



# Retek<sup>®</sup> Integration Bus<sup>™</sup> 10.3.2

## Installation Guide



The software described in this documentation is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

No part of this documentation may be reproduced or transmitted in any form or by any means without the express written permission of Retek Inc., Retek on the Mall, 950 Nicollet Mall, Minneapolis, MN 55403.

Information in this documentation is subject to change without notice.

Retek provides product documentation in a read-only-format to ensure content integrity. Retek Customer Support cannot support documentation that has been changed without Retek authorization.

**Corporate Headquarters:**

Retek Inc.  
Retek on the Mall  
950 Nicollet Mall  
Minneapolis, MN 55403  
888.61.RETEK (toll free US)  
+1 612 587 5000

**European Headquarters:**

Retek  
110 Wigmore Street  
London  
W1U 3RW  
United Kingdom  
Switchboard:  
+44 (0)20 7563 4600  
Sales Enquiries:  
+44 (0)20 7563 46 46  
Fax: +44 (0)20 7563 46 10

Retek® Integration Bus™ is a trademark of Retek Inc.

Retek and the Retek logo are registered trademarks of Retek Inc.

This unpublished work is protected by confidentiality agreement, and by trade secret, copyright, and other laws. In the event of publication, the following notice shall apply:

©2003 Retek Inc. All rights reserved.

All other product names mentioned are trademarks or registered trademarks of their respective owners and should be treated as such.

Printed in the United States of America.



## ***Customer Support***

### **Customer Support hours:**

Customer Support is available 7x24x365 via e-mail, phone and Web access.

Depending on the Support option chosen by a particular client (Standard, Plus, or Premium), the times that certain services are delivered may be restricted. Severity 1 (Critical) issues are addressed on a 7x24 basis and receive continuous attention until resolved, for all clients on active maintenance.

| <b>Contact Method</b> | <b>Contact Information</b> |
|-----------------------|----------------------------|
|-----------------------|----------------------------|

|                        |  |
|------------------------|--|
| <b>Internet (ROCS)</b> | <a href="http://www.retek.com/support">www.retek.com/support</a><br>Retek's secure client Web site to update and view issues |
|------------------------|--|

|               |                    |
|---------------|--------------------|
| <b>E-mail</b> | support@rettek.com |
|---------------|--------------------|

|              |   |
|--------------|---|
| <b>Phone</b> | US & Canada: 1-800-61-RETEK (1-800-617-3835)<br>World: +1 612-587-5800<br>EMEA: 011 44 1223 703 444<br>Asia Pacific: 61 425 792 927 |
|--------------|---|

|             |   |
|-------------|---|
| <b>Mail</b> | Retek Customer Support<br>Retek on the Mall<br>950 Nicollet Mall<br>Minneapolis, MN 55403 |
|-------------|---|

### **When contacting Customer Support, please provide:**

- Product version and program/module name.
- Functional and technical description of the problem (include business impact).
- Detailed step by step instructions to recreate.
- Exact error message received.
- Screen shots of each step you take.



# Contents

|  |           |
|--|-----------|
| <b>Chapter 1 – Introduction.....</b>   | <b>1</b>  |
| <b>Chapter 2 – Install SeeBeyond e*Gate Integrator.....</b>                      | <b>3</b>  |
| <b>Chapter 3 – RIB schema .....</b>  | <b>5</b>  |
| Import .....   | 5         |
| Preliminary steps .....  | 5         |
| Create a new schema .....  | 7         |
| Import RIB Components .....  | 8         |
| Install Delta Release.....   | 10        |
| Preliminary Steps.....   | 10        |
| Import RIB Components .....  | 11        |
| RIB schema configuration.....  | 13        |
| Step 1: Modify the main Participating Host and Control Broker configuration..... | 14        |
| Step 2: Modify the JMS IQ Manager configuration .....                            | 16        |
| Step 3: Modify Connection Point configurations .....                             | 18        |
| Step 4: Delete unused e*Ways .....   | 24        |
| Step 5: Add/Copy e*Ways for additional components.....                           | 29        |
| Step 6: Edit the rib.properties file to correspond to the system .....           | 30        |
| Step 7: Create/modify startup scripts.....                                       | 30        |
| Step 8: Starting the RIB components for a new/modified schema .....              | 31        |
| <b>Chapter 4 – Database triggers and Oracle dependencies .</b>                   | <b>33</b> |
| Database triggers .....  | 33        |
| Oracle dependencies.....   | 33        |
| Verify RIB Error Hospital database tables.....                                   | 33        |
| RIB_DOCTYPES table and DTD files .....   | 36        |
| <b>Chapter 5 – RIB Administration Tool installation.....</b>                     | <b>37</b> |
| Web-based version install .....  | 37        |
| Prerequisites .....  | 37        |
| Install RIB Administration Tool.....   | 37        |
| Internationalization .....   | 40        |
| Install JRE .....  | 41        |
| Test Error Hospital GUI Applet .....   | 41        |
| Files and classes contained in the war file .....                                | 41        |
| Classes .....  | 41        |
| Jars and other files .....   | 42        |
| Language-specific files .....  | 43        |

|   |           |
|---|-----------|
| <b>Chapter 6 – WebSphere Integration .....</b>            | <b>45</b> |
| rib.properties file .....                                 | 45        |
| Error Hospital entries.....                               | 45        |
| Logging entries .....                                     | 46        |
| JNDI/JMS Configuration.....                               | 46        |
| Implementation classes.....                               | 47        |
| Publishing configuration.....                             | 48        |
| rib_subscriber.xml file .....                             | 51        |
| Generic JMS Provider .....                                | 51        |
| Message Listener Ports.....                               | 51        |
| Data Source .....   | 52        |
| <b>Chapter 7 – ISO Integration .....</b>                  | <b>53</b> |
| Manual Steps .....  | 53        |
| ISO Reference .....                                       | 55        |
| rib.properties file .....                                 | 55        |
| Example of a messaging component configuration file ..... | 57        |
| ribmessaging.cfg file .....                               | 58        |

# Chapter 1 – Introduction

This manual details the installation of the Retek Integration Bus (RIB). An overview of this process is as follows:

- 1 The SeeBeyond e\*Gate Integrator product (version 4.5.3) is installed. This involves installing the registry host and all participating host software, plus Graphical User Interface hosts for development and system monitoring. See Chapter 2 for details.
- 2 The RIB schema is imported into the e\*Gate Integrator product. This is explained in Chapter 3.
- 3 Update the database connection points, JMS queues, and CLASSPATH configuration values. Also, delete unused adapters. This is explained in Chapter 3.
- 4 Verify the Error Hospital tables exist; make DTD files available on the network; and install the Hospital GUI components.
- 5 Application Server specific configurations.





## Chapter 2 – Install SeeBeyond e\*Gate Integrator

The Retek Integration Bus (RIB) leverages SeeBeyond's e\*Gate Integrator for supplying the needed messaging facilities for integrating applications.

The following steps need to be completed successfully to install the e\*Gate product:

- 1 A Registry host that will contain the central database of the message formats, as well as publication, subscription and transformation logic needs to be installed.
- 2 At least one Participating host, which implements the publishers, subscribers, and transformations, needs to be installed.
- 3 The required e\*Gate add-ons need to be installed.
  - Batch e\*Way 4.5.4 add-on
  - Oracle e\*Way 4.5.3 add-on (included in this is the JDBC e\*Way)
- 4 The GUI hosts that are used to monitor the operation of the system and to extend or further develop the system's capabilities need to be installed.
  - e\*Gate GUI 4.5.3
- 5 There may be a number of required e\*Gate ESR (patches) to be installed. A listing of any of these will be included in the RIB Release Notes. ESRs can be found on Retek's fulfillment center (<http://fulfillment.retek.com>).

### Notes:

- All three types of hosts can be present on the same physical machine. However, GUI hosts must execute on a Microsoft Windows platform.
- e\*Gate requires a Java Runtime Environment (JRE) version 1.3.1. This is bundled with the e\*Gate install.
- e\*Gate Monitor and e\*Gate Enterprise Manager applications use the Exceed X-windows application. If a version of Exceed exists on a GUI host, then one must install the e\*Gate version into a different directory. The e\*Gate version is *not* a full installation of Exceed.

The instructions for installing the SeeBeyond e\*Gate Integrator system are documented in the *e\*Gate Integrator Installation Guide*. This document is found on Disk 2 of the SeeBeyond installation disk set (docs\eGate\_Install\_Guide.pdf)



## Chapter 3 – RIB schema

### Import

If this is a Delta release of the RIB (see Release Notes to identify delta releases), skip to the [Install Delta Release](#) section of this document.

The RIB software is distributed in a single messaging schema. This schema contains all of the RIB's publishing and subscribing e\*Ways (adapters) and Connection Points. It also contains a single JMS Intelligent Queue Manager.

Once the RIB schema has been imported, a system administrator must configure the connection points. Additional configuration modifications may also be needed, such as e\*Way CLASSPATH. These types of changes are detailed in Chapter 3.

The final modifications to the system are due to the site-specific deployment of the system. These changes include distributed components to different hosts, creating fail-over hosts, developing additional event types, adapters, connection points and collaborations for integrating an enterprise's non-Retek applications to the RIB. It also includes creating security roles and privileges. These activities are not considered part of the installation and are not documented in this manual. For more information on these activities, see the *SeeBeyond e\*Gate Users Guide*.

### Preliminary steps

To create and import the RIB schema, take the following preliminary steps:

- 1 For security reasons, create an "egate" user that will own the e\*Gate files and execute the software.
- 2 Log onto the Unix system where eGate was installed using this account.
- 3 Copy the RIB tar files from the RIB installation CD(s) to the location where you are planning to install the RIB software. This location will be known as the RETEK\_INSTALL\_DIR in the remainder of this section. The RIB tar files are named RIBFor<APP><version>.tar (where the application is RDM, RCOM, RMS, ISO, RDC, etc.). An example of the <RETEK\_INSTALL\_DIR> directory name could be called "INSTALL", located directly under the "egate" user's home directory. Future releases of the RIB can be installed into this directory, as the directories will have new version numbers in their names.  
(eg: /files0/egate/INSTALL/)
- 4 Once you have copied the RIB tar file(s) to <RETEK\_INSTALL\_DIR>, extract each file in this directory and change the permissions on the extracted files to make them writable.
  - tar xvf 'filename'
  - chmod -R 755 \*

- 5 Change directories to <RETEK\_INSTALL\_DIR>/RIB<version>.
- 6 Edit the file egate\_profile. Make sure the settings for the following variables are correct for your environment.
  - EHOME – The directory where SeeBeyond e\*Gate was installed.
  - RETEK\_INSTALL\_DIR – The directory created in step 3 above.
  - EGATE\_SERVER\_NAME – The name or IP address of the server e\*Gate and the RIB software are installed on.
  - EGATE\_SERVER\_PORT – The port that the e\*Gate Registry Host was installed on during the SeeBeyond e\*Gate Integrator install.
  - Platform specific section (Sun Solaris, IBM AIX, HP-UX) - Uncomment the section that is applicable to your operating system and ensure that the other two sections are commented out.
- 7 Edit the “egate” user’s .profile located in the “egate” user’s home directory. Add an entry at the end of this file that sources the egate\_profile modified in step 6 above.
  - <RETEK\_INSTALL\_DIR>/RIB<version>/egate\_profile  
(eg: . /files0/egate/INSTALL/RIB<version>/egate\_profile)
  - Ensure that ‘.’ (dot colon) is at the beginning of the egate user’s PATH variable.  
(eg: PATH=.:\${PATH} ; export PATH)

Source the .profile after making these modifications or start a new Unix session before continuing.
- 8 If there was an earlier attempt at installing the 10.3 version of the RIB, it must be inactivated by renaming it:
  - a Make sure that all e\*Ways, the control broker, and the registry are shut down. On Unix, the following command will show the active processes:
 

```
> ps -ef | grep stc | grep -v grep
```

If stc processes are still running, be sure to shut down all the stc processes (eg: kill -9)
  - b Rename the \$EHOME/server/registry/RIB103.rdb file to \$EHOME/server/registry/RIB103.rdb.bak.
  - c Rename the \$EHOME/server/registry/repository/RIB103 directory to \$EHOME/server/registry/repository/RIB103.bak.
  - d The RIB103.rdb.bak file and RIB103.bak directory can be deleted at a later time once the new version has been successfully installed.

- 9 Start the e\*Gate registry – the following command can be run manually, or the start\_egate script can be run; it is located at:  
`<RETEK_INSTALL_DIR>/RIB<version>/Rib_Support/`

```

> let CB_PORT=$EGATE_SERVER_PORT+1
> $EHOME/server/bin/stcregd -ss -ln
$EGATE_SERVER_NAME -bd $EHOME/server -pr
$EGATE_SERVER_PORT -pc $CB_PORT -mc 1024 !> /dev/null

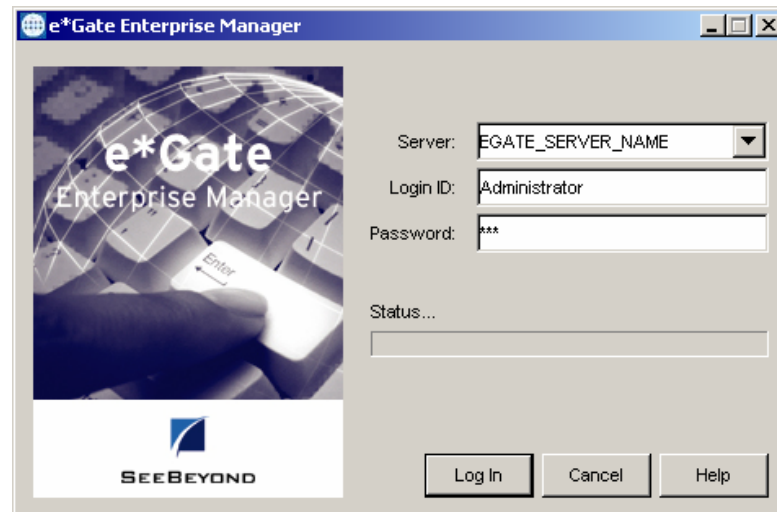
```

## Create a new schema

The RIB schema is imported through a two-step process. The first step involves creating a new schema. This new schema is empty and does not contain any RIB modules.

