).

#### Default

The default delivery mode is **Persistent**.

### 6.1.3 Specifying Message Selectors

You can specify message selectors for topic subscribers and queue receivers with the **Message selector** property. To display JMS client properties, refer to [“Configuring JMS Clients” on page 76](#).

To specify a message selector, enter a message selector string as per JMS specification syntax. For example:

```
JMSType = 'car' AND color = 'blue'
```

### 6.1.4 Specifying Message Priorities

You can specify message priorities for topic publishers and queue senders with the **Priority** property. To display JMS client properties, refer to [“Configuring JMS Clients” on page 76](#).

The level of message priority that you set for the JMS client causes all messages produced by this client to have that same priority level. For example, if you set the priority level to 2, all messages sent by that client have message priority level 2.

You can also specify message priorities in Collaborations with the JMS OTD with the *setPriority* method. Collaboration message priorities override JMS client message priorities. For more information, refer to [“Setting JMS Message Properties in Collaborations” on page 69](#).

#### Default

The default delivery mode is 4.

#### Required Values

An integer between 0 and 9, where 0 through 4 is normal priority and 5 through 9 is expedited priority.



## 6.1.5 Specifying the Transaction Mode

You can set the transaction mode for producers and consumers with the **Transaction mode** property. To display JMS client properties, refer to [“Configuring JMS Clients” on page 76](#).

The **Transaction mode** property specifies whether messages for this session use one of the following transaction modes:

### Transacted Mode

When you set the transaction mode to **Transacted**, the JMS session is handled as a transacted session: message acknowledgement is handled automatically by commit, and recovery is handled automatically by rollback. The message destination processes the messages as units consisting of a set of produced and consumed messages. When the transaction commits, the unit of input is acknowledged, and the unit of output is sent. If a transaction is rolled back, the produced messages are deleted and the consumed messages are automatically recovered.

For the JMS IQ Manager, each message destination has its own session. Therefore, in a simple topic A > Collaboration A > topic B Project, there are two sessions and therefore two transactions. This means that duplicate messages and loss of messages may occur unless you use XA mode.

### XA Mode

When the transaction mode is set to **XA**, the JMS session uses the JTA XAResource API to support multiple resources (distributed transactions). Messages in XA-enabled sessions are processed per the two-phase commit protocol. In the first phase, the resource manager sends a query to commit to the receivers and waits for the receivers to respond with a confirmation. In the second phase, the resource manager receives confirmation from all receivers, and commits the message to all receivers.

### Default

The default transaction mode is **Transacted**.

### Required Values

**Transacted** or **XA**.

## 6.1.6 Specifying Security Properties

JMS client security properties must be specified if security is enabled for the message server. To enable message server security for the JMS IQ Manager, refer to [“Enabling Security” on page 89](#).

You can specify the following security settings for JMS clients:

- the user name
- password
- security realm
- authentication
- auditing

- authorization

## Specifying Security Name and Password

You can specify a user name and password for topic subscribers and queue receivers with the **Name** and **password** properties; for the topic publishers and queue senders, use the **Connection password** and **Connection user ID** properties. To display JMS client properties, refer to [“Configuring JMS Clients” on page 76](#).

The user name and password is used for the JMS client to connect to message servers. JMS clients send security information only if the **Use for this connection** property is enabled as described in [“Use for JMS Connection” on page 82](#).

User names and passwords are defined in the Environment in the Enterprise Designer; right-click the Environment in the Enterprise Designer and click **User Management**.

JMS client security properties must be specified if security is enabled for the message server. To enable message server security for the JMS IQ Manager, refer to [“Enabling Security” on page 89](#).

## Specifying the Security Realm

You can specify the security realm for topic subscribers and queue receivers with the **SecurityRealm** property. To display JMS client properties, refer to [“Configuring JMS Clients” on page 76](#).

The security realm is used for the JMS client to connect to message servers. Currently only file realms are supported. The security realm defines users. To set up users, right-click the Environment in the Enterprise Designer and click **User Management**.

JMS clients send the security information only if the **Use for this connection** property is enabled as described in [“Use for JMS Connection” on page 82](#).

JMS client security properties must be specified if security is enabled for the message server. To enable message server security for the JMS IQ Manager, refer to [“Enabling Security” on page 89](#).

### Default

The default is **default**. Currently, adding security realms is unavailable for the JMS IQ Manager.

## Use for JMS Connection

You can send security information for this producer or consumer with the **Use for JMS connection** property. To display JMS client properties, refer to [“Configuring JMS Clients” on page 76](#).

When you set this property to **True**, the security information (user name, password, security realm) is sent when the JMS client connects to a message server.

This property must be enabled when message server security is enabled for JMS clients to be able to connect to the message server. To enable message server security for the JMS IQ Manager, refer to [“Enabling Security” on page 89](#).

### Default

**False**; by default, JMS clients do not send security information to message servers.

### Required Values

**True or False.**

## Enabling Auditing

You can enable user activity logging for this producer or consumer with the **Audit** property. To display JMS client properties, refer to [“Configuring JMS Clients” on page 76](#).

When the **Audit** property is enabled, the integration server logs by which users have accessed the message server. This information is captured in the *integrationservername.log*, where *integrationservername* is the name of the integration server. This file resides in the **logicalhost\logs** directory.

### Default

**True**; user activity logging is enabled.

### Required Values

**True or False.**

## Enabling Authentication

You can enable user authentication for this producer or consumer with the **Authentication** property. To display JMS client properties, refer to [“Configuring JMS Clients” on page 76](#).

When the **Authentication** property is set to **True**, the integration server verifies the user name and password.

### Default

**False**; by default, the integration server does not authenticate user name and password information.

### Required Values

**True or False.**

## Enabling Authorization

You can enable user authorization for this producer or consumer with the **Audit** property. To display JMS client properties, refer to [“Configuring JMS Clients” on page 76](#).

When the **Authorization** property is enabled, the integration server uses its role definitions to verify the user’s permissions.

### Default

**False**; permission authorization is disabled.

## Required Values

**True or False.**

### 6.1.7 Setting Durability

You can enable or disable concurrent processing for topic subscribers with the **Durability** property. To display JMS client properties, refer to [“Configuring JMS Clients” on page 76](#).

The **Durability** property specifies whether the subscriber to this JMS connection is durable. When a subscriber is nondurable, the client only sees messages on a topic when the subscriber is active. If the subscriber is inactive, and messages are published, the messages are lost.

When a subscriber is durable, messages are not lost even when the subscriber is inactive because the message server retains the messages until they are retrieved by the subscriber or until the messages expire. A durable subscriber registers with the message server as a durable subscriber with the name *source\_destination*, for example, *topicA\_CollaborationA*. When a subscriber becomes inactive, the message server retains the unexpired messages for a subsequent subscriber object with the same identity to resume the subscription.

To ensure messages are not lost due to subscriber inactivity, set the **Durability** property to **Durable**. There is a trade-off in performance.

#### Default

By default, JMS clients are **Durable**.

#### Required Values

**Durable or Nondurable.**

### 6.1.8 Setting the Number of ServerSessionPool Threads

You can specify the maximum number of threads (1 - 100) per ServerSessionPool to be used for concurrent processing with the **Server session pool size** property. This property is available for topic subscribers and queue receivers. To display JMS client properties, refer to [“Configuring JMS Clients” on page 76](#).

This property is used on conjunction with the connection consumer setting of the **Concurrency** property ([“Setting JMS Client Message Processing Order” on page 79](#)). You can specify the number of message driven beans (MDBs) or server session pool to assign to a JMS Collaboration to process messages concurrently. When you use connection consumer with fully concurrent or protected concurrent FIFO processing, this connection consumer configuration allows the integration server to assign multiple threads to execute the Collaboration on a particular message destination.

The maximum number of messages that a connection consumer can load into a server session at one time is set by the **Server session batch size** property. This property is set at 1 and cannot be changed.

For an overview about message processing, refer to [“Message Processing Order” on page 27](#).

### **Default**

By default, the maximum number of threads per server session pool is 5.

### **Required Values**

An integer of 1 or larger, depending on the capability of the system, indicating the number of threads.

# Adding and Configuring Message Servers

This chapter describes how to add message servers to Logical Hosts in the Enterprise Designer. Once you have added message servers, you can specify configuration properties for JMS IQ Managers.

This chapter describes configurations for JMS IQ Manager and the 4.5.x JMS IQ Manager. This chapter does not include configurations for the WebLogic and WebSphere JMS servers; these must be configured in the WebLogic and WebSphere application server. For information, refer to the relevant WebLogic and WebSphere documentation.

## In This Chapter:

- [Adding Message Servers to Logical Hosts](#) on page 86
- [Configuring Message Servers](#) on page 88
- [4.5.x SeeBeyond Message Server Configurations](#) on page 109
- [Deploying Topics and Queues to Message Servers](#) on page 110
- [Reconfiguring Message Servers](#) on page 112

---

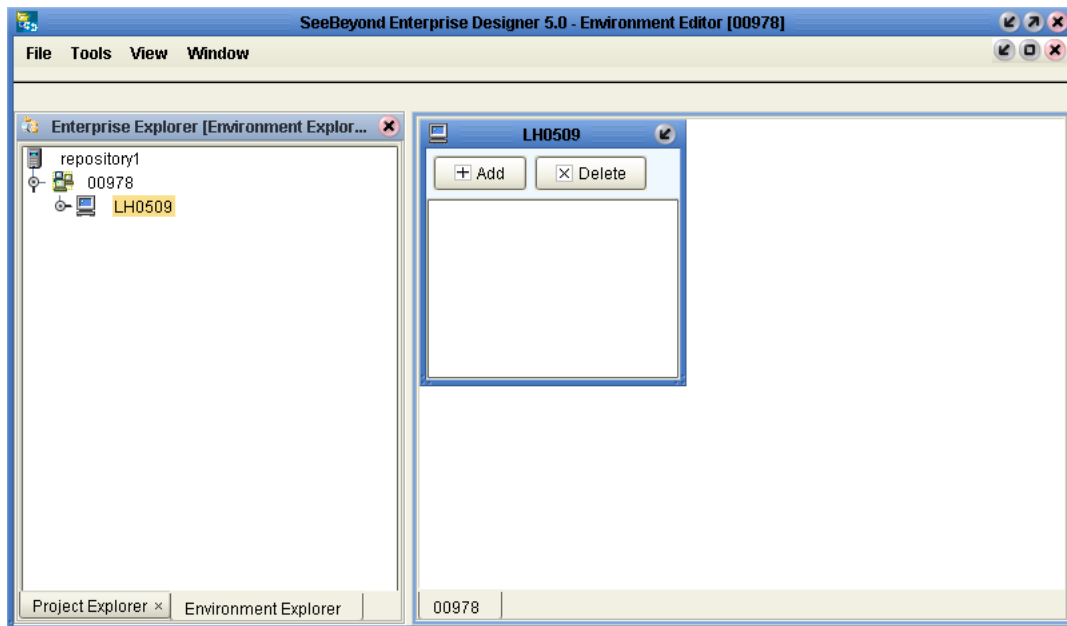
## 7.1 Adding Message Servers to Logical Hosts

Before you can configure a message server you must add it to its Logical Host. The procedure below describes how you add message servers to Logical Hosts in eGate Environments.

### To add message servers to Logical Hosts

- 1 In Enterprise Designer, click the **Environment Explorer** tab.
- 2 Click the Repository and expand or create the Environment that contains the Logical Host to which you want to add a message server.

**Figure 36** Expanding the Environments Folder



- 3 If necessary, create a Logical Host by right-clicking the Environment and selecting **New Logical Host**.
- 4 Right-click the Logical Host to which you want to add a message server and click one of the following options:
  - ◆ **New SeeBeyond Message Server 4.5.x** to create a JMS IQ Manager to connect to a 4.5.x SeeBeyond Message Server.
  - ◆ **New JMS IQ Manager** to create a JMS IQ Manager.
  - ◆ **New WebLogic JMS Server** to create a WebLogic JMS server.
  - ◆ **New WebSphere JMS Server** to create a WebSphere JMS server.

*Note:* The options for the WebLogic and WebSphere JMS servers and the 4.5.x message server are available only if you uploaded the *.sar* for these message servers. For information about installing *.sar* files, refer to **“Installing Message Server .sar Files” on page 35**.

This adds a message server to the Logical Host. To configure JMS IQ Managers, refer to the following sections:

- For JMS IQ Managers, **“Configuring Message Servers” on page 88**.
- For 4.5.x SeeBeyond Message Servers, **“4.5.x SeeBeyond Message Server Configurations” on page 109**

---

## 7.2 Configuring Message Servers

Once you have added a message server to its Logical Host, you can configure the message server as described below. For information about adding message servers to Logical Hosts, refer to the section above.

### To configure message servers

- 1 In Enterprise Designer, click the **Environment Explorer** tab.
- 2 Click the Repository and expand the Environment that contains the Logical Host for which you want to configure a message server.
- 3 Expand the Logical Host, right-click the message server, and click **Properties**. The **Properties** dialog box appears.
- 4 Expand the folders as necessary and specify the properties. The table below shows where to find detailed configuration information for each message server type.

Message Server Type	Configuration
JMS IQ Manager	page 88
4.5.x SeeBeyond Message Server	page 109

- 5 Click **OK**.

