

SUN SEEBEYOND

eGATE™ INTEGRATOR JMS REFERENCE GUIDE

Release 5.1.0



Copyright © 2006 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, U.S.A. All rights reserved. Sun Microsystems, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed at <http://www.sun.com/patents> and one or more additional patents or pending patent applications in the U.S. and in other countries. U.S. Government Rights - Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements. Use is subject to license terms. This distribution may include materials developed by third parties. Sun, Sun Microsystems, the Sun logo, Java, Sun Java Composite Application Platform Suite, SeeBeyond, eGate, eInsight, eVision, eTL, eXchange, eView, eIndex, eBAM, eWay, and JMS are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon architecture developed by Sun Microsystems, Inc. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd. This product is covered and controlled by U.S. Export Control laws and may be subject to the export or import laws in other countries. Nuclear, missile, chemical biological weapons or nuclear maritime end uses or end users, whether direct or indirect, are strictly prohibited. Export or reexport to countries subject to U.S. embargo or to entities identified on U.S. export exclusion lists, including, but not limited to, the denied persons and specially designated nationals lists is strictly prohibited.

Copyright © 2006 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, Etats-Unis. Tous droits réservés. Sun Microsystems, Inc. détient les droits de propriété intellectuelle relatifs à la technologie incorporée dans le produit qui est décrit dans ce document. En particulier, et ce sans limitation, ces droits de propriété intellectuelle peuvent inclure un ou plus des brevets américains listés à l'adresse <http://www.sun.com/patents> et un ou les brevets supplémentaires ou les applications de brevet en attente aux Etats - Unis et dans les autres pays. L'utilisation est soumise aux termes de la Licence. Cette distribution peut comprendre des composants développés par des tierces parties. Sun, Sun Microsystems, le logo Sun, Java, Sun Java Composite Application Platform Suite, Sun, SeeBeyond, eGate, eInsight, eVision, eTL, eXchange, eView, eIndex, eBAM et eWay sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd. Ce produit est soumis à la législation américaine en matière de contrôle des exportations et peut être soumis à la réglementation en vigueur dans d'autres pays dans le domaine des exportations et importations. Les utilisations, ou utilisateurs finaux, pour des armes nucléaires, des missiles, des armes biologiques et chimiques ou du nucléaire maritime, directement ou indirectement, sont strictement interdites. Les exportations ou réexportations vers les pays sous embargo américain, ou vers des entités figurant sur les listes d'exclusion d'exportation américaines, y compris, mais de manière non exhaustive, la liste de personnes qui font objet d'un ordre de ne pas participer, d'une façon directe ou indirecte, aux exportations des produits ou des services qui sont régis par la législation américaine en matière de contrôle des exportations et la liste de ressortissants spécifiquement désignés, sont rigoureusement interdites.

Version 20060316190523

Contents

List of Tables	7
----------------	---

List of Figures	8
-----------------	---

Chapter 1

Introduction	9
About This Document	9
What's in This Document	9
Scope	10
Intended Audience	10
Text Conventions	10
Screenshots	11
Related Documents	11
Sun Microsystems, Inc. Web Site	11
Documentation Feedback	11

Chapter 2

About Message Servers in eGate Integrator	12
Supported Message Servers	12
About the SeeBeyond JMS IQ Manager	12
Message Delivery Order	13
Message Priorities	14
Security	14
Distributed Transactions	14
Connection Consumer	14
About the Sun Java System Message Queue	15
Message Servers in Java CAPS Projects	15
The eGate Integrator Model	15
Project Design Phase: Topics, Queues and JMS Properties	17
Environment Design Phase: Message Server Configurations	18
Deployment Profile Design Phase: Creating Mappings	19
Activating Deployment Profile Phase: Configuration File	19
Message Servers at Runtime	19

Quick Overview: Building Java CAPS Projects with JMS	20
JMS IQ Manager Administration Tools	20
Notes About the Java Message Queue	21

Chapter 3

Functional Overview of the JMS IQ Manager	23
About the JMS IQ Manager Database	23
About Segments	23
About the Database Location	24
The Default Database Configuration	24
Message Processing Order	25
JMS IQ Manager FIFO Delivery Modes	26
Fully Concurrent Processing	26
Protected Concurrent Processing	27
Fully Serialized Processing	28
JMS Client Concurrency Effect on FIFO Delivery Modes	29
Serial Processing Across a Destination Group	30
About Performance	30
Managing Resources	31
Throttling Producers	31

Chapter 4

Installing Message Server .sar Files	34
Installing Message Server .sar Files	34

Chapter 5

Building JMS Business Logic for Java CAPS Projects	35
Adding Message Destinations to Projects	35
Creating Collaboration Definitions with JMS OTDs	36
Using JMS Methods in Collaboration Definitions	37
JMS Methods Available for JMS OTDs	37
JMS Methods Available for JMS Messages	52
Using JMS Message Properties in Collaborations	57
Binding Collaboration Definitions to Services	60
Linking JMS Collaborations to Message Destinations	61

Chapter 6

Configuring JMS Clients 64

Configuring JMS Clients	64
Setting JMS Client Message Processing Order	66
Specifying Persistent or Non-Persistent Delivery Mode	67
Specifying Message Selectors	67
Specifying Message Priorities	67
Specifying the Transaction Mode	68
Specifying the Idle Timeout	69
Specifying the Maximum Wait Time	69
Specifying the Maximum Pool Size	69
Specifying the Steady Pool Size	69
Setting Durability	70
Setting the Number of ServerSessionPool Threads	70

Chapter 7

Configuring Message Servers 72

Adding Message Servers to Logical Hosts	72
Configuring Java Message Queue Connections	73
Configuring the JMS IQ Manager Connections	74
Configuring Runtime JMS IQ Managers	75
Setting Segment Options	77
Specifying the Location of JMS IQ Manager Database Files	77
Specifying Segment Size	77
Specifying the Prefix for Database Filenames	78
Specifying the Minimum Number of Segments	79
Specifying the Maximum Number of Segments	79
Enabling Disk Synchronization	80
Setting Journaling and Message Expiration Options	80
Enabling Message Expiration	81
Specifying the Maximum Lifetime for Live Messages	81
Enabling Journaling	81
Specifying Time To Live for Journaled Messages	82
Specifying the Location of the Journal Database	82
Configuring Throttling Options	83
Specifying the Throttling Threshold for Message Destinations	83
Specifying the Server Throttling Threshold	83
Specifying the Throttling Lag	84
Configuring FIFO Modes	84
Specifying Fully Serialized Queues	84
Specifying Protected Concurrent Queues	85
Specifying Special FIFO Mode Expiration Time	85
Specifying Time Dependency	86
Setting Up Security	86
Enabling Authentication and Authorization	86
Specifying the Default Realm	87

Enabling Sun Java System Directory Server	88
Enabling Microsoft Active Directory Service	91
Enabling a Generic LDAP Server	94
Setting Diagnostic Options	97
Specifying the Logging Level	97
Specifying the Journal Logging Level	97
Specifying the Maximum Log File Size	98
Specifying the Maximum Number of Backup Log Files	98
Enabling Alerts	99
Configuring SRE JMS IQ Managers	99
Deploying Topics and Queues to Message Servers	99

Chapter 8

Managing JMS IQ Managers	102
About Administration Tools	102
Managing JMS IQ Managers with the Enterprise Manager	102
Monitoring Topics and Queues	103
Viewing Message Properties	104
Viewing Message Destination Summaries	105
Viewing Message Payload	105
Managing JMS IQ Managers with the MS Control Utility	105
Syntax	106
Flags and Arguments	106
Shutting Down the Server	108
Viewing JMS IQ Manager Statistics	109
Viewing All Topics for a JMS IQ Manager	109
Changing Topic Message Contents	109
Viewing Statistics for Topics	110
Viewing Properties of All Subscribers	110
Viewing Properties of All Subscribers to Topics	110
Viewing All Queues for a JMS IQ Manager	111
Displaying Statistics for Queues	111
Viewing Properties of All Receivers	111
Viewing Properties of All Receivers of Queues	111
Republishing Messages from Topics	111
Republishing Messages from Queues	112
Browsing Journaled Messages	112
Backing Up	113
Browsing Archives	113
Setting Timeout	114

Index	115
--------------	------------

List of Tables

Table 1	Text Conventions	10
Table 2	JMS IQ Manager Administration Tools	21
Table 3	Benefits and Costs—Fully Concurrent Processing	27
Table 4	Benefits and Costs—Protected Concurrent Processing	28
Table 5	Benefits and Costs—Fully Serialized Processing	28
Table 6	JMS Client Concurrency Effect on FIFO modes—Topic	29
Table 7	JMS Client Concurrency Effect on FIFO modes—Queue	30
Table 8	Publisher Throttling	32
Table 9	JMS Message Properties	58
Table 10	Configuring Java Message Queue Connections	74
Table 11	Configuring JMS IQ Manager Connections	75
Table 12	LDAP Server Roles	87
Table 13	Sun Java System Directory Server Properties	88
Table 14	Active Directory Properties	91
Table 15	OpenLDAP Directory Server Properties	95
Table 16	Topic Properties	103
Table 17	Queue Properties	104
Table 18	Message Properties	104
Table 19	MS Control Utility Flags and Arguments	106

List of Figures

Figure 1	eGate Integrator Model	16
Figure 2	Message Server in the eGate Integrator Model	16
Figure 3	Java CAPS Projects	18
Figure 4	Adding Message Servers to an Environment	18
Figure 5	Deployment Profile Example	19
Figure 6	JMS IQ Manager Database Structure	25
Figure 7	Fully Concurrent Processing	27
Figure 8	Protected Concurrent Processing	27
Figure 9	Fully Serialized Processing	28
Figure 10	Multiple Application Server Configuration	29
Figure 11	Creating Topics and Queues	35
Figure 12	JMS Message Priorities	59
Figure 13	Setting JMS Message Properties	60
Figure 14	Binding Collaboration Definitions to Services	60
Figure 15	Changing a Service to a Collaboration	61
Figure 16	Collaboration Binding Window	62
Figure 17	Binding Services to eWays	62
Figure 18	Collaboration Connections to Destinations	63
Figure 19	Finished Connectivity Map	63
Figure 20	JMS Client Properties Icon	64
Figure 21	Configuring JMS Clients	65
Figure 22	Expanding the Environments Folder	72
Figure 23	Configuring Java Message Queue Connection Information	74
Figure 24	Configuring JMS IQ Manager Connection Information	75
Figure 25	Configuring Runtime JMS IQ Managers	76
Figure 26	Initial Deployment Profile	100
Figure 27	Deployment Profile Example	101

Introduction

This chapter provides information about this document, its related documents and provides a list of documents referenced. It also provides the URL for the Sun Web site and feedback and support email addresses.

What's in This Chapter

- [About This Document](#) on page 9
- [Related Documents](#) on page 11
- [Sun Microsystems, Inc. Web Site](#) on page 11
- [Documentation Feedback](#) on page 11

1.1 About This Document

The section below provide information about this document, such as an overview of its contents, scope, and intended audience.

1.1.1 What's in This Document

This document describes the different message server available for the Sun Java™ Application Platform Suite (CAPS). This guide describes how you configure and manage message servers.

This guide contains the following chapters:

- [Chapter 2 “About Message Servers in eGate Integrator” on page 12](#) provides an overview of the eGate™ Integrator message servers, as well as message server administration tools available in eGate Integrator.
- [Chapter 3 “Functional Overview of the JMS IQ Manager” on page 23](#) 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 Message Server .sar Files” on page 34](#) describes how to install the .sar file for SRE JMS IQ Manager support.

- **Chapter 5 “Building JMS Business Logic for Java CAPS Projects” on page 35** describes how to use Enterprise Designer to add message destinations (topics and queues) and build a JMS OTD.
- **Chapter 6 “Configuring JMS Clients” on page 64** describes how to configure JMS properties for message servers in Java CAPS.
- **Chapter 7 “Configuring Message Servers” on page 72** describes how to use Enterprise Designer to configure message servers.
- **Chapter 8 “Managing JMS IQ Managers” on page 102** describes how you can use Enterprise Manager and the MS Control utility to manage JMS IQ Managers.

1.1.2 Scope

This document describes how to use, configure, and manage message servers in eGate Integrator.

This guide assumes that you are familiar with all information in the *Java Message Server Specification* version 1.1.

This guide refers to the *Sun SeeBeyond eGate Integrator User’s Guide* for Enterprise Designer-specific procedures, and to the *Sun Seebeyond eGate Integrator System Administration Guide* for Enterprise Manager-specific procedures.

1.1.3 Intended Audience

This document is intended for Java CAPS Project designers and administrators with detailed knowledge of the JMS API.

1.1.4 Text Conventions

The following conventions are observed throughout this document.

Table 1 Text Conventions

Text Convention	Used For	Examples
Bold	Names of buttons, files, icons, parameters, variables, methods, menus, and objects	<ul style="list-style-type: none">▪ Click OK.▪ On the File menu, click Exit.▪ Select the eGate.sar file.
Monospaced	Command line arguments, code samples; variables are shown in <i>bold italic</i>	<code>java -jar <i>filename</i>.jar</code>
Blue bold	Hypertext links within document	See Text Conventions on page 10
<u>Blue underlined</u>	Hypertext links for Web addresses (URLs) or email addresses	http://www.sun.com

1.1.5 Screenshots

Depending on what products you have installed, and how they are configured, the screenshots in this document may differ from what you see on your system.

1.2 Related Documents

The following SeeBeyond documents provide additional information about Java CAPS message servers:

- *Sun SeeBeyond eGate Integrator User's Guide*
- *Sun SeeBeyond eGate Integrator System Administration Guide*
- *Sun Java Composite Application Platform Suite Deployment Guide*
- *Sun Java Composite Application Platform Suite Installation Guide*
- *Java Message Server Specification version 1.1*
- *Sun Java System Message Queue 3 Technical Overview*
- *Sun Java System Message Queue 3 Administration Guide*

1.3 Sun Microsystems, Inc. Web Site

The Sun Microsystems web site is your best source for up-to-the-minute product news and technical support information. The site's URL is:

<http://www.sun.com>

1.4 Documentation Feedback

We appreciate your feedback. Please send any comments or suggestions regarding this document to:

CAPS_docsfeedback@sun.com

About Message Servers in eGate Integrator

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

What's in This Chapter

- [Supported Message Servers](#) on page 12
- [About the SeeBeyond JMS IQ Manager](#) on page 12
- [About the Sun Java System Message Queue](#) on page 15
- [Message Servers in Java CAPS Projects](#) on page 15
- [Quick Overview: Building Java CAPS Projects with JMS](#) on page 20
- [JMS IQ Manager Administration Tools](#) on page 20

2.1 Supported Message Servers

eGate Integrator provides two built-in message servers; the Java™ Message Queue and the Sun SeeBeyond JMS IQ Manager, also referred to as the STCMS. eGate Integrator also supports several non-Java Java CAPS message servers such as the WebSpere MQ.