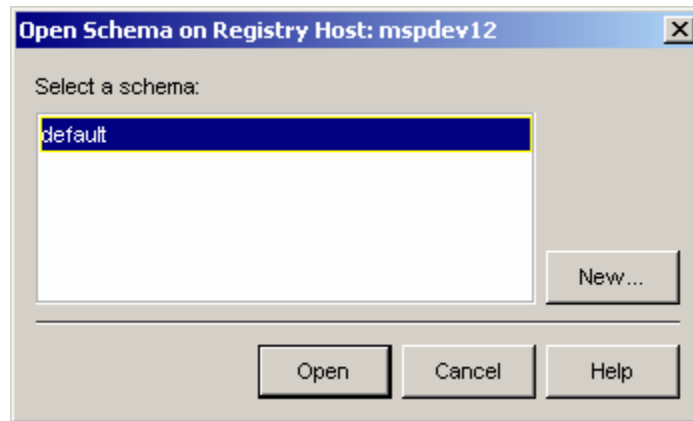
- 1 Log in to the e\*Gate registry using the e\*Gate Enterprise Manager GUI tool. Log in as Administrator, using the password that was set during the installation of e\*Gate Integrator.

**Note:** For the Server: field, enter the EGATE\_SERVER\_NAME or IP address as specified in the egate\_profile modified previously. Also, because the e\*Gate Registry Host may not be running on the default port (23001), it is good practice to always specify the port along with the hostname. The format is <hostname>:<port>, (e.g. mspdev14:23001).



*e\*Gate Enterprise Manager dialog box*

The Open Schema dialog box is displayed.



*Open Schema dialog box*

- 2 Click **New** to create a new schema. The New Schema dialog box is displayed.
- 3 In the Enter New Schema Name field, enter RIB103.
- 4 Click **Open**.

You have now successfully created an empty schema named RIB103.

## Import RIB Components

The second step of the RIB Schema import process entails the actual importing of the RIB components into the newly created “RIB103” base schema.

## Load RIB Components - Automated Instructions

A script to register the new RIB Schema, “RIB103”, and insert all of the registry modules can be found in the following directory:

```
<RETEK_INSTALL_DIR>/RIB<version>/Migration_Scripts
```

### Notes:

- If your RIB103 schema has a password for the Administrator user that is different than the default of “STC” you will need to edit the install script and replace the two occurrences of “STC” with your password (case sensitive).
- The system must be able to locate the unzip and zip utilities for the importways script to work. If not, update the PATH variable in the egate user’s .profile file.
- Additionally, a working version of perl must be installed on your machine. Be sure the PATH variable contains an entry to where this file resides.
- The DISPLAY variable must be set to the IP address of the machine that the install will run on.

**Unless all of these requirements are met, the manual method to import the RIB modules will have to be used.**

- 1 From this directory, run the “install” script.
- 2 Once the script has completed, use Enterprise Manager to check to ensure each module was successfully loaded into the schema. If the Enterprise Manager is already running, select the Refresh option under the View menu. Otherwise, start the Enterprise Manager.
- 3 Following the schema import, start the control broker in order to ensure that the schema is functioning properly.

**Note:** The RIB schema will not function properly until the system has been completely configured!

The “install” script creates soft links in \$EHOME for the following 5 scripts:

- start\_egate →  
`<RETEK_INSTALL_DIR>/RIB<version>/Rib_Support/start_egate`
- start\_rib →  
`<RETEK_INSTALL_DIR>/RIB<version>/Rib_Support/start_rib`
- Egate.txt →  
`<RETEK_INSTALL_DIR>/RIB<version>/Egate/Eways/Egate.txt`
- start\_cb →  
`<RETEK_INSTALL_DIR>/RIB<version>/Rib_Support/start_cb`
- plist → `<RETEK_INSTALL_DIR>/RIB<version>/Rib_Support/plist`

- 4 Continue the install process: [RIB schema configuration](#).

## Load RIB Components - Manual Instructions

These instructions are provided in order to individually load modules into the schema. Skip this section if the automated process was used and completed successfully.

Each RIB module can be loaded into the RIB103 schema manually, if necessary, by running the “Import Definitions from File” feature of the e\*Gate Enterprise Manager GUI.

- 1 From a Windows PC, which has the e\*Gate GUI installed, put the `<MODULE>.zip` file(s) to be loaded on an accessible drive.
- 2 Start the e\*Gate Enterprise Manager.
- 3 Log in to the RIB103 schema.
- 4 Select File > Import Definitions from File. The Import Wizard is displayed.
- 5 Click **Next**. The Step 1 page is displayed.
- 6 Select the “Module” radio button. Click **Next**. The Step 2 page is displayed.
- 7 Locate/select the `<MODULE>.zip` file to be imported into the RIB103 schema. Click **Next**.
- 8 Click **Finish**. The Import Component dialog box is displayed, asking for confirmation as to which Participating Host/Control Broker pair to import into. Click **OK**.

- 9 Repeat the above steps for each <MODULE>.zip file that you wish to import.
- 10 Continue the install process: [RIB schema configuration](#).

## Install Delta Release

### Preliminary Steps

- 1 Log into the UNIX system as the “egate” user.
- 2 Copy the RIB .tar files from the RIB installation CD to the <RETEK\_INSTALL\_DIR> directory.
- 3 Once the RIB .tar files have been copied over, extract each file and change the permissions on the extracted files to make them writeable.
  - `tar xvf 'filename'`
  - `chmod -R 755 *`
- 4 Change directories to <RETEK\_INSTALL\_DIR>/RIB<version>.
- 5 Edit the file `egate_profile`. Make sure the settings for the following variables are correct for your environment:
  - EHOME – The directory where SeeBeyond e\*Gate was installed.
  - RETEK\_INSTALL\_DIR – The install directory referenced in Step 2.
  - EGATE\_SERVER\_NAME – The name or IP address of the server e\*Gate and the RIB software are installed on.
  - EGATE\_SERVER\_PORT – The port that the e\*Gate Registry Host was installed on during the SeeBeyond e\*Gate Integrator install.
- 6 Edit the “egate” user’s .profile located in the “egate” user’s home directory. Add an entry at the end of this file that sources the `egate_profile` modified in step 5 above.
  - `. <RETEK_INSTALL_DIR>/RIB<version>/egate_profile`  
(eg: `. /files0/egate/INSTALL/RIB<version>/egate_profile`)

Ensure that ‘.’ (dot colon) is at the beginning of the `egate` user’s `PATH` variable.

(eg: `PATH=.:${PATH} ; export PATH`)

Source the “egate” user’s .profile after making these modifications or start a new Unix session before continuing.



## Import RIB Components

### Load RIB Components – Automated Instructions

A script to insert the new registry modules can be found in the following directory:

```
<RETEK_INSTALL_DIR>/RIB<version>/Migration_Scripts
```

#### Notes:

- If your RIB103 schema has a password for the Administrator user that is different than the default of “STC” you will need to edit the install script and replace the two occurrences of “STC” with your password (case sensitive).
- The system must be able to locate the unzip and zip utilities for the importways script to work. If not, update the PATH variable in the egate user’s .profile file.
- Additionally, a working version of perl must be installed on your machine. Be sure the PATH variable contains an entry to where this file resides.
- The DISPLAY variable must be set to the IP address of the machine that the install will run on.

**Unless all of these requirements are met, the manual method to import the RIB modules will have to be used.**

- 1 From this directory, run the “install” script.
- 2 Once the script has completed, use Enterprise Manager to check to ensure each module was successfully loaded into the schema. If the Enterprise Manager is already running, select the Refresh option under the View menu. Otherwise, start the Enterprise Manager.
- 3 Following the schema import, start the control broker in order to ensure that the schema is functioning properly.

**Note:** The RIB schema will not function properly until the system has been completely configured.

The “install” script creates soft links in \$EHOME for the following 5 scripts:

- start\_egate → <RETEK\_INSTALL\_DIR>/RIB<Version>/Rib\_Support/start\_egate
- start\_rib → <RETEK\_INSTALL\_DIR>/RIB<Version>/Rib\_Support/start\_rib
- Egate.txt → <RETEK\_INSTALL\_DIR>/RIB<Version>Egate/Eways/Egate.txt
- start\_cb → <RETEK\_INSTALL\_DIR>/RIB<Version>/Rib\_Support/start\_cb
- plist → <RETEK\_INSTALL\_DIR>/RIB<Version>/Rib\_Support/plist

Change directories to \$EHOME and run the “start\_cb” script.

- 4 Continue the install process: [RIB schema configuration](#).

## Load RIB Components - Manual Instructions

These instructions are provided in order to individually load modules into the schema. Skip this section if the automated process was used and completed successfully.

Each RIB module can be loaded into the RIB103 schema manually, if necessary, by running the “Import Definitions from File” feature of the e\*Gate Enterprise Manager GUI.

- 1 From a Windows PC, which has the e\*Gate GUI installed, copy the <RETEK\_INSTALL\_DIR>/RIBfor<APP><version>/Egate/eways-in/<MODULE>.zip file(s) to a drive accessible by the Windows PC.  
(e.g. /files0/egate/INSTALL/RIBforISO<version>/Egate/eways-in/ewInvReqToRMS.zip)  
(e.g. /files0/egate/INSTALL/RIBforRMS<version>/Egate/eways-in/ewItemZonePrcFromRMS.zip)
- 2 Open the e\*Gate Enterprise Manager.
- 3 Log into the RIB103 schema.
- 4 Select File → Import Definitions from File. The Import Wizard is displayed.
- 5 Click **Next**. The Step 1 page is displayed.
- 6 Select the “Module” radio button. Click **Next**. The Step 2 page is displayed.
- 7 Locate/select the <MODULE>.zip file to be imported into the RIB103 schema. Click **Next**.
- 8 Click **Finish**. The Import Component dialog box is displayed, asking for confirmation as to which Participating Host/Control Broker pair to import into. Click **OK**.
- 9 Repeat the above steps for each <MODULE>.zip file that you wish to import.
- 10 Continue the install process: [RIB schema configuration](#)

## RIB schema configuration

After the RIB schema has been imported, the schema must be configured for the site-specific environment. This section details the minimum configuration changes needed to get the RIB schema into an operational state. It assumes that all schema components will run on a single host and that all databases referenced are accessible from this host.

This chapter details the minimum changes needed for the RIB to run. It assumes that the RIB is deployed on a single host and that only a single JMS IQ Manager is needed. This deployment configuration is *not* appropriate for all RIB installations. Production environment deployments may choose to distribute different specific e\*Ways and JMS queues among multiple hosts. This type of production deployment is not covered in this manual.

The following steps are required to configure the RIB schema:

- 1 Modify the main Participating Host and Control Broker configuration.
- 2 Modify the JMS IQ Manager configuration.
- 3 Modify Connection Point configurations.
- 4 Delete unused e\*Ways.
- 5 Add/Copy e\*Ways for additional components.
- 6 Edit the rib.properties file to correspond to the system.
- 7 Create/modify startup scripts.

## Step 1: Modify the main Participating Host and Control Broker configuration

The first step in the RIB messaging schema configuration is to modify the main participating host and control broker's configuration. The RIB103 schema includes a single participating host and control broker that contains all of the messaging e\*Ways and associated components. If these are not modified, then the configuration will attempt to resolve host names and ports as specified by the supplied/shipped configuration.

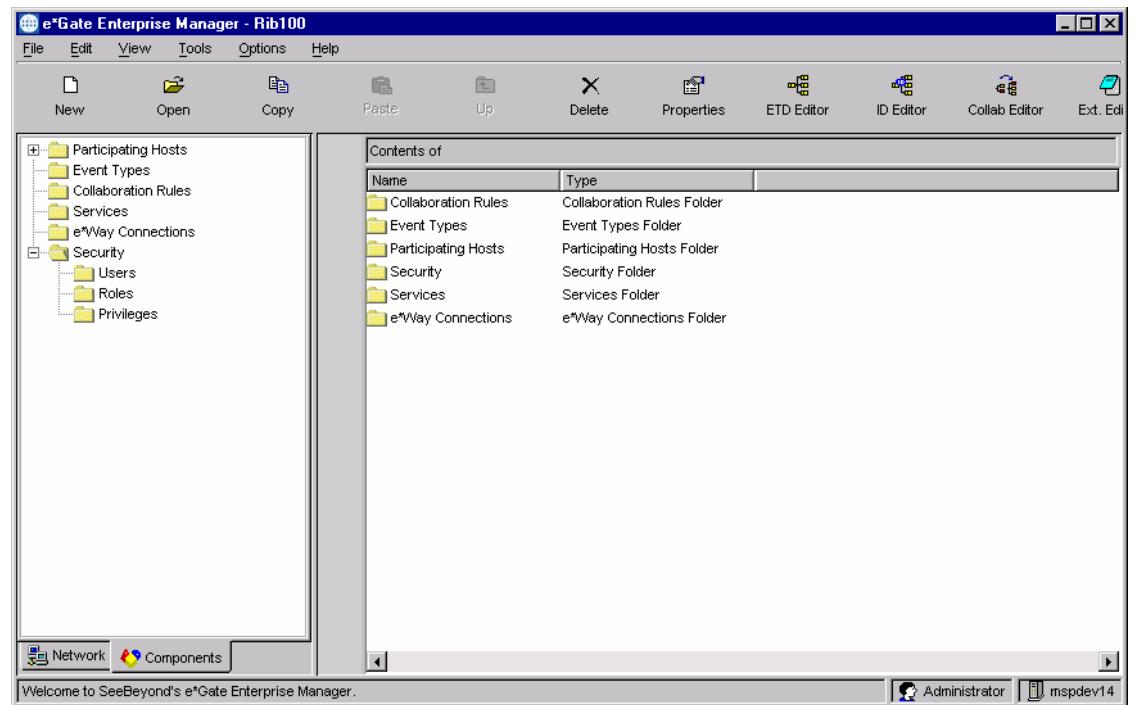
Alternatively, you can change the configuration of the participating host and its supplied control broker.

Changing these configurations is a manual process performed with the e\*Gate Enterprise Manager application. This application must be installed on a Microsoft Windows 2000 or Microsoft Windows NT platform. Specific platform requirements are detailed in the SeeBeyond *e\*Gate Integrator Installation Guide*.

These instructions modify both the names and IP address of the participating host and control broker. The name of the control broker must match any start-up scripts used.