---

## 7.3 JMS IQ Manager Configurations

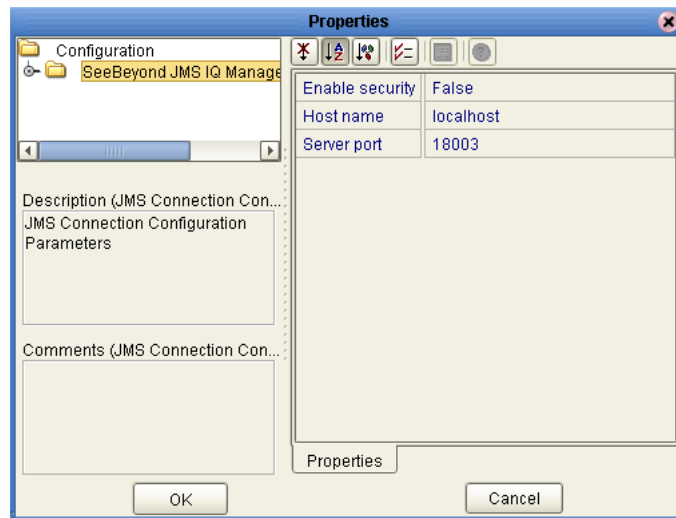
This section describes the JMS IQ Manager configuration options. For procedural information regarding configuring JMS IQ Managers, refer to [“Configuring Message Servers” on page 88](#). For information about the JMS IQ Manager, refer to [“About the JMS IQ Manager” on page 14](#).

### 7.3.1 Setting General Options

You set general configurations for the JMS IQ Manager in the **JMS IQ Manager** tab of the **JMS IQ Manager Properties** dialog box. The figure below shows the **JMS IQ Manager** tab.



**Figure 37** JMS IQ Manager—General Configurations



## Enabling Security

You can enable security in the properties for JMS IQ Managers with the **Enable security** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

To enable security, set this property to **True**. To disable security, set this property to **False**.

When you enable security, you must enter a username and password for each JMS client that subscribes or publishes to the JMS IQ Manager as described in [“Specifying Security Name and Password” on page 82](#). You must also set the **Use for connection** property for those JMS clients as described in [“Use for JMS Connection” on page 82](#).

To manage a Project at runtime that has security enabled, you must provide the **-username** and **-userpassword** flag for the STC MS Control Utility as described in [“Flags and Arguments” on page 124](#).

### Default

The default setting for the **Enable security** property is **False**; JMS IQ Manager security is disabled by default.

## Specifying the Host Name

You can specify the name of the system the JMS IQ Manager runs on in the properties for JMS IQ Managers with the **Host name** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

You can use this property to indicate to an external JNDI server where the system can look up the ConnectionFactory information for the SeeBeyond JMS IQ Manager.

### Default

The default host name is **localhost**.

## Specifying the Server Port Number

You can set the JMS IQ Manager port number in the properties for JMS IQ Managers with the **Server port** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Server port** property specifies the TCP/IP port number that the JMS IQ Manager listens on. Each JMS IQ Manager must have a unique port number per system. If you add JMS IQ Managers, change this setting to an available port.

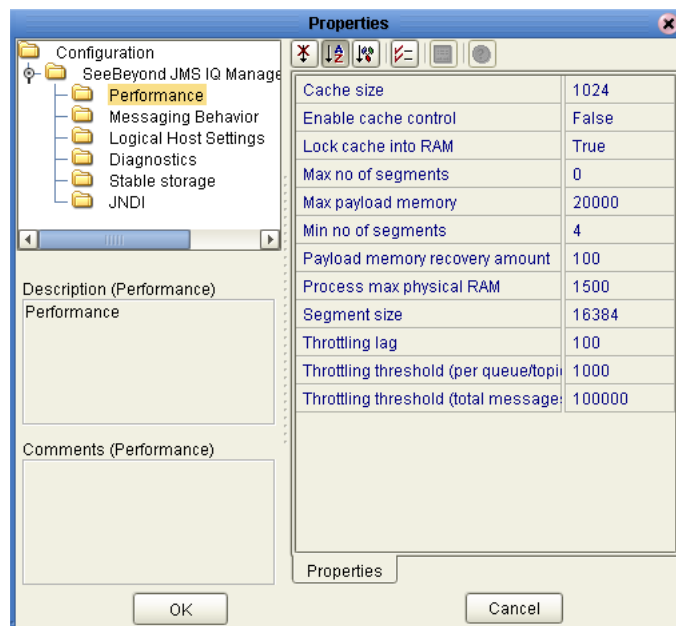
### Default

The default port number depends on at what point you add the JMS IQ Manager to the Logical Host. The logical host's management agent uses the first three ports from the base port, and the integration server uses four ports. So if your base port is 18000, and you first specify the integration server, the JMS IQ Manager port is 18007.

## 7.3.2 Setting Performance Options

You set performance configurations for the JMS IQ Manager in the **Performance** tab of the **JMS IQ Manager Properties** dialog box. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#). The figure below shows the **Performance** tab.

**Figure 38** JMS IQ Manager—Performance Configurations



The JMS IQ Manager enables you to set options specifically to tune the server for optimal performance tailored to your business environment. This section describes each option in detail. Many of these options are related. For detailed information about how these options affect the performance of the JMS IQ Manager, refer to [“About Performance” on page 31](#).

## Specifying Cache Size

You can specify JMS IQ Manager cache size in the properties for JMS IQ Managers with the **Cache size** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Cache size** property specifies the total number of pages in the read cache. This is the amount of memory read from the disk each time the JMS IQ Manager refreshes its message memory.

A page is 512 bytes on Windows, 1024 bytes on UNIX. If you set this property to 2048 pages (1 MB in Windows), to write a 10 MB message to disk, the server performs 10 writes.

You can use this property to optimize performance. If the server processes fixed message sizes, you can set this value to accommodate that size. If you have messages smaller than 1 MB you can set this value to the maximum size of the message plus overhead of two to three pages. For example, for 0.5 MB messages on Windows you would set this value to 5120:  $1024 + (2 \times 2048)$ .

It is not recommended to set this value to the size of the largest message because too much cache memory would be wasted.

### Cache Size and Segment Size

Set the cache size to a number of pages less than the amount of pages specified for segment size. For more information about the **Segment size** property, refer to [“Specifying Segment Size” on page 95](#).

### Default

The default cache size is 1024 pages, which is 0.5 MB for Windows (1024 pages x 512 bytes) and 1 MB for UNIX (1024 pages x 1024 bytes).

### Required Values

An integer from 1 through 999,999,999 indicating the number of pages.

## Enabling Cache Control

You can enable cache control in the properties for JMS IQ Managers with the **Enable cache control** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Enable cache control** property specifies whether the JMS IQ Manager controls cache synchronization to disk. When you disable cache control, the operating system controls the synchronization schedule. Disabling cache control increases performance, but also increases risk of message loss in the event of system failure.

### Default

The default setting for the **Enable cache control** property is **True**; the JMS IQ Manager controls cache synchronization to disk.

## Required Values

**True or False.**

## Locking Cache Into RAM

You can enable cache locking in the properties for JMS IQ Managers with the **Lock cache into RAM** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Lock cache into RAM** property enables or disables whether cache is locked into RAM. This property works for JMS IQ Managers running on Windows platforms only.

The Windows VirtualLock API function locks the server disk cache into physical memory, ensuring that subsequent access to the region does not incur a page fault (a swap-out to disk). You can use this property with the **Cache size** property to improve performance. For more information, refer to [“Specifying Cache Size” on page 91](#).

## Default

By default, this property is set to **True** and cache is locked into RAM.

## Required Values

**True or False.**

## Specifying the Maximum Number of Segments

You can specify the maximum segment number in the properties for JMS IQ Managers with the **Max no of segments** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Max no of segments** property specifies the upper limit for the number of database files (segments) that the JMS IQ Manager creates for its stable message storage. You use this property to limit the amount of disk space that the JMS IQ Manager uses. If the JMS IQ Manager attempts to write data that exceeds this limit, it exits gracefully and logs an error message in the JMS IQ Manager log.

The JMS IQ Manager uses the JMS IQ Manager database to store live persistent messages as well as journaled messages. For more information, refer to [“Setting Stable Storage and Journaling Options” on page 104](#) and [“Specifying Persistent or Non-Persistent Delivery Mode” on page 79](#).

In addition to limiting the maximum number of segments, you can also specify the size limit for segments. For more information about the **Segment size** property, refer to [“Specifying Segment Size” on page 95](#).

You can also specify the minimum number of segments created initially for stable message storage with the **Min no of segments** property as described in [“Specifying the Minimum Number of Segments” on page 94](#).

It is not advisable to use the JMS IQ Manager as a semi-permanent storage medium without sufficient memory and disk resources. To manage the memory and disk resources needed by the JMS IQ Manager, use the publisher throttling feature, controlled by the **Throttling threshold (total messages)**, **Throttling threshold (per**

**queue/topic**), and **Throttling lag** properties. For details, refer to [“Managing Resources” on page 32](#).

#### Default

The default is 0. This value causes the JMS IQ Manager to create new files as needed, limited only by available disk space.

#### Required Values

An integer from 0 through 99,999 indicating the number of segments.

### Specifying the Maximum Amount of Payload Memory

You can specify the payload memory amount in the properties for JMS IQ Managers with the **Max payload memory** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Max payload memory** property specifies the upper limit of the amount of cache, in KB, allocated for the server to keep message payloads in cache. This represents the cache of message data that the server uses to fulfill read requests. When payload memory use increases beyond the **Max payload memory** threshold, the JMS IQ Manager begins memory recovery and cleanup.

This property work together with the **Payload Memory Padding** property as described in the section below.

#### Default

The default is 20,000 KB (20 MB).

#### Required Values

An integer from 1 through 999,999,999 indicating KB. The upper limit depends on available memory resources.

### Specifying Padding for the Payload Memory Threshold

You can specify the payload memory recovery amount in the properties for JMS IQ Managers with the **Payload memory recovery amount** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Payload memory recovery amount** property specifies the amount of KB of cache to recover in a memory recovery and cleanup operation.

#### Default

The default is 100 KB.

#### Required Values

An integer from 0 through 999,999 indicating KB. The upper limit must be less than the amount specified for the **Max payload memory** property.

## Specifying the Minimum Number of Segments

You can specify the minimum segment number in the properties for JMS IQ Managers with the **Min no of segments** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Min no of segments** property specifies the minimum number of database files (segments) that the JMS IQ Manager creates initially for stable message storage. When the minimum is exceeded, the server allocates additional segments on an as-needed basis, up to the number of files specified for the **Max no of segments** property as described in [“Specifying the Maximum Number of Segments” on page 92](#).

The JMS IQ Manager uses the JMS IQ Manager database to store live persistent messages. For more information, refer to [“Setting Stable Storage and Journaling Options” on page 104](#) and [“Specifying Persistent or Non-Persistent Delivery Mode” on page 79](#).

In addition to limiting the maximum number of segments, you can also specify the size limit for segments. For more information about the **Segment size** property, refer to [“Specifying Segment Size” on page 95](#).

### Default

The default is 4 segments.

### Required Values

An integer from 1 through 99,999 indicating the number of segments.

## Specifying the Maximum Amount of Physical RAM for Processes

You can specify the amount of physical RAM in the properties for JMS IQ Managers with the **Process max physical RAM** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Process max physical RAM** property specifies the amount of KB indicating the maximum size of the working set of the JMS IQ Manager. This property is only used when the JMS IQ Manager runs on Windows platforms.

The working set of a process is the set of memory pages currently visible to the process in physical RAM memory. These pages are resident and available for an application to use without triggering a page fault. It is not always advisable to increase this setting; the message service will require less page swapping, but other components may require more.

### Default

The default is 1500 KB.

### Required Values

An integer equal to or larger than 0, indicating KB. This number is limited by the physical memory size. If you set this property to 0, the JMS IQ Manager sets this value.

## Specifying Segment Size

You can specify database segment size in the properties for JMS IQ Managers with the **Segment size** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The JMS IQ Manager database consists of database files (segments). By default, these segments are named **stcms\*.dbs** and reside in the message server folder on the Logical Host. The **Segment size** property specifies the total number of pages in a single segment file. A page is 512 bytes on Windows, 1024 bytes on UNIX. The default segment size is 16,384 pages, which is 8 MB for Windows and 16 MB for UNIX.

Set the segment size to the total number of anticipated durable subscribers plus (single maximum message size/page size), plus a few pages for overhead. For example, for 100 anticipated subscribers and a maximum message size of 10,000 bytes on UNIX:

$$100 + (10,000 / 1024) = 198$$

Then add a few pages for overhead, so enter 210.

With this setting, there may only be one (100,000 byte) message in each segment. The ideal segment size depends on the circumstances. If the slowest subscriber lags behind the fastest publisher by a certain number of messages, you can set the segment size so that this number of messages will fit a single segment.

The JMS IQ Manager cleans up the database by recycling segments for which all messages have either expired or have been retrieved by their subscribers.

A lower segment size setting results in more efficient use of the disk because smaller segments turn over more rapidly and thus provide more effective use of server memory. However, a lower segment size means that more new segments may need to be allocated, which requires more time than freeing a cleaned-up segment. In addition, if a transaction is larger than the specified segment size, the server rolls back the transaction. You must then increase the **Segment size** property to an amount larger than the message, and apply the change to the Logical Host as described in [“Reconfiguring Message Servers” on page 112](#).

A high segment size setting can be advantageous in that cleanup runs less often; but each cleanup takes longer. However, cleaning up two small segments requires more time than cleaning up one large segment, so you can set a large segment size to increase performance on systems that are constrained by disk I/O speed rather than memory or space.

### Default

The default segment size is 16,384 pages, which is 8 MB for Windows and 16 MB for UNIX.

### Required Values

An integer indicating the number of pages from 1 through 999,999,999. Set this property to at least twice the total number of anticipated durable subscribers.



