SeeBeyond ICAN Suite

eGate Integrator for eInsight Enterprise Service Bus User's Guide

Release 5.0.4

Version 20040602173257.

Contents

Chapter 7

Environments 110

Chapter 8

Chapter 9

List of Figures

List of Tables

Chapter 1

Introduction

This chapter describes the general purpose, scope, and organization of this document, and also provides references to additional sources of relevant information.

1.1 Purpose and Scope

This User's Guide provides general information about the features and operation of SeeBeyond® eGate Integrator in creating and deploying eGate Projects. For information on eGate Integrator system management, see the *eGate Integrator System Administration Guide*.

Note: *Any operational explanations provided in this document are generic, for reference purposes only, and do not necessarily address the specifics of setting up individual eGate Projects.*

1.2 Intended Audience

This User's Guide is intended for personnel who are involved in integrating software applications using eGate Integrator. To a large extent, these are individuals who will be using the eGate Enterprise Designer to build eGate Projects to accomplish this task. This guide also provides a basic overview of the eGate product for those attempting to gain a general understanding of how eGate Integrator works.

This guide assumes that the reader is an experienced computer user, familiar with Windows-style GUI operations, and also has an in-depth understanding of the operating system(s) on which eGate Integrator will be installed.

Note: *The eGate Integrator graphical user interface (GUI) runs only on Windows. Refer to the eInsight Enterprise Service Bus Installation Guide for a list of operating systems on which eGate Integrator itself can run.*

1.3 Organization of Information

This document provides information about eGate Integrator 5.0 and includes the following chapters and appendices:

- **Chapter 1 "Introduction"** describes the purpose of this User's Guide, including writing conventions and a list of related documents.
- **Chapter 2 "System Overview"** provides an overview of the general structure, architecture, and operation of eGate Integrator, and it's place within the SeeBeyond ICAN Suite.
- **Chapter 3 "Enterprise Manager"** provides a detailed overview of the Enterprise Manager, including its structure and operation.
- **Chapter 4 "Enterprise Designer"** provides a detailed overview of the Enterprise Designer, including its structure and operation.
- **Chapter 5 "eGate Project Components"** explains how to create a Connectivity Map and use the Configuration Editor to modify eWay and JMS connections between Connectivity Map components.
- **Chapter 6 "Object Type Definitions"** describes how to create Object Type Definitions (OTDs).
- **Chapter 7"Environments"**explains how to create and populate eGate Environments, and how to configure and start Logical Hosts.
- **Chapter 8 "Project Deployment"** explains how to create and activate Deployment Profiles.
- **Chapter 9 "Web Services"** describes how to use eGate Integrator in concert with other ICAN Suite components to create Web services.

In addition, the **Glossary** on page 189 lists various terms used in this User's Guide.

1.4 Writing Conventions

The following writing conventions are observed throughout this document.

Table 1 Writing Conventions

Text	Convention	Example
Button, file, icon, parameter, variable, method, menu, and object names.	**Bold** text	▪ Click **OK** to save and close. ▪ From the **File** menu, select **Exit**. ▪ Select the **logicalhost.exe** file. ▪ Enter the **timeout** value. ▪ Use the **getClassName()** method. ▪ Configure the **Inbound** File eWay.
Command line arguments and code samples	`Fixed` font. Variables are shown in ***bold italic***.	`bootstrap -p` ***`password`***
Hypertext links	**Blue** text	**http://www.seebeyond.com**

Additional Conventions

Windows Systems

For the purposes of this guide, all references to **Windows** apply to Microsoft Windows Server 2003, Windows XP, and Windows 2000.

Path Name Separator

This guide uses a backslash (\) as the separator within path names. If you are working on a UNIX system, please substitute a forward slash (/).

1.5 Supporting Documents

The following SeeBeyond documents provide additional information about the eGate Integrator system as explained in this guide:

- *eGate Integrator JMS Reference Guide*
- *eGate Integrator System Administration Guide*
- *eGate Integrator Tutorial*
- *eInsight Enterprise Service Bus Installation Guide*
- *SeeBeyond ICAN Suite Deployment Guide*
- *SeeBeyond ICAN Suite Primer*

For information on a specific add-on product (for example, an eWay Intelligent Adapter), see the User's Guide for that product. A complete list of SeeBeyond documentation is included in the *SeeBeyond ICAN Suite Primer*.

The documentation for the SeeBeyond ICAN Suite is distributed as a collection of online documents, which can be accessed through the Enterprise Manager (see **Documentation** on page 31). These documents are in Adobe Acrobat format, which requires that Acrobat Reader be installed on your computer. Acrobat Reader can be from Adobe Systems as a free download from the following URL:

http://www.adobe.com

1.6 The 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

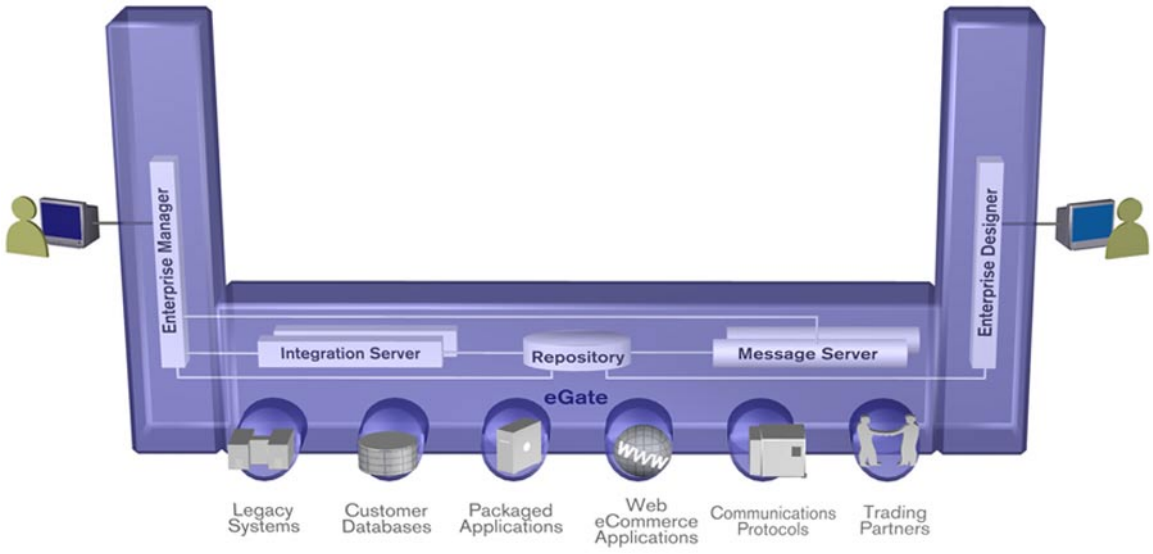
System Overview

This chapter provides an overview of the conceptual operation and general architecture of the eGate Integrator part of eInsight Enterprise Service Bus.

2.1 Introduction

SeeBeyond® eGate™ Integrator is a fully J2EE certified and Web services-based, distributed integration platform that serves as the foundation of the SeeBeyond Integrated Composite Application Network™ (ICAN™) Suite. eGate Integrator provides the core integration platform, comprehensive systems connectivity, guaranteed messaging and robust transformation capabilities while providing a unified, single sign-on environment for integration development, deployment, monitoring and management. eGate Integrator supports portability of integrations across common J2EE application servers through a completely open, J2EE-certified and Web services-based architecture.

Figure 1 eGate Integrator



As shown in Figure 1, the heart of eGate Integrator is the Repository, which is a comprehensive store of information common to the entire enterprise. An integrated

UDDI-compliant server allows publication and discovery of Web services. The run-time environment employs J2EE-compliant integration servers as operational engines and JMS-compliant message servers for the propagation of messages. The flexibility of the eGate system allows the option of deployment to a SeeBeyond run-time environment or to third-party application servers, across a distributed network of hardware platforms.

Enterprise Manager provides a unified, browser-based framework for managing all aspects of the run-time environment, as well as installing and updating all ICAN Suite components. Enterprise Designer provides a unified, graphical development environment for integrating systems and developing composite applications using Web services.

eGate Integrator can communicate with and link multiple applications and databases across a variety of different operating systems. eGate performs with a wide variety of hardware, message standards, operating systems, databases, and communication protocols in both real-time and batch (scheduled) integration modes.

2.2 Integration Model

SeeBeyond addresses application integration by means of an eGate Project, which contains the business logic required to solve the specific problem. The Project contains the various logical components and supporting information required to perform the routing, processing, and caching of messages containing the relevant data from one application to another. All Project information is stored in the Repository.

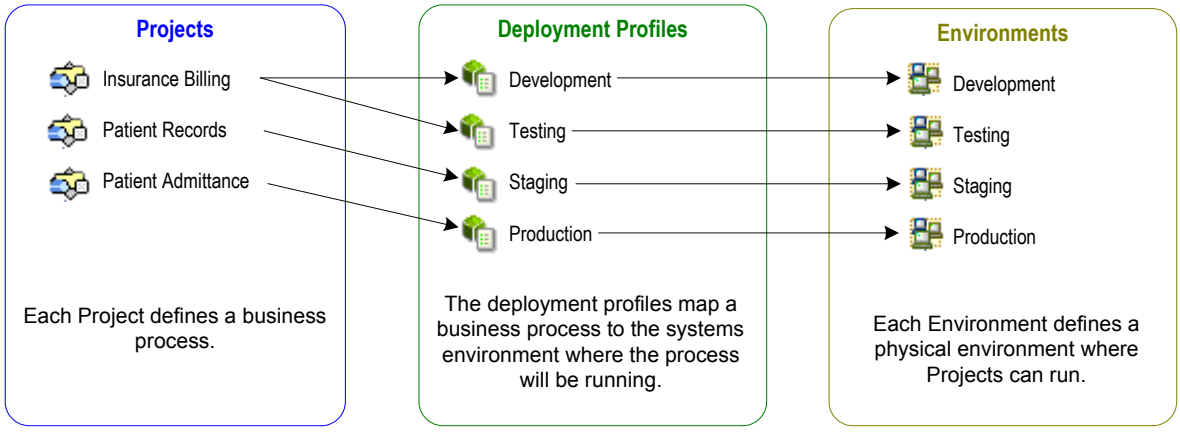
Projects are created using tools contained within Enterprise Designer and, once deployed, can be run and monitored using Enterprise Manager. Projects can also be set up to be run from the business process level using the SeeBeyond eInsight Business Process Manager, if that product is also installed.

Projects are run within individual sets of system definitions, referred to as Logical Hosts. These are defined within Environments, which represent the physical resources required to implement the Project. Projects are mapped to the individual Environments by means of deployment profiles, which are defined within the Enterprise Designer and become part of the Project. Activating the deployment profile deploys the Project to the associated Environment.

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

The finished Project, of course, will run in your production Environment; separate Environments, having the same structure as the production Environment, should be created for development and testing. You may also want some additional Environments, such as staging. The following figure illustrates the eGate Integrator implementation model using a healthcare-related example.

Figure 2 eGate Integrator Implementation Model



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

In broad outline, implementing an integration Project using eGate Integrator includes the following steps:

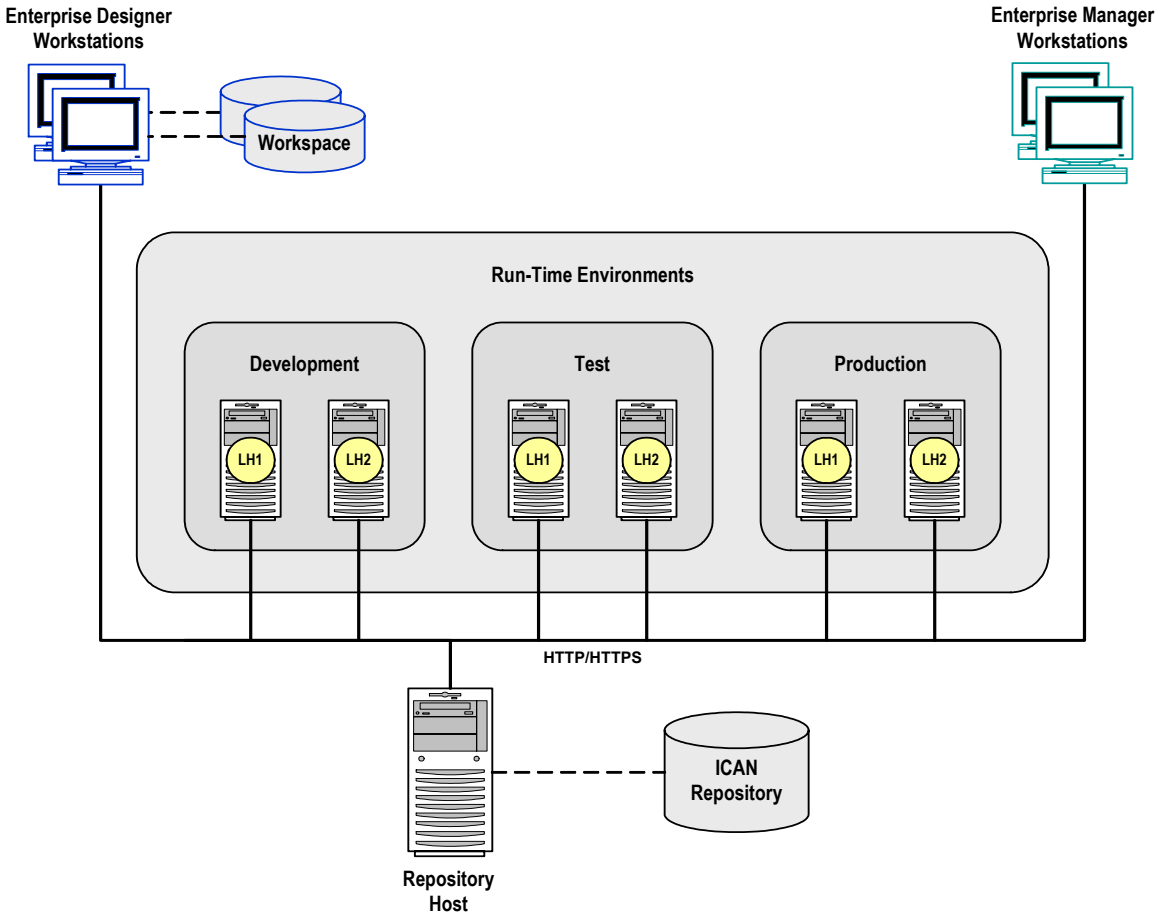
1. **Design your Project.**
2. **Define your Environments.**
3. **Create your Deployment Profiles.**
4. **Deploy the eGate Project.**

These implementation steps are all accomplished using Enterprise Designer, which is introduced in **Enterprise Designer** on page 24 and developed further in subsequent chapters.

2.3 System Architecture

SeeBeyond's eGate Integrator employs a versatile architecture that is ideally suited to distributed computing environments. As a result, the various components of an eGate Integrator system can reside on the same hardware platform (assuming adequate system resources), or be distributed across several different hardware platforms in the enterprise network. Figure 3 shows an example system implementation that is highly distributed.

Figure 3 Distributed eGate Integrator System



Note: *In this scenario it is assumed that all instances of eGate are of the same release.*

2.3.1 Repository

The setup, components, and configuration information for the elements of a Project are stored in the Repository. The Repository also stores all of the product binary files that are required at run time by the Logical Hosts. The components and configurations are downloaded to the Logical Host during the initial bootstrap process and as needed after design-time configuration changes are made.

As shown in Figure 3, a single Repository serves the entire enterprise. This common Repository is used for development, testing, and production purposes. Communication between the Repository and other ICAN components can be configured to use either HTTP or HTTPS. The Enterprise Designer and Enterprise Manager clients can communicate with the Repository through a firewall. The Repository makes Web Services available via a UDDI registry.

2.3.2 Run-Time Environments

An eGate Environment represents the total system required to implement a Project. It consists of a collection of Logical Hosts, capable of hosting components of the ICAN Suite, along with information about external systems involved in the implementation.

- **Logical Hosts**
  Each Environment contains one or more system definitions. Each definition must include one or more **integration servers** such as the SeeBeyond Integration Server, which are the engines that run eGate services and eWays, and one or more **message servers** such as the SeeBeyond JMS IQ Manager, which manage JMS topics (publish-and-subscribe messaging) and queues (point-to-point messaging). Each collection of integration servers, message servers, and additional software modules comprise what is known as a *Logical Host*.
- **External Systems**
  An external system is a representation of a real, physical system that exists within the specific Environment, with configuration properties for locating and accessing that system.

In the example system shown in Figure 3, the production environment is split across two hardware platforms, each supporting a single Logical Host. Separate environments for development and testing should duplicate the structure of the production environment. The test environment should be supported by hardware similar to that supporting the production environment, to allow performance and load testing to give representative throughput results. The hardware supporting the development environment, however, does not usually have the same performance requirements as that supporting the test and production environments.

An eGate Project is created within the development environment, then migrated to the test environment, and finally to the production environment. This migration path is a necessary and highly critical practice in implementing a working system.

Note again that there is no requirement for the components shown in Figure 3 to run on separate systems; all could run on a single system, provided that resources (CPU, memory and disk) are sufficient to support concurrent usage.

2.4 User Interfaces

eGate Integrator uses two basic graphical user interfaces (GUIs), each of which addresses a different set of users. Enterprise Manager is an interface used by the entire ICAN Suite, the primary users of which are system administrators. Enterprise Designer is used by personnel who are involved in defining a software system for integrating the various enterprise applications using eGate Integrator and eInsight Business Process Manager.

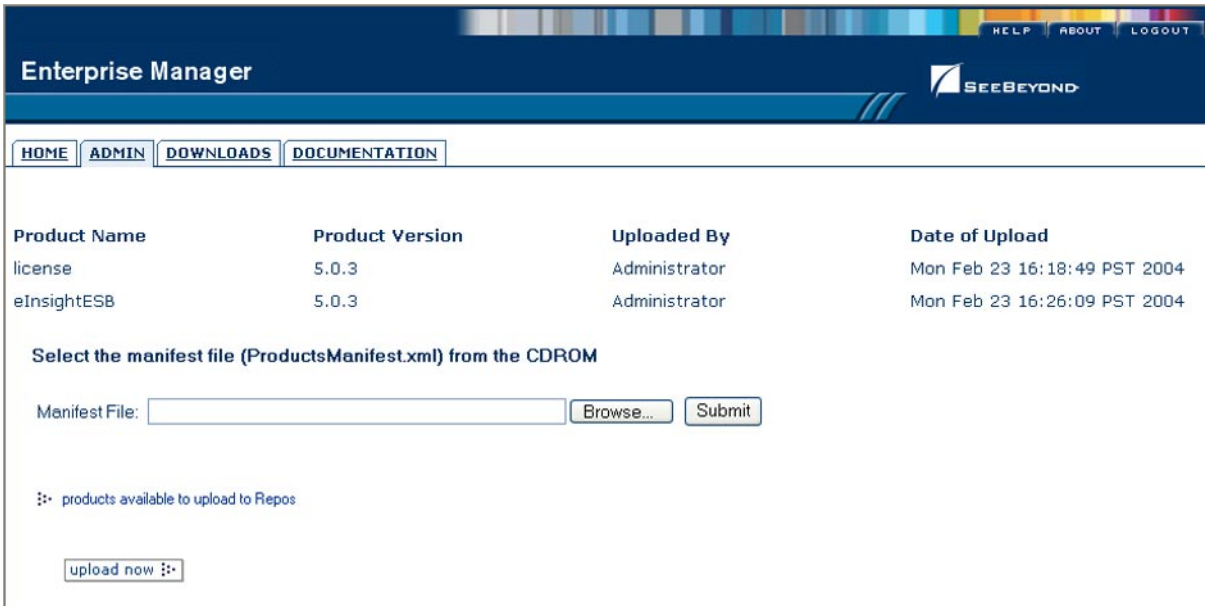
2.4.1 Enterprise Manager

Enterprise Manager is a Web-based application that works within Microsoft Internet Explorer. It is used throughout the SeeBeyond ICAN Suite for:

- Installing and updating ICAN Suite products
- Accessing ICAN Suite product documentation
- Managing and monitoring runtime components

The Enterprise Manager is described in **Enterprise Manager** *on page 27*. Figure 4 shows the Enterprise Manager **Admin** page, used in product installation.

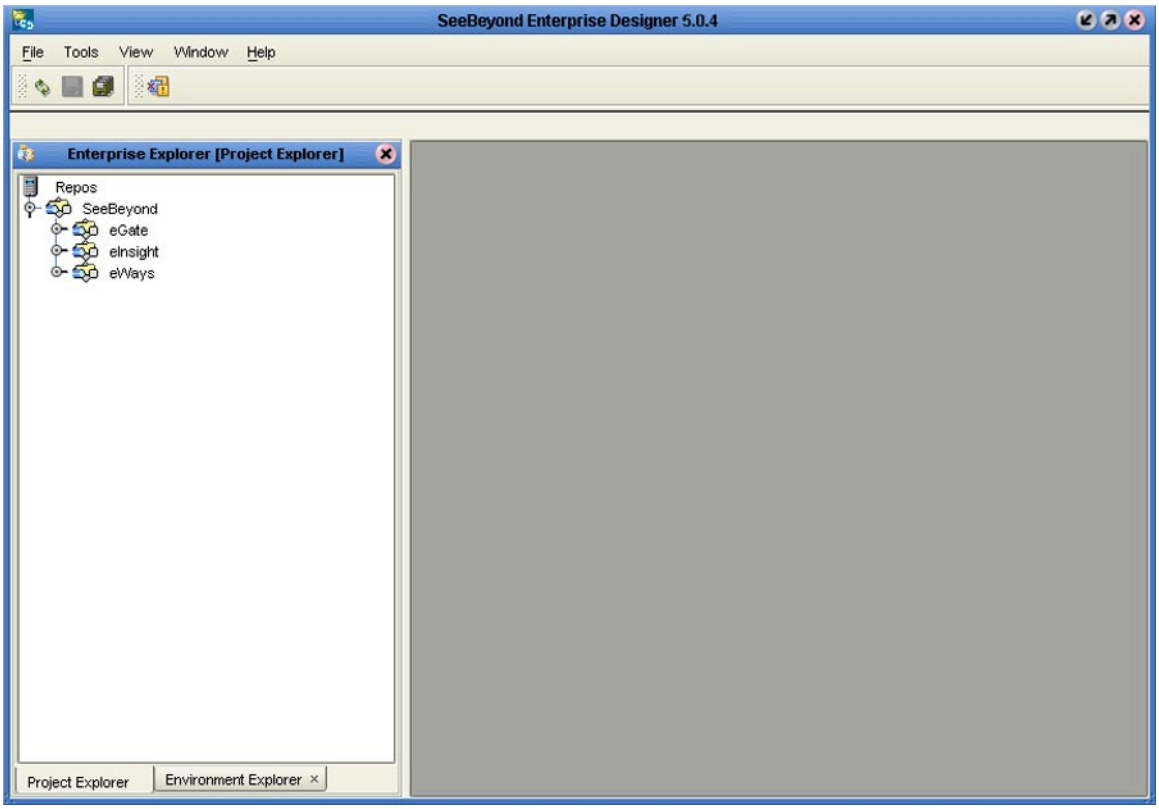
Figure 4 SeeBeyond Enterprise Manager



2.4.2 Enterprise Designer

The SeeBeyond Enterprise Designer is used to create and configure the logical components and physical resources of an eGate Project. Through this GUI (see Figure 5), you can develop Projects to process and route data through an eGate Integrator system. Enterprise Designer is also used by other components of the ICAN Suite.

Figure 5 Enterprise Designer

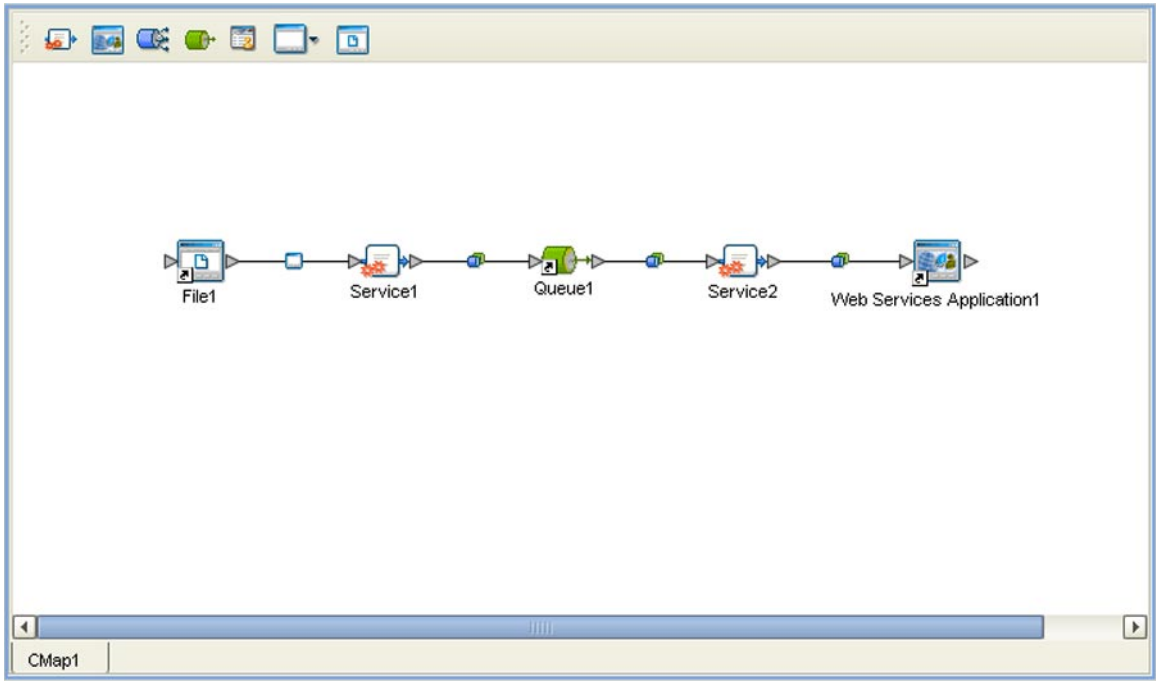


The major features of the Enterprise Designer are the Enterprise Explorer on the left, and an editor panel on the right—which is initially blank. The Enterprise Explorer follows the familiar Windows Explorer format, displaying a tree structure. The editor panel displays a variety of editors, depending upon what component is selected in the Enterprise Explorer. These editors include, for example:

- Connectivity Map Editor
- OTD Editor
- Environment Editor
- Deployment Editor

The Connectivity Map Editor (see Figure 6) provides a graphic example of one of these editors, in which logical components of a Project can be created and connected. eGate uses Connectivity Maps to intuitively configure the end-to-end flow of messages within an integration. The integration developer can to drag and drop the various components onto the Connectivity Map canvas and link them together to specify message flow. The features and usage of the Connectivity Map Editor are described in **eGate Project Components** on page 67.

Figure 6 Connectivity Map Editor



The Enterprise Designer also includes the design-time functionality for other ICAN products, such as eInsight and eXchange. For more information on using other ICAN products in the Enterprise Designer, see the product documentation for those products.

For more information on the Enterprise Designer, see **Enterprise Designer** *on page 33*.

Chapter 3

Enterprise Manager

This chapter provides an introduction to the ICAN Suite Enterprise Manager.

3.1 Overview

Enterprise Manager is a Web-based interface with which you can install and update eGate Integrator, and monitor and manage deployed eGate components.

Important: *Enterprise Manager works only with Microsoft Internet Explorer.*

3.1.1 Installing and Updating eGate

eGate Integrator components are uploaded from the installation media (CD-ROMs) to the Repository server via the Enterprise Manager. These products are then available to be downloaded and installed from the Repository server. For information on installing and updating eGate components, see the *SeeBeyond ICAN Suite Installation Guide*.

3.1.2 Monitoring and Managing eGate

The Enterprise Manager allows you to monitor and manage deployed eGate components in real-time.

- **The ICAN Monitor** on page 32 describes the basic features of the ICAN Monitor interface. ICAN Monitor usage for specific tasks is described in the *eGate Integrator System Administration Guide*.

3.2 Starting Enterprise Manager

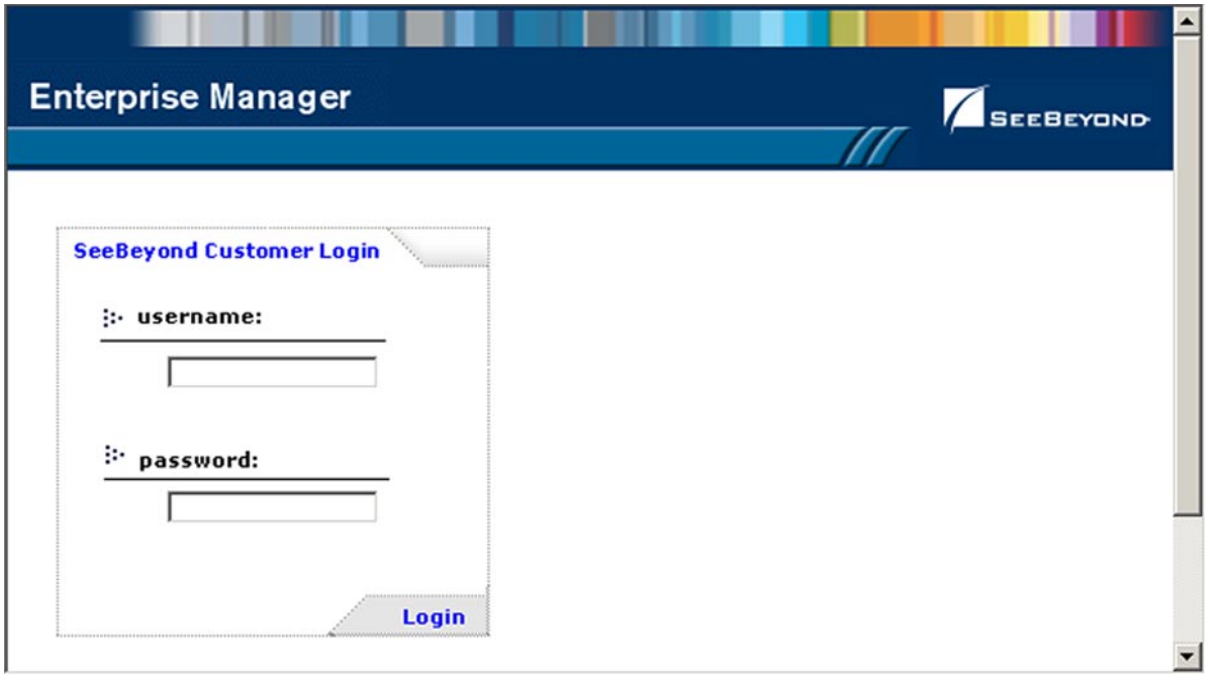
To start the Enterprise Manager

1. Launch Internet Explorer.
2. Enter **http://*hostname:portnumber*** in the **Address** box to display the SeeBeyond Customer Login window shown in Figure 7.

Note: *The **hostname** is the fully-qualified, network-addressable host name of the server where you installed the Repository. The **portnumber** is the number of the port you entered during installation of the Repository. See the SeeBeyond ICAN Suite Installation Guide.*

Important: *The TCP/IP host name must be alphanumeric.*

Figure 7 Enterprise Manager Login

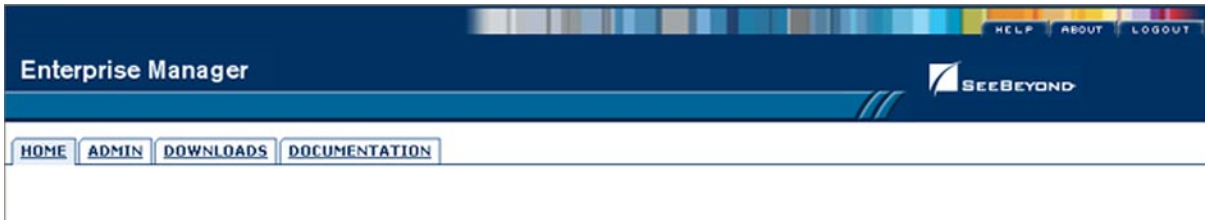


3. Enter your login ID and password in the **Username** and **Password** boxes and click **Login**.

3.3 The Enterprise Manager Interface

Once you have logged in, you see the full Enterprise Manager user interface (see Figure 8).

Figure 8 Enterprise Manager GUI



The Enterprise Manager is organized into four pages, as described in the following table. Each page is accessed by clicking the appropriate tab.

Table 2 Enterprise Manager - Pages

Page	Function
Home	The Home page is used for accessing the ICAN Monitor, which is the main subject of this chapter. See **Home** on page 30.
Admin	The Administration page is used for installing and updating ICAN components. See the *SeeBeyond ICAN Suite Installation Guide* for information.
Downloads	The Downloads page is used in installing and updating ICAN components. See the *SeeBeyond ICAN Suite Installation Guide* for information.
Documentation	The Documentation page is used for accessing ICAN Suite documentation. See **Documentation** on page 31, and the following *Note*.

Note: *You must download the documentation SAR files from the installation disk before you can access any documents using the Documentation page (see the SeeBeyond ICAN Suite Installation Guide).*

There are also three small tabs in the upper-right corner of the Enterprise Manager, which are described in the following table.

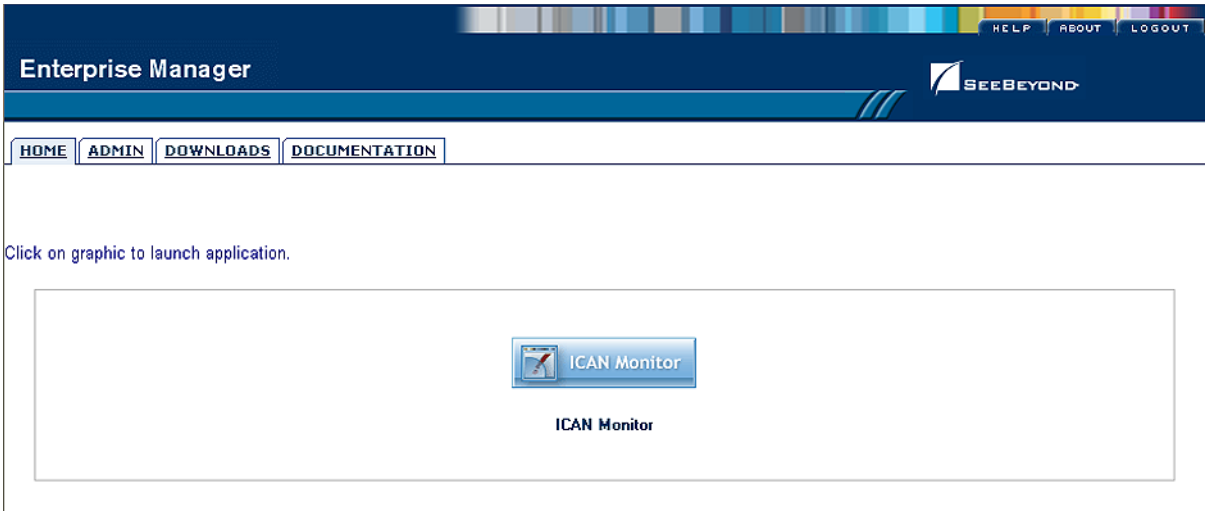
Table 3 Enterprise Manager - Control Tabs

Tab	Function
Help	The Help tab provides access to the online help system.
About	The About tab displays the installed version of the product (this tab is not present on the Documentation page).
Logout	The Logout tab logs you out of the Enterprise Manager and returns you to the Login page.

3.3.1 Home

The Enterprise Manager's **Home** page (see Figure 9) contains a link to the ICAN Monitor. Click the **Monitor** icon to launch the ICAN Monitor (see **The ICAN Monitor** on page 32).

Figure 9 ICAN Monitor Launch Icon

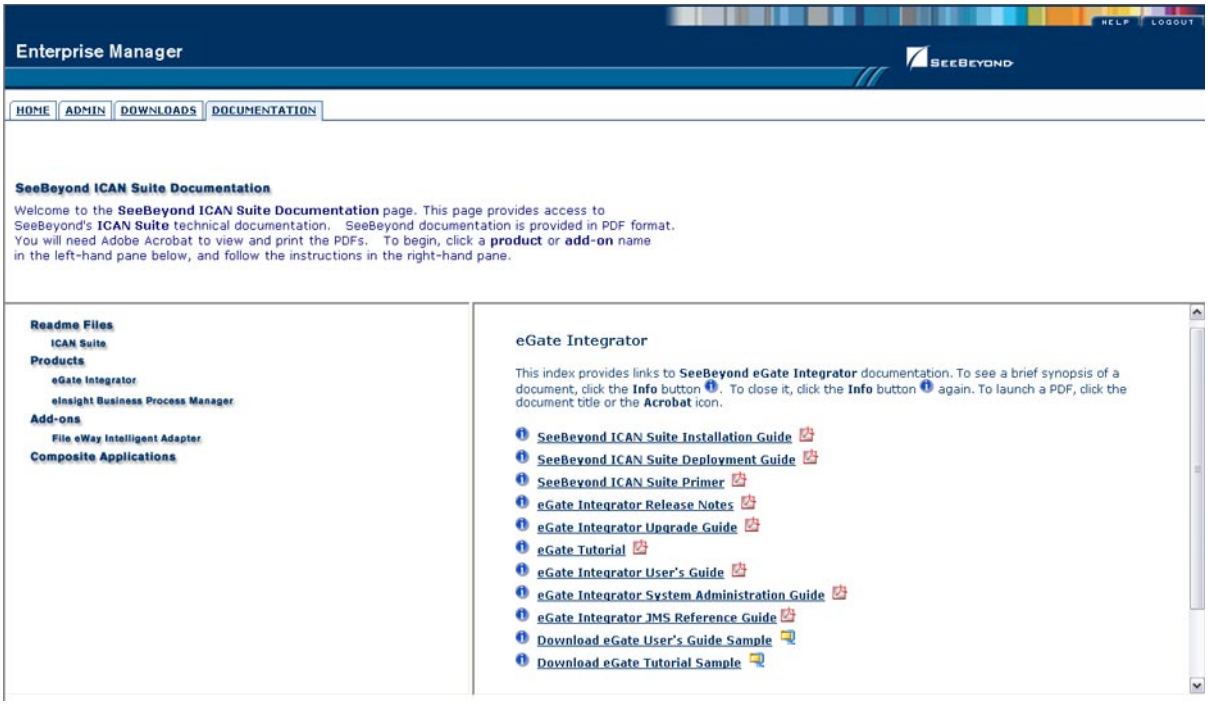


Note: *If connection problems are encountered, close all Internet Explorer windows and retry.*

3.3.2 Documentation

The **Documentation** page (see Figure 10) contains links to the latest versions of the SeeBeyond ICAN documentation in Adobe Acrobat (PDF) format, and also any sample Project files (in ZIP format). Shown is the current set for eGate Integrator.

Figure 10 Documentation Page



The provided documentation is organized into the major categories listed in Table 4.

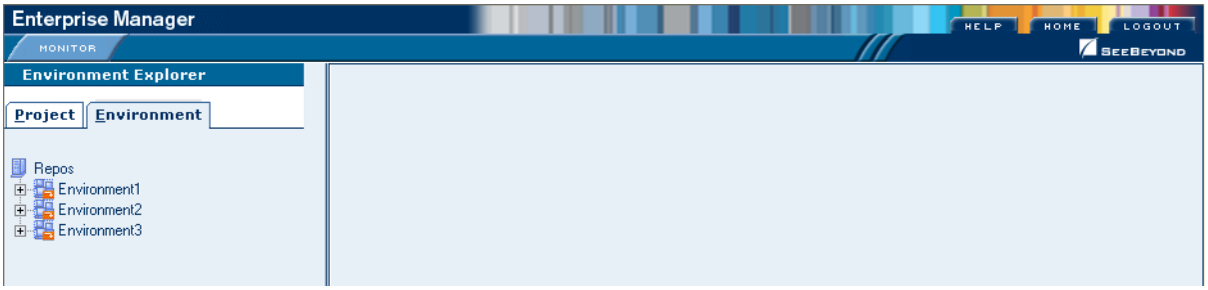
Table 4 Document Categories

Category	Contents
Readme Files	Includes information regarding the latest operating system and hardware requirements, cautions and caveats regarding known issues, and supplementary information arising after the publication of other documentation.
Products	Documentation regarding SeeBeyond core products, such as eGate Integrator and eInsight Business Process Manager. Also includes example Project files, if available.
Add-ons	Documentation regarding optional, ancillary products such as eWays and OTD Libraries.

3.4 The ICAN Monitor

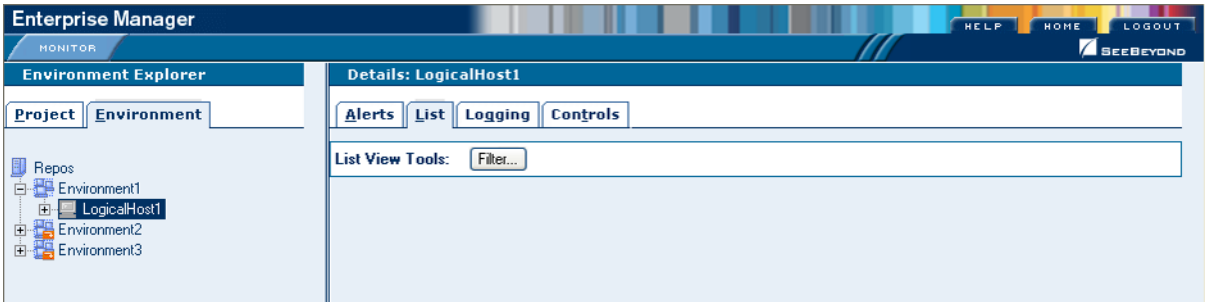
The ICAN Monitor has structure similar to that of the Enterprise Designer, with an Explorer panel on the left and a Details panel on the right. Initially, the Details panel is blank as shown in Figure 11.

Figure 11 ICAN Monitor Interface - Initial



Like the Enterprise Manager itself, the ICAN Monitor's **Details** area is organized into sections represented by tabs (see Table 5). Which tabs are present depends upon the component selected in the Explorer. For example, selecting the Logical Host displays the Monitor page shown in Figure 12.

Figure 12 ICAN Monitor Interface - Environment



At times, the Details panel will have two parts, to display an additional level of information. In this case, different tabs will be displayed in the upper and lower panels.

Table 5 ICAN Monitor Interface - Details Tabs

Tab	Function
Alerts	Displays all alerts for the component selected in the Explorer.
List	Displays a list presenting information about the component selected in the Explorer.
Logging	Displays all log messages for the component selected in the Explorer.
Controls	Displays controls that allow an Administrator to intervene in the run-time process and perform tasks such as starting and stopping components.

Note: *See the eGate Integrator System Administration Guide for detailed information regarding Monitor usage.*

Chapter 4

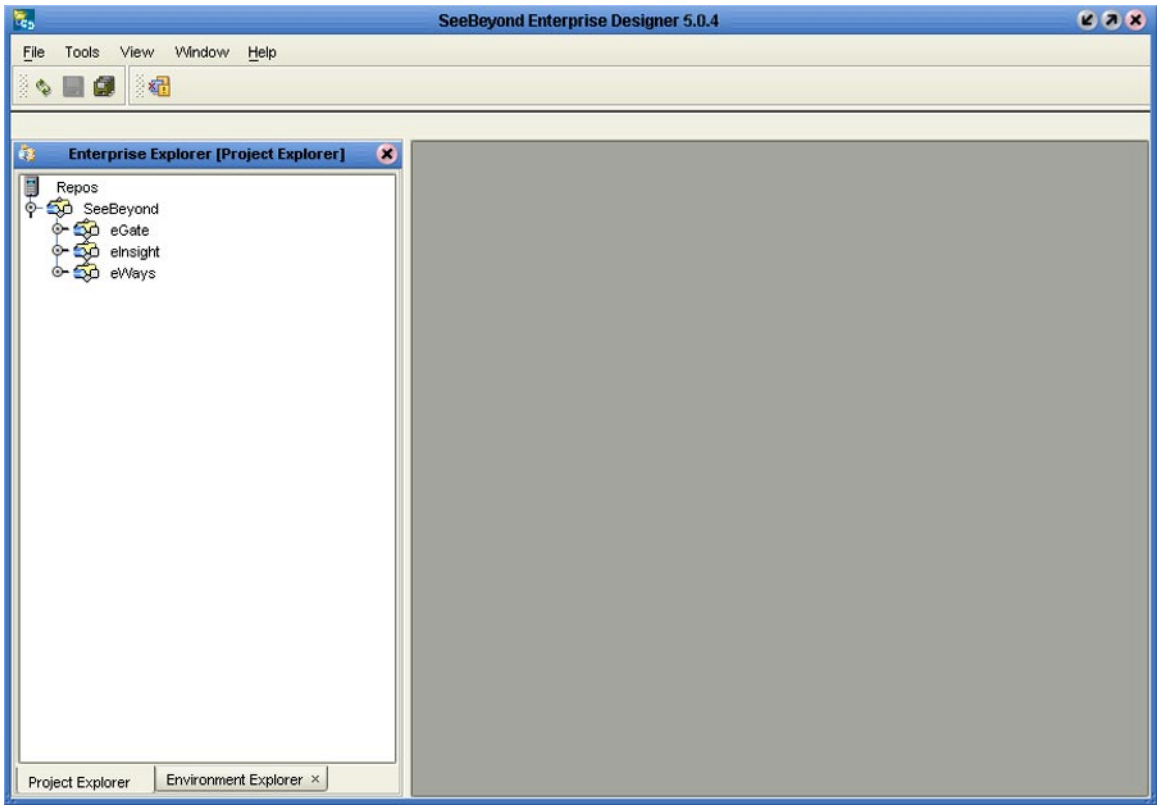
Enterprise Designer

This chapter presents an overview of the major features of the Enterprise Designer.

4.1 Overview

The Enterprise Designer graphical user interface (GUI) is used to create and configure the logical components and physical resources of an eGate Project. Through this GUI (see Figure 13), you can develop Projects to process and route data through an eGate Integrator system.

Figure 13 SeeBeyond Enterprise Designer



The procedure for invoking the Enterprise Designer is described in **Starting Enterprise Designer** on page 35.

4.1.1 User Interface

The Enterprise Designer contains the customary graphical interface features, which are described in the following sections of this chapter:

- **Menus** on page 37 describes the options contained in the individual menus.
- **Toolbar** on page 40 describes the functionality of the toolbar icons.
- **Browser Buttons** on page 40 describes the browser buttons that appear throughout the Enterprise Designer, in various wizards and dialog boxes.

The major features of the Enterprise Designer are the Enterprise Explorer on the left, and an editor panel on the right—which is initially blank. The Enterprise Explorer follows the familiar Windows Explorer format, displaying a tree structure. The Enterprise Explorer provides two views of the ICAN system, which are described in the following sections of this chapter:

- **Project Explorer** on page 41
- **Environment Explorer on page 42**

4.1.2 Editors

The editor panel displays a variety of editors, depending upon what component is selected in the Enterprise Explorer. These editors are described in the following sections of this chapter:

- **Connectivity Map Editor** on page 43
- **OTD Editor** on page 44
- **Environment Editor** on page 45
- **Deployment Editor** on page 46

4.1.3 Analysis and Archiving Tools

The Enterprise Designer includes several analysis and archiving tools, which are described in the following sections of this chapter:

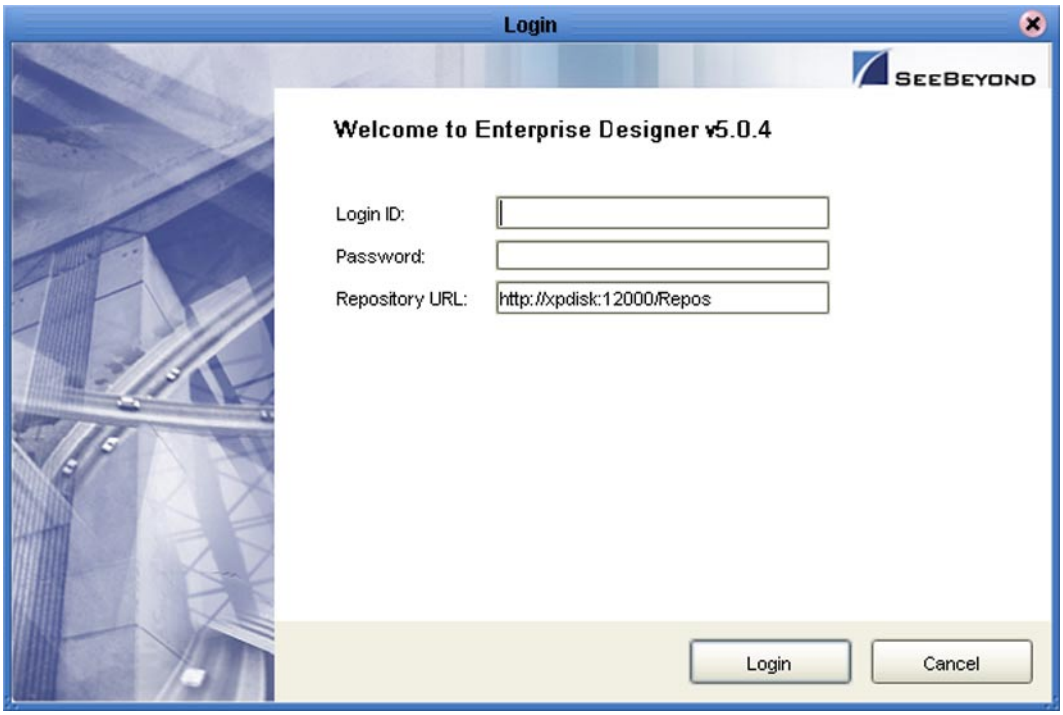
- **Project/Environment Import** on page 47, which allows you to import a Project that has been created elsewhere.
- **Project/Environment Export** on page 51, which allows you to export a Project to an external file so that it may be used elsewhere.
- **Impact Analyzer** on page 56, which helps you visualize how a change to one part of a Project would affect the rest of the Project.
- **Version Control** on page 58, which allows you to maintain multiple versions of Project components.

4.2 Starting Enterprise Designer

To start the Enterprise Designer on a Windows Platform

1 Run the batch file ***ICAN-root*\edesigner\bin\runed.bat** to display the *Login* dialog box shown in Figure 14 (placing a shortcut on your desktop streamlines this procedure).

Figure 14 Login Dialog Box



2 Click in the *Login ID* text box, and enter your login ID.

3 Tab to the *Password* text box, and enter your password.

4 The URL for the Repository should be displayed in the *Repository URL* text box. If it is incorrect, edit the URL before proceeding. See the *SeeBeyond ICAN Suite Installation Guide* for details.

5 Click **Login** to complete the login process and display the Enterprise Designer GUI shown in Figure 13. A progress monitor will appear while the process is running.

To start the Enterprise Designer on a UNIX Platform

1 Run the script ***ICAN-root*/edesigner/bin/runed.sh** to display the *Login* dialog box shown in Figure 14.

2 Click in the *Login ID* text box, and enter your login ID.

3 Tab to the *Password* text box, and enter your password.

4 The URL for the Repository should be displayed in the *Repository URL* text box. If it is incorrect, edit the URL before proceeding. See the *SeeBeyond ICAN Suite Installation Guide* for details.

5 Click **Login** to complete the login process and display the Enterprise Designer GUI shown in Figure 13.

4.3 Interface Features

4.3.1 Menus

The menu bar provides access to a variety of options for managing your Project. The individual menus are described in the following tables.

File Menu

Table 6 File Menu Options

Option	Function
Save	Saves changes to the selected objects (to the local workspace only).
Save All	Saves changes to all objects currently open in the editor (to the local workspace only).
Exit	Closes the Enterprise Designer.

Tools Menu

Table 7 Tools Menu Options

Option	Function
Impact Analyzer	Displays a dialog box in which you can view how one component of a Project impacts other components. See **Impact Analyzer** on page 56.
Options	Displays the Options Setup dialog box, in which you can specify selected options such as heap sizes and language extensions. See **Options Setup** on page 39.
Update Center	Displays a series of dialog boxes in which you can check for program updates. See the *eGate Integrator Installation Guide*.

View Menu

Table 8 View Menu Options

Option	Function
Environment Explorer	Activates the **Environment Explorer** tab on the Enterprise Explorer. See **Environment Explorer on page 42**.
Project Explorer	Activates the **Project Explorer** tab on the Enterprise Explorer. See **Project Explorer** on page 41.

Window Menu

Table 9 Window Menu Options

Option	Function
Cascade	Displays all open windows so that each window slightly overlaps the others in the Project Editor.
Tile	Displays all open windows in a stacked tile pattern.
Horizontal Layout	Displays all open windows from top to bottom.
Vertical Layout	Displays all open windows from left to right.
Minimize All	Minimizes all open windows so that only the title bar displays at the bottom of the Editor canvas.
Restore All	Returns minimized windows to their original position on the Editor canvas.
Close All	Closes all open windows.