### Modify the configuration

- 1 Open the e\*Gate Enterprise Manager.
- 2 Connect the e\*Gate Enterprise Manager to the RIB103 schema. The following window is displayed:

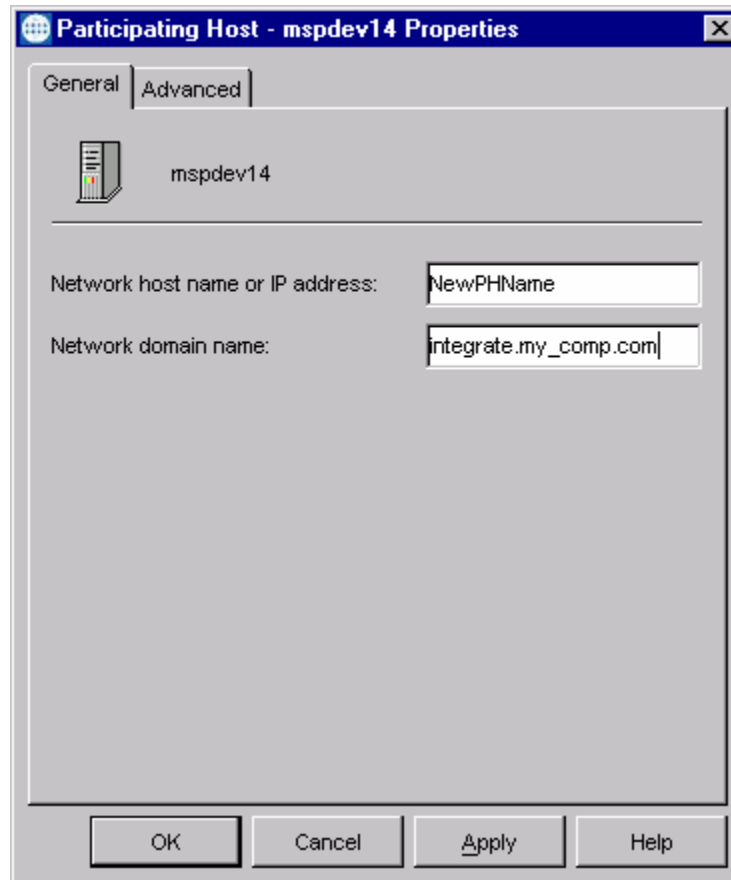


*The main e\*Gate Enterprise Manager window*

- 3 Right-click on the first active participating host displayed. A command list is displayed.

**Note:** An active participating host is one *without* the string “(inactive)” as part of its name. If there is *not* a participating host *without* the “(inactive)” string, refer to the SeeBeyond System Administrator’s Guide for instructions on how to activate the correct participating host.

- 4 Select **Properties....** The Participating Host Properties dialog box is displayed.



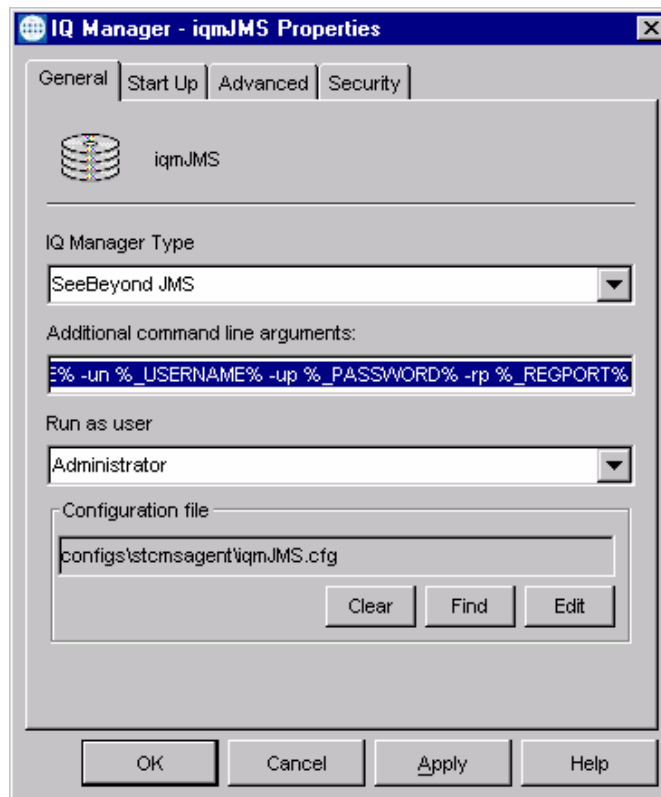
*Participating Host Properties window*

- 5 In the Network host name or IP address field, enter the e\*Gate server name.
- 6 In the Network domain name field, enter the correct network domain name for your environment.
- 7 Click **OK**.

## Step 2: Modify the JMS IQ Manager configuration

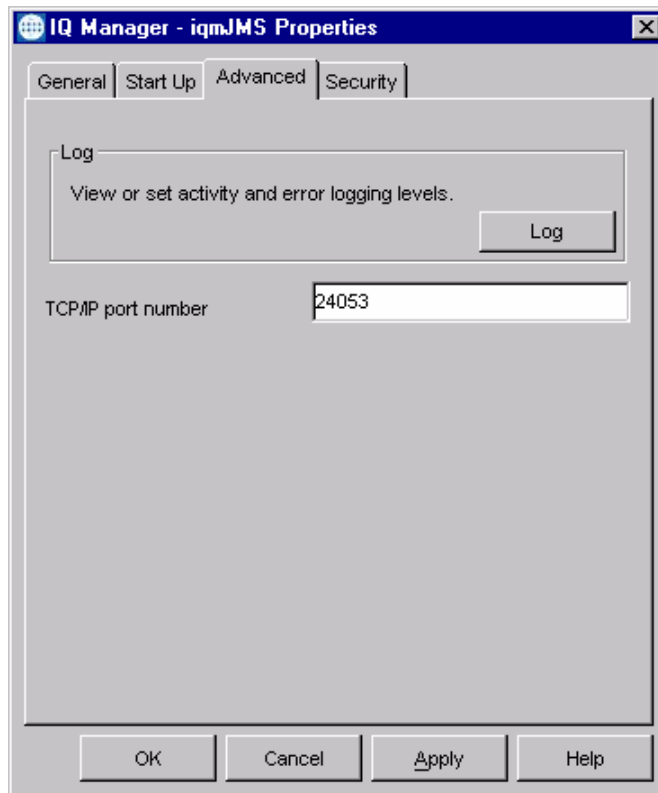
The JMS IQ Manager is initially configured to use the TCP/IP port number of **24053** for all e\*Gate publishers and subscribers to connect to. If this port is used by other applications, then the JMS IQ Manager will not be able to be started. Complete this step only if port 24053 is **NOT** available.

- 1 In the main e\*Gate Enterprise Manager window, right-click on the **iqmJMS** queue manager. (The iqmJMS queue manager is towards the bottom of the Components frame, below all of the e\*Ways.)
- 2 Select **Properties....** The IQ Manager Properties dialog box is displayed.



***IQ Manager Properties dialog box for iqmJMS***

- 3 Click on the Advanced tab at the top of the window.



*Advance IQ Manager Properties window*

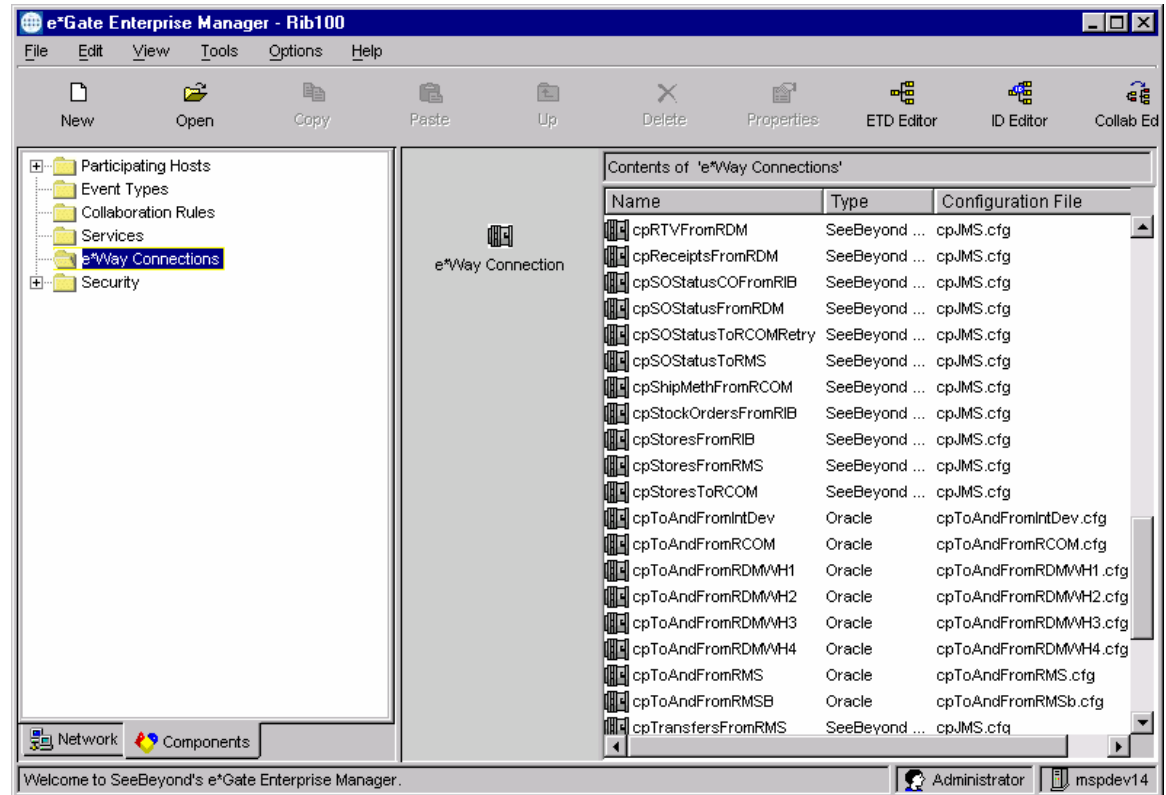
- 4 In the TCP/IP port number field, change the port number to an available port.
- 5 Click **OK**. Note the port number for the next step.

**Note:** If there are multiple instances of e\*Gate running on a single Unix server, close attention must be paid to the registry, control broker and JMS ports. Runtime problems will be encountered if instances share the same ports.

## Step 3: Modify Connection Point configurations

The next step is to modify the Connection Point configurations to reflect the JMS IQ Manager and Oracle databases used. This is performed in the e\*Gate Enterprise Manager application.

- From the main window, click on the e\*Way Connections folder. The window changes to reflect the available connections.



### Connection Points

There are two types of connection points supplied with the RIB: SeeBeyond JMS and Oracle.

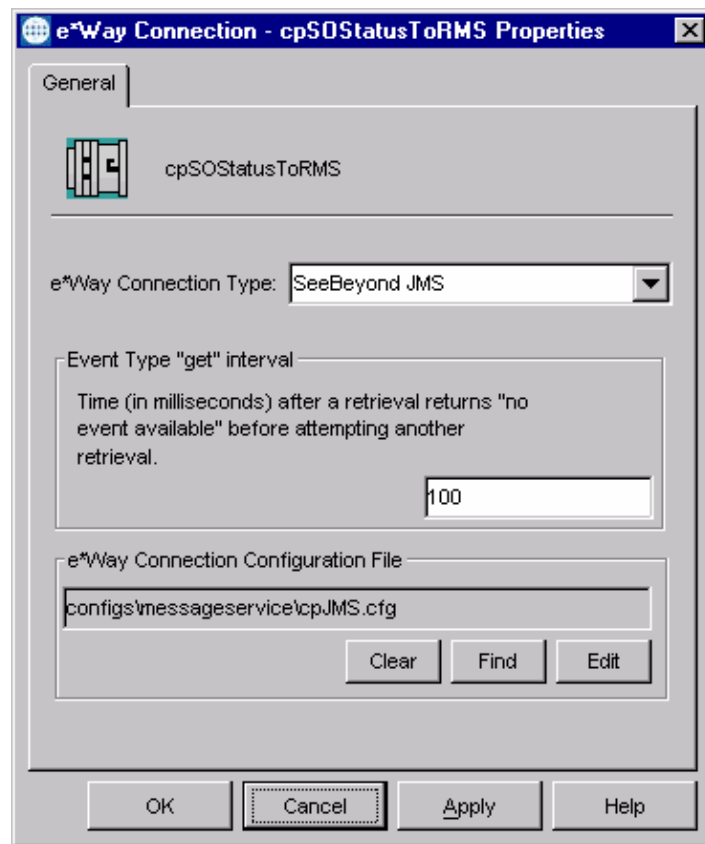


## Change SeeBeyond JMS Connection Points

The SeeBeyond JMS connection points must connect to a known JMS IQ Manager. This requires knowledge of both the port number and host name. The host name is the name of the host used in step 1 “Modify the main Participating Host and Control Broker configuration”. The TCP/IP port number is initially set to 24053. Change the TCP/IP port number only if Step 3 changed the port number of the iqmJMS IQ Manager. Otherwise, leave the port number as 24053.

Ensure that the connection point connects to the correct JMS IQ Manager. Note that in the default installation, all SeeBeyond JMS Connection points share the same configuration file. This being the case, only *one* of the SeeBeyond JMS connections needs to be modified.

- 1 Locate any one of the SeeBeyond JMS connection points, right click on it and select **Properties**. The e\*Way Connection Properties dialog box is displayed.



*e\*Way Connection Properties window*

- 2 To change the address of the JMS IQ Manager the connection point connects to, edit the configuration file from *one* of the connection points using it. Multiple connection points may use the same connection point for sending messages to and from the JMS queue. The RIB schema initially uses only a single JMS queue for all messages.

- Click **Edit** to change the address of the queue associated with the e\*Way Connection Configuration File section of this properties window. The Connection Point configuration file edit dialog box is displayed.