## Specifying the Throttling Threshold for the Total Number of Messages

You can specify the throttling threshold for the total number of messages for all topics and queues in the properties for JMS IQ Managers with the **Throttling threshold (total messages)** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Throttling threshold (total messages)** property specifies the maximum number of messages for all message destinations combined before the JMS IQ Manager starts throttling producers. For a detailed explanation and an example, see [“Throttling Producers” on page 32](#).

### Default

The default is 100,000 messages.

### Required Values

An integer from 0 through 999,999,999 indicating number of messages. If you specify 0, producers are never throttled.

## Specifying the Throttling Threshold for Message Destinations

You can specify the throttling threshold per message destination in the properties for JMS IQ Managers with the **Throttling threshold (per queue/topic)** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Throttling threshold (per queue/topic)** property specifies the maximum number of messages per topic or queue after which all producers of the message destination are throttled. Once a producer is throttled, the JMS IQ Manager stops reading messages from it until the number of messages it produces has dropped to below the threshold of the total of **Throttling threshold (per queue/topic) – Throttling lag**. For information about the **Throttling lag** property, refer to the section below.

### Default

The default is 1000 messages.

### Required Values

An integer from 0 through 999,999,999 indicating the number of messages. If set to 0, the publishers are never throttled.

## Specifying the Throttling Lag

You can specify the number of messages that must be dequeued before message producers are no longer throttled. To do so, you use the **Throttling lag** property in the JMS IQ Managers properties. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

You use the **Throttling lag** property combined with the **Throttling threshold (per queue/topic)** property. Once a producer is throttled, the JMS IQ Manager stops reading messages from it until the number of messages it produces has dropped to below the threshold of the total of **Throttling threshold (per queue/topic) – Throttling lag**. For information about the **Throttling threshold (per queue/topic)** property, refer to the section above.



## Default

The default is 1000 messages.

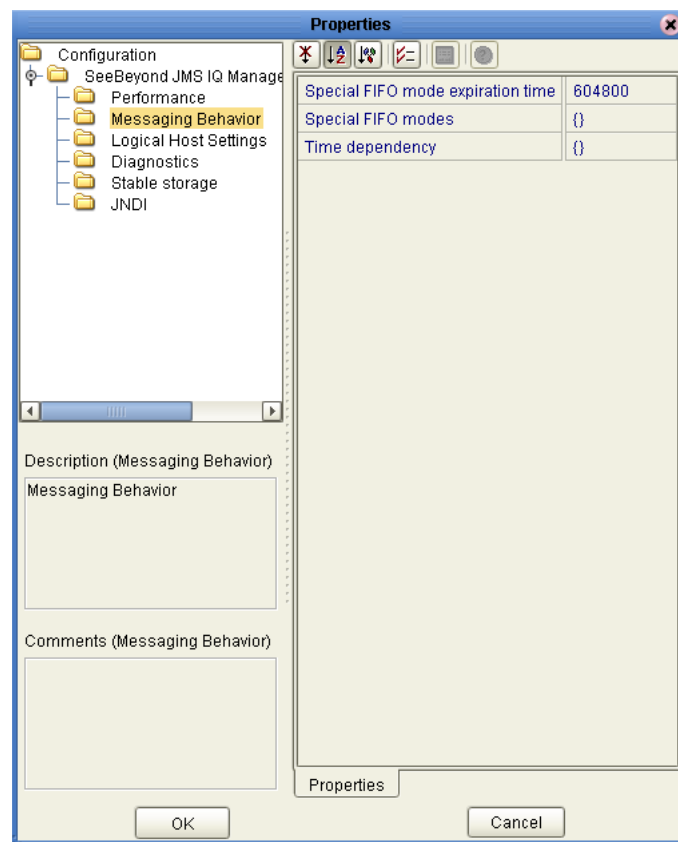
## Required Values

An integer from 0 through 99,999,999. The value must be set to less than that of the **Throttling threshold (per queue/topic)** property.

### 7.3.3 Setting Delivery Ordering Options

You set delivery order configurations for the JMS IQ Manager in the **Messaging Behavior** tab of the **JMS IQ Manager Properties** dialog box. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#). The figure below shows the **Messaging Behavior** tab.

**Figure 39** JMS IQ Manager—Delivery Order Configurations



For an overview of delivery ordering options, refer to [“Message Processing Order” on page 27](#).

## Specifying Special FIFO Mode Expiration Time

You can specify an expiration time for FIFO mode in the properties for JMS IQ Managers with the **Special FIFO mode expiration time** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

With this property, you specify the maximum number of seconds to delay a commit request in special FIFO mode.

#### Default

The default FIFO mode expiration time is 604800 seconds (168 hours).

## Specifying Special FIFO Modes

You can specify the FIFO message delivery order in the properties for JMS IQ Managers with the **Special FIFO modes** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

You can specify three different modes for FIFO delivery modes: fully concurrent, protected concurrent, or fully serialized. Use the following syntax to specify the mode for each message destination:

`{destination:mode}`

where *destination* is the name of the topic or queue and where *mode* is 0 for fully concurrent, 1 for protected concurrent, and 2 for fully serialized.

For example: `{billingtopic:1;admittancetopic:2}`

In fully concurrent mode, consumers can retrieve messages from a destination only when all other messages have been received or are in the process of being received. Consumers can then commit messages without restrictions. This means that the messages can be committed out of sequence, for example, a cancellation may be committed before the order is committed. This type of delivery order is not always desirable.

In protected concurrent mode, the consumer can retrieve messages just as in fully concurrent mode (only after all messages have been received or are being received), but messages can only be committed if all previous messages have been committed.

In fully serialized mode, consumers can only retrieve messages after all previous messages for the message destination have been received *and* committed.

To implement serialized mode across multiple integration servers, you must set the JMS clients for the consumers involved to serial mode. For more information, refer to [“Setting JMS Client Message Processing Order” on page 79](#).

For a general overview of message processing order, refer to [“Message Delivery Order” on page 14](#).

#### Default

By default, all message destinations use fully concurrent delivery mode (0).

## Specifying Delivery Order by Time Dependency Group

You can specify the message delivery order in the properties for JMS IQ Managers with the **Time dependency** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

By default, messages are processed by and delivered to Collaborations in the order in which they were created (committed to their destination), independent of messages

associated with any other destination. When you use the **Time dependency** property, ordering causes the message processing order to be dependent on messages associated with *other* destinations. These destinations are specified as the time dependency group using the **Time dependency** property.

Messages associated with any of the destinations in the time dependency group are ordered in fully serialized mode. In other words, a message associated with a destination in this group is processed only after all older messages associated with any other destination in the time dependency group have been processed.

For a general overview of message processing order, refer to **“Message Delivery Order” on page 14**.

### Required Values

**T:TopicA;T:TopicB;Q:QueueA;Q:QueueB;...;T:TopicN;Q:QueueN**

where *TopicA*, *TopicB*, and *TopicN* are the names of topics. *QueueA*, *QueueB*, and *QueueN* are the queues. You can specify any combination of topics and/or queues.

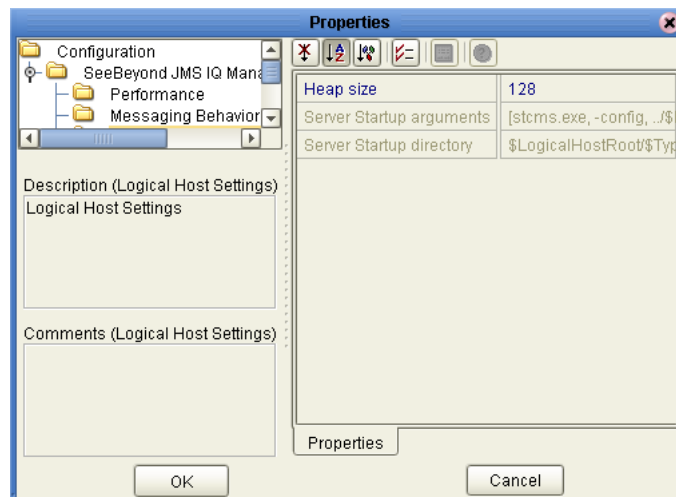
If you specify a message destination that does not exist, Enterprise Manager enables time-based order for all other destinations and ignores the unknown name. This allows you to add topics and queues in your project at a later time.

*Note:* When you specify time dependency, you cannot use a colon (:) or semicolon (;) in topic or queue names because they are already used for the time dependency value.

## 7.3.4 Setting Logical Host Options

You set Logical Host settings as they relate to the JMS IQ Manager in the **Logical Host Settings** tab of the **JMS IQ Manager Properties** dialog box. To display JMS IQ Manager properties, refer to **“Configuring Message Servers” on page 88**. The figure below shows the **Logical Host Settings** tab.

**Figure 40** JMS IQ Manager—Logical Host Configurations



## Specifying Heap Size

You can specify Logical Host heap size in the properties for JMS IQ Managers with the **Heap size** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Heap size** property specifies the maximum amount of memory allocated in the Logical Host for the Java Virtual Machine (JVM).

### Default

The default is 128 MB.

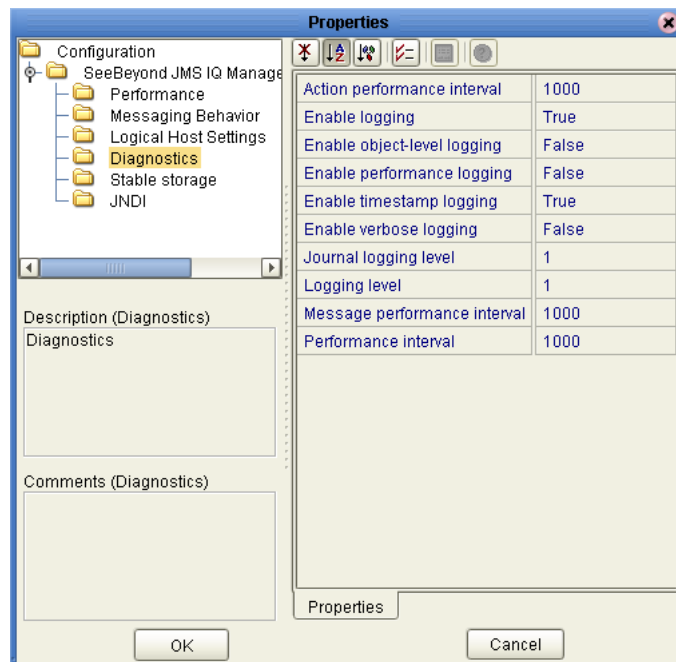
### Required Values

An integer in MB between 120 and the maximum amount available on the system.

## 7.3.5 Setting Diagnostic Options

You set diagnostic configurations for the JMS IQ Manager in the **Diagnostics** tab of the **JMS IQ Manager Properties** dialog box. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#). The figure below shows the **Diagnostics** tab.

**Figure 41** JMS IQ Manager—Diagnostic Configurations



## Specifying the Action Interval for Performance Statistics Logging

You can specify the action interval in the properties for JMS IQ Managers with the **Action interval** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Action interval** property specifies how many times the ActionManager's Update function is called before it logs its performance statistics.

#### Default

The default value is 1000 messages.

#### Required Values

An integer larger than 0.

### Enabling Logging

You can enable JMS IQ Manager logging in the properties for JMS IQ Managers with the **Enable logging** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Enable logging** property enables or disables logging. When you enable logging, JMS IQ Manager diagnostic information is written to a JMS IQ Manager log file. The name of the log file is *JMSIQMgrname.log*, where *JMSIQMgrname* is the name of the JMS IQ Manager. This log file resides in the *logicalhostname\logicalhost\logs* directory, where *logicalhostname* is the name of the Logical Host.

The log file can hold up to 10 MB of diagnostic information. If the JMS IQ Manager attempts to log more than 10 MB, the log file is renamed to *JMSIQMgrname.log.1*, and a new file is created with the name *JMSIQMgrname.log.2*.

#### Default

The default setting for the **Enable logging** property is **True**; JMS IQ Manager logging is enabled by default.

#### Required Values

**True** or **False**.

### Enabling Object-level Logging

You can enable object logging in the properties for JMS IQ Managers with the **Enable object-level logging** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Enable object logging** property specifies whether the JMS IQ Manager log includes object logging. Object information constitutes constructor and deconstructor information for objects. This information may be helpful to deduce when objects were created and deleted. This logging level is primarily used as a diagnostic tool when a JMS IQ Manager terminates unexpectedly.

For the JMS IQ Manager to create a log, the **Enable logging** property must be enabled as described in [“Enabling Logging” on page 101](#). The **Enable logging** property is enabled by default.

#### Default

The default setting for the **Enable object logging** property is **False**; object logging is disabled by default.

## Required Values

**True or False.**

## Enabling Performance Logging

You can have the JMS IQ Manager log include performance information in the properties for JMS IQ Managers with the **Enable performance logging** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

When performance logging is enabled, the JMS IQ Manager log includes performance information such as how long it takes to process messages and to write to disk.

For the JMS IQ Manager to create a log, the **Enable logging** property must be enabled as described in [“Enabling Logging” on page 101](#). The **Enable logging** property is enabled by default.

### Default

The default setting for the **Enable performance logging** property is **False**; performance logging is disabled by default.

## Required Values

**True or False.**

## Enabling Timestamp Logging

You can have the JMS IQ Manager log include timestamp information in the properties for JMS IQ Managers with the **Enable timestamp logging** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

When timestamp logging is enabled, the JMS IQ Manager log includes timestamp information for every event logged. This may help determine how event relate to each other.

For the JMS IQ Manager to create a log, the **Enable logging** property must be enabled as described in [“Enabling Logging” on page 101](#). The **Enable logging** property is enabled by default.