Help Menu

Table 10 Help Menu Options

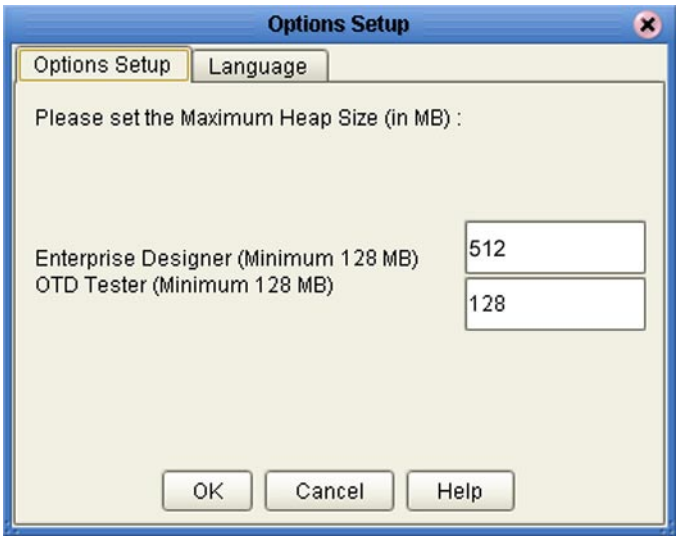
Option	Function
About Enterprise Designer	Displays an information box giving the version number, copyright information, and ICAN Repository connection information.
Contents	Displays the online help for all installed components of the ICAN Suite that operate within Enterprise Designer.
Help Sets	For future use (currently duplicates the Contents option).

Options Setup

Heap Size

The Heap Size tab allows you to increase the heap size of Enterprise Designer itself and the OTD Tester (see Figure 15). Although the default heap size settings should be adequate for most applications, occasions may arise when you will need to allocate additional memory in one or more of these modules to accommodate large file sizes.

Figure 15 Options Setup - Heap Size Dialog



Language

The Language tab allows you to control the use of extended language options in International versions of eGate Integrator.

Figure 16 Options Setup - Language Dialog

4.3.2 Toolbar

Table 11 Enterprise Designer Toolbar Icons

Icon	Function
	**Refresh All from Repository** refreshes the Project Explorer and Environment Explorer to display the current contents of the Repository. (You are prompted to save any changes before the refresh occurs.) Open editors are not refreshed.
	**Save** saves changes made to the selected Project to the local workspace only—the Repository is *not* updated. This icon is inactive if no changes have been made.
	**Save All** saves changes made to all open Projects to the local workspace only—the Repository is *not* updated. This icon is inactive if no changes have been made.
	Displays the **Impact Analyzer** dialog box, which allows you to view how one component of a Project impacts other components.

4.3.3 Browser Buttons

The following buttons are used throughout the Enterprise Designer, in wizards and file selection dialog boxes. They correspond to standard Windows browser buttons.

Table 12 Browser Buttons

Button	Function
	**Up One Level** returns you to the parent folder or directory.
	**Home** returns you to the root folder or directory.
	**Create New Folder** creates a new folder under the current folder.
	**List** displays folder/file names only.
	**Details** displays details of the folders or files (name, type, date last modified, etc.).

4.4 Enterprise Explorer

The Enterprise Explorer organizes the components of a Project into tabs that display different views of an eGate system.

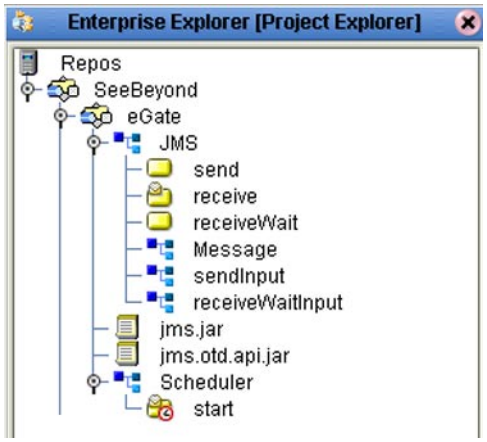
- **Project Explorer** on page 41 deals with logical components.
- **Environment Explorer on page 42** deals with physical resources, including the Logical Host and Integration Server.

Note: *The Project and Environment trees are initially loaded only to the Project or Environment level. The contents of a Project or Environment are loaded when you expand the particular node. This causes a slight delay when you expand the node, but eliminates a potentially-significant delay when you open Enterprise Designer, due to the large size of some OTD libraries.*

4.4.1 Project Explorer

The **Project Explorer** tab includes folders and icons that represent the names and contents of Projects. Some example components of a Project are shown in Figure 17.

Figure 17 Enterprise Explorer: Project Explorer View



Details of the features and usage of the Project Explorer are found in **eGate Project Components** on page 67.

4.4.2 Environment Explorer

An Environment consists of Logical Hosts capable of hosting eGate components and information about external systems which may be involved with an eGate configuration.

Figure 18 Enterprise Explorer: Environment Explorer View

Details of the features and usage of the Environment Explorer are found in **Environments** on page 110.

4.5 Enterprise Designer Editors

The editor panel—which is initially blank—displays a variety of editors, depending upon what component is selected in the Enterprise Explorer.

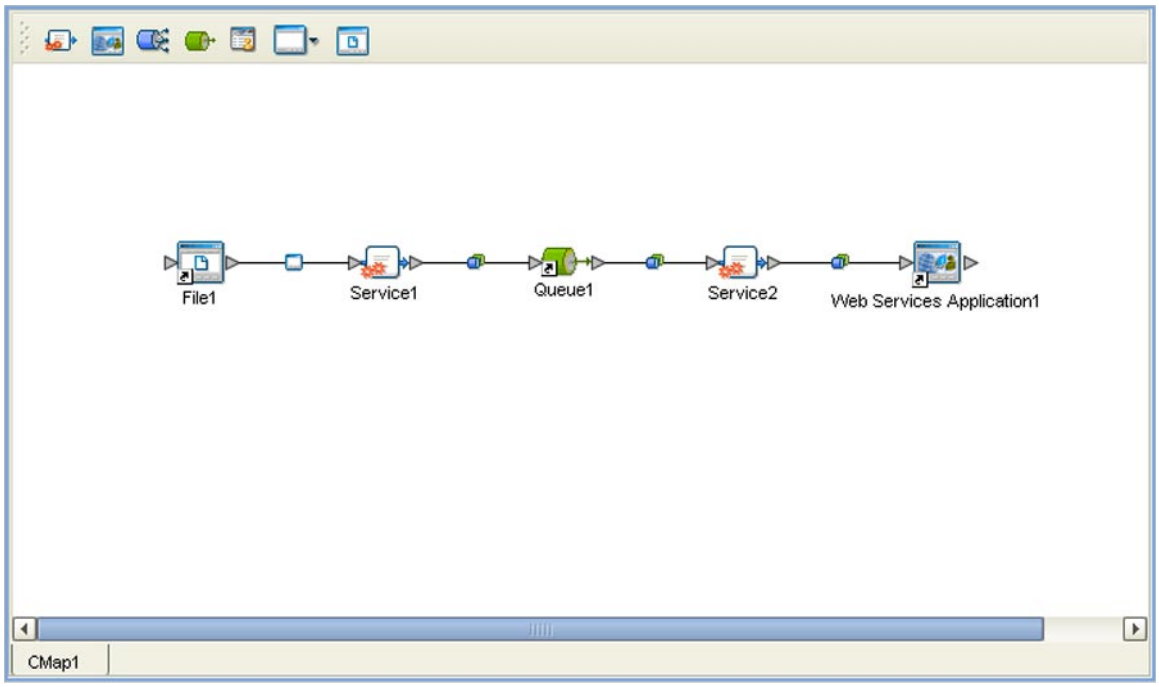
Note: *See the eInsight Enterprise Service Bus User's Guide for a detailed explanation of the steps involved in setting up an eInsight ESB Project.*

4.5.1 Connectivity Map Editor

A Connectivity Map is a graphical representation of your Project, containing the various logical components comprising the Project and the links between them. The Connectivity Map Editor, shown in Figure 19, allows you to create your Project by simply dragging and dropping icons onto a Project canvas and then connecting them to form data paths. You then can configure the components by means of dialog boxes that are displayed by clicking on the component icons.

See **Using the Connectivity Map Editor** on page 77 for detailed information.

Figure 19 Connectivity Map Editor

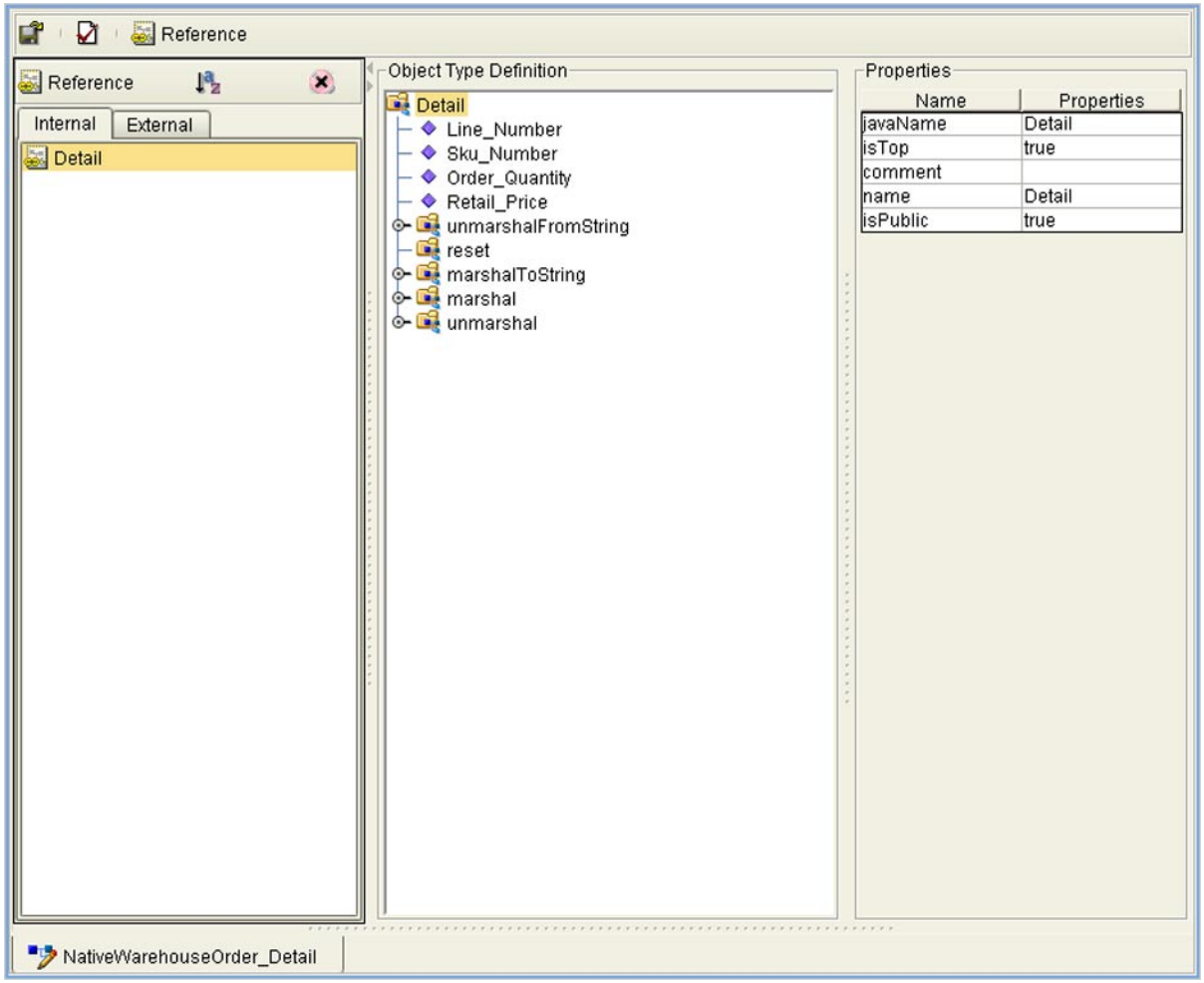


4.5.2 OTD Editor

The OTD Editor window, as shown in Figure 20, displays the source files used to create the Object Type Definitions (OTDs) to use with a Project. You use an OTD wizard tool to create OTD files and add them to the **Project Explorer** tab.

See **OTD Editor** on page 90 for detailed information.

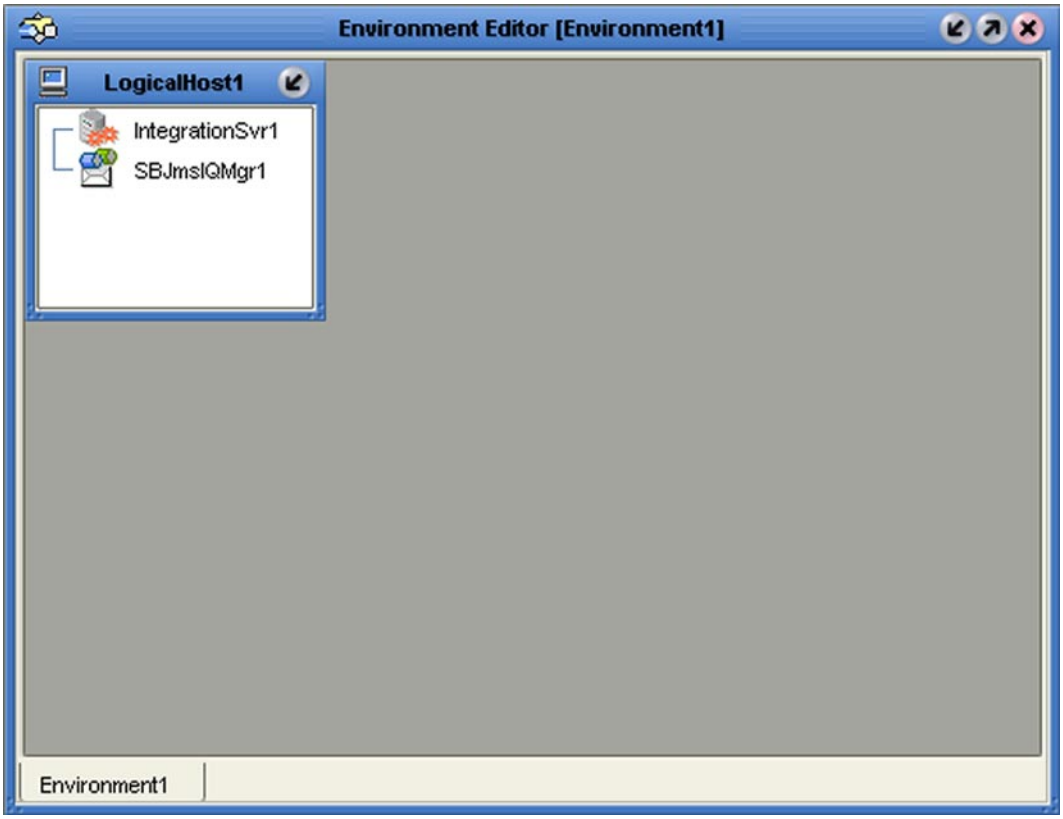
Figure 20 OTD Editor



4.5.3 Environment Editor

The Environment Editor provides a canvas in which you can create and customize an Environment. Here you can see the various components (Logical Hosts, servers, and external systems) included in the selected Environment. An environment containing example Logical Hosts is shown in Figure 21.

Figure 21 Environment Editor

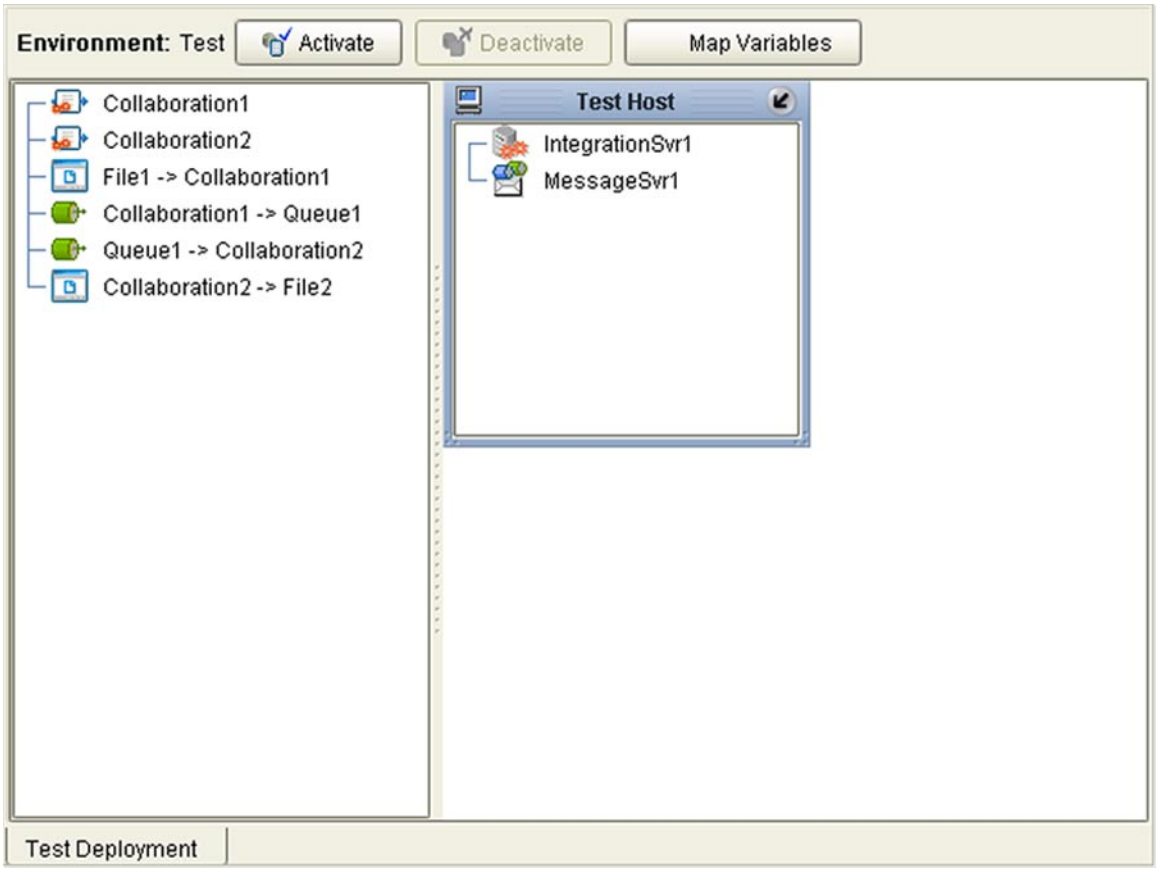


Note: *Unlike changes to Project-related configuration properties, changes to Environment-related properties do not require redeployment, just application.*

4.5.4 Deployment Editor

The Deployment Editor, as shown in Figure 22, contains information about how Project components will be deployed in an Environment. See **The Deployment Editor** on page 155 for detailed information

Figure 22 Deployment Editor



4.6 Additional Tools and Features

4.6.1 Project/Environment Import

The import function allows you to import an eGate Project or Environment file using the Enterprise Designer. Both follow essentially the same procedure.

Important: *Products installed in the source Repository must be installed in the Repository into which the Project is imported.*

When importing a Project, note that:

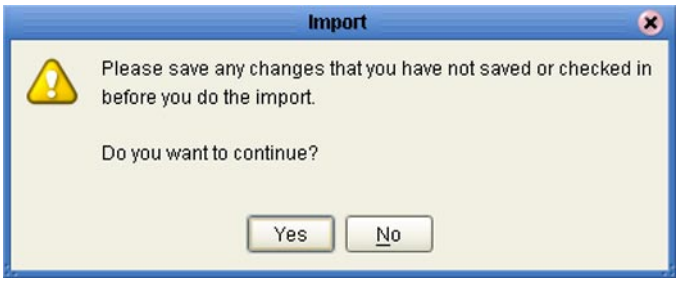
- Existing Projects are not affected by the imported Project.
- During import, if another Project having the same name exists in the target Repository, you will receive an error message and the existing file will not be overwritten.
- If you have not installed all of the necessary products (such as eWays) that a Project requires, you will not be able to import that Project and will get an error message.
- References are validated during import.
- Project deployment objects are not imported, because they have references to both Project and Environment elements that are not required at the Project level.

Importing a Project Using Enterprise Designer

To import a Project using Enterprise Designer

1. From the Repository context menu (for Projects) or the Project context menu (for Sub-Projects), select **Import.**
2. The message box shown in Figure 23 appears, prompting you to save your changes.

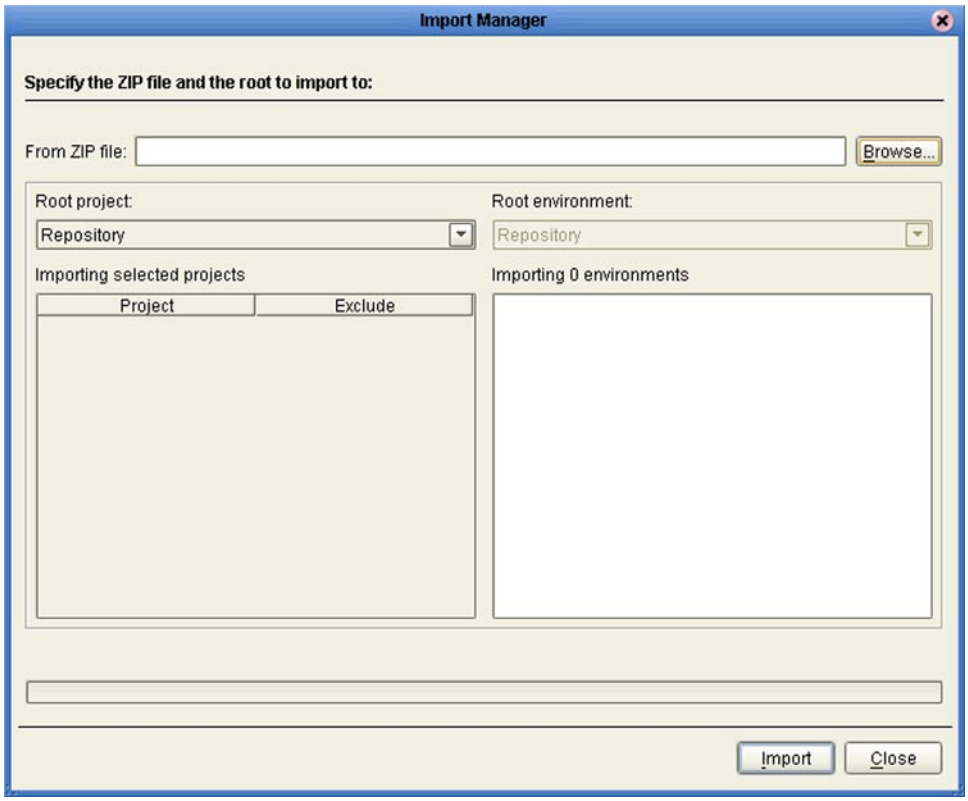
Figure 23 Import Message Box



A. If you want to save your changes, but have not already done so, click **No**. Save your changes, and then re-select **Import**, as in step 1.

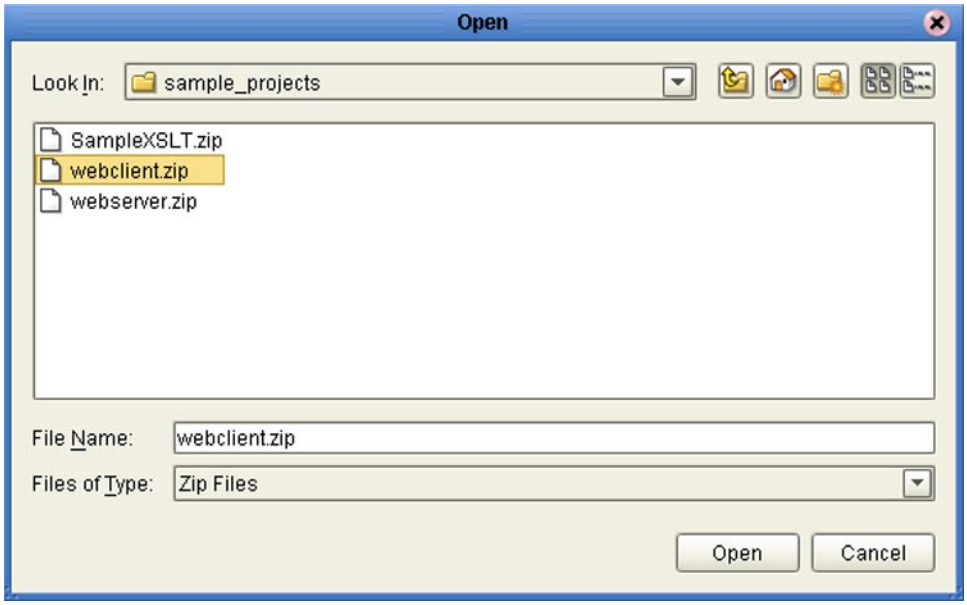
B. If you have saved any desired changes, click **Yes** to display the dialog box shown in Figure 24.

Figure 24 Import Manager Dialog Box (1)



3 Click the **Browse** button to display the *Open File* dialog box, as shown in Figure 25. If you browse to an Environment file, the *Root environment* field will be enabled.

Figure 25 Open File Dialog Box

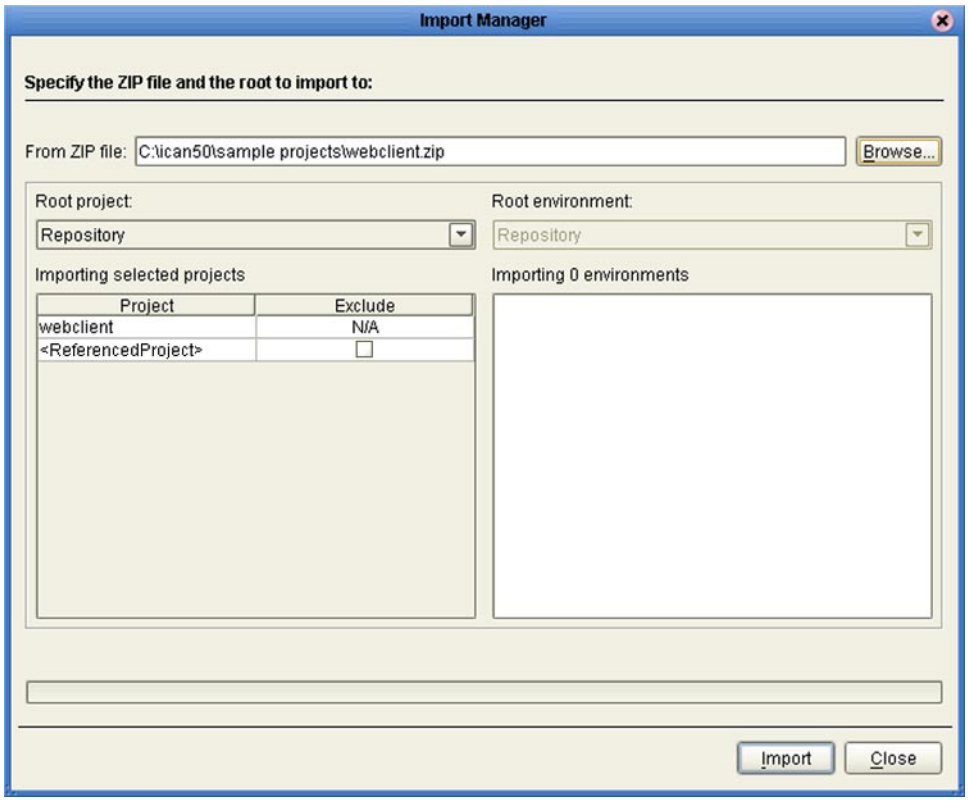


4 Locate and select the Project or Environment file that you want to import.

5 Click **Open** to import the file.

6 The Import Manager dialog box appears as shown in Figure 26.

Figure 26 Import Manager Dialog Box



Note: *If the Project you are importing contains references to another Project, and the other Project already resides in your Repository, you have the option of excluding the referenced Project from the import by checking the box that appears in the Exclude column. The references will be retargeted to the Project existing in the Repository.*

7 Click **Import** to import the file.

8 The Import Status message box shown in Figure 27 appears after the file has been imported successfully.

Figure 27 Import Status Message Box

9 Click **OK** to close the message box.

10 When you are finished importing files, click **Close** to close the Import Manager dialog box. The Project Explorer will now automatically be refreshed from the Repository.

Importing a Project Using the Command Line

You can also import a Project using the following command-line script.

Location of script file:

ICAN-root\repository\util\importProject.bat (or importProject.sh)

Command Syntax:

importProject ***username password importfile rootprojectname***

where:

- ***importfile*** is the name and path of the archive file containing the Project or Environment you are importing.
- ***rootprojectname*** is the name of the parent Project, under which the imported Project will be a sub-Project. If the Project is not to be imported as a sub-Project, then leave this parameter as an empty string (“”).

To import a Project using the import script

1 Open a command prompt and change directory to *ICAN-root*\repository\util.

2 To extract a Project contained in the file **c:\project4import.zip** and import it into the Repository, type:

```
importProject username password c:\project4import.zip ""
```

To import a Project as a sub-Project using the import script

1 Open a command prompt and change directory to *ICAN-root*\repository\util.

2 To extract a Project contained in the file **c:\project4import.zip** and import it into the Repository as a sub-Project of **mainProject**, type:

```
importProject username password c:\project4import.zip mainProject
```

To import an Environment using the import script

1 Open a command prompt and change directory to *ICAN-root*\repository\util.

2 To extract an Environment contained in the file **c:\environment4import.zip** and import it into the Repository, type:

```
importProject username password c:\environment4import.zip ""
```

3 The Environment, including all Projects deployed to that Environment, will be imported.

4.6.2 Project/Environment Export

The export function allows you to export an eGate Project and/or Environment to an external file using either the Enterprise Designer or a command-line script.

When exporting a Project, note that:

- The exported Project may have references to elements that are in other Projects. A list of such references is generated during the export process.
- Project deployment objects are not exported, because they have references to both Project and Environment elements that are not required at the Project level.

Exporting a Project Using Enterprise Designer

To export a Project or Environment using Enterprise Designer

1 From the Project context menu, select **Export** to display the Export Manager dialog box. If you do not have any existing Environments in your Repository, you will see the dialog box shown in Figure 28. If you do, you will see the dialog box shown in Figure 29.

Figure 28 Export Manager Dialog Box (1a)

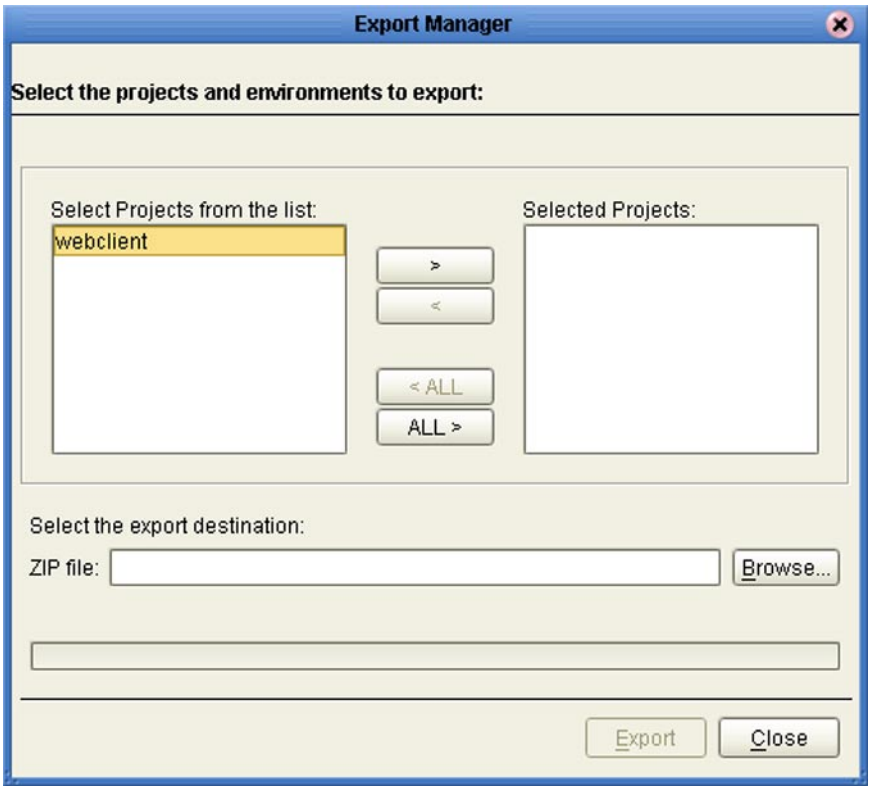
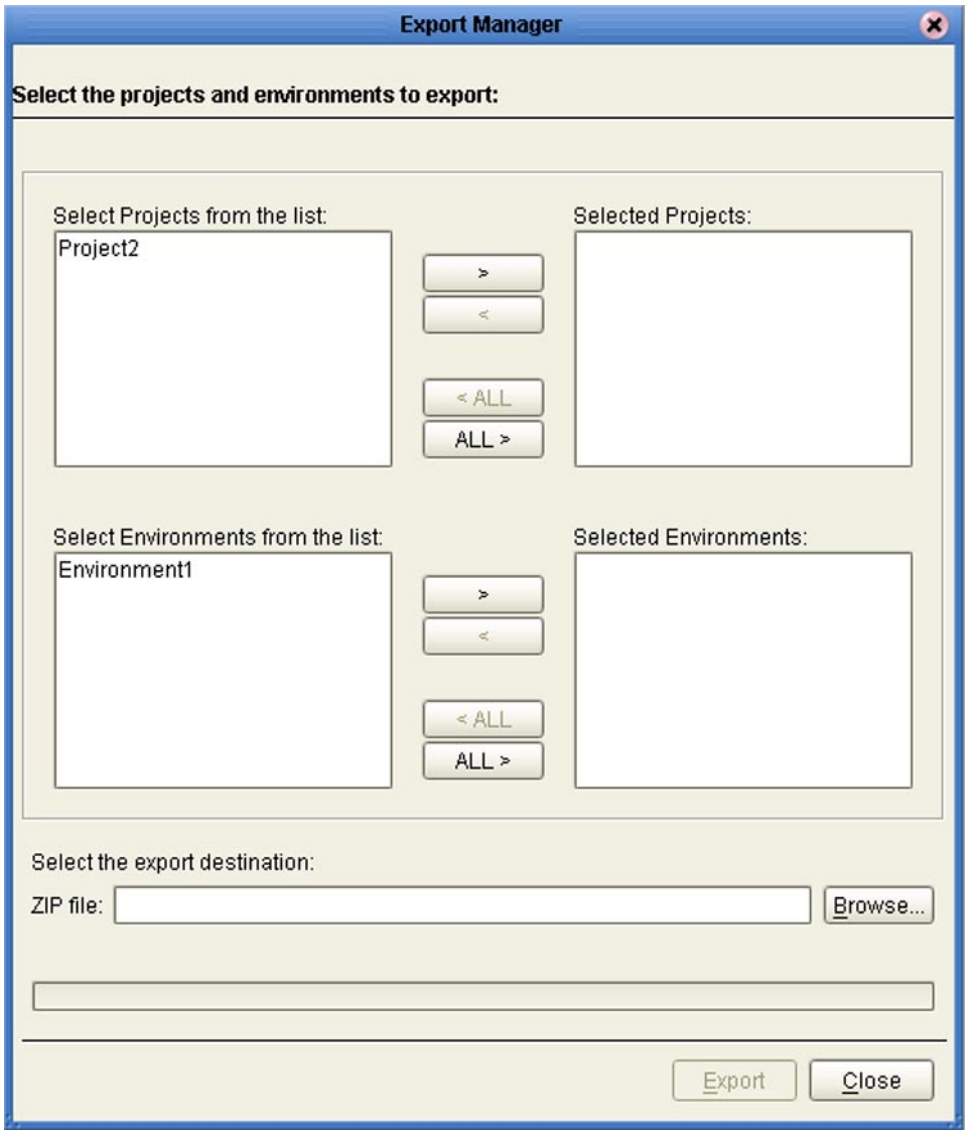
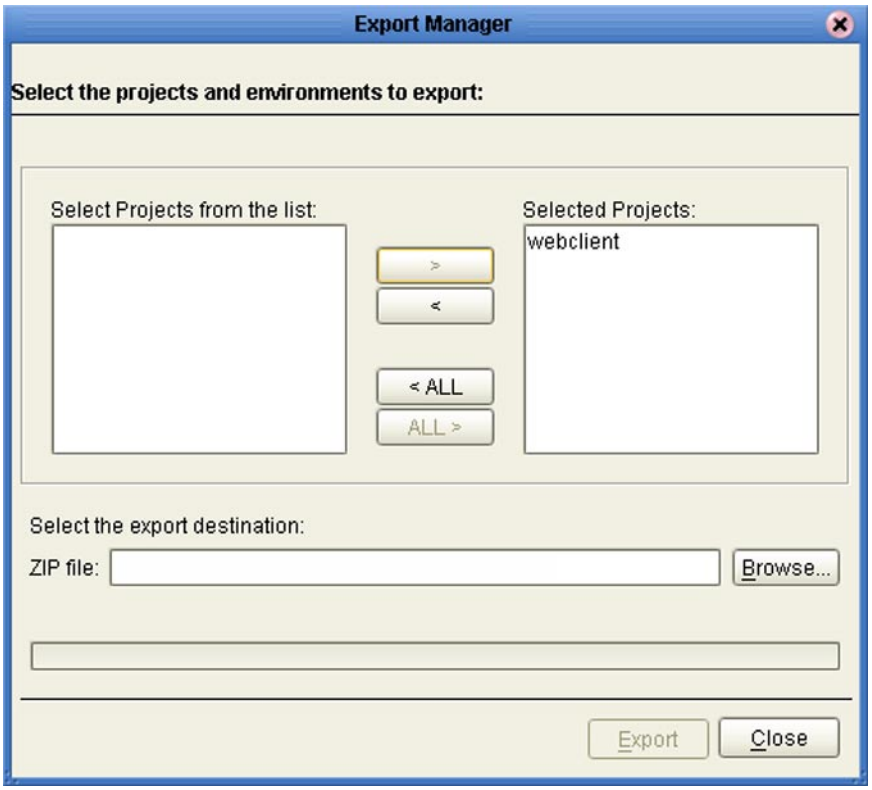


Figure 29 Export Manager Dialog Box (1b)



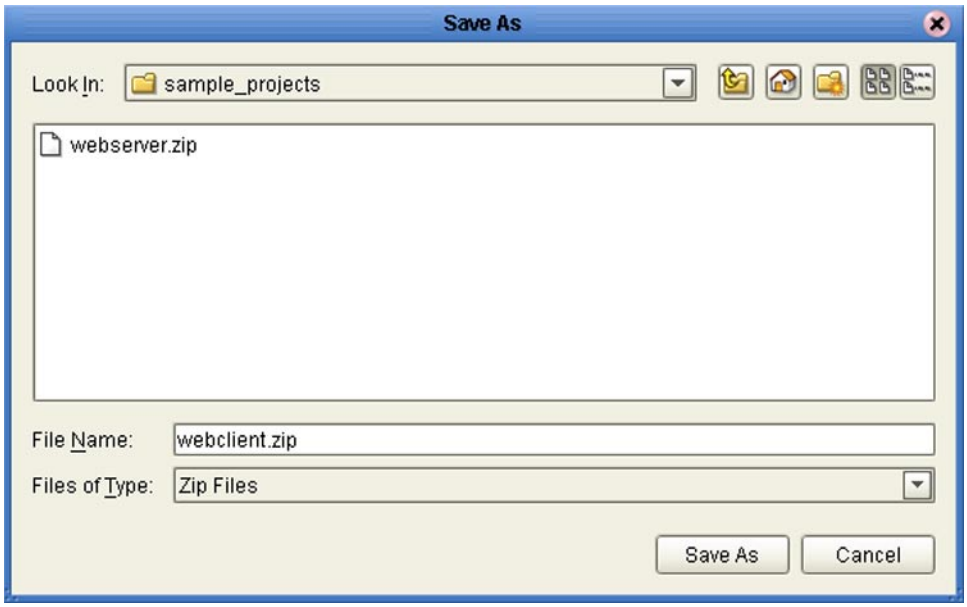
2 Highlight the desired Project(s) or Environment(s) in the displayed list, and transfer them to the *Selected Projects* or *Selected Environments* panel using the arrow buttons (see Figure 30).

Figure 30 Export Manager Dialog Box (2)



3 Click the **Browse** button to display the *Save As* dialog box, as shown in Figure 31.

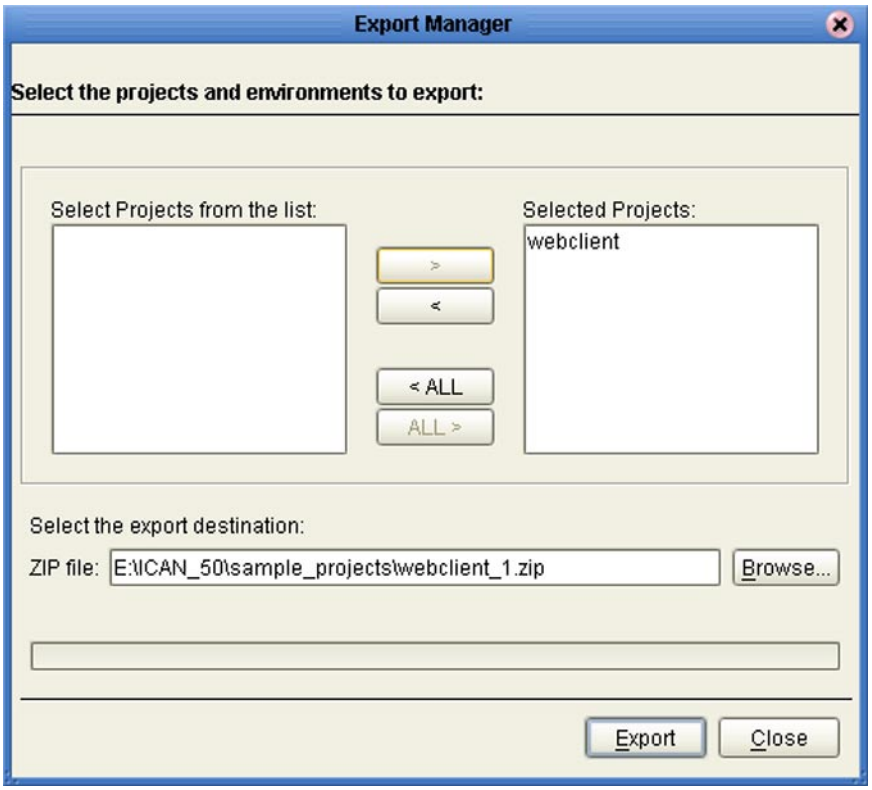
Figure 31 Save As Dialog Box



4 Select the export destination and change the export file name, if desired.

5 Click **Save As** to enter the file name.

Figure 32 Enter File Name Dialog Box (2)



6 Click **Export** to export the Project file (this process may take a few minutes).

7 The Export Status message box shown in Figure 33 appears after the file has been exported successfully.

Figure 33 Export Status Message Box

8 Click **OK** to close the message box.

Exporting a Project Using the Command Line

You can also export a Project or Environment using the following command-line script.

Location of script file:

ICAN-root\repository\util\exportProject.bat (or exportProject.sh)

Command Syntax:

```
exportProject username password exportfile projectname
    environmentname
```

where:

- ***exportfile*** is the name and path for the archive file to contain the Project and/or Environment you are exporting.
- ***projectname*** is the name of the Project you are exporting. If you are exporting an Environment only, leave this parameter as an empty string ("").
- ***environmentname*** is the name of the Environment you are exporting. If you are exporting a Project only, leave this parameter as an empty string ("")

To export a Project using the export script

1. Open a command prompt and change directory to ***ICAN-root***\repository\util.
2. To save the Project **myProject** to the file **c:\project4export.zip**., type:

```
exportProject username password c:\project4export.zip myProject ""
```

To export an Environment using the export script

1. Open a command prompt and change directory to ***ICAN-root***\repository\util.
2. To save the Environment **myEnvironment** to the file **c:\environment4export.zip**., type:

```
exportProject username password c:\environment4export.zip ""
    myEnvironment.
```

To export a Project and an Environment using the export script

1. Open a command prompt and change directory to ***ICAN-root***\repository\util.
2. To save the Project **myProject** and Environment **myEnvironment** to the file **c:\projenv4export.zip**., type:

```
exportProject username password c:\projenv4export.zip myProject
    myEnvironment.
```

4.6.3 Impact Analyzer

The Impact Analyzer helps you determine how a change to one component of a Project or Environment will affect other components in that Project or Environment.

To perform an Impact Analysis

1. Select a component in either the Project Explorer or Environment Explorer.
2. Click the **Impact Analyzer** button, or select **Impact Analyzer** from the Tools menu, to display the *Impact Analyzer* dialog box shown in Figure 34.
3. In the *Please show me* drop-down list, select items you would like to view. You have the option of viewing either:
   - Objects that have references to the selected object.
   - Objects that are referenced by the selected object.
4. You can filter the number of listed objects using the *Please show me impacted objects in* drop-down list; the default is the entire **Repository**.
5. You can print the object list by clicking **Print** to display the Windows *Print* dialog box.

Figure 34 Impact Analyzer Dialog Box

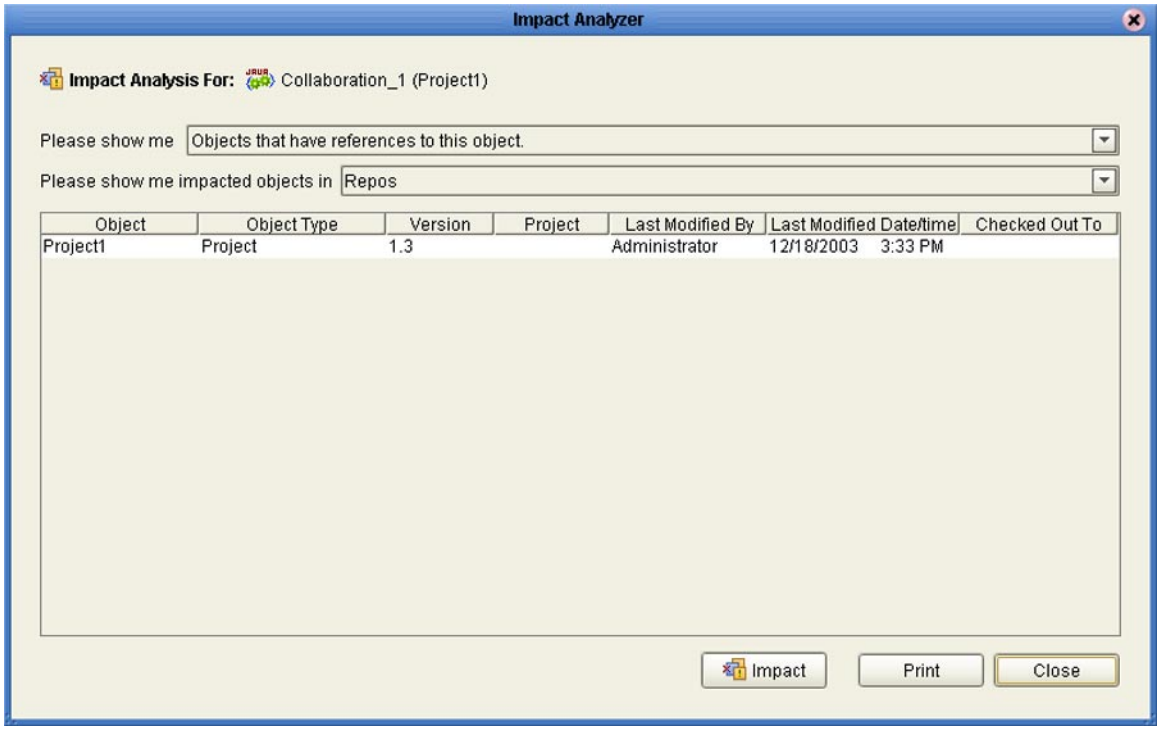
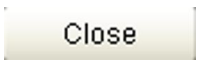


Table 13 Impact Analyzer Command Buttons

Button	Function
	Performs an impact analysis for the object selected from the object list (not fully implemented).
	Displays the Windows **Print** dialog box, which you can use to print the object list.
	Closes the **Impact Analyzer** dialog box.

4.6.4 Version Control

Version control allows you to maintain multiple versions of selected Project or Environment components. The version history of each component is recorded to a log file, and can be viewed by means of a menu option (see **Viewing a Component's Version History** on page 59).

Important: *More than one person concurrently using the same user ID will circumvent this version control system, and one person's work can be overwritten by another. You should ensure that all personnel using Enterprise Designer use unique IDs.*

Checked-In State

When a component is checked in to the version control system, it is locked against modification until being checked out, and a lock is displayed in the component's icon in the Enterprise Explorer. Figure 35 shows the lock, using the OTD icon as an example. See **Checking a Component In** on page 60, **Checking a Component In Without Revisions** on page 61, and **Checking In a Previous Version as the Latest Version** on page 65.

Figure 35 *Checked In* Icon (OTD Example)

Checked-Out State

When the latest version of a component is checked out from the version control system, it is locked against another user checking it out. A writing pad icon (see Figure 36) is displayed next to the component's icon in the Enterprise Explorer, indicating that it is checked out. See **Checking a Component Out** on page 62.

Figure 36 *Checked Out* Icon

Retrieved State

When any version of a component is retrieved from the version history dialog box, it is *not* locked against another user checking it out or retrieving it. A combined writing pad/warning icon (see Figure 37) is displayed next to the component's icon in the Enterprise Explorer, indicating that it is in your workspace—but warning you that it is not locked in any way. See **Retrieving a Component to Your Workspace** on page 63 and **Checking In a Previous Version as the Latest Version** on page 65.

Figure 37 *Retrieved* Icon

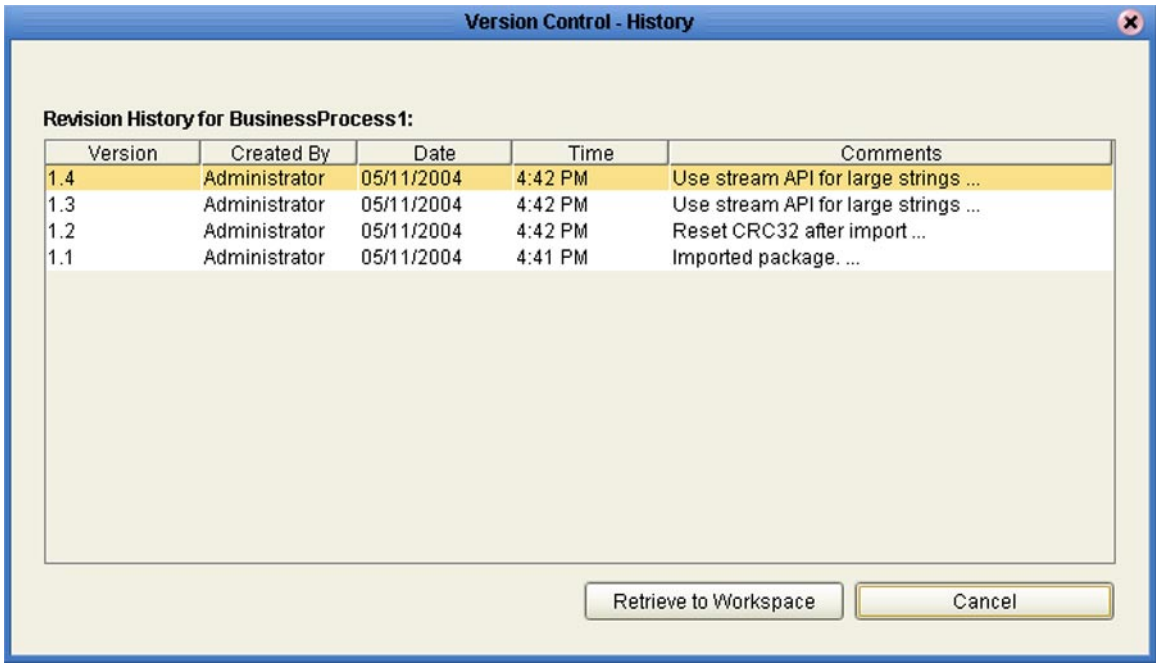
Viewing a Component's Version History

To view the version history for a component

1 In the Enterprise Explorer, select the component and right-click to display its context menu.

2 Select **Version History** to display the *Version Control - History* information box shown in Figure 38.

Figure 38 Version Control - History Dialog Box



3 Right-click in the *Comments* column to display the full text of the comment.

4 Click **Cancel** to close the box.

Note: *If a version is checked out or retrieved to a user's workspace, the appropriate icon also appears in the Version column.*

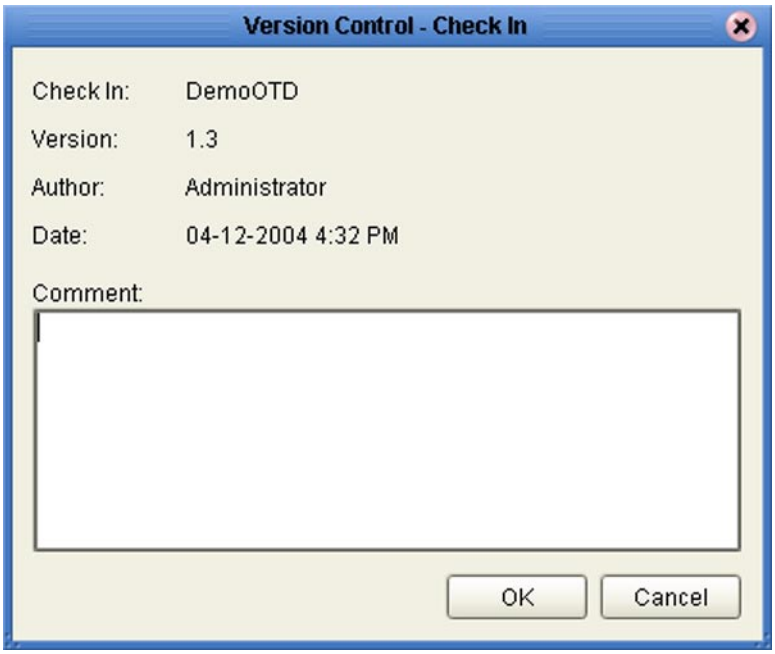
Checking a Component In

Once you have created and configured a component for the first time, or created a revised version of an existing component, you must check that component in to save it to the common area of the Repository and release your lock on the object.