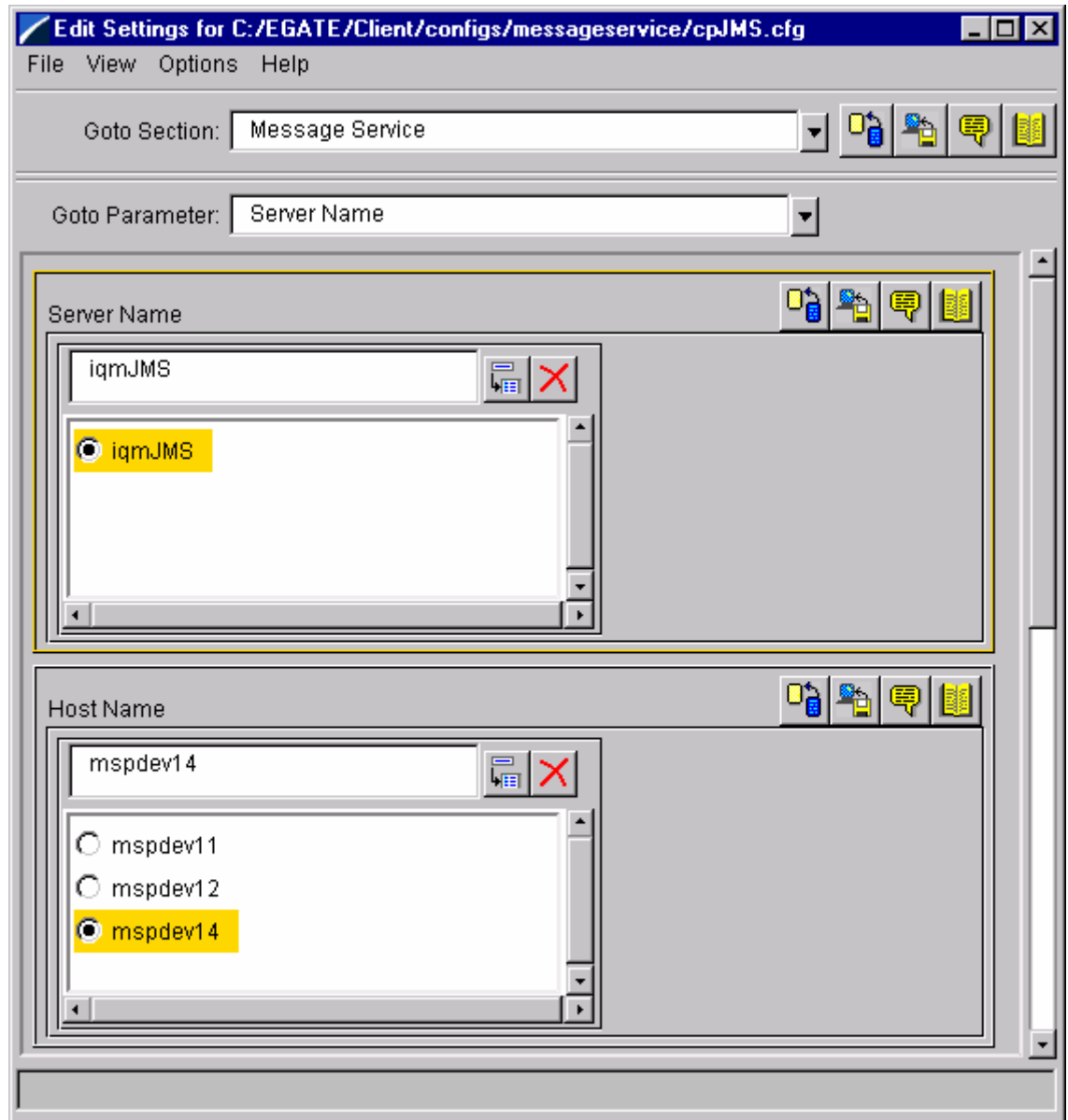
The screenshot shows a Windows-style dialog box titled "Edit Settings for C:/EGATE/Client/configs/messageservice/cpJMS.cfg". It has a menu bar with "File", "View", "Options", and "Help". Below the menu bar are two dropdown menus: "Goto Section:" set to "General Settings" and "Goto Parameter:" set to "Connection Type". The main area contains four sections, each with a title, a set of radio buttons, and a set of four icons (a document with a blue arrow, a document with a yellow arrow, a speech bubble, and a document with a red X).

- Connection Type:** Radio buttons for "Queue" and "Topic". "Topic" is selected.
- Transaction Type:** Radio buttons for "Internal", "Non-Transactional", and "XA-compliant". "XA-compliant" is selected.
- Delivery Mode:** Radio buttons for "Non-Persistent" and "Persistent". "Persistent" is selected.
- Maximum Number of Bytes to read:** A text box containing "16777216" and a radio button for "5000".

At the bottom of the dialog box, a status bar displays the text: "com.stc.common.collabService.SBYNJMSFactory" selected.

*Connection Point Configuration Edit window (General Settings Section)*

- 4 In the GoTo Section field, select the **Message Service** section from the drop-down list.



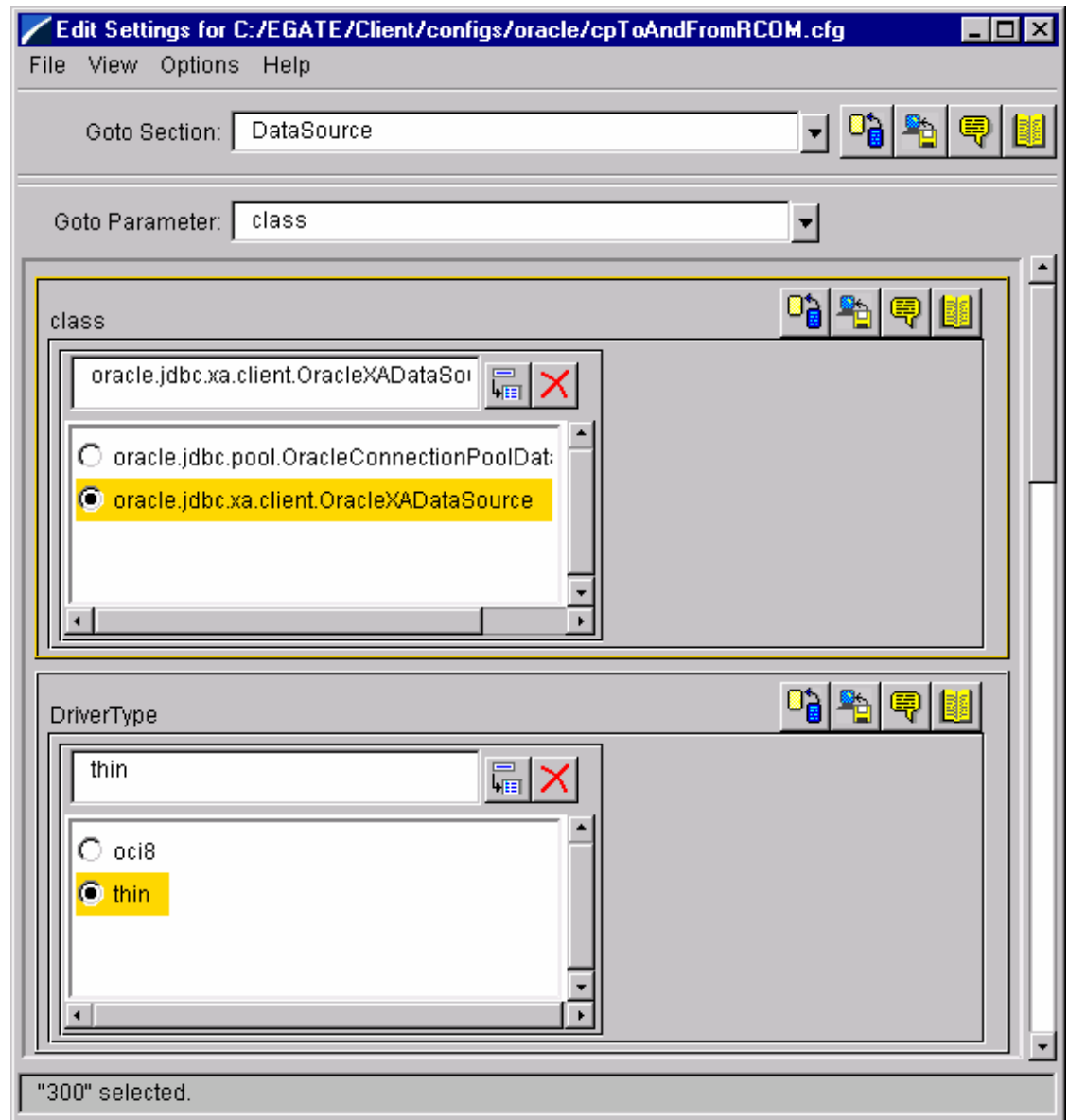
***Connection Point Configuration Edit (Message Service Section)***

- 5 In the Host Name field, enter the name of the host on which the JMS IQ Manager resides. If all components are running on the same host, this is the same name as specified in Step 1.
- 6 If the JMS IQ Manager's TCP/IP Port number was changed in Step 3, then scroll down to the Port Number field and enter the number used in Step 3.
- 7 Select File > Save to save the new configuration.
- 8 Select File > Promote To Runtime to make the configuration change take effect in the schema.
- 9 Select File > Close to exit the window.

## Changing Oracle Database Connection points

All of the Oracle database Connection points must be altered to reflect the database instance and the user-ID/login for each of the applications.

- 1 Open the Connection Point Configuration window for the Oracle Connection Points, in the same manner as was done in the previous section for the SeeBeyond JMS Connection points.



*Oracle Database Configuration Edit window (DataSource Section)*

- 2 All configuration parameters of interest are found in the **DataSource** section.
- 3 The table below lists which parameters should be changed.

| Parameter Name | Description  |
|----------------|--|
| Class          | Specifies the name of the Java class in the JDBC driver (Usually oracle.jdbc.xa.client.OracleXADataSource) |
| DriverType     | This is the JDBC driver type (Usually thin)  |
| ServerName     | Name of server to connect to. Must have a valid Oracle listener.   |
| PortNumber     | Database connection port number. (Usually 1521.)   |
| DatabaseName   | Database System ID (SID).  |
| user name      | Login name to use  |
| Password       | Login password to use. This is stored in an encrypted form and displayed as a series of asterisks.         |

- 4 Change all Oracle database connection point settings.

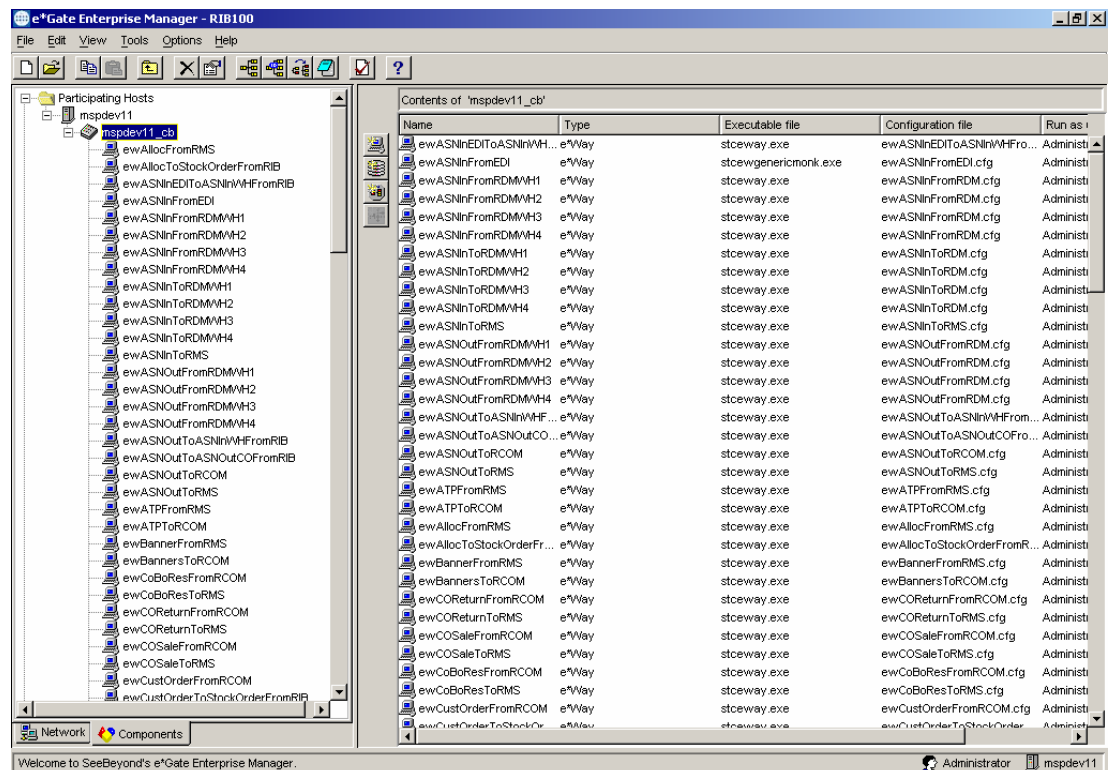
You can delete the connection points used by publishers and subscribers for applications not installed. However, there is no harm in leaving these connection points as is.

**Note:** Be sure to save and promote to runtime any changes made in the connection point configuration files.

## Step 4: Delete unused e\*Ways

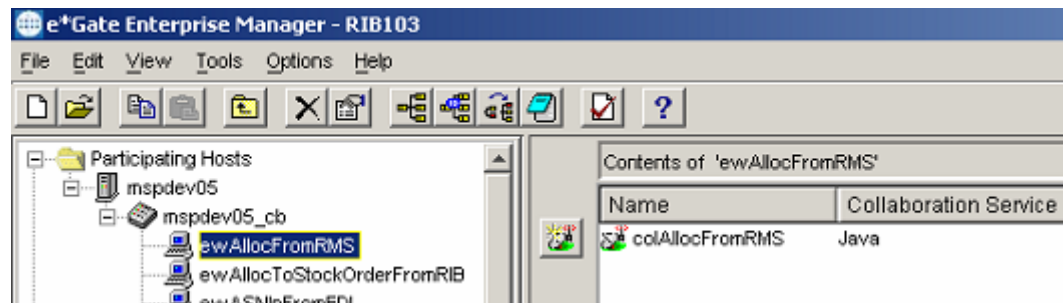
If the entire suite of RIB supported applications are not available or are not used, then delete the e\*Ways associated with these applications. For messages that are directed to multiple applications or application instances, the presence of an e\*Way will cause the JMS queue to store messages until all subscribers have received them. If a subscriber exists and never starts nor successfully consumes a message, then the JMS queue will never delete its copy of the message. Eventually, the JMS queue will exceed its configured message storage limits and message publication will halt.

- 1 From the main e\*Gate Enterprise Manager window, click on the Components tab in the lower left corner of the screen.
- 2 Expand the Participating Hosts folder in the left hand side frame, if not already expanded.
- 3 Expand the control broker containing all of the RIB e\*Ways so that the list of e\*Ways is presented.