### Default

The default setting for the **Enable timestamp logging** property is **False**; timestamp logging is disabled by default.

## Required Values

**True or False.**

## Enabling Verbose Logging

You can have the JMS IQ Manager log include full-length messages in the properties for JMS IQ Managers with the **Enable verbose logging** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

For the JMS IQ Manager to create a log, the **Enable logging** property must be enabled as described in [“Enabling Logging” on page 101](#). The **Enable logging** property is enabled by default.

### Default

The default setting for the **Enable verbose logging** property is **False**; verbose logging is disabled by default.

### Required Values

**True or False.**

## Specifying the Journal Logging Level

You can specify the type of journaling errors to be included in the JMS IQ Manager journal log in the properties for JMS IQ Managers with the **Journal Logging level** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Journal logging level** property specifies the threshold severity level at which the system issues informational, warning, and error messages.

Set this property to 0 for maximum debugging; this setting logs all messages. Specify a setting of 1 to log warning messages, error messages, and fatal messages. To log only error messages and fatal messages, set this property to 2. To log only fatal messages, specify 3.

By default, the journal log file resides in the journal directory in the JMS IQ Manager directory. The journal directory can be specified with the **Journal directory** property as described in [“Specifying the Location of the Journal Database” on page 107](#).

### Default

The default is 1; the JMS IQ Manager journal log includes warning messages, error messages, and fatal messages.

### Required Values

0, 1, 2, or 3.

## Specifying the Logging Level

You can specify the type of errors to be included in the JMS IQ Manager log in the properties for JMS IQ Managers with the **Logging level** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Logging level** property specifies the threshold severity level at which the system issues informational, warning, and error messages.

Set this property to 0 for maximum debugging; this setting logs all messages. Specify a setting of 1 to log warning messages, error messages, and fatal messages. To log only error messages and fatal messages, set this property to 2. To log only fatal messages, specify 3.



For the JMS IQ Manager to create a log, the **Enable logging** property must be enabled as described in [“Enabling Logging” on page 101](#). The **Enable logging** property is enabled by default.

#### Default

The default is 1; the JMS IQ Manager log includes warning messages, error messages, and fatal messages.

#### Required Values

0, 1, 2, or 3.

### Specifying the Message Interval for Performance Statistics Logging

You can specify the message interval in the properties for JMS IQ Managers with the **Message interval** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Message interval** property specifies the number of messages the JMS IQ Manager processes before it logs its performance statistics.

#### Default

The default value is 1000 messages.

### Specifying the Interval for Performance Statistics Logging

You can specify the performance interval in the properties for JMS IQ Managers with the **Performance interval** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Performance interval** property specifies how many `IMessages` the `IMessageManager` needs to create before it logs its performance statistics.

#### Default

The default is 1000 messages.

#### Required Values

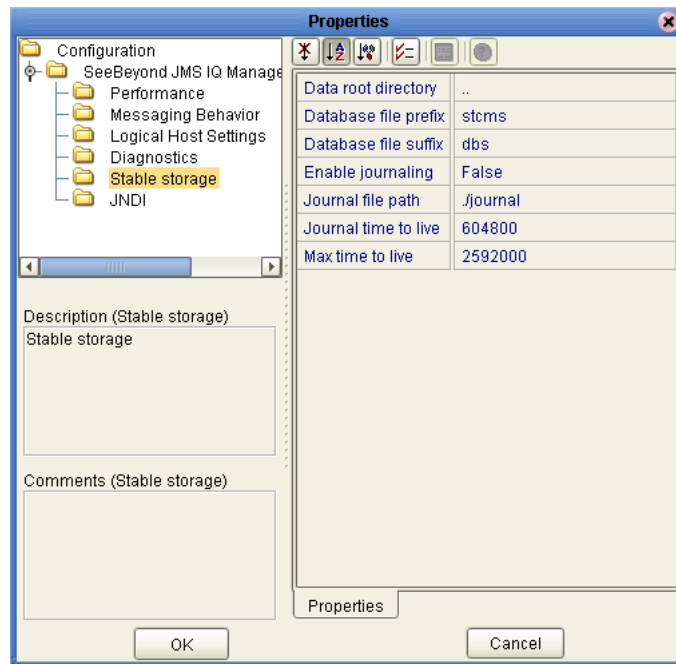
An integer larger than 0, indicating the number of messages.

## 7.3.6 Setting Stable Storage and Journaling Options

You set storage and journaling configurations for the JMS IQ Manager in the **Stable Storage** tab of the **JMS IQ Manager Properties** dialog box. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#). The figure below shows the **Stable Storage** tab.



**Figure 42** JMS IQ Manager—Stable Storage and Journaling Configurations



Journaling messages allows you to republish messages at a later date. You can journal messages by configuring the JMS IQ Manager for the following:

- Enabling journaling (journaling is disabled by default)
- Specifying the amount of time for journaled messages to live (24 hours by default)
- Specifying the location for the journal database

To republish journaled messages, you use the STC MS Control utility as described in [“Republishing Messages from Topics” on page 129](#) or [“Republishing Messages from Queues” on page 130](#). You can also use the STC MS Control utility to browse journaled messages with the `-journaler` flag. For information, refer to [“Browsing Journaled Messages” on page 130](#).

## Specifying the Location of JMS IQ Manager Database Files

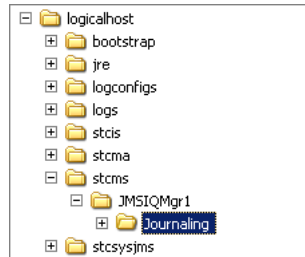
You can specify where you want to JMS IQ Manager database files to reside in the properties for JMS IQ Managers with the **Data directory** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The data directory holds the JMS IQ Manager database files (`stcms*.dbms`). You can specify the location as an absolute path or a path relative to the `logicalhost\stcms\bin` directory. Using an absolute path for the data directory allows you to store the JMS IQ Manager files on a different system, for example, for backup purposes.

If journaling is enabled, the data directory contains a **Journaling** directory, unless another location has been specified for the **Journal directory** property. The Journaling directory holds the journaling database files. For information, refer to [“Specifying the Location of the Journal Database” on page 107](#). Journaling is disabled by default.

The figure below shows the directory structure for a data root defined as `..` as in the default setting.

**Figure 43** JMS IQ Manager Database Files Location—Data Directory



### Default

The default setting is `..` (one directory up), which stores the database files in the following directory:

*logicalhostname* \ *logicalhost* \ *stcms* \ *JMSIQMgrname*

where *logicalhostname* is the name of the Logical Host and where *JMSIQMgrname* is the name of the JMS IQ Manager.

### Specifying the Prefix for Database Filenames

You can specify the database filename prefix in the properties for JMS IQ Managers with the **Database file prefix** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Database file prefix** property specifies the prefix for the JMS IQ Manager database filenames. The database files are used to store persistent messages.

### Default

The default value is **stcms**; database filenames are **stcms\*.dbs** by default.

### Specifying the File Extension for Database Files

You can specify the database file extension in the properties for JMS IQ Managers with the **Database file extension** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Database file extension** property specifies the file extension for the JMS IQ Manager database files used for persistent message storage.

### Default

The default database file extension is **dbs**; database filenames are **stcms\*.dbs** by default.

## Enabling Journaling

You can enable or disable journaling in the properties for JMS IQ Managers with the **Enable journaling** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

When you enable journaling, every inbound message is automatically copied to the journal database. The message is then held in the journal database for the duration of time to live specified for journaled messages. By default, the expiration time for a journaled message is 24 hours. To change the time to live for journaled messages, refer to [“Specifying Time To Live for Journaled Messages” on page 108](#). The journaled message time to live is completely independent of when the live counterpart of the message is consumed by its publisher.

When a journaled message expires, it is not deleted from the journal database—it remains there until you back up the topics or queues. When you back up, all messages in the journal database are included in the archive, and the journal expired messages are removed from the journal database. It is recommended that you back up daily when journaling is enabled; otherwise, the journal database retains journal expired messages and may grow exceedingly large. Because the journal database and the JMS IQ Manager database are located on the same system, it is important to avoid running out of disk space.

To back up, you use the MS Control Utility. For more information, refer to [“Backing Up” on page 131](#).

When messages are in the journal database, you can view them but not edit them. You can use either the Enterprise Manager or the STC MS Control utility to view and republish journaled messages. For more information about the MS Control utility, see [“Managing JMS IQ Managers with the MS Control Utility” on page 123](#).

### Default

The default setting for the **Enable journaling** property is **False**; journaling is disabled by default.

### Required Values

**True or False.**

## Specifying the Location of the Journal Database

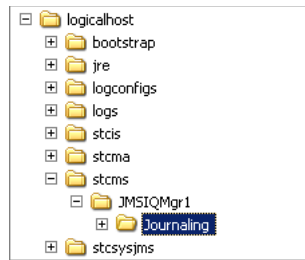
You can specify the location of the **Journaling** directory in the properties for JMS IQ Managers with the **Journal directory** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Journaling directory** holds the journal database files and the journaling log file. You can enter an absolute path or a path relative to the **logicalhost\bootstrap\bin** directory. Using an absolute path for the data directory allows you to store the journal database files on a different system, for example, for backup purposes.

The JMS IQ Manager only creates Journaling directory when journaling is enabled. Journaling is disabled by default. For more information, refer to [“Enabling Journaling” on page 107](#).

The figure below shows the location of the **Journaling** directory for the default setting.

**Figure 44** JMS IQ Manager Database Files Location—Data Directory



### Default

The default setting is `../stcms`, which stores the journal database files and the journaling log file in the following directory:

***logicalhostname*\logicalhost\stcms\JMSIQMgrname\Journaling**

where *logicalhostname* is the name of the Logical Host and where *JMSIQMgrname* is the name of the JMS IQ Manager.

## Specifying Time To Live for Journalled Messages

You can specify the time to live for journalled messages in the properties for JMS IQ Managers with the **Journal time to live** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Journal time to live** property specifies the maximum amount of time in seconds that a journalled message lives before it expires. The JMS IQ Manager journals messages only when journaling is enabled as described in [“Enabling Journaling” on page 107](#). Journaling is disabled by default.

When a journalled message expires, it is not deleted from the journal database—it remains there until you back up the topics or queues. When you back up, all messages in the journal database are included in the archive (.zip file), and the journal expired messages are removed from the journal database.

To back up, you use the MS Control Utility. For more information, refer to [“Backing Up” on page 131](#).

### Default

The default is 86400 seconds (24 hours).

### Required Values

An integer from 0 through 999,999,999 indicating seconds. If you specify 0, journalled messages expire immediately.

## Specifying Time To Live for Live Messages

You can specify the message expiration time in the properties for JMS IQ Managers with the **Max time to live** property. To display JMS IQ Manager properties, refer to [“Configuring Message Servers” on page 88](#).

The **Max time to live** property specifies the maximum amount of time (in seconds) before a live message expires. After it expires, the message is removed from the queue whether it has been consumed or not. If you specify 0, the message never expires.

#### Default

The default is 2592000 seconds (30 days).

#### Required Values

An integer from 0 through 999,999,999 indicating seconds.

---

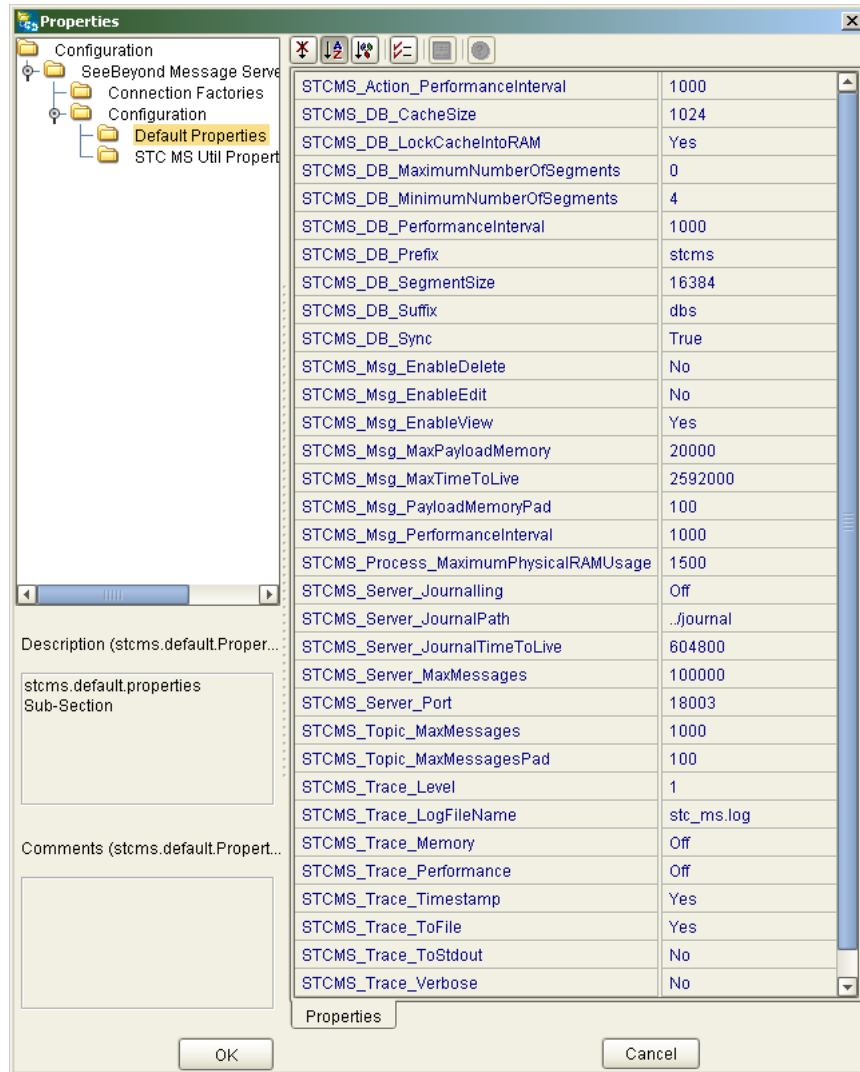
## 7.4 4.5.x SeeBeyond Message Server Configurations

If your Project connects to a 4.5.x SeeBeyond Message Server, and you have added a 4.5.x SeeBeyond Message Server to your Environments in the Enterprise Designer, you can configure the message server as described in [“Configuring Message Servers” on page 88](#).