To check in a new version of a Project or Environment component

1. In the Enterprise Explorer, select the component and right-click to display its context menu.
2. Select **Check In** to display the *Version Control - Check In* dialog box shown in Figure 39.

Figure 39 Version Control - Check In Dialog Box



3. Type in a description of the changes in the new version.
4. Click **OK** to check the new version in.

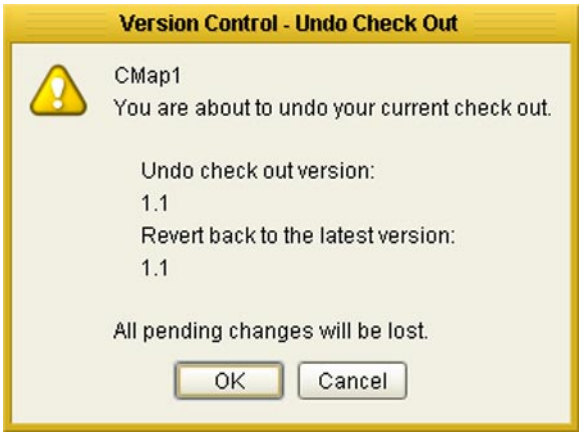
Checking a Component In Without Revisions

When you have checked the latest version of a component out and want to check it back in without any revisions, you can simply cancel the check-out by using the following procedure. The version number is not incremented.

To check in a Project or Environment component without revisions

1. In the Enterprise Explorer, select the component and right-click to display its context menu.
2. Select **Undo Check Out** to display the *Version Control - Undo Check Out* dialog box shown in Figure 40. (This option is not available for all components.)

Figure 40 Version Control - Undo Check Out Dialog Box



3. Click **OK** to check the currently checked-out version back in.

Note: *This procedure is also valid for retrieved versions.*

Checking a Component Out

You can check out the current version of a component by using the following procedure.

Note: *Only one user can have a file checked out for editing at a time. If another user attempts to check out the same file, they will receive a message indicating that the file is currently checked out.*

To check out the latest version of a Project or Environment component

1. In the Enterprise Explorer, select the component and right-click to display its context menu.
2. Select **Check Out** to display the *Version Control - Check Out* dialog box shown in Figure 41.

Figure 41 Version Control - Check Out Dialog Box

3. Click **OK** to check the component out.
4. Click the **Save** or **Save All** icon to place the version in your Repository workspace.

Retrieving a Component to Your Workspace

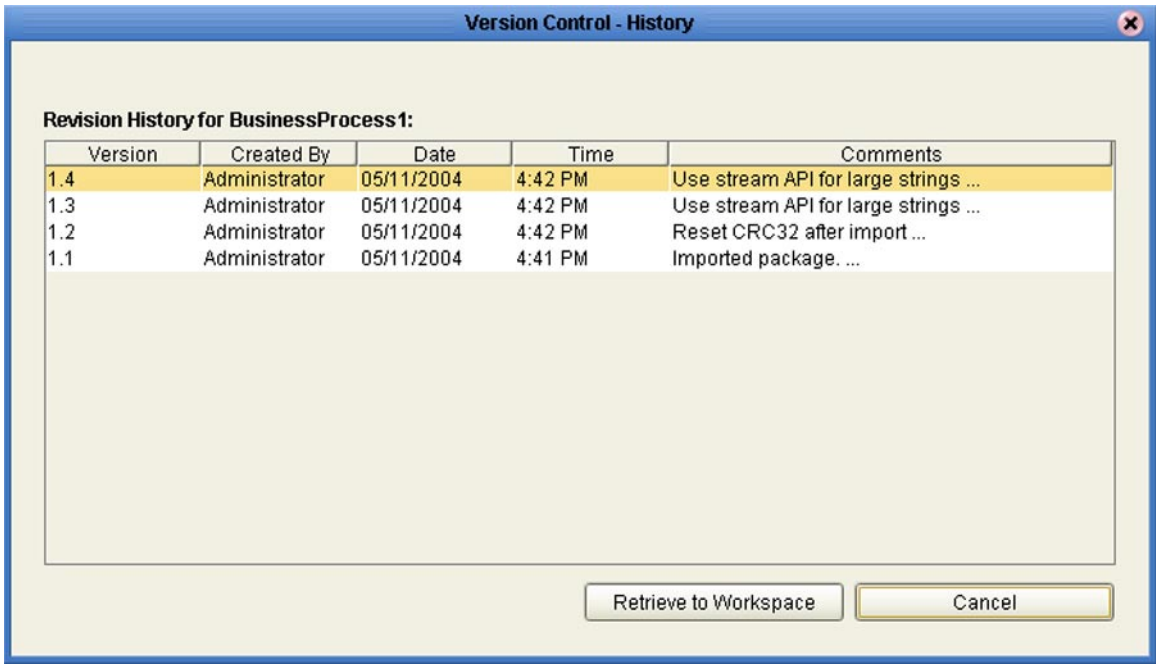
You can retrieve either the current or a previous version of a component by retrieving it from the Version History information box. Retrieving does *not* lock the file from being checked out or retrieved by other users for editing. To check a retrieved version back in as the latest version, you must use the **Make Latest** option described in **Checking In a Previous Version as the Latest Version** on page 65.

Note: *Currently applies only to Business Processes.*

To retrieve an older version of a Project or Environment component

1. In the Enterprise Explorer, select the component and right-click to display its context menu.
2. Select **Version History** to display the *Version Control - History* dialog box shown in Figure 38.

Figure 42 Version Control - History Dialog Box



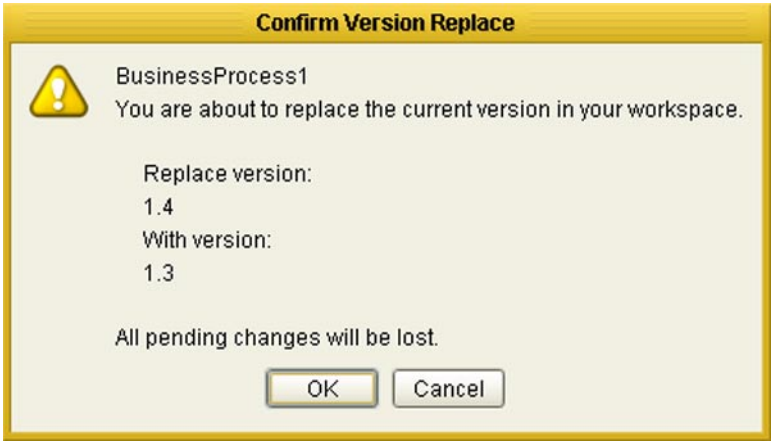
3. Select the version you want to retrieve and click **Retrieve to Workspace**.
   A. If you are attempting to retrieve the *latest* version of the component, you will be presented with the dialog box shown in Figure 43.

Figure 43 Access File Dialog Box

- **Check Out for Edit** copies the file to your workspace and locks it—the file becomes read-only to other users. This is the same mechanism as described in **Checking a Component Out** on page 62. To check the latest version out from the dialog box, you must select this option and click **OK**. You will then be presented with the dialog box shown previously (Figure 41).
- **Retrieve to Workspace** copies the file to your workspace, but does not prevent it from being checked out or retrieved by other users. This is the default setting for the dialog box; simply click **OK**.

B If you are attempting to retrieve a *previous* version of the component, you will be presented with the dialog box shown in Figure 44. Clicking **OK** will overwrite any other version you have retrieved to your workspace, or replace the currently checked-in version in your workspace only—other users will be unaffected.

Figure 44 Confirm Version Replace Dialog Box



Note: *If you have the latest version of the component checked out to your workspace, the* ***Checked Out*** *icon will appear in the Version column of the Version History dialog and the* ***Retrieve to Workspace*** *button will be disabled. You must check the latest version back in to version control before you can retrieve any version.*

Checking In a Previous Version as the Latest Version

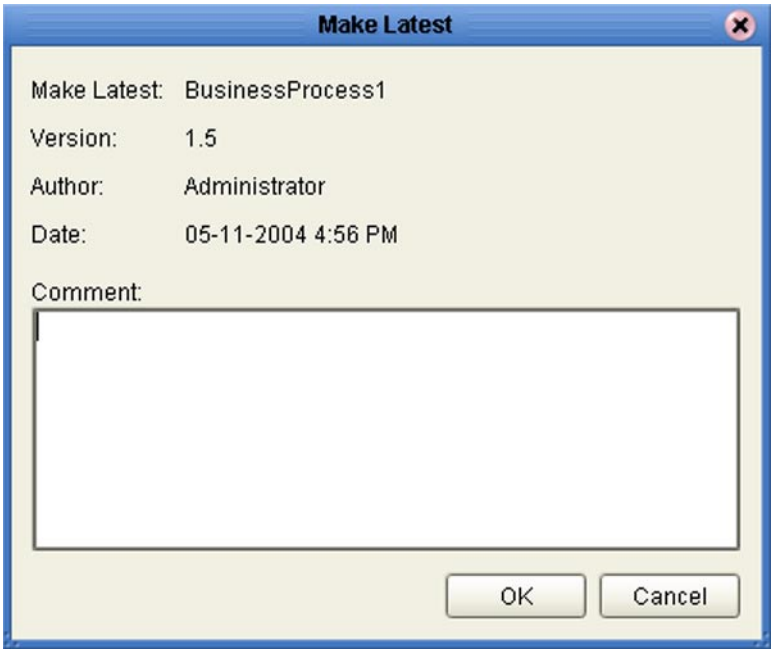
If you have retrieved a previous version of a component to your workspace, you can check it in to the version control system as the latest version by selecting the *Make Latest* option.

Note: *Currently applies only to Business Processes.*

To make a previous version of a component become the latest version

1. In the Enterprise Explorer, select the component and right-click to display its context menu.
2. Click **Make Latest** to display a confirmation dialog box.
   - A If the latest checked-in version of the component has not changed since you retrieved the previous version, you will see the dialog box shown in Figure 44.

Figure 45 Make Latest Dialog Box



Type in a description of the changes in this version and click **OK**. The version in your workspace will be checked in as the latest version of the component.

- B If the latest checked-in version of the component is different from the one that was current when you retrieved the previous version, you will first see the dialog box shown in Figure 46.

Figure 46 Confirm Latest Version Override Dialog Box

If you are sure you want to replace the current latest version, click **OK** to display the dialog box shown previously (Figure 45). Type in a description of the changes in this version and click **OK**. The version in your workspace will be checked in as the latest version of the component.

Important: *This situation can occur if another user has made changes to the latest version—you must use caution when checking in your version, since the other user's changes will be superseded.*

Command-line Utilities

If you encounter problems with the version control system, there are two command-line utilities—a Repository version control utility and a workspace cleanup script—that can be run by personnel with Administrator privileges. These utilities should be used as a last resort, and with the utmost caution. See the *eGate Integrator System Administration Guide* for information.

Chapter 5

eGate Project Components

This chapter describes the eGate components of an eInsight Project, and the use of the Enterprise Designer in defining your Project.

5.1 Overview

An eInsight Project represents the logical system designed to solve either all or part of a business problem. Projects are created using tools contained within the Enterprise Designer, and are deployed to specific Logical Hosts in specific Environments by means of Deployment Profiles (see **Environments** on page 110). Components developed for use in one Project can be used in another, and a Project can internally reference another Project.

Note: *See the eInsight Enterprise Service Bus User's Guide for a detailed explanation of the steps involved in setting up an eInsight ESB Project.*

5.1.1 Project Components

The components found in a typical Project are described in the following sections of this chapter:

- **Using the Connectivity Map Editor** on page 77
- **Web Service Application** on page 79
- **External Applications** on page 80
- **Component Connections** on page 82

Behind the scenes, and not explicitly shown in a Connectivity Map, are other Project components such as:

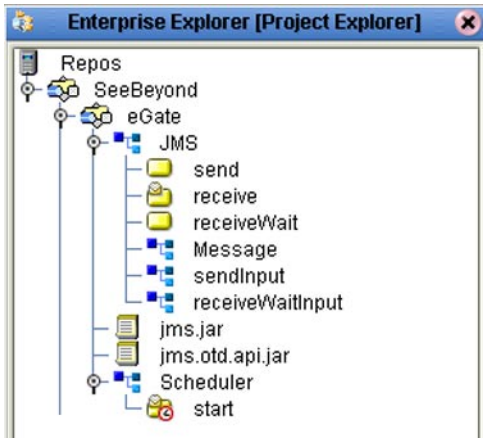
- **Object Type Definitions**

  Object Type Definitions (OTDs) are sets of rules that define the encoding of an object. They describe messages that are propagated through eGate, and the methods available for operating on them, and also interactions with external APIs. See **Object Type Definitions** on page 87.

5.2 The Project Explorer

A Project consists of logical constructs and configurations designed to solve some or all of a business problem. The **Project Explorer** displays the contents of the Repository that belong to the selected Project (see Figure 47).

Figure 47 Project Explorer



The Project Explorer is used in conjunction with the Connectivity Map Editor (see **Using the Connectivity Map Editor** on page 77) to create and configure a Project. Each component in the Project Explorer has an icon to identify the component type (see **Project Explorer Icons** on page 69). Right-clicking on a component displays a context menu for that component (see **Context Menus** on page 70), from which you can select appropriate actions.

Note: *Select **Refresh All from Repository** before you open any Project component (such as a service) to ensure that you open the latest version of the component.*

5.2.1 Project Explorer Icons

The icons described in Table 14 appear in the Project Explorer.

Table 14 Project Icons

Icon	Description
	Represents the **Repository**, which is the central ICAN database where all Project information is saved. Binary files required at run time are also stored here.
	Represents the **Project** or subproject.
	Represents a **Connectivity Map**, which contains the business logic and information about the data transmission. A lock displayed in the lower-left corner indicates that the Connectivity Map is currently checked into the version control system (see OTD example).
	Represents a **Project variable** or **constant**.
 	Represents an **Object Type Definition** (OTD) file. A lock displayed in the lower-left corner indicates that the OTD is currently checked into the version control system.
	Represents a **Deployment Profile**, which specifies how Project components will be deployed in an Environment.

5.2.2 Context Menus

Right-clicking a component in the Project Explorer displays a context menu for that component. Only those menu options that are allowed for the component in its current state are activated.

Repository Menu

Figure 48 Repository Menu

Table 15 Repository Menu Options

Option	Function
Project	Adds a new Project to the Repository.
Sort by Type	Places all objects in order by grouping object types.
Sort by Name	Places all objects in alphabetical order.
Sort by Date	Places all objects in order by creation date, from oldest to newest.
Import	Displays a dialog box with which you can import a Project or Environment into the Repository.
Export	Displays a dialog box with which you can export a Project or Environment from the Repository to an archive file.
Refresh All from Repository	Refreshes the Project Explorer to display the current contents of the Repository. (Open editors are not refreshed.)
User Management	Displays the User Management dialog box, where an Administrator can manage user access to the Repository with options for adding, modifying, and deleting users. See the *eGate Integrator System Administration Guide*.
Properties	Displays a dialog box showing the configuration properties of your Repository. See the *eGate Integrator System Administration Guide*.

Project Menu

Figure 49 Project Menu

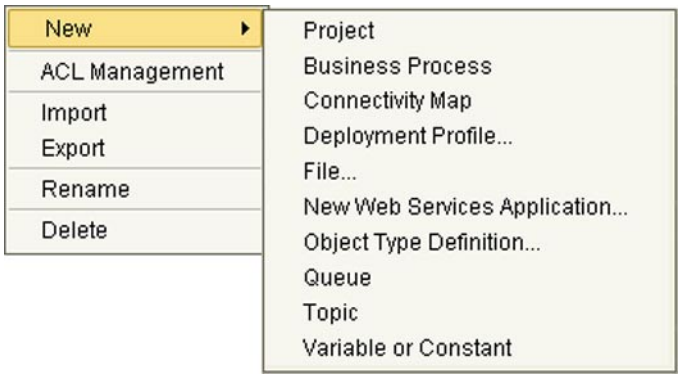


Table 16 Project Menu Options

Option	Option	Function
New	Project	Adds a Subproject folder to the selected Project.
	Business Process	If eInsight Business Process Manager is installed, displays the user interface for creating a new Business Process.
	Connectivity Map	Adds a Connectivity Map to the Project. See **Using the Connectivity Map Editor** on page 77.
	Deployment Profile	Displays a dialog box with which you can create a Deployment Profile for the selected Project. See **The Deployment Editor** on page 155.
	File	If the File eWay is installed, displays a dialog box with which you can create an external file to use with the Project. (This is an example of an External Application—the applications that are displayed in the menu depend upon which eWays are installed on your system.)
	New Web Services Appl.	Adds a Web services application to the selected Project. See **Web Service Application** on page 79.
	Object Type Definition	Displays the **OTD Wizard**, with which you can create an Object Type Definition (OTD) file. See **Using the OTD Wizards** on page 88 for more information.
	Queue	Adds a queue to your Project.
	Topic	Adds a topic to your Project.
	Variable or Constant	Displays a dialog box with which you can add a constant or variable icon to your Project.
ACL Management		Displays the ACL Properties dialog box, with which an Administrator can assign read/write/delete privileges to users for the selected Project. See the *eGate Integrator System Administration Guide*.

Table 16 Project Menu Options

Option	Option	Function
Import		Displays a dialog box with which you can import a Project as a Subproject under the selected Project. See **Project/ Environment Import** on page 47.
Export		Displays a dialog box with which you can export the selected Project. See **Project/Environment Export** on page 51.
Rename		Activates the field, allowing you to rename the selected Project.
Delete		Deletes the selected Project, subject to the following conditions: ▪ You have *delete* privileges for the Project (see ACL Management, above). ▪ The Project is not checked out by anyone other than yourself. If these conditions are true, a dialog box is displayed in which you confirm that you want to delete the selected Project. Clicking **Yes** then deletes the Project.

Connectivity Map Menu

Figure 50 Connectivity Map Menu

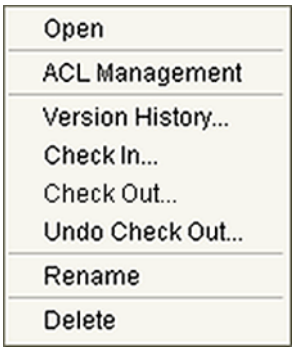


Table 17 Connectivity Map Menu Options

Command	Function
Open	Opens the Connectivity Map Editor for the selected Connectivity Map. See **Using the Connectivity Map Editor** on page 77.
ACL Management	Displays the ACL Properties dialog box, with which an Administrator can assign read/write/delete privileges to users for the selected Connectivity Map. See the *eGate Integrator System Administration Guide*.
Version History	Displays a dialog box with which you can track the version history for the selected Connectivity Map. See **Viewing a Component's Version History** on page 59 for more information.
Check In	Displays a dialog box with which you can check in a new version of the selected Connectivity Map. See **Checking a Component In** on page 60 for more details.
Check Out	Displays a dialog box with which you can check out the current version of the selected Connectivity Map. See **Checking a Component Out** on page 62 for more information.
Undo Check Out	Displays a dialog box with which you can undo the check-out of the selected Connectivity Map. See **Checking a Component In Without Revisions** on page 61 for more information.
Rename	Activates the field, allowing you to rename the selected Connectivity Map.
Delete	Deletes the selected Connectivity Map, subject to the following conditions: ▪ You have *delete* privileges for the Connectivity Map (see ACL Management, above). ▪ The Connectivity Map is not checked out by anyone other than yourself. If these conditions are true, a dialog box is displayed in which you confirm that you want to delete the selected Connectivity Map. Clicking **Yes** then deletes the Connectivity Map.

Object Type Definition Menu

Figure 51 OTD Menu

Table 18 OTD Menu Options

Command	Function
Open	Opens the OTD Editor, showing the selected OTD. See **OTD Editor** on page 90.
ACL Management	Displays the ACL Properties dialog box, with which an Administrator can assign read/write/delete privileges to users for the selected OTD. See the *eGate Integrator System Administration Guide*.
Version History	Displays a dialog box with which you can track the version history for the selected OTD. See **Viewing a Component's Version History** on page 59 for more information.
Check In	Displays a dialog box, with which you can check in a new version of the selected OTD. See **Checking a Component In** on page 60 for more details.
Check Out	Displays a dialog box with which you can check out the current version of the selected OTD. See **Checking a Component Out** on page 62 for more information.
Relaunch (XSD OTD only)	Relaunches the XSD OTD Wizard, so that you can re-define an XSD OTD while retaining the original OID. The following conditions apply: ▪ You must have *write* privileges for the OTD (see ACL Management, above). ▪ The OTD must not be checked out by anyone other than yourself. ▪ The OTD must not be imported from another Project. ▪ The OTD must have been created in eGate Integrator 5.0.4 (or later release).
Rename	Activates the field, allowing you to rename the selected OTD.
Delete	Deletes the selected OTD, subject to the following conditions: ▪ You have *delete* privileges for the OTD (see ACL Management, above). ▪ The OTD is not checked out by anyone other than yourself. If these conditions are true, a dialog box is displayed in which you confirm that you want to delete the selected OTD. Clicking **Yes** then deletes the OTD.

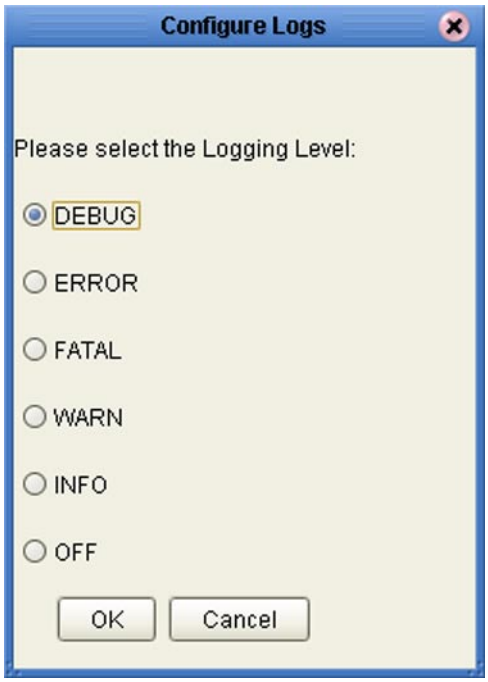
Deployment Profile Menu

Figure 52 Deployment Profile Menu

Table 19 Deployment Profile Menu Options

Command	Function
Version History	Displays a dialog box with which you can track the version history for the selected Deployment Profile. See **Viewing a Component's Version History** on page 59 for more information.
Check In	Displays a dialog box, with which you can check in a new version of the selected Deployment Profile. See **Checking a Component In** on page 60 for more details.
Check Out	Displays a dialog box with which you can check out the current version of the selected Deployment Profile. See **Checking a Component Out** on page 62 for more information.
Undo Check Out	Displays a dialog box with which you can undo the check-out of the selected Deployment Profile. See **Checking a Component In Without Revisions** on page 61 for more information.
Open	Opens the Deployment Editor, showing the selected Deployment Profile. See **The Deployment Editor** on page 155.
Rename	Activates the field, allowing you to rename the selected Deployment Profile.
Delete	Deletes the selected Deployment Profile, subject to the following conditions: ▪ You have *delete* privileges for the Deployment Profile (see ACL Management, above). ▪ The Deployment Profile is not checked out by anyone other than yourself. If these conditions are true, a dialog box is displayed in which you confirm that you want to delete the selected Deployment Profile. Clicking **Yes** then deletes the Deployment Profile.

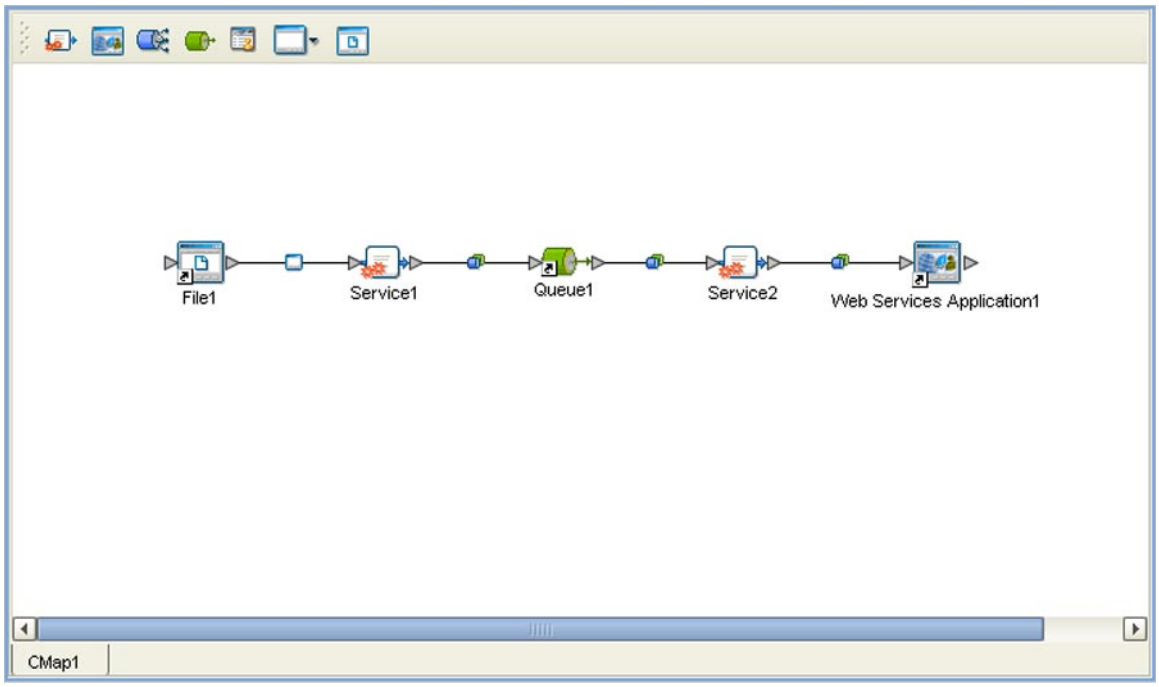
Table 19 Deployment Profile Menu Options

Command	Function
Configure Log	Displays a dialog box in which you can set the logging level for the selected Deployment Profile.  The default value is DEBUG. See the *eGate Integrator System Administration Guide* for information about logging.
ACL Management	Displays the ACL Properties dialog box, with which an Administrator can assign read/write/delete privileges to users for the selected Deployment Profile. See the *eGate Integrator System Administration Guide*.

5.3 Using the Connectivity Map Editor

When you create a new Connectivity Map in the Enterprise Explorer, the editor panel displays the Connectivity Map Editor (see Figure 53). To define your Project, you simply drag icons from the toolbar to the workspace, or canvas, to populate the Connectivity Map with the necessary components. You subsequently link the components by dragging the cursor from one to the other.

Figure 53 Connectivity Map Window



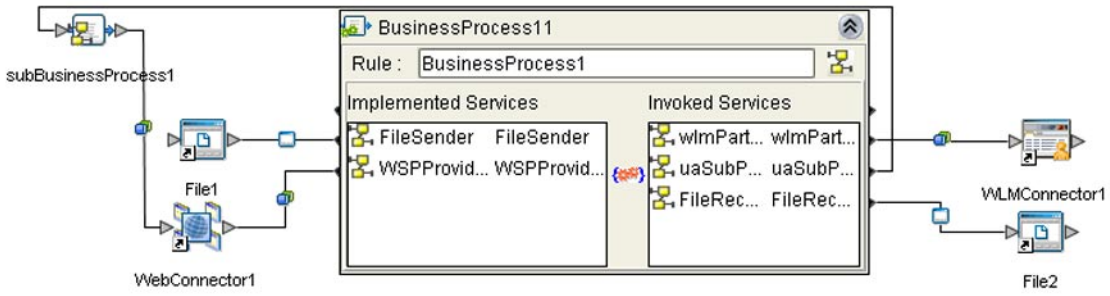
The drag-and-drop components include services, queues, topics, schedulers, and external applications. Additional components, such as eWays and JMS Clients, are placed automatically when you link the components you have placed manually.

A service provides a framework for a business process, which contains the information required to execute a set of business rules. See the *eInsight Enterprise Service Bus User's Guide* for information regarding business processes.

A *message destination* is a container for stored data, and can follow either the topic or queue JMS model. A *topic* is a message destination that conforms to the publish-and-subscribe messaging paradigm.A *queue* is a message destination that conforms to the point-to-point messaging paradigm. See the *eGate Integrator JMS Reference Guide* for detailed information on message destinations.

When there are multiple destinations, the Connectivity Map Editor cannot resolve which output port connects to which destination. Because of this, the Business Process must be defined first, and the connections must be drawn by opening the Business Process Binding box in Connectivity Map (see Figure 54).

Figure 54 Linking Multiple Message Destinations



The Connectivity Map Editor toolbar contains the icons listed in Table 20, plus additional icons representing eGate add-ons and other ICAN components that you may have installed.

Table 20 Connectivity Map Toolbar Icons

Icon	Component	Function
	Service	A logical component that provides the framework for a business process. See the *eInsight Enterprise Service Bus User's Guide* for information.
	Queue	A Message Destination that conforms to the point-to-point messaging paradigm, having one sender and one receiver. See the *eGate Integrator JMS Reference Guide* for information.
	Topic	A Message Destination that conforms to the publish/subscribe messaging paradigm, having one sender (publisher) and multiple receivers (subscribers). See the *eGate Integrator JMS Reference Guide for* information.
	Web Services Application	Represents a Web services application (see **Web Service Application** on page 79).
	External Applications	Represents an application external to eGate. Click the arrow beside the icon to view a list of specific applications to which you can connect. See **External Application Drop-Down Menu** on page 80.
	Scheduler	Represents a scheduling component of the Connectivity Map. Use this component to set data transfer to occur at set intervals. See **Schedulers** on page 80.

It is important to understand that the logical components appearing in the Connectivity Map are essentially *placeholders* that refer to the "actual" components that exist in the Repository and appear in the Project Explorer. Renaming or deleting a queue or topic in the Connectivity Map only affects the placeholder, not the object in the Repository.

Also, renaming or deleting a queue or topic in the Repository will not affect the existence or name of the associated placeholder in the Connectivity Map. The change

will, however, be reflected in the *tooltips* for the placeholder. This allows you to re-assign the placeholder without disrupting the continuity of the Connectivity Map.

5.4 Web Service Application

A Web Services Application (or Web Services External System) represents either an external Web Service that can be invoked by a Web Client containing an eInsight Business Process, or a Web Server containing a Business Process that is exposed as a Web Service (see **Web Services** on page 169). In the case of the Web Server, the exposed Web Service is listed in the SeeBeyond UDDI Registry.

The Properties dialog box (see Figure 55) appears automatically when you create a New Web Services External System in the Environment Explorer. You can also invoke the dialog box by right-clicking the Web Services Application in the Enterprise Explorer and selecting **Properties**. You should assign the desired values for the listed properties.

Figure 55 Web Service Application Properties Dialog Box

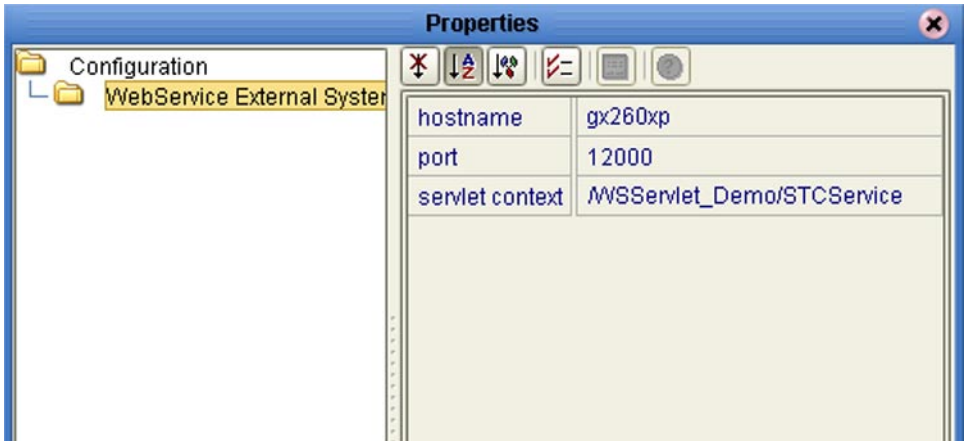


Table 21 Web Service Application Properties

Property	Description
hostname	Name of the computer hosting the application.
port	Port on which the application can be accessed.
servlet context	The path and name of the Web Services Application.

If you do not specify the properties for the Web Service, then they are assigned automatically upon deployment. The name automatically assigned to the **servlet context** property can be lengthy and complex, however, so you should rename it appropriately. This property controls the name of the Web Service Application (**.war**) file.

Additionally, if you are creating more than one Web Service Application for deployment in any given Environment, you must ensure that the respective servlet

context properties have different values so that each application will have a different URL.

5.5 External Applications

The basic purpose of eGate Integrator is to facilitate the interchange of data between external business applications. These business applications are collectively referred to as external applications, and are represented in the Project by logical proxies for the specific applications involved. An external application can be identified with an ERP application such as SAP or PeopleSoft, a DBMS such as Oracle or SQL, or with a particular communications protocol, such as TCP/IP or HTTPS.

External applications are logical representations of external software applications that are being integrated by the eGate system. These are linked to a Service by means of an eWay. Clicking the drop-down arrow beside the external application icon displays a menu showing those applications corresponding to eWays that have been purchased and installed, plus the Scheduler. An example is shown in Figure 56.

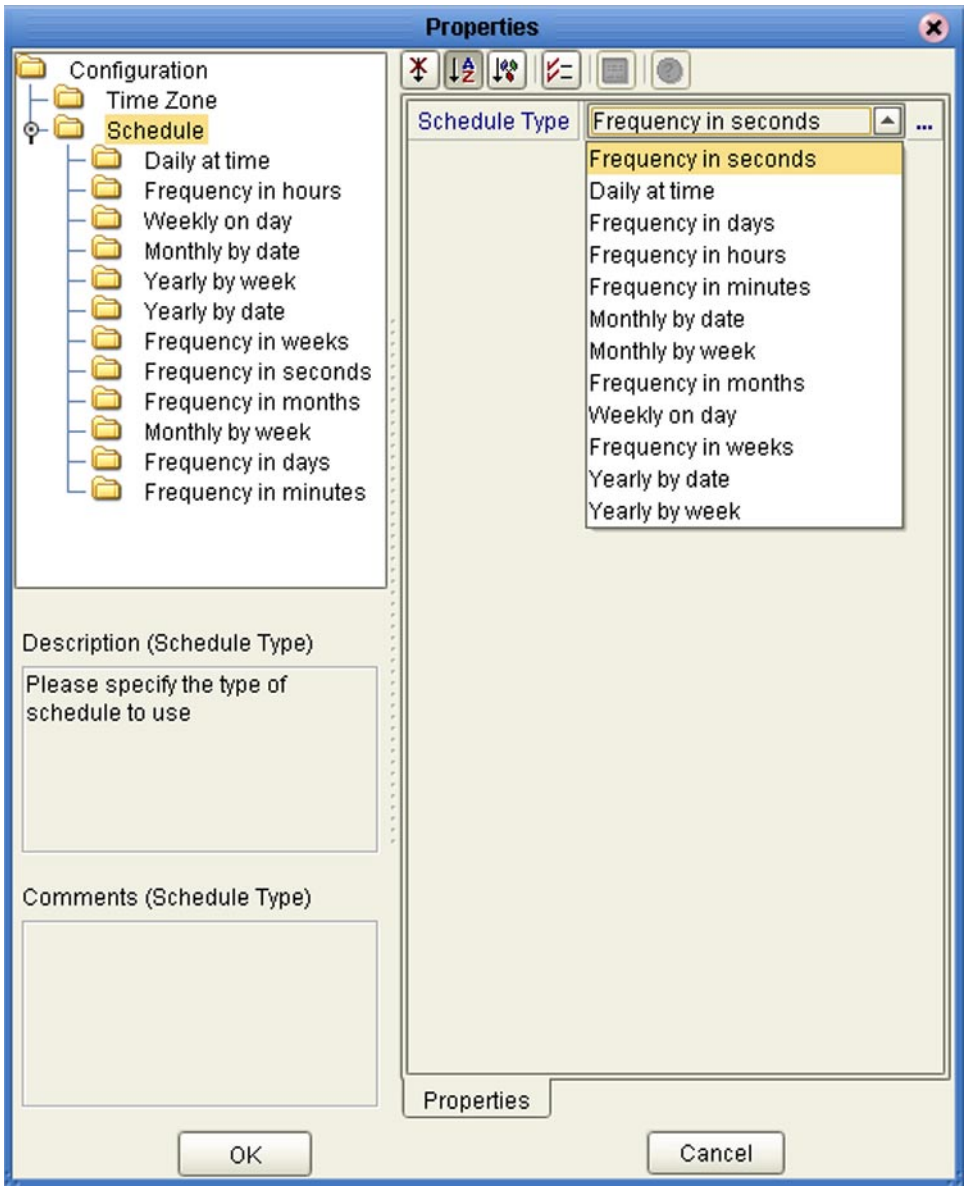
Figure 56 External Application Drop-Down Menu

Selecting the check box beside an individual external application adds that icon to the toolbar; clearing the check box removes it from the toolbar.

5.5.1 Schedulers

A Scheduler allows a service to be performed at a prescribed interval. The interval can be static, or can be made dynamic by using a Project variable for the interval value. Once the scheduler is connected to a service in the Connectivity Map, double-clicking the JMS Client displays the Properties dialog box for that scheduler (seeFigure 57).

Figure 57 Scheduler Properties Dialog Box



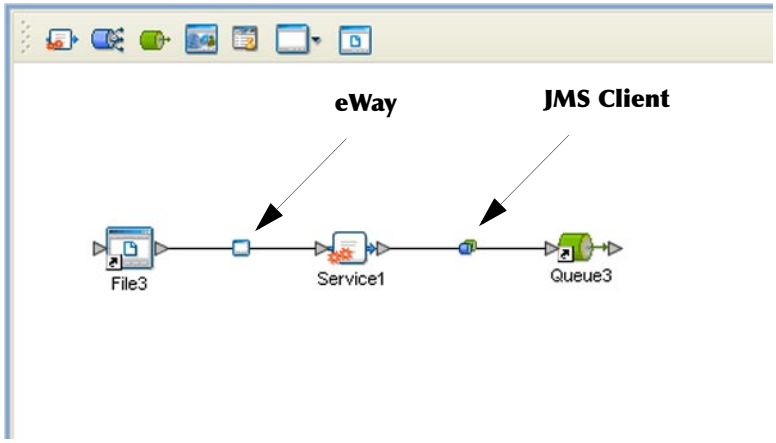
Selecting **Schedule** displays the **Schedule Type** property field which you set to the type of schedule you want to use. Selecting the corresponding node in the explorer tree displays the property field for that schedule type, in which you specify the desired value. The text in the *Description* box will include the appropriate units.

Selecting **Time Zone** displays the **Time Zone** property field in which you specify your local time zone, so that your schedule will be synchronized to the local time, if appropriate.

5.6 Component Connections

When you link two components on a Connectivity Map, the Enterprise Designer places either an eWay or JMS Client connection icon on the link, depending upon the type of components you are linking (see Figure 58).

Figure 58 Connection Icons in a Connectivity Map

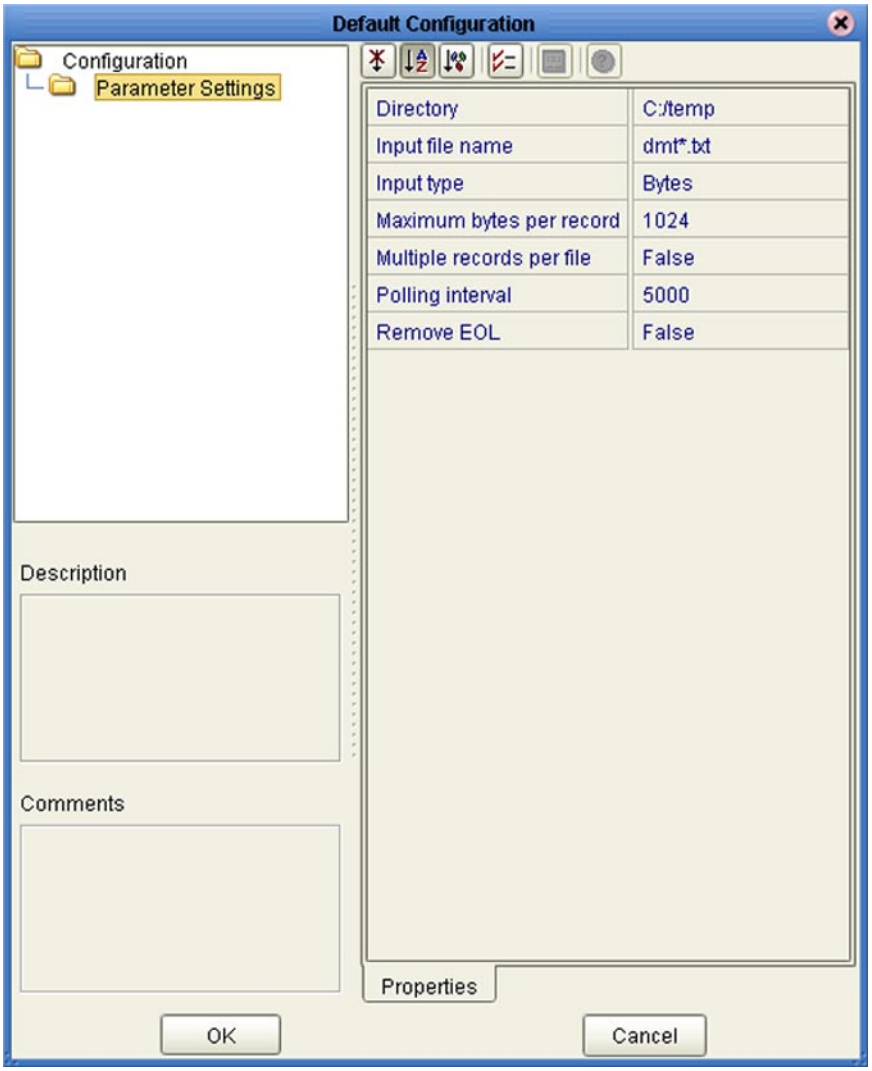


- When you link an external application with a Business Process, the Enterprise Designer automatically adds an eWay Connection icon to the link. The eWay enables communication and movement of data between the external application and the eGate system. The eWay configuration specifies the logical connection properties for the link. See the individual eWay Intelligent Adapter User's Guides for specific information.
- When you link a Service with a Message Destination (queue or topic), the Enterprise Designer adds a JMS Client Connection icon. The JMS Client configuration specifies the logical connection properties for the linked Service. See the *eGate Integrator JMS Reference Guide* for information.

5.6.1 Configuring a Connection

Double-clicking an eWay or JMS Client connection icon in the Connectivity Map displays the Default Configuration dialog box. As an example, Figure 59 shows a dialog box that lists the configuration properties for a File eWay.

Figure 59 Default Configuration Dialog Box



Note: *The first time you double-click an eWay or JMS Client icon, you will see a Templates dialog box. Here, you must designate an eWay to be inbound or outbound. Clicking* ***OK*** *will then display the Default Configuration dialog box.*

The constituent parts of the Default Configuration dialog box are:

- The **Configuration Tree** includes folders that contain configuration and connection properties for the selected eWay or message destination.
- The **Toolbar** contains a series of buttons used to sort and modify the information listed in the Properties folder, as described in Table 22.
- The **Description** box contains a brief description of the contents of the item currently selected in the Configuration Tree.
- The **Comments** box is for user comments about the item selected in the Configuration Tree.

Table 22 Configuration Dialog Box Toolbar Buttons

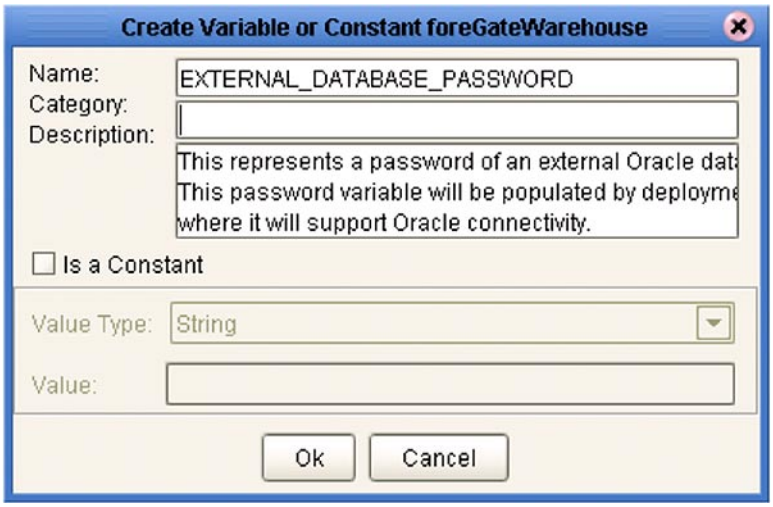
Button	Command	Function
	Unsorted	Displays configuration properties in their default order.
	Sort by Name	Sorts configuration properties alphabetically by name.
	Sort by Type	Displays configuration properties by property type.
	Show Editable Properties Only	Displays only the properties of an eWay or message destination that can be modified.
	Customizer	Displays the **Customizer** dialog box, which you can use to customize the selected eWay or message destination.
	Help	Displays the online help documentation for the Configuration Editor.

5.7 Defining Constants and Variables

You can define variables and constants for a specific Project. Variables function as placeholders, having values that are determined when you create a specific Deployment Profile (see **Mapping Variables** on page 162). Project variable values can be literals or Environmental constants.

For example, Figure 60 shows a project variable defined to represent a password of a database user in a target environment. System managers will assign an actual value to this variable in the deployment profile editor. The value of the assigned project variable—an Environment constant— is then used to connect the database in the target environment.

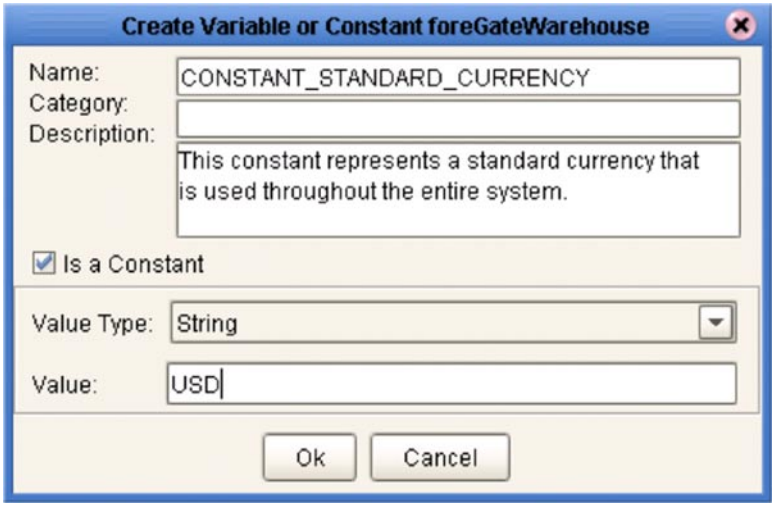
Figure 60 Project Variable Creation



Project constants are name/value pairs that are visible across the Project. For example, Figure 61 shows a standard currency defined to be used globally throughout the system.

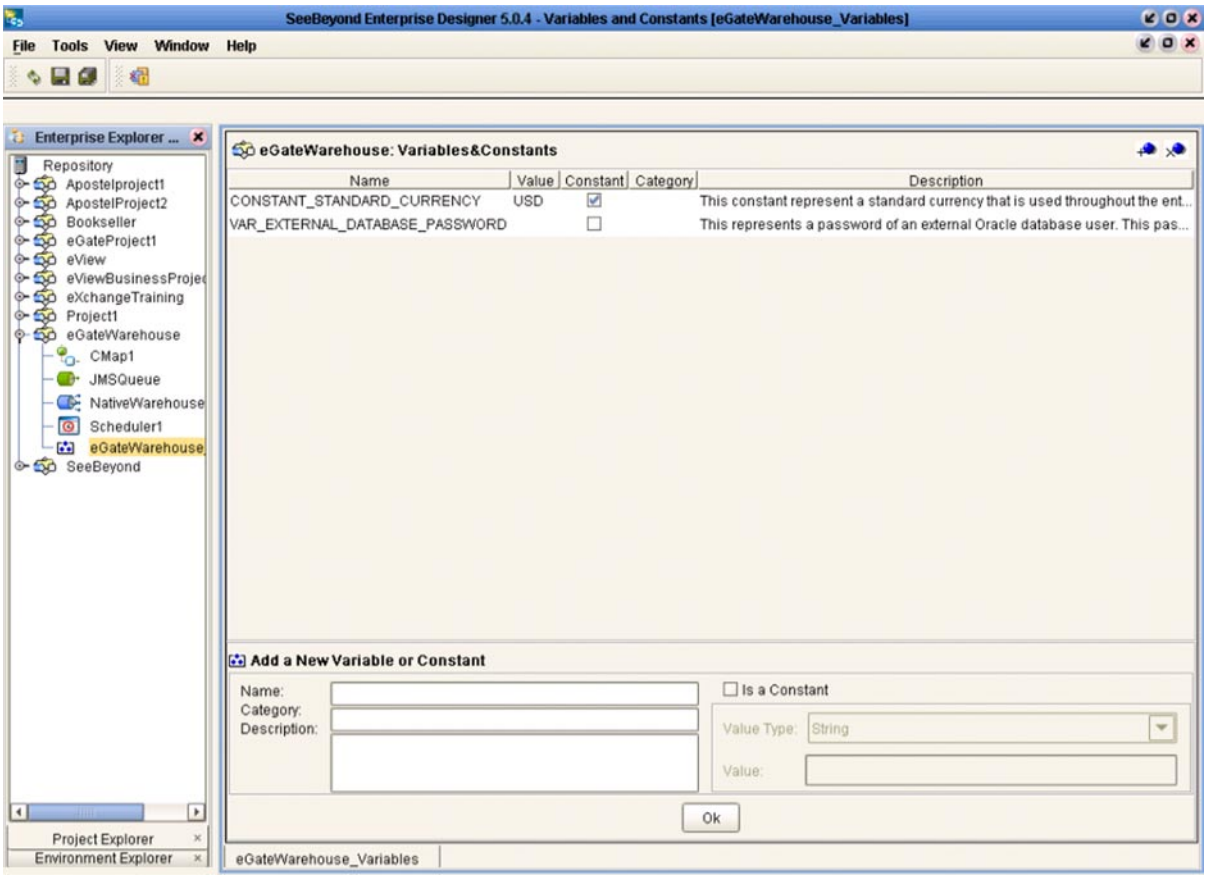
Note: *When you create an Project constant, you assign a permanent value to it—which cannot be overridden.*

Figure 61 Project Constant Creation



Constants and variables are automatically added to a Variables and Constants object group within the Project (see Figure 62).

Figure 62 Variables and Constants Object Group



Chapter 6

Object Type Definitions

This chapter contains a brief introduction to OTDs, and describes the procedures for creating externally-defined OTDs.

6.1 Overview

Object Type Definitions (OTDs) describe external data formats that characterize the input and output data structures in an eGate Project. OTDs typically have a specific external representation format that is used to store and transport the OTD contents through a Project. The OTD defines both this external representation and the run-time structure. At run time, an OTD instance is accessed from BPEL using XPath expressions.

6.1.1 OTD Types

Externally-Defined OTDs

Externally-defined OTDs are based on formats or standards external to eGate Integrator, such as Document Type Definition (DTD), Web Services Definition Language (WSDL), and XML Schema Definition (XSD). Some of these OTDs are *messagable*, others are API-based. Externally-defined OTDs are read-only.

