SeeBeyond ICAN Suite

eGate Integrator for eInsight Enterprise Service Bus User's Guide

Release 5.0.2

Version 20031222150504.

Contents

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 Projects"** explains how to create a Connectivity Map and use the Configuration Editor to modify eWay and JMS connections between Connectivity Map components.
- **Chapter 6 "Web Services"** describes how to use eGate Integrator in concert with other ICAN Suite components to create Web services.
- **Chapter 7 "Object Type Definitions"** describes how to create Object Type Definitions (OTDs).
- **Chapter 8"Environments"**explains how to create and populate eGate Environments, and how to configure and start Logical Hosts.
- **Chapter 9 "Project Deployment"** explains how to create and activate Deployment Profiles.
- **Appendix A"OTD Interfaces"**describes the two types of OTD frameworks.
- **Appendix B "The Schema Runtime Environment"** describes the SRE Monitor, used to manage e*Gate 4.x schemas running within eGate 5.0.

In addition, the **Glossary** on page 164 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 29). 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

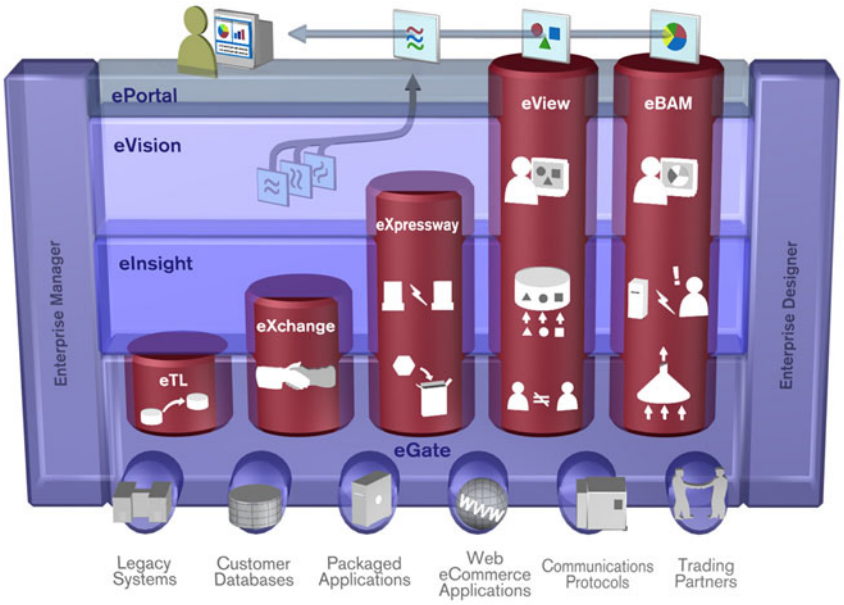
System Overview

This chapter provides an overview of the conceptual operation and general architecture of the eGate Integrator system.

2.1 Introduction

SeeBeyond's Integrated Composite Application Network (ICAN) Suite provides businesses with a comprehensive, unified eBusiness infrastructure to connect, integrate, and manage enterprise-wide software applications running on various computer systems. The full ICAN Suite is depicted in Figure 1.

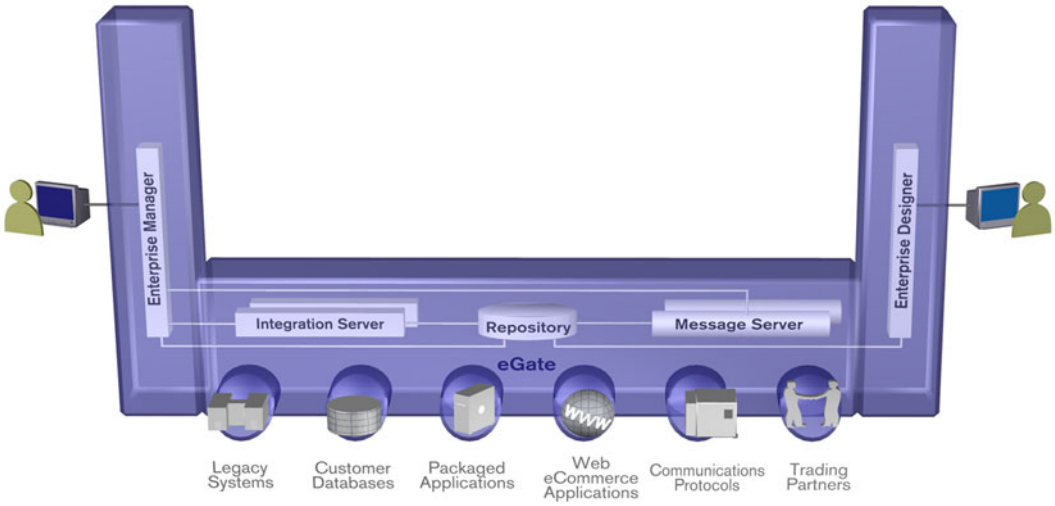
Figure 1 SeeBeyond ICAN Suite



SeeBeyond's eGate Integrator provides the "backbone" for the ICAN suite, integrating the various components of the Suite and all other connected components of the business enterprise. As shown in Figure 2, eGate Integrator includes the Enterprise Manager and Enterprise Designer, which provide graphical user interfaces for managing, configuring, and controlling the entire ICAN Suite and the business

processes running therein. See **Enterprise Manager** on page 25 and **Enterprise Designer** on page 23.

Figure 2 eGate Integrator



Other major constituents of eGate Integrator shown in Figure 2 are the Integration Server, the JMS IQ Manager, and the Repository, all of which will be described briefly later in this chapter. The flexibility of the eGate system allows the option of deploying it across a distributed network of hardware platforms, if desired, and running it on any combination of SeeBeyond, BEA WebLogic, and IBM WebSphere servers.

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 the 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 Enterprise Service Bus.

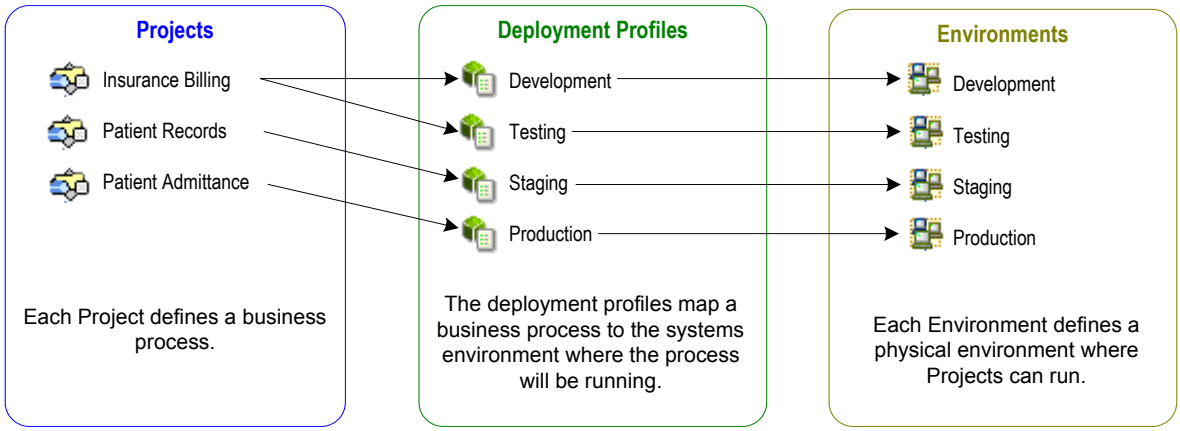
Projects are run within Logical Hosts, which are individual, runtime instances of eGate Integrator. Logical Hosts are defined within Environments, which represent the physical resources required to implement the Project. Projects are mapped to the individual Environments by means of Deployment Profiles, 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 3 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, an eGate Integrator implementation includes the following steps:

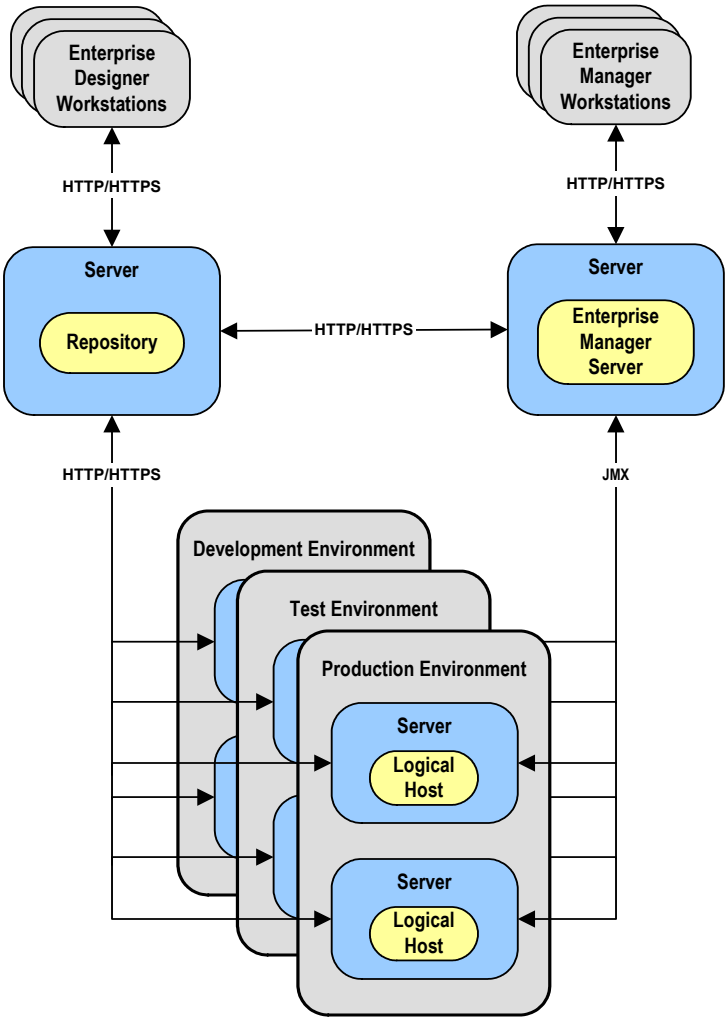
1. **Design your Project.**
2. **Define your Environments.**
3. **Create your Deployment Profiles.**
4. **Deploy the eGate Project.**

These implementation steps are all accomplished using the eGate Enterprise Designer, which is introduced in **Enterprise Designer** on page 23 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 4 shows an example system implementation that is highly distributed.

Figure 4 Typical eGate Integrator System



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 runtime 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 4, a single Repository serves the entire enterprise. This common Repository is used for development, testing, and production purposes. Communication between the Repository and other eGate components can be configured to use either HTTP or HTTPS. The Enterprise Designer and Enterprise Manager clients can communicate with the Repository and Enterprise Manager servers through a firewall.

2.3.2 Environments

An eGate Environment represents the physical 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 Logical Hosts. A Logical Host contains one or more **integration servers**, which are the engines that run eGate Collaborations 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).
- **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 4, the production environment is split across two hardware platforms, each running 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.

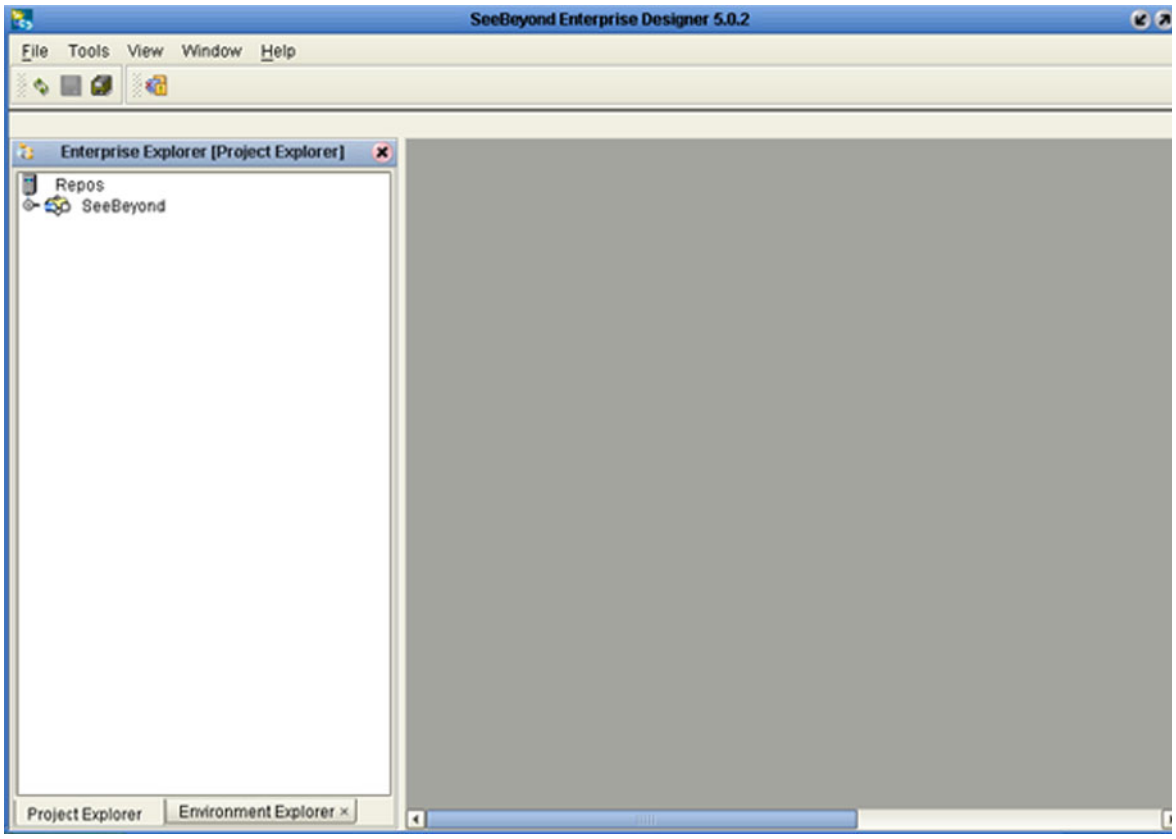
Note again that there is no requirement for the components shown in Figure 4 to run on separate systems; all could run on a single system, provided that resources (CPU, memory, and disk) are sufficient to support the concurrent usage.

2.4 User Interfaces

2.4.1 Enterprise Designer

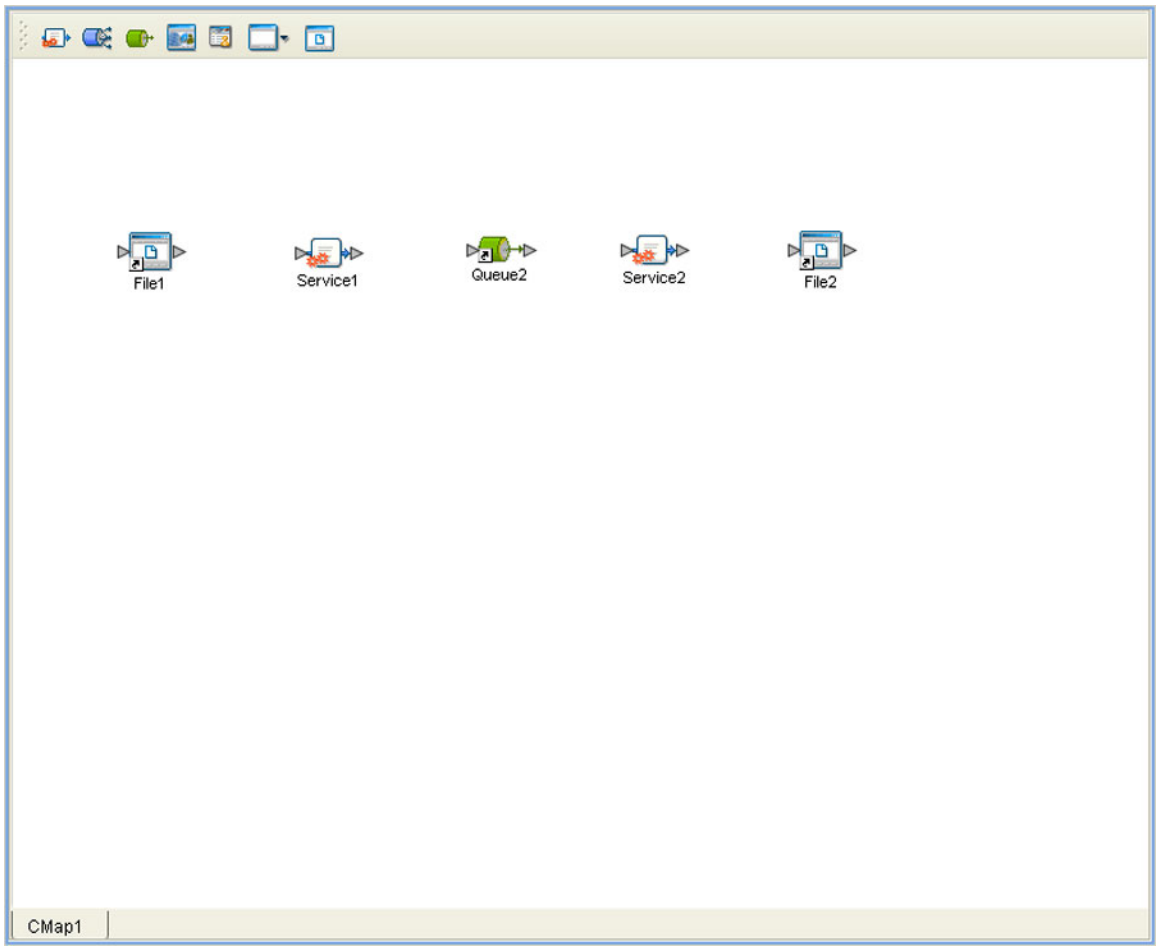
The Enterprise Designer graphical user interface (GUI) is used to create and configure the logical components and physical resources of an eGate Project. Through this GUI (see Figure 5), you can develop Projects to process and route data through an eGate Integrator system.

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. The Connectivity Map Editor (see Figure 6) provides a graphic example of one of these, in which logical components of a Project are created and connected.

Figure 6 Connectivity Map Editor



The features and usage of the Connectivity Map Editor are described in **eGate Projects** on page 61. Other editors are displayed for creating and modifying Object Type Definitions, Collaboration Definitions, Deployment Profiles, and other Project components.

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 32* .

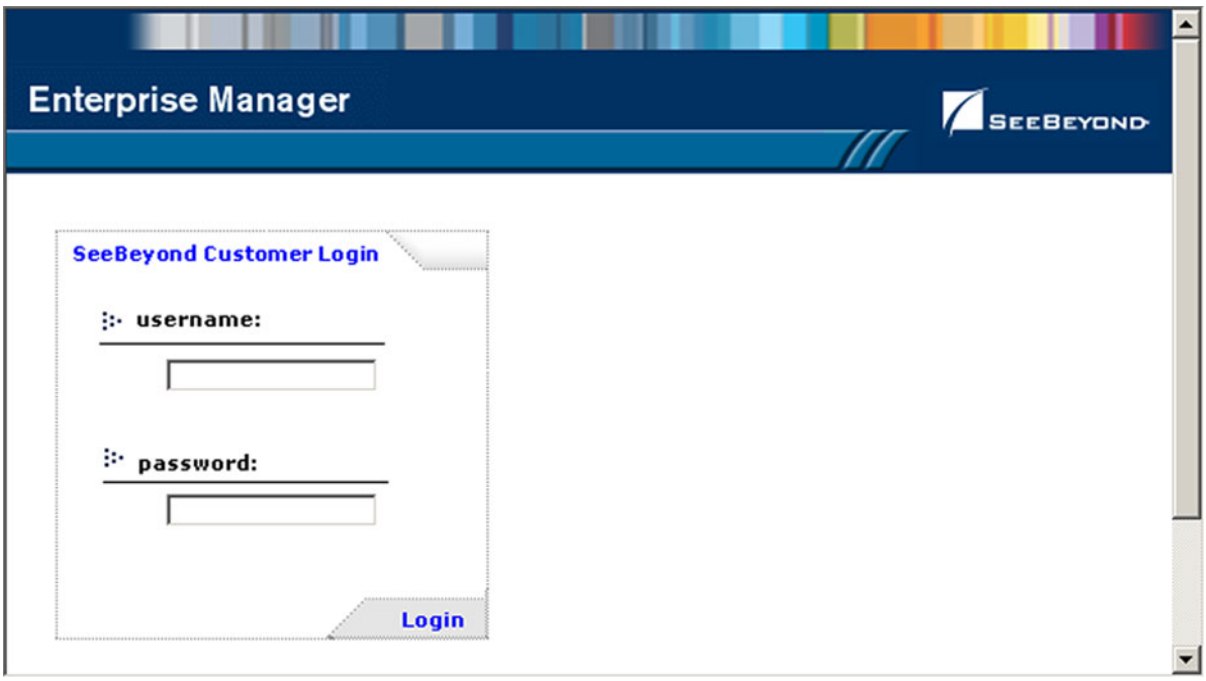
2.4.2 Enterprise Manager

The Enterprise Manager is a Web-based application you use for:

- Managing and monitoring eGate runtime components.
- Installing ICAN Suite products into the Repository.
- Downloading and installing products from the Repository.
- Accessing other Web-based ICAN Suite products.
- Accessing ICAN Suite product documentation.

The Enterprise Manager (see Figure 7) is accessed via Microsoft Internet Explorer,.

Figure 7 SeeBeyond Enterprise Manager Login



For more information on the Enterprise Manager, see **Enterprise Manager** *on page 26*.

For more information on Web Services capability, see **Web Services** on page 79.

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 *eInsight Enterprise Service Bus 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 Enterprise Monitor** on page 30 describes features of the Monitor interface itself.
- The Enterprise Manager also contains a facility that allows you to monitor and manage schemas from e*Gate 4.5.x in eGate 5.0, using the Schema Runtime Environment. See **The SRE Monitor** on page 295.

3.2 Starting Enterprise Manager

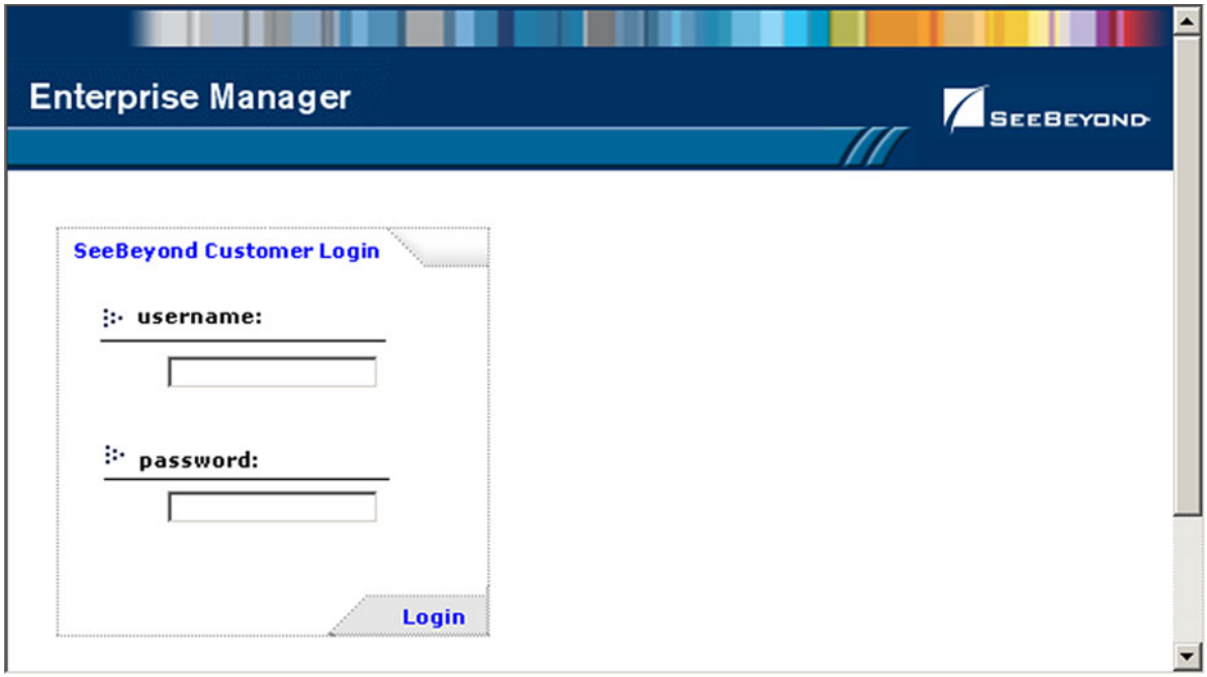
To start the Enterprise Manager

1. Launch Internet Explorer.
2. Enter **http://*hostname:portnumber*** in the **Address** box to display the SeeBeyond Customer Login window shown in Figure 8.

Note: *The **hostname** is the TCP/IP 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 eInsight Enterprise Service Bus Installation Guide.*

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

Figure 8 Enterprise Manager Login

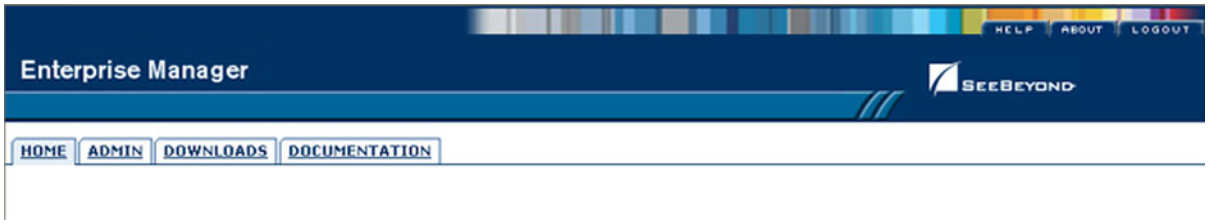


3. Enter your login ID and password in the **Username** and **Password** boxes and click **Login**.

3.3 The Enterprise Manager Interface

Once you have logged in, you see the full Enterprise Manager user interface (see Figure 9).

Figure 9 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

Tab	Function
Home	The Home tab is used for accessing the eGate Monitor, which is the main subject of this chapter. See **Home** on page 29.
Administration	The Administration tab is used in installing and updating ICAN components. See the *eInsight Enterprise Service Bus Installation Guide* for information.
Downloads	The Downloads tab is used in installing and updating ICAN components. See the *eInsight Enterprise Service Bus Installation Guide* for information.
Documentation	The Documentation tab is used for accessing ICAN Suite documentation. See **Documentation** on page 29, and the following *Note*.

Note: *You must download the documentation SAR files from the installation disk before you can access any documents from this page (see the eInsight Enterprise Service Bus Installation Guide).*

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

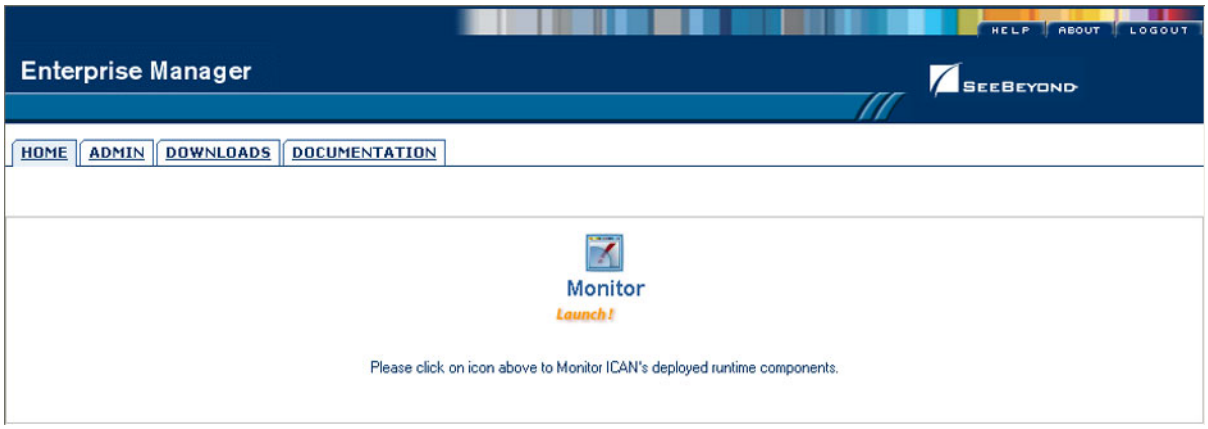
Table 3 Enterprise Manager - Control Tabs

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 displayed *only* on the Home page.
Home	The Home tab returns you to the Home page. This tab is displayed on all pages *other than* the Home 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** tab (see Figure 10) contains a link to the Enterprise Monitor. Click the **Monitor** icon to launch the Monitor (see **The Enterprise Monitor** on page 30).

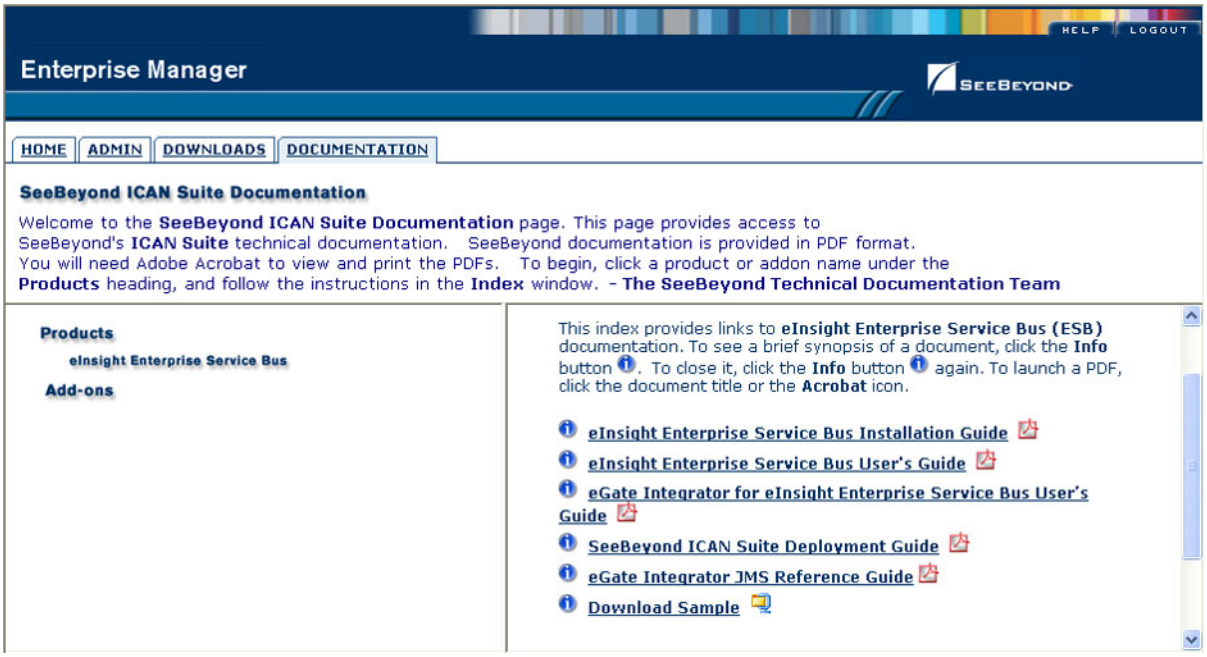
Figure 10 Monitor Launch Window



3.3.2 Documentation

The **Documentation** tab (see Figure 11) contains links to the latest versions of the SeeBeyond ICAN documentation in PDF format, and also any sample Project files. Shown is the set for eInsight Enterprise Service Bus.

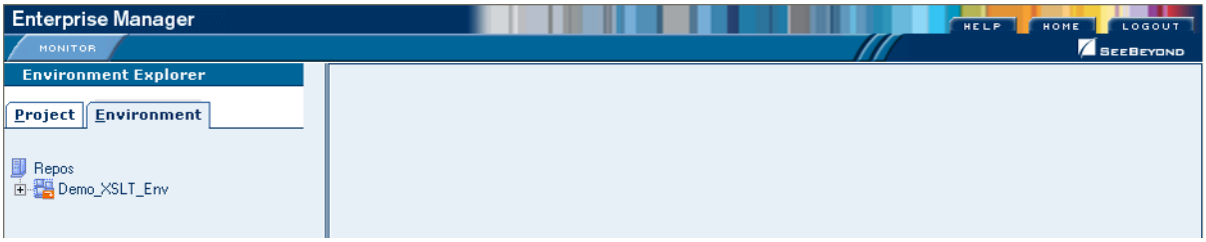
Figure 11 Documentation Tab



3.4 The Enterprise Monitor

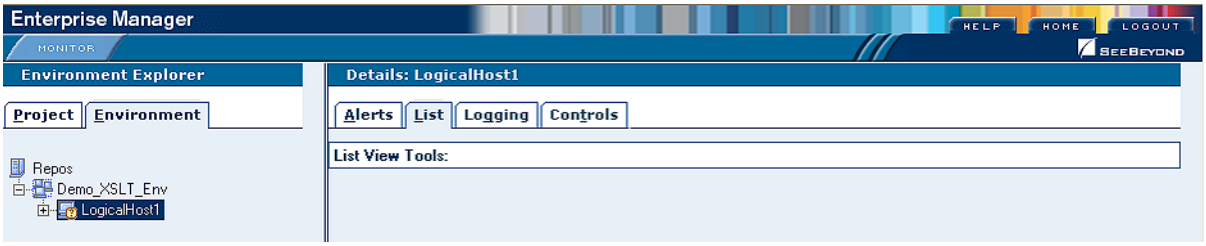
The Monitor has structure similar to that of the Enterprise Designer, with an Explorer panel on the left and a Details panel on the right. Initially, the Details panel is blank as shown in Figure 12.

Figure 12 Monitor Interface - Initial



Like the Enterprise Manager itself, the Monitor's **Details** area is organized into sections represented by tabs. 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 13 .

Figure 13 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. The full set of tabs is described in Table 4.

Table 4 Monitor Interface - Details Tabs

Tab	Function
Alerts	Displays functionality-related information about 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 rolling transactions forward or backward.
Summary	Displays a summary of information regarding the component selected in the upper Details panel.

Table 4 Monitor Interface - Details Tabs

Tab	Function
Consumption	Displays the number of messages processed by the component selected in the upper Details panel, and the number of messages still pending.

Note: *Not all listed tabs are always present, and tabs may be divided between an upper and lower Details panel.*

The Monitor interface offers the following viewing controls:

- **ALT** and drag the cursor to scroll.
- **CTRL** and click to zoom out.
- **CTRL-SHIFT** and click to zoom in.

Note: *See the eGate Integrator System Administration Guide for examples of monitor usage.*

Chapter 4

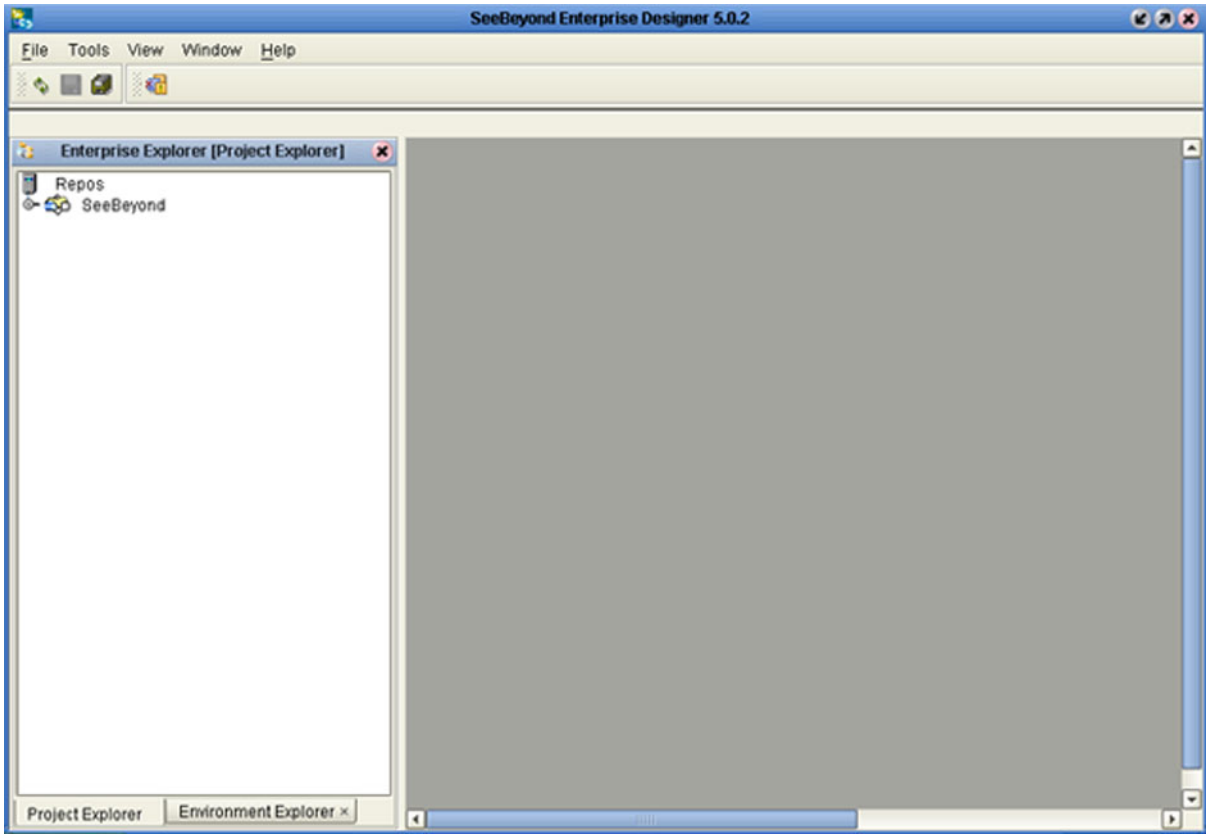
Enterprise Designer

This chapter describes the various 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 14), you can develop Projects to process and route data through an eGate Integrator system.

Figure 14 SeeBeyond 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 Enterprise Explorer provides two views of the ICAN system, which are described in the following sections of this chapter:

- **Project Explorer** on page 40
- **Environment Explorer on page 41**

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

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.

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

- **Menus** on page 36 describes the options contained in the individual menus.
- **Toolbar** on page 38 describes the functionality of the toolbar icons.
- **Browser Buttons** on page 39 describes the browser buttons that appear throughout the Enterprise Designer, in various wizards and dialog boxes.