To be able to add and configure a 4.5.x SeeBeyond Message Server, you must have installed `stcmessageserver45x.sar` as described in [“Installing Message Server .sar Files” on page 35](#).

The figure below shows the 4.5.x Message Server Properties dialog box.

**Figure 45** 4.5.x Message Server Properties Dialog Box



## 7.5 Deploying Topics and Queues to Message Servers

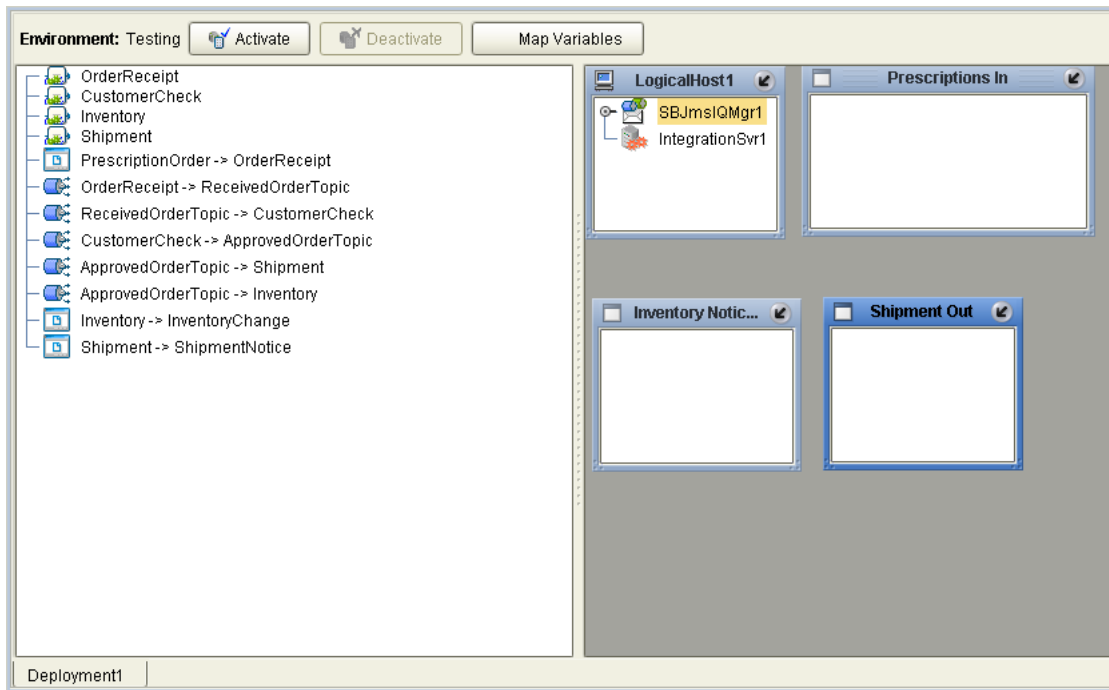
Deploying message destinations to message server is the last step in building an eGate Project, after you have created the Environment and configured the Environment components. To deploy message destinations, you create a Deployment Profile as described below.

**Note:** *Attempting to create JMS IQ Manager under a Logical Host with an Integration Server while other users have their Logical Host with an Integration Server and JMS IQ Manager checked out, causes an error stating the Logical Host is checked out. In such an instance, create the JMS IQ Manager before you create the Integration Server.*

### To deploy topics and queues to message servers

- 1 In the Project Explorer tab of the Enterprise Designer, right-click the Project, click **New**, and click **Deployment Profile**.
- 2 Enter the name of the profile, click the Environment this profile uses, and click **OK**. The Deployment Profile displays, showing the components in the Environment, and the Connectivity Map components.

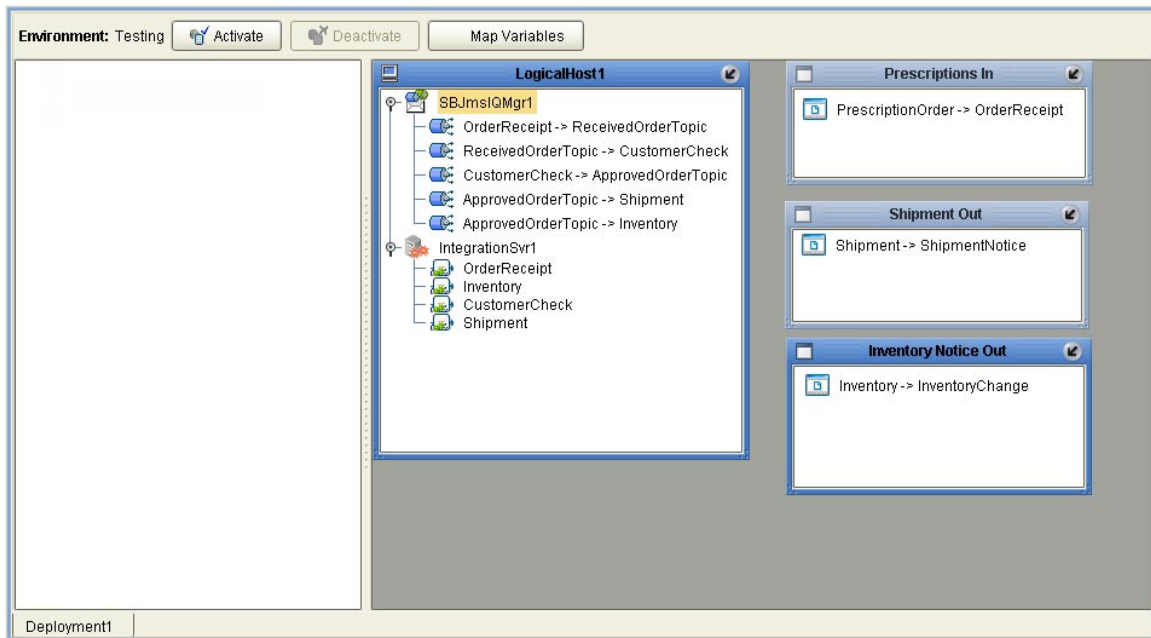
**Figure 46** Initial Deployment Profile



- 3 Drag the topics and queues from the Deployment Profile tab to the message server, being cautious not to split inbound and outbound message destinations across separate servers.
- 4 Drag the Collaborations to the appropriate integration servers.
- 5 Drag the eWay components to the eWays.

The figure below shows an example of a deployment profile.

**Figure 47** Deployment Profile Example



You can now activate the deployment profile and run the Project. For detailed information about activation and running Projects, refer to the *eGate Integrator User's Guide*.

## 7.6 Reconfiguring Message Servers

To reconfigure an existing message server, you change the configuration and then apply the changes to the Logical Host as described below. If you also made changes to the Project, for example, Connectivity Map components such as JMS client properties, you must also reactivate the Deployment Profile. The procedure below describes how you reconfigure message servers.

### To reconfigure message servers

- 1 In the **Environment Explorer** tab of the Enterprise Designer, expand the Environment and the Logical Host for which you want to reconfigure a message server.
- 2 Right-click the message server to be reconfigured and click **Properties**. The **Properties** dialog box appears.
- 3 Change the properties as necessary. The table below shows where to find information about the configuration options.

Message Server Type	Configuration
JMS IQ Manager	page 88
4.5.x SeeBeyond Message Server	page 109



- 4 Click **OK** to accept the changes and close the **Properties** dialog box.
- 5 Right-click the Logical Host that contains this message server.
- 6 Click **Apply**.

# Managing JMS IQ Managers

eGate Integrator provides two tools to manage the JMS IQ Managers: the Enterprise Manager and the STC Message Server Control command line utility.

This chapter describes each tool, the differences between the tools, and how to manage message servers with these tools.

## In This Chapter:

- [About Administration Tools](#) on page 114
- [Managing JMS IQ Managers with the Enterprise Manager](#) on page 114
- [Managing JMS IQ Managers with the MS Control Utility](#) on page 123

---

## 8.1 About Administration Tools

eGate Integrator provides two administration tools for JMS IQ Manager management:

- Enterprise Manager
- STC MS Control utility

The Enterprise Manager is a web application that allows you to start and stop message servers, monitor message destinations, and view message properties and payloads. For information, refer to [Managing JMS IQ Managers with the Enterprise Manager](#) on page 114.

The STC MS Control utility is a command-line utility that enables you to manage many advanced aspects of the JMS IQ Managers. For information, refer to [Managing JMS IQ Managers with the MS Control Utility](#) on page 123.

---

## 8.2 Managing JMS IQ Managers with the Enterprise Manager

With the Enterprise Manager, you can manage JMS IQ Managers as follows:

- [Starting and Stopping JMS IQ Managers](#) on page 116
- [Monitoring Topics and Queues](#) on page 117

- [Viewing Message Properties](#) on page 119
- [Viewing Message Payload](#) on page 123

The section below describe how to use the Enterprise Manager to manage JMS IQ Managers.

## 8.2.1 Starting Enterprise Manager

The procedure below describes how you start Enterprise Manager.

### To start Enterprise Manager

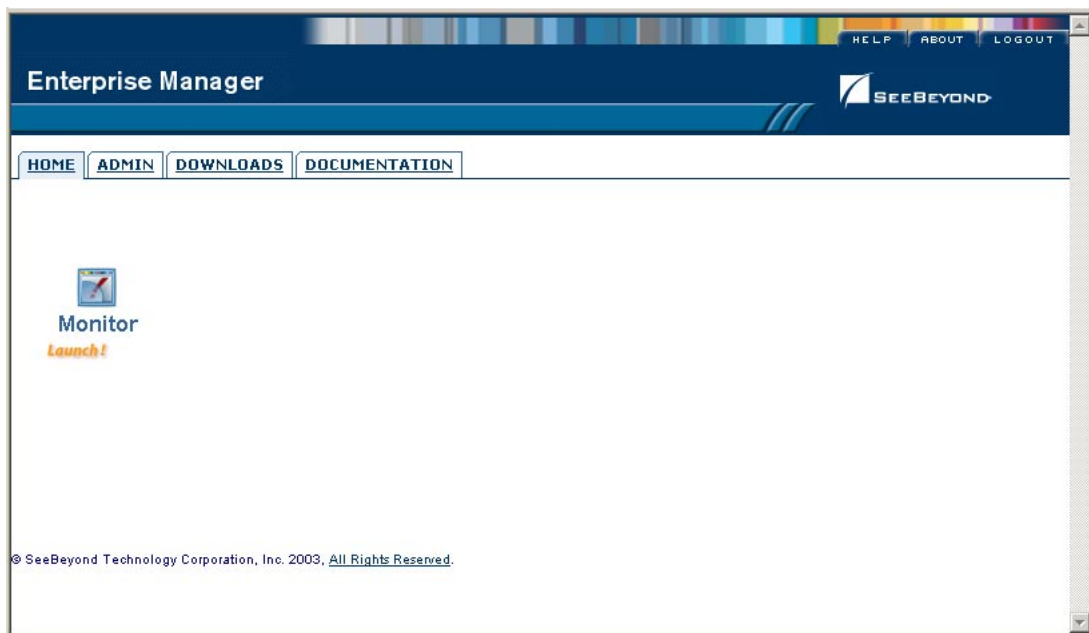
- 1 On the Windows **Start** menu, click **Run**.
- 2 Enter the following and click **OK**:

**http://<computername>:<portnumber>**

where <computername> is the computer name or the TCP/IP address of the system with the Logical Host and where <portnumber> is the port number of the Logical Host.

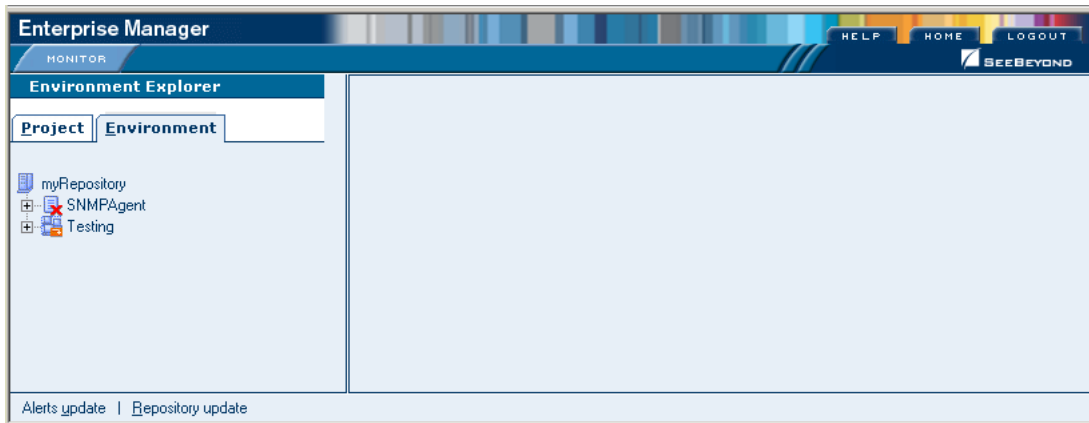
- 3 Enter your user name and password and click **Login**. The Enterprise Manager home page appears.

