SeeBeyond ICAN Suite

eGate Integrator for eInsight Enterprise Service Bus User's Guide

Release 5.0.3

Version 20040303143240.

Contents

Chapter 5

Chapter 6

Chapter 9

List of Figures

List of Tables

Chapter 1

Introduction

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

1.1 Purpose and Scope

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

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

1.2 Intended Audience

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

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

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

1.3 Organization of Information

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

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

In addition, the **Glossary** on page 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 30). These documents are in Adobe Acrobat format, which requires that Acrobat Reader be installed on your computer. Acrobat Reader can be from Adobe Systems as a free download from the following URL:

http://www.adobe.com

1.6 The SeeBeyond Web Site

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

http://www.seebeyond.com

Chapter 2

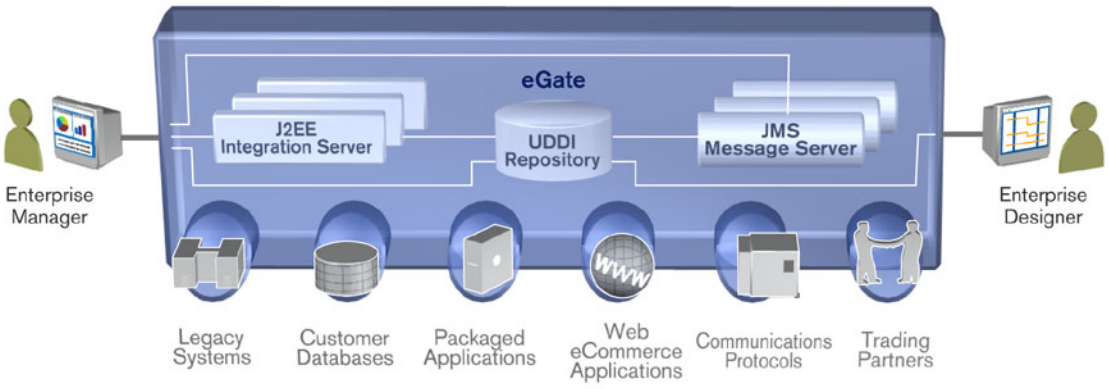
System Overview

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

2.1 Introduction

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

Figure 1 eGate Integrator



As shown in Figure 1, the heart of eGate Integrator is the Repository, which is a comprehensive store of information common to the entire enterprise. An integrated UDDI-compliant server allows publication and discovery of Web services. The run-time environment employs J2EE-compliant integration servers as operational engines and JMS-compliant message servers for the propagation of messages. The flexibility of the

eGate system allows the option of deployment to a SeeBeyond run-time environment or to third-party application servers, across a distributed network of hardware platforms.

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

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

2.2 Integration Model

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

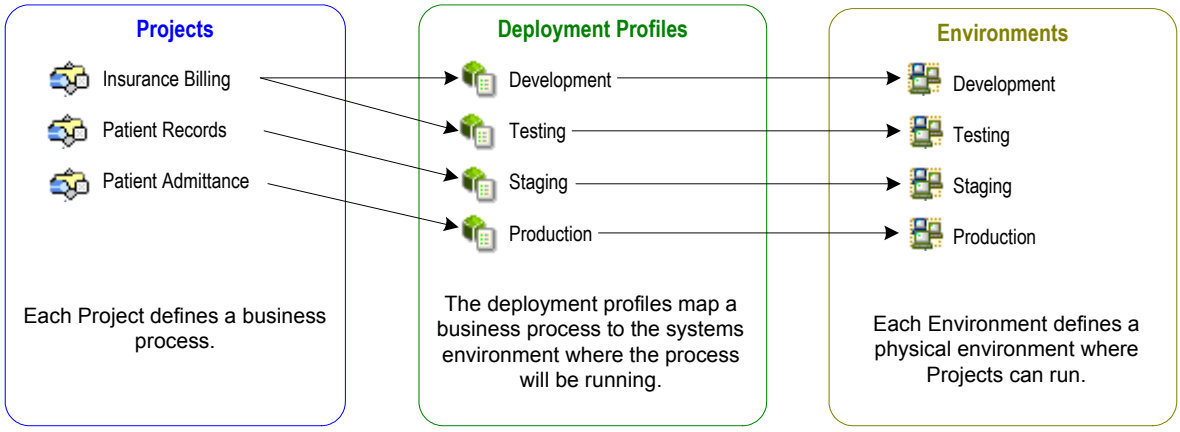
Projects are created using tools contained within Enterprise Designer and, once deployed, can be run and monitored using Enterprise Manager. Projects can also be set up to be run from the business process level using the SeeBeyond eInsight Business Process Manager, if that product is also installed.

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

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

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

Figure 2 eGate Integrator Implementation Model



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

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

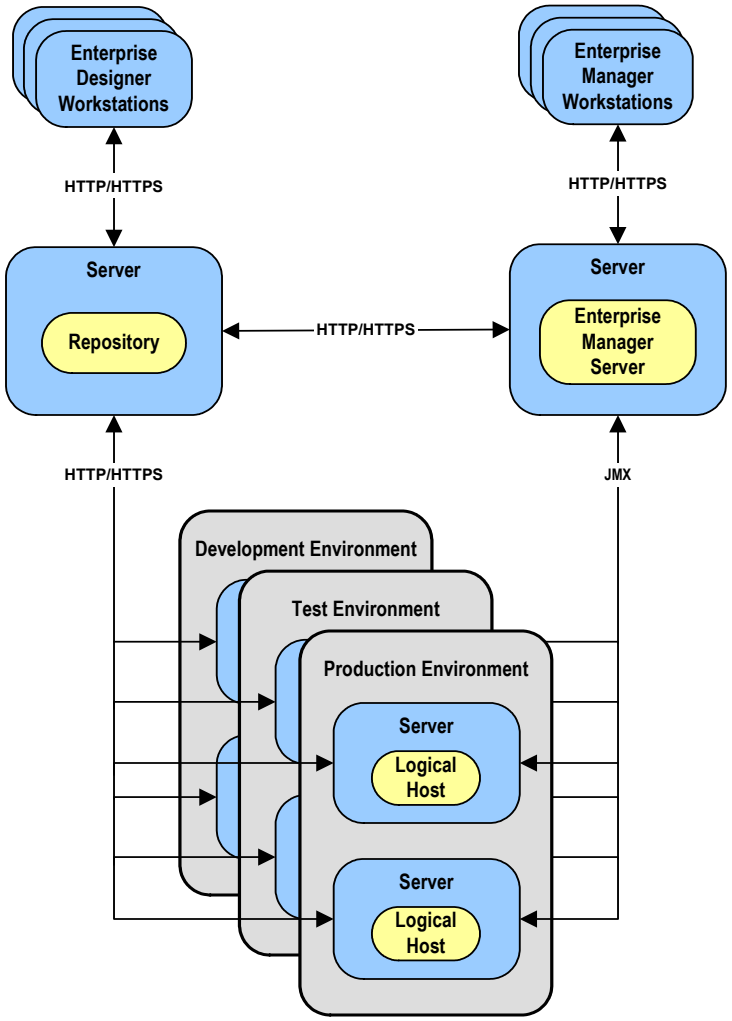
1. **Design your Project.**
2. **Define your Environments.**
3. **Create your Deployment Profiles.**
4. **Deploy the eGate Project.**

These implementation steps are all accomplished using Enterprise Designer, which is introduced in **Enterprise Designer** on page 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 3 shows an example system implementation that is highly distributed.

Figure 3 Typical eGate Integrator System



In Figure 3, the servers shown in blue are hardware platforms; the entities shown in yellow are software.

2.3.1 Repository

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

As shown in Figure 3, a single Repository serves the entire enterprise. This common Repository is used for development, testing, and production purposes. Communication between the Repository and other ICAN components can be configured to use either HTTP or HTTPS. The Repository makes Web Services available via a UDDI-compliant server. For more information on Web Services capability, see **Web Services** on page 145.

The Enterprise Designer and Enterprise Manager clients can communicate with the Repository through a firewall.

2.3.2 Environments

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

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

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

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

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

2.4 User Interfaces

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

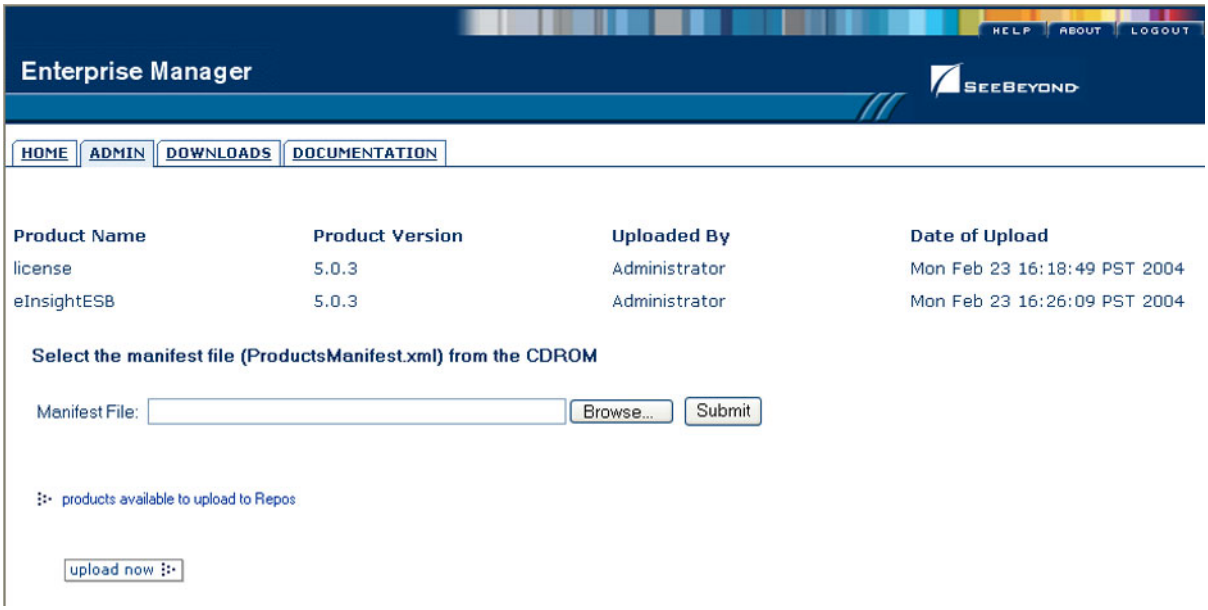
2.4.1 Enterprise Manager

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

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

The Enterprise Manager is described in **Enterprise Manager** *on page 26*. Figure 4 shows the Enterprise Manager **Admin** page, used in product installation.

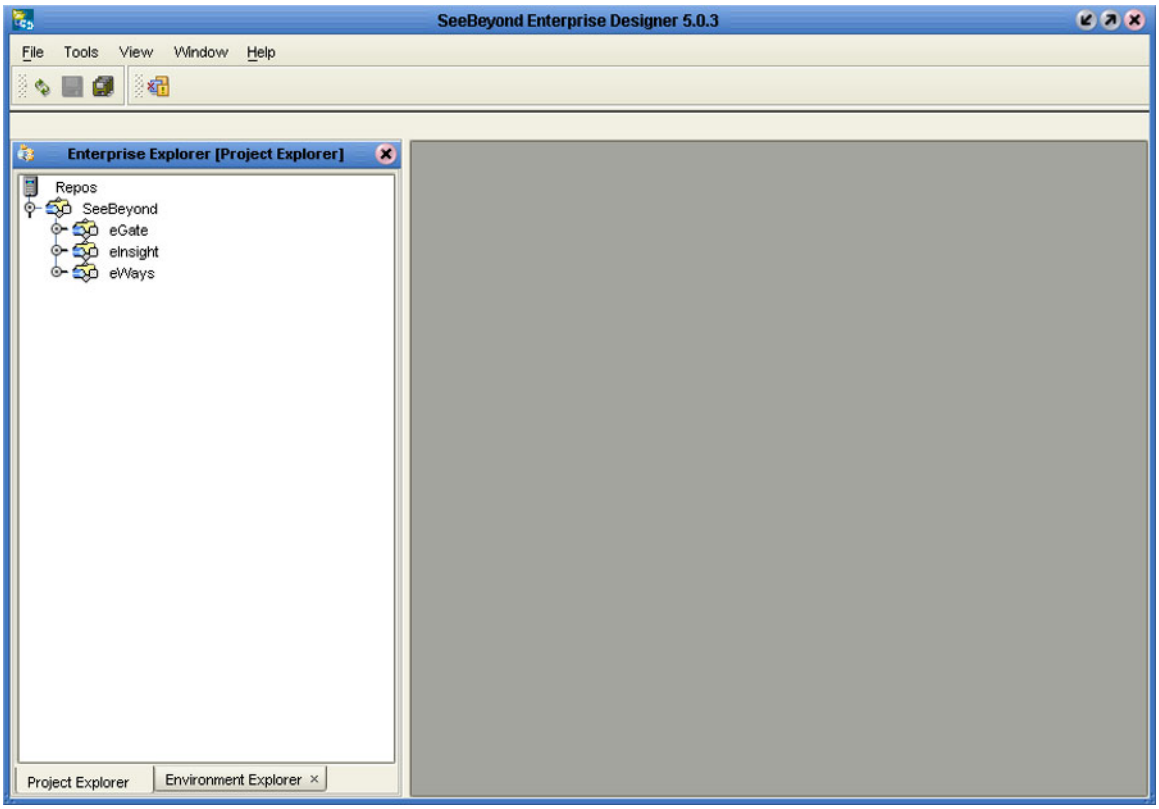
Figure 4 SeeBeyond Enterprise Manager



2.4.2 Enterprise Designer

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

Figure 5 Enterprise Designer

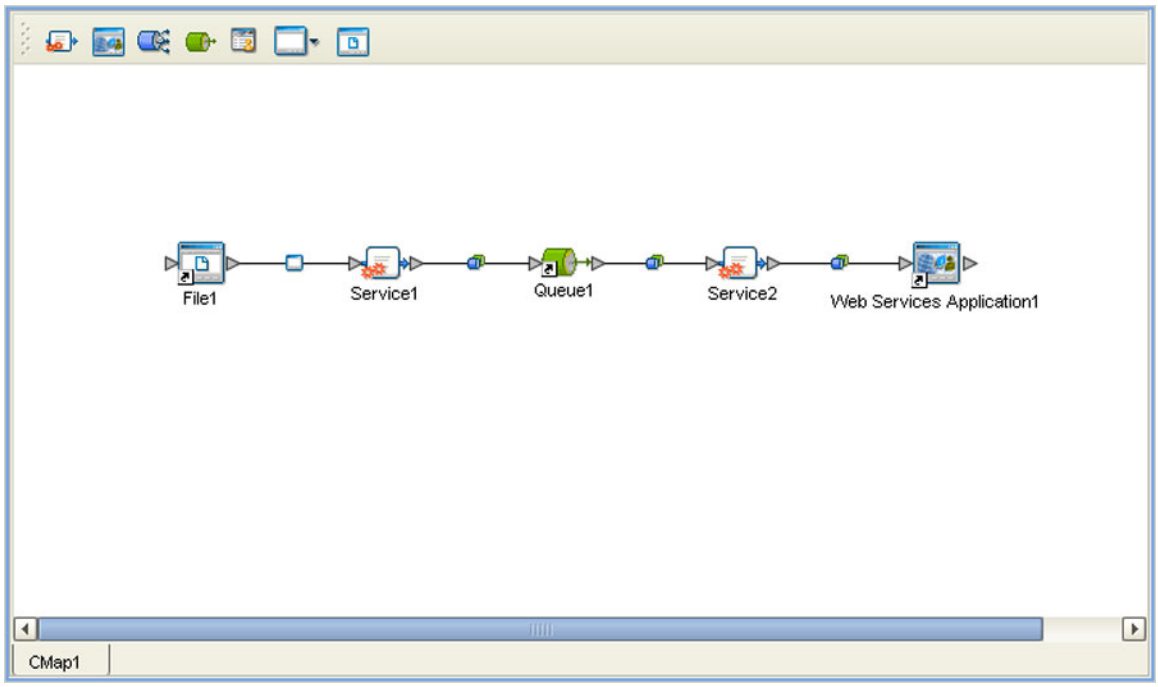


The major features of the Enterprise Designer are the Enterprise Explorer on the left, and an editor panel on the right—which is initially blank. The Enterprise Explorer follows the familiar Windows Explorer format, displaying a tree structure. The editor panel displays a variety of editors, depending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

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

The Connectivity Map Editor (see Figure 6) provides a graphic example of one of these editors, in which logical components of a Project can be created and connected. eGate uses Connectivity Maps to intuitively configure the end-to-end flow of messages within an integration. The integration developer can to drag and drop the various Collaborations, Intelligent Queues and external-system eWay adapters onto the Connectivity Map canvas and link them together to specify message flow. The features and usage of the Connectivity Map Editor are described in **eGate Projects** on page 64.

Figure 6 Connectivity Map Editor



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

For more information on the Enterprise Designer, see **Enterprise Designer** *on page 37* .

Chapter 3

Enterprise Manager

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

3.1 Overview

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

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

3.1.1 Installing and Updating eGate

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

3.1.2 Monitoring and Managing eGate

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

- **The ICAN Monitor** on page 31 describes the basic features of the ICAN Monitor interface. ICAN Monitor usage for specific tasks is described in the *eGate Integrator System Administration Guide*.
- **The SRE Monitor** on page 32 describes an optional facility that allows you to monitor and manage e*Gate 4.x schemas in eGate 5.0, using the Schema Runtime Environment.

3.2 Starting Enterprise Manager

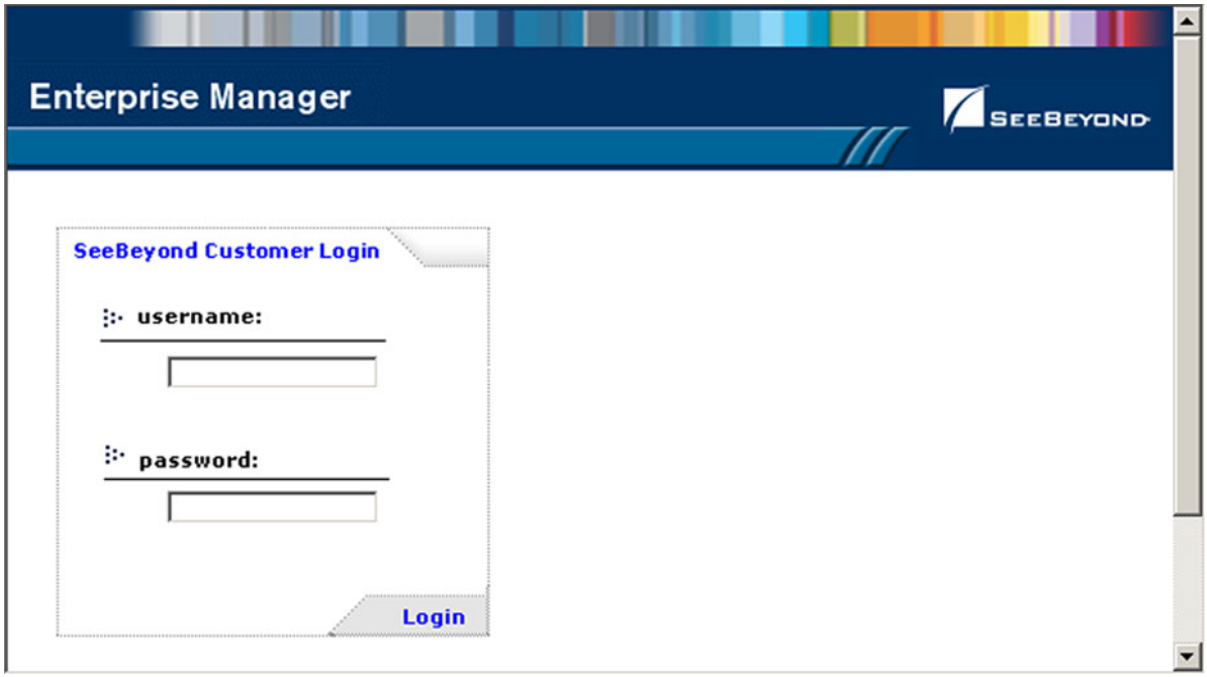
To start the Enterprise Manager

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

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

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

Figure 7 Enterprise Manager Login

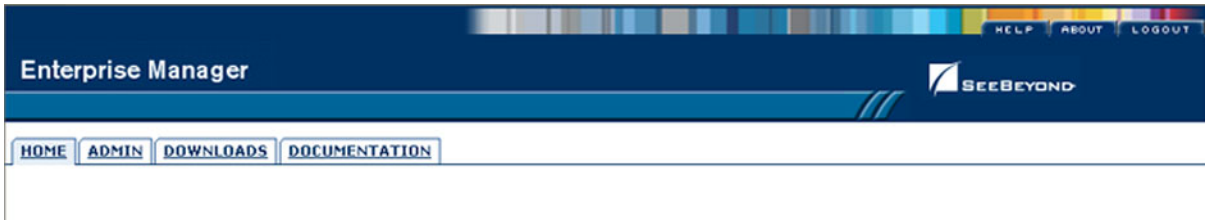


3. Enter your login ID and password in the **Username** and **Password** boxes and click **Login**.

3.3 The Enterprise Manager Interface

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

Figure 8 Enterprise Manager GUI



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

Table 2 Enterprise Manager - Pages

Tab	Function
Home	The Home page is used for accessing the ICAN Monitor, which is the main subject of this chapter. See **Home** on page 29.
Admin	The Administration page is used for installing and updating ICAN components. See the *SeeBeyond ICAN Suite Installation Guide* for information.
Downloads	The Downloads page is used in installing and updating ICAN components. See the *SeeBeyond ICAN Suite Installation Guide* for information.
Documentation	The Documentation page is used for accessing ICAN Suite documentation. See **Documentation** on page 30, and the following *Note*.

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

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

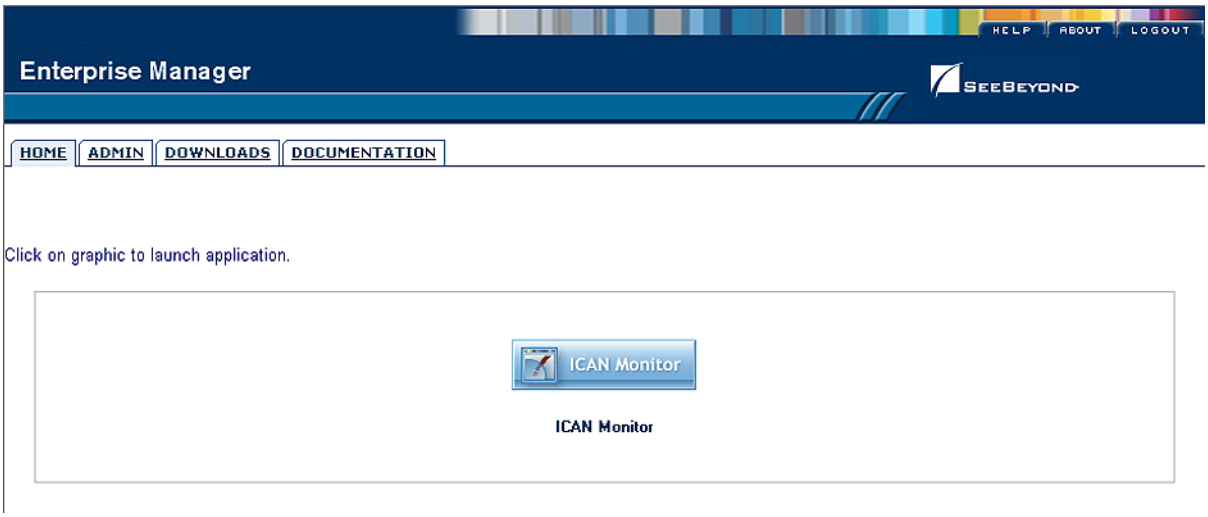
Table 3 Enterprise Manager - Control Tabs

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

3.3.1 Home

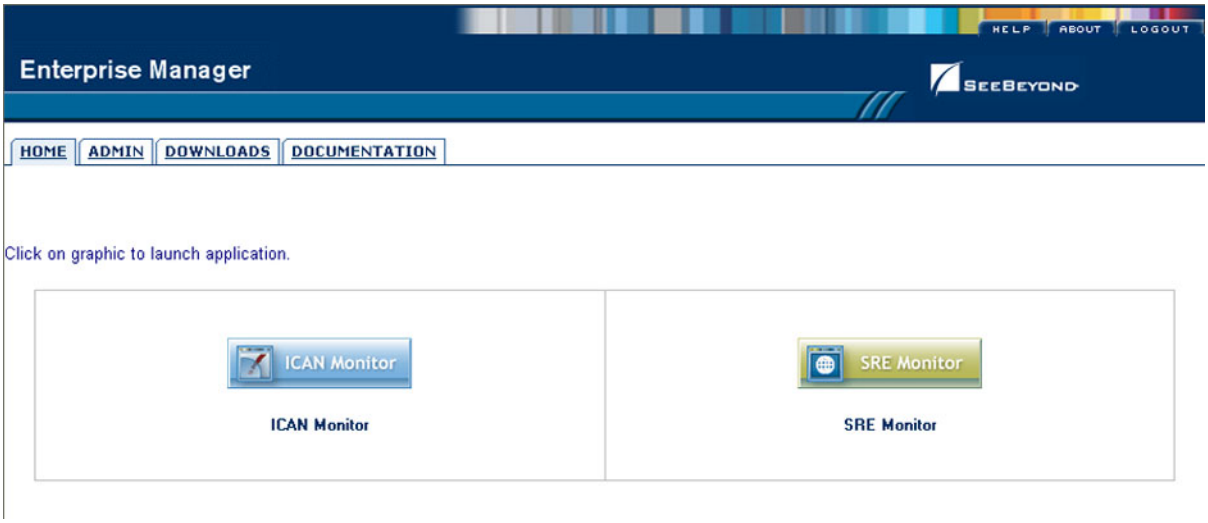
The Enterprise Manager's **Home** tab (see Figure 9) contains a link to the ICAN Monitor. Click the **Monitor** icon to launch the ICAN Monitor (see **The ICAN Monitor** on page 31).

Figure 9 ICAN Monitor Launch Icon



If the SRE Monitor (see **The SRE Monitor** on page 32) is installed, its icon is also displayed on this page, as shown in Figure 10.

Figure 10 ICAN Monitor/SRE Monitor Launch Icons

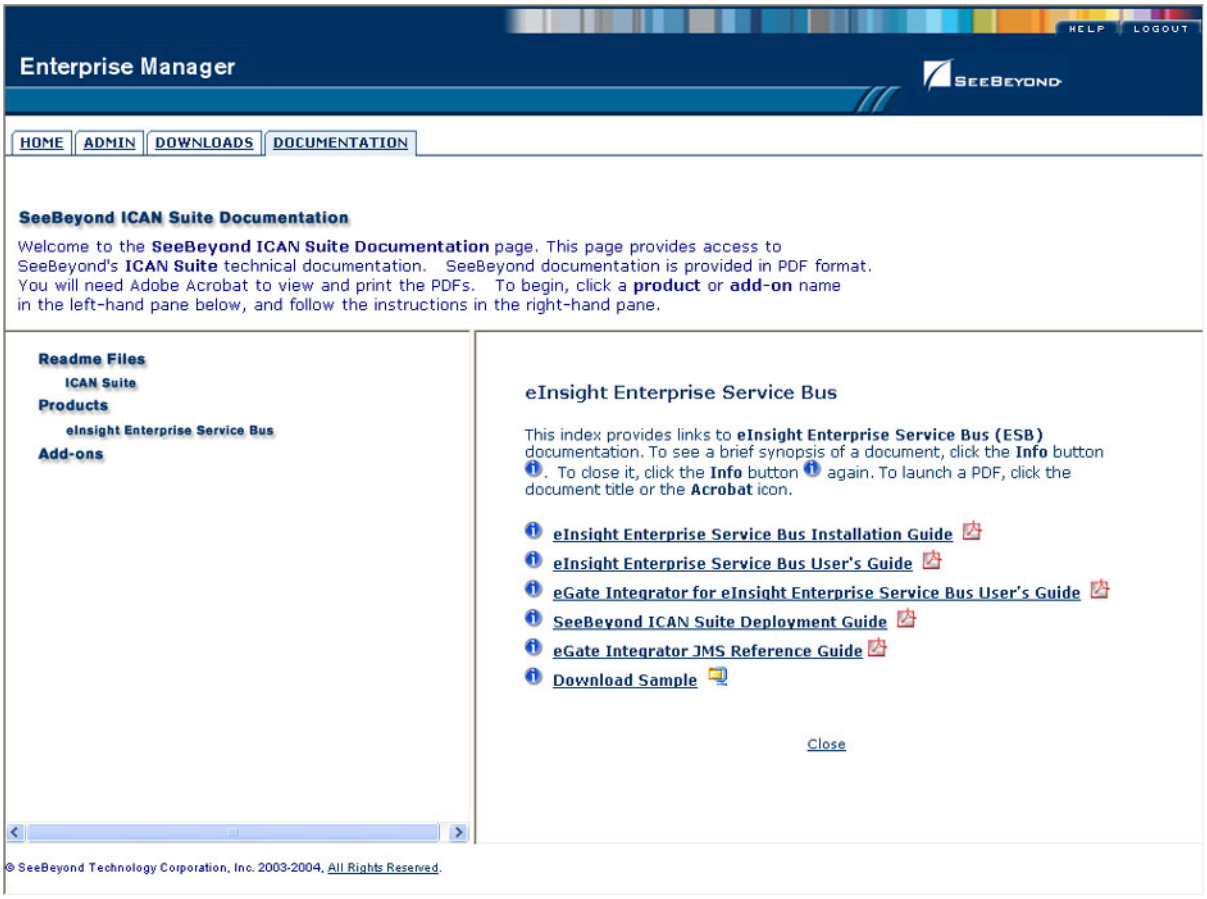


Note: *If connection problems are encountered, close all Internet Explorer windows and retry.*

3.3.2 Documentation

The **Documentation** tab (see Figure 11) contains links to the latest versions of the SeeBeyond ICAN documentation in Adobe Acrobat (PDF) format, and also any sample Project files (in ZIP format). Shown is the current set for eGate Integrator.

Figure 11 Documentation Tab



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

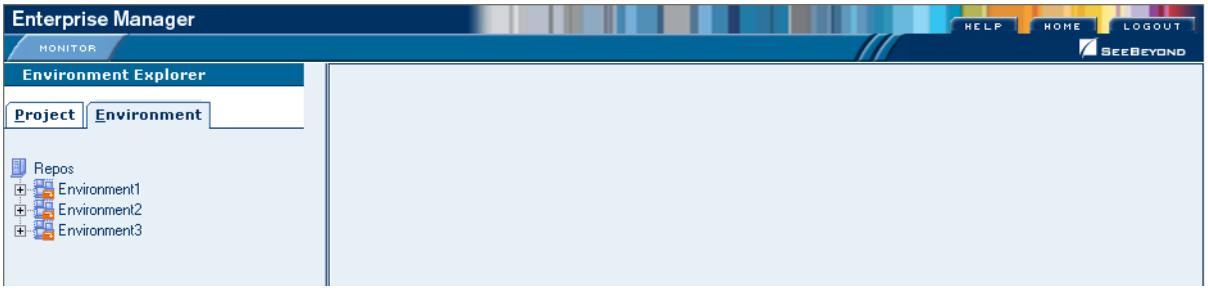
Table 4 Document Categories

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

3.4 The ICAN Monitor

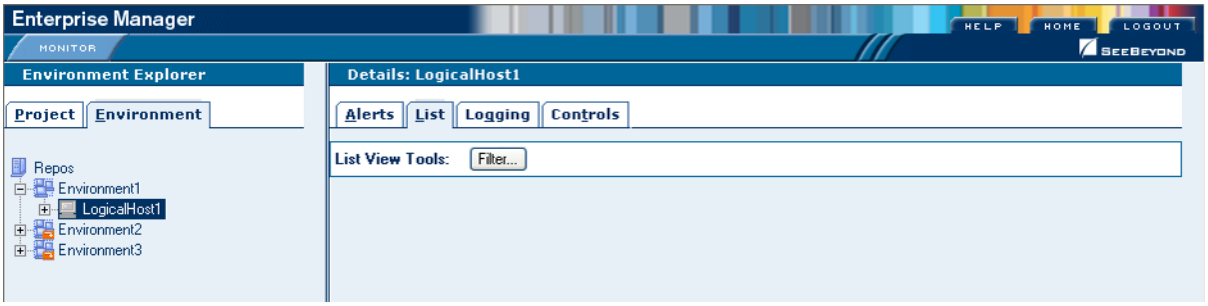
The ICAN Monitor has structure similar to that of the Enterprise Designer, with an Explorer panel on the left and a Details panel on the right. Initially, the Details panel is blank as shown in Figure 12.

Figure 12 ICAN Monitor Interface - Initial



Like the Enterprise Manager itself, the ICAN Monitor's **Details** area is organized into sections represented by tabs (see Table 5). Which tabs are present depends upon the component selected in the Explorer. For example, selecting the Logical Host displays the Monitor page shown in Figure 13 .

Figure 13 ICAN Monitor Interface - Environment



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

Table 5 ICAN Monitor Interface - Details Tabs

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

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

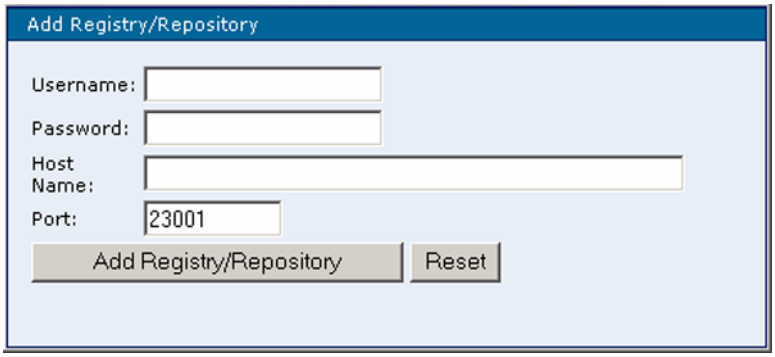
3.5 The SRE Monitor

eGate 5.0 provides a completely different operating environment from earlier versions of the product (e*Gate). The Schema Runtime Environment (SRE) allows you to use schemas developed for e*Gate 4.x with eGate 5.0 by providing the necessary environmental components. Instructions for installing and using the SRE are contained in the SeeBeyond documentation for the SRE.