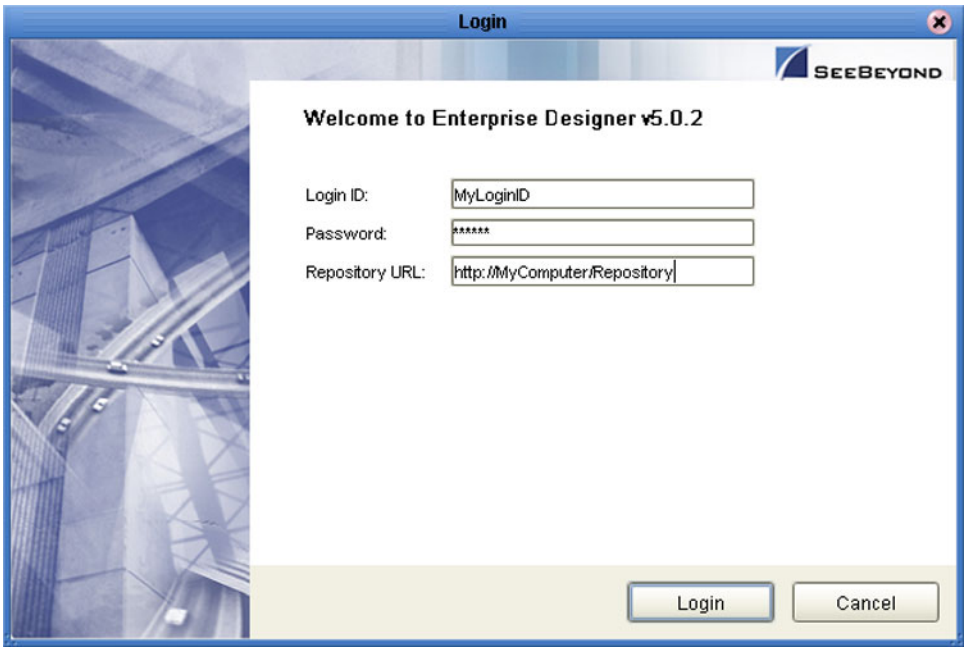
The procedure for invoking the Enterprise Designer is described in **Starting Enterprise Designer** on page 34.

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 15 (placing a shortcut on your desktop streamlines this procedure).

Figure 15 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 eInsight Enterprise Service Bus Installation Guide for details.

5 Click **Login** to complete the login process and display the Enterprise Designer GUI shown in Figure 14.

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 15.

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 eInsight Enterprise Service Bus Installation Guide for details.

5 Click **Login** to complete the login process and display the Enterprise Designer GUI shown in Figure 14.

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.

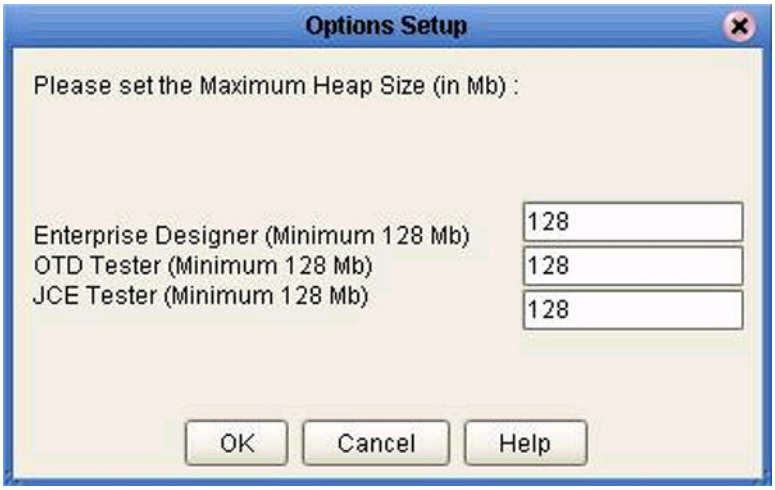
File Menu

Table 5 File Menu Options

Option	Function
Save	Saves changes to the selected Project.
Save All	Saves changes to all Projects.
Exit	Closes the Enterprise Designer.

Tools Menu

Table 6 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 the maximum heap size for selected components: 
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 7 View Menu Options

Option	Function
Environment Explorer	Activates the **Environment Explorer** tab on the Enterprise Explorer. See **Environment Explorer on page 41**.
Project Explorer	Activates the **Project Explorer** tab on the Enterprise Explorer. See **Project Explorer** on page 40.

Window Menu

Table 8 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 Project Editor.
Restore All	Returns minimized windows to their original position on the Project Editor.
Close All	Closes all open windows.

Help Menu

Table 9 Help Menu Options

Option	Function
Contents	Displays the online help for all installed components of the ICAN Suite, opening to the initial ICAN topic.
Help Sets	Displays the online help for all installed components of the ICAN Suite, opening to the initial topic for the selected coponent.

4.3.2 Toolbar

Table 10 Enterprise Designer Toolbar Icons

Icon	Function
	**Save saves** changes to the selected Project (inactive if no changes have been made).
	**Save All** saves changes to all Projects (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 11 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

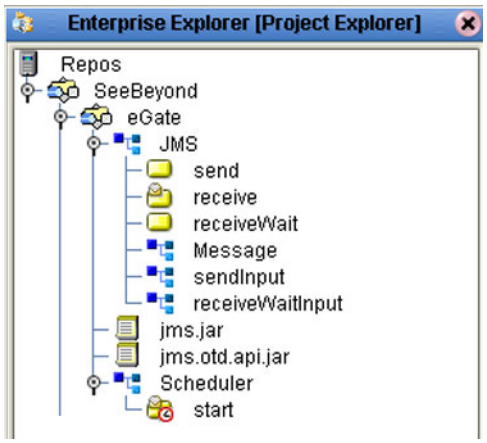
The Enterprise Explorer organizes the components of a Project into tabs that display different views of an eGate system.

- **Project Explorer** on page 40 deals with logical components.
- **Environment Explorer on page 41** deals with physical resources, including the Logical Host and Integration Server.

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 16.

Figure 16 Enterprise Explorer: Project Explorer View



Details of the features and usage of the Project Explorer are found in **eGate Projects** on page 61.

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 17 Enterprise Explorer: Environment Explorer View

Details of the features and usage of the Environment Explorer are found in **Environments** on page 121.

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. These editors are briefly 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

Note: *See the eGate Integrator Tutorial for an end-to-end demonstration of the steps involved in setting up a Project.*

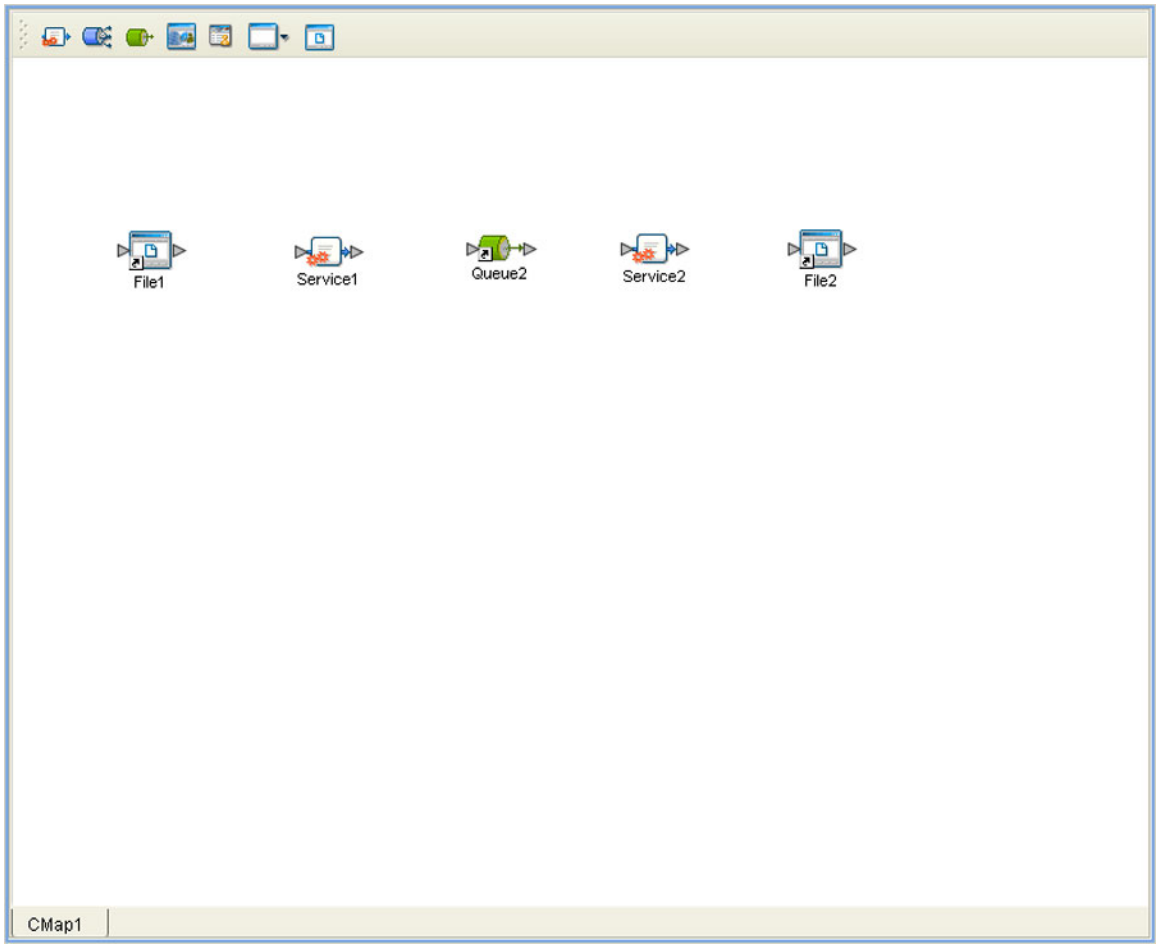
4.5.1 Connectivity Map Editor

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 18, 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.

Note: *You should create your Collaboration Definitions before using the Connectivity Map to connect components.*

See **Using the Connectivity Map Editor** on page 67 for detailed information.

Figure 18 Connectivity Map Editor

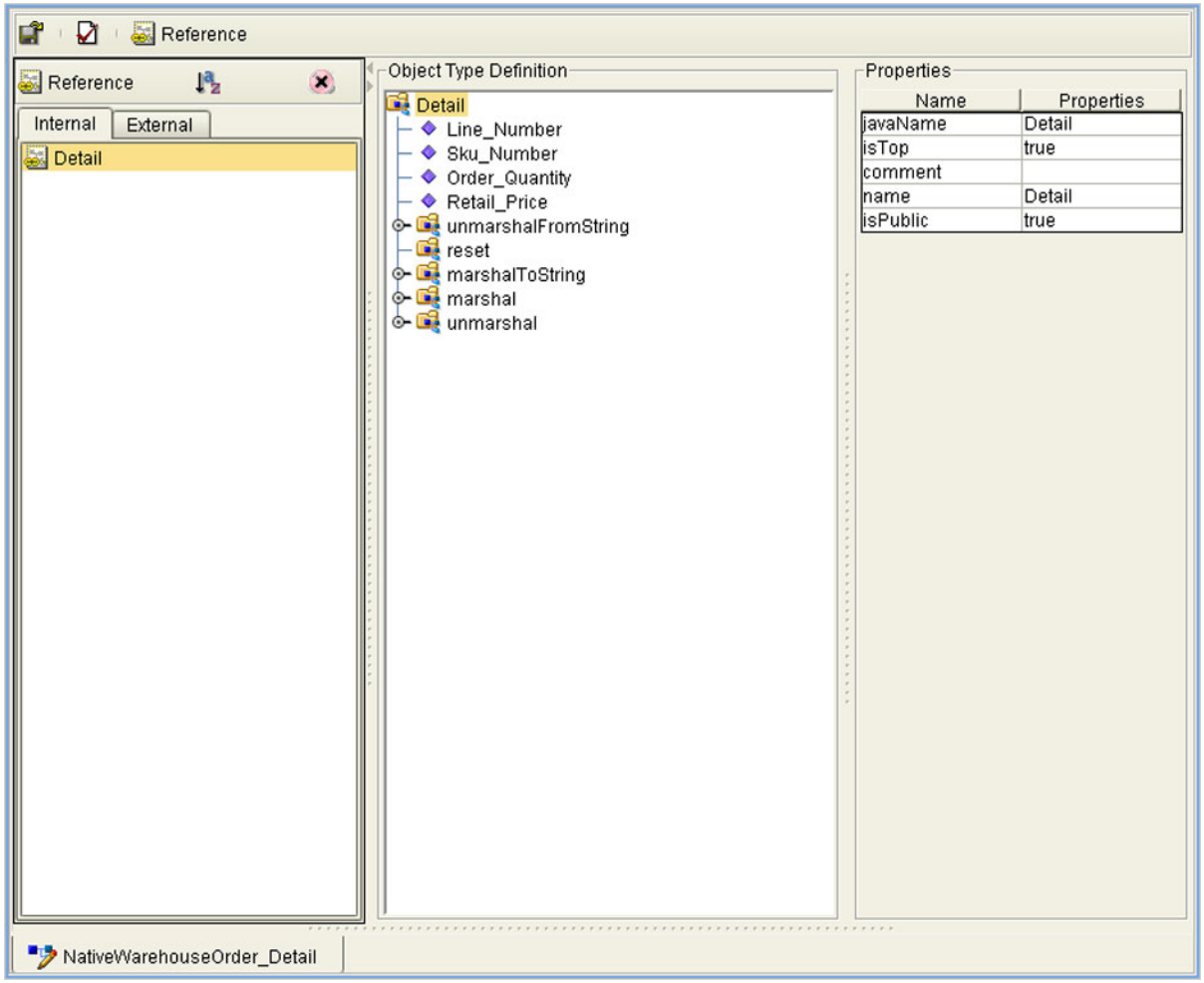


4.5.2 OTD Editor

The OTD Editor window, as shown in Figure 19, displays the source files used to create the Object Type Definitions (OTDs) to use with a Project. You use an OTD wizard tool to compile OTD files and add them to the **Project Explorer** tab.

See **Using the OTD Editor** on page 115 for detailed information.

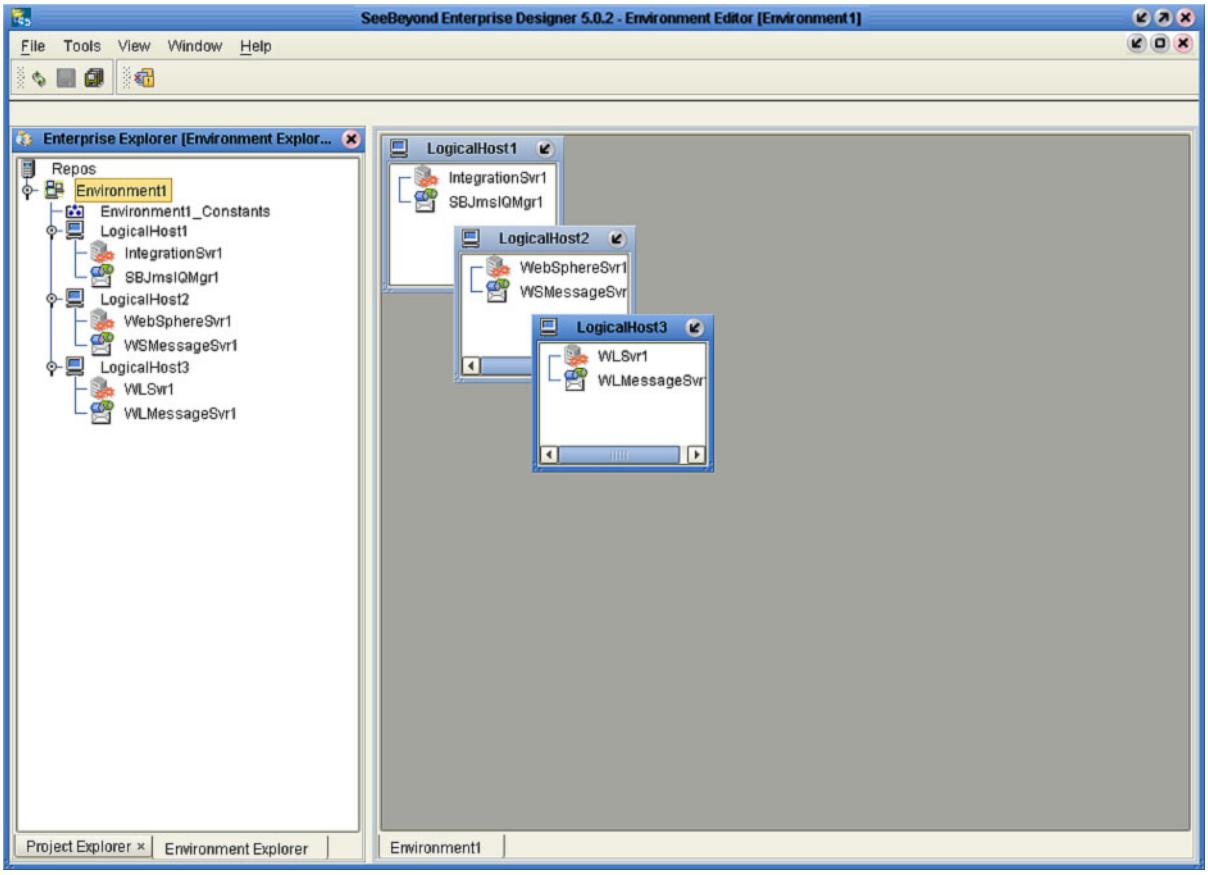
Figure 19 OTD Editor



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 20.

Figure 20 Environment Editor

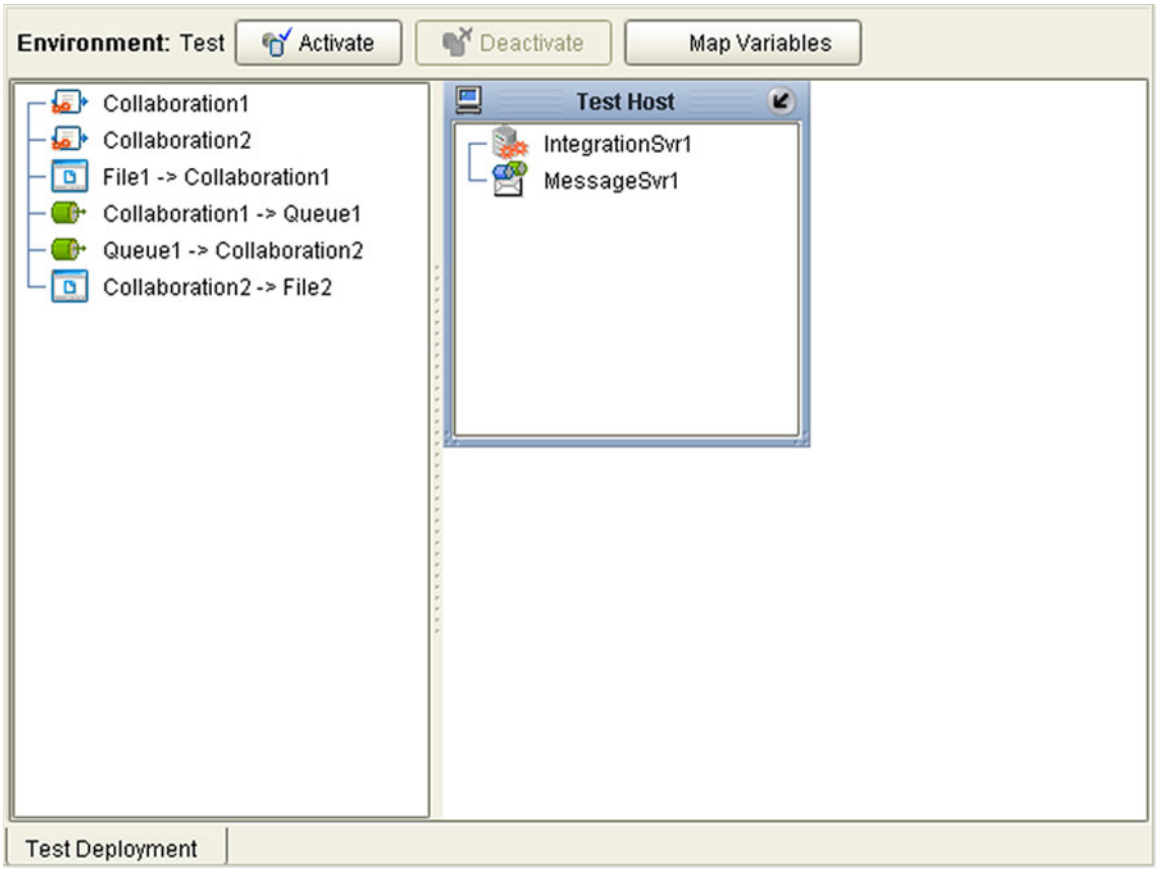


Note: *Unlike changes to Project-related configuration properties, changes to Environment-related properties do not require redeployment.*

4.5.4 Deployment Editor

The Deployment Editor, as shown in Figure 21, contains information about how Project components will be deployed in an Environment. See **Using the Deployment Editor** on page 151 for detailed information

Figure 21 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.

Important: *APIs 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.
- You can specify a new Project name and location (in Project Explorer) during import.
- 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.

Note: *A record of this process can be found in:*

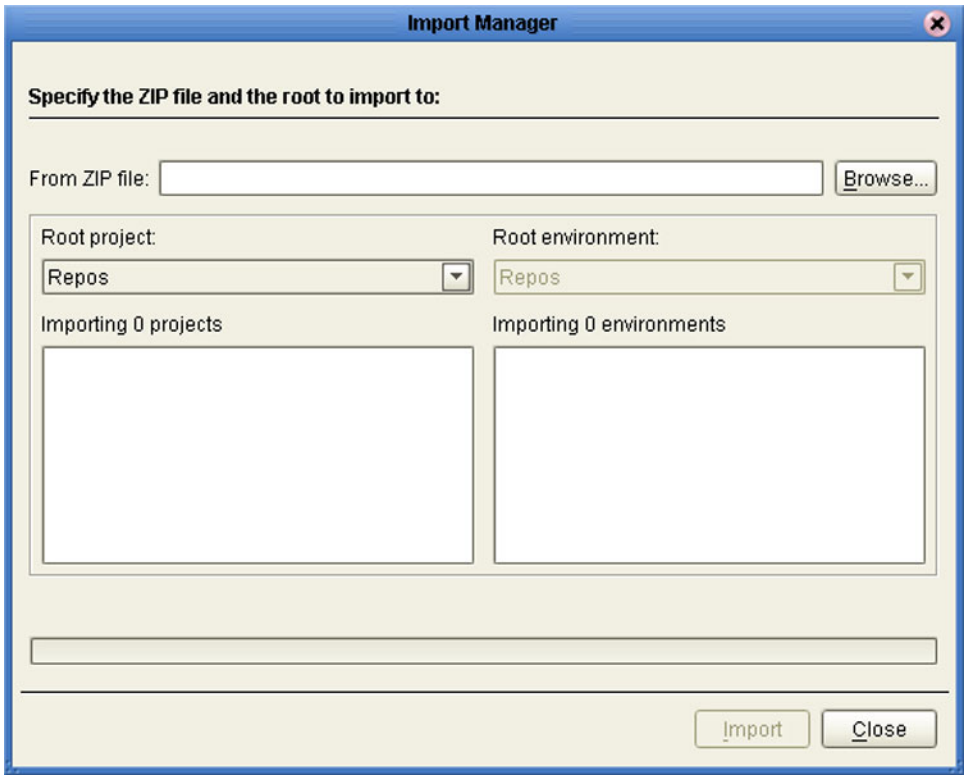
ICAN-root`\repository\logs\repository.log`

Importing a Project Using Enterprise Designer

To import a Project using Enterprise Designer

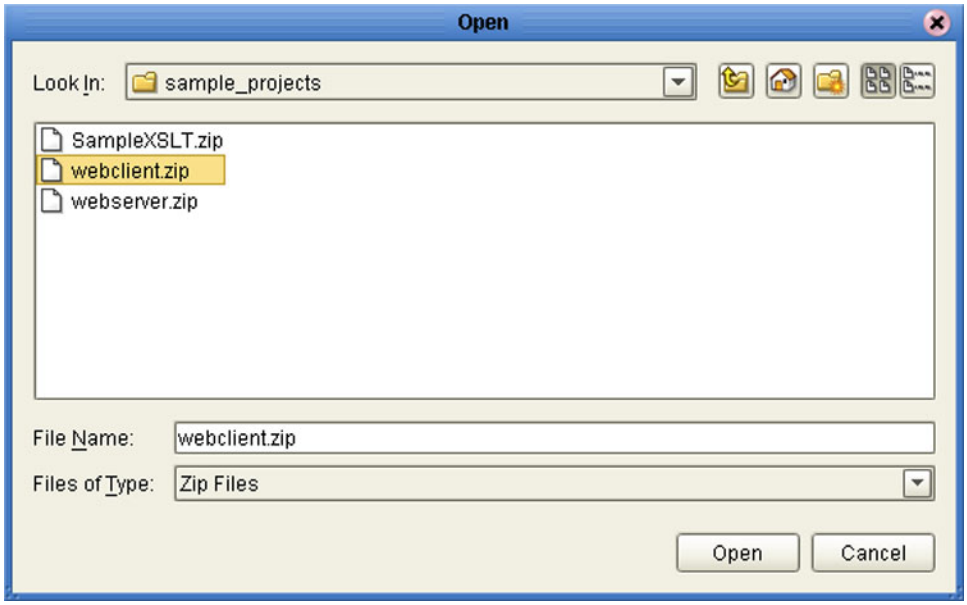
1. From the Repository context menu (for Projects) or the Project context menu (for Sub-Projects), select **Import** to display the dialog box shown in Figure 22.

Figure 22 *Import Manager* Dialog Box (1)



2 Click the **Browse** button to display the *Open File* dialog box, as shown in Figure 23. If you browse to an Environment file, the *Root environment* field will be enabled.

Figure 23 *Open File* Dialog Box

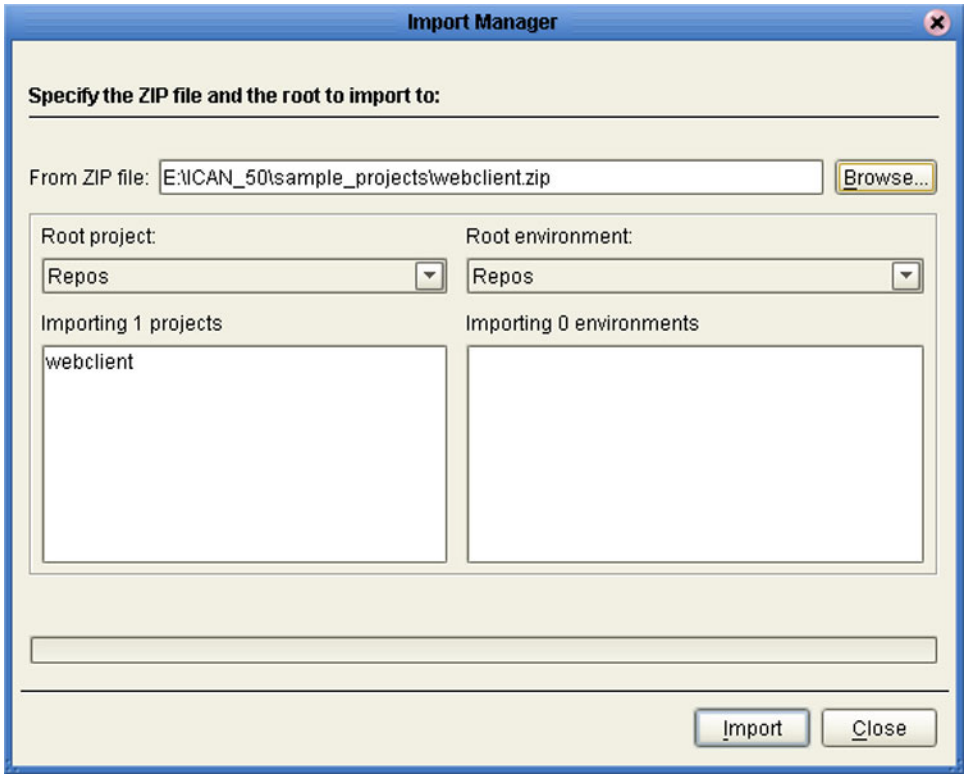


3 Locate and select the Project or Environment file that you want to import.

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

5 The Import Manager dialog box appears as shown in Figure 24, in which you specify the desired destination and file name (if different from the original).

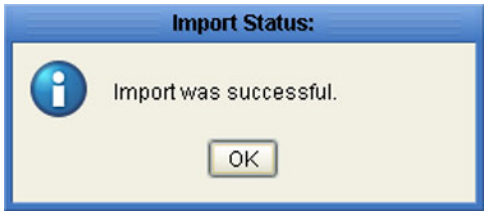
Figure 24 *Import Manager* Dialog Box



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

7 The Import Status message box shown in Figure 25 appears after the file has been imported successfully.

Figure 25 *Import Status* Message Box



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

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 fully-qualified archive file name for the Project you are importing.
- ***`rootprojectname`*** is either the name of the parent Project or the ICAN root directory. If the Project is to be attached to ***ICAN-root***, 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. Type (for example): **importProject Administrator stc c:\project4import.zip myExistingProject**.

   This will extract the Project that exists in the file **c:\project4import.zip** and attach it as a Sub-Project of **myExistingProject**.

4.6.2 Project/Environment Export

The export function allows you to export an eGate Project 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.

Note: *A record of this process can be found in:*

ICAN-root`\repository\logs\repository.log`

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 26. If you do, you will see the dialog box shown in Figure 27.

Figure 26 *Export Manager* Dialog Box (1a)

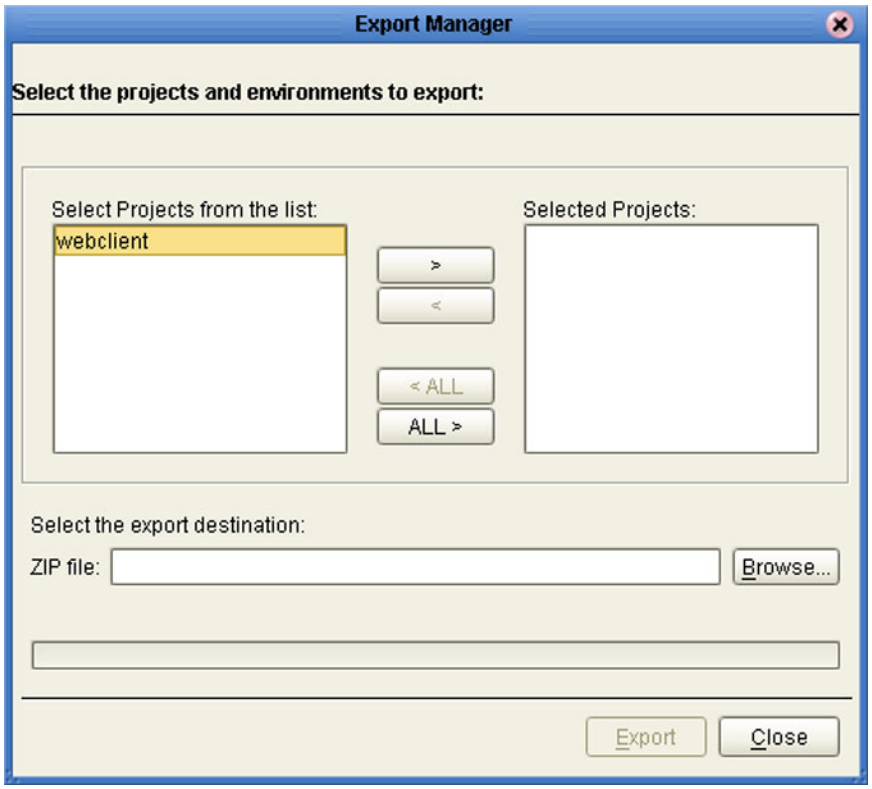
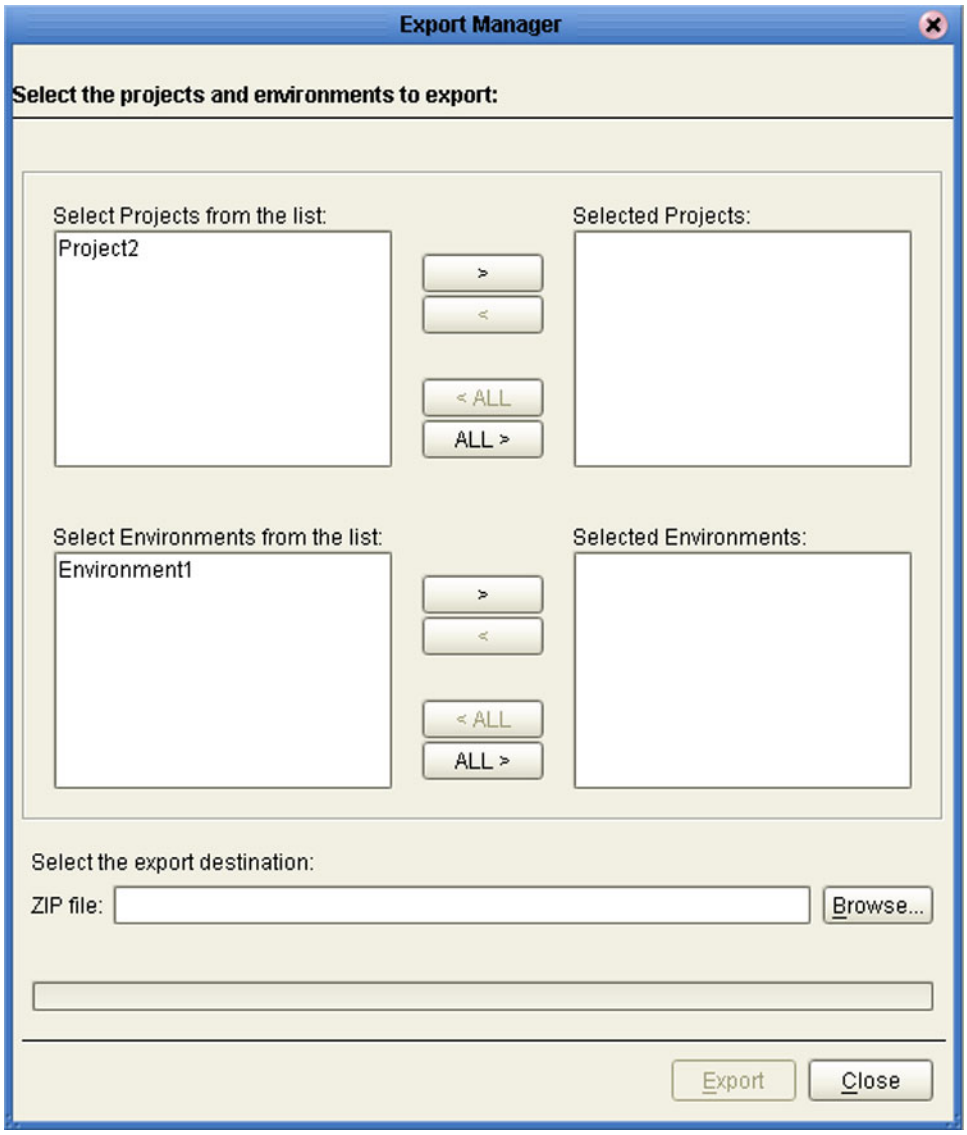
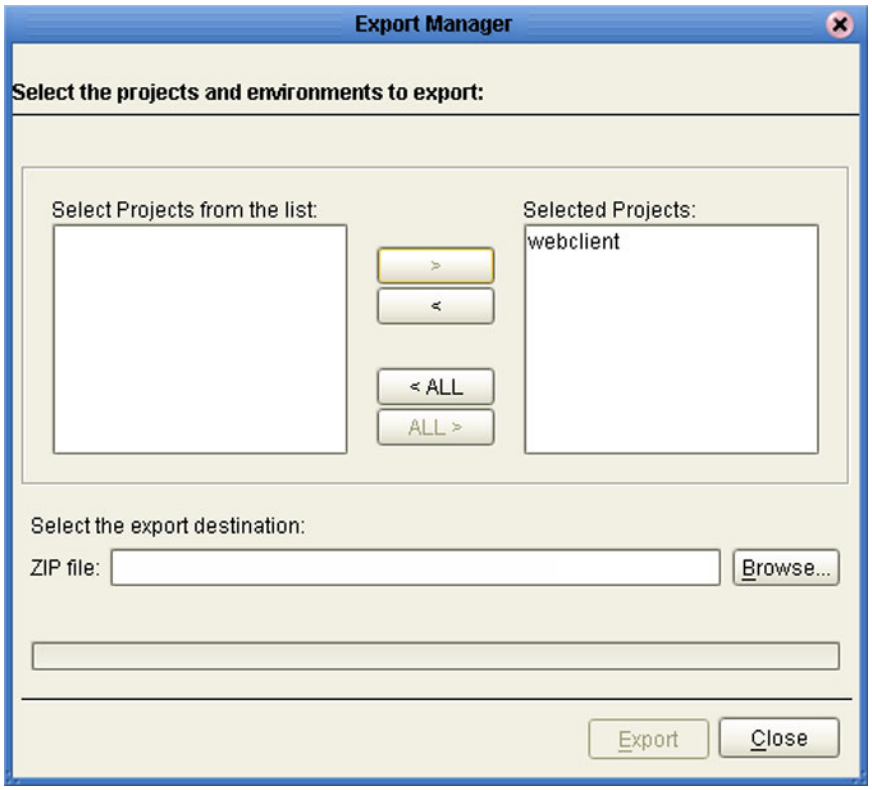


Figure 27 *Export Manager* Dialog Box (1b)



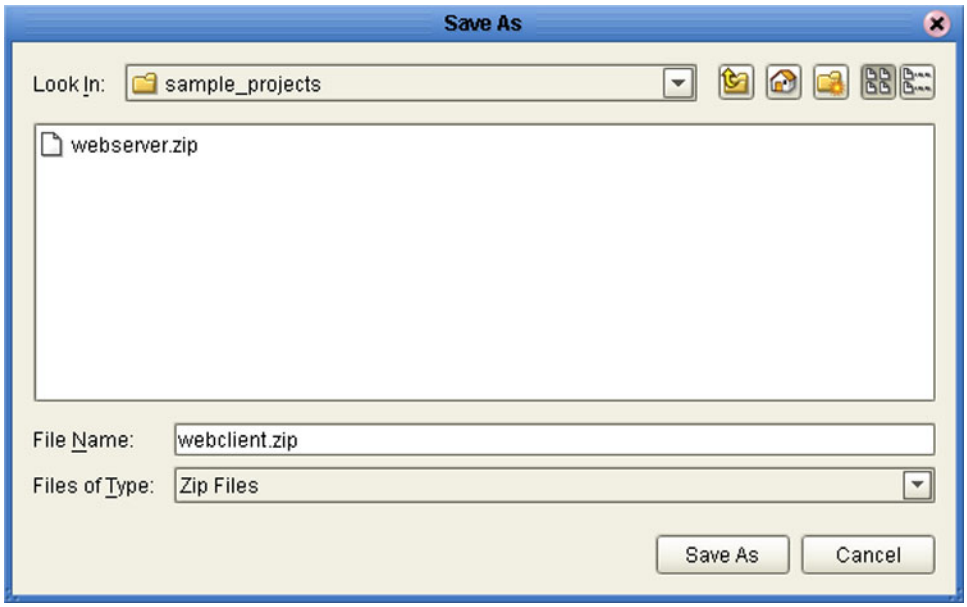
2 Highlight the desired Project(s) or Environment(s) in the displayed list, and transfer them to the *Selected Projects* or *Selected Environments* panel using the arrow buttons (see Figure 28).