**Figure 48** Enterprise Manager Home Page



- 4 Click the **Monitor** icon. The **Monitor** tab of the Enterprise Manager appears.

**Figure 49** Enterprise Manager—Monitor Tab



You use the left pane in Enterprise Manager to start and stop message servers as described in the section below. You can also use it to select topics and queues to monitor messages as described in [“Monitoring Topics and Queues” on page 117](#).

## 8.2.2 Starting and Stopping JMS IQ Managers

To start and stop JMS IQ Managers

- 1 Start Enterprise Manager as described in [“Starting Enterprise Manager” on page 115](#).
- 2 Expand the Environment and Logical Host for which you want to monitor message servers.
- 3 Right-click the JMS IQ Manager you want to start or stop.

**Figure 50** Starting or Stopping Message Servers



- To start the JMS IQ Manager, click **Start Server**.  
To stop the JMS IQ Manager, click **Stop Server**.

### 8.2.3 Monitoring Topics and Queues

You can monitor message traffic in topics and queues using the Enterprise Manager.

Message servers distinguish between topics and queues in the following way:

- A **topic** conforms to the *publish-and-subscribe* (pub/sub) messaging domain, where one *publisher* broadcasts messages to *potentially many subscribers*. When the Message Server publishes a message on a topic, it ensures that all subscribers receive the message.
- A **queue** conforms to the *point-to-point* (p2p, or PTP) messaging domain, where one *sender* delivers message to *exactly one receiver*. When the message server sends a message to a queue, it ensures it is received once and only once, even though there may be many receivers “listening” to the queue. This is equivalent to the subscriber pooling in other queue implementations.

Except for this distinction between pub/sub and PTP, topics and queues are otherwise quite similar:

- Each topic or queue maintains a *sequence* of messages in progress; each message has a timestamp called its *enqueue time* that indicates when it was published or sent.
- Messages are held in virtual memory until delivered; the difference between the *highest sequence number* and the *lowest sequence number* is an indicator of how much memory is being used by that topic/queue.

- Messages that have been read and committed by their subscribers/receivers are subject to *cleanup*. After cleanup, the lowest sequence number is increased by the number of messages that were delivered and successfully committed.

**To monitor topics and queues**

- 1 Start Enterprise Manager as described in **“Starting Enterprise Manager” on page 115**.
- 2 Expand the Environment, the Logical Host, and the message server for which you want to view topics or queues.
- 3 Click the topic or queue to be viewed. The right pane shows the topics or queues for the message server as shown below.

**Figure 51** Viewing Topics



The table below describes the topic properties shown in the Enterprise Manager.

**Table 10** Topic Properties

Property Name	Description
Topic Name	Name of the topic.
Min Seq #	Sequence number of the oldest message available for this topic. If there are no messages available, this field shows the sequence number of the last message processed.
Max Seq #	Sequence number of the most recent message available for this topic. If there are no messages available, this field shows the sequence number of the last message processed.
Avail. Count	Number of messages for this topic that are still unprocessed by at least one subscriber.

**Table 10** Topic Properties (Continued)

Property Name	Description
# Subscribers	Number of subscribers <i>registered to consume</i> messages for this topic (including durable subscribers that are currently disconnected).
Last Pub Date/Time	The date and timestamp of the most recent message currently available in the topic. This field is blank when there are no messages available.

The table below describes the queue properties shown in the Enterprise Manager.

**Table 11** Queue Properties

Property Name	Description
Queue Name	The name for this queue.
Min Seq #	Sequence number of the oldest message available for this queue. If there are no messages available, this field shows the sequence number of the last message processed.
Max Seq #	Sequence number of the most recent message available for this queue. If there are no messages available, this field shows the sequence number of the last message processed.
# Receivers	Number of receivers for this queue.
Avail. Count	Number of messages in this queue that are unprocessed.
Last Pub Date/Time	The date and timestamp of the most recent message currently available in the queue. This field is blank when there are no messages available.

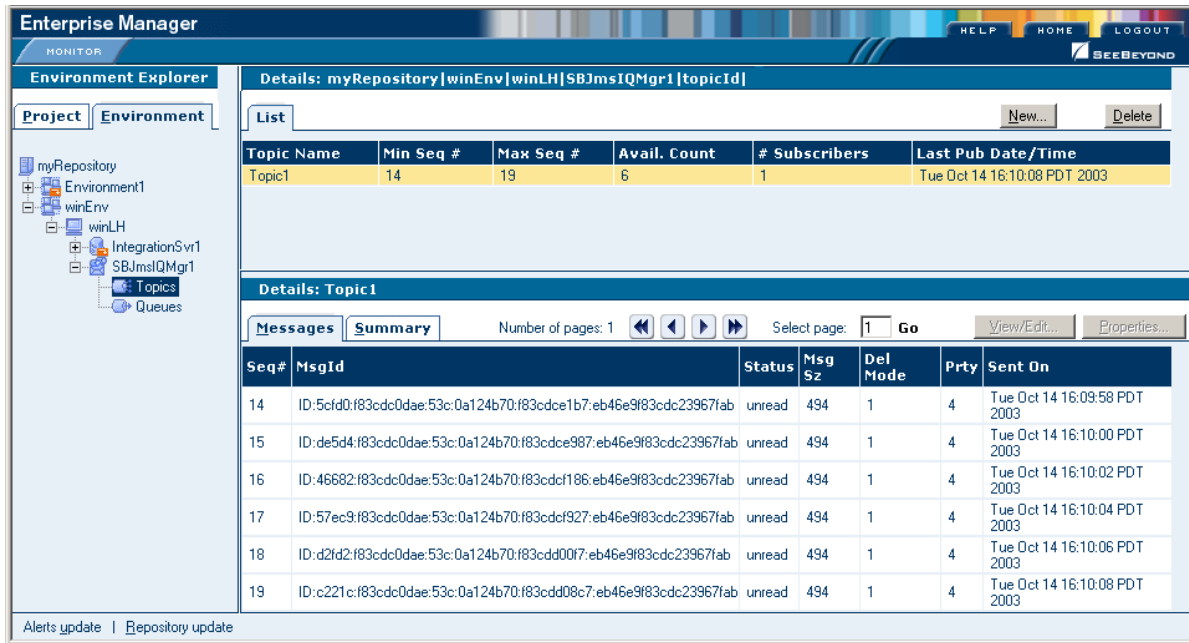
## 8.2.4 Viewing Message Properties

You can view message properties in the Enterprise Manager. The procedure below describes how you view the messages, and provides a list of the message properties with descriptions.

To view message properties

- 1 In the **Monitor** tab of the Enterprise Manager, select the topic or queue for which you want to view a message as described in **“Monitoring Topics and Queues” on page 117**.
- 2 Under **List**, click the topic or queue for which you want to view message properties.

Figure 52 Viewing Message Properties



- Under **Details: Topics**, scroll to the message to see its properties. The table below describes the properties listed.

Table 12 Message Properties

Property Name	Description
idx	Index number for the message.
MsgId	Unique identification number for the message.
Seq #	Sequence number of the message.
Status	Message status.
Pyld Sz	Total byte size of the stored message.
Prty	Priority of the message.
Sent On	Date and time that the message was sent.

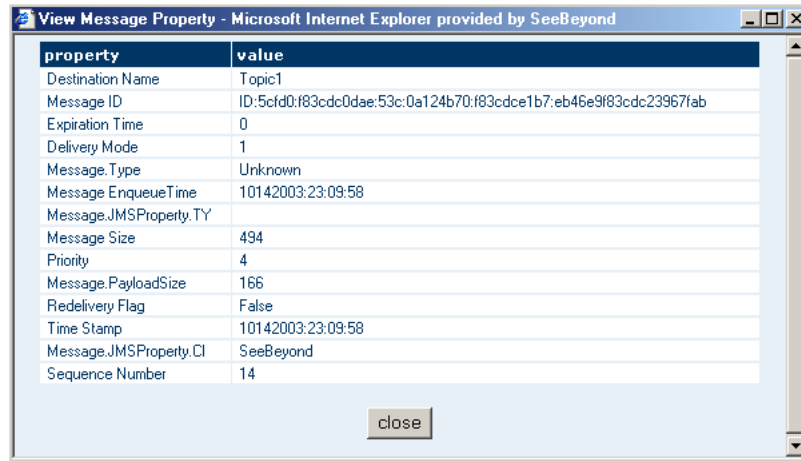
## 8.2.5 Viewing Message Details

To view message details

- In the **Monitor** tab of the Enterprise Manager, select the topic or queue for which you want to view a message as described in **“Viewing Message Properties” on page 119**.
- In the **Message** tab, click the message for view you want to view details.
- Click **Properties**. The **View Message Property** window appears.



**Figure 53** Viewing Message Details



The table below describes the message details listed.

**Table 13** Message Details

Property Name	Description
Destination Name	The name of the topic or queue.
Message ID	Unique identification number for the message.
Expiration Time	The message time to live in seconds.
Delivery Mode	1 indicates a persistent message, 2 indicates a nonpersistent message.
Message Type	The message type of the message, such as text or bytes.
Message Enqueue Time	The date and time when the message was received by its message destination. For example, an enqueue time of 09302003:21:53:44 indicates that the message was enqueued at 9:53pm on Sept. 30, 2003.
Message JMS Property TY	JMS property.
Message Size	The size of the message including the JMS header in bytes.
Priority	Priority of the message from 0 to 9, which 9 as the highest priority.
Message Payload Size	The message payload size in bytes.
Redelivery Flag	Indicates whether this message is set for redelivery.
Message JMS Property Cl	JMS property.
Sequence number	The sequence number of the message.

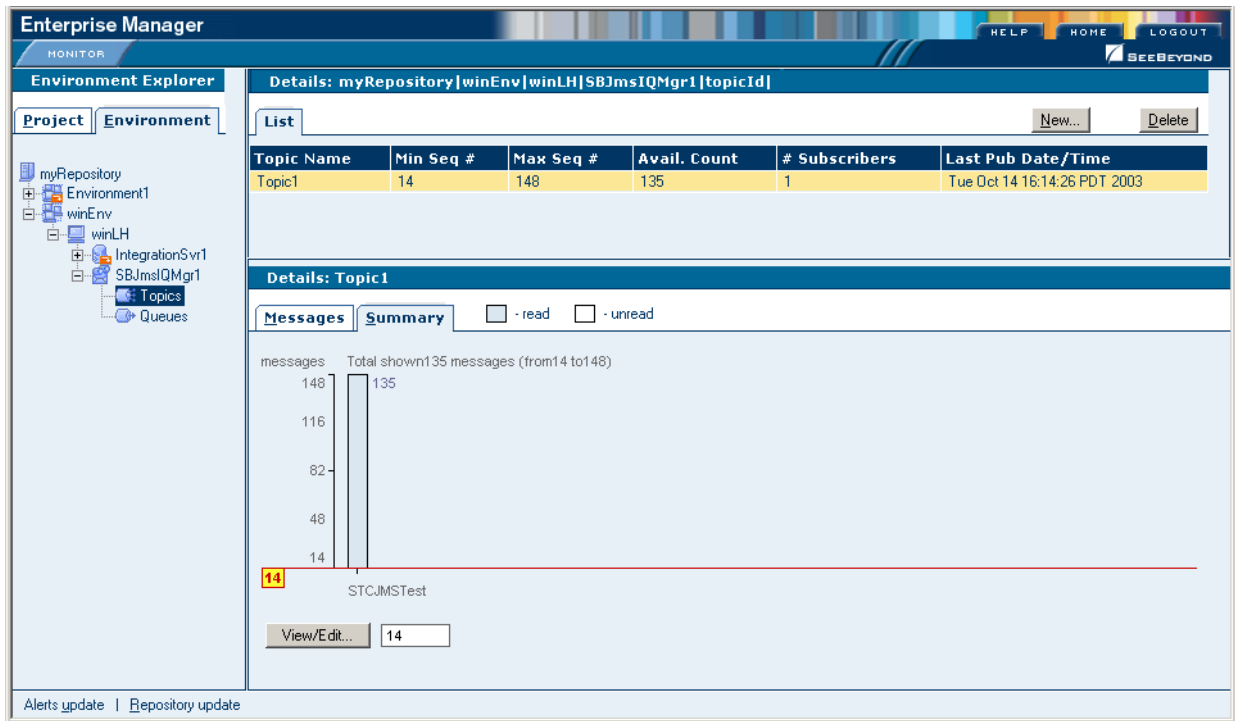
### 8.2.6 Viewing Message Destination Summaries

You can view the summary for a topic or queue as described below.