Java CAPS supports several types of message servers. eGate Integrator comes with its own JMS implementations: the SeeBeyond JMS IQ Manager and the Schema Run-time Environment (SRE) JMS IQ Manager. The Sun SeeBeyond JMS IQ Manager is simply referred to as the JMS IQ Manager.

2.2 About the SeeBeyond 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 1.1 and includes the following features:

- [Message Delivery Order](#) on page 13
- [Message Priorities](#) on page 14

- [Security](#) on page 14
- [Distributed Transactions](#) on page 14
- [Connection Consumer](#) on page 14

2.2.1 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 for queues.
- Specifying a set of message destinations (a time order group) for which fully serialized processing occurs.
- Configuring topics and queues 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 25](#).

Special FIFO Modes for All Destinations

For each 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 [“Configuring FIFO Modes” on page 84](#).

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 older messages in the time order group have been received or are in the process of being received. For information, refer to [Setting Diagnostic 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 topic subscribers as connection consumers to improve message throughput through concurrent processing. You can set the JMS client configuration with the **Concurrency** property as described in [“Configuring JMS Clients” on page 64](#).

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 default 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 process messages concurrently using multiple integration servers; these integration servers may run on different systems.

This configuration does affect FIFO processing. For information, refer to [“JMS Client Concurrency Effect on FIFO Delivery Modes” on page 29](#).

2.2.2 Message Priorities

eGate Integrator enables you to set message priorities for topic publishers and queue senders as well as Collaborations. The default message priority is 4. The 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. The eGate message priority implementation adheres to the recommended standards in the Java Specification: in most circumstances, messages with higher priorities are delivered before message with lower priorities.

For information about setting the priority level for topic publishers and queue senders, refer to [“Configuring JMS Clients” on page 64](#).

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 [Using JMS Message Properties in Collaborations](#) on page 57.

2.2.3 Security

The eGate Integrator provides role-based security for the JMS IQ Manager by using authentication via file realm, Active Directory service, and Sun Java System Directory Server version 5.2, formerly known as Sun ONE Directory Server. When authentication is enabled, access to the JMS IQ Manager is only granted when the connection has a valid user ID and password.

JMS IQ Manager security is disabled by default. To enable security, refer to [“Enabling Authentication and Authorization” on page 86](#).

2.2.4 Distributed Transactions

The JMS IQ Manager enables you to configure JMS properties for distributed transactions (XA).

2.2.5 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 information about how this configuration affects message processing order, refer to [“JMS Client Concurrency Effect on FIFO Delivery Modes” on page 29](#).

For information about connecting Java CAPS Projects to SRE schemas, refer to the *Sun SeeBeyond eGate Integrator Upgrade Guide*.

2.3 About the Sun Java System Message Queue

eGate Integrator enables you to set up Java CAPS Projects so that they can be deployed to the Sun Java Enterprise System (JES). To create a JES deployment, do the following:

- 1 Install the support files as described in [Installing Message Server .sar Files](#) on page 34.
- 2 In your Java CAPS Project, select a JES application server and message server. For information, refer to [Adding Message Servers to Logical Hosts](#) on page 72.
- 3 Deploy the Project as described in the *Sun SeeBeyond eGate Integrator User's Guide*.

2.4 Message Servers in Java CAPS Projects

eGate Integrator provides an efficient way of designing and deploying your Java CAPS Projects 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

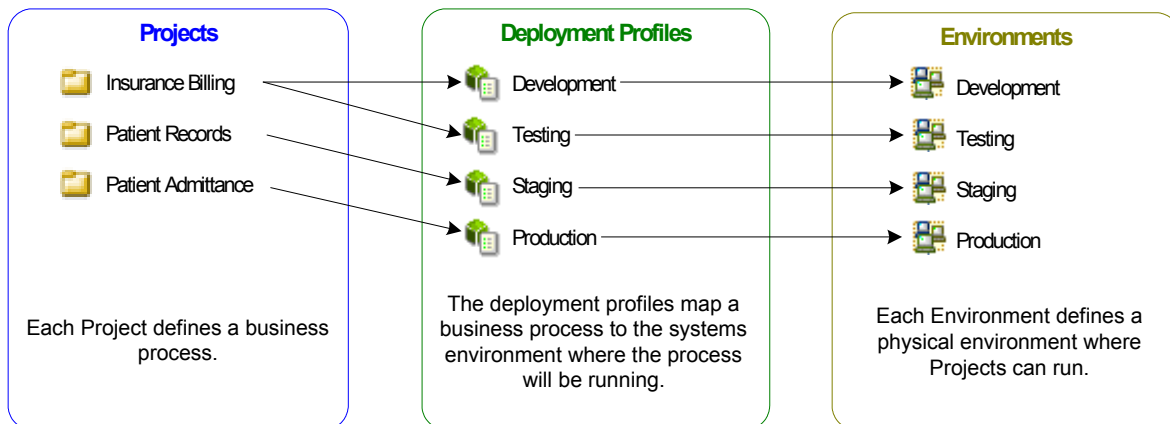
eGate Integrator models the design and deployment phases after business workflows. In broad outline, an eGate Integrator implementation includes the following steps:

- 1 **Designing a Java CAPS 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 **Creating a Java CAPS 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 **Creating 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 Java CAPS Project.**
To deploy the Java CAPS Project, you deploy the Java CAPS Project .EAR file. This creates configuration files 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.

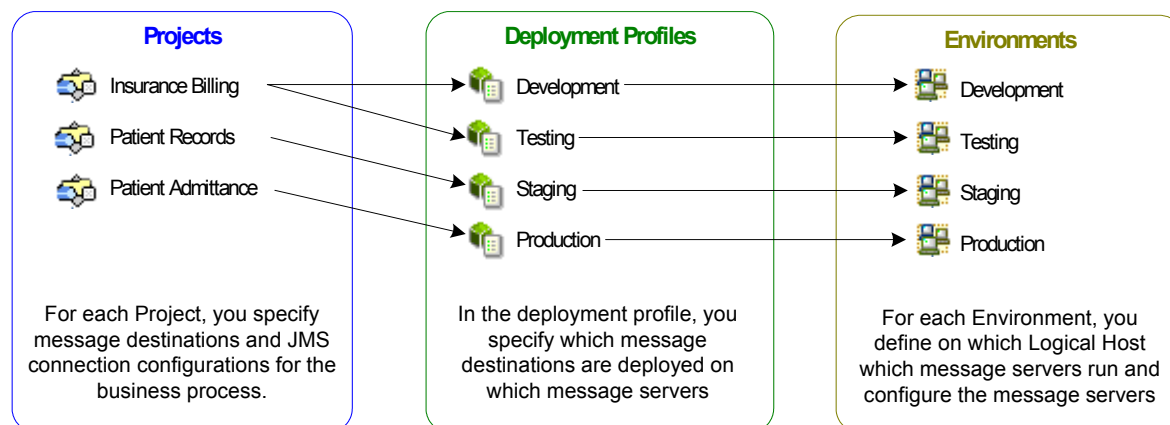
Figure 1 eGate Integrator Model



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 a Java CAPS Project in Enterprise Designer. In a Java CAPS 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 icons 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.

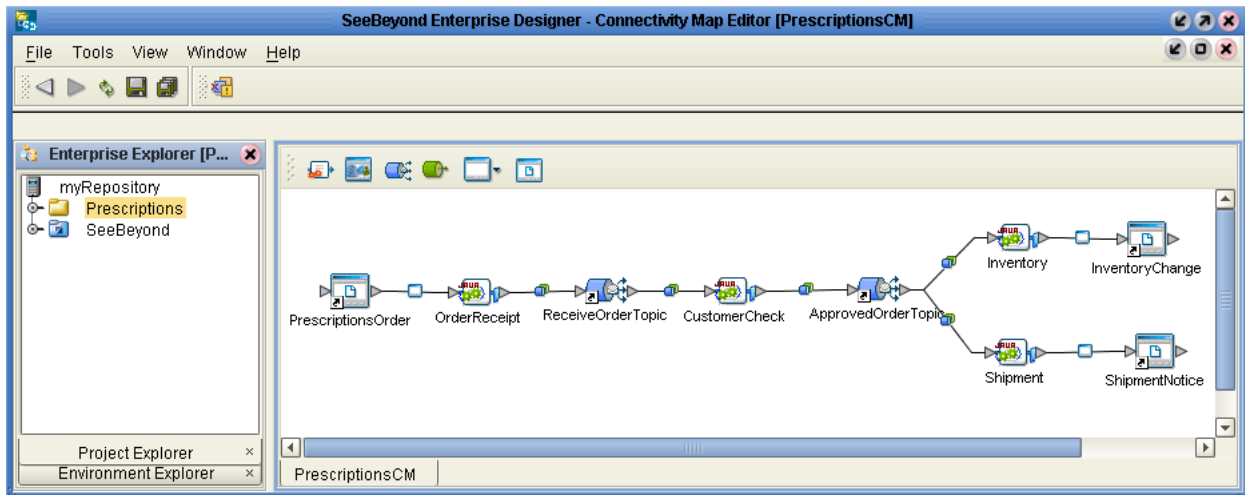
After defining the Collaboration Definitions, you create the relationships between the components by associating Collaboration Definitions with Collaborations. This creates links between the components. On links between message destinations and their subscribers and publishers, there is a JMS client 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, “Building JMS Business Logic for Java CAPS Projects” on page 35](#) and [Chapter 6, “Configuring JMS Clients” on page 64](#).

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

Figure 3 Java CAPS Projects



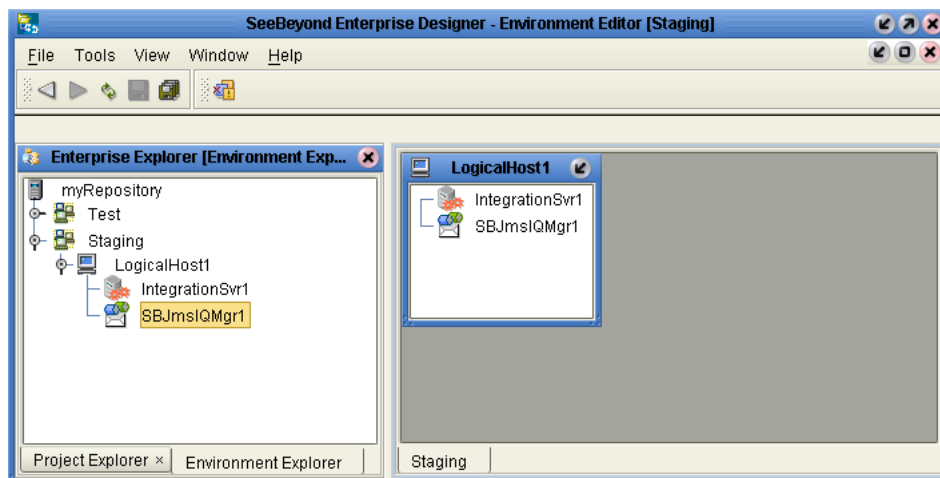
For more information about building Projects, refer to [“Project Design Phase: Topics, Queues and JMS Properties”](#) on page 17.

2.4.3 Environment Design Phase: Message Server Configurations

In the Java CAPS 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 [“Configuring Message Servers”](#) on page 72.

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**.

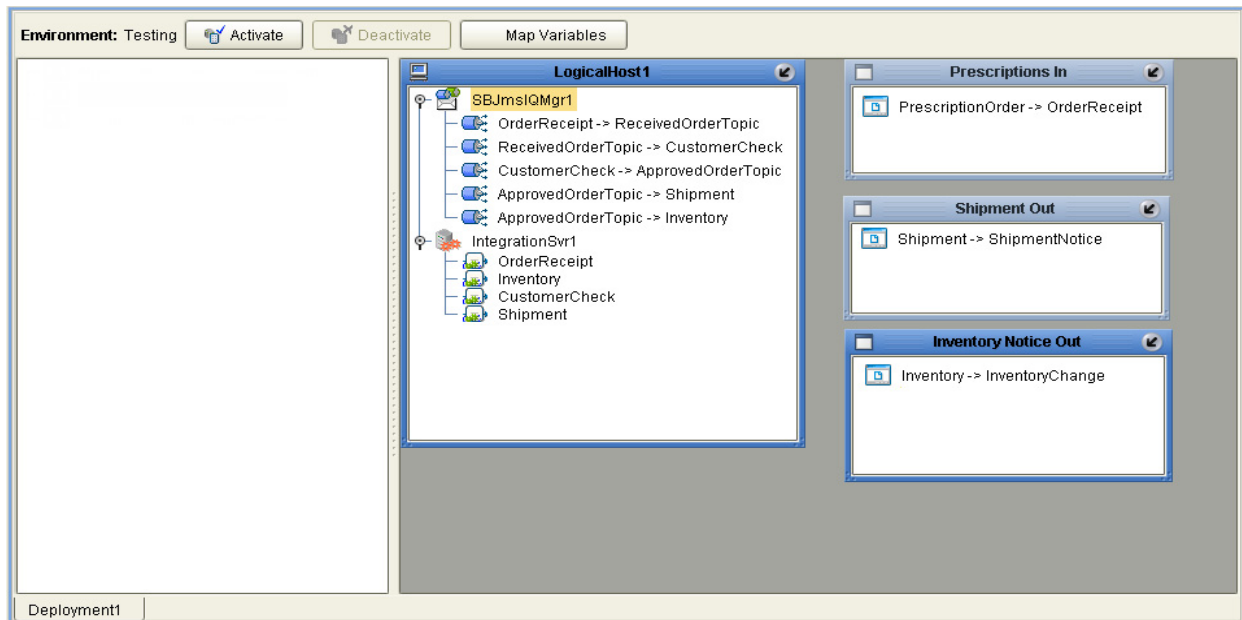
Figure 4 Adding Message Servers to an Environment



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 99](#). 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 Java CAPS Project is deployed, you can manage the JMS IQ Manager with the Enterprise Manager as described in [“Managing JMS IQ Managers with the Enterprise Manager” on page 102](#).

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 105](#).

2.5 Quick Overview: Building Java CAPS Projects with JMS

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

- 1 Install the SRE JMS IQ Manager if necessary. For information, refer to [“Installing Message Server .sar Files” on page 34](#).
- 2 Create a Project and Connectivity Map as described in the *Sun SeeBeyond eGate Integrator User’s Guide*.
- 3 Add Project components to the Connectivity Map; this includes adding message destinations as described in [“Adding Message Destinations to Projects” on page 35](#).
- 4 Create OTDs as described in the *Sun SeeBeyond eGate Integrator User’s 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 [“Creating Collaboration Definitions with JMS OTDs” on page 36](#).
- 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 37](#).
- 7 Bind Collaboration Definitions to Services as described in [“Binding Collaboration Definitions to Services” on page 60](#).
- 8 Link the Connectivity Map components; this includes connecting JMS Web Service Collaborations to their destinations as described in [“Linking JMS Collaborations to Message Destinations” on page 61](#).
- 9 Configure eWays and JMS clients. For JMS client configuration, refer to [“Configuring JMS Clients” on page 64](#).
- 10 Create a Java CAPS Environment and add message servers as described in [“Adding Message Servers to Logical Hosts” on page 72](#).
- 11 Configure the message server as described in [“Configuring the JMS IQ Manager Connections” on page 74](#).
- 12 Deploy the Java CAPS Project as described in the *Sun SeeBeyond eGate Integrator User’s Guide*.