Figure 28 *Export Manager* Dialog Box (2)



3 Click the **Browse** button to display the *Save As* dialog box, as shown in Figure 29.

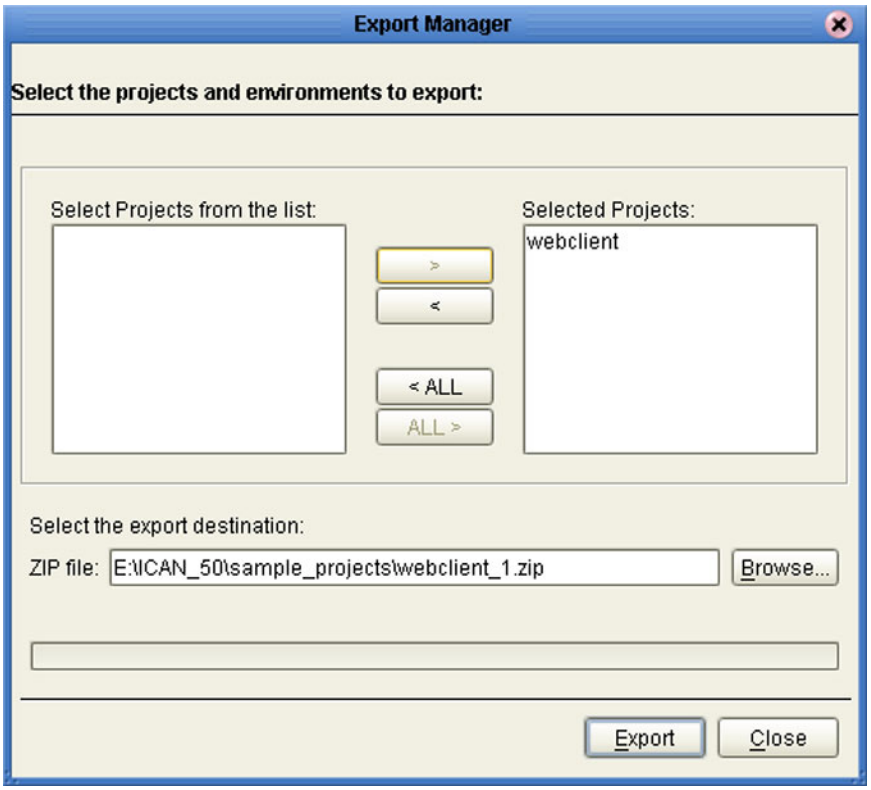
Figure 29 *Save As* Dialog Box



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

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

Figure 30 *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 31 appears after the file has been exported successfully.

Figure 31 *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 fully-qualified archive file name for the Project you are exporting, indicating where it is to be stored.
- ***projectname*** is the name of the Project you are exporting.
- ***environmentname*** is the name of the Environment you are exporting.

To import a Project using the import script

1. Open a command prompt and change directory to *ICAN-root*\repository\util.
2. Type (for example): **importProject Administrator stc c:\project4export.zip myProject myEnvironment**.

   This will save the existing Project **myProject** and Environment **myEnvironment** to the file **c:\project4export.zip**.

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 32.
3. In the *Please show me* drop-down list, select items you would like to view.
4. From the list of objects that appears, select one for which you would like to perform an impact analysis.
   - You can print the object list by clicking **Print** to display the Windows *Print* dialog box.
5. Click **Impact** to see how that object would be affected by a change to the component you selected in step 1.

Figure 32 Impact Analyzer Dialog Box

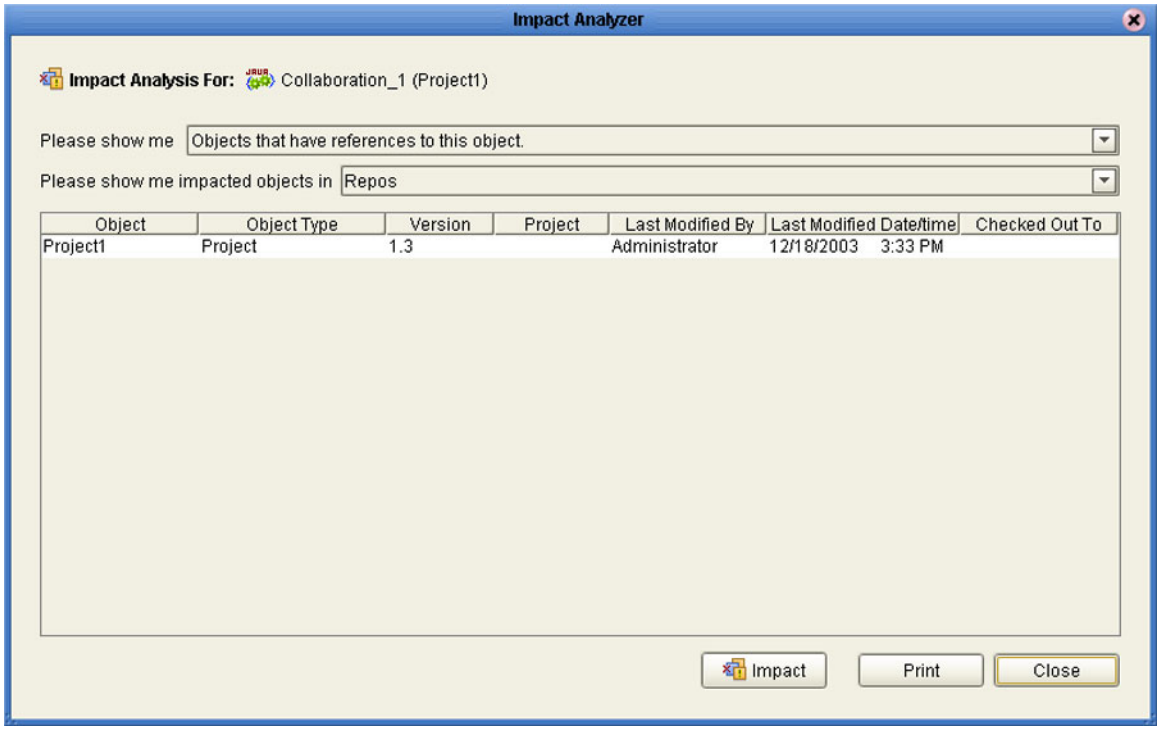
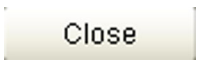


Table 12 Impact Analyzer Command Buttons

Button	Function
	Performs an impact analysis for the object selected from the object list.
	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 a Project or Environment component. The version history of each component is recorded to a log file, and can be viewed by means of a menu option.

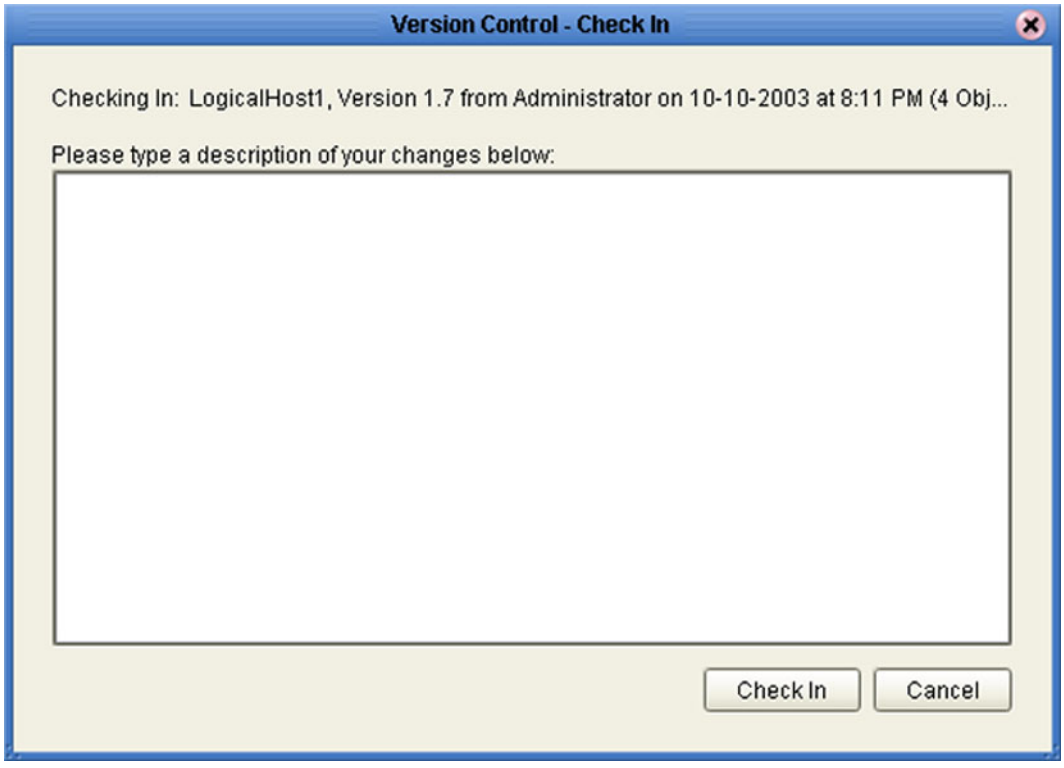
Checking a Component In

Once you have created and configured a Project or Environment component, you can check that object in by using the following procedure.

To check in a version of a Project or Environment component

1. Click the Project or Environment Explorer tab in the Enterprise Explorer.
2. Right-click on a component to display its context menu.
3. Select **Check In** to display the *Version Control - Check In* dialog box shown in Figure 33.

Figure 33 Version Control - Check In Dialog Box



4. Type in a description of the changes in the new version.
5. Click **Check In** to save your changes to a new version.

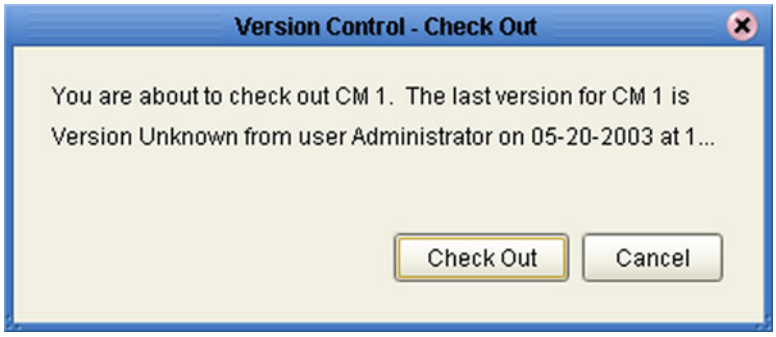
Checking a Component Out

Once an object has been checked in, you can check it out by using the following procedure.

To check out a version of a Project or Environment component

1. Click the Project or Environment Explorer tab from the Enterprise Explorer.
2. Right-click on a component to display its context menu.
3. Select **Check Out** to display the *Version Control - Check Out* dialog box shown in Figure 34.

Figure 34 Version Control - Check Out Dialog Box



4. Click **Check Out** to open the component.

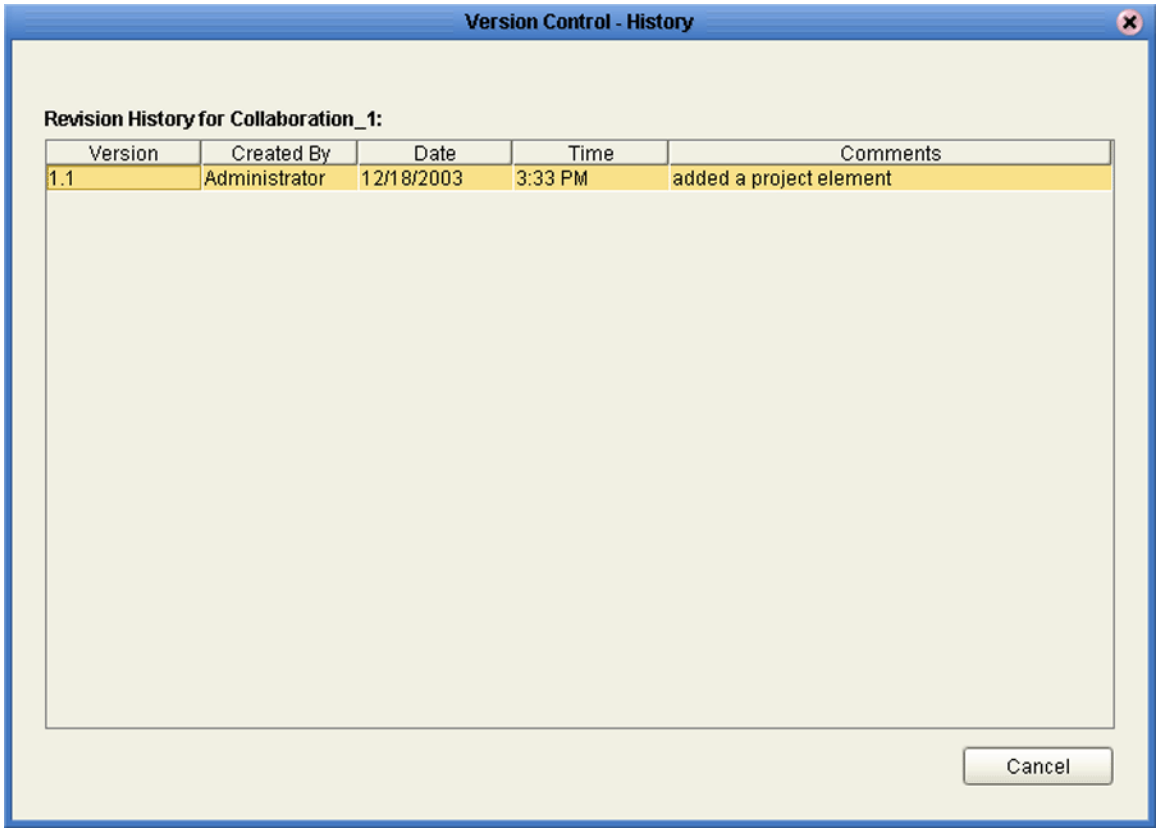
Note: *Only one user can have a file checked out for editing at a time. If another user attempts to check out the same file, they will receive a message indicating that the file is currently checked out.*

Viewing a Component's Version History

To view the version history for a component

1. Click the Project or Environment Explorer tab in the Enterprise Explorer.
2. Right-click on a component to display its context menu.
3. Select **Version History** to display the *Version Control - History* dialog box shown in Figure 35.

Figure 35 Version Control - History Dialog Box



Chapter 5

eGate Projects

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

5.1 Overview

An eGate 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 121). An end-to-end example of Project implementation is given in the *eGate Integrator Tutorial*.

An eGate Project is related to an Activity in an eInsight business process. Components developed for use in one Project can be used in another, and a Project can internally reference another Project.

5.1.1 Project Components

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

- **Services** on page 69
- **External Applications** on page 71
- **Schedulers** on page 71
- **Component Connections** on page 73
- **Message Destinations** on page 70

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

- **Collaboration Definitions**

  A Collaboration Definition defines the logical operation taking place in the related Collaboration, and is based on an Object Type Definition.

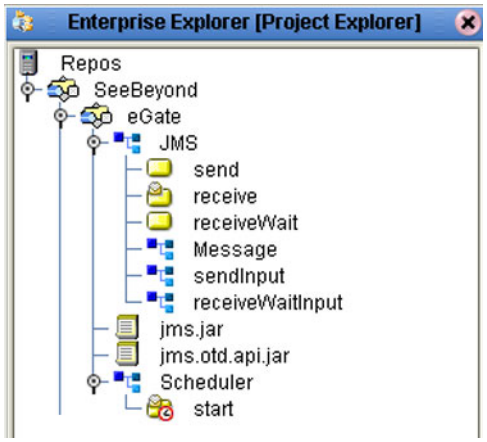
- **Object Type Definitions**

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.

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 36).

Figure 36 Project Explorer



The Project Explorer is used in conjunction with the Connectivity Map Editor (see **Using the Connectivity Map Editor** on page 67) to create and configure a Project.

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

5.2.1 Project Explorer Icons

The icons described in Table 13 appear in the Project Explorer.

Table 13 Project Icons

Icon	Description
	Represents the **Repository**, which is the database where all Projects and contents are saved.
	Represents the **Project** or subproject.
	Represents a **Connectivity Map**, which contains the business logic and information about the data transmission. .
	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 on a component in the Project Explorer displays a context menu for that component. Included here are descriptions of options for the following component context menus:

- **Repository Menu** on page 64
- **Project Menu** on page 65
- **Connectivity Map Menu** on page 66

Repository Menu

Figure 37 Repository Menu

Table 14 Repository Menu Options

Option	Function
Project	Adds a new Project to the **Project Explorer** tab.
Sort by Type	Places Project components in order by grouping component types.
Sort by Name	Places Projects and Project components in alphabetical order.
Sort by Date	Places Projects in order by creation date from oldest Project 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 another location.
Refresh All from Repository	Updates Enterprise Designer with Project/Environment configurations stored in the Repository. (Open editors are not refreshed.)
User Management	Displays the User Management dialog box, where you 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 the properties of your Repository. See the *eGate Integrator System Administration Guide*.

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

Project Menu

Figure 38 Project Menu

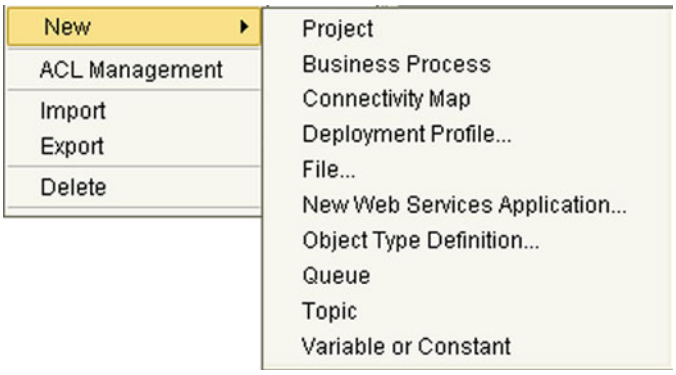


Table 15 Project Menu Options

Option	Option	Function
New	Project	Adds a Subproject folder to the selected Project.
	Business Process	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 67.
	Deployment Profile	Displays a dialog box with which you can assign a Deployment Profile to the selected Project. See **Using the Deployment Editor** on page 151.
	File	Displays a dialog box with which you can create an external file to use with the Project.
	New Web Services ...	Adds a third-party Web service application to the Project Explorer. See **SeeBeyond Web Services** on page 80.
	Object Type Definition	Displays the **OTD Wizard**, with which you can create an Object Type Definition (OTD) file. See **Using the OTD Wizard** on page 100 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 you can assign read and/or write privileges to users for the selected Project. See the *eGate Integrator System Administration Guide*.
Import		Displays a dialog box with which you can import a Project as a Subproject under the selected Project.
Export		Displays a dialog box with which you can export a Project as a Subproject under the selected Project.

Table 15 Project Menu Options

Option	Option	Function
Delete		Deletes the selected Project.

Connectivity Map Menu

Figure 39 Connectivity Map Menu

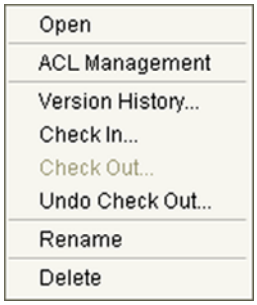


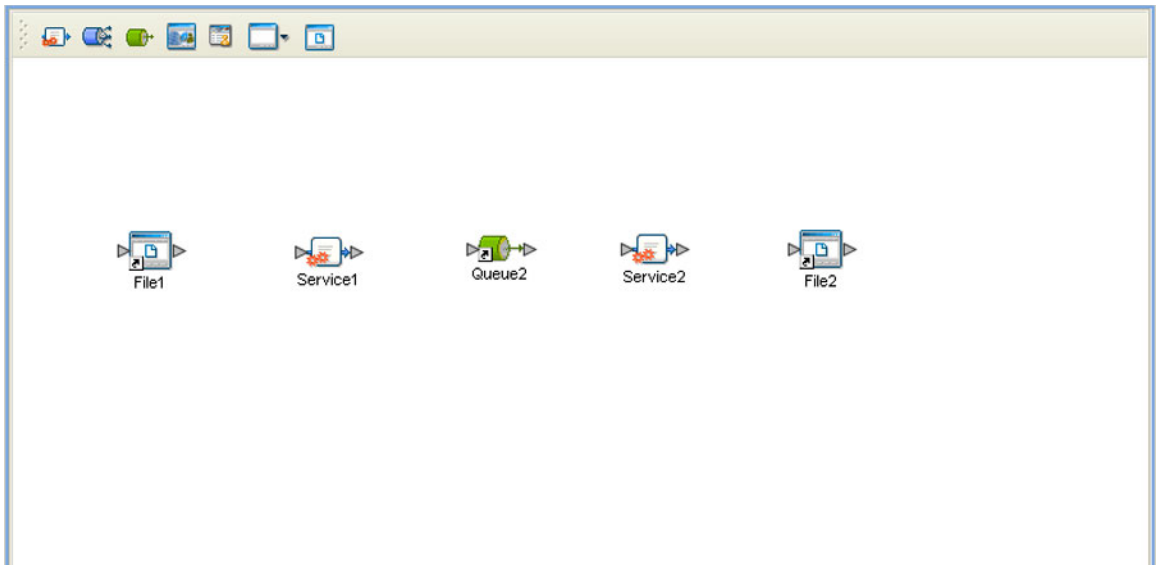
Table 16 Connectivity Map Menu Options

Command	Function
Open	Opens the **Connectivity Map Editor** for the selected Connectivity Map. See **Using the Connectivity Map Editor** on page 67.
ACL Management	Displays the **ACL Properties** dialog box, with which you can assign read and/or write privileges to users for the selected Project. See the *eGate Integrator System Administration Guide*.
Version History	Displays a dialog box with which you can track the version history for OTDs and Collaboration Definitions. Version control allows users to maintain multiple versions of the same OTD and Collaboration Definition files. See **Viewing a Component's Version History** on page 60 for more information.
Check In	Displays a dialog box, with which you can check in the current version of a Project. Refer to **Checking a Component In** on page 58 for more details.
Check Out	Displays a dialog box with which you can check out a version of a Project. See **Checking a Component Out** on page 59 for more information.
Undo Check Out	Reverses the Check Out command, returns you to the previous state.
Rename	Allows you to rename the selected Connectivity Map.
Delete	Displays a dialog box in which you confirm that you want to delete the selected Connectivity Map. Clicking **Yes** then deletes the Connectivity Map.

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 40). 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.

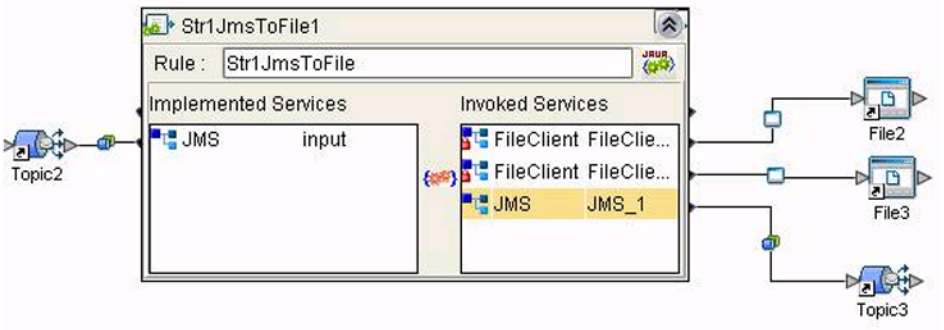
Figure 40 Connectivity Map Window



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.

When there are multiple destinations, as with a JMS topic, the Connectivity Map Editor cannot resolve which output port to connect to which destination. Because of this, the Collaboration definition must be created first, and the connections must be drawn by opening the Collaboration Binding box in Connectivity Map (see Figure 41).

Figure 41 Linking JMS Topics



The Connectivity Map Editor toolbar contains the icons listed in Table 17, plus additional icons representing eGate add-ons and other ICAN components that you may have installed.

Table 17 Connectivity Map Toolbar Icons

Icon	Component	Function
	Service	A logical component that provides the framework for a process or Collaboration. See **Service Component** on page 69.
	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 Service External Application	Represents a third-party Web service application external to eGate. See **SeeBeyond Web Services** on page 80.
	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 71.
	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 71.

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 Services

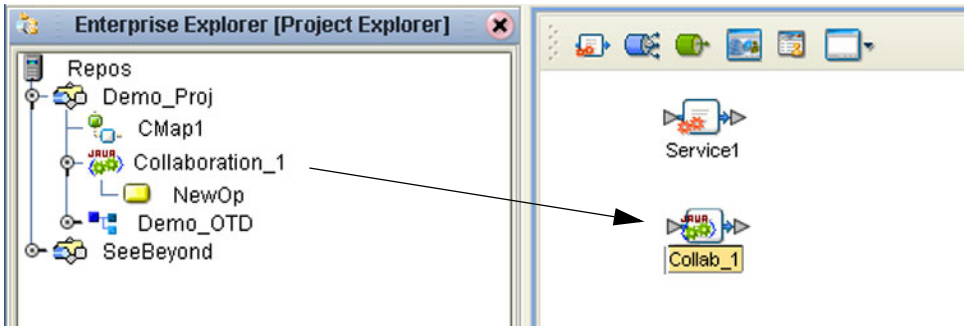
A service provides a framework for a process or a Collaboration, which contains the information required to execute a set of business rules.

5.4.1 Collaborations

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

The Collaboration acts as a service having a publication or subscription relationship with each linked entity. The link is provided by a JMS Client connection (see **Component Connections** on page 73. Dragging a Collaboration from the Project Explorer to the Service icon in the Connectivity Map defines the service as a Collaboration (see Figure 42).

Figure 42 Service Component



Note: *Any changes made to the names of the Collaborations should be done when the Collaboration is created. If the name is changed later, the Collaboration should be opened, regenerated if applicable, and saved again. For safe measure, this should also be done before creating the Connectivity Map and Deployment Profile.*

Connection-related properties for the Collaboration (or other service) are configured in the adjoining JMS Client. These properties include:

- Concurrent or serial processing
- Transaction mode (transacted or XA)
- Security

All properties, and the procedures for configuring them, are detailed in the *eGate Integrator JMS Reference Guide*.

5.5 Message Destinations

A Message Destination is a container for stored data, and can follow either the topic or queue JMS model.

5.5.1 Topics

A *topic* is a message destination that conforms to the publish-and-subscribe messaging paradigm.

5.5.2 Queues

A *queue* is a message destination that conforms to the point-to-point messaging paradigm.

5.6 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 43.

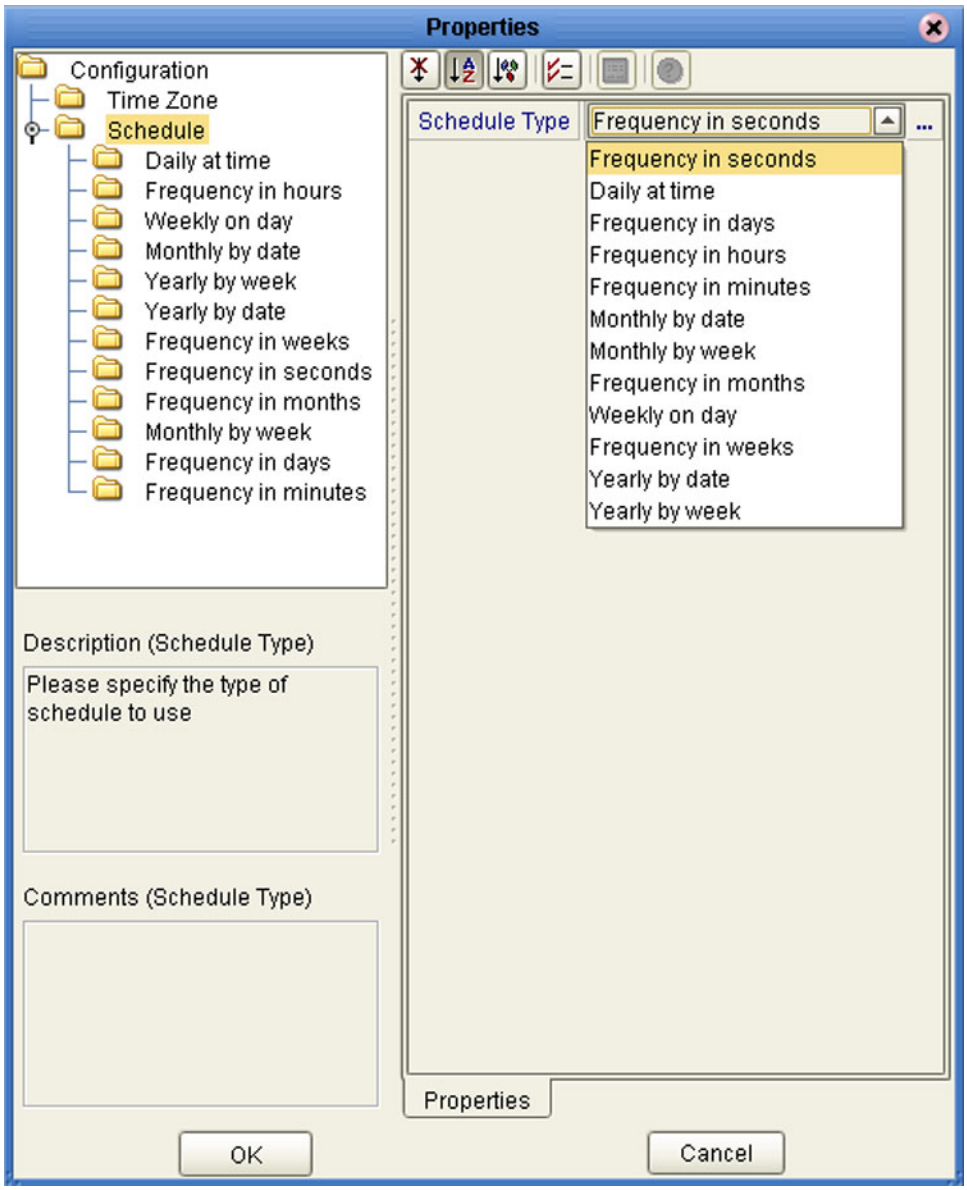
Figure 43 External Application Drop-Down Menu

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

5.6.1 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 44).

Figure 44 Scheduler Properties Dialog Box



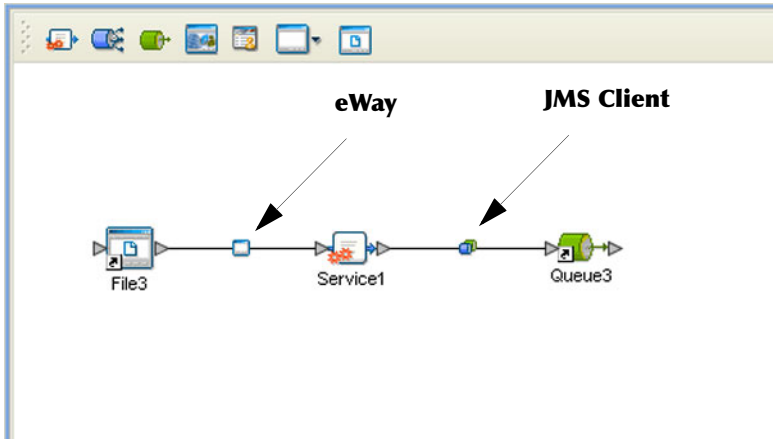
Selecting **Schedule** displays the **Schedule Type** property field which you set to the type of schedule you want to use. Selecting the corresponding node in the explorer tree displays the property field for that schedule type, in which you specify the desired value. The text in the *Description* box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

5.7 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 45).

Figure 45 Connection Icons in a Connectivity Map



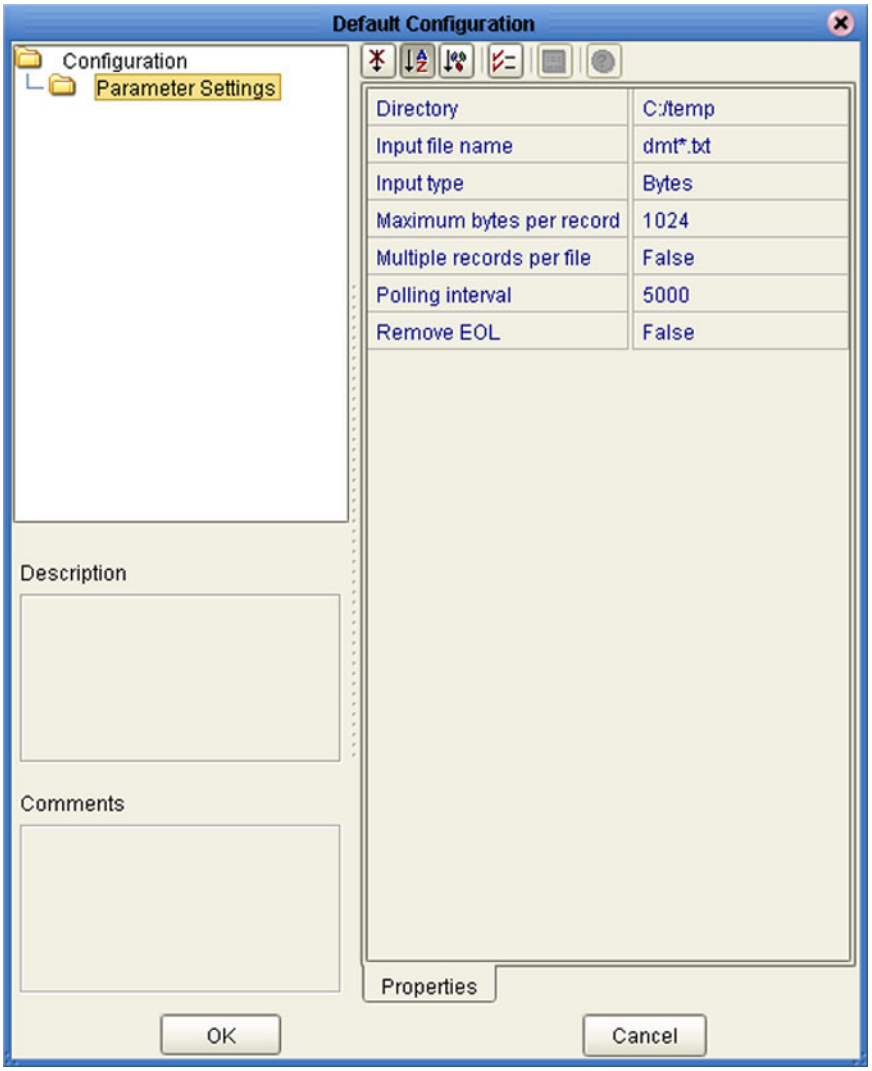
- When you link an external application with a Collaboration, the Enterprise Designer automatically adds an eWay Connection icon to the link. The eWay enables communication and movement of data between the external application and the eGate system. The eWay configuration specifies the logical connection 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.

Note: *You **must** configure the JMS properties, since no default values are assigned. Failure to do so results in an error at a later time.*

5.7.1 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 46 shows a dialog box that lists the configuration properties for a File eWay.

Figure 46 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, or a Service to be a publisher or subscriber. 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 18.
- The **Properties** folder lists the default properties for the selected eWay or JMS Client. See, respectively, the individual eWay Intelligent Adapter User's Guides and the *eGate Integrator JMS Reference Guide* for information.
- The **Description** box contains a brief description of the contents of the item currently selected in the Configuration Tree.
- The **Comments** box lists additional information about the item selected in the Configuration Tree.

Table 18 Configuration Dialog Box Toolbar Buttons

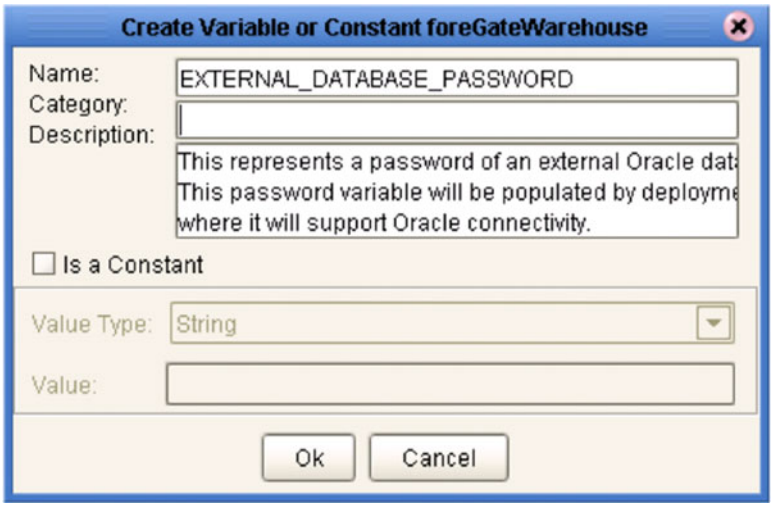
Button	Command	Function
	Unsorted	Displays configuration properties in their default order.
	Sort by Name	Sorts configuration properties alphabetically by name.
	Sort by Type	Displays configuration properties by property type.
	Show Editable Properties Only	Displays only the properties of an eWay or 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.

5.8 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 157). Project variable values can be literals or Environmental constants.