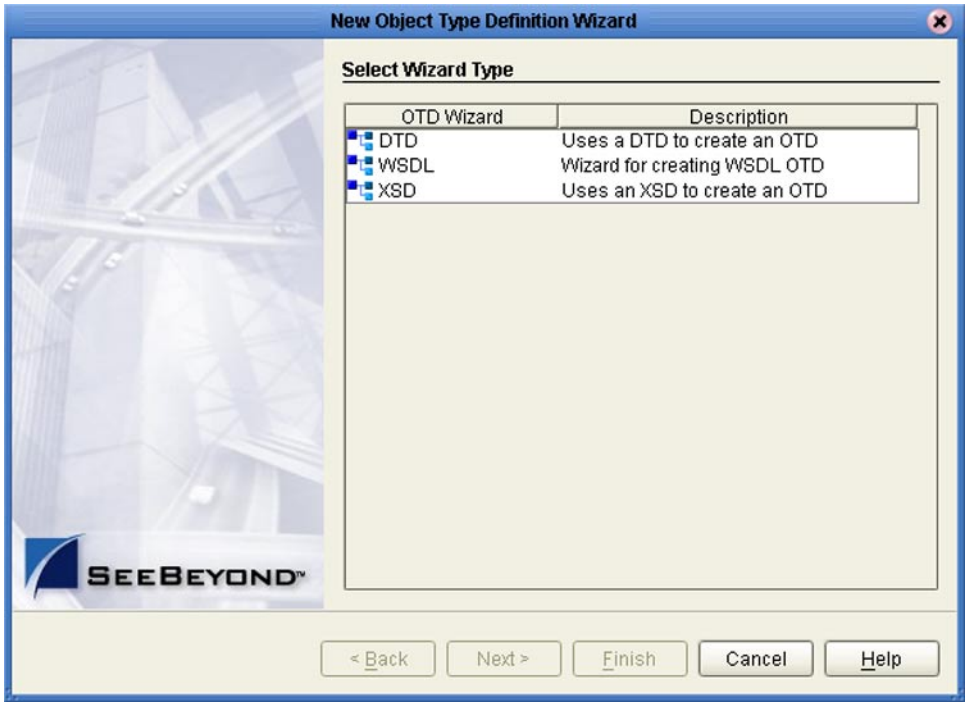
6.2 Using the OTD Wizards

Wizards are provided in the Enterprise Designer GUI to guide you through the OTD building process. These Wizards call back-end builders that actually implement the building of the code, based on the provided information.

Right-click on a Project in the Enterprise Explorer to display the Project context menu, then select **New Object Type Definition** to display the OTD Wizard, shown in Figure 63. The initial dialog allows you to select a specific OTD Wizard. The basic Wizards supplied with eGate Integrator are described in:

- **Using the DTD Wizard** on page 96
- **Using the WSDL Wizard** on page 101
- **Using the XSD Wizard** on page 105

Figure 63 OTD Wizard Selection Dialog



Additional OTD Wizards are supplied with eGate add-on components, and are described in the User's Guides for the specific products. When these products are installed, the OTD Wizards are added to the list shown in Figure 63.

The OTD Wizards guide you through the initial phases of creating an Object Type Definition, and then invoke the OTD Editor. The user interface is highly self-explanatory, but details of the navigation buttons are listed in Table 23 for your reference.

Table 23 OTD Wizard Navigation Buttons

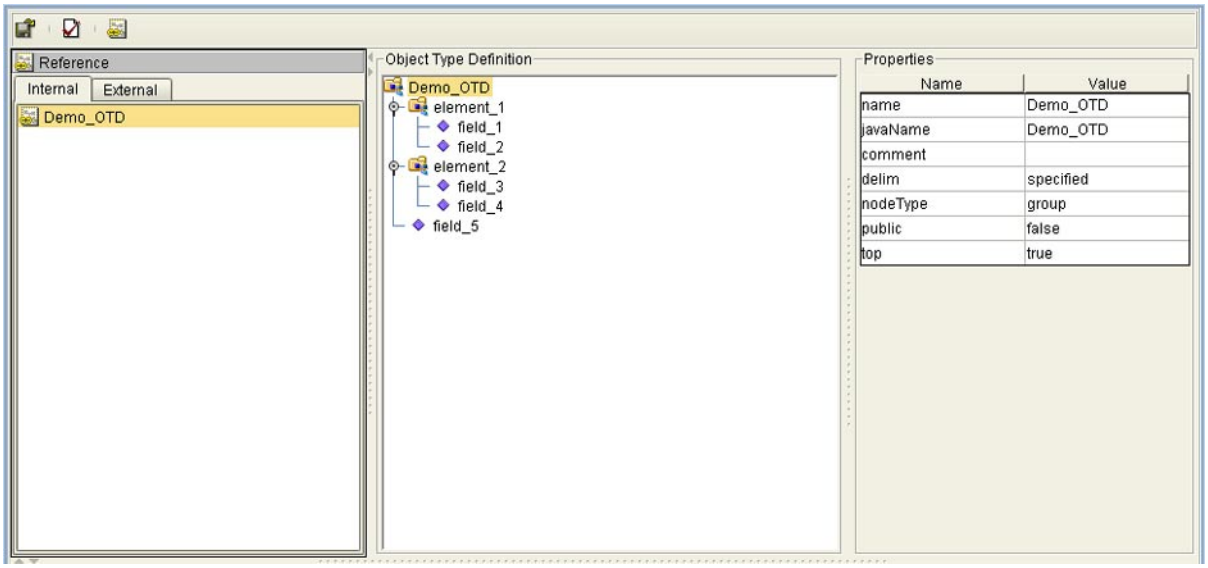
Button	Function
< Back	Returns to the previous step in the wizard. This button is disabled on the first step.
Next >	Goes to the next step in the wizard. This button is disabled on the last step.
Finish	Saves all OTD settings and closes the wizard. This button is only enabled on the last step.
Cancel	Closes the wizard without saving the OTD.
Help	Displays the online help documentation for the **OTD Wizard** dialog box.

6.3 OTD Editor

After you create an OTD file using the OTD Wizard, the OTD Editor appears in the editor panel of the Enterprise Designer, as shown in Figure 64. You can also invoke the OTD Editor by selecting **Open** in the context menu for an existing OTD in the Project Explorer. OTDs are saved to the Project automatically.

Note: *Remember that externally-defined OTDs are read-only, and cannot be edited. You can, however, test them to verify correctness of the build.*

Figure 64 OTD Editor



Major features of the OTD Editor interface are:

- **Reference**
  This area contains internal and external templates for the OTD file.
- **Object Type Definition**
  This area displays each field and element included in the OTD file.
- **Properties**
  This area displays details about the OTD file or field selected in the *Object Type Definition* list.

Table 24 OTD Editor Toolbar Icons

Icon	Command	Function
	Save as New Name in Repository	Saves current OTD under a new name in the Repository.
	Tester	Displays/refreshes the Tester area.
	Toggle Reference Tab Panel	Displays/hides the Reference area.

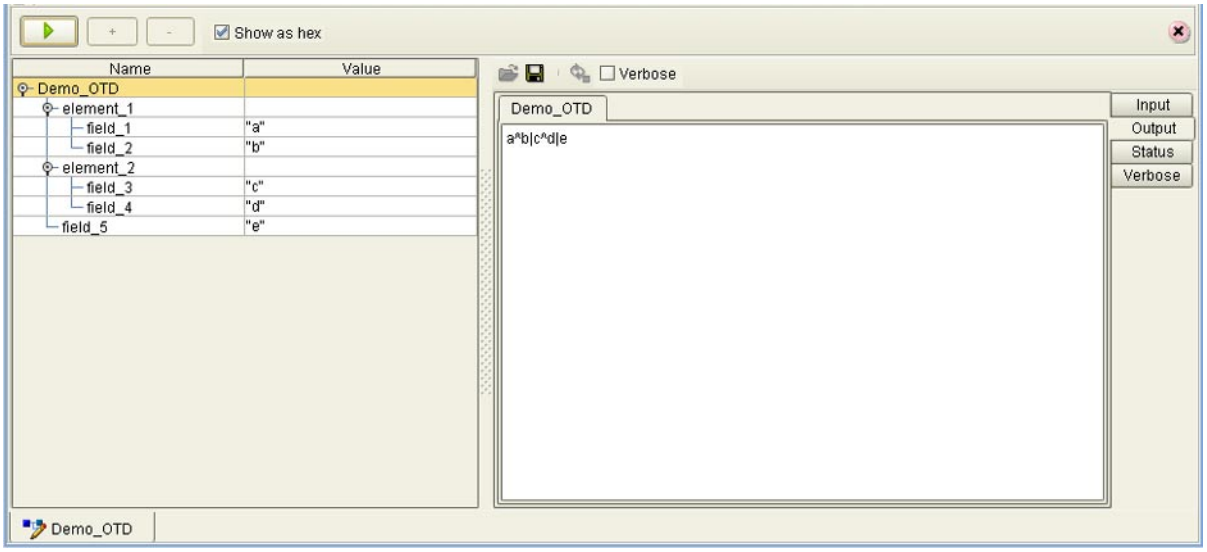
6.3.1 OTD Tester

The OTD tester provides a facility to verify the correctness of OTDs, for example to:

- Prevent data errors at runtime.
- Verify that all required data elements are available.
- Verify that all used data formats are correct.

Clicking the Tester icon (see Table 24) in the OTD Editor toolbar displays the OTD Tester as the lower part of the editor, as shown in Figure 65.

Figure 65 OTD Tester



The data display panel on the right has four data display modes, selectable by tabs. The **Input** tab is selected by default. Use of the OTD Tester is described in **Using the OTD Tester** on page 93.

Table 25 OTD Tester Buttons

Button	Command	Function
	Run Tester	Runs the tester with the entered values.
	Add Instance	Adds an instance of the selected repeating node, after the selected instance. You can also select the length field for a repeating node, in which case the new node will be added as the first instance.
	Delete Instance	Deletes the selected instance of a repeating node.

Next to the OTD Tester buttons is a check box labeled **Show as hex**. When checked, the values for **byte[]** nodes are shown in hexadecimal format (default setting). When not checked, the values are shown as regular alphanumeric text.

Table 26 OTD Tester Icons

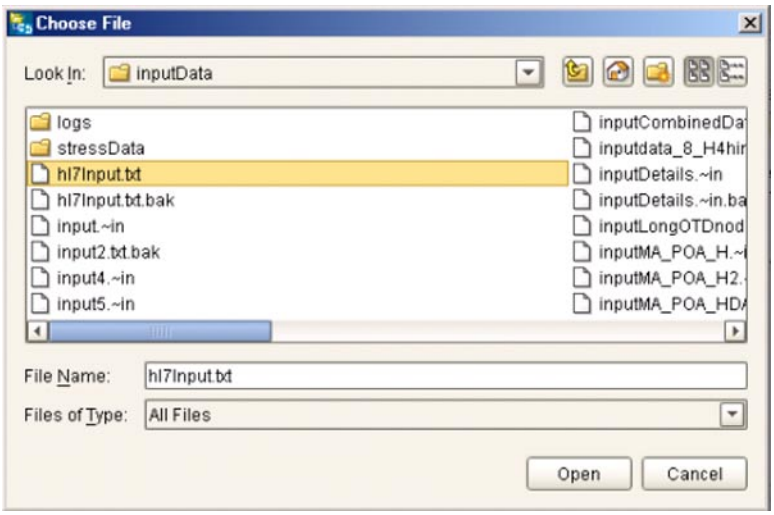
Icon	Command	Function
	Open	Displays file browser.
	Save	Saves displayed file.
	Refresh	Repopulates the OTD object elements with the values from the data display panel.

6.3.2 Using the OTD Tester

To use the OTD tester

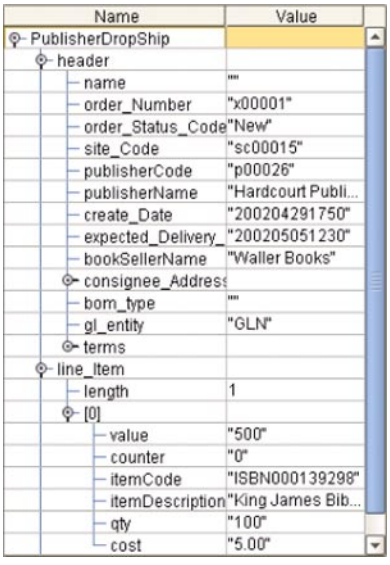
1. Open or create an OTD.
2. Display the OTD Tester
3. Provide the input test data either by selecting a data file (see Figure 66), or by entering the data manually.

Figure 66 Select Data File



4. Click the **Run Tester** button to test the selected OTD.
5. Verify the output by checking the values for each element for correctness (see Figure 67).

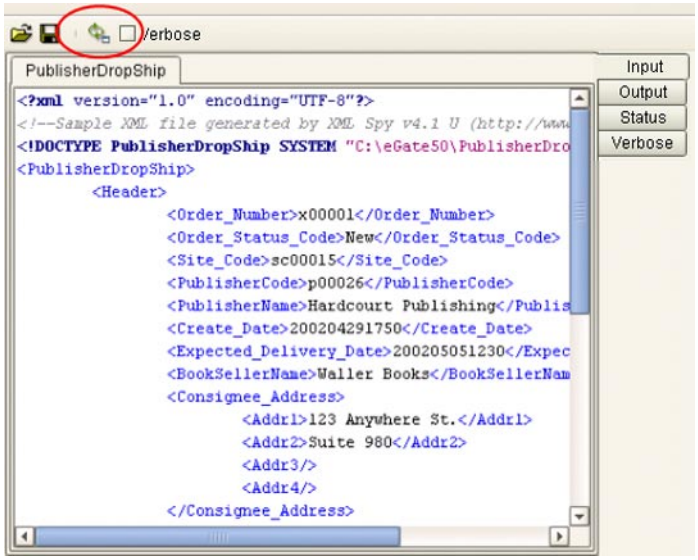
Figure 67 Object Elements and Values



6 Save your input test data to a file for re-use by selecting the **Input** data display and clicking the **Save** icon.

7 You can also change your test data in the Input data display, then re-test the OTD by clicking the **Refresh** icon (see Figure 68) to repopulate your OTD object elements with the new values.

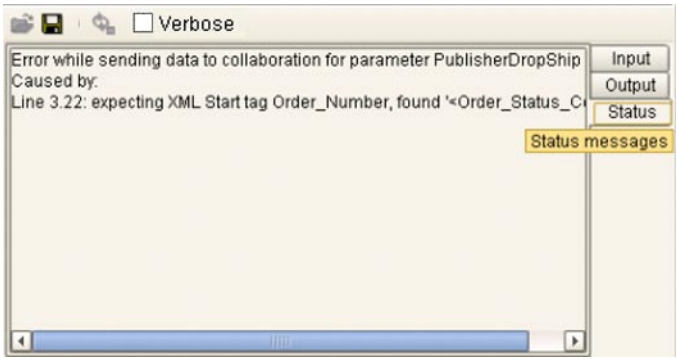
Figure 68 Data Display: Refresh Icon



Note: *Data, which you can edit for testing, is shown in black type.*

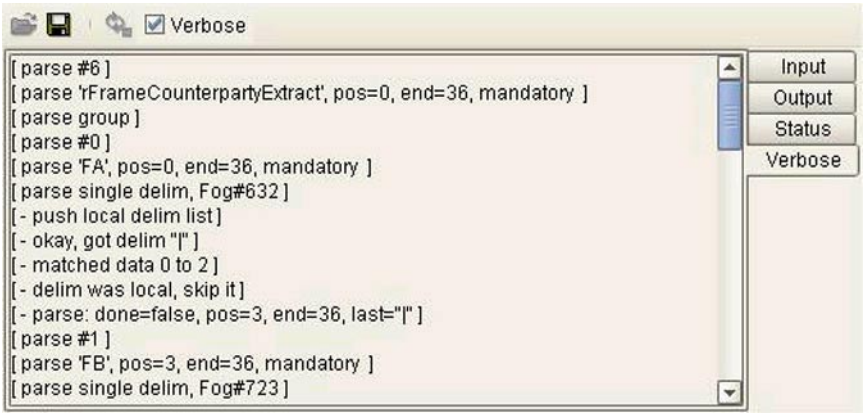
8 If there are errors in your input data, the **Status** data display is automatically invoked, showing the appropriate error messages (see Figure 69).

Figure 69 Status Data Display



9 For selected OTDs, the **Verbose** option provides a trace of parsing actions during the *unmarshal* process to aid in debugging the OTD structure. When this option is available, the OTD will activate the **Verbose** check box. Selecting the **Verbose** check box causes parsing information to appear on the **Verbose** data display (see Figure 70). The format and content of the data display are OTD-specific.

Figure 70 Verbose Data Display



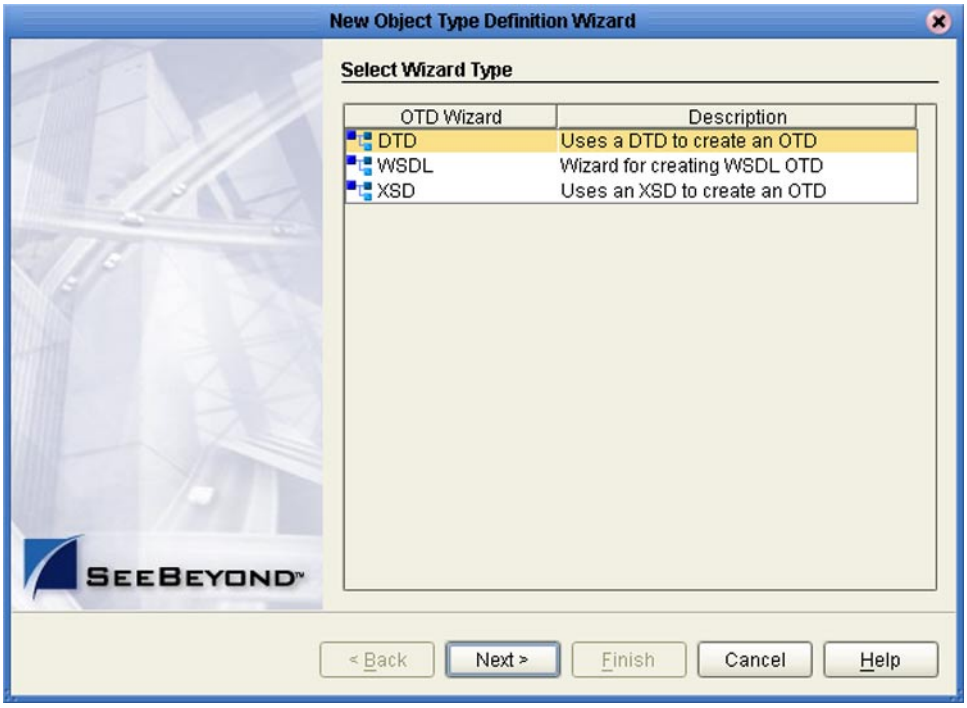
6.4 Creating Externally-Defined OTDs

6.4.1 Using the DTD Wizard

To create an OTD file from a DTD file

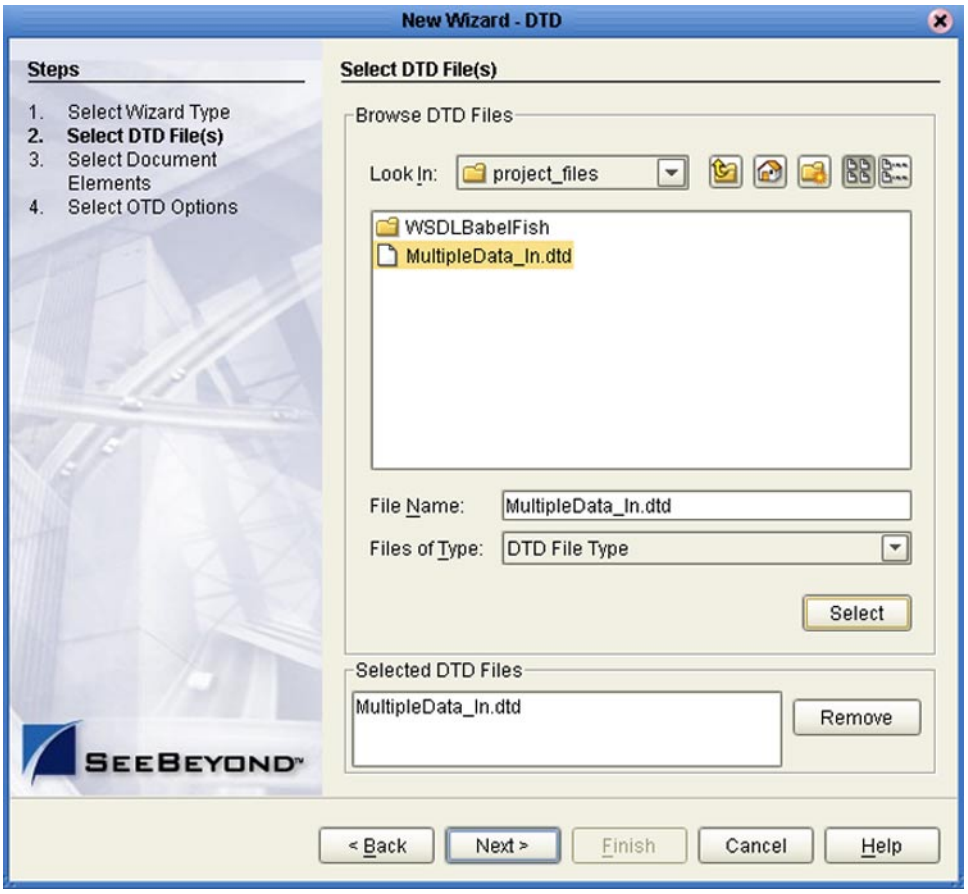
1 In the *Select Wizard Type* dialog, select **DTD** from the *OTD Wizard* list (see Figure 71) to create an OTD file from a Document Type Definition (DTD) file.

Figure 71 OTD Wizard Selection: DTD Wizard



2 Click **Next** to display the *Select DTD File(s)* dialog box, shown in Figure 72.

Figure 72 Select DTD File(s) Dialog Box



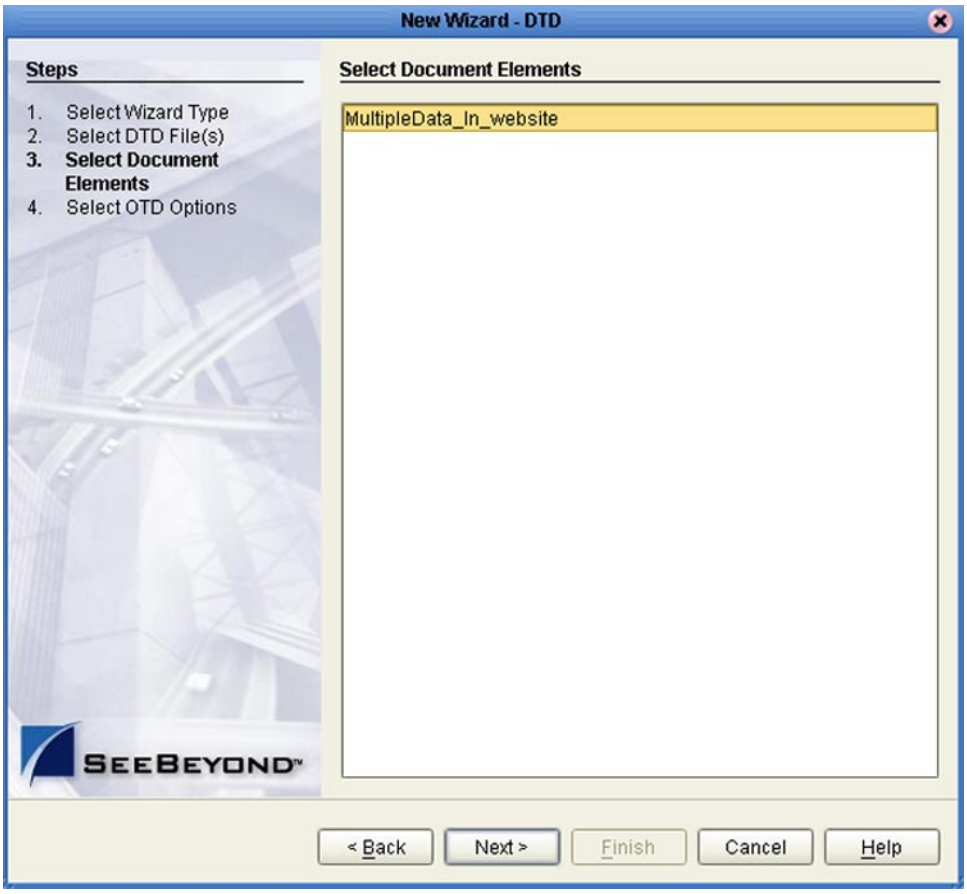
3 In the *Look In* drop-down list, navigate to the DTD file or files that you want to use to create the OTD. Click **Select** to add the files to the *List of Selected DTDs*.

- If the DTD file does not contain all information required for building an OTD (such as element definitions) a warning box such as that shown in Figure 82 will be displayed.

Figure 73 Cannot Create OTD Warning Box

4 Click **Next** to display the *Select Document Elements* dialog box, shown in Figure 74.

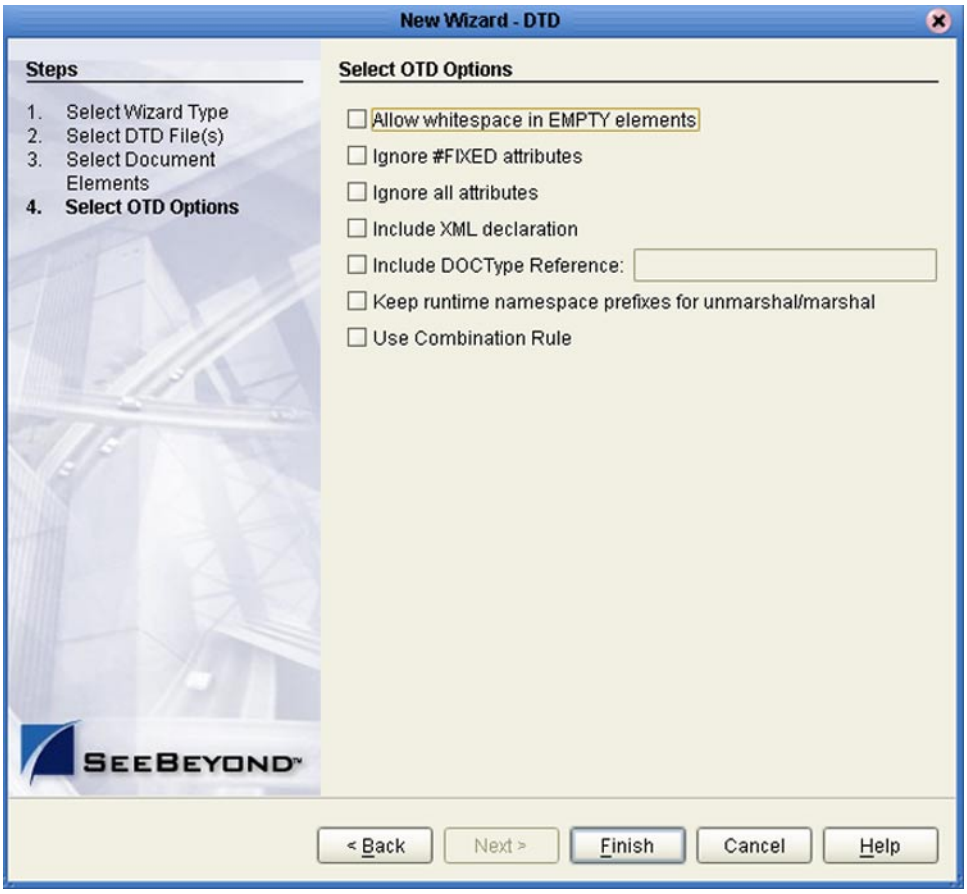
Figure 74 Select Document Elements Dialog Box



5 Select the elements of the document that you want to include in the OTD.

6 Click **Next** to display the *Select OTD Options* dialog box, shown in Figure 75.

Figure 75 Select OTD Options Dialog Box



7 Select the check boxes next to the OTD options you want to enable (see Table 27).

Table 27 DTD OTD Options

Option	Description
Allow whitespace in EMPTY elements	If an element is defined as EMPTY, this option controls whether or not white spaces are allowed within the element in the XML instance document forming the DTD.
Ignore #FIXED attributes	This option controls whether or not attributes defined as FIXED are ignored during the unmarshal and marshal processes. ▪ If this option is *not* selected, the attribute is recognized and saved into the OTD's runtime structure during the unmarshal process, and also appears in the output during the marshal process. ▪ If this option *is* selected, the attribute is ignored and neither of the above occurs.

Option	Description
Ignore all attributes	This option controls whether or not all attributes are ignored during the unmarshal and marshal processes. If both this option and the *Keep runtime namespace prefixes ...* option (below) are selected, only namespace attributes will be handled during the unmarshal process and consequently presented in the output during the marshal process. (The *namespace* attribute has the form **xmlns:XX**.)
Include XML declaration	This option controls whether or not the XML declaration **<?xml version="1.0" encoding="....."?>** appears in the output during the marshal process.
Include DOC Type Reference	This option controls whether or not the **"<!DOCTYPE ..."** string appears in the output during the marshal process.
Keep runtime namespace prefixes for unmarshal/ marshal	This option controls whether or not the namespace prefixes used during the marshal process are identical to those used in the unmarshal process. ▪ If this option is selected, all namespace attributes will be preserved once they appear in the XML instance document, and the namespace prefixes used in the marshal process will be exactly as they were presented in the XML document during the unmarshal process. ▪ If this option is *not* selected, then the namespace prefixes used in the marshal process might be different than the ones presented in the XML document during the unmarshal process (for example, the namespace prefixes that are presented in the XSD file might be used). **Note:** A consequence of selecting this option is that if there is no unmarshal process performed before the marshal process, then there will be no namespace attributes presented in the output (see the comment for the option below).
Use Combination Rule	Not currently used.

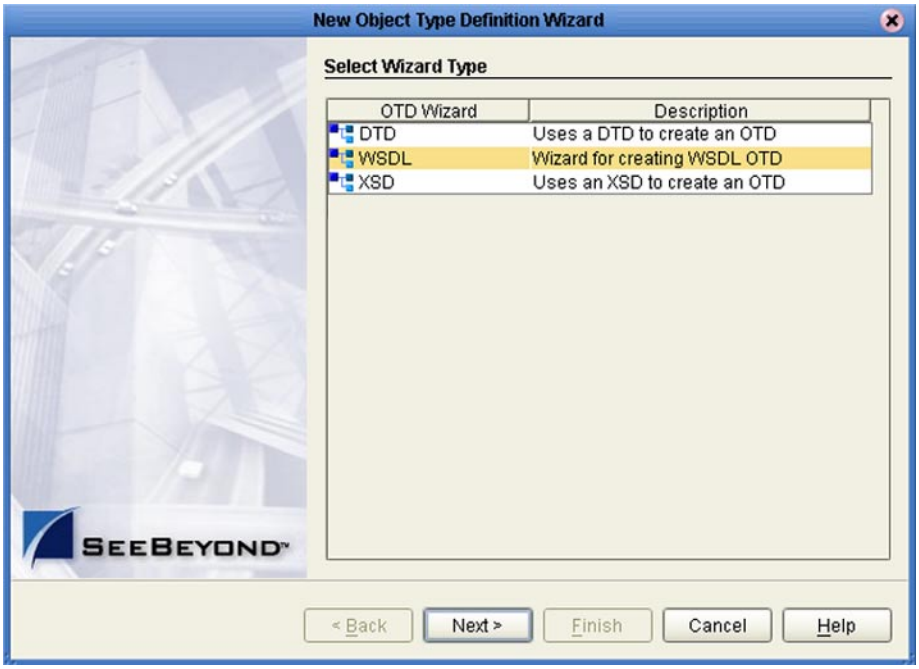
8 Click **Finish** to add the OTD to your Project and invoke the OTD Editor.

6.4.2 Using the WSDL Wizard

To create an OTD file from a WSDL file

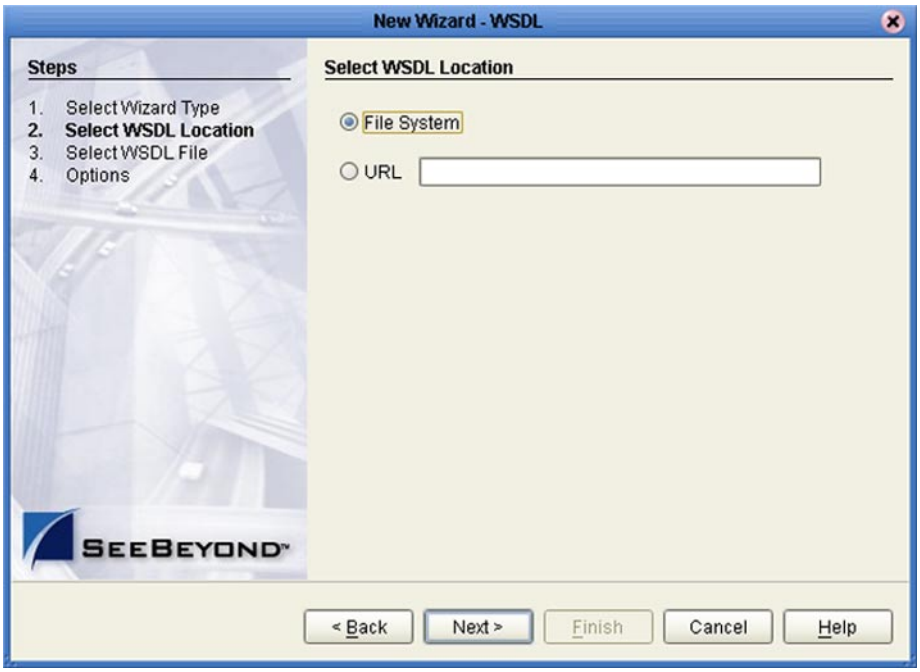
1 In the *Select Wizard Type* dialog, select **WSDL** from the *OTD Wizard* list (see Figure 76) to create an OTD file from an WSFL file.

Figure 76 OTD Wizard Selection: WSDL Wizard



2 Click **Next** to display the *Select WSDL File Location* dialog, shown in Figure 77

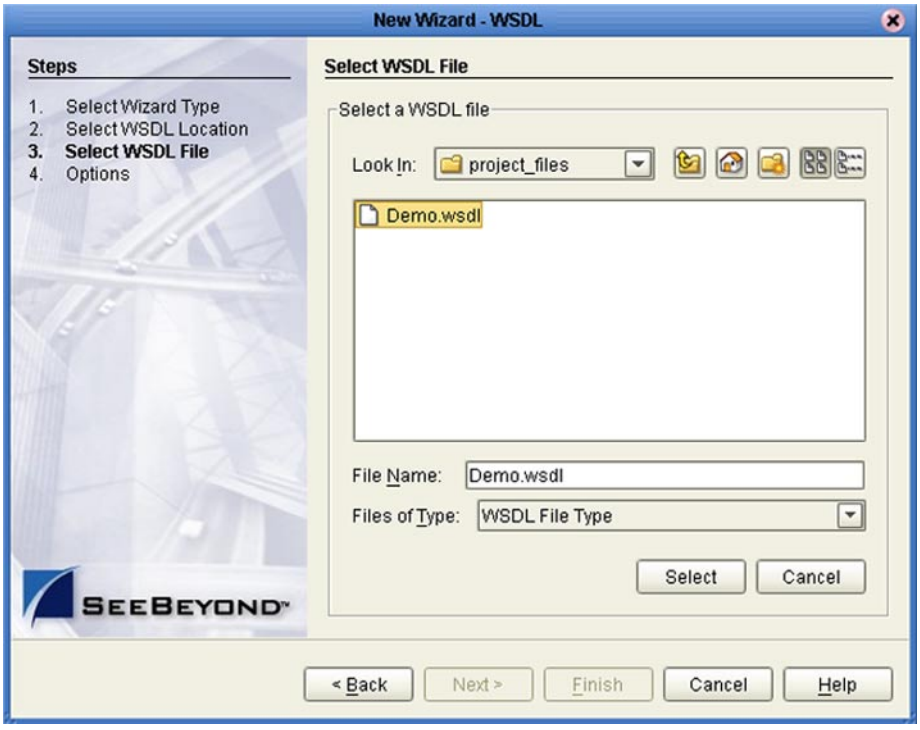
Figure 77 WSDL Wizard: Select WSDL Location



3 In the *Select WSDL Location* dialog, select **File System** or enter a **URL**, depending upon where your WSDL file is located.

4 Click **Next** to display the *Select WSDL File* dialog, shown in Figure 78.

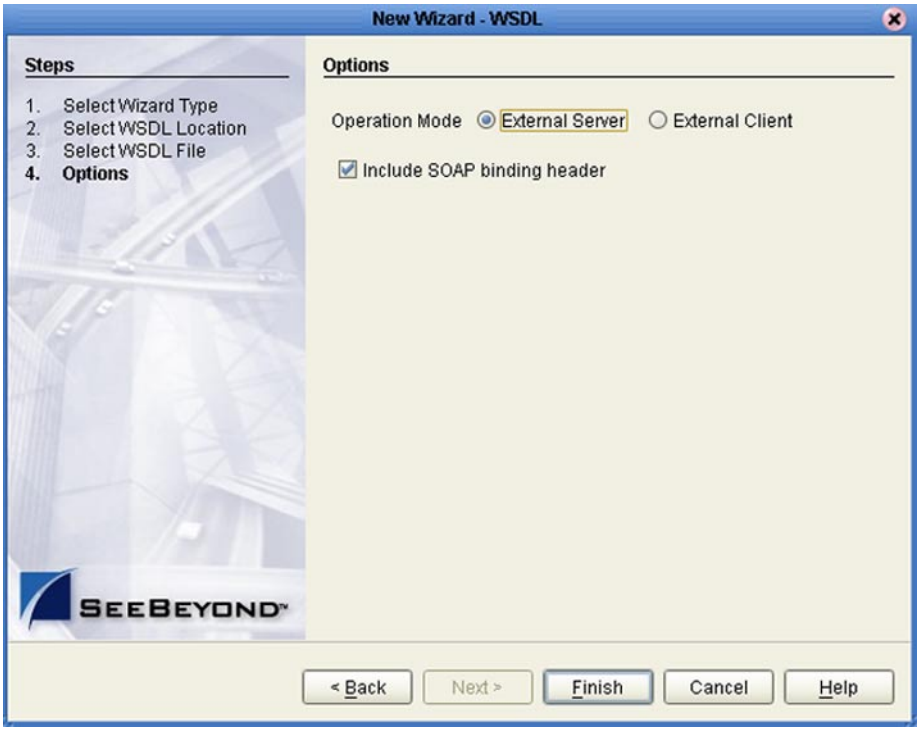
Figure 78 WSDL Wizard: Select WSDL File



5 In the *Look In* drop-down list, navigate to the WSDL file or files that you want to use to create the OTD. Click **Select** to add the files to the *List of Selected WSDLs*.

6 Click **Next** to display the *Options* dialog, shown in Figure 79.

Figure 79 WSDL Wizard: Select OTD Options



7 Select the check boxes next to the OTD options you want to enable:

- If you are using a Web client, select **External Server**.
- If you are using a Web server, select **External Client**.
- To include the SOAP binding header in the WSDL file, select the check box.

8 Click **Finish** to add the OTD to your Project and invoke the OTD Editor.

WSDL OTD Structure

The WSDL OTD has the following basic structure:

Root Node
PortType_XXX
Operation_XXX
Input_XXX
Output_XXX
PortType_XXX
Operation_XXX
Input_XXX
Output_XXX (and so on)

Where **XXX** is the name for each element given in the original WSDL file.

WSDL Operation Elements

To tie your messages together as a request-response pair corresponding to a method call, you must define operations using the WSDL **<operation>** element. A WSDL operation specifies which message is the *input* and which message is the *output*.

Inside the WSDL file's **<operation>** element**,** you specify your **<input>** and **<output>** elements. Each element refers to the corresponding message by its fully qualified name. The collection of all WSDL operations (that is, methods) exposed by your service is called a **portType** and is defined using the WSDL **<portType>** element.

The **<operation>** element is a child of **<portType>**. You can name the **<portType>** whatever you want. The port type **name** attribute provides a unique name among all the PortTypes defined within the enclosing WSDL file. Each WSDL operation is named via the **name** attribute.

Each operation within a WSDL OTD (like its WSDL file counterpart) uses one of the following operation modes for communication:

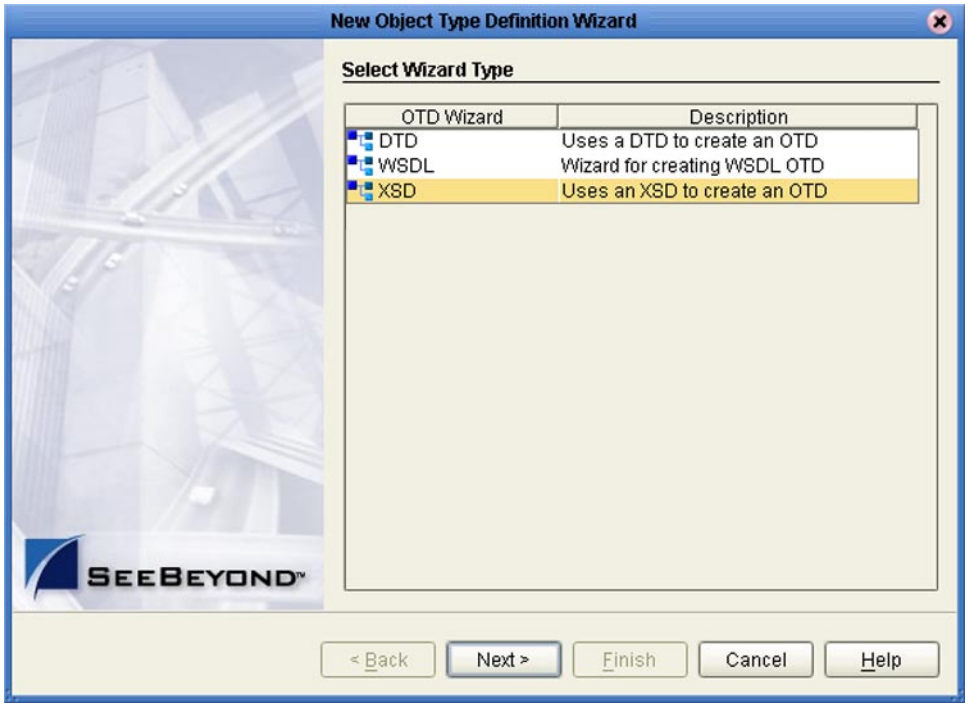
- **One-way:** The server receives a message from the client; also referred to as "fire and forget."
- **Request-response:** The server receives a message from the client and sends a correlated message back

6.4.3 Using the XSD Wizard

To create an OTD file from an XSD file

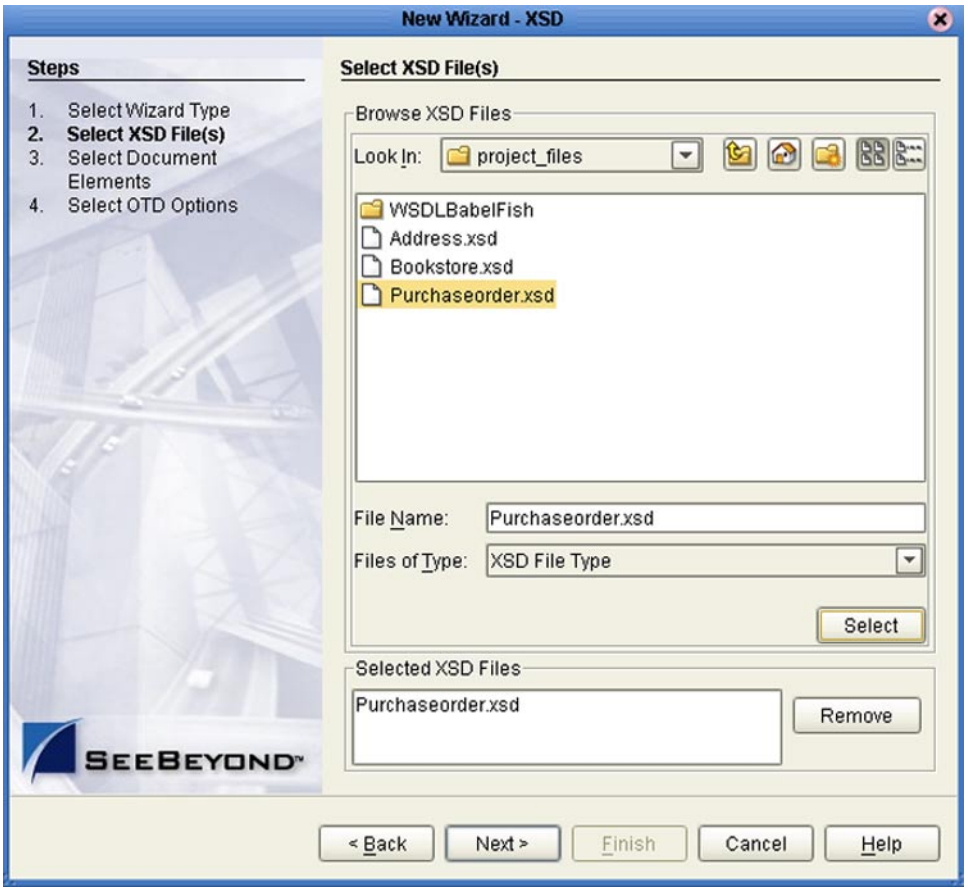
1 In the *Select Wizard Type* dialog, select **XSD** from the *OTD Wizard* list (see Figure 80) to create an OTD file from an XSD file.

Figure 80 OTD Wizard Selection: XSD Wizard



2 Click **Next** to display the Select XSD File(s) dialog box, shown in Figure 81.

Figure 81 XSD Wizard: Select XSD File(s)



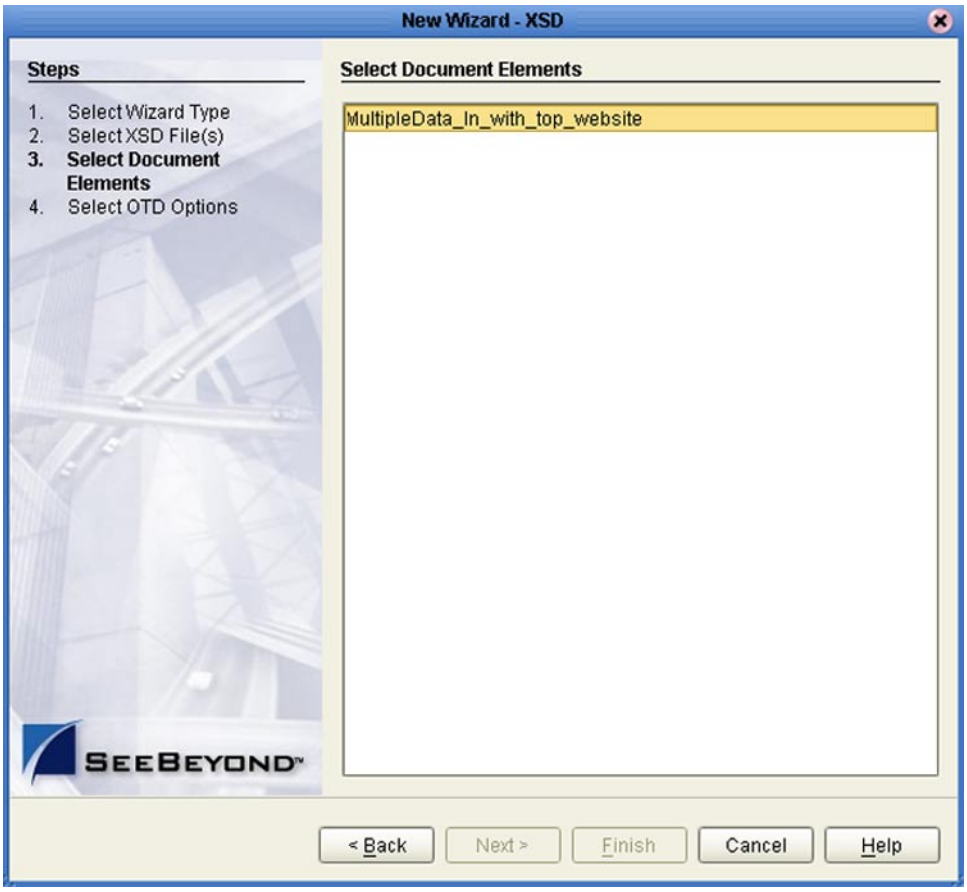
3 In the *Look In* drop-down list, navigate to the XSD file or files that you want to use to create the OTD. Click **Select** to add the files to the *List of Selected XSDs*.

- If the XSD file does not contain all information required for building an OTD (such as element definitions) a warning box such as that shown in Figure 82 will be displayed.

Figure 82 Cannot Create OTD Warning Box

4 Click **Next** to display the *Select Document Elements* dialog box, shown in Figure 83.

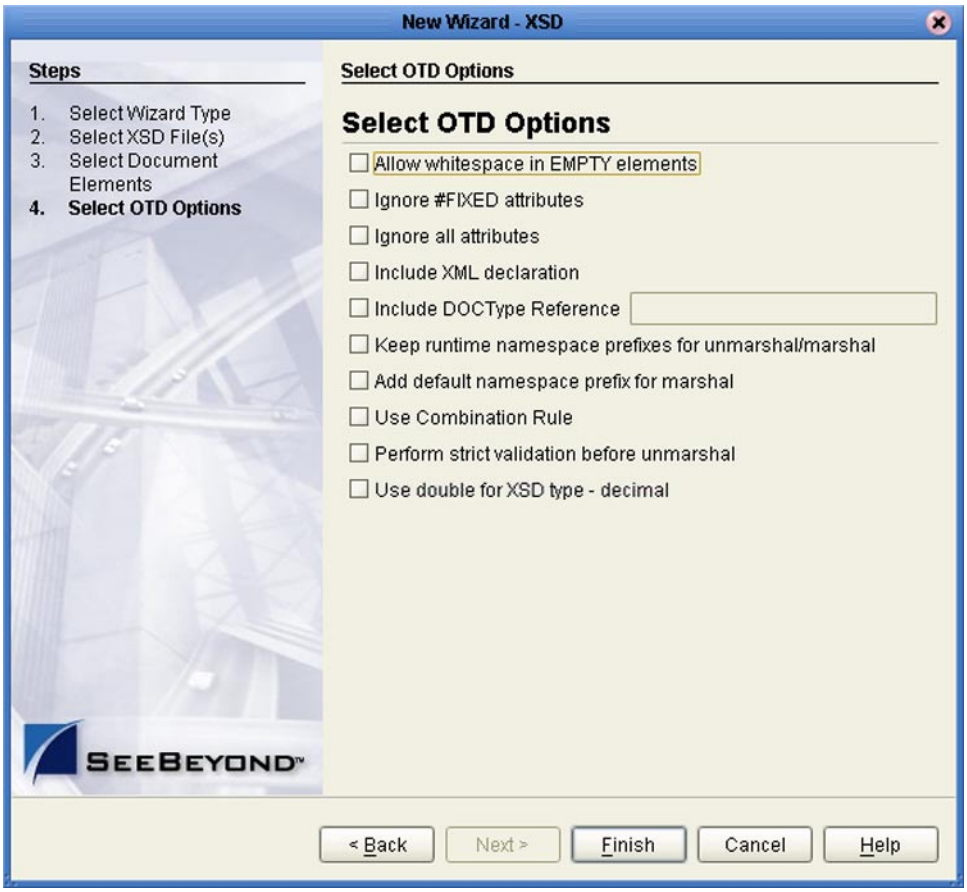
Figure 83 Select Document Elements Dialog Box



5 Select the elements of the document that you want to include in the OTD.

6 Click **Next** to display the *Select OTD Options* dialog box, shown in Figure 84.

Figure 84 Select OTD Options Dialog Box



7 Select the check boxes next to the OTD options you want to enable (see Table 28).

Table 28 XSD OTD Options

Option	Description
Allow whitespace in EMPTY elements	Not currently used for XSD OTDs.
Ignore #FIXED attributes	This option controls whether or not attributes defined as FIXED are ignored during the unmarshal and marshal processes. ▪ If this option is *not* selected, the attribute is recognized and saved into the OTD's runtime structure during the unmarshal process, and also appears in the output during the marshal process. ▪ If this option *is* selected, the attribute is ignored and neither of the above occurs.

Option	Description
Ignore all attributes	This option controls whether or not all attributes should be ignored in the unmarshal and marshal processes. If both this option and the *Keep runtime namespace prefixes ...* option (below) are selected, only namespace attributes will be handled during the unmarshal process and consequently presented in the output during the marshal process. (The *namespace* attribute has the form **xmlns:XX**.)
Include XML declaration	This option controls whether or not the XML declaration **<?xml version="1.0" encoding="....."?>** appears in the output during the marshal process.
Include DOC Type Reference	Not currently used for XSD OTDs.
Keep runtime namespace prefixes for unmarshal/ marshal	This option controls whether or not the namespace prefixes used during the marshal process are identical to those used in the unmarshal process. ▪ If this option is selected, all namespace attributes will be preserved once they appear in the XML instance document, and the namespace prefixes used in the marshal process will be exactly as they were presented in the XML document during the unmarshal process. ▪ If this option is *not* selected, then the namespace prefixes used in the marshal process might be different than the ones presented in the XML document during the unmarshal process (for example, the namespace prefixes that are presented in the XSD file might be used). **Note:** A consequence of selecting this option is that if there is no unmarshal process performed before the marshal process, then there will be no namespace attributes presented in the output (see the comment for the option below).
Add default namespace prefix for marshal	This option controls whether or not the prefix of the default target namespace of an element is applied to the element during the marshal process. ▪ If both this flag and the *Keep runtime namespace prefixes ...* option (above) are selected, then the default target namespace of an element will be applied to the element during the marshal process, *if it is a root element*. ▪ If the *Keep runtime namespace prefixes ...* option is *not* selected, then the elements are qualified based on the XSD definition and this flag has no effect.
Use Combination Rule	Not currently used.
Perform strict validation before unmarshal	Not currently used.
Use double for XSD type - decimal	If this option is checked, Enterprise Designer will use Java double type for the decimal type in the OTD. If not checked (default) Java BigDecimal type is used, which can handle big decimals with high precision.