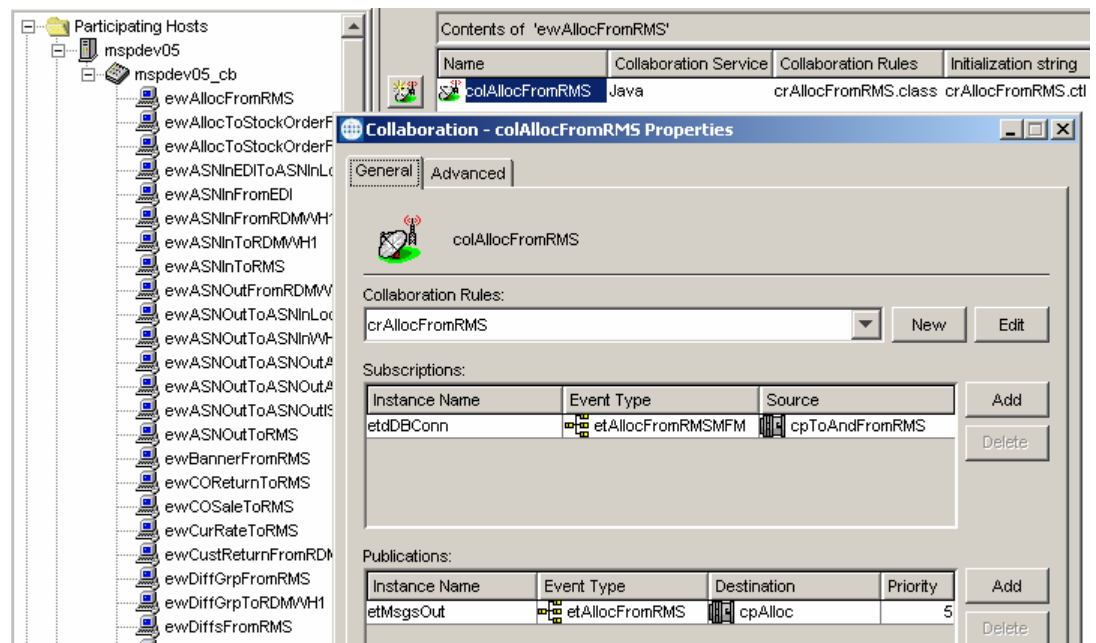
*Expanded control broker*

- 4 Highlight the e\*Way you wish to delete by clicking on it in the left hand frame. This will give you visibility in the right hand window to the collaboration associated with this e\*Way.



*Viewing a collaboration*

- 5 Open the collaboration associated with the e\*Way you wish to delete by highlighting and double-clicking on it.



*Collaboration properties window*

- 6 At this point you must determine if the e\*Way you wish to delete is a publishing e\*Way or a subscribing e\*Way. You can identify a publishing e\*Way by determining its source of subscription. This is identified in the Subscriptions section of the Collaboration –properties window (opened in step 5 above). This column indicates which connection settings the e\*Way will use. If this connection is to an outside application (I.E. RMS, RDM, ISO, RCOM,ect) this e\*Way is considered a publishing e\*Way. The naming convention employed by Retek makes these easily identifiable. Examples include cpToAndFromRMS, cpToAndFromRDMWH1, and cpToAndFromISO.

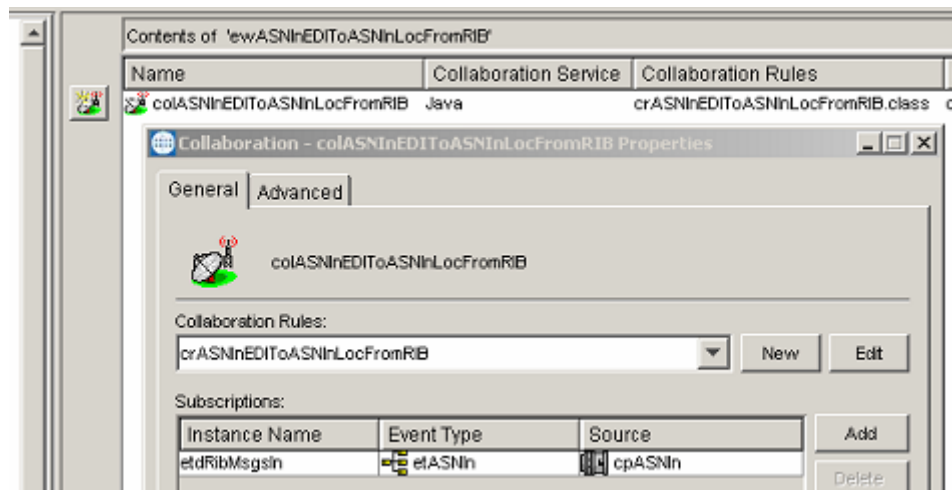
If you have determined that the e\*Way you wish to delete is a publishing e\*Way you may now skip ahead to step 8. If the e\*Way you wish to delete is a subscribing e\*Way, please proceed with the next step.

- 7 You have determined that your e\*Way is a subscribing e\*Way. Because this e\*Way is a subscribing e\*Way, there are additional pieces to remove that if left in a SeeBeyond schema could ultimately cause the performance issues alluded to in the opening paragraph.

When a registry is started in a SeeBeyond schema, it creates a unique subscriber for each subscribing e\*Way/collaboration in that schema on the JMSQueue. SeeBeyond then ensures that each subscriber receives a message once. Until all subscribers for a particular message indicate to SeeBeyond that they have handled that message, a copy of the message is kept in the JMSQueue. Because of this “journaling” of messages, we must be sure that if we remove a subscribing e\*Way from a SeeBeyond schema, we remove its subscriber as well. If we fail to do this, a copy of the message will be stored in the JMSQueue taking up memory resources.

#### Removing a Subscriber:

- a To remove a subscriber from a JMSQueue you must first identify the Topic which the e\*Way is subscribing to. The Topic can be found by opening the collaboration properties window and looking under the Event Type column in Subscriptions section of the window. Topics will always begin with the two letters “et” (I.E. etASNIn).

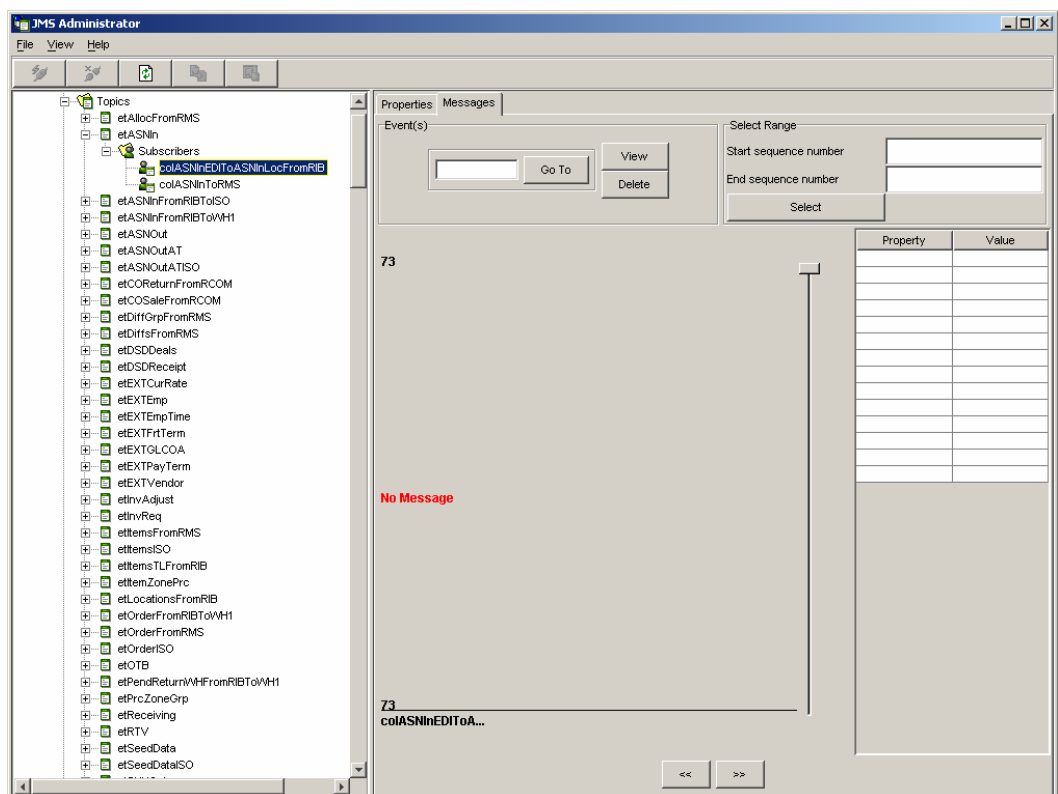


*Identify the Topic in the Event Type column*

- b Be sure the JMS IQ is running. Open the JMS Administrator tool. Instructions on use of the JMS Administrator can be found in the SeeBeyond document – JMS Intelligent Queue User’s Guide – Chapter 5: Managing Events and JMS IQ Managers.
- c From the list of topics given in the JMS Administrator, find the topic that matches the Topic you found in step A above. Once you find your topic, expand the Topic and Subscriber nodes in the GUI. Here you see a list of all collaborations that contain subscribers to this topic. Highlight the collaboration which matches the collaboration whose properties you looked at in step A.

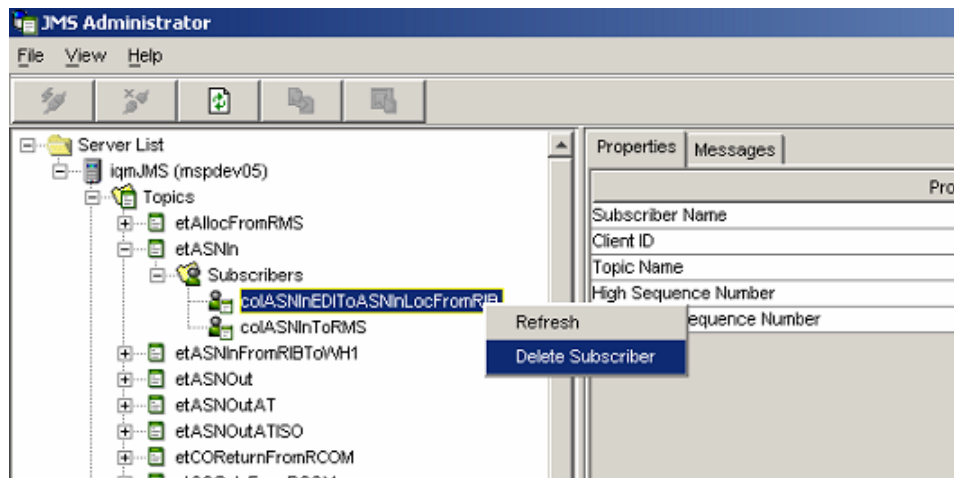


- d Before proceeding to the next step it is essential the user understand the significance of deleting a subscriber. The deletion of a subscriber also deletes any data from the JMSQueue for which that subscriber may be waiting to process. When deleting a subscriber that was not desired as part of a schema, this is of no concern. However, when deleting a subscriber because the functionality of its e\*Way has been replaced, special care must be taken to insure no loss of data occurs. Before deleting subscriber which is being replaced by another, be sure that all data in the JMSQueue for the old subscriber has been processed. This can be done by shutting down the publishing e\*Ways/process, clicking on the Messages tab in the right hand pane of the JMS Administrator and looking to see that No Message exists for the subscriber you wish to delete. If a message does exist, wait for the subscribing application to finish processing the message before proceeding.



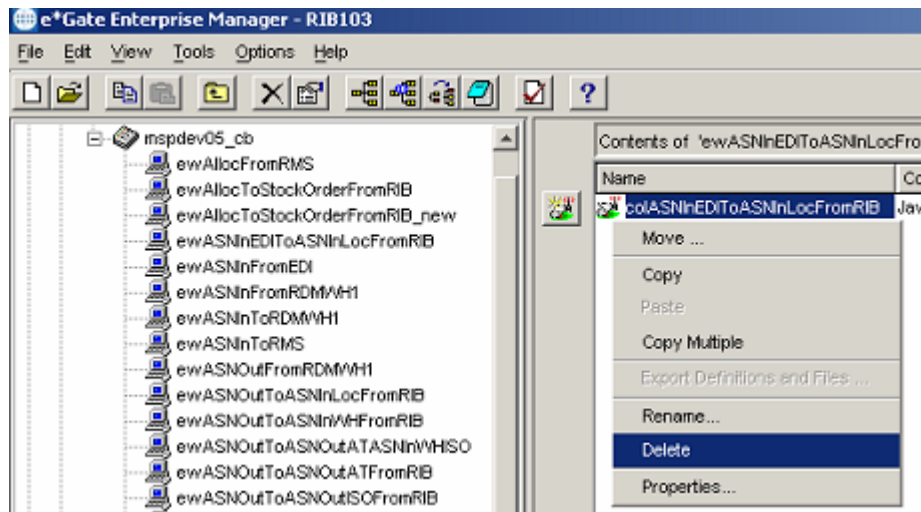
*Check to see no messages exist for subscriber*

- e Right click on the appropriate subscriber/collaboration and choose “Delete Subscriber” from the dropdown menu (note: the e\*Way which corresponds to this subscriber must not be running in order to delete).



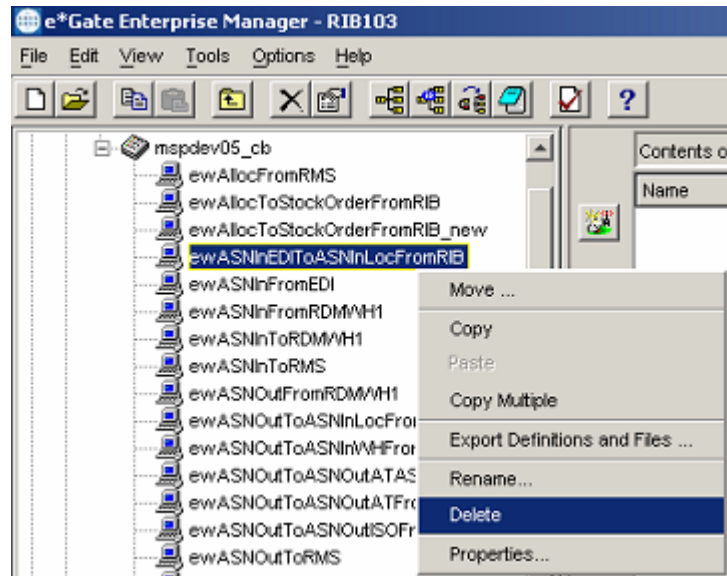
*Deleting a subscriber*

- f Close the JMS Administrator and return to the e\*Gate Enterprise Manager.
- 8 Right-click on the collaboration. Choose **Delete** to delete the collaboration. A confirmation dialog box is displayed.