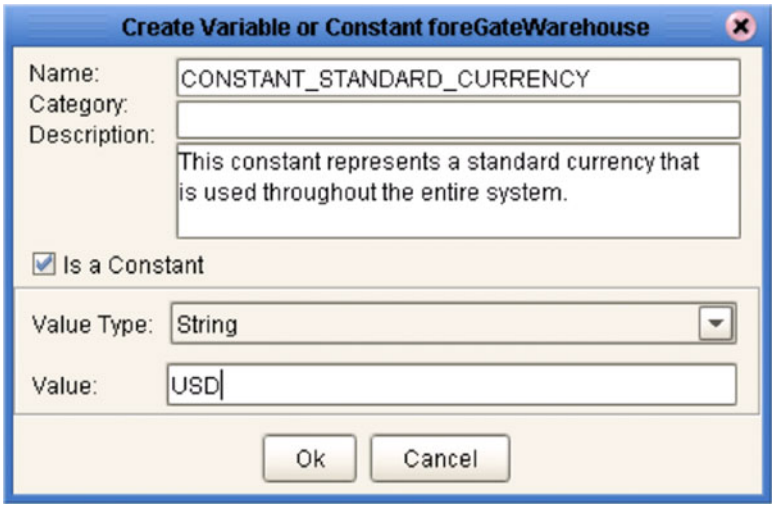
For example, a project variable is defined to represent a password of a database user in a target environment (see Figure 47). System managers will assign an actual value to this variable in the deployment profile editor. The value of the assigned project variable—an Environment constant— is then used to connect the database in the target environment.

Figure 47 Project Variable Creation



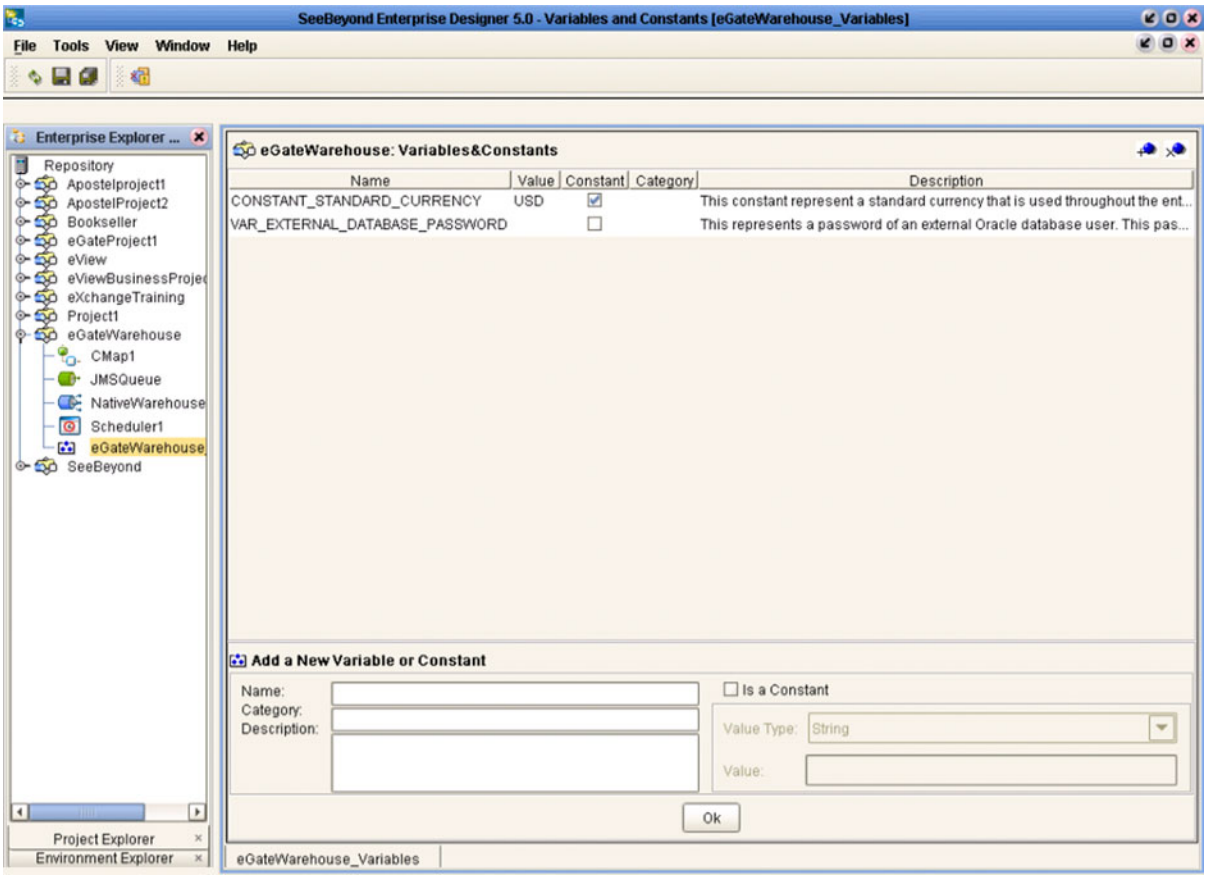
A constant is a name-value pair; when you create a constant you assign a permanent value to it, which cannot be overridden. An example of this would be a standard currency used globally throughout the system (see Figure 48).

Figure 48 Project Constant Creation



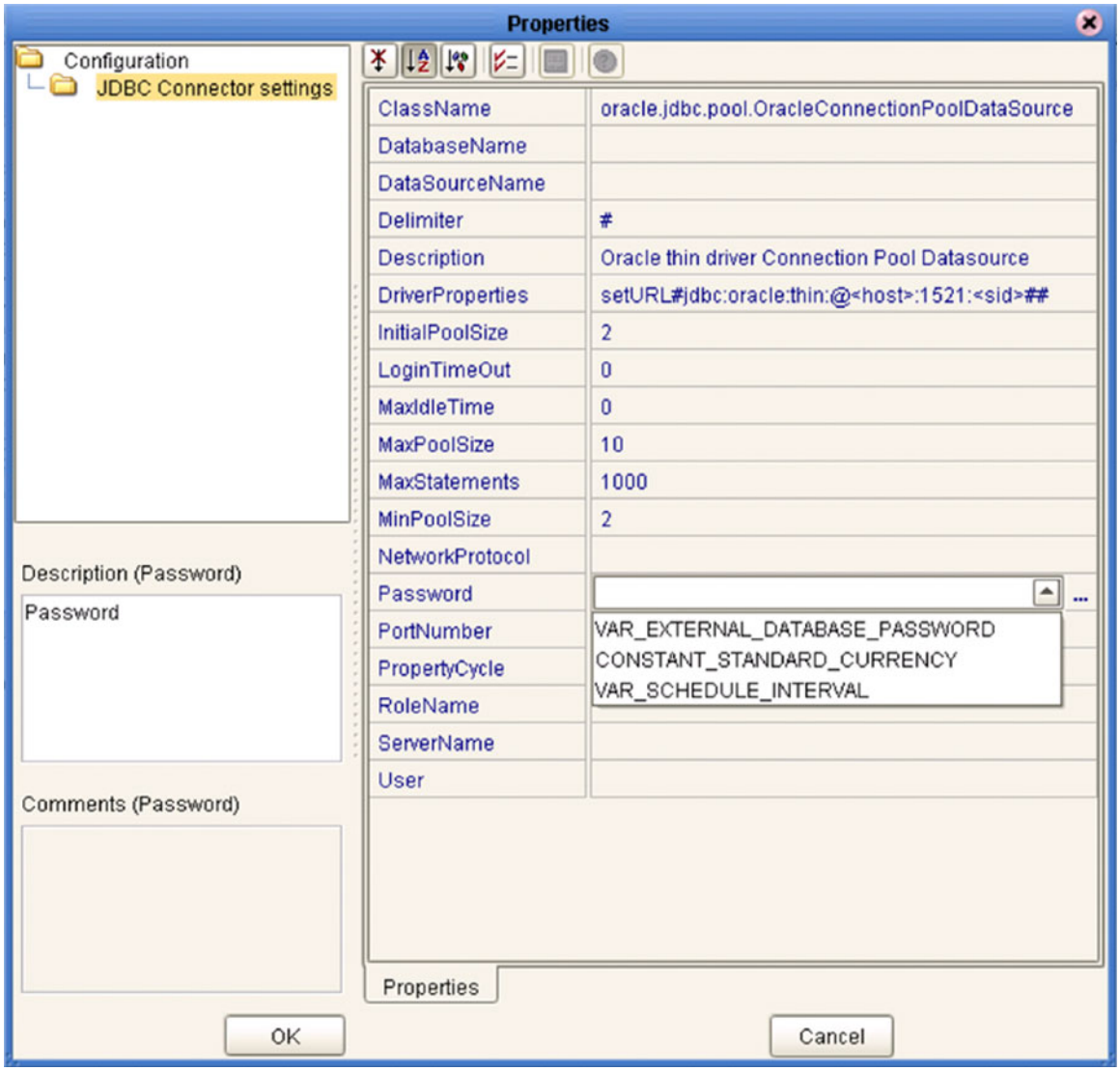
These constants and variables are automatically added to a Variables and Constants object group within the Project (see Figure 49).

Figure 49 Variables and Constants Object Group



The Project variables and constants can be referenced as properties within the Project. For example, the password variable described previously can be selected from the Variables and Constants object group to provide the Password property in the JDBC Connector settings dialog box (see Figure 50).

Figure 50 Connector Properties



See **Mapping Variables** on page 157 for information on setting the values for Project variables.

Chapter 6

Web Services

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

6.1 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.

6.2 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. The associated business processes are developed within eInsight Enterprise Service Bus. See **Building a Web Client** on page 83 and **Building a Web Server** on page 92.

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 106.

- **WSDL Editor**

  See the *eInsight Enterprise Service Bus User's Guide*.

- **WSDL Interface Designer**

  See the *eInsight Enterprise Service Bus User's Guide*.

- **WSDL Viewer**

  See the *eInsight Enterprise Service Bus User's Guide*.

- **UDDI Repository**

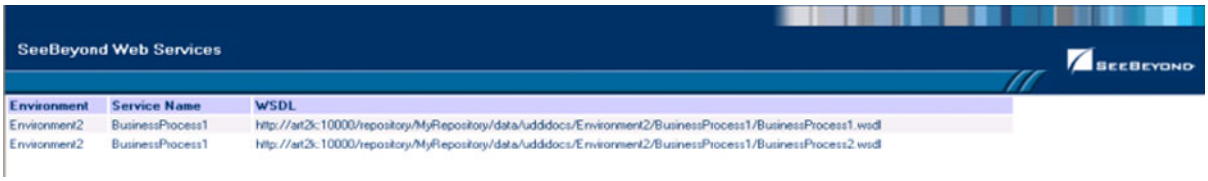
  All ICAN objects that can be accessed as Web services are presented in the SeeBeyond UDDI Repository. See **UDDI Repository** on page 81.

6.3 UDDI Repository

In general, all ICAN objects that expose themselves as a Web service (such as an eInsight business process) are presented in the SeeBeyond UDDI Repository (see Figure 51). The URL of this repository is:

```
http://ICAN_Suite_host_name:enterprise_manager_installation_port/
     stcuddi
```

Figure 51 SeeBeyond UDDI Repository



Each entry in the UDDI Repository includes:

- The ICAN environment name.
- The actual (Web) Services name.
- The location of the Web Service's WSDL file.

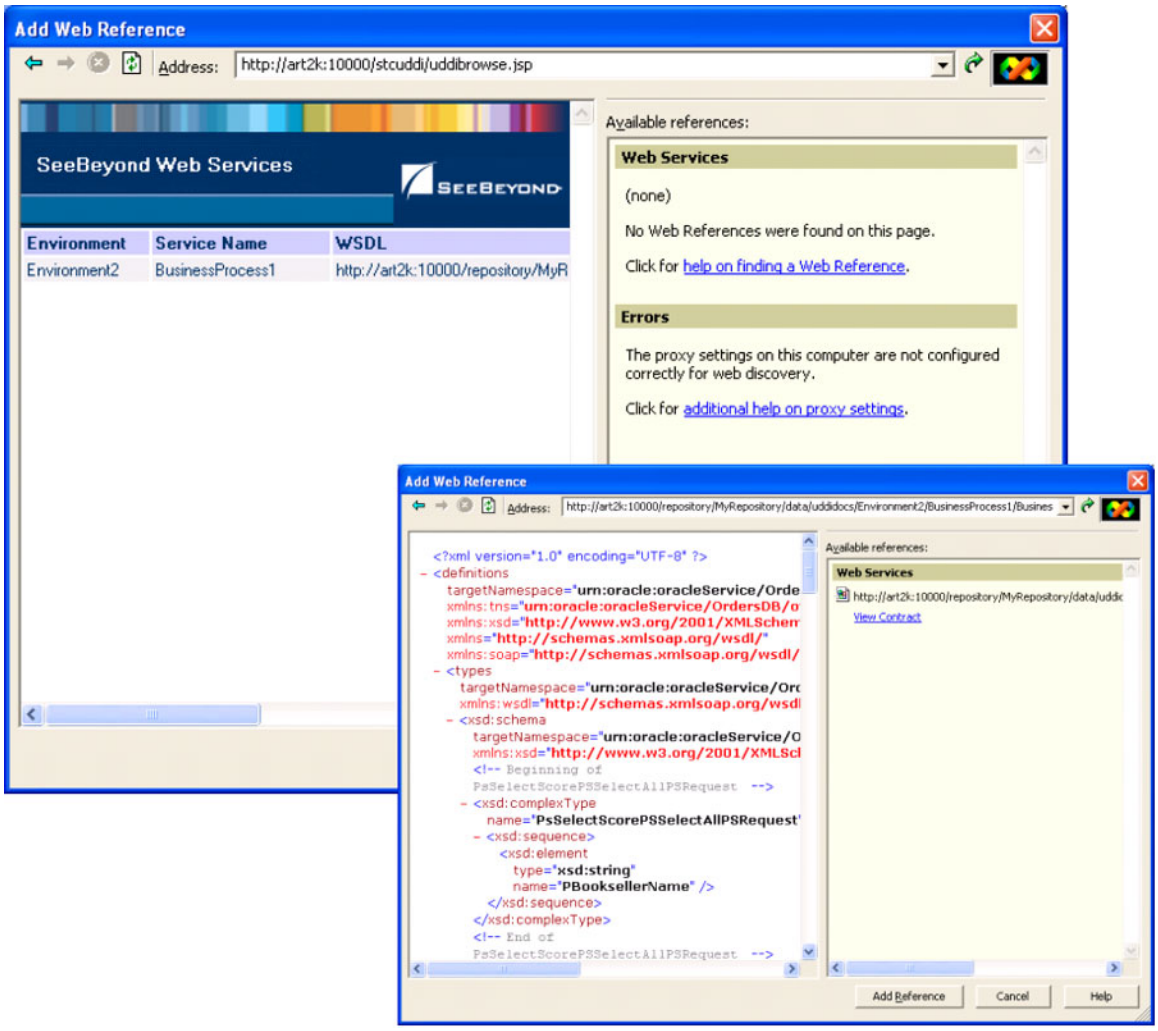
By selecting an entry its WSDL file is displayed, as shown in Figure 52.

Figure 52 Example Web Service WSDL File

```
<?xml version="1.0" encoding="UTF-8" ?>
- <definitions targetNamespace="urn:oracle:oracleService/OrdersDB/otdGetCreditScore" xmlns:tns="urn:oracle:oracleService/OrdersDB/otdGetCreditScore"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
  - <types targetNamespace="urn:oracle:oracleService/OrdersDB/otdGetCreditScore" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
    - <xsd:schema targetNamespace="urn:oracle:oracleService/OrdersDB/otdGetCreditScore" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
        <!-- Beginning of PsSelectScorePSSelectAllPSRequest  -->
      - <xsd:complexType name="PsSelectScorePSSelectAllPSRequest">
        - <xsd:sequence>
            <xsd:element type="xsd:string" name="PBooksellerName" />
          </xsd:sequence>
        </xsd:complexType>
        <!-- End of PsSelectScorePSSelectAllPSRequest  -->
        <!-- Beginning of PsSelectScorePSSelectOnePSResponseType  -->
      - <xsd:complexType name="PsSelectScorePSSelectOnePSResponseType">
        - <xsd:sequence>
            <xsd:element type="xsd:decimal" name="CREDIT_SCORE" />
          </xsd:sequence>
        </xsd:complexType>
        <!-- End of PsSelectScorePSSelectOnePSResponseType  -->
        <!-- Beginning of PsSelectScorePSSelectMultiplePSResponseType  -->
      - <xsd:complexType name="PsSelectScorePSSelectMultiplePSResponseType">
        - <xsd:sequence>
            <xsd:element type="xsd:long" name="rowCount" />
            <xsd:element type="xsd:decimal" name="CREDIT_SCORE" />
          </xsd:sequence>
        </xsd:complexType>
        <!-- End of PsSelectScorePSSelectMultiplePSResponseType  -->
        <!-- Beginning of PsSelectScorePSSelectAllPSResponseType  -->
      - <xsd:complexType name="PsSelectScorePSSelectAllPSResponseType">
        - <xsd:sequence>
            <xsd:element type="xsd:long" name="rowCount" />
```

The SeeBeyond UDDI Repository can be used in a third party tool, for example Microsoft Visual Studio (see Figure 53). In this example, a so-called *Web reference* (to the UDDI repository) is added to a C# project.

Figure 53 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 Collaborations (see **Using the WSDL Wizard** on page 106).

6.4 Building a Web Client

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

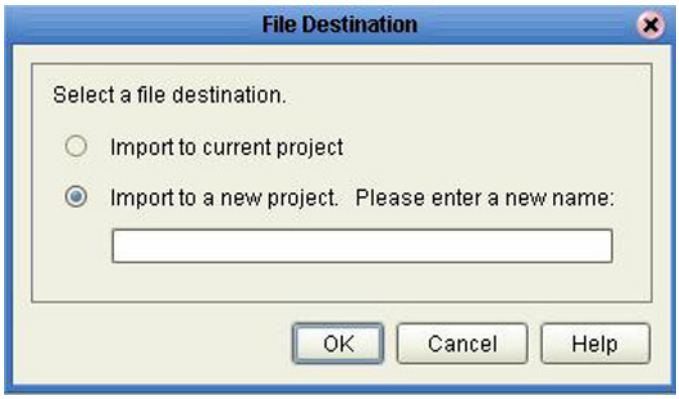
1. Build an Object Type Definition (OTD).
2. Develop a business process.
3. Create the eGate Project.
4. Deploy the Project to the selected Environment.

The Project used in the following example is available as **webclient.zip**, contained in the eGate User Guide Sample file included with this User's Guide.

To import the sample project

1. The sample files are uploaded with the User's Guide **.sar** file and downloaded from the Enterprise Manager's Documentation tab. Extract the samples from the Enterprise Manager to a local file.
2. From the Enterprise Designer's Project Explorer pane, right-click the Repository and click **Import Project** form the selection menu. The **Select File to Import** dialog box appears.
3. Browse to the directory that contains the sample project zip file. Select the sample file (**webclient.zip**) and click **Open**.
4. From the *File Destination* dialog box (see Figure 54), select **Import to a new Project**, enter the name of the Project, and click **OK**.

Figure 54 File Destination Dialog Box



5. After the import has successfully completed, select the **Repository** in the Project Explorer and click **Refresh All from Repository**.

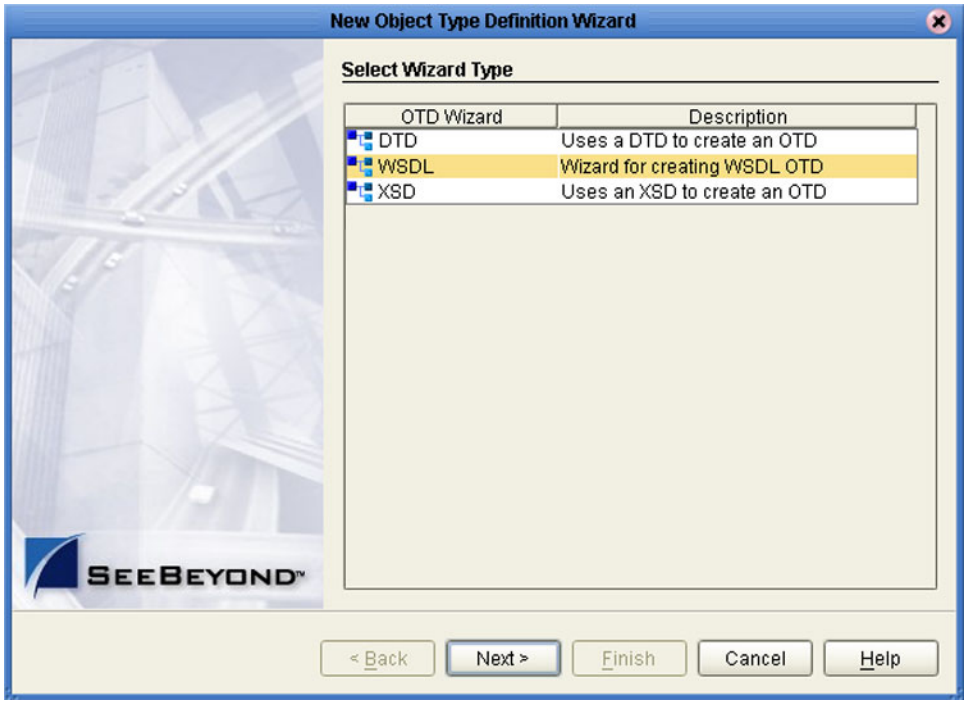
6.4.1 Object Type Definition

To create the WSDL OTD

1 In Project Explorer, create a new OTD.

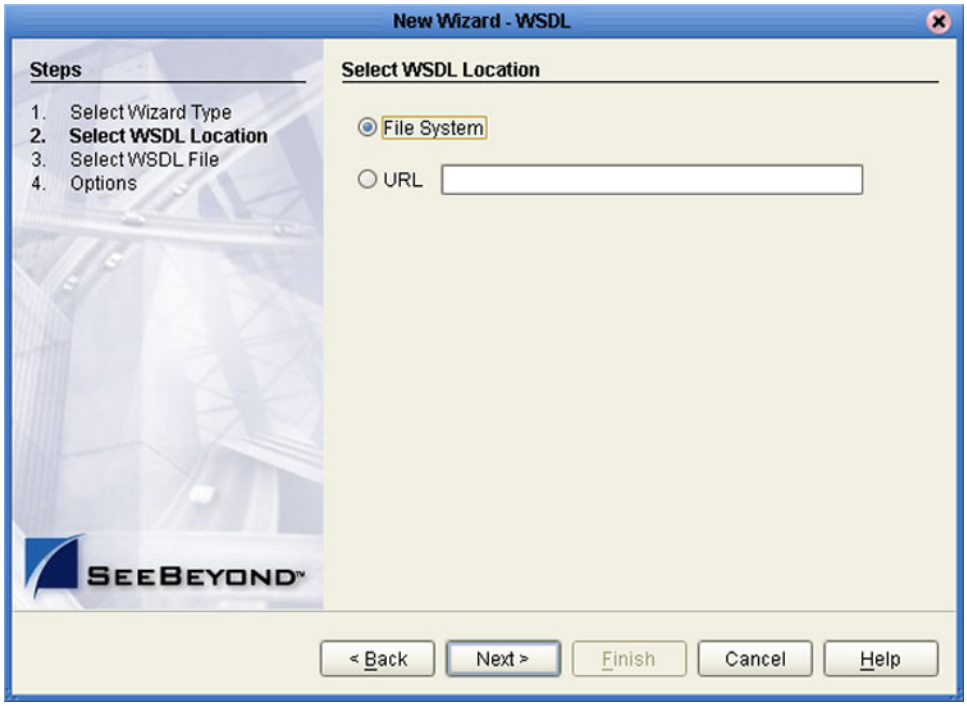
2 Select the **WSDL** OTD Wizard (see Figure 55).

Figure 55 Select WSDL Wizard



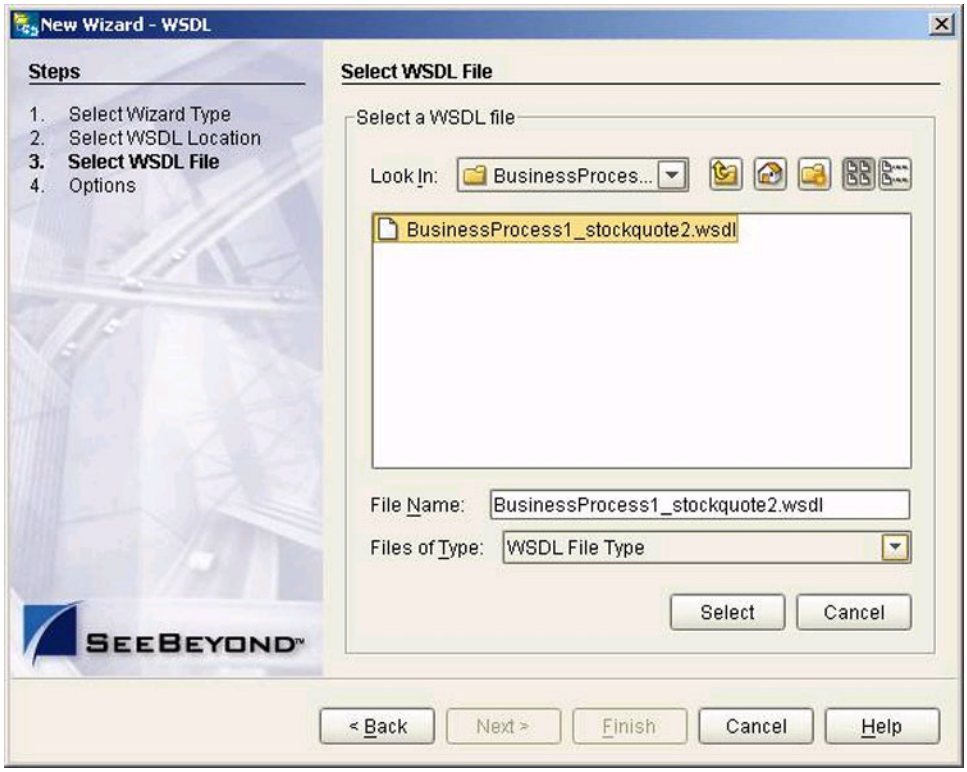
3 Select the WSDL file location and click **Next**. In this example, the file is located in the local file system (see Figure 56).

Figure 56 Select File Location



4 Select the WSDL file on which you want to base the OTD (see Figure 57). The file itself is shown in **Figure 62 on page 89**.

Figure 57 Select WSDL File

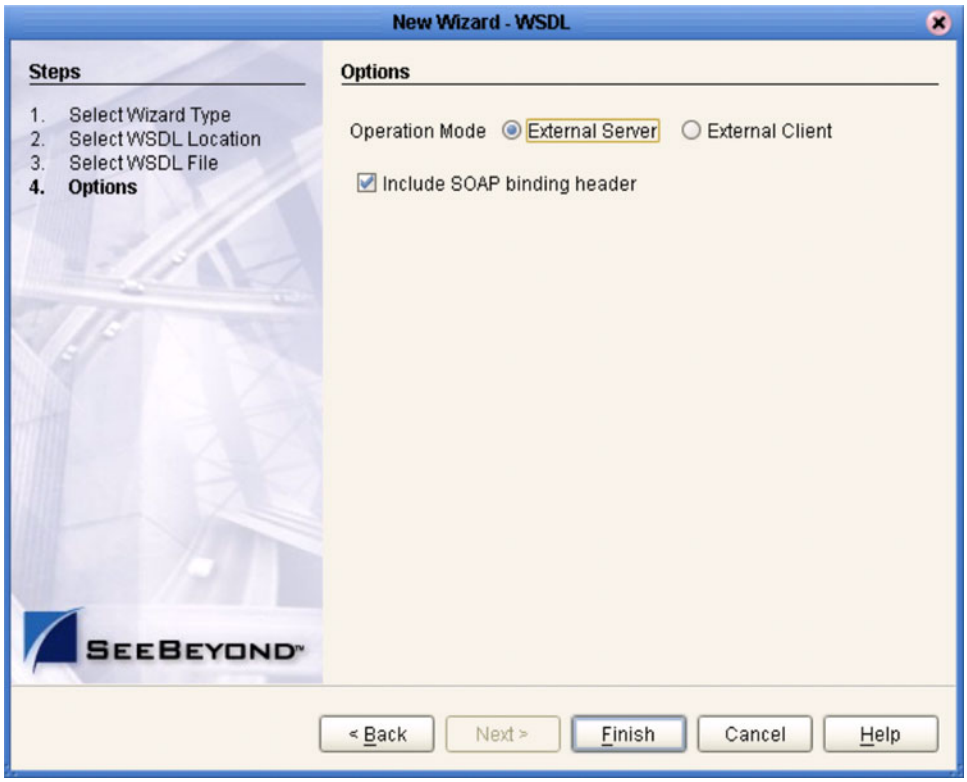


5 For a Web Client, select the following options (see Figure 58):

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

B Select **Include SOAP binding header** .

Figure 58 Select External Server



6 Click **Finish**.

6.4.2 eInsight Business Process

The example business process, developed in eInsight, is shown in Figure 59 (see the *eInsight Enterprise Service Bus User's Guide* for details).

Figure 59 Web Client Business Process

The **receive** rule for the business process is shown in Figure 60, and the **write** rule is shown in Figure 61.

Figure 60 Web Client Business Process *Receive* Rule

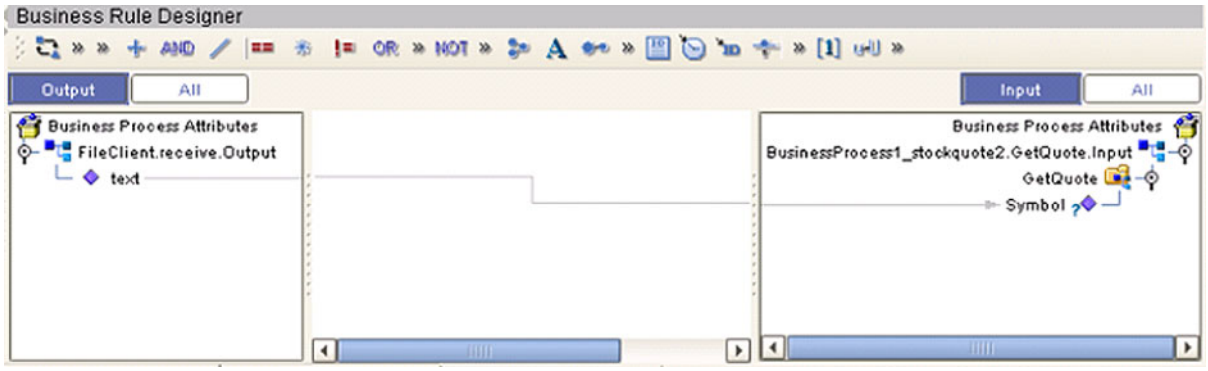
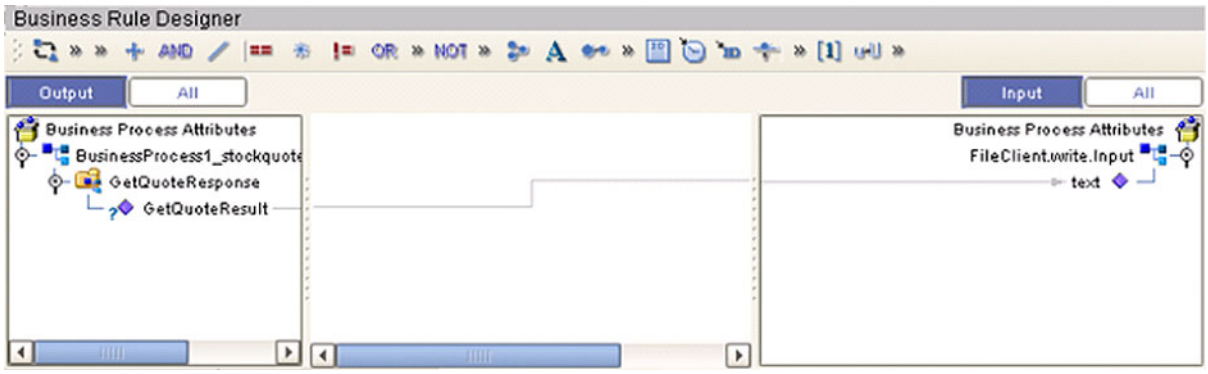


Figure 61 Web Client Business Process *Write* Rule



The WSDL file describing the business process is shown in Figure 62.

