Oracle SeeBeyond JMS Intelligent Queue User's Guide

Release 5.0.5 for Schema Run-time Environment (SRE)

Version 20100728161821.

Contents

Chapter 4

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces you to this guide, its general purpose and scope, and its organization. It also provides sources of related documentation and information.

This guide explains how to use the e*Gate Integrator implementation of Java™ Message Service (JMS™) to provide intelligent queue (IQ™) capabilities. This explanation includes:

- General overview of the feature and its prerequisites.
- Configuration of the Oracle SeeBeyond JMS IQ Manager, e*Way Connection, and IQ components that constitute the implementation.

This document does not contain information about installation or general e*Gate configuration. For information about these and related topics, see **"Supporting Documents" on page 12**.

The reader of this guide is presumed to be a Java developer or system administrator with responsibility for maintaining the e*Gate system. You must be familiar with Java and the Java Message Service API.

1.1 Contents of This Guide

This guide contains the following information:

- **Chapter 2, “Oracle SeeBeyond JMS IQ Service Overview” on page 13** provides an introduction to the Oracle SeeBeyond JMS IQ service, as well as a summary of its use, operation, and requirements.
- **Chapter 3, “Creating and Configuring JMS IQ Managers and IQs” on page 16** describes how JMS IQ Managers work, including FIFO and time-based Event delivery methods. This chapter also includes procedures for creating and configuring JMS IQ Managers and JMS IQs.
- **Chapter 4, “Configuring JMS e*Way Connections” on page 40** describes how you configure e*Way Connections to work with Oracle SeeBeyond JMS IQs.
- **Chapter 5, “Managing Events and JMS IQ Managers” on page 48** describes how you configure Oracle SeeBeyond JMS IQ Managers and JMS IQs.
- **Chapter 6 “Using the STC Message Server Control Utility” on page 68** describes the **stcms*.exe** command-line utilities that you can use to manage Oracle SeeBeyond JMS IQs.

1.2 Writing Conventions

The writing conventions listed in this section are observed throughout this document.

Hypertext Links

When you are using this guide online, cross-references are also hypertext links and appear in **blue text** as shown below. Click the **blue text** to jump to the section.

> For information on these and related topics, see **“Parameter, Function, and Command Names” on page 11**.

Command Line

Text to be typed at the command line is displayed in a special font as shown below.

```
java -jar ValidationBuilder.jar
```

Variables within a command line are set in the same font and bold italic as shown below.

```
stcregutil -rh host-name -rs schema-name -un user-name
-up password -ef output-directory
```

Code and Samples

Computer code and samples (including printouts) on a separate line or lines are set in Courier as shown below.

```
Configuration for BOB_Promotion
```

However, when these elements (or portions of them) or variables representing several possible elements appear within ordinary text, they are set in *italics* as shown below.

> *path* and *file-name* are the path and file name specified as arguments to **-fr** in the **stcregutil** command line.

Notes and Cautions

Points of particular interest or significance to the reader are introduced with ***Note***, ***Caution***, or ***Important***, and the text is displayed in *italics*, for example:

Note: *The Actions menu is only available when a Properties window is displayed.*

User Input

The names of items in the user interface such as icons or buttons that you click or select appear in **bold** as shown below.

> Click **Apply** to save, or **OK** to save and close.

File Names and Paths

When names of files are given in the text, they appear in **bold** as shown below.

> Use a text editor to open the **ValidationBuilder.properties** file.

When file paths and drive designations are used, with or without the file name, they appear in **bold** as shown below.

> In the **Open** field, type **D:\setup\setup.exe** where **D:** is your CD-ROM drive.

Parameter, Function, and Command Names

When names of parameters, functions, and commands are given in the body of the text, they appear in **bold** as follows:

> The default parameter **localhost** is normally only used for testing.
>
> The Monk function **iq-put** places an Event into an IQ.
>
> You can use the **stccb** utility to start the Control Broker.

1.3 Supporting Documents

The following documents provide additional information about other features or products mentioned in this guide:

- *Creating an End-to-end Scenario with e*Gate Integrator*
- *e*Gate API Kit Developer's Guide*
- *e*Gate Integrator Alert and Log File Reference Guide*
- *e*Gate Integrator Collaboration Services Reference Guide*
- *e*Gate Integrator Installation Guide*
- *e*Gate Integrator Intelligent Queue Services Reference Guide*
- *e*Gate Integrator System Administration and Operations Guide*
- *e*Gate Integrator User's Guide*
- *Monk Developer's Reference*
- *Oracle eBusiness Integration Suite Primer*
- *Oracle eBusiness Integration Suite Deployment Guide*
- *Standard e*Way Intelligent Adapter User's Guide*

See the *Oracle eBusiness Integration Suite Primer* for a complete list of eBI Suite-related documentation. You can also refer to the appropriate Microsoft Windows or UNIX documents, if necessary.

Note: *For information about how to use a specific add-on product (for example, an e*Way Intelligent Adapter), see the user's guide for that product.*

Chapter 2

Oracle SeeBeyond JMS IQ Service Overview

This chapter provides a general overview of the Oracle SeeBeyond JMS IQ Service: how it operates, how it compares with the Oracle SeeBeyond Standard IQ Services, and system requirements.

2.1 JMS and e*Gate: Introduction

The Java Message Service (JMS) specification has become an industry standard for enterprise messaging systems.

Note: *The e*Gate term **Event** is interchangeable with the JMS term **message**.*

The Oracle SeeBeyond JMS IQ Service, which is compliant with JMS version 1.0.2, provides all the standard capabilities of Oracle SeeBeyond Intelligent Queues to existing and new Collaborations written in Monk, C, or Java, including:

- Persistent nonvolatile storage of Events (messages) and message (Event) routing.
- Guaranteed delivery.
- API access from Monk, C, or Java.

In other words: Every e*Gate component that publishes or subscribes to an IQ managed by an Oracle SeeBeyond Standard IQ service can publish or subscribe in exactly the same way to exactly the same IQ managed by the Oracle SeeBeyond JMS IQ Service.

However, in addition to functioning as a normal IQ implementation, the Oracle SeeBeyond JMS IQ Service offers the following additional benefits:

- Significantly faster performance and better scalability.
- Support for standard JMS message types.
- Publication and subscription of Events by external Java applications using the industry-standard open API (when used with the e*Gate API Kit for JMS).
- Run-time subscribers.
- XA compliance and support for Guaranteed Exactly Once Delivery of events.

- Increased flexibility. For example, a Oracle SeeBeyond JMS IQ Service can function as a synchronous communication channel for two or more external systems without using any IQs at all, or it can even function as a kind of bridge between two or more schemas.

The following architectural diagrams show the similarities and differences between Oracle SeeBeyond Standard IQ Services and Oracle SeeBeyond JMS IQ Services.

Figure 1 Architecture Using Standard IQ Manager

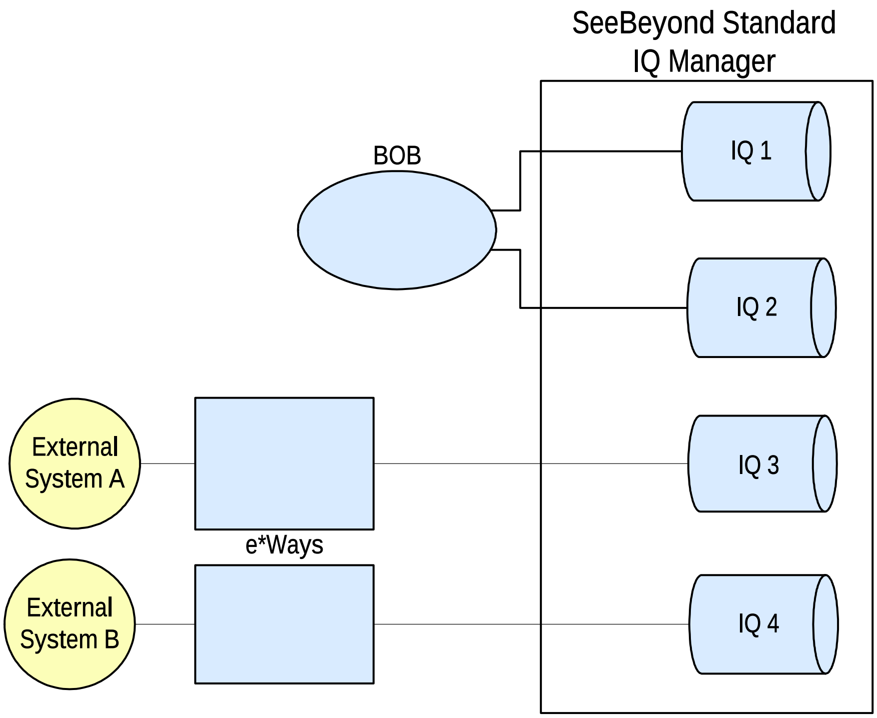
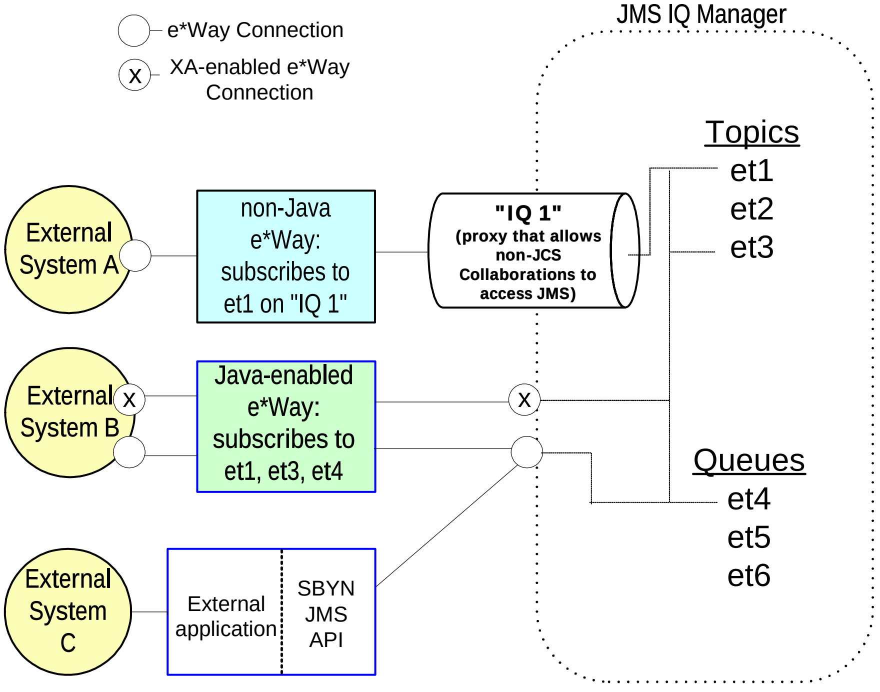


Figure 2 Architecture Using Oracle SeeBeyond JMS IQ Manager


e*Way Connection
X
XA-enabled e*Way Connection
JMS IQ Manager
Topics
et1
et2
et3
External System A
non-Java e*Way: subscribes to et1 on "IQ 1"
"IQ 1"
(proxy that allows non-JCS Collaborations to access JMS)
External System B
X
Java-enabled e*Way: subscribes to et1, et3, et4
X
Queues
et4
et5
et6
External System C
External application
SBYN JMS API

2.2 For Further Information

The *e*Gate API Kit Developer's Guide* contains complete information about architecture and APIs, as well as in-depth information about such special topics as the Java and COM+ Clients, and the Compensating Resource Manager (CRM). You can also find details on the various messaging models (publish/subscribe, point-to-point, and request/reply), sample code, and a sample schema implementation.

Chapter 3

Creating and Configuring JMS IQ Managers and IQs

This chapter describes how to set up JMS IQ Managers and IQs. This guide also refers to the JMS IQ Manager as the STC Message Server or **stcms**.

Note: *The JMS term "topic" is used interchangeably with the e*Gate term "Event Type"; the JMS term "message" is used interchangeably with the e*Gate term "Event"; and the term "server" is used generically for "Oracle SeeBeyond JMS IQ Manager."*

In This Chapter:

3.1 How JMS IQ Managers Work

On startup, the server performs the following sequence of operations:

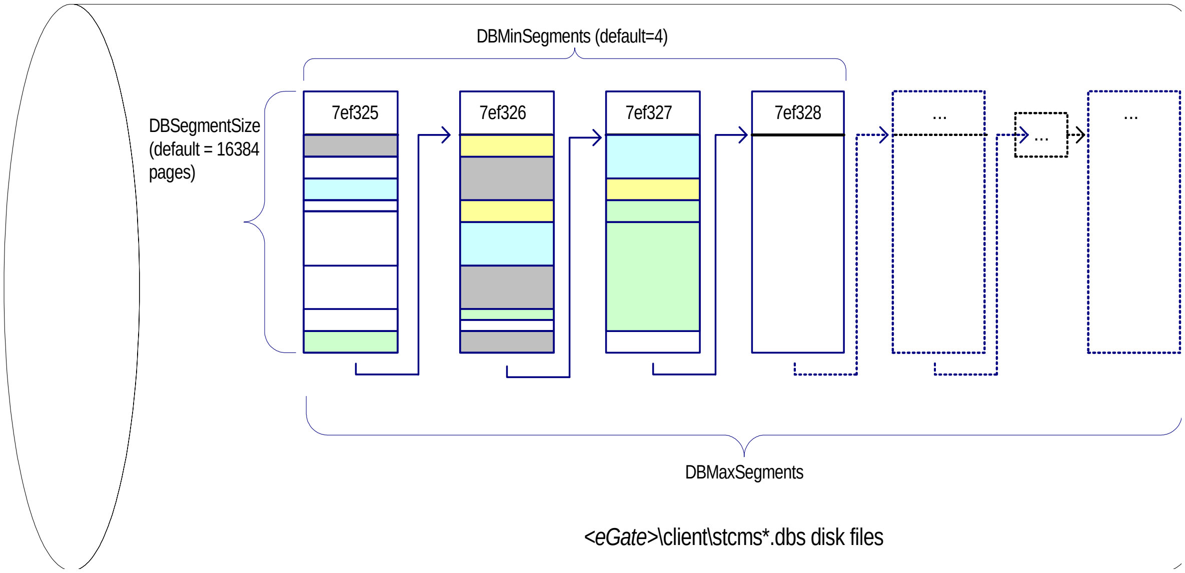
1. It allocates sufficient disk space to hold the minimum number of segments.

   Figure 3 shows a server allocation of four segments, numbered 73f325 through 73f328.

2. As Events arrive, they are appended to the first segment until the segment is full.
3. When a segment is full, the server puts subsequent Events in the first free segment.

   Figure 3 shows that the third segment, 7ef327 in file **stcms73f327.dbs**, is almost full.

Figure 3 Oracle SeeBeyond JMS IQ Manager Operation


DBMinSegments (default=4)
DBSegmentSize (default = 16384 pages)
7ef325
7ef326
7ef327
7ef328
...
...
...
DBMaxSegments
<eGate>\client\stcms*.dbs disk files

4 If there is no free segment, the server allocates a new segment if possible.

5 When an Event has been delivered to its last subscriber:

- For a non-transactional Event, the Event is dequeued immediately and the slot it occupied is marked eligible; but
- For a transactional (Internal or XA-compliant) Event, the Event is dequeued (and its slot marked eligible) only after all subscribers have committed it.

6 When all Events in a segment have expired or been dequeued, the server cleans up the segment, freeing it for re-use.

In Figure 3, the first segment (7ef325) is almost ready for cleanup.

3.1.1 About Segments

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, and so their default file names have the form **stcms*.dbs**. By default, segment files reside in the **..\stcms** folder of **<*eGate*>\client\bin\** (typically, **C:\eGate\client\stcms\**).

The default segment size is 16384 pages (in Windows a page is 512 bytes; in UNIX it is 1024 bytes). The default number of segments allocated initially is 4.

To configure segment defaults, edit the properties of the Oracle SeeBeyond JMS IQ Manager, open its configuration file, and access its DB Settings section; see **“DB Settings” on page 26**.

3.1.2 About Performance, Memory, and Disk Space

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

Because every Event 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 Events in server memory continuously until the corresponding segment is cleaned up.

Because available server memory can easily be exceeded for systems handling very large Events with long dwell times, there are several configuration parameters to help you manage a memory-bound server; see **"Managing Resources" on page 18**.

General Tips

- Segment size (in bytes: 512 bytes/page for Windows, 1024 for UNIX) must always be larger than the largest expected Event—preferably by an order of magnitude.

Tips for Maximizing Performance

In decreasing order of importance:

- Use the fastest disk possible.
- Keep as many Events in server memory as possible.
- Use the largest disk possible.

Site-Specific Considerations

- Allocating a new segment requires more time than freeing a cleaned-up segment.
- Smaller segments turn over more rapidly and thus provide more effective use of server memory. However, since 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.