*Deleting the collaboration*

- 9 Right-click on the e\*Way. Choose **Delete** to delete the e\*Way. A confirmation dialog box is displayed.



#### *Deleting the e\*Way*

- 10 Log onto the Unix system using the “egate” user that owns the e\*Gate files and executes the software.
- 11 Navigate to the Unix \$HOME directory. The Egate.txt file contains a listing of all of the e\*Ways contained the RIB 10.3.2 release. Modify this file to no longer contain a reference to the e\*Way you have just removed from your schema.
- 12 Repeat this process to delete all e\*Ways which are not desired.

### Step 5: Add/Copy e\*Ways for additional components

Occasionally, there will be a need to add additional e\*Ways to the imported schema. Often times, it is possible to copy an existing e\*Way, reconfigure the various pieces that make up the e\*Way, and continue from there. If it is necessary to add/copy e\*Ways, please refer to the SeeBeyond e\*Gate Users Guide for the correct procedures.

## Step 6: Edit the rib.properties file to correspond to the system

Next, edit the rib.properties file to reflect the site-specific mappings and properties. The RIB import process copied a rib.properties file from <RETEK\_INSTALL\_DIR>/RIB<version>/Rib\_Support/src to the \$EHOME/client/classes directory. If a previous version of this file existed, it was renamed to rib.properties.bak. Some of the types of properties in this file are:

- Error Hospital specifics such as the max number of attempts to try for a failing message and the delay between each attempt.
- Multi-channel indication for Purchase Orders.
- Multi-threading settings.
- Facility ID mappings. These must correspond to codes in the RMS application for the correct routing of message to RDM instances.
- Logging specific settings.
- “no event” sleep duration settings.
- Directory location of RIBLOGS logging files.

If the rib.properties file in \$EHOME/client/classes is not to be used, then append the full path of the directory containing the file that will be used to the beginning of the CLASSPATH variable setting.

See the Retek Integration Bus Operations Guide for more information on the values for the rib.properties file.

## Step 7: Create/modify startup scripts

The final installation step is to create RIB startup scripts for Unix systems. These scripts start up the SeeBeyond e\*Gate registry and the control broker for the RIB103 Schema. Default scripts are provided for starting the registry service and the control broker in the \$EHOME directory. Refer to the “start\_egate” and a “start\_cb” scripts.

The commands that implement this functionality are stcregd and stccb. When executed, they run as daemons. Depending on logging and other parameters, they may log items to their stderr or stdout files. These commands are detailed further in the following manuals:

- Retek Integration Bus Operations Guide
- SeeBeyond e\*Gate Integrator User’s Guide
- SeeBeyond e\*Gate Integrator System Administration and Operations Guide

## Step 8: Starting the RIB components for a new/modified schema

Start\_rib is intended for use in starting the e\*Way components of a new schema or when new e\*Way components have been added to an existing schema. The RIB 10.3.2 architecture requires that the first time an e\*Way is started in a schema, it is started in a particular order (subscriber, TAFR, publisher). The start\_rib script uses an input file called Egate.txt to identify an e\*Way as a subscriber, a TAFR, or a publisher. Egate.txt contains a listing of all of the e\*Way components packaged with RIB 10.3.2. Start\_rib is executed with one or several of the parameters listed within its usage. To view the start\_rib usage, change Unix directories to \$EHOME and run start\_rib with no parameters.

It is necessary that start\_rib is run to start the e\*Way components immediately after installation and configuration of the RIB 10.3.2 components. The following command should be executed while the registry and control broker are running: start\_rib JMS SUB TAFR. After these processes have successfully started (check to be certain connection points were accurately set and DB connection was successful), if the client is integrating with ISO, start the ISO services following SIM instruction. Run start\_rib once more using the following command: start\_rib PUB HOSP. All of the e\*Way process listed in the Egate.txt file should now be running.

If desired, new e\*Way components may be started using the start\_rib command by adding the new components to the start\_rib input file - Egate.txt. This file can be located in the \$EHOME directory.

If a full install is not being preformed, and integrating with ISO, proceed to [Chapter 7 – ISO Integration](#).



## Chapter 4 – Database triggers and Oracle dependencies

### Database triggers

Once the RIB has been installed and configured, the publishing applications need to be told to begin to publish data. There are multiple ways to initiate the publishing process. Each Retek product's operation guide contains this information.

### Oracle dependencies

In order for the Retek 10.3 RIB to function correctly, *you must install Oracle's XML Developer's Kit for PL/SQL on your database server*. This can be downloaded from Oracle Technology Network. The version of XML Developer's Kit for PL/SQL *must be dated 3/28/2002 or later* – there is a bug in prior version that will prevent the RIB from working correctly.

### Verify RIB Error Hospital database tables

One feature of the RIB is the Error Hospital subsystem used to store and retry messages that have processing problems by a subscribing application. This facility allows for non-dependent messages to continue to be processed by the application until the failure has been resolved and the message successfully consumed.

These tables were created during the database portion of the RCOM, RDM, or RMS installations. The DDL to create these tables can be found on these products' installation CDs.

**Note:** For more detailed information on the RIB Error Hospital, refer to the <RETEK\_INSTALL\_DIR>/RIB103/Rib\_Support/doc/index.html file.

For each Error Hospital, verify that the three hospital tables exist and that they have the correct columns. The three tables necessary are:

### **RIB\_MESSAGE**

```
DROP TABLE RIB_MESSAGE CASCADE CONSTRAINTS ;

CREATE TABLE RIB_MESSAGE (
    MESSAGE_NUM          NUMBER (8)      NOT NULL,
    LOCATION              VARCHAR2 (60)   NOT NULL,
    FAMILY                VARCHAR2 (25)   NOT NULL,
    TYPE                  VARCHAR2 (30)   NOT NULL,
    ID                    VARCHAR2 (255),
    RIB_MESSAGE_ID        VARCHAR2 (255),
    PUBLISH_TIME          DATE,
    IN_QUEUE              VARCHAR2 (1)    NOT NULL,
    MESSAGE_DATA           CLOB           NOT NULL,
    ATTEMPT_COUNT          NUMBER (4)      NOT NULL,
    MAX_ATTEMPTS           NUMBER (4)      NOT NULL,
    NEXT_ATTEMPT_TIME     DATE,
    DELETE_PENDING        VARCHAR2 (1)    NOT NULL,
    TOPIC_NAME             VARCHAR2 (255),
    THREAD_VALUE           NUMBER (22,8),
    JMS_QUEUE_ID          VARCHAR2 (255),
    CUSTOM_FLAG            VARCHAR2 (1)    DEFAULT 'F' NOT NULL,
    CUSTOM_DATA            CLOB,
    REASON_CODE            VARCHAR2 (10)   NOT NULL,
    CONSTRAINT PK_RIB_MESSAGE
    PRIMARY KEY ( MESSAGE_NUM ) ) ;
```

**Note:** The field MESSAGE\_NUM in the table RIB\_MESSAGE has an associated script to create the sequence. It is required to run this script – **rib\_message\_seq.sql**. This script can be found in the <RETEK\_INSTALL\_DIR>/RIBfor<APP><version>/XML\_Uilities directory.



**RIB\_MESSAGE\_FAILURE**

```
DROP TABLE RIB_MESSAGE_FAILURE CASCADE CONSTRAINTS ;
```

```
CREATE TABLE RIB_MESSAGE_FAILURE (
    MESSAGE_NUM    NUMBER (8)      NOT NULL,
    SEQ_NUMBER     NUMBER (2)      NOT NULL,
    TIME           DATE            NOT NULL,
    LOCATION       VARCHAR2 (60)   NOT NULL,
    DESCRIPTION    VARCHAR2 (4000) NOT NULL,
    CONSTRAINT PK_RIB_MESSAGE_FAILURE
    PRIMARY KEY ( MESSAGE_NUM, SEQ_NUMBER ) ) ;
```

**RIB\_MESSAGE\_ROUTING\_INFO**

```
DROP TABLE RIB_MESSAGE_ROUTING_INFO CASCADE CONSTRAINTS
;
```

```
CREATE TABLE RIB_MESSAGE_ROUTING_INFO (
    MESSAGE_NUM    NUMBER (8)      NOT NULL,
    SEQ_NUMBER     NUMBER (2)      NOT NULL,
    NAME           VARCHAR2 (25)   NOT NULL,
    VALUE          VARCHAR2 (25)   NOT NULL,
    DETAIL1_NAME   VARCHAR2 (25),
    DETAIL1_VALUE  VARCHAR2 (300),
    DETAIL2_NAME   VARCHAR2 (25),
    DETAIL2_VALUE  VARCHAR2 (300),
    CONSTRAINT PK_RIB_MESSAGE_ROUTING_INFO
    PRIMARY KEY ( MESSAGE_NUM, SEQ_NUMBER ) ) ;
```

**Note:** If these tables need to be created outside of a Retek Application's installation2 scripts called 'rib\_ddl.sql' and 'rib\_message\_seq.sql' can be found in the <RETEK\_INSTALL\_DIR>/RIBfor<APP><version>/XML\_Uilities directory.

If your database already has these tables in place from a RIB installation prior to version 10.3, the rib\_message table may need to be updated with new table changes. Executing the 0001\_rib\_message.sql script, which can be found in the <RETEK\_INSTALL\_DIR>/RIB103/DBC directory, applies these changes to the table. If the rib\_message table contains data, the data should be exported before the script is run, and imported back into the table.

Check the structure of the rib\_message table, prior to running this script, to determine if this script needs to be applied or not.

**Note:** This script contains table changes that include the addition of a new NON-nullable column, 'custom\_flag' (position 17), that should be defaulted to 'F' when importing data back into the table. All other new columns can be null.

## RIB\_DOCTYPES table and DTD files

The RIB\_DOCTYPES table should also have been created and populated by the Retek Application installation(s) (e.g. RMS). The integration with RCOM now requires that the DTD files themselves be network accessible.

To accomplish this, all of the DTD files should be deployed to a web server running at the client. Once this is done, and the URL to the DTDs is known, the value in the DOC\_TYPE\_URL column will need to reflect this location.

The rib\_doctypes\_rms.dat file which is found in the <RETEK\_INSTALL\_DIR>/RIBforRMS/XML\_Uutilities directory is the script that RMS uses to populate the table with data. The URLs can be globally replaced with the new one and the script re-run.

**Note:** If custom RMS, a custom version of this script will exist. It should be applied instead.

## Chapter 5 – RIB Administration Tool installation

The RIB Administration Tool is a web-based tool provided to perform RIB administration functions, which include the Hospital Administration GUI, the Message Statistics GUI and the RIB Properties Editor GUI.

There are two versions of the Hospital Administration GUI that can be used to administer problem messages that have been put in the hospital database tables. One version is a java executable/application and one is web/browser-based. Retek recommends you use the web/browser-based version, if possible, because it does not require anything other than java runtime to be installed on individual workstations.

This section explains the installation of the web-based version; see the next section for the executable/application-based version of the Hospital Administration GUI.

### Web-based version install

#### Prerequisites

- 1 A J2EE-compliant web server from which to deploy the gui.war file. Jakarta Tomcat is the web server that the RIB Administration Tool was developed on, and is the recommended web server to use.
- 2 Java Runtime Engine (JRE) version 1.4 installed on all workstations/PCs that will be accessing the GUI via their web browser.

**Note:** If these two prerequisites cannot be met, install the java executable/application version of the Hospital UI (see next section).

#### Install RIB Administration Tool

- 1 Copy gui.war from the  
`<RETEK_INSTALL_DIR>/RIB<version>/Rib_Hospital_Gui/build` directory to the web server and deploy it. If deploying on Tomcat, place the gui.war file in the Tomcat /webapps directory and reload the server.
- 2 After deployment, locate and modify the gui.properties configuration file in the web application directory (for Tomcat, this would be in `$CATALINA_HOME/webapps/gui/`). The entries in this file that must be changed are listed below:

```
#####
# GUI Project Variables
GUI.ProjectHost=
GUI.ProjectPort=
GUI.ProjectName=gui
GUI.TimingsLogFile.Path=
GUI.TimingsLogFile.Name=timings_rib.log
GUI.rib.properties.default.FilePath=
```

```
GUI.rib.properties.default.BackupFileExt=.bak
```

Where:

- GUI.ProjectHost is the name or IP address of the server that the J2EE web server is running on. GUI.ProjectPort is the http listener port of the J2EE web server. These are the values that will be set for all of the GUI applets, and will override the applet's `baseurl.getContext` lookups to find the URL to the servlets. If for any reason this lookup does not find the correct host and port, or if a servlet residing on a different host or port is preferred, set these values appropriately.
- GUI.ProjectName should be set in the properties file to contain the name of the project installation (installed application name) on the web server. The applets will use this name to build the URL to the servlets. The default installation name is "gui".
- GUI.TimingsLogFile.Path and the GUI.TimingsLogFile.Name should be set to contain the default path to the timings log file and the default name for the log file for the Message Statistics GUI Applet. When this applet is loaded, it will display a window where the user can enter the path to the log file and the parameters to pass into the RibTimings class to gather the statistics. If no value is entered, the log file path text field on this window will initially be blank. The TimingsLogFile.Name is defaulted to "timings\_rib.log".
- GUI.rib.properties.default.FilePath should be set to the default file path of the rib.properties file. This will be displayed in the RIB Properties Editor's connection window as the default File Name, which the user can modify before retrieving the file from the server.
- GUI.rib.properties.default.BackupFileExt should be set to contain the default file extension the RIB Properties Editor will use when creating a backup copy of the rib.properties on the server. This will be displayed in a dialog that appears on saving the file. The user can modify the extension of the backup file to whatever they choose before the file is saved.

Example:

```
#####
# GUI Project Variables
GUI.ProjectHost=localhost
GUI.ProjectPort=8080
GUI.ProjectName=gui
GUI.TimingsLogFile.Path=/files0/egate/timings/
GUI.TimingsLogFile.Name=timings_rib.log
GUI.rib.properties.default.FilePath=/files0/egate/egate/
client/classes/
GUI.rib.properties.default.BackupFileExt=rib.properties.
bak
```

**Note:** All File Path entries in this properties file should end with a file separator character, since the file name will be appended to the end of the path (ie “/” or “\”).