The SRE Monitor enables you to manage e*Gate 4.x schemas running in the Schema Runtime Environment from within eGate 5.0. The SRE Monitor interface generally resembles the ICAN Monitor, but differs somewhat in detail (see Figure 14). Only the Environment Explorer is displayed, which has two additional icons in the upper left corner; these are described in Table 6.

Figure 14 SRE Monitor

Table 6 SRE Monitor Explorer Icons

Icon	Function
	The **Add Registry/Repository** icon displays the **Add Registry/Repository** dialog box, in which you specify the desired Registry or Repository's name and port, and your user name and password. 
	The **Refresh Registry** icon refreshes the SRE Registry and the Explorer tree following changes to component status.

Instructions for installing the SRE Monitor are contained in the *SeeBeyond ICAN Suite Installation Guide*.

3.5.1 Starting and Using the SRE Monitor

To start the SRE Monitor

1. Start the SRE Monitor server, as described in the *SeeBeyond ICAN Suite Installation Guide*.
2. Launch Internet Explorer and access Enterprise Manager, as described in **Starting Enterprise Manager** on page 27.
3. On the Enterprise Manager Home page, click the **SRE Monitor** icon shown in Figure 15 to display the initial page of the SRE Monitor (see Figure 16).

Figure 15 Enterprise Manager Home Page

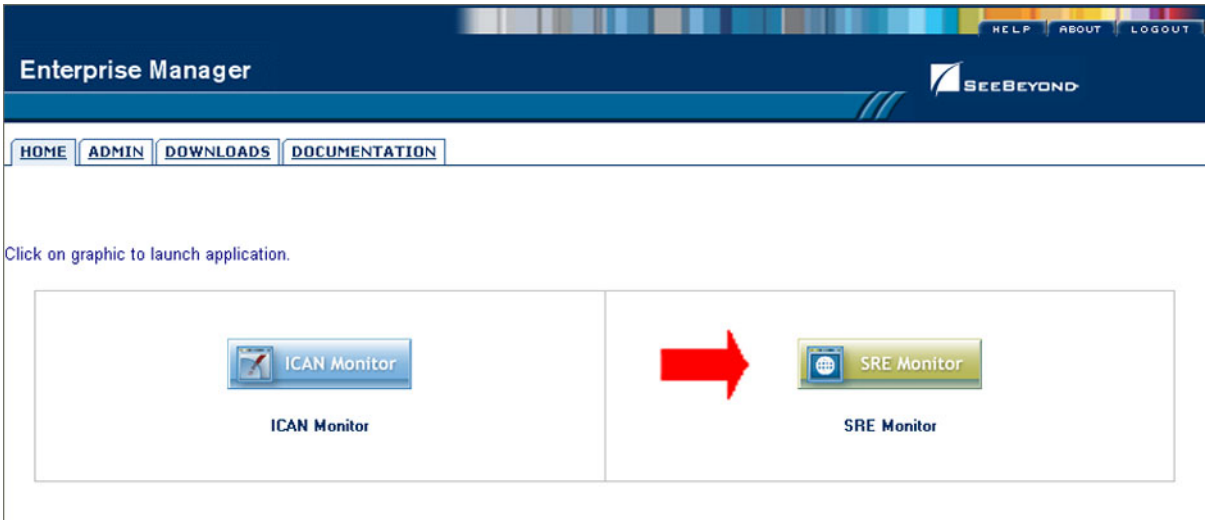
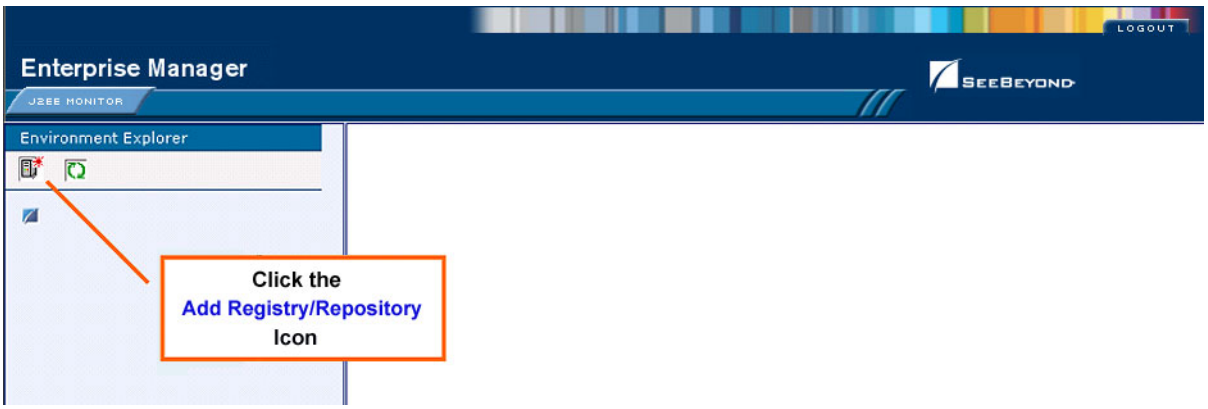
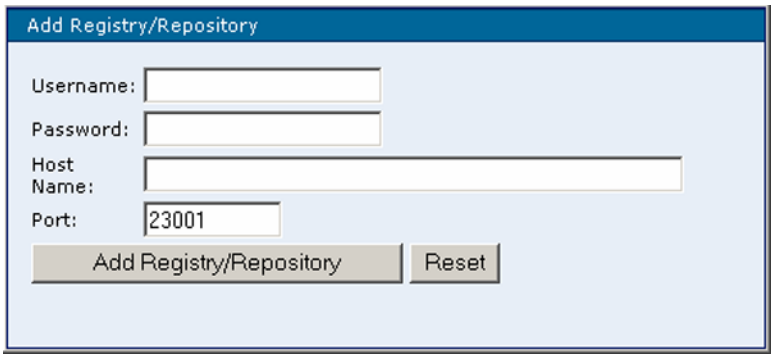


Figure 16 SRE Monitor Initial Page



4. Click the **Add Registry/Repository** icon in the upper left corner of the Explorer. This displays the dialog box shown in Figure 17.

Figure 17 Add Registry/Repository Dialog Box



5 Enter your login ID and password, and the Repository Host Name and Port, and click **Add Registry/Repository**.

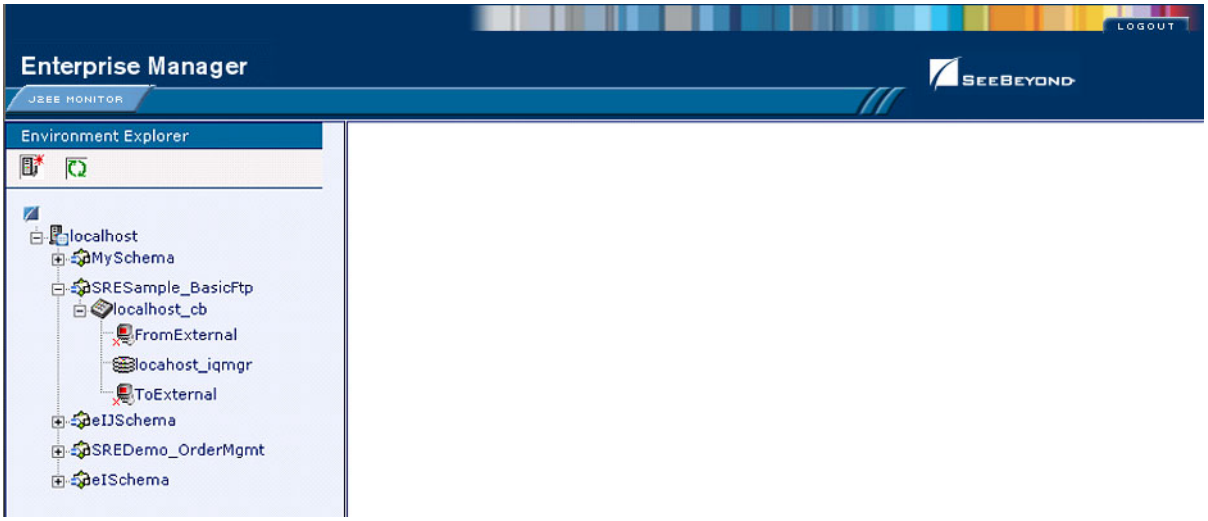
Note: *The **Host Name** is the host name of the server where you installed the e*Gate 4.x Registry, and the **Port** is the number of the port you entered during installation of the e*Gate 4.x Registry. See the SeeBeyond ICAN Suite Installation Guide.*

Important: *The Host Name must be composed of alphanumeric characters only.*

To use the SRE Monitor

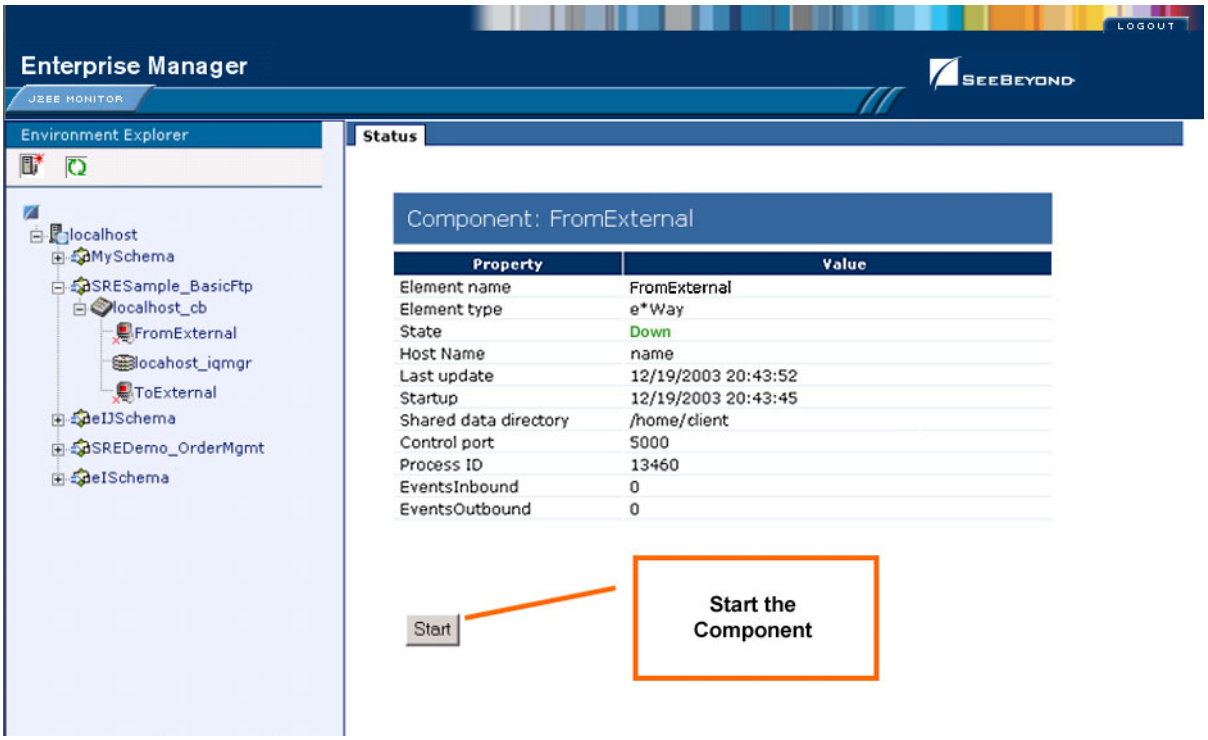
1 After adding the Registry, expand the Control Broker in the Explorer tree to view the SRE components, as shown in Figure 18.

Figure 18 Viewing SRE Components



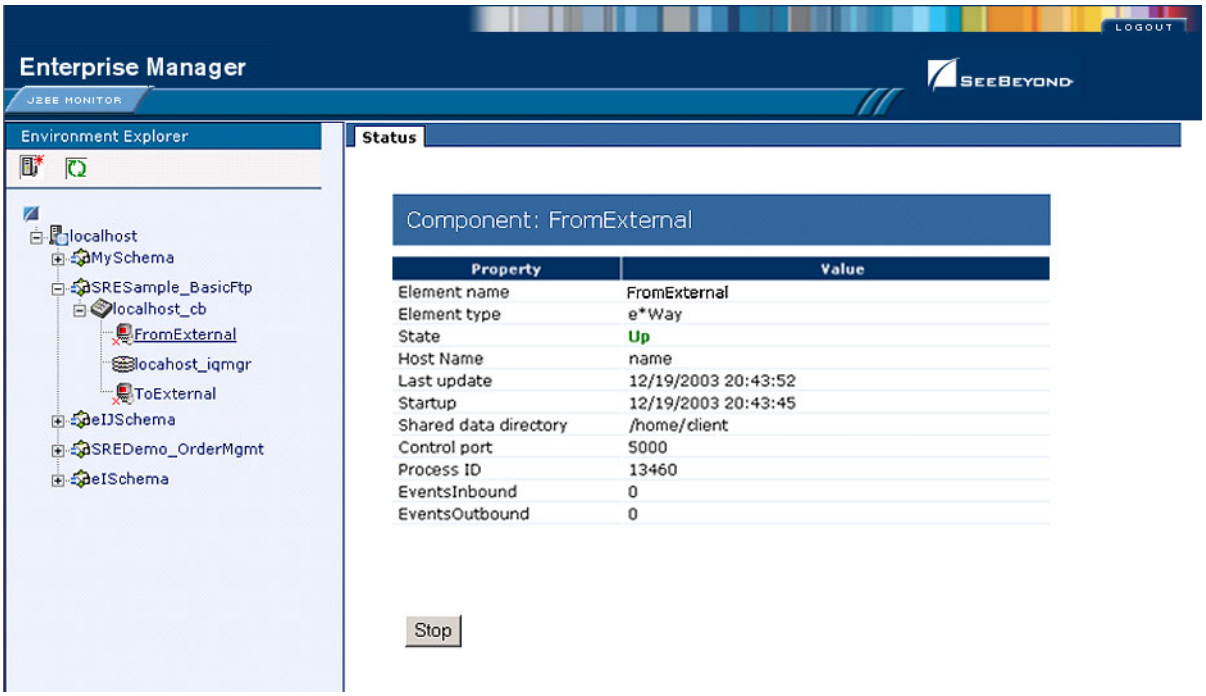
2 Click on a component to display its status. As an example, the status of the *FromExternal* e*Way is shown in Figure 19. You can start the component by clicking the **Start** button, which becomes a **Stop** button when the component is running.

Figure 19 FromExternal e*Way Status (Not Running)



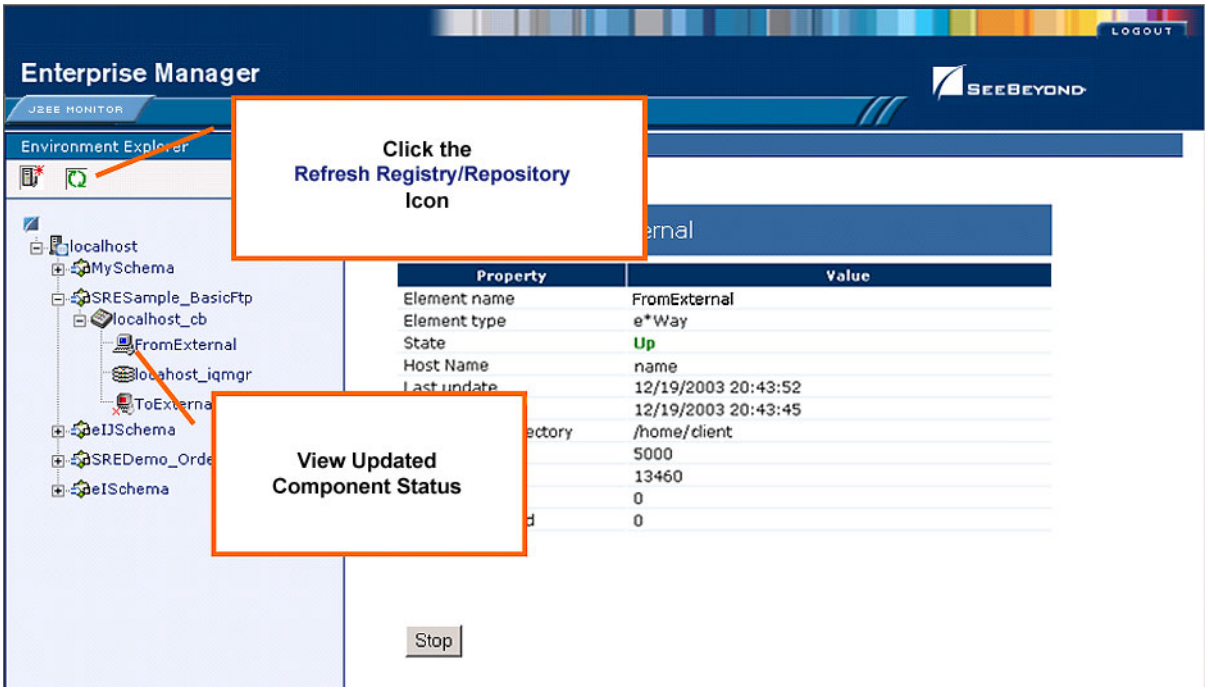
3 The status page is updated after you start or stop the component, as shown in Figure 20.

Figure 20 FromExternal e*Way Status (Running)



4 To refresh the Explorer tree, click the **Refresh Repository** icon, as shown in Figure 21.

Figure 21 Refreshing the Explorer Tree



Chapter 4

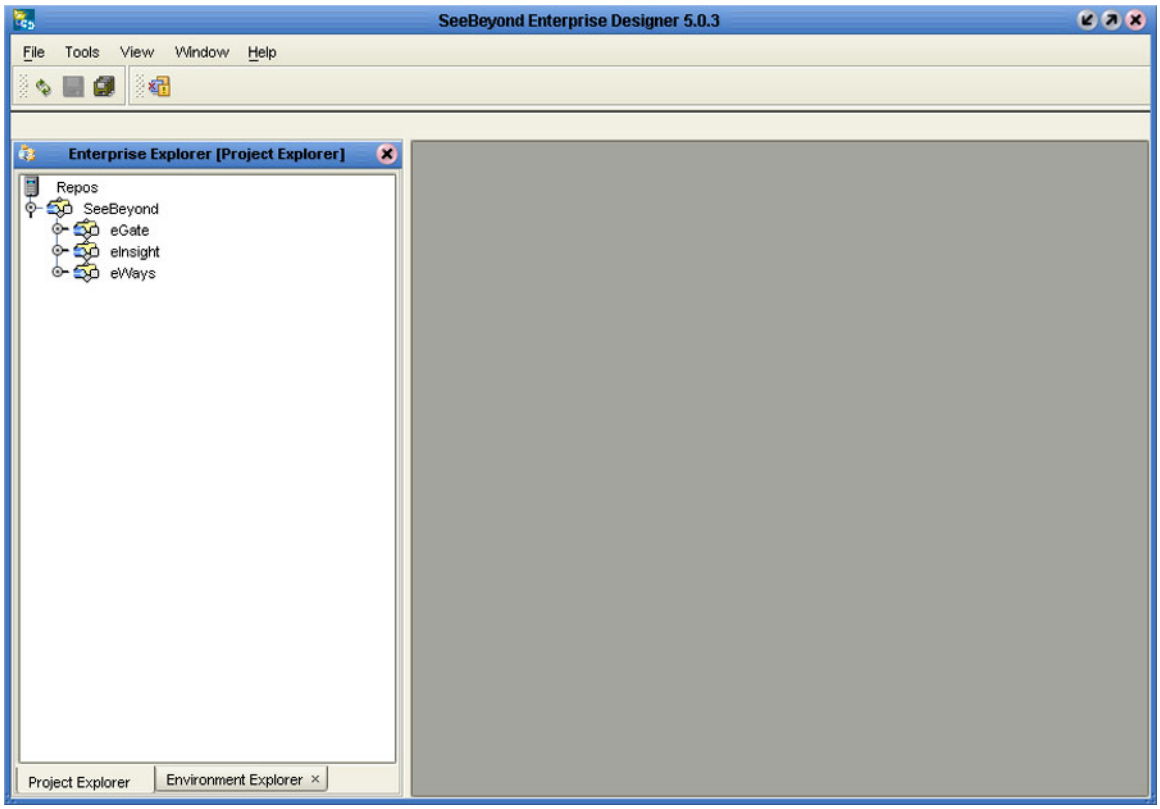
Enterprise Designer

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 22), you can develop Projects to process and route data through an eGate Integrator system.

Figure 22 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 43
- **Environment Explorer on page 44**

4.1.1 Editors

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

- **Connectivity Map Editor** on page 45
- **OTD Editor** on page 46
- **Environment Editor** on page 47
- **Deployment Editor** on page 48

4.1.2 Analysis and Archiving Tools

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

- **Project/Environment Import** on page 49, which allows you to import a Project that has been created elsewhere.
- **Project/Environment Export** on page 54, which allows you to export a Project to an external file so that it may be used elsewhere.
- **Impact Analyzer** on page 59, which helps you visualize how a change to one part of a Project would affect the rest of the Project.
- **Version Control** on page 61, which allows you to maintain multiple versions of Project components.

4.1.3 User Interface

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

- **Menus** on page 40 describes the options contained in the individual menus.
- **Toolbar** on page 42 describes the functionality of the toolbar icons.
- **Browser Buttons** on page 42 describes the browser buttons that appear throughout the Enterprise Designer, in various wizards and dialog boxes.

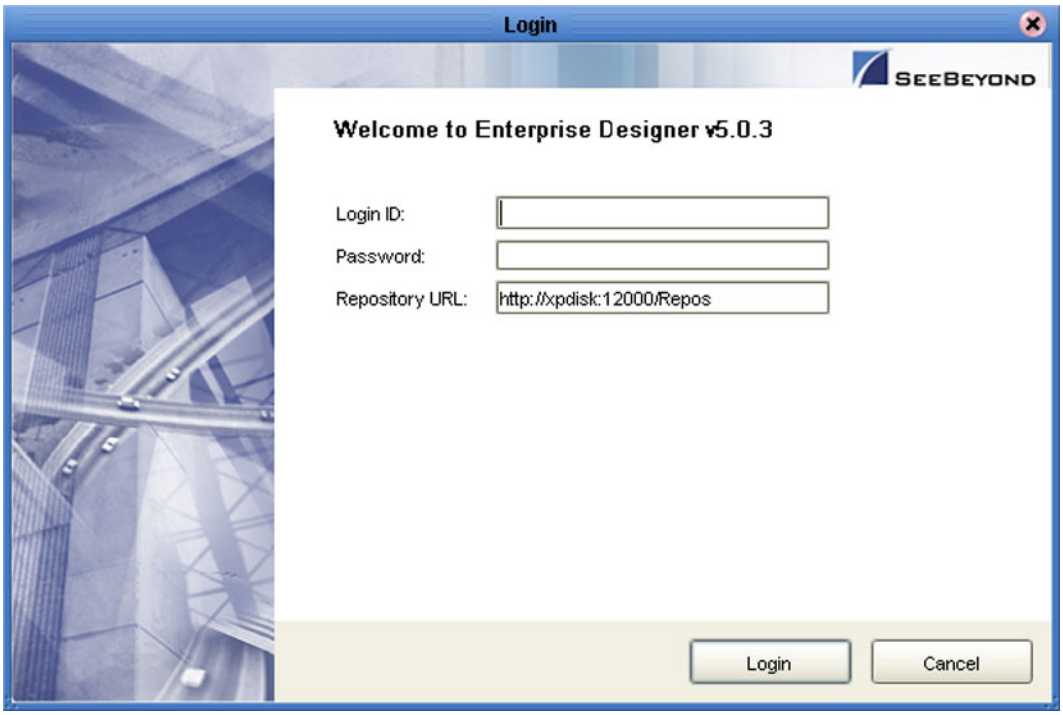
The procedure for invoking the Enterprise Designer is described in **Starting Enterprise Designer** on page 39.

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

Figure 23 Login Dialog Box



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

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

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

5 Click **Login** to complete the login process and display the Enterprise Designer GUI shown in Figure 22. A progress monitor will appear while the process is running.

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

7 Click **Login** to complete the login process and display the Enterprise Designer GUI shown in Figure 22.

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.

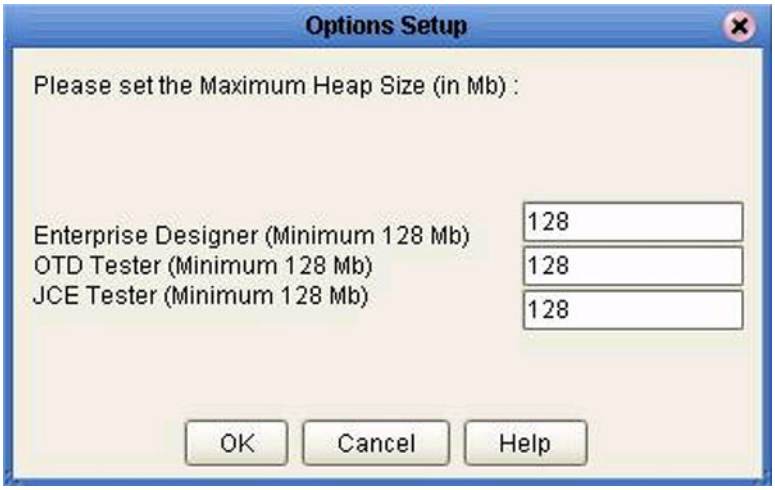
File Menu

Table 7 File Menu Options

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

Tools Menu

Table 8 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 59.
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 9 View Menu Options

Option	Function
Environment Explorer	Activates the **Environment Explorer** tab on the Enterprise Explorer. See **Environment Explorer on page 44**.
Project Explorer	Activates the **Project Explorer** tab on the Enterprise Explorer. See **Project Explorer** on page 43.

Window Menu

Table 10 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 11 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 component.
About Enterprise Designer	Displays an information box giving the version number, copyright information, and currently-active Repository connection information.

4.3.2 Toolbar

Table 12 Enterprise Designer Toolbar Icons

Icon	Function
	**Refresh All from Repository** refreshes the Project Explorer and Environment Explorer to display the current contents of the Repository. (You are prompted to save any changes before the refresh occurs.) Open editors are not refreshed.
	**Save** saves changes 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 13 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

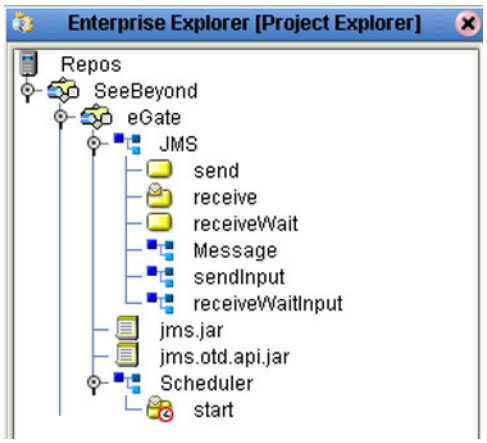
The Enterprise Explorer organizes the components of a Project into tabs that display different views of an eGate system.

- **Project Explorer** on page 43 deals with logical components.
- **Environment Explorer on page 44** 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 24.

Figure 24 Enterprise Explorer: Project Explorer View



Details of the features and usage of the Project Explorer are found in **eGate Projects** on page 64.

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 25 Enterprise Explorer: Environment Explorer View

Details of the features and usage of the Environment Explorer are found in **Environments** on page 104.

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 45
- **OTD Editor** on page 46
- **Environment Editor** on page 47
- **Deployment Editor** on page 48

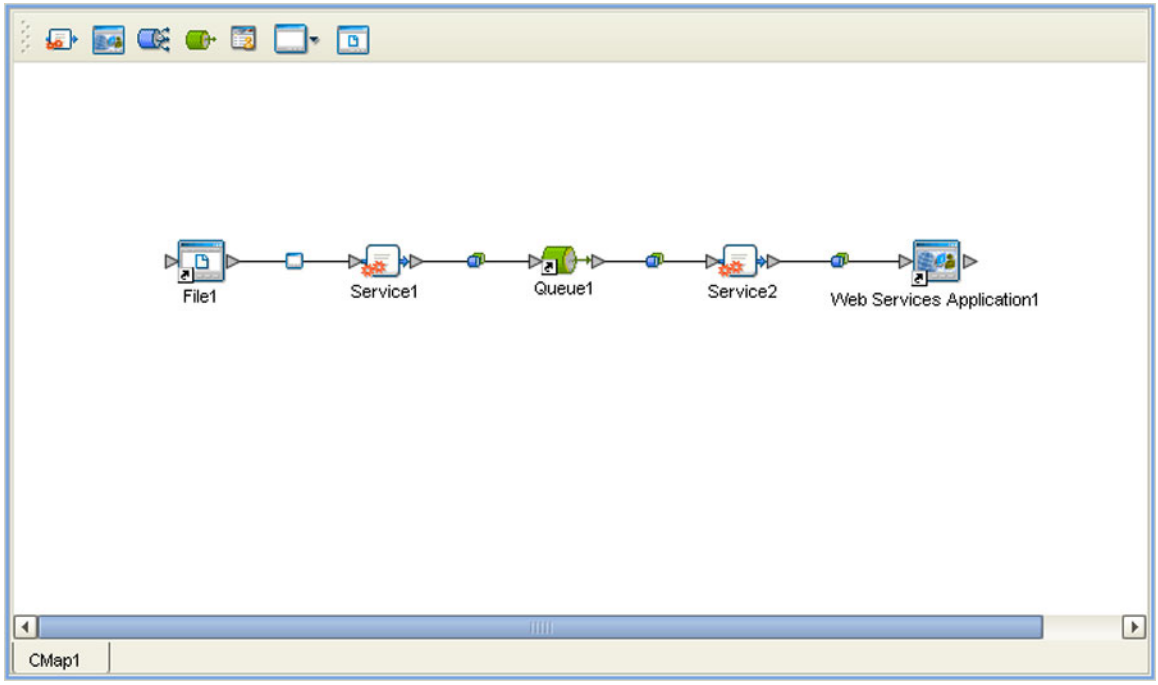
4.5.1 Connectivity Map Editor

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 26, 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 70 for detailed information.

Figure 26 Connectivity Map Editor

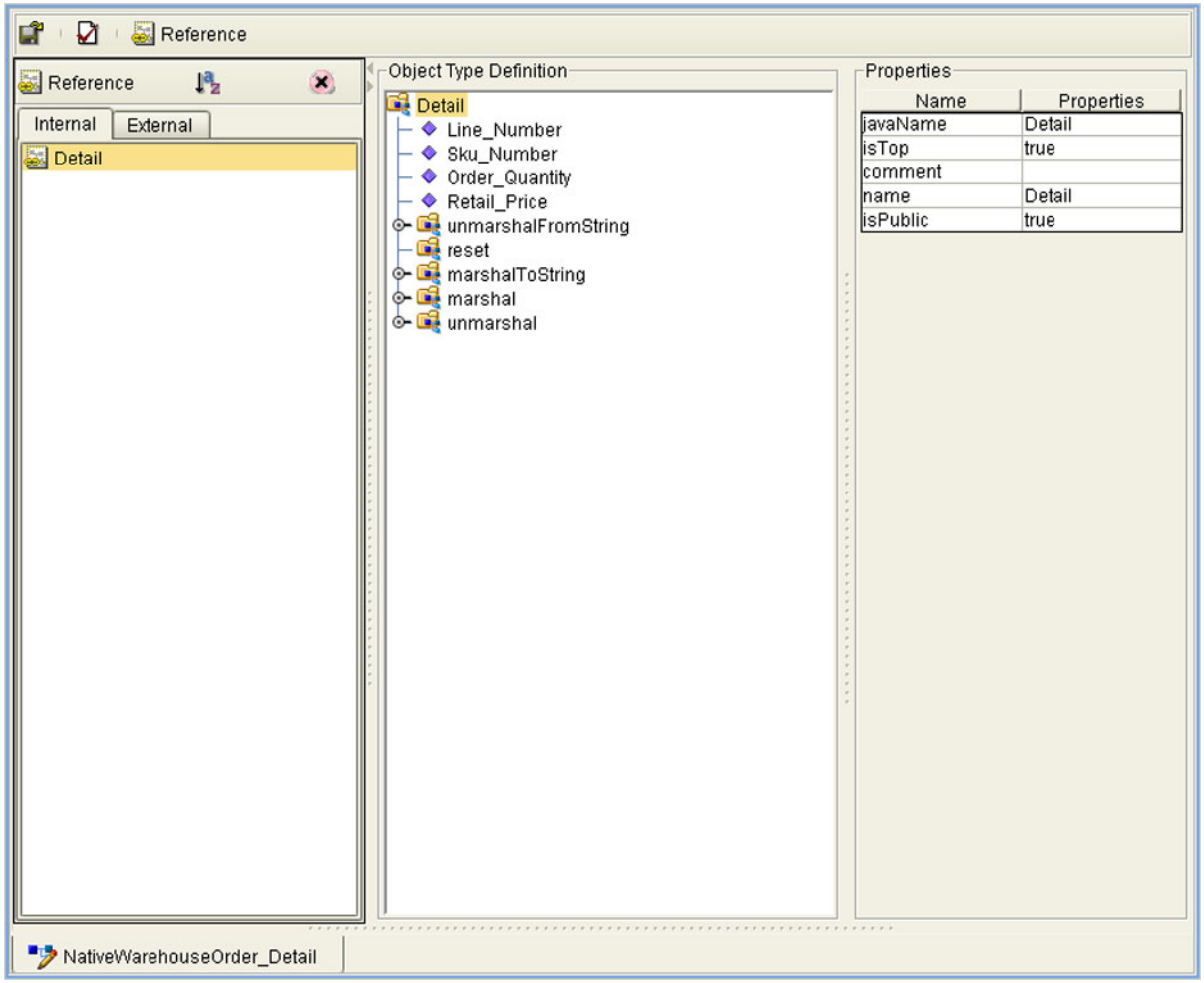


4.5.2 OTD Editor

The OTD Editor window, as shown in Figure 27, displays the source files used to create the Object Type Definitions (OTDs) to use with a Project. You use an OTD wizard tool to create OTD files and add them to the **Project Explorer** tab.

See **Using the OTD Editor** on page 98 for detailed information.