To view message destination summaries

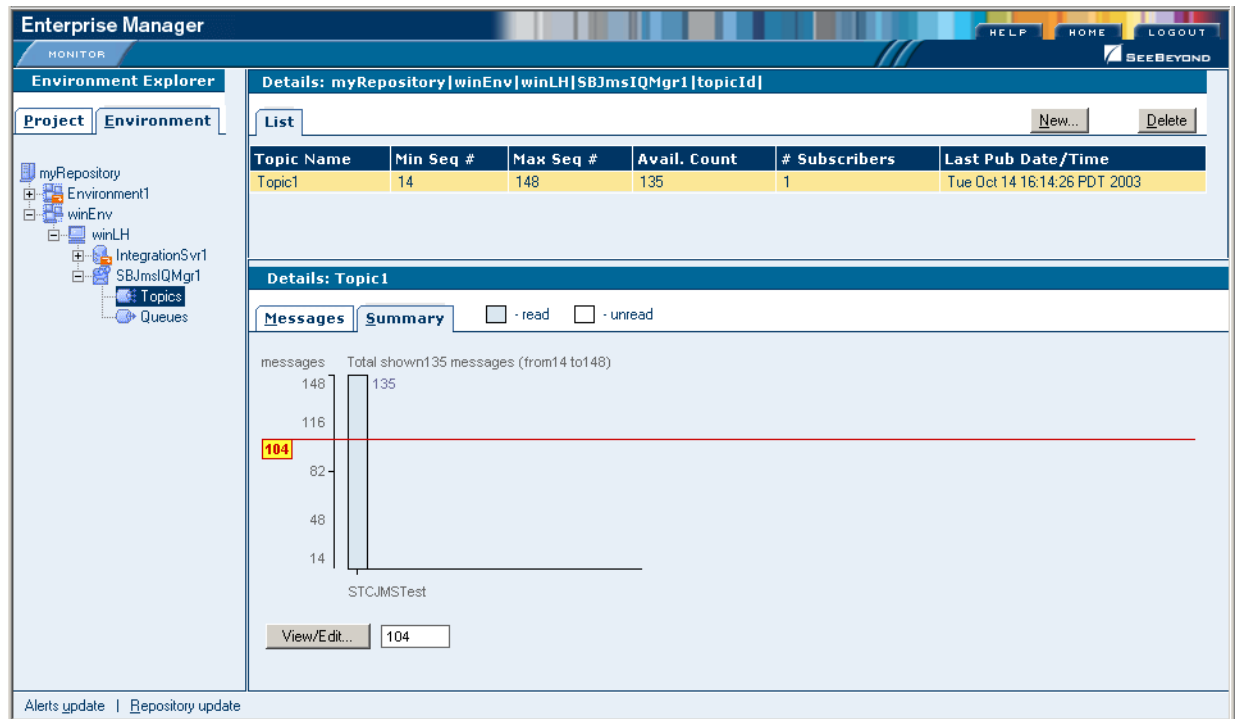
- 1 Find the topic or queue for which you want to view its summary as described in [“Viewing Message Properties” on page 119](#).
- 2 Click **Summary**. The **Summary** page appears.

Figure 54 Summary Page



- 3 To select a message, you can enter the message number in the **View/Edit** box, or you can drag the message slider as shown below.

Figure 55 Using the Message Slider to Select a Message



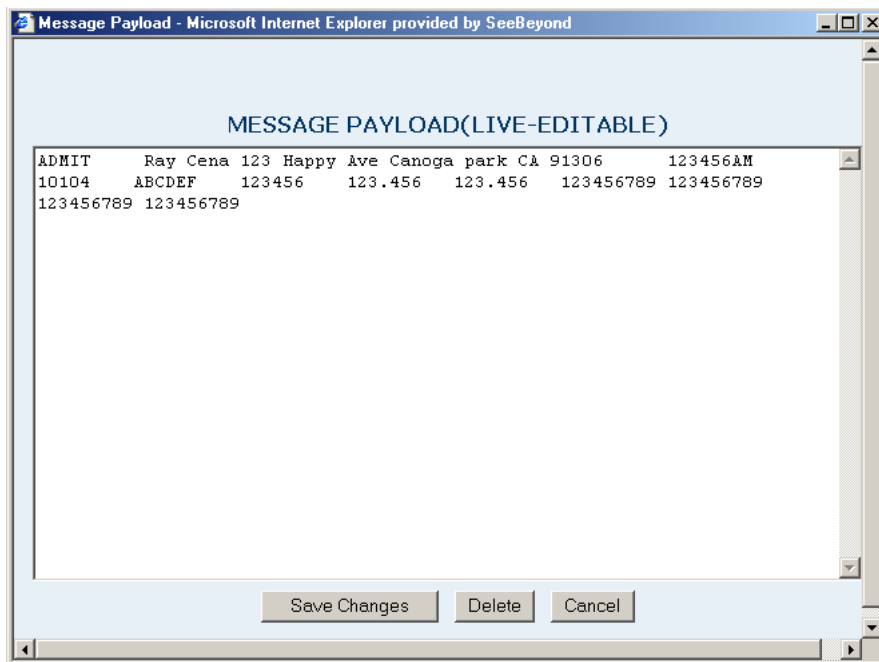
## 8.2.7 Viewing Message Payload

You can view the payload of text messages using Enterprise Manager as described below.

To view message payload

- 1 Find the message for which you want to view the payload as described in [“Viewing Message Properties” on page 119](#).
- 2 Under **Details: Topics**, click the message for which you want to view the payload.
- 3 Click **View/Edit**. The **Edit Payload for Message** dialog box appears.

**Figure 56** Viewing Message Payload



---

## 8.3 Managing JMS IQ Managers with the MS Control Utility

With the MS Control utility you can manage JMS IQ Managers as follows:

- Display the version of the MS server or of the utility.
- Shut down the server.
- For a specified server: List, create, or delete topics or queues.
- For a specified topic: List, create, or delete subscribers, retrieve a topic message list, or view topic statistics.
- For a specified queue: List, create, or delete receivers, retrieve a queue message list, or view queue statistics.
- For a specified message: View, delete, or modify message content.

- Create, delete, modify, monitor, or list the contents of a particular queue or topic.
- View or modify a particular message or its type.
- Fetch or delete a range of messages.
- Journal, back up, and archive messages.

Preferences for the MS Control utility are stored in the **stcmsutil.Properties** file. See [“stcmsutil.Properties” on page 132](#).

### 8.3.1 Syntax

The MS Control utility has the following syntax for all flags other than **--help** and **--version**:

```
stcmsctrlutil -host hostname -port portnumber [-offset portoffset]
               -flag

stcmsctrlutil -host hostname -port portnumber [-offset portoffset]
               -flag argument1 [argument2 [argument3]]
```

If JMS IQ Manager security is enabled, you must specify a user name and password as follows:

```
stcmsctrlutil -host hostname -port portnumber -username username
               - userpassword userpassword [-offset portoffset] -flag
```

For more information about security, refer to [“Specifying Security Properties” on page 81](#).

For **--help** and **--version**, the syntax is as follows:

```
stcmsctrlutil --help

stcmsctrlutil --version
```

### 8.3.2 Flags and Arguments

**Table 14** MS Control Utility Flags and Arguments

Shortcut	Flag arguments	Purpose
	<b>--version</b>	View utility version information.
	<b>--help</b>	View help information.
	<b>-username</b> <i>username</i>	Supply the user name to connect to the JMS IQ Manager. This flag is mandatory when JMS IQ security is enabled.
	<b>-userpassword</b> <i>userpassword</i>	Supply the password to connect to the JMS IQ Manager. This flag is mandatory when JMS IQ security is enabled.
	<b>-msversion</b>	View server version information.
	<b>-shutdown</b>	Shut down the server.
	<b>-status</b>	View server status.

**Table 14 MS Control Utility Flags and Arguments (Continued)**

Shortcut	Flag arguments	Purpose
-tl	-topiclist	List all topics for this server.
-sla	-sublistall	List all subscribers for all topics combined.
-slft	-sublistfortopic <i>topicname</i>	List all subscribers for the specified topic.
-ts	-topicstat <i>topicname</i>	View statistics for the specified topic.
-ql	-queueulist	List all queues for this server.
-rlfq	-recvlistforqueue <i>queuename</i>	List all receivers for the specified queue.
-rla	-recvlistall	List all receivers for all queues combined.
-qs	-queuestat <i>queuename</i>	View statistics for a specific queue.
	-host <i>hostname</i>	Specify the name of the Logical Host. If not specified, the default is: <b>-host localhost</b> For hosts other than localhost and flags other than --help and --version, <b>-host is required.</b>
	-port <i>portnumber</i>	Specify the TCP/IP port of the Logical Host that this Message Server is listening to. If not specified, the default is: <b>-port 7555</b> For ports other than 7555 and flags other than --help and --version, <b>-port is required.</b>
	-offset <i>portoffset</i>	Specify a server port offset number.
-ct	-createtopic <i>topicname</i>	Create a new topic with the specified name.
-dt	-deletetopic <i>topicname</i>	Delete the specified topic.
-cq	-createqueue <i>queuename</i>	Create a new queue with the specified name.
-dq	-deletequeue <i>queuename</i>	Delete the specified queue.
-cs	-createsub <i>topicname subname clientname</i>	Create a new subscriber for the specified topic and client. For <i>clientname</i> , specify <b>eGate</b> .
-ds	-deletesub <i>topicname subname clientname</i>	Delete a certain subscriber from the specified topic and client. For <i>clientname</i> , specify <b>eGate</b> .
-qml	-qmsglist <i>queuename seqnumber numbermessages</i>	List all messages for the specified queue, starting at or above the specified sequence number, and listing no more than <i>numbermessages</i> altogether.
-tml	-tmsglist <i>topicname seqnumber numbermessages</i>	List all messages for the specified topic, starting at or above the specified sequence number, and listing no more than <i>numbermessages</i> altogether.
-gqm	-qmessage <i>queuename</i>	Retrieve the particular message designated by <i>seqnumber</i> for the specified queue. If the specified queue contains no message with this sequence number, an error is returned.
-gtm	-tmessage <i>topicname seqnumber</i>	Retrieve the particular message designated by <i>seqnumber</i> for the specified topic. If the specified topic contains no message with this sequence number, an error is returned.

**Table 14 MS Control Utility Flags and Arguments (Continued)**

Shortcut	Flag arguments	Purpose
<b>-dtm</b>	<b>-deltmsg</b> <i>topicname seqnumber</i>	Delete the message at the specified sequence number in the specified topic.
<b>-dqm</b>	<b>-delqmsg</b> <i>queuename seqnumber</i>	Delete the message at the specified sequence number in the specified queue.
<b>-ctm</b>	<b>-changetmsg</b> <i>topicname seqnumber</i>	Change the content of the message at the specified sequence number in the specified topic, reading from standard input (the command prompt, or whatever file or piped command it specifies).
<b>-cqm</b>	<b>-changeqmsg</b> <i>queuename seqnumber</i>	Change the content of the message at the specified sequence number in the specified queue, reading from standard input (the command prompt, or whatever file or piped command it specifies).
	<b>-msgtype</b> <i>type</i>	Specify the data type of the content of the message. Must be <b>bytes</b> or <b>text</b> .
<b>-dqm</b>	<b>-delqmsg</b> <i>queuename</i>	Delete the message at the specified sequence number in the specified queue.
<b>-lt</b>	<b>-locktopic</b> <i>topicname</i>	Lock a topic from being accessed, prevent any subscriber from receiving messages from it.
<b>-ut</b>	<b>-unlocktopic</b> <i>topicname</i>	Unlock a topic, restoring access to all subscribers.
<b>-tmi</b>	<b>-tmimport</b> <i>topicname seqno nmgs</i>	Republish messages from a topic.
<b>-qmi</b>	<b>-qmimport</b> <i>topicname seqno nmgs</i>	Republishing messages from a queue.
<b>-j</b>	<b>-journaler</b>	Browse journaled messages.
<b>-ar</b>	<b>-archiver</b> <i>directoryname</i>	Browse specified archive.
<b>-b</b>	<b>-backup</b> <i>file date</i>	Back up messages.
<b>-t</b>	<b>-timeout</b> <i>seconds</i>	Specify the timeout in seconds.

### 8.3.3 Displaying the Utility Version

```
stcmsctrlutil --version
Control Utility version 5.0.6778 (Sept 22 2003)
Copyright (c) 2003, SeeBeyond Technology Corporation.
All Rights Reserved.
```

### 8.3.4 Shutting Down the Server

```
stcmsctrlutil -host localhost -port 24055 -shutdown
```

### 8.3.5 Viewing Message Server Statistics

```
stcmsctrlutil -host localhost -port 24055 -status
Up since: Fri June 13 13:13:10 2003
```

```
Memory used by data messages: 67.535 K(Bytes)
Total messages passed through: 664 Message(s)
Number of message queues: 5
Number of connections: 7
Port number: 24055
```

### 8.3.6 Viewing All Topics for a Server

```
stcmsctrlutil -host localhost -port 24055 -topiclist
Topic List:
  SeeBeyond.MS.Control
  Broadcast
  STCTemporaryTopic.2.1
```

### 8.3.7 Changing Topic Message Contents

To change the contents (payload) of a message for a specified topic, you use the **-changetmsg** flag. You must specify whether the message type is bytes or text. The message cannot be processed while you are changing the contents.

#### To change topic message contents

- 1 Use the following command syntax to specify the contents change:

```
stcmsctrlutil.exe -p portnumber -ctm topicname seqnumber -msgtype
type
```

where *portnumber* is the number of the port, where *topicname* is the name of the topic that contains the message, where *seqnumber* is the sequence number of the message, and where *type* is either bytes or text indicating the message type of the original message.

- 2 Press **ENTER**.
- 3 Type the new contents.
- 4 Press **CTRL-Z**.

The command line below shows an example of changing of T0:

```
stcmsctrlutil -ctm T0 182-host -p 24053 -msgtype text
NEWCONTENTS
^Z
Executed function: CHANGE_TOPIC_MESSAGE
Message: 182 has been changed
```

The command line below shows the changed contents of the message:

```
stcmsctrlutil.exe -p 24053 -tmessage T0 182 -msgtype text
Executed function: GET_TOPIC_MESSAGE
Text message
NEWCONTENTS
```

### 8.3.8 Viewing Statistics for Topics

The following example shows how to view statistics for a topic.