- 2 Locate and modify the gui.servlet.properties file in the web application directory under WEB-INF/classes (for Tomcat, this would be in \$CATALINA\_HOME/webapps/gui/WEB-INF/classes). The entries in this file that can be changed are listed below:

Where:

```
#####
# GUI Project Variables
GUI.jdbc.driver=oracle.jdbc.driver.OracleDriver
GUI.rib.properties.SessionTimeout=900
GUI.rib.properties.local.FilePath=
```

- GUI.jdbc.driver should be set to the driver used to log in to the database for the main Portal login. The default driver that is contained the gui.war is an Oracle database driver.
- GUI.rib.properties.SessionTimeout should be set to the amount of time in which a session is timed out after being idle. The index.jsp will set the HttpSession.setMaxInactiveInterval(); The default is 900 seconds (15 minutes).
- GUI.rib.properties.local.FilePath should be set to the directory where the Rib Properties should locally save the file while editing it. The default is to set this to <appserver-installation-directory>/<installed-application-name>/temp/, but can be changed to any directory on the application server.

Example:

```
#####
# GUI Project Variables
GUI.jdbc.driver=oracle.jdbc.driver.OracleDriver
GUI.rib.properties.SessionTimeout=900
GUI.rib.properties.local.FilePath=/files0/jakarta-
tomcat/webapps/gui/temp/
```

**Note:** All File Path entries in this properties file should end with a file separator character, since the file name will be appended to the end of the path (ie “/” or “\”).

## Internationalization

The RIB Administration Tool is initially configured to run in English. To set the RIB Administration Tool to the language of your choice:

- 1 Edit two files in the web application directory: `gui.properties` and `gui.servlet.properties`. Set the `GUI.language` and `GUI.country` values to the language you wish to use. If no language or country code is specified for these properties, the applets will default to the locale in which they are running.

The following settings are available:

For English:        `GUI.language=en`

`GUI.country=US`

For French:        `GUI.language=fr`

`GUI.country=FR`

For German:        `GUI.language=de`

`GUI.country=DE`

For Spanish:        `GUI.language=es`

`GUI.country=ES`

For Korean:        `GUI.language=ko`

`GUI.country=KO`

For Japanese:        `GUI.language=ja`

`GUI.country=JP`

Default locale:        `GUI.language=`

`GUI.country=`

- 2 In the web application directory, there is a folder containing specific files for each language (based on the language code : DE, EN, ES, FR, JA, KO). Copy the files from this directory into the web application directory (overwriting the existing files).
- 3 If Jakarta Tomcat is the web server being used, it must be reloaded for the changes in steps 1 and 2 above to take effect. Other web servers may require reloading as well.

## Install JRE

- 1 Java Runtime Engine (JRE) 1.4 must be installed on all workstations/PC's that will be accessing the GUI via their web browser.

**Note:** The 1.4 JRE can be downloaded at no charge from Sun's Java web site (<http://java.sun.com/j2se/downloads.html>).

## Test Error Hospital GUI Applet

- 1 To test the GUI, enter the following URL in a browser:

<http://<server>:<port>/<ProjectName>/index.jsp>

- Server = name or IP address of the server that the J2EE web server is running on (should be the same value as GUI.ProjectHost in gui.properties)
- Port = http listener port of the J2EE web server (should be the same value as GUI.ProjectPort in gui.properties)
- ProjectName = value of GUI.ProjectName in gui.properties

Example: <http://localhost:8080/gui/index.jsp>

## Files and classes contained in the war file

### Classes

**com.retek.rib.gui.AppletCoder:** used for encoding and decoding information sent from applets to servlets

**com.retek.rib.gui.AppletDialog:** used for error message dialogs for all applets

**com.retek.rib.gui.DBConnection:** used by index.jsp to test authentication with main RIB Administration login

**com.retek.rib.gui.GUIHelp:** used to display help dialog for RIB Administration Tool

**com.retek.rib.gui.HospitalUIApplet:** main Hospital Administration class, contains all applet GUI code

**com.retek.rib.gui.HospitalUIDBHelper:** Hospital Administration class, contains TableModel implementation and command calls

**com.retek.rib.gui.HospitalUIHelper:** Hospital Administration class, contains calls to servlet

**com.retek.rib.gui.HospitalUIQueries:** Hospital Administration class, contains queries to retrieve and update the message data

**com.retek.rib.gui.HospitalUIServlet:** Hospital Administration servlet class

**com.retek.rib.gui.PropertiesUI:** main RIB Properties Editor class, contains all applet GUI Code

**com.retek.rib.gui.PropertiesServlet:** RIB Properties Editor servlet class

**com.retek.rib.gui.PropsHelper:** RIB Properties Editor class, contains calls to servlet

**com.retek.rib.gui.ServletHelper:** used for generic servlet methods

**com.retek.rib.gui.StatisticsUI:** main Message Statistics class, contains all applet GUI code

**com.retek.rib.gui.StatsDBHelper:** Message Statistics class, contains TableModel implementation

**com.retek.rib.gui.StatsHelper:** Message Statistics class, contains calls to servlet

**com.retek.rib.gui.TableMap** and **com.retek.rib.gui.TableSorter:** classes used for TableModel implementation for both applets

**com.retek.rib.gui.TimingsServlet:** Message Statistics servlet class

## Jars and other files

**js/apps.js:** javascript file for RIB Administration index page

**taglibs/gui.tld:** tag library for RIB Administration index page

**WEB-INF/lib/classes12.jar:** contains Oracle Database Driver

**WEB-INF/lib/retex-rib-support.jar:** contains base code for Hospital Administration and Message Statistics functionality

**WEB-INF/lib/retex-sbyn.jar:** contains base code for Hospital Administration

**WEB-INF/lib/etdRibMessages.jar:** contains base code for Hospital Administration

**WEB-INF/lib/stcjs.jar:** contains base code for Hospital Administration

**WEB-INF/web.XML:** contains servlet mappings and session defaults

**WEB-INF/classes/gui.servlet.properties** and **gui.properties:** properties files used by RIB Administration Tool and applets

**WEB-INF/classes/rib.properties:** properties file used for Hospital Administration



## Language-specific files

**lang/\*.properties:** Language-specific properties files for the java resource bundles to provide displayed text in the RIB Administration Tool in the proper language

- The language-specific properties files consist of:
  - GUIHelp\_\*.properties:** Resource bundle for the GUIHelp applet
  - HospitalUI\_\*.properties:** Resource bundle for the HospitalUIApplet
  - RibPropertiesUI\_\*.properties:** Resource bundle for the RibPropertiesUI applet
  - Servlet\_\*.properties:** Resource bundle for the ServletHelper
  - StatisticsUI\_\*.properties:** Resource bundle for the StatisticsUI applet
  - DE/\*:** German HTML and JSP files
  - EN/\*:** English HTML and JSP files
  - ES/\*:** Spanish HTML and JSP files
  - FR/\*:** French HTML and JSP files
  - JA/\*:** Japanese HTML and JSP files
  - KO/\*:** Korean HTML and JSP files
- The language-specific JSP files consist of:
  - errorpage.jsp:** error page for RIB Administration index and login pages
  - index.jsp:** main index page for RIB Administration
  - login.jsp:** main login page for RIB Administration
- The language-specific HTML files consist of:
  - HospitalUIHelp.html:** Help file for the HospitalUIApplet
  - GUIHelp.html:** Help file for the GUIHelp applet
  - StatisticsHelp.html:** Help file for the StatisticsUI applet
  - PropertiesUIHelp.html:** Help file for the RibPropertiesUI applet



## Chapter 6 – WebSphere Integration

This chapter will briefly review the configuration required for integrating with RCOM running in a WebSphere 5.0 application server. All of this is done as part of the installation of the RCOM application, but is noted here for reference.

### rib.properties file

**Note:** The editing of this file is documented in the RCOM Installation Guide.

In the rcom-j2ee-rib.jar file, you will find a file named rib.properties. This file contains the RIB specific properties used by the RIB subscribing Message-Driven Beans and publishing Stateless Session Beans that are deployed on the Application Server. See below.

### Error Hospital entries

```
#####
# These are the RIB hospital properties.
hospital.attempt.max=5
hospital.attempt.delay=10
hospital.attempt.delayIncrement=10
```

This section details the entries used for retrying messages from the Error Hospital.

**hospital.attempt.max** – This is the maximum number of attempts to try to push this record through the RIB automatically, once this retry count is exceeded the message remains the Error Hospital DB but is no longer retried automatically.

**hospital.attempt.delay** – value (in seconds) used to calculate the next attempt time

**hospital.attempt.delayIncrement** – value (in seconds) used to calculate the next attempt time.

The next attempt time is calculated as:

$\text{hospitalAttemptDelay} + (\text{hospitalAttemptDelyIncrement} * \text{attempt count})$

This is done so that the delay between each attempt is longer than the previous delay.

## Logging entries

```
#####
# Default logging level verbose? [Y or N]
log.default.verbose=N
#####
# Path where RIB and Timings log files will be written.
# It must end with
# a directory separator / or \.
log.default.file_path=/files2/websph/WebSphere/AppServer
/logs/server1/
```

The RIB has its own logging capabilities. The RIB support Java classes contain logging logic which write to RIB log files which are written to a user specified directory.

**log.default.file\_path** - Path where RIB and Timings log files will be written. It must end with a directory separator / or \.

**log.default.verbose** – [ Y or N] Specifies whether or not the RIB code should be logged at a verbose level.

## JNDI/JMS Configuration

```
#####
# These are JNDI names used to lookup DataSource and
# TransactionManager
# used by the RIB.
rcom.jndi.db=jdbc/OracleRibDs
rcom.jndi.jms.factory=XAConnectionFactory
#####
# This is the hostname and port of the eGate JMS
# provider.
rib.jms.hostname=<servername> e.g. mspdev14.retek.int
rib.jms.port=<portname> e.g. 24053
# Write each JMS message (XML) out to a file? [Y, N,
# True or False]
rib.jms.write_file=False
#####
# These properties are used to interface with RCOM
# (J2EE). Only applicable
# if RIB is not deployed in same AppServer Container.
rcom.jndi.context.factory=com.ibm.websphere.naming.WsnInitialContextFactory
rcom.jndi.url=iiop://mspdev03.retek.int:2809
```

These configurations specify the locations of the JNDI naming service and the JMS Queue.

**rcom.jndi.context.factory** – This is the JNDI context factory to use for JNDI lookups. This is defaulted to the WebSphere Initial Context Factory.

**rcom.jndi.url** – This is the URL to the JNDI service.

**rcom.jndi.db** – This is the JNDI name for the data source.

**rcom.jndi.jms.factory** – This is the JNDI name for the JMS XA Topic Connection Factory. This is used for creating a Topic Connection in order to place messages on the JMS Queue.

**rib.jms.hostname** – This is the JMS hostname (servername) of the JMS Queue.

**rib.jms.port** – This is the JMS port of the JMS Queue.

**rib.jms.write\_file** – This property specifies whether or not to write each JMS XML message out to a file.

## Implementation classes

```
#####
# Version of AlertPublisher, RibMessage, etc. the RIB is
# using.
alertPublisherImpl=com.retek.rib.alert.NullAlertPublisher
ribMessageImpl=com.retek.rib.sbyn.RibMessageWrapper
ribMessagesImpl=com.retek.rib.sbyn.RibMessagesWrapper
routingInfoImpl=com.retek.rib.sbyn.RoutingInfoWrapper
failureImpl=com.retek.rib.sbyn.FailureWrapper
```

In order to promote pluggable, platform specific implementations, the RIB allows the specification of platform-specific classes for a variety of functions. These functions include the actual creation of a RibMessages XML message and the interface to an alert mechanism. The following entries are used to specify what Java classes should be used for these functions:

**alertPublisherImpl** -- Interface to the Alerting mechanism

Values: com.retek.rib.sbyn.alert.EgateAlertPublisher (SeeBeyond)

**ribMessageImpl** – Class used to create a ribMessage node within a RibMessages container.

Values: com.retek.rib.sbyn.RibMessageWrapper (SeeBeyond)

**ribMessagesImpl** – Class used to create a RibMessages container.

Values: com.retek.rib.sbyn.RibMessagesWrapper (SeeBeyond)

**routingInfoImpl** – Class used to create the Routing Information Section within a ribMessage node.

Values: com.retek.rib.sbyn.RoutingInfoWrapper (SeeBeyond)

**failureImpl** – Class used to create, store and copy message failure information

Values: com.retek.rib.sbyn.FailureWrapper (SeeBeyond)

## Publishing configuration

```
#####
#List of publishing topics for each message family
COBORES=etCoBoResFromRCOM
CODSRCPT=etCODSRcptFromRCOM
CORETURN=etCOReturnFromRCOM
CORRESPONDENCE=etCorrespondenceFromRCOM
COSALE=etCOSaleFromRCOM
CUSTORDER=etCustOrderFromRCOM
DSPO=etDSPOFromRCOM
INVADJUST=etInvAdjust
PAYMENTS=etPaymentsFromRCOM
PENDRETURN=etPendReturnFromRCOM
WOSTATUS=etWOStatusFromRCOM
WOINT=etWOIntFromRCOM

#####
#Flag to force all published messages into the hospital
instead of JMS
publisher.force_hospital=False
```

These properties are used to configure publishing from WebSphere to the JMS Queue.

**<family-name>=<topic-name>** - These properties map a publishing family name to the correct topic name on the JMS Queue (eg. COBORES = etCoBoResFromRCOM).

**publisher.force\_hospital** – This property forces all messages published from WebSphere directly to the hospital table in the database instead of the JMS Queue.