Figure 62 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"
```

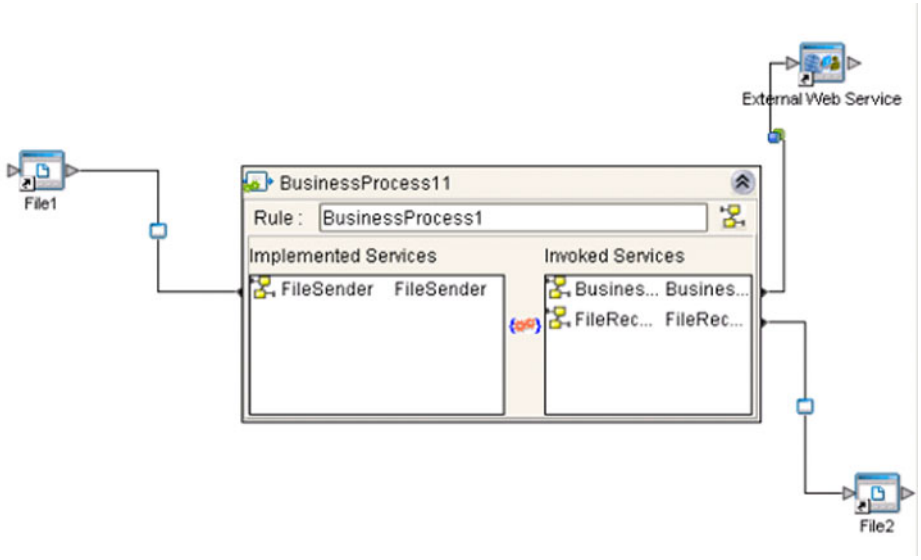
6.4.3 eGate Project

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.

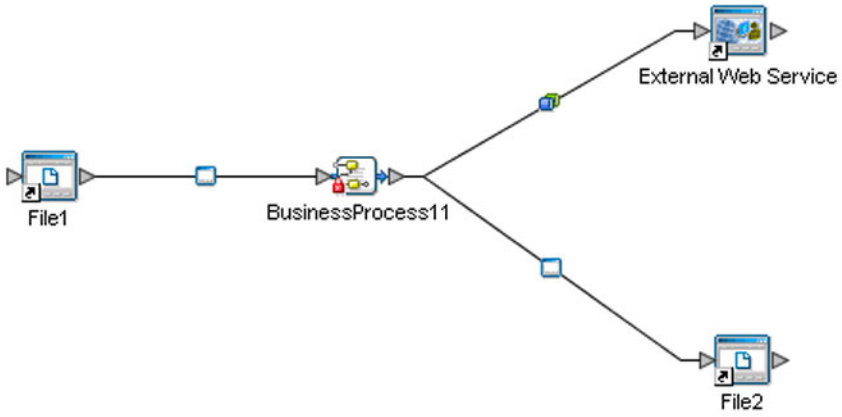
The business process is connected as shown in Figure 63.

Figure 63 Map Business Process



The completed Connectivity Map for the example Project is shown in Figure 64.

Figure 64 Web Client Connectivity Map

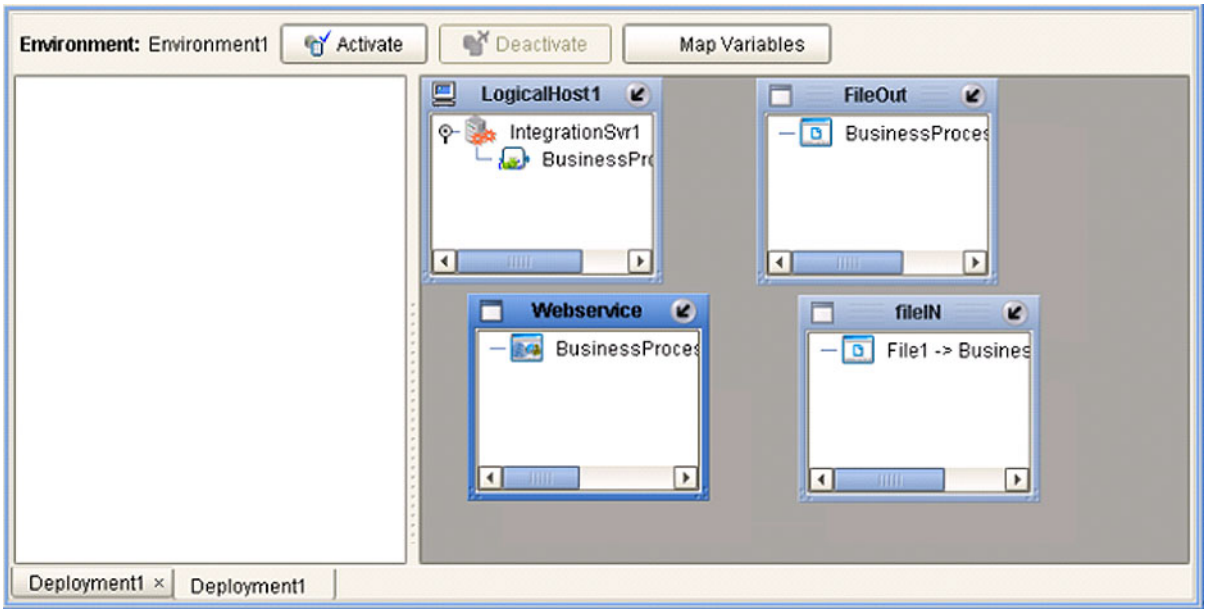


The Web client example Project appears in the Project Explorer as shown in Figure 65.

Figure 65 Web Client Example Project

The example Project is deployed as shown in Figure 66.

Figure 66 Project Deployment



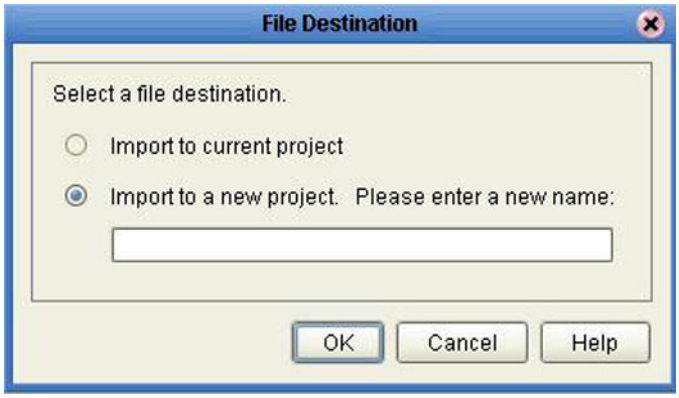
6.5 Building a Web Server

Here we briefly demonstrate the procedure for building a Web server. The Project used in the following example is available as **webserver.zip**, contained in the eGate User Guide Sample file included with this User's Guide.

To import the sample project

1. The sample files are uploaded with the User's Guide **.sar** file and downloaded from the Enterprise Manager's Documentation tab. Extract the samples from the Enterprise Manager to a local file.
2. From the Enterprise Designer's Project Explorer pane, right-click the Repository and click **Import Project** form the selection menu. The **Select File to Import** dialog box appears.
3. Browse to the directory that contains the sample project zip file. Select the sample file (**webserver.zip**) and click **Open**.
4. From the *File Destination* dialog box (see Figure 54), select **Import to a new Project**, enter the name of the Project, and click **OK**.

Figure 67 File Destination Dialog Box



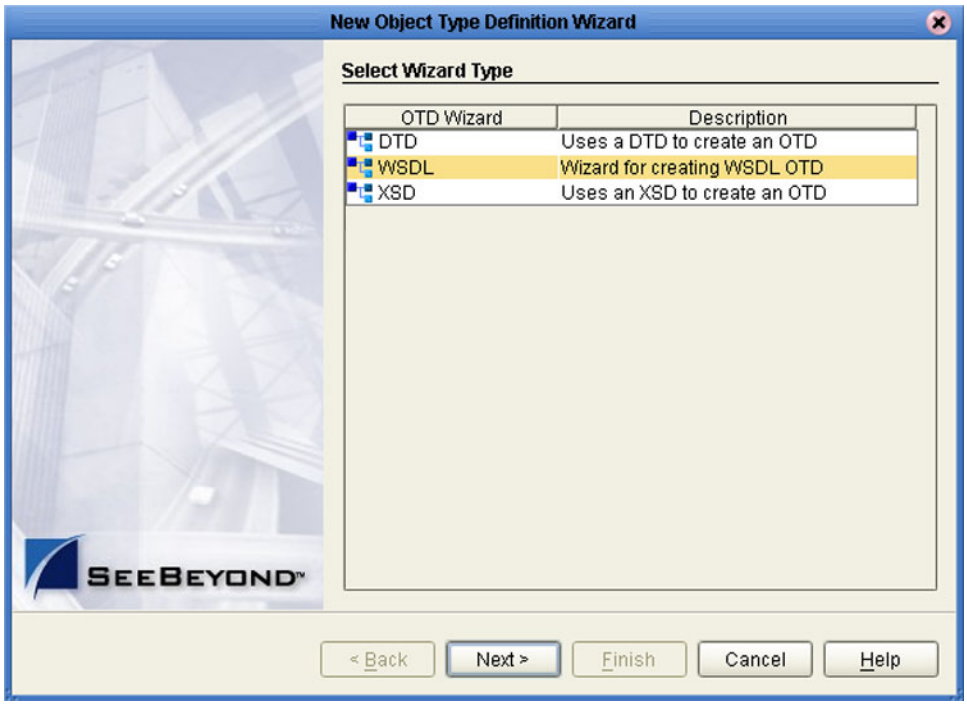
5. After the import has successfully completed, select the **Repository** in the Project Explorer and click **Refresh All from Repository**.

To build a Web Server Using the ICAN Suite

1 In Project Explorer, create a new OTD.

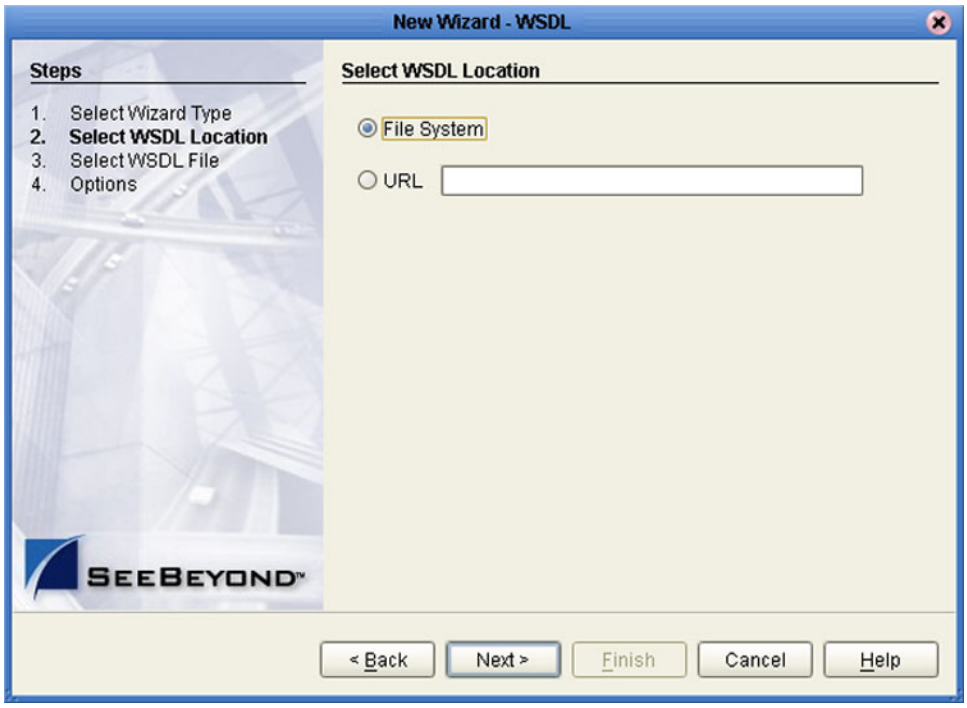
2 Select the WSDL OTD Wizard (see Figure 55).

Figure 68 Select WSDL Wizard



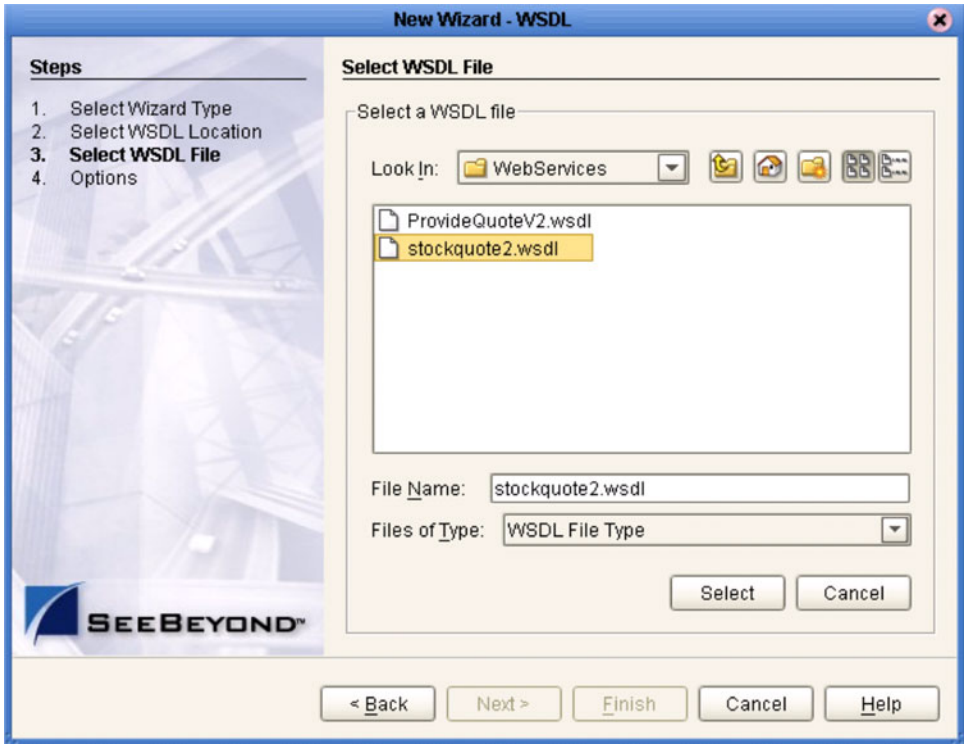
3 Select the WSDL file location (see Figure 69).

Figure 69 Select File Location



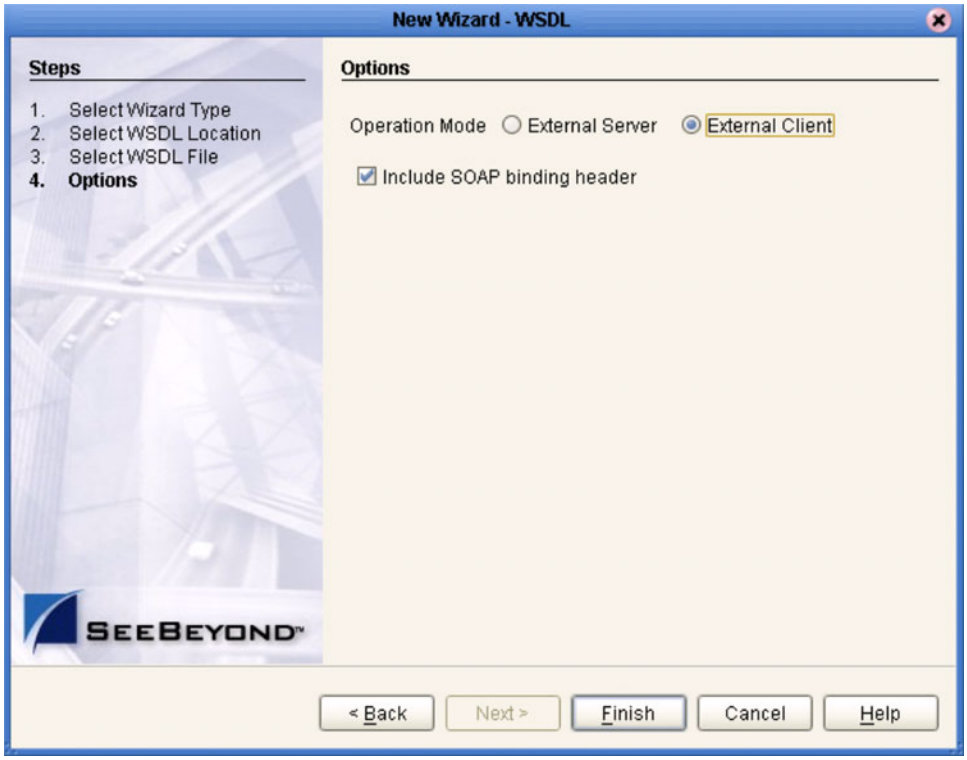
4 Select the WSDL file you want to use for the OTD (see Figure 70).

Figure 70 Select WSDL File



5 For a Web server, select External Client (see Figure 71).

Figure 71 Select External Client

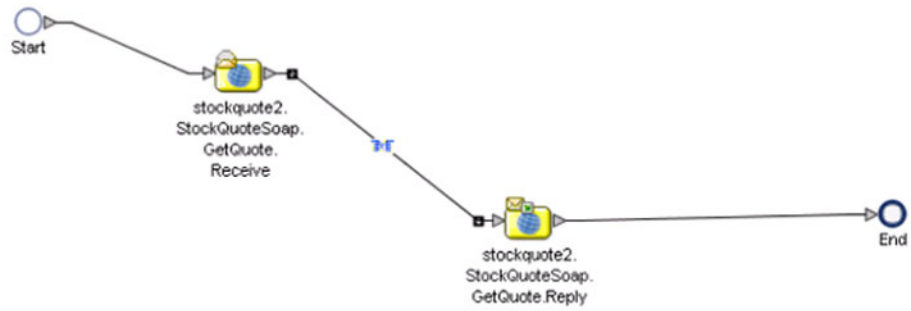


6 Click **Finish**.

6.5.1 eInsight Business Process

The example business process, developed in eInsight, is shown in Figure 72 (see the *eInsight Enterprise Service Bus User's Guide* for details).

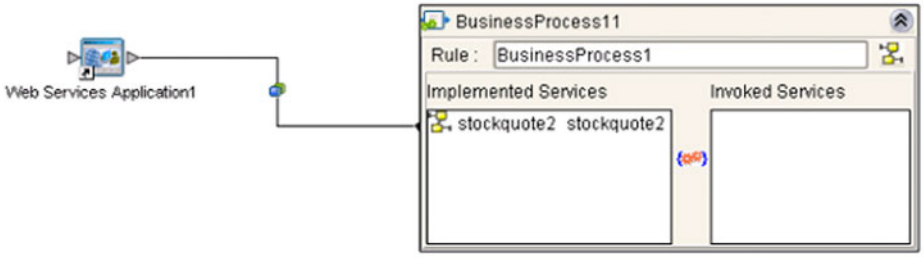
Figure 72 Web Server Business Process



6.5.2 eGate Project

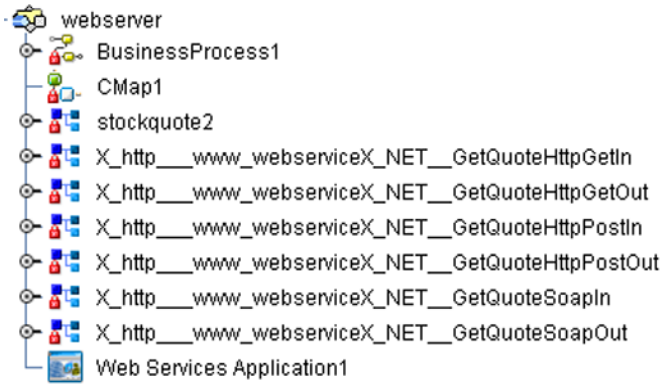
The business process is connected as shown in Figure 73, using the Enterprise Designer Connectivity Map Editor.

Figure 73 Connectivity Map



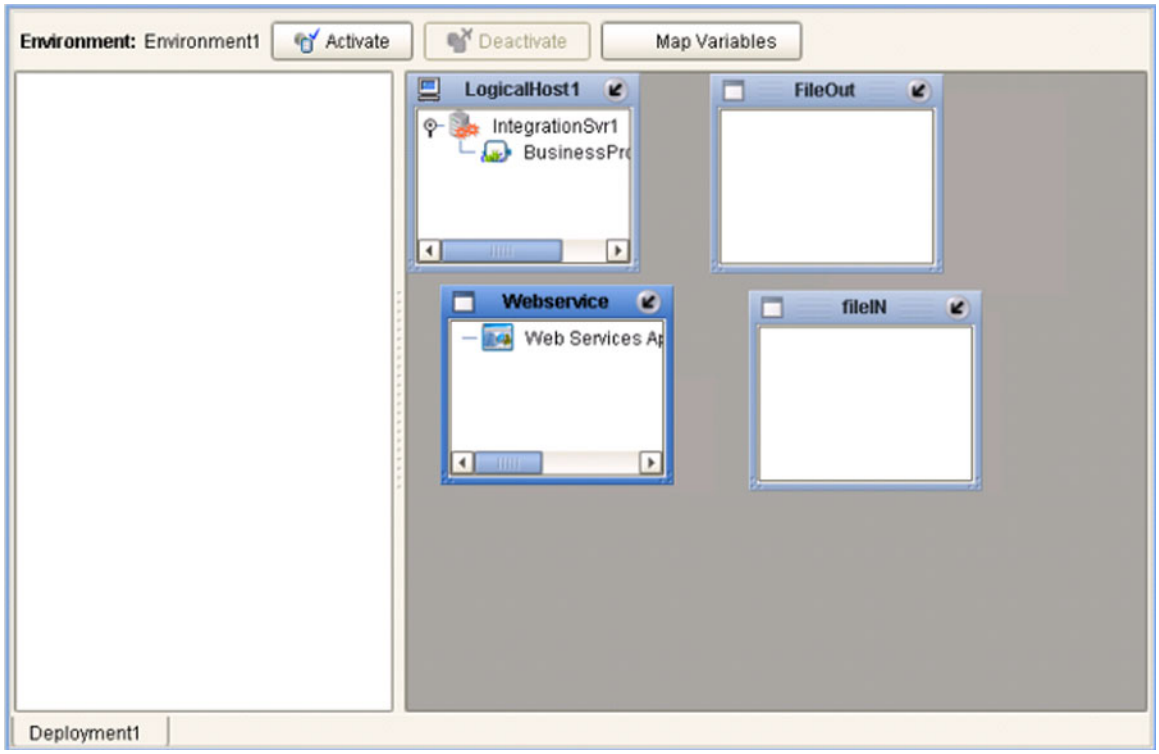
The Web server example Project appears in the Project Explorer as shown in Figure 74.

Figure 74 Web Server Example Project



The Project is deployed as shown in Figure 75.

Figure 75 Project Deployment



Chapter 7

Object Type Definitions

This chapter describes the OTD creation process. The Enterprise Designer includes two tools, the OTD Wizard and OTD Editor, to help you create and customize OTDs.

7.1 Overview

An Object Type Definition (OTD) is a description of a complex hierarchical data structure that can be accessed and manipulated by your code in a Collaboration. OTDs typically have a specific external representation format that is used to store and transport the OTD contents through the parts of a eGate Project. The OTD defines both the run-time structure and the external representation.

At run time, an OTD instance is accessed using the so-called *bean-like* accessors, or from BPEL using XPath expressions. In Java, the nodes comprising the hierarchy of the data structure have an interface similar to Java beans: each node has a set of properties with *get* and *set* methods to manipulate them. There is also a so-called generic interface (see **Appendix A**).

Typically, a collaboration will receive a message containing the external representation of a particular OTD. It will use the *unmarshal* method of an instance of that OTD to parse the data and make it accessible though the hierarchical data structure. Then it will perform some operation: for example, copying parts of the data to another OTD instance. Finally, it will invoke the *marshal* method on the other OTD instance to render the contents of its data structure as a single, serialized data stream for further transport.

7.1.1 The *Bean-like* Interface

The hierarchical data structure of an OTD is represented at run time by a set of generated Java classes. These classes follow the Java bean rule. They have a set of zero or more properties, each of which has a specific type and a given name, and may be optional and/or repeating. In contrast to regular bean properties, a OTD node property has two distinct names: a display name, that can be a virtually-arbitrary string, and a Java name that is the accessor basename.

For example, if a node has a property with Java name "X", then the implementing class for that node will have a method "*getX*". The Java name is normally derived from the display name, modified to suit the restrictions on Java identifiers, and supplied automatically by eGate.

7.1.2 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), XML Schema Definition (XSD), and various proprietary formats such as SAP BAPI. Some of these OTDs are *messagable*, others are API-based. Externally defined OTDs are read-only.

7.1.3 Building OTDs

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.

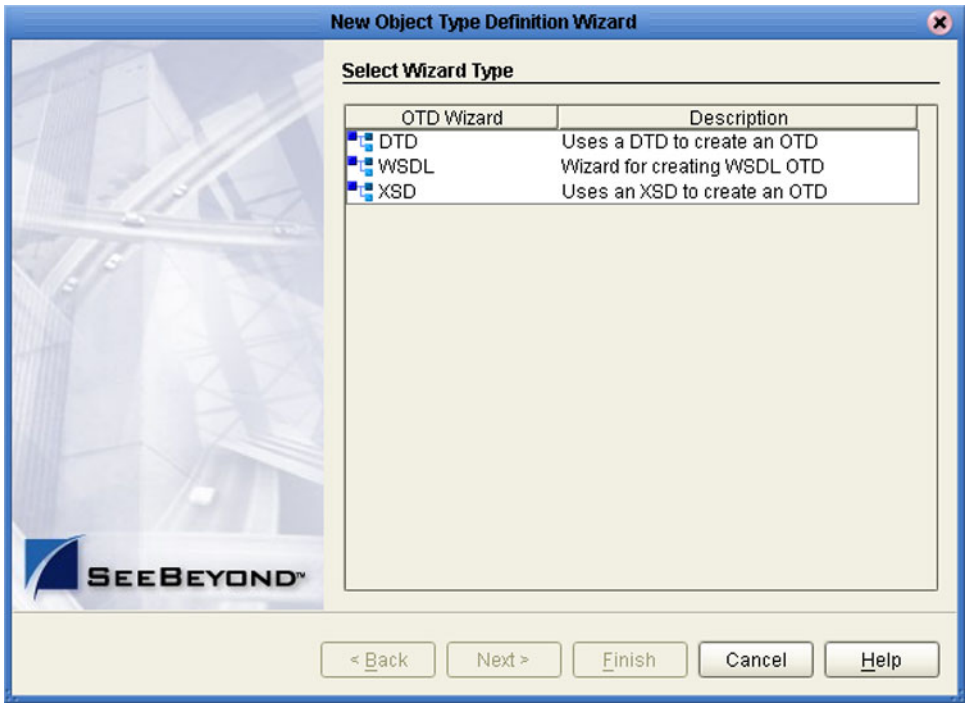
Note: *In order to be compatible with BPEL, the top element of an OTD must implement the OtdRoot and OtdNode interfaces; therefore, the top node cannot be a leaf node. As a result, OTDs cannot support a pure string-BLOB, since this would require the top element to be implemented as a java.lang.String object.*

7.2 Using the OTD Wizard

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 76. 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 102
- **Using the WSDL Wizard** on page 106
- **Using the XSD Wizard** on page 111

Figure 76 OTD Wizard Selection Dialog



Additional OTD Wizards are supplied with eGate Add-ons, 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 76.

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 19 for your reference.

Table 19 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.

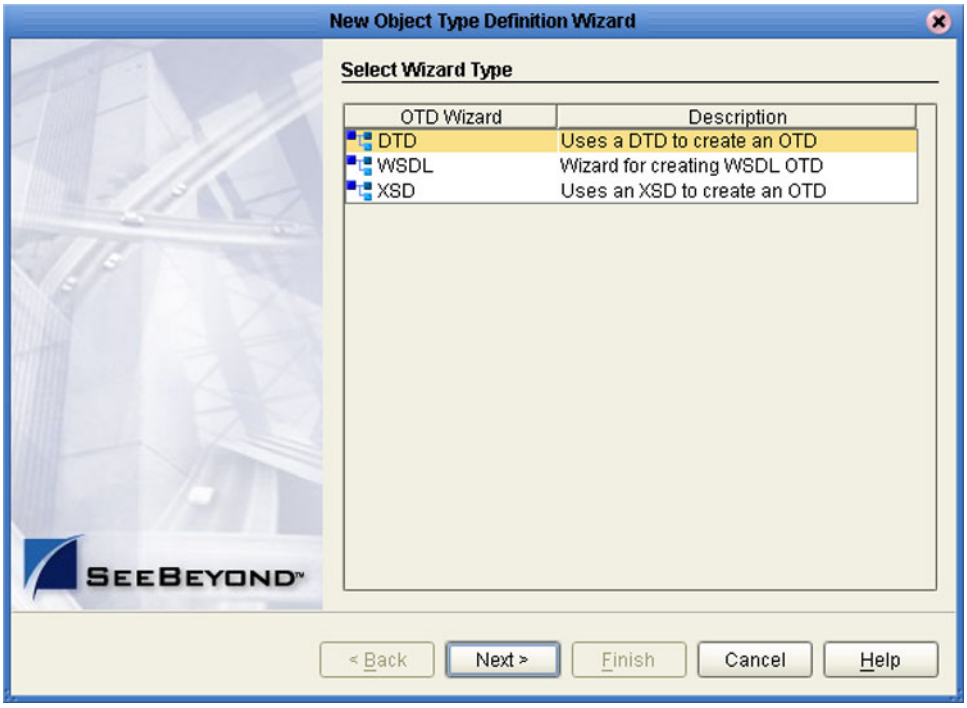
7.3 Externally-Defined OTDs

7.3.1 Using the DTD Wizard

To create an OTD file from a DTD file

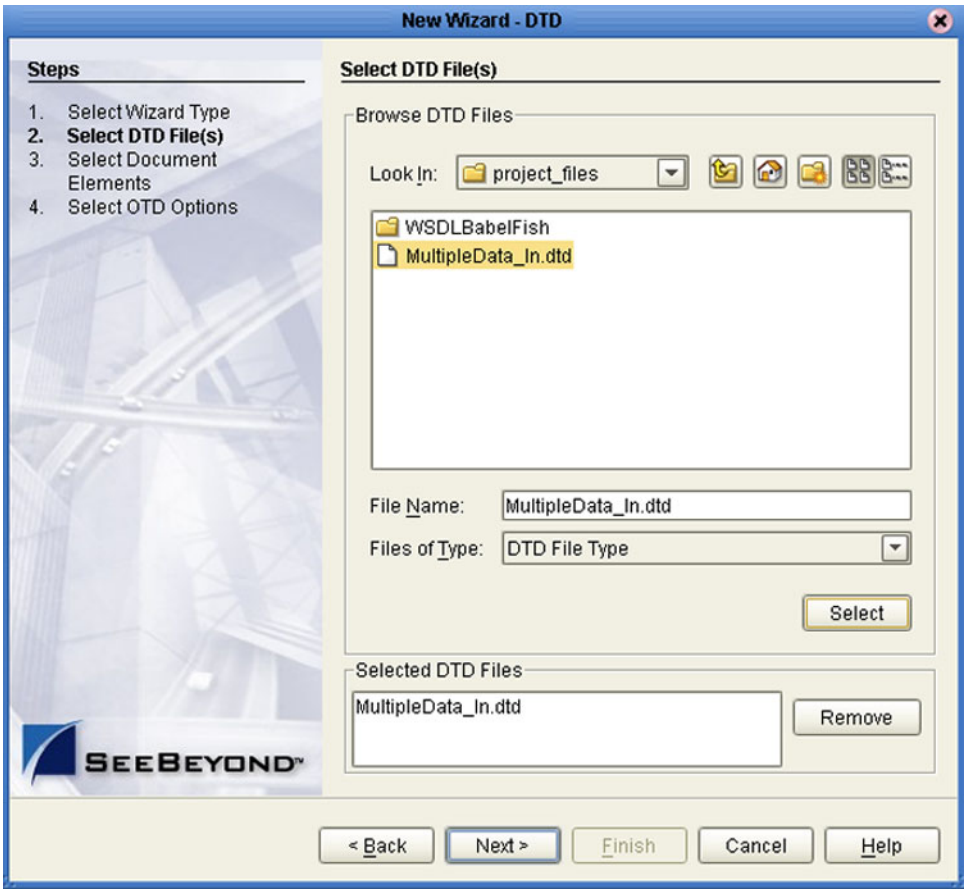
1 In the *Select Wizard Type* dialog, select **DTD** from the *OTD Wizard* list (see Figure 77) to create an OTD file from a Data Type Definition (DTD) file.

Figure 77 OTD Wizard Selection: DTD Wizard



2 Click **Next** to display the *Select DTD File(s)* dialog box, shown in Figure 78.

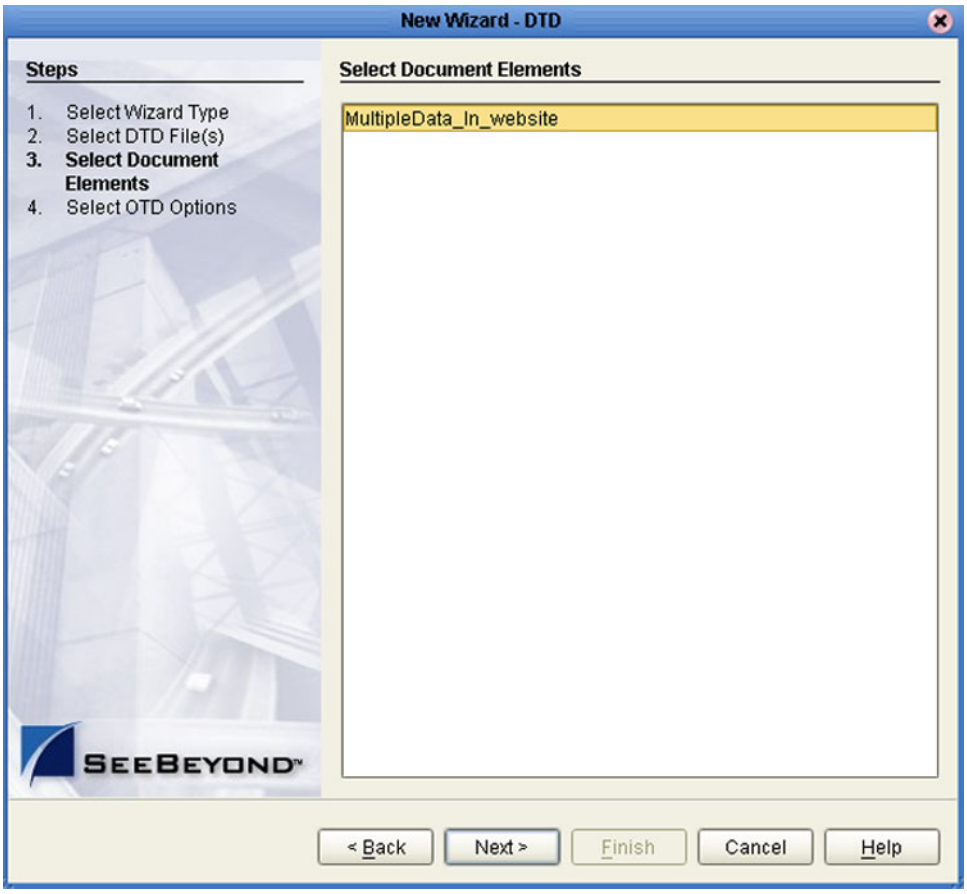
Figure 78 Select DTD File(s) Dialog Box



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*.

4 Click **Next** to display the *Select Document Elements* dialog box, shown in Figure 79.

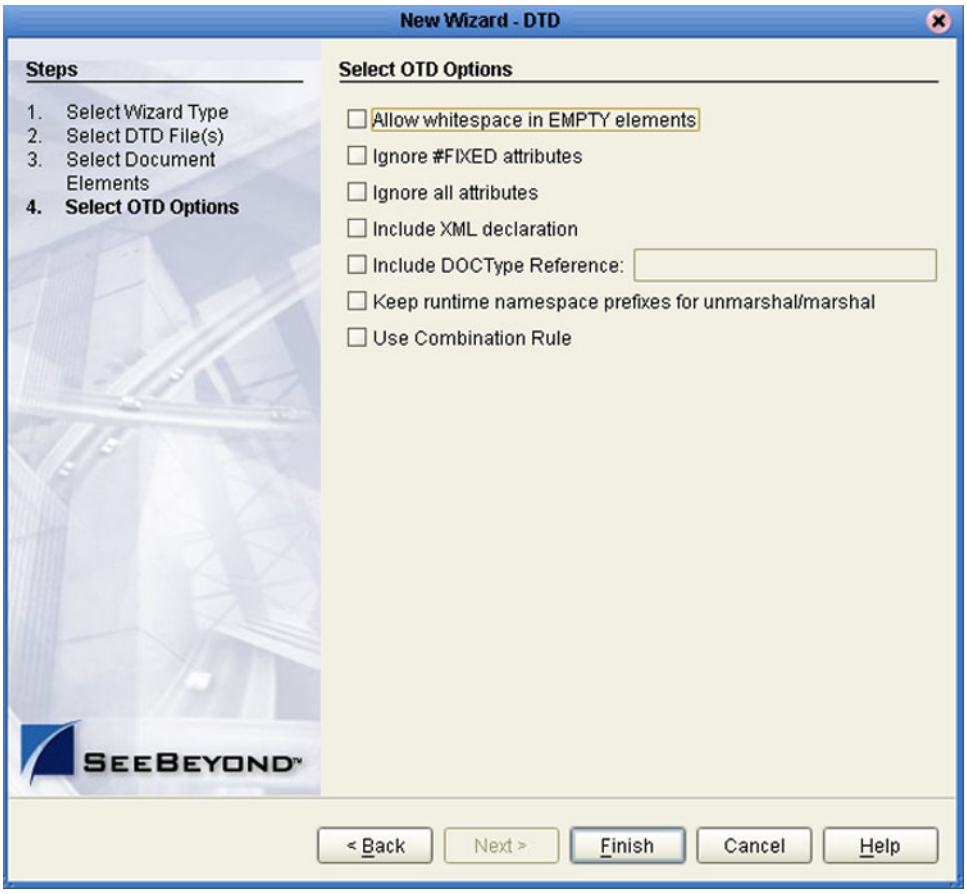
Figure 79 Select Document Elements Dialog Box



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 80.

Figure 80 Select OTD Options Dialog Box



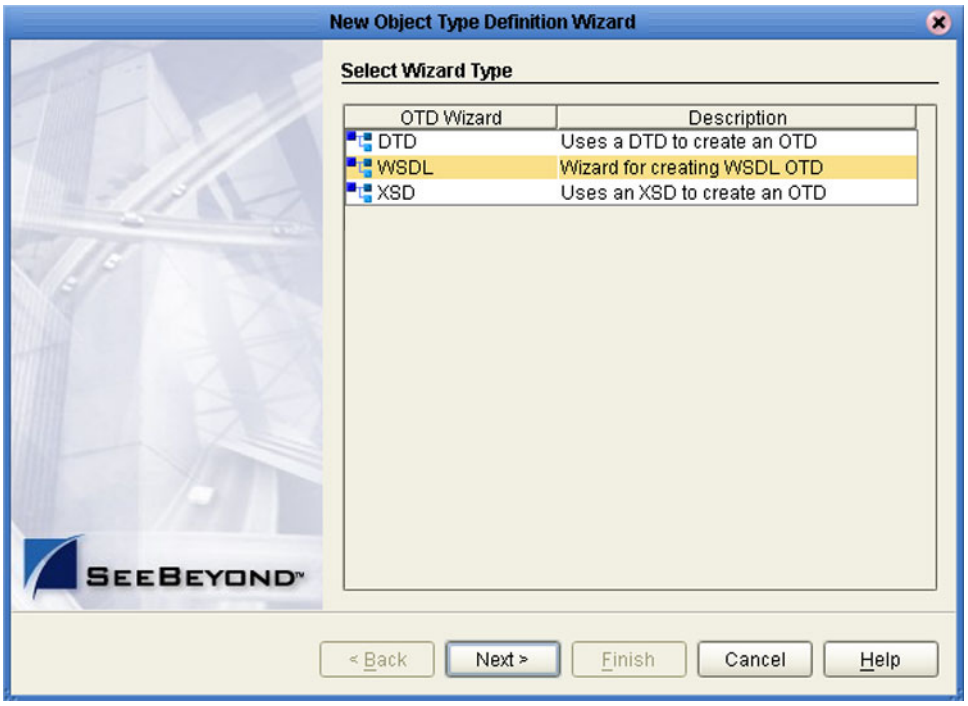
7 Select the check boxes next to the OTD options you want to enable, and click **Finish** to add the OTD to the Enterprise Designer with the selected OTD options.

7.3.2 Using the WSDL Wizard

To create an OTD file from a WSDL file

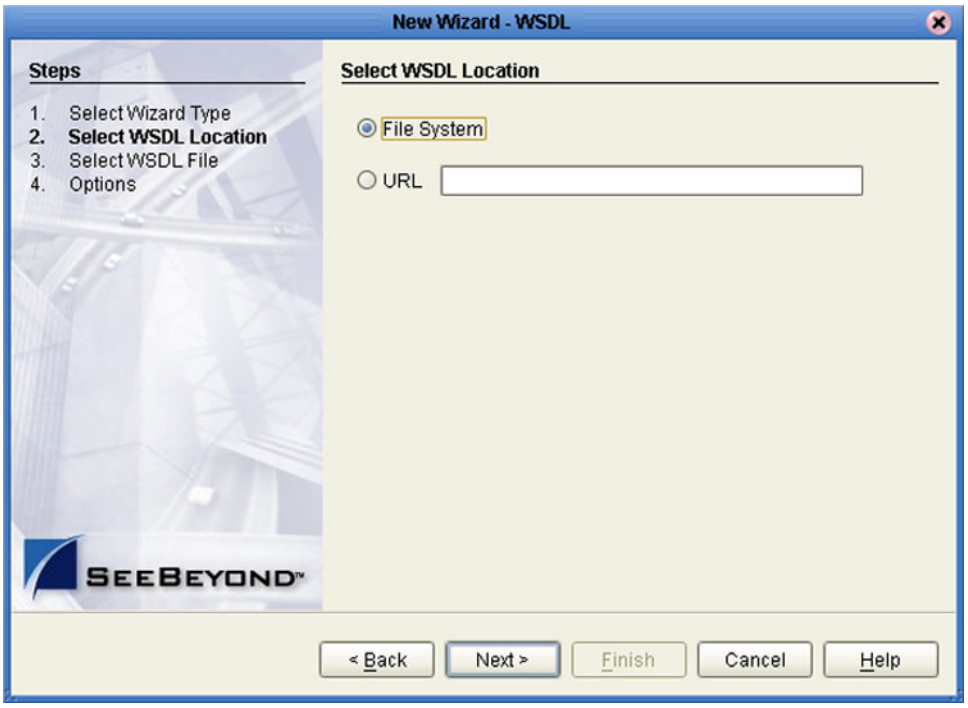
1 In the *Select Wizard Type* dialog, select **WSDL** from the *OTD Wizard* list (see Figure 81) to create an OTD file from an WSFL file.

Figure 81 OTD Wizard Selection: WSDL Wizard



2 Click **Next** to display the *Select WSDL File Location* dialog, shown in Figure 82

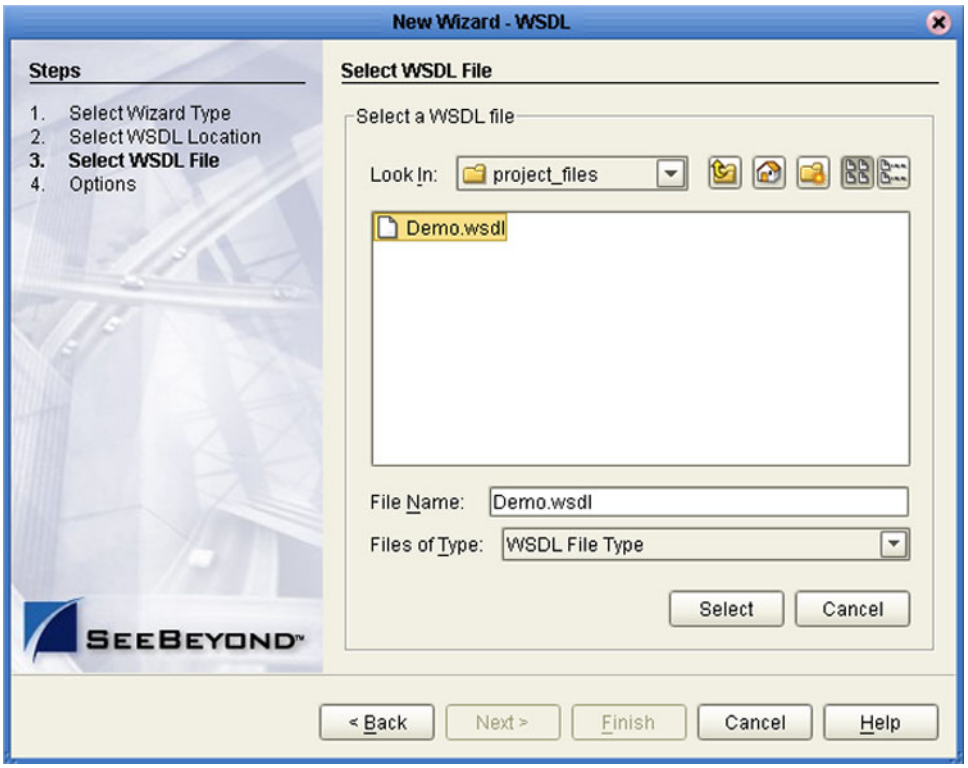
Figure 82 WSDL Wizard: Select WSDL Location



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 83.

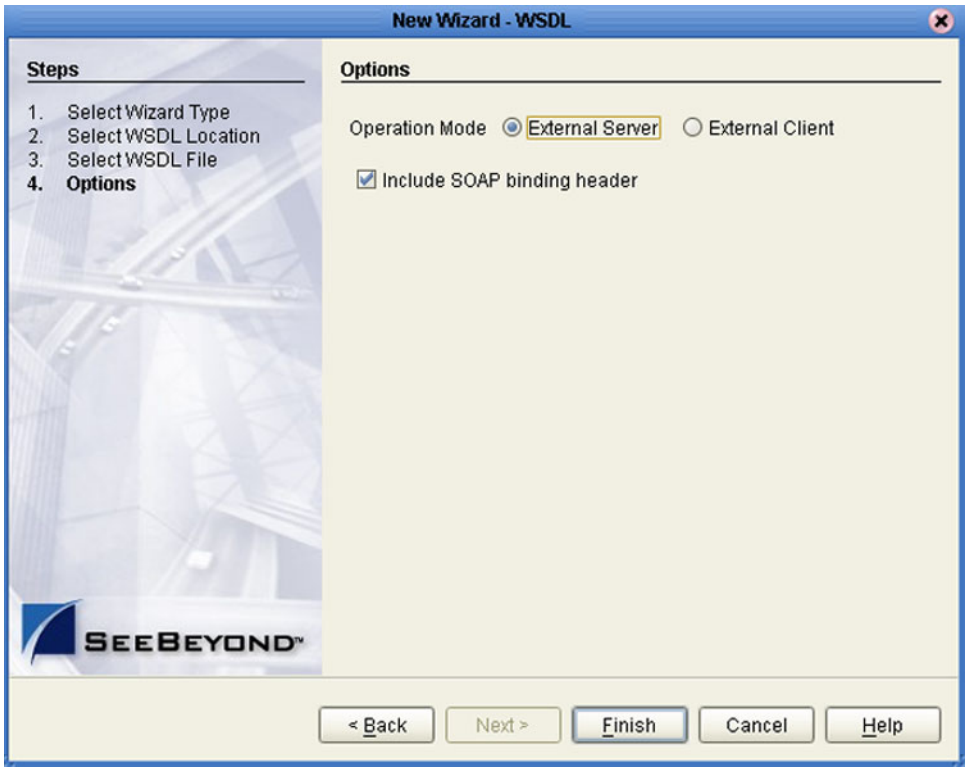
Figure 83 WSDL Wizard: Select WSDL File



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 84.

Figure 84 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 the Enterprise Designer with the selected OTD options.

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:

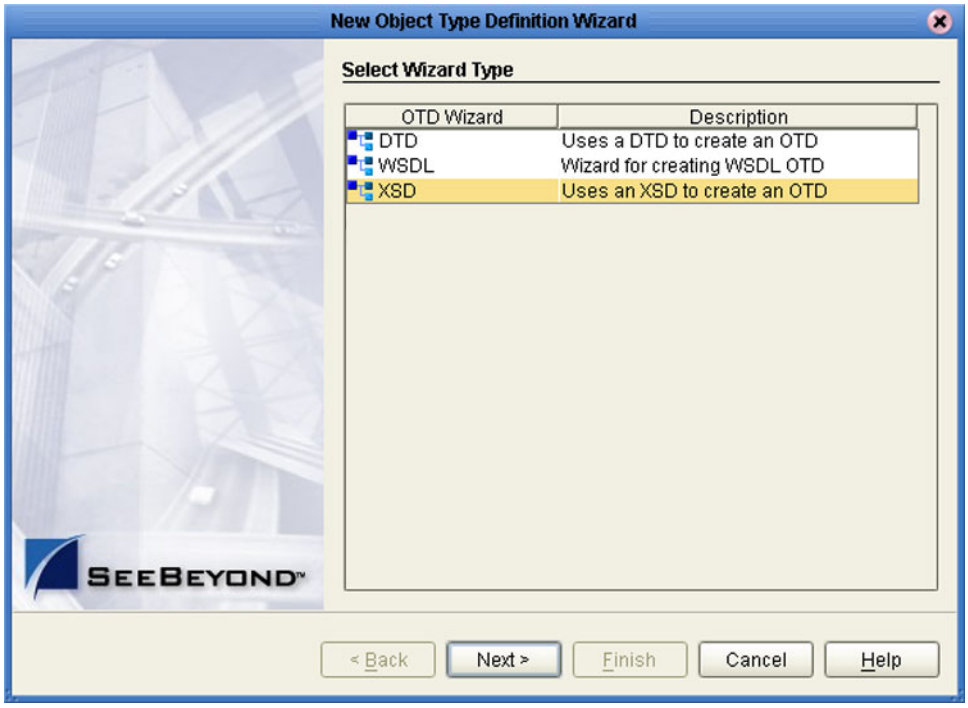
- **One-way:** The server receives a message from the client; also referred to as "fire and forget."
- **Request-response:** The server receives a message from the client and sends a correlated message back

7.3.3 Using the XSD Wizard

To create an OTD file from an XSD file

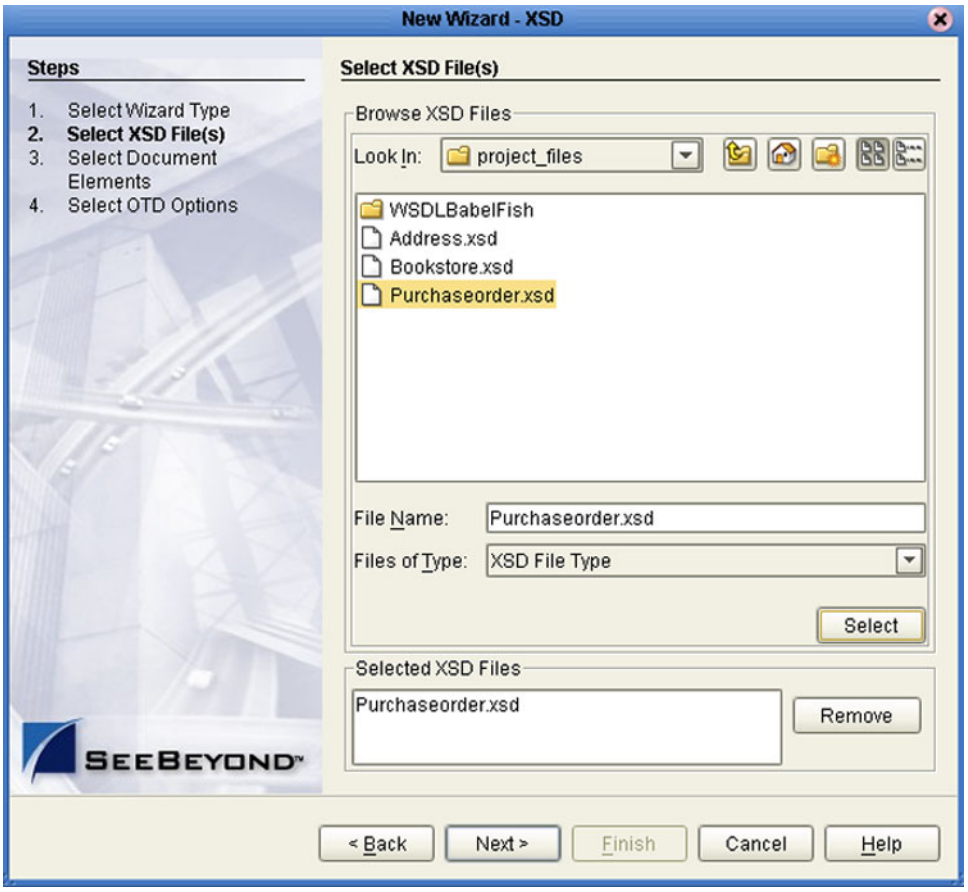
1 In the *Select Wizard Type* dialog, select **XSD** from the *OTD Wizard* list (see Figure 85) to create an OTD file from an XSD file.

Figure 85 OTD Wizard Selection: XSD Wizard



2 Click **Next** to display the Select XSD File(s) dialog box, shown in Figure 86.

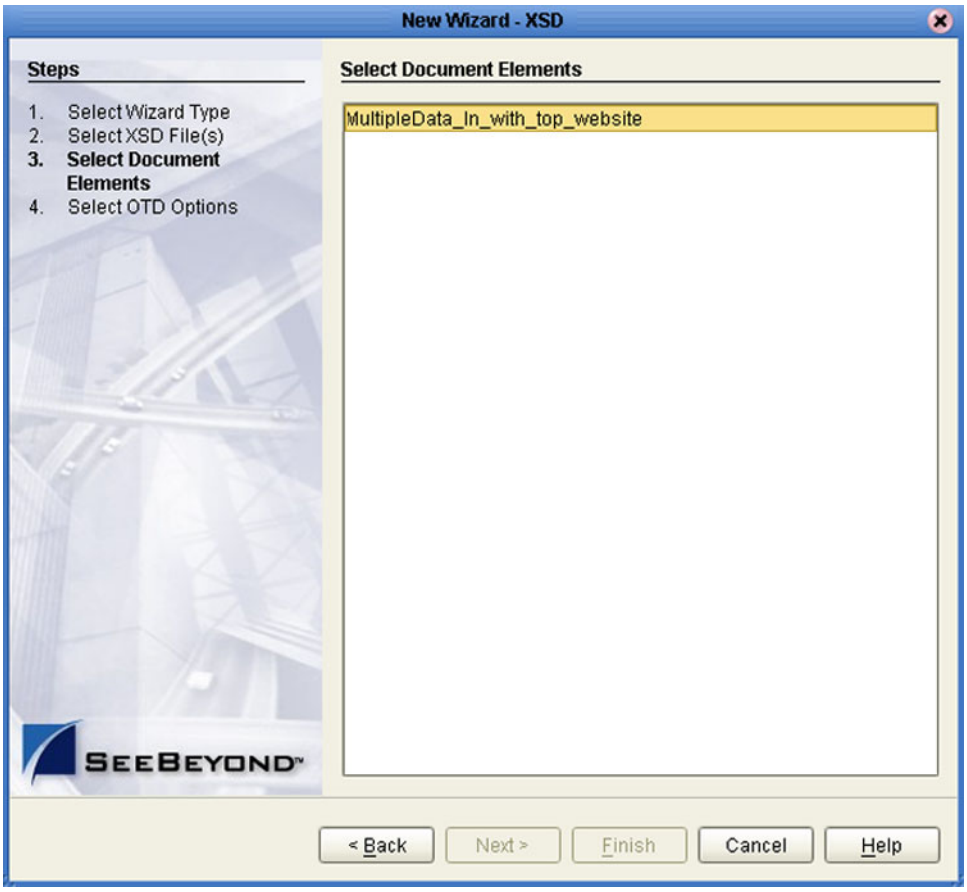
Figure 86 XSD Wizard: Select XSD File(s)



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*.

4 Click **Next** to display the *Select Document Elements* dialog box, shown in Figure 79.

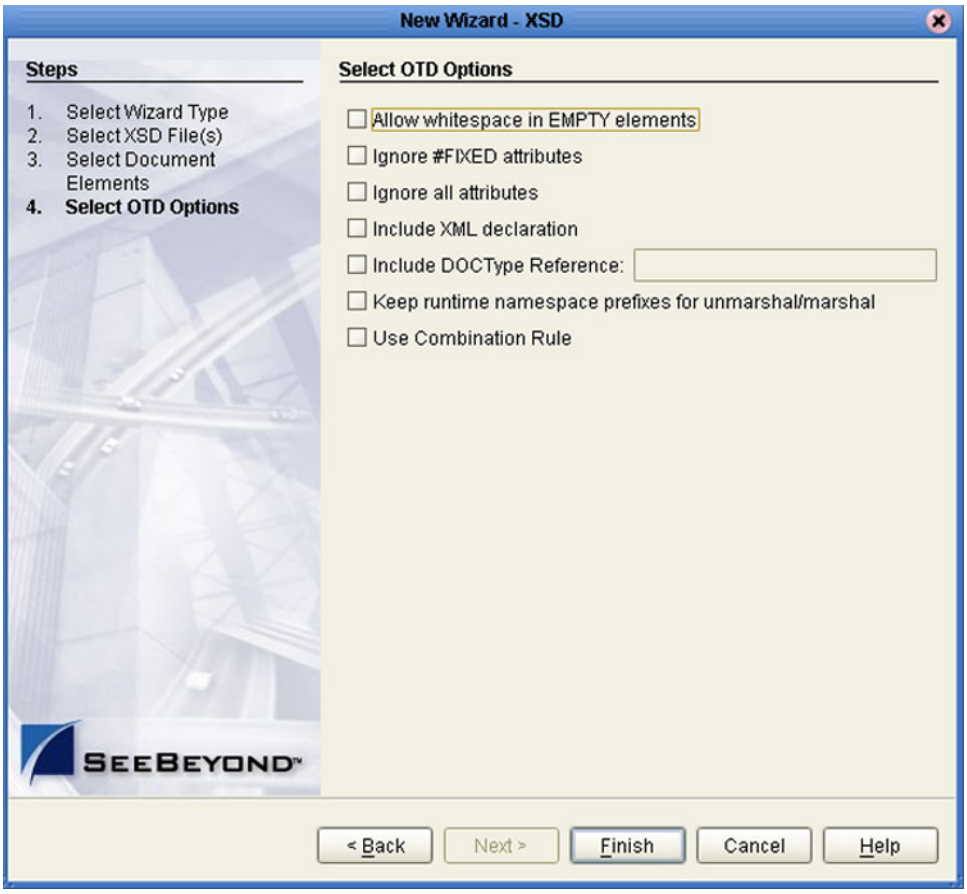
Figure 87 Select Document Elements Dialog Box



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 80.

Figure 88 Select OTD Options Dialog Box



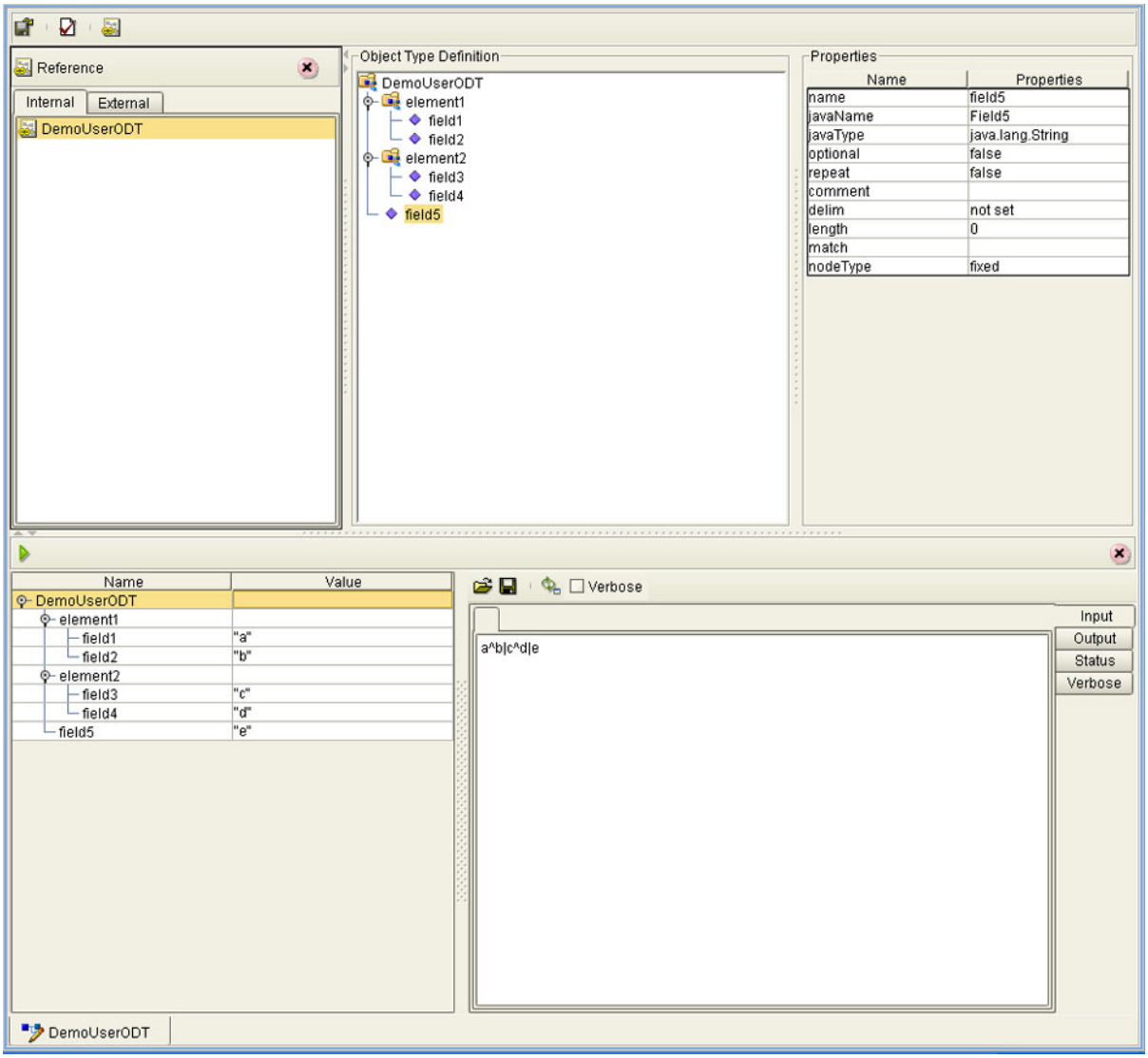
7 Select the check boxes next to the OTD options you want to enable, and click **Finish** to add the OTD to the Enterprise Designer with the selected OTD options.

7.4 Using the 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 89. 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.

Important: *If you delete an OTD in the Project Explorer, any Collaboration Definitions that have been built using that OTD will be affected. It is recommended that you run the Impact Analyzer before attempting to delete any OTDs (see* **Impact Analyzer** *on page 56**).*

Figure 89 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.
- **Tester**
  This area displays in the bottom part of the window when you click **Tester**. Use this area to perform tests on the contents of the OTD.
- **Toolbars**
  Several toolbars appear in the OTD Editor, containing icons as described in Table 20.

Table 20 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.
	Sort by Name	Sorts list alphabetically by name.
	Run Tester	Runs the tester with the entered values.
	Open	Displays file browser.
	Save	Saves displayed file.
	Refresh	Repopulates the OTD object elements with the values from the data display panel.

7.4.1 Node Management

The OTD Editor allows you to:

- **Add** nodes and elements to an OTD.
- **Delete** nodes and elements from an OTD.

  When a node is *deleted*, both the node and its associated 'children' (data elements) are deleted.
- **Prune** nodes in an OTD.

  When a node is *pruned*, only its associated 'children' (data elements) are deleted, while the node itself is preserved. Pruning can only be performed on nodes.

These commands are accessed from the node context menu.

7.5 Using the OTD Tester

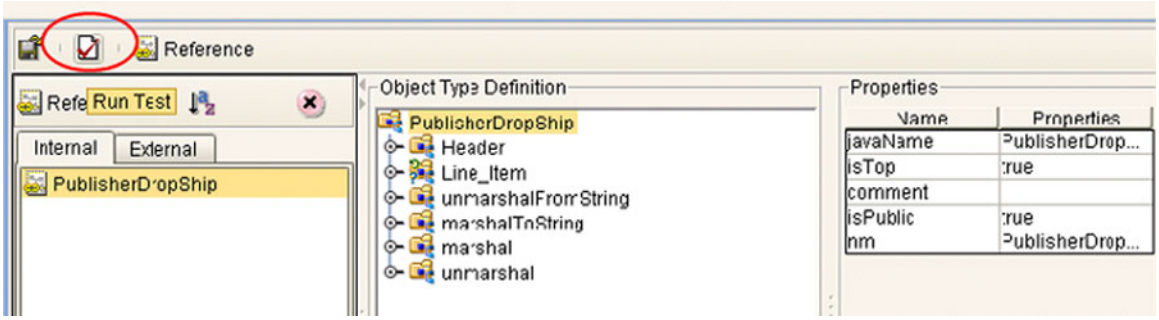
The OTD tester provides a facility to verify the correctness of OTDs, for example to:

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

To use the OTD tester

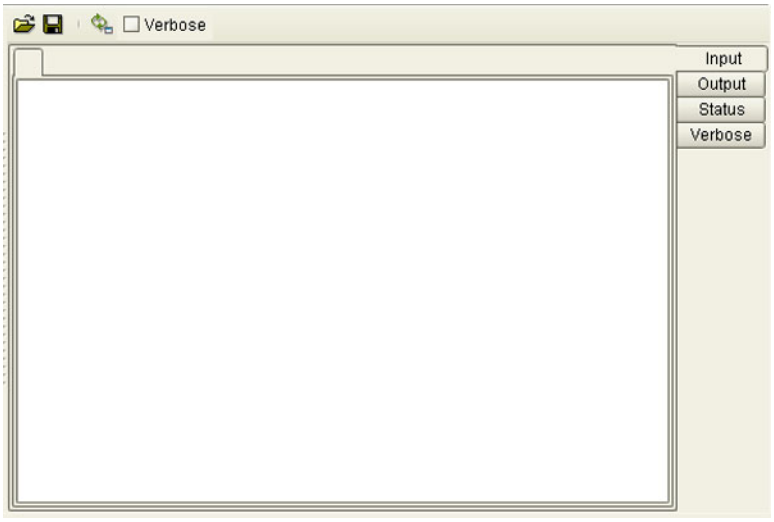
1. Open or create an OTD.
2. Click the **Tester** icon (see Figure 90).

Figure 90 OTD Tester



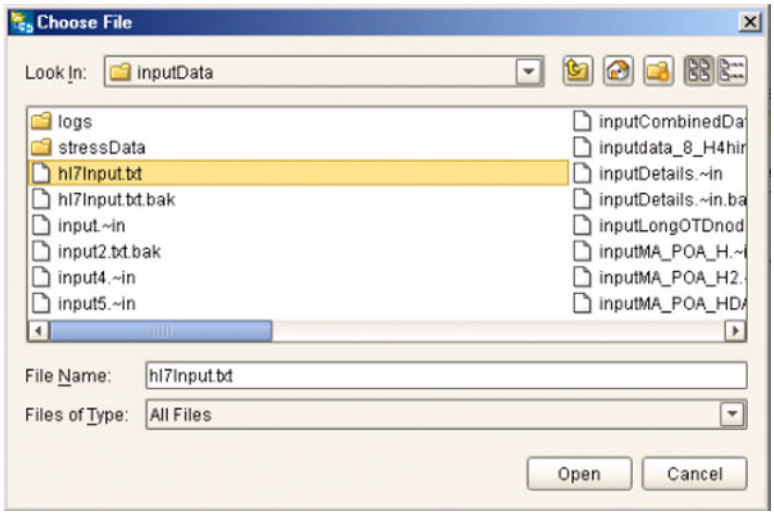
A test panel will appear below the OTD detail area of the editor. Note that there are four data display modes, selectable by tabs (see Figure 91). The Input tab is selected by default.

Figure 91 Test Panel Data Display



3 You can provide the input test data either by selecting a data file (see Figure 92), or by entering the data manually.

Figure 92 Select Data File



4 Click the **Run Tester** icon (green arrow) to test the selected OTD.

5 Verify the output by checking the values for each element for correctness (see Figure 93).

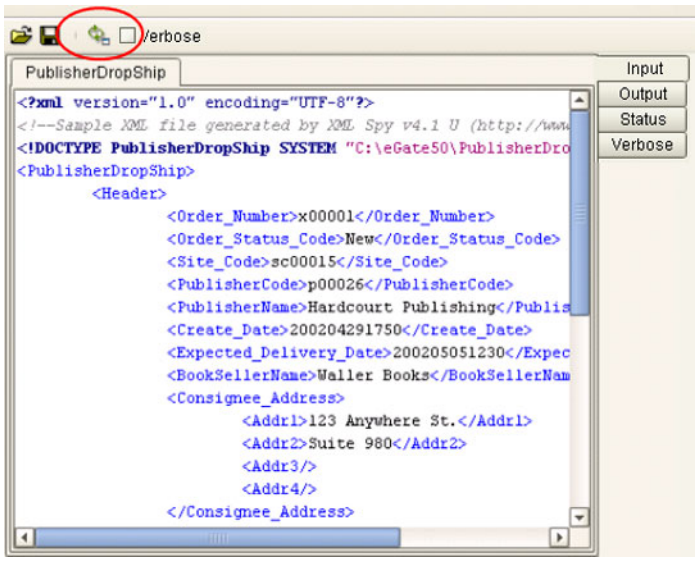
Figure 93 Object Elements and Values

Name	Value
PublisherDropShip	
header	
name	""
order_Number	"x00001"
order_Status_Code	"New"
site_Code	"sc00015"
publisherCode	"p00026"
publisherName	"Hardcourt Publi...
create_Date	"200204291750"
expected_Delivery_	"200205051230"
bookSellerName	"Waller Books"
consignee_Address	
bom_type	""
gl_entity	"GLN"
terms	
line_Item	
length	1
[0]	
value	"500"
counter	"0"
itemCode	"ISBN000139298"
itemDescription	"King James Bib...
qty	"100"
cost	"5.00"

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

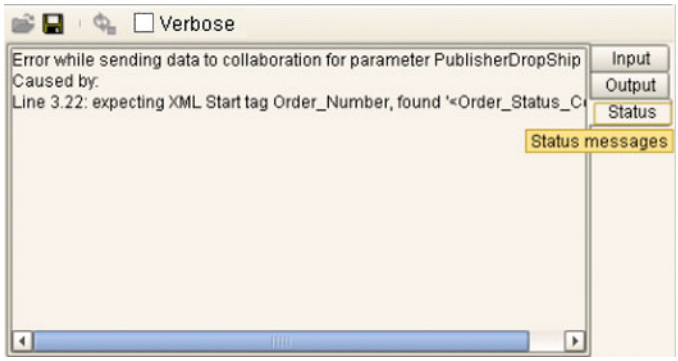
7 You can also change your test data in the Input data display, then re-test the OTD by clicking the **Refresh** icon (see Figure 94) to repopulate your OTD object elements with the new values.

Figure 94 Data Display: Refresh Icon



8 If there are errors in your input data, the **Status** data display is automatically invoked, showing the appropriate error messages (see Figure 95).

Figure 95 Status Data Display



Chapter 8

Environments

Projects are run within Logical Hosts, which are individual, runtime instances of eGate Integrator. Logical Hosts are defined within *Environments*, which represent the physical resources required to implement the Project. This chapter describes the process of defining eGate Environments, and the various components of an Environment.

8.1 Overview

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

8.2 Environment Explorer

Figure 96 Enterprise Explorer: Environment Explorer View

8.2.1 Environment Explorer Icons

The icons described in Table 21 appear in the Environment Explorer.

Table 21 Environment Icons

Icon	Function
	Represents the **Repository**, which is the database where all Projects and contents are saved.
	Represents the **Environment**, which contains Logical Hosts and information about external systems.
	Represents a **Logical Host**, which contains the software and other installed components that are required at runtime.
	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 an **Integration Server**.
	Represents a **JMS IQ Manager** or third-party **message server**, which is used to store and forward eGate system messages.

8.2.2 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 123
- **Environment Menu** on page 124
- **Logical Host Menu** on page 125

Repository Menu

Figure 97 Repository Menu

Table 22 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 in which you can modify the SNMP agent properties.
Save Changes to Repository	Saves all changes made in the Environment Explorer.
Refresh All from Repository	Updates Enterprise Designer with Project/Environment configurations stored in the Repository. (Open editors are not refreshed.)

Environment Menu

Figure 98 Environment Menu

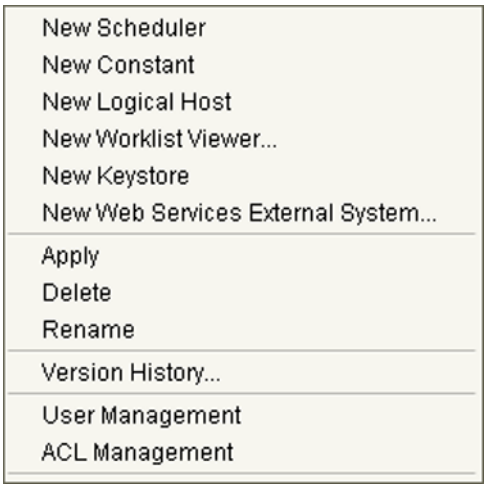


Table 23 Environment Menu Options

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 Environment. See **Defining Environmental Constants** on page 127.
New Logical Host	Adds a new Logical Host to the selected Environment.
New Worklist Viewer	See the *eInsight Enterprise Service Bus User's Guide* for information.
New Keystore	Adds a new keystore to the selected Environment.
New Web Service ...	Adds a third-party Web service application to the Project Explorer. See **SeeBeyond Web Services** on page 80.
Apply	Saves changes to the selected Environment.
Delete	Displays a dialog box in which you confirm that you want to delete the selected Environment. Clicking **Yes** then deletes the Environment.
Rename	Allows you to rename the selected Environment.
Version History	Displays a dialog box with which you can track the version history for Environments. Version control allows users to maintain multiple versions of the same Environment. See **Viewing a Component's Version History** on page 60 for more information.
User Management	Displays a dialog box with which you can manage Repository access. See the *eGate Integrator System Administration Guide*.
ACL Management	Displays the **ACL Properties** dialog box, with which you can assign read and/or write privileges to users for the selected Project. See the *eGate Integrator System Administration Guide*.

Logical Host Menu

Figure 99 Logical Host Menu

Table 24 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.
Delete	Displays a dialog box in which you confirm that you want to delete the selected Logical Host. Clicking **Yes** then deletes the Logical Host.
Rename	Allows you to rename the selected Logical Host.
Apply	Saves changes to the selected Logical Host.
ESR Setup	Displays a dialog box with which you can select emergency software releases (ESRs) to add to the Environment.
Version History	Displays a dialog box with which you can track the version history for Logical Hosts. Version control allows users to maintain multiple versions of the same Logical Host. See **Viewing a Component's Version History** on page 60 for more information.
Check In	Displays a dialog box, with which you can check in the current version of an Environment. Refer to **Checking a Component In** on page 58 for more details.
Check Out	Displays a dialog box with which you can check out a version of an Environment. See **Checking a Component Out** on page 59 for more information.
ACL Management	Displays the **ACL Properties** dialog box, with which you can assign read and/or write privileges to users for the selected Environment See the *eGate Integrator System Administration Guide*.

Table 24 Logical Host Menu Options

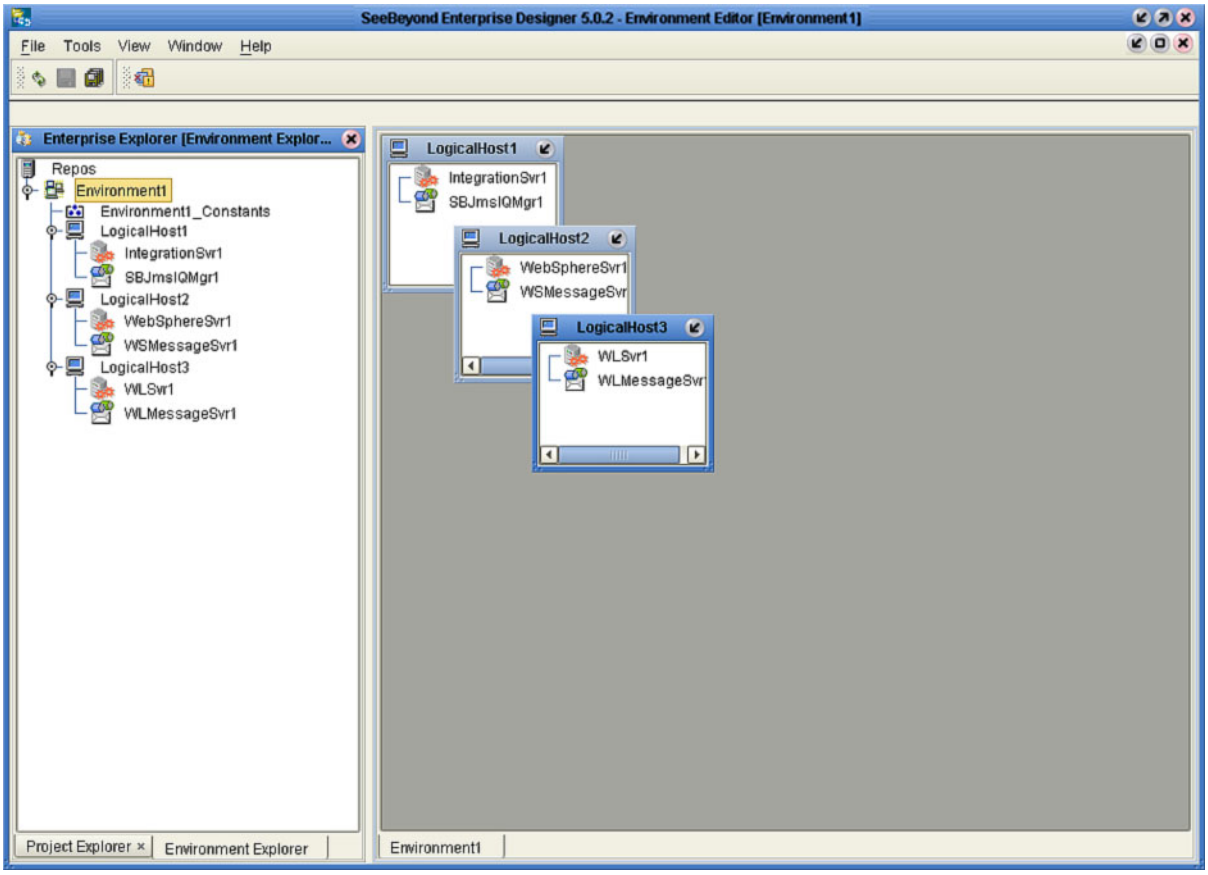
Option	Function
Properties	Displays a dialog box with which you can modify the default settings for the selected Logical Host.

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

8.3 Environment Editor

Clicking on 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 100).

Figure 100 Environment Editor



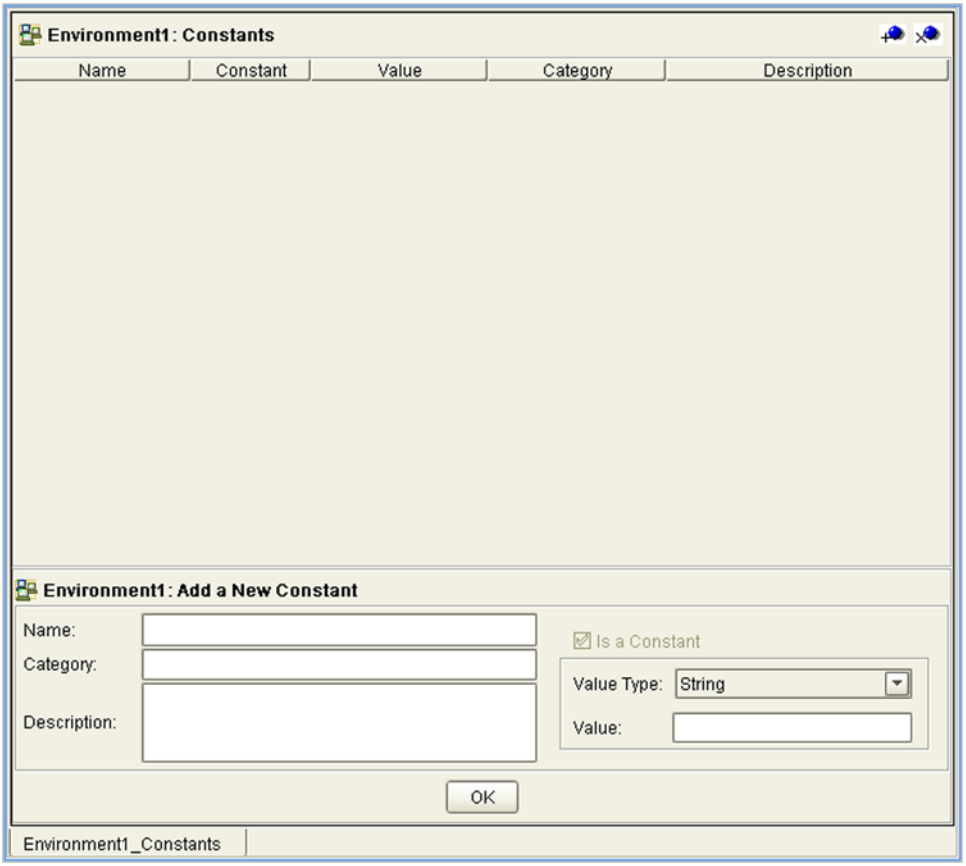
Here you can see the various components (Logical Hosts, servers, and external systems) included in the selected Environment. New Environments are added through the use of the Repository context menu (see **Repository Menu** on page 123). Components are added to the Environment by selecting options in the Environment and Logical Host context menus (see **Environment Menu** on page 124 and **Logical Host Menu** on page 125, respectively).

8.3.1 Defining Environmental Constants

You can define constants for a specific Environment. Environmental constants are name/value pairs that are visible across the Environment. When you create a constant you assign a permanent value to it, which cannot be overridden.

Selecting the **New Constant** option from the Environment context menu displays the Constants panel in the Environment Editor (see Figure 101).

Figure 101 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.

Table 25 Environmental Constants Panel Icons

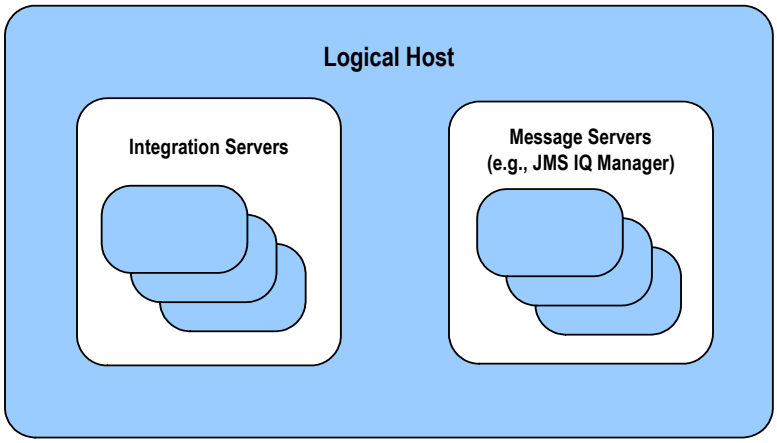
Icon	Name	Function
	Add a New Constant	Adds a new constant to the list.
	Delete a Highlighted Constant	Deletes the selected constant from the list.

8.4 Logical Hosts

8.4.1 Overview

A Logical Host is an instance of the eGate runtime environment that is 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. The Logical Host contains both Integration Servers and Message Servers, as illustrated in Figure 102.

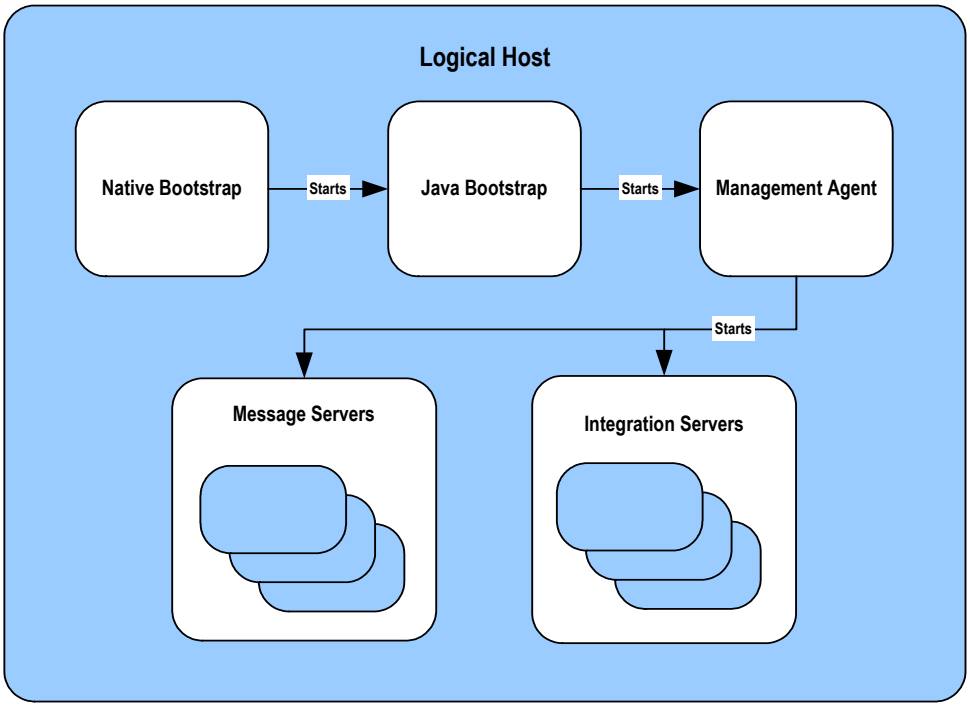
Figure 102 Logical Hosts



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 JMS IQ Managers and Integration Servers.

At run time, a platform-specific bootstrap script starts the Java bootstrap program that 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 103 illustrates this sequence.

Figure 103 Startup Sequence



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

8.4.2 Installing the Logical Host as a Windows Service

Installing the Logical Host as a Windows service configures the Logical Host to start up automatically at system startup, and restart automatically after an abnormal system shutdown.

Note: *You must have Administrator privileges on the local Windows computer in order to configure the Logical Host to start as a service (the installation script writes to the Windows registry, which requires Administrator privileges).*

To install the Logical Host as a Windows Service

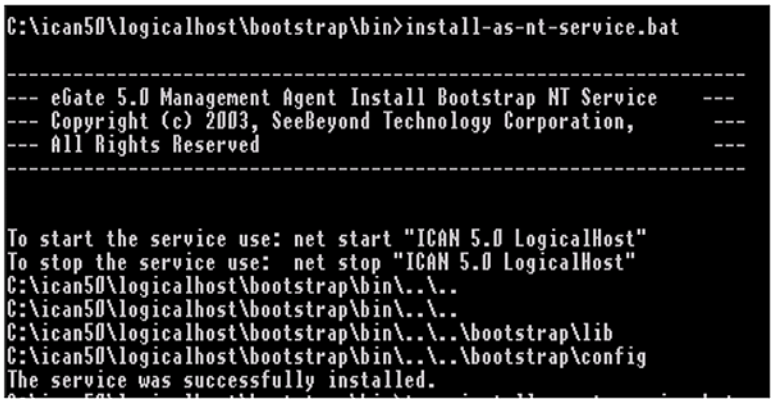
1. Configure the **logical-host.properties** file as described in **Modifying the Logical Host Properties File** on page 135.
2. Change to the directory containing **bootstrap.bat**:

   ```
   cd ICAN-root\logicalhost\bootstrap\bin\
   ```

3. Double-click **install-as-nt-service.bat**.

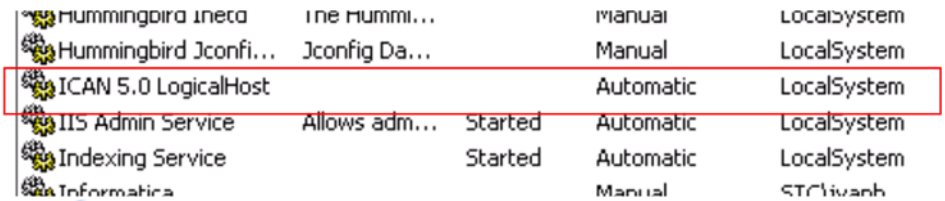
   This runs the script that configures the Windows service to automatically run the Logical Host at system startup. The only parameter is the name of your Logical Host Service; the default is "ICAN 5.0 Logical Host" (see Figure 104).

Figure 104 Install as Service Script



4. Verify the installation by opening the Windows Services facility and searching for the Logical Host name (see Figure 105). Note that by default, the service is listed as *Automatic*—it will not be running, however, until you click **Start** or reboot the computer.

Figure 105 Windows Services List



To remove the Logical Host Windows Service

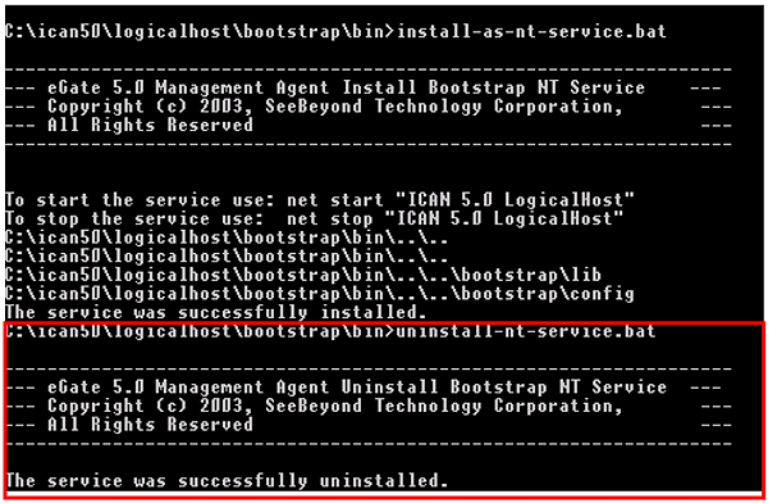
1 Change to the directory containing **bootstrap.bat**:

```
cd ICAN-root\logicalhost\bootstrap\bin\
```

2 Double-click **uninstall-nt-service.bat**.

This runs the script that removes the Windows service that automatically starts the Logical Host at system startup (see Figure 106).

Figure 106 Uninstall as Service Script



8.4.3 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 107).

Figure 107 Logical Host Configuration Properties

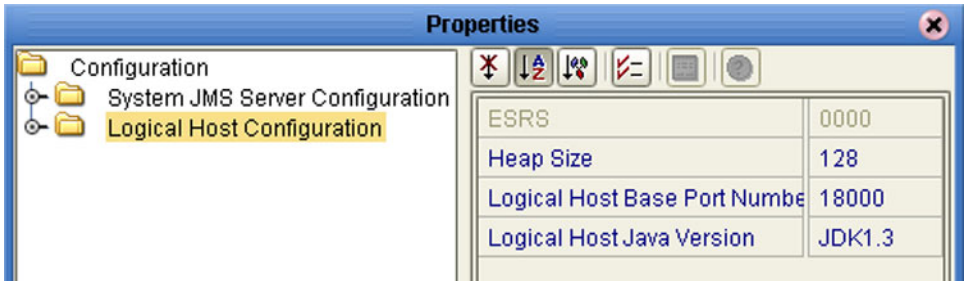


Table 26 Logical Host Configuration Properties List

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 reside in the same Environment, 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*).

8.4.4 Modifying the Logical Host Properties File

If you want to install a Logical Host as a Windows service, you must set the Logical Host default startup configuration in the **logical-host.propterties** file. Setting the default configuration also avoids having to specify the arguments if you choose to start the Logical Host manually at the command prompt. If you do specify the arguments when you manually start the Logical Host, what you enter will override the default values; see **Starting a Logical Host** on page 138.

To modify the Logical Host Properties file

1. Ensure that the Logical Host is not running.
2. Navigate to ***ICAN-root***\logicalhost\bootstrap\config\logical-host.properties.
3. Use a text editor (such as Windows WordPad) to open the file.

Figure 108 Example logical-host.properties File

