SeeBeyond™ eBusiness Integration Suite

SeeBeyond JMS Intelligent Queue User's Guide

Release 4.5.2

Version 20020513113622.

Contents

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.

1.1 Document Purpose and Scope

This guide explains how to use the SeeBeyond Technology Corporation™ (SeeBeyond™) 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 SeeBeyond JMS IQ Manager, e*Way Connection, and IQ components that constitute the implementation.

Important: *Any operation explanations given here are generic, for reference purposes only, and do not necessarily address the specifics of setting up and/or operating individual e*Gate systems.*

This document does not contain information on installation or general e*Gate configuration. For information on these and related topics, see **"Supporting Documents" on page 11**.

1.2 Intended Audience

The reader of this guide is presumed to be a Java developer or system administrator with responsibility for maintaining the e*Gate system. You should have moderate knowledge of the Java programming language, good familiarity with the Java Message Service API, and thorough familiarity with Windows-style GUI operations.

1.3 Organization of Information

This document is organized topically as follows:

- **Chapter 1 "Introduction" on page 8**gives a general preview of this document, its purpose, scope, and organization.
- **Chapter 2 "SeeBeyond JMS IQ Service Overview" on page 12**provides an introduction to the SeeBeyond JMS implementation, as well as a summary of the feature's use, operation, and prerequisites.
- **Chapter 3 "SeeBeyond JMS IQ Manager Setup" on page 15**explains how to configure the SeeBeyond JMS IQ Manager component.
- **Chapter 4 "JMS e*Way Connection Configuration" on page 30**explains how to configure the e*Way Connection component to operate with the SeeBeyond JMS IQ Service.
- **Chapter 5 "SeeBeyond JMS IQ Component" on page 38**explains how to configure the SeeBeyond JMS IQ component to operate with the SeeBeyond JMS IQ Service.
- **Chapter 6 "SeeBeyond JMS IQ Manager Administration" on page 40**explains how to use the JMS Administrator to monitor topics and queues managed by the SeeBeyond JMS IQ Service.
- **Chapter 7 "STC MS Control Utility" on page 50** lists and describes the **stcms*.exe** command-line utilities that can be used to query and control the SeeBeyond JMS IQ Service.

1.4 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 10**.

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.

1.5 Supporting Documents

The following SeeBeyond 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*
- *SeeBeyond eBusiness Integration Suite Primer*
- *SeeBeyond eBusiness Integration Suite Deployment Guide*
- *Standard e*Way Intelligent Adapter User's Guide*

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

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

1.6 SeeBeyond Web Site

The SeeBeyond Web site is your best source for up-to-the-minute product news and technical support information. The site's URL is

http://www.SeeBeyond.com

Chapter 2

SeeBeyond JMS IQ Service Overview

This chapter provides a general overview of the SeeBeyond JMS IQ Service: how it operates, how it compares with the 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 SeeBeyond JMS IQ Service, which is compliant with JMS version 1.0.2, provides all the standard capabilities of 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 a SeeBeyond Standard IQ service can publish or subscribe in exactly the same way to exactly the same IQ managed by the SeeBeyond JMS IQ Service.

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

- 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 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 SeeBeyond Standard IQ Services and SeeBeyond JMS IQ Services.

Figure 1 Architecture Using Standard IQ Manager

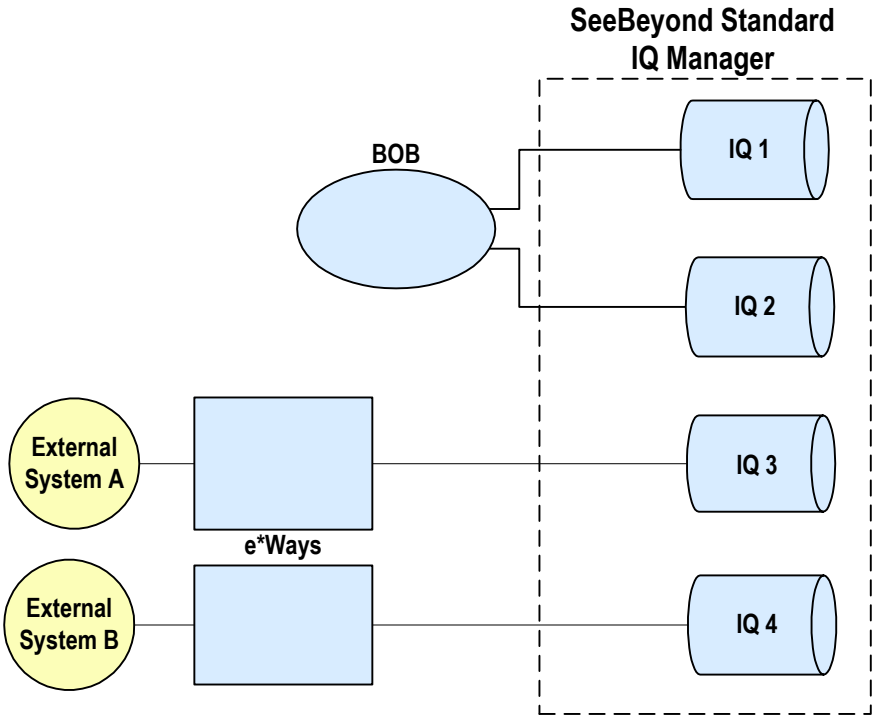
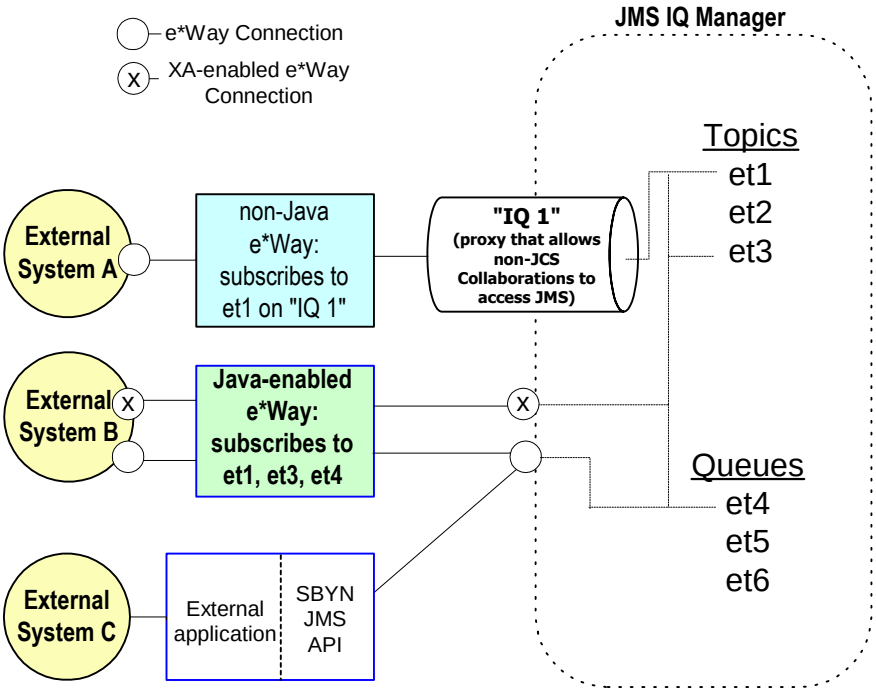


Figure 2 Architecture Using SeeBeyond JMS IQ Manager



2.2 System Requirements

The SeeBeyond JMS IQ Service is available on the following operating systems:

- Windows 2000, Windows 2000 SP1, and Windows 2000 SP2
- Windows NT 4.0 SP6a
- Solaris 2.6, 7, and 8
- AIX 4.3.3 and AIX 5.1
- HP-UX 11.0 and HP-UX 11i
- Compaq *Tru64* UNIX V4.0F and V5.0A
- Red Hat Linux 6.2
- Japanese Windows 2000, Windows 2000 SP1, and Windows 2000 SP2
- Japanese Windows NT 4.0 SP6a
- Japanese Solaris 2.6, 7, and 8
- Japanese HP-UX 11.0
- Korean Windows 2000, Windows 2000 SP1, and Windows 2000 SP2
- Korean Windows NT 4.0 SP6a
- Korean Solaris 8
- Korean AIX 4.3.3
- Traditional Chinese (BIG5) Windows 2000, Windows 2000 SP1, and Windows 2000 SP2
- Traditional Chinese (BIG5) Windows NT 4.0 SP6a
- Traditional Chinese (BIG5) Solaris 8

Note: *The SeeBeyond JMS IQ Service is* ***not*** *available on OS/390 at this time.*

To use the SeeBeyond JMS IQ Service, you need the following:

- An e*Gate Participating Host, version 4.5 or later. For AIX operating systems, you need an e*Gate Participating Host, version 4.5.1 or later.
- A TCP/IP network connection.