8 Click **Finish** to add the OTD to your Project and invoke the OTD Editor.

Chapter 7

Environments

This chapter describes the process of defining eGate Environments, and the various components of an Environment.

7.1 Overview

Projects are run within *Logical Hosts*, which contain the logical resources required by the Project at run time. The Logical Hosts, in turn, are defined within *Environments*, which represent the physical resources required to implement the Project. The Environment also contains information about external systems with which the eGate Project interacts.

7.1.1 Environment Components

The components found in a typical run-time Environment are described in the following sections of this chapter:

- **Logical Hosts** on page 121
- **Integration Servers** on page 125
- **Message Servers** on page 143

7.2 Environment Explorer

The **Environment Explorer** displays the contents of the Repository that belong to the selected Environment (see Figure 85).

Figure 85 Enterprise Explorer: Environment Explorer View

The Environment Explorer is used in conjunction with the Environment Editor to create and configure the components of a run-time Environment. Each component in the Environment Explorer has an icon to identify the component type (see **Environment Explorer Icons** on page 112). Right-clicking on a component displays a context menu for that component (see **Context Menus** on page 113), from which you can select appropriate actions.

7.2.1 Environment Explorer Icons

The icons described in Table 29 appear in the Environment Explorer.

Table 29 Environment Icons

Icon	Function
	Represents the **Repository**, which is the central ICAN database where all Project information is saved. Binary files required at run time are also stored here.
	Represents a run-time **Environment**, which contains Logical Hosts and information about external systems.
	Represents a **Logical Host**, which contains the various logical components and files that are required at run time.
	Represents an **Environmental Constant**, which you can use to automate eWay and message destination configuration changes.
	Represents a **Scheduler** component of an Environment, which you can use to set data transfer to occur at set intervals.
	Represents a **SeeBeyond Integration Server** or third-party **application server**, which manages the Business Process interactions of an eGate Project. The integration server is deployed to a Logical Host.
	Represents a **SeeBeyond JMS IQ Manager** or third-party **message server**, which is used to store and forward eGate system messages. The message server is deployed to a Logical Host.

7.2.2 Context Menus

Right-clicking on a component in the Environment Explorer displays a context menu for that component. Included here are descriptions of options for the following component context menus:

- **Repository Menu** on page 113
- **Environment Menu** on page 114
- **Logical Host Menu** on page 115

Repository Menu

Figure 86 Repository Menu

Table 30 Repository Menu Options

Option	Function
New Environment	Displays a dialog box with which you can create a new Environment.
Configure SNMP Agent	Displays a dialog box with which you can modify the SNMP agent properties.
Save Changes to Repository	Saves any changes you have made in the Environment Editor to the Repository.
Refresh All from Repository	Refreshes the Environment Explorer to display the current contents of the Repository. (Open editors are not refreshed.)

Environment Menu

Figure 87 Environment Menu

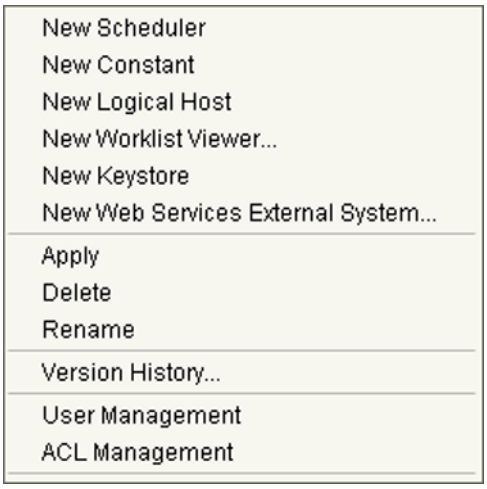


Table 31 Environment Menu Options

Option	Function
New Scheduler	Displays a dialog box with which you can add a new scheduling component to the selected Environment.
New Constant	Displays a dialog box with which you can add a constant to the selected Environment. See **Defining Environmental Constants** on page 119.
New Logical Host	Adds a new Logical Host to the selected Environment.
New Worklist Viewer	This option is present only when eInsight Business Process Manager is installed. See the *eInsight Business Process Manager User's Guide* for information.
New Keystore	Adds a new keystore to the selected Environment.
New Web Services External System	Adds a Web services external system to the selected Environment. See **Web Service Application** on page 79.
Apply	Applies changes to the selected Environment.
Delete	If you have *delete* privileges for the Environment (see ACL Management, below), a dialog box is displayed in which you confirm that you want to delete the selected Environment. Clicking **Yes** then deletes the Environment.
Rename	Activates the field, allowing you to rename the selected Environment.
Version History	Displays a dialog box with which you can track the version history for the selected Environment. See **Viewing a Component's Version History** on page 59 for more information.
User Management	Displays a dialog box with which an Administrator can manage message server access. See the *eGate Integrator System Administration Guide*.

Table 31 Environment Menu Options

Option	Function
ACL Management	Displays the **ACL Properties** dialog box, with which an Administrator can assign read/write/delete privileges to users for the selected Environment. See the *eGate Integrator System Administration Guide*.

Logical Host Menu

Figure 88 Logical Host Menu

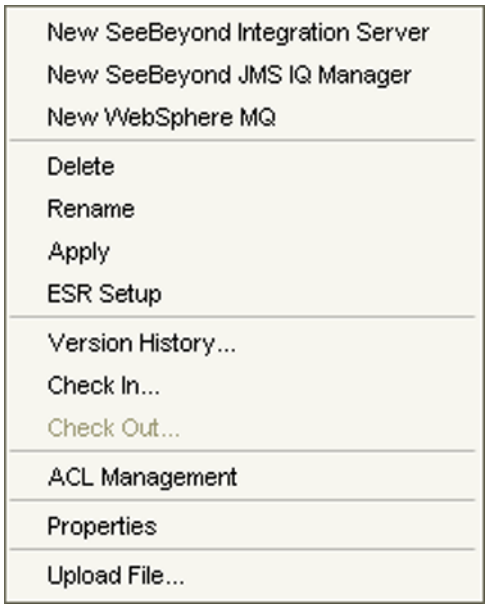


Table 32 Logical Host Menu Options

Option	Function
New SeeBeyond Integration Server	Adds a new SeeBeyond Integration Server to the selected Logical Host.
New SeeBeyond JMS IQ Manager	Adds a new SeeBeyond JMS IQ Manager to the selected Logical Host. See the *eGate Integrator JMS Reference Guide* for details.
New WebSphere MQ	Adds a new IBM WebSphere MQ message server to the selected Logical Host. See the *eGate Integrator JMS Reference Guide* for details.
Delete	Deletes the selected Logical Host, subject to the following conditions: ▪ You have *delete* privileges for the Logical Host (see ACL Management, below). ▪ The Logical Host is not checked out by anyone other than yourself. If these conditions are true, a dialog box is displayed in which you confirm that you want to delete the selected Logical Host. Clicking **Yes** then deletes the Logical Host.
Rename	Activates the field, allowing you to rename the selected Logical Host.

Table 32 Logical Host Menu Options

Option	Function
Apply	Applies changes to the selected Logical Host.
ESR Setup	Displays a dialog box with which an Administrator can select emergency software releases (ESRs) to add to the Logical Host.
Version History	Displays a dialog box with which you can track the version history for Logical Hosts. See **Viewing a Component's Version History** on page 59 for more information.
Check In	Displays a dialog box, with which you can check in a new version of an Logical Host. Refer to **Checking a Component In** on page 60 for more details.
Check Out	Displays a dialog box with which you can check out the current version of an Logical Host. See **Checking a Component Out** on page 62 for more information.
ACL Management	Displays the **ACL Properties** dialog box, with which an Administrator can assign read and/or write privileges to users for the selected Logical Host See the *eGate Integrator System Administration Guide*.
Properties	Displays a dialog box with which you can modify the configuration properties for the selected Logical Host.
Upload File	Allows you to upload third-party libraries (**.jar** files) to the Logical Host.

Note: *If you are using BEA WebLogic and/or IBM WebSphere, the Application Servers and JMS Message Servers for these products will also appear in the context menu (see Figure 89).*

Figure 89 Logical Host Menu with Third-Party Servers

Integration Server

The menu shown in Figure 90 is used for both the SeeBeyond Integration Server and third-party application servers. The configuration properties, however, are different.

Figure 90 Integration Server Menu

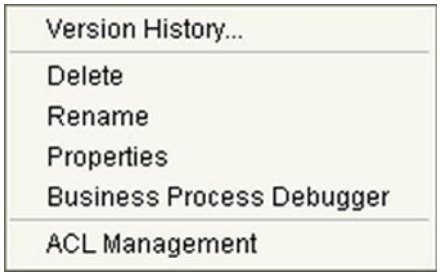


Table 33 Integration Server Menu Options

Option	Function
Version History	Displays a dialog box with which you can track the version history for the selected integration server. See **Viewing a Component's Version History** on page 59 for more information.
Delete	Deletes the selected integration server, subject to the following conditions: ▪ You have *delete* privileges for the integration server (see ACL Management, below). ▪ The integration server is not checked out by anyone other than yourself. If these conditions are true, a dialog box is displayed in which you confirm that you want to delete the selected integration server. Clicking **Yes** then deletes the integration server.
Rename	Activates the field, allowing you to rename the selected integration server.
Properties	Displays a dialog box with which you can modify the configuration properties for the selected integration server.
Business Process Debugger	Displays the Business Process Debugger. (This option appears in the menu only if eInsight Business Process Manager is installed in your ICAN Suite). See the *eInsight Business Process Manager User's Guide* for information.
ACL Management	Displays the **ACL Properties** dialog box, with which an administrator can assign read/write/delete privileges to users for the selected integration server. See the *eGate Integrator System Administration Guide*.

SeeBeyond JMS IQ Manager

The menu shown in Figure 91 is used for both the SeeBeyond JMS IQ Manager and third-party message servers. The configuration properties, however, are different.

Figure 91 JMS IQ Manager Menu

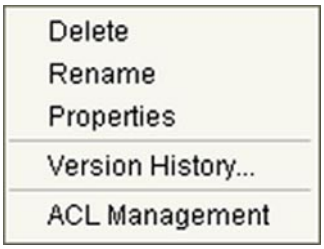


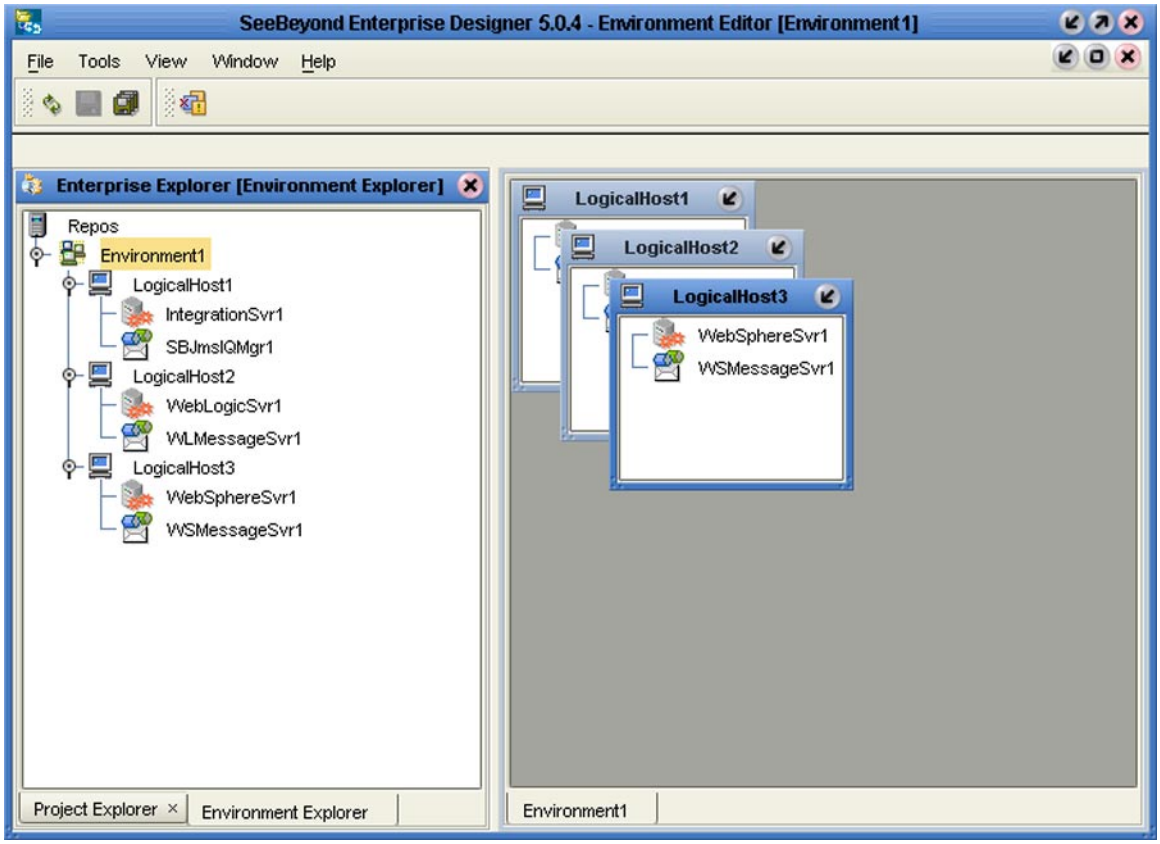
Table 34 Integration Server Menu Options

Option	Function
Delete	Deletes the selected message server, subject to the following conditions: ▪ You have *delete* privileges for the message server (see ACL Management, below). ▪ The message server is not checked out by anyone other than yourself. If these conditions are true, a dialog box is displayed in which you confirm that you want to delete the selected message server. Clicking **Yes** then deletes the message server.
Rename	Activates the field, allowing you to rename the selected message server.
Properties	Displays a dialog box with which you can modify the configuration properties for the selected message server.
Version History	Displays a dialog box with which you can track the version history for the selected message server. See **Viewing a Component's Version History** on page 59 for more information.
ACL Management	Displays the **ACL Properties** dialog box, with which an administrator can assign read/write/delete privileges to users for the selected message server. See the *eGate Integrator System Administration Guide*.

7.3 Environment Editor

Clicking an Environment icon in the Environment Explorer invokes the Environment Editor, which provides a canvas in which you can create and customize an Environment (see Figure 92).

Figure 92 Environment Editor

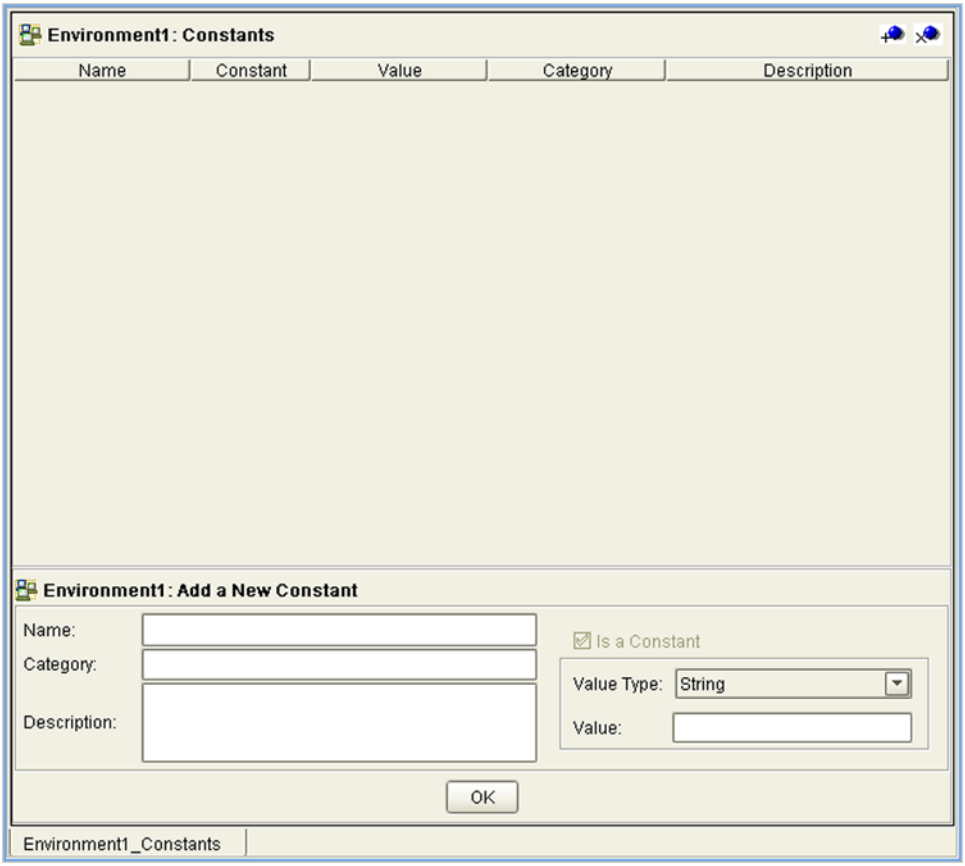


Here you can see the various components (Logical Hosts, servers, and external systems) included in the selected Environment. New Environments are added through the use of the Repository context menu (see **Repository Menu** on page 113). Components are added to the Environment by selecting options in the Environment and Logical Host context menus (see **Environment Menu** on page 114 and **Logical Host Menu** on page 115, respectively).

7.3.1 Defining Environmental Constants

Environmental constants are name/value pairs that are visible across the Environment. Selecting the **New Constant** option from the Environment context menu displays the Constants panel in the Environment Editor (see Figure 93).

Figure 93 Environmental Constants Panel



All constants defined for the specific Environment are listed in the *Constants* section of the panel, along with their various properties. New constants are added using the *Add a New Constant* section of the panel.

Note: *When you create an Environmental constant, you assign a permanent value to it—which cannot be overridden.*

Table 35 Environmental Constants Panel Icons

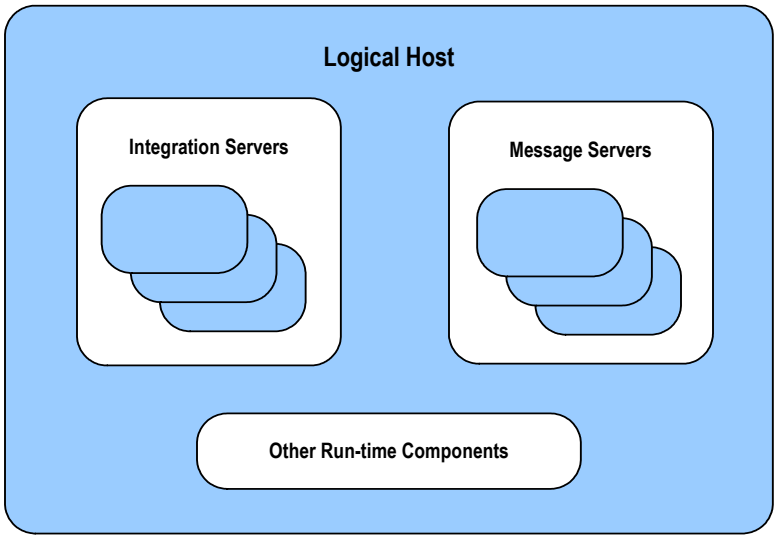
Icon	Name	Function
	Add a New Constant	Adds a new constant to the list.
	Delete a Highlighted Constant	Deletes the selected constant from the list.

7.4 Logical Hosts

7.4.1 Overview

A Logical Host contains the eGate run-time components that are installed on a host hardware platform. A Logical Host can be a member of only one Environment, but each Environment can contain multiple Logical Hosts. Both integration servers and message servers are deployed to the Logical Host, as illustrated in Figure 94.

Figure 94 Logical Hosts



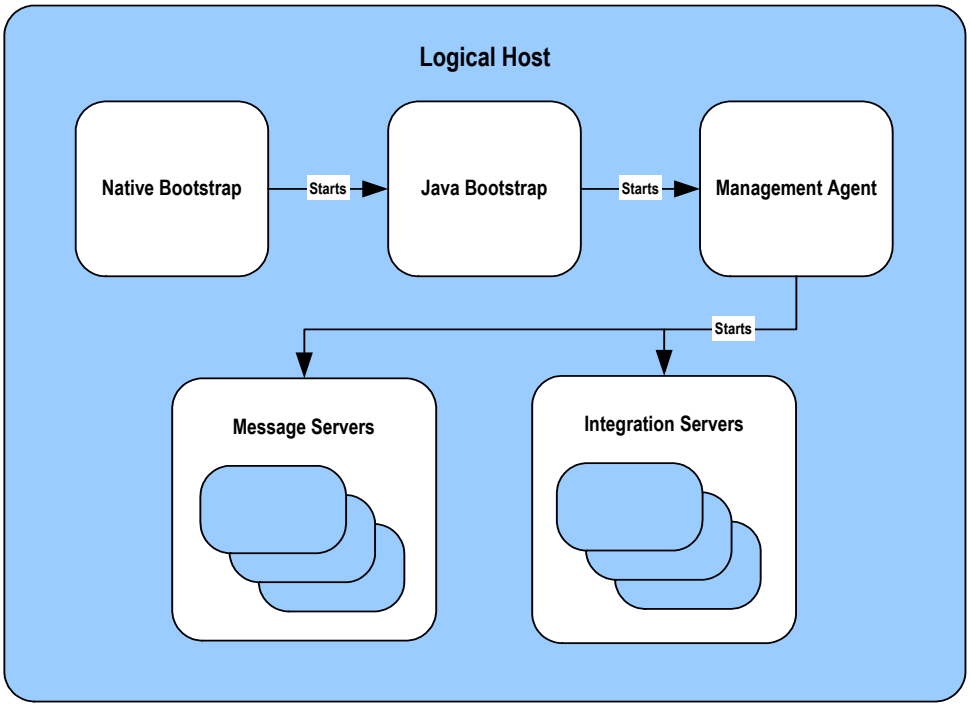
The master service of the Logical host is the Management Agent. This service starts the other services on the Logical Host as part of the bootstrap process. The Management Agent also communicates with the Enterprise Manager via JMX (Java Management Extensions) to report the status of the message servers and integration servers.

At run time, a platform-specific bootstrap script starts a Java bootstrap program. The first time the bootstrap runs, it downloads the Management Agent, message server, and integration server from the Repository. The Management Agent is then started, which in turn starts the message server(s) and integration server(s). Figure 95 illustrates this sequence.

Subsequent bootstraps simply start the existing Management Agent without downloading component configurations from the Repository, unless you force a download by:

- Selecting **Apply** from the Logical Host context menu (see **Logical Host Menu** on page 115).
- Using the command-line option (**-f**). See the *eGate Integrator System Administration Guide* for information.

Figure 95 Startup Sequence



Each Logical Host has a separate bootstrap process. The process is started from a batch file (***ICAN-root***\logicalhost\bootstrap\bin\bootstrap.bat) or script (***ICAN-root***/logicalhost/bootstrap/bin/bootstrap.sh). This file or script finds the Repository via command-line parameters or from the configuration file (***ICAN-root***\logicalhost\bootstrap\config\logical-host.properties). See the *eGate Integrator System Administration Guide* for additional information.

7.4.2 Configuring a Logical Host

To access the configuration properties for a Logical Host

1. Right-click a Logical Host in the Environment Explorer tree to display the context menu for that Logical Host instance.
2. Select **Properties** from the context menu to display the **Properties** dialog box.
3. Select the **Logical Host Configuration** node in the properties tree to display the Logical Host Configuration Section, which contains the top-level configuration properties for the Logical Host (see Figure 96).

Figure 96 Logical Host Configuration Properties

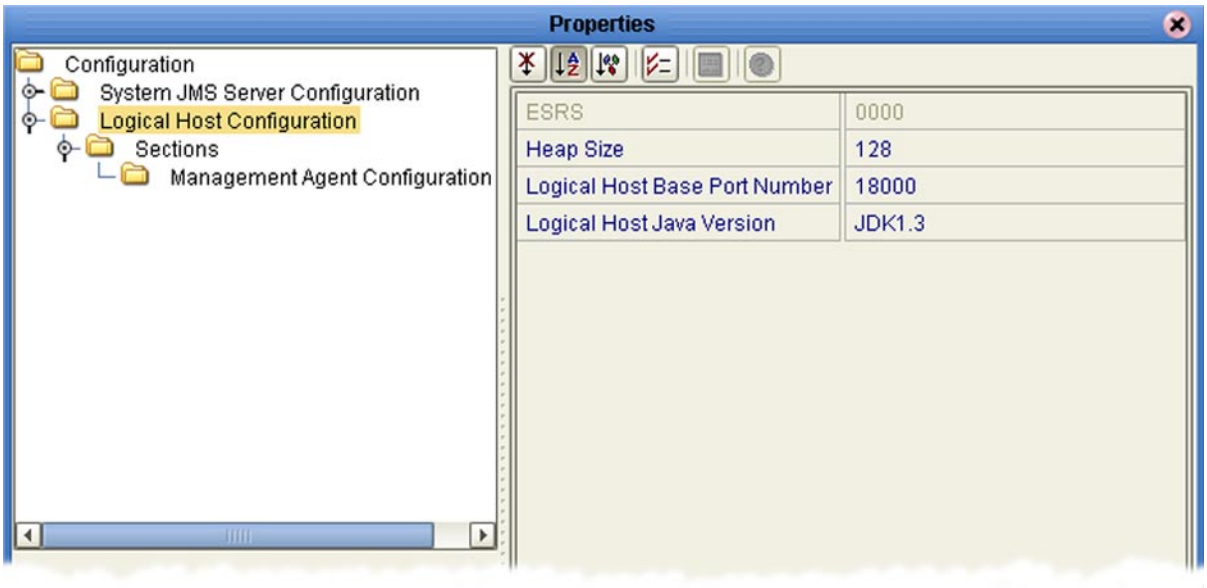


Table 36 Logical Host Configuration Properties List

Property	Description
ESRS	Shows a list of all installed Emergency Software Releases (ESRs).
Heap Size	Specifies the Heap size in Megabytes; the minimum size is **128** Mb, which is the default value. Note that this property is only for the bootstrap and management processes, and does not affect the integration server or any runtime components that are processing data.
Logical Host Base Port Number	Specifies the base port number for the Logical Host. The default value is **18000**. When multiple Logical Hosts reside on a single hardware platform, you must configure the base port numbers; see the following section.
Logical Host Java Version	Specifies the Java version being used to the eWay RAR file generation program, so that any generated file will be properly compatible. The default value is **JDK1.3**.

Configuring the Base Port Number

If multiple Logical Hosts concurrently run on the same computer, you must ensure that each Logical Host has a different base port number to avoid conflicts. This base port number is propagated throughout the Logical Host, so that the various components are automatically given successive port numbers following that assigned to the Logical Host itself.

The number of port numbers used in a Logical Host varies according to the specific implementation, so when assigning new base port numbers you need to skip successive

numbers by an adequate amount. The default base port number is 18000, so base port numbers of 19000, 20000, and so on are recommended.

If you need to assign a specific port number to a particular Logical Host component, the automatic numbering process will skip the component port number you have assigned manually (*be sure this port number is not used elsewhere*).

Note: *While Windows accepts port numbers below 12000, UNIX does not.*

Figure 97 Management Agent Configuration Properties

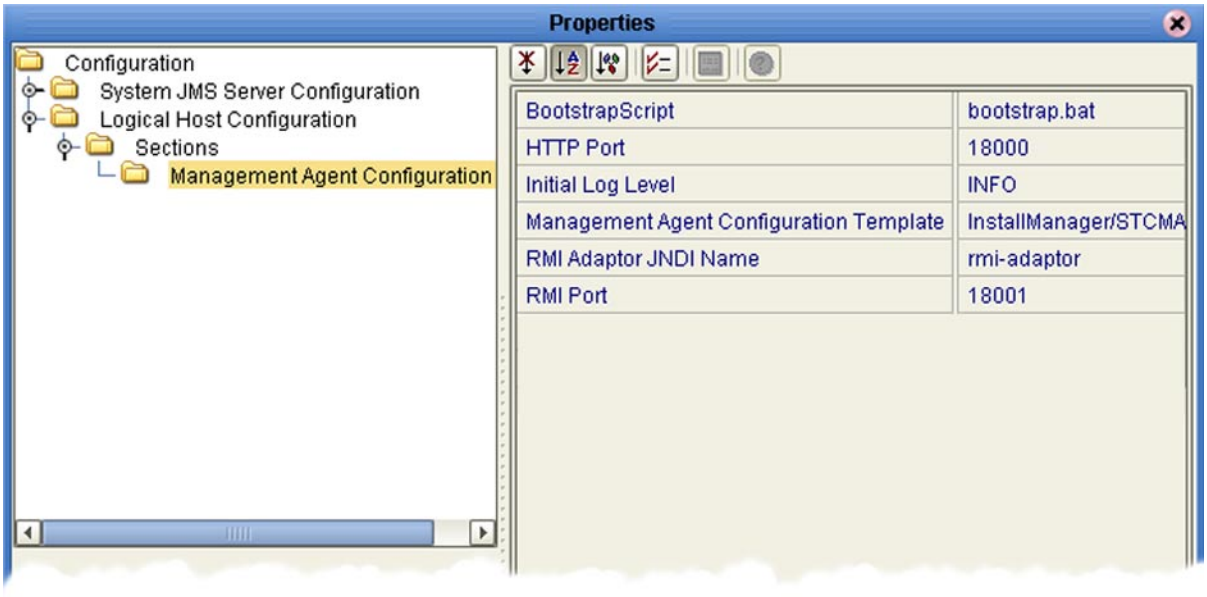


Table 37 Management Agent Configuration Properties List

Property	Description
BootstrapScript	Name of the bootstrap script; the default is **bootstrap.bat**.
HTTP Port	The HTTP port; the default value is **18008**.
Initial Log Level	The initial log level (OFF, ALL, DEBUG, INFO, WARN, ERROR, FATAL); the default value is **INFO**. See the *eGate Integrator System Administration Guide* for log level details.
Management Agent Configuration Template	The name and path of the configuration template; the default value is **InstallManager/STCMA/common/config/templates/ManagementAgent-config.vm**.
RMI Adaptor JNDI Name	The name of the RMI adapter; the default value is **rmi-adaptor**.
RMI Port	The RMI port; the default value is **18001**.

7.5 Integration Servers

The Logical Host contains one or more instances of a J2EE integration server, which is the engine that runs eInsight Business Process for processing business logic, and eWays that communicate with external applications. The integration server provides services for security, transactions, business rules execution, and connectivity management. eGate Integrator contains the SeeBeyond Integration Server, and also supports the use of third-party application servers such as BEA WebLogic and IBM WebSphere (see **Deploying Projects to Third-Party Servers** on page 163).

7.5.1 Configuring an Integration Server

To access the configuration properties for an integration server

1. Right-click an integration server in the Environment Explorer tree to display the context menu for that instance.
2. Select **Properties** from the context menu to display the **Properties** dialog box.
3. Select the **IS Configuration** node in the properties tree to display the top-level IS configuration properties (see Figure 98).

Figure 98 Top-level IS Configuration Properties

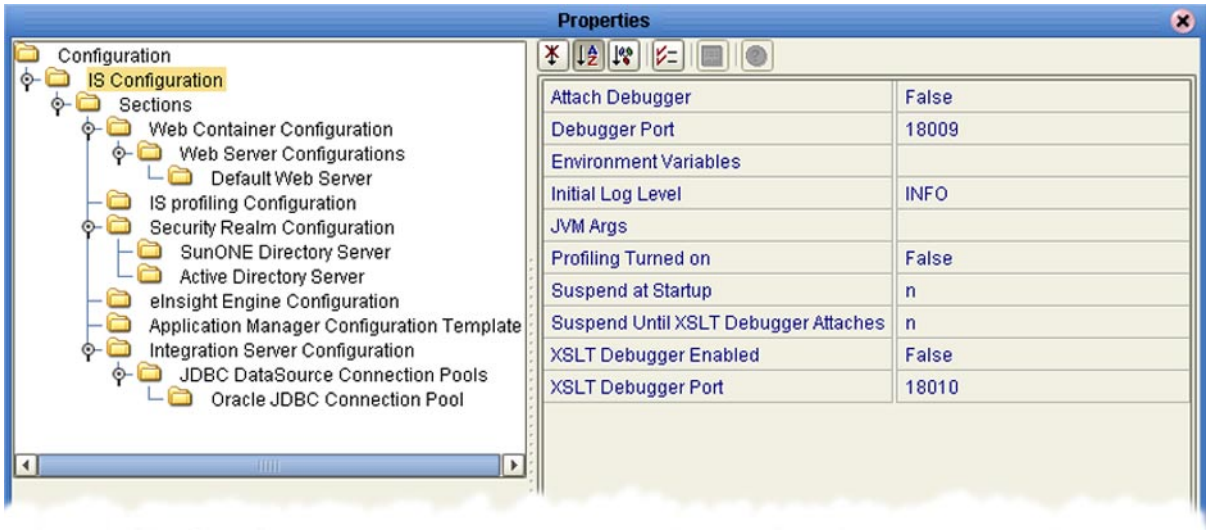


Table 38 Top-level IS Configuration Properties List

Property	Description
Attach Debugger	Enables/disables debugging for the IS. The default is **False** (disabled).
Debugger Port	This property is used only when the Debugger is enabled. The default value depends upon the number of the Logical Host base port.

Table 38 Top-level IS Configuration Properties List

Property	Description
Environment Variables	Specifies user-defined Environment Variables. Each element has the format name=value. When present, these values override the system settings, so that *all* required variables must be set. There is no default.
Initial Log Level	The initial log level (OFF, ALL, DEBUG, INFO, WARN, ERROR, FATAL); the default value is **INFO**. See the *eGate Integrator System Administration Guide* for log level details.
JVM Args	Java Virtual Machine (JVM) arguments. Each element in the collection should specify one, and only one, argument.
Profiling Turned On	Enables/disables performance monitoring for the IS. The default is **False** (disabled). To enable, change to **True** (as shown) and configure the properties described in **Performance Monitoring (Profiling)** on page 130.
Suspend at Startup	Allows the VM to begin executing before the debugger application attaches. The default is **n** (do not suspend).
Suspend Until XSLT Debugger Attaches	Allows the VM to begin executing before the XSLT debugger application attaches. The default is **n** (do not suspend).
XSLT Debugger Enabled	Enables/disables XSLT debugging for the IS. The default is **False** (disabled).
XSLT Debugger Port	This property is used only when the XSLT Debugger is enabled. The default depends upon the value of the Logical Host base port.

The IS Configuration node contains several sections, each containing detailed configuration properties for a particular IS component (including the integration server itself). You can also access these properties by selecting **Properties** from the context menus for the appropriate nodes.

Web Container

Properties included here are used for setting up Web services.

Figure 99 Web Container Configuration Properties

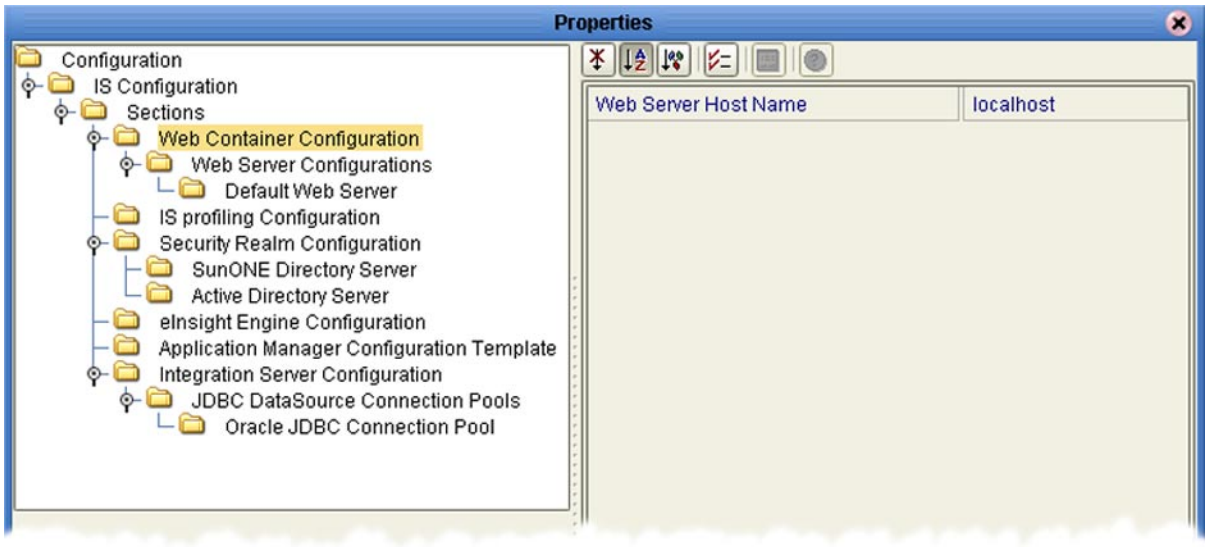


Table 39 Web Container Configuration Properties List

Property	Description
Web Server Host Name	Specifies the host name; the default is **localhost**.

Web Server Configurations

This directory serves as a container for the configuration properties for all Web servers associated with the selected integration server. You should create new sets of configuration properties for each server, and preserve the default set as a template.

To create a new set of Web server configuration properties

1. Right-click Web Server Configurations to display the context menu, and click **Create New Section**.
2. Give the newly created section an appropriate name.
3. In the properties for the newly created section, make any necessary changes to the default properties.
4. Click **OK**.

Default Web Server

The Default Web Server properties serve as a template for the individual Web servers.

Figure 100 Default Web Server Properties

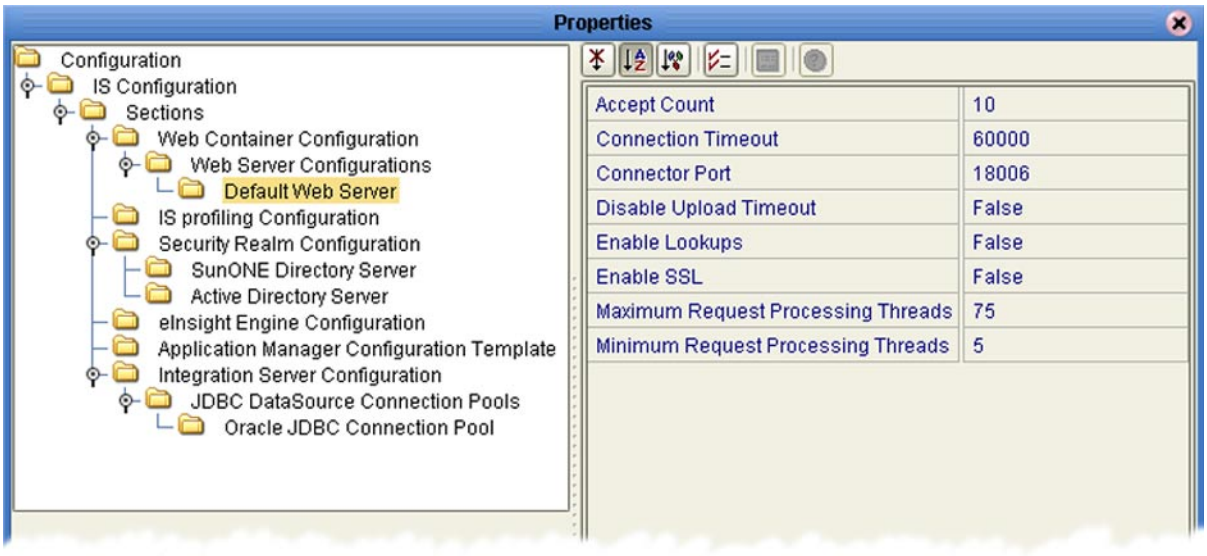


Table 40 Default Web Server Properties List

Property	Description
Accept Count	Specifies the maximum acceptable number of incoming connection requests when all possible request processing threads are in use. Any requests received beyond this number when the queue is full are refused. The default value is **10**.
Connection Timeout	Specifies the time period in milliseconds that this connector will wait for the request URI line to be presented, after accepting the connection. The default value is **60,000** ms (1 min).
Connector Port	Specifies the connection port for the Web server. The default value is **18006**.
Disable Upload Timeout	Allows the servlet container to use a different, and longer, connection timeout while a servlet is executing. This gives the servlet a longer time to complete execution, and/or provides a longer timeout during data upload. The default value is **False**.
Enable Lookups	If set to **True**, calls are made requesting getRemoteHost() to perform DNS lookups in order to return the actual host name of the remote client. If set to **False**, the DNS lookup is bypassed and the IP address is returned in string form, thereby improving performance. The default value is **False**.
Enable SSL	Specifies whether or not to enable the Secure Sockets Layer (SSL) protocol. The default value is **False**.

Table 40 Default Web Server Properties List

Property	Description
SSL Client Authentication Required	Displayed only if **Enable SSL** is set to **True**. Set this property to **True** if you want to require SSL Client Authentication. The default value is **False**.
SSL Keystore Password	Displayed only if **Enable SSL** is set to **True**. Enter your desired Keystore password (there is no default value).
Maximum Request Processing Threads	Specifies the maximum number of request processing threads to be created by this connector, thereby determining the maximum number of simultaneous requests that can be handled. The default value is **75**.
Minimum Request Processing Threads	Specifies the number of request processing threads to be created by this connector when it is first started. This value must be less than the value set for the Maximum Request Processing Threads property. The default value is **5**.

Performance Monitoring (Profiling)

You can monitor the performance of the integration server by using the built-in *Heap Analysis* tool, which is enabled and configured using the Profiling Configuration dialog box (see Figure 101).

Figure 101 Profiling Configuration Properties

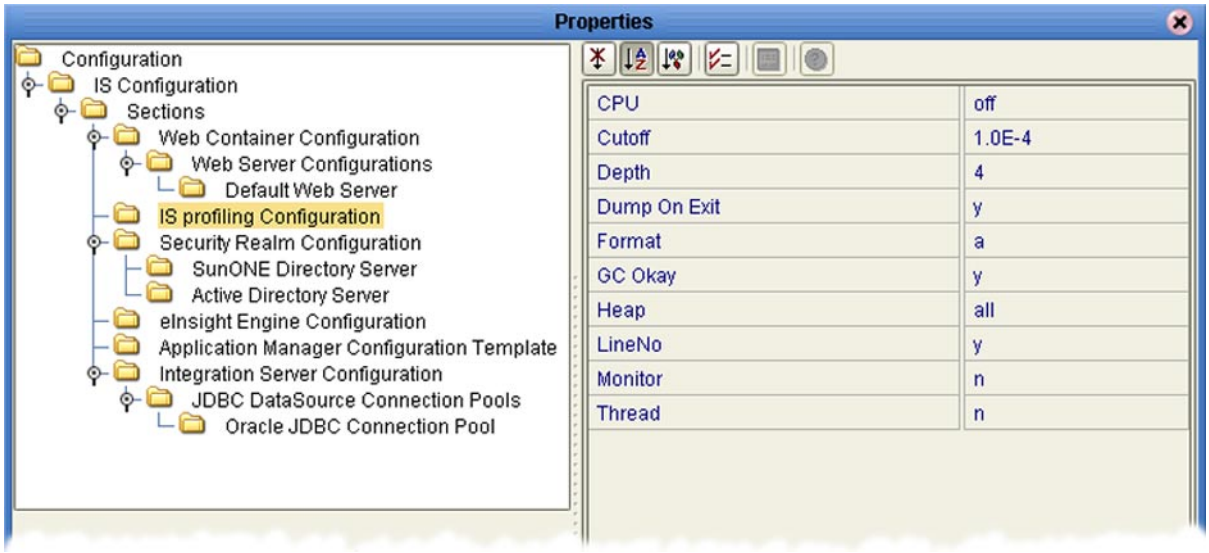


Table 41 Profiling Configuration Properties List

Property	Description
CPU	Specifies whether or not CPU usage is included in the trace. The default value is **off**.
Cutoff	Specifies the output cutoff point. The default value is **1.0E-4**.
Depth	Specifies the stack trace depth. The default value is **4**.
Dump on Exit	Specifies whether or not to dump on exit. The default value is **y** (yes).
Format	Specifies ASCII (**a**) or binary (**b**) output. The default value is **a** (ASCII).
GC Okay	Specified whether or not to allow garbage collection (GC) during sampling. The default value is **y** (yes).
Heap	Specifies the blocks of memory to include in traces. The default value is **all**.
LineNo	Specifies whether or not to include line numbers in traces. The default value is **y** (yes).
Monitor	Specifies whether or not to include monitor contention. The default value is **n** (no).
Thread	Specifies whether or not to include the thread in traces. The default value is **n** (no).

Security Realm

These properties pertain to the Lightweight Directory Access Protocol (LDAP), if used. Subdirectories contain properties for SunONE Directory Server and Microsoft Active Directory Server. See the *eGate Integrator System Administration Guide* for information regarding Security Realm configuration.

Figure 102 Security Configuration Properties

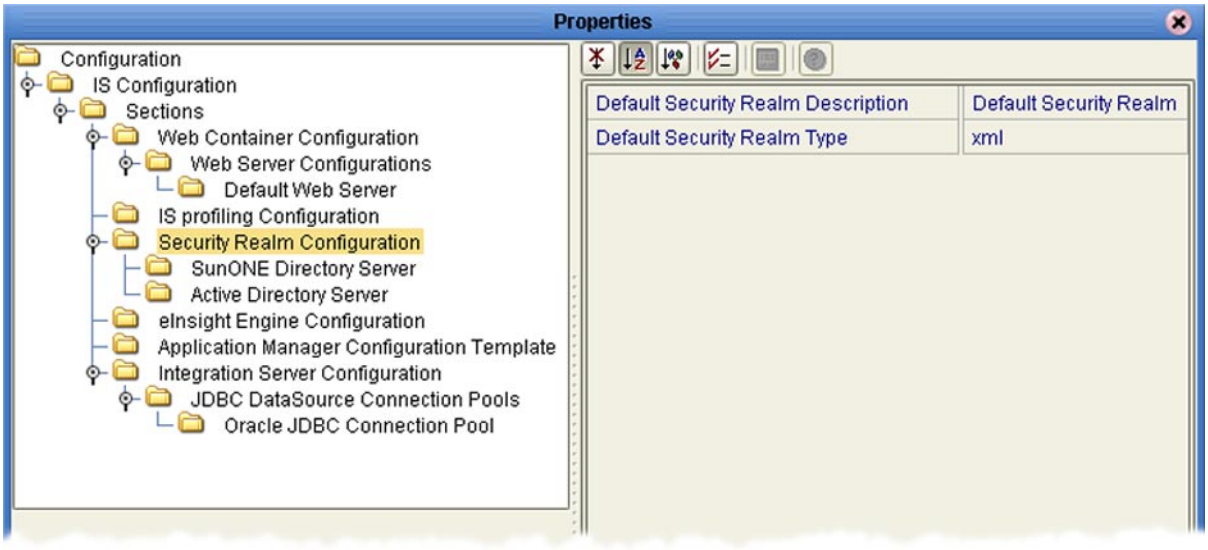


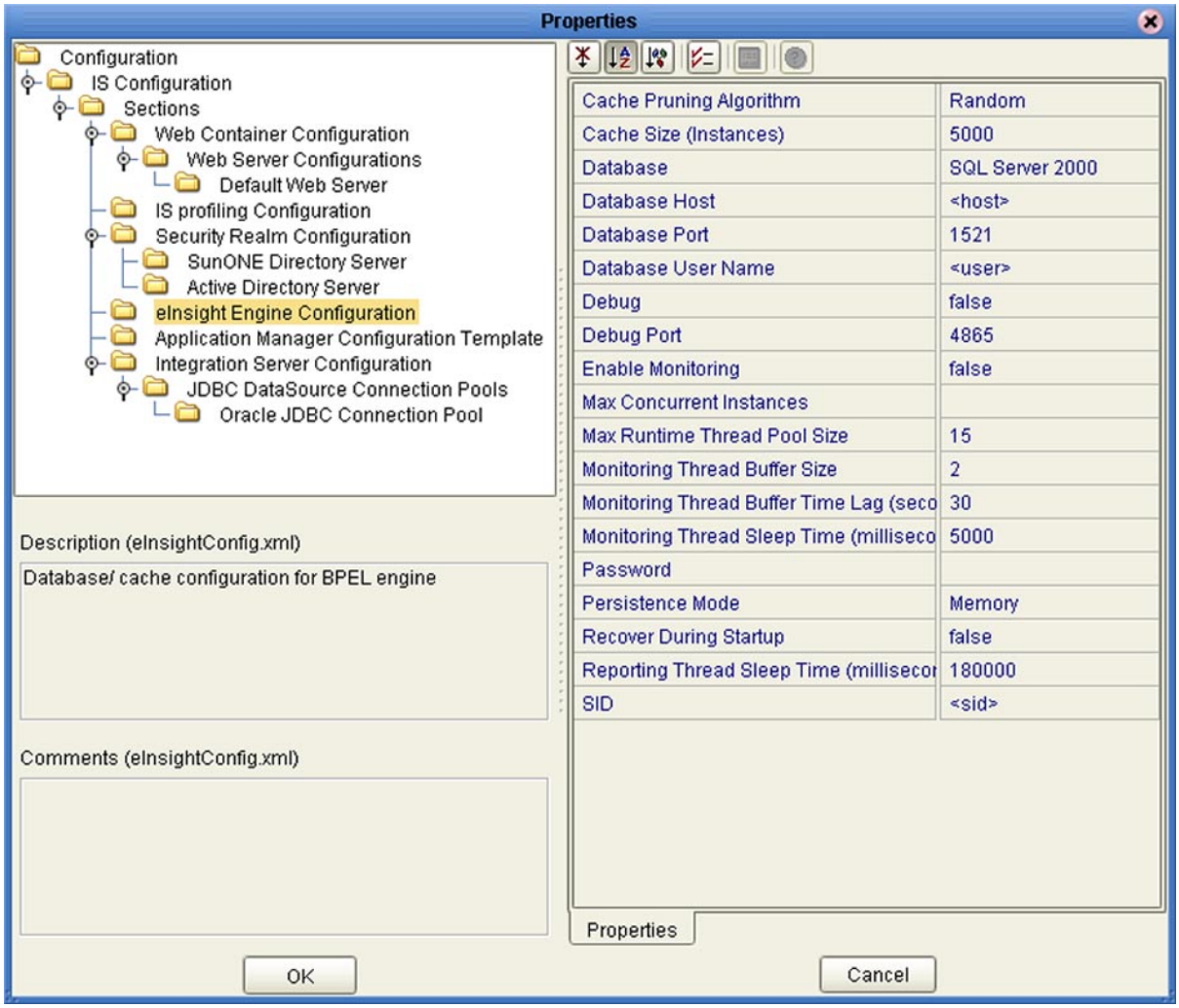
Table 42 Security Realm Configuration Properties List

Property	Description
Default Security Realm Description	Specifies the name for the default LDAP Security Realm. The default value is **Default Security Realm**.
Default Security Realm Type	Specifies the default LDAP Security Realm type. The default value is **xml**.

eInsight Engine

This configuration node is displayed only if you have eInsight Business Process Manager installed on your system. The configuration properties relate to the BPEL engine's database cache; see the *eInsight Business Process Manager User's Guide* for information regarding these properties (see Figure 103).

Figure 103 eInsight Engine Configuration Properties



Application Manager

You can set integration server thread pool variables using the Application Manager Configuration Properties dialog box (see Figure 104).

Figure 104 Application Manager Configuration Properties

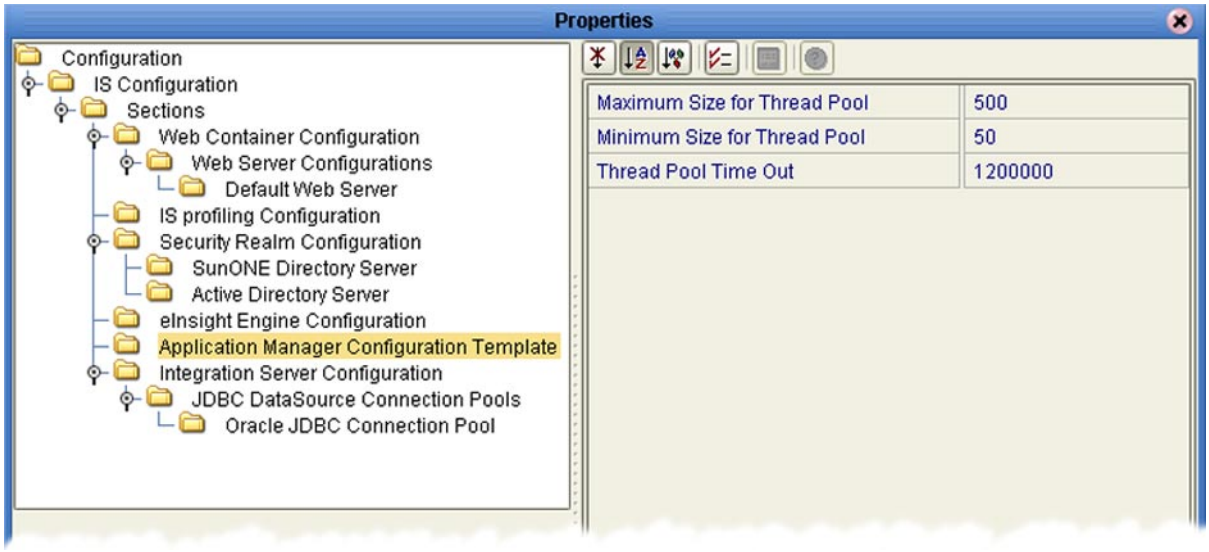


Table 43 Application Manager Configuration Properties List

Property	Description
Maximum Size for Thread Pool	Specifies the maximum size for the thread pool. The default value is **500**.
Minimum Size for Thread Pool	Specifies the minimum size for the thread pool. The default value is **50**.
Thread Pool Time Out	Specifies the timeout interval for the thread pool, measured in milliseconds. The default value is **1,200,000** ms (20 min).