2.6 JMS IQ Manager Administration Tools

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

- 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	“Configuring JMS Clients” on page 64 and “Configuring Message Servers” on page 72
Enterprise Manager	Application that allows you to start and stop message servers, view alerts, monitor, view, and edit live messages, for JMS IQ Managers	“Managing JMS IQ Managers with the Enterprise Manager” on page 102
STC MS Control Utility	Command-line tool that allows you to configure and manage JMS IQ Managers	“Managing JMS IQ Managers with the MS Control Utility” on page 105

2.7 Notes About the Java Message Queue

The following are important issues to be aware of when working with the Java Message Queue:

- The connector supports only sync concurrency mode; concurrent processing is unavailable for topics.
- The URL syntax must be `mq://host:port/serviceName?option1=value1&option2=value2`.
 - ♦ SJSMQ `jms` and `ssljms` services are supported.
 - ♦ If `serviceName` is omitted, `jms` service will be used as default.
 - ♦ To support one or more SJSMQ message server addresses, add a comma and another URL, e.g. `mq://host:port/serviceName?option1=value1&option2=value2, mq://host:port/serviceName?option1=value1&option2=value2`
 - ♦ Options can include options for JMSJCA and options for the client runtime. The latter options are propagated to the connection factories.
- The connector has been certified with client runtime version 3.6 SP4 (Build 1). When deploying into Sun Java System Application Server 8.1, make sure that the version of the `imqjmsra.jar` that is in the classpath of the server is of this version or higher.
- To enable concurrent processing on queues, make sure to provide this option in the configuration of the SJSMQ broker:


```
imq.autocreate.queue.maxNumActiveConsumers=-1
```

The default number of messages sent to one queue-receiver; this is likely too many to have a reasonable concurrency. Therefore it is recommended to set this additional parameter in the broker configuration:

```
imq.autocreate.queue.consumerFlowLimit=10
```

- The management MBean for management of messages in queues and topics is limited to queues only because the client runtime does not provide functionality for the management of topics.

To make management messages persistent, add this option in the configuration file of the SJSMQ broker:

```
imq.metrics.topic.persist=true
```

Refer to the documentation of SJSMQ for a description of this option, as well as other options such as `imq.metrics.interval`, `imq.metrics.topic.interval` and `imq.metrics.topic.timetolive`.

Functional Overview of the JMS IQ Manager

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

What's in This Chapter

- [About the JMS IQ Manager Database](#) on page 23
- [Message Processing Order](#) on page 25
- [About Performance](#) on page 30

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 the 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 default 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 Java CAPS 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 Segment Options” on page 77](#).

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 77](#). By default, the database resides in the following directory:

```
..\..\domains\logicalhostname\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 JMSIQMgrname directory contains a **Journal** directory, unless another location has been specified for the **Journal** property. The Journaling directory holds the journaling database files. For information, refer to [“Specifying the Location of the Journal Database” on page 82](#). 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).

3.1.3 The Default Database Configuration

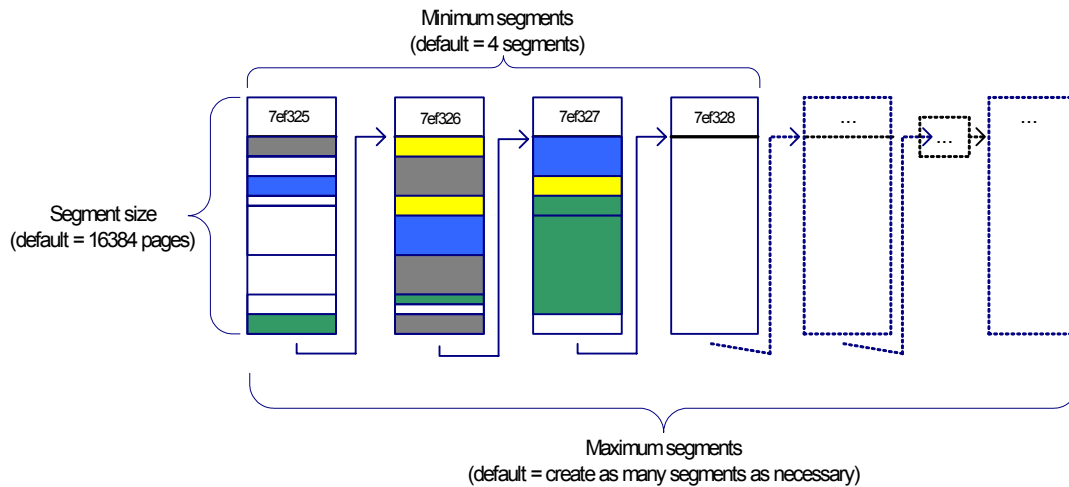
The default configuration for the JMS IQ Manager database is as follows:

- 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 6 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 6 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 6 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 6, 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 Java CAPS 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 29](#).
- You can specify a set of message destinations with a specific processing order.
For information, see [“Serial Processing Across a Destination Group” on page 30](#).

3.2.1 JMS IQ Manager FIFO Delivery Modes

You can specify several FIFO delivery modes for Java CAPS Projects; these ordering modes apply to all message destinations in the Java CAPS Project. The FIFO modes (for queues only) are:

- *Fully concurrent first-in, first-out (FIFO) order*
Receivers 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*
Receivers can retrieve messages when all older messages have been received or are being received, but must commit using time sequence.
- *Fully serialized FIFO order*
Receivers read a messages only after all messages have been received and commit messages using time sequence.

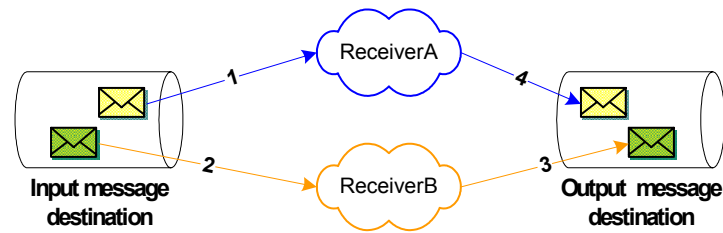
Setting the FIFO delivery mode only comes into play when multiple receivers or multiple processes within a single receiver subscribe to the same message destination. For a single consumer with a single process, processing for queues is by default fully serialized. The process of processing a message is as follows:

- 1 The receiver requests a message (receiver is ready to receive a message)
- 2 The receiver receives a message
- 3 The receiver processes the message

Fully Concurrent Processing

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

Figure 7 Fully Concurrent Processing



The figure above shows a sample delivery sequence for fully concurrent processing. In step 1 and 2, the receivers retrieve their messages from the input queue. Both receivers 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, receivers 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

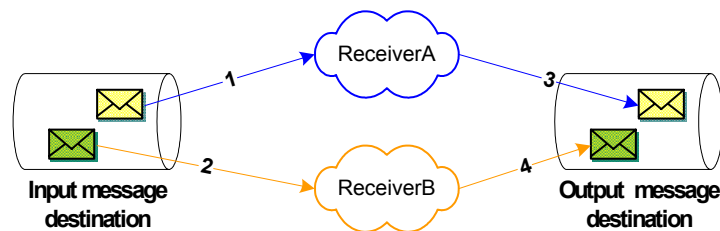
Benefits	Costs
Provides the best performance	Delivery not time sequenced
Receivers are not hampered by other receivers	

You specify fully concurrent processing for JMS IQ Managers with the **Special FIFO modes** property as described in [“Configuring FIFO Modes” on page 84](#).

Protected Concurrent Processing

In protected concurrent mode, receivers 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 8 Protected Concurrent Processing



The figure above shows a sample delivery sequence for protected concurrent processing. In step 1 and 2, the receivers retrieve their messages from the input queue. Both receivers 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.

ReceiverB might be ready to commit its message before ReceiverA, but must wait until ReceiverA commits its message (step 3). Only when ReceiverA's message has been committed, can ReceiverB 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

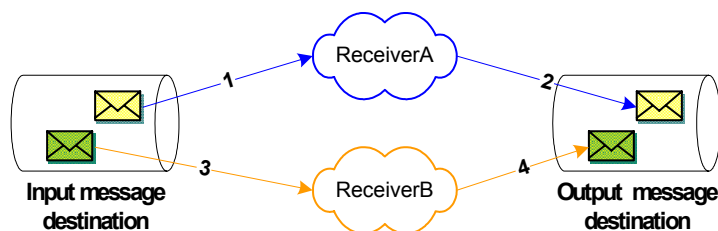
Benefits	Costs
Provides better performance than serialized processing	Provides less performance than fully concurrent processing
Messages are delivered by time sequence	

You specify protected concurrent processing for JMS IQ Managers with the **Special FIFO modes** property as described in [“Configuring FIFO Modes” on page 84](#).

Fully Serialized Processing

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

Figure 9 Fully Serialized Processing



The figure above shows a sample delivery sequence for serialized processing. In step 1, ReceiverA retrieves its message. ReceiverB might at this point be ready to receive its message, but must wait until ReceiverA has committed its message. After ReceiverA commits the message in step 2, ReceiverB 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 [“Configuring FIFO Modes” on page 84](#).

3.2.2 JMS Client Concurrency Effect on FIFO Delivery Modes

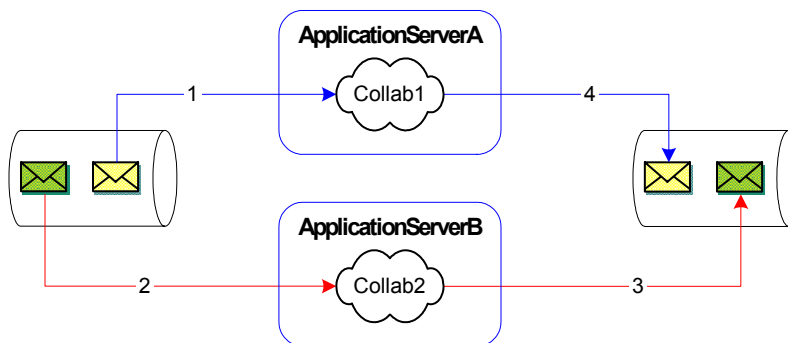
Apart from specifying FIFO delivery modes for queues as described above, you can also configure topics and queues for concurrency: connection consumer or serial mode.

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 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 10 Multiple Application Server Configuration



The table below lists how the connection consumer and serial mode settings 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

Concurrency mode	FIFO mode: fully concurrent
serial mode	fully serialized
connection consumer	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

Concurrency mode	FIFO mode: fully serialized	FIFO mode: protected concurrent	FIFO mode: fully concurrent
1 integration server serial mode	fully serialized no concurrency	protected concurrent no concurrency	fully serialized no concurrency
1 integration server connection consumer	fully serialized no concurrency	protected concurrent concurrent	no strict order maintained concurrent
>1 integration servers serial mode	fully serialized no concurrency	protected concurrent concurrent	no strict order maintained concurrent
>1 integration servers 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 Diagnostic Options](#) on page 97.

3.3 About Performance

Because of the large assortment of configuration parameters, you have a high degree of control over processing speed, memory use, and disk space. The JMS IQ Manager properties work together to allow you to fine-tune your system according to load and hardware constraints. For information, refer to [“Setting Segment Options” on page 77](#).

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 31](#).

To maximize performance

- Use the fastest disk possible.
- Allocating a new segment requires more time than freeing a cleaned-up segment.
- Set the segment size to lower because 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 the 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:

- **Server Throttling Threshold** 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, **Per-Destination Throttling Threshold** 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 **Server Throttling Threshold** 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 **Per-Destination Throttling Threshold** 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.*

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:

- **Server Throttling Threshold**=100,000
- **Per-Destination Throttling Threshold**=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.

Table 8 Publisher Throttling (Continued)

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:	
11:43	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 Message Server .sar Files

This chapter describes how to install support or SRE JMS IQ Manager.

To be able to design Java CAPS Projects that use this message server, you must install its .sar file.

What's in This Chapter

- [Installing Message Server .sar Files](#) on page 34

4.1 Installing Message Server .sar Files

The procedure below describes an overview of how to install the SRE JMS IQ Manager. For detailed installation instructions, refer to the *Sun Java Composite Application Platform Suite Installation Guide*.

Before you install a message server .sar file, install and download the following items using the Java CAPS Installer:

- Sun SeeBeyond Repository
- eGate Integrator
- Enterprise Designer
- Enterprise Manager
- Logical Host

To install message server .sar files

- 1 In the **Administrator** page of the Java CAPS Installer, click **Click to install additional products**.
- 1 In the **Administrator > Select** page, expand **Core Product**, select **JMSClientToSREJMSIQMgr** and click **Next**.
- 2 In the **Administrator > Upload** page, select the requested .sar files and click **Install Products**.

When the installation is finished, the "Installation Completed" message appears.

Building JMS Business Logic for Java CAPS Projects

This chapter describes how to build the business logic for JMS Java CAPS Projects in JMS OTDs, and how to add message destinations (topics and queues) to Projects in the Enterprise Designer.

What's in This Chapter

- [Adding Message Destinations to Projects](#) on page 35
- [Creating Collaboration Definitions with JMS OTDs](#) on page 36
- [Using JMS Methods in Collaboration Definitions](#) on page 37
- [Using JMS Message Properties in Collaborations](#) on page 57
- [Binding Collaboration Definitions to Services](#) on page 60
- [Linking JMS Collaborations to Message Destinations](#) on page 61

5.1 Adding Message Destinations to Projects

To start implementing JMS in a Java CAPS 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.
- 2 Create or click a Connectivity Map and drag the **Topic** or **Queue** icons from the Connectivity Map toolbar onto the canvas.

Figure 11 Creating Topics and Queues



- 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.

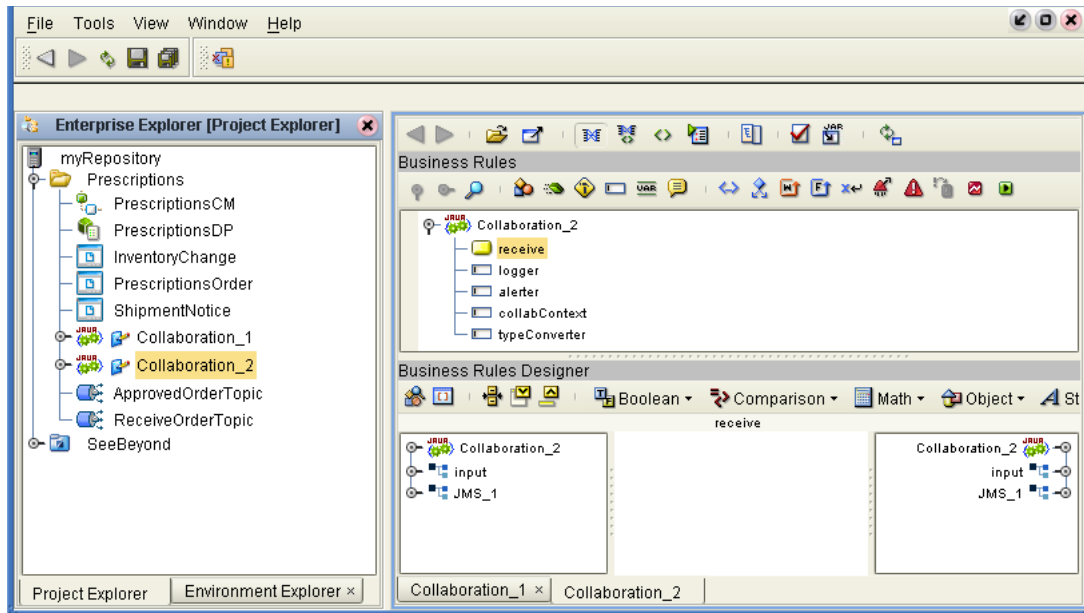
5.2 Creating Collaboration Definitions with JMS OTDs

To create Collaborations that read from topics or queues, you add the JMS OTD to the Collaboration Definition. This indicates to the Collaboration from which message destination it expects to receive messages, and allows you to build the JMS business rules.

The procedure below describes how to add the JMS OTD to Collaboration Definitions.

To create Collaboration Definitions with JMS OTDs

- 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 **Collaboration Definition (Java)**. The **Collaboration Definition Wizard—Step 1** dialog box appears.
- 3 Enter the name for the Collaboration in the **Collaboration Name** box and click **Next**.
- 4 Double-click **Sun SeeBeyond**, **eGate**, and **JMS** to open the JMS folder. This folder contains the *Receive* method.
- 5 Click **Receive** and click **Next**. The **Collaboration Definition Wizard—Step 3** dialog box appears.
- 6 Double-click **Sun SeeBeyond**, **eGate**, and **JMS**. This adds the JMS OTD to the Collaboration Definition.
- 7 Add any other OTDs necessary for this Collaboration Definition.
- 8 Click **Finish**. The **Collaboration Editor** window displays the Collaboration.



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.

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 37
- [JMS Methods Available for JMS Messages](#) on page 52

5.3.1 JMS Methods Available for JMS OTDs

Collaboration Definitions that use the JMS OTD as described in [“Creating Collaboration Definitions with JMS OTDs” on page 36](#) have the several JMS methods available. The table below lists the JMS methods available for the JMS OTD.

A second set of JMS methods is available for JMS messages received in the Collaboration—messages of the type `com.stc.connectors.jms.Message`. Those JMS methods are listed in [“JMS Methods Available for JMS Messages” on page 52](#).

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.

requestReplyTo(message, destName)

Syntax

```
requestReplyTo(message, destName)
```

Description

Sends a message to the explicit destination and waits for the reply message by using Topic/QueueRequestor with blocking fashion.

Parameters

Name	Type	Description
<i>message</i>	com.stc.connectors.jms.Message	The message object to send
<i>destName</i>	java.lang.String	The destination name

Return Value

Returns	Type
The reply message.	com.stc.connectors.jms.Message

Exceptions

Throws JMSEException when a message is null, destName is null, or the JMS provider fails to send and receive the message due to an internal error.

requestReplyTo(timeout, message, destName)

Syntax

```
requestReplyTo(timeout, message, destName)
```

Description

Sends a message to the explicit destination and receives the reply message by using Topic/QueueRequestor. Timeout is to be applied.

Parameters

Name	Type	Description
<i>timeout</i>	java.lang.long	The timeout in milliseconds
<i>message</i>	com.stc.connectors.jms.Message	The message object to send
<i>destName</i>	java.lang.String	The destination name

Return Value

Returns	Type
The reply message	com.stc.connectors.jms.Message

Exceptions

Throws JMSEException when a message is null, destName is null, or when the JMS provider fails to send and receive the message due to an internal error, or timeout expires.

requestReply(message)

Syntax

```
requestReply (message)
```

Description

Sends a message to the implicit destination and waits for the reply message by using Topic/QueueRequestor with blocking fashion.

Parameters

Name	Type	Description
<i>message</i>	com.stc.connectors.jms.Message	The message object to send

Return Value

Returns	Type
The reply message.	com.stc.connectors.jms.Message

Exceptions

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

requestReply(timeout, message)

Syntax

```
requestReply(timeout, message)
```

Description

Sends a message to the implicit destination and receives the reply message by using Topic/QueueRequestor. Timeout is to be applied.

Parameters

Name	Type	Description
<i>timeout</i>	java.lang.long	The timeout in milliseconds
<i>message</i>	com.stc.connectors.jms.Message	The message object to send

Return Value

Returns	Type
The reply message.	com.stc.connectors.jms.Message

Exceptions

Throws JMSEException if a message is null, JMS provider fails to send and receive the message due to an internal error, or timeout expires.

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.

Name	Type	Description
<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>	<i>int</i>	The delivery mode; 2 indicates persistent messages or 1 indicates nonpersistent messages.
<i>priority</i>	<i>int</i>	The message priority (0 through 9, with 9 being the highest priority).
<i>timetolive</i>	<i>long</i>	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>	<i>java.lang.String</i>	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>	<i>int</i>	The delivery mode; 2 indicates persistent messages or 1 indicates nonpersistent messages.
<i>priority</i>	<i>int</i>	The message priority (0 through 9, with 9 being the highest priority).
<i>timetolive</i>	<i>long</i>	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.
<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.

5.3.2 JMS Methods Available for JMS Messages

Collaboration Definitions that use the JMS OTD as described in [“Creating Collaboration Definitions with JMS OTDs” on page 36](#) 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 methods 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, UnsupportedEncodingException.

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 Using JMS Message Properties in Collaborations

For Collaboration Definitions built with the JMS OTD as described in [“Creating Collaboration Definitions with JMS OTDs” on page 36](#), 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)
Type	Text, Bytes, Map, Stream		getJMSType() setJMSType(string)

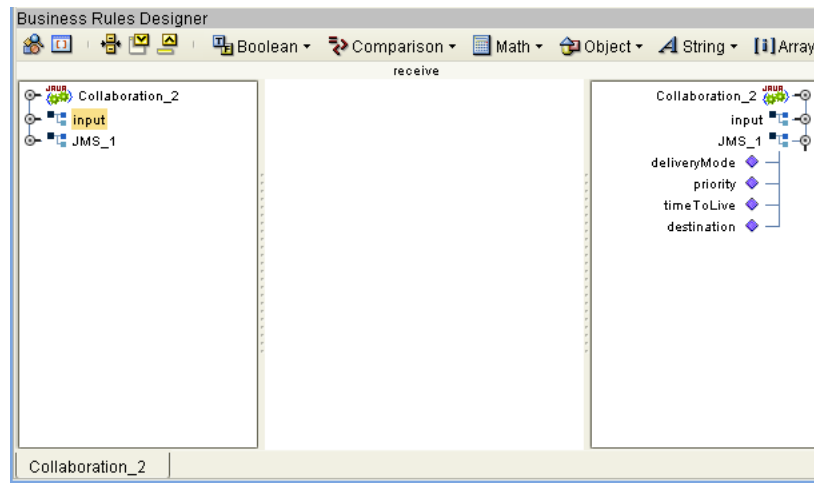
For more information about:

- Delivery mode and message priority, see [“Configuring JMS Clients” on page 64](#).
- Time to live, see [“Specifying the Maximum Lifetime for Live Messages” on page 81](#).

To set JMS message properties in Collaborations

- 1 In the Java Collaboration Rules Editor, expand **JMS_1**. The message priorities display as follows:

Figure 12 JMS Message Priorities




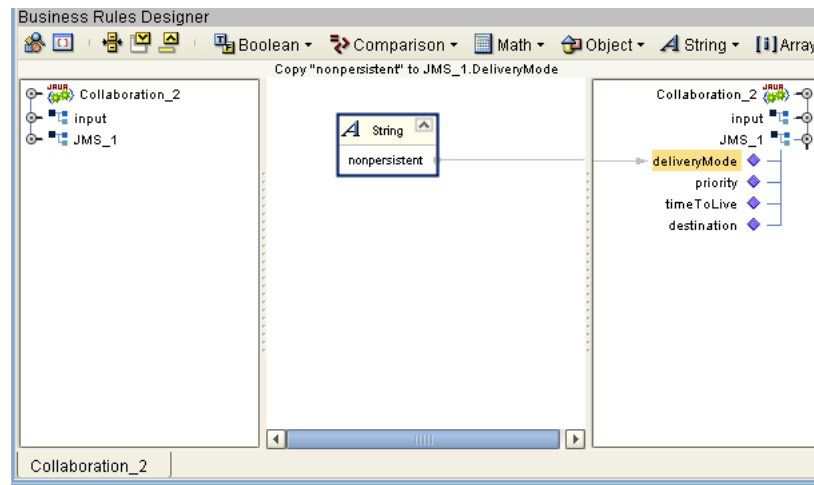
- 2 Click **Create Literal** . The **Create Literal** dialog box appears.
- 3 In the **Type** box, click the following option depending on the type of message property to be defined:
 - ♦ Delivery mode: string
 - ♦ Message priority: int
 - ♦ Time to live: long
 - ♦ Timeout: long
 - ♦ Destination: string
- 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 13 Setting JMS Message Properties



6 Click **Save**.

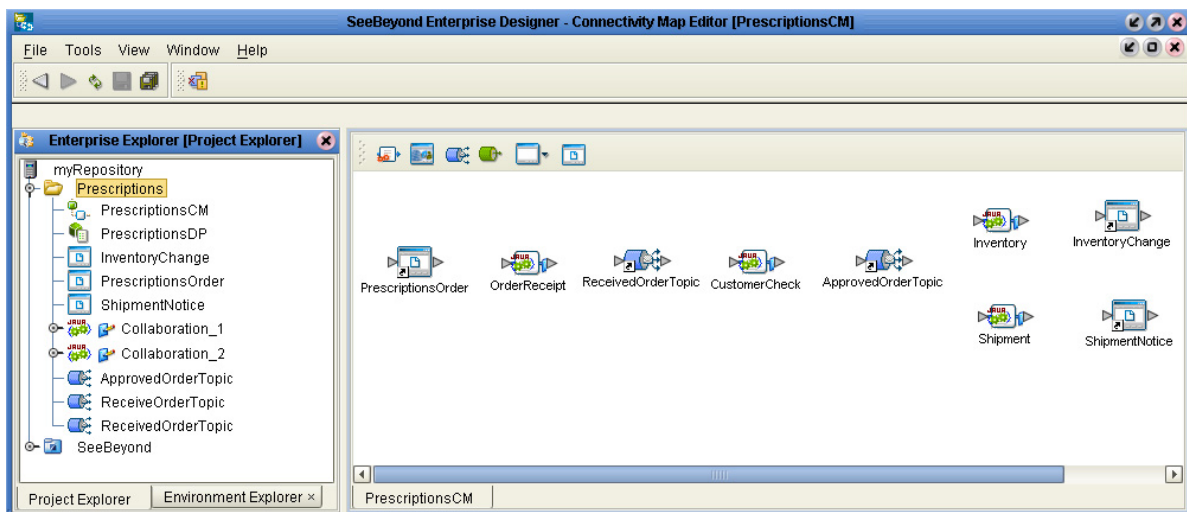
5.5 Binding Collaboration Definitions to Services

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

To bind Collaboration Definitions to services

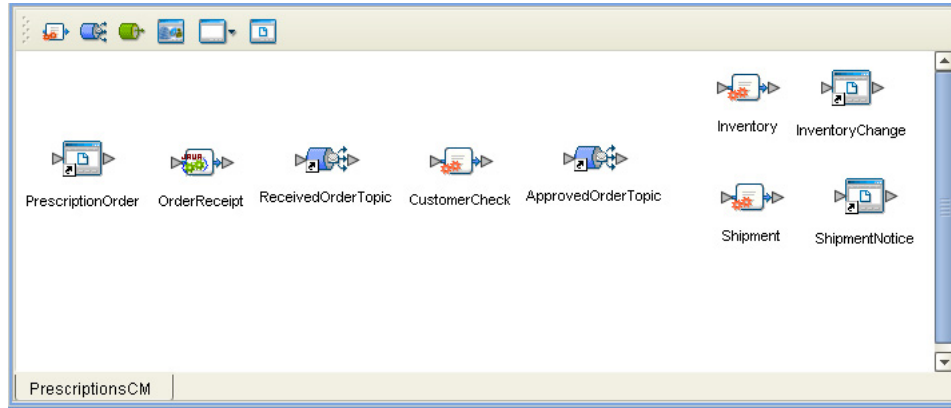
- 1 In the Project Explorer tab of the Enterprise Designer, expand the Project and open the Connectivity Map for the Project if necessary.
- 2 Click and hold the Collaboration Definition to be bound to a service.
- 3 Drag the definition over the service icon in the Connectivity Map as shown below.

Figure 14 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 15 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.

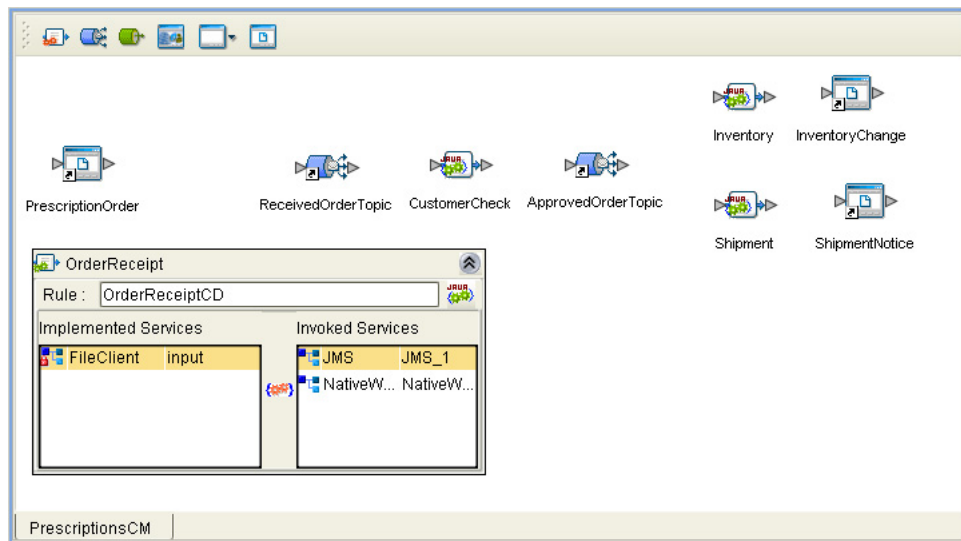
5.6 Linking JMS Collaborations to Message Destinations

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

To link JMS Collaborations to message destinations

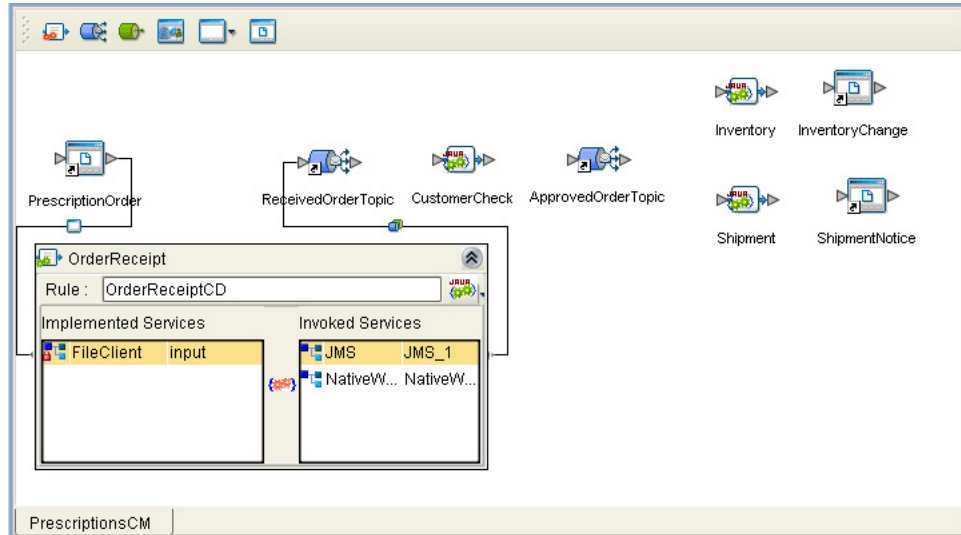
- 1 Double-click the Collaboration. The Collaboration binding windows appears as shown below.

Figure 16 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.

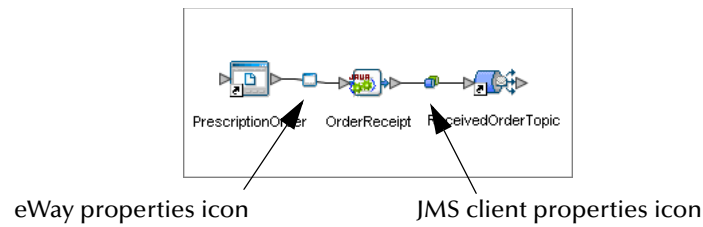
Figure 17 Binding Services to eWays



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

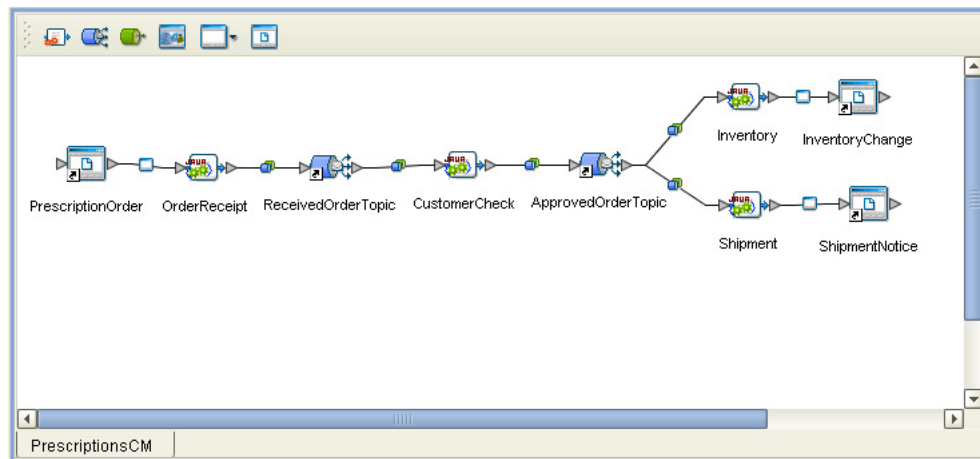
As you can see in the figure below, each connection contains an icon—in this case an eWay properties icon and a JMS client properties icon.

Figure 18 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 19 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 can configure the JMS clients.

What's in This Chapter

- [Configuring JMS Clients](#) on page 64

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 for the JMS IQ Manager.

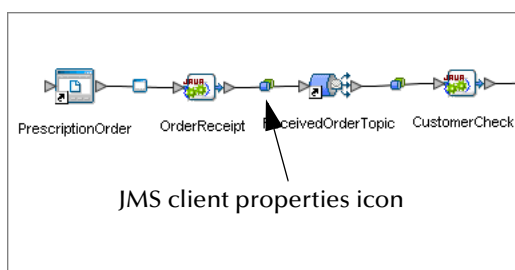
On the links between message destinations and other components there will be green/blue icons as shown in [Figure 20 on page 64](#). You double-click these JMS client properties icons to configure JMS clients.

You can specify JMS client properties for JMS IQ Managers. For non-JMS IQ Manager message servers, configure JMS clients with the message server management tools.

To configure JMS clients

- 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 Double-click the Connectivity Map. The Connectivity Map appears, showing the JMS client properties icon.

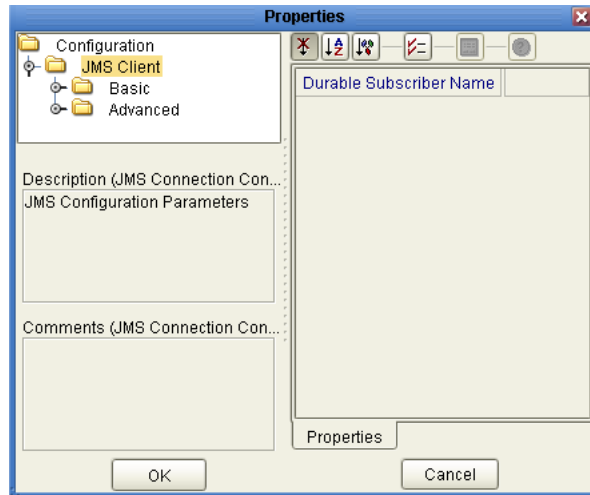
Figure 20 JMS Client Properties Icon



- 3 Double-click the JMS client properties icon. The **Properties** dialog box appears.

- 4 For inbound topics, click **JMS Client** and enter the name for the durable subscriber in the **Durable Subscriber Name** box.

Figure 21 Configuring JMS Clients



- 5 Expand the **JMS Client** folder and click **Basic**.

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

- 6 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 66
Delivery mode	Topic publishers Queue senders	page 67
Message selector	Topic subscribers Queue receivers	page 67
Priority	Topic publishers Queue senders	page 67
Transaction mode	Topic subscribers Topic publishers Queue receivers Queue senders	page 68
Idle timeout	Outbound	page 69
Maximum wait time	Outbound	page 69
Maximum pool size	Outbound	page 69
Steady pool size	Outbound	page 69

- 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 70
Server session pool	Topic subscribers Queue receivers	page 70

- 8 To reconfigure an existing JMS client, redeploy the Deployment Profile.

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 64](#).

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 67](#)).

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 application servers) on a system. This configuration does affect FIFO processing. For information, refer to ["Message Processing Order" on page 25](#).

You specify the maximum number of threads per server session pool as described in ["Setting the Number of ServerSessionPool Threads" on page 70](#). 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 64](#).

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 messages to be 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.

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 64](#).

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

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

If you use identifier start characters, it must be a character for which the method *Character.isJavaIdentifierStart* returns true as per JMS specification. If the identifier is invalid, the ICAN Monitor may show invalid message property names.

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 64](#).

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 [Using JMS Message Properties in Collaborations](#) on page 57.

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 64](#).

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. This setting prevents message loss and duplicate messages, even when a system unexpectedly shuts down.

Default

The default transaction mode is **Transacted**.

Required Values

Transacted or **XA**.

6.1.6 Specifying the Idle Timeout

You can set the transaction mode for outbound message destinations with the **Idle timeout** property. To display JMS client properties, refer to [“Configuring JMS Clients” on page 64](#).

The **Idle timeout** property specifies the number of seconds to wait before returning a connection to the pool.

Default

The default is 30 seconds.

6.1.7 Specifying the Maximum Wait Time

You can set the transaction mode for outbound message destinations with the **Maximum wait time** property. To display JMS client properties, refer to [“Configuring JMS Clients” on page 64](#).

The **Maximum wait time** property specifies the maximum amount of milliseconds to wait for acquiring a connection before throwing an exception.

Default

The default is 30000 milliseconds.

6.1.8 Specifying the Maximum Pool Size

You can set the transaction mode for outbound message destinations with the **Maximum pool size** property. To display JMS client properties, refer to [“Configuring JMS Clients” on page 64](#).

The **Maximum pool size** property specifies the maximum number of connections to be made to the message server.

Default

The default is 32 seconds.

6.1.9 Specifying the Steady Pool Size

You can set the transaction mode for outbound message destinations with the **Steady pool size** property. To display JMS client properties, refer to [“Configuring JMS Clients” on page 64](#).

The **Steady pool size** property specifies the minimum and initial number of connections maintained in the pool.

Default

The default is four connections.

6.1.10 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 64](#).

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**.

6.1.11 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 64](#).

This property is used in conjunction with the connection consumer setting of the **Concurrency** property ([“Setting JMS Client Message Processing Order” on page 66](#)). 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 25](#).

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.

Configuring Message Servers

This chapter describes how to add message servers to Logical Hosts in the Enterprise Designer. Once you have added a messages server, you can specify its connection information. This chapter also describes how to configure runtime message servers.

This chapter does not include configurations for third-party message servers and the Java Message Queue. Use the product-specific administration tool for those servers.

What's in This Chapter

- [Adding Message Servers to Logical Hosts](#) on page 72
- [Configuring Java Message Queue Connections](#) on page 73
- [Configuring the JMS IQ Manager Connections](#) on page 74
- [Configuring Runtime JMS IQ Managers](#) on page 75
- [Configuring SRE JMS IQ Managers](#) on page 99
- [Deploying Topics and Queues to Message Servers](#) on page 99

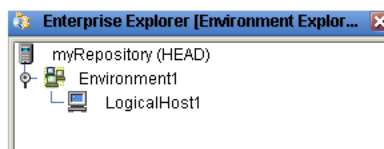
7.1 Adding Message Servers to Logical Hosts

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 Java CAPS 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 or JMS IQ Manager.

Figure 22 Expanding the Environments Folder



- 3 If necessary, create a Logical Host by right-clicking the Environment and selecting **New** and then **Logical Host**.
- 4 Right-click the Logical Host to which you want to add a message server, click **New**, and then click one of the following options:
 - ♦ **New Java System JMS Server** to create a Java Message Queue.
 - ♦ **SeeBeyond JMS IQ Manager** to create a JMS IQ Manager.
 - ♦ **WebSphere MQ** to create a WebSphere MQ.
 - ♦ **SRE JMS IQ Manager** to create an SRE JMS IQ Manager to connect to an SRE SeeBeyond Message Server.

Note: If you do not see the option for the message server you need, you have not installed the .sar file for that message server. For information about installing .sar files, refer to [“Installing Message Server .sar Files” on page 34](#).

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

- For JMS IQ Managers, [“Configuring the JMS IQ Manager Connections” on page 74](#).
- For SRE JMS IQ Managers, [“Configuring SRE JMS IQ Managers” on page 99](#).

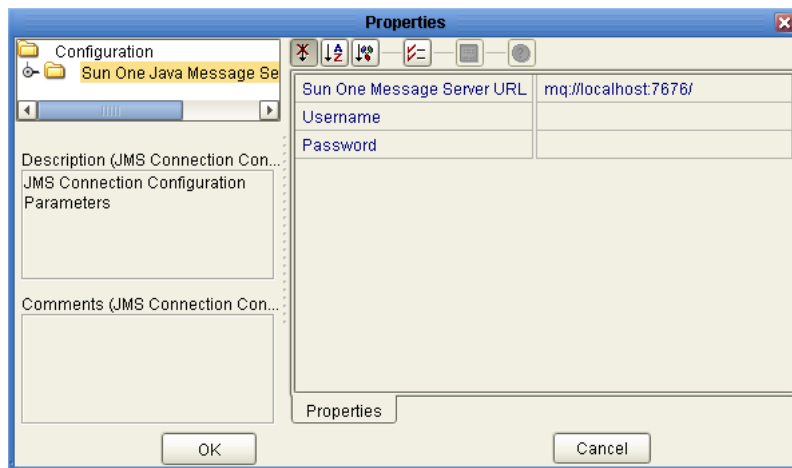
7.2 Configuring Java Message Queue Connections

The section below describes how to configure Java Message Queues connections.

To configure Java Message Queue connections

- 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 Java Message Queue, and click **Properties**. The **Properties** dialog box appears.

Figure 23 Configuring Java Message Queue Connection Information



- 4 Enter the following information:

Table 10 Configuring Java Message Queue Connections

Property	Enter
Sun One Message Server URL	The URL to connect to the Java Message Queue
Username	The username with which to connect to Java Message Queue
Password	The password to use for the Java Message Queue connection

- 5 Click **OK**.

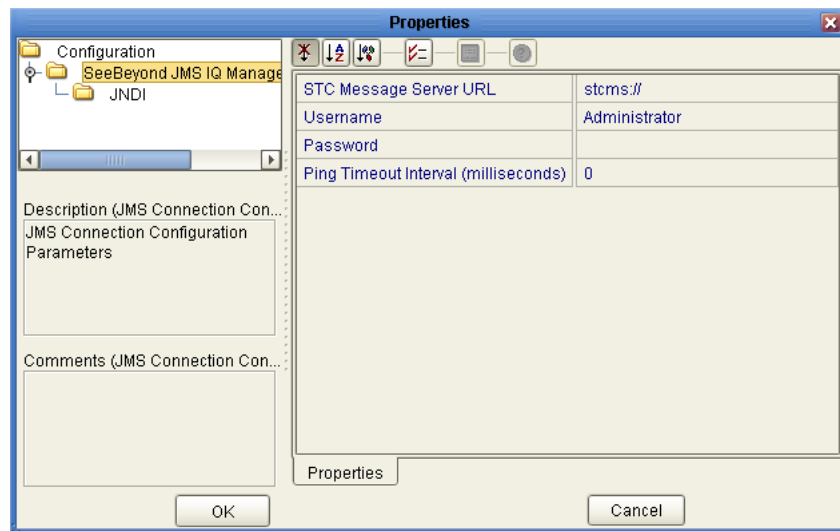
7.3 Configuring the JMS IQ Manager Connections

The section below describes how to configure JMS IQ Manager connections.

To configure JMS IQ Manager connections

- 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 JMS IQ Manager, and click **Properties**. The **Properties** dialog box appears.

Figure 24 Configuring JMS IQ Manager Connection Information



- 4 Enter the following information:

Table 11 Configuring JMS IQ Manager Connections

Property	Enter
STC Message Server URL	The URL to connect to for the JMS IQ Manager
Username	The username with which to connect to the JMS IQ Manager
Password	The password to use for the JMS IQ Manager connection
Ping timeout interval (milliseconds)	The number of milliseconds after which pinging the JMS IQ Manager times out

- 5 Click **OK**.

7.4 Configuring Runtime JMS IQ Managers

This section describes how to configure runtime JMS IQ Managers.

To configure runtime JMS IQ Managers

- 1 Start the Integration Configuration Tool as follows:
 - ♦ Run the command `http://localhostname:portnumber`, or
 - ♦ In the Enterprise Manager Explorer pane, right-click the Logical Host and click **Configure Integration Server**.

The Integration Server Administration application displays.

- 2 Click the JMS IQ Manager. The JMS IQ Manager configurations appears as follows.

Figure 25 Configuring Runtime JMS IQ Managers

Configuration > SeeBeyond JMS IQ Manager

Stable Storage Messaging Behavior Access Control Diagnostics Miscellaneous

Stable Storage Save Load Defaults

* Indicates required field

Segment ?

* Data Directory: ?
Path to the stcms database

* Block Size: ?
Segment block size

* Segment Size: ?
Specifies the segment size

* Minimum Number of Segment: ?
Specifies minimum number of segment

* Maximum Number of Segment: ?
Specifies maximum number of segment

Sync To Disk: ☐ Enabled ?
Controls cache synchronization to disk

Journaling and Expiration ?

Enable Message Expiration: ☒ Enabled ?
Enables message expiration.

* Maximum Lifetime: Second ?
Specifies message expiration time

Enable Journal: ☐ Enabled ?
Enables message journaling

* Journaling Maximum Lifetime: Second ?
Specifies journal expiration time

* Journal Directory: ?
Specifies journal location

Save Load Defaults

- 3 Enter the configuration properties for the JMS IQ Manager. The table below shows where to find detailed information for the configuration options.

JMS IQ Manager Configuration	Information
Segment settings	page 77
Journaling and message expiration	page 80
Throttling options	page 83
FIFO modes	page 84
Time dependency	page 86
Access control (security)	page 88
Diagnostics	page 97
Alert	page 91

- 4 Click **Save**.

7.4.1 Setting Segment Options

You set storage and journaling configurations for the JMS IQ Manager in the **Stable Storage** tab. To display this tab, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#). 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 111](#) or [“Republishing Messages from Queues” on page 112](#). 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 112](#).

Specifying the Location of JMS IQ Manager Database Files

You can specify where you want to JMS IQ Manager database files to reside for JMS IQ Managers with the **Data directory** property. To display this property, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#). The **Data directory** property is located in the **Stable Storage** tab.

The data directory holds the JMS IQ Manager database files (**stcms*.dbs**). 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 82](#). Journaling is disabled by default.

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 directory where the Logical Host **.zip** file was extracted and where *JMSIQMgrname* is the name of the JMS IQ Manager.

Specifying Segment Size

You can specify database segment size for JMS IQ Managers with the **Segment size** property. To display this property, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#). The **Segment size** property is located in the **Performance** tab.

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 a value larger than the sum of:

- The anticipated number of subscribers
- The anticipated maximum transaction size in bytes divided by the page size in bytes; the transaction size is the sum of the sizes of all messages in one transaction. If transactions span no more than one message, the maximum transaction size is equal to the size of the largest message.
- 10 extra pages for overhead

For example, on a UNIX system (where the page size is 1 KB) where you expect no more than 100 subscribers and that messages will not exceed 100 KB, and that only one message will be sent/received per transaction, you would set the segment size to at least $100 + (100\text{kb}/1\text{kb}) + 10 = 210$ pages

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.

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 larger than 1. Set this property to at least twice the total number of anticipated durable subscribers.

Specifying the Prefix for Database Filenames

You can specify the database filename prefix for JMS IQ Managers with the **Database file prefix** property. To display this property, refer to [“Configuring Runtime JMS IQ](#)

Managers” on page 75. The **Database file prefix** property is located in the **Stable Storage** tab.

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 Minimum Number of Segments

You can specify the minimum segment number for JMS IQ Managers with the **Min Number of Segment** property. To display this property, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#). The **Minimum Number of Segments** property is located in the **Performance** tab.

The **Minimum Number 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 **Maximum Number of Segments** property as described in [“Specifying the Maximum Number of Segments” on page 79](#).

The JMS IQ Manager uses the JMS IQ Manager database to store live persistent messages. For more information, refer to [“Setting Segment Options” on page 77](#) and [“Configuring JMS Clients” on page 64](#).

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 77](#).

Default

The default is 4 segments.

Required Values

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

Specifying the Maximum Number of Segments

You can specify the maximum segment number for JMS IQ Managers with the **Maximum Number of Segments** property. To display this property, refer to [“Configuring the JMS IQ Manager Connections” on page 74](#). The **Maximum Number of Segments** property is located in the **Performance** tab.

The **Maximum Number 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 Segment Options” on page 77](#) and [“Configuring JMS Clients” on page 64](#).

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 77](#).

You can also specify the minimum number of segments created initially for stable message storage with the **Minimum Number of Segments** property as described in [“Specifying the Minimum Number of Segments” on page 79](#).

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 **Server Throttling Threshold**, **Per-Destination Throttling Threshold**, and **Throttling lag** properties. For details, refer to [“Managing Resources” on page 31](#).

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.

Enabling Disk Synchronization

You can specify whether the JMS IQ Manager controls cache synchronization to disk for JMS IQ Managers with the **Sync to Disk** property. To display this property, refer to [“Configuring the JMS IQ Manager Connections” on page 74](#). The **Sync to Disk** property is located in the **Performance** tab.

The **Sync to Disk** 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 **Sync to Disk** property is disabled by default.

7.4.2 Setting Journaling and Message Expiration Options

You set journaling configurations for the JMS IQ Manager in the **Stable Storage** tab. To display this tab, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#).

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 111](#) or [“Republishing Messages from Queues” on page 112](#). 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 112](#).

Enabling Message Expiration

You can enable or disable message expiration for JMS IQ Managers with the **Enable Message Expiration** property. To display this property, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#). The **Enable Message Expiration** property is located in the **Stable Storage** tab.

When you enable message expiration, message are removed from the queue after the amount of time set for the **Maximum Lifetime** property has expired. For information about the **Maximum Lifetime** property, refer to [“Specifying the Maximum Lifetime for Live Messages” on page 81](#).

Default

By default, message expire after 30 days.

Specifying the Maximum Lifetime for Live Messages

You can specify the message expiration time for JMS IQ Managers with the **Maximum Lifetime** property. To display this property, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#). The **Maximum Lifetime** property is located in the **Stable Storage** tab.

The **Maximum Lifetime** 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).

Enabling Journaling

You can enable or disable journaling for JMS IQ Managers with the **Enable Journal** property. To display this property, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#). The **Enable Journal** property is located in the **Stable Storage** tab.

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 82](#). 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 113](#).

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 105](#).

Default

Journaling is disabled by default.

Specifying Time To Live for Journaled Messages

You can specify the time to live for journaled messages for JMS IQ Managers with the **Journal Maximum Lifetime** property. To display this property, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#). The **Journal Maximum Lifetime** property is located in the **Stable Storage** tab.

The **Journal Maximum Lifetime** property specifies the maximum amount of time in seconds that a journaled message lives before it expires. The JMS IQ Manager journals messages only when journaling is enabled as described in [“Enabling Journaling” on page 81](#). Journaling is disabled by default.

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 (.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 113](#).

Default

The default is 604800 seconds (7 days).

Specifying the Location of the Journal Database

You can specify the location of the journaling directory for JMS IQ Managers with the **Journal Directory** property. To display this property, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#). The **Journal Directory** property is located in the **Stable Storage** tab.

The **Journal directory** holds the journal database files and the journaling log file. You can enter an absolute path or a path relative to the `logicalhost\is\stcms\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 a journal directory when journaling is enabled. Journaling is disabled by default. For more information, refer to [“Enabling Journaling” on page 81](#).

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

Default

By default, the journal database files are stored in the following folder:

logicalhostname\logicalhost\stcms\JMSIQMgrname\Journal

where *logicalhostname* is the name of the directory where the Logical Host is running.

Required Values

An absolute path or a path relative to the **logicalhost\is\stcms\bin** directory.

7.4.3 Configuring Throttling Options

You set throttling options for the JMS IQ Manager in the **Messaging Behavior** tab. To display this tab, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#).

Specifying the Throttling Threshold for Message Destinations

You can specify the throttling threshold per message destination for JMS IQ Managers with the **Per-Destination Throttling Threshold** property. To display this property, refer to [“Configuring the JMS IQ Manager Connections” on page 74](#). The **Per-Destination Throttling Threshold** property is located in the **Performance** tab.

The **Per-Destination Throttling Threshold** 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 **Per-Destination Throttling Threshold – 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 Server Throttling Threshold

You can specify the throttling threshold for the total number of messages on the server with the **Server Throttling Threshold** property. To display this property, refer to [“Configuring the JMS IQ Manager Connections” on page 74](#). The **Server Throttling Threshold** property is located in the **Performance** tab.

The **Server Throttling Threshold** 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 31](#).

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 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 this property, refer to [“Configuring the JMS IQ Manager Connections” on page 74](#). The **Throttling Lag** property is located in the **Performance** tab.

You use the **Throttling Lag** property combined with the **Per-Destination Throttling Threshold** 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 **Per-Destination Throttling Threshold – Throttling Lag**. For information about the **Per-Destination Throttling Threshold** property, refer to the section above.

Default

The default is 100 messages.

Required Values

An integer from 0 through 99,999,999. The value must be set to less than that of the **Per-Destination Throttling Threshold** property.

7.4.4 Configuring FIFO Modes

You set FIFO modes options for the JMS IQ Manager in the **Messaging Behavior** tab. To display this tab, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#).

You can specify three different modes for FIFO delivery modes: fully concurrent, protected concurrent, or fully serialized.

In fully concurrent mode, receivers can retrieve messages from a destination only when all other messages have been received or are in the process of being received. Receivers 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.

For a general overview of message processing order, refer to [“Message Delivery Order” on page 13](#).

Specifying Fully Serialized Queues

You can specify the fully serialized queued for JMS IQ Managers with the **Fully Serialized Queues** property. To display this property, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#). The **Fully Serialized Queues** property is located in the **Messaging Behavior** tab.

You can specify three different modes for FIFO delivery modes: fully concurrent, protected concurrent, or fully serialized.

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

To implement serialized mode across multiple application servers, you must set the JMS clients for the consumers involved to serial mode. For more information, refer to [“Configuring JMS Clients” on page 64](#).

For a general overview of message processing order, refer to [“Message Delivery Order” on page 13](#).

Default

By default, all message destinations use fully concurrent delivery mode (see [“Configuring FIFO Modes” on page 84](#)).

Specifying Protected Concurrent Queues

You can specify the FIFO delivery mode for queues as protected concurrent with the **Fully Concurrent Queues** property. To display this property, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#). The **Protected Concurrent Queues** property is located in the **Messaging Behavior** tab.

You can specify three different modes for FIFO delivery modes: fully concurrent, protected concurrent, or fully serialized queues.

In protected concurrent mode, a receiver 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.

For a general overview of message processing order, refer to [“Message Delivery Order” on page 13](#).

Default

By default, all message destinations use fully concurrent delivery mode (see [“Configuring FIFO Modes” on page 84](#)).

Specifying Special FIFO Mode Expiration Time

You can specify an expiration time for FIFO modes with the **FIFO Expiration Time** property. To display this property, refer to [“Configuring the JMS IQ Manager Connections” on page 74](#). The **FIFO Expiration Time** property is located in the **Messaging Behavior** tab.

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).

7.4.5 Specifying Time Dependency

You set time dependency options for JMS IQ Managers in the **Messaging Behavior** tab. To display this tab, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#).

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 set specific time dependency option, 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 Topics** and **Time Dependency Queues** properties.

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.

Message properties such as JMS priority have no effect when time dependency is used.

For a general overview of message processing order, refer to [“Message Delivery Order” on page 13](#).

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.*

7.5 Setting Up Security

You set security options for JMS IQ Managers in the **Access Control** tab. To display this tab, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#).

Enabling Authentication and Authorization

You can enable user authentication and authorization for a producer or consumer with the **Require Authentication** property. To display this property, refer to [“Configuring the JMS IQ Manager Connections” on page 74](#). The **Require Authentication** property is located in the **Access Control** tab.

When authentication is enabled, access to the JMS IQ Manager is only granted when the connection has a valid user ID and password.

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 106](#).

Configuring JMS LDAP Server Roles

You must also configure your LDAP server when enabling the JMS IQ Manager for LDAP. Configure the LDAP server for one or more of the following roles:

Table 12 LDAP Server Roles

Role	Description
ms.application	This role can create connections, publishers, durable subscribers, subscribers, receivers, and senders. It can also unsubscribe, shut down, suspend, and resume.
ms.administrator	This role can create connections, publishers, durable subscribers, subscribers, receivers, and senders. It can also unsubscribe, shut down, edit, view, delete, create, suspend, and resume. Note: This role has the most permissions.
ms.operator	This role can create connections, publishers, durable subscribers, subscribers, receivers, and senders. It can also unsubscribe and view.
ms.connection	This role can create connections.
ms.receiver	This role can create connections, durable subscribers, subscribers, and receivers. It can also unsubscribe.
ms.sender	This role can create connections, publishers, and senders.
ms.viewer	This role can create connections, publishers, and subscribers. It can also view.
ms.gui	This role can create connections, publishers, and subscribers. It can also view, shut down, edit, delete, create, suspend, and resume.

For detailed information about configuring the LDAP server for roles, refer to the *eGate Integrator System Administrator Guide*.

Default

By default, JMS IQ Manager security is enabled.

Specifying the Default Realm

You can specify the default realm for topic subscribers and queue receivers with the **Default Realm** property. To display this property, refer to [“Configuring the JMS IQ Manager Connections” on page 74](#). The **Default Realm** property is located in the **Access Control** tab.

This security realm setting is used by the Integration Server for JMS Client interaction with J2EE components. The security realm must be configured as one of the security realms in the **security.xml** configuration file in the Integration Server directory.

Currently file realms, Active Directory service, and Sun Java System Directory Server are supported. The security realm defines users. To set up users, 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 Authentication and Authorization” on page 86](#).

Default

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

7.5.1 Enabling Sun Java System Directory Server

You configure the JMS IQ Manager to use a Sun Java System Directory Server in the **Access Control** tab. To display this tab, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#).

After enabling and configuring the JMS IQ Manager for the Sun Java System Directory Server, you must also configure the Sun server itself. For information, refer to the *eGate Integrator System Administrator Guide*.

The default values set for the Java System Directory are intended to match the standard schema of Sun Java System Directory Server. If you have not changed the standard schema, then all you need to do is change **localhost** in the **Naming Provider URL** property and **CAPS** in the **GroupsParentDN**, **Naming Security Principal**, **Role’s Parent DN**, and **User’s Parent DN** properties to match your environment. If you have changed the standard schema, be sure to check each property and (if necessary) modify the default value.

The table below describes each Sun Java System property.

Table 13 Sun Java System Directory Server Properties

Property	Description
GroupDNAttributeNamelnGroup	The name of the Distinguished Name attribute in group entries. The default value is entrydn .
GroupNameFieldlnGroupDN	The name of the group name field in group Distinguished Names. The default value is cn .
GroupOfUserFilterUnderGroupsParentDN	The LDAP search filter used to retrieve all of a user’s groups. This property follows the syntax supported by the java.text.MessageFormat class with {1} marking where the user’s Distinguished Name should be inserted. The default value is uniquemember={1} .

Table 13 Sun Java System Directory Server Properties (Continued)

Property	Description
GroupsParentDN	<p>The parent Distinguished Name of the group entries. In other words, this property specifies the root entry of the Groups portion of the LDAP directory.</p> <p>The default value is ou=Groups,dc=ican,dc=com.</p> <p>Be sure to change ican to an appropriate value for your environment. In addition, determine whether you need to set the value of the SearchGroupsSubTree property to True.</p>
Initial Naming Factory	<p>The fully qualified name of the factory class that creates the initial context. The initial context is the starting point for JNDI naming operations.</p> <p>The default value is com.sun.jndi.ldap.LdapCtxFactory.</p>
Naming Provider URL	<p>The URL of the JNDI service provider.</p> <p>The default value is ldap://localhost:389.</p> <p>Be sure to change localhost to an appropriate value for your environment.</p>
Naming Security Authentication	<p>The security level to use in JNDI naming operations.</p> <p>The default value is simple.</p>
Naming Security Credentials	<p>The password of the naming security principal.</p> <p>The default value is STC.</p>
Naming Security Principal	<p>The security principal used for connecting to the LDAP server.</p> <p>The default value is uid=Administrator,ou=People,dc=ican,dc=com.</p> <p>Be sure to change ican to an appropriate value for your environment.</p>

Table 13 Sun Java System Directory Server Properties (Continued)

Property	Description
Role's Parent DN	<p>The parent Distinguished Name of the role entries. In other words, this property specifies the root entry of the Roles portion of the LDAP directory.</p> <p>The default value is dc=ican,dc=com.</p> <p>Be sure to change ican to an appropriate value for your environment. In addition, determine whether you need to set the value of the SearchRolesSubTree property to True.</p>
RoleNameAttributeNameInUser	<p>The name of the role name attribute in user entries.</p> <p>The default value is nsroledn.</p>
RoleNameFieldInRoleDN	<p>The name of the role name field in role Distinguished Names.</p> <p>The default value is cn.</p>
SearchGroupsSubTree	<p>By default, the Groups portion of the LDAP directory is searched only one level below the root entry. To enable searches of the entire subtree, set the value to True.</p> <p>The default value is False.</p>
SearchRolesSubTree	<p>By default, the Roles portion of the LDAP directory is searched only one level below the root entry. To enable searches of the entire subtree, set the value to True.</p> <p>The default value is False.</p>
SearchUsersSubTree	<p>By default, the Users portion of the LDAP directory is searched only one level below the root entry. To enable searches of the entire subtree, set the value to True.</p> <p>The default value is False.</p>

Table 13 Sun Java System Directory Server Properties (Continued)

Property	Description
User's Parent DN	<p>The parent Distinguished Name of the user entries. In other words, this property specifies the root entry of the Users portion of the LDAP directory.</p> <p>The default value is ou=People,dc=ican,dc=com.</p> <p>Be sure to change ican to an appropriate value for your environment. In addition, determine whether you need to set the value of the SearchUsersSubTree property to True.</p>
UserDNAttributeNameInUser	<p>The name of the Distinguished Name attribute in user entries.</p> <p>The default value is entrydn.</p>
UserIDAttributeNameInUser	<p>The name of the user ID attribute in user entries.</p> <p>The default value is uid.</p>

7.5.2 Enabling Microsoft Active Directory Service

You configure the JMS IQ Manager to use the Microsoft Active Directory service in the **Active Directory Service** tab. To display this tab, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#).

After enabling and configuring the JMS IQ Manager for the Active Directory service, you must also configure the service itself. For information, refer to the *eGate Integrator System Administrator Guide*.

The default values set for the Active Directory Service are intended to match the standard schema of Active Directory. If you have not changed the standard schema, then all you need to do is change **localhost** in the **Naming Provider URL** property and **CAPS** in the **GroupsParentDN**, **Naming Security Principal**, **Role's Parent DN**, and **User's Parent DN** properties to match your environment. If you have changed the standard schema, be sure to check each property and (if necessary) modify the default value.

The table below describes each Active Directory Service property.

Table 14 Active Directory Properties

Property	Description
GroupDNAttributeNameInGroup	<p>The name of the Distinguished Name attribute in group entries.</p> <p>The default value is distinguishedName.</p>

Table 14 Active Directory Properties

Property	Description
GroupNameFieldInGroupDN	<p>The name of the group name field in group Distinguished Names.</p> <p>The default value is cn.</p>
GroupOfUserFilterUnderGroupsParentDN	<p>The LDAP search filter used to retrieve all of a user's groups. This property follows the syntax supported by the java.text.MessageFormat class with {1} marking where the user's Distinguished Name should be inserted.</p> <p>The default value is (&(member={1})(objectclass=group)).</p>
GroupsParentDN	<p>The parent Distinguished Name of the group entries. In other words, this property specifies the root entry of the Groups portion of the LDAP directory.</p> <p>The default value is cn=users,dc=ican,dc=com.</p> <p>Be sure to change ican to an appropriate value for your environment. In addition, determine whether you need to set the value of the SearchGroupsSubTree property to True.</p>
Initial Naming Factory	<p>The fully qualified name of the factory class that creates the initial context. The initial context is the starting point for JNDI naming operations.</p> <p>The default value is com.sun.jndi.LdapCtxFactory.</p>
Naming Provider URL	<p>The URL of the JNDI service provider.</p> <p>The default value is ldap://localhost:389.</p> <p>Be sure to change localhost to an appropriate value for your environment.</p>
Naming Security Authentication	<p>The security level to use in JNDI naming operations.</p> <p>The default value is simple.</p>
Naming Security Credentials	<p>The password of the naming security principal.</p> <p>The default value is STC.</p>

Table 14 Active Directory Properties

Property	Description
Naming Security Principal	<p>The security principal used for connecting to the LDAP server.</p> <p>The default value is cn=Administrator,cn=Users,dc=ican,dc=com.</p> <p>Be sure to change ican to an appropriate value for your environment.</p>
Role's Parent DN	<p>The parent Distinguished Name of the role entries. In other words, this property specifies the root entry of the Roles portion of the LDAP directory.</p> <p>The default value is ou=ICANRoles,dc=ican,dc=com.</p> <p>Be sure to change ican to an appropriate value for your environment. In addition, determine whether you need to set the value of the SearchRolesSubTree property to True.</p>
RoleDNAttributeNameInRole	<p>The name of the Distinguished Name attribute in role entries.</p> <p>The default value is cn.</p>
RolesOfUserFilterUnderRolesParentDN	<p>The LDAP search filter used to retrieve all of a user's roles. This property follows the syntax supported by the java.text.MessageFormat class with {1} marking where the user's Distinguished Name should be inserted.</p> <p>The default value is (&(member={1})(objectclass=group)).</p>
SearchGroupsSubTree	<p>By default, the Groups portion of the LDAP directory is searched only one level below the root entry. To enable searches of the entire subtree, set the value to True.</p> <p>The default value is False.</p>
SearchRolesSubTree	<p>By default, the Roles portion of the LDAP directory is searched only one level below the root entry. To enable searches of the entire subtree, set the value to True.</p> <p>The default value is False.</p>

Table 14 Active Directory Properties

Property	Description
SearchUsersSubTree	<p>By default, the Users portion of the LDAP directory is searched only one level below the root entry. To enable searches of the entire subtree, set the value to True.</p> <p>The default value is False.</p>
User's Parent DN	<p>The parent Distinguished Name of the user entries. In other words, this property specifies the root entry of the Users portion of the LDAP directory.</p> <p>The default value is cn=Users,dc=ican,dc=com.</p> <p>Be sure to change ican to an appropriate value for your environment. In addition, determine whether you need to set the value of the SearchUsersSubTree property to True.</p>
UserDNAttributeNameInUser	<p>The name of the Distinguished Name attribute in user entries.</p> <p>The default value is distinguishedName.</p>
UserIDAttributeNameInUser	<p>The name of the user ID (that is, the login ID) attribute in user entries.</p> <p>The default value is sAMAccountName.</p>

7.5.3 Enabling a Generic LDAP Server

You enable a generic LDAP server for the JMS IQ Manager with the Enable Generic LDAP Server property. To display this property, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#).

After enabling and configuring the JMS IQ Manager for the a generic LDAP server, you must also configure the server itself.

The default values for the generic LDAP server are intended to match the standard schema. If you have not changed the standard schema, then all you need to do is change **localhost** in the **Naming Provider URL** property and **CAPS** in the **GroupsParentDN**, **Role's ParentDN**, and **User's ParentDN** properties to match your environment. If you have changed the standard schema, be sure to check each property and (if necessary) modify the default value.

The table below describes each generic LDAP property.

Table 15 OpenLDAP Directory Server Properties

Property	Description
GroupNameFieldInGroupDN	<p>The name of the group name field in group Distinguished Names.</p> <p>The default value is cn.</p>
GroupsOfUserFilterUnderGroupsParentDN	<p>The LDAP search filter used to retrieve all of a user's groups. This property follows the syntax supported by the java.text.MessageFormat class with {1} marking where the user's Distinguished Name should be inserted.</p> <p>The default value is uniqueMember={1}.</p>
GroupsParentDN	<p>The parent Distinguished Name of the group entries. In other words, this property specifies the root entry of the Groups portion of the LDAP directory.</p> <p>The default value is ou=Groups,dc=ican,dc=com.</p> <p>Be sure to change ican to an appropriate value for your environment. In addition, determine whether you need to set the value of the SearchGroupsSubTree property to True.</p>
Initial Naming Factory	<p>The fully qualified name of the factory class that creates the initial context. The initial context is the starting point for JNDI naming operations.</p> <p>The default value is com.sun.jndi.Ldap.LdapCtxFactory.</p>
Naming Provider URL	<p>The URL of the JNDI service provider.</p> <p>The default value is ldap://localhost:389.</p> <p>Be sure to change localhost to an appropriate value for your environment.</p>
Naming Security Authentication	<p>The security level to use in JNDI naming operations.</p> <p>The default value is simple.</p>

Table 15 OpenLDAP Directory Server Properties

Property	Description
Role's ParentDN	<p>The parent Distinguished Name of the role entries. In other words, this property specifies the root entry of the Roles portion of the LDAP directory.</p> <p>The default value is ou=ICANRoles, dc=ican,dc=com.</p> <p>Be sure to change ican to an appropriate value for your environment. In addition, determine whether you need to set the value of the SearchRolesSubTree property to True.</p>
RoleNameAttributeNameInRole	<p>The name of the role name attribute in user entries.</p> <p>The default value is cn.</p>
RolesOfUserFilterUnderRolesParentDN	<p>The LDAP search filter used to retrieve all of a user's roles. This property follows the syntax supported by the java.text.MessageFormat class with {1} marking where the user's Distinguished Name should be inserted.</p> <p>The default value is uniquemember={1}.</p>
SearchGroupsSubTree	<p>By default, the Groups portion of the LDAP directory is searched only one level below the root entry. To enable searches of the entire subtree, set the value to True.</p> <p>The default value is False.</p>
SearchRolesSubTree	<p>By default, the Roles portion of the LDAP directory is searched only one level below the root entry. To enable searches of the entire subtree, set the value to True.</p> <p>The default value is False.</p>
SearchUsersSubTree	<p>By default, the Users portion of the LDAP directory is searched only one level below the root entry. To enable searches of the entire subtree, set the value to True.</p> <p>The default value is False.</p>

Table 15 OpenLDAP Directory Server Properties

Property	Description
User's ParentDN	<p>The parent Distinguished Name of the user entries. In other words, this property specifies the root entry of the Users portion of the LDAP directory.</p> <p>The default value is ou=People,dc=ican,dc=com.</p> <p>Be sure to change ican to an appropriate value for your environment. In addition, determine whether you need to set the value of the SearchUsersSubTree property to True.</p>
UserIDAttributeNameInUser	<p>The name of the user ID attribute in user entries.</p> <p>The default value is uid.</p>

7.5.4 Setting Diagnostic Options

You set diagnostic configurations for the JMS IQ Manager in the **Diagnostics** tab. To display this tab, refer to [“Configuring the JMS IQ Manager Connections” on page 74](#).

Specifying the Logging Level

You can specify the type of errors to be included in the JMS IQ Manager log for JMS IQ Managers with the **Logging level** property. To display this property, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#). The **Logging level** property is located in the **Diagnostic** tab.

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

Set this property to INFO for maximum debugging; this setting logs all messages. WARN to log warning messages, error messages, and fatal messages. To log only error messages and fatal messages, set this property to ERROR. To log only fatal messages, specify FATAL.

Default

By default, the JMS IQ Manager logs warning messages, error messages, and fatal messages.

Specifying the Journal Logging Level

You can specify the type of journaling errors to be included in the JMS IQ Manager journal log with the **Logging Level of Journaler** property. To display this property, refer to [“Configuring Runtime JMS IQ Managers” on page 75](#). The **Journal logging level** property is located in the **Diagnostic** tab.

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

Set this property to INFO for maximum debugging; this setting logs all messages. WARN to log warning messages, error messages, and fatal messages. To log only error messages and fatal messages, set this property to ERROR. To log only fatal messages, specify FATAL.

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 82](#).

Default

By default, the JMS IQ Manager journal log includes warning messages, error messages, and fatal messages.

Specifying the Maximum Log File Size

You can specify the maximum size for the JMS IQ Manager log file with the **Max log file size** property. To display this property, refer to [“Configuring the JMS IQ Manager Connections” on page 74](#). The **Maximum Log File Size** property is located in the **Diagnostic** tab.

If the JMS IQ Manager attempts to log more than the specified log file size, the log file is rename to **JMSIQMgrname.log.1**, and a new file is created with the name **JMSIQMgrname.log.2**. Be default, the JMS IQ Manager can create five log files. You can set a limit to the number of log files with the **Number of Backup Log Files** property as described in [“Specifying the Maximum Number of Backup Log Files” on page 98](#).

Default

The default log file size is 10 MB.

Required Values

An integer larger than 0, indicating the MB of the log file size.

Specifying the Maximum Number of Backup Log Files

You can specify the maximum number of JMS IQ Manager log files in the JMS IQ Manager properties with the **Number of backup log files** property. To display this property, refer to [“Configuring the JMS IQ Manager Connections” on page 74](#). The **Number of backup log files** property is located in the **Diagnostic** tab.

If the JMS IQ Manager attempts to log more than the specified log file size, the log file is rename to **JMSIQMgrname.log.1**, and a new file is created with the name **JMSIQMgrname.log.2**. With the **Number of backup log files** property you can set a limit to the number of log files created.

Default

The default number of backup log files created is five.

Required Values

An integer larger than 0, indicating the number of backup log files.

7.5.5 Enabling Alerts

You can enable alert generation for the JMS IQ Manager with the Enable Alert property on the Miscellaneous tab. To display this tab, refer to [“Configuring the JMS IQ Manager Connections” on page 74](#).

Default

Alert generation is disabled by default.

7.6 Configuring SRE JMS IQ Managers

To have a Java CAPS Project connect to an SRE SeeBeyond Message Server at runtime, you add an SRE JMS IQ Manager to the Logical Host in the Enterprise Designer.

To be able to add and configure an SRE JMS IQ Manager, you must have installed **JMSClientToSREJMSIQMgr.sar** as described in [“Installing Message Server .sar Files” on page 34](#).

The SRE JMS IQ Manager is a container that tells eGate how to connect to the SRE SeeBeyond Message Server, which must already be running. You enter the connection information (hostname and port number) in the SRE JMS IQ Manager properties in the Java CAPS Environment.

7.7 Deploying Topics and Queues to Message Servers

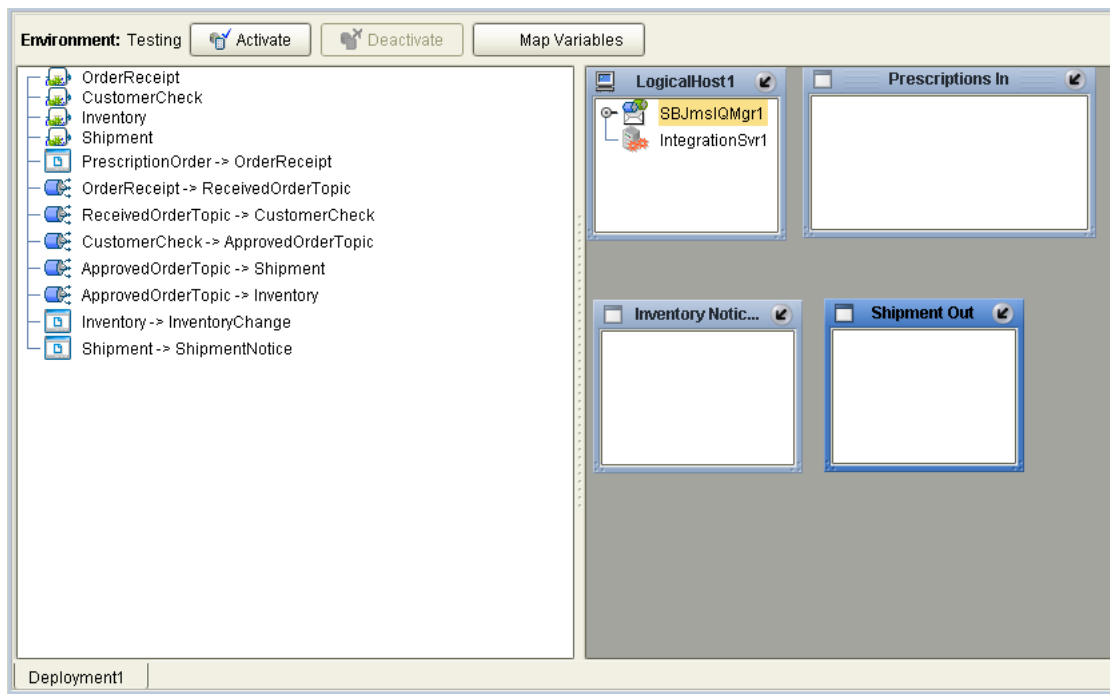
Deploying message destinations to message servers is the last step in building a Java CAPS Project, after you have created the Environment and configured the Environment components. To deploy message destinations, you create a Deployment Profile as described below.

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

To deploy topics and queues to message servers and JMS IQ Managers

- 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.

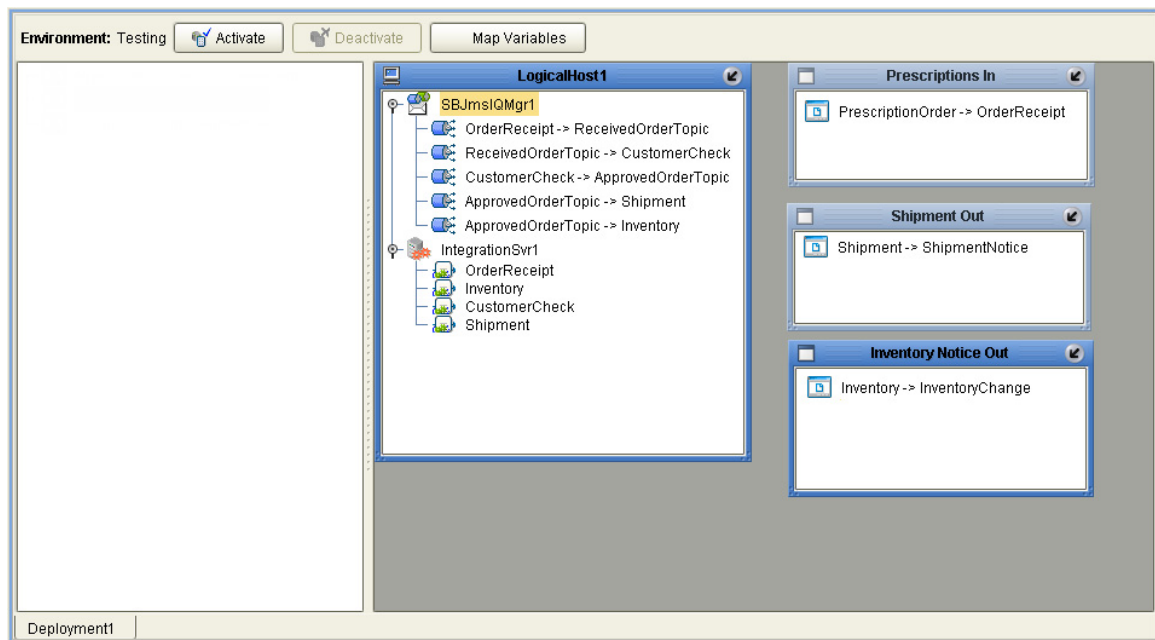
Figure 26 Initial Deployment Profile



- 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 application servers.
- 5 Drag the eWay components to the eWays.

The figure below shows an example of a Deployment Profile.

Figure 27 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*.

For information about integrating Java Integration Projects and JMS IQ Managers in SRE schemas, refer to the *Sun Java Composite Application Platform Suite Upgrade Guide*.

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.

What's in This Chapter

- [About Administration Tools](#) on page 102
- [Managing JMS IQ Managers with the Enterprise Manager](#) on page 102
- [Managing JMS IQ Managers with the MS Control Utility](#) on page 105

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 monitor message destinations, and view message properties and payloads. For information, refer to [Managing JMS IQ Managers with the Enterprise Manager](#) on page 102.

The 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 105.

8.2 Managing JMS IQ Managers with the Enterprise Manager

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

- [Monitoring Topics and Queues](#) on page 103
- [Viewing Message Properties](#) on page 104
- [Viewing Message Payload](#) on page 105

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

8.2.1 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 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 In Enterprise Manager, expand the Environment, the Logical Host, and the message server for which you want to view topics or queues.
- 2 Click the topic or queue to be viewed.

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

Table 16 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.
# Subscribers	Number of subscribers <i>registered to consume</i> messages for this topic (including durable subscribers that are currently disconnected).

Table 16 Topic Properties (Continued)

Property Name	Description
Last Pub Date/Time	The date and timestamp of the most recent message currently available in the topic. This field shows the last publication date and time of the last message when there are no messages available.

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

Table 17 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 shows the last publication date and time of the last message when there are no messages available.

8.2.2 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 Enterprise Manager, select the topic or queue for which you want to view a message as described in [“Monitoring Topics and Queues” on page 103](#).
- 2 Under **List**, click the topic or queue for which you want to view message properties.
- 3 Under **Details: messagedestinationname**, click **Messages**.
- 4 Select the unread or journaled message and click **Properties**. The table below describes the properties listed.

Table 18 Message Properties

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.

Table 18 Message Properties (Continued)

Property Name	Description
Message Type	The message type of the message, such as text or bytes.
Message EnqueueTime	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 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.
Correlation ID	Correlation identification.
Message PayloadSize	The message payload size in bytes.
Redelivery Flag	Indicates whether this message is set for redelivery.
Time Stamp	Day, date, and time when the message was received.
Sequence number	The sequence number of the message.

8.2.3 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 104](#).
- 2 Click **Summary**. The **Summary** page appears.
- 3 To select a message, you can enter the message number in the **View/Edit** box, or you can drag the message slider.

8.2.4 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 104](#).
- 2 Under **Details: messagedestinationname**, click the message for which you want to view the payload.
- 3 Click **View/Edit**. The **Edit Payload for Message** dialog box appears.

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.

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 [“Setting Up Security” on page 86](#).

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

```
stcmsctrlutil --help
```

```
stcmsctrlutil --version
```

8.3.2 Flags and Arguments

Table 19 MS Control Utility Flags and Arguments

Shortcut	Flag arguments	Purpose
	--version	View utility version information.
	--help	View help information.
	-username <i>username</i>	Supply the user name to connect to the JMS IQ Manager. This flag is mandatory when JMS IQ security is enabled.

Table 19 MS Control Utility Flags and Arguments (Continued)

Shortcut	Flag arguments	Purpose
	-userpassword <i>userpassword</i>	Supply the password to connect to the JMS IQ Manager. This flag is mandatory when JMS IQ security is enabled.
	-msversion	View server version information.
	-shutdown	Shut down the server.
	-status	View server status.
-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	-queuelist	List all queues for this server.
-rlfq	-recvlistforqueue <i>queueenamel</i>	List all receivers for the specified queue.
-rla	-recvlistall	List all receivers for all queues combined.
-qs	-queuestat <i>queueenamel</i>	View statistics for a specific queue.
	-host <i>hostname</i>	Specify the name of the Logical Host. If not specified, the default is: -host localhost For hosts other than localhost and flags other than --help and --version, -host is required.
	-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: -port 7555 For ports other than 7555 and flags other than --help and --version, -port is required.
	-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>queueenamel</i>	Create a new queue with the specified name.
-dq	-deletequeue <i>queueenamel</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 eGate .
-ds	-deletesub <i>topicname subname clientname</i>	Delete a certain subscriber from the specified topic and client. For <i>clientname</i> , specify eGate .
-qml	-qmsglist <i>queueenamel 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.

Table 19 MS Control Utility Flags and Arguments (Continued)

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

8.3.3 Shutting Down the Server

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

8.3.4 Viewing JMS IQ Manager Statistics

```
stcmsctrlutil -host localhost -port 24055 -status
Up since: Tue Oct 14 20:54:23 2003
Memory used by data messages: 950.729 K(Bytes)
Total messages passed through: 1900331
Total messages retained: 3555
Number of message queue(s): 9
Number of connection(s): 14
Port number: 18007
Process ID: 2780
Server state: Ready and running...
```

8.3.5 Viewing All Topics for a JMS IQ Manager

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

8.3.6 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 the contents of a message on topic T0:

```
stcmsctrlutil -ctm T0 182 -p 18007 -msgtype text
NEWCONTENTS
^Z
Message: 182 has been changed
```

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

```
stcmsctrlutil.exe -p 18007 -tmessage T0 182 -msgtype text
NEWCONTENTS
```

8.3.7 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 show **Suspended: No status**.

```
stcmsctrlutil -host localhost -port 18007 -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
```

8.3.8 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
```

8.3.9 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
```

8.3.10 Viewing All Queues for a JMS IQ Manager

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

8.3.11 Displaying Statistics for Queues

```
stcmsctrlutil -host localhost -port 24055 -queuestat 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
```

8.3.12 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
```

8.3.13 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
```

8.3.14 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
Last imported sequence number = 491
Last imported sequence number = 497
```

8.3.15 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
Last imported sequence number = 500
Import failed
Import failed on sequence number: 500
```

This example will fail because 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.

8.3.16 Browsing Journaled Messages

To browse journaled messages, you use the **-journaler** flag. The **-journaler** flag receives information from the journaler instead of the JMS IQ Manager. 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 IQ Manager.

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

```
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 IQ Manager:

```
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
```

8.3.17 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 IQ Manager.

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
```

8.3.18 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
```

8.3.19 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 JMS IQ Manager 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 -timeout 15
```

If the **-timeout** flag is not used in subsequent commands, the default timeout is used.

Index

C

- configuration parameters
 - DB Settings
 - DBMaxSegments 79
 - DBMinSegments 79
 - DBSegmentSize 77
 - DB Sync
 - DBSync 80
 - Message Settings
 - MaxTimeToLive 81
 - Server Settings
 - Journaling 81
 - JournalPath 82
 - JournalTimeToLive 82
 - ServerMaxMessages 83
 - TimeDependency
 - TimeDependency 86
 - Topic Settings
 - TopicMaxMessages 83
 - TopicMaxMessagesPad 84
 - Trace Settings
 - TraceLevel 97
 - TraceTimestamp 86
- conventions, text 10

D

- DBMaxSegments 79
- DBMinSegments 79
- DBSync 80
- document purpose and scope 9

J

- JMS client
 - reconfiguring 66
- Journaling 81
- JournalPath 82
- JournalTimeToLive 82

L

- Logical Host
 - system JMS 73

M

- MaxTimeToLive 81

O

- organization of information, document 9

R

- reconfiguring
 - JMS client 66

S

- screenshots 11
- ServerMaxMessages 83
- stcmsctrlutil
 - syntax 106
- subtree properties 90, 93, 96
- supporting documents 11
- system JMS 73

T

- text conventions 10
- TimeDependency 86
- TopicMaxMessages 83
- TopicMaxMessagesPad 84
- TraceLevel 97
- TraceTimeStamp 86