2.3 For Further Information

The *e*Gate API Kit Developer's Guide* contains complete information on architecture and APIs, as well as in-depth information on such special topics as the Java and COM+ Clients, the Compensating Resource Manager (CRM), and the Message Selector. 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

SeeBeyond JMS IQ Manager Setup

This chapter describes how to create and configure a SeeBeyond JMS IQ Manager.

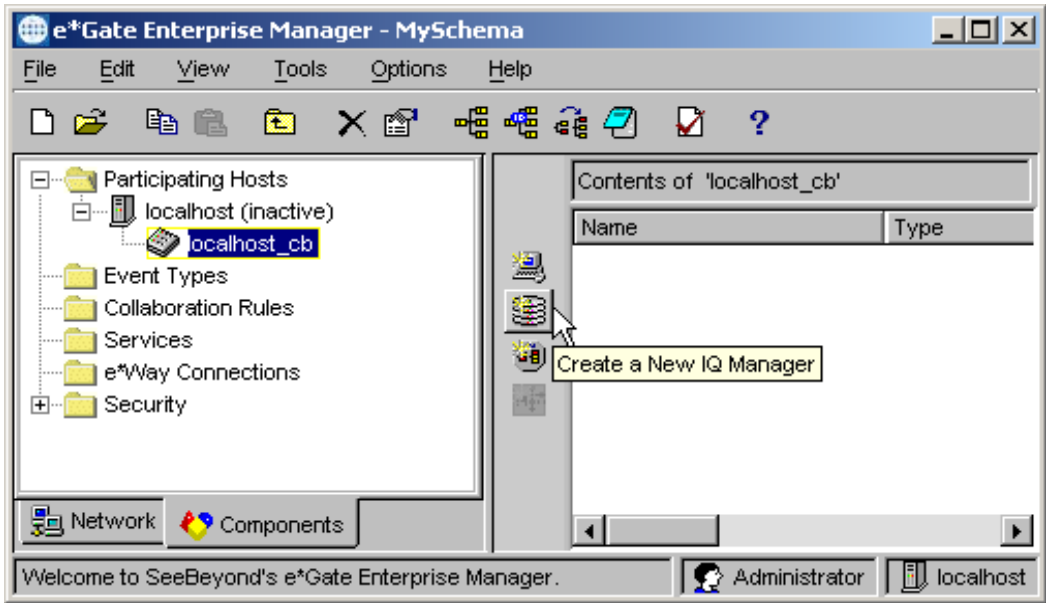
3.1 SeeBeyond JMS IQ Manager

Use the Enterprise Manager to set up or modify IQ Managers.

To create and configure a new SeeBeyond JMS IQ Manager

1. In the Enterprise Manager's Navigator pane, click the **Components** tab (if necessary) and click the Control Broker of the Participating Host where you want to create the new IQ Manager.
2. In the Palette, click **Create a New IQ Manager**. See Figure 3.

Figure 3 Create a New IQ Manager



3. In the **New IQ Manager Component** dialog box, enter the name of the new IQ Manager and then click **OK**.

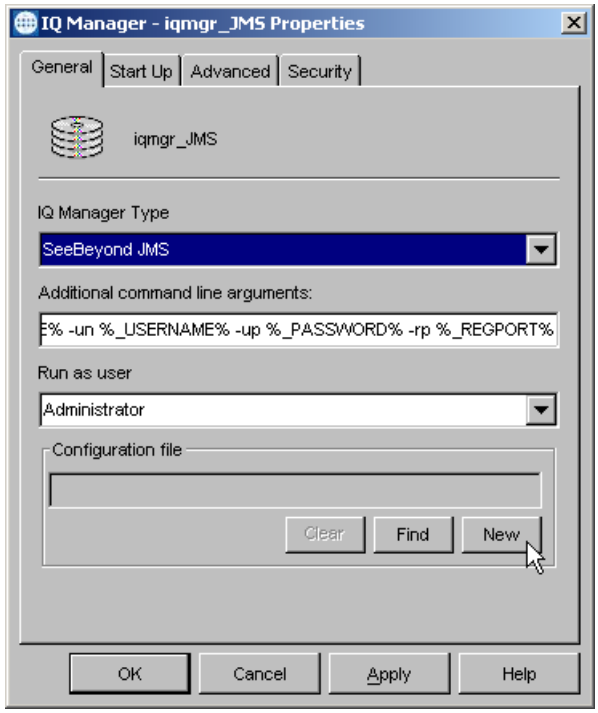
   The new IQ Manager appears in the component pane.

4 In the component pane, select the new IQ Manager and then edit its properties.

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

5 If necessary, in the **IQ Manager Type** list, click **SeeBeyond JMS**. See Figure 4.

Figure 4 IQ Manager Properties - General Tab



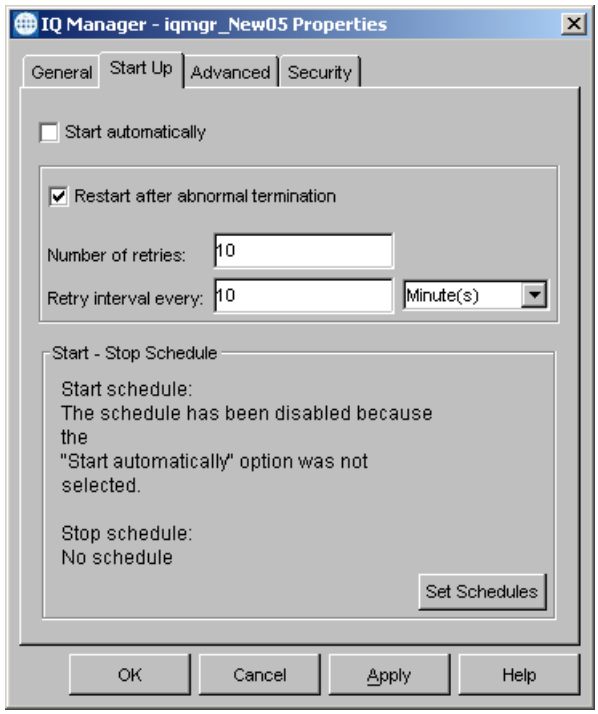
6 Under **Configuration File**, click **New**. When the **Edit Settings** dialog box appears, set the configuration parameters for this configuration file. Parameters are listed and explained in **"Overview of Server Operation and Configuration" on page 19**.

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

7 After setting the configuration parameters and adding user notes, save the **.cfg** file (keeping the same name as the name of the IQ Manager and accepting the default location), and close the **Edit Settings** dialog box.