Integration Server

Detailed, low-level configuration of the integration server is performed using the Integration Server Configuration Properties dialog box (see Figure 105).

Figure 105 Integration Server Configuration Properties

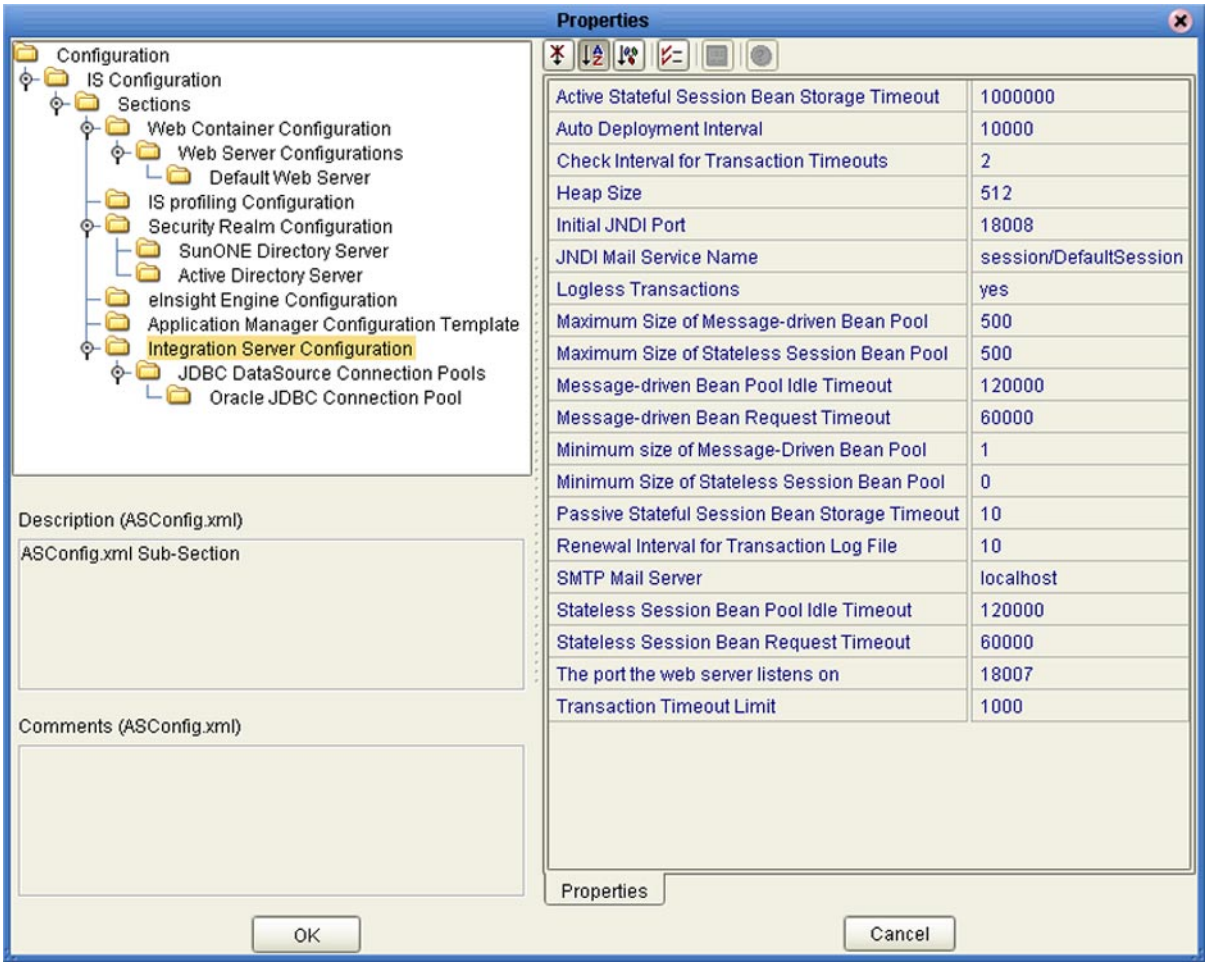


Table 44 Integration Server Configuration Properties List

Property	Description
Active Stateful Session Bean Storage Timeout	Specifies the interval after which an Active Stateful Session Bean is removed from storage, measured in minutes. The default value is **1,000,000** min, which ensures that it will not be removed unintentionally.
Auto Deployment Interval	Specifies the interval at which the auto-deployer checks the deployment directory for files, measured in milliseconds. The default value is **10,000** ms.
Check Interval for Transaction Timeouts	Specifies the interval between checks for transaction timeouts, measured in minutes. The default value is **2** min.

Table 44 Integration Server Configuration Properties List

Property	Description
Heap Size	Specifies the Heap size in Megabytes; the minimum size is **512** Mb, which is the default value. Increasing this value increases the JVM size.
Initial JNDI Port	Specifies the initial port required by the Naming Service class for startup. The default value depends upon the value of the Logical Host base port.
JNDI Mail Service Name	Specifies the name of the JNDI mail service. The default value is **session/DefaultSession**.
Logless Transactions	Specifies whether or not logless transactions are allowed. The default value is **yes**.
Maximum Size of Message-driven Bean Pool	Specifies the maximum number of Message-driven Beans allowed in the Message-driven Bean pool at one time. The default value is **500**.
Maximum Size of Stateless Session Bean Pool	Specifies the maximum number of Stateless Session Beans allowed in the Stateless Session Bean pool at one time. The default value is **500**.
Message-driven Bean Pool Idle Timeout	Specifies the timeout interval for the Message-driven Bean pool, measured in milliseconds. The default value is **120,000** ms (2 min).
Message-driven Bean Request Timeout	Specifies the interval after which a Message-driven Bean request times out, measured in milliseconds. The default value is **60,000** ms (1 min).
Minimum Size of Message-driven Bean Pool	Specifies the minimum number of Message-driven Beans allowed in the Message-driven Bean pool at one time. The default value is **1**.
Minimum Size of Stateless Session Bean Pool	Specifies the maximum number of Stateless Session Beans allowed in the Stateless Session Bean pool at one time. The default value is **1**.
Passive Stateful Session Bean Storage Timeout	Specifies the interval after which a Passive Stateful Session Bean is removed from storage, measured in minutes. The default value is **10** min.
Renewal Interval for Transaction Log File	Specifies the interval for renewing the Transaction Service log file, measured in hours. The default value is **10** hr.
SMTP Mail Server	Specifies the name of the SMTP mail host server. The default value is **localhost**.
Stateless Session Bean Pool Idle Timeout	Specifies the timeout interval for the Stateless Bean pool, measured in milliseconds. The default value is **120,000** ms (2 min).
Stateless Session Bean Pool Request Timeout	Specifies the interval after which a Stateless Bean request times out, measured in milliseconds. The default value is **60,000** ms (1 min).
The Port the Web Server Listens On	Specifies the port the Web server listens on. The default value depends upon the value of the Logical Host base port.

Table 44 Integration Server Configuration Properties List

Property	Description
Transaction Timeout Limit	Specifies the time limit for transactions to time out, measured in seconds. The default value is **1,000** sec.

JDBC DataSource Connection Pools

This directory serves as a container for the configuration properties for all JDBC DataSource Connection Pools associated with the selected integration server. The only set of default configuration properties currently furnished is for Oracle JDBC.

Oracle JDBC Connection Pool

Connection Pool properties for an Oracle database associated with the integration server are specified in the Oracle JDBC Connection Pool dialog box (see Figure 106).

Figure 106 Oracle JDBC Connection Pool Properties

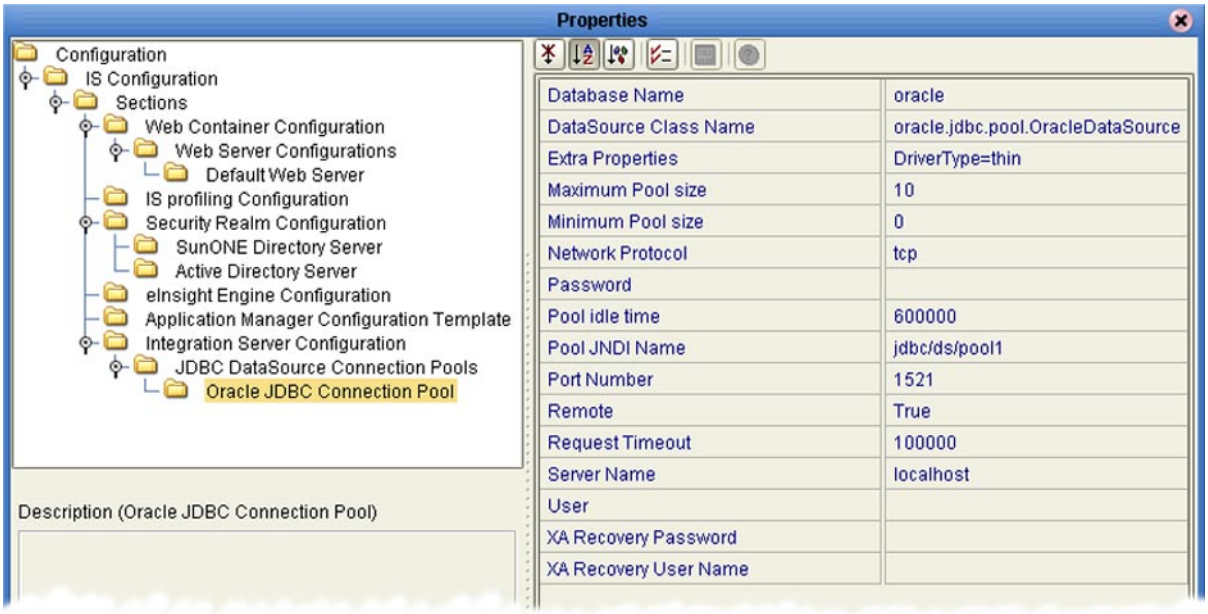


Table 45 Oracle JDBC Connection Pool Properties List

Property	Description
Database Name	Specifies the name of the database for which the pool is created. The default value is **oracle**.
DataSource Class Name	Specifies the name of the DataSource class. The default value is **oracle jdbc pool OracleDataSource**.
Extra Properties	Specifies custom properties for the DataSource, using semicolon-separated key-value pairs. The default value is **DriverType=thin**.

Table 45 Oracle JDBC Connection Pool Properties List

Property	Description
Maximum Pool Size	Specifies the maximum number of connections in the pool. The default value is **10**.
Minimum Pool Size	Specifies the minimum number of connections in the pool. The default value is **0**.
Network Protocol	Specifies the network protocol. The default value is **tcp**.
Password	Specifies the password for the connection. There is no default value.
Pool Idle Time	Specifies the maximum time period in milliseconds that a connection may remain unused before it is removed from the pool in order to reduce the pool size. The default value is **600,000** ms (10 min).
Pool JINI Name	Specifies the unique JINI name of the DataSource pool. The pool is bound in the java/namespace for local access or into the global namespace for remote access. The default value is **jdbc/ds/pool1**.
Port Number	Specifies the port number on which the server receives data. The default value is **1521**.
Remote	Specifies whether or not the DataSource should be bound into the global remote JINI namespace for access by remote clients. The default value is **true**.
Request Timeout	Specifies the maximum time period in milliseconds that a request for connection from the pool may block all other connections currently in use. The default value is **100,000** ms.
Server Name	Specifies the host name of the database server or IP address where the database server is running. The default value is **localhost**.
User	Specifies the user name authorized for creating connections. There is no default value.
XA Recovery Password	For XA DataSources only, specifies the password to use for XA transaction recovery. There is no default value.
XA Recovery User Name	For XA DataSources only, specifies the user name to use for XA transaction recovery. There is no default value.

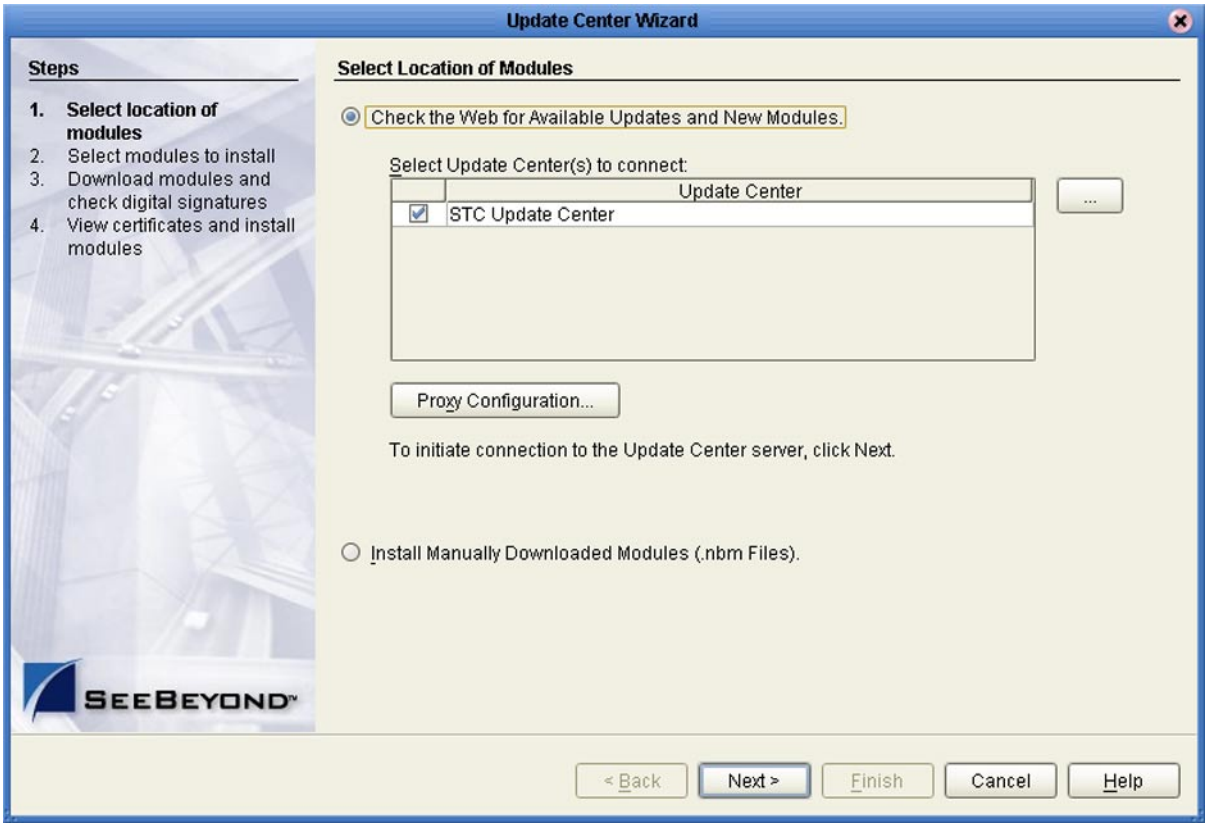
7.5.2 Using a Proxy Server

The following procedure allows eGate Projects to use a proxy server during run time. This allows you to reference a WSDL file using a URL that points outside your firewall.

To configure a proxy server

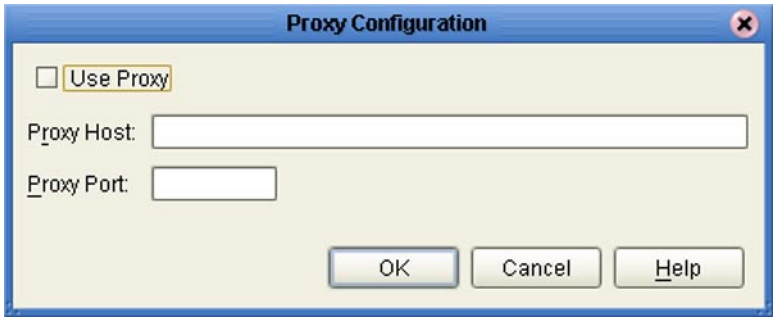
1 From Enterprise Designer's **Tools** menu, select **Update Center** to display the Update Center Wizard (see Figure 107).

Figure 107 Update Center Wizard



2 Click the **Proxy Configuration** button to display the Proxy Configuration dialog box shown in Figure 108.

Figure 108 Proxy Configuration Dialog Box



3 Check the **Use Proxy** box, and fill in the fully-qualified name of the proxy host and port for the proxy server.

4 Click **OK** to finish.

7.5.3 Deploying User-Defined Stateless Session Beans

User-defined stateless session beans can be deployed to the eGate Integration Server following the procedure outlined in this section.

Note: *The deployment of stateful session beans, entity beans, and message-driven beans is not currently supported.*

To deploy a stand-alone SLSB to the eGate Integration Server

1. Create and compile the EJB.
2. Write the **ejb-jar.xml** and **seebeyond-ejb.xml** deployment descriptors for your EJB.
3. Create a **.jar** file with the deployment descriptors in the **\META-INF** directory and the code in the root.
4. Move the **.jar** file into the **\logicalhost\stcis\deploy\new***integration_server_name* directory for deployment. The Integration Server will automatically pick up the **.jar** file from this location and deploy the EJB.

Examples of the EJBs and associated **.xml** files are as follows.

Example Remote Interface

```
package ejb.CustomApp;
import java.rmi.RemoteException;
import java.rmi.Remote;
import javax.ejb.*;

public interface CustomApp extends EJBObject, Remote
{
  public String getId() throws RemoteException;
}
```

Example Home Interface

```
package ejb.CustomApp;

import javax.ejb.*;
import java.rmi.Remote;
import java.rmi.RemoteException;
import java.util.*;

public interface CustomAppHome extends EJBHome
{
   public CustomApp create() throws CreateException, RemoteException;
}
```

Example Stateless Session Bean (SLSB)

```
package ejb.CustomApp;

import javax.ejb.*;
import java.io.Serializable;
import java.util.*;
import java.rmi.*;
import javax.naming.Context;
import javax.naming.InitialContext;

// import addtional classes as needed "CustomController"
```

```
public class CustomAppBean implements SessionBean
{
  private SessionContext ctx;
  private CustomController mCustom;

  public void setSessionContext( SessionContext context )
  {
     this.ctx = context;
  }

  public void ejbCreate()
  {
     try {
       javax.naming.Context context = new InitialContext();
       // lookup Custom application
       Object ref = context.lookup("ejb/CustomController");
       CustomControllerHome CustomHome =
(CustomControllerHome)javax.rmi.PortableRemoteObject.narrow(ref,
CustomControllerHome.class);
       mCustom = CustomHome.create();
     } catch (Exception e) {
     System.out.println( e.getMessage() );
     }
  }

  public String getId()
  {
     SystemObjectPK key = new SystemObjectPK( "SBYN", "0000000001" );
     String EUID= "Not Found";
     try {
    EUID = mCustom.getEUID( key );
     }
     catch (Exception e) {
     System.out.println("===> Exception: " );
     System.out.println( e.getMessage() );
     }

     return( EUID );
  }

  // add addtional EJB methods

}
```

Example ejb-jar.xml file for the above SLSB

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ejb-jar PUBLIC '-//Sun Microsystems, Inc.//DTD Enterprise
JavaBeans 2.0//EN' 'http://java.sun.com/dtd/ejb-jar_2_0.dtd'>
<!-- Generated XML! -->
<ejb-jar>
    <display-name>ServiceBeans</display-name>
    <enterprise-beans>
        <session>
            <description><![CDATA[Custom App Session Bean]]></
description>
            <display-name>Custom App</display-name>
            <ejb-name>CustomApp</ejb-name>
            <home>ejb.CustomApp.CustomAppHome</home>
            <remote>ejb.CustomApp.CustomApp</remote>
            <ejb-class>ejb.CustomApp.CustomAppBean</ejb-class>
            <session-type>Stateless</session-type>
            <transaction-type>Bean</transaction-type>
            <ejb-ref>
                <ejb-ref-name>ejb/CustomController</ejb-ref-name>
```

```
                    <ejb-ref-type>Session</ejb-ref-type>

<home>com.stc.eindex.ejb.Custom.CustomControllerHome</home>
                    <remote>com.stc.eindex.ejb.Custom.CustomController</
remote>
                    <ejb-link>CustomController</ejb-link>
                </ejb-ref>
            </session>
        </enterprise-beans>
</ejb-jar>
```

Example seebeyond-ejb.xml file for the above SLSB

```
<sbyn-ejb-deployment-descriptor>
    <enterprise-beans>
        <session>
            <ejb-name>CustomApp</ejb-name>
            <jndi-name>ejb/CustomApp</jndi-name>
            <security>
                <authorize>no</authorize>
                <authenticate>no</authenticate>
                <security-audit>no</security-audit>
            </security>
            <pool-min>1</pool-min>
        </session>
    </enterprise-beans>
</sbyn-ejb-deployment-descriptor>
```

SLSB Deployment verification

Examine the log file **\logicalhost\logs\stc_is_*integration_server_name*.log**. You should find text such as "**CustomApp (EJB) was successfully deployed**" confirming deployment.

To remove a stand-alone SLSB from the eGate Integration Server

1. Shut down the Logical Host containing the Integration Server where the SLSB is deployed.
2. Remove the **.jar** file created in the preceding deployment procedure from the **\logicalhost\stcis\repository\applications*integration_server_name*\EJB** directory.
3. Restart the Logical Host.

7.6 Message Servers

The Logical Host contains one or more Message Servers, which manage JMS topics (publish-and-subscribe messaging) and queues (point-to-point messaging). eGate Integrator includes the SeeBeyond JMS IQ Manager as its Java Messaging Service (JMS) implementation. The JMS IQ Manager conforms to the Java Message specification 1.0.2b, and supports both topic (publish-and-subscribe) and queue (point-to-point) messaging styles.

This section presents an overview of the configuration properties for the SeeBeyond JMS IQ Manager. Third-party application servers such as BEA WebLogic and IBM WebSphere incorporate their own message servers. For more information on the JMS IQ Manager, and deploying Project components to third-party message servers, see the *eGate Integrator JMS Reference Guide*.

7.6.1 SeeBeyond JMS IQ Manager Configuration

General Configuration

These properties cover the basic configuration of the JMS IQ Manager.

Figure 109 JMS IQ Manager - General Configuration Properties

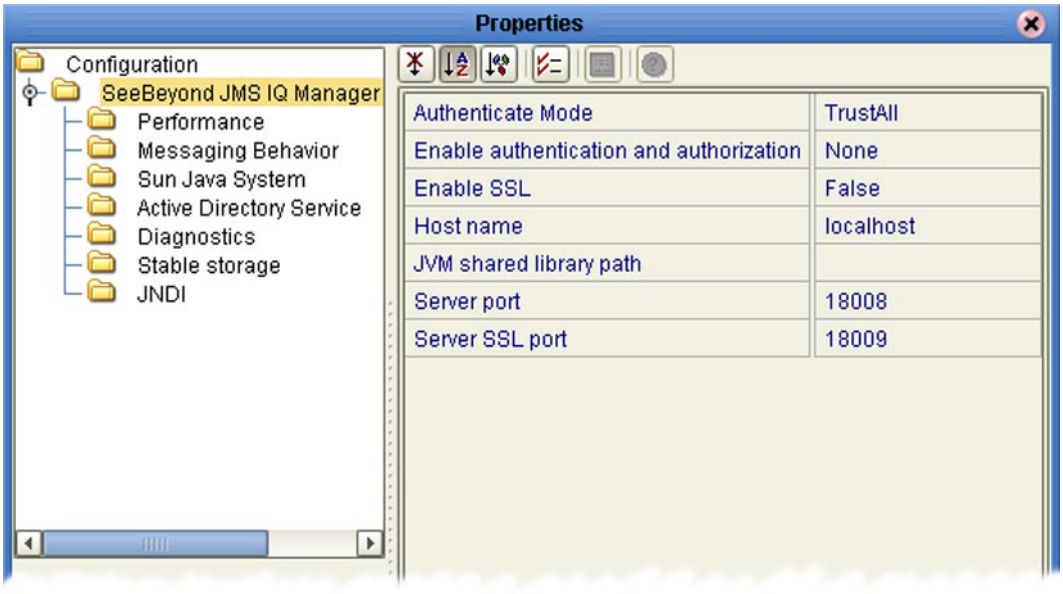


Table 46 JMS IQ Manager - General Configuration Properties List

Property	Description
Authenticate Mode	Specifies whether or not the JMS IQ Manager authenticates components attempting to connect to it. The options are **Authenticate** and **TrustAll**; the default value is **TrustAll**.
Enable authentication and authorization	Specifies whether or not security is enabled for the JMS IQ Manager, requiring a user name and password. The options are: ▪ **None** (no security—default value) ▪ **File** (use File realm) ▪ **SunONE** (use SunONE LDAP server) ▪ **AD** (use Active Directory LDAP server) If you specify either **SunONE** or **AD**, you must also configure your LDAP server. See **Sun Java System** on page 148 and **Active Directory Service** on page 149.
Enable SSL	Specifies whether or not to enable the Secure Sockets Layer (SSL) protocol for the JMS IQ Manager's TCP/IP connections. The options are **True** and **False**; the default value is **False**.
Host name	Specifies the name of the host system for the JMS IQ Manager. The default value is **localhost**.
JVM shared library path	Specifies the path to the Java Virtual Machine shared library. The value depends on the operating system under which the Logical Host is running—use the ellipsis (...) button to select the correct value. The default value is **../../jre/bin/client/jvm.dll**, for Windows platforms.
Server port	Specifies the TCP/IP port number; the default value depends upon the number of the Logical Host base port. Each JMS IQ Manager must have a unique port number per system.
Server SSL port	Specifies the port on which the JMS IQ Manager listens for SSL connections; the default value depends upon the number of the Logical Host base port.

Performance

The Performance properties allow you to specify memory usage for optimum system performance. See the *eGate Integrator JMS Reference Guide* for additional information.

Figure 110 Performance Configuration Properties

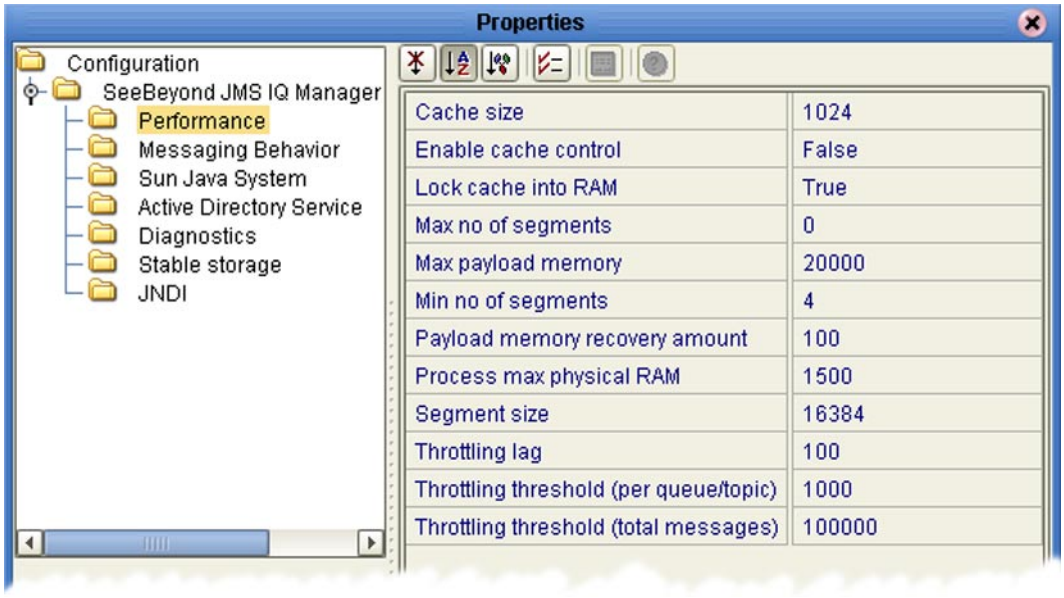


Table 47 Performance Configuration Properties List

Property	Description
Cache size	Specifies the total number of pages in the *read* cache (a page is 512 bytes on Windows and 1024 bytes on UNIX). The default value is **1024** pages (0.5 MB for Windows, 1 MB for UNIX).
Enable cache control	Specifies whether or not the JMS IQ Manager controls the cache synchronization to disk. The options are **True** and **False**; the default is **False**.
Lock cache into RAM	Specifies whether or not the cache is locked into physical memory. The options are **True** and **False**; the default is **True**.
Max. no. of segments	Specifies the upper limit for the number of database files that the JMS IQ Manager creates for its stable message storage. The default value is **0**, which causes the JMS IQ Manager to create files as needed—limited only by disk space.
Max. payload memory	Specifies the upper limit for the amount of cache, in kB, allocated for the server to keep message payloads in cache. The default value is **20,000** kB.

Table 47 Performance Configuration Properties List

Property	Description
Min. no. of segments	Specifies the minimum number of database files that the JMS IQ Manager initially creates for its stable message storage. The default value is **4** segments (files).
Payload memory recovery amount	Specifies the amount of cache, in kB, to recover in a recovery and cleanup operation. The default value is **100** kB.
Process max. physical RAM	Specifies the upper limit for the amount of RAM, in kB, allocated for use by the JMS IQ Manager as working memory (Windows platforms only). The default value is **1500** kB.
Segment size	Specifies the total number of pages in a single database file (segment). The default value is **16,384** pages (8 MB for Windows, 16 MB for UNIX).
Throttling lag	Specifies the number of messages that must be dequeued before message producers are no longer throttled. The default value is **100**.
Throttling threshold (per queue/topic)	Specifies the maximum number of messages after which all message producers for the message destination (topic or queue) are throttled. The default value is **1000** messages.
Throttling threshold (total messages)	Specifies the maximum number of messages for all message destinations, after which the JMS IQ Manager begins throttling producers. The default value is **100,000** messages.

Messaging Behavior

The Messaging Behavior properties allow you to configure the order of message delivery. See the *eGate Integrator JMS Reference Guide* for additional information.

Figure 111 Messaging Behavior Configuration Properties

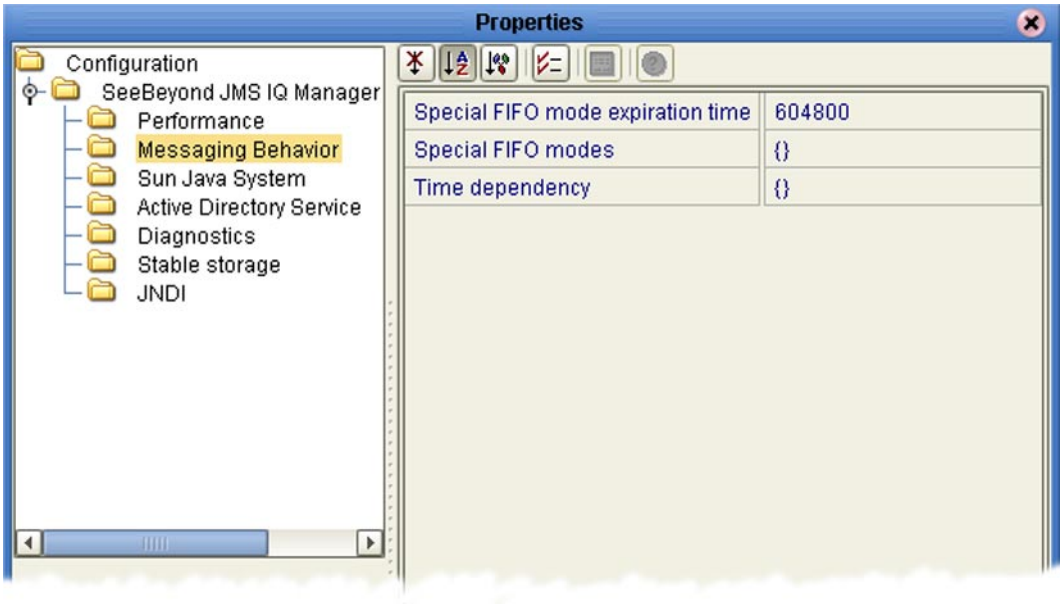


Table 48 Messaging Behavior Configuration Properties List

Property	Description
Special FIFO mode expiration time	Specifies the expiration time in seconds for first-in, first-out (FIFO) mode operation. The default value is **604,800** seconds (7 days).
Special FIFO modes	Specifies the FIFO delivery order: ▪ fully concurrent (**0**) ▪ protected concurrent (**1**) ▪ fully serialized (**2**) The default value is (**0**). See the *eGate Integrator JMS Reference Guide* for specification syntax.
Time dependency	Specifies whether or not the processing order is dependent on messages are associated with other destinations. The default value is **0**, indicating no such dependency. See the *eGate Integrator JMS Reference Guide* for syntax.

Sun Java System

These configuration properties are used only if you are using SunONE Directory Server as an LDAP server. See the *eGate Integrator System Administration Guide* for additional information.

Figure 112 Sun Java System Configuration Properties

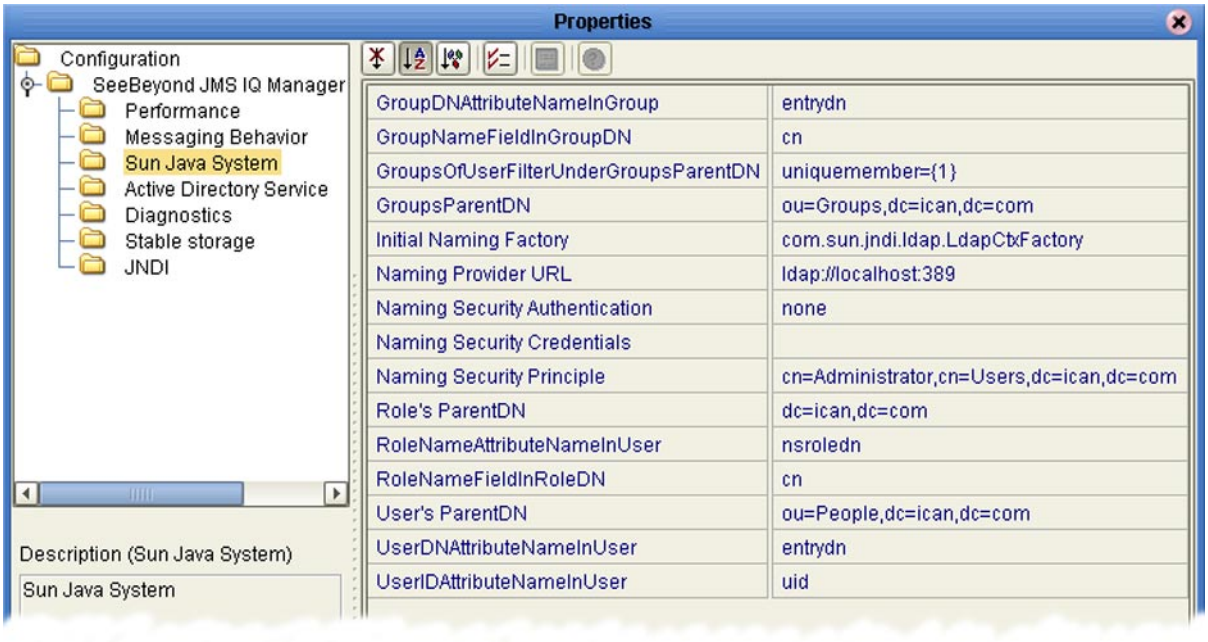


Table 49 Sun Java System Configuration Properties List

Property	Default Value
GroupDNAttributeNameInGroup	entrydn
GroupNameFieldInGroupDN	cn
GroupOfUserFilterUnderGroupsParentDN	uniquemember={1}
GroupsParentDN	cn=Groups,dc=ican,dc=com
Initial Naming Factory	com.sun.jndi.ldap.LdapCtxFactory
Naming Provider URL	ldap#10.18.73.63:489
Naming Security Authentication	none
Naming Security Credentials	
Naming Security Principle	cn=Administrator,cn=Users,dc=ican,dc=com
Role's ParentDN	dc=ican,dc=com
RoleDNAttributeNameInRole	nsroledn
RolesOfUserFilterUnderGroupsParentDN	cn
User's ParentDN	cn=People,dc=ican,dc=com
UserDNAttributeNameInUser	entrydn

Table 49 Sun Java System Configuration Properties List

Property	Default Value
UserIDAttributeNameInUser	uid

Active Directory Service

These configuration properties are used only if you are using the Microsoft Active Directory Service as an LDAP server. See the *eGate Integrator System Administration Guide* for additional information.

Figure 113 Active Directory Service Configuration Properties

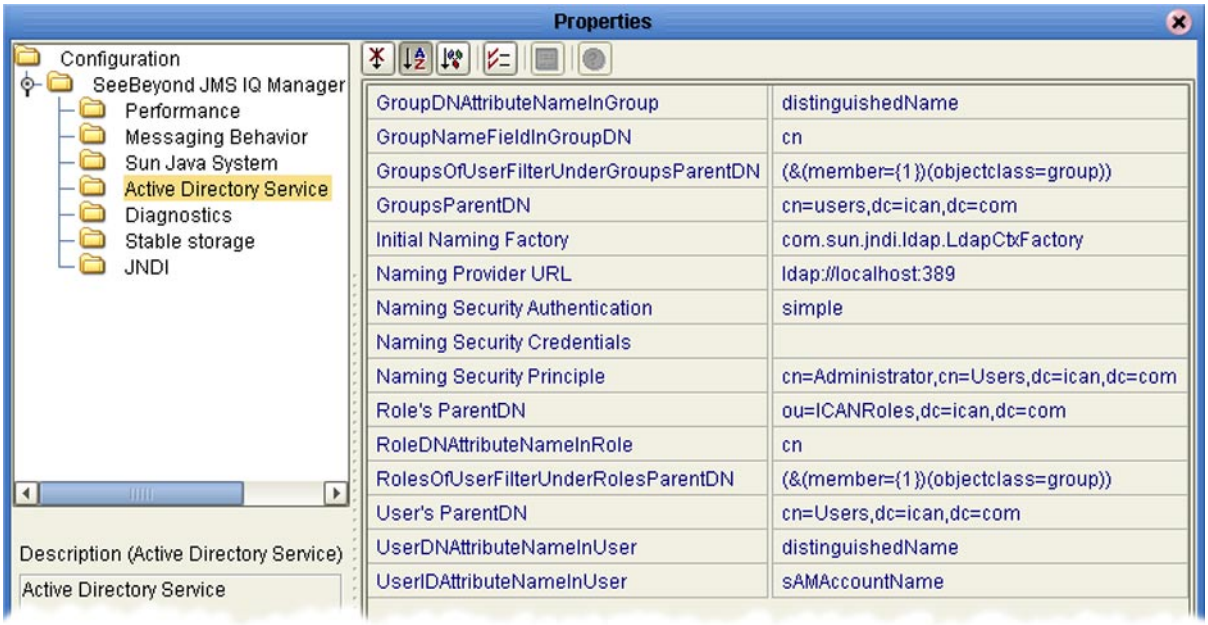


Table 50 Active Directory Service Configuration Properties List

Property	Default Value
GroupDNAttributeNameInGroup	distinguishedName
GroupNameFieldInGroupDN	cn
GroupOfUserFilterUnderGroupsParentDN	(&(member={1})(objectclass=group))
GroupsParentDN	cn=users,dc=ican,dc=com
Initial Naming Factory	com.sun.jndi.ldap.LdapCtxFactory
Naming Provider URL	ldap#10.18.73.63:389
Naming Security Authentication	simple
Naming Security Credentials	
Naming Security Principle	cn=Administrator,cn=Users,dc=ican,dc=com
Role's ParentDN	ou=ICANRoles,dc=ican,dc=com

Table 50 Active Directory Service Configuration Properties List

Property	Default Value
RoleDNAttributeNameInRole	cn
RolesOfUserFilterUnderGroupsParentDN	(&(member={1})(objectclass=group))
User's ParentDN	cn=users,dc=ican,dc=com
UserDNAttributeNameInUser	distinguishedName
UserIDAttributeNameInUser	sAMAccountName

Diagnostics

The Diagnostics properties allow you to configure the logging operations for the JMS IQ Manager. See the *eGate Integrator JMS Reference Guide* for additional information.

Figure 114 Diagnostics Configuration Properties

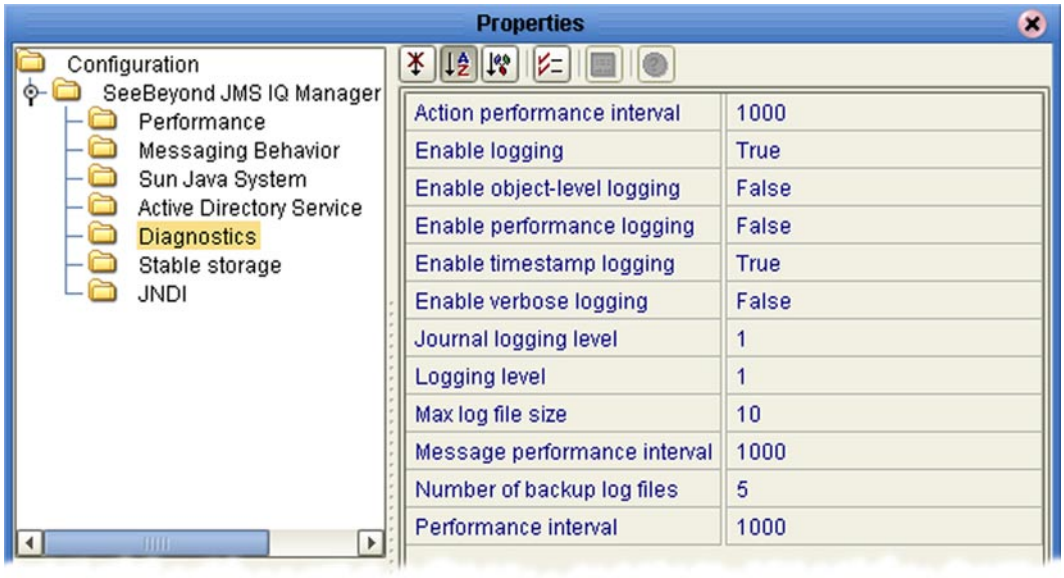


Table 51 Diagnostics Configuration Properties List

Property	Description
Action performance interval	Specifies how many times the ActionManager's *Update* function is called before logging its performance statistics. The default is **1000**.
Enable logging	Specifies whether or not diagnostic information is written to the JMS IQ Manager log file. **True** (the default) enables logging, **False** disables it.

Table 51 Diagnostics Configuration Properties List

Property	Description
Enable object-level logging	Specifies whether or not object (constructor/deconstructor) information is written to the JMS IQ Manager log file. **True** enables logging, **False** (the default) disables it.
Enable performance logging	Specifies whether or not performance information is written to the JMS IQ Manager log file. **True** enables logging, **False** (the default) disables it.
Enable timestamp logging	Specifies whether or not timestamp information is written to the JMS IQ Manager log file. **True** (the default) enables logging, **False** disables it.
Enable verbose logging	Specifies whether or not full-length messages are written to the JMS IQ Manager log file. **True** enables logging, **False** (the default) disables it.
Journal logging level	Specifies the threshold severity level at which the system issues informational, warning, and error messages, and writes them to the JMS IQ Manager journal log. ▪ all messages (**0**) ▪ warning, error, and fatal messages (**1**) ▪ error and fatal messages (**2**) ▪ fatal messages only (**3**) The default level is (**1**).
Logging level	Specifies the threshold severity level at which the system issues informational, warning, and error messages, and writes them to the JMS IQ Manager log. ▪ all messages (**0**) ▪ warning, error, and fatal messages (**1**) ▪ error and fatal messages (**2**) ▪ fatal messages only (**3**) The default level is (**1**).
Max. log file size	Specifies the maximum size of the JMS IQ Manager log file, in MB. The default is **10** MB.
Message performance interval	Specifies the number of messages the JMS IQ Manager processes before logging its performance statistics. The default is **1000**.
Number of backup log files	Specifies the maximum number of JMS IQ Manager backup log files in the stack. The default is **5**.
Performance interval	Specifies the number of IMessages the IMessageManager creates before logging its performance statistics. The default is **1000**.

Stable Storage

The Stable Storage properties allow you to configure the storage and journaling operations for the JMS IQ Manager. See the *eGate Integrator JMS Reference Guide* for additional information.

Figure 115 Stable Storage Configuration Properties

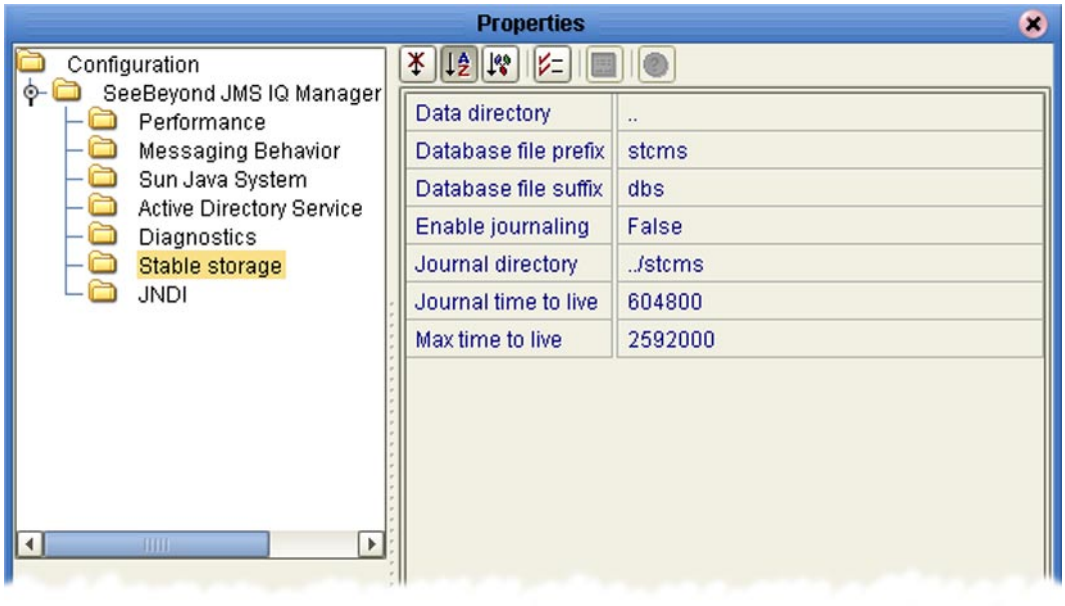


Table 52 Stable Storage Configuration Properties List

Property	Description
Data directory	Specifies the location for the JMS IQ Manager database files. The path can be either absolute or relative to the **logicalhost** directory. The default is **..** (one directory up from the **logicalhost** directory).
Database file prefix	Specifies the prefix for the JMS IQ Manager database file names. The default is **stcms** (for default file names of *stcms*.dbs*).
Database file suffix	Specifies the extension for the JMS IQ Manager database file names. The default is **dbs** (for default file names of *stcms*.dbs*).
Enable journaling	Enables or disables journaling, whereby every inbound message is automatically copied to the journal database. **True** enables, **False** disables; the default is **False**.
Journal directory	Specifies the location for the journal database files. The path can be either absolute or relative to the **logicalhost** directory. The default is **../stcms**.

Table 52 Stable Storage Configuration Properties List

Property	Description
Journal time to live	Specifies the maximum amount of time, in seconds, a journaled message persists before expiring. The default is **604,800** seconds (7 days).
Max. time to live	Specifies the maximum amount of time, in seconds, a live message persists before being removed from the queue. The default is **2,592,000** seconds (30 days).

JNDI

The JNDI configuration properties are not user-definable.

Chapter 8

Project Deployment

This chapter describes the process of creating deployment profiles and activating the deployed projects.

8.1 Deployment Profiles

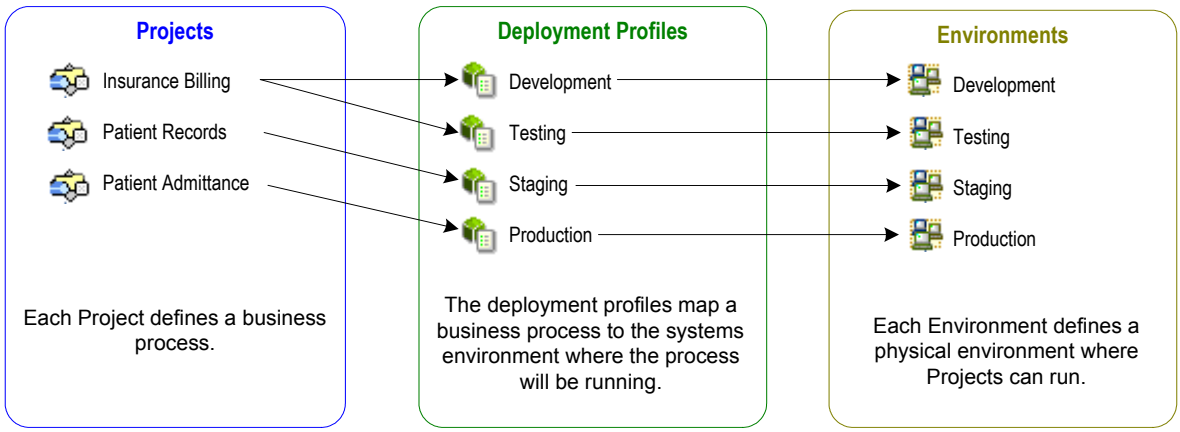
Deployment Profiles define specific instances of a Project in a particular Environment. A deployment profile contains information about the assignment of Services and Message Destinations to integration and message servers (JMS IQ Managers). It also contains version information for all relevant objects in the Project. The Enterprise Designer includes a Deployment Editor, which you can use to create and customize deployment profiles.

Note that:

- Each Project can have zero or more Deployment Profiles, but each of a Project's active Deployment Profiles must be in a separate Environment.
- Each Environment can have zero or more Deployment Profiles assigned to it, but any given Environment can have only one Deployment Profile from a given Project.

Repeating Figure 2 from the **System Overview** on page 19:

Figure 116 eGate Integrator Implementation Model



8.2 The Deployment Editor

The Deployment Editor (see Figure 117) allows you to create a new Deployment Profile or edit an existing one. To create a new Deployment Profile, right-click on a Project in the Project Explorer to display its context menu. From the menu, select **New > Deployment Profile**. To edit an existing Deployment Profile, simply click on its icon.

Figure 117 Deployment Editor Window

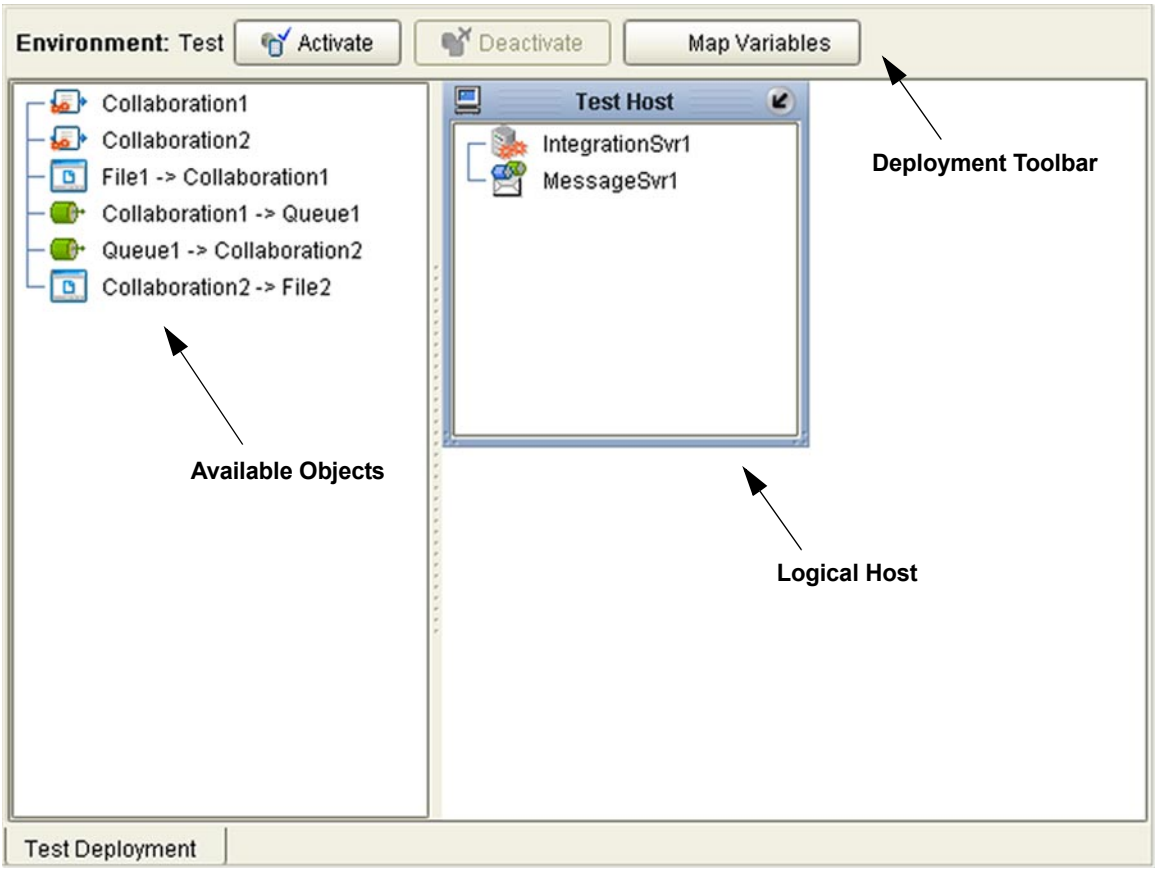


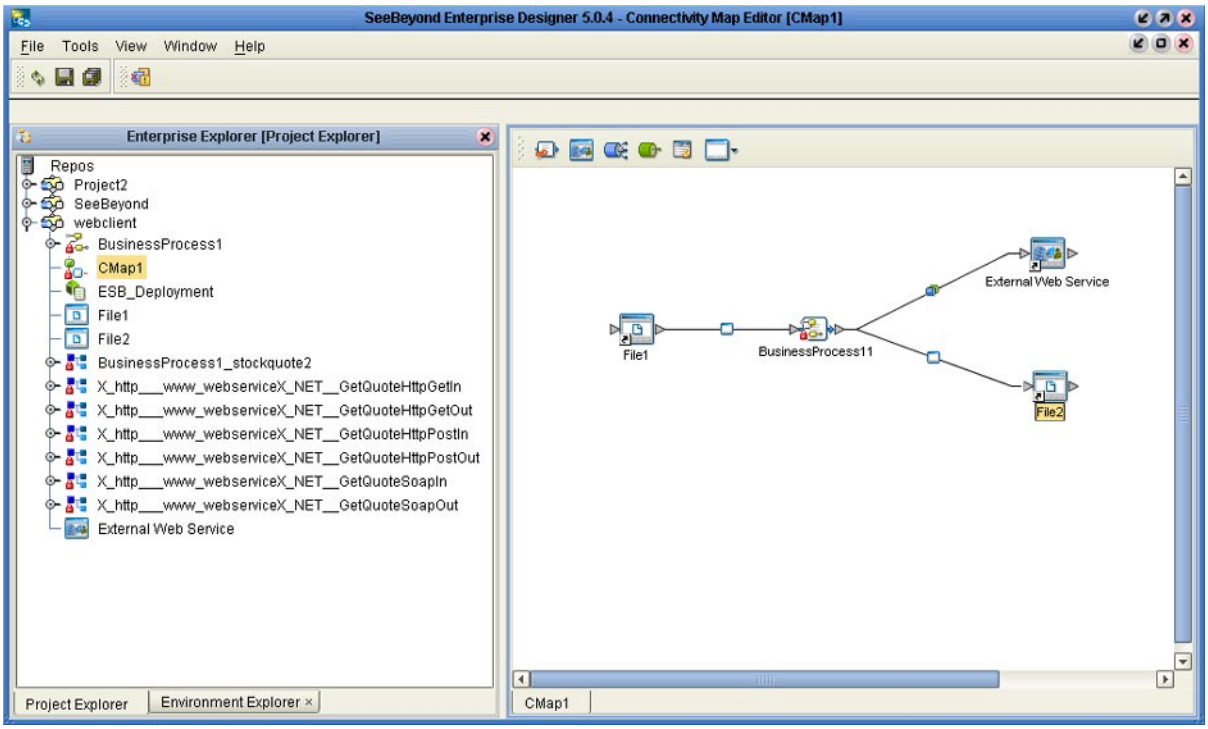
Table 53 Deployment Toolbar Buttons

Button	Function
Activate	Starts the Project by creating an enterprise archive (EAR) file based on the Connectivity Map and linking this file with the application server. See **Activating and Deactivating Deployment Profiles** on page 159.
Deactivate	Stops the Project by terminating the link between the EAR file and the application server, sets the Deployment Profile to *inactive*, and saves to the Repository.
Map Variables	Allows you to assign names and values to Project variables for the specific Deployment Profile. See **Mapping Variables** on page 162.

8.3 Creating a Deployment Profile

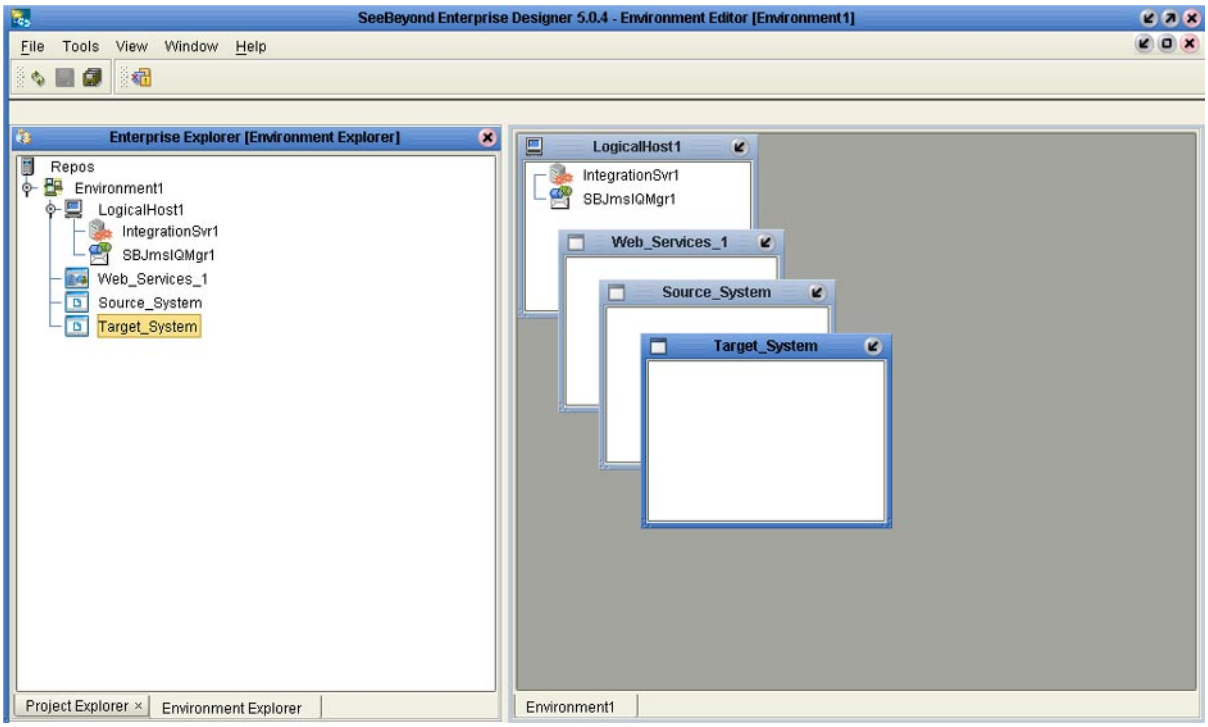
The Web Client Project shown in Figure 118 will be used as a deployment example.

Figure 118 Web Client Example Project



1. In the Environment Explorer, create an Environment and right-click on the Environment to display its context menu.
2. From the menu, select the Environment components you need and name them appropriately. They will appear as shown in Figure 119.

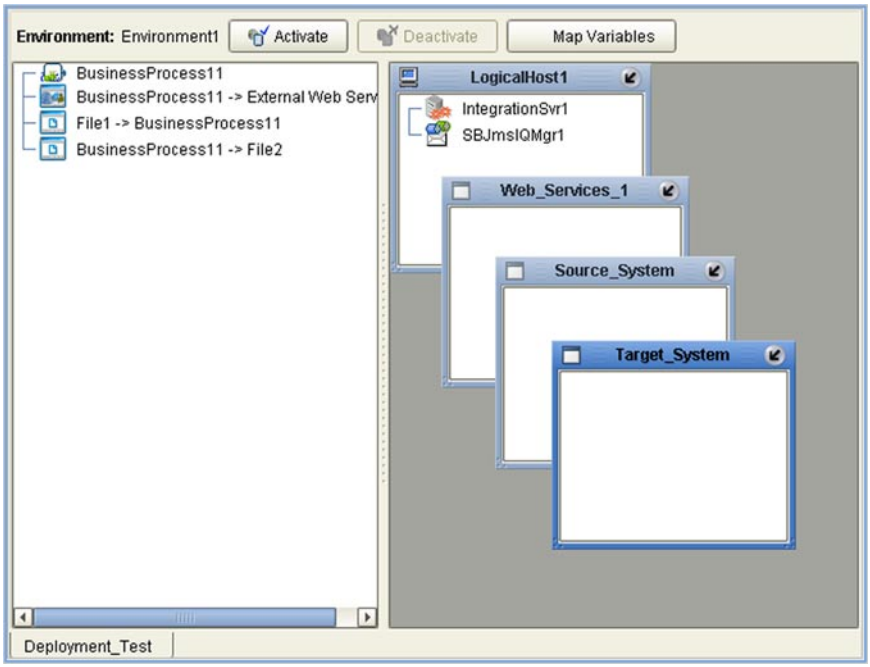
Figure 119 Web Client Example Environment



3 In the Project Explorer, right-click on the Project to display its context menu.

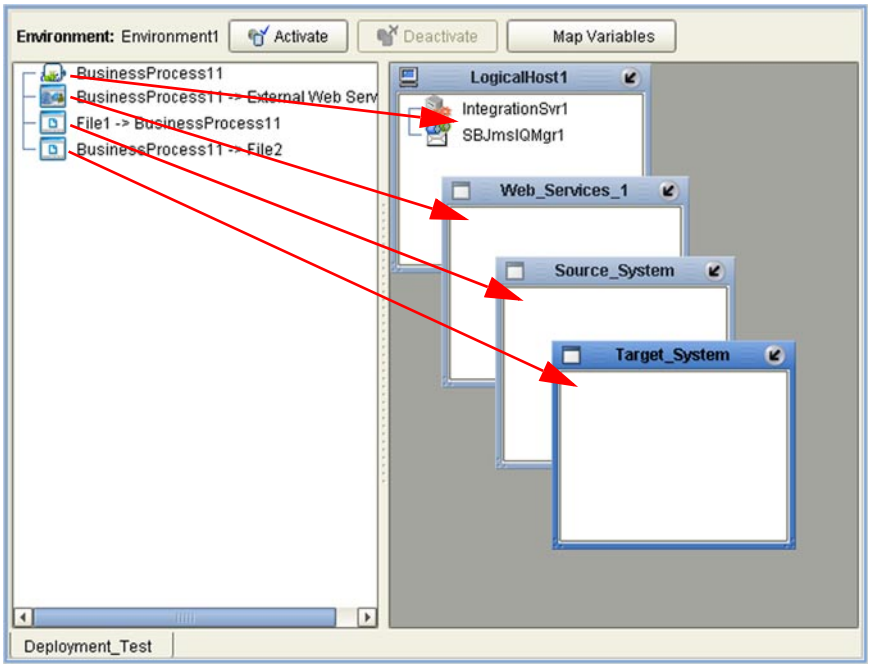
4 From the menu, select **New > Deployment Profile**. The Deployment Profile Editor appears, displaying the Environment you created (see Figure 120).

Figure 120 Example Deployment Profile (1)



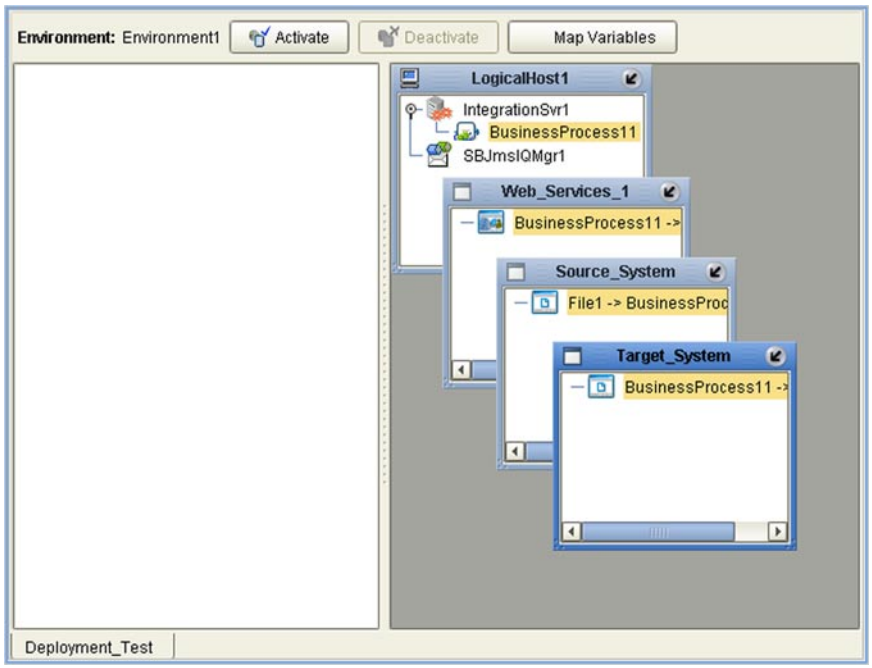
5 Drag the Project components from the left panel and drop them into the appropriate Environment components in the right panel, as illustrated in Figure 121.

Figure 121 Example Deployment Profile (2)



6 When the Environment components are fully populated, the left panel will be blank, as shown in Figure 122. You should now **Save** the profile.

Figure 122 Example Deployment Profile (3)



8.4 Activating and Deactivating Deployment Profiles

When activating or deactivating a Deployment Profile, you have the option of applying the changes to the Logical Host either immediately or at a later time. By activating the Deployment Profile without immediately applying the changes, you can check the validity of the entire Deployment Profile.

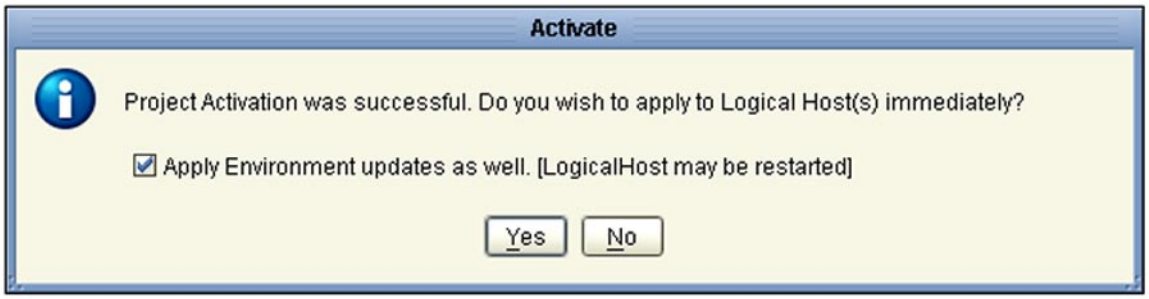
If you have multiple Deployment Profiles to deploy, you might work more efficiently by activating each of the Deployment Profiles without applying the changes, and then applying all of the changes to the Logical Host at a later time (*heed the warning given in the following procedures*).

8.4.1 Using Enterprise Designer

To activate a Deployment Profile

1. In the Project Explorer, select the Deployment Profile you want to activate.
2. Click the **Activate** button, which invokes the dialog box shown in Figure 123:

Figure 123 Activate Dialog Box

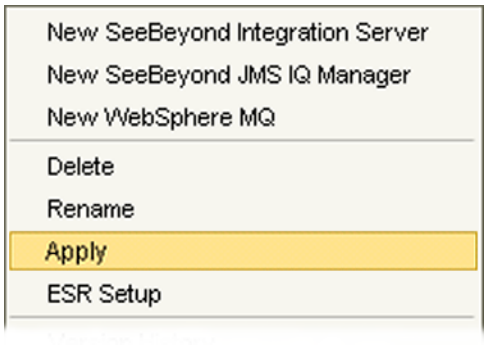


3. Select your response based on the following criteria:
   - If the Logical Host is running, and you wish to apply the changes immediately, click **Yes**.
   - If the Logical Host has not yet been bootstrapped, or you wish to apply the changes at a later time, click **No**.
4. If you click **Yes**, the information box shown in Figure 124 will appear after the changes have been applied. Click **OK** to proceed.

Figure 124

5 To apply the changes at a later time, right-click the Logical Host and select **Apply** from the menu (see Figure 125). This will download the latest configuration from the Repository to that Logical Host.

Figure 125 Logical Host Context Menu - Apply



6 The information box shown in Figure 124 will appear after the changes have been completed. Click **OK** to proceed.

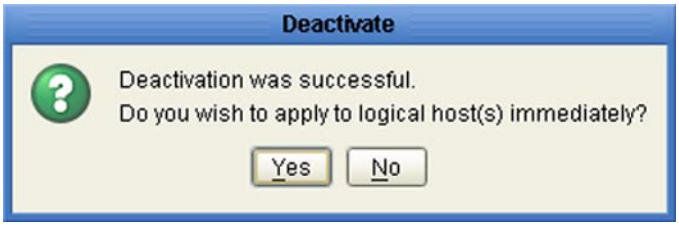
Important: *Use the ICAN Monitor to verify that your changes have been applied before again running the apply command.*

To deactivate a Deployment Profile

1 In the Project Explorer, select the Deployment Profile you want to deactivate.

2 Click the **Deactivate** button, which invokes the dialog box shown in Figure 126:

Figure 126 Deactivate Dialog Box



3 Select your response based on the following criteria:

- If the Logical Host is running, and you wish to apply the changes immediately, click **Yes**.
- If the Logical Host has not yet been bootstrapped, or you wish to apply the changes at a later time, click **No**.

4 To apply the changes at a later time, right-click the Logical Host and select **Apply** from the menu (see Figure 125). This will apply all of the changes for that Logical Host.

Important: *Use the ICAN Monitor to verify that your changes have been applied before again running the apply command.*

8.4.2 Using a Command-line Script

A script named **CmdLineUtil.bat** (or **.sh**) allows you to deploy and undeploy projects via the command line. The *apply* and *unapply* commands provide the same function as clicking **Activate** or **Deactivate**, and then clicking **Yes**, in Enterprise Designer.

This command-line utility allows you three options for deployment:

- **Apply to all Logical Hosts in the Environment** (you must provide the Environment name).
- **Apply to a single Logical Host** (you must provide the Environment name and the Logical Host name).
- **Apply to a specific Deployment Profile** (you must provide the Project name and the Deployment Profile name).

The command-line utility also allows you to activate or deactivate from any computer, not only the one on which the Logical Host is installed. User name/password authentication is performed before the utility can be used.

This utility must be downloaded using Enterprise Manager as described in the *SeeBeyond ICAN Suite Installation Guide*. Before running the utility:

- The JAVA_HOME environmental variable must be set on the host computer.
- The Repository must be running.
- You must first activate or deactivate the Project in Enterprise Designer.
- You must perform an initial bootstrap of the Logical Host(s) to download the components from the Repository.

Important: *Use the ICAN Monitor to verify that your changes have been applied before again running the apply or unapply command.*

8.5 Mapping Variables

Project variables function as placeholders, having values that are determined when you create a specific Deployment Profile. These values can be literals or Environmental constants. Clicking the **Map Variable** button displays the Deployment Profile Mappings panel, where you can assign names (see Figure 127) and values (see Figure 128).

Figure 127 Deployment Profile Mappings

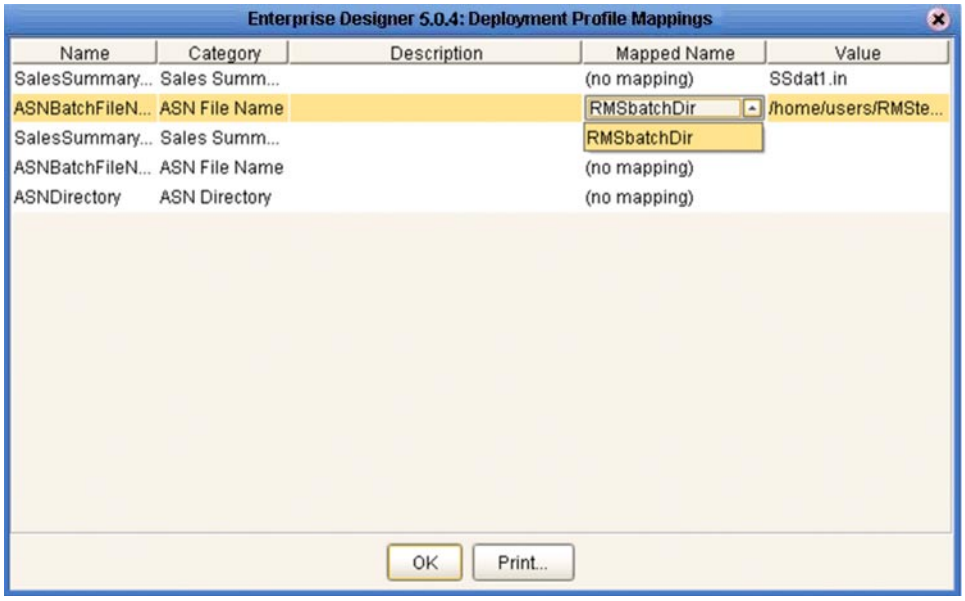
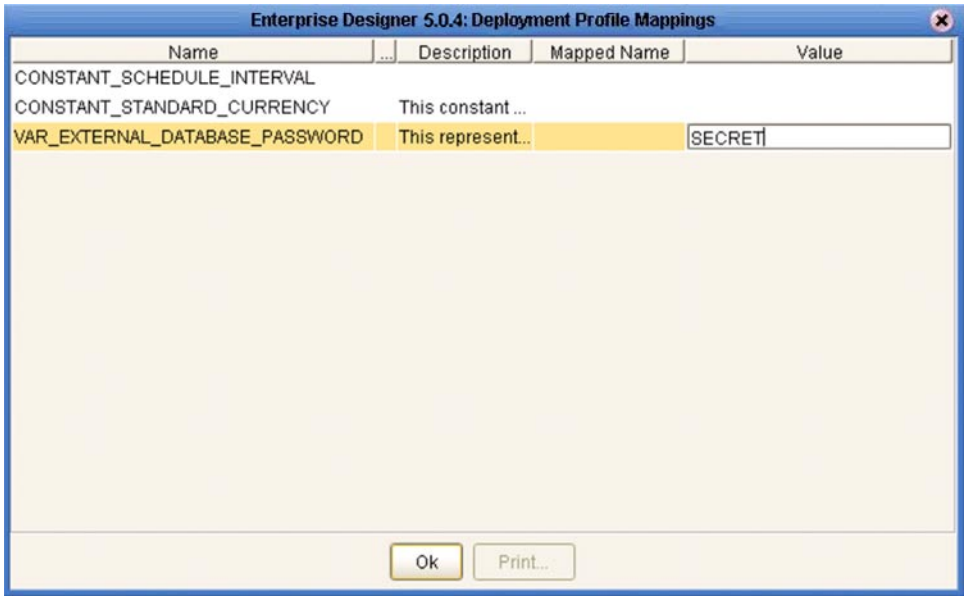


Figure 128 Project Variable Value Entry



8.6 Deploying Projects to Third-Party Servers

SeeBeyond's eGate Integrator allows you to develop Projects using Enterprise Designer and deploy them to a BEA WebLogic or IBM WebSphere environment. The SAR files for these third-party products must be installed prior to deployment, as described in the *eGate Integrator JMS Reference Guide*.

Because of the versions of the Java Connection Architecture supported by WebLogic and WebSphere, the following restrictions apply:

- Services deployed to WebLogic or WebSphere are restricted to those internal to eGate Integrator itself (between message destinations), and those associated with outbound eWays.
- Not all SeeBeyond eWays support third-party servers. Check the individual eWay User's Guides regarding such support, and also any additional configuration that may be necessary for compatibility with WebLogic or WebSphere.

8.6.1 BEA WebLogic

Note: *Before using the WebLogic JMS, you must install additional* ***.jar*** *files as described below. For additional information, see the eGate Integrator JMS Reference Guide.*

To install additional .jar files

1. Download the **log4j.jar** file from the location below (this location may change).

   http://jakarta.apache.org/log4j/docs
2. Download the **xerces.jar** file from the location below (this location may change).

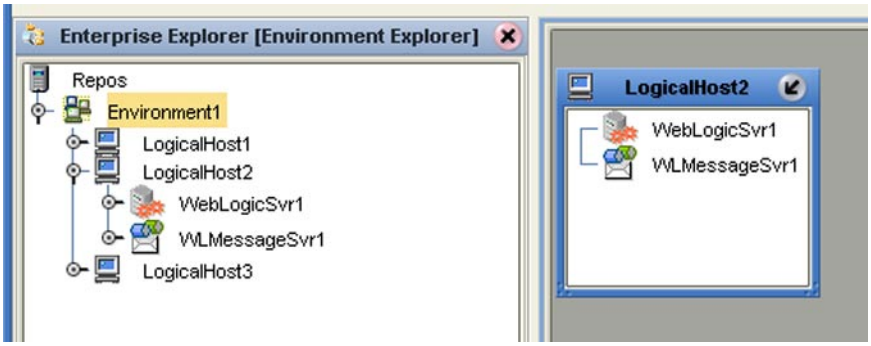
   http://xml.apache.org/dist/xerces-j
3. Place both **.jar** files into the ***weblogic8x*\server\lib** directory.
4. Add the **.jar** files to the *set CLASSPATH* segment of the **startWLS.cmd** file located in the ***weblogic8x*\server\bin** directory. The text to be added is:

```
%WL_HOME%\server\lib\log4j.jar;%WL_HOME%\server\lib\xerces.jar
```

To deploy an eGate Project to a BEA WebLogic 8.0 or 8.1 environment

1. Create the following components in Enterprise Designer (see Figure 129):
   - A A new environment
   - B A Logical Host
   - C A WebLogic J2EE application server
   - D A WebLogic JMS message server

Figure 129 WebLogic Deployment (1)

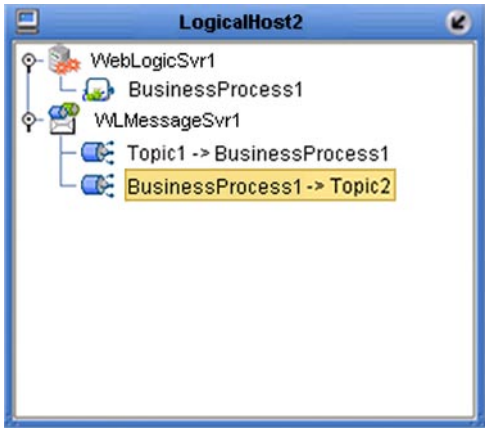


2 Create a new Deployment Profile to bind the Connectivity Map to the new WebLogic environment (see Figure 130).

A Drag the two topics and drop onto the WebLogic message server.

B Drag the Business Process and drop onto the WebLogic application server.

Figure 130 WebLogic Deployment (2)



3 Activate the Deployment Profile.

Activating the Deployment Profile creates an Environment Archive (EAR) file, which contains all files necessary to create and run an application in WebLogic. This file can be found in the following location:

```
ICAN-root\repository\data\files\WLEnvironmrntName\
    ProjectName_DeploymentProfileName.ear
```

Note: *The remainder of this procedure is performed in the WebLogic user interface, and is only outlined here. Please refer to your BEA WebLogic documentation for current information regarding interface layout and deployment details.*

4 Start the BEA WebLogic server.

5 Navigate to **Server Administration Console > Deployments > Applications**.

6 Perform the following steps:

A Add a new JMS Connection Factory.

B Enter a JNDI name for the JMS Connection Factory:

```
jms\connectionfactory\xa-topic\
     LogicalHostName_MessageServerName
```

For example, the default name would be:

```
jms\connectionfactory\xa-topic\LogicalHost1_WLMessageSvr1
```

C Verify that the WebLogic JMS Server Destination names for topics match those in eGate.

D Select **Deploy a new Application**.

E Upload and install the EAR file described in step 3.

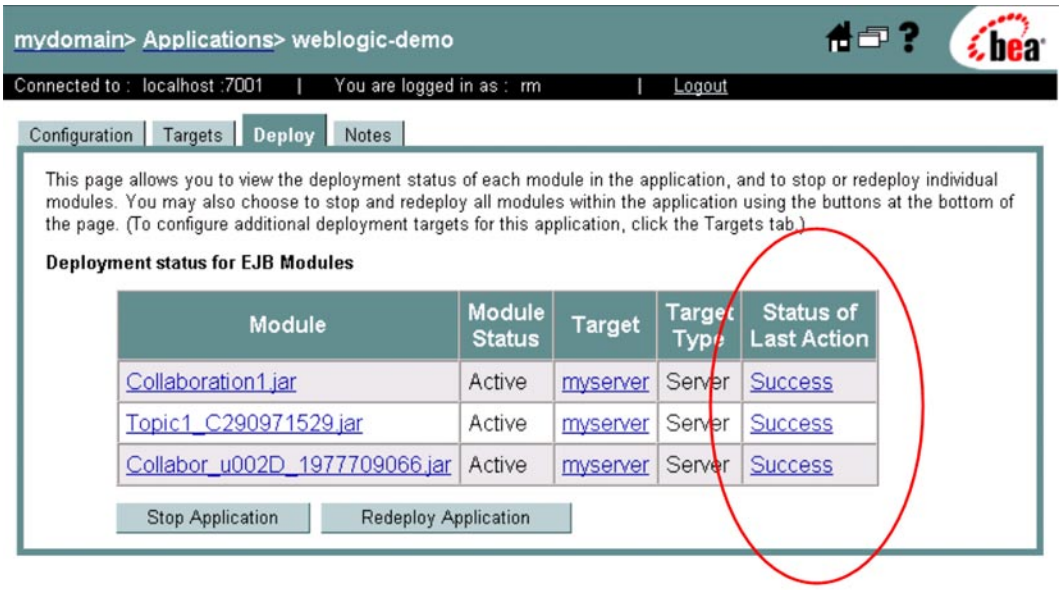
F Select the EAR file you just installed as the archive for the new application.

G Enter a name for the new application.

H Click **Deploy**.

I Verify the success of the deployment (see Figure 131, which shows a WebLogic 8.1 example).

Figure 131 WebLogic Deployment Verification



8.6.2 IBM WebSphere

Note: *Before using the WebSphere JMS, you must install the* ***log4j.jar*** *file. For additional information, see the eGate Integrator JMS Reference Guide.*

To install log4.jar

1 Download the **log4j.jar** file from the location below (this location may change).

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

2 Place the **log4j.jar** file into the **\WebSphere\AppServer\lib** directory.

To deploy an eGate Project to an IBM WebSphere 5.0.x or 5.1 environment

1 Create the following components in Enterprise Designer (see Figure 132):

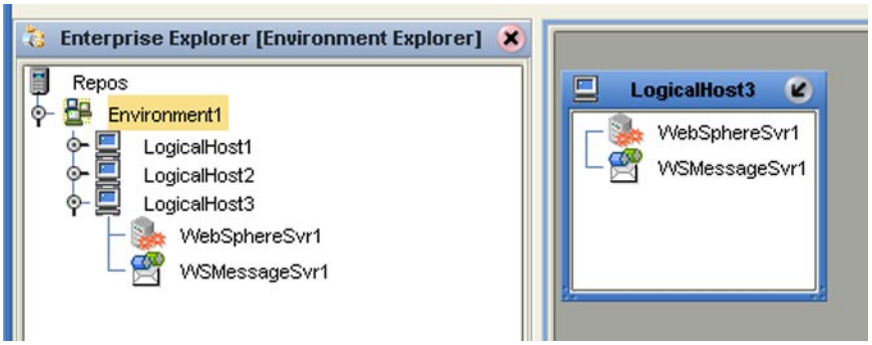
A A new environment

B A Logical Host

C A WebSphere J2EE application server

D A WebSphere JMS message server

Figure 132 WebSphere Deployment (1)

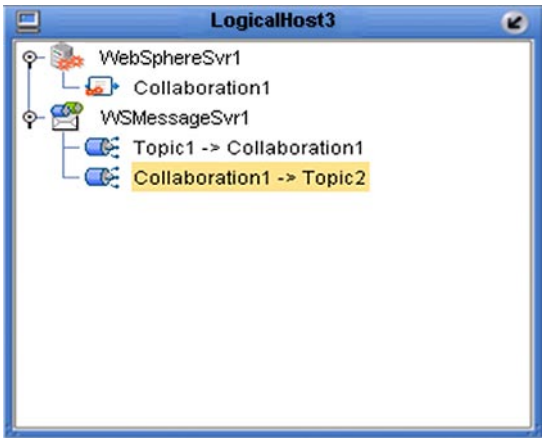


2 Create a new Deployment Profile to bind the Connectivity Map to the new WebSphere environment (see Figure 133).

A Drag the two topics and drop onto the WebSphere message server.

B Drag the Business Process and drop onto the WebSphere application server.

Figure 133 WebSphere Deployment (2)



3 Activate the Deployment Profile.

The activated Deployment Profile creates an Environment Archive (EAR) file, which contains all files necessary to create and run an application in WebSphere. This file can be found in the following location:

```
ICAN-root\repository\data\files\WSEnvironmentName\
    ProjectName_DeploymentProfileName.ear
```

Note: The remainder of this procedure is performed in the WebSphere user interface, and is only outlined here. Please refer to your IBM WebSphere documentation for current information regarding interface layout and deployment details.

4 Start the IBM WebSphere server.

5 From the Administrative Console, navigate to **Servers > Application Servers > *server_name* > Message Listener Service > Listener Ports**.

6 Add a new Listener port.

7 Enter a Connection Factory JNDI name for the new port:

```
jms\connectionfactory\xa-topic\
    LogicalHostName_MessageServerName
```

For example, the default name would be:

```
jms\connectionfactory\xa-topic\LogicalHost1_WSMessageSvr1
```

This binds the JNDI name with the WebSphere Message Server Listener port.

8 From the Administrative Console, navigate to **Applications > Enterprise Applications > Install New Application**.

9 In *Preparing for the application installation*:

A Enter the path for the EAR file described in step 3 and click **Next**.

B Select **Generate Default Bindings** and click **Next**.

10 In *Step 1, Provide options ...*:

A Check **Deploy EJBs**.

B Enter the application name.

C Click **Next**.

11 In *Step 2, Provide options ...*, click **Next**.

12 In *Step 3, Provide Listener Ports ...*, accept the default value and click **Next**.

Note: *The Listener port number should match the port number entered in step 6.*

13 In *Step 4, Provide JNDI Names ...*, accept the default value and click **Next**.

14 In *Step 5, Provide EJB references ...*, accept the default value and click **Next**.

15 In *Step 6, Map resource references ...*, enter the JNDI name from step 7, and click **Next**.

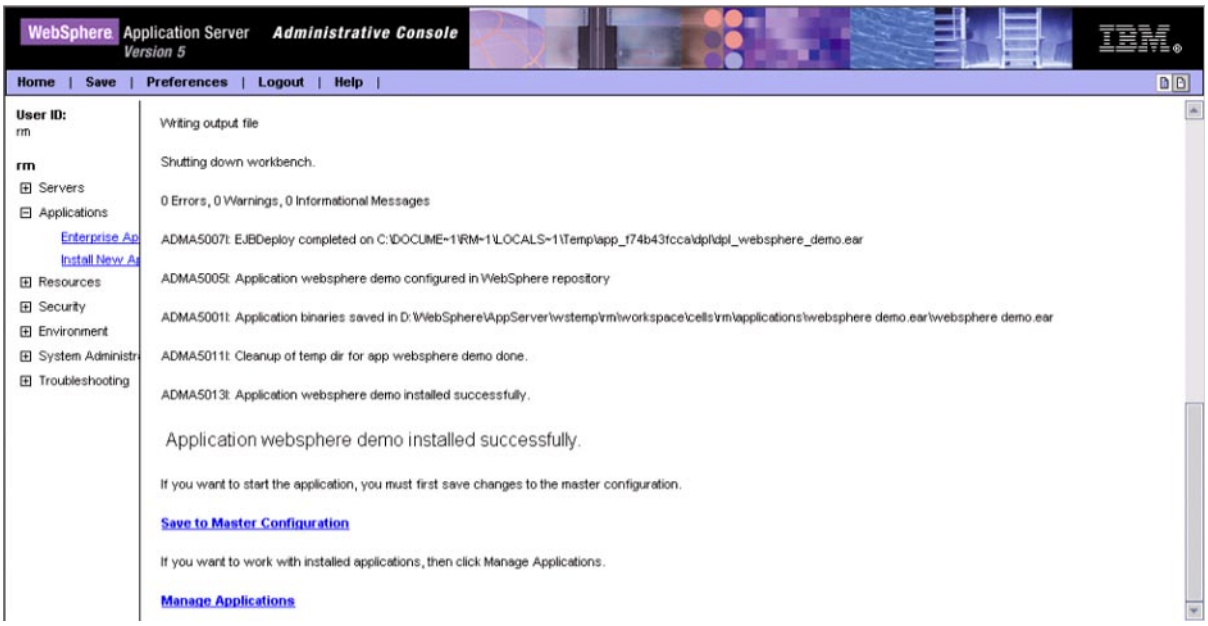
16 In *Step 7, Map modules ...*, check all modules and click **Next**.

17 In *Step 8,* (protection levels), check all modules and click **Next**.

18 In *Step 9, Summary,* click **Finish**.

19 Verify the success of the deployment (see Figure 134, which shows a WebSphere 5 example).

Figure 134 WebSphere Deployment Verification



Chapter 9

Web Services

This chapter describes the use of the Web Services capability of eGate Integrator, acting with other components of the ICAN Suite.

9.1 Overview

Basically, Web Services enables communication and data transfer between diverse applications using the Internet. In doing so, it provides a means for implementing EAI (Enterprise Application Integration) within an organization, or B2B (Business-to-Business) integration between partner organizations. This capability is achieved by wrapping back-end systems to present a common, standardized interface to the connecting network.

Four related technologies are used to transform and transport data within Web Services:

- **XML** (Extensible Markup Language)

  Provides a language for defining both the data itself and the way to process it.

- **WSDL** (Web Services Description Language)

  Defines the interfaces, data types, interactions, and mappings used in the Web Services. WSDL files are used to invoke and operate Web services on the Internet and to access and invoke remote applications and databases.

- **SOAP** (Simple Object Access Protocol)

  Defines a communications envelope that is mappable to HTTP and provides a format for transmitting XML documents over a network.

- **UDDI** (Universal Description, Discovery, and Integration)

  Provides a mechanism for storing and categorizing information that allows publication of services and discovery of external services.

9.2 SeeBeyond Web Services

eGate Integrator provides the capability to create either a client or a server to receive WSDL file from a remote server, or send WSDL files to a remote client. eGate works in conjunction with eInsight Business Process Manager, in which the associated business processes are developed. See **Building a Web Client** on page 174 and **Building a Web Server** on page 181.

The ICAN Suite contains the following components that implement the Web Services capability:

- **WSDL Wizard**

  The WSDL Wizard creates an OTD from a WSDL file. See **Using the WSDL Wizard** on page 101.

- **WSDL Editor**

  See the *eInsight Business Process Manager User's Guide*.

- **WSDL Interface Designer**

  See the *eInsight Business Process Manager User's Guide*.

- **WSDL Viewer**

  See the *eInsight Business Process Manager User's Guide*.

- **UDDI Registry**

  All ICAN objects represented in the Repository that can be accessed as Web services are presented via a UDDI-compliant server. See **UDDI Registry** on page 171.

9.3 UDDI Registry

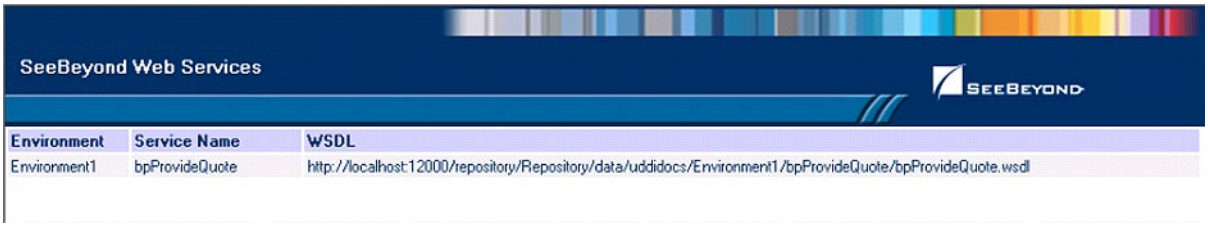
In general, all ICAN objects that expose themselves as Web services (such as eInsight business processes) are listed in a UDDI registry. The deployment activation process automatically publishes entries to the UDDI registry and creates the necessary sections in the WSDL files to expose it as a Web service.

This registry can be viewed on the SeeBeyond Web Services page (see Figure 135), part of Enterprise Manager. The URL of the SeeBeyond Web Services page is:

```
http://hostname:portnumber/stcuddi
```

*Note: The **hostname** is the fully-qualified, network-addressable host name of the server where you installed the Repository. The **portnumber** is the number of the port you entered during installation of the Repository.*

Figure 135 SeeBeyond UDDI Registry



Each entry on the SeeBeyond Web Services page includes:

- The ICAN environment name.
- The actual (Web) Service name.
- The URL for the associated WSDL file.

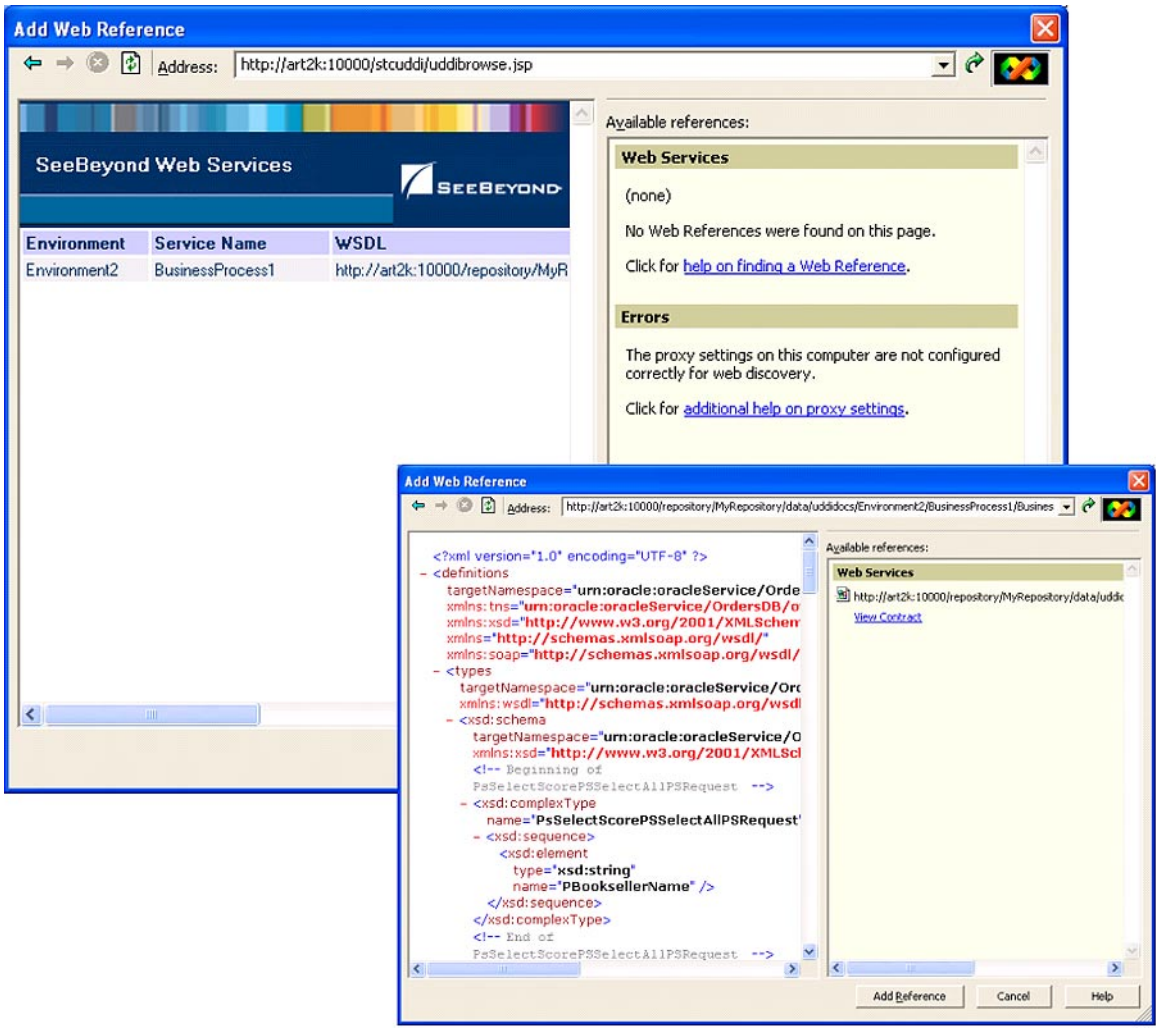
Select an entry (line item) to display its WSDL file, as shown in Figure 136.

Figure 136 Example Web Service WSDL File

```
<?xml version="1.0" encoding="UTF-8" ?>
- <definitions targetNamespace="http://seebeyond/quoteservice" xmlns:tns="http://seebeyond/quoteservice"
  xmlns:sbynpx="http://bpel.seebeyond.com/hawaii/5.0/privateExtension/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
- <types targetNamespace="http://seebeyond/quoteservice">
  - <xsd:schema targetNamespace="http://seebeyond/quoteservice"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema">
      <xsd:element type="xsd:string" name="QuoteRequest" />
      <xsd:element type="xsd:string" name="QuotePrice" />
      <xsd:element type="xsd:string" name="QuoteCompany" />
    </xsd:schema>
  </types>
- <message name="msgProvideQuoteRequest">
    <part name="TICKER" element="tns:QuoteRequest" />
  </message>
- <message name="msgQuoteResponse">
    <part name="price" element="tns:QuotePrice" />
    <part name="company" element="tns:QuoteCompany" />
  </message>
- <portType name="ptProvideQuote">
```

The SeeBeyond UDDI Registry can be used in a third party tool, for example Microsoft Visual Studio (see Figure 137). In Visual Studio's Solution Explorer, right-click on *References* and enter the URL of the SeeBeyond Web Services page as a Web Reference.

Figure 137 Microsoft Visual Studio Example



eGate Integrator can exchange data with Internet and Web Services applications using the Web Services Description Language (WSDL). This language is XML-based and is used to define Web services and describe how to access them. The WSDL OTD Wizard is used to build OTDs that are used in the Project Business Process (see **Using the WSDL Wizard** on page 101).

9.3.1 Using UDDI Browsers

A third-party UDDI Browser can be used to access the SeeBeyond UDDI Registry for publishing and inquiry. To add the SeeBeyond UDDI Registry to the browser, you need to enter the information given in Table 54 into the browser's *Add UDDI Registry* facility.

Table 54 UDDI Registry Information

Parameter	Value
Name	SeeBeyond UDDI
Inquiry URL	http://***hostname:portnumber***/stcuddi/inquiry
Publish URL	http://***hostname:portnumber***/stcuddi/publish
Username	***your_username***
Password	***your_password***

9.4 Building a Web Client

Here we briefly demonstrate the procedure for building a Web client. The steps involved are:

1. Build an Object Type Definition (OTD).
2. Develop a business process.
3. Create the eGate Project.
4. Deploy the Project to the selected Environment.

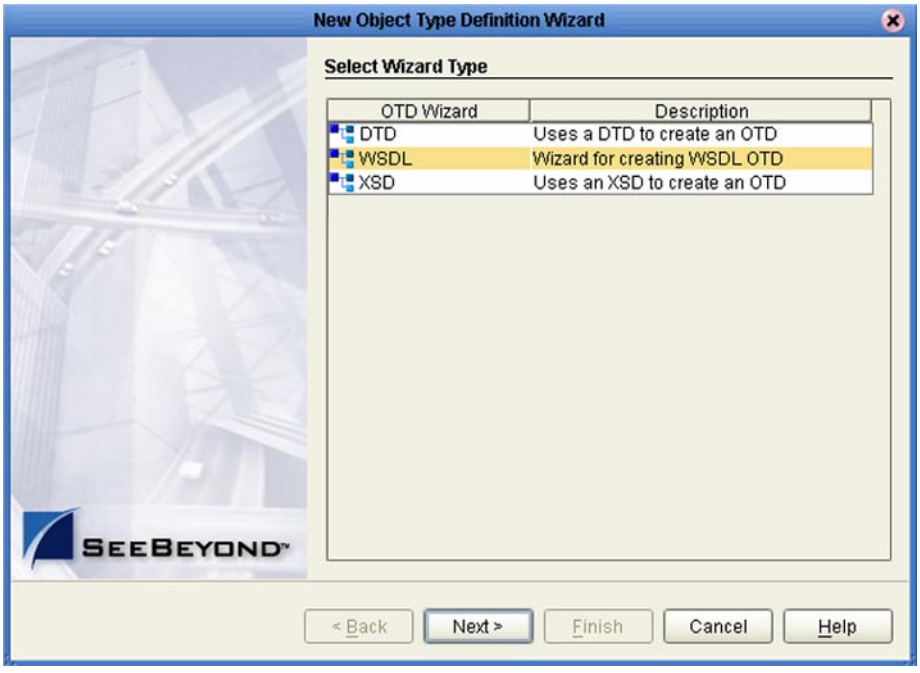
The Project used in the following example is contained in the *eGate User's Guide Sample* listed on the Enterprise Manager's Documentation page. To use this example Project, download the sample file **eGate_User_Guide_Sample.zip** and extract the contents to a convenient directory. Import the file **webclient.zip** into your Repository following the procedure described in **Project/Environment Import** on page 47. The files **input.txt** and **output.txt** are simple text files for testing purposes.

9.4.1 Object Type Definition

To create a Web Client OTD

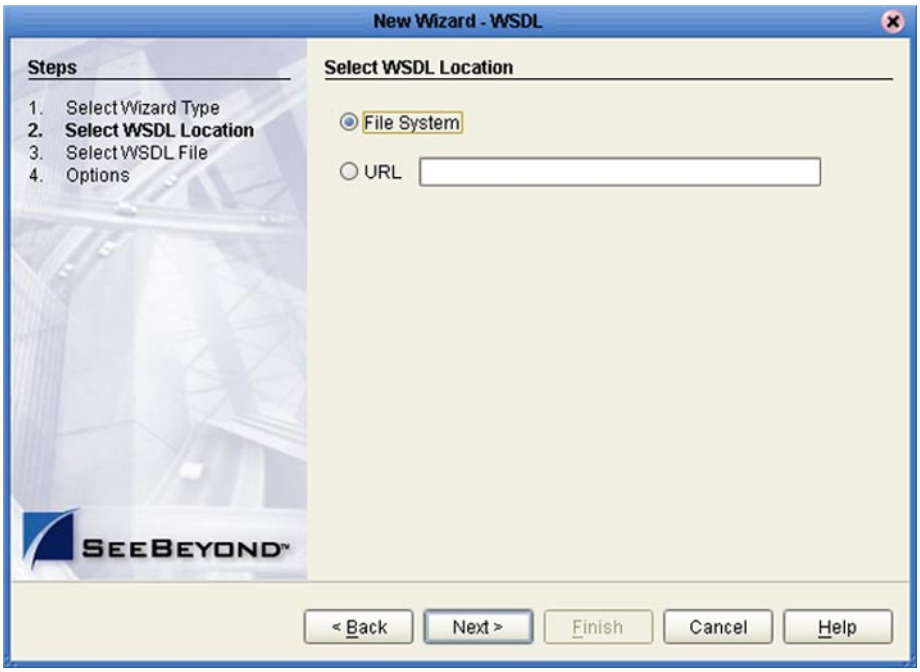
1. Select **New > Object Type Definition** from your Project's context menu.
2. Select **WSDL** from the initial Wizard dialog (see Figure 138) and click **Next**.

Figure 138 Select WSDL Wizard



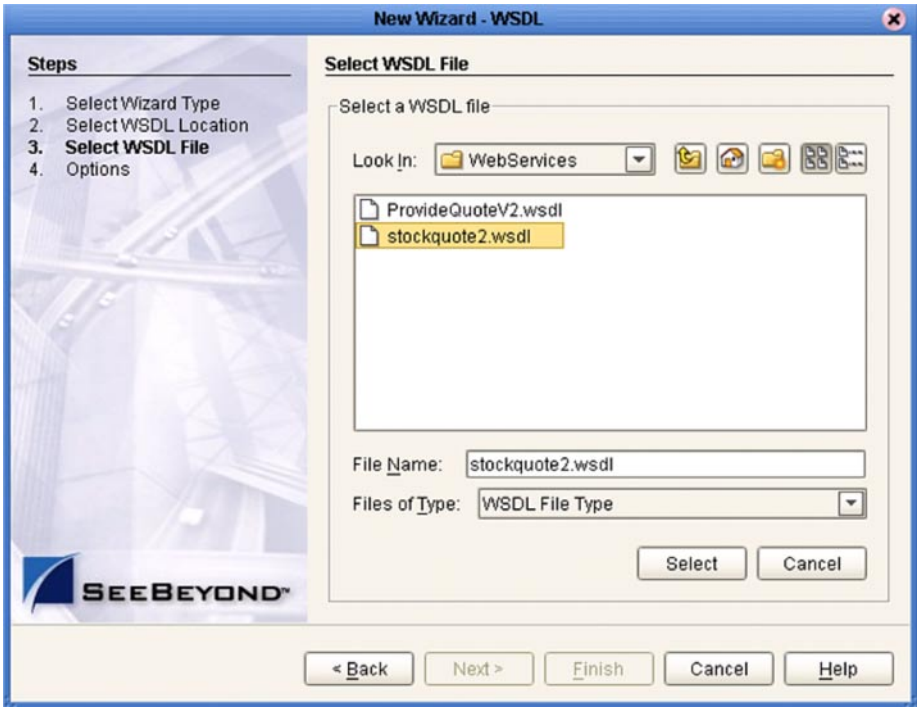
3. Select the WSDL file location; in this example, the local file system (see Figure 139) and click **Next**.

Figure 139 Select File Location



4 Select the WSDL file on which you want to base the OTD (see Figure 140) and click **Next**.

Figure 140 Select WSDL File

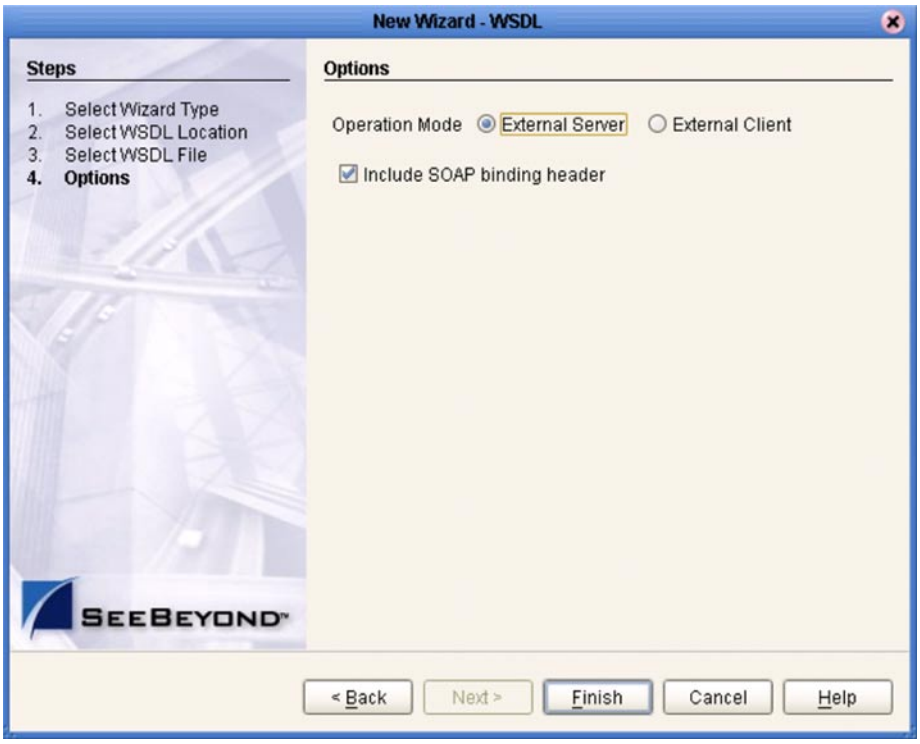


5 For a Web Client, select the following options (see Figure 141):

A Select **External Server** as the Operation Mode.

B Select **Include SOAP binding header** .

Figure 141 Select External Server



6 Click **Finish**.

9.4.2 eInsight Business Process

The example business process, developed in eInsight Business Process Manager, is shown in Figure 142 (see the *eInsight Business Process Manager User's Guide* for details).

Figure 142 Web Client Business Process

The **receive** rule for the business process is shown in Figure 143, and the **write** rule is shown in Figure 144.

Figure 143 Web Client Business Process *Receive* Rule

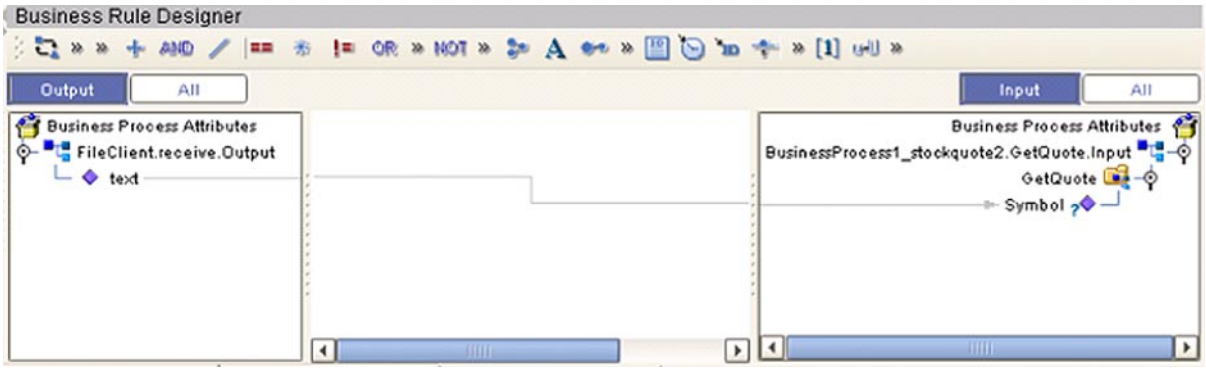
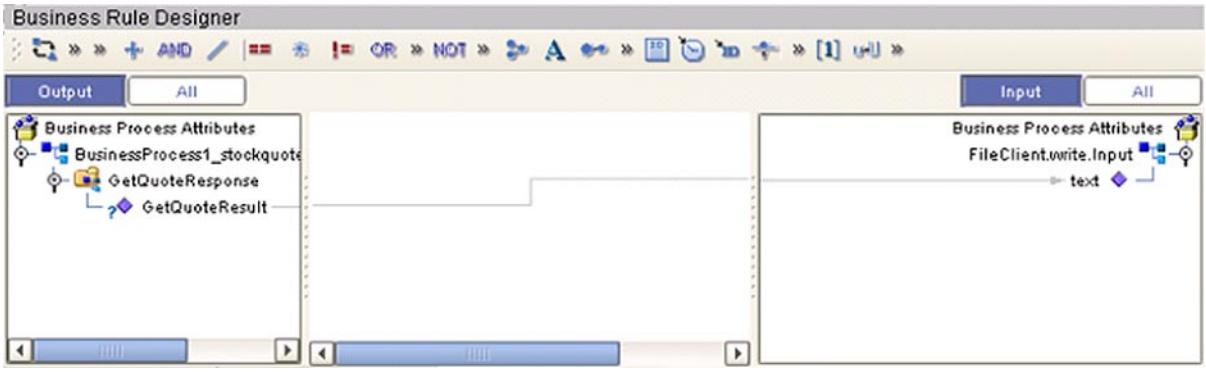


Figure 144 Web Client Business Process *Write* Rule



The WSDL file describing the business process is shown in Figure 145.

Figure 145 Sample WSDL File