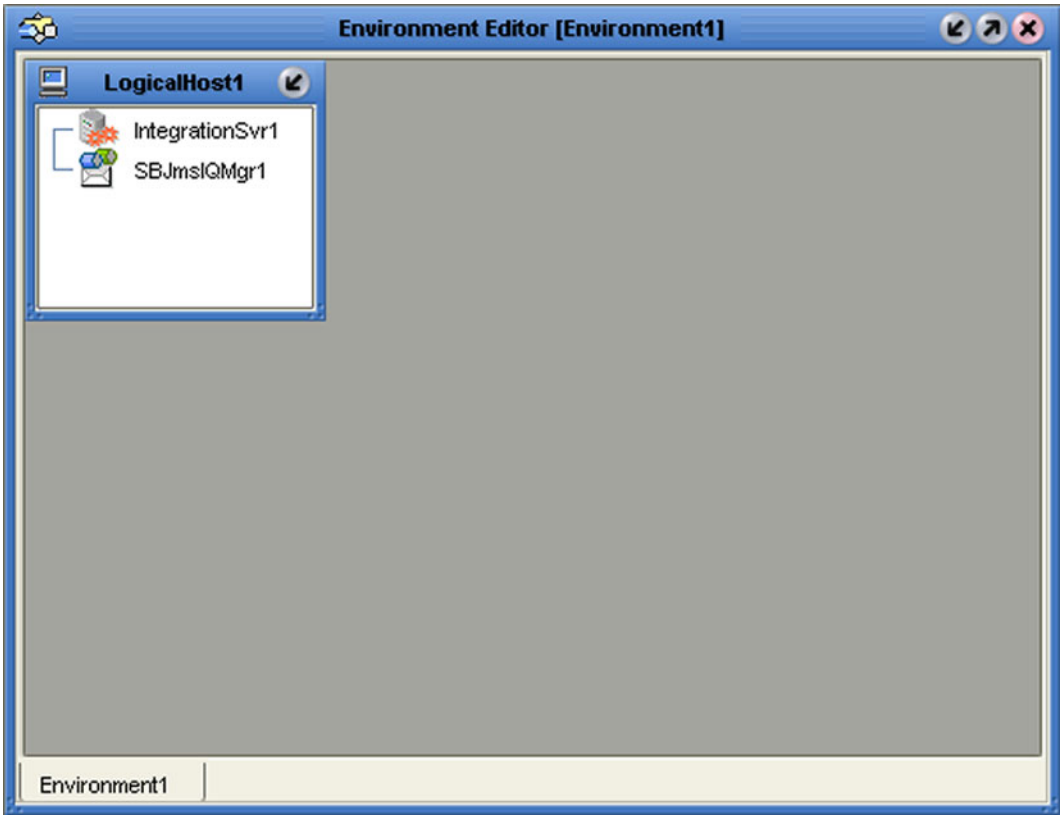
Figure 27 OTD Editor



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 28.

Figure 28 Environment Editor

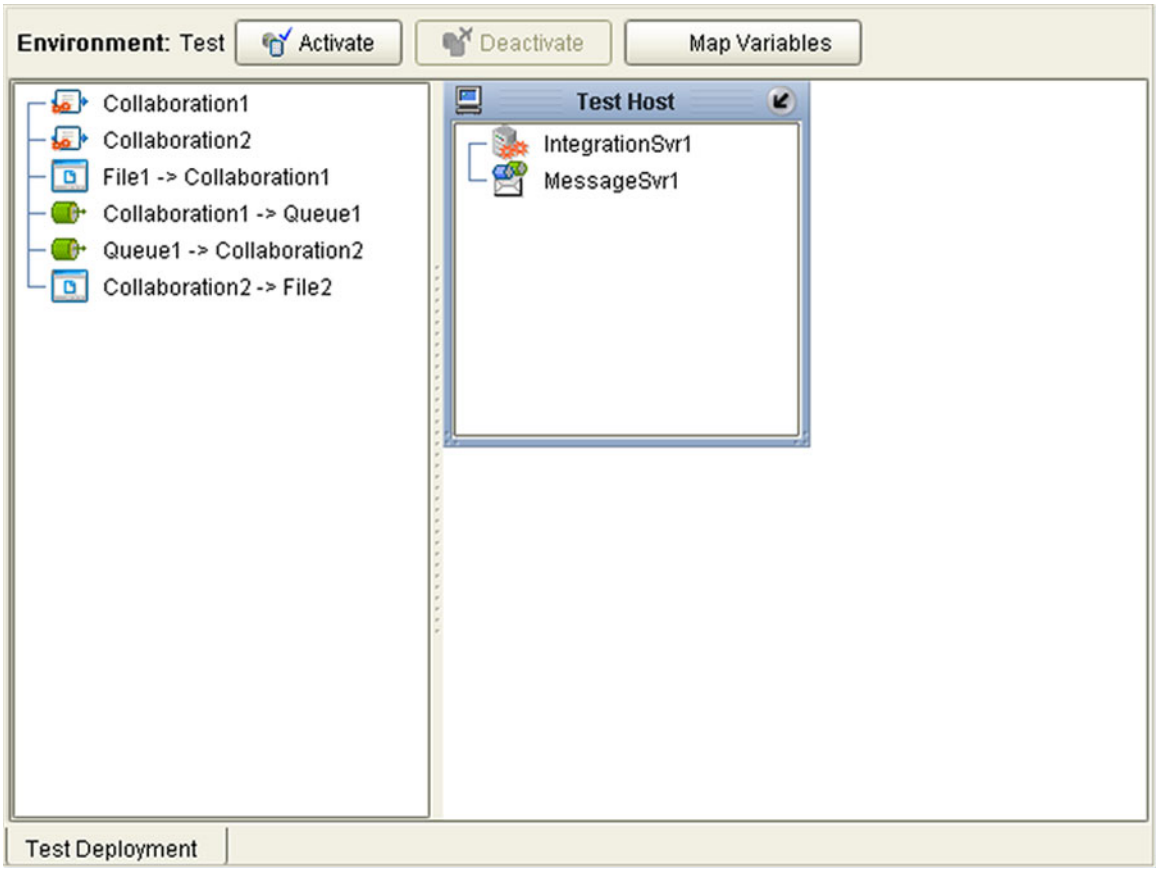


Note: *Unlike changes to Project-related configuration properties, changes to Environment-related properties do not require redeployment, just application.*

4.5.4 Deployment Editor

The Deployment Editor, as shown in Figure 29, contains information about how Project components will be deployed in an Environment. See **The Deployment Editor** on page 132 for detailed information

Figure 29 Deployment Editor



4.6 Additional Tools and Features

4.6.1 Project/Environment Import

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

Important: *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:*

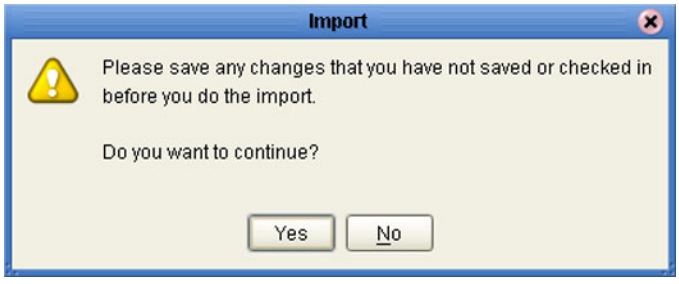
ICAN-root`\repository\logs\repository.log`

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.**
2. The message box shown in Figure 30 appears, prompting you to save your changes.

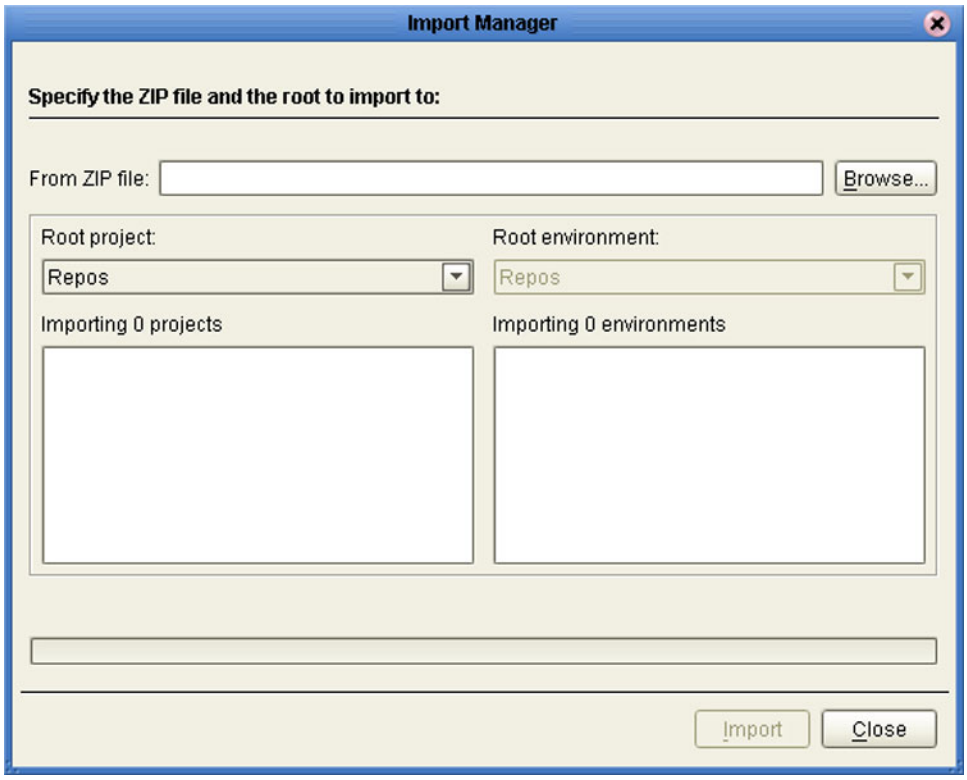
Figure 30 Import Message Box



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

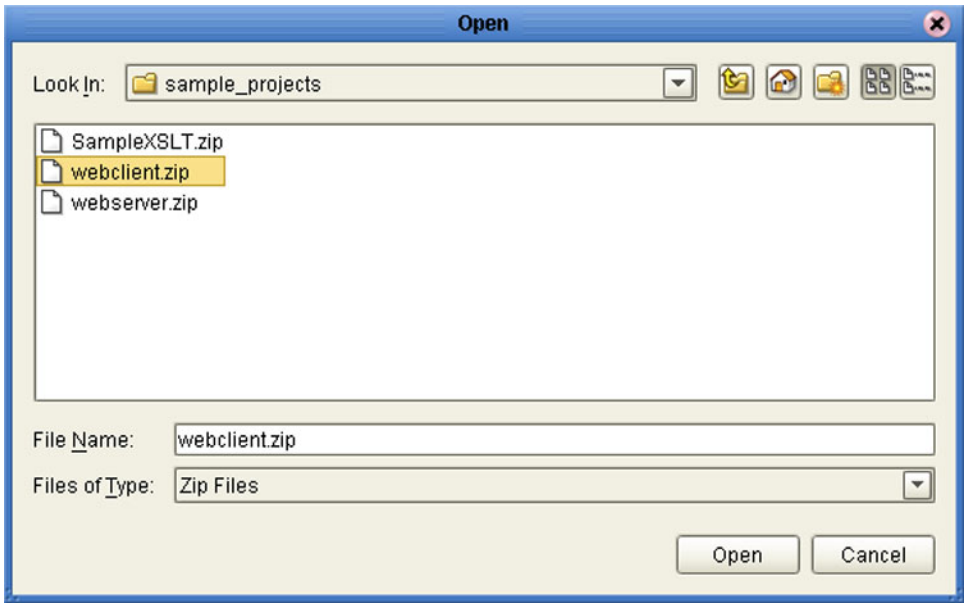
B If you have saved any desired changes, click **Yes** to display the dialog box shown in Figure 31.

Figure 31 Import Manager Dialog Box (1)



3 Click the **Browse** button to display the *Open File* dialog box, as shown in Figure 32. If you browse to an Environment file, the *Root environment* field will be enabled.

Figure 32 Open File Dialog Box

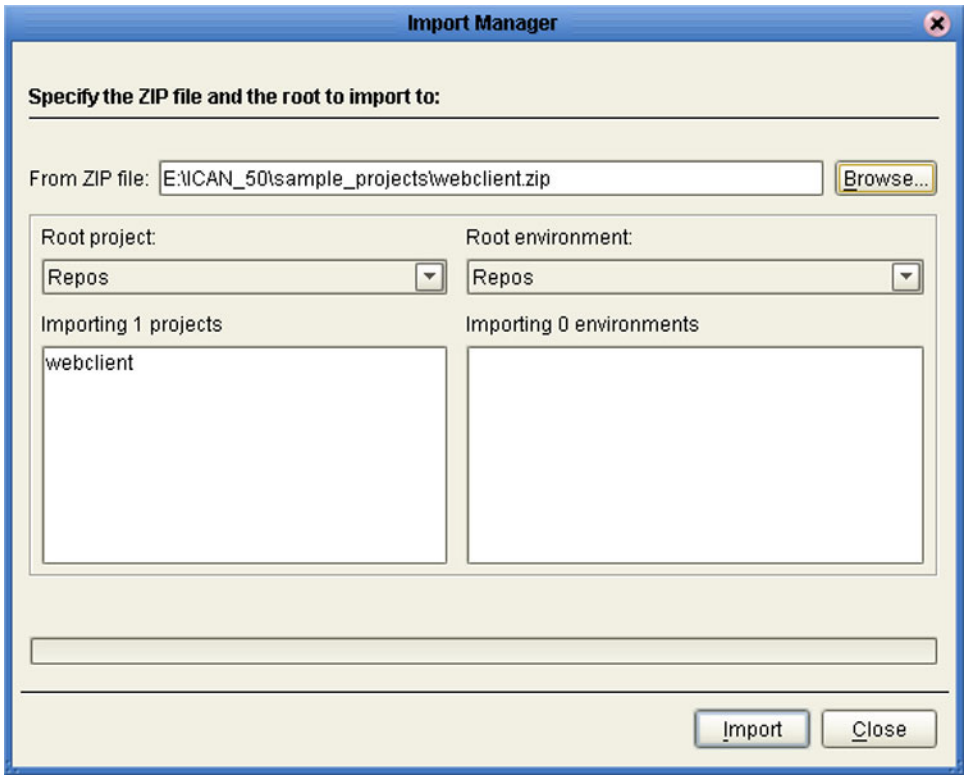


4 Locate and select the Project or Environment file that you want to import.

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

6 The Import Manager dialog box appears as shown in Figure 33, in which you specify the desired destination and file name (if different from the original).

Figure 33 Import Manager Dialog Box



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

8 The Import Status message box shown in Figure 34 appears after the file has been imported successfully.

Figure 34 Import Status Message Box

9 Click **OK** to close the message box. The Project Explorer will automatically be refreshed from the Repository.

Importing a Project Using the Command Line

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

Location of script file:

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

Command Syntax:

`importProject` ***`username password importfile rootprojectname`***

where:

- ***`importfile`*** is the 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 35. If you do, you will see the dialog box shown in Figure 36.

Figure 35 Export Manager Dialog Box (1a)

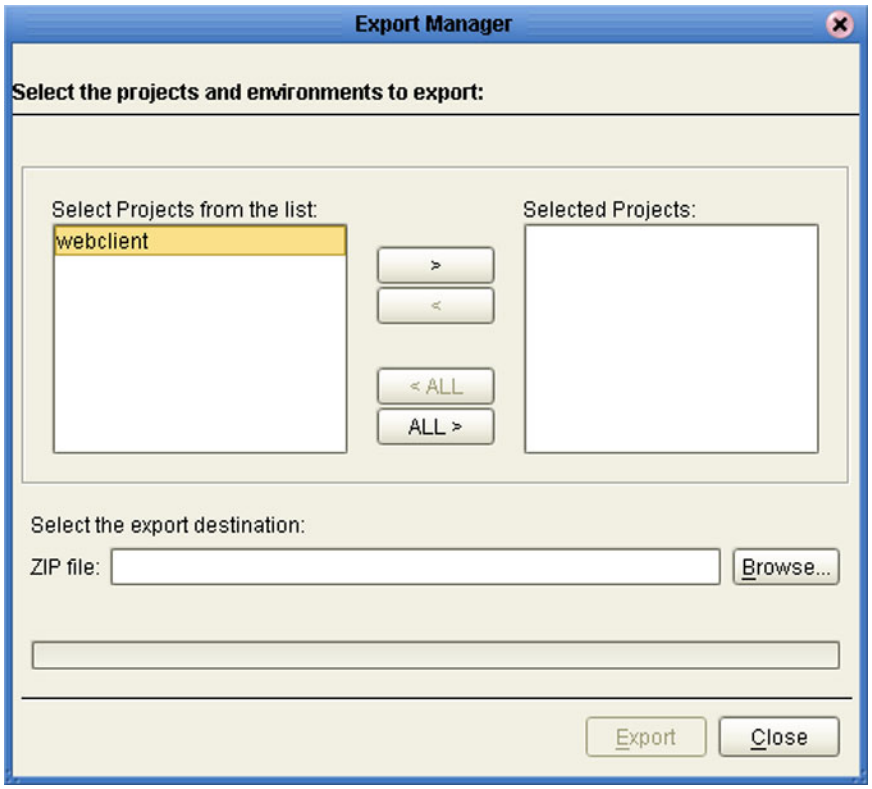
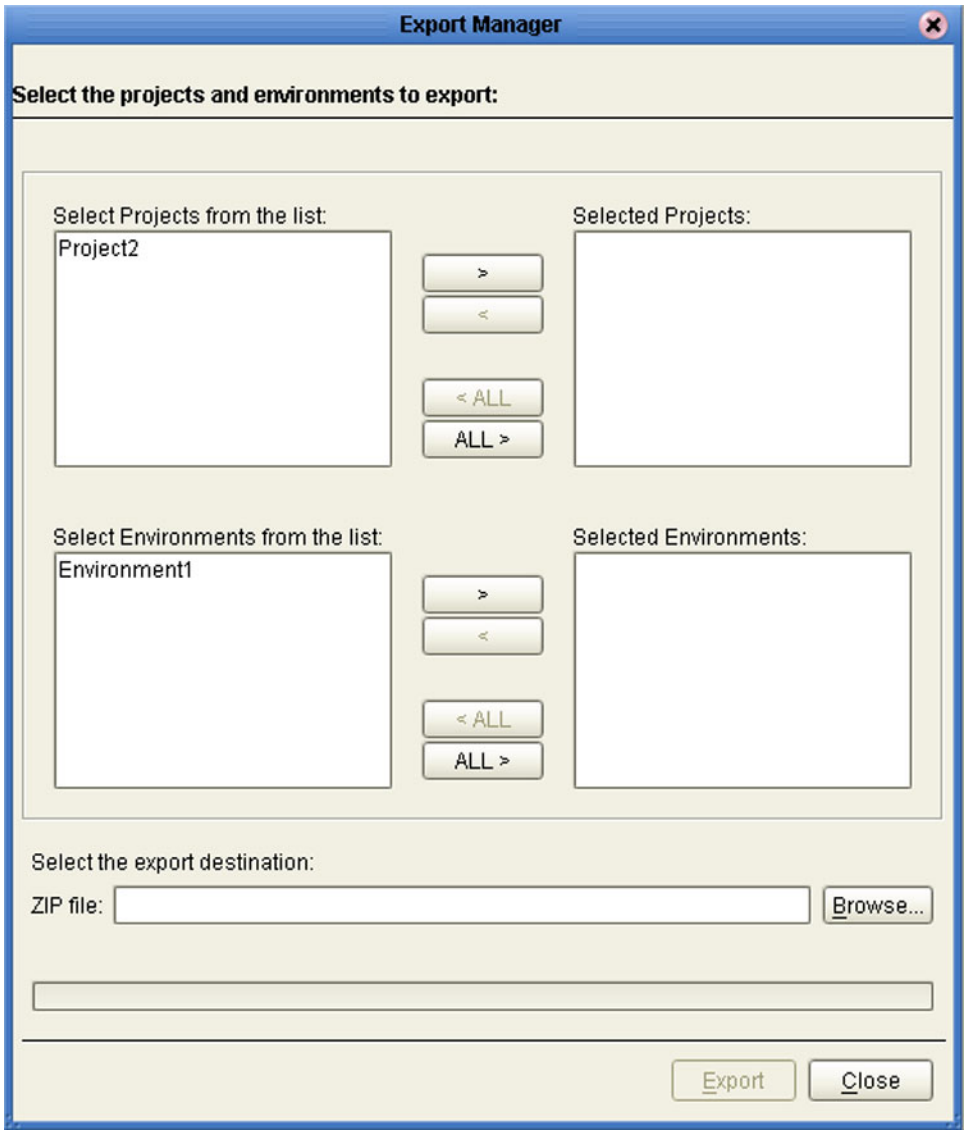
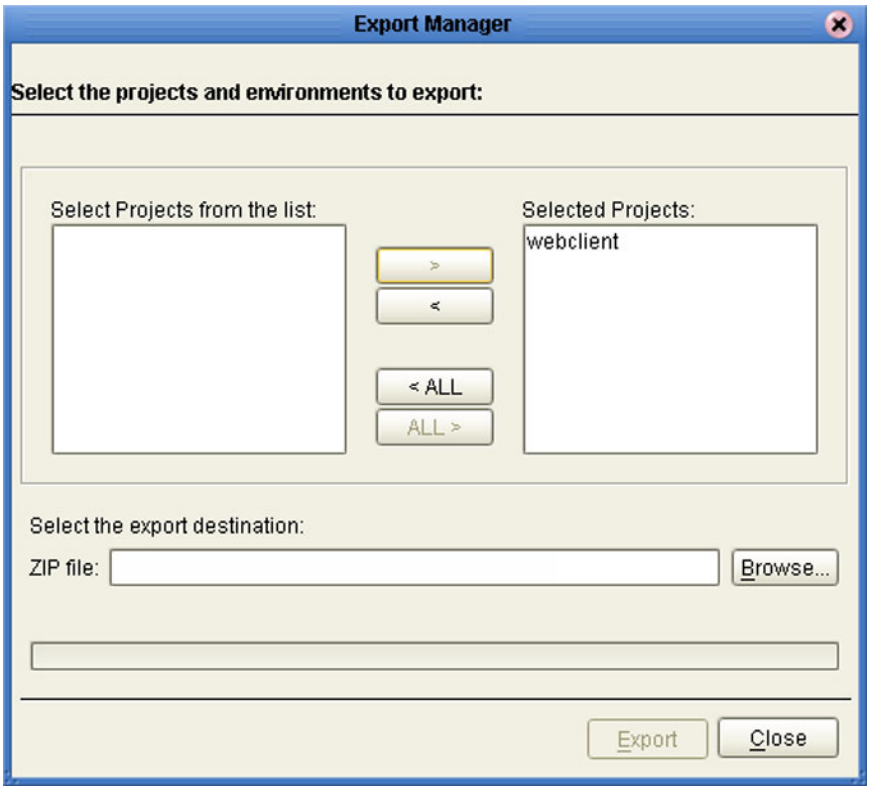


Figure 36 Export Manager Dialog Box (1b)



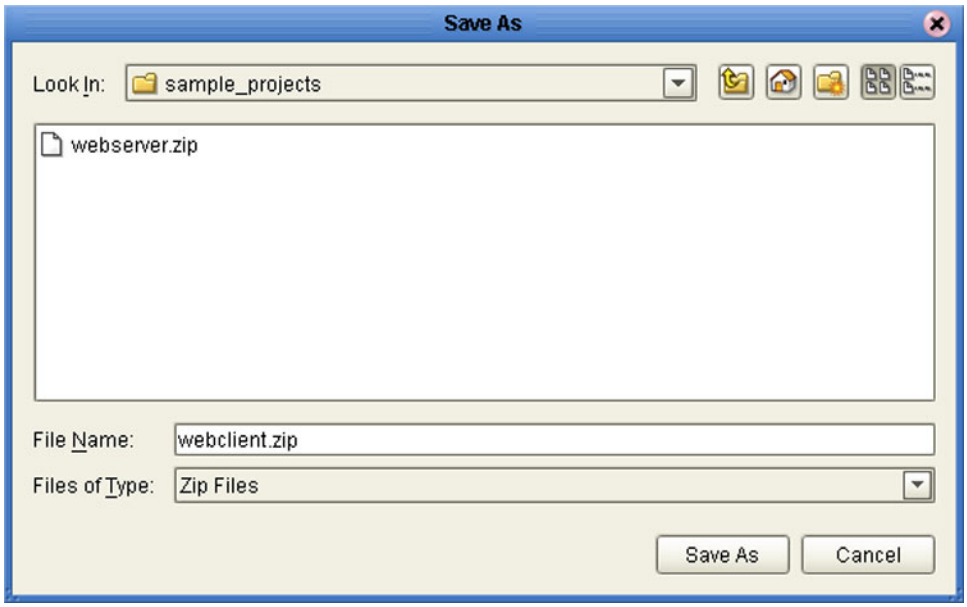
2 Highlight the desired Project(s) or Environment(s) in the displayed list, and transfer them to the *Selected Projects* or *Selected Environments* panel using the arrow buttons (see Figure 37).

Figure 37 Export Manager Dialog Box (2)



3 Click the **Browse** button to display the *Save As* dialog box, as shown in Figure 38.

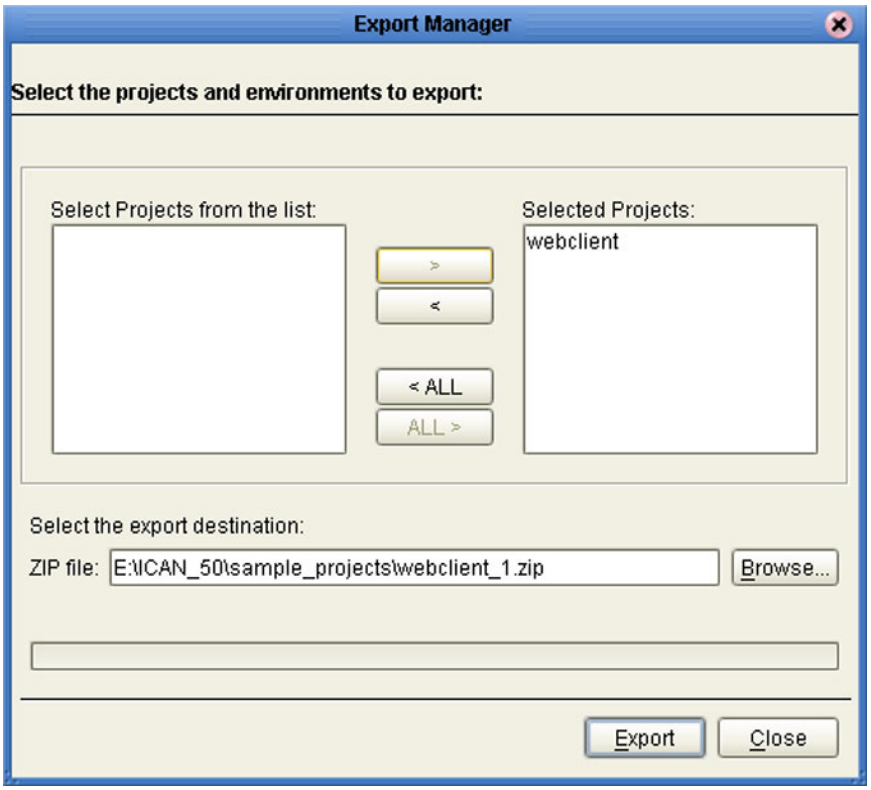
Figure 38 Save As Dialog Box



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

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

Figure 39 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 40 appears after the file has been exported successfully.

Figure 40 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 export a Project using the export script

1. Open a command prompt and change directory to *ICAN-root*\repository\util.
2. Type (for example): **exportProject 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 41.
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 41 Impact Analyzer Dialog Box

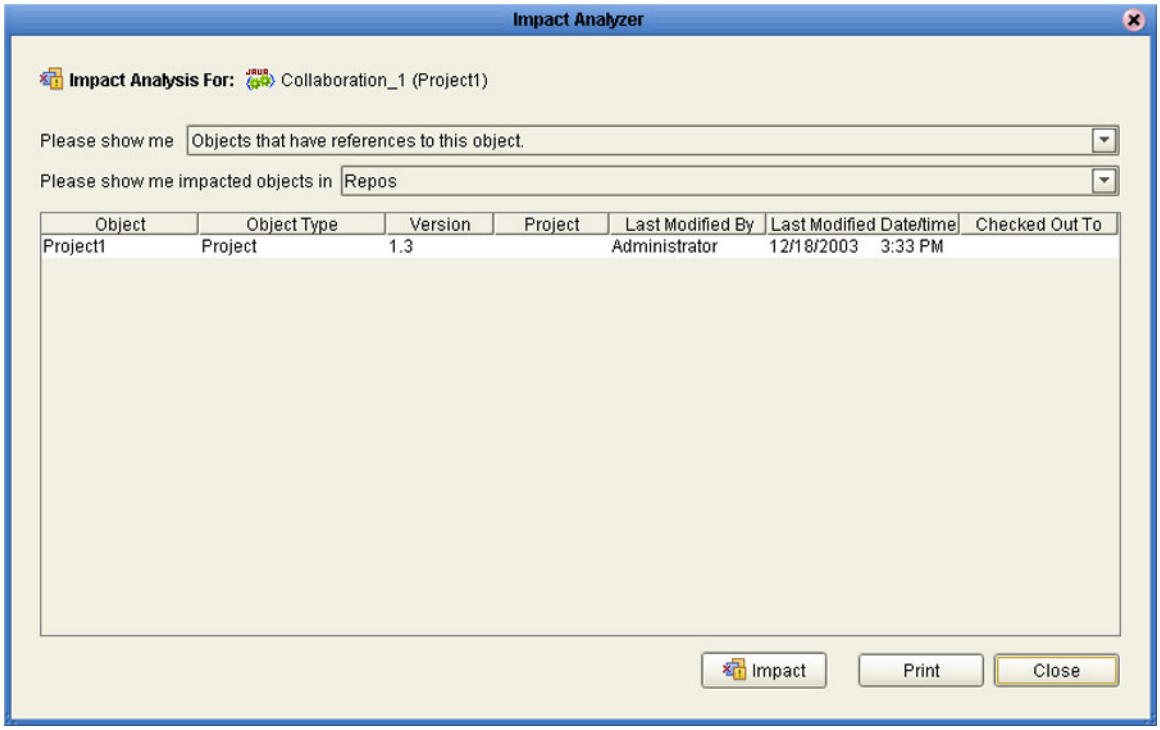
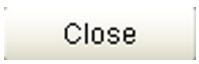


Table 14 Impact Analyzer Command Buttons

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.

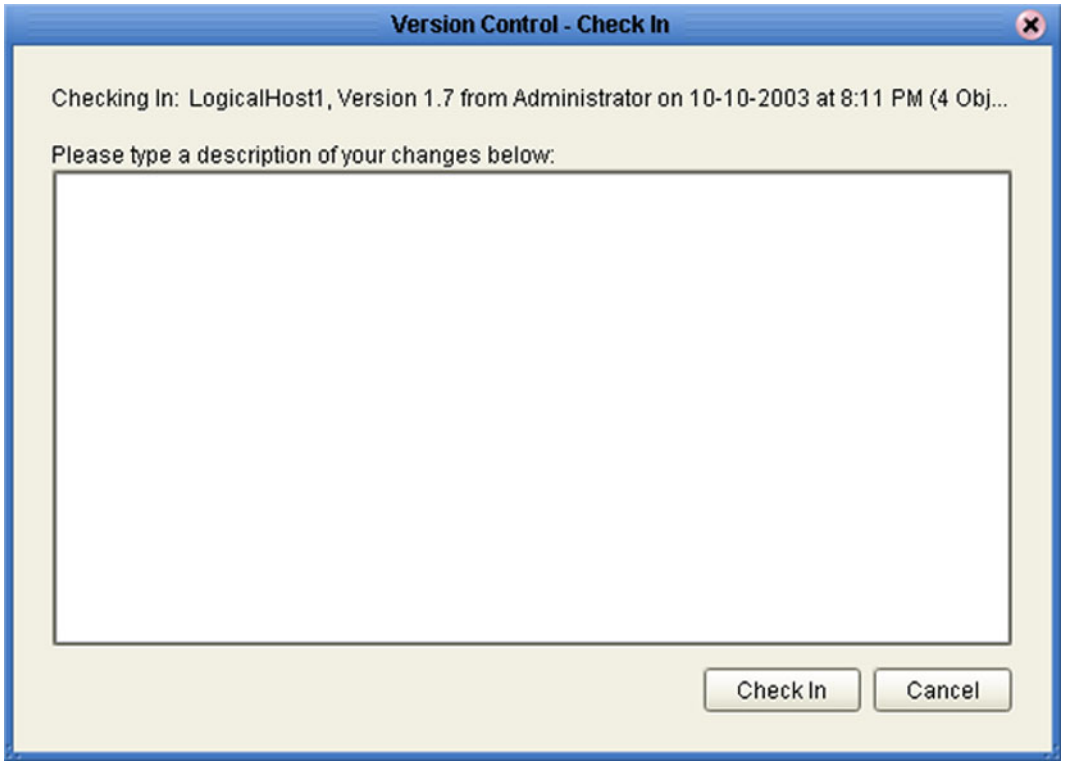
Checking a Component In

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 42.

Figure 42 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.

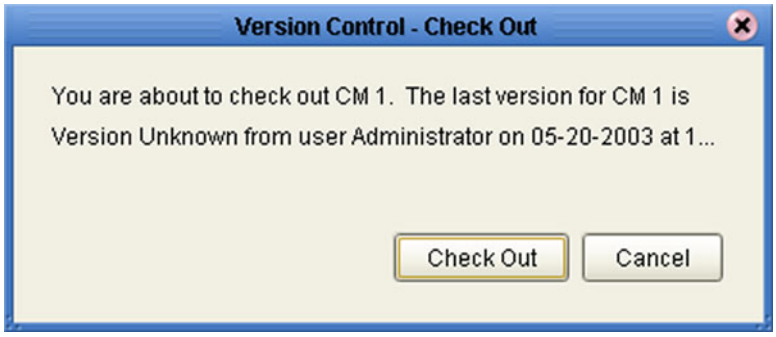
Checking a Component Out

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 43.

Figure 43 Version Control - Check Out Dialog Box



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

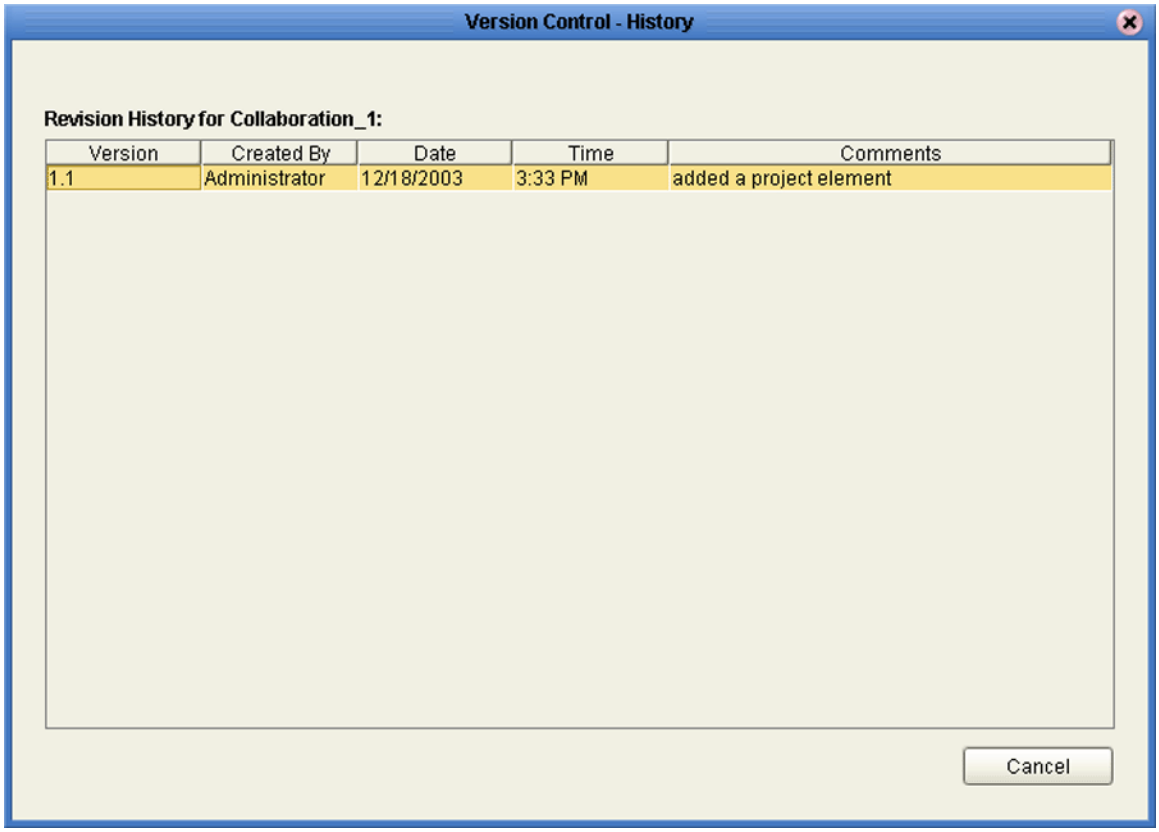
Note: *Only one user can have a file checked out for editing at a time. If another user attempts to check out the same file, they will receive a message indicating that the file is currently checked out.*

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 44.

Figure 44 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 104).

If you are also using eInsight Business Process Manager, 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.

Note: *See the eGate Integrator Tutorial for an end-to-end demonstration of the steps involved in setting up a 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 72
- **External Applications** on page 73
- **Schedulers** on page 73
- **Component Connections** on page 75
- **Message Destinations** on page 73

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. It is based on an Object Type Definition.

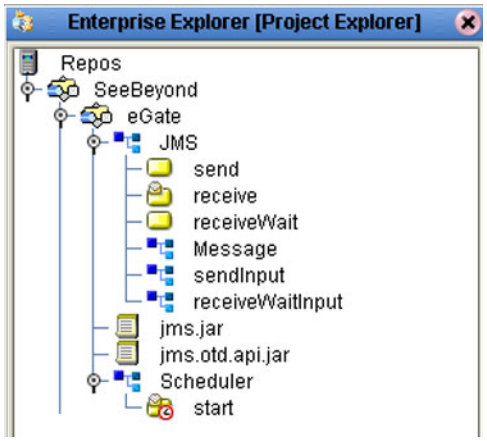
- **Object Type Definitions**

  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 45).

Figure 45 Project Explorer



The Project Explorer is used in conjunction with the Connectivity Map Editor (see **Using the Connectivity Map Editor** on page 70) 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 15 appear in the Project Explorer.

Table 15 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 67
- **Project Menu** on page 68
- **Connectivity Map Menu** on page 69

Repository Menu

Figure 46 Repository Menu

Table 16 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	Refreshes the Project Explorer and Environment Explorer to display the current contents of 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 47 Project Menu

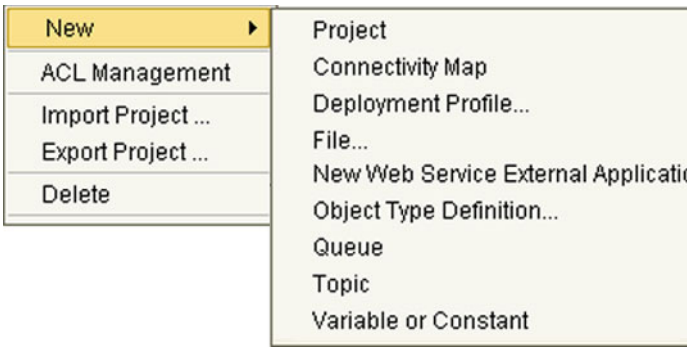


Table 17 Project Menu Options

Option	Option	Function
New	Project	Adds a Subproject folder to the selected Project.
	Business Process	If eInsight Business Process Manager is installed, displays the user interface for creating a new Business Process.
	Connectivity Map	Adds a Connectivity Map to the Project. See **Using the Connectivity Map Editor** on page 70.
	Deployment Profile	Displays a dialog box with which you can assign a Deployment Profile to the selected Project. See **The Deployment Editor** on page 132.
	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 146.
	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 81 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 the selected Project.

Table 17 Project Menu Options

Option	Option	Function
Delete		Deletes the selected Project.

Connectivity Map Menu

Figure 48 Connectivity Map Menu

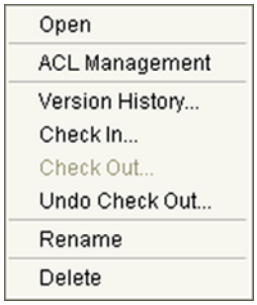


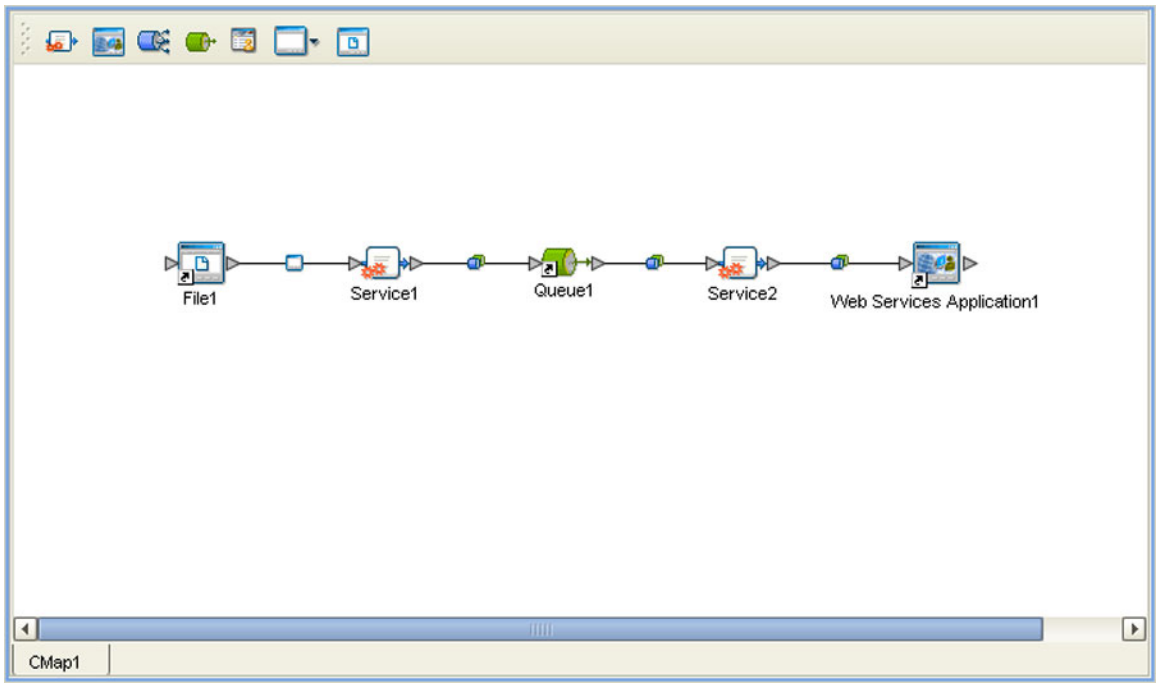
Table 18 Connectivity Map Menu Options

Command	Function
Open	Opens the **Connectivity Map Editor** for the selected Connectivity Map. See **Using the Connectivity Map Editor** on page 70.
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 63 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 61 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 62 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 49). 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.

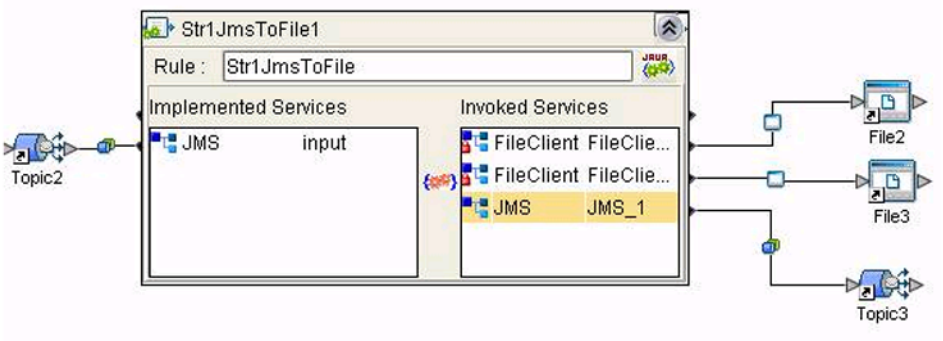
Figure 49 Connectivity Map Window



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 connects 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 50).

Figure 50 Linking JMS Topics



The Connectivity Map Editor toolbar contains the icons listed in Table 19, plus additional icons representing eGate add-ons and other ICAN components that you may have installed.

Table 19 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 72.
	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 146.
	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 73.
	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 73.

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

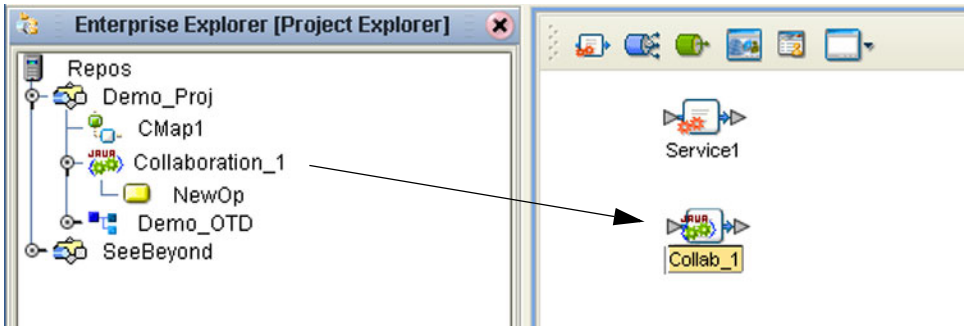
A service provides a framework for a process or a Collaboration, which contains the information required to execute a set of business rules.

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.

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 75. Dragging a Collaboration from the Project Explorer to the Service icon in the Connectivity Map defines the service as a Collaboration (see Figure 51).

Figure 51 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 52.

Figure 52 External Application Drop-Down Menu

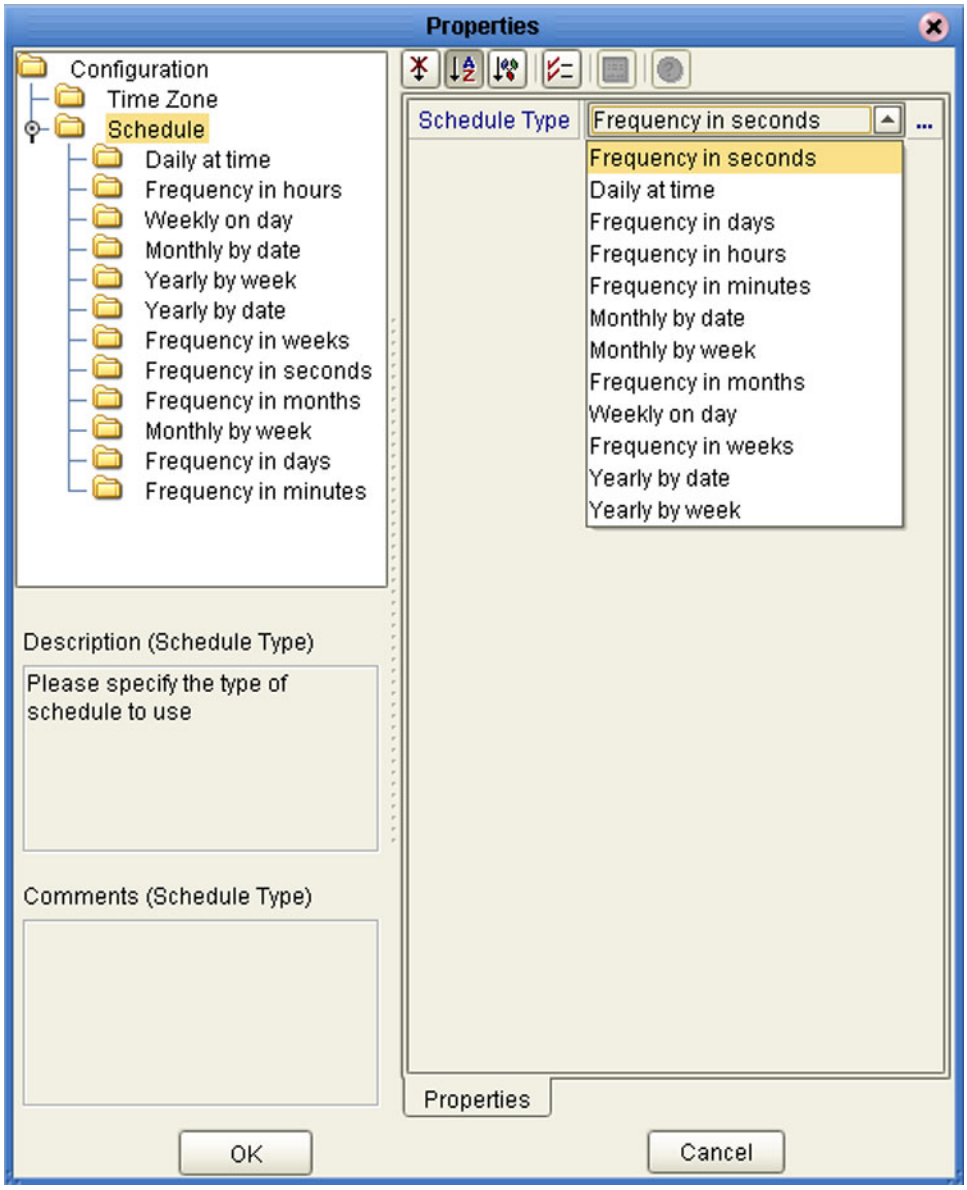
Selecting the check box beside an individual external application adds that icon to the toolbar; clearing the check box removes it from the toolbar.

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 53).

Figure 53 Scheduler Properties Dialog Box



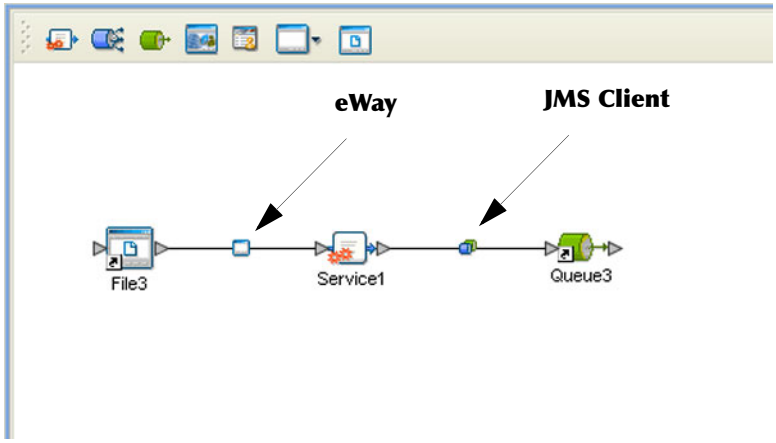
Selecting **Schedule** displays the **Schedule Type** property field which you set to the type of schedule you want to use. Selecting the corresponding node in the explorer tree displays the property field for that schedule type, in which you specify the desired value. The text in the *Description* box 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 54).

Figure 54 Connection Icons in a Connectivity Map

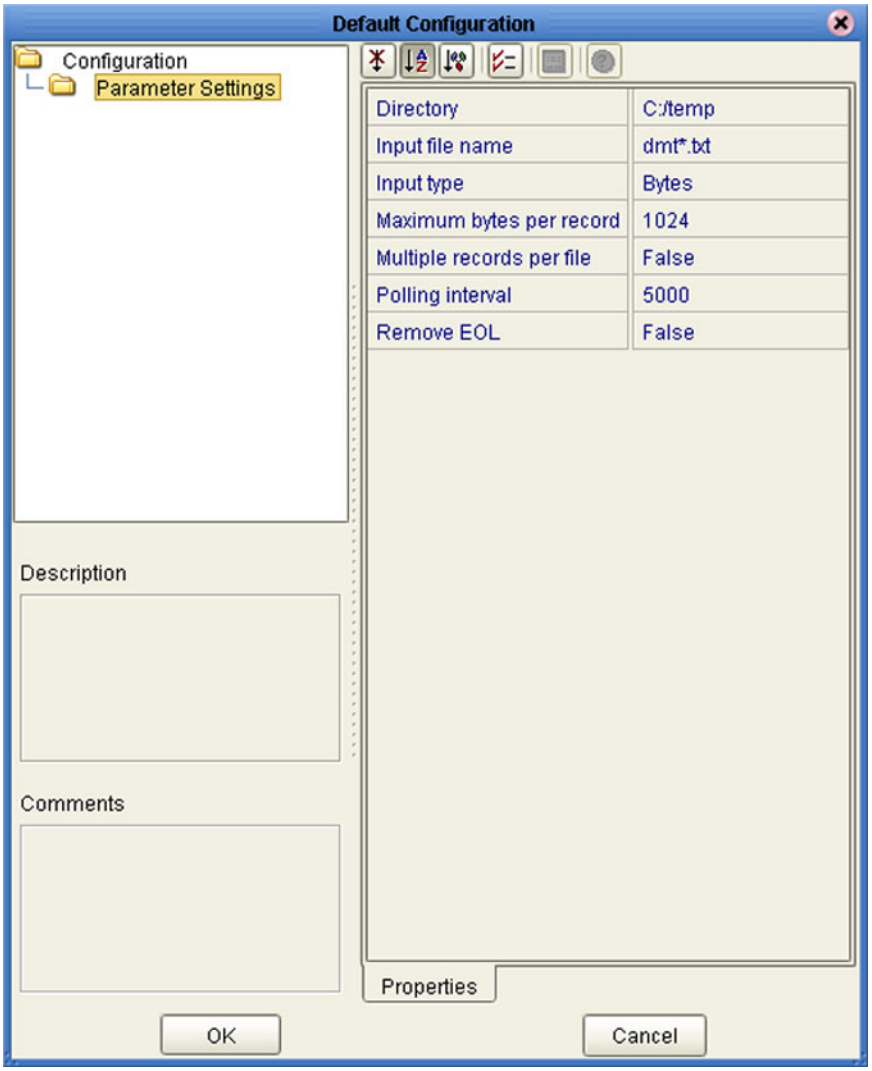


- When you link an external application with a Collaboration, the Enterprise Designer automatically adds an eWay Connection icon to the link. The eWay enables communication and movement of data between the external application and the eGate system. The eWay configuration specifies the logical connection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

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

Figure 55 Default Configuration Dialog Box



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

The constituent parts of the Default Configuration dialog box are:

- The **Configuration Tree** includes folders that contain configuration and connection properties for the selected eWay or message destination.
- The **Toolbar** contains a series of buttons used to sort and modify the information listed in the Properties folder, as described in Table 20.
- The **Description** box contains a brief description of the contents of the item currently selected in the Configuration Tree.
- The **Comments** box is for user comments about the item selected in the Configuration Tree.

Table 20 Configuration Dialog Box Toolbar Buttons

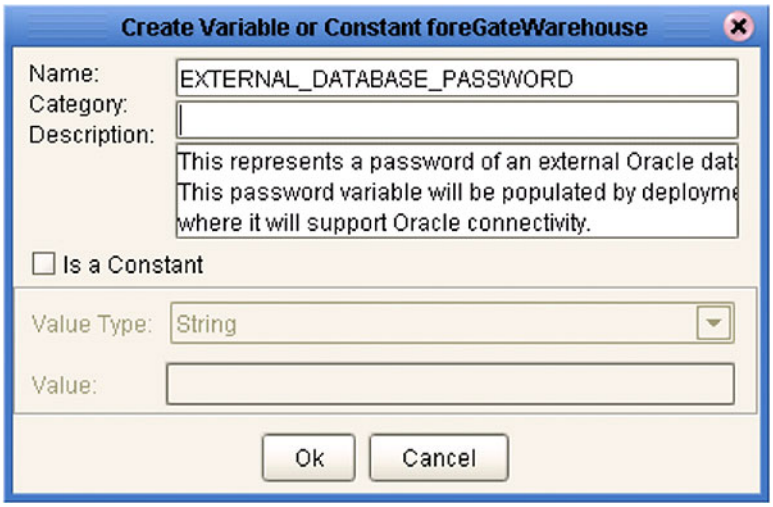
Button	Command	Function
	Unsorted	Displays configuration properties in their default order.
	Sort by Name	Sorts configuration properties alphabetically by name.
	Sort by Type	Displays configuration properties by property type.
	Show Editable Properties Only	Displays only the properties of an eWay or 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 274). Project variable values can be literals or Environmental constants.

For example, Figure 56 shows a project variable defined to represent a password of a database user in a target environment. System managers will assign an actual value to this variable in the deployment profile editor. The value of the assigned project variable—an Environment constant— is then used to connect the database in the target environment.

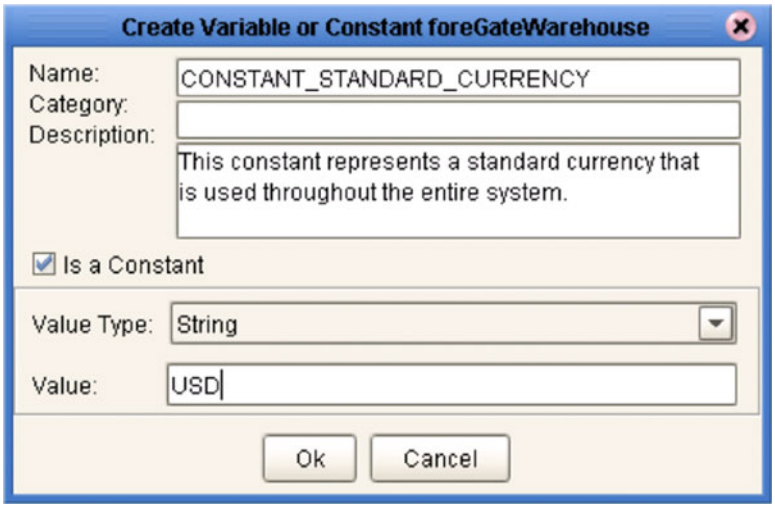
Figure 56 Project Variable Creation



Project constants are name/value pairs that are visible across the Project. For example, Figure 57 shows a standard currency defined to be used globally throughout the system.

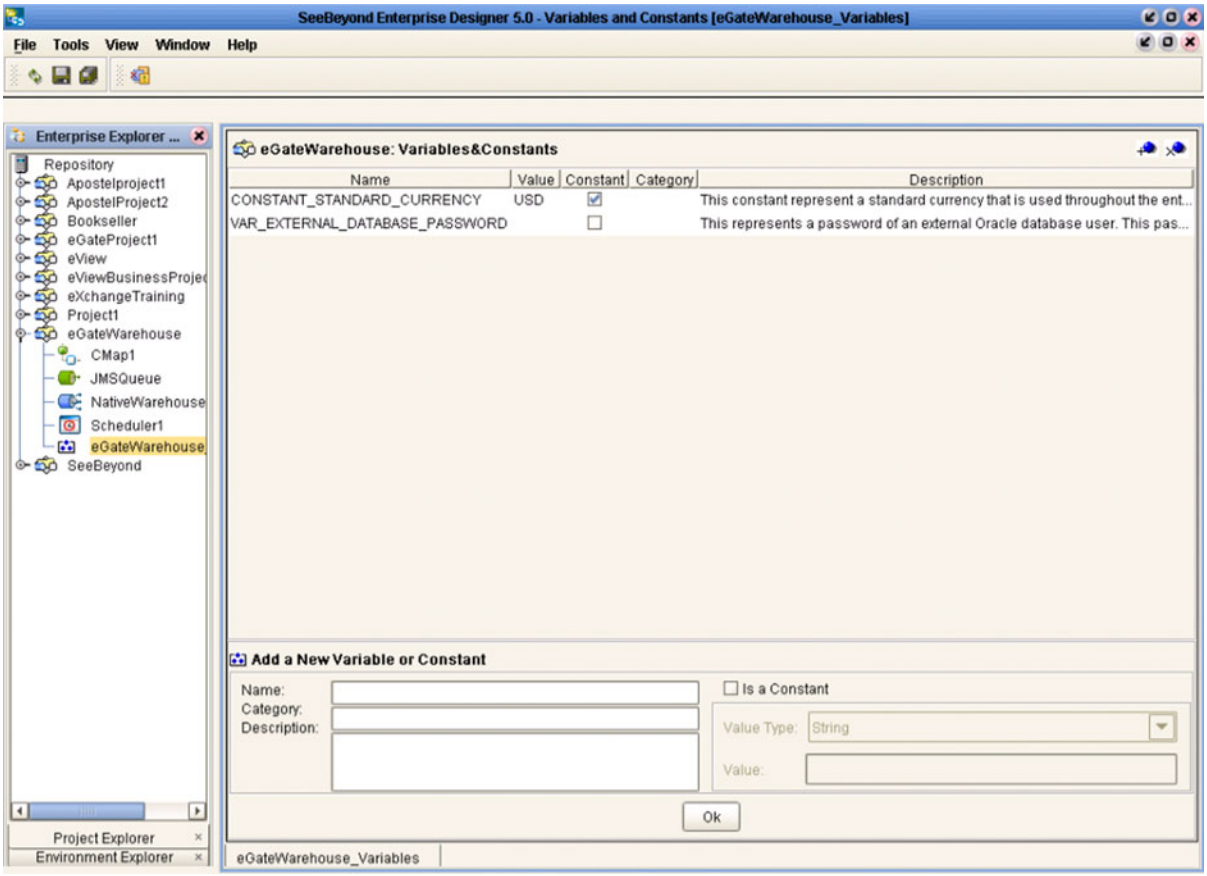
Note: *When you create an Project constant, you assign a permanent value to it—which cannot be overridden.*

Figure 57 Project Constant Creation



Constants and variables are automatically added to a Variables and Constants object group within the Project (see Figure 58).

Figure 58 Variables and Constants Object Group



Chapter 6

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.

6.1 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 from BPEL using XPath expressions.

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.

6.1.1 OTD Types

Externally-Defined OTDs

Externally-defined OTDs are based on formats or standards external to eGate Integrator, such as Document Type Definition (DTD), Web Services Definition Language (WSDL), 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.

6.1.2 Building OTDs

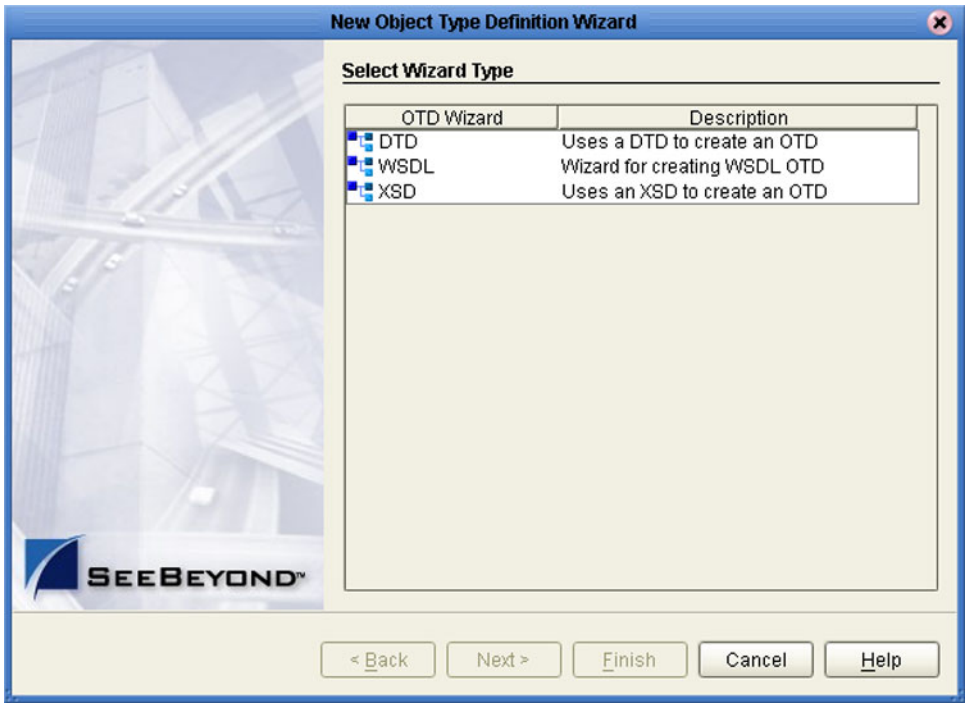
Wizards are provided in the Enterprise Designer GUI to guide you through the OTD building process. These Wizards call back-end builders that actually implement the building of the code, based on the provided information.

6.2 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 59. 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 83
- **Using the WSDL Wizard** on page 88
- **Using the XSD Wizard** on page 93

Figure 59 OTD Wizard Selection Dialog



Additional OTD Wizards are supplied with eGate add-on components, and are described in the User's Guides for the specific products. When these products are installed, the OTD Wizards are added to the list shown in Figure 59.

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 21 for your reference.

Table 21 OTD Wizard Navigation Buttons

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

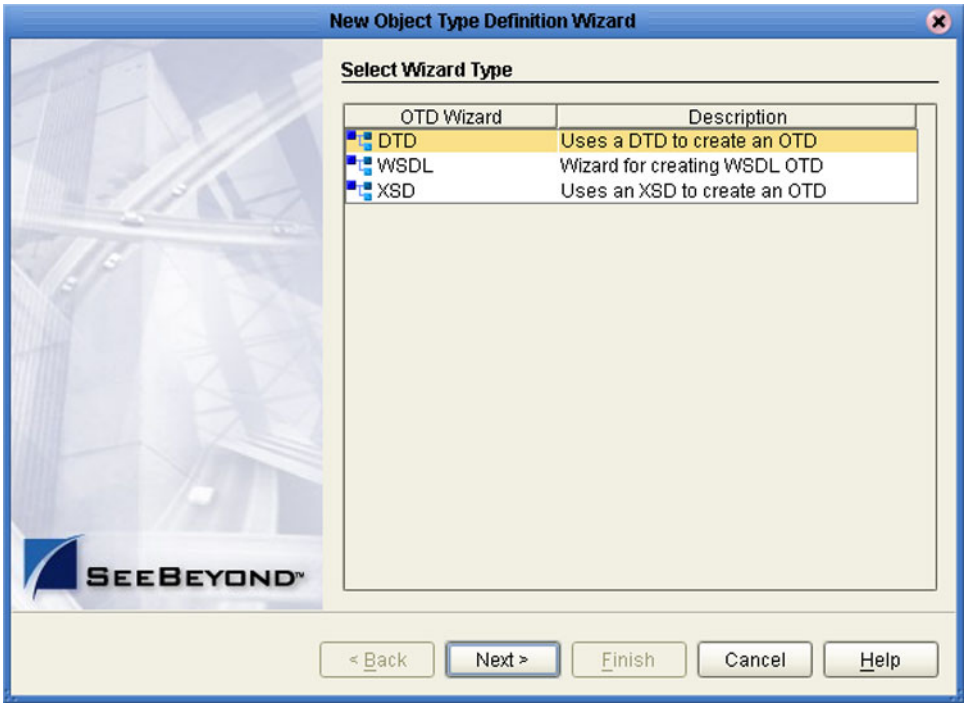
6.3 Externally-Defined OTDs

6.3.1 Using the DTD Wizard

To create an OTD file from a DTD file

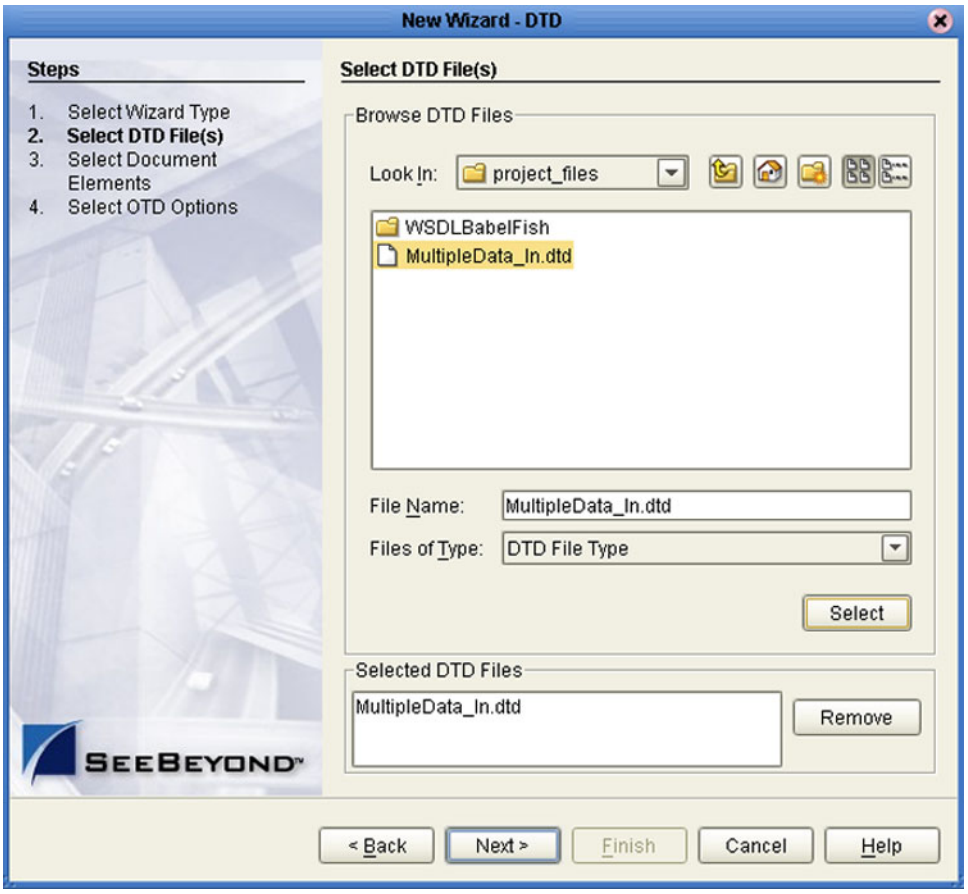
1 In the *Select Wizard Type* dialog, select **DTD** from the *OTD Wizard* list (see Figure 60) to create an OTD file from a Data Type Definition (DTD) file.

Figure 60 OTD Wizard Selection: DTD Wizard



2 Click **Next** to display the *Select DTD File(s)* dialog box, shown in Figure 61.

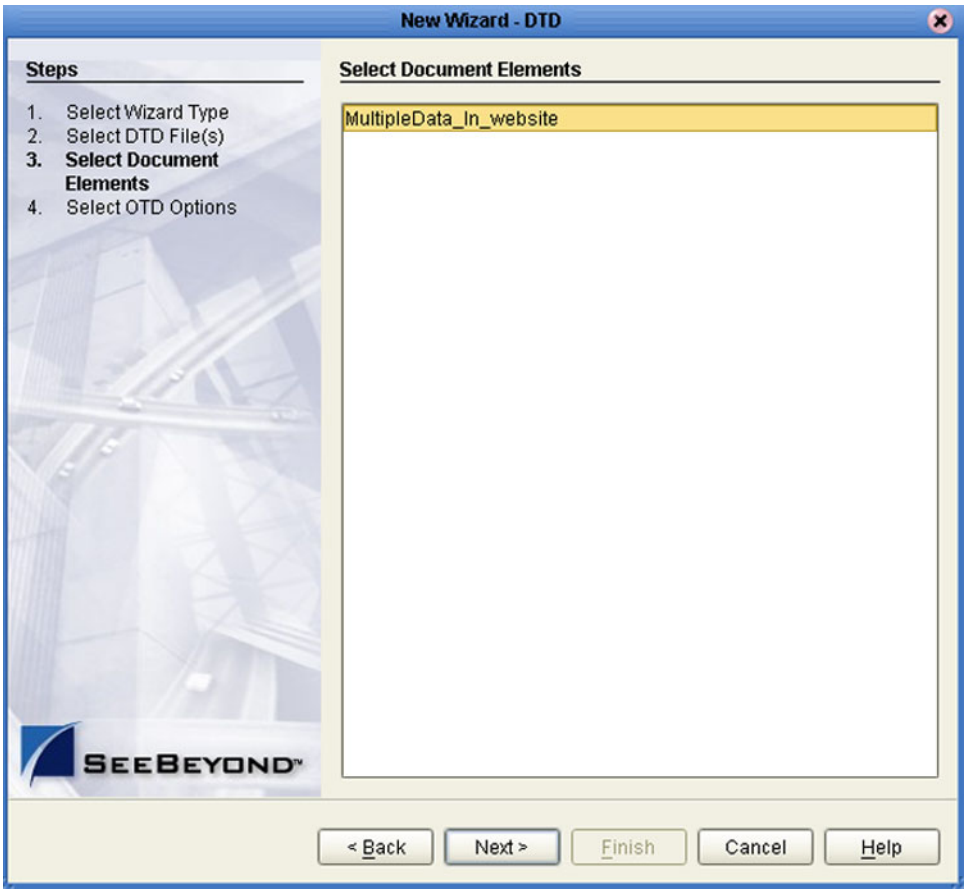
Figure 61 Select DTD File(s) Dialog Box



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 62.

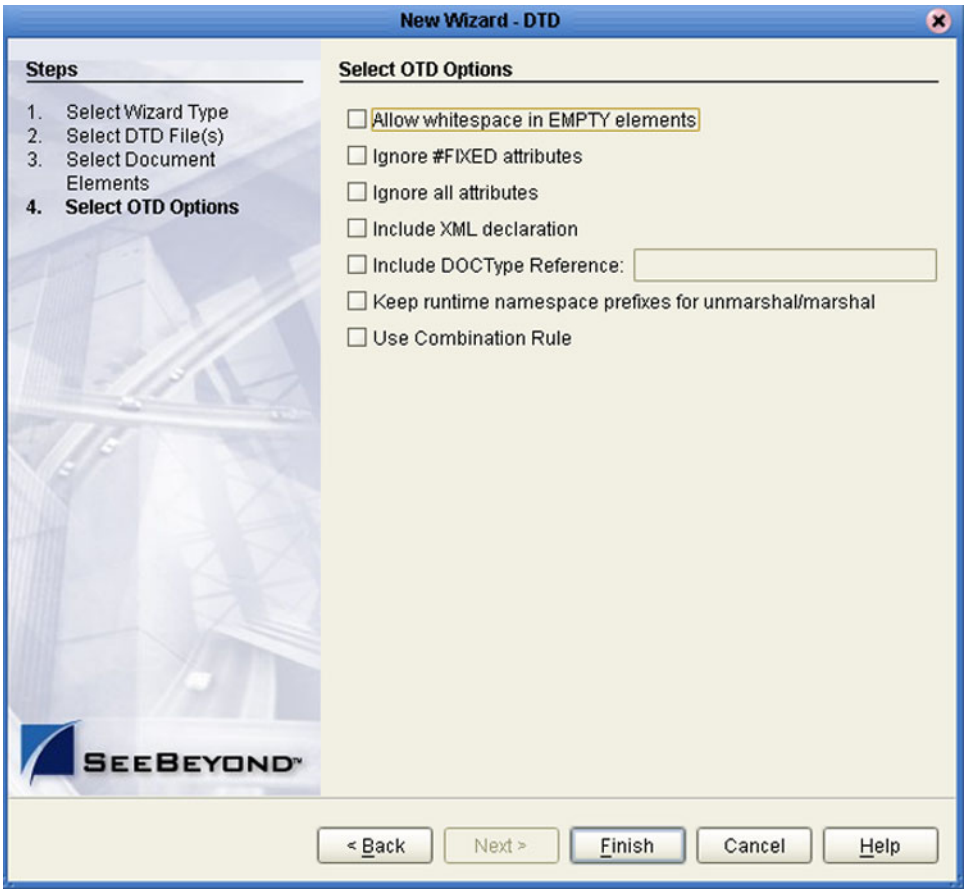
Figure 62 Select Document Elements Dialog Box



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 63.

Figure 63 Select OTD Options Dialog Box



7 Select the check boxes next to the OTD options you want to enable (see Table 22).

Table 22 DTD OTD Options

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

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

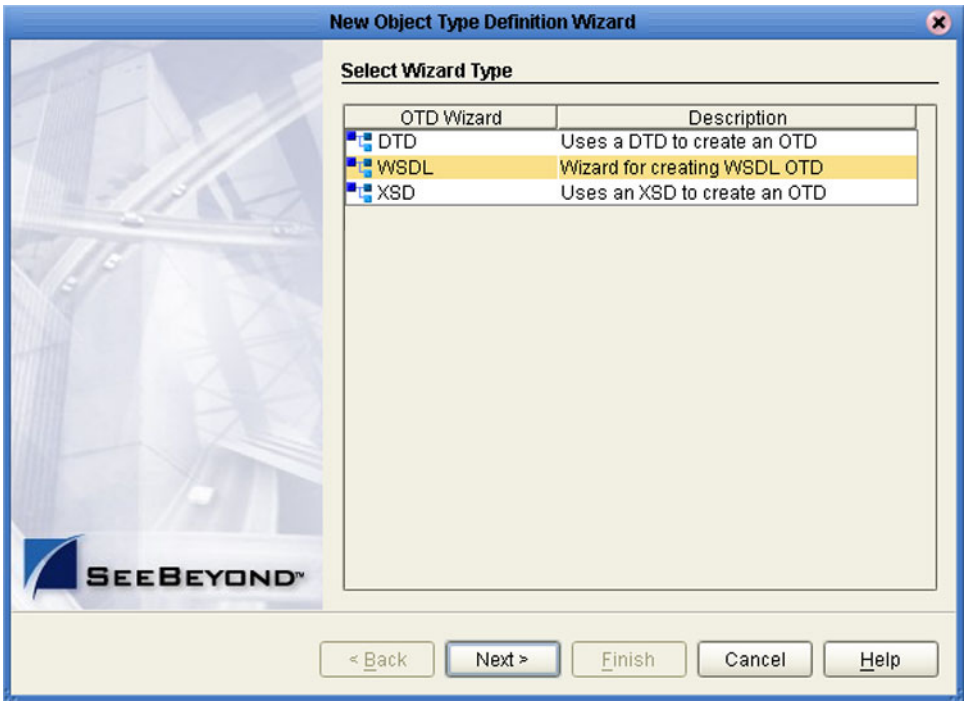
8 Click **Finish** to add the OTD to the Project.

6.3.2 Using the WSDL Wizard

To create an OTD file from a WSDL file

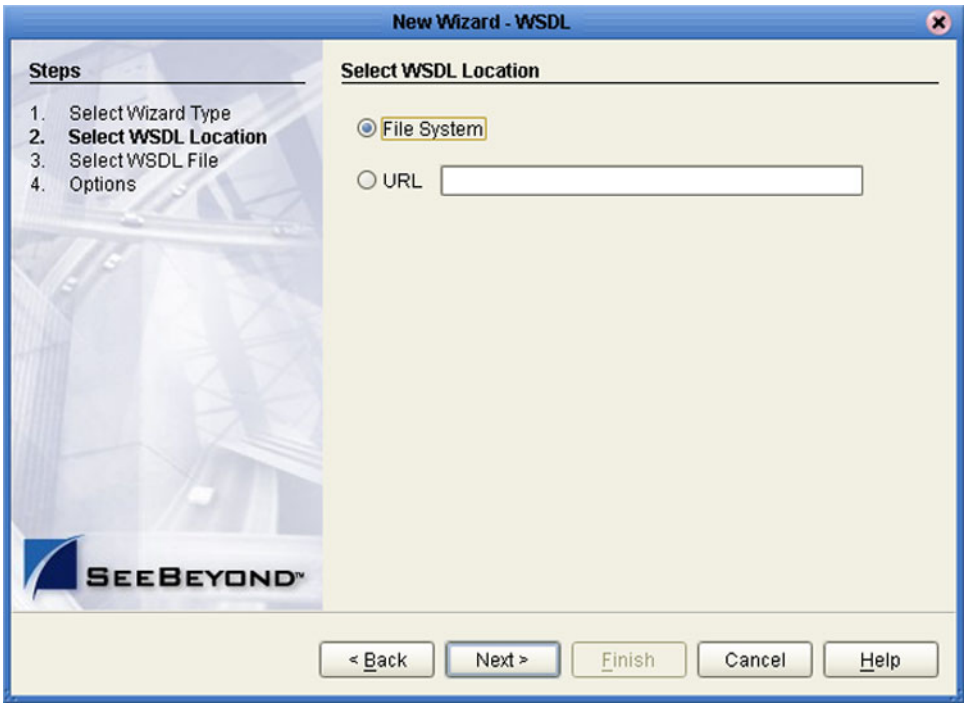
1 In the *Select Wizard Type* dialog, select **WSDL** from the *OTD Wizard* list (see Figure 64) to create an OTD file from an WSFL file.

Figure 64 OTD Wizard Selection: WSDL Wizard



2 Click **Next** to display the *Select WSDL File Location* dialog, shown in Figure 65

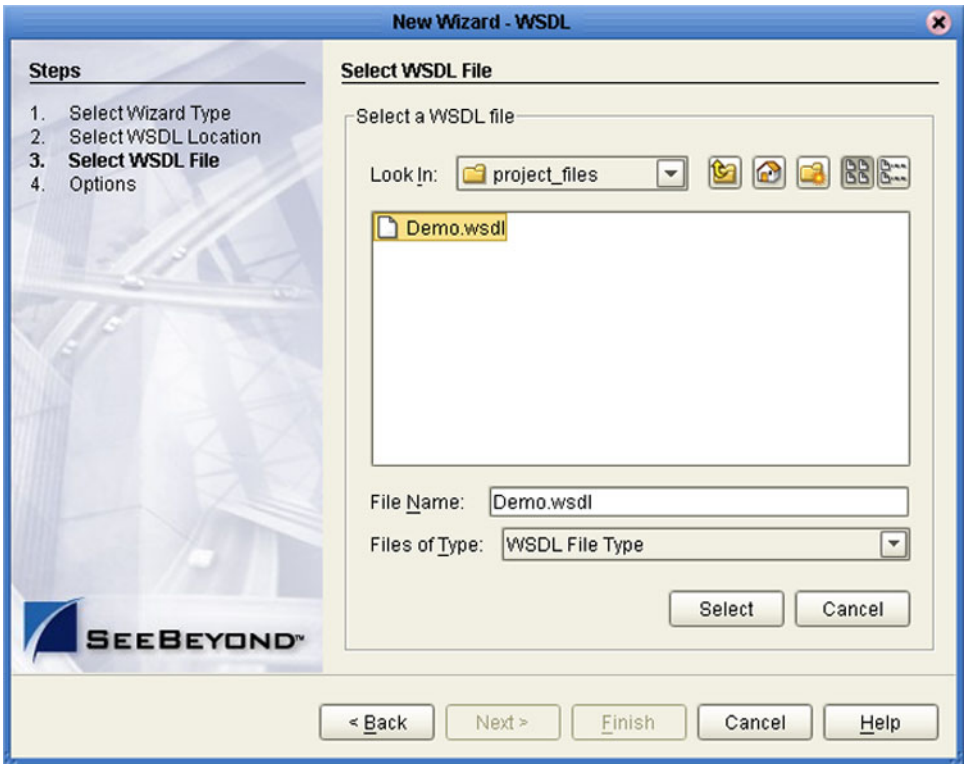
Figure 65 WSDL Wizard: Select WSDL Location



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 66.

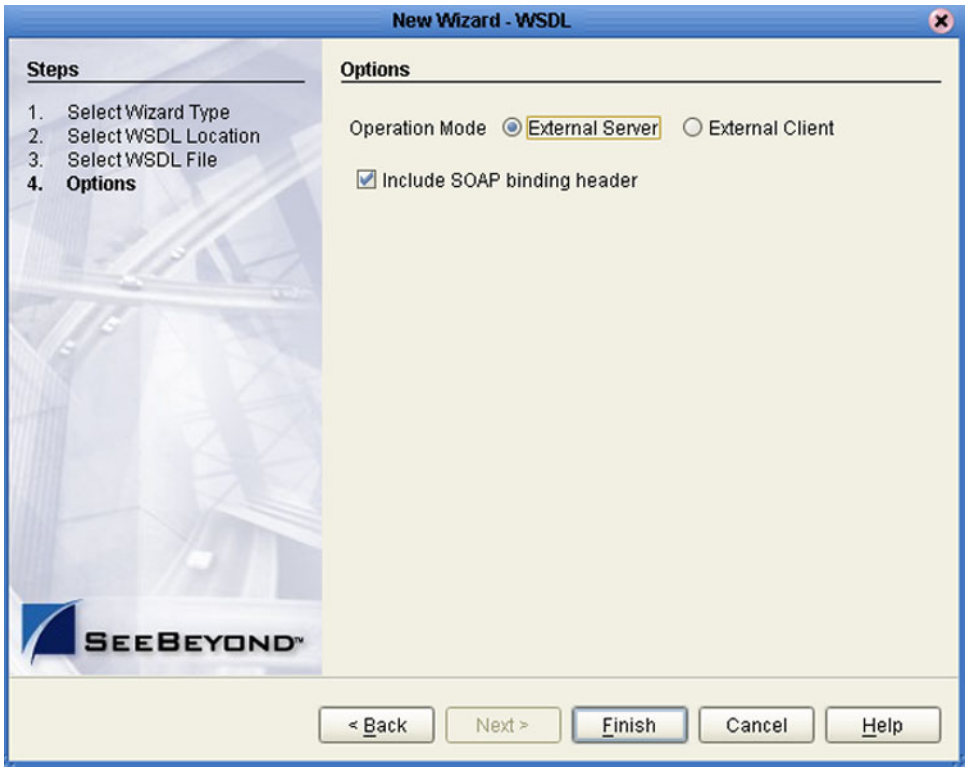
Figure 66 WSDL Wizard: Select WSDL File



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 67.

Figure 67 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:

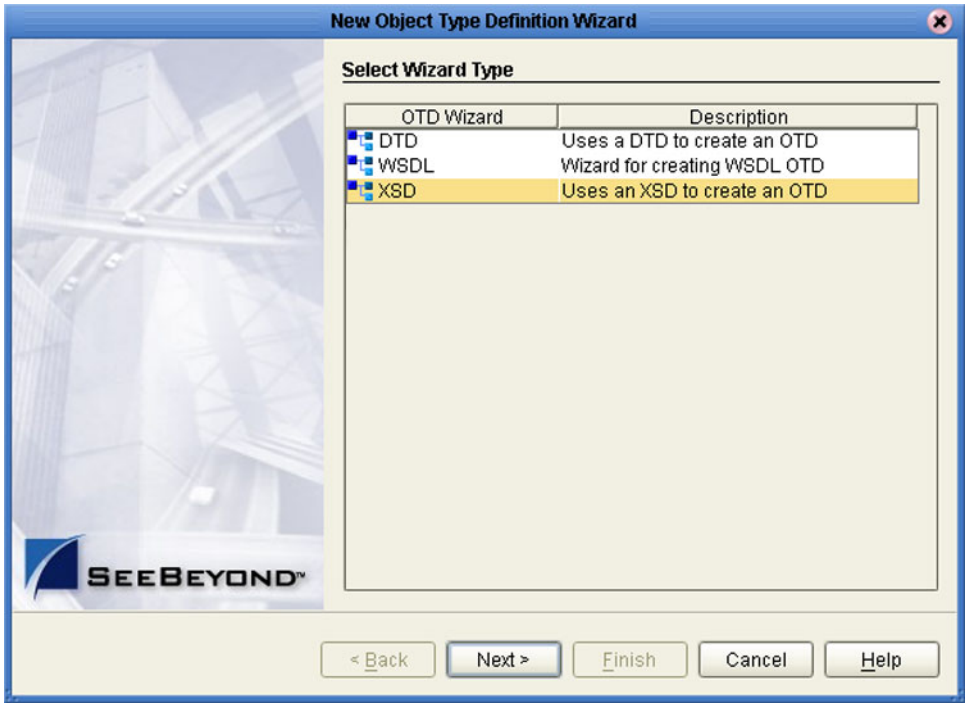
- **One-way:** The server receives a message from the client; also referred to as "fire and forget."
- **Request-response:** The server receives a message from the client and sends a correlated message back

6.3.3 Using the XSD Wizard

To create an OTD file from an XSD file

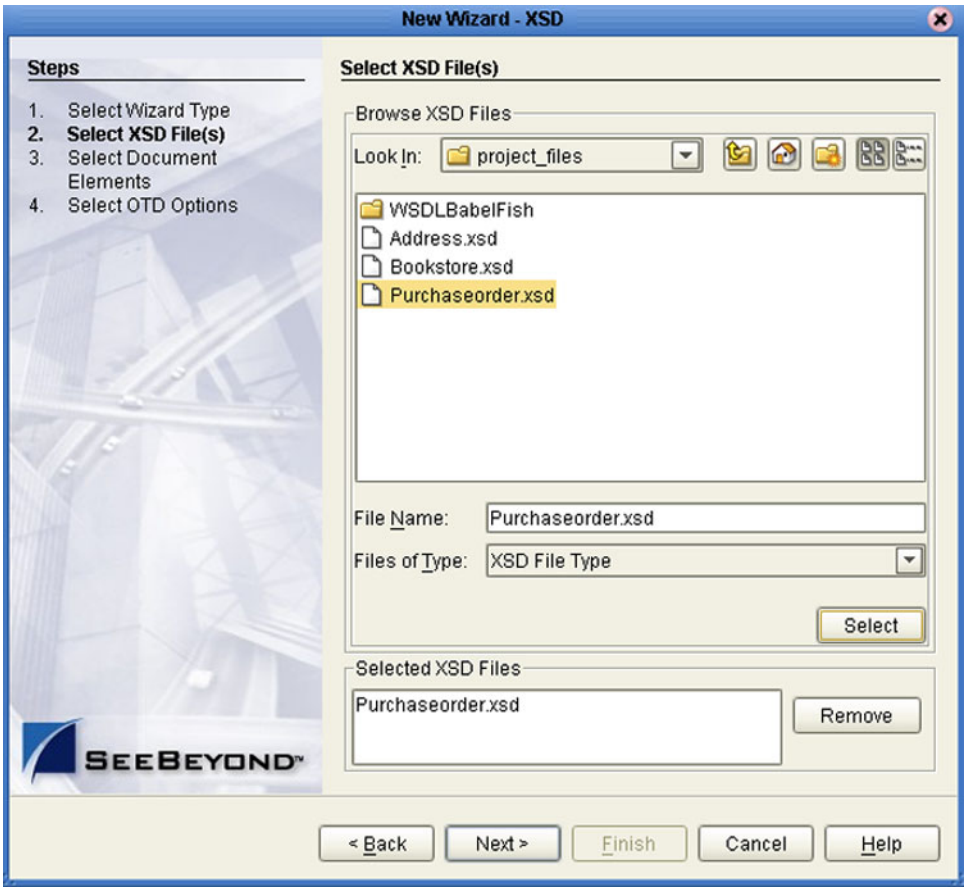
1 In the *Select Wizard Type* dialog, select **XSD** from the *OTD Wizard* list (see Figure 68) to create an OTD file from an XSD file.

Figure 68 OTD Wizard Selection: XSD Wizard



2 Click **Next** to display the Select XSD File(s) dialog box, shown in Figure 69.

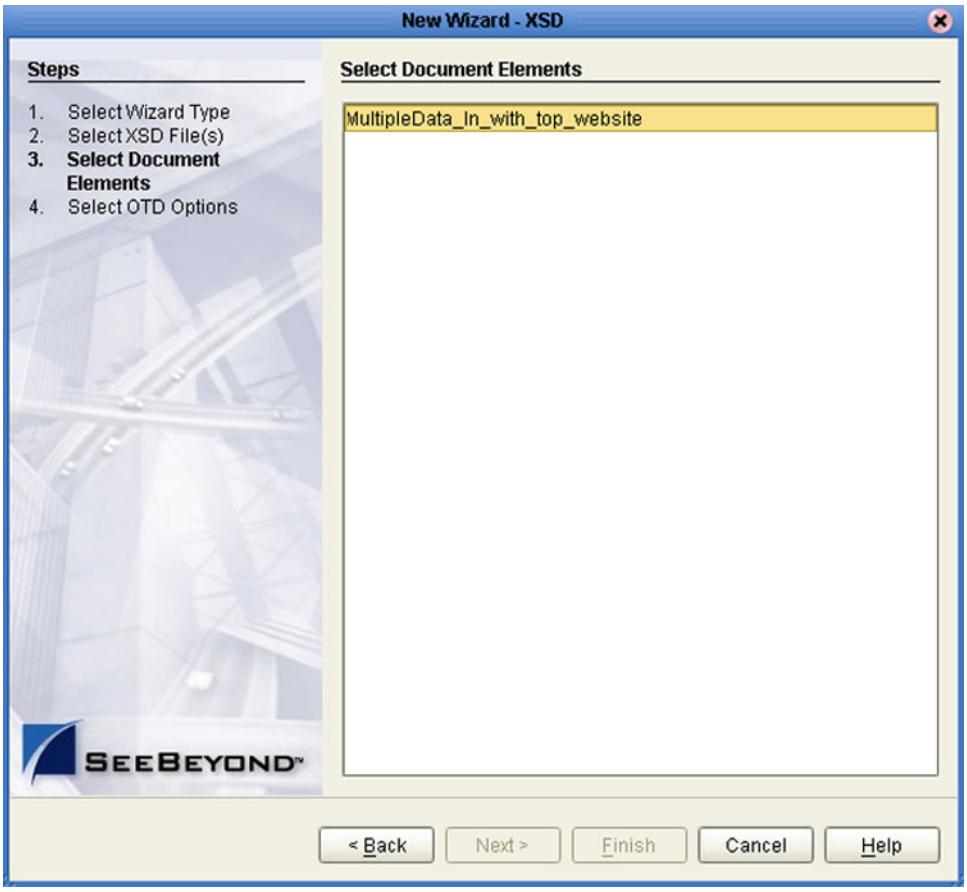
Figure 69 XSD Wizard: Select XSD File(s)



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 62.

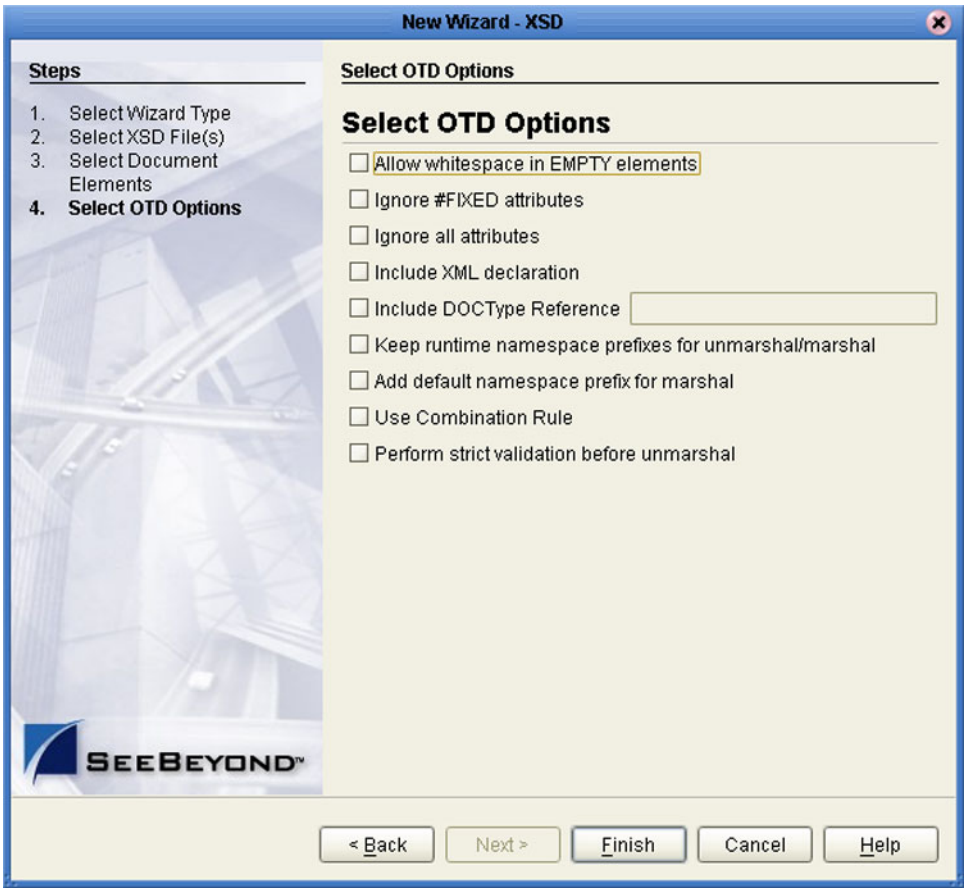
Figure 70 Select Document Elements Dialog Box



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 63.

Figure 71 Select OTD Options Dialog Box



7 Select the check boxes next to the OTD options you want to enable (see Table 23).

Table 23 XSD OTD Options

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

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

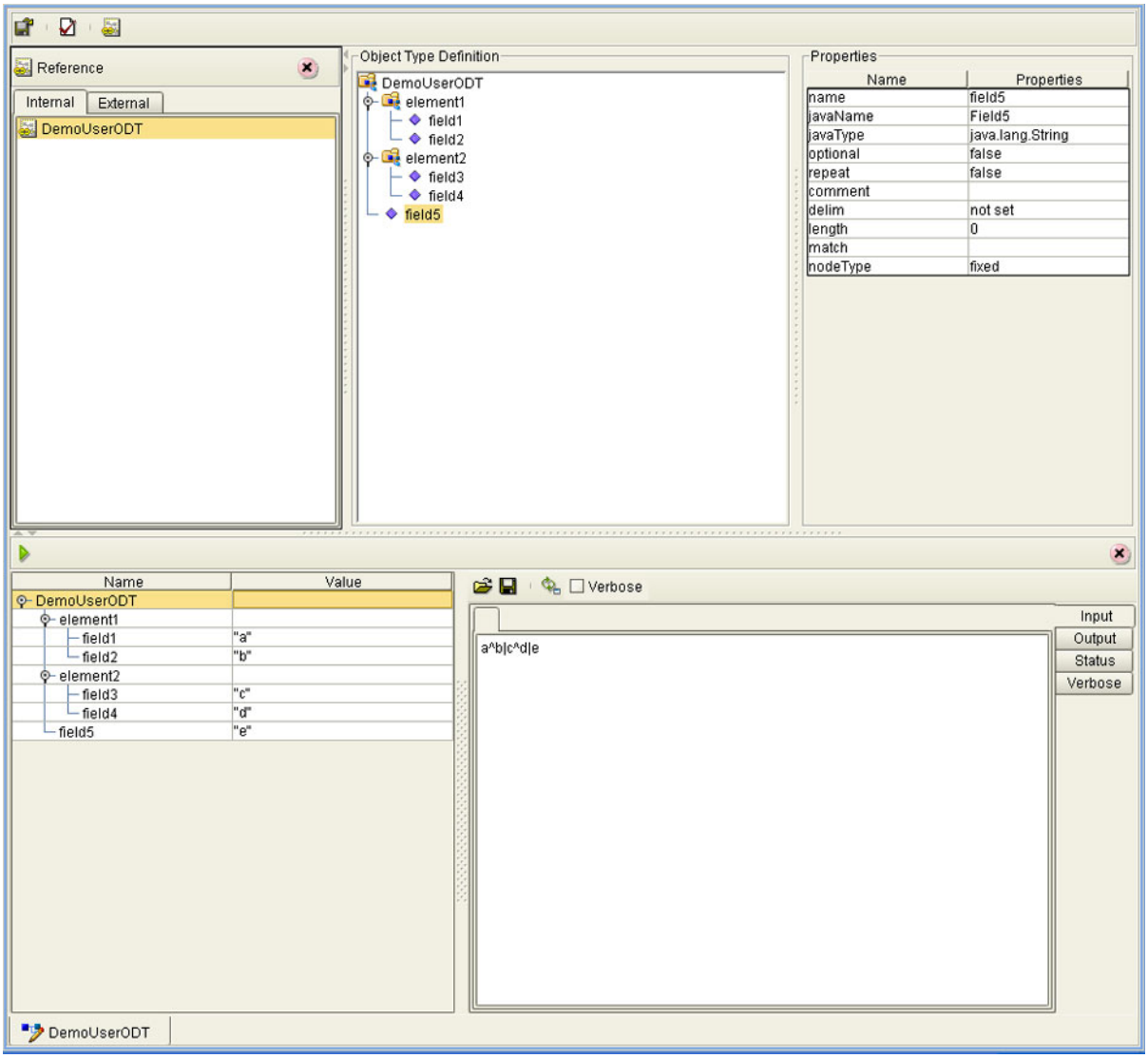
8 Click **Finish** to add the OTD to the Project.

6.4 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 72. 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 59**).*

Figure 72 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 24.

Table 24 OTD Editor Toolbar Icons

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

6.4.1 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.

6.5 Using the OTD Tester

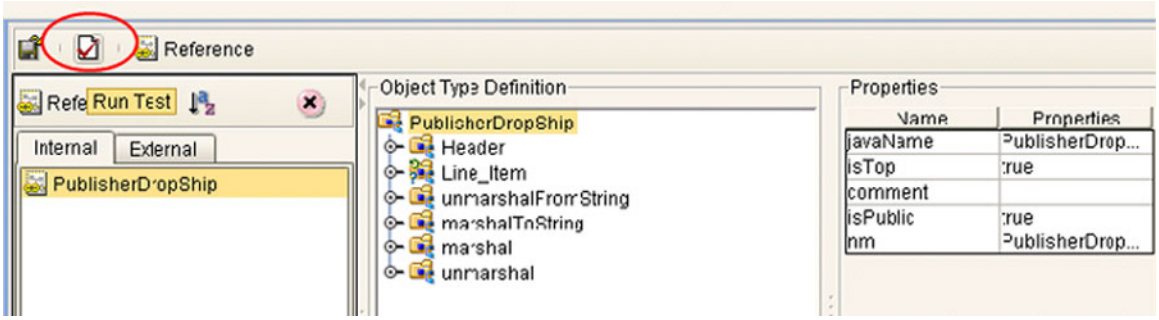
The OTD tester provides a facility to verify the correctness of OTDs, for example to:

- 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

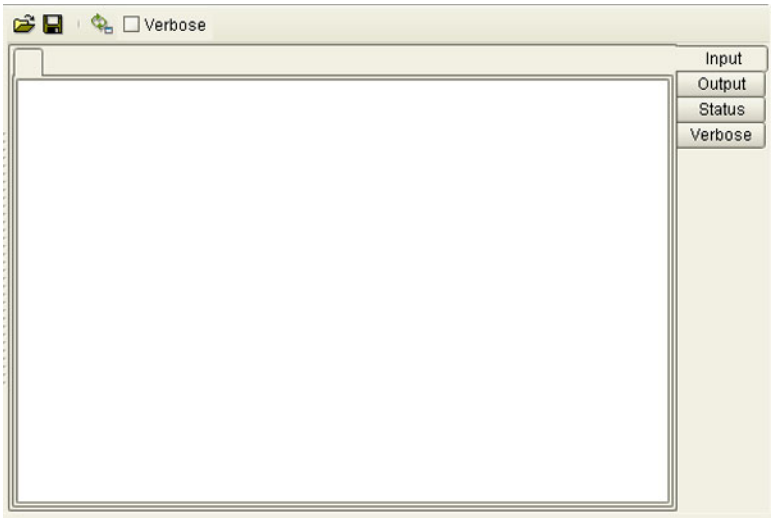
1. Open or create an OTD.
2. Click the **Tester** icon (see Figure 73).

Figure 73 OTD Tester



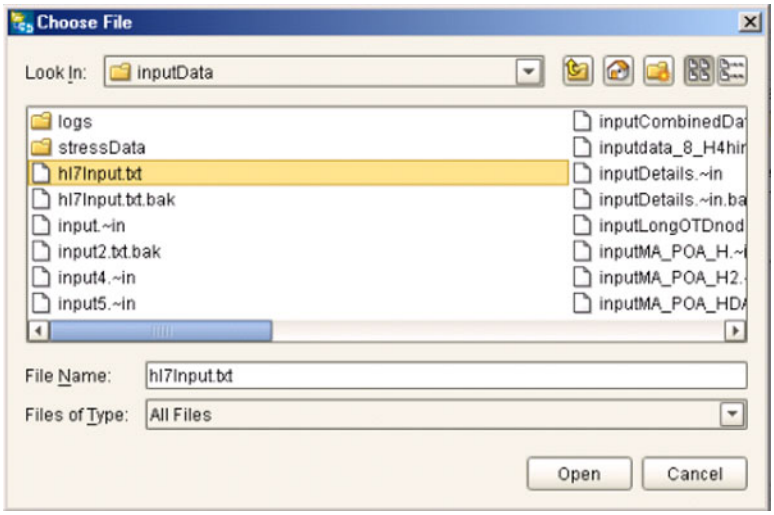
A test panel will appear below the OTD detail area of the editor. Note that there are four data display modes, selectable by tabs (see Figure 74). The Input tab is selected by default.

Figure 74 Test Panel Data Display



3 You can provide the input test data either by selecting a data file (see Figure 75), or by entering the data manually.

Figure 75 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 76).

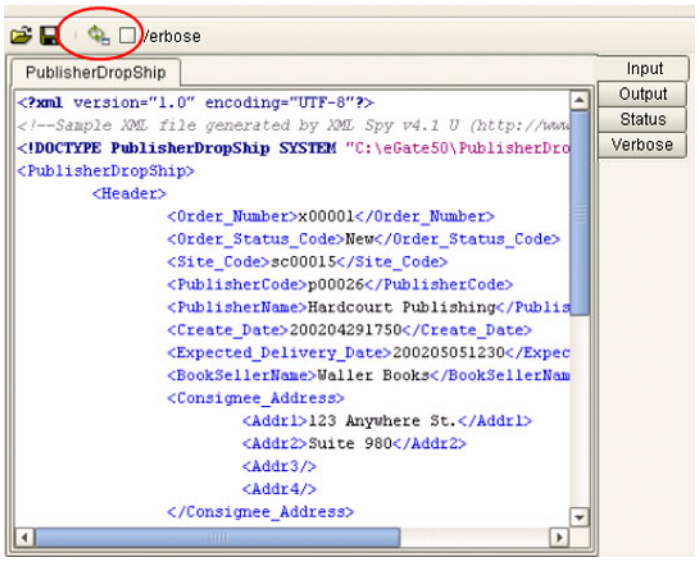
Figure 76 Object Elements and Values

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.

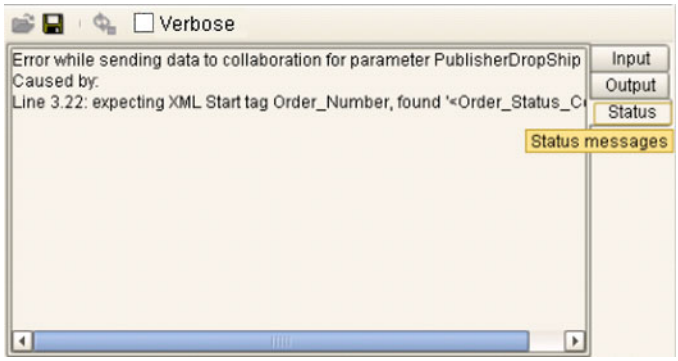
7 You can also change your test data in the Input data display, then re-test the OTD by clicking the **Refresh** icon (see Figure 77) to repopulate your OTD object elements with the new values.

Figure 77 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 78).

Figure 78 Status Data Display



Chapter 7

Environments

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

7.1 Overview

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

7.2 Environment Explorer

Figure 79 Enterprise Explorer: Environment Explorer View

7.2.1 Environment Explorer Icons

The icons described in Table 25 appear in the Environment Explorer.

Table 25 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.

7.2.2 Context Menus

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

- **Repository Menu** on page 106
- **Environment Menu** on page 107
- **Logical Host Menu** on page 108

Repository Menu

Figure 80 Repository Menu

Table 26 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	Refreshes the Project Explorer and Environment Explorer to display the current contents of the Repository. (Open editors are not refreshed.)

Environment Menu

Figure 81 Environment Menu

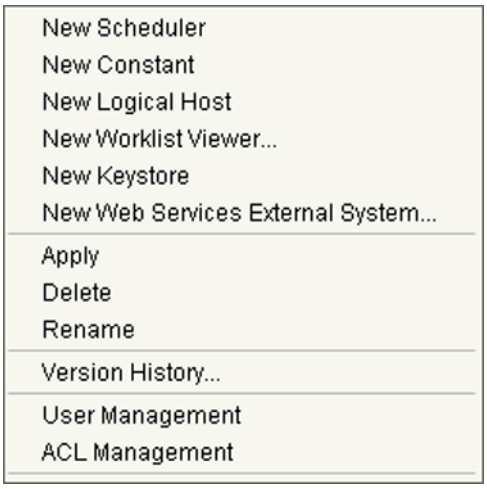


Table 27 Environment Menu Options

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 110.
New Logical Host	Adds a new Logical Host to the selected Environment.
New Worklist Viewer	This option is present only when eInsight Business Process Manager is installed. See the *eInsight Business Process Manager User's Guide* for information.
New Keystore	Adds a new keystore to the selected Environment.
New Web Service ...	Adds a third-party Web service application to the Project Explorer. See **SeeBeyond Web Services** on page 282.
Apply	Applies 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 71 for more information.
User Management	Displays a dialog box with which you can manage message server access. See the *eGate Integrator System Administration Guide*.

Table 27 Environment Menu Options

Option	Function
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*.

Logical Host Menu

Figure 82 Logical Host Menu

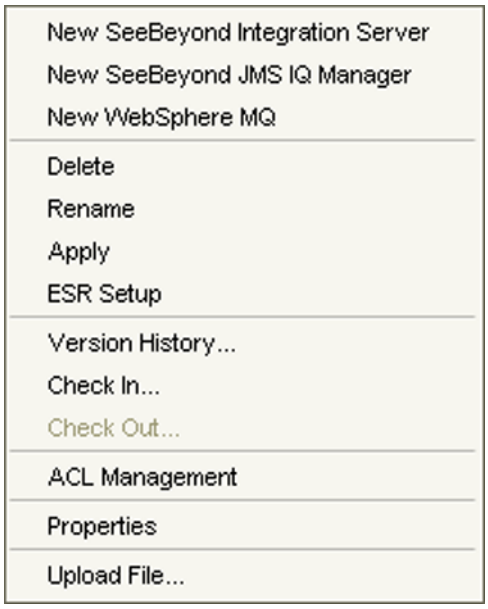


Table 28 Logical Host Menu Options

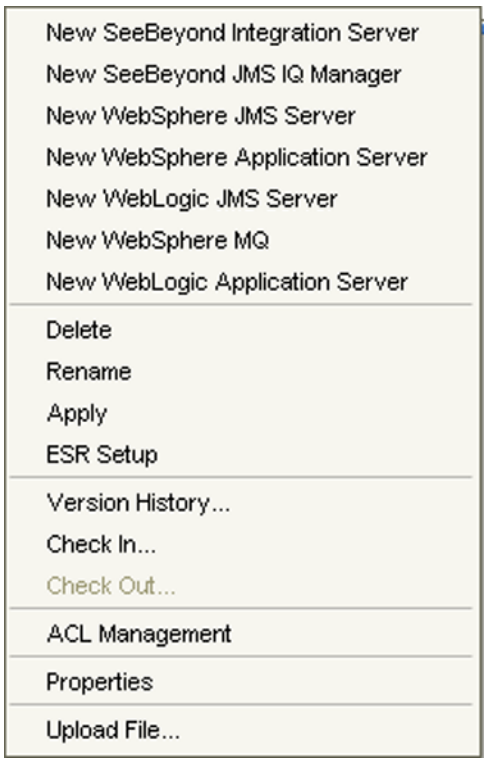
Option	Function
New SeeBeyond Integration Server	Adds a new SeeBeyond Integration Server to the selected Logical Host.
New SeeBeyond JMS IQ Manager	Adds a new SeeBeyond JMS IQ Manager to the selected Logical Host.
New WebSphere MQ	Adds a new IBM WebSphere MQ message server to the selected Logical Host.
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	Applies 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 Logical Host..

Table 28 Logical Host Menu Options

Option	Function
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 71 for more information.
Check In	Displays a dialog box, with which you can check in the current version of an Logical Host. Refer to **Checking a Component In** on page 69 for more details.
Check Out	Displays a dialog box with which you can check out a version of an Logical Host. See **Checking a Component Out** on page 70 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 Logical Host See the *eGate Integrator System Administration Guide*.
Properties	Displays a dialog box with which you can modify the default settings for the selected Logical Host.
Upload File	Allows you to upload third-party libraries (**.jar** files) to the Logical Host.

Note: *If you are using BEA WebLogic and/or IBM WebSphere, the Application Servers and JMS Message Servers for these products will also appear in the context menu (see Figure 83).*

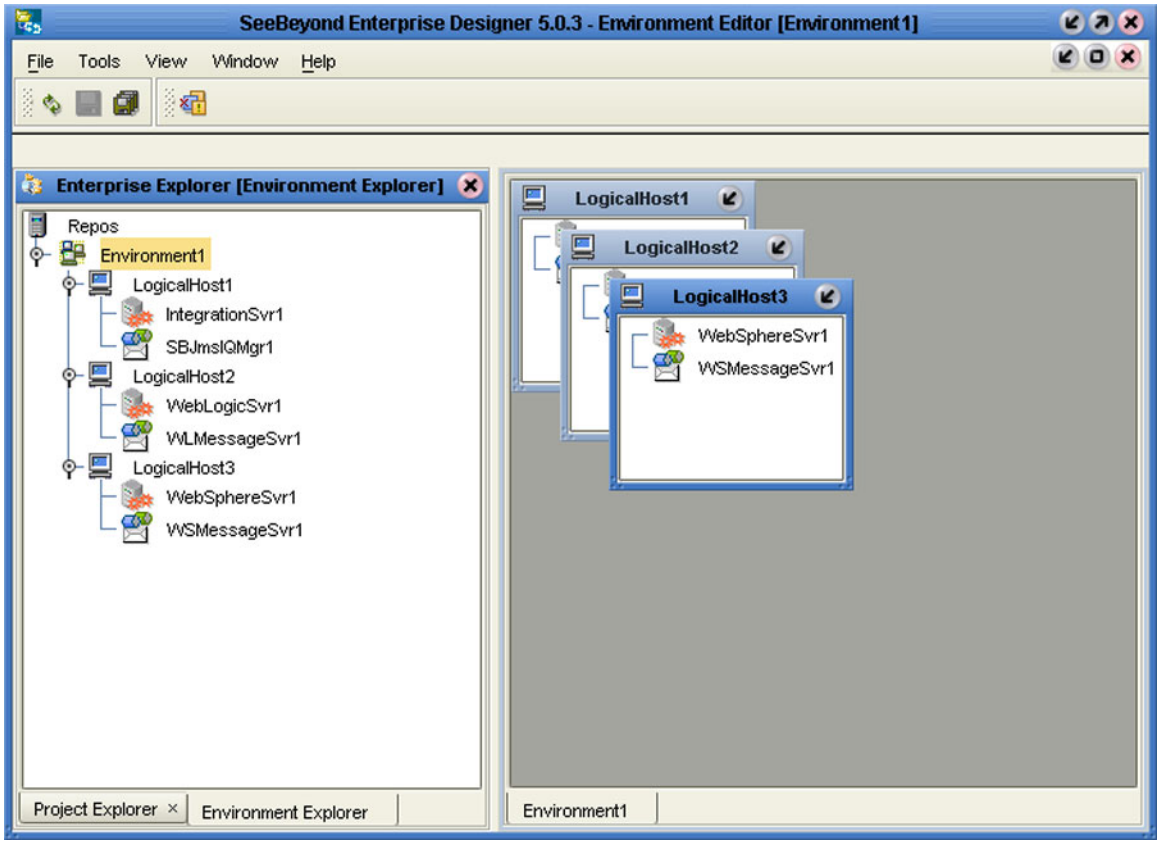
Figure 83 Logical Host Menu with Third-Party Servers



7.3 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 84).

Figure 84 Environment Editor

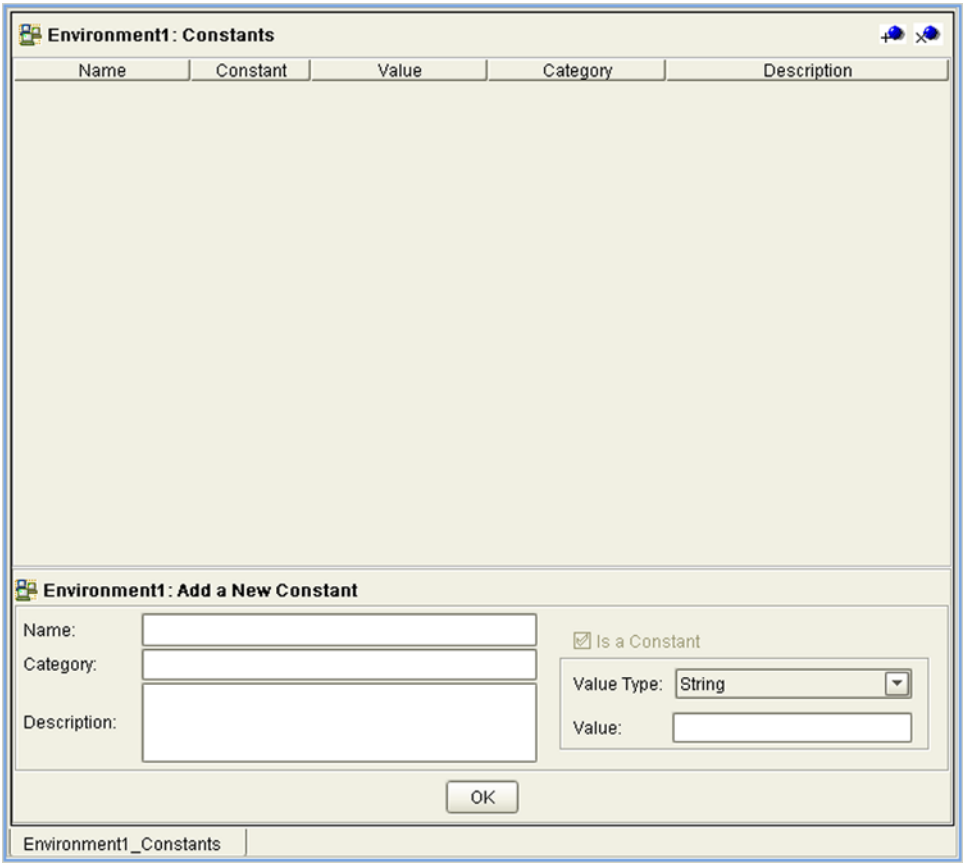


Here you can see the various components (Logical Hosts, servers, and external systems) included in the selected Environment. New Environments are added through the use of the Repository context menu (see **Repository Menu** on page 106). Components are added to the Environment by selecting options in the Environment and Logical Host context menus (see **Environment Menu** on page 107 and **Logical Host Menu** on page 108, respectively).

7.3.1 Defining Environmental Constants

Environmental constants are name/value pairs that are visible across the Environment. Selecting the **New Constant** option from the Environment context menu displays the Constants panel in the Environment Editor (see Figure 85).

Figure 85 Environmental Constants Panel



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

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

Table 29 Environmental Constants Panel Icons

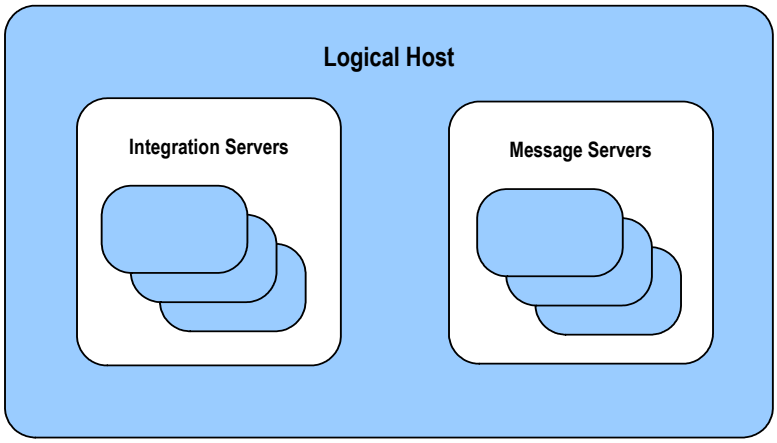
Icon	Name	Function
	Add a New Constant	Adds a new constant to the list.
	Delete a Highlighted Constant	Deletes the selected constant from the list.

7.4 Logical Hosts

7.4.1 Overview

A Logical Host 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 86.

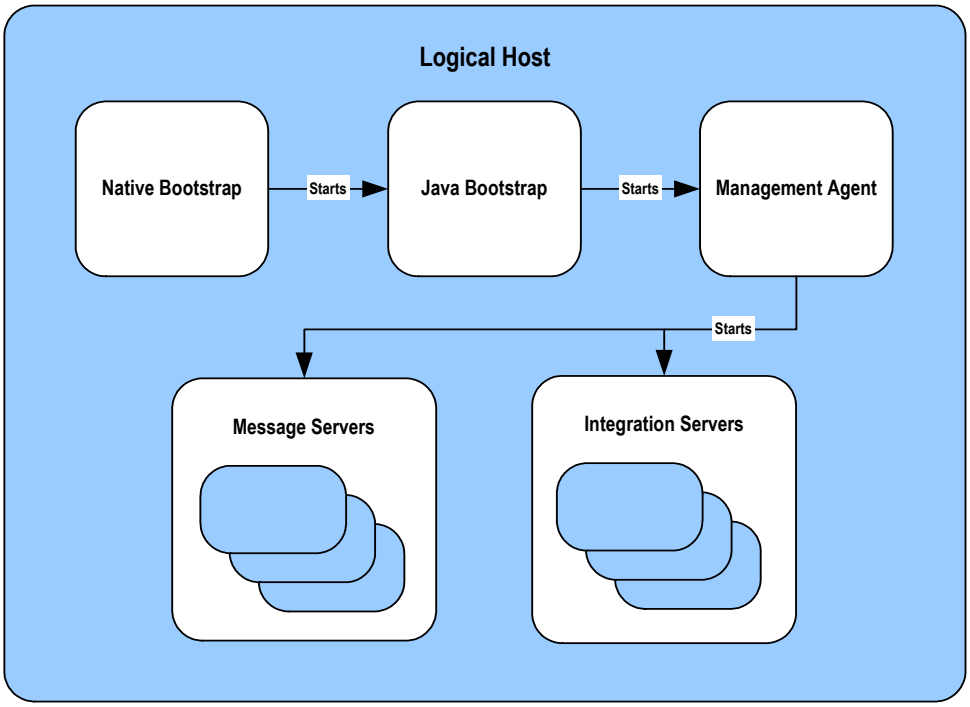
Figure 86 Logical Hosts



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

At run time, a platform-specific bootstrap script starts 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 87 illustrates this sequence.

Figure 87 Startup Sequence



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

7.4.2 Configuring a Logical Host

To access the configuration properties for a Logical Host

1. Right-click a Logical Host in the Environment Explorer tree to display the context menu for that Logical Host instance.
2. Select **Properties** from the context menu to display the **Properties** dialog box.
3. Select the **Logical Host Configuration** node in the properties tree to display the Logical Host Configuration Section, which contains the top-level configuration properties for the Logical Host (see Figure 88).

Figure 88 Logical Host Configuration Properties

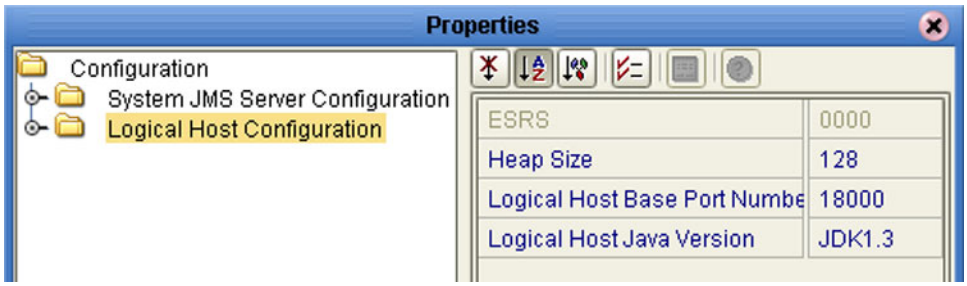


Table 30 Logical Host Configuration Properties List

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

Configuring the Base Port Number

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

The number of port numbers used in a Logical Host varies according to the specific implementation, so when assigning new base port numbers you need to skip successive numbers by an adequate amount. The default base port number is 18000, so base port numbers of 19000, 20000, and so on are recommended.

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

Note: *While Windows will accept port numbers below 12000, UNIX cannot.*

7.5 Integration Servers

The Logical Host contains one or more instances of a J2EE integration server, which is the engine that runs eGate Collaborations for processing business logic, and eWays that communicate with external applications. The integration server provides services for security, transactions, business rules execution, and connectivity management. eGate Integrator contains the SeeBeyond Integration Server, and also supports the use of third-party application servers such those supplied by BEA WebLogic and IBM WebSphere for this purpose (see **Deploying Projects to Third-Party Servers** on page 275).

7.5.1 Configuring an Integration Server

To access the configuration properties for an integration server

1. Right-click an integration server in the Environment Explorer tree to display the context menu for that instance.
2. Select **Properties** from the context menu to display the **Properties** dialog box.
3. Select the **IS Configuration** node in the properties tree to display the top-level IS configuration properties (see Figure 89).

Figure 89 Top-level IS Configuration Properties

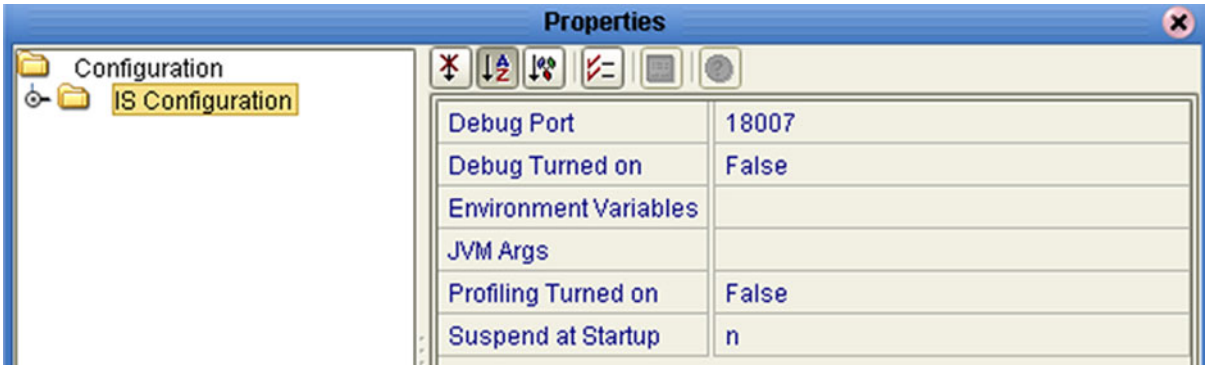


Table 31 Top-level IS Configuration Properties List

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 31 Top-level IS Configuration Properties List

Property	Description
JVM Args	Java Virtual Machine (JVM) arguments. Each element in the collection should specify one, and only one, argument.
Profiling Turned On	Enables/disables performance monitoring for the IS. The default is **False** (disabled). To enable, change to **True** (as shown) and configure the properties described in **Performance Monitoring (Profiling)** on page 119.
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 (if eInsight is installed)
- 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 90 Web Container Configuration Properties

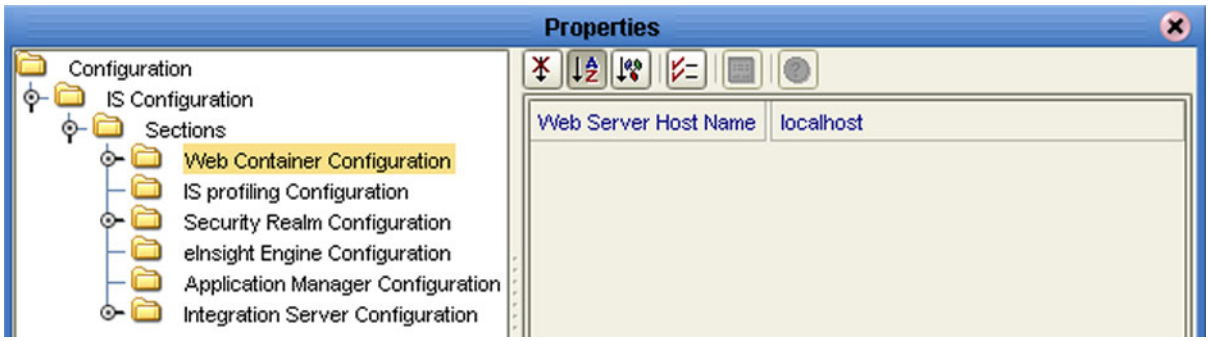


Table 32 Web Container Configuration Properties List

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

Default Web Server

The default Web server properties are contained in a subdirectory under the Web Container directory.

Figure 91 Default Web Server Properties

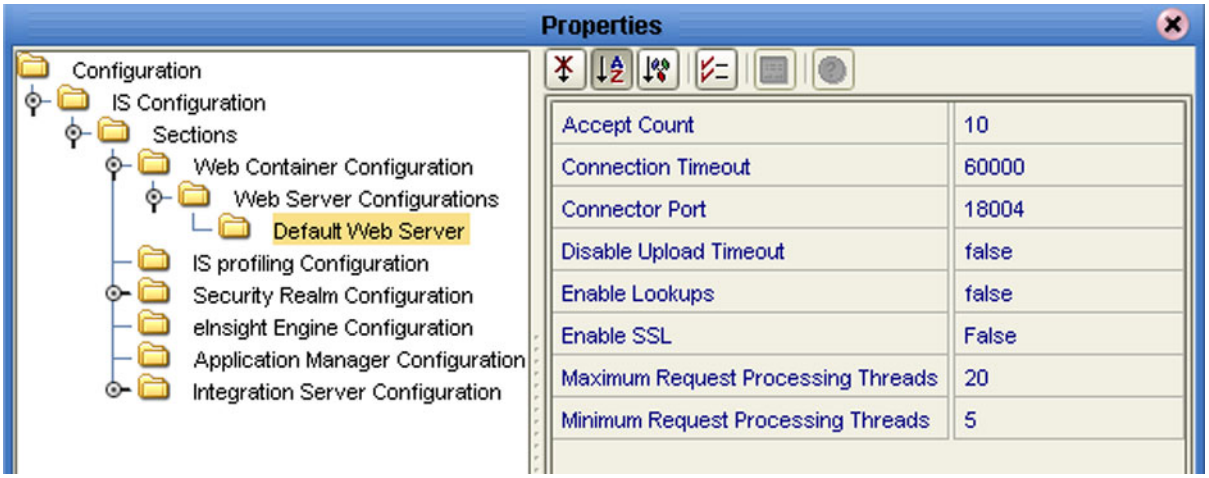


Table 33 Default Web Server Properties List

Property	Description
Accept Count	Specifies the maximum acceptable number of incoming connection requests when all possible request processing threads are in use. Any requests received beyond this number when the queue is full are refused. The default value is **10**.
Connection Timeout	Specifies the time period in milliseconds that this connector will wait for the request URI line to be presented, after accepting the connection. The default value is **80000** ms.
Connector Port	Specifies the connection port for the Web server. The default value is **18004**.
Disable Upload Timeout	Allows the servlet container to use a different, and longer, connection timeout while a servlet is executing. This gives the servlet a longer time to complete execution, and/or provides a longer timeout during data upload. The default value is **false**.
Enable Lookups	If set to **true**, calls are made requesting getRemoteHost() to perform DNS lookups in order to return the actual host name of the remote client. If set to **false**, the DNS lookup is bypassed and the IP address is returned in string form, thereby improving performance. The default value is **false**.
Enable SSL	Specifies whether or not to enable the Secure Sockets Layer (SSL) protocol. The default value is **false**.
Maximum Request Processing Threads	Specifies the maximum number of request processing threads to be created by this connector, thereby determining the maximum number of simultaneous requests that can be handled. The default value is **20**.
Minimum Request Processing Threads	Specifies the number of request processing threads to be created by this connector when it is first started. This value must be less than the value set for the Maximum Request Processing Threads property. The default value is **5**.

Performance Monitoring (Profiling)

You can monitor the performance of the integration server by using the built-in *Heap Analysis* tool, which is enabled and configured using the Profiling Configuration dialog box (see Figure 92).

Figure 92 Profiling Configuration Properties

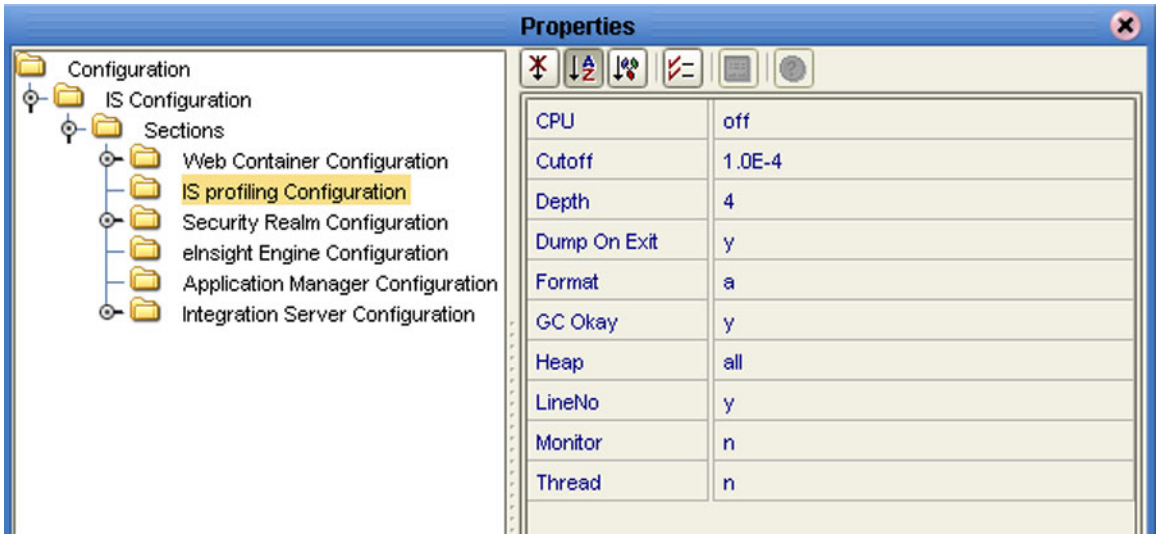


Table 34 Profiling Configuration Properties List

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

Security Realm

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

Figure 93 Security Configuration Properties

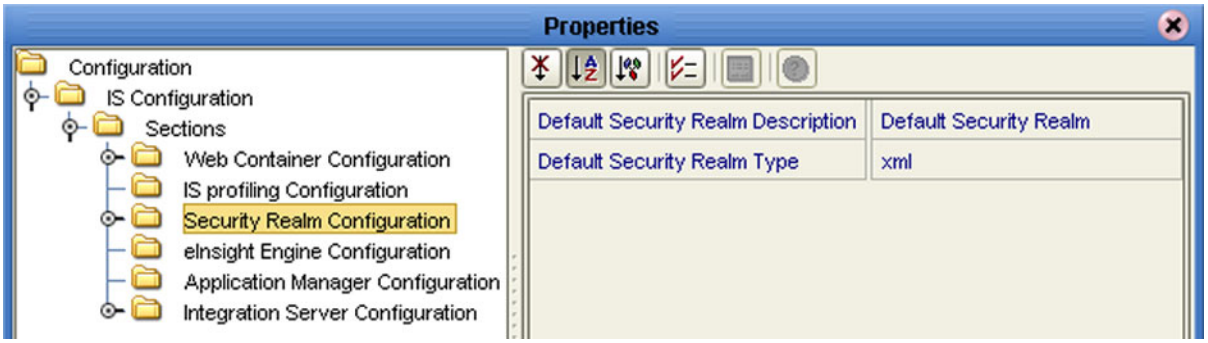


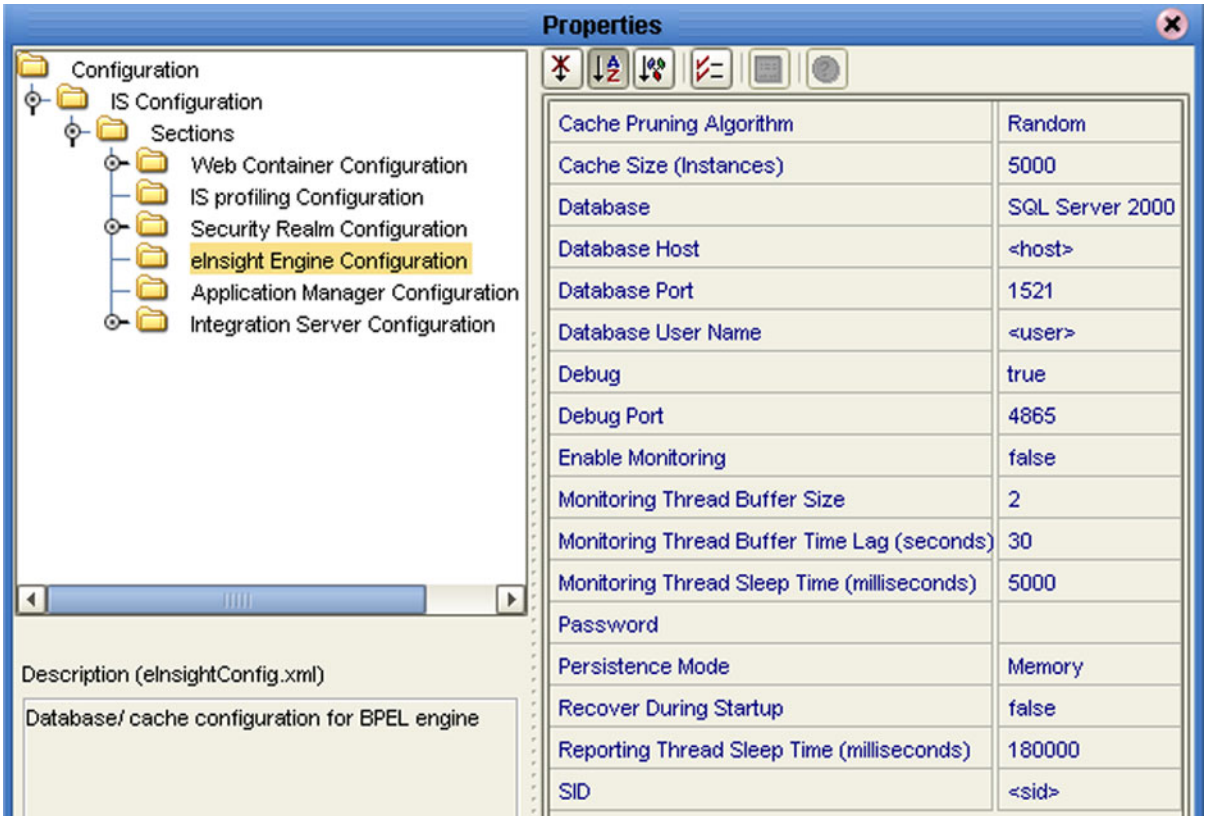
Table 35 Security Realm Configuration Properties List

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**.

eInsight Engine

This configuration node is displayed only if you have eInsight Business Process Manager installed on your system. The configuration properties relate to the BPEL engine's database cache; see the *eInsight Business Process Manager User's Guide* for information regarding these properties (see Figure 94).

Figure 94 eInsight Engine Configuration Properties



Application Manager

You can set integration server thread pool variables using the Application Manager Configuration Properties dialog box (see Figure 95).

Figure 95 Application Manager Configuration Properties

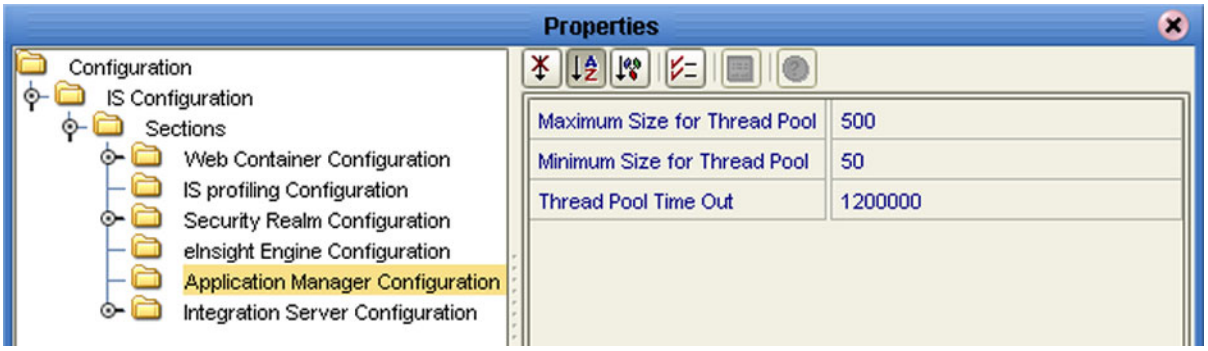


Table 36 Application Manager Configuration Properties List

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 96).

Figure 96 Integration Server Configuration Properties

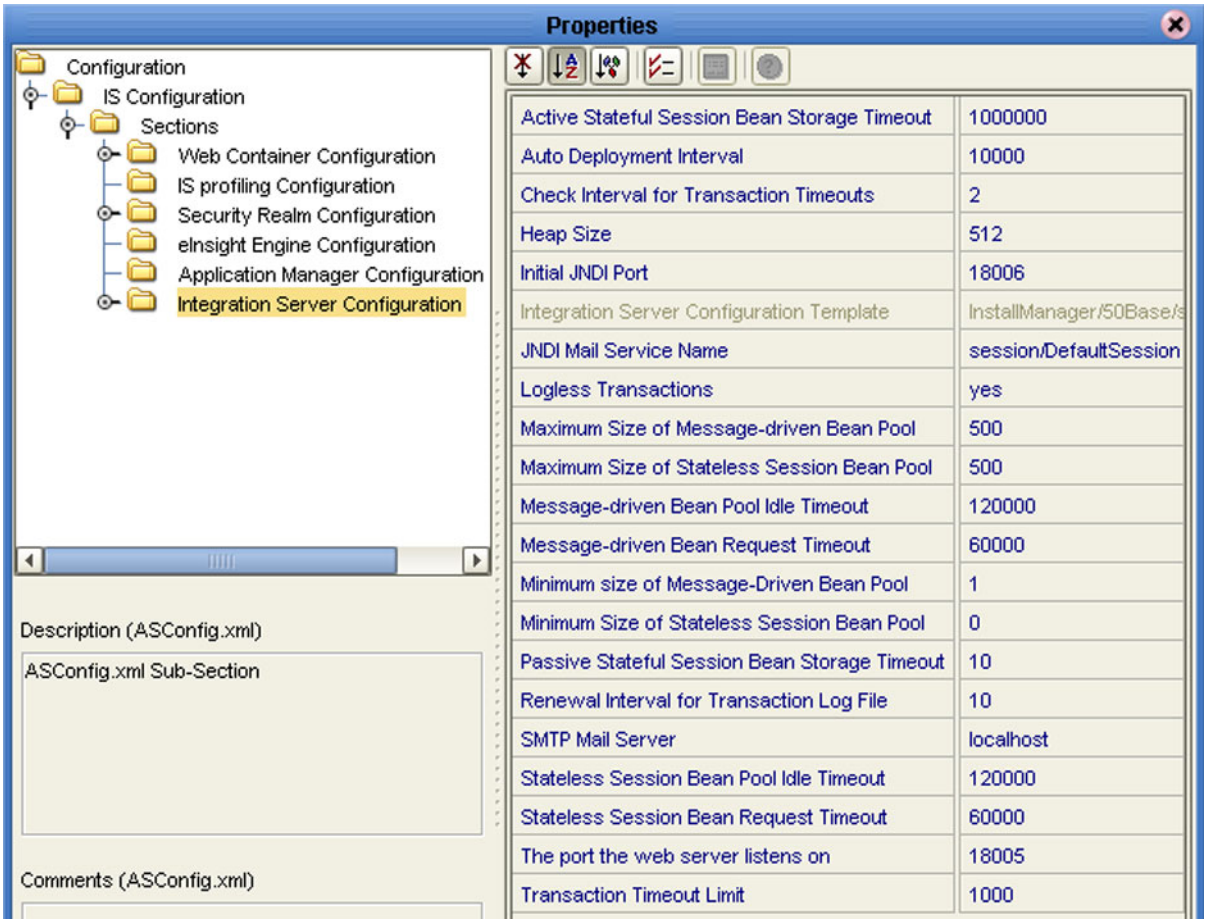


Table 37 Integration Server Configuration Properties List

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.
Check Interval for Transaction Timeouts	Specifies the interval between checks for transaction timeouts, measured in minutes. The default value is **2** min.

Table 37 Integration Server Configuration Properties List

Property	Description
Heap Size	Specifies the Heap size in Megabytes; the minimum size is **512** Mb, which is the default value. Increasing this value increases the JVM size.
Initial JNDI Port	Specifies the initial port required by the Naming Service class for startup. The default value depends upon the value of the Logical Host base port.
JNDI Mail Service Name	Specifies the name of the JNDI mail service. The default value is **session/DefaultSession**.
Logless Transactions	Specifies whether or not logless transactions are allowed. The default value is **yes**.
Maximum Size of Message-driven Bean Pool	Specifies the maximum number of Message-driven Beans allowed in the Message-driven Bean pool at one time. The default value is **500**.
Maximum Size of Stateless Session Bean Pool	Specifies the maximum number of Stateless Session Beans allowed in the Stateless Session Bean pool at one time. The default value is **500**.
Message-driven Bean Pool Idle Timeout	Specifies the timeout interval for the Message-driven Bean pool, measured in milliseconds. The default value is **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.
Renewal Interval for Transaction Log File	Specifies the interval for renewing the Transaction Service log file, measured in hours. The default value is **10** hr.
SMTP Mail Server	Specifies the name of the SMTP mail host server. The default value is **localhost**.
Stateless Session Bean Pool Idle Timeout	Specifies the timeout interval for the Stateless Bean pool, measured in milliseconds. The default value is **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.

Table 37 Integration Server Configuration Properties List

Property	Description
Transaction Timeout Limit	Specifies the time limit for transactions to time out, measured in seconds. The default value is **1000** sec.

Oracle JDBC Connection Pool

Connection Pool properties for an Oracle database associated with the integration server are specified in the Oracle JDBC Connection Pool dialog box (see Figure 97).

Figure 97 Oracle JDBC Connection Pool Properties

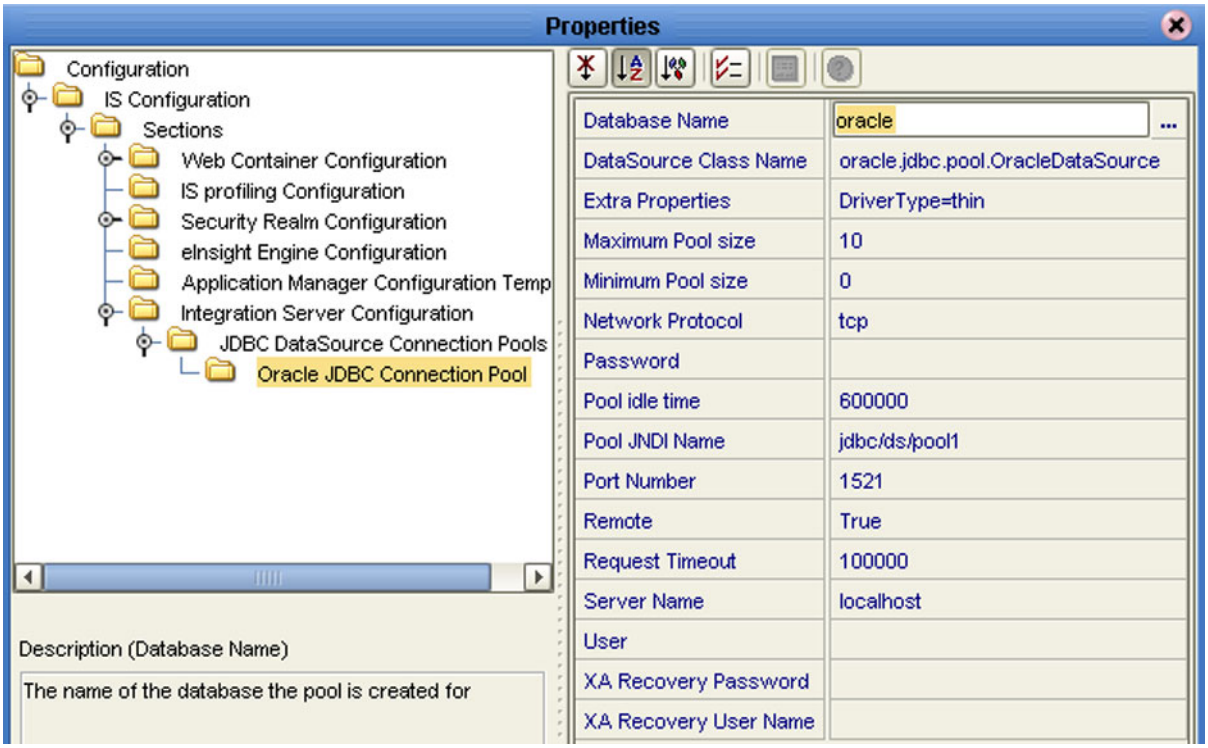


Table 38 Oracle JDBC Connection Pool Properties List

Property	Description
Database Name	Specifies the name of the database for which the pool is created. The default value is **oracle**.
DataSource Class Name	Specifies the name of the DataSource class. The default value is **oracle jdbc pool OracleDataSource**.
Extra Properties	Specifies custom properties for the DataSource, using semicolon-separated key-value pairs. The default value is **DriverType=thin**.
Maximum Pool Size	Specifies the maximum number of connections in the pool. The default value is **10**.

Table 38 Oracle JDBC Connection Pool Properties List

Property	Description
Minimum Pool Size	Specifies the minimum number of connections in the pool. The default value is **0**.
Network Protocol	Specifies the network protocol. The default value is **tcp**.
Password	Specifies the password for the connection. There is no default value.
Pool Idle Time	Specifies the maximum time period in milliseconds that a connection may remain unused before it is removed from the pool in order to reduce the pool size. The default value is **600000** ms.
Pool JINI Name	Specifies the unique JINI name of the DataSource pool. The pool is bound in the java/namespace for local access or into the global namespace for remote access. The default value is **jdbc/ds/pool1**.
Port Number	Specifies the port number on which the server receives data. The default value is **1521**.
Remote	Specifies whether or not the DataSource should be bound into the global remote JINI namespace for access by remote clients. The default value is **true**.
Request Timeout	Specifies the maximum time period in milliseconds that a request for connection from the pool may block all other connections currently in use. The default value is **100000** ms.
Server Name	Specifies the host name of the database server or IP address where the database server is running. The default value is **localhost**.
User	Specifies the user name authorized for creating connections. There is no default value.
XA Recovery Password	For XA DataSources only, specifies the password to use for XA transaction recovery. There is no default value.
XA Recovery User Name	For XA DataSources only, specifies the user name to use for XA transaction recovery. There is no default value.

7.5.2 Deploying User-Defined Stateless Session Beans

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

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

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

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

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

Example Remote Interface

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

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

Example Home Interface

```
package ejb.CustomApp;

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

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

Example Stateless Session Bean (SLSB)

```
package ejb.CustomApp;

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

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

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

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

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

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

     return( EUID );
  }

  // add addtional EJB methods

}
```

Example ejb-jar.xml file for the above SLSB

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

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

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

Example seebeyond-ejb.xml file for the above SLSB

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

SLSB Deployment verification

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

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

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

7.6 Message Servers

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

Third-party application servers such as BEA WebLogic and IBM WebSphere incorporate their own message servers. For more information on the JMS IQ Manager, and deploying Project components to third-part message servers, see the *eGate Integrator JMS Reference Guide*.

Chapter 8

Project Deployment

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

8.1 Deployment Profiles

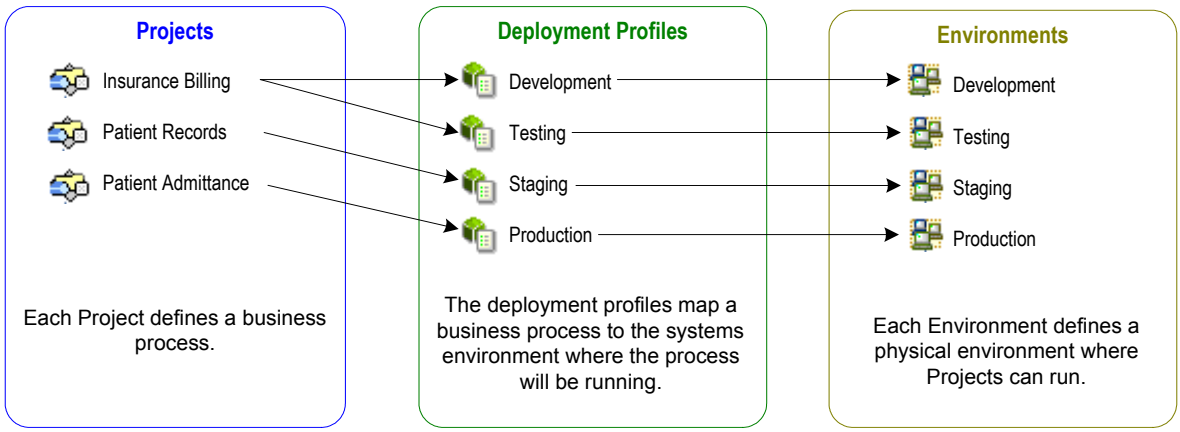
Deployment Profiles define specific instances of a Project in a particular Environment. A deployment profile contains information about the assignment of Services and Message Destinations to integration and message servers (JMS IQ Managers). It also contains version information for all relevant objects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

Note that:

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

Repeating Figure 2 from the **System Overview** on page 18:

Figure 98 eGate Integrator Implementation Model



8.2 The Deployment Editor

The Deployment Editor (see Figure 99) allows you to create a new Deployment Profile or edit an existing one. To create a new Deployment Profile, right-click on a Project in the Project Explorer to display its context menu. From the menu, select **New > Deployment Profile**. To edit an existing Deployment Profile, simply click on its icon.

Figure 99 Deployment Editor Window

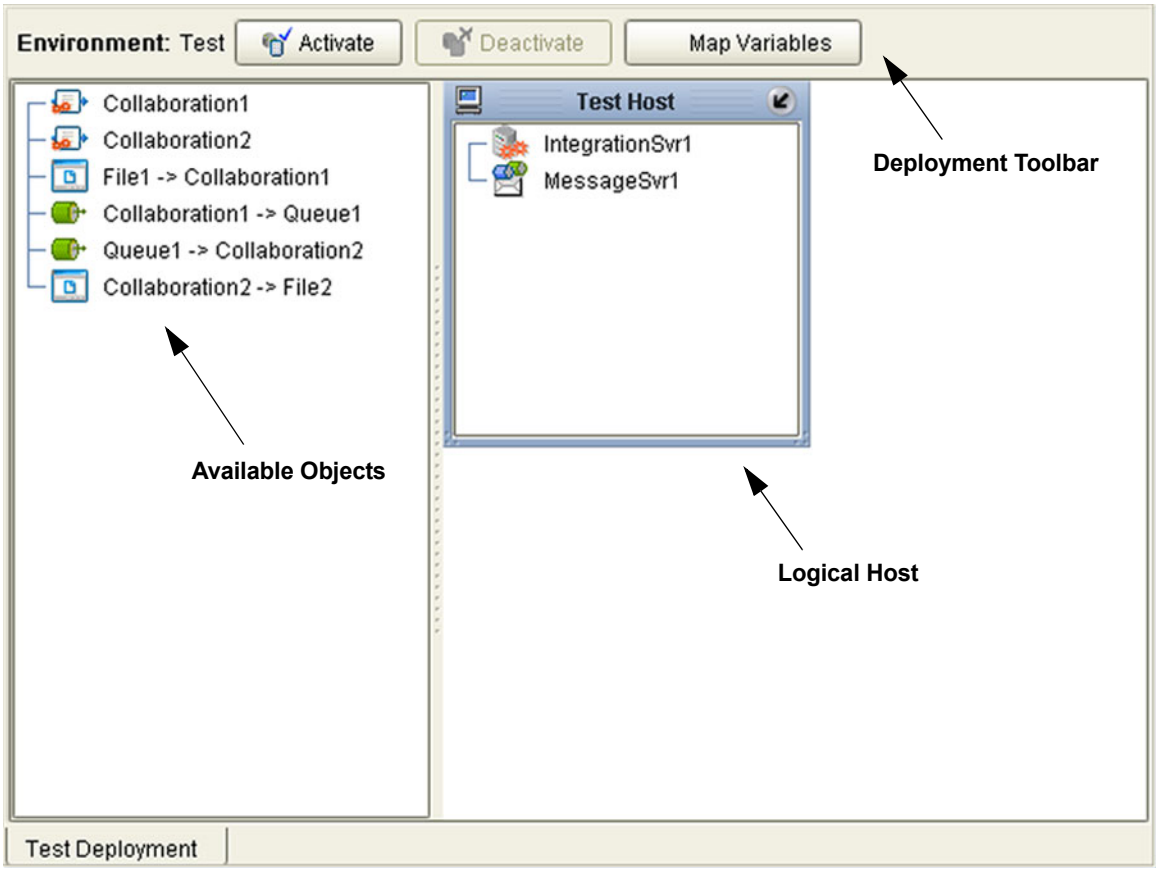


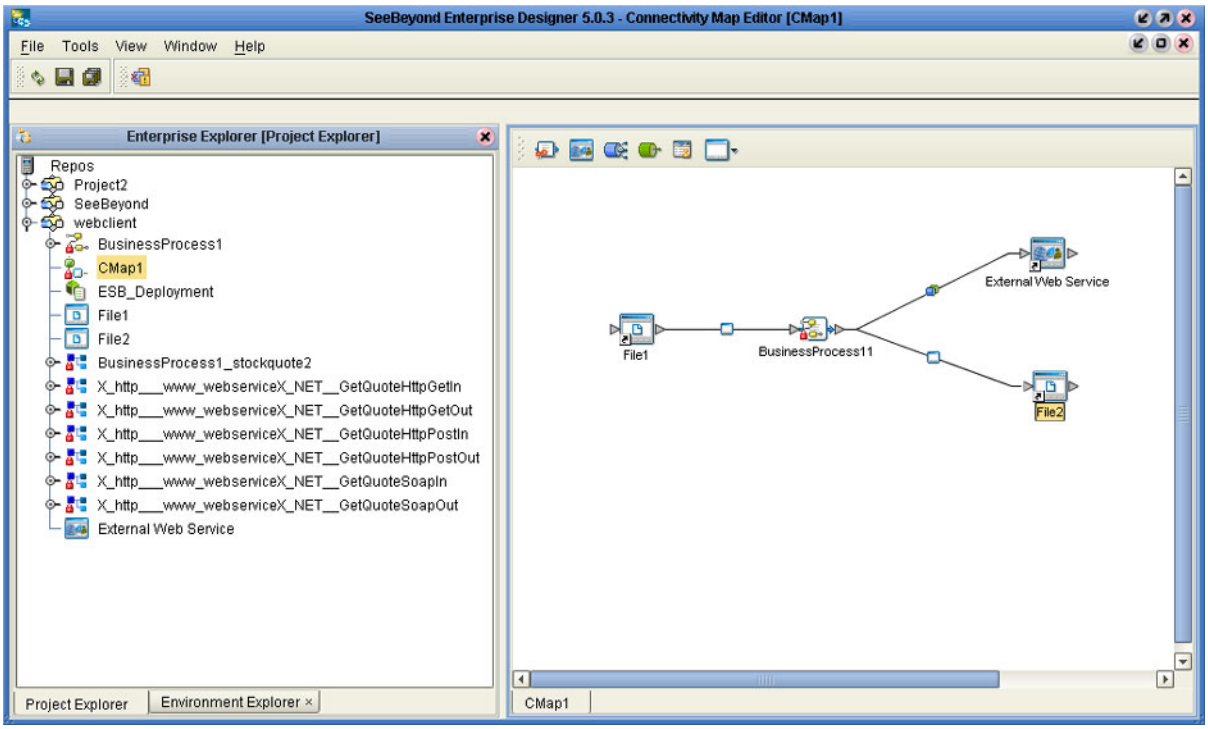
Table 39 Deployment Toolbar Buttons

Button	Function
Activate	Starts the Project by creating an enterprise archive (EAR) file based on the Connectivity Map and linking this file with the application server. See **Activating and Deactivating Deployment Profiles** on page 136.
Deactivate	Stops the Project by terminating the link between the EAR file and the application server, sets the Deployment Profile to *inactive*, and saves to the Repository.
Map Variables	Allows you to assign names and values to Project variables for the specific Deployment Profile. See **Mapping Variables** on page 138.

8.3 Creating a Deployment Profile

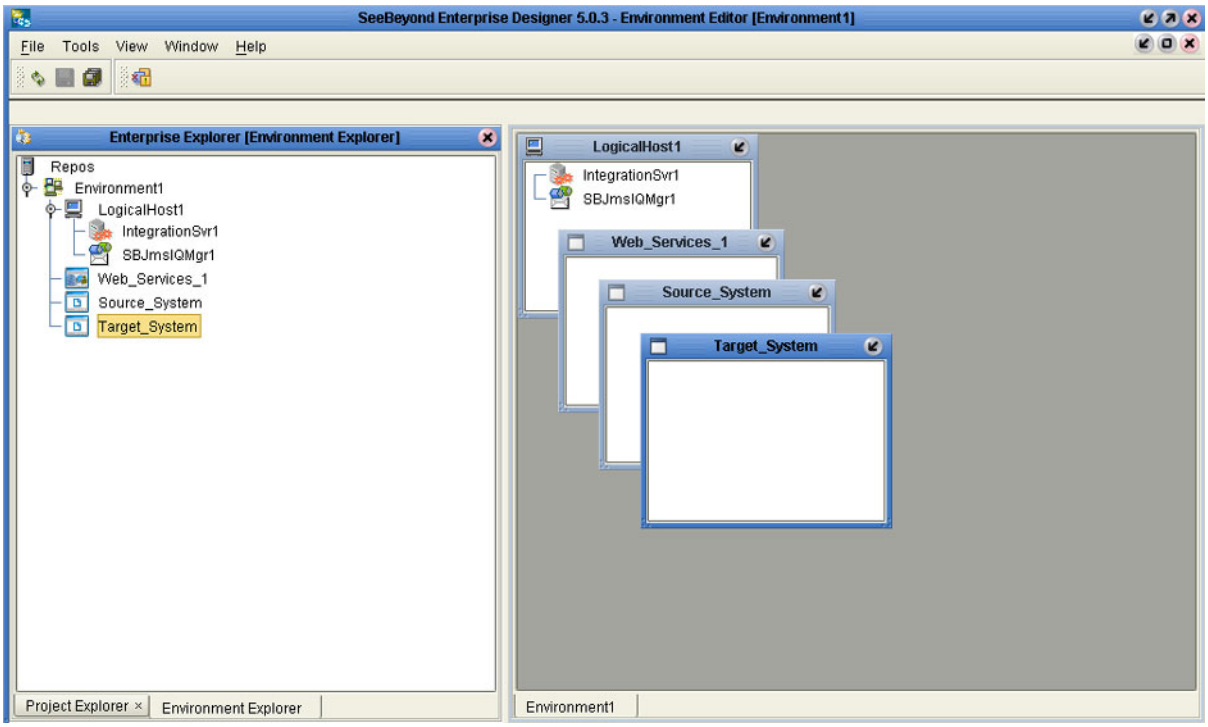
The Web Client Project shown in Figure 100 will be used as a deployment example.

Figure 100 Web Client Example Project



1. In the Environment Explorer, create an Environment and right-click on the Environment to display its context menu.
2. From the menu, select the Environment components you need and name them appropriately. They will appear as shown in Figure 101.

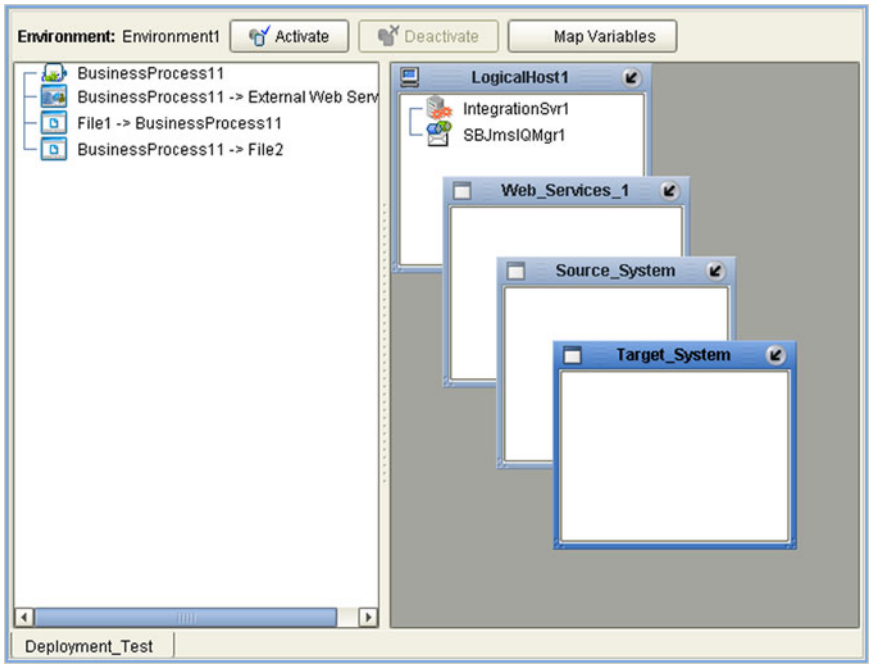
Figure 101 Web Client Example Environment



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

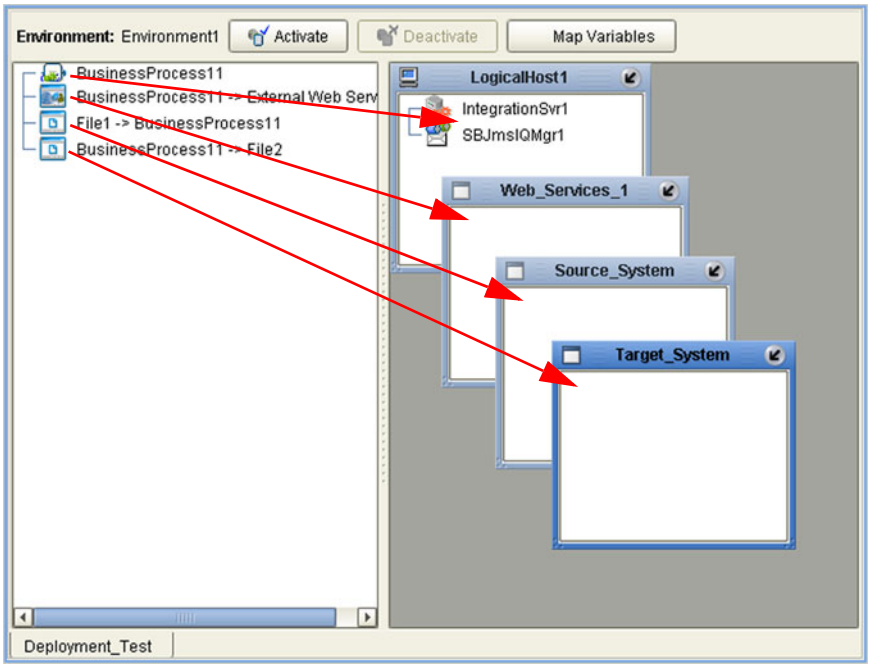
4 From the menu, select **New > Deployment Profile**. The Deployment Profile Editor appears, displaying the Environment you created (see Figure 102).

Figure 102 Example Deployment Profile (1)



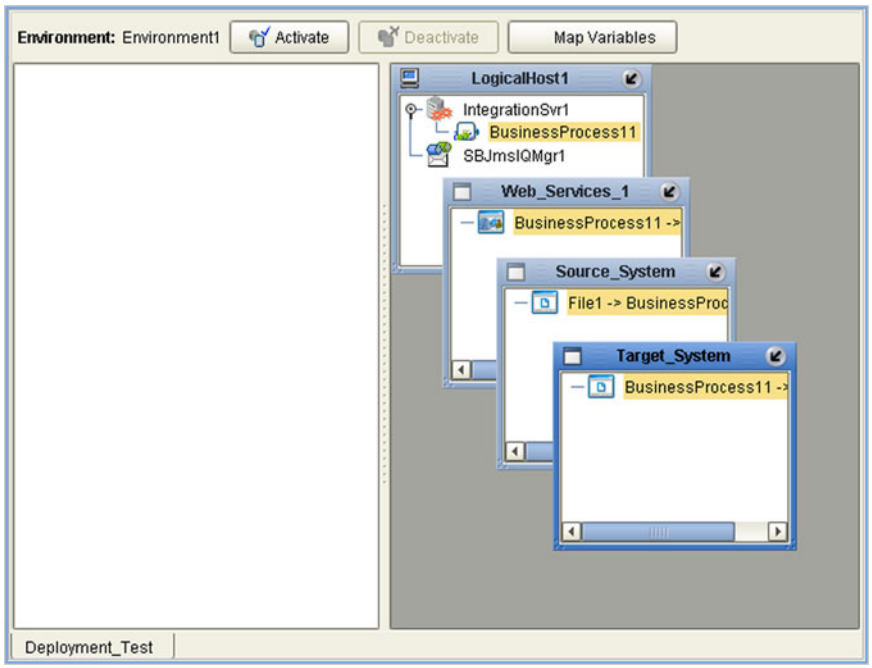
5 Drag the Project components from the left panel and drop them into the appropriate Environment components in the right panel, as illustrated in Figure 103.

Figure 103 Example Deployment Profile (2)



6 When the Environment components are fully populated, the left panel will be blank, as shown in Figure 104. You should now **Save** the profile.

Figure 104 Example Deployment Profile (3)



8.4 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.

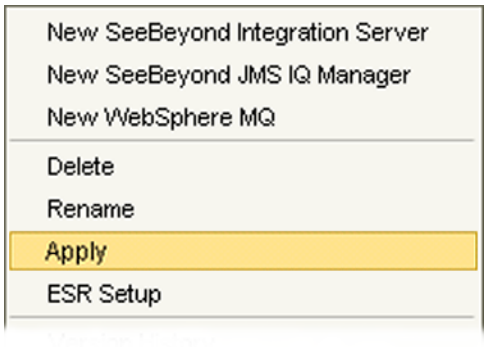
To activate a Deployment Profile

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

Figure 105 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 106). This will apply all of the changes for that Logical Host.

Figure 106 Logical Host Context Menu - Apply

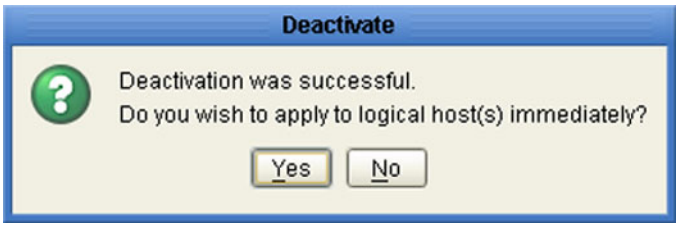


Note: *The **Apply** action assumes that the Logic Host is running, since it invokes a trigger to the Logical Host causing it to download the latest settings from the Repository and deploy those settings to all components on the Logical Host.*

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 107 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 106). 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 application server, leftover work directories in the repository/application directory are deleted.*

8.4.1 Command-line Activation and Deactivation

A script named **CmdLineUtil.sh** (or **.bat**) allows you to deploy and undeploy a specific project via the command line. The *deploy* and *undeploy* commands function in the same way as the **Activate** (or **Deactivate**) and **Apply Changes to LogicalHost** commands in Enterprise Designer, with the one exception that the *deploy* command does not try to regenerate project file—it only copies the existing project file from the Repository to the Logical Host.

8.5 Mapping Variables

Project variables function as placeholders, having values that are determined when you create a specific Deployment Profile. These values can be literals or Environmental constants. Clicking the **Map Variable** button displays the Deployment Profile Mappings panel, where you can assign names (see Figure 108) and values (see Figure 109).

Figure 108 Deployment Profile Mappings

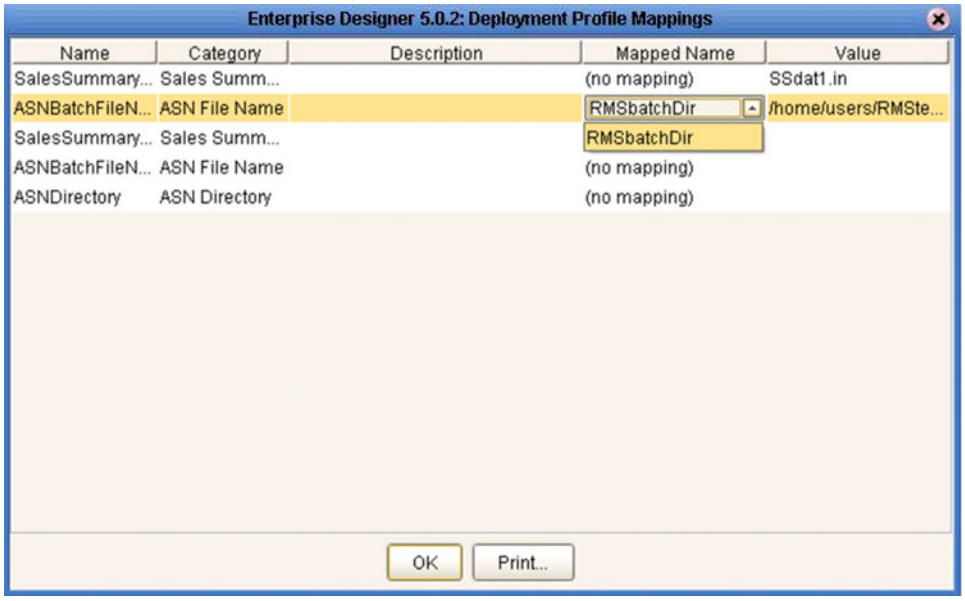
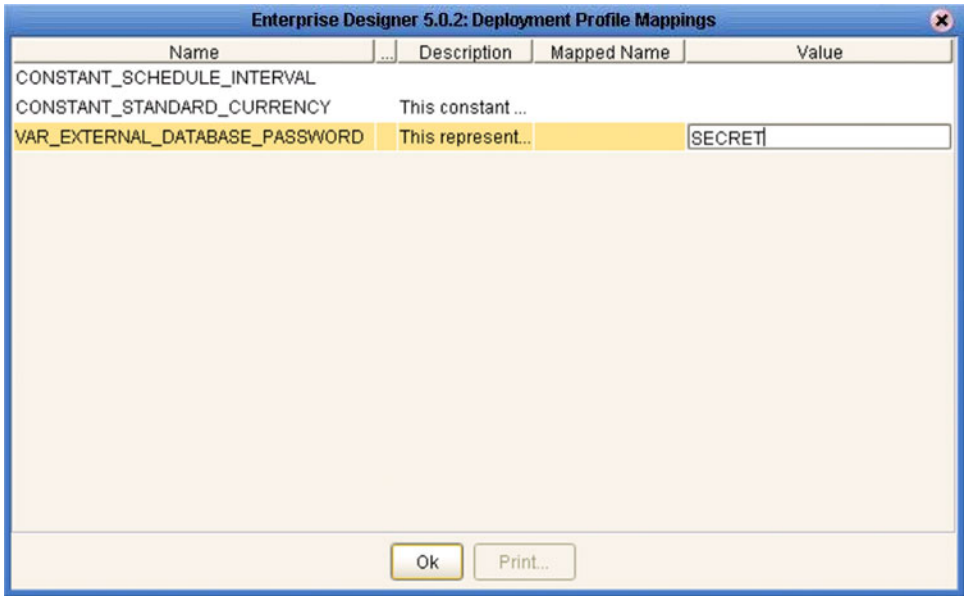


Figure 109 Project Variable Value Entry



8.6 Deploying Projects to Third-Party Servers

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

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

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

8.6.1 BEA WebLogic

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

To install additional .jar files

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

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

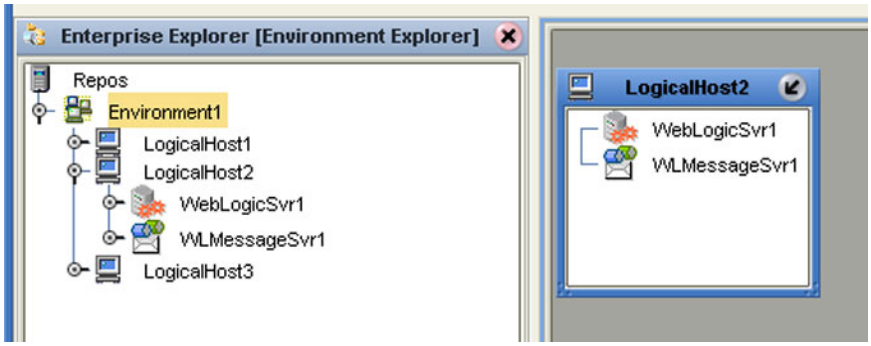
   http://xml.apache.org/dist/xerces-j
3. Place both **.jar** files into the ***weblogic8x*\server\lib** directory.
4. Add the **.jar** files to the *set CLASSPATH* segment of the **startWLS.cmd** file located in the ***weblogic8x*\server\bin** directory. The text to be added is:

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

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

1. Create the following components in Enterprise Designer (see Figure 110):
   A. A new environment
   B. A Logical Host
   C. A WebLogic J2EE application server
   D. A WebLogic JMS message server

Figure 110 WebLogic Deployment (1)

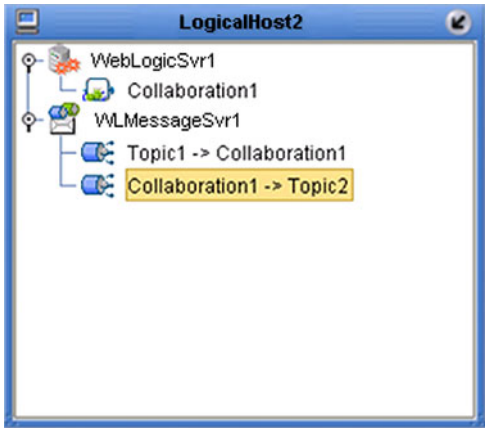


2 Create a new Deployment Profile to bind the Connectivity Map to the new WebLogic environment (see Figure 111).

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

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

Figure 111 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.

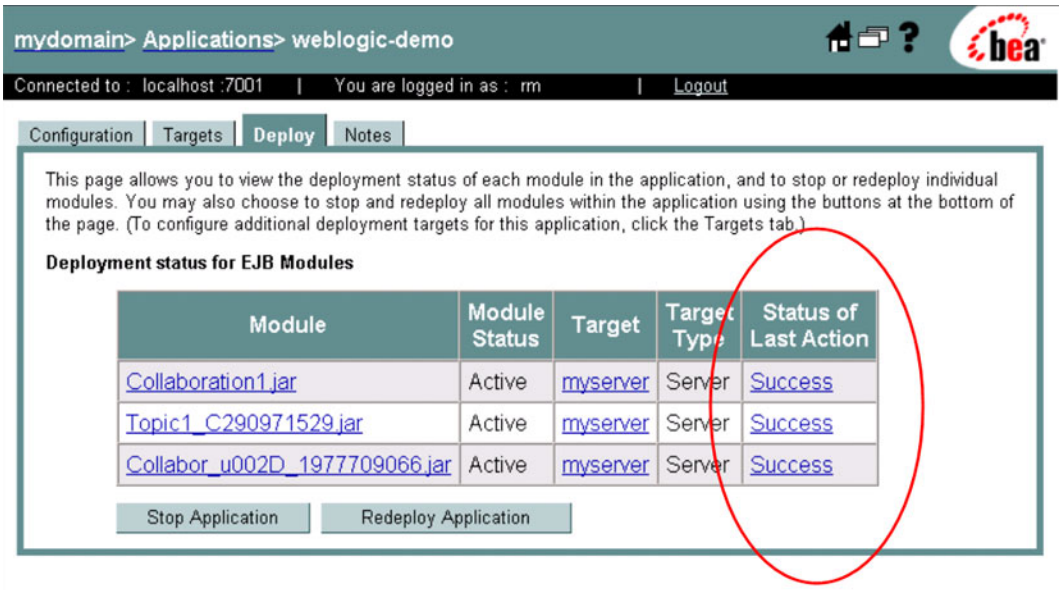
F Select the EAR file you just installed as the archive for the new application.

G Enter a name for the new application.

H Click **Deploy**.

I Verify the success of the deployment (see Figure 112 , which shows a WebLogic 8.1 example).

Figure 112 WebLogic Deployment Verification



8.6.2 IBM WebSphere

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

To install log4.jar

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

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

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

To deploy an eGate Project to an IBM WebSphere 5.0 or 5.0.1 environment

1 Create the following components in Enterprise Designer (see Figure 113):

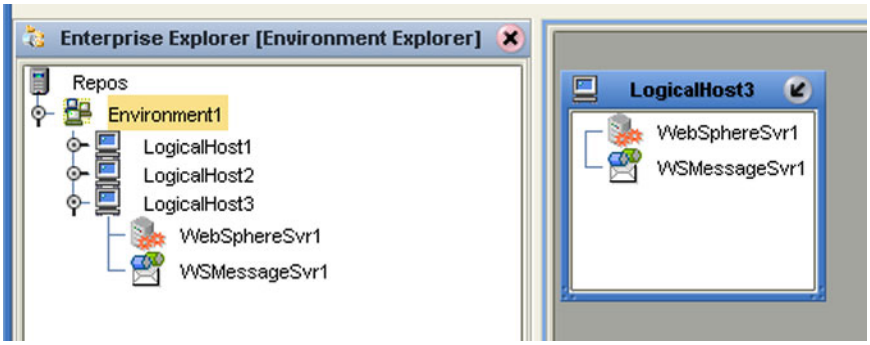
A A new environment

B A Logical Host

C A WebSphere J2EE application server

D A WebSphere JMS message server

Figure 113 WebSphere Deployment (1)

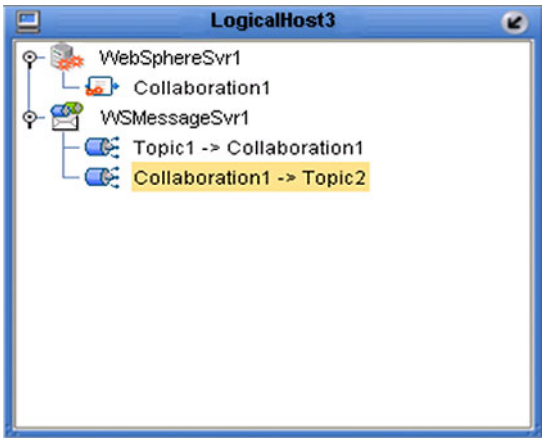


2 Create a new Deployment Profile to bind the Connectivity Map to the new WebSphere environment (see Figure 114).

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

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

Figure 114 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**.

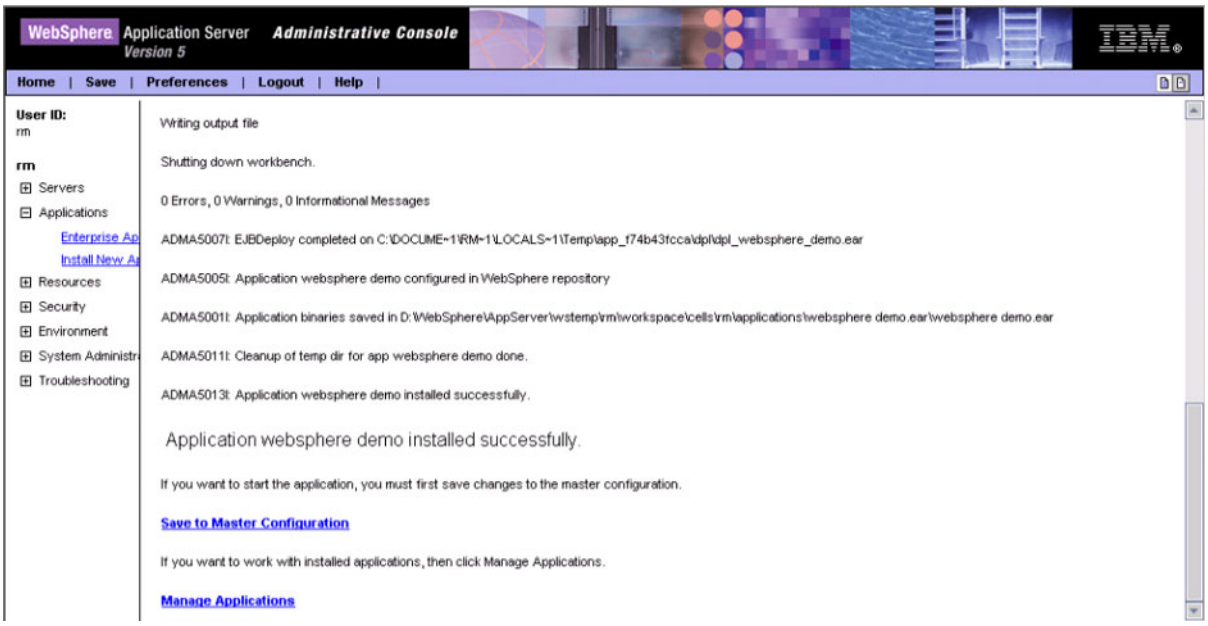
16 In *Step 7, Map modules ...*, check all modules and click **Next**.

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 115 , which shows a WebSphere 5 example).

Figure 115 WebSphere Deployment Verification



Chapter 9

Web Services

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

9.1 Overview

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

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

- **XML** (Extensible Markup Language)

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

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

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

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

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

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

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

9.2 SeeBeyond Web Services

eGate Integrator provides the capability to create either a client or a server to receive WSDL file from a remote server, or send WSDL files to a remote client. eGate works in conjunction with eInsight Business Process Manager, in which the associated business processes are developed. See **Building a Web Client** on page 149 and **Building a Web Server** on page 158.

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 88.

- **WSDL Editor**

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

- **WSDL Interface Designer**

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

- **WSDL Viewer**

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

- **UDDI Repository**

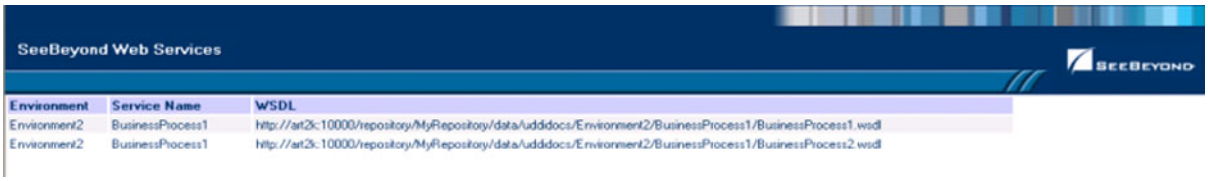
  All ICAN objects represented in the Repository that can be accessed as Web services are presented via a UDDI-compliant server. See **UDDI Repository** on page 147.

9.3 UDDI Repository

In general, all ICAN objects that expose themselves as a Web service (such as an eInsight business process) are presented via a UDDI-compliant server (see Figure 116). The URL of this server is:

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

Figure 116 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.

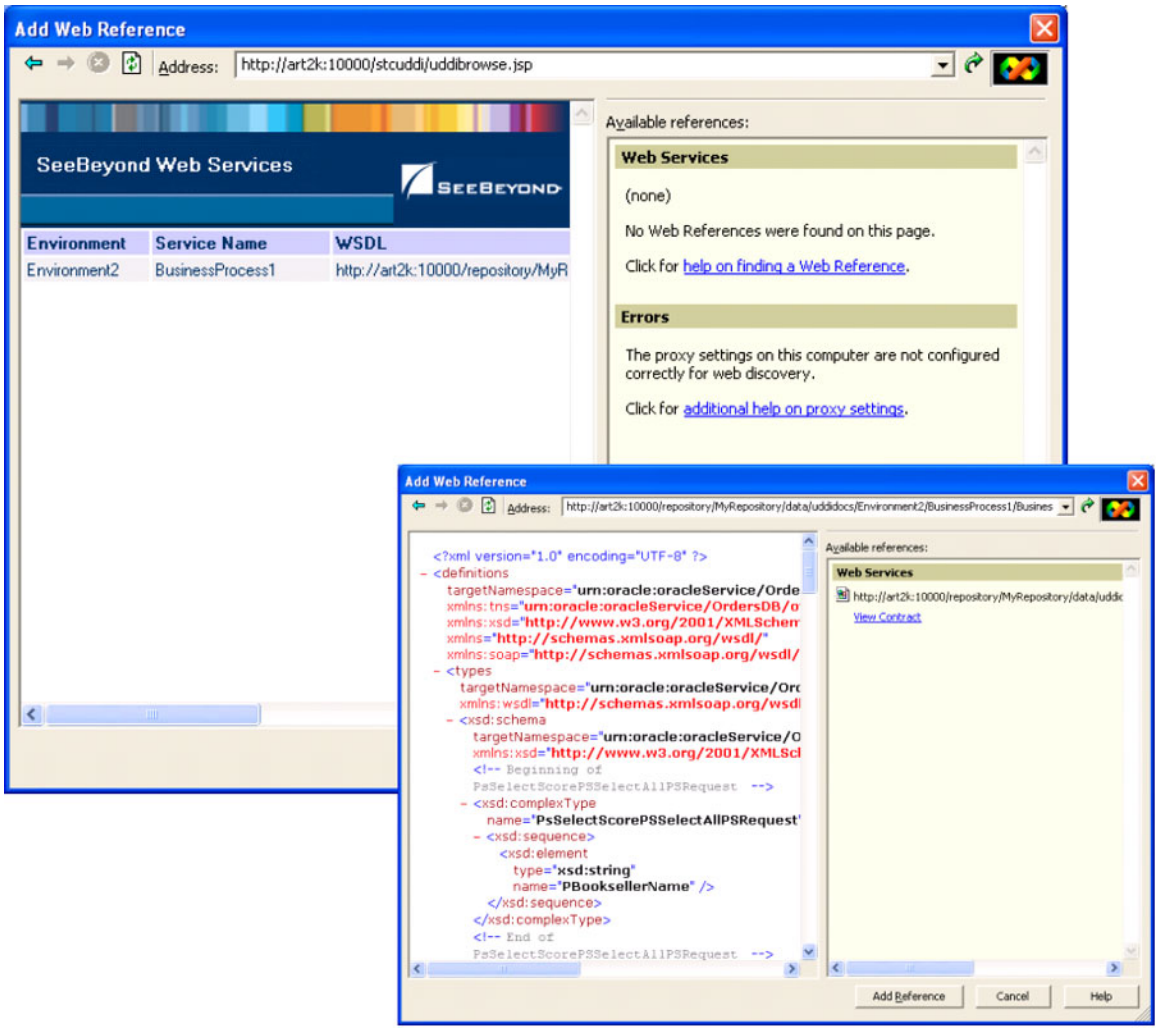
By selecting an entry its WSDL file is displayed, as shown in Figure 117.

Figure 117 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 118). In this example, a so-called *Web reference* (to the UDDI Repository) is added to a C# project.

Figure 118 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 88).

9.4 Building a Web Client

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

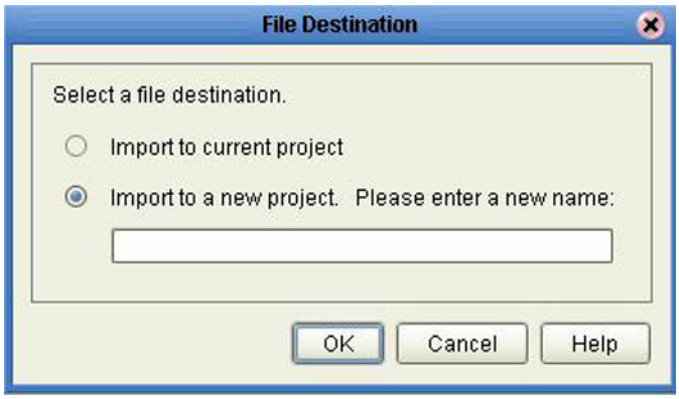
1. Build an Object Type Definition (OTD).
2. Develop a business process.
3. Create the eGate Project.
4. Deploy the Project to the selected Environment.

The Project used in the following example is 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 119), select **Import to a new Project**, enter the name of the Project, and click **OK**.

Figure 119 File Destination Dialog Box



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

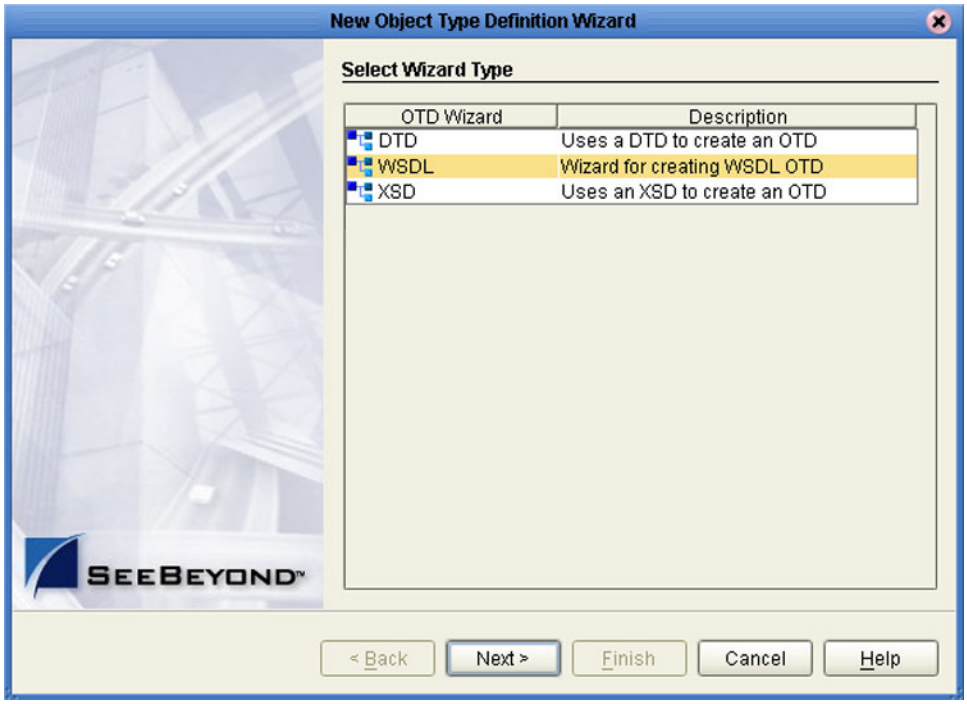
9.4.1 Object Type Definition

To create the WSDL OTD

1 In Project Explorer, create a new OTD.

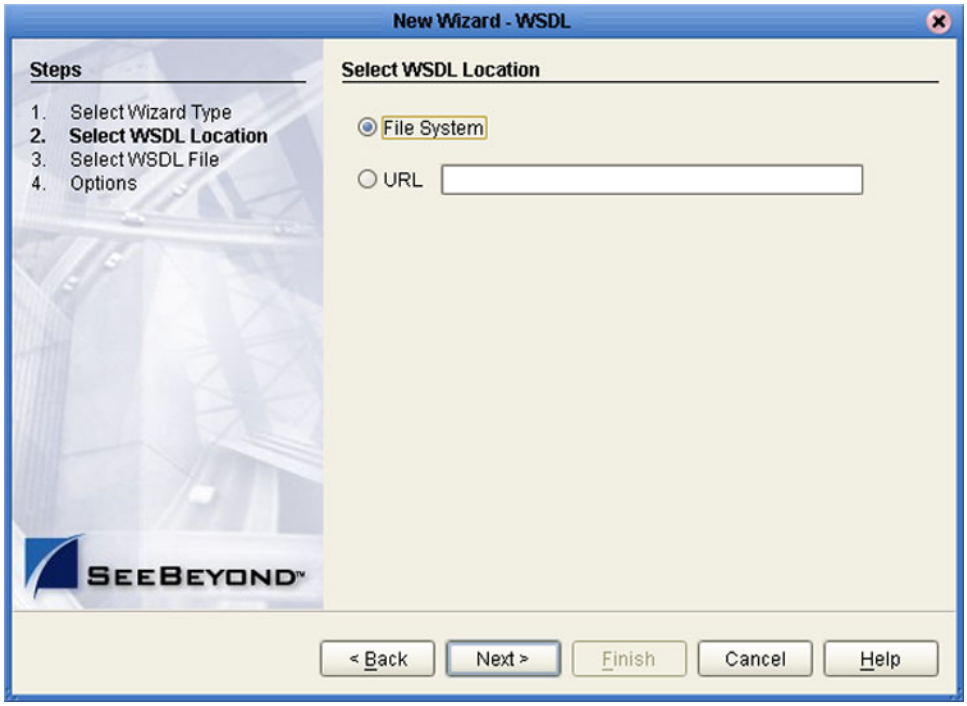
2 Select the **WSDL** OTD Wizard (see Figure 120).

Figure 120 Select WSDL Wizard



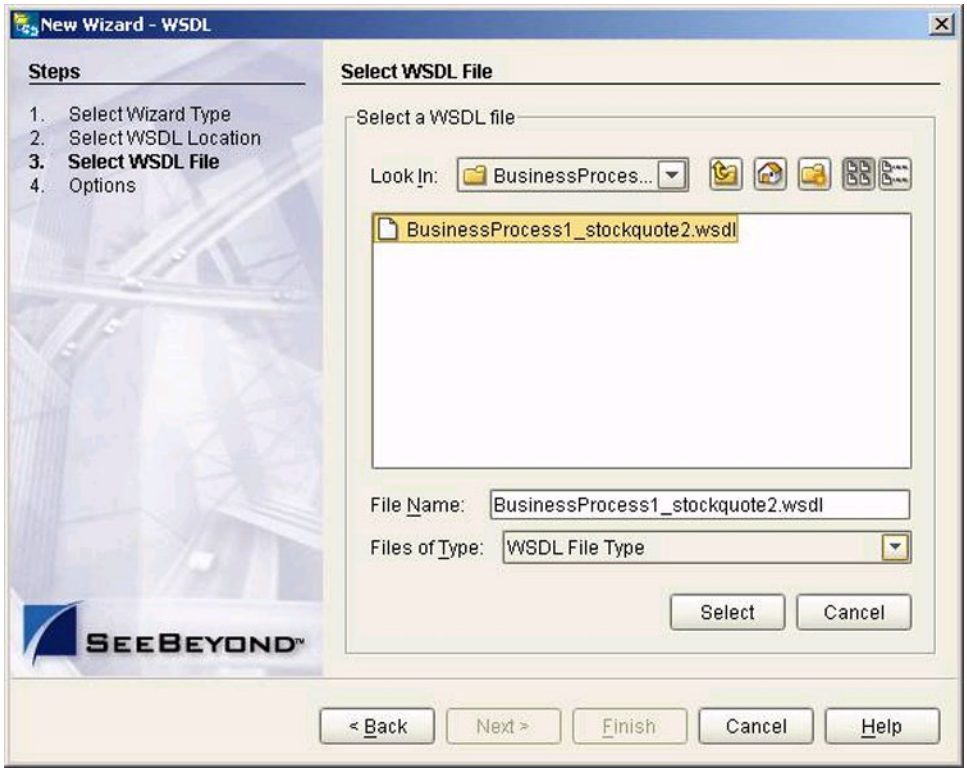
3 Select the WSDL file location and click **Next**. In this example, the file is located in the local file system (see Figure 121).

Figure 121 Select File Location



4 Select the WSDL file on which you want to base the OTD (see Figure 122). The file itself is shown in **Figure 127 on page 155**.

Figure 122 Select WSDL File

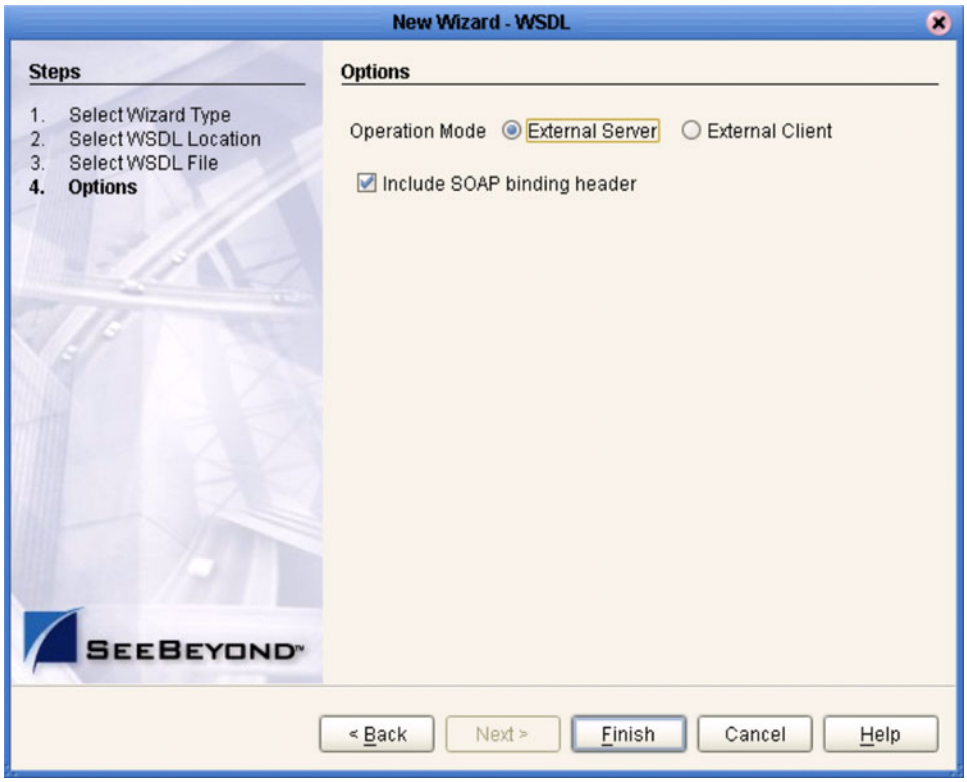


5 For a Web Client, select the following options (see Figure 123):

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

B Select **Include SOAP binding header** .

Figure 123 Select External Server



6 Click **Finish**.

9.4.2 eInsight Business Process

The example business process, developed in eInsight Business Process Manager, is shown in Figure 124 (see the *eInsight Business Process Manager User's Guide* for details).

Figure 124 Web Client Business Process

The **receive** rule for the business process is shown in Figure 125, and the **write** rule is shown in Figure 126.

Figure 125 Web Client Business Process *Receive* Rule

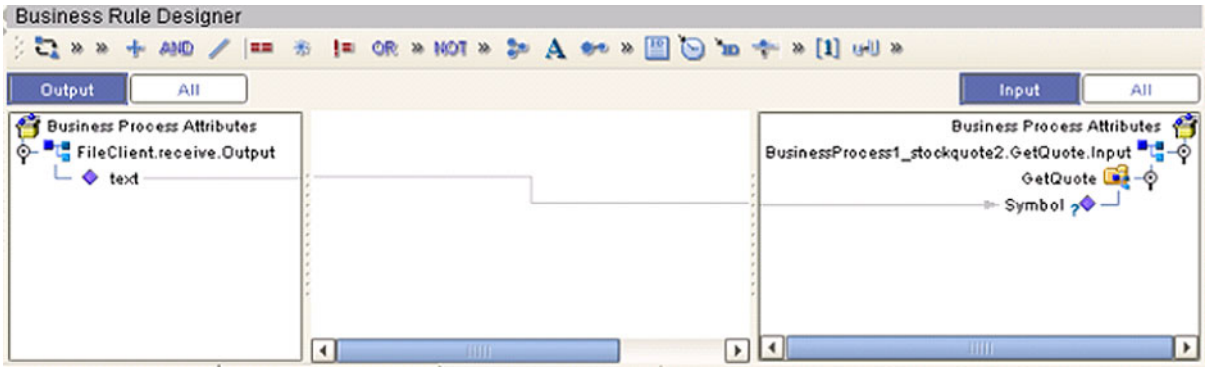
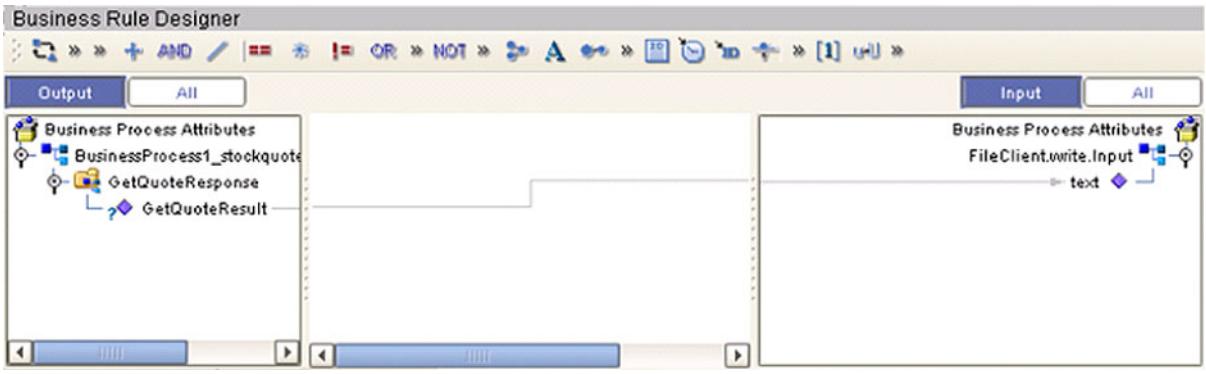


Figure 126 Web Client Business Process *Write* Rule



The WSDL file describing the business process is shown in Figure 127.

Figure 127 Sample WSDL File

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