8 Click the **Start Up** tab. See **Figure 5 on page 17**.

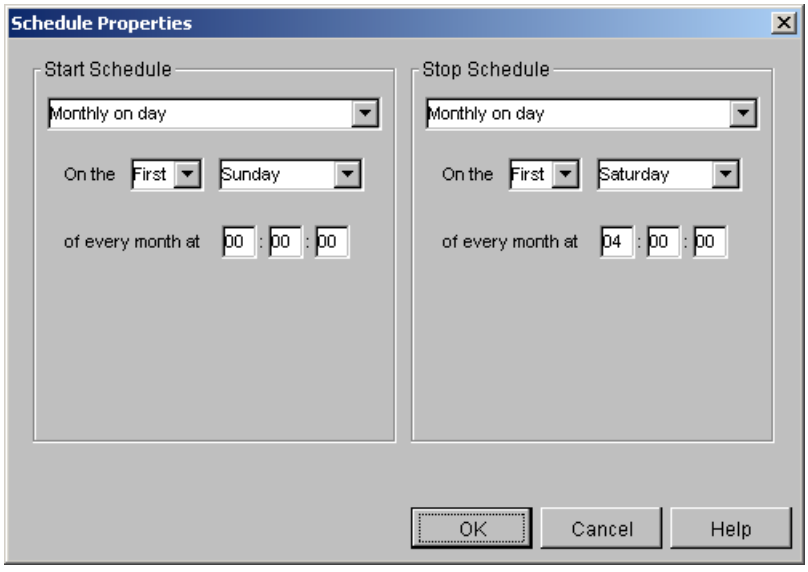
Figure 5 IQ Manager Properties - Start Up Tab



9 If you want this IQ Manager to stop and start on a schedule you define, click the **Start automatically** check box; otherwise, leave it unchecked.

10 If you selected the **Start automatically** check box, you can also specify a start and stop schedule by clicking **Set Schedules**. See Figure 6.

Figure 6 IQ Manager Properties - Start Up - Set Schedules



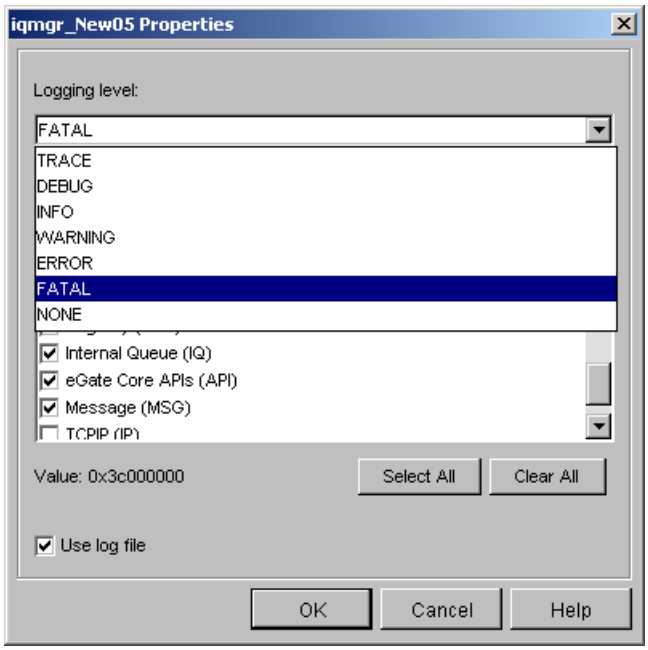
11 If you want this IQ Manager to require manual restart after an abnormal halt, clear the **Restart after abnormal termination** check box; otherwise, leave it checked.

- By default, if this check box is checked, the system will attempt a total of ten retries, spacing the attempts at ten-minute intervals. You can increase or decrease the number of retries and the interval length.

12 To view or set the activity logging level and error logging level, click the **Advanced** tab, click **Log**, and then select each of the logging levels (of TRACE, DEBUG, INFO, WARNING, ERROR, FATAL, and NONE) and set the debugging flags for that level. See Figure 7.

Note: *For complete information on logging levels, debug flags, and activating logging for any component, see the* ***e*Gate Integrator Alert and Log File Reference Guide****.*

Figure 7 IQ Manager Properties - Advanced - Log



13 The Advanced tab also allows you to set the TCP/IP port number to a value other than the default (24053).

14 When you have finished setting options in the **Advanced** tab, click **Apply**.

15 To view or set privilege assignments, click the **Security** tab, click **Privilege**, and then add roles and assign privileges as needed. When you have finished, click **OK**.

Note: *For complete information on assigning privileges and associating privileges with roles, see the* ***e*Gate Integrator System Administration and Operations Guide****.*

16 In the **IQ Manager Properties** dialog box, click **OK** to apply the changes, close the dialog box, and return to Enterprise Manager.

3.2 Overview of Server Operation and Configuration

Terminology

The SeeBeyond JMS IQ Manager is sometimes also called the STC Message Server, or simply the *Server*. The prefix **stcms** refers to the SeeBeyond JMS IQ Manager.

3.2.1 SeeBeyond JMS IQ Manager (Server) Operation

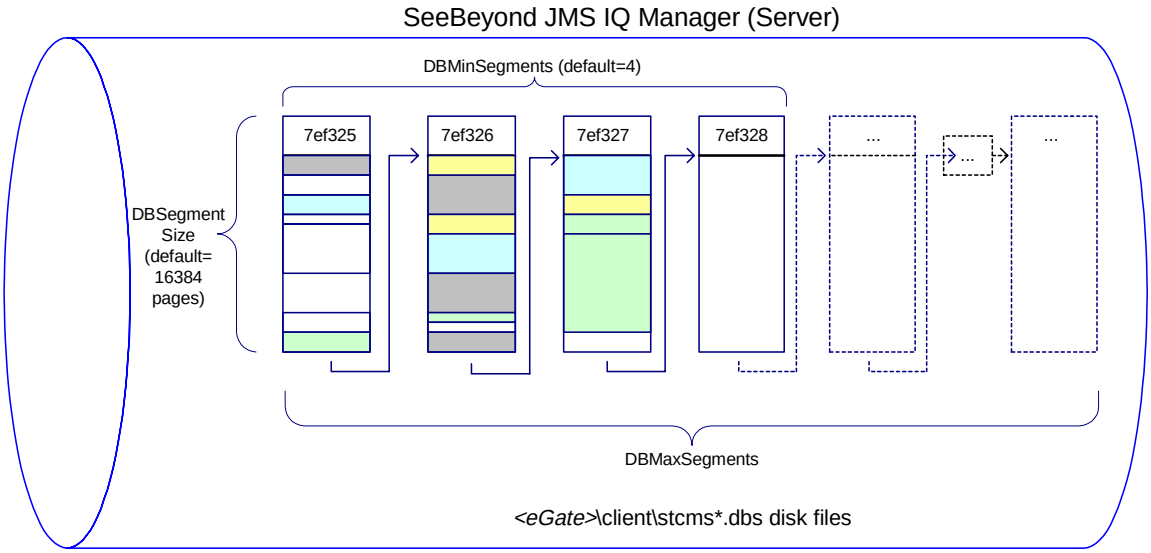
On startup, the Server performs the following sequence of operations:

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

   In Figure 8, the Server allocated 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 8 shows that the third segment, 7ef327 in file **stcms73f327.dbs**, is almost full.

Figure 8 SeeBeyond JMS IQ Manager Operation



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 8, the first segment (7ef325) is almost ready for cleanup.

3.2.2 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 SeeBeyond JMS IQ Manager, open its configuration file, and access its DB Settings section; see **"DB Settings" on page 23**.

3.2.3 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 21**.

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.

3.2.4 Managing Resources

As for any server, there are configuration parameters purely for managing memory (such as MaxPayloadMemory and PayloadMemoryPad) and memory/disk tradeoffs (such as DBCacheSize for controlling disk write memory cache, DBSync on UNIX for increasing speed over reliability, and DBCacheIntoRAM on Windows for reducing page faults). In addition, for the 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 "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 commonest 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.*

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.

3.3 Configuration Parameters and Their Settings

SeeBeyond JMS IQ Manager configuration parameters are organized into five sections: DB, Message, Action, Server, Topic, and Trace. Parameters for each section are described in detail below.

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 "SeeBeyond JMS IQ Manager."*

3.3.1 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).

Format: Use the **forward** slash (**/**) as a path delimiter.

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.