```
################################################################################
#                                                                              #
#                          Logical Host Properties                             #
#                                                                              #
################################################################################

# These properties are automatically persisted by the bootstrap sequence.
# They are used by default if none are provided at the command line.

################################################################################
# repository.url: (USER MODIFIABLE)
#               Specifies the remote URL for connecting to the repository.
#               Takes the form:
#                   http://<repository-server-hostname>:<port>/
#                    <repository-name>
#               For example:
#                   http://localhost:10000/myRep
################################################################################
repository.url=

################################################################################
# repository.username: (USER MODIFIABLE)
#               Username for connecting to the repository.
################################################################################
repository.username=

################################################################################
# repository.password: (USER MODIFIABLE)
#               Plain text form of password used for connecting to the
#               repository. Any value provided here will be cleared out
#               by the system and written in encrypted form to the
#               repository.password.encrypted field.
################################################################################
repository.password=

################################################################################
# repository.password.encrypted:
#               Encrypted form of the repository password. NOTE: This value
#               is generated by the system, so it is improper to edit this
#               field manually.
################################################################################
repository.password.encrypted=

################################################################################
# physical.host.name: (USER CONFIGURABLE)
#               Specifies the physical host on which this logical host is
#               running. The host name should include the domain name.
#               Example: host.company.com
################################################################################
physical.host.name=

################################################################################
# logical.host.environment.name: (USER MODIFIABLE)
#               Specifies the name of the environment containing the
#               current logical host.
################################################################################
logical.host.environment.name=

################################################################################
# logical.host.name: (USER MODIFIABLE)
#               Specifies the name of the current logical host.
################################################################################
logical.host.name=
```

```
################################################################################
# logical.host.root.dir:
#               Specifies the root directory of a logical host
#               installation.
################################################################################
logical.host.root.dir=

################################################################################
# os.type:
#               Specifies the OS type of the machine on which logical host
#               is going to run
################################################################################
os.type=


# *** THE FOLLOWING PROPERTIES ARE PRIVATE; MODIFY AT YOUR OWN RISK ***

# bootstrap
managementagent.jar.path=lib/managementagent.jar

# deployment manager
deployment.manager.jar.relative.path=deploymentmanager.jar

# management agent
managementagent.config.file=./config/ManagementAgent-config.xml
managementagent.command.line.windows=cmd.exe /c ..\\jre\\bin\\java.exe -Xrs -
Dlogical.host.properties.file=..\\bootstrap\\config\\logical-host.properties -classpath
.\\config;.\\lib\\managementagent.jar;.\\lib\\com.stc.hnsems.jar;.\\lib\\jta.jar;.\\lib
\\jms.jar;.\\lib\\jta.jar;.\\lib\\concurrent.jar;.\\lib\\log4j.jar;.\\lib\\jargs.jar;.\
\lib\\mx4j.jar;.\\lib\\mx4j-
tools.jar;.\\lib\\stcjms.jar;.\\lib\\stcrepository.jar;.\\lib\\stccore.jar;.\\lib\\stcr
epositoryclient.jar;.\\lib\\stcrepositoryserver.jar;.\\lib\\velocity.jar;.\\lib\\ejb.ja
r;.\\lib\\commons-
collections.jar;..\\stcis\\lib\\com.stc.isbootstrap.jar;..\\stcis\\lib\\gnu-regexp-
1.1.4.jar;.\\lib\\com.stc.isapi.jar;..\\stcis\\lib\\com.stc.isimpl.jar;.\\lib\\com.stc.
compatimpl.jar;.\\lib\\com.stc.eventmanagementapi.jar;.\\lib\\xmlrpc.jar;.\\lib\\smscli
ent.jar;.\\lib\\com.stc.stcutil4Jimpl.jar;.\\lib\\xercesImpl.jar;.\\lib\\xalan.jar;.\\l
ib\\xml-apis.jar;.\\lib\\stcqueueviewer.jar;.\\lib\\stcjournaller.jar;.\\lib\\jakarta-
regexp-1.2.jar;;.\\lib\\commons-codec-1.1.1-dev.jar;
com.stc.processmanager.ma.ManagementAgent

managementagent.command.line.unix=sh ../bootstrap/bin/magent.sh

managementagent.shutdown.windows=cmd.exe /c ..\\jre\\bin\\java.exe -
Djava.ext.dirs=.\\lib com.stc.sms.mbeans.ManagementAgentControl -f
.\\config\\ManagementAgent-config.xml -c shutdown

managementagent.shutdown.unix=sh ./jre/bin/java -Djava.ext.dirs=./lib
com.stc.sms.mbeans.ManagementAgentControl -f ./config/ManagementAgent-config.xml -c
shutdown


# misc
repository.relative.url=data/repository/SBYN00.properties
log4j.config.path=bootstrap/config/log4j.xml
jre.zip.name=jre.zip
```

4 In the beginning of the **logical-host.properties** file, locate the properties listed in **Table 27 on page 137** (most are marked USER MODIFIABLE, and may appear in a different order than that shown):

5 Enter the appropriate values for your system.

Note: *Do not enter spaces before or after the equal (=) sign and the property values. Spaces are allowed only in the value itself.*

Table 27 User-Modifiable Logical Host Properties

Property	Description
logical.host.environment.name	Specifies the name of the Logical Host environment deployment; for example, **EnvironmentDeployment1**.
logical.host.name	Specifies the name for the Logical Host; for example, **logicalhost1**.
logical.host.root.dir	Specifies the full path of the Logical Host directory; for example, **c:\ican50\logicalhost**.
os.type	Specifies the operating system type under which logical host is going to run.
physical.host.name	Specifies the physical host on which this logical host is running. The host name should include the domain name, for example: **host.company.com**.
repository.url	Specifies the path to your Repository; for example, **http://*hostname*:*port*/*repositoryname*** where: ▪ ***hostname*** is the physical name of the computer on which the Repository resides; for example, **localhost**. ▪ ***port*** is the port number that the Repository uses to receive requests; for example, **9999**. ▪ ***repositoryname*** is the name you specified for the Repository; for example, **Repository1**.
repository.username	Specifies the user name you are using to access the Repository; for example, **Administrator**.
repository.password	Specifies the password you are using to access the Repository; for example, **myPassword**. When you launch the bootstrap process, this password is encrypted and written to the **repository.password.encrypted** property. After the encrypted password has been written, this **repository.password** value is removed from the **logical-host.properties** file.
repository.password.encrypted	This property is automatically updated based on changes made to the **repository.password** property. *Do not enter anything for this property or modify its contents*.

6 Save the **logical-host.properties** file.

7 The Logical Host may now be started, either manually or as a Windows service.

8.4.5 Starting a Logical Host

The Logical Host is started by means of a bootstrap script that is specific to the platform on which it is running. You can start the Logical Host manually by opening a command prompt and typing a startup command having a string of arguments appended to it. These include both optional and required initial (first-time only) arguments, as listed in Table 28. If you have configured the logical-host.properties file (see **Modifying the Logical Host Properties File** on page 135), you can use the default values by omitting the arguments.

The startup command string uses the following syntax:

```
bootstrap argument1 ... argumentN
```

For example, the command

```
bootstrap -h
```

displays the online *Help* entry that explains the command line arguments.

Table 28 Startup Command Arguments

Parameter	Description	Req/Opt
-d *debug*	Overrides bootstrap sequence. Displays all cached (default) argument values.	Optional
-e *environment name*	The name of the Environment to which this Logical Host belongs.	Required (first time only)
-h *help*	Overrides bootstrap sequence. Displays the usage report.	Optional
-i *id*	The user ID used for accessing the Repository. Note that the user ID is the same as the username, and that the Administrator can set up more than one user ID.	Required (first time only)
-l *logicalhost name*	The name of this Logical Host.	Required (first time only)
-n *physical host name*	The name of this Physical Host.	Required (first time only)
-p *password*	The password used for accessing the Repository.	Required (first time only)
-r *repository URL*	The root URL for the Repository containing the Logical Host data.	Required (first time only)

Note: *Required (first time only) indicates that the argument is required the first time you start the Repository. You do not need to use it again unless you change the name of the Logical Host.*

Starting a Logical Host on a Windows System

To start a Logical Host for the first time, or to override existing default values

1. Open a command prompt on your desktop.
2. Change to the directory containing **bootstrap.bat**:

   ```
   cd ICAN-root\logicalhost\bootstrap\bin\
   ```

3. Type the startup command including all of the required parameters shown in Table 28:

   ```
   bootstrap argument1 ... argumentN
   ```

4. Click **Enter**.

To start a Logical Host after the first time, or to use existing default values

1. Open a command prompt on your desktop.
2. Change to the directory containing **bootstrap.bat**:

   ```
   cd ICAN-root\logicalhost\bootstrap\bin
   ```

3. Double-click **bootstrap.bat**.

Starting a Logical Host on a UNIX System

To start a Logical Host

1. Open a command prompt on your desktop.
2. Change to the directory containing **bootstrap.bat**:

   ```
   cd ICAN-root/logicalhost/bootstrap/bin/
   ```

3. Run the bootstrap script using the following command:

   ```
   ./bootstrap.sh arguments
   ```

Starting a Logical Host on a Red Hat Linux System

To start a Logical Host

1. Open a command prompt on your desktop.
2. Change to the directory containing **bootstrap.bat**:

   ```
   cd ICAN-root/logicalhost/bootstrap/bin/
   ```

3. Run the bootstrap script using the following command:

   ```
   ./bootstrap.sh arguments
   ```

4. After the bootstrap command is executed, the script prompts you for the RedHat server release number.

8.4.6 Stopping a Logical Host

The Logical Host can be shut down either from the Enterprise Monitor or by means of the following command-line procedure.

To stop a Logical Host

1. Open a command prompt on your desktop.
2. Change to the directory containing **bootstrap.bat**:

```
cd ICAN-root\logicalhost\bootstrap\bin\
```

3. Run the file **shutdown.bat** (or **shutdown.sh**).

8.5 Integration Servers

The Logical Host contains one or more Integration Servers, which are the engines that run eGate Collaborations for processing business logic, and eWays that communicate with external applications. It provides services for security, transactions, business rules execution, and connectivity management. The SeeBeyond Integration Server is based on Java 2 Enterprise Edition (J2EE). eGate Integrator also supports third-party integration servers such as BEA WebLogic and IBM WebSphere (see **Deploying Projects to Third-Party Servers** on page 158).

8.5.1 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 IS 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 109).

Figure 109 Top-level IS Configuration Properties

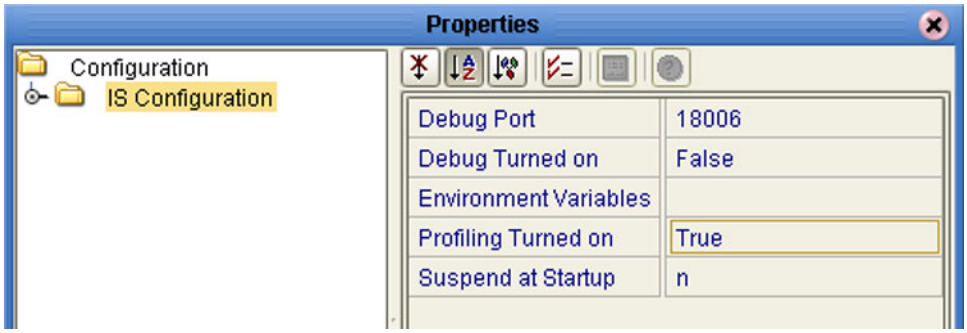


Table 29 Top-level IS Configuration Properties List

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

Table 29 Top-level IS Configuration Properties List

Property	Description
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 143.
Suspend at Startup	Allows the VM to begin executing before the debugger application attaches. The default is **n** (do not suspend).

The IS Configuration node contains several sections, each containing detailed configuration properties for a particular IS component (including the integration server itself). These components are:

- Web Container
- IS Profiling Configuration
- Security Configuration Template
- eInsight Engine Configuration
- Application Manager Configuration Template
- Integration Server Configuration

You can 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 110 Web Container Configuration Properties

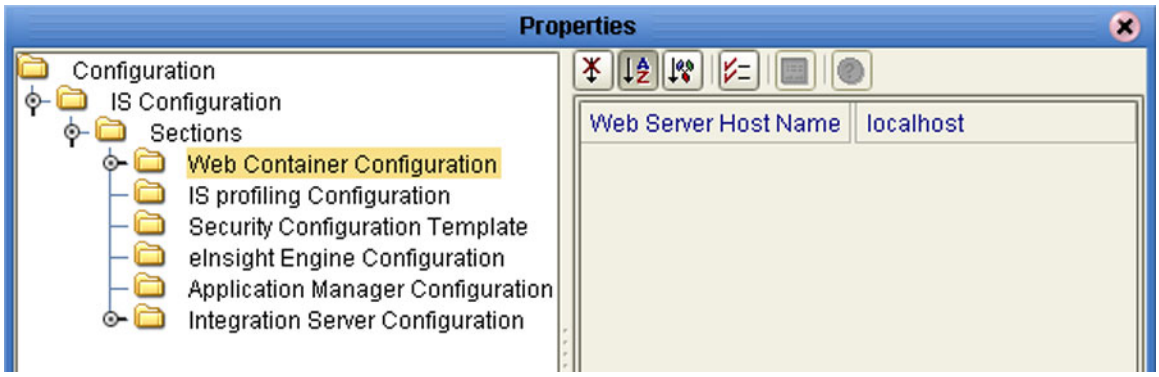


Table 30 Web Container Configuration Properties List

Property	Description
Web Server Host Name	When setting up a Web server, this property specifies the host name; the default is **localhost**.

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 111).

Figure 111 Profiling Configuration Properties

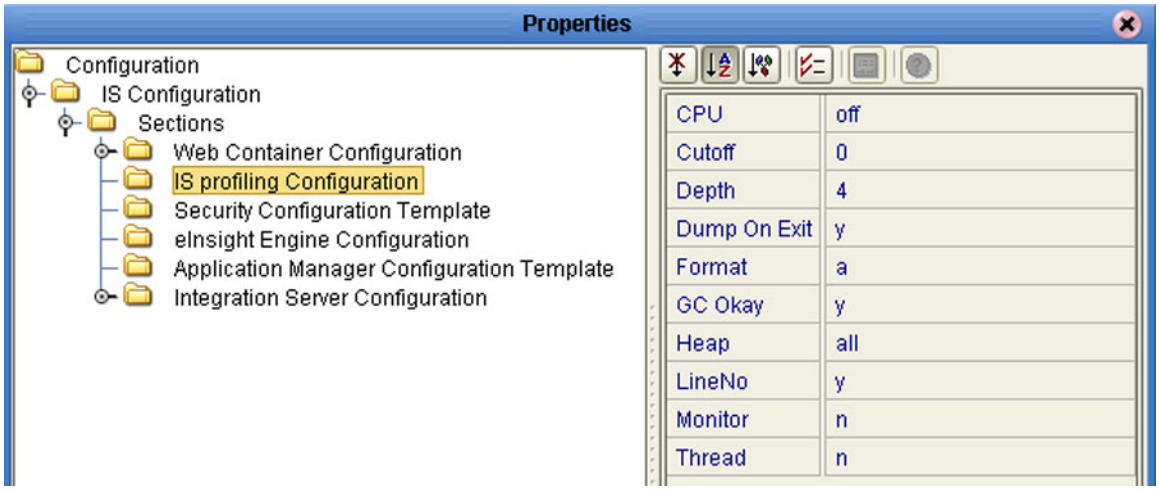


Table 31 Profiling Configuration Properties List

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 **0**.
Depth	Specifies the stack trace depth. The default value is **4**.
Dump on Exit	Specifies whether or not to dump on exit.
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

These properties pertain to the Lightweight Directory Access Protocol (LDAP), if used. Additional information can be found in the *eGate Integrator System Administration Guide*.

Figure 112 Security Configuration Properties

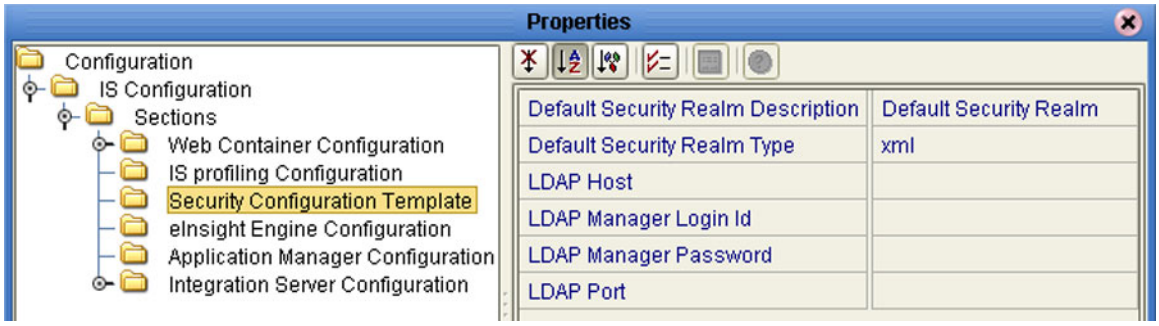


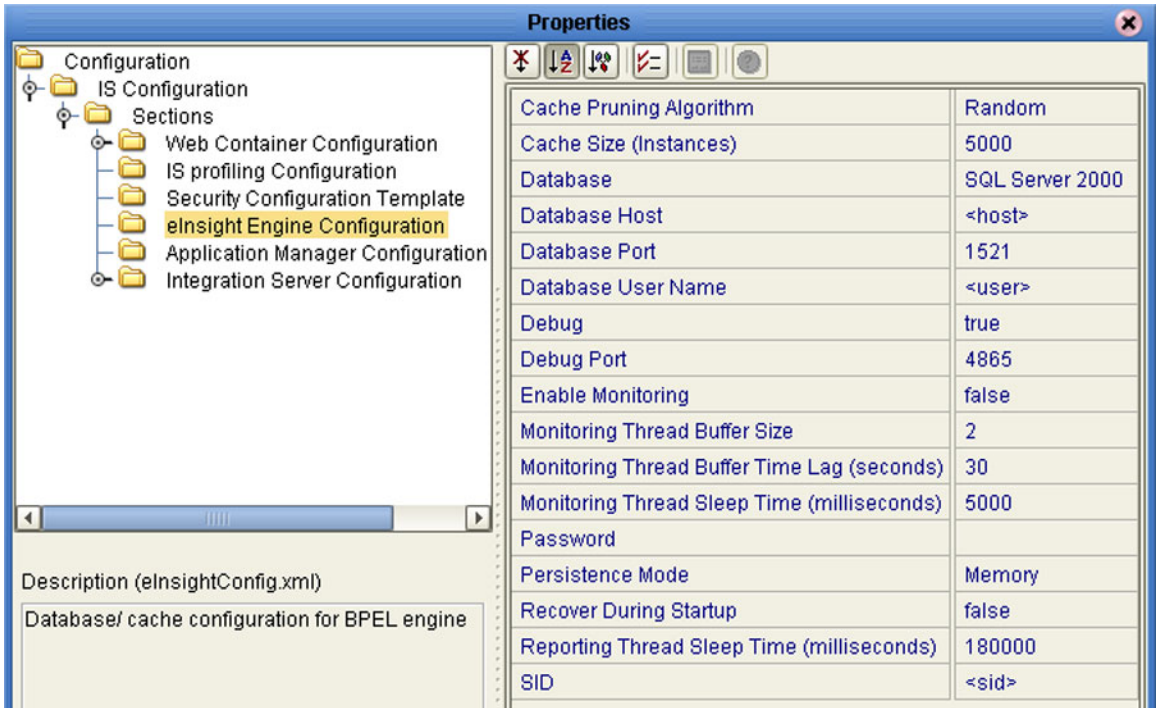
Table 32 Security Configuration Properties List

Property	Description
Default Security Realm Description	Specifies the default LDAP Security Realm description. The default value is **Default Security Realm**.
Default Security Realm Type	Specifies the default LDAP Security Realm type. The default value is **xml**.
LDAP Host	Specifies the LDAP host platform. There is no default value.
LDAP Manager Login Id	Specifies the LDAP manager login ID. There is no default value.
LDAP Manager Password	Specifies the LDAP manager password. There is no default value.
LDAP Port	Specifies the LDAP port. There is no default value.

eInsight Engine

These configuration properties relate to the BPEL engine's database cache; see the *eInsight Enterprise Service Bus User's Guide* for information regarding these properties.

Figure 113 eInsight Engine Configuration Properties



Application Manager

You can set integration server thread pool variables using the Application Manager Configuration Properties dialog box (see Figure 114).

Figure 114 Application Manager Configuration Properties

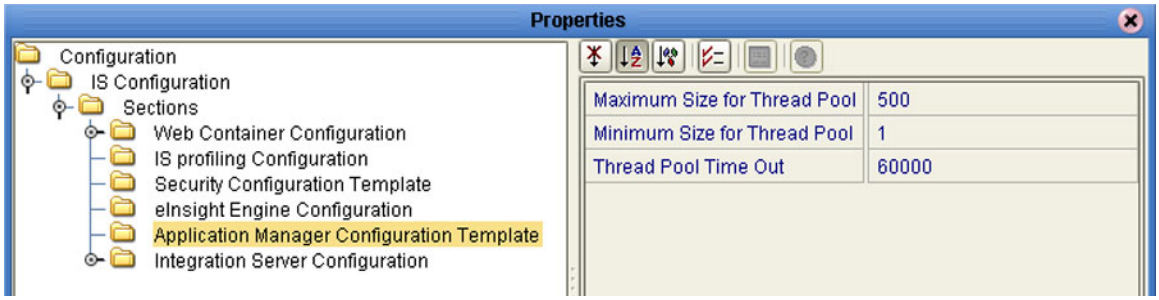


Table 33 Application Manager Configuration Properties List

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 **1**.
Thread Pool Time Out	Specifies the timeout interval for the thread pool, measured in milliseconds. The default value is **60000**.

Integration Server

Detailed, low-level configuration of the integration server is performed using the Integration Server Configuration Properties dialog box (see Figure 115).

Figure 115 Integration Server Configuration Properties

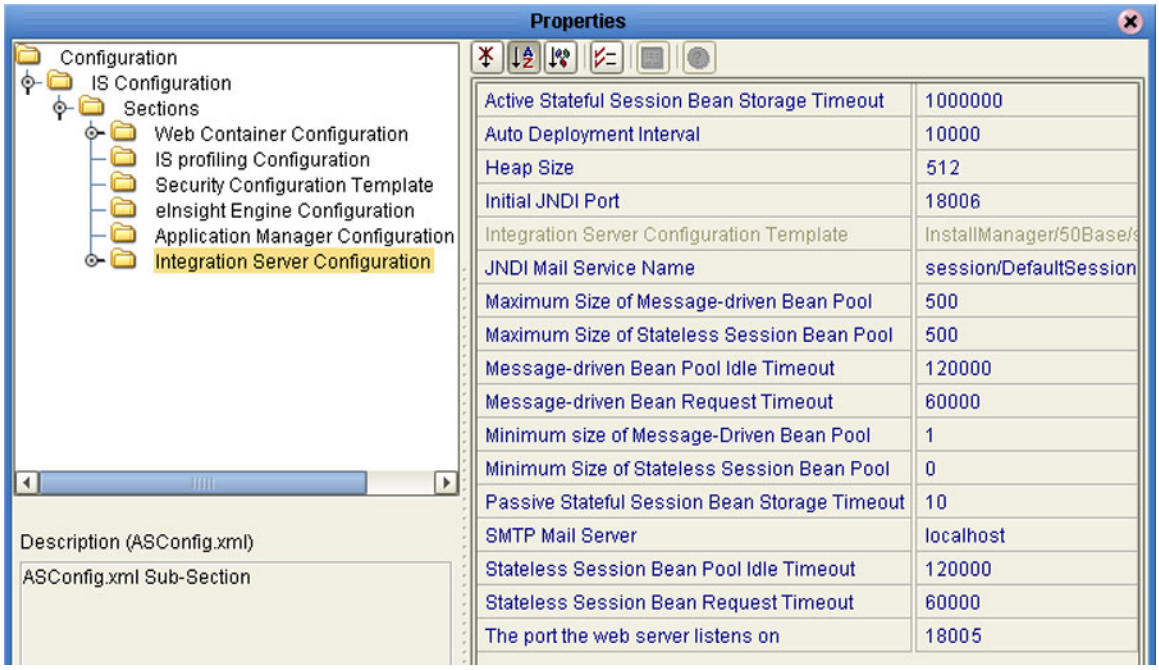


Table 34 Integration Server Configuration Properties List

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 **1000000** 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 **10000** ms.
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.
Integration Server Configuration Template	Specifies the location of the configuration template (not user-modifiable).
JNDI Mail Service Name	Specifies the name of the JNDI mail service. The default value is **session/DefaultSession**.

Table 34 Integration Server Configuration Properties List

Property	Description
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 **120000** ms.
Message-driven Bean Request Timeout	Specifies the interval after which a Message-driven Bean request times out, measured in milliseconds. The default value is **60000** ms.
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.
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 **120000** ms.
Stateless Session Bean Pool Request Timeout	Specifies the interval after which a Stateless Bean request times out, measured in milliseconds. The default value is **60000** ms.
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.

8.6 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.

Third-party integration 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-part message servers, see the *eGate Integrator JMS Reference Guide*.

Chapter 9

Project Deployment

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

9.1 Deployment Profiles

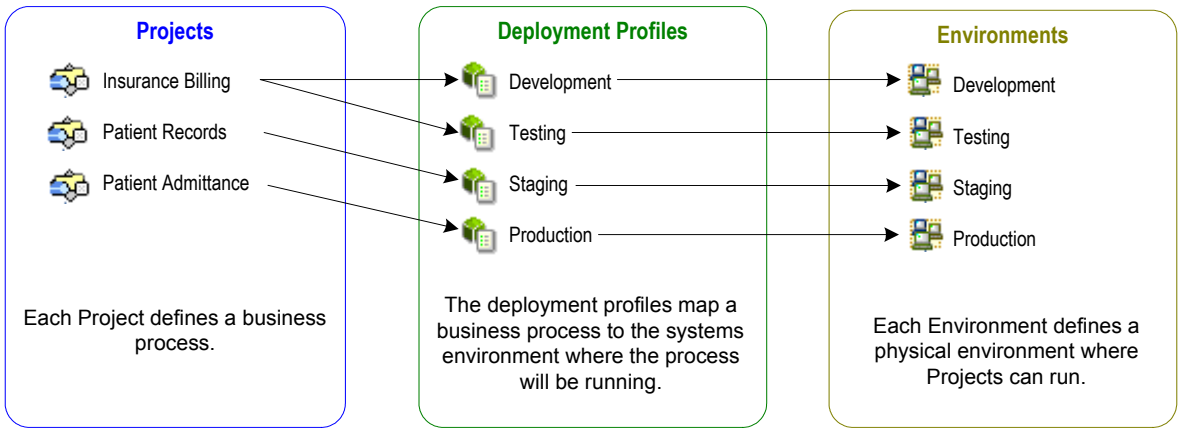
Deployment Profiles define specific instances of a Project in a particular Environment. A deployment profile contains information about the assignment of Services and Message Destinations to integration and message servers (JMS IQ Managers). It also contains version information for all relevant objects 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 3 from the **System Overview** on page 18:

Figure 116 eGate Integrator Implementation Model



9.2 Using the Deployment Editor

The Deployment Editor (see Figure 117) appears when you create a new Deployment Profile or click on an existing Deployment Profile icon in the Project Explorer tab.

Figure 117 Deployment Editor Window

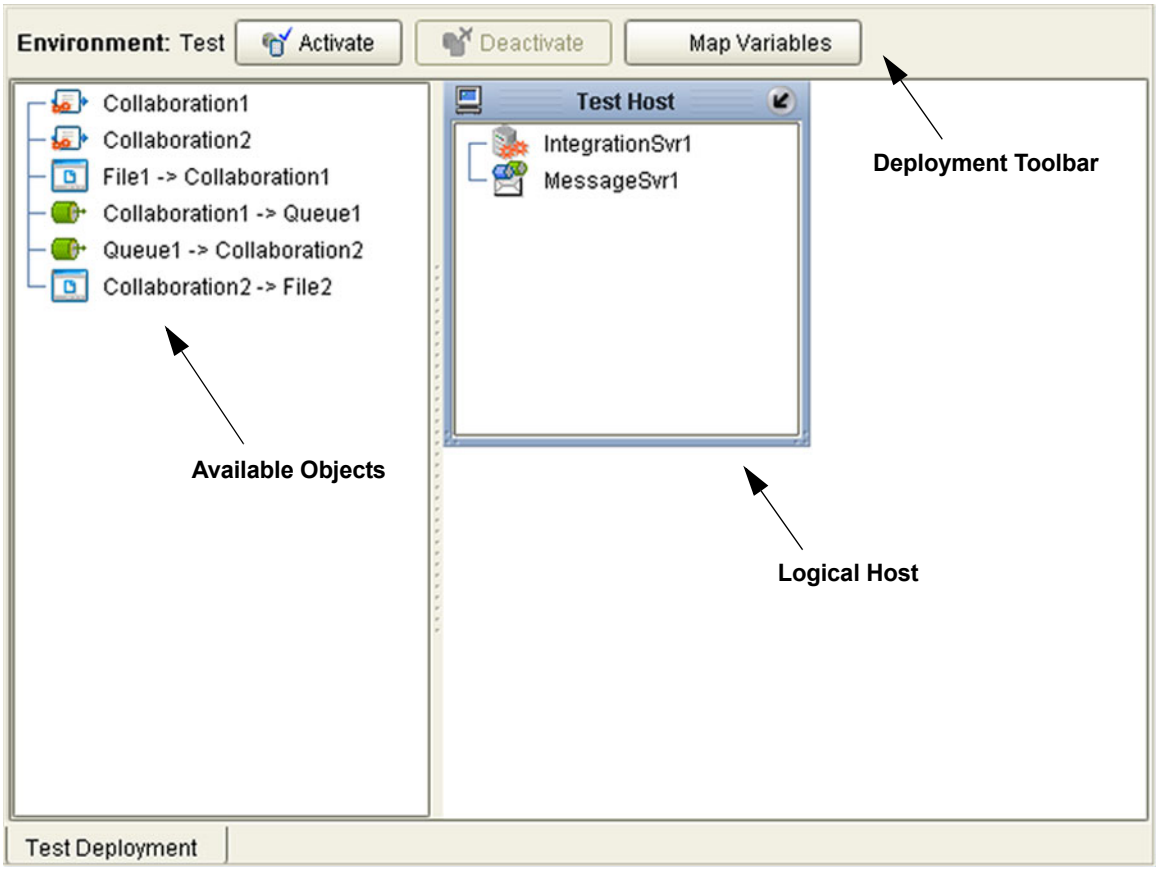


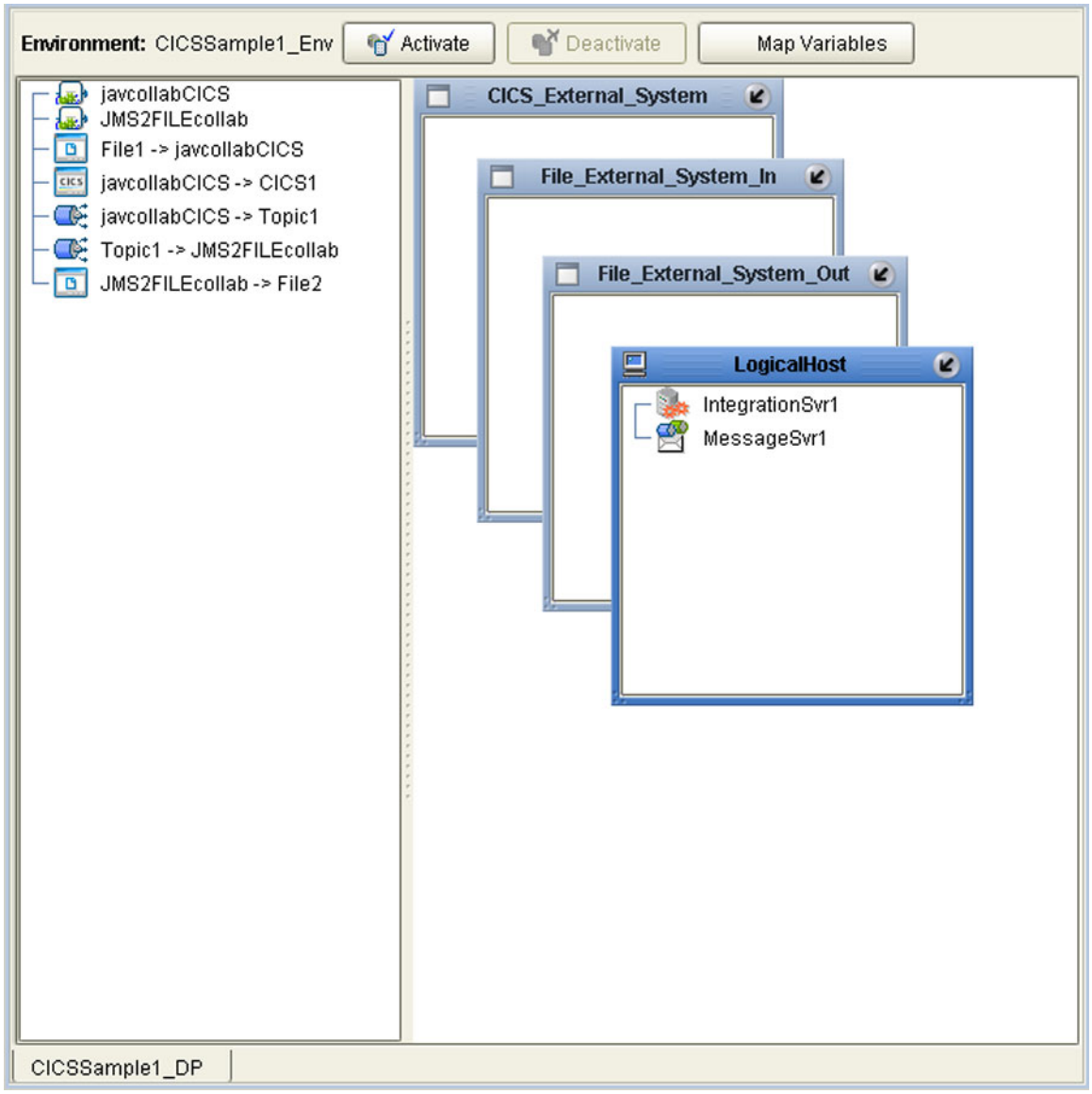
Table 35 Deployment Toolbar Buttons

Button	Function
Activate	Starts the Project by creating an enterprise archive (EAR) file based on the Connectivity Map and linking this file with the SeeBeyond Integration Server. See **Activating and Deactivating Deployment Profiles** on page 155.
Deactivate	Stops the Project by terminating the link between the EAR file and the SeeBeyond Integration 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 157.

9.3 Creating a Deployment Profile

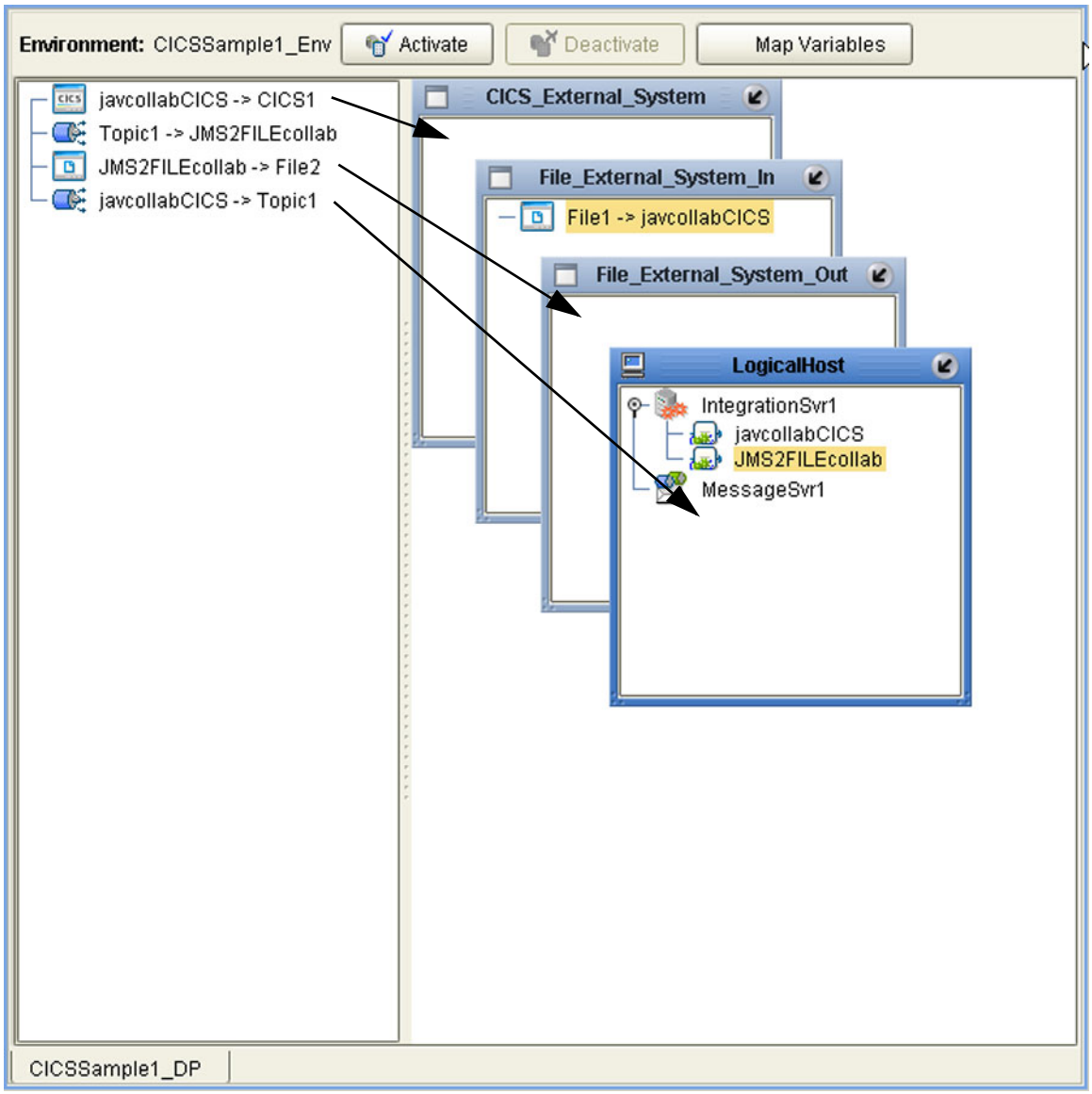
In the Environment Explorer, create an Environment and right-click on the Environment to display its context menu. From the menu, select the Environment components you need and name them appropriately. They will appear on the Deployment Editor canvas, as shown in Figure 118.

Figure 118 Example Deployment Profile (1)



Drag the Project components from the left panel and drop them into the appropriate Environment components in the right panel, as illustrated in Figure 119. As you do so, they will disappear from the left panel.

Figure 119 Example Deployment Profile (2)

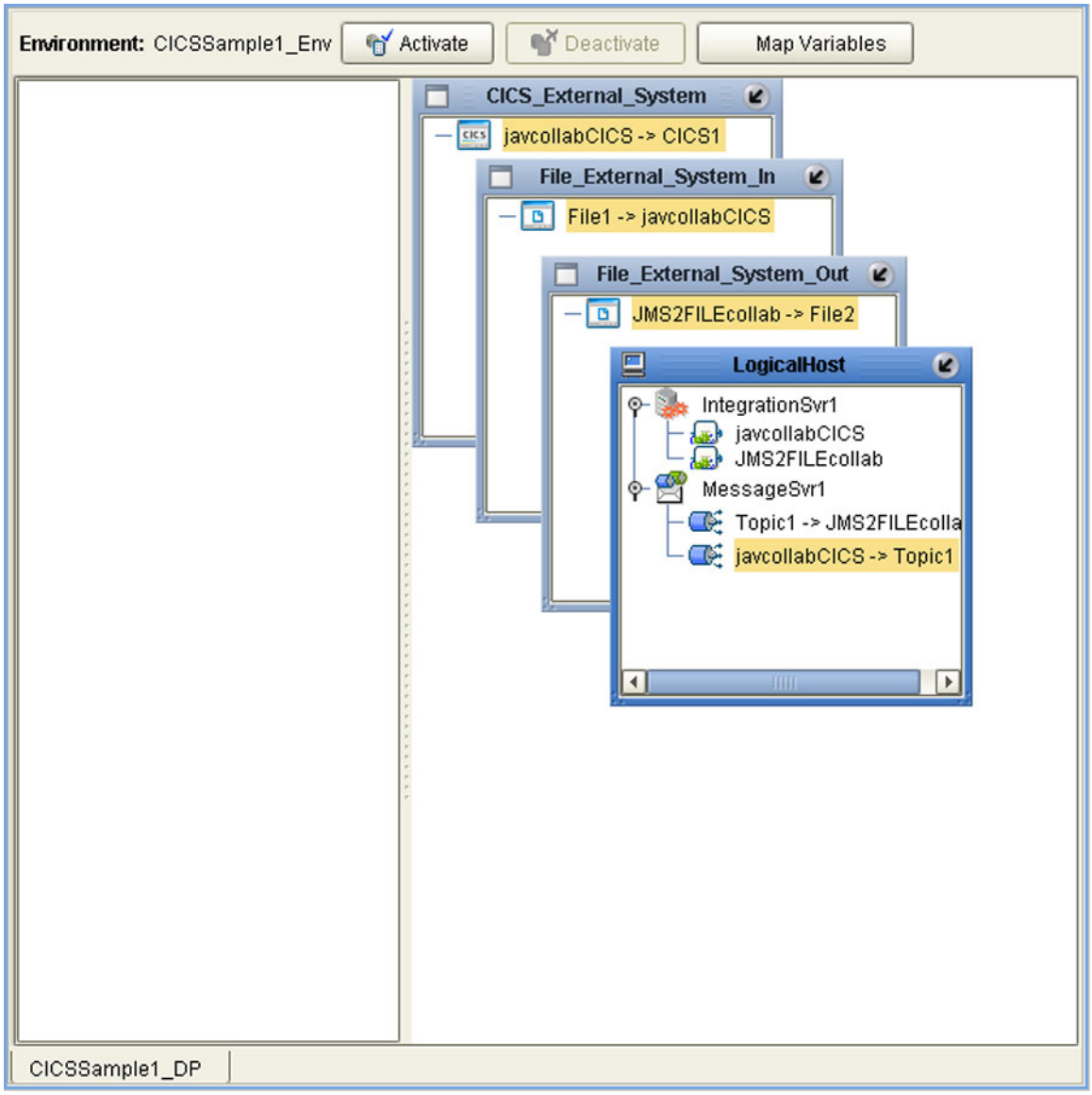


Note that:

- The eWay objects are placed into their appropriate External Systems.
- Collaboration objects are placed into the appropriate Integration Server on the appropriate Logical Host.
- Topic and queue objects are placed into the appropriate Message Server (JMS IQ Manager) on the appropriate Logical Host.

When the Environment components are fully populated, the left panel will be blank, as shown in Figure 120. The Deployment Profile is now ready to be saved and Activated.

Figure 120 Example Deployment Profile (3)



9.4 Activating and Deactivating Deployment Profiles

Using the Activate and Deactivate toolbar buttons, you have the option of immediately applying the changes to the Logical Host or deferring the changes to a later time. Activating the Deployment Profile without applying the changes checks the validity of the entire Deployment Profile.

Another advantage to activating the Deployment Profile without applying the changes comes into play when you have multiple Deployment Profiles to deploy at once. To save time, you can activate each of the Deployment Profiles without applying the changes. Then when you do apply all of the changes to the Logical Host in one batch.

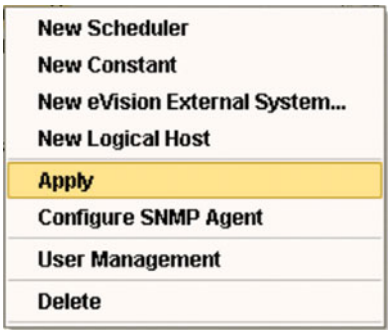
To activate a Deployment Profile

1. In the Deployment Profile, select the Deployment you wish to activate.
2. Click the **Activate** button. The following message appears:

Figure 121 Activate Dialog Box

3. Answer the question following these 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**. To apply the changes later, right-click the Logical Host and select **Apply** from the menu (see Figure 122). This will apply all of the changes for that Logical Host.

Figure 122 Logical Host Context Menu - Apply

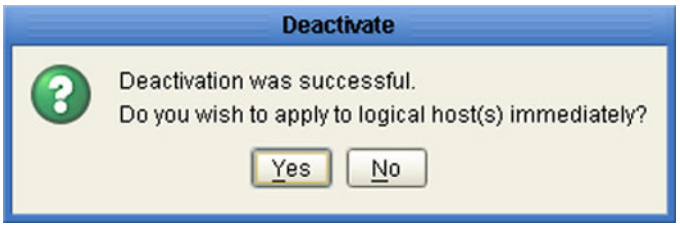


Note: *The **Apply** action assumes that the Logic Host is running, since it invokes a trigger to the Logical Host causing it to download the latest settings from the Repository and deploy those settings to all components on the Logical Host.*

To deactivate a Deployment Profile

1. In the Deployment Profile, select the Deployment you wish to deactivate.
2. Click the **Deactivate** button. The following message appears:

Figure 123 Activate Dialog Box



3. Answer the question following these criteria:
   - If the Logical Host is running, and you wish to apply the changes immediately, click **Yes**.
4. If the Logical Host has not yet been bootstrapped, or you wish to apply the changes at a later time, click **No**. To apply the changes later, right-click the Logical Host and select **Apply** from the menu (see Figure 122). This will apply all of the changes for that Logical Host. See the ***Note*** following the activation procedure.

Note: *In Windows and NFS, application working directories cannot be deleted during deactivation. This is because .jar files s in these directories have been added to a ClassLoader and the JVM maintains **locks/handles?** on any such files. At the subsequent startup of the Integration Server, leftover work directories in the repository/application directory are deleted.*

9.5 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 124) and values (see Figure 125).

Figure 124 Deployment Profile Mappings

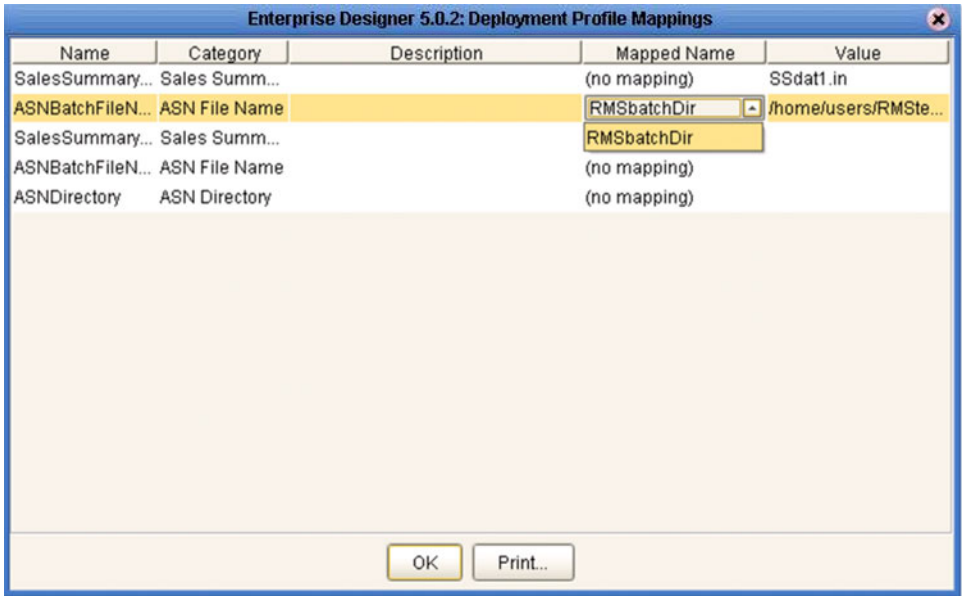
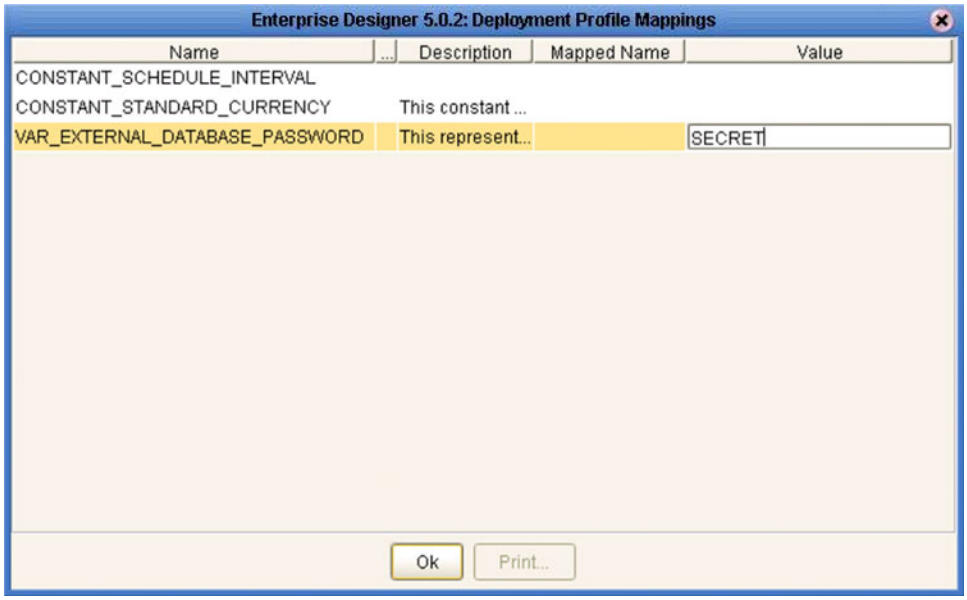


Figure 125 Project Variable Value Entry



9.6 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.

9.6.1 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).

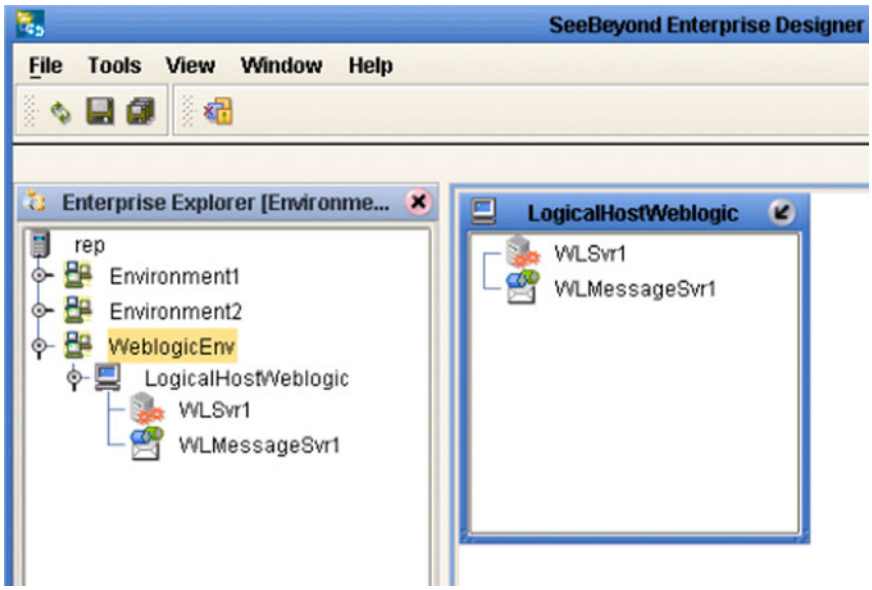
   http://xml.apache.org/dist/xerces-j
3. Place both **.jar** files into the ***weblogic8x*\server\lib** directory.
4. Add the **.jar** files to the *set CLASSPATH* segment of the **startWLS.cmd** file located in the ***weblogic8x*\server\bin** directory. The text to be added is:

```
%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 126):
   - A A new environment
   - B A Logical Host
   - C A WebLogic Integration Server
   - D A WebLogic JMS Message Server

Figure 126 WebLogic Deployment (1)

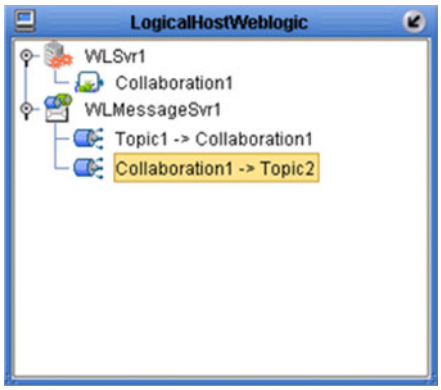


2 Create a new Deployment Profile to bind the Connectivity Map to the new WebLogic environment (see Figure 127).

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

B Drag the Collaboration and drop onto the WebLogic integration server.

Figure 127 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.

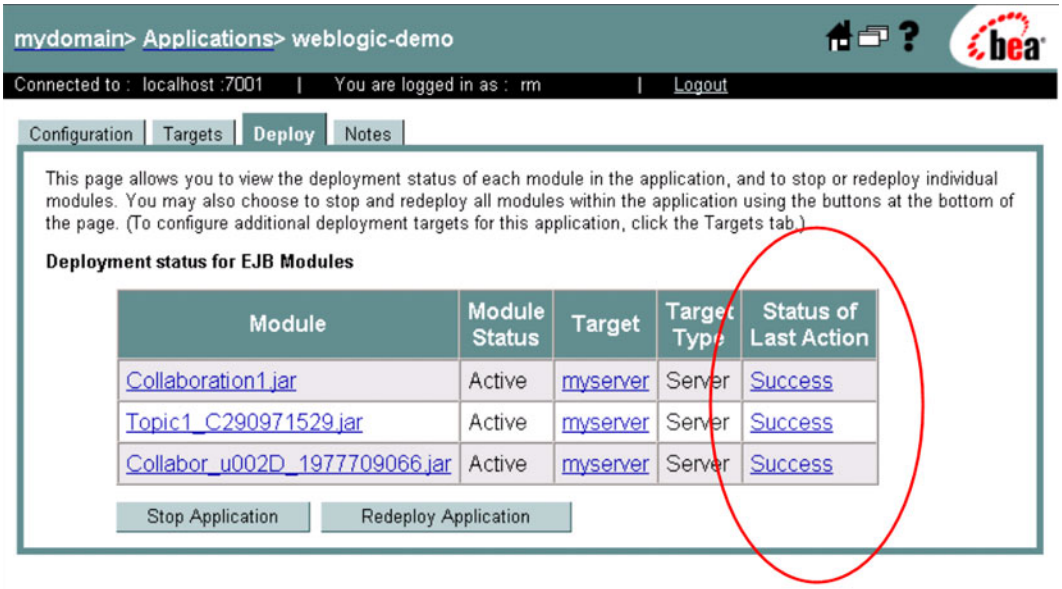
F Select the EAR file you just installed as the archive for the new application.

G Enter a name for the new application.

H Click **Deploy**.

I Verify the success of the deployment (see Figure 128 , which shows a WebLogic 8.1 example).

Figure 128 WebLogic Deployment Verification



9.6.2 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 or 5.0.1 environment

1 Create the following components in Enterprise Designer (see Figure 129):

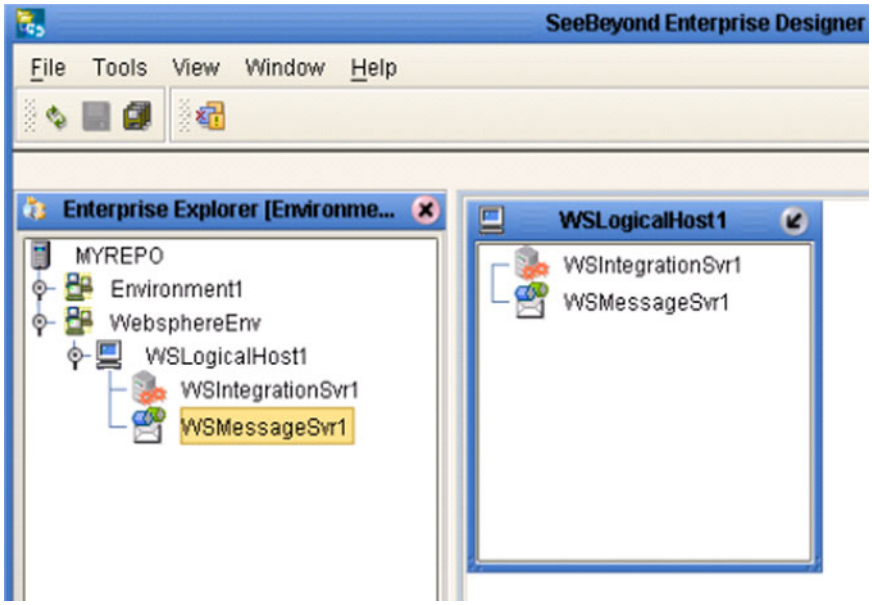
A A new environment

B A Logical Host

C A WebSphere Integration Server

D A WebSphere JMS Message Server

Figure 129 WebSphere Deployment (1)

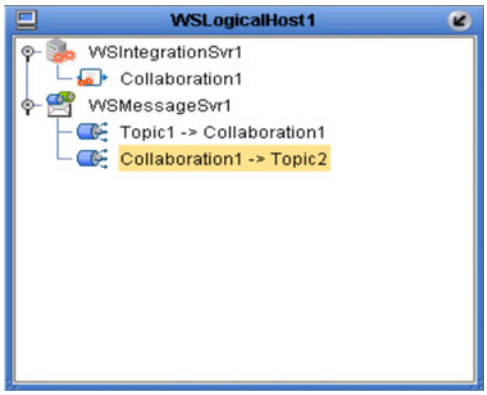


2 Create a new Deployment Profile to bind the Connectivity Map to the new WebSphere environment (see Figure 130).

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

B Drag the Collaboration and drop onto the WebSphere integration server.

Figure 130 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**.

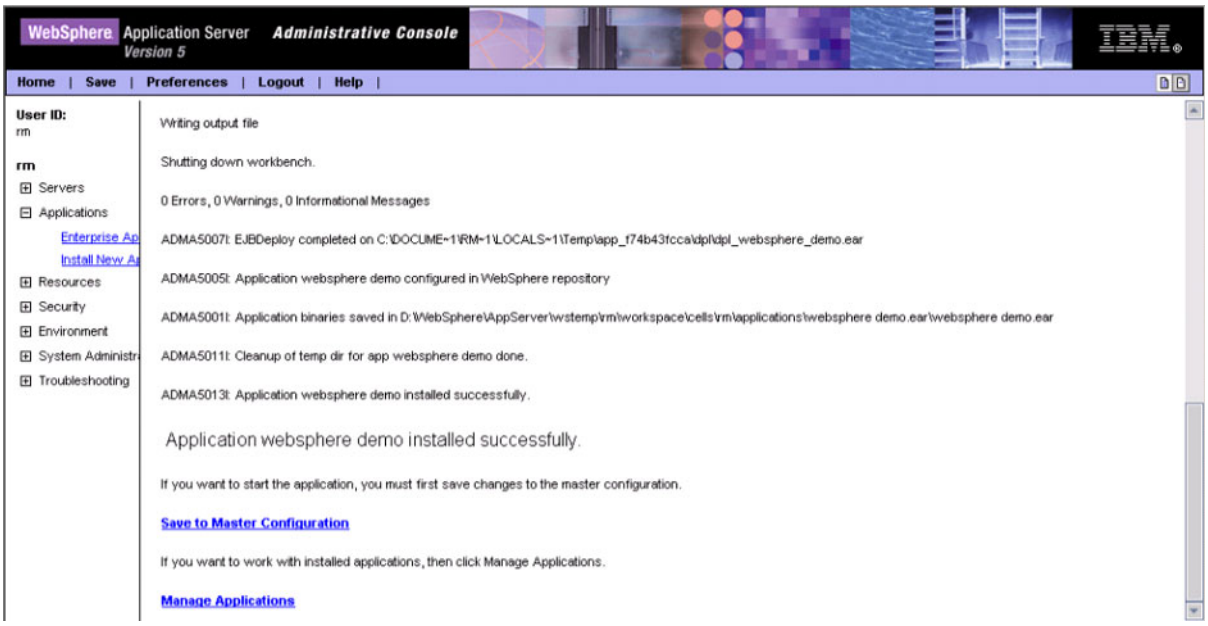
16 In *Step 7, Map modules ...*, check all modules and click **Next**.

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 131 , which shows a WebSphere 5 example).

Figure 131 WebSphere Deployment Verification



Glossary

BI
Business integration (also Business Intelligence).

Collaboration
See **"Service"** and **"Collaboration Definition"**.

Collaboration Definition
The encoding of business rules, in Java or XSLT format. Typically, the encoding consists of operations on OTDs (see **"OTD" on page 167**). 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 Collaborations, Passthrough Collaborations, topics, queues, and eWays. A Connectivity Map is created under a Project. A Project may have multiple Connectivity Maps.

Constants
A name or value pair that is visible across a Project.

CRM
Customer Relations Management

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

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

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
In relational databases (RDBMSs) data mapping is the relationship and data flow between source and target objects. Mapping involves structuring the relationship between source and target objects.

Data Mart
A smaller, focused, database designed to help managers make business decisions. (A data warehouse is a larger, enterprise, database(s).)

Data Mining
Used to synthesize or isolate unique data patterns to predict future behaviors or to filter data to select patterns that help discover previously unknown relationships among data. Commonly used by marketers who acquire and distill consumer information.

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

Data Warehouse
A copy or view of enterprise transaction data (sometimes non-transaction data) that is used for reporting. The data is often summarized and always structured for queries and analysis.

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

Dimension Table
Dimension tables describe the business entities of an enterprise; also called lookup or reference tables.

Dirty Data
Dirty data contains, but is not limited to, incorrect data including spelling errors, punctuation errors, incorrect data referencing, incomplete, inconsistent, outdated, and redundant data.

Drill Down
To move from summary to more detailed data by “drilling down” to get it. In database terminology this might mean starting with a general category and drilling down to a specific field in a record.

eGate System
See **"Project"**.

Environment
A collection of physical resources and their configurations that are used to host Project components. An Environment contains logical hosts and external systems.

EPR
Enterprise Resource Management

ETL
Extract, Transform, Load. Extract is the process of reading data from a source database and extracting 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 data into a larger 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 in an eGate Project of an external application.

External System
A representation in an eGate Project of an external application system.

Extraction
Data are extracted from a source using software tools. This first step in ETL initially "gets" the data.

Fact Table
A fact table typically contains two types of columns: those containing facts and those that contain foreign keys to dimension tables. Fact tables contain detail facts and/or summary facts.

ICAN Suite
The SeeBeyond Integrated Composite Application Network Suite.

Integration Server
J2EE 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.

Join
Matches records, which are joined by a common field, in two tables in a relational database. Often part of a Select query.

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
An instance of the eGate runtime Environment that is installed on a machine. A Logical Host contains the software and other installed components that are required at runtime, such as application and message servers.

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 Maps.

Metadata
"Data about data." Metadata describes "how," "when," and "who" about structure and format, of a particular set of data. ETL tools are used to generate and maintain a central metadata repository.

Non-normalized Data
Non-normalized data cannot be cross-referenced accurately, if at all, and causes manageability issues. Non-normalized data may be converted to normalized data.

Normalized Data
Normalization is a common database design process used to remove redundant or incorrect organization and data. The design and normalization of the database will create a maintainable data set that can be cross-referenced.

Normalized data is not only easier to analyze but also easier to expand. Normalization involves removing redundancy and correcting incorrect data structure and organization.

OLAP
Online analytical processing.

OTD
An acronym for Object Type Definition. OTDs contain the data structure and rules that define an object. An OTD is used in Java Collaboration Definitions 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 query methods:

Choose – With this easy-to-use method, the database system presents a list of parameters from which you can choose. This method is not as flexible as other 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 flexibility and power to make requests for information in the form of a stylized query using a query language. This is the most complex and powerful method.

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, even though 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

Data that has not been turned into "information," through processing. Although factual and "real," raw data is unorganized.

Relational Database (RDBMS)

Short for Relational Database Management System, most often referred to as RDBMS. Data is stored in related tables. Relational databases can be viewed in many different ways.

In this system a single database can be spread across several tables. (RDBMS differs from flat-file databases where each database is self-contained as a single file or table.)

Repository

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

Schema Runtime Environment

An add-on in eGate 5.0 that provides the upgrade path for e*Gate 4.x users to upgrade to eGate 5.0. Also known as the SRE.

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

Staging Data
Data that is to be processed before entering the warehouse.

Subproject
An independent Project that is included as part of another Project and listed on the Enterprise Explorer tree beneath the main Project icon.

Table
Refers to data arranged in rows and columns, like a spreadsheet. In relational database management 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 potentially many subscribers. When the SeeBeyond JMS IQ Manager publishes a message on a topic, it ensures that all subscribers receive the message.

Transformation
Data that are extracted from databases are transformed into a desired form, using various tools that cleanse, merge, purge, aggregate, calculate, audit, remove redundancy, standardize, etc.

XSLT

An acronym for Extensible Stylesheet Language Transformations. A file format used in eGate to generate Collaboration Definitions.

Warehouse
See **"Data Warehouse"**.

e*Gate 4.x Terms in eGate 5.0

Table 36 provides definitions for the terms that are new with eGate release 5.0, as well as equivalent terms from eGate release 4.x.

Table 36 eGate 5.0 Terms

5.0 Term	4.x Equivalent Term
Collaboration	Collaboration
Collaboration Definition	Collaboration Definition
Connection	eWay Connection
Connectivity Map	Closest: Network View of an entire Schema
Deploy	Run the Control Broker
Deployment	<none>
Deployment Profile	Closest: Schema
Enterprise Designer	Enterprise Manager
Enterprise Manager	Enterprise Monitor
Environment	Schema (except only includes physical information, not business logic)
eWay	eWay Connection eWay
eWay Configuration	eWay Connection Configuration
External Application	eWay Connection
External System	eWay Connection
JMS Connection	eWay Connection
Integration Server	<none>
Link	JMS eWay Connection
Linked Message Destination	<none>
Logical Host	Participating Host
Message Destination	Topic or queue
Message Server	JMS IQ Manager
Object Type Definition (OTD)	Event Type Definition (ETD)
Process Manager	Control Broker
Project	Schema (except not including physical layer)
Queue	JMS queue
Repository	Registry
Subproject	Schema

Table 36 eGate 5.0 Terms (Continued)

5.0 Term	4.x Equivalent Term
Topic	JMS topic
XSLT	<none>

Index

F

I

J

L

M

N

O

P

Q

R

S

T

V

W

X