Example:

```
#####
# These are the RIB hospital properties.
hospital.attempt.max=5
hospital.attempt.delay=10
hospital.attempt.delayIncrement=10
#####
# Default logging level verbose? [Y or N]
log.default.verbose=N
#####
# Path where RIB and Timings log files will be written.
# It must end with
# a directory separator / or \.
log.default.file_path=/files2/websph/WebSphere/AppServer
/logs/server1/
#####
# Log message times? [Y or N], and the file to write
# timings log
# entries to. Only specify the file name as it will be
# prepended
# with the log.default.file_path property. If no
# entries for an
# e*Way, it will default to N.
#
log.MDB.timings=Y
log.MDB.timings_logfile=timings_rib.log
#####
# These are JNDI names used to lookup DataSource and
# TransactionManager
# used by the RIB.
rcom.jndi.db=jdbc/OracleRibDs
#rcom.jndi.tm=java:comp/env/TransactionManager
rcom.jndi.jms.factory=XAConnectionFactory
#####
# This is the hostname and port of the eGate JMS
# provider.
rib.jms.hostname=<servername> e.g. mspdev14.retek.int
rib.jms.port=<portname> e.g. 24053
# Write each JMS message (XML) out to a file? [Y, N,
# True or False]
```

```

rib.jms.write_file=False
#####
# Version of AlertPublisher, RibMessage, etc. the RIB is
using.
alertPublisherImpl=com.retek.rib.alert.NullAlertPublishe
r
ribMessageImpl=com.retek.rib.sbyn.RibMessageWrapper
ribMessagesImpl=com.retek.rib.sbyn.RibMessagesWrapper
routingInfoImpl=com.retek.rib.sbyn.RoutingInfoWrapper
failureImpl=com.retek.rib.sbyn.FailureWrapper
#####
# These properties are used to interface with RCOM
(J2EE). Only applicable
# if RIB is not deployed in same AppServer Container.
#rcom.jndi.context.factory=com.ibm.websphere.naming.WsnI
nitialContextFactory
#rcom.jndi.url=iiop://mspdev03.retek.int:2809
#####
#List of publishing topics for each message family
COBORES=etCoBoResFromRCOM
CODSRCPT=etCODSRCptFromRCOM
CORETURN=etCOReturnFromRCOM
CORRESPONDENCE=etCorrespondenceFromRCOM
COSALE=etCOSaleFromRCOM
CUSTORDER=etCustOrderFromRCOM
DSPO=etDSPOFromRCOM
INVADJUST=etInvAdjust
PAYMENTS=etPaymentsFromRCOM
PENDRETURN=etPendReturnFromRCOM
WOSTATUS=etWOStatusFromRCOM
WOINT=etWOIntFromRCOM

#####
#Flag to force all published messages into the hospital
instead of JMS
publisher.force_hospital=False

```



## rib\_subscriber.xml file

Log4j is used for logging and performance measurements in the RIB Publishing and Subscribing code. These logs are created based on the family type of the message being processed. The logging is defined in the rib\_subscriber.xml file. The editing of this file is defined in the RCOM installation guide.

Each publishing and subscribing family has its own logger and appender defined. This controls the logger name and associated file created by log4j. An example of an ASNOut family logger and appender is shown below. This same template is used for every family, with the “ASNOut” replaced for the new family name.

Example:

```
<!-- ASNOUT -->
<logger name="rib.sub.timings.asnout">
    <level value="INFO"/>
    <appender-ref ref="aASNOut" />
</logger>
<appender name="aASNOut"
class="org.apache.log4j.FileAppender">
    <param name="File" value="/Program
Files/WebSphere/AppServer/logs/server1/timings_asnout.log"/>
    <layout class="org.apache.log4j.PatternLayout">
        <param name="ConversionPattern"
value="%d{yyyy.MM.dd
HH:mm:ss.SSS} | ASNOutToRCOM | %t | | %m%n"/>
    </layout>
</appender>
```

## Generic JMS Provider

The Generic JMS Provider is fully configured as part of the RCOM installation. From the WebSphere Admin Console, click Resources -> Generic JMS Providers. You will see “**SeeBeyond JMS Provider**” as the available resource. The JMS Connection Factory as well as the JMS Destinations is defined here.

## Message Listener Ports

The Message Listener Ports are also fully configured as part of the RCOM installation. From the WebSphere Admin Console, click Servers -> Application Servers -> server1 -> Message Listener Service -> Listener Ports. You will see all of the WebSphere Listener Ports defined here.

## Data Source

Finally, the Oracle DataSources are fully configured as part of the RCOM installation. From the WebSphere Admin Console, click Resources -> JDBC Providers. You will see “**Oracle JDBC Thin Driver (XA)**” as the available resource. All of the RCOM DataSources are defined here. The “**Oracle Rib Datasource**” is the DataSource that the RIB utilizes.

## Chapter 7 – ISO Integration

This chapter will review the steps required for integrating RIB 10.3.X with SIM.

### Manual Steps

Log into the SIM application server as the user who performed the SIM install.

- 1 Create a temporary install directory for integrating RIB with SIM. This temporary directory will be known as <ISO\_TMPINSTALL\_DIR> for the remainder of this section:

```
> mkdir INSTALL
```

- 2 Copy the file RIBForISO<version>.tar to <ISO\_TMPINSTALL\_DIR>.

- 3 Once you have copied the file, extract the contents.

```
> tar xvf 'RIBForISO<version>.tar'
```

- 4 Ensure all files located in <ISO\_TMPINSTALL\_DIR> have the correct permissions by issuing the following command in <ISO\_TMPINSTALL\_DIR>:

```
> chmod -R 755 *
```

- 5 Edit the file iso\_profile. This file is located at <ISO\_TMPINSTALL\_DIR>/RIB<version>. Make sure the settings for the following variables are correct for your environment.

- ISO\_TMPINSTALL\_DIR - Same directory created in step 1 above
- ISO\_INSTALL\_DIR - The directory where ISO was installed.

**Note:** The install script expects ISO\_TMPINSTALL\_DIR to include the temporary install directory created in step 1 above (eg: /files0/jadmin/INSTALL) and ISO\_INSTALL\_DIR to include the full directory path leading up to server<platform> (eg: /files0/jadmin/iso)

- 6 In <ISO\_TMPINSTALL\_DIR>/RIB<version> run iso\_profile:

```
> . /iso_profile
```

- 7 In <ISO\_TMPINSTALL\_DIR>/RIB<version>, run installisoconfig.

```
> ./installisoconfig
```

## 8 Steps for the RIB Error Hospital:

- Check the SIM schema for the existence of the following RIB error hospital tables: RIB\_MESSAGE, RIB\_MESSAGE\_FAILURE, and RIB\_MESSAGE\_ROUTING\_INFO.
- If the RIB error hospital tables do not exist, check to ensure that the following tablespaces exist: LOB\_DATA, RETEK\_DATA, and INDEX\_DATA

**Note:** Do not run the following sql scripts if the rib error hospital tables above already exist. These scripts will drop and re-create the tables, thus deleting all data in the tables.

- run rib\_ddl.sql as the SIM schema to create the required RIB tables. This script is located at  
<ISO\_TMPINSTALL\_DIR>/RIBforISO<version>/XML\_Uutilities.
- run rib\_message\_seq.sql as the SIM schema to create the sequence number for the table RIB\_MESSAGE. This script is located at:  
<ISO\_TMPINSTALL\_DIR>/RIBforISO<version>/XML\_Uutilities.

**Note:** Since the database scripts drop and re-create the tables, ignore any errors resulting from trying to drop objects that don't exist. These errors are expected.

## 9 Edit the file ribmessaging.cfg. This file is located at &lt;ISO\_INSTALL\_DIR&gt;/server&lt;platform&gt;/retek/sim/files/prod/config. Make sure the setting for the following variable is correct for your environment. This value must match exactly values listed in the SeeBeyond configuration file cpJMS.cfg of HostName+Port Number (I.E cpJMS.cfg-HostName: mspdev05.retek.int, cpJMS.cfg-Port Number: 37053, ribmessaging.cfg-BROKER: mspdev05.retek.int:37053).

- BROKER=<EGATE\_SERVER\_NAME>:<EGATE\_SERVER\_PORT>  
(eg: BROKER=10.1.1.164:25053)

## 10 Edit the file messaging.cfg. This file is located at &lt;ISO\_INSTALL\_DIR&gt;/server&lt;platform&gt;/retek/sim/files/prod/config. Make sure the setting for the following variable is correct for your environment. This value must match exactly values listed in the SeeBeyond configuration file cpJMS.cfg of HostName+Port Number (I.E cpJMS.cfg-HostName: mspdev05.retek.int, cpJMS.cfg-Port Number: 37053, ribmessaging.cfg-BROKER: mspdev05.retek.int:37053).

- BROKER=<EGATE\_SERVER\_NAME>:<EGATE\_SERVER\_PORT>  
(eg: BROKER=10.1.1.164:25053)

- 11 Edit the file `riblog4j.properties`. This file is located at `<ISO_INSTALL_DIR>/server<platform>/retek/sim/files/prod/config`. Replace all occurrences of `<ISO_INSTALL_DIR>` in this file with the full directory path leading up to the `/server<platform>` directory.
  - Ex: `log4j.appender.ItemsSubscriber.File=<ISO_INSTALL_DIR>/serverUnix/retek/sim/log/itemsmessagingcomponent.log`  
(eg: `log4j.appender.ItemsSubscriber.File=/files0/jadmin/sim10.1/serverUnix/retek/sim/log/itemsmessagingcomponent.log`)
- 12 Edit the file `rib.properties`. This file is located at `<ISO_INSTALL_DIR>/server<platform>/retek/sim/files/prod/config`. Replace the occurrence of `<ISO_INSTALL_DIR>` in this file with the full directory path leading up to the `/server<platform>` directory.
  - Ex: `log.default.file_path=<ISO_INSTALL_DIR>/serverUnix/retek/sim/log/`  
(eg: `log.default.file_path =/files0/jadmin/sim10.1/serverUnix/retek/sim/log/`)
- 13 After successful completion to `installisoconfig`, the temporarily directory `ISO_TMPINSTALL_DIR` can be removed

## ISO Reference

The following sections are noted here for reference.

### rib.properties file

In the `rib-redsky.jar` file, you will find a file called, “`rib.properties`”. This file contains the RIB specific properties used by the RIB subscribing messaging components under ISO. These messaging components will be deployed in an ISO container, one for each subscribing API. Some of the important sections of this file are illustrated below:

```
#####
# These are the RIB hospital properties.
hospital.attempt.max=5
hospital.attempt.delay=10
hospital.attempt.delayIncrement=10
#####
# These are properties that are also used in the process
# of putting a message in the hospital. The difference
# here
# is that these properties control some of the concrete
# classes
```

```
# that are used in this process.  
failureImpl=com.retek.rib.sbyn.collab.FailureWrapper  
routingInfoImpl=com.retek.rib.sbyn.collab.RoutingInfoWrapper  
routingInfoDetailImpl=com.retek.rib.sbyn.collab.RoutingInfoDetailWrapper  
ribMessageImpl=com.retek.rib.sbyn.collab.RibMessageWrapper  
ribMessagesImpl=com.retek.rib.sbyn.collab.RibMessagesWrapper
```

## Example of a messaging component configuration file

We will use the ASNIIn API for an example of a configuration file. These configuration files can be found in the usual ISO Chelsea “config” directory.

# The topic name from which to accept messages.

TOPIC\_NAME=etASNOOutISO

# Makes the subscription durable (see JMS specification).

DURABLE\_SUBSCRIBER=true

# The type of component – Publisher or Subscriber.

JMS\_COMPONENT\_TYPE=Subscriber

# The messaging group to which to listen.

MESSAGING\_GROUP=

# Module name to be used for the Rib's context object.

MODULE\_NAME=RibMessagingComponent

# Sub-module name to be used for the Rib's context object.

# This will be the same as the message family name.

SUB\_MODULE\_NAME=ASNOOut

# If TRUE, only a single thread will be used to call the

# processMessages(ArrayList) method. If FALSE, multiple

# threads may call this method. Default is TRUE.

SINGLE\_THREADED=TRUE

# Disconnect from the server for the specified number of minutes between checks

# for messages. Note that this does not make sense for a non-durable topic-based

# subscription. Therefore, a value of 0 will cause the component to stay

# connected. The default is 0.

CONNECTION\_INTERVAL=0

# The config file to use for setting up messaging. The default is

# "messaging.cfg".

MESSAGING\_CONFIG=ribmessaging.cfg

# Remote Object Lookup Name

REMOTE\_NAME=ASNOOutMessagingComponent

# Collect Performance Statistics

PERFORMANCE=true

```

# Chelsea Logging Properties
LOGGING_IMPL=com.chelseasystems.cr.logging.LoggingFileServices
LOGGING_FILE_NAME=../log/asnoutmessagingcomponent.log
LOGGING_LEVEL=4
LOGGING_PAUSE=5000
LOGGING_SYSTEM_OUT=true
LOGGING_SYSTEM_ERR=true
# Log4j Logging – for any logging done by the RIB, Log4j logging will be
# used. Any logging done by the Chelsea container will use the Chelsea
# logging services. Both types of logging will use the same file.
LOGGING_LOG4J_LEVEL=DEBUG
LOGGING_LOG4J_MAX_FILE_SIZE=1024KB
LOGGING_LOG4J_MAX_BACKUP_INDEX=1
LOGGING_LOG4J_PATTERN_FORMAT=%d [%t] %-5p %c - %m%n

```

## ribmessaging.cfg file

This configuration file is used to configure the JMS messaging parameters across all of the RIB's APIs, publishing and subscribing.

# The client impl is the class that implements the MessagingServices contract.

```
CLIENT_IMPL=com.retek.rib.redsky.RibSeeBeyondJmsServices
```

# The time-to-live for messages sent to the server.

```
MESSAGE_LIFETIME=1800000
```

# This property has to do with who controls the transaction. When set to “true”, the Chelsea framework is in control of a global transaction. When “false”, commits or rollbacks must be done explicitly in the application code. For the #RIB, this should always be “true”.

```
USE_SESSION_TRANSACTION=true
```

# POS device-specific class uses Store and Register in global repository.

# Currently, the RIB does not use grouping, so this entry is irrelevant.

```
GROUPING_UTIL=com.chelseasystems.cr.messaging.grouping.POSMessagingGroupService
```

# Default group name. Again, the RIB does not currently use grouping, so this entry is irrelevant.

```
DEFAULT_GROUP=RTK
```

# The Broker is the JMS server address and port.

# For the RIB, this will be the server name and port of the SeeBeyond JMS



# queue.

BROKER=10.1.1.164:25053

# Username and password are set via administration of the JMS server. For the  
# RIB, SeeBeyond does not make use of username and password, so these should  
# be blank.

USERNAME=

PASSWORD=

# Number of times to try getting a connection to JMS server

MAX\_CONNECTION\_TRIES=2

# Number of seconds to pause between connection attempts

PAUSE\_BETWEEN\_TRIES=2