Be sure to 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 database system disk cache. A larger cache means better performance for more active data files.

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.*

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 will create and maintain for its persistent message store and transaction log. This effectively limits the amount of disk space that the Server will use. 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 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 21**.

Required Values

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

LockCacheIntoRAM

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.*

3.3.2 Message Settings

The **Message Settings** parameters govern message data memory and expiration settings on the SeeBeyond JMS IQ Manager (Server).

MaxPayloadMemory

Description

Specifies the maximum amount (in kilobytes) of message data payloads allowed to be in physical memory at any moment while the Server is running. When message data memory usage increases beyond the MaxPayloadMemory threshold, the Server will begin 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 smallish 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 message expires. After it expires, the message is permanently scratched.

Required Values

An integer between **0** and **999999999**. The default is **2592000** (in other words, 30*24*60*60 seconds = 30 days). The special value **0** means “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**.

3.3.3 Server Settings

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 "SeeBeyond JMS IQ Manager."*

The **Server Settings** parameter sets the upper limit on the total number of messages that the Server will track 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 21**.

Required Values

An integer between **0** and **999999999**. The default is **100000**. When set to **0**, publishers are never throttled.

3.3.4 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**.

3.3.5 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, you should set this 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**.

Chapter 4

JMS e*Way Connection Configuration

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

- Section 4.1 (**"Creating and Configuring JMS e*Way Connections" on page 31**) provides a step-by-step procedure that allows you to create, view, or edit a SeeBeyond JMS e*Way Connection.
- Section 4.2 (**"Configuration Parameters for JMS e*Way Connections" on page 33**) lists and describes the configuration parameters and their settings.

Overview: Advantages of JMS e*Way Connections

While the 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 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 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 Enterprise Manager to set up or modify SeeBeyond JMS e*Way Connections.

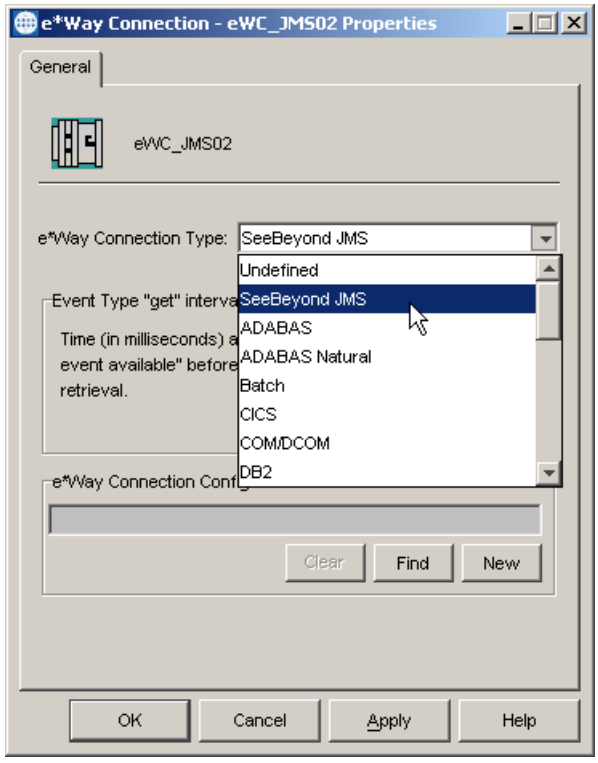
To create and configure a new SeeBeyond JMS e*Way Connection

1. In the Enterprise Manager'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**. See Figure 9.

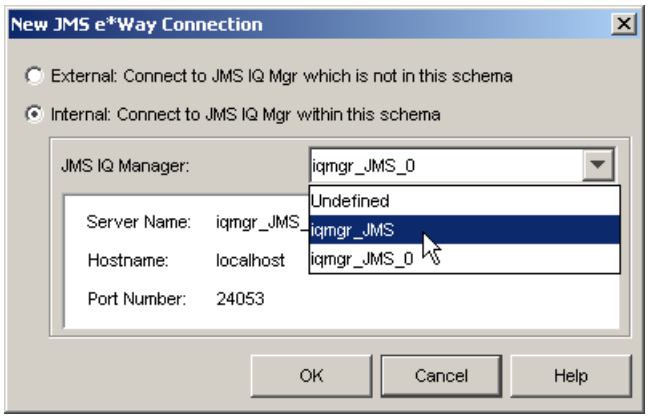
Figure 9 e*Way Connection Properties Dialog Box



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

   The **New JMS e*Way Connection** dialog box appears. See Figure 10.

Figure 10 New JMS e*Way Connection Dialog Box



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 33**.

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

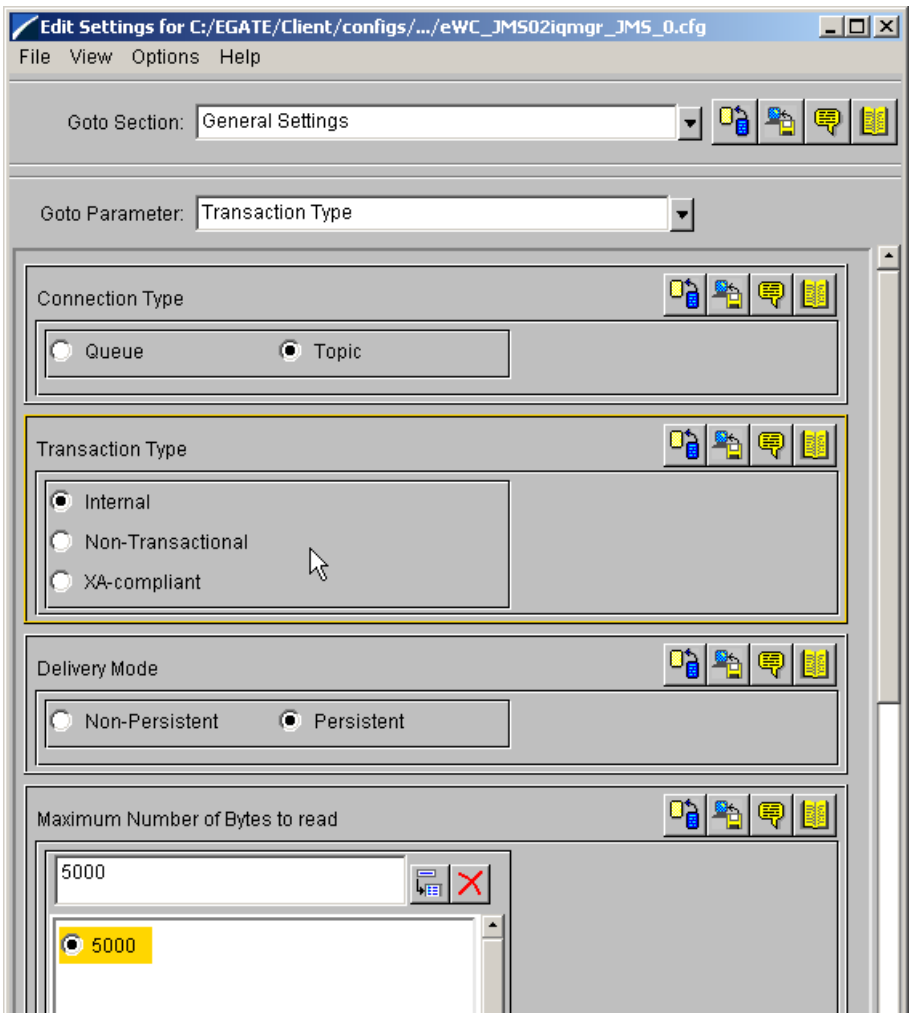
For SeeBeyond JMS e*Way Connections, configuration parameters are organized into two sections:

- **General Settings** on page 33
- **Message Service Settings** on page 36

4.2.1 General Settings

The General Settings control overall properties of the e*Way Connection. See Figure 11.