```
<process name="BusinessProcess1"
         targetNamespace="http://127.0.0.1:12000/repository/webclient/BusinessProcess1"
         sbynpxp:end_YLoc="123.0"
         sbynpxp:start_YLoc="120.0"
         sbynpxp:linkStyle="angular"
         sbynpxp:start_XLoc="50.0"
         sbynpxp:end_XLoc="508.0"
         xmlns:tns="http://127.0.0.1:12000/repository/webclient/BusinessProcess1"
         xmlns:sbynpx="http://bpel.seebeyond.com/hawaii/5.0/privateExtension/"
         xmlns:slink="ServiceLinkTypes/SeeBeyond/eInsight/e3273l:f8eaf3f6cf:-7fff"
         xmlns:ns0="http://www.webserviceX.NET/"
         xmlns:sbynruntime="http://bpel.seebeyond.com/hawaii/5.0/privateExtension/runtime/"
         xmlns:sbyncreation="http://bpel.seebeyond.com/hawaii/5.0/privateExtension/creation"
         xmlns:nsl="urn:fileservice"
         xmlns:sbynpxp="http://bpel.seebeyond.com/hawaii/5.0/privateExtension/presentation/"
         xmlns="http://schemas.xmlsoap.org/ws/2002/07/business-process/"
         xmlns:sbyntracing="http://bpel.seebeyond.com/hawaii/5.0/privateExtension/tracing/"
         xmlns:sbyninc="http://bpel.seebeyond.com/hawaii/5.0/privateExtension/incompleteModel">
    <!-- partners definition -->
    <partners>
        <partner name="BusinessProcess1_stockquote2"
```

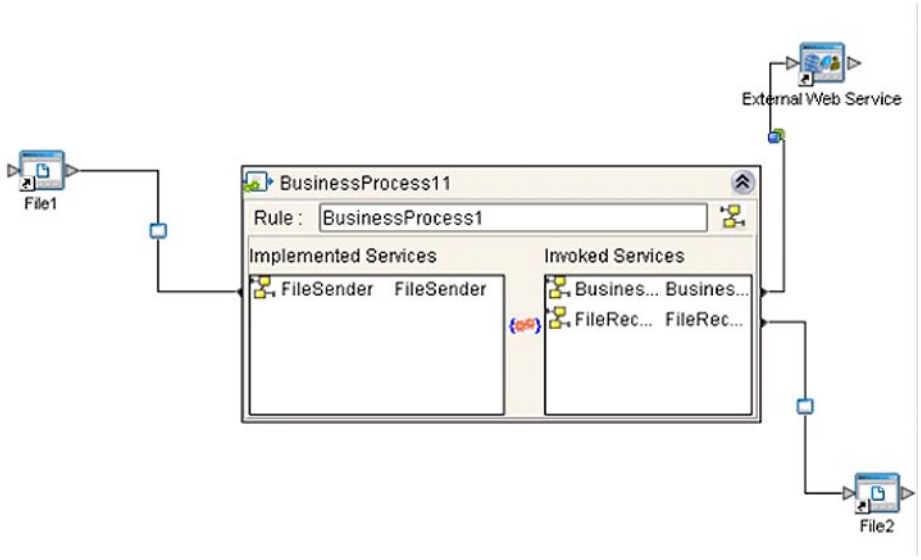
9.4.3 eGate Project

The Project components are created and mapped in the Enterprise Designer Connectivity Map Editor. The example Project contains:

- Two external files and accompanying File eWays.
- An External Web Service.
- A service, into which you drag and drop the eInsight business process from the Project Explorer.

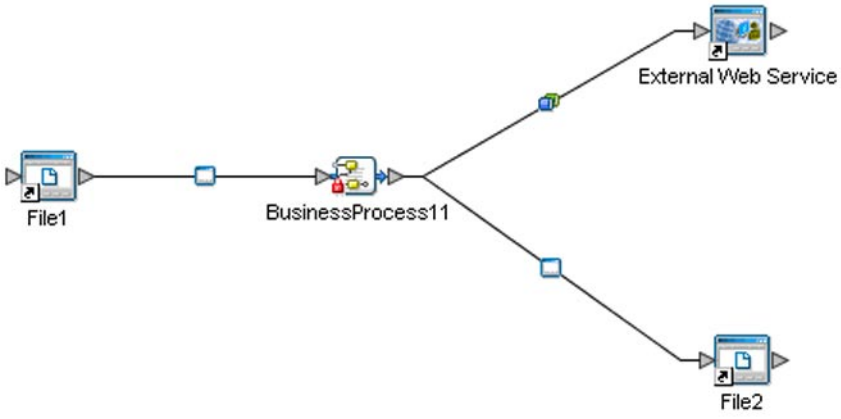
The business process is connected as shown in Figure 146.

Figure 146 Map Business Process



The completed Connectivity Map for the example Project is shown in Figure 147.

Figure 147 Web Client Connectivity Map



The Web client example Project appears in the Project Explorer as shown in Figure 148.

Figure 148 Web Client Example Project

The example Project is deployed to a run-time Environment as shown in Figure 149.and Figure 150

Figure 149 Web Client Deployment (1)

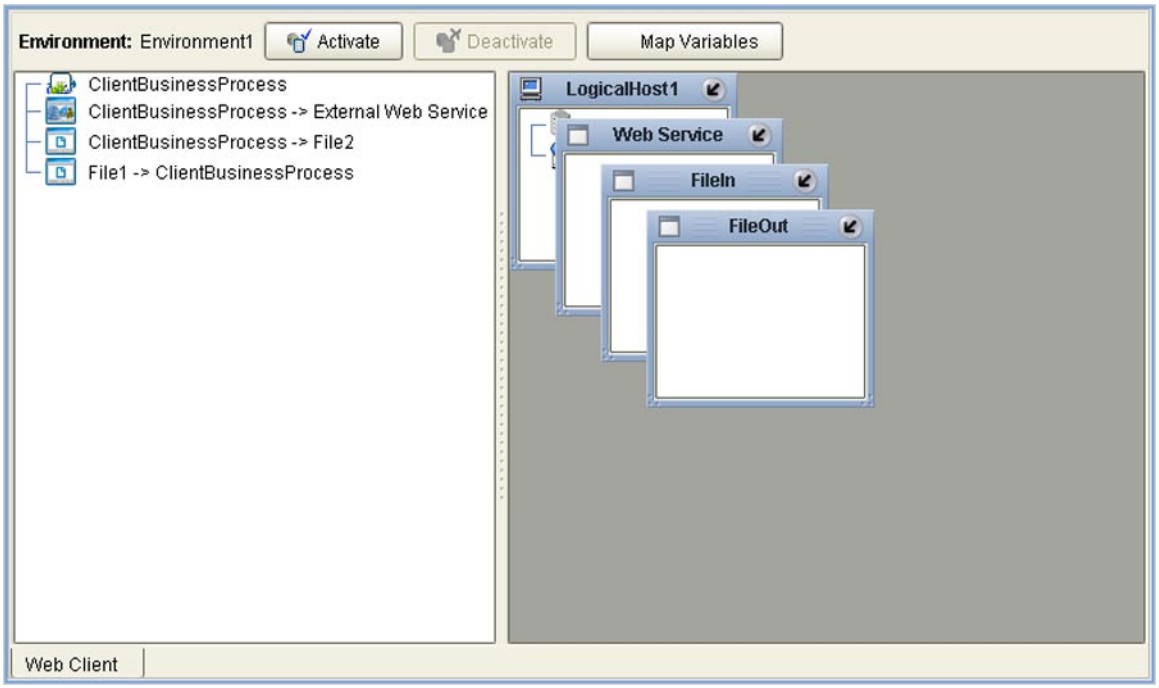
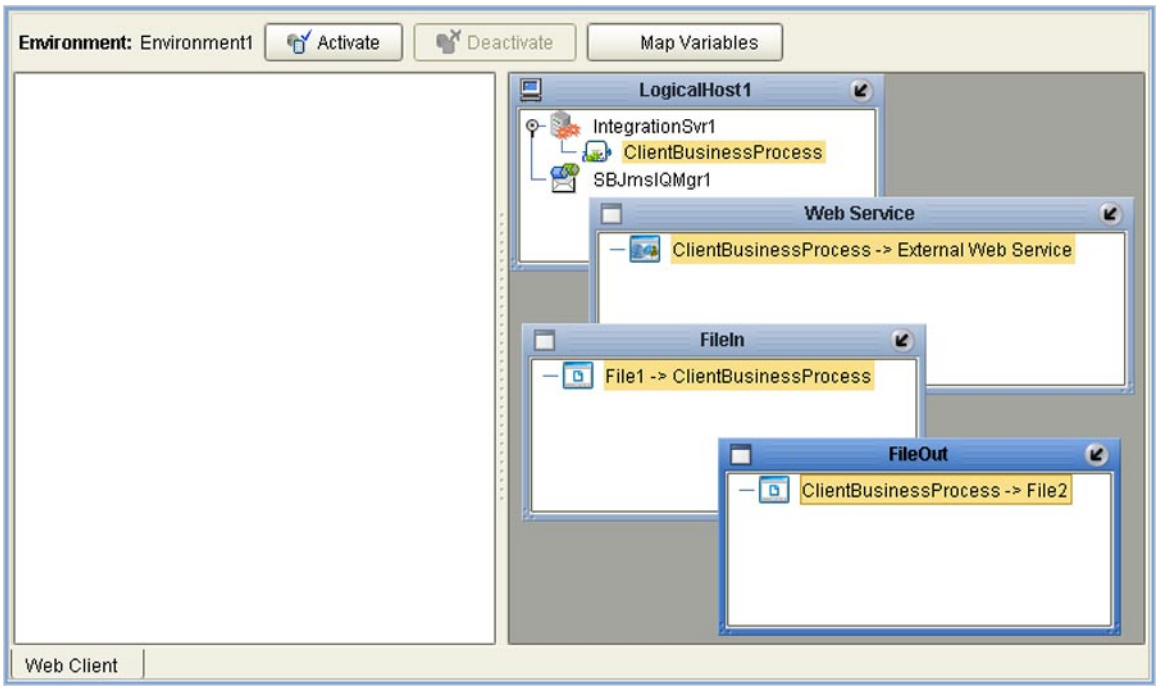


Figure 150 Web Client Deployment (2)



9.5 Building a Web Server

Here we briefly demonstrate the procedure for building a Web server. As with the Web client, the steps involved are:

1. Build an Object Type Definition (OTD).
2. Develop a business process.
3. Create the eGate Project.
4. Deploy the Project to the selected Environment.

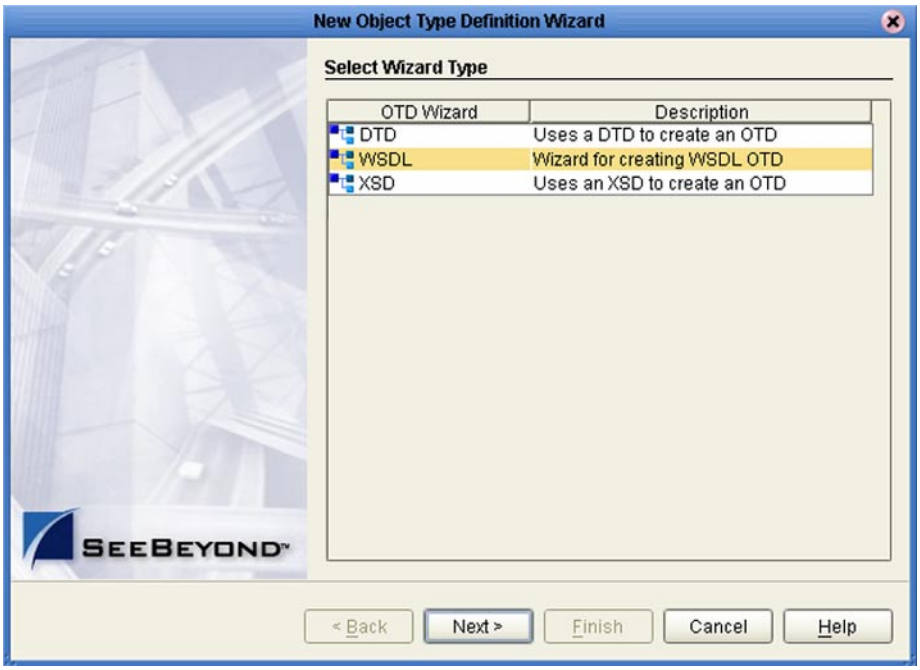
The Project used in the following example is contained in the *eGate User's Guide Sample* listed on the Enterprise Manager's Documentation page. To use this example Project, download the sample file **eGate_User_Guide_Sample.zip** and extract the contents to a convenient directory. Import the file **webserver.zip** into your Repository following the procedure described in **Project/Environment Import** on page 47.

9.5.1 Object Type Definition

To create a Web Server OTD

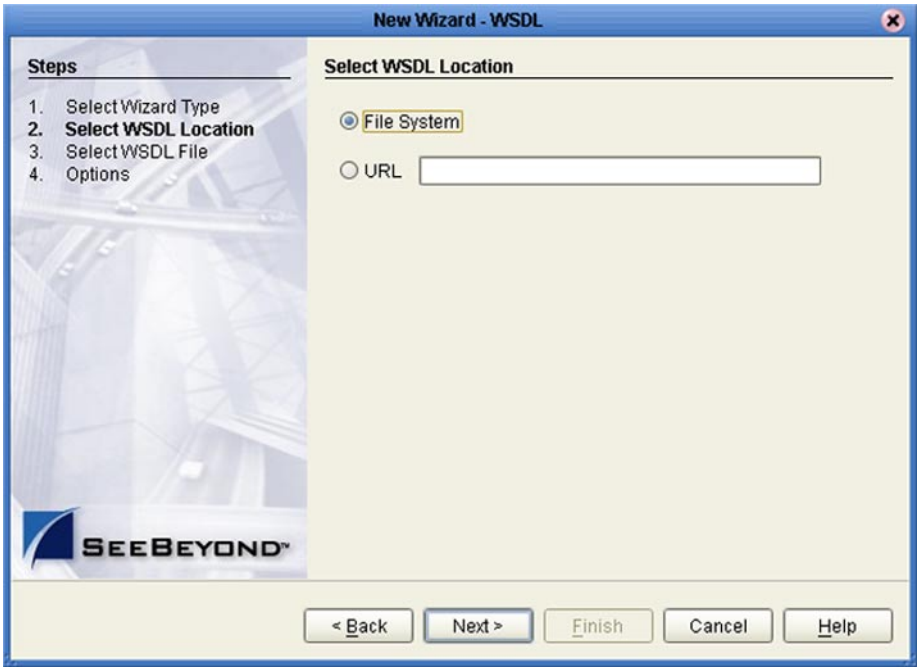
1. Select **New > Object Type Definition** from your Project's context menu.
2. Select **WSDL** from the initial Wizard dialog (see Figure 151) and click **Next**.

Figure 151 Select WSDL Wizard



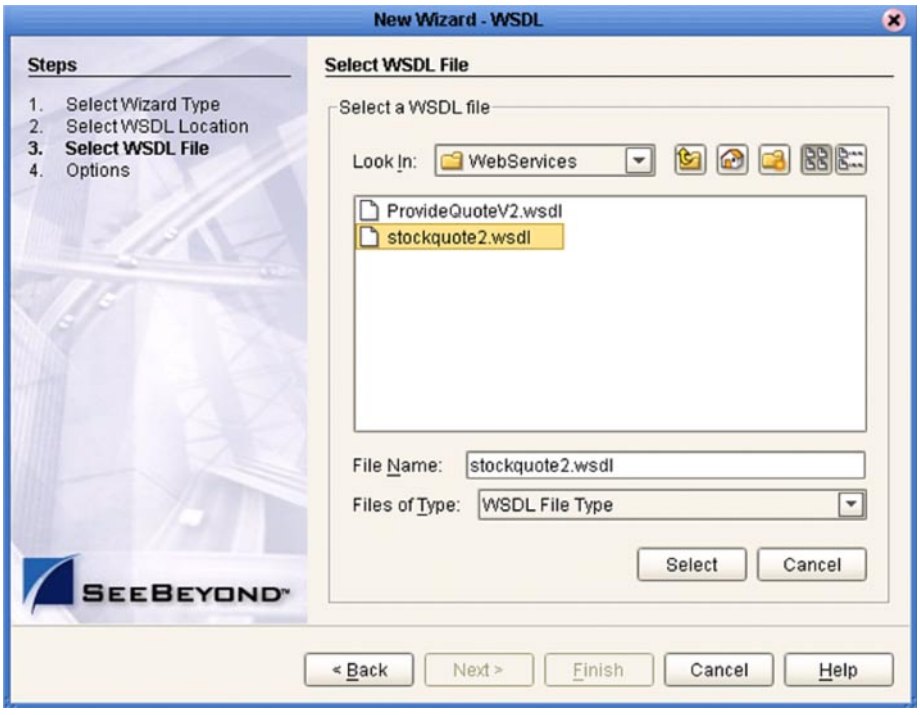
3 Select the WSDL file location; in this example, the local file system (see Figure 152) and click **Next**.

Figure 152 Select File Location



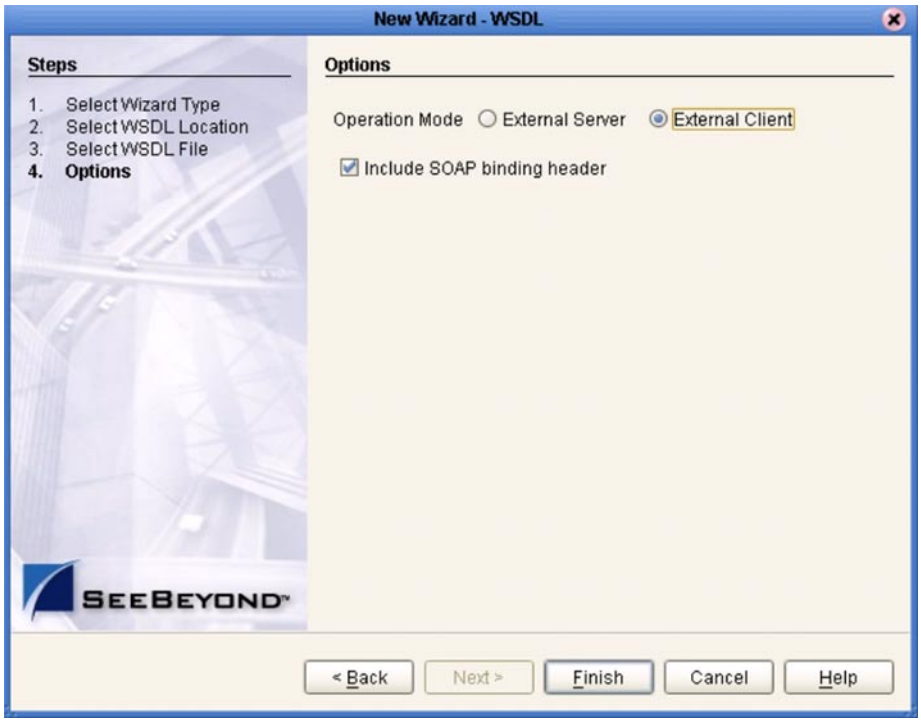
4 Select the WSDL file you want to use for the OTD (see Figure 153) and click **Next**.

Figure 153 Select WSDL File



5 For a Web server, select External Client (see Figure 154) and click **Finish**.

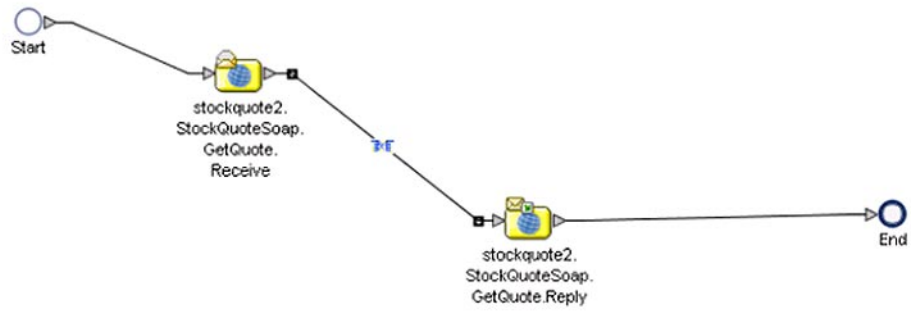
Figure 154 Select External Client



9.5.2 eInsight Business Process

The example business process, developed in eInsight Business Process Manager, is shown in Figure 155 (see the *eInsight Business Process Manager User's Guide* for details).

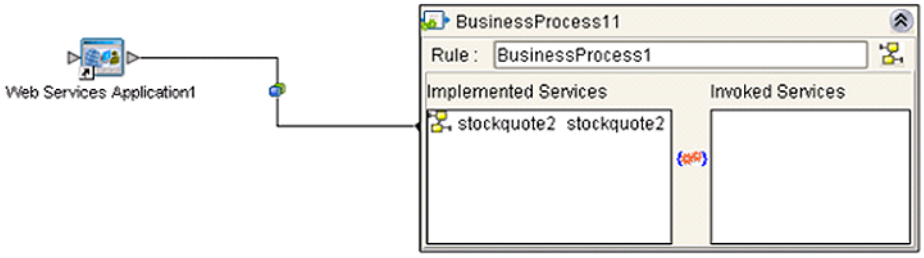
Figure 155 Web Server Business Process



9.5.3 eGate Project

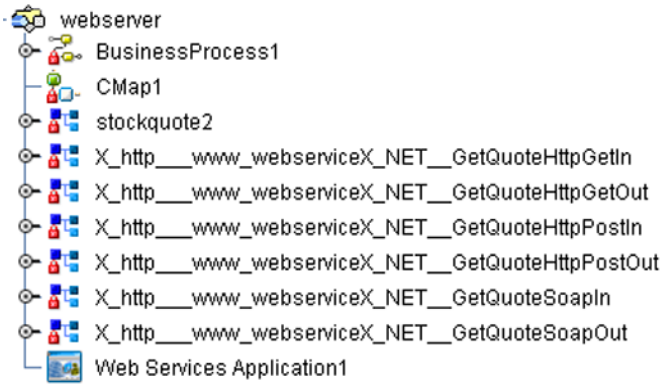
The business process is connected as shown in Figure 156, using the Enterprise Designer Connectivity Map Editor.

Figure 156 Connectivity Map



The Web server example Project appears in the Project Explorer as shown in Figure 157.

Figure 157 Web Server Example Project



The Project is deployed to a run-time Environment as shown in Figure 158 and Figure 159.

Figure 158 Web Server Deployment (1)

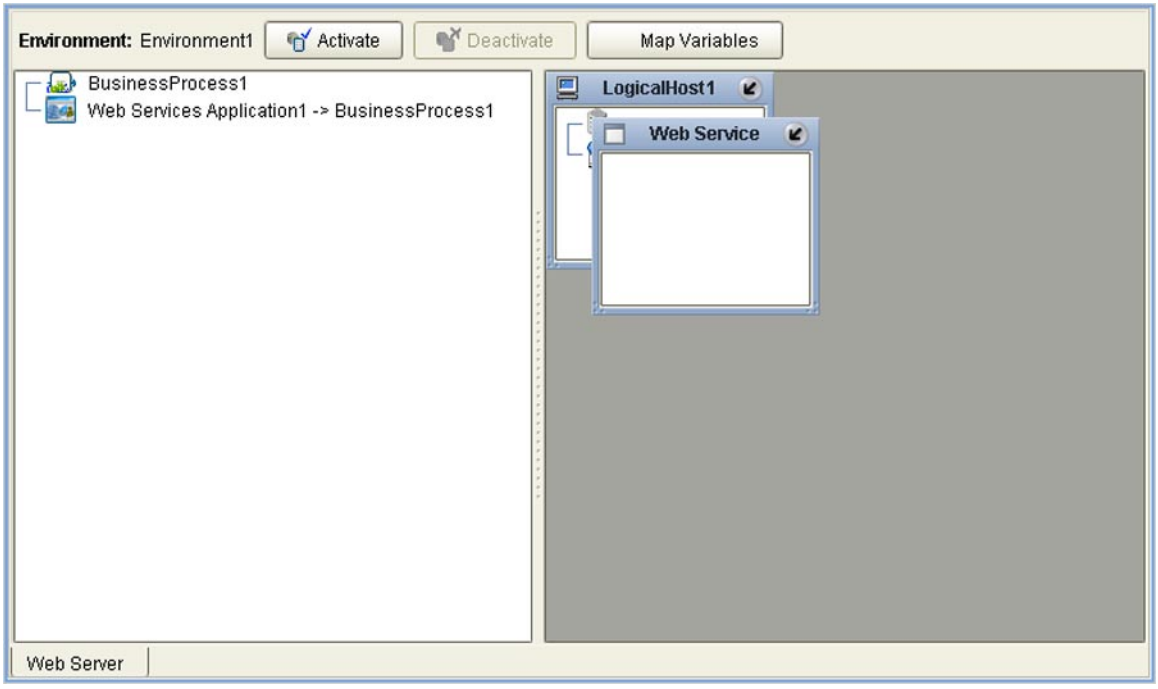
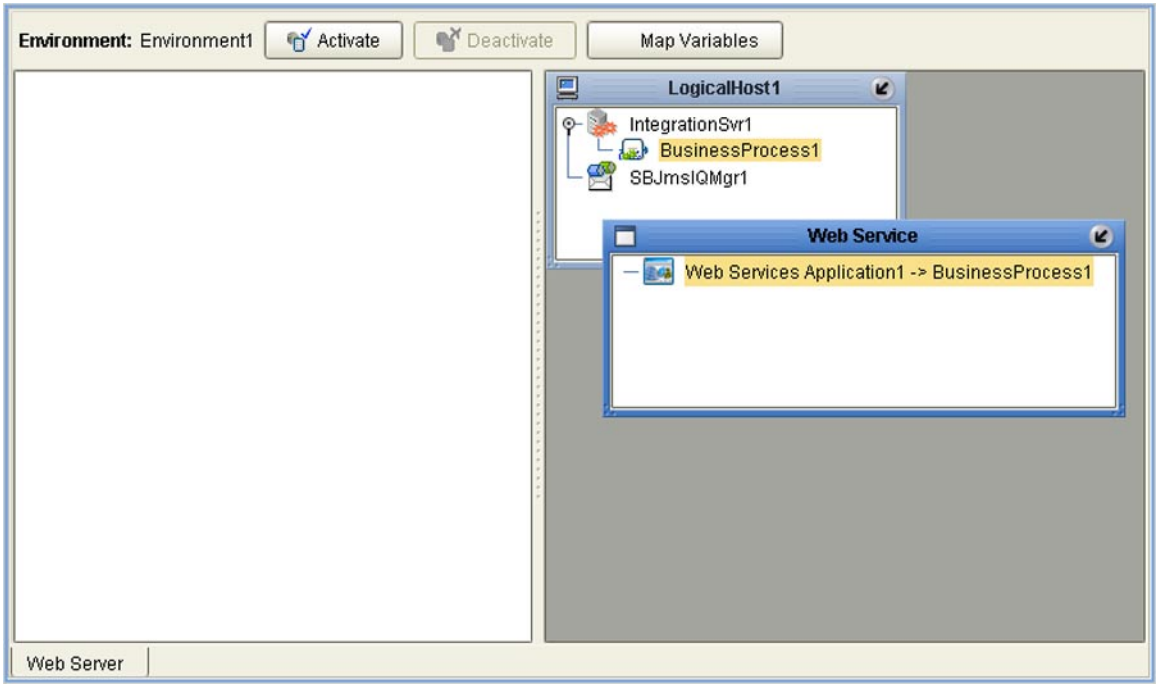


Figure 159 Web Server Deployment (2)



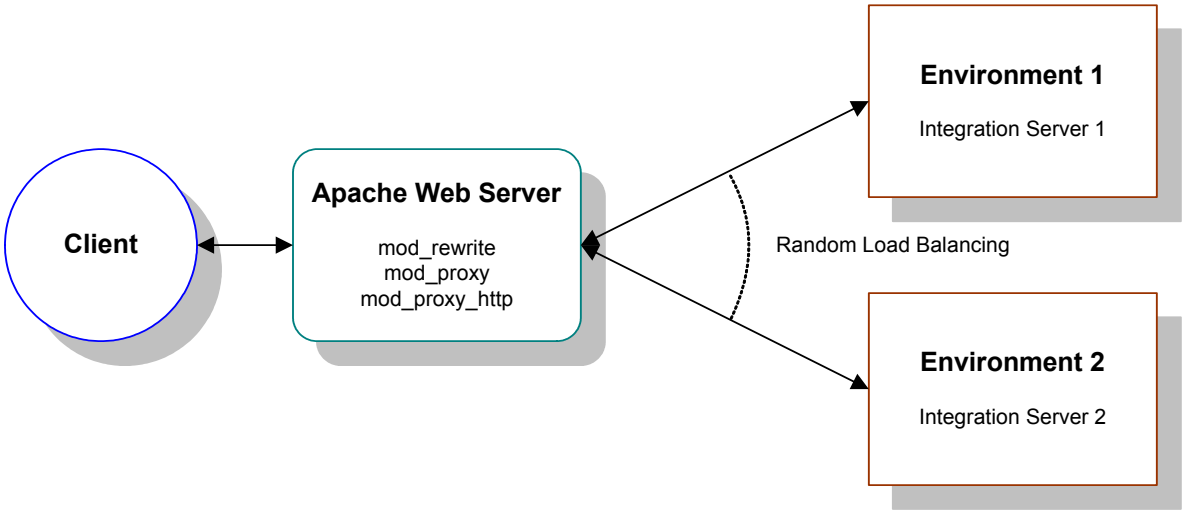
9.6 Load Balancing

You can build a system that will load balance HTTP requests in a way suitable for ICAN Web Services by using an Apache-based Web server. To accomplish this you will need the following:

1. An Apache-based HTTP Web server such as Apache 2.0 or IBM HTTP 2.0.
2. Modify the Apache configuration file to enable **mod_rewrite**, **mod_proxy**, and **mod_proxy_http**.
3. Create a mapping file that includes the name of your business process that is published as a Web Service and a list of URL endpoints where it is available.
4. Add the **mod_rewrite** rewrite rule to your Apache configuration file.

Note: *This scheme only provides simple random load-balancing. It does not provide fail-over nor does it provide any heuristics or any other ability to distribute work based on server load.*

Figure 160 Load Balancing Example



9.6.1 Configuring the Apache Server

The main configuration file for Apache, **httpd.conf**, is located in *APACHE*/config. To enable **mod_proxy**, **mod_proxy_http**, and **mod_rewrite**, uncomment the following lines shown in bold text (these options are usually disabled—commented out—by default).

```
#LoadModule mime_magic_module modules/mod_mime_magic.so
LoadModule proxy_module modules/mod_proxy.so
#LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule proxy_http_module modules/mod_proxy_http.so
#LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
LoadModule rewrite_module modules/mod_rewrite.so
LoadModule setenvif_module modules/mod_setenvif.so
```

9.6.2 Creating the Mapping File

Before you create the Web Services and the mapping file you need to be aware of the mapping rule that will be used. The URL generated for the SeeBeyond Web Services endpoint is of the following form (you must rename the ***bold_italic*** section in the servlet context property as described in **Web Service Application** on page 79):

```
WSServlet_WSCalc_u02F_IS1_WSC_WSCALC_bpCalc1/STCService
```

The mapping file (**ws_server.txt**.) uses the business process name as the key to find a list of URL endpoints on which that service will be available. This file is a simple text file of the following form:

```
Key alternate_URL1|alternate_URL2
```

The content below is a typical example—however, it should actually appear on a single line with no space between the alternate URLs. The alternate URLs contain the actual endpoints for the Web Services, including the host IP address and port.

```
bpCalc1
     127.0.0.1:18004/WSServlet_WSCalc_u02F_IS1_WSC_WSCALC_bpCalc1/STCService|
     127.0.0.1:19004/WSServlet_WSCalc_u02F_IS2_WSC_WSCALC_bpCalc1/STCService
```

The mapping file must be stored on the same system that is running Apache, and is usually found in the configuration directory, for example:

```
C:\Program Files\IBM HTTP Server 2.0\conf\ws_server.txt.
```

9.6.3 Adding the Rewrite Rule

The code section below is the rewrite rule; lines beginning with '**#**' are comments:

```
# Start of ICAN rewrite rules
<IfModule mod_rewrite.c>
#RewriteLog "C:\Program Files\IBM HTTP Server 2.0\logs\rewrite.log"
#RewriteLogLevel 5
RewriteMap SERVICES "rnd:C:\Program Files\IBM HTTP Server
2.0\conf\ws_server.txt"

<Location ~ /WSServlet_ >
    RewriteEngine On

    RewriteRule "([^/_]+(?=/STCService))" "http://${SERVICES:$1}" [P,L]
    RewriteRule "(.*)" "http://localhost:80/notfound.html" [G]
</Location>
</IfModule>
# End of rewrite directives.
```

The purpose of this rule is to intercept requests for ICAN-formatted URLs of the form

```
http://LOADBALANCER:9999/WSServlet_XXXXXXX_bpCalc1/STCService
```

and rewrite them to a form using the entries in the mapping file **ws_server.txt**. It does this by extracting the key (shown in bold) from the requested URL. This key is the last element of the variable part of the endpoint URL created by ICAN.

- If it cannot find a suitable match, it will return a *gone* (410) error.
- If it finds a suitable key but does not find a match in the mapping file, it will return a *not found* (404) error.
- If a key is matched but the service is not available on that URL, it will return a *bad gateway* (502) error.

9.6.4 Debugging

There are a number of ways to debug the internal processing of Apache, the rewrite logging probably being the most useful. You can specify an additional rewrite log and a log level within the rewrite section of the **httpd.conf** file. The logging level is on a scale of 0 to 5, with 0=off and 5=maximum debugging.

An example of a successful rewrite request log is shown below, using log level 5 (timestamps omitted):

```
[rid#596d60/initial] (3) [per-dir /WSServlet_/] add path info postfix: C:/
     Program Files/IBM HTTP Server 2.0/htdocs/en_US/WSServlet_WSCalc_
     u02F_IS1_WSC_WSCALC_bpCalc1 -> C:/Program Files/IBM HTTP Server 2.0/
     htdocs/en_US/WSServlet_WSCalc_u02F_IS1_WSC_WSCALC_bpCalc1/STCService

[rid#596d60/initial] (3) [per-dir /WSServlet_/] applying pattern '([^/_]+(?=/
     STCService))' to uri 'C:/Program Files/IBM HTTP Server 2.0/htdocs/en_US/
     WSServlet_WSCalc_u02F_IS1_WSC_WSCALC_bpCalc1/STCService'

[rid#596d60/initial] (5) cache lookup OK: map=SERVICES[txt] key=bpCalc1 ->
     val=127.0.0.1:18004/WSServlet_WSCalc_u02F_IS1_WSC_WSCALC_bpCalc1/
     STCService

[rid#596d60/initial] (5) randomly choosen the subvalue `127.0.0.1:18004/
     WSServlet_WSCalc_u02F_IS1_WSC_WSCALC_bpCalc1/STCService'

[rid#596d60/initial] (2) [per-dir /WSServlet_/] rewrite C:/Program Files/IBM
     HTTP Server 2.0/htdocs/en_US/WSServlet_WSCalc_u02F_IS1_WSC_WSCALC_
     bpCalc1/STCService -> http://127.0.0.1:18004/WSServlet_WSCalc_u002F_
     IS1_WSC_WSCALC_bpCalc1/STCService

[rid#596d60/initial] (2) [per-dir /WSServlet_/] forcing proxy-throughput with
     http://127.0.0.1:18004/WSServlet_WSCalc_u02F_IS1_WSC_WSCALC_bpCalc1/
     STCService

[rid#596d60/initial] (1) [per-dir /WSServlet_/] go-ahead with proxy request
     proxy:http://127.0.0.1:18004/WSServlet_WSCalc_u02F_IS1_WSC_WSCALC_
     bpCalc1/STCService [OK]
```

Glossary

BPEL
BPEL (Business Process Execution Language), also known as BPEL4WS (Business Process Execution Language for Web Services), is an XML-based language designed to enable task sharing for either a distributed or grid computing environment. It combines and replaces **WSDL** and Microsoft's XLANG specification.

Collaboration
A logical operation performed between some combination of message destinations and external applications. The operation is defined by a Collaboration Definition (see next entry), which can be encoded in either Java or **XSLT**.

Collaboration Definition
The encoding of business rules, in Java or XSLT format. Typically, the encoding consists of operations on an **Object Type Definition (OTD)**. Several Collaborations can have the same Collaboration Definition.

Connection
Consists of the configuration information that enables an eWay to connect to an external system.

Connectivity Map
Contains business logic and routing information about the data transmission. A Connectivity Map usually includes one or more **Collaboration**s, **Topic**s, **Queue**s, and **eWay**s. A Connectivity Map is created under a **Project**. A Project may have multiple Connectivity Maps.

Constant
A static name-value pair that is visible across a **Project**.

CRM
Acronym for Customer Relations Management.

Data Cleansing
Refers to transforming data for accurate and effective use in a database or data management system by cleansing “dirty” or redundant data. Data must be cleansed of errors in structure and content before it is useful in data warehousing and integration.

Data Dictionary
Defines the organization of a database and lists all files in the database, the number of records in each file, and the names and types of each field. Although the dictionary

does not contain actual data, it does contain essential information for managing the database. The data dictionary is often hidden from end users

Data Integrity
Refers to the accuracy and validity of data. Data integrity can be compromised in many ways, including human error through data entry, or through faulty logic in programming. Computer viruses, software bugs and many other factors can also compromise data integrity.

Data Mapping
Refers to establishing the relationship and data flow pattern between source and target objects, usually within the context of relational database management systems (RDBMSs).

Data Transformation
Data transformation is necessary after extracting data from legacy data formats, or any format that requires **Data Cleansing**. Data is transformed for efficient use for Business-to-Business Enterprise Data Integration.

DBCS
Acronym for Double-Byte Character Set.

Deployment Profile
Contains the information about how the **Project** components will be deployed in an Environment. A Project can have multiple Deployment Profiles, but only one Deployment Profile can be activated for a Project in any one **Environment**.

Derived Collaboration
A Collaboration that inherits operations from another, according to standard object-oriented practice.

DTD
A Document Type Definition (DTD) specifies how an associated document, written either in the Standard Generalized Markup Language (SGML) or of the Extensible Markup Language (**XML**), is to be processed.

Enterprise Designer
The **Project** design tool within eGate Integrator.

Enterprise Service Bus (ESB)
A category of software, incorporating native Web services support, that provides a low-end alternative to a comprehensive integration broker suite—offering limited functionality, but less complexity and lower cost.

Environment
A collection of physical resources and their configurations that are used to host eGate **Project** components. An Environment contains Logical Hosts and external systems.

ERM
Acronym for Enterprise Resource Management.

ETL
A three-phase (extract/transform/load) process used, for example, to generate and maintain a central **Metadata** repository.

- **Extract** is the process of reading data from a source database and obtaining the desired subset of data.
- **Transform** is the process of converting the extracted data from its previous form into the desired form.
- **Load** is the process of writing the transformed data into the target database.

eWay
A link between a **Collaboration** and an external connection including the message server connection (topic or queue) or external application.

External Application
A logical representation of an application external to the ICAN Suite.

External System
A representation of a computer system hosting an application external to the ICAN Suite.

Extraction
The process of reading data from a source database and obtaining the desired subset of data (see **ETL**).

HTML
HTML (HyperText Markup Language) is the set of markup symbols or codes (tags) inserted in a file intended for display on a Web page. HTML describes the content of the Web page (primarily text and graphics) only in terms of how it is to be displayed and interacted with.

HTTP
HTTP (HyperText Transfer Protocol) is the set of rules for transferring files—text, graphics, audio, video—on the World Wide Web.

ICAN Suite
The SeeBeyond Integrated Composite Application Network Suite.

Impact Analyzer
A module within Enterprise Designer that analyzes and predicts the impact a specified change would have on other components in the **Project**.

Integration Server
J2EE-compliant software platform that houses the business logic container used to run Collaborations and JCA connectors (eWays). Provides transaction services, persistence, and external connectivity.

JMS IQ Manager
JMS-compliant, guaranteed delivery store, forwarding, and queueing service.

Link
The JMS Connection between a **Collaboration** and a topic or queue in a JMS-compliant message server.

Linked Message Destination
A reference to a **Message Destination** defined in another **Connectivity Map**.

Logical Host
A Logical Host contains the eGate run-time components, including integration servers and message servers, that are installed on a host hardware platform.

Management Agent
Uses J2EE technology to manage and monitor an eGate 5.0 deployment that may contain other application servers in addition to the SeeBeyond Integration Server. Defines management interfaces and services designed for distributed environments, focusing on providing functionality for managing networks, systems, and applications.

Message Destination
A general term for a topic or queue. Two or more Projects can share a message destination that has the same name and is deployed on the same message server. A single Project may also have a single message destination referenced in multiple **Connectivity Map**s.

Metadata
Metadata describes the structure and format of a particular set of data. **ETL** tools are used to generate and maintain a central metadata repository.

Non-normalized Data
Data that has not been converted to normalized data (see next entry).

Normalized Data
Data that has been processed to remove redundant or incorrect data structure and organization, thereby creating a maintainable data set that can be cross-referenced. Normalized data is not only easier to analyze but also easier to expand.

Object Type Definition (OTD)
Object Type Definitions contain the data structure and rules that define an object. OTDs are used in **Collaboration Definition**s for creating data transformations and interfacing with external systems.

Project
Contains a collection of logical components, configurations, and files that are used to solve business problems. A Project organizes the files and packages and maintains the settings that comprise an eGate system in SeeBeyond's **Enterprise Designer**.

Query
A request for information from a database. There are three basic query methods:

- **Choose** – With this simplified method, the database system presents a list of parameters from which you can choose. This method is the least flexible of the three methods.
- **Query by example (QBE)** – With this method, the system lets you specify fields and values to define a query.
- **Query language** – With this method, you have the ability to make requests for information in the form of a stylized query using a query language. This is the most complex and powerful of the three methods.

Queue

A JMS queue is a shareable object that conforms to the *point-to-point* (p2p, or PTP) messaging domain, where one sender delivers a message to exactly one receiver. When the SeeBeyond **JMS IQ Manager** sends a message to a queue, it ensures it is received once and only once, although there may be many receivers "listening" to the queue. This is equivalent to the subscriber pooling in other queue implementations. You can reference a queue that exists in another **Connectivity Map** or **Project**.

Raw Data

Unprocessed data, as obtained from the source (see also **Non-normalized Data**).

Relational Database (RDBMS)

Short for Relational Database Management System, most often referred to as RDBMS, in which data is stored in related tables and can be viewed in many different ways. Relational databases differ from flat-file databases, in which each database is self-contained as a single file or table.

Repository

Stores and manages the setup, component, and configuration information for eGate **Project**s. The Repository also provides monitoring services for Projects, which include version control and impact analysis.

SBCS

Acronym for Single-Byte Character Set.

Schema Runtime Environment (SRE)

An add-on feature in eGate 5.0 that allows **Collaboration**s developed in e*Gate 4.x to be used in and controlled from eGate 5.0, thereby providing an interim upgrade path for e*Gate 4.x users.

Service

Contains the information about executing a set of business rules. These business rules can be defined in a Java or XSLT **Collaboration Definition**, Business Process, eTL Definition, or other service. A Service also contains binding information for connecting to JMS **Topic**s, **Queue**s, **eWay**s, and other services.

SI/SO

Acronym for Shift-In/Shift-Out.

SOAP

SOAP (Simple Object Access Protocol) enables a program running in one operating system to communicate with another program running in either the same or a different operating system, using **HTTP** and **XML** as the mechanisms for information exchange.

Subproject

An independent **Project** that is included as part of another Project, and is displayed in the Enterprise Explorer tree as a "branch" beneath the main Project.

Table

Refers to data arranged in rows and columns, as in a spreadsheet. In **Relational Database (RDBMS)** systems, all information is stored in tables.

Topic

A JMS topic is a shareable object that conforms to the *publish-and-subscribe* (pub/sub) messaging domain, where one publisher broadcasts messages to one or more subscribers. When the SeeBeyond **JMS IQ Manager** publishes a message on a topic, it ensures that all subscribers receive the message.

Transformation

The process of converting the data extracted from its source into the form required by its target program or system. This process includes **Data Cleansing**, **Data Mapping**, data normalization (see **Normalized Data**), and other sub-processes.

UDDI

UDDI (Universal Description, Discovery, and Integration) is an **WSDL**-based registry that enables businesses to list themselves and their services on the Internet.

Version Control

Features that maintain the integrity of a program or **Project** by controlling the ability of an individual to modify the program or Project, and providing an audit trail for accepted modifications.

WSDL

WSDL (Web Services Flow Language) is an **XML**-based language, derived from **SOAP**, used to describe the services a business offers via the Internet. WSDL provides the means of expressing business services in the **UDDI** registry.

XML

XML (Extensible Markup Language) is a superset of **HTML**, which also describes the content in terms of what data is being described. XML is *extensible* because—unlike HTML—the markup symbols are unlimited and self-defining.

XSD

XSD (XML Schema Definition) specifies how to formally describe the elements in an **XML** document. It is more powerful than, and generally replaces, the older Document Type Definition (DTD). eGate Integrator makes use of **Object Type Definition (OTD)**s described in XSD, as well as DTD.

XSLT

XSLT (Extensible Stylesheet Language Transformation) is a language for transforming XML documents into other XML documents. It is designed for use as part of XSL, which is a stylesheet language for XML. eGate Integrator makes use of **Collaboration Definition**s coded in XSLT.

eGate 5.0 Terminology

Table 55 lists terminology that is new with eGate release 5.0 along with equivalent terms from eGate release 4.x, where applicable.

Table 55 Terminology Cross-Reference

eGate 5.0 Term	Equivalent e*Gate 4.x Term
Connection	e*Way Connection
Connectivity Map	Schema Network View (closest)
Deployment	Running the Control Broker
Deployment Profile	<none> (part of Schema)
Enterprise Designer	Enterprise Manager
Enterprise Manager	Enterprise Monitor
Environment	Schema (physical layer only)
eWay	e*Way, e*Way Connection
eWay Configuration	e*Way Connection Configuration
External Application	e*Way Connection
External System	e*Way Connection
JMS Connection	e*Way Connection
ICAN Monitor	Enterprise Monitor
Integration Server	<none>
Link	JMS e*Way Connection
Linked Message Destination	<none>
Logical Host	Participating Host
Message Destination	Topic or queue
Message Server	MS IQ Manager
Object Type Definition (OTD)	Event Type Definition (ETD)
Process Manager	Control Broker
Project	Schema (logical layer only)
Queue	MS queue
Repository	Registry
Subproject	Schema (logical layer only)
Topic	JMS topic

Index

T

U

V

W

X