The Suspended entry shows whether topic is suspended and all subscribers stop receiving messages. This usually shows No. After the JMS IQ Manager restarts, all topics have Suspended: No status.

```
stcmsctrlutil -host localhost -port 24055 -topicstat Broadcast
Topic Name: Broadcast
First sequence number: 0
Last sequence number: 5
First enqueue time: 05172001:16:30:30
Last enqueue time: 05172001:16:30:42
Number of current subscribers: 0
Number of total subscribers: 2
Message count: 6
Lowest subscriber sequence: 0
Highest subscriber sequence: 3
Suspended: No
```

### 8.3.9 Viewing Properties of All Subscribers

```
stcmsctrlutil -host localhost -port 24055 -sublistall
Number Of Subscriber(s): 4
Subscriber name: NonDurable1
  Client ID:
  Topic name: SeeBeyond.MS.Control
  Committed sequence: 0
  High sequence: 0
Subscriber name: subscriber1
  Client ID: Client
  Topic name: Broadcast
  Committed sequence: 0
  High sequence: 3
Subscriber name: subscriber2
  Client ID: Client
  Topic name: Broadcast
  Committed sequence: 3
  High sequence: 6
Subscriber name: NonDurable2
  Client ID:
  Topic name: STCTemporaryTopic.2.1
  Committed sequence: 0
  High sequence: 0
```

### 8.3.10 Viewing Properties of All Subscribers to Topics

```
stcmsctrlutil -host localhost -port 24055 -sublistfortopic STC
Number Of Subscriber(s): 2
Subscriber name: subscriber1
  Client ID: Client
  Topic name: STC
  Committed sequence: 0
  High sequence: 3
Subscriber name: subscriber2
  Client ID: Client
  Topic name: STC
  Committed sequence: 3
  High sequence: 6
```

### 8.3.11 Viewing All Queues for a Server

```
stcmsctrlutil -host localhost -port 24055 -queuelist
Queue List:
  MyQueue0
  PTP
```



### 8.3.12 Displaying Statistics for Queues

```
stcmsctrlutil -host localhost -port 24055 -queuostat PTP
Queue Name: PTP
First enqueue time: 02011970:00:00:00
Last enqueue time: 02011970:00:00:00
Number of current receivers: 2
Message count: 0
Messages sent and committed: 0
```

### 8.3.13 Viewing Properties of All Receivers

```
stcmsctrlutil -host localhost -port 24055 -recvlistall
Number Of Receiver(s): 3
Receiver ID: 14235659
    Queue name: MyQueue0
    Session ID: 1
    Committed messages: 0
    Uncommitted messages: 0
Receiver ID: 14274653
    Queue name: PTP
    Session ID: 3
    Committed messages: 434
    Uncommitted messages: 0
Receiver ID: 14291939
    Queue name: PTP
    Session ID: 4
    Committed messages: 432
    Uncommitted messages: 1
```

### 8.3.14 Viewing Properties of All Receivers of Queues

```
stcmsctrlutil -host localhost -port 24055 -recvlistforqueue PTP
Number Of Receiver(s): 2
Receiver ID: 14274653
    Queue name: PTP
    Session ID: 3
    Committed messages: 434
    Uncommitted messages: 0
Receiver ID: 14291939
    Queue name: PTP
    Session ID: 4
    Committed messages: 432
    Uncommitted messages: 1
```

### 8.3.15 Republishing Messages from Topics

To republish messages from topics, you use the **-tmimport** flag.

The command line below republishes five journaled messages from topic T0 starting from message with sequence number 491.

```
stcmsctrlutil -j -tmi T0 491 5
Executed function: IMPORT
Importing messages
Executed function: REPUBLISH_STATUS
Last imported sequence number = 491
Executed function: REPUBLISH_STATUS
Last imported sequence number = 497
```

### 8.3.16 Republishing Messages from Queues

To republish messages from queues, you use the **-qmimport** flag.

The command line below republishes five journaled messages from queue T0 starting from message with sequence number 500.

```
stcmsctrlutil -j -qmi T0 500 5
Executed function: IMPORT
Importing messages
Executed function: REPUBLISH_STATUS
Last imported sequence number = 500
Executed function: REPUBLISH_STATUS
Import failed
Import failed on sequence number: 500
```

There are no messages. Import failed trying to republish first sequence number. To republish messages from archive you must specify the **-ar** flag and archive directory instead of the **-journaler** flag.

### 8.3.17 Browsing Journaled Messages

To browse journaled messages, you use the **-journaler** flag. The **-journaler** flag receives information from the journaler instead of the JMS server. The journaler does not support information about subscribers and receivers; flags such as **-sublistall**, **-deletesub**, or **-recvlistall** do not work with the **-journaler** flag. You cannot delete journaled messages, topics, or queues. The MS Control utility displays information about journaled topics and queues in the same format as the JMS server.

For example, the command line below displays the topic message list from the JMS server:

```
stcmsctrlutil -tl
Number Of Topic(s): 4
Topic List:
    STCMS.Control
    STCMS.Journal
    T0
    STCTemporaryTopic.1031789365648.1031789335025.1
```

The command line below displays the topic message list from the journaler:

```
stcmsctrlutil -j -tl
Number Of Topic(s): 1
Topic List:
    T0
```

The command line below displays information about queue Q0 from the JMS server:

```
stcmsctrlutil -qs Q0
Queue Name: Q0
First enqueue time: 01011970:00:00:00
Last enqueue time: 01011970:00:00:00
Number of current receivers: 0
Message count: 0
Messages sent and committed: 1001
Min sequence Number: 0
Max sequence Number: 0
```

The command line below displays the same information about queue Q0 but from the journaler:

```
stcmsctrlutil -j -qs Q0
Queue Name: Q0
First enqueue time: 09122003:00:14:07
Last enqueue time: 09122003:00:14:28
Number of current receivers: 0
Message count: 1001
Messages sent and committed: 0
Min sequence Number: 0
Max sequence Number: 1000
```

The command line below displays the information about one message with sequence number 0. Because the message has been consumed, the MS Control utility cannot display this information.

```
stcmsctrlutil -qml Q0 0 1
Number Of Messages(s): 0
```

The command line below displays the same information as above, but from the journaler. The message is not journal expired, which enables the MS Control utility to display the message properties.

```
stcmsctrlutil -j -qml Q0 0 1
Number Of Messages(s): 1
Message[1]:
Message.SeqNo=0
Message.Timestamp=1031789647260
Journaler.ExpirationTime=1031809647260
Message.Size=228
Message.JMSProperty.TS=1031789647260
Message.JMSProperty.EX=0
Message.JMSProperty.DM=1
Message.JMSProperty.TY=ASCII
Message.JMSProperty.PR=0
Message.JMSProperty.RD=false
Message.JMSProperty.MI=ID:377:3b742aa5:950:0a01beee:3d7fdc4f104
Message.UserProperty.JMS_ProducerID=BENCH
```

### 8.3.18 Backing Up

The **-backup** flag creates a zip file that contains all messages (regardless of whether they are live or journaled) for all queues and topics up to the specified date. After you create this zip file, you can unzip it and then browse the archive using the **-archive** flag.

If you are using a non-default port, you must specify the **-p** flag with the port number of the JMS server.

The command line below shows an example of **-backup**:

```
stcmsctrlutil -backup c:\eGate\client\Archiver\Ar09112003.zip "09/11/
2003"
Backup finished. Archived messages: 2003
```

### 8.3.19 Browsing Archives

When you have backed up the topics and queues for a particular date, you can browse the archive (a .zip file) with the **-archive** flag.

The **-archive** flag functions similarly to **-journaler** except that you specify the directory where you unzipped the archive. The MS Control utility displays information in the same format as **-journaler**.

The command line below displays the same information as **-journaler**, but it reads this information from `c:\eGate\client\Archiver\backup`.

```
stcmsctrlutil -ar c:\eGate\client\Archiver\backup -tl
Number Of Topic(s): 1
Topic List:
    T0
```

The command line below displays information about topic T0.

```
stcmsctrlutil -ar c:\eGate\client\Archiver\backup -ts T0
Topic Name: T0
First sequence number: 0
Last sequence number: 1000
First enqueue time: 09122003:00:14:17
Last enqueue time: 09122003:00:14:00
Number of current subscribers: 0
Number of total subscribers: 0
Message count: 1001
Lowest subscriber sequence: 0
Highest subscriber sequence: 0
```

The command line below displays a description of the message with sequence number 1 from the archive `c:\eGate\client\Archiver\backup`.

```
stcmsctrlutil -ar c:\eGate\client\Archiver\backup -tml T0 1 1
Number Of Messages(s): 1
Message[1]:
Message.SeqNo=1
Message.Timestamp=1031789654330
Journaler.ExpirationTime=1031809654330
Message.Size=228
Message.JMSProperty.EX=0
Message.JMSProperty.TS=1031789654330
Message.JMSProperty.DM=1
Message.JMSProperty.TY=ASCII
Message.JMSProperty.PR=0
Message.JMSProperty.MI=ID:45c:3b742aa6:950:0a01beee:3d7fdc5614a
Message.JMSProperty.RD=false
Message.UserProperty.JMS_ProducerID=BENCH
```

### 8.3.20 Setting Timeout

You can use the **-timeout** flag to increase the timeout for the MS Control utility for retrieving messages. The default timeout is five seconds. If the message is not received within five seconds, the utility exits and you see the message "Timeout to receive message from the server, exiting stcmsctrlutil API." This may happen when the server is busy. Increasing the timeout as shown below may resolve this problem.

The command line below shows how you increase the timeout to 15 seconds:

```
stcmsctrlutil -j -tl -t 15
```

---

## 8.4 stcmsutil.Properties

The **stcmsutil.Properties** file is located in the message server\bin folder in the message server in the location where the eGate Logical Host is installed. This file holds current

settings and preferences for the MS Control utility. Table 15 shows the property names and their preset values if never modified.

**Table 15** stcmsutil.Properties - Message Service Utility Property Names and Values

Property	Preset value	Description
Util.Client.Library	stc_msclient.dll	Specifies the filename of the dynamic link library (DLL) file used by the C++ client.
stcmsctrlutil.Trace.ToStdout	No	Specifies whether <b>stcmsctrlutil.exe</b> writes messages to standard output (usually the console).
stcmsctrlutil.Trace.Verbose	No	Specifies whether trace messages written by <b>stcmsctrlutil.exe</b> provide a lengthy full explanation.
stcmsctrlutil.Trace.ToFile	Yes	Specifies whether trace messages written by <b>stcmsctrlutil.exe</b> are written to a file. Also see <b>stcmsctrlutil.Trace.LogFile</b> .
stcmsctrlutil.Trace.Level	2	Specifies threshold severity level at which <b>stcmsctrlutil.exe</b> issues informational, warning, and error messages: <ul style="list-style-type: none"> <li>▪ <b>0</b> Informational messages (in addition to all three categories below).</li> <li>▪ <b>1</b> Warning messages (in addition to both categories below).</li> <li>▪ <b>2</b> Error messages (in addition to the category below).</li> <li>▪ <b>3</b> Fatal messages.</li> </ul>
stcmsctrlutil.Trace.Performance	No	Specifies whether performance level tracing is turned on.
stcmsctrlutil.Trace.Memory	No	Specifies whether memory level tracing is turned on.
stcmsctrlutil.Trace.TimeStamp	No	Specifies whether <b>stcmsctrlutil.exe</b> prints timestamps in the trace log file.
stcmsctrlutil.Trace.LogFile	../logs/stcmsctrlutil.log	Specifies the path and file name where trace messages written by <b>stcmsctrlutil.exe</b> are logged. Meaningful only when <b>stcmsctrlutil.Trace.ToFile = Yes</b> .

# Index

## C

- configuration parameters
  - DB Settings
    - DBCacheSize 91
    - DBLockCacheIntoRAM 92
    - DBMaxSegments 92
    - DBMinSegments 94
    - DBSegmentSize 95
    - DBSuffix 106
  - DB Sync
    - DBSync 91
  - Message Settings
    - MaxPayloadMemory 93
    - MaxTimeToLive 89, 108
    - PayloadMemoryPad 93
  - Server Settings
    - Journaling 107
    - JournalPath 107
    - JournalTimeToLive 108
    - ServerMaxMessages 96
  - TimeDependency
    - TimeDependency 98
  - Topic Settings
    - TopicMaxMessages 96
    - TopicMaxMessagesPad 96
  - Trace Settings
    - TraceFile 101
    - TraceLevel 103
    - TraceMemory 101
    - TraceTimestamp 98, 102
    - TraceVerbose 102
- conventions
  - path name separator 12
  - Windows 12

## D

- DBCacheSize 91
- DBLockCacheIntoRam 92
- DBMaxSegments 92
- DBMinSegments 94
- DBSuffix 106
- DBSync 91
- document

- conventions 12
- document purpose and scope 11

## J

- JMS client
  - reconfiguring 78
- Journaling 107
- JournalPath 107
- JournalTimeToLive 108

## L

- Logical Host
  - system JMS 87

## M

- MaxPayloadMemory 93
- MaxTimeToLive 89, 108

## O

- organization of information, document 11

## P

- PayloadMemoryPad 93

## R

- reconfiguring
  - JMS client 78

## S

- SeeBeyond Web site 13
- ServerMaxMessages 96
- stcmsctrlutil
  - syntax 124
  - usage examples 126
- stcmsutil.Properties file
  - described 132
  - preset values 133
- supporting documents 13
- system JMS 87

## T

- technical support
  - SeeBeyond Web site 13
- TimeDependency 98
- TopicMaxMessages 96

## Index

TopicMaxMessagesPad 96

TraceLevel 103

TraceMemory 101

TraceTimeStamp 98, 102

TraceToFile 101

TraceVerbose 102

## W

writing conventions 12