Figure 11 JMS e*Way Connection - General Settings



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 on 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

Default: (empty String)

Constraints: If supplied, the String must be a syntactically correct SQL92 expression of up to 512 bytes, including the ***NULL*** terminator.

Important:*This parameter does not check syntax. Therefore, if the string you supply contains syntax errors, the selector is ignored and the subscriber will not be created.*

Description

Allows you to specify a set of filtering criteria for messages; the only messages that will be delivered will be those for which the headers and properties match the selector. (Message selectors cannot reference message *body* values.) If a message selector is an empty String, its value is treated as null, indicating that there is no message selector for the message consumer.

A message selector is a String composed according to a subset of the SQL92 conditional expression syntax. The selector matches a message if and only if the SQL expression evaluates to Boolean ***TRUE*** when the message's header field and the property values are substituted for the corresponding identifiers within the selector. For example:

```
LName = 'Jones' OR (Age > 16 AND Weight <= 85.00) OR JMSPriority < 5
```

This example selects only messages where lastname=**Jones**, where age/weight criteria are met, or high-priority messages; the order of evaluation for message selectors is from left to right within precedence level, and parentheses can be used to change this order.

For more information about the SeeBeyond implementation of message selectors, see the *e*Gate API Kit Developer's Guide*, chapter 4 ("Implementing the Message Service"), especially section 4.2.4 ("The Message Selector"). For additional information, download **jms-1_0_2b-spec.pdf** (*Java™ Message Service Specification* version 1.0.2b) from

http://java.sun.com/products/jms/docs.html

and refer to section 3.8.1 ("Message Selection"). For more information on SQL92 syntax generally, see *X/Open CAE Specification Data Management: Structured Query Language (SQL), Version 2*, ISBN: 1-85912-151-9 March 1996.

Factory Class name

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

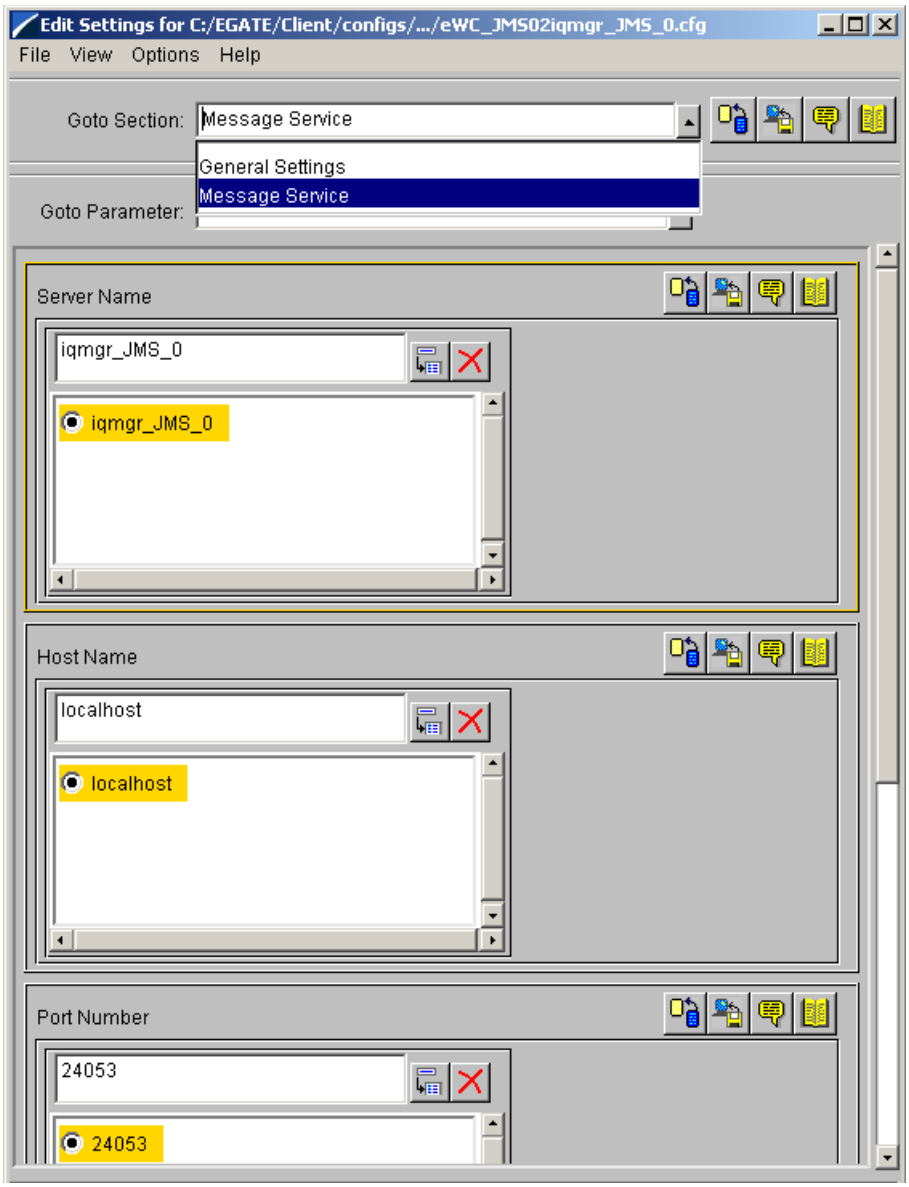
Description

For 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. See Figure 12.

Figure 12 JMS e*Way Connection - Message Service Settings



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

SeeBeyond JMS IQ Component

This chapter describes how to create and configure a SeeBeyond JMS IQ component—that is, a queue that uses the SeeBeyond JMS IQ Service.

Important: *Although IQ components are required by Monk Collaborations, they are not necessary for Java Collaborations. For Java Collaborations, you should use the SeeBeyond JMS e*Way Connection and allow the SeeBeyond JMS IQ Manager to maintain queuing internally. For details, refer to* **Chapter 4 "JMS e*Way Connection Configuration" on page 30**.

5.1 Creating and defining the IQ

For Monk Collaborations (see Note above), use the Enterprise Manager to create the new IQ component and define the IQ Service it uses.

To create a new IQ component

1. In Enterprise Manager, in the Navigator's **Components** tab, open the Participating Host and Control Broker containing the SeeBeyond JMS IQ Manager whose IQs you will be creating.
2. In the Navigator, select the JMS IQ Manager—in other words, select an IQ Manager whose IQ Service is set to **SeeBeyond JMS**—and then, in the Palette, click the **Create a New IQ** tool.
3. In the **New IQ Component** dialog box, type a name for the IQ and then click **OK**.

   The new IQ component appears in the Components Editor pane.

To set a new IQ component to use the SeeBeyond JMS IQ Service

1. In the Components Editor, double-click the newly created IQ.

   The **IQ Properties** dialog box appears. In the **General** tab, **Service** is preset to **STC_JMS_IQ**. This is the service used by all SeeBeyond JMS IQ components.
2. Do not enter anything for **Initialization string**; this box is reserved for future use.
3. **Event Type "get" interval**: Higher values yield higher polling frequency rates and the greater load on the IQ Manager, or enter a number lower than 100 to decrease the polling frequency and the load on the IQ Manager.
4. Do not select or clear **Auto Recovery**; this check box is reserved for future use.

5 For reference, the **Database** tab contains the following information:

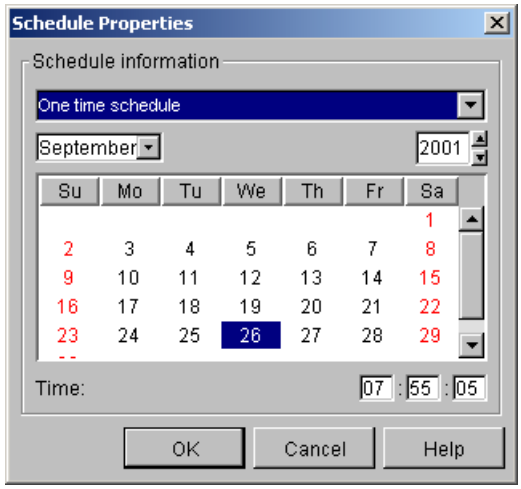
- **Hostname:** Echoes the name of the machine running the JMS Server.
- **Port number**: Echoes the port number set in the **IQ Manager Properties** dialog box, **Advanced** tab, in step **13 on page 18**.