3.1.3 Managing Resources

As for any server, there are configuration parameters purely for managing disk access memory (such as MaxPayloadMemory and PayloadMemoryPad) and memory/disk trade-offs (such as DBCacheSize for controlling read memory, DBSync on UNIX for increasing speed over reliability, and DBCacheIntoRAM on Windows for reducing page faults). In addition, for the Oracle SeeBeyond JMS IQ Manager there are special configuration parameters that specifically deal with Events, Event Types, and publishers. Three of these are discussed in detail below.

Throttling Publishers

Note: *The JMS term "topic" is used interchangeably with the e*Gate term "Event Type"; the JMS term "message" is used interchangeably with the e*Gate term "Event"; and the term "server" is used generically for "Oracle SeeBeyond JMS IQ Manager."*

When the amount of server memory allocated to messages reaches a certain limit, the server 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 publisher.

Publisher throttling is done on a per-topic basis. This caters to the most common reasons for approaching server memory limit in an otherwise well-tuned system:

- A particular topic (Event Type) has a period of abnormally heavy traffic. Throttling all publishers of such a topic gives the topic's subscribers a chance to catch up while maintaining normal throughput for other topics.
- A particular subscriber falters or fails, causing a backup of all topics it subscribes to. If the subscriber problem is transient and normal, then throttling all its publishers gives it "breathing room" to catch up on the backlog. And, if the subscriber problem is serious, then throttling its publishers allows unaffected topics to flow freely while the problem can be diagnosed and repaired without taking the system off-line.

Three configuration parameters govern publisher throttling:

- **ServerMaxMessages** sets the server limit. When the system is below this threshold, it does not throttle any publishers even if they are eligible for throttling.
- When publisher throttling is in effect, **TopicMaxMessages** sets the per-topic limit. The system stops reading messages from publishers of any topic that has exceeded this limit. It resumes reading messages on the topic only when one or both of the following criteria are met:
    - The server falls below the ServerMaxMessages threshold; or
    - The topic has so few messages that the system can stop throttling its publishers.
- **TopicMaxMessagesPad** determines how many messages on this topic must be dequeued before throttling can stop.

Note: *Each message in a topic counts against the topic's **TopicMaxMessages** 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 subscribers.*

Note: *When a BOB sends messages to a JMS IQ faster than the BOB receives messages from the same JMS IQ, a deadlock occurs. To avoid this situation, do not use the JMS IQ for long term storage and change the **ServerMaxMessages** setting.*

Example of Publisher Throttling and Unthrottling

Table 1 illustrates a scenario where a server becomes stressed and starts to compensate by using publisher throttling. (The server uses default values for throttling parameters: **ServerMaxMessages**=100000; **TopicMaxMessages**=1000; **TopicMaxMessagesPad**=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 server is no longer stressed, the same topic is allowed build up an even greater backlog without having its publisher throttled.

Table 1 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 stressed.
11:38	100307	800	500	150	Server is stressed, but Topic_A is unaffected—its subscribers are keeping up well enough.
11:39	101283	1100	800	225	Server still stressed, 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 stressed, Pub1 will stay throttled until the number of undelivered messages falls below 900.
11:41	104031	1350	1300	375	Pub1 is throttled; Sub1 is nearly caught up; Sub2 is catching up, but has 975 undelivered messages.
11:42	103204	1350	1350	449	Pub1 is throttled; Sub1 has caught up; Sub2 has 901 undelivered messages — still too many.
11:43	102762	1350	1350	451	Although server is still stressed, it unthrottles Pub1 now that the undelivered message count for Topic_A has fallen below 900.
11:44	101095	1375	1370	525	Server is stressed, but Topic_A is unaffected — it has only 850 undelivered messages.
11:45	100028	1575	1500	600	Server is stressed, but Topic_A is unaffected — it has only 975 undelivered messages.
11:46	99248	1900	1700	675	Server is no longer stressed; no publishers are throttled even though Sub2 has more than 1000 undelivered messages.

3.2 Managing Event Delivery Sequence

Depending on your schema design, e*Gate applies three methods of Event delivery for schemas that use JMS IQs:

- First in, first out (FIFO)
- Non-FIFO
- Time-based sequencing

FIFO and non-FIFO

FIFO and non-FIFO are default event delivery methods; which delivery method occurs depends on the schema design. With FIFO, Events are delivered in order of enqueue time. FIFO occurs for Collaborations that subscribe to the same Event Type published under a single IQ Manager. Otherwise, non-FIFO occurs. In non-FIFO, subscribing Collaborations retrieve Events in round-robin fashion from each Event Type/publisher combination to which they subscribe. The IQ Manager then processes the oldest unread Event for each Event Type/publisher combination. Events are not processed based on enqueue time across publishers; only per publisher.

Time-based Sequencing

If your business environment requires a specific Event delivery order for multiple Event Types published under a single IQ Manager, you can specify a time-based sequence as described in **"Specifying Time-based Event Delivery" on page 22**. To set a time-based Event delivery order, you specify a list of Event Types. For each Event Type specified, no Events of that Event Type are delivered to subscribing Collaborations if there is an Event with an older enqueue time.

Note About Mixing Standard and JMS IQs

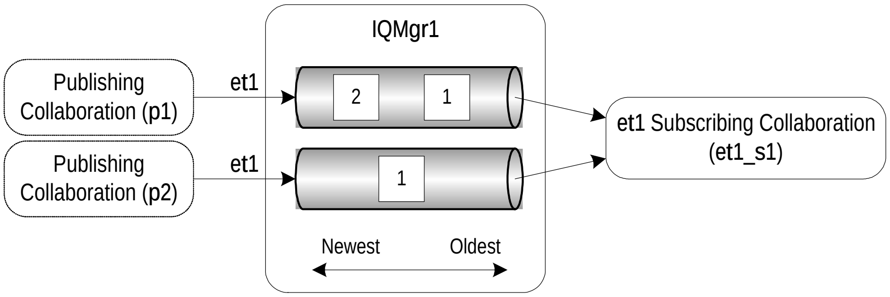
If you have a schema with both JMS and Standard IQs, the delivery method of Events applies only for the same type of IQ, not across IQs types. For example, if your Standard IQ has FIFO enabled, and the schema allows for FIFO within the JMS IQs, FIFO occurs within the JMS IQs and within the Standard IQs. However, FIFO does not occur across the Events in the Standard *and* the JMS IQs.

3.2.1 FIFO Event Delivery Order

With FIFO Event delivery, Events are delivered in order of enqueue time. FIFO occurs for Collaborations that subscribe to the same Event Type published by Collaborations under a single IQ Manager.

The figure below shows an example of the Event delivery order in a schema where FIFO applies. In this schema, the Collaborations **p1** and **p2** publish the same Event Type, **et1**, to a single IQ Manager. A single Collaboration subscribes to Event Type et1. The Events shown in the IQs arrived in right-to-left sequence.

Figure 4 FIFO Event Delivery Order


IQMgr1
Publishing Collaboration (p1)
et1
2
1
Publishing Collaboration (p2)
et1
1
et1 Subscribing Collaboration (et1_s1)
Newest
Oldest

The list below refers to the Events as ***pub-n***, where *pub* is the publishing Collaboration and *n* is the Event's sequence number, for example **p1-1**. In this schema, Events are processed as follows:

- **et1_s1** asks the IQ Manager for the oldest Event of Event Type **et1**.
- **IQMgr1** returns **p1-1**
- **et1_s1** asks the IQ Manager for the oldest Event of Event Type **et1**.
- **IQMgr1** returns **p2-1**
- **et1_s1** asks the IQ Manager for the oldest Event of Event Type **et1**.
- **IQMgr1** returns **p1-2**

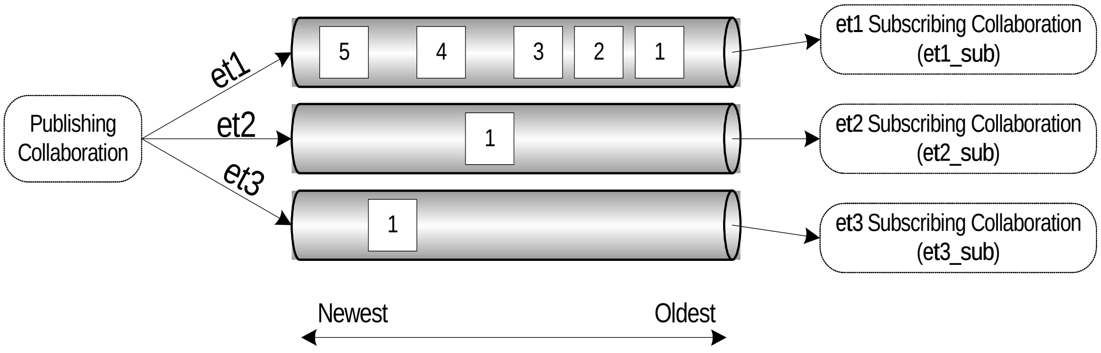
3.2.2 Specifying Time-based Event Delivery

By default, depending on your schema design, event delivery is either FIFO or non-FIFO. Depending on your business environment, this may create a situation where related Events are processed in an inappropriate sequence. For example, a "Cancel Admit Patient" Event might be processed before the "Admit Patient" Event has been processed if these are two Events of different Event Types, or if the Events are not published to the same IQ Manager. To avoid such a situation, you can specify a time-based processing sequence with the TimeDependency setting as described in **"TimeDependency Settings" on page 34**.

With the TimeDependency setting, you specify a list of Event Types. For each Event Type specified, no Events of that Event Type are delivered to subscribing Collaborations if there is an Event with an older commit time.

The figure below shows an application of time-based event processing. This schema has three event types, **et1**, **et2**, and **et3**, as members of the time ordering list. The Events shown arrived in right to left sequence, that is, three Events of type **et1**, then one **et2** Event, and so on.

Figure 5 Time-based Event Processing


Publishing Collaboration
et1
et2
et3
5
4
3
2
1
1
1
et1 Subscribing Collaboration (et1_sub)
et2 Subscribing Collaboration (et2_sub)
et3 Subscribing Collaboration (et3_sub)
Newest
Oldest

When you specify a time-based sequence for **et1**, **et2**, and **et3**, the Events are processed as shown below. This section refers to the Events as **et1-*n***, where *n* is the Event's sequence number, for example **et1-1**.

- **et2_sub** does not receive **et2-1** until **et1_sub** commits **et1-3**. If **et2_sub** requests an Event before this point, the IQ Manager replies *"No message available."*
- **et1_sub** does not receive **et1-4** until **et2_sub** commits **et2-1**. If **et1_sub** requests an Event before this point, the IQ Manager replies *"No message available."*

- **et3_sub** does not receive **et3-1** until **et1_sub** commits **et1-4**. If **et3_sub** requests an Event before this point, the IQ Manager replies *"No message available."*
- **et1_sub** does not receive **et1-5** until **et3_sub** commits **et3-1**. If **et1_sub** requests an Event before this point, the IQ Manager replies *"No message available."*

Important Notes About Time-based Sequencing

Use time-based sequencing with caution; it could potentially lead to significant waiting periods. For example, consider the situation above where the e*Way containing ***et1_sub*** is shut down immediately after ***et1_sub*** commits ***et1-1***. In this case, ***et2_sub*** and ***et3_sub*** is blocked until the e*Way containing ***et1_sub*** is brought back up and commits ***et1-2*** and ***et1-3***.

If you are using time-based sequencing, it is important that all system times are in sync. For example, if system A in New York is at 9:46 PM, system B in Los Angeles needs to be at 6:46 PM; in GMT, the systems are then in sync. If a system shuts down, make sure that the system time is correct at startup.

Adding Time-based Sequencing Information to Log Files

To have the JMS Administrator log file include time-based sequencing related information, set the trace settings as follows:

- Set the **TraceLevel** option to **0**
- Set the **TraceVerbose** option to **Yes**

For details about setting trace levels, refer to **"Trace Settings" on page 32**.

3.3 Creating JMS IQ Managers

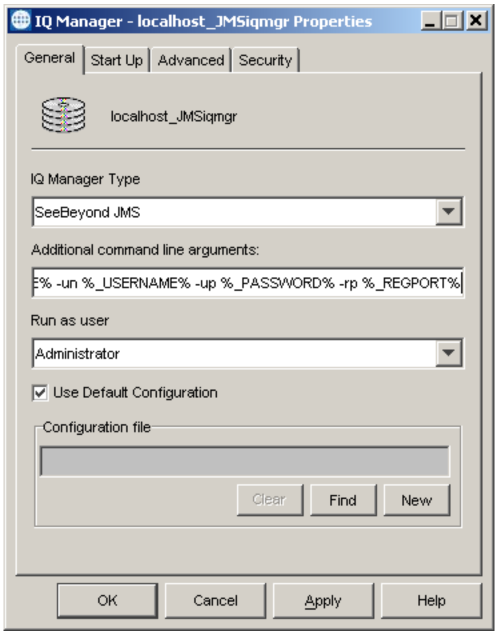
The procedure below describes how you create JMS IQ Managers.

To use JMS IQ Managers successfully, do not edit the **.egate.store** so that the SystemData and SharedEx variables point to different directories. Doing so causes the JMS IQ Managers to malfunction during startup.

To create JMS IQ Managers

1. In the **Components** tab of the e*Gate Schema Designer, expand the **Participating Hosts** folder, and click the Control Broker where you want to create the new IQ Manager (***hostname*_cb**).
2. Click **Create a New IQ Manager**.
3. In the **Name** box, type the name of the new IQ Manager and click **OK**.
4. In the **Component** tab, double-click the new IQ Manager.

The **IQ Manager Properties** dialog box appears.



IQ Manager Type: SeeBeyond JMS

Additional command line arguments: F% -un %_USERNAME% -up %_PASSWORD% -rp %_REGPORT%

Run as user: Administrator

☑ Use Default Configuration

Configuration file:

5. To use a configuration file other than the default configuration, click **New**.

   The **Edit Settings** dialog box appears. Set the configuration parameters for this configuration file. For information about the configuration parameters, refer to **"How JMS IQ Managers Work" on page 16**. You may want to add user notes to describe non-standard settings.

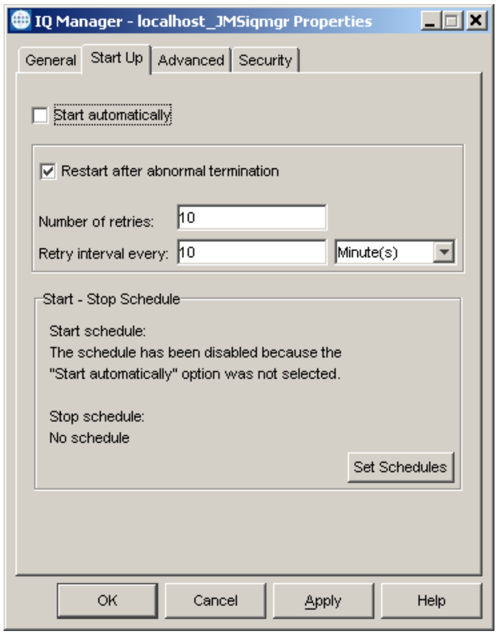
6. On the **File** menu, click **Save**.

7. In the **Save As** dialog box, enter the configuration file name and click **Save**.

8. On the **File** menu, click **Close** to close the **Edit Settings** dialog box.

9. If you want to update message selectors with new values for the existing durable subscribers, then add the **-cs** flag to the string in the **Additional command line arguments** field.

10 In the **IQ Properties** dialog box, click the **Start Up** tab.



☐ Start automatically

☑ Restart after abnormal termination

Number of retries: 10

Retry interval every: 10 Minute(s)

Start - Stop Schedule

Start schedule:
The schedule has been disabled because the "Start automatically" option was not selected.

Stop schedule:
No schedule

11 To have this IQ Manager stop and start on a schedule you define, select **Start automatically**.

12 Clear **Restart after abnormal termination** if you want the IQ Manager to require manual restart after an abnormal halt.

By default, the JMS IQ Manager restarts automatically after it halts abnormally unless you clear this check box.

13 If the IQ Manager is set to restart automatically, select the number of times to attempt a restart, and the interval between attempts. The default is to attempt 10 retries at 10-minute intervals.

14 To define the schedule for automatic startup, click **Set Schedules**, enter the schedule information, and click **OK**.

15 To change TCP/IP port number through which the IQ Manager communicates with the Control Broker, click the **Advanced** tab, and enter the number in the **TCP/IP port number** box.

This value is set during installation; usually you do not need to change it. If your schema has multiple IQ Managers, each IQ Manager must have a unique TCP/IP port number.

16 To view or set privilege assignments, click the **Security** tab, click **Privilege**, and add roles and assign privileges.

For more information about privileges refer to the *e*Gate Integrator System Administration and Operations Guide*.

17 Click **OK** to close the **IQ Manager Properties** dialog box.

3.4 Configuring JMS IQ Managers

After you created a JMS IQ Manager as described in **"Creating JMS IQ Managers" on page 23**, you can specify configuration settings for the IQ Manager.

JMS IQ Managers have the following configuration options:

To configure a JMS IQ Manager

1. In the **Components** tab of the e*Gate Schema Designer, expand the **Participating Hosts** folder, and click the Control Broker for which you want to configure a JMS IQ Manager (***hostname*_cb**).
2. Double-click the JMS IQ Manager (***hostname*_igmgr**) to display the IQ Manager Properties dialog box.
3. Click **Edit**.

   The **Edit Settings** dialog box appears.
4. In the **GoTo Section** list, click one of the options below. For information about each of these settings, refer to the page number listed.
   - **DB Settings** on page 26
   - **Message Settings** on page 29
   - **Server Settings** on page 31
   - **Topic Settings** on page 32
   - **Trace Settings** on page 32
   - **TimeDependency Settings** on page 34
5. In the **GoTo Parameter** list, click the parameter you want to configure.
6. When you have configured the parameters, click **Save** on the **File** menu.

3.4.1 DB Settings

The **DB Settings** parameters govern the persistent message store, transaction-log files, and the disk write memory cache.

DBPath

Preset to: (blank).

Unless other values have been changed, a blank setting defaults to:
<*eGate*>`\Client\stcms\<`*`IQManagerUID`*`>` (see discussion below).

Description

Specifies the directory in which the persistent message store and transaction-log files reside.

Required Values

A string (if specified at all).

If no value is specified, the system uses the value of the **MessageServiceData** variable in the file **.egate.store** as the base data directory. It then creates a subdirectory under that directory, named with the unique ID (UID) of the JMS IQ Manager, to store the data files. This guarantees that no two JMS IQ Managers will accidentally use the same base data directory for holding their **.dbs** files.

- On Windows systems, the **.egate.store** file normally contains the line:

  ```
  MessageServiceData=C:\EGATE\Client\stcms
  ```

  If you keep all defaults, the persistent store files will reside in:

  ```
  C:\EGATE\Client\stcms\<IQManagerUID>
  ```

  (where <*IQManagerUID*> is replaced by the actual UID of the IQ Manager)

- On UNIX systems, the **.egate.store** file normally contains the line:

  ```
  MessageServiceData=/usr/egate/client/stcms
  ```

  If you keep all defaults, the persistent store files will reside in:

  ```
  /usr/egate/client/stcms/<IQManagerUID>
  ```

(where <*IQManagerUID*> is replaced by the actual UID of the IQ Manager)

If specified, DBPath overrides the **.egate.store** setting for the **MessageServiceData** variable:

- If DBPath is specified as a *relative* path, data files are stored relative to the directory containing the **stcms.exe** executable; this is usually <*eGate*>**\client\bin\**.
- If DBPath is specified as a *fully qualified* path, data files are stored in the directory specified.

If you choose to define DBPath for multiple IQ Managers, specify a different path for each JMS IQ Manager to avoid conflict. If you cannot specify a different path for each, then you must specify different DBSuffix settings for each JMS IQ Manager sharing the same DBPath.

DBSuffix

Description

Specifies the characters to use as a file extension for the file names of the persistent message store and transaction-log files.

Required Values

A string. The default is **dbs**

Note: *If you keep all defaults, the persistent store files will have the form C:\eGate\client\stcms\<stcmsIQManagerUID>\stcms#####.**dbs**.*

DBCacheSize

Description

Specifies the total number of pages in the read cache. This is the number of bytes read from the disk each time the JMS IQ Manager refreshes its message memory.

Required Values

An integer between 1 and 999999999. The default is 1024. (A page is 512 bytes on Windows, 1024 bytes on UNIX.) Range: **1 - DBSegmentSize**

DBSegmentSize

Description

Specifies the total number of pages in a single DB file. (A page is 512 bytes on Windows, 1024 bytes on UNIX.)

Required Values

An integer between 1 and 999999999. The default is 16384.

Limits:
Set this to at least **(2 * the total number of anticipated durable subscribers).**

Note: *__DBSegmentSize__ must be set greater than the size of the largest anticipated message. Allow a generous margin of error. Larger __DBSegmentSize__ is advantageous in that cleanup runs less often; but the cleanup does run longer. Smaller __DBSegmentSize__ results in more efficient use of the disk, but if the message is larger than the segment size set, the server exits.*

DBMinSegments

Description

Specifies the minimum number of files, of size **DBSegmentSize**, initially created and maintained by the server for its persistent message store and transaction log. When the minimum is exceeded, the server allocates additional segments on an as-needed basis, up to the maximum set by **DBMaxSegments**.

Required Values

An integer between 1 and 99999. The default is 4.

Limits:
Must be **>= 1**, up to **99999.**

DBMaxSegments

The special value **0** means "no limit"; this causes the server to create new files as needed, limited only by available disk space.

Description

Specifies the maximum total number of files that the server creates and maintains for its persistent message store and transaction log. This effectively limits the amount of disk space that the server uses. If the server needs to write data that would exceed this limit, it exits gracefully and outputs an appropriate error message to the trace log.

The Oracle SeeBeyond JMS IQ Manager should not be used as a semi-permanent storage medium without sufficient memory and disk resources. To control the memory and disk resources needed by the server, use the publisher throttling feature, controlled by the **ServerMaxMessages**, **TopicMaxMessages,** and **TopicMaxMessagesPad** settings. For details, see **"Managing Resources" on page 18**.

Required Values

An integer between 0 and 99999. The default is **0**.

DBLockCacheIntoRAM

Description

The Windows VirtualLock API function locks the server disk cache into physical memory, ensuring that subsequent access to the region will not incur a page fault (a swap-out to disk). This variable can be used in conjunction with **DBCacheSize** to improve performance.

Note: *Only used in Windows; the administrator privilege is required.*

DBSync

Description

True (the default) specifies that Message Service controls the cache synchronization to disk. False specifies that the operating system controls the synchronization schedule.

3.4.2 Message Settings

The **Message Settings** parameters govern message data memory and expiration settings on the server.

MaxPayloadMemory

Description

Specifies the memory allotted for disk accesses, namely the maximum amount (in kilobytes) of message data payloads allowed to be in physical memory at any moment while the server is running. This represents the cache of message data that the server uses to fulfill read requests. When message data memory usage increases beyond the MaxPayloadMemory threshold, the server begins memory garbage collection and recovery.

Required Values

An integer between **1** and **999999999**. The default is **20000**. The upper limit depends on available memory resources.

PayloadMemoryPad

Description

When **MaxPayloadMemory** (see above—the threshold beyond which the server begins memory recovery and cleanup) is exceeded, the server attempts to recover the exceeded memory plus a small amount more; the extra amount (in KB) is specified by the parameter **PayloadMemoryPad**.

Required Values

An integer between **0** and **999999**. The default is **100**. The upper limit must be less than **MaxPayloadMemory**.

MaxTimeToLive

Description

Specifies the maximum amount of time (in seconds) before a live message expires. After it expires, the message is removed from the IQ whether it has been consumed or not.

Required Values

An integer between **0** and **999999999**. The default is **2592000** (in other words, 30*24*60*60 seconds = 30 days). If you specify **0**, the message never expires.

EnableEdit

Description

Turns on/off the ability to edit message contents.

Required Values

Yes or **No**.

EnableView

Description

Turns on/off the ability to view message contents.

Required Values

Yes or **No**.

EnableDelete

Description

Turns on/off the ability to delete messages.

Required Values

Yes or **No**.

3.4.3 Server Settings

The **Server Settings** parameter sets the upper limit on the total number of messages that the server tracks before throttling publishers.

ServerMaxMessages

Description

Specifies the maximum number of total messages for all topics combined that the server permits before it starts throttling publishers. For a detailed explanation and an example, see **"Throttling Publishers" on page 18**.

Required Values

An integer between **0** and **999999999**. The default is **100000**. If you specify **0**, publishers are never throttled.

Journaling

Description

Specifies whether journaling is enabled. For more information, refer to **"Enabling Journaling" on page 64**.

Journaling options can also be specified for a specific Collaboration. Collaboration journaling settings override JMS server settings for journaling. For more information, refer to **"Enabling Journaling for a Collaboration" on page 65**.

When you enable journaling, it is recommended you archive daily to avoid the journal database becoming exceedingly large. Archiving removes journal expired Events from the journal database. You use the STC MS Control utility to archive. For more information, refer to **"Backing Up" on page 76**.

JournalTimeToLive

Description

Specifies the maximum amount of time in seconds for a journaled message to live before it expires and is stored in the archive database. For more information, refer to **"Enabling Journaling" on page 64**.

Journaling options can also be specified for a specific Collaboration. Collaboration journaling setting override JMS server settings for journaling. For more information, refer to **"Enabling Journaling for a Collaboration" on page 65**.

Required Values

An integer between **0** and **999999999**. The default is **86400** (one day in seconds). If you specify **0**, journaled messages expire immediately.

JournalPath

Description

Specifies the location where the journal files are stored. You can enter an absolute path or a path relative to the location of the message service executable. For more information, refer to **"Enabling Journaling" on page 64**.

Required Values

A string (if specified at all).

If no value is specified, the system creates a **Journal** folder in the e*Gate client directory, for example: c:\eGate\Client\Journal.

3.4.4 Topic Settings

The **Topic Settings** parameters set the upper limit on the total number of messages the server handles and govern per-topic traffic thresholds.

TopicMaxMessages

Description

Specifies the maximum number of messages permitted for any particular topic. When the number of messages on a topic reaches this value, all publishers of the topic are throttled. Once a publisher is throttled, the server stops reading messages from it until the number of topics in the queue it publishes to has dropped to below the threshold of (**TopicMaxMessages – TopicMaxMessages Pad**).

Required Values

An integer between **0** and **999999999**. The default is **1000**. If set to **0**, the publishers are never throttled.

TopicMaxMessagesPad

Description

Used in conjunction with **TopicMaxMessages** parameter. Specifies the number of messages that must be dequeued before publishers to the topic are unthrottled.

Required Values

An integer between **0** and **99999999**. The default is **100**. The value must be set to less than that of **TopicMaxMessages**.

3.4.5 Trace Settings

The **Trace Settings** parameters govern trace and debug logging behaviors.

TraceToFile

Description

Specifies whether informational, warning, and error messages are written to the log file.

Required Values

Yes or **No**. The default is Yes.

TraceLevel

Description

Specifies the threshold severity level at which the system issues informational, warning, and error messages.

Required Values

0, **1**, **2**, or **3**. The default is **1**.

0 Informational messages (in addition to all three categories below).

1 Warning messages (in addition to both categories below).

2 Error messages (in addition to the category below).

3 Fatal messages (only)

Note: *For maximum debugging, use the setting **0**. To increase performance, set this option to **1** or greater in production.*

TraceMemory

Description

Sets memory-level tracing on or off.

Required Values

off or **on**. The default is **off**.

TraceToStdout

Description

Specifies whether to (also) print debugging/trace information to standard output.

Required Values

Yes or **No**. The default is **No**.

TraceVerbose

Description

Specifies whether debugging/trace information will issue complete full-length messages.

Required Values

Yes or **No**. The default is **No**.

TraceTimestamp

Description

Specifies whether to print timestamps in the debugging/trace log file.

Required Values

Yes or **No**. The default is **No**.

JournalTraceLevel

Description

Specifies the threshold severity level at which the system issues informational, warning, and error messages regarding journaled Events.

Required Values

0, **1**, **2**, or **3**. The default is **1**.

0 Informational messages (in addition to all three categories below).

1 Warning messages (in addition to both categories below).

2 Error messages (in addition to the category below).

3 Fatal messages (only)

Note: *For maximum debugging, use the setting **0**. To increase performance, set this option to **1** or greater in production.*

3.4.6 TimeDependency Settings

TimeDependency

Description

Specifies time-based sequencing: the order in which to process Event Types depending on the Events' commit times—the oldest Event first, the newest Event last. By default, Collaborations process Event Types on first in, first out (FIFO) basis, but only for one Event Type within the same IQ Manager. For more information, refer to **"Specifying Time-based Event Delivery" on page 22**.

Required Values

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

where *TopicA*, *TopicB*, and *TopicN* are the logical names of JMS topics. *QueueA*, *QueueB*, and *Queue N* are the logical names of Event Types associated with JMS queues (JMS queues refer to e*Gate subscriber pooling, not IQs). You can specify any combination of topics and/or queues. For more information about topics and queues, refer to **"Managing Topics and Queues" on page 56**.

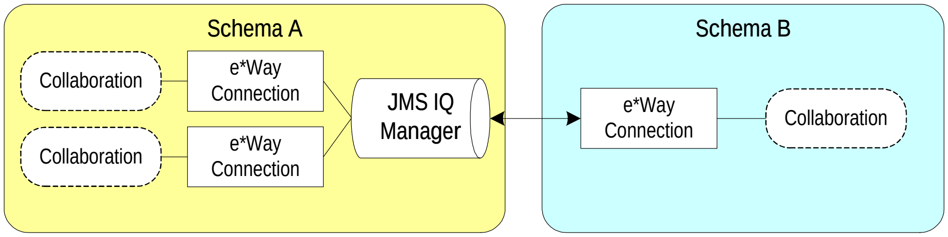
If you specify a topic or queue that does not exist, JMS Administrator enables time dependency for all other topics and queues and ignores the unknown name. This allows you to add topics and queues in your schema at a later time.

Note: *When you specify the TimeDependency, you cannot use a colon (:) or semicolon (;) as a configuration file delimiter in your schema because they are already used for the TimeDependency value.*

3.4.7 Bridging Schemas with JMS IQ Managers

Bridge schemas with JMS IQ Managers is an effective way of sharing Events between schemas. The JMS IQ Manager can reside in either schema.The schema without the JMS IQ Manager must have an e*Way Connection that contains location information for the JMS IQ Manager, such as its host and port number. The Collaborations in this schema must subscribe to Events using e*Way Connections, not JMS IQs.

The figure below shows bridging between schemas.



If the Events are topics (not queues), start the subscribing Collaborations first (schema B) before starting the publishing Collaborations in schema A when you run the schemas for the first time. Otherwise, the JMS IQ Manager does not save the Events until the subscribing Collaboration connects. After the subscribing Collaboration connects for the first time, the JMS IQ Manager registers the subscribing Collaboration. At that point you can stop and start all schema components in any order.

This is not an issue if the Events are defined as queues. You define Events are topics or queues by setting the connection type in the e*Way Connection properties.

Both schemas must be running the same version of e*Gate. For example, you cannot have schema A running e*Gate 4.5.2 and schema B running e*Gate version 4.5.3.

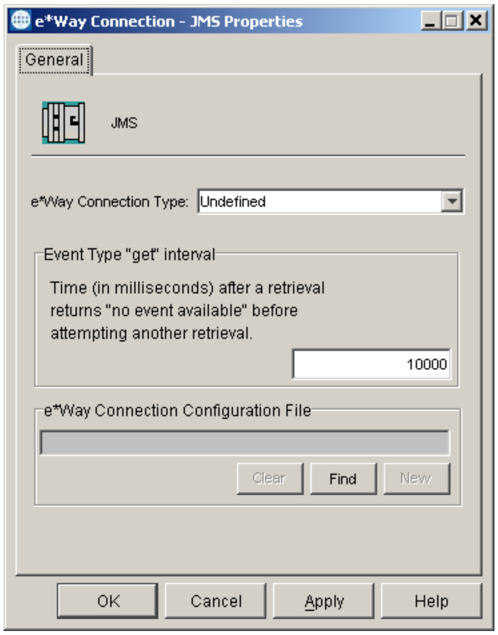
The procedure below describes how to create the e*Way Connection that specifies the JMS IQ Manager location. This e*Way Connection is created for the schema without the JMS IQ Manager.

To create an e*Way Connection that specifies the location of a bridged JMS IQ Manager

1. In the **Components** tab of the Schema Designer, click the **e*Way Connections** folder.
2. Click **Create a New e*Way Connection** .
3. In the **Name** box, type the name of the e*Way Connection and click **OK**.

4 In the Contents pane, double-click the new e*Way Connection.

The **e*Way Connection Properties** dialog box displays.



e*Way Connection Type: Undefined

Event Type "get" interval

Time (in milliseconds) after a retrieval returns "no event available" before attempting another retrieval.

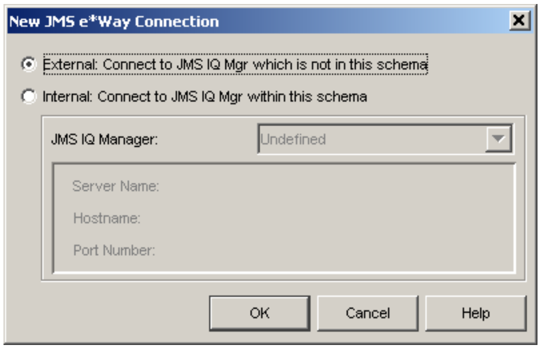
10000

e*Way Connection Configuration File

5 In the **e*Way Connection Type** list, click **SeeBeyond JMS**.

6 Click **New**.

7 Click **External: Connect to JMS IQ Mgr which is not in this schema**.



● External: Connect to JMS IQ Mgr which is not in this schema

○ Internal: Connect to JMS IQ Mgr within this schema

JMS IQ Manager: Undefined

Server Name:

Hostname:

Port Number:

8 Click **OK**.

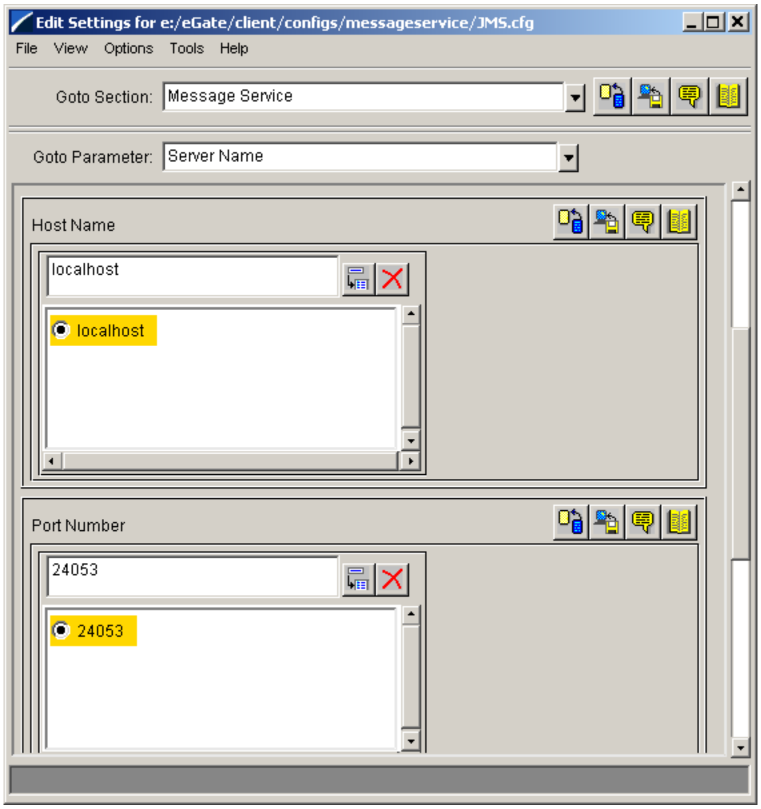
The **Edit Settings** dialog box appears.

9 In the **GoTo Section** list, click **Message Service**.

10 In the **Service Name** box, type the name of the JMS IQ Manager, and press **ENTER**.

11 In the **Host Name** box, type the name of host where the JMS IQ Manager is located and press **ENTER**.

12 In the **Port Number** box, type the port number of the host and press **ENTER**.



Edit Settings for e:/eGate/client/configs/messageservice/JMS.cfg

Goto Section: Message Service

Goto Parameter: Server Name

Host Name: localhost

● localhost

Port Number: 24053

● 24053

13 On the **File** menu, click **Save**.

3.5 Creating and Configuring JMS IQs

This section describes how to create and configure JMS IQs. JMS IQs are typically used in a schema with Monk Collaborations. For a schema with Java Collaborations, use the JMS e*Way Connection and have the JMS IQ Manager maintain queuing internally. For details, refer to **Chapter 4 “Configuring JMS e*Way Connections” on page 40**.

The following Monk functions do not support JMS IQs:

- *iq-get-header*
- *iq-mark-unusable*
- *iq-peek*

The procedure below describes how you use the e*Gate Schema Designer to create and configure JMS IQs.

To create and configure JMS IQs

1 In the **Components** tab in the e*Gate Schema Designer, expand the **Participating Host** folder, and click the JMS IQ Manager.

2 Click **Create a New IQ** .

3. In the **Name** box, type the IQ name and click **OK**.
4. Double-click the new IQ in the Contents pane.
5. In the **Event Type "get" interval** box, type the number of milliseconds you want the IQ Manager to wait between IQ polls when the IQ is empty. The default (100 ms) is recommended. Higher polling frequency yields a greater load on the IQ Manager.
6. Select the **Auto Recovery** check box to enable the IQ for auto recovery after an unexpected shutdown. When you select this option, the IQ Manager writes log files for this IQ that are used for the recovery.
7. To change the IQ from simple pub/sub to subscriber pool, click the **Advanced** tab.
8. Under **IQ behavior**, select one of the following:
   - **Simple publish/subscribe**: all Collaborations that subscribe to an Event Type receive all Events for that Event Type.
   - **Subscriber pool:** point-to-point transmission—as Events are received for a particular Event Type, they are received by only one subscriber in the pool (usually the subscriber that is idle longest). When Events are published for multiple subscribers, selecting this check box enables only the first subscriber to receive the Event. Leaving the check box clear (the default) enables all subscribers to receive the Event.
9. When you have finished setting the IQ component's properties, click **OK**.

3.6 Deleting JMS IQs

You can delete single JMS IQs or IQ Managers from schemas in e*Gate Schema Designer. When you delete an IQ Managers, all its IQs are deleted.

After deleting IQs, you may want to delete the **.DBS** files from the STCMS directory in the e*Gate client installation directory. For example, **c:\egate\client\stcms**. Doing so removes all queues, topics, subscriber data, and so on.

To delete JMS IQ Managers

1. In the **Components** tab in the e*Gate Schema Designer, expand the **Participating Host** and the Control Broker folders.
2. Click the IQ Manager to be deleted.
3. Press DELETE.
4. Click **OK**.

To delete JMS IQs

1. In the **Components** tab in the e*Gate Schema Designer, expand the **Participating Host** and the Control Broker folders.
2. Click the IQ Manager.
3. In the Contents pane, click the IQ to be deleted.

4. Press DELETE.
5. Click **OK**.

Chapter 4

Configuring JMS e*Way Connections

This chapter describes how to create and configure JMS e*Way Connections:

- Section 4.1 (**"Creating and Configuring JMS e*Way Connections" on page 41**) provides a step-by-step procedure that allows you to create, view, or edit a JMS e*Way Connection.
- Section 4.2 (**"Configuration Parameters for JMS e*Way Connections" on page 43**) lists and describes the configuration parameters and their settings.

Overview: Advantages of JMS e*Way Connections

While the Oracle SeeBeyond JMS IQ Manager can be accessed as an "IQ" (actually a proxy), e*Way Connections have many advantages:

- e*Way Connections are JMS-compliant and, where required, can be XA-compliant; JMS IQ proxies merely allow access to the JMS IQ Manager as if it were an IQ.
- Accessing the Oracle SeeBeyond JMS IQ Manager through an IQ proxy requires the Collaboration to publish and subscribe to a physical component, whereas accessing it through an e*Way Connection allows publication and subscription to a logical unit, which can be either a topic (broadcast) or a queue (point-to-point).
- JMS e*Way Connections provide support for Event Linking and Sequencing (ELS); JMS IQ proxies do not.
- JMS e*Way Connections provide support for external applications via the JMS API Kit (see the *e*Gate API Kit Developer's Guide*); JMS IQ proxies do not.

Note: *Collaborations using e*Way Connections to the Oracle SeeBeyond JMS IQ Manager must use the Java Collaboration Service (JCS). Event Type instances that use JMS e*Way Connections can be set to* ***either*** *In* ***or*** *Out but not both—in other words, you cannot use an In/Out Event Type instance with a JMS e*Way Connection.*

4.1 Creating and Configuring JMS e*Way Connections

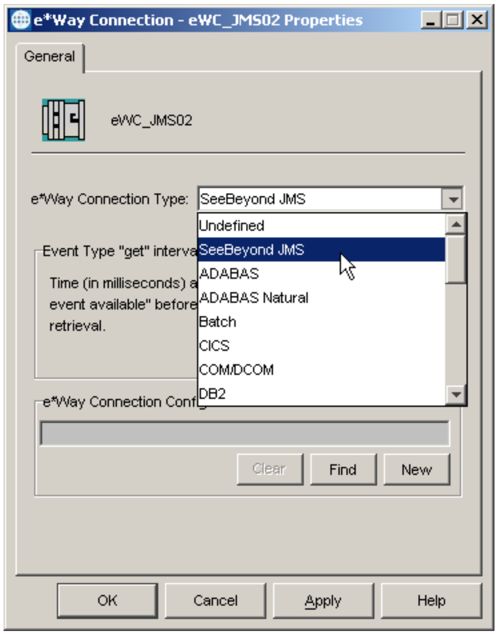
Use the Schema Designer to set up or modify JMS e*Way Connections.

To create and configure a new JMS e*Way Connection

1. In the Schema Designer's Navigator pane, click the **Components** tab (if necessary), and then click the **e*Way Connections** folder.
2. In the Palette, click **Create a New e*Way Connection**.
3. In the dialog box, enter the name of the new e*Way Connection, and then click **OK**.

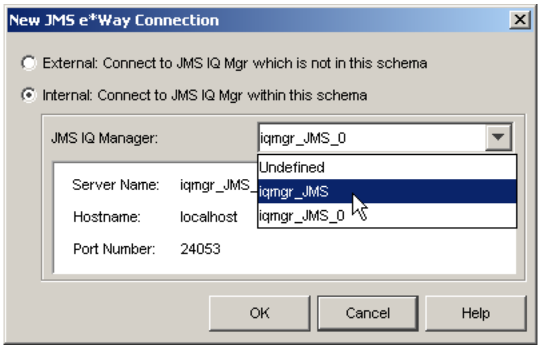
   A new e*Way Connection appears in the component pane.
4. In the component pane, select the new e*Way Connection and then edit its properties.

   The **e*Way Connection -** <*name*> **Properties** dialog box appears.
5. Under **e*Way Connection Type**, click **SeeBeyond JMS** as shown below.



6. Under **e*Way Connection Configuration File**, click **New**.

   The **New JMS e*Way Connection** dialog box appears.



7 Do one of the following:

- To create an entirely new set of configuration parameters, click the **External: Connect to JMS IQ Mgr which is not in this schema** option, click **OK**, and then use the **Edit Settings** dialog box (see below) to set the configuration parameters for this e*Way Connection.
- To inherit the configuration parameters from a pre-existing JMS IQ Manager, click the **Internal: Connect to Existing JMS IQ Manager within this schema** option, select a JMS IQ Manager from the list (its server name, hostname, and IP address are displayed), and then click **OK**. A configuration file (**.cfg** file) is created based on the configuration parameters of the selected JMS IQ Manager. If you want to view or modify the e*Way Connection configuration parameters or add user notes, click **Edit** and use the **Edit Settings** dialog box (see below) to set the configuration parameters for this configuration file.

Parameters are listed and explained in **"Configuration Parameters for JMS e*Way Connections" on page 43**.

Note: *You may want to add user notes to flag or explain any nonstandard settings.*

8 After using the **Edit Settings** dialog box to set configuration parameters and add user notes, save the **.cfg** file (using the same name as the e*Way Connection, and accepting the default location), close the **Edit Settings** dialog box, and then click **OK** to close the **e*Way Connection Properties** dialog box.

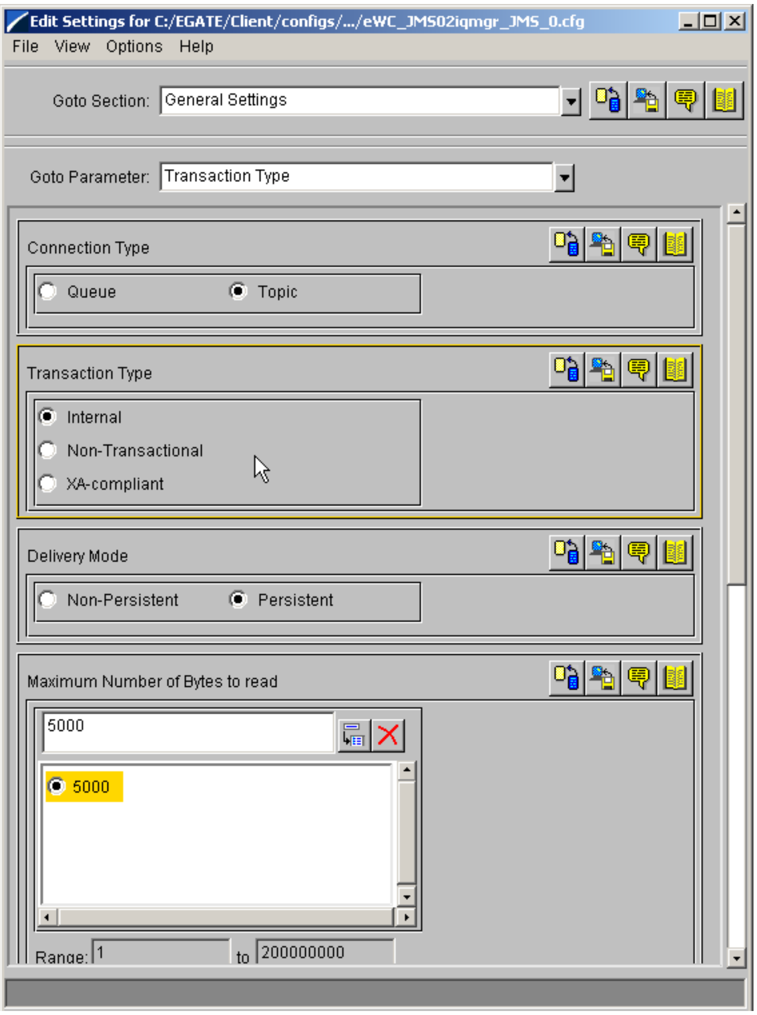
4.2 Configuration Parameters for JMS e*Way Connections

For JMS e*Way Connections, configuration parameters are organized into two sections:

- **General Settings** on page 43
- **Message Service Settings** on page 46

4.2.1 General Settings

The general settings control overall properties of the e*Way Connection. The screen below shows the **Edit Settings** dialog box where you set the general settings.



Edit Settings for C:/EGATE/Client/configs/.../eWC_JMS02iqmgr_JMS_0.cfg

Goto Section: General Settings

Goto Parameter: Transaction Type

Connection Type

○ Queue ● Topic

Transaction Type

● Internal
○ Non-Transactional
○ XA-compliant

Delivery Mode

○ Non-Persistent ● Persistent

Maximum Number of Bytes to read

5000

● 5000

Range: 1 to 200000000

Connection Type

Choices: **Topic** (=default); or **Queue**

Description

For classic publication/subscription behavior, where each message is delivered to *all* current subscribers to the topic (Event Type), select **Topic**.

For point-to-point behavior (equivalent to "subscriber pooling" for conventional IQs), where each message is delivered only *one* recipient in the pool, select **Queue.**

Transaction Type

Choices: **Internal** (=default); **Non-Transactional**; or **XA-compliant**

Description

In **Internal** (transactional) style, a commit is necessary: The message is not saved until the either a commit or a rollback is received.

In **Non-Transactional** mode, the message is automatically saved on the server; no commit is necessary. Operations on topics or queues are immediately acted on, and so a **receive()** is considered complete as soon as the message is received. This means that rollback does not occur: If a Collaboration exits or throws an exception, the message will be lost, since it has already been received. If this behavior is undesirable, set the Delivery Mode parameter to either **Internal** or **XA-compliant**.

In **XA-compliant** transactional style, a two-phase commit is done: The sender sends a prepare, and the commit occurs if and only if all receivers are prepared. Collaborations that use Guaranteed Exactly Once Delivery (GEOD) of Events require XA-compliant transaction types. Consult the XA Processing chapter of the *e*Gate User's Guide* for information about XA use and restrictions.

Delivery Mode

Choices: **Non-Persistent**; or **Persistent** (=default)

Description

Setting **Delivery Mode** to **Persistent** guarantees that the JMS IQ Manager stores each message safely to disk. Setting it to **Non-Persistent** does not guarantee that the message is stored safely to disk. **Non-Persistent** provides better performance but no recovery.

Important: *If the JMS IQ Manager halts when in* ***Non-Persistent*** *mode, undelivered messages are lost.*

Maximum Number of Bytes to read

Default: **5000**

Range: **1 to 200000000**

Description

Your setting for this parameter depends on the size of your messages. For example, if you can anticipate that very large messages will be read, set this parameter accordingly.

Default Outgoing Message Type

Default: **Bytes**

Choices: **Bytes** or **Text**

Description

For messages whose payload is known to be incompatible with other messaging systems, or whose payload is unknown, keep this option set to **Bytes**.

For messages that carry no payload, or carry only a simple TextMessage payload (such as XML documents), you can set this option to **Text.**

Message Selector

For more information about the implementation of message selectors, see the *e*Gate API Kit Developer's Guide*.

Factory Class name

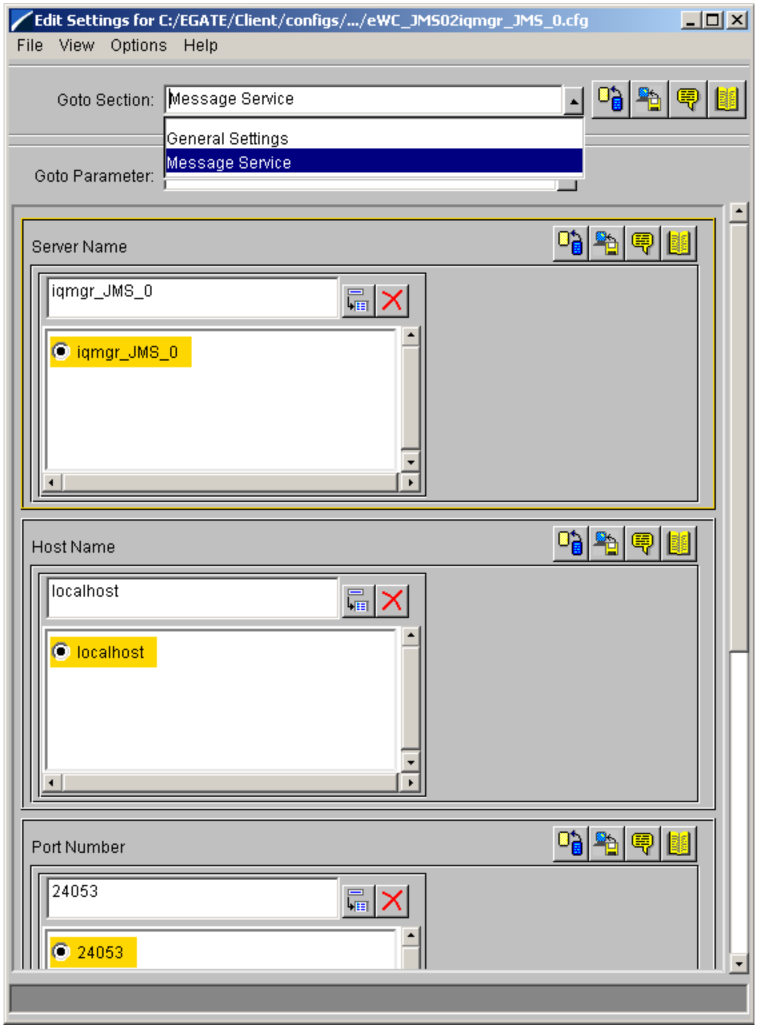
Default: **com.stc.common.collabService.SBYNJMSFactory**

Description

For Oracle SeeBeyond e*Way Connections, keep the default setting: **com.stc.common.collabService.SBYNJMSFactory**

4.2.2 Message Service Settings

Message Service Settings specify low-level information required to establish the JMS. The screen below shows the **Edit Settings** dialog box where you set the Message Service Settings.



Goto Section: Message Service

Server Name: iqmgr_JMS_0

Host Name: localhost

Port Number: 24053

Server Name

Description

Specifies the name of the server—in other words, the JMS IQ Manager—with which e*Gate communicates.

Required Values

A valid server name.

Host Name

Description

Specifies the name of the host with which e*Gate communicates.

Required Values

A valid host name.

Additional Information

This needs to be the same as the host that the JMS IQ Manager is running on.

Port Number

Description

Specifies the port number of the JMS IQ Manager (or message service server) on the host with which e*Gate communicates.

Required Values

A valid port number from 2000 through 1000000000.

Maximum Message Cache Size

Description

Specifies the maximum size of the message cache in bytes.

Required Values

An integer from 1 through 2147483647 (=2^31-1).

Chapter 5

Managing Events and JMS IQ Managers

To manage Events, you use the JMS Administrator in Schema Manager. With JMS Administrator, you can manage live, journaled, and archived Events. You can also use JMS Administrator to manage JMS IQ Managers, such as refreshing, connecting to, and disconnection from IQ Managers. This chapter describes how you use JMS Administrator.

In This Chapter:

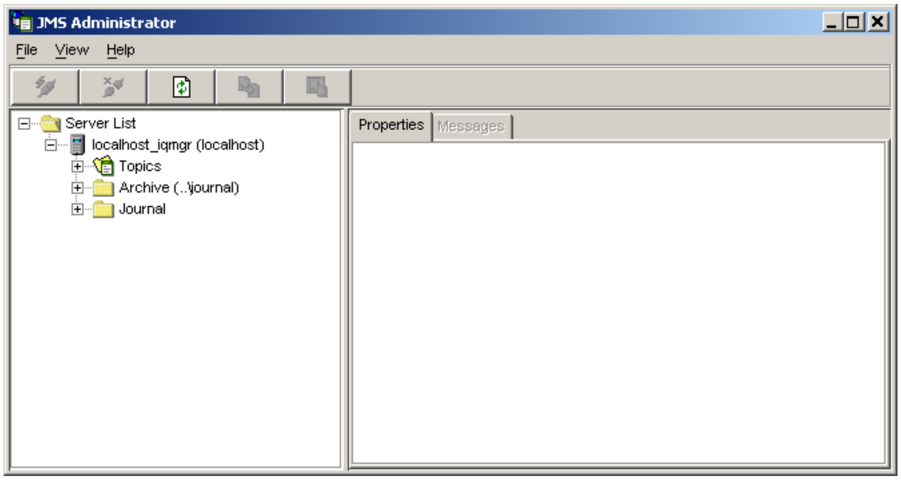
5.1 Starting JMS Administrator

The procedure below describes how you start JMS Administrator.

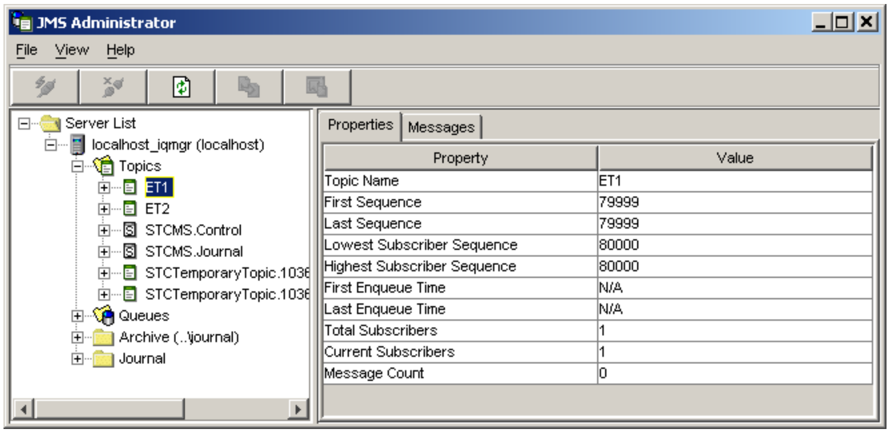
To start JMS Administrator

1. Run the schema you want to monitor.
2. Start Schema Manager and open the schema.
3. On the toolbar, click **Launch JMS Administrator** to start the JMS Administrator.

4 Expand **Server List** folder and the Participating Host folder as shown below.

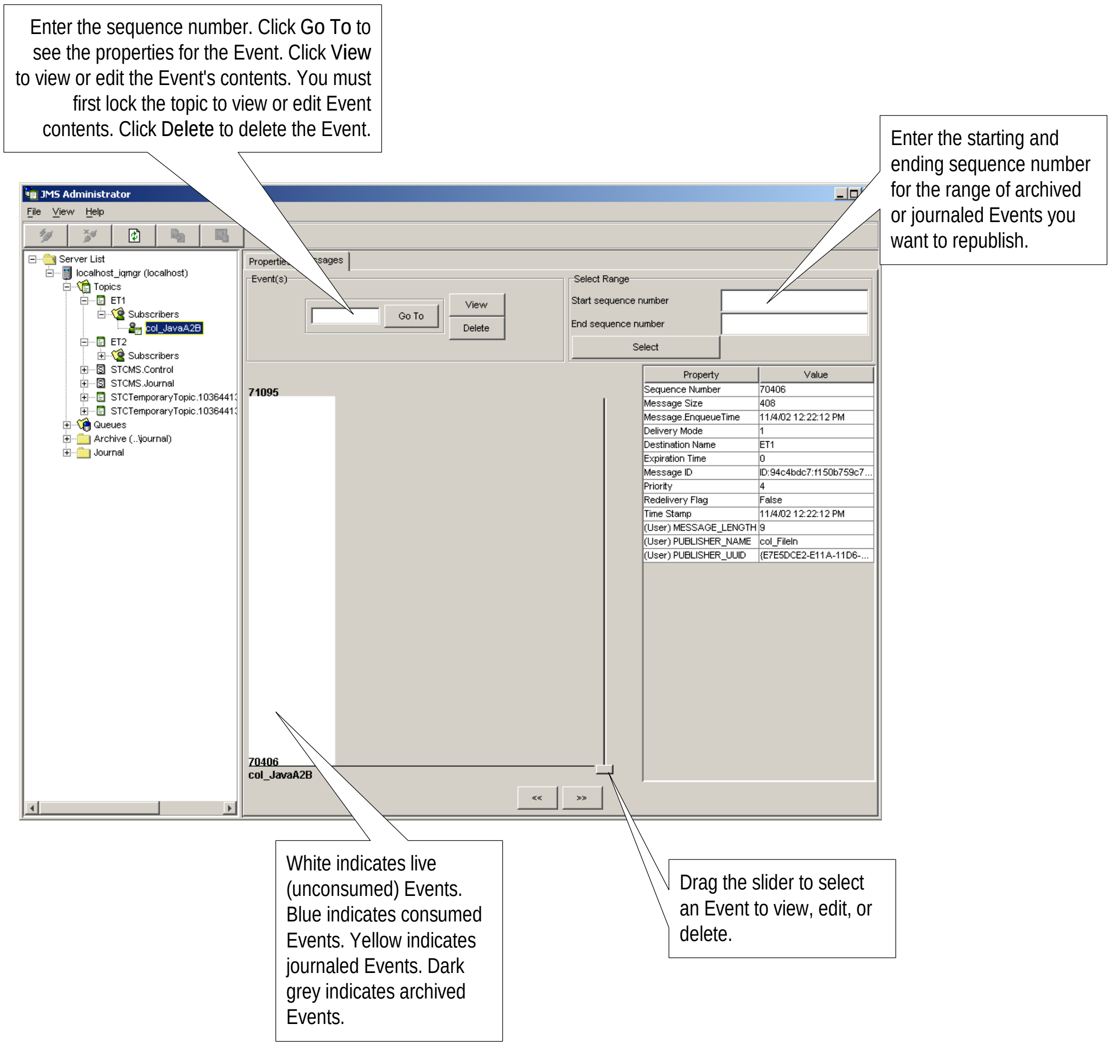


You use the left pane in JMS Administrator to manage JMS IQ Managers as described in the section below. You can also use it to select subscribers, publishers, live Events, archived Events, and journaled Events. The right pane shows the properties for the selection in the left pane in the **Properties** tab. For example, the screen below shows the properties for Event Type ET1. For more information about properties, refer to **"Managing Topics and Queues" on page 56**.



Property	Value
Topic Name	ET1
First Sequence	79999
Last Sequence	79999
Lowest Subscriber Sequence	80000
Highest Subscriber Sequence	80000
First Enqueue Time	N/A
Last Enqueue Time	N/A
Total Subscribers	1
Current Subscribers	1
Message Count	0

The right pane also has a **Messages** tab. You use this tab to view and edit Event Contents, Event properties and republish archived or journaled Events. The **Messages** tab is only available when live, journaled, or archived Events are available for monitoring. The screen below shows the **Messages** tab.



Property	Value
Sequence Number	70406
Message Size	408
Message.EnqueueTime	11/4/02 12:22:12 PM
Delivery Mode	1
Destination Name	ET1
Expiration Time	0
Message ID	ID:94c4bdc7:f150b759c7...
Priority	4
Redelivery Flag	False
Time Stamp	11/4/02 12:22:12 PM
(User) MESSAGE_LENGTH	9
(User) PUBLISHER_NAME	col_FileIn
(User) PUBLISHER_UUID	{E7E5DCE2-E11A-11D6-...

Enter the sequence number. Click **Go To** to see the properties for the Event. Click **View** to view or edit the Event's contents. You must first lock the topic to view or edit Event contents. Click **Delete** to delete the Event.

Enter the starting and ending sequence number for the range of archived or journaled Events you want to republish.

White indicates live (unconsumed) Events. Blue indicates consumed Events. Yellow indicates journaled Events. Dark grey indicates archived Events.

Drag the slider to select an Event to view, edit, or delete.

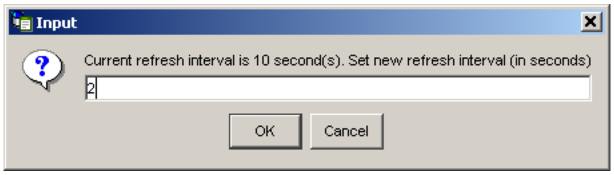
For more information, about viewing and editing Event contents, refer to **"Managing Events" on page 60**. For information about journaled Events, refer to **"Journaling Events" on page 64** and **"Republishing Journaled Events" on page 66**. For information about archived Events, refer to **"Republishing Archived Events" on page 66**.

5.2 Enabling Auto Refresh

You can configure JMS Administrator so that it automatically refreshes the JMS IQ Manager, topics, and queues at a specified time interval. The default interval is five seconds. The procedure below describes how you enable auto refresh and set the refresh interval.

To enable auto refresh

1. Start the JMS IQ Administrator as described in **"Starting JMS Administrator" on page 48**.
2. On the **View** menu, click **Settings**, and then click **Set Auto Refresh Interval**.
3. Type the numbers of seconds for the interval as shown below.



Current refresh interval is 10 second(s). Set new refresh interval (in seconds)

2

4. Click **OK**.
5. On the **View** menu, click **Settings**, and then click **Enable Auto Refresh**.

5.3 Managing JMS IQ Managers

With JMS Administrator you can manage JMS IQ Managers as follows:

- Add JMS IQ Managers
- Delete JMS IQ Managers
- Refresh JMS IQ Managers
- Connect to JMS IQ Managers
- Disconnect from JMS IQ Managers
- Bind objects
- Show bindings

5.3.1 Adding JMS IQ Managers for Monitoring

The JMS Administrator displays the IQ Managers in the schema you are monitoring. It also allows you to add other JMS IQ Managers for monitoring if you want to do so. When you add a JMS IQ Manager, you must enter a name for the manager as well as the TCP/IP address of the port number. The TCP/IP port number of the Participating Host is set in e*Gate Schema Designer the **Advanced** tab of the Properties dialog box for each JMS IQ Manager. For a JMS IQ Manager in the "Up" state, you can view this information in Schema Manager by clicking the **Control** tab.

To add a JMS IQ Manager for monitoring

1. Start the JMS IQ Administrator as described in **"Starting JMS Administrator" on page 48**.
2. Right-click the **Server List** folder and click **Add Server**.
3. In the **Please enter host name** box, type the name of the Participating Host of the JMS IQ Manager you want to add.

4. Click **OK**.
5. In the **Please enter port number** box, type the TCP/IP port number for this JMS IQ Manager.
6. Click **OK**.

5.3.2 Deleting JMS IQ Managers from Monitoring

JMS Administrator allows you to monitor two types of JMS IQ Managers: those that are part of the schema you are monitoring, and those that you have added manually as described in the section above.

JMS Administrator allows you to delete JMS IQ Managers. This does not remove the JMS IQ Managers from the schema (if it is part of a schema). It merely removes the JMS IQ Manager from the JMS Administrator window. This is useful mostly for JMS IQ Managers that you have added but no longer need to monitor.

You cannot permanently delete an IQ Manager that is part of a schema. The next time you start JMS Administrator, the IQ Manager displays again as part of the schema.

To delete a JMS IQ Manager from monitoring

1. Start the JMS IQ Administrator as described in **"Starting JMS Administrator" on page 48**.
2. Expand the **Server List** folder.
3. Right-click the JMS IQ Manager you want to delete and click **Delete Server**.
4. Click **OK**.

5.3.3 Refreshing, Connecting to, or Disconnecting from JMS IQ Managers

To refresh, connect to, or disconnect from JMS IQ Managers

1. Start the JMS IQ Administrator as described in **"Starting JMS Administrator" on page 48**.
2. Expand the **Server List** folder and the JMS IQ Manager folder (***hostname*_iqmgr**).
3. To refresh the JMS IQ Manager selected, click **Refresh** .
4. To connect to the JMS IQ Manager selected, click **Connect** .
5. To refresh the JMS IQ Manager selected, click **Disconnect** .

When a JMS IQ Manager is disconnected, you see the following icon:

5.3.4 Binding Objects for JNDI Lookup

If you use a Java Naming and Directory Interface^{TM} (JNDI) provider, you can use the JMS Administrator to bind e*Gate objects for JNDI lookup. e*Gate is compliant with JNDI version 1.2.1 and provides the following connection factories:

- **STCTopicConnectionFactory**
- **STCQueueConnectionFactory**
- **STCXATopicConnectionFactory**
- **STCXAQueueConnectionFactory**

You can bind connection factories, IQs, and topics as described in the sections below.

Configuring e*Gate for JNDI Binding

Before you use Schema Manager to create, remove, and view JNDI bindings, you must configure e*Gate for your JNDI service provider as described below.

To configure e*Gate for JNDI binding

1. Put the necessary **.jar** files for your JNDI provider in the ***egate*\client\JRE\1.3\lib\ext** directory, where *egate\client* is the directory where the e*Gate Participating Host is located. The 1.3 folder may be organized differently depending on your platform.
   - For File System Context (FS Context), this is **fscontext.jar** and **providerutil.jar**
   - For Lightweight Directory Access Protocol (LDAP) and other JNDI providers, follow the provider's instructions on which files you need.
2. On the drive where the Participating Host is located, create a directory where you want to store binding information. For example: **c:\jndibindings**.
3. Create a **jndi.properties** file as follows:
   - For FS Context JNDI service provider:

     ```
     java.naming.factory.initial=com.sun.jndi.fscontext.RefFSContextFactory
     java.naming.provider.url=file:///bindingpath
     ```

     where *bindingpath* is the path to the directory created in step 2.
   - For UNIX, specify the path to a mounted drive for the directory created in step 2:

     ```
     java.naming.factory.initial=com.sun.jndi.fscontext.RefFSContextFactory
     java.naming.provider.url=file:/drive/jndibindings
     ```

     where *drive* is a mounted drive.
   - For LDAP, create a **jndi.properties** file for the LDAP JNDI service provider. For example:

     ```
     java.naming.factory.initial=com.sun.jndi.ldap.LdapCtxFactory
     java.naming.provider.url=ldap://egate1.stc.com:389/o=jndi
     java.naming.security.authentication=simple
     java.naming.security.principal=cn=Manager,o=jndi
     java.naming.security.credentials=secret
     ```

4 Save the file as **jndi.properties** in the ***egate**client*\JRE\1.3\lib** directory, where *egate**client* is the directory where the e*Gate Participating Host is located. You can use only one JNDI service provider at a time.

Binding STC Connection Factories, Topics, and IQs

You bind e*Gate objects with the JMS Administrator. Before you can do this, you must configure e*Gate for your specific JNDI provider as described in the section above. You can bind connection factories, topics, and subscriber pool IQs. You cannot bind IQs that have IQ behavior configured as simple publish/subscribe or to not store journaled messages.

The connection factory bindings are associated with a JMS server running on a particular server on a particular port. If the server or the port changes for the JMS server, you must rebind the object.

To bind STC connection factories, topics, and IQs

1 Start the JMS IQ Administrator as described in **"Starting JMS Administrator" on page 48**.

2 In the left pane, expand the **Server List** folder.

3 To bind a connection factory, right-click the IQ Manager and click **Bind STC Connection Factory** and click the factory object you want to bind.

To bind a topic, right-click the topic and click **Bind Topic**.

To bind an IQ, right-click the IQ and click **Bind Queue**.

4 In the **Enter name to bind** box, specify the binding name, indicating the object to be bound. For example: AltasXACTF.

For LDAP, you must specify **cn=** before the binding name. For example: cn=AltasCQF.

5 Click **OK**.

5.3.5 Viewing or Removing JNDI Bindings

To view or remove JNDI bindings

1 Start the JMS IQ Administrator as described in **"Starting JMS Administrator" on page 48**.

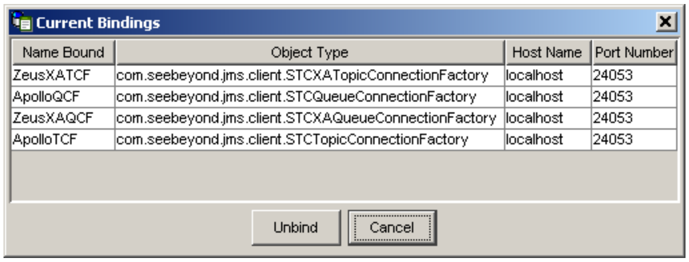
2 In the left pane, expand the **Server List** folder.

3 To view or remove connection factory bindings, right-click the JMS IQ Manager.

To view or remove topic bindings, right-click the topic.

To view or remove IQ binding, right-click the IQ.

4 Click **Show Bindings**. The **Current Bindings** dialog box shows the bindings that are currently specified. The screen below shows the current connection factory bindings.



Name Bound	Object Type	Host Name	Port Number
ZeusXATCF	com.seebeyond.jms.client.STCXATopicConnectionFactory	localhost	24053
ApolloQCF	com.seebeyond.jms.client.STCQueueConnectionFactory	localhost	24053
ZeusXAQCF	com.seebeyond.jms.client.STCXAQueueConnectionFactory	localhost	24053
ApolloTCF	com.seebeyond.jms.client.STCTopicConnectionFactory	localhost	24053

5 To remove a binding, click the binding and click **Unbind**.

5.3.6 Viewing JMS IQ Manager Properties

To view JMS IQ Manager properties

1 Start the JMS IQ Administrator as described in **"Starting JMS Administrator" on page 48**.

2 Expand the **Server List** folder.

3 Click the JMS IQ Manager (***hostname*_iqmgr**) you want to view.

4 Click **Refresh** 🔄.

The right pane displays the properties of the JMS IQ Manager selected. The table below describes the properties listed.

Table 2 JMS IQ Manager Properties

Property Name	Description
Up Time	Date and time when this JMS IQ Manager was most recently started. Displayed in local format.
Total Message Memory Used	Running byte count of memory allocated for all uncommitted message payloads.
Total Message Count	Total number of messages processed since this JMS IQ Manager has been up. This total includes data messages, acknowledgements, and so on.
Total Number of Event Types	Number of Event Types (the total sum of topics and queues) that this JMS IQ Manager manages, including temporary topics and queues.
Total Number of Connections	Total count of JMS IQ Manager connections currently open for all client types combined: topic publishers and subscribers, and queue senders and receivers.
Port Number	The particular TCP/IP port of the Participating Host that this JMS IQ Manager is listening to.

5.4 Managing Topics and Queues

JMS IQ Managers 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 JMS IQ Manager publishes an Event 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 JMS IQ Manager sends an Event 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 IQ implementations.

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

- Each topic or queue maintains a *sequence* of messages in progress; each message has a timestamp called its *enqueue time* that indicates when it was published or sent.
- Messages are held in virtual memory until delivered; the difference between the *highest sequence number* and the *lowest sequence number* is an indicator of how much memory is being used by that topic/queue.
- Messages that have been read and committed by their subscribers/receivers are subject to *cleanup*. After cleanup, the lowest sequence number is increased by the number of messages that were delivered and successfully committed.

Note: *When a bar chart shows several subscribers, the bar graph that is "seen" by a particular subscriber may reflect all messages published for these subscribers, and thus may contain messages of no interest to the particular subscriber "seeing" it.*

STCMS.Control, STCMS.Journal, and STCTemporary Topics

Every JMS IQ Manager has a special topic, **STCMS.Control**, which has a nondurable "subscriber" that allows the Oracle SeeBeyond Message Service utilities to monitor the status of the JMS IQ Manager and its topics; for information about these utilities, see **Chapter 6**. You also see the special topic **STCMS.Journal**. This topics is where the JMS IQ Manager publishes messages to be retrieved by the journaler.

e*Gate also creates temporary topics with one or more transient subscribers and topic names like **STCTemporaryTopic.***n.m*. All these special topics can be ignored.

Receiver Names

The names of queue receivers for point-to-point messages are tracked by their location in memory rather than by their logical names. This means that names that receivers have within the e*Gate schema do not display.

5.4.1 Viewing Topic Properties

To view topic properties

1. Start the JMS IQ Administrator as described in **"Starting JMS Administrator" on page 48**.
2. Expand the **Server List** folder and the JMS IQ Manager folder (***hostname*_iqmgr**).
3. Expand the **Topics** folder.
4. Click the topic for which you want to view properties.
5. To refresh the topic, click **Refresh** .

The right pane displays the properties of the topic selected. The table below describes the properties listed.

Table 3 Topic Properties

Property Name	Description
Topic Name	Logical name of the topic (Event Type).
First Sequence	Sequence number of the oldest message that has not yet been processed by all subscribers (the oldest available message).
Last Sequence	Sequence number of the most recent message.
Lowest Subscriber Sequence	Sequence number of the most recent message that has been committed by at least all subscribers.
Highest Subscriber Sequence	Sequence number of the most recent message that has been committed by at least one subscriber.
First Enqueue Time	Timestamp of the first message for this topic that has still not been processed.
Last Enqueue Time	Timestamp of the most recent message for this topic that has still not been processed.
Total Subscribers	Number of subscribers *registered to consume* messages for this topic (including subscribers that are currently disconnected).
Current Subscribers	Number of currently connected subscribers.
Message Count	Number of messages on this topic that are still unprocessed by at least one subscriber.

5.4.2 Viewing Subscribing Collaboration Properties

To view subscribing Collaboration properties

1. Start the JMS IQ Administrator as described in **"Starting JMS Administrator" on page 48**.
2. Expand the **Server List** folder and the JMS IQ Manager folder (***hostname*_iqmgr**).
3. Expand the **Topics** folder and the Event Type folder.
4. Expand the **Subscribers** folder to display the subscribing Collaborations.

5. Open the topic's **Subscribers** folder.
6. To refresh the topic, click **Refresh**.

The right pane displays the properties of the Collaboration selected. The table below describes the properties listed.

Table 4 Subscribing Collaboration Properties

Property Name	Description
Subscriber Name	Logical name of this subscriber.
Client ID	When combined with the subscriber name, the client ID provides a unique identifier for a durable subscriber.
Topic Name	Logical name of the topic (Event Type) this subscriber is listening for.
High Sequence Number	Sequence number of the most recent message that this subscriber has read.
Committed Sequence Number	Sequence number of the most recent message that this subscriber has read and committed.

5.4.3 Viewing Queue Properties and Refreshing Queues

Whether the **Queues** folders appear in JMS Administrator depends on whether your schema has queues.

To view queue properties and refresh queues

1. Start the JMS IQ Administrator as described in **"Starting JMS Administrator" on page 48**.
2. Expand the **Server List** folder and the JMS IQ Manager folder (***hostname*_iqmgr**).
3. Expand the **Queues** folder.
4. Click the queue for which you want to view properties.
5. To refresh the queue, click **Refresh**.

The right pane displays the properties of the queue selected. The table below describes the properties listed.

Table 5 Queue Properties

Property Name	Description
Queue Name	Logical name of the Event Type for this queue.
First Enqueue Time	Timestamp of the first message in this queue that has still not been processed.
Last Enqueue Time	Timestamp of the most recent message in this queue.
Current Receivers	Number of receivers for this queue.

Table 5 Queue Properties (Continued)

Property Name	Description
Message Count	Total number of messages in this queue that have already been processed and deleted.
Outstanding	Total number of messages in this queue that have not yet been processed.

5.4.4 Viewing and Refreshing Queue Receivers

To list the receivers of a queue

1. Start the JMS IQ Administrator as described in **"Starting JMS Administrator" on page 48**.
2. Expand the **Server List** folder and the JMS IQ Manager folder (***hostname*_iqmgr**).
3. Expand the **Queues** folder and the **Receivers** folder.
4. To refresh the queue, click **Refresh** .

The right-hand pane displays a list of the receivers for the selected queue.

5.4.5 Viewing Receiver Properties

To view receiver properties

1. Start the JMS IQ Administrator as described in **"Starting JMS Administrator" on page 48**.
2. Expand the **Server List** folder and the JMS IQ Manager folder (***hostname*_iqmgr**).
3. Expand the **Queues** folder and the **Receivers** folder.
4. To refresh the queue, click **Refresh** .

The right pane displays the properties of the receiver selected. The table below describes the properties listed.

Table 6 Receiver Properties

Property Name	Description
Receiver ID	Address in virtual memory of this receiver.
Queue Name	Logical name of the queue (Event Type) sending to this receiver.
Session ID	ID of the current JMS session.
Committed Sequence	Number of messages that this queue receiver has already committed.
Uncommitted Sequence	Number of messages that this queue receiver has read but not yet committed.

5.4.6 Creating Topics or Queues

To create topics or queues for JMS IQ Managers

1. Start the JMS IQ Administrator as described in **"Starting JMS Administrator" on page 48**.
2. In the left pane, expand the **Server List** folder.
3. Right-click the JMS IQ Manager (***hostname*_iqmgr**).
4. To create a topic, click **Create Topic**.

   To create a topic, click **Create Queue**.
5. In the **Please enter the topic/queue name** box, type the name of the new topic or queue.
6. Click **OK**.

This creates a new topic or queue under the **Topic** or **Queues** folder.

5.4.7 Deleting Topics or Queues

You can delete topics and queues in JMS. You cannot permanently delete a topic or queue that is part of a schema. If you do so, the next time you start JMS Administrator, the deleted topic or queue displays again as part of the schema.

To delete topics or queues for JMS IQ Managers

1. Start the JMS IQ Administrator as described in **"Starting JMS Administrator" on page 48**.
2. In the left pane, expand the **Server List** folder.
3. Expand the JMS IQ Manager folder and the **Topics** or **Queues** folder.
4. Right-click the topic or queue you want to delete.
5. To delete a topic, click **Delete Topic**.

   To delete a topic, click **Delete Queue**.
6. Click **OK**.

5.5 Managing Events

You can manage live, journaled, and archived Events in JMS Administrator with the **Messages** tab. The **Messages** tab is only available when live, journaled, or archived Events are available for monitoring. For example, the **Messages** tab only becomes available for live Events when your schema is running and publishing live Events.

5.5.1 Viewing Event Properties

You can view live, journaled, and archived Event properties in JMS Administrator. The procedure below describes how you view the Events, and provides a list of the message properties with descriptions. To view archived Event properties, you must have an unzipped archive available for import as described in **"Republishing Archived Events" on page 66**.

To view Events, you must have viewing enabled for the IQ Manager as described in **"EnableView" on page 30**.

To view Event properties

1. Start the JMS IQ Administrator as described in **"Starting JMS Administrator" on page 48**.
2. In the left pane, expand the **Server List** folder and the JMS IQ Manager (***hostname*_iqmgr**).
3. To view the properties of live events, expand the **Topics** or **Queues** folder and click the Event Type.

   To view properties of live events for a particular subscribing Collaboration, expand the **Topics** folder and click the Collaboration.

   To view the properties of journaled events, expand the **Journal** and **Journaled Topics** or **Queues** folders.

   To view the properties archived Events, click the **Archive** folder, enter the archive directory, and expand the **Archive Topics** or **Queues** folder.
4. Click the **Messages** tab.

The right pane displays the properties of the Event selected. The table below describes the properties listed.

Table 7 Event Properties

Property Name	Description
Sequence Number	Number of the Event.
Message Size	Total byte size of the stored Event.
Message.Timestamp	Time that the Event was sent.
Message.EnqueueTime	Date and time the Event was received.
Journaler.ExpirationTime	Date and time the journaled Event expires. The default journal expiration time is one day after the Event is enqueued. You can set the duration for the journal expiration as described in **"JournalTimeToLive" on page 31**. Events remain in the journal database until they are archived. Archiving removes journal expired Events from the journal database. If you enable journaling, archiving daily is recommended. For more information, refer to **"Archiving" on page 76**.
Delivery Mode	Mode of Event delivery: persistent or non-persistent.
Destination Mode	Topic or queue value where the Event was sent.

Table 7 Event Properties (Continued)

Property Name	Description
Expiration Time	Current GMT time plus the live expiration time. The live expiration time is 30 days be default. To specify a different duration, refer to **"MaxTimeToLive" on page 30**. You can also specify the live expiration time for a Collaboration as described in the *e*Gate Integrator User's Guide*. If this field displays 0, no live expiration time was set for Collaboration; the live expiration time is the MaxTimeToLive setting.
Message ID	Unique identification number for the Event.
Priority	Priority level (default is 4).
Redelivery Flag	If Event was delivered but not acknowledged, the value for this property is 1.
Reply To	Name of the topic where the reply message is sent.
Time Stamp	Date and time the Event was received.
Max Messages	Maximum number of Events that can be displayed.
Min Messages	Minimum number of Events that can be displayed.
Topic Name	Name of the topic name under which this Events is filed.
Type	Type identifier that is supplied by a client.

Note: *The Event properties shown in the JMS Administrator now include user properties defined for messages through JMS API functions.*

5.5.2 Viewing and Editing Event Contents

You can view the contents of live, journaled, and archived Events in JMS Administrator. You can also edit the contents of live Events, but not journaled or archived Events. To view archived Events, you must have an unzipped archive available for import as described in **"Republishing Archived Events" on page 66**.

To be able to view or edit live Event contents, you must first lock the Event Type so that you can make your changes before the Event is published. You must unlock the Event Type once you are done viewing and editing for the Events to be processed. If you exit the JMS Administrator without unlocking an Event Type, the Event Type remains locked and unprocessed.

Locking a topic does not change the expiration time of the Events. It is possible for an Event to expire while the topic is locked.

To view and edit Events, you must have viewing and editing enabled for the IQ Manager as described in**"EnableView" on page 30** and **"EnableEdit" on page 30**.

To view and edit Events

1. Start the JMS IQ Administrator as described in **"Starting JMS Administrator" on page 48**.
2. In the left pane, expand the **Server List** folder and the JMS IQ Manager (***hostname*_iqmgr**).

3 To view the properties of live events, expand the **Topics** or **Queues** folder.

To view the properties of journaled events, expand the **Journal** and **Journaled Topics** or **Queues** folders.

To view archived Events, click the **Archive** folder, enter the archive directory, and expand the **Archive Topics** or **Queues** folder.

4 Click the Event Type for which you want to view or edit an Event.

5 Click the **Messages** tab.

6 For live Events only, right-click the Event Type and click **Lock**.

This locks all live Events for the Event Type selected; a lock icon appears by the Event Type folder.

7 Select the Event by dragging the scroll bar, or type the Event sequence number in the **Go To** box and click **Go To**.

8 Click **View**.

The Event appears in the text editor.

9 To edit live Events, click **Edit**, edit the Event, and click **OK**.

10 Click the locked Event Type and click **Unlock**.

5.5.3 Deleting Events

You can delete Events in JMS Administrator during monitoring. You can only delete one Event at a time. If you select a range of Events, only the first Event is deleted.

To delete Events, you must have deleting enabled for the IQ Manager as described in **“EnableDelete” on page 30**.

To delete Events

1 Start the JMS IQ Administrator as described in **“Starting JMS Administrator” on page 48**.

2 In the left pane, expand the **Server List** folder and the JMS IQ Manager (***hostname*_iqmgr**).

3 To delete live events, expand the **Topics** or **Queues** folder.

To delete journaled events, expand the **Journal** and **Journaled Topics** or **Queues** folders.

To delete archived Events, click the **Archive** folder, enter the archive directory, and expand the **Archive Topics** or **Queues** folder.

4 Click the Event Type for which you want to view or edit an Event.

5 Click the **Messages** tab.

6 Select the Event by dragging the scroll bar, or type the Event sequence number in the **Go To** box and click **Go To**.

7 Click **Delete**.

5.6 Journaling Events

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

- Enabling journaling.
- Setting a path for the journal location.
- Setting a journal expiration time.

Journaling is disabled by default, so you must manually enable it before you can use it.

When you enable journaling, every Event that is enqueued is copied to the journal database. The Event is then held in the journal database for the duration of the journal expiration time regardless of at what point the Event is consumed. By default, this is one day. To change the journal expiration time, refer to **"JournalTimeToLive" on page 31**.

Once the journaled Event expires, it remains in the journal database until you archive. When you archive, all Events in the journal database are archived, and the journal expired Events are removed from the journal database. If you do not archive regularly, the journal database retains journal expired Events and may grow exceedingly large. Because the journal database and the IQ database are located on the same system, it is important to avoid running out of disk space. *It is recommended that you archive daily when journaling is enabled.*

You archive with the **-backup** flag of the STC MS Control utility. For more information, refer to **"Backing Up" on page 76**.

When Events are in the journal database, you can view them but not edit them. You can use either the JMS Administrator or the STC MS Control utility to view and republish journaled Events. For more information about the MS Control utility, see **"Using the STC Message Server Control Utility" on page 68**.

You can also enable journaling solely for a specific Collaboration as described in **"Enabling Journaling for a Collaboration" on page 65** by specifying the journal expiration time. This is most commonly used for schemas with standard IQs, but you can use it in combination with JMS IQ Managers. The journal settings for the Collaboration take precedence.

Journaling Log File

When journaling is enabled, journaling activities and errors are monitored and logged in the following file:

***egate\client*\logs\journal_*iqmgr*.log**

where *egate\client* is the installation directory for the e*Gate Participating Host, and where *iqmgr* is the name of the JMS IQ Manager.

5.6.1 Enabling Journaling

Before you can journal Events, you must enable the JMS IQ Manager that handles those Events for journaling. By default, journaling is disabled. You can also specify a location for the journal files, and the duration after which you want journaled Events to expire. For more information about these settings, refer to **"Server Settings" on page 31**.

To enable journaling

1. In the **Components** tab of the e*Gate Schema Designer, expand the **Participating Hosts** folder, then double-click the IQ manager (***hostname*_iqmgr**) for which you want to enable journaling.

   The **IQ Manager - Properties** dialog box appears.

2. Click **Edit**.
3. In the **GoTo Section** list, click **Server Settings**.
4. In the **GoTo Parameter** list, click **Journalling**.
5. Under **Journalling**, click **Yes**. This enables journaling and archiving.
6. Under **JournalTimeToLive**, type the maximum number of seconds for an Event to remain in the journaler before it expires and is stored in the archive database. The default is 86400 (one day in seconds). The journal expiration time is the Event's enqueue time plus the duration set for the journal expiration time.
7. Under **JournalPath**, specify the path to the locations where the journal files are stored. This can be an absolute path or a path relative to the directory where the e*Gate Participating Host is installed.
8. On the **File** menu, click **Save**.

If the IQ Manager manages multiple queues, these settings are for all queues; they cannot be set individually.

5.6.2 Enabling Journaling for a Collaboration

If your schema uses standard IQs instead of JMS Connections, you can enable journaling for a Collaboration. This setting takes precedence over journal settings specified for the IQ Manager as described in the previous section. If you do not specify journal settings for a Collaboration, the JMS Administrator uses the journal settings for the IQ Manager.

To enable journaling for a Collaboration

1. In the **Components** tab of the e*Gate Schema Designer, expand the **Participating Host** folder, and click the Control Broker (***hostname*_cb)**.
2. Click the e*Way for the Collaboration.
3. Double-click the Collaboration in the Contents pane.
4. Under **Publications**, click the Event Type.
5. Click **Advanced**.
6. Click **Expiration**.
7. Under **Journalled Event Types expire after,** select the amount of time after which a journaled Event expires.

The journal expiration time is the Event’s enqueue time plus the duration set for the journal expiration time. By default, the journal expiration time is one day.

8. Click **OK**.
9. Click **Apply**.

5.7 Republishing Journaled Events

To republish journaled Events, you can use the JMS Administrator as shown below, or the MS Control Utility as described in **“Republishing Messages from Topics” on page 74** and **“Republishing Messages from Queues” on page 74**.

If the journal database is empty when you start JMS Administrator, you are unable to view journaled Events that arrive in the journal database at a later time. You must first disconnect and reconnect to the JMS server to be able to view these journaled Events.

To republish journaled Events

1. Start the JMS IQ Administrator as described in **“Starting JMS Administrator” on page 48**.
2. In the left pane, expand the **Server List** folder and the JMS IQ Manager (***hostname*_iqmgr**).
3. Expand the **Journal** and **Journal Topics** or **Queues** folder.
4. Click the **Messages** tab.
5. In the **Start Sequence Number** box, type the sequence number of the start of the range of Event you want to republish.
6. In the **End Sequence Number** box, type the sequence number of the end of the range of Event you want to republish.
7. Click **Select**.

   The selected range displays in green.
8. Click **Republish to live** to republish the Events.

5.8 Republishing Archived Events

You can republish archived Events using JMS Administrator as described below. Before you can do so, you must have an unzipped archive available. To create an archive, you use the STC MS Control utility as described in **“Backing Up” on page 76**.

You can also use the MS Control utility to republish archived Events as described in **“Republishing Messages from Topics” on page 74**.

To republish archived Events

1. Unzip the archive file in an empty archive directory. Do not unzip multiple archive files in the same directory.
2. Start the JMS IQ Administrator as described in **“Starting JMS Administrator” on page 48**.
3. In the left pane, expand the **Server List** folder and the JMS IQ Manager (***hostname*_iqmgr**).
4. Click the **Archive** folder.
5. Type the path to the archive directory and click **OK**.
6. Expand **Archive Topics** or **Queues** folder and click the Event Type.
7. Click the **Messages** tab.
8. In the **Start Sequence Number** box, type the sequence number of the start of the range of Event you want to republish.
9. In the **End Sequence Number** box, type the sequence number of the end of the range of Event you want to republish.
10. Click **Select**.

    The selected range displays in green.

11. Click **Import to live** to republish the Events.

Chapter 6

Using the STC Message Server Control Utility

This chapter includes information about the Oracle SeeBeyond Java Message Service (MS) command line utility, **stcmsctrlutil.exe**.

6.1 Overview

The MS Control utility provides a command-line interface to an active MS server. From a command line or batch program, you can use the MS Control utility to:

- 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 Event or its type.
- Fetch or delete a range of Events.
- Journal, back up, and archive Events.

Preferences for the MS Control utility are stored in the **stcmsutil.Properties** file. See **"stcmsutil.Properties" on page 78**.

6.2 Syntax

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

```
stcmsctrlutil -host host-name -port port-number [-offset port-offset]
           -flag

stcmsctrlutil -host host-name -port port-number [-offset port-offset]
           -flag argument1 [argument2 [argument3]]
```

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

```
stcmsctrlutil --help

stcmsctrlutil --version
```

6.3 Flags and Arguments

Table 8 MS Control Utility Flags and Arguments

Shortcut	Flag *arguments*	Purpose
	--version	View utility version information.
	-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** *topicname*	List all subscribers for the specified topic.
-ts	**-topicstat** *topicname*	View statistics for the specified topic.
-ql	**-queuelist**	List all queues for this server.
-rlfq	**-recvlistforqueue** *queuename*	List all receivers for the specified queue.
-rla	**-recvlistall**	List all receivers for all queues combined.
-qs	**-queuestat** *queuename*	View statistics for a specific queue.
	-host *hostname*	Specify the name of the Participating 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 *portnumber*	Specify the TCP/IP port of the Participating Host that this JMS IQ Manager 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 *portoffset*	Specify a server port offset number.
-ct	**-createtopic** *topicname*	Create a new topic with the specified name.
-dt	**-deletetopic** *topicname*	Delete the specified topic.
-cq	**-createqueue** *queuename*	Create a new queue with the specified name.
-dq	**-deletequeue** *queuename*	Delete the specified queue.

Table 8 MS Control Utility Flags and Arguments **(Continued)**

Shortcut	Flag *arguments*	Purpose
-cs	**-createsub** *topicname subname clientname*	Create a new subscriber for the specified topic and client. For *clientname*, specify **eGate**.
-ds	**-deletesub** *topicname subname clientname*	Delete a certain subscriber from the specified topic and client. For *clientname,* specify **eGate**.
-qml	**-qmsglist** *queuename seqnumber numbermessages*	List all Events for the specified queue, starting at or above the specified sequence number, and listing no more than *numbermessages* altogether.
-tml	**-tmsglist** *topicname seqnumber numbermessages*	List all Events for the specified topic, starting at or above the specified sequence number, and listing no more than *numbermessages* altogether.
-gqm	**-qmessage** *queuename*	Retrieve the particular message designated by *seqnumber* for the specified queue. If the specified queue contains no message with this sequence number, an error is returned.
-gtm	**-tmessage** *topicname seqnumber*	Retrieve the particular message designated by *seqnumber* for the specified topic. If the specified topic contains no message with this sequence number, an error is returned.
-dtm	**-deltmsg** *topicname seqnumber*	Delete the message at the specified sequence number in the specified topic.
-dqm	**-delqmsg** *queuename seqnumber*	Delete the message at the specified sequence number in the specified queue.
-ctm	**-changetmsg** *topicname seqnumber*	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** queuename *seqnumber*	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 *type*	Specify the data type of the content of the Event. Must be **bytes** or **text**.
-dqm	**-delqmsg** *queuename*	Delete the message at the specified sequence number in the specified queue.
-lt	**-locktopic** *topicname*	Lock a topic from being accessed, preventing any subscriber from receiving messages from it.
-ut	**-unlocktopic** *topicname*	Unlock a topic, restoring access to all subscribers.
-tmi	**-tmimport** *topicname seqno nmgs*	Republish messages from a topic.
-qmi	**-qmimport** *topicname seqno nmgs*	Republishing messages froma queue.

Table 8 MS Control Utility Flags and Arguments **(Continued)**

Shortcut	Flag *arguments*	Purpose
-j	**-journaler**	Browse journaled Events.
-ar	**-archiver** *directoryname*	Browse specified archive.
-b	**-backup** *file date*	Back up Events.
-t	**-timeout** *seconds*	Specify the timeout in seconds.
	--help	View help information.

6.3.1 Displaying the Utility Version

```
stcmsctrlutil --version
Control Utility version 4.5.3.4122 (Sept 22 2002)
Copyright (c) 2002, SeeBeyond Technology Corporation.
All Rights Reserved.
```

6.3.2 Shutting Down the Server

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

6.3.3 Viewing Statistics about Message Service

```
stcmsctrlutil -host localhost -port 24055 -status
Up since: Wed Feb 13 13:13:10 2002
Memory used by data messages: 67.535 K(Bytes)
Total messages passed through: 664 Message(s)
Number of message queues: 5
Number of connections: 7
Port number: 24055
```

6.3.4 Viewing All Topics for a Server

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

6.3.5 Changing Topic Contents

To change the contents of a topic, you use the **-changetmsg** flag. You must specify the message type as bytes or text. You cannot change the original message type when you change the contents.

To change topic 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 name, where *seqnumber* is the sequence number of the topic, and where *type* is either bytes or text indicating the original message type.

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

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

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

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

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

6.3.6 Viewing Statistics for Topic “Broadcast”

```
stcmsctrlutil -host localhost -port 24055 -topicstat Broadcast
Topic Name: Broadcast
First sequence number: 0
Last sequence number: 5
First enqueue time: 05172001:16:30:30
Last enqueue time: 05172001:16:30:42
Number of current subscribers: 0
Number of total subscribers: 2
Message count: 6
Lowest subscriber sequence: 0
Highest subscriber sequence: 3
```

6.3.7 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
```

6.3.8 Viewing Properties of All Subscribers to Topic “STC”

```
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
```

6.3.9 Viewing All Queues for a Server

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

6.3.10 Displaying Statistics for Queue “PTP”

```
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
```

6.3.11 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
        Uncommited messages: 0
Receiver ID: 14291939
        Queue name: PTP
        Session ID: 4
        Committed messages: 432
        Uncommited messages: 1
```

6.3.12 Viewing Properties of All Receivers of Queue “PTP”

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

```
Receiver ID: 14291939
        Queue name: PTP
        Session ID: 4
        Committed messages: 432
        Uncommitted messages: 1
```

6.3.13 Republishing Messages from Topics

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

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

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

6.3.14 Republishing Messages from Queues

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

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

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

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

6.3.15 Browsing Journaled Events

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

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

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

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

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

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

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

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

```
stcmsctrlutil -j -qs Q0
Queue Name: Q0
First enqueue time: 09122002:00:14:07
Last enqueue time: 09122002:00:14:28
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
```

6.3.16 Backing Up

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

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

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

```
stcmsctrlutil -backup c:\eGate\client\Archiver\Ar09112002.zip "09/11/
2002"
Backup finished. Archived messages: 2002
```

6.3.17 Archiving

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: 09122002:00:14:17
Last enqueue time: 09122002:00:14:00
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
```

6.3.18 Setting Timeout

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

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

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

6.4 stcmsutil.Properties

The **stcmsutil.Properties** file is located in the **bin** directory in the location where the e*Gate Participating Host is installed. This file holds current settings and preferences for the MS Control utility. Table 9 shows the property names and their preset values if never modified.

Table 9 stcmsutil.Properties - Message Service Utility Property Names and Values

Property	Preset value	Description
Util.Client.Library	stc_msclient.dll	Specifies the filename of the Dynamic Link Library (DLL) file used by the C++ client.
stcmsctrlutil.Trace.ToStdout	No	Specifies whether **stcmsctrlutil.exe** writes messages to standard output (usually the console).
stcmsctrlutil.Trace.Verbose	No	Specifies whether trace messages written by **stcmsctrlutil.exe** provide a lengthy full explanation.
stcmsctrlutil.Trace.ToFile	Yes	Specifies whether trace messages written by **stcmsctrlutil.exe** are written to a file. Also see **stcmsctrlutil.Trace.LogFile**.
stcmsctrlutil.Trace.Level	0	Specifies threshold severity level at which **stcmsctrlutil.exe** issues informational, warning, and error messages: ▪ **0** Informational messages (in addition to all three categories below). ▪ **1** Warning messages (in addition to both categories below). ▪ **2** Error messages (in addition to the category below). ▪ **3** Fatal messages.
stcmsctrlutil.Trace.Performance	No	Specifies whether performance level tracing is turned on.
stcmsctrlutil.Trace.Memory	No	Specifies whether memory level tracing is turned on.
stcmsctrlutil.Trace.TimeStamp	No	Specifies whether **stcmsctrlutil.exe** prints timestamps in the trace log file.
stcmsctrlutil.Trace.LogFile	../logs/stcmsctrlutil.log	Specifies the path and file name where trace messages written by **stcmsctrlutil.exe** are logged. Meaningful only when **stcmsctrlutil.Trace.ToFile** = **Yes**.

Chapter 7

Using the ICAN Suite with WebLogic Through JMS

You can use Oracle SeeBeyond JMS IQ Service to allow ICAN Suite for SRE products to operate with Oracle WebLogic Server.

In This Chapter:

7.1 Connecting to WebLogic Through Oracle SeeBeyond JMS

The ICAN SUite for SRE provides connectivity to Oracle WebLogic Server through the Oracle SeeBeyond JMS IQ Service. Perform the following steps to connect to the WebLogic server.

- **Preparing the e*Gate Schema** on page 79
- **Setting up the ICAN Suite Interface** on page 80
- **Configuring JNDI Bindings to the JMS Server** on page 81
- **Configuring the WebLogic Server** on page 82
- **Create the WebLogic MDB** on page 83

7.1.1 Preparing the e*Gate Schema

Using WebLogic with the ICAN Suite using JMS requires the following changes in the e*Gate Schema Manager:

- The Participating Host network hostname must be the actual hostname and not 'localhost'. Verify this under Participating Hosts in the Schema Designer Explorer panel.
- If a JMS IQ Manager does not exist, create one. For instructions, see **Creating JMS IQ Managers** on page 23.

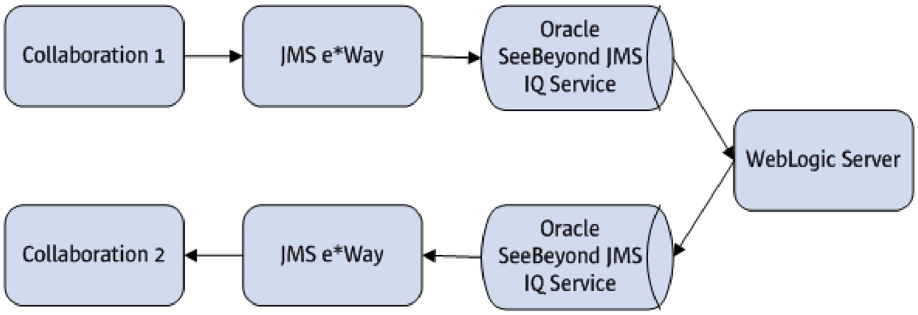
7.1.2 Setting up the ICAN Suite Interface

You can use either JMS e*Way Connections or a JMS IQ interface to transfer data from the Collaborations to the WebLogic server and back again. The following sections describe both scenarios.

Using JMS e*Way Connections

To use JMS e*Way Connections to provide connectivity with the JMS IQ Service, you need to create the Java Collaboration rules and corresponding Collaborations to publish to or subscribe from the JMS e*Way Connection. A typical scenario might look like Figure 6.

Figure 6 WebLogic Scenario Using JMS e*Way Connections



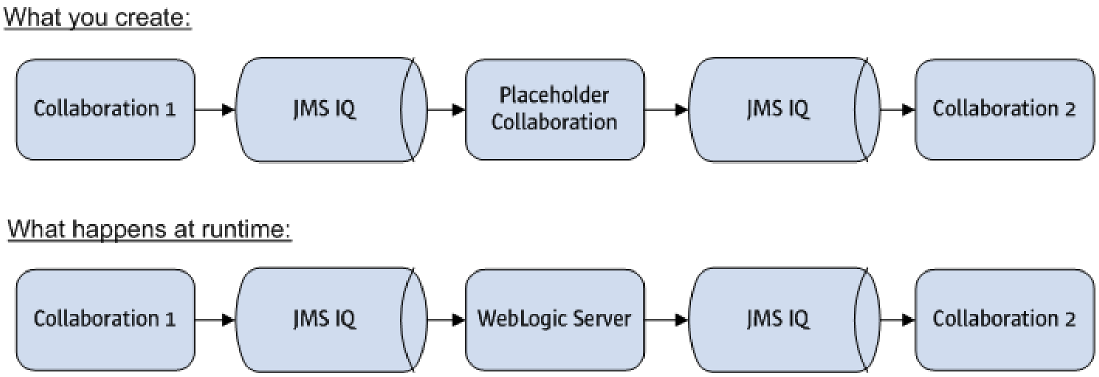
You can use either a JMS topic or JMS queue with JMS e*Way Connections. A message driven bean running on WebLogic can receive messages from Collaboration 1 through the e*Way and JMS service, and then send message back to Collaboration 2 through the JMS service and e*Way. Messages transferred to and from JMS can be either text messages or bytes messages.

For more information about JMS e*Way Connections, see **Chapter 4**, **"Configuring JMS e*Way Connections" on page 40**.

Using the JMS IQ Interface

To use the JMS IQ interface, you need to create the JMS IQ. You also need to create Collaborations that publish to and subscribe from the JMS IQ, along with a Collaboration that is simply a placeholder for the component running on the WebLogic server. At runtime, the component that hosts the placeholder Collaboration is not started and the WebLogic component assumes the identity of the placeholder.

Figure 7 WebLogic Scenario Using a JMS IQ Interface



This configuration only uses JMS queue and not JMS topics. The names of the JMS queues will be the name of the Event Type published by Collaboration 1 and the name of the Event Type subscribed to by Collaboration 2. A message driven bean running on WebLogic can receive messages from Collaboration 1 through first queue, and then send message back to Collaboration 2 through the second queue. Messages transferred to and from JMS can only be bytes messages.

To create the JMS IQ

1. In the schema explorer panel of the e*Gate Schema Designer, expand **Participating Hosts** until you see the IQ Manager you created to use with WebLogic.
2. Select the IQ Manager, and then click **Create a New IQ** (the button in the middle of the Schema Designer).
3. On the **New IQ Component** dialog box, enter a name for the IQ and then click **OK**.

   The new JMS IQ appears in the right panel of the Schema Designer.
4. Right click the new IQ and then select **Properties**.

   The IQ Properties window appears.
5. Configure the IQ as you would normally, but under **IQ Behavior** on the **Advanced** tab, select **Subscriber Pool**.

7.1.3 Configuring JNDI Bindings to the JMS Server

You need to configure JNDI bindings for the WebLogic component to use. To do this, you first need to configure e*Gate Integrator for the JNDI service provider, and then you can create the bindings.

Configuring e*Gate Integrator for JNDI Binding for WebLogic Interoperability

Before you use Schema Manager to create, remove, and view JNDI bindings, you must configure e*Gate for your JNDI service provider. You can download the .jar files as part of the Java Web Services Developer Pack (http://java.sun.com/webservices/downloads/previous/webservicespack.jsp).

For complete instructions, see **"Configuring e*Gate for JNDI Binding" on page 53**.

Creating JNDI Bindings for WebLogic Interoperability

In this step, you bind a connection factory or factories for the WebLogic component to use. If the WebLogic component uses JMS Topics, you need to bind a TopicConnectionFacgtory object; if the component uses queues, you need to bind a Queue ConnectionFactory object. In addition, if the WebLogic component is transactional, you need to select an XA connection factory object to bind.

For complete instructions, see **"Binding STC Connection Factories, Topics, and IQs" on page 54**. Before beginning, start the Control Broker, Schema Manager, JMS IQ Manager, and other associated schema components. Then launch the JMS Administrator.

7.1.4 Configuring the WebLogic Server

When you created the JNDI bindings, the binding information was written to a file named **.bindings** in the *bindingpath* you specified when you configured e*Gate Integrator. You will copy this and other files to the WebLogic server when you configure the server.

To configure the WebLogic Server

1 Copy the following files to the WebLogic server:

- Copy the **.bindings** file to the system where the WebLogic server is running.
- Copy the **fscontext.jar**, **providerutil.jar**, and **stcjms.jar** files from **<eGate>\client\classes** to the WebLogic domain **\lib** directory; for example:

```
C:\Oracle\Middleware\user_projects\domains\domain1\lib
```

Note: *You need to add the last three files to the WebLogic Server's classpath if you copied them to a different directory than the domain \lib directory.*

2 Restart the WebLogic server.

3 Create a new JMS System Module:

A Log in to the WebLogic Admin Console, and go to Services > Messaging > JMS Modules.

B Create a new JMS System Module, and select the AdminServer or any other server as the target.

C Select the newly created JMS System Module, and select **New** to create a new resource.

D Select **Foreign Server** from the list and click **Next**.

E Enter a name, click **Next**, and then click **Finish**.

F Select the Foreign JMS Server you created, and set the value for the JNDI Initial Context Factory as follows:

```
com.sun.jndi.fscontext.RefFSContextFactory
```

G Set the JNDI Connection URL as follows, replacing <path_to_binding_file> with the actual path to the file you copied earlier:

```
file://<path_to_binding_file>
```

For example:

```
file://C:/JNDI/
```

H Click **Save**.

4 Create connection factories:

A Click the Connection Factories tab.

B Click **New**, and enter a name for the factory.

C Enter the following JNDI names:

- **Local JNDI Name**: The name the WebLogic component will use for lookup. To avoid confusion, use the same name as the Remote JNDI Name.
- **Remote JNDI Name**: The JNDI name used when you created the JNDI Bindings for Connection Factories using the JMS Administrator.

D Repeat the above steps to create any additional connection factories.

5 Create destinations for the topics and queues:

A Click the Destination tab and click **New**.

B Enter the following properties:

- **Name**: A unique name for the destination.
- **Local JNDI Name**: The name the WebLogic component will use for lookup. To avoid confusion, use the same name as the Remote JNDI Name.
- **Remote JNDI Name**: The JNDI name used when you created the JNDI Bindings for Topics and Queues using the JMS Administrator.

7.1.5 Create the WebLogic MDB

This section provides an example of how to create an MDB to communicate with the Oracle SeeBeyond JMS Service. Your actual implementation would be based on your specific requirements.

To create the WebLogic MDB

1 In a temporary directory create a **meta-inf** subdirectory.

2 In the **meta-inf** subdirectory, create a Message Driven Bean descriptor named **ejb-jar.xml**, and create a WebLogic EJB descriptor named **weblogic-ejb-jar.xml**.

3 In the Message Driven Bean descriptor, **ejb-jar.xml**, specify whether it receives messages from a Topic or a Queue. For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<ejb-jar xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xmlns:ejb="http://
java.sun.com/xml/ns/javaee/ejb-jar_3_0.xsd"
xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://
java.sun.com/xml/ns/javaee/ejb-jar_3_0.xsd" version="3.0">
```

```
   <display-name>SREWLInterOP</display-name>
   <enterprise-beans>
      <message-driven>
         <ejb-name>HelloMDB</ejb-name>
         <ejb-class>HelloMDB</ejb-class>
         <transaction-type>Container</transaction-type>
         <messaging-type>javax.jms.MessageListener</messaging-type>
         <message-destination-type>javax.jms.Queue
         </message-destination-type>
      </message-driven>
   </enterprise-beans>
   <assembly-descriptor>
      <container-transaction>
         <method>
            <ejb-name>HelloMDB</ejb-name>
            <method-name>*</method-name>
         </method>
         <trans-attribute>Required</trans-attribute>
      </container-transaction>
   </assembly-descriptor>
</ejb-jar>
```

4 In the Weblogic EJB descriptor, **weblogic-ejb-jar.xml**, specify the incoming JNDI name for the Topic or Queue and the JNDI name for the connection factory. These are the local JNDI names you used when configuring the Weblogic server. For example:

```
<?xml version="1.0"?>
<!DOCTYPE weblogic-ejb-jar PUBLIC "-//BEA Systems, Inc.//DTD WebLogic
7.0.0 EJB//EN" "http://www.bea.com/servers/wls700/dtd/weblogic-ejb-
jar.dtd">
<weblogic-ejb-jar>
   <weblogic-enterprise-bean>
      <ejb-name>HelloMDB</ejb-name>
         <message-driven-descriptor>
            <pool>
               <max-beans-in-free-pool>8</max-beans-in-free-pool>
               <initial-beans-in-free-pool>1
               </initial-beans-in-free-pool>
            </pool>
         <destination-jndi-name>STCMS_GenericInEvent
         </destination-jndi-name>
         <connection-factory-jndi-name>STCMS_XAQueueFactory
         </connection-factory-jndi-name>
      </message-driven-descriptor>
   </weblogic-enterprise-bean>
</weblogic-ejb-jar>
```

5 In the temporary directory, write the Message Driven Bean code. Following is a sample for **HelloMDB.java**:

```
import javax.jms.BytesMessage;
import javax.jms.Destination;
import javax.jms.Message;
import javax.jms.MessageListener;
import javax.jms.Session;
import javax.jms.TextMessage;
import javax.jms.Queue;
import javax.jms.QueueConnection;
import javax.jms.QueueConnectionFactory;
import javax.jms.QueueSender;
import javax.jms.QueueSession;
import javax.naming.Context;
import javax.naming.InitialContext;
public class HelloMDB implements MessageListener
{
   public HelloMDB()
   {
   }
   private static String DEST_LOOKUP_NAME = "STCMS_Topic2";
   private static String CF_LOOKUP_NAME = "STCMS_XAQueueFactory";
   String PubID = "{65905790-6851-11DF-A48C-94018F17D48D}";

   public void onMessage(Message m)
   {
     String textmessage = "";
     Destination dest = null;

       try {
           if(m instanceof TextMessage) {
               textmessage = ((TextMessage)m).getText();
               System.out.println("Received TextMessage "+textmessage);
           } else if (m instanceof BytesMessage) {
               BytesMessage bmsg = ((BytesMessage)m);
               int len = (int)bmsg.getBodyLength();
               byte[] bytes = new byte[len];
               bmsg.readBytes(bytes);
               textmessage = new String(bytes);
               System.out.println("Received BytesMessage "+textmessage);
           } else {
               textmessage = m.toString();
               System.out.println("Receive Non text/byte Message "+textmessage);
           }
           dest = m.getJMSReplyTo();
           if(dest != null) {
              System.out.println(“Message specified ReplyTo Dest "+dest);
           }

           Context context = new InitialContext();
           System.out.println(" Looking up connection factory and destination. ConnectionFactory:
              "+CF_LOOKUP_NAME+" destination : "+DEST_LOOKUP_NAME);
           if(dest == null)
              dest = (Destination) context.lookup(DEST_LOOKUP_NAME);
              QueueConnectionFactory cf = (QueueConnectionFactory)context.lookup(CF_LOOKUP_NAME);
              QueueConnection conn = cf.createQueueConnection(); conn.start();

              QueueSession session = conn.createQueueSession(true, Session.AUTO_ACKNOWLEDGE);
              QueueSender producer = session.createSender((Queue)dest);

              BytesMessage msg1 = session.createBytesMessage();
              byte [] msg_bytes = ("From HelloMDB "+textmessage).getBytes();
              msg1.writeBytes(msg_bytes);
              msg1.setStringProperty("PUBLISHER_UUID", PubID);
              msg1.setStringProperty("MESSAGE_LENGTH", msg_bytes.length+"");
              producer.send(msg1);
              producer.close();
              session.commit();
              System.out.println(" Message Sent to "+dest);
              try {
                  session.close();
                  conn.close();
              } catch(Throwable ignore){
                  ignore.printStackTrace();
              }

       } catch(Throwable e) {
          e.printStackTrace();
       }
   }
   public static void main(String args[]){
      System.out.println("This is MDB ");
   }
}
```

JMS e*Way Connections can handle both TextMessage and BytesMessage, so you can let the MDB send either one when using JMS e*Way Connections to communicate with WebLogic.

On the other hand, the SRE JMS IQ Interface can only handle BytesMessage, so the MDB can only send BytesMessage when using the JMS IQ Interface. For the IQ Interface you need to set two string properties of a message before sending the message out. The two string properties are PUBLISHER_UUID and MESSAGE_LENGTH. In the above example, these two lines set the properties:

```
msg1.setStringProperty("PUBLISHER_UUID", PubID);
msg1.setStringProperty("MESSAGE_LENGTH", msg_bytes.length+"");
```

The PUBLISHER_UUID can be found by exporting the schema using the **stcregutil** utility with the following syntax:

```
stcregutil -rh <host> -rs <schema name> -un <user name> -up
<password> -bu -e <export file name>
```

For example:

```
stcregutil -rh localhost -rs file2jms -un Administrator -up STC
-bu -e file2jms.txt
```

Open the exported file with a text editor and look for the placeholder Collaboration name in the REGISTRY_PUBLISHER section. A string similar to this {65905790-6851-11DF-A48C-94018F17D48D} following the collaboration name is the PUBLISHER_UUID.

6. Compile the MDB code. For example:

   ```
   javac HelloMDB.java
   ```

7. Create the MDB .jar file. For example:

   ```
   jar cvf HelloMDB.jar HelloMDB.class META-INF
   ```

8. Deploy the MDB to Weblogic.

Index