6 Do not use the **External** tab to set or reset your configuration file unless you have been instructed to do so by SeeBeyond Support personnel.

7 Click the **Advanced** tab and make the following changes if necessary:

- If you want to unset or modify the IQ cleanup schedule, click **Set Schedule** and use the **Schedule Properties** dialog box to specify a frequency and to set exact date and/or times. See Figure 13.

Figure 13 IQ Component Properties - Advanced Tab - Schedule Properties



- In the Advanced tab, under **IQ behavior**, set the IQ behavior:
  - **Simple publish/subscribe** causes "broadcasting"—in other words, all IQs that subscribe to a topic (Event Type) receive all messages (Events) for that topic.
  - **Subscriber pool** causes "point-to-point" transmission—as messages (Events) are received for a particular topic (Event Type), they are received by only one subscriber in the pool (usually the subscriber that happens to be idle longest).

8 When you have finished setting the IQ component's properties, click **OK**.

In the component pane, the IQ component shows that it is set to use the STC_JMS_IQ IQ Service.

Chapter 6

SeeBeyond JMS IQ Manager Administration

This chapter describes how to use the JMS Administrator application to monitor the status of SeeBeyond JMS IQ Managers and their topics and queues.

6.1 Overview

The JMS Administrator is a special-purpose application accessed from e*Gate Monitor. Similar to IQ Administrator (formerly called "Queue Viewer"), the JMS Administrator provides real-time information on SeeBeyond JMS IQ Managers, queues, and topics.

6.1.1 Purpose and Features

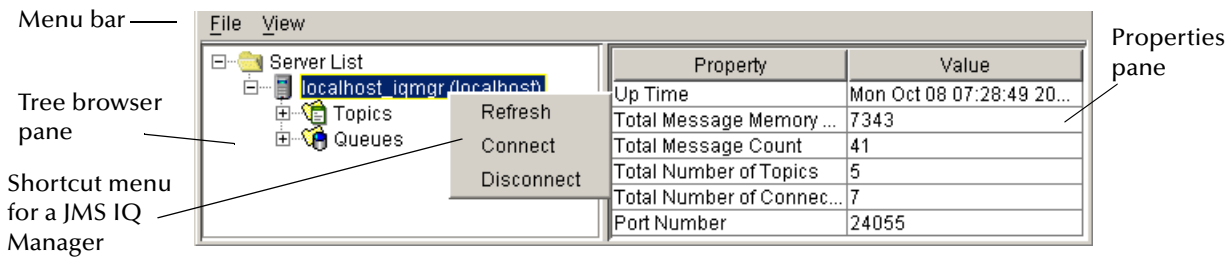
JMS Administrator displays information about the following items within a schema:

- SeeBeyond JMS IQ Managers, listed under **Server List**.
- Aggregate information on topics and queues, as properties of each IQ Manager.
- Individual Event Types: the **Topics** folder holds broadcast Event Types and their subscribers; the **Queues** folder holds point-to-point Event Types and their receivers.

6.1.2 Graphical User Interface (GUI)

At this release, JMS Administrator presents information in a simple two-pane layout: on the left, the tree browser allows you to navigate up or down to items of interest; on the right, the properties pane shows properties and values for the current selection. When you right-click an item in the tree browser, a shortcut menu appears, providing easy access to commands specific to that item. See Figure 14.

Figure 14 GUI Map for JMS Administrator



6.2 Using JMS Administrator

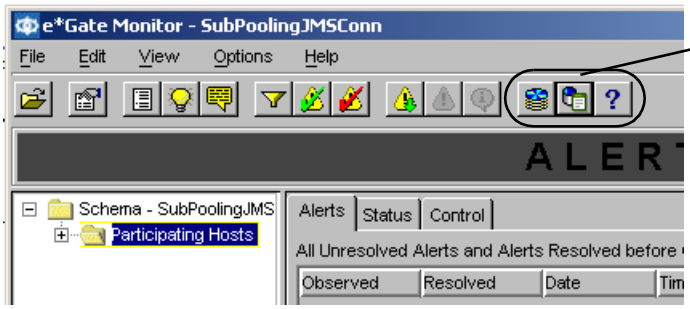
You access JMS Administrator via a toolbar button of e*Gate Monitor.

To start JMS Administrator

1 Start e*Gate Monitor and log in to the schema you want to monitor.

The toolbar buttons for the IQ Administrators are on the far right. See Figure 15.

Figure 15 The e*Gate Monitor Toolbar



2 On the toolbar, click JMS Administrator  .

The JMS Administrator appears. See **Figure 14 on page 40**.

6.2.1 Working With JMS IQ Managers (Servers)

The highest level of the tree browser is the **Server List** folder. You use the **Server List** folder to view existing JMS IQ Managers and add new ones. With each JMS IQ Manager in the list, you can connect to it, disconnect from it, or view its properties.

To list all JMS IQ Managers in the schema

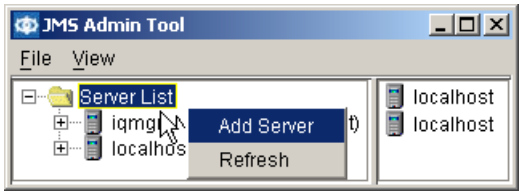
- In the tree browser, open the **Server List** folder.

The current JMS IQ Managers are listed.

To add a placeholder for a JMS IQ Manager

1 In the tree browser, right-click the **Server List** folder and click **Add Server** on the shortcut menu. See Figure 16.

Figure 16 Adding a Channel to a JMS IQ Manager



2 In the first **Input** dialog box, enter the name of the Participating Host of the JMS IQ Manager you want to add and then click **OK**.

3 In the second **Input** dialog box, enter the TCP/IP port number for this JMS IQ Manager and then click **OK**.

Note: *The TCP/IP port number of the Participating Host is set in the **Advanced** tab of the Properties dialog box for each JMS IQ Manager. For a JMS IQ Manager in the Up state, you can learn this information from e*Gate Monitor by clicking the **Control** tab.*

The tree browser displays the name of Participating Host you specified. However, it does not appear in the right pane until it is connected.

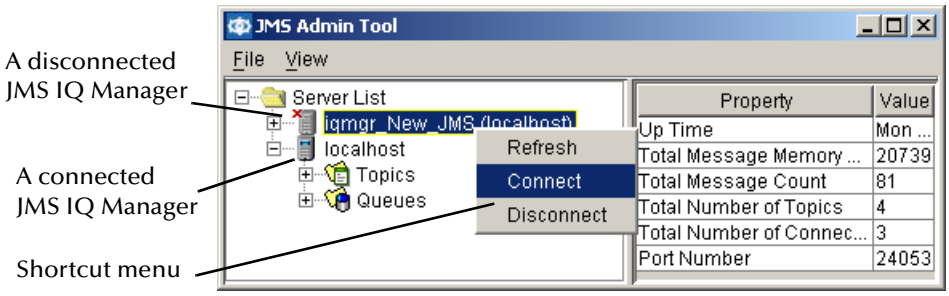
To refresh, connect to, or disconnect from a JMS IQ Manager

1 In the tree browser, under **Server List**, right-click the name of a Participating Host.

2 In the shortcut menu, do one of the following:

- Click **Refresh** to re-display the list with updated information (see Table 2 **"JMS IQ Manager (Server) Properties" on page 46**).
- Click **Connect** to connect to the selected JMS IQ Manager. See Figure 17.
- Click **Disconnect** to disconnect from the selected JMS IQ Manager.

Figure 17 Connecting to a JMS IQ Manager



The connected/disconnected state of each JMS IQ Manager is shown by its icon in the Server List: a small green square represents a connection; a small red X, a disconnection.

To view the properties of a JMS IQ Manager

- In the tree browser, open the **Server List** folder, right-click the name of the JMS IQ Manager you want to browse, and click **Refresh** on the shortcut menu.

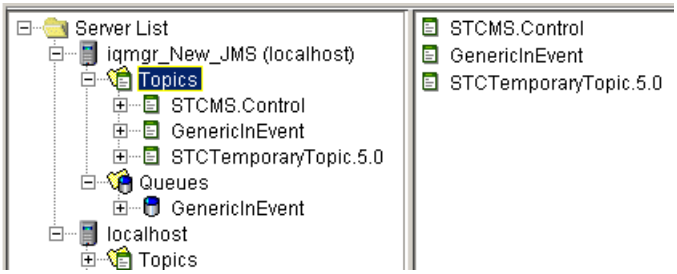
For information on JMS IQ Manager properties, see **"Properties of a JMS IQ Manager (Server)" on page 46**.

To list the topics and queues of a JMS IQ Manager

1 In the tree browser, under **Server List**, open the JMS IQ Manager whose topics and queues you want to list.

2 For the **Topics** folder and the **Queues** folder under this JMS IQ Manager, click ⊞ to view the contents of the folders. See Figure 18.

Figure 18 Displaying Topics and Queues



6.2.2 Working With Topics and Queues (Event Types)

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 a message (also called 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 a message (=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 behavior is equivalent to the *Subscriber Pooling* feature of 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 still 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 mesages of no interest to the particular subscriber "seeing" it.*

Special Topics

Every JMS IQ Manager has a special topic, **STCMS.Control**, which has a nondurable "subscriber" that allows the SeeBeyond Message Service utilities to monitor the status of the JMS IQ Manager and its topics; for information on these utilities, see **Chapter 7**. In addition, e*Gate creates temporary topics with one or more transient subscribers and topic names like **STCTemporaryTopic.***n.m*. All these special topics can be ignored.

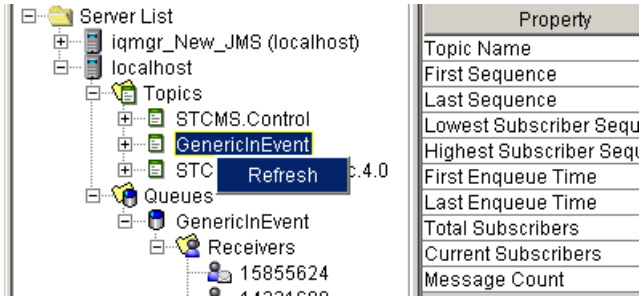
Receiver Names

At this release, the names of queue receivers for point-to-point messages are tracked by their location in memory rather than by their logical names—in other words, the names that receivers have within the e*Gate schema are not displayed.

To view the properties of a topic

1. In the tree browser, open the **Server List** folder and double-click the JMS IQ Manager that contains the topic whose properties you want to browse.
2. Open the **Topics** folder, right-click the topic name, and click **Refresh** on the shortcut menu. See Figure 19.

Figure 19 Topic Properties



For information on topic properties, see **"Properties of a Topic" on page 47**.

To list the subscribers of a topic

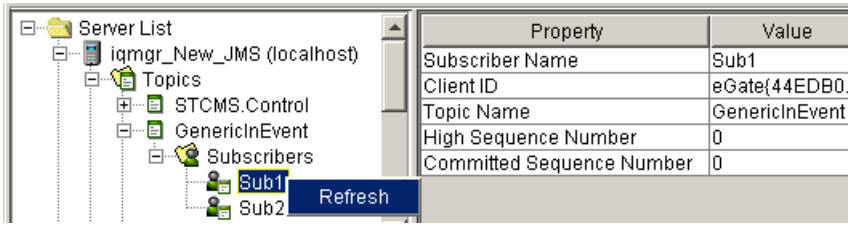
1. In the tree browser, open the **Server List** folder and double-click the JMS IQ Manager that contains the topic whose subscribers you want to list.
2. Open the **Topics** folder, find the correct topic, and click ⊞ to expand it.
3. Right-click the **Subscribers** folder and click **Refresh** on the shortcut menu.

The right-hand pane displays a list of the subscribers for the selected topic.

To view the properties of a subscriber

1. Use the tree browser to navigate to the topic containing the subscriber whose properties you want to view.
2. Open the topic's **Subscribers** folder.
3. Right-click the name of the subscriber whose properties you want to view, and click **Refresh** on the shortcut menu. See Figure 20.

Figure 20 Subscriber Properties

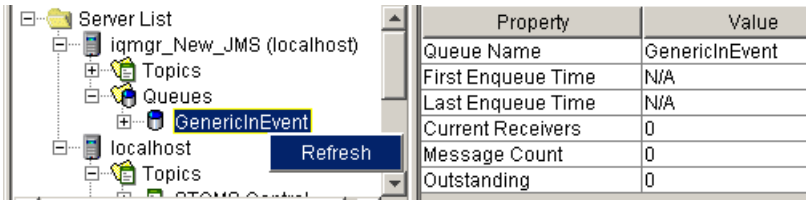


For information on subscriber properties, see **"Properties of a Subscriber to a Topic" on page 48**.

To view the properties of a queue

1. In the tree browser, open the **Server List** folder and double-click the JMS IQ Manager that contains the queue whose properties you want to browse.
2. Open the **Queues** folder, right-click the queue name, and click **Refresh** on the shortcut menu. See Figure 21.

Figure 21 Queue Properties



For information on queue properties, see **"Properties of a Queue" on page 48**.

To list the receivers of a queue

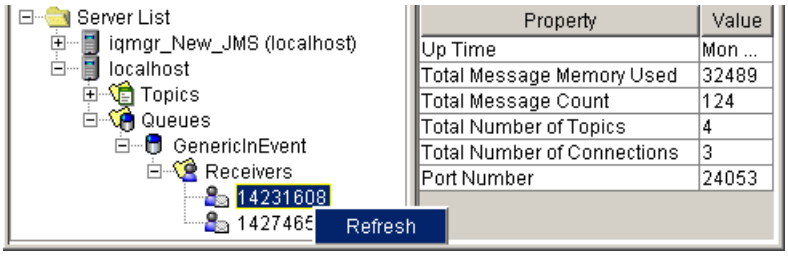
1. In the tree browser, open the **Server List** folder and double-click the JMS IQ Manager that contains the queue whose receivers you want to list.
2. Open the **Queues** folder, find the correct queue, and click ⊞ to expand it.
3. Right-click the **Receivers** folder and click **Refresh** on the shortcut menu.

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

To view the properties of a receiver

1. Use the tree browser to navigate to the queue containing the receiver whose properties you want to view.
2. Open the queue's **Receivers** folder.
3. Right-click the name of the receiver whose properties you want to view, and click **Refresh** on the shortcut menu. See Figure 22.

Figure 22 Receiver Properties



For information on receiver properties, see **"Properties of a Receiver of a Queue" on page 48**.

6.3 Properties

JMS Administrator allows you to see the property names and current values for each of the following:

Properties of a JMS IQ Manager (Server)

When you click the name of a JMS IQ Manager in the **Server List** folder, the properties pane displays the properties and current values of the JMS IQ Manager. See Table 2.

Table 2 JMS IQ Manager (Server) Properties

Property Name	Type	Description
Up Time	date/time	Date and time when this JMS IQ Manager was most recently started. Displayed in local format.
Total Message Memory Used	positive integer	Running bytecount of memory allocated for all uncommitted data message payloads.
Total Message Count	nonnegative integer, up to 2^64-1	Total messages processed since this JMS IQ Manager has been up. Includes not just data messages but also acknowledgments, etc.
Total Number of Event Types	nonnegative integer	Number of Event Types (topics plus queues) that this JMS IQ Manager is managing. Includes temporary topics and temporary queues.

Table 2 JMS IQ Manager (Server) Properties (Continued)

Property Name	Type	Description
Total Number of Connections	nonnegative integer	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	positive integer	The particular TCP/IP port of the Participating Host that this JMS IQ Manager is listening to.

Properties of a Topic

When you click the name of a topic (under the **Topics** folder under a particular JMS IQ Manager), the properties pane displays the properties and current values of the topic. See Table 3.

Table 3 Topic Properties

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

Properties of a Subscriber to a Topic

When you click the name of a subscriber (in the **Subscribers** folder under a particular topic), the properties pane displays the properties and current values of the subscriber. See Table 4.

Table 4 Subscriber Properties

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

Properties of a Queue

When you click the name of a queue (under the **Queues** folder under a particular JMS IQ Manager), the properties pane displays the properties and current values of the queue. See Table 5.

Table 5 Queue Properties

Property Name	Type	Description
Queue Name	string	Logical name of the Event Type for this queue.
First Enqueue Time	date/time	Timestamp of the first message in this queue that has still not been processed.
Last Enqueue Time	date/time	Timestamp of the most recent message in this queue.
Current Receivers	nonnegative integer	Number of receivers for this queue.
Message Count	nonnegative integer	Total number of messages in this queue that have already been processed and deleted.
Outstanding	nonnegative integer	Total number of messages in this queue that have not yet been processed.

Properties of a Receiver of a Queue

When you click the name of a receiver (in the **Receivers** folder under a particular queue), the properties pane displays the properties and current values of the receiver. See Table 6.

Table 6 Receiver Properties

Property Name	Type	Description
Receiver ID	nonnegative integer < 2^32-1	Address in virtual memory of this receiver.
Queue Name	string	Logical name of the queue (Event Type) sending to this receiver.
Session ID	nonnegative integer	ID of the current JMS session.
Committed Sequence	nonnegative integer	Number of messages that this queue receiver has already committed.
Uncommitted Sequence	nonnegative integer	Number of messages that this queue receiver has read but not yet committed.

Chapter 7

STC MS Control Utility

This chapter includes information on the SeeBeyond Java Message Service command-line utility, **stcmsctrlutil.exe**.

7.1 Overview

The STC Message Service Control utility **stcmsctrlutil.exe** provides a command-line interface to an active Message Service server. From a command line or batch program, you can **stcmsctrutil.exe** it to:

- Learn the version of the MS server or of the utility.
- Shut down the server.
- For a specified server: List, create, or delete topics or queues.
- For a specified topic: List, create, or delete subscribers, retrieve a topic message list, or view topic statistics.
- For a specified queue: List, create, or delete receivers, retrieve a queue message list, or view queue statistics.
- For a specified message: View, delete, or modify message content.
- Create, delete, modify, monitor, or list the contents of a particular queue or topic.
- View or modify a particular message (Event) or its type.
- Fetch or delete a range of messages (Events).
- Lock or unlock a particular queue or topic so that it access is denied or restored to all its subscribers.

User preferences for **stcmsctrlutil.exe** are stored in the **stcmsutil.Properties** file. See **"stcmsutil.Properties" on page 56**.

7.2 Syntax

There are only two options of **stcmsctrlutil** that do not require you to supply a host name and port number:

```
stcmsctrlutil --help

stcmsctrlutil --version
```

For all other uses of **stcmsctrlutil**, you must supply a host name and a port number and, optionally, a port offset, in addition to other flags and arguments, using one of the following formats.

```
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]]
```

7.3 Flags and Arguments

Table 7 Flags and Arguments for Utility **stcmsctrlutil**

Shortcut	Flag *arguments*	Purpose
	--help	to display help information
	--version	to display utility version information
	-host *hostName*	to 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*	to 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*	to specify a server port offset number
	-msversion	to display server version information
	-shutdown	to shut down the server
	-status	to display server status
-tl	**-topiclist**	to list all topics for this server
-sla	**-sublistall**	to list all subscribers for all topics combined
-slft	**-sublistfortopic** *topicName*	to list all subscribers for the specified topic
-ts	**-topicstat** *topicName*	to display statistics for the specified topic
-ct	**-createtopic** *topicName*	to create a new topic with the specified name

Table 7 Flags and Arguments for Utility **stcmsctrlutil (Continued)**

Shortcut	Flag *arguments*	Purpose
-dt	**-deletetopic** *topicName*	to delete the specified topic
-cs	**-createsub** *topicName subName clientName*	to create a new subscriber for the specified topic and client. For *clientName*, specify **eGate**.
-ds	**-deletesub** *topicName subName clientName*	to delete a certain subscriber from the specified topic and client. For *clientName,* specify **eGate**.
-tml	**-tmsglist** *topicName seqNumber numberMessages*	to list all messages (=Events) for the specified topic, starting at or above the specified sequence number, and listing no more than *numberMessages* altogether.
-gtm	**-tmessage** *topicName seqNumber*	to get 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*	to delete the message at the specified sequence number in the specified topic.
-ctm	**-changetmsg** *topicName seqNumber*	to 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).
-lt	**-locktopic** *topicName*	to lock a topic from being accessed, preventing any subscriber from receiving messages from it.
-ut	**-unlocktopic** *topicName*	to unlock a topic, restoring access to all subscribers.
-ql	**-queuelist**	to list all queues for this server
-rla	**-recvlistall**	to list all receivers for all queues combined
-rlfq	**-recvlistforqueue** *queue-name*	to list all receivers for the specified queue
-qs	**-queuestat** *queue-name*	to display statistics for a specific queue
-cq	**-createqueue** *queueName*	to create a new queue with the specified name
-dq	**-deletequeue** *queueName*	to delete the specified queue
-qml	**-qmsglist** *queueName seqNumber numberMessages*	to list all messages (=Events) for the specified queue, starting at or above the specified sequence number, and listing no more than *numberMessages* altogether.
-gqm	**-qmessage** *queueName*	to get 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.
-dqm	**-delqmsg** *queueName*	to delete the message at the specified sequence number in the specified queue.

Table 7 Flags and Arguments for Utility **stcmsctrlutil (Continued)**

Shortcut	Flag *arguments*	Purpose
-cqm	**-changeqmsg** *sequenceNumber*	to 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).
-lq	**-lockqueue** *queueName*	to lock a queue from being accessed, preventing any receiver from receiving messages from it.
-uq	**-unlockqueue** *queueName*	to unlock a queue, restoring access to all receivers.
	-msgtype *type*	to specify the data type of the content of the message(=Event). Must be either **bytes** or **text**

7.3.1 Examples

To display the version of the utility

```
O:\Debug\bin>stcmsctrlutil --version
Control Utility version 4.5.2.4122 (Feb 22 2002)
Copyright (c) 2002, SeeBeyond Technology Corporation.
All Rights Reserved.
```

To shut down the server

```
O:\Debug\bin>stcmsctrlutil -host localhost -port 24055 –shutdown
```

To display statistics about Message Service

```
O:\Debug\bin>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
```

To list all topics for this server

```
O:\Debug\bin>stcmsctrlutil -host localhost -port 24055 -topiclist
Topic List:
    SeeBeyond.MS.Control
    Broadcast
    STCTemporaryTopic.2.1
```

To display statistics for topic “Broadcast”

```
C:>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
```

To view properties of all subscribers

```
O:\Debug\bin>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
```

To view properties of all subscribers subscribed to topic "STC"

```
C:>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
```

To list all queues for this server

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

To display statistics for queue "PTP"

```
C:>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
```

To view properties of all receivers

```
O:\Debug\bin>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
```

To view properties of all receivers of queue "PTP"

```
C:>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
```

7.4 stcmsutil.Properties

The **stcmsutil.Properties** file holds current settings and preferences for the STC MS utility, **stcmsctrlutil.exe**. Table 8 shows the property names and their preset values if never modified.

Table 8 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**.

Index

Symbols

A

C

D

E

I

M

O

P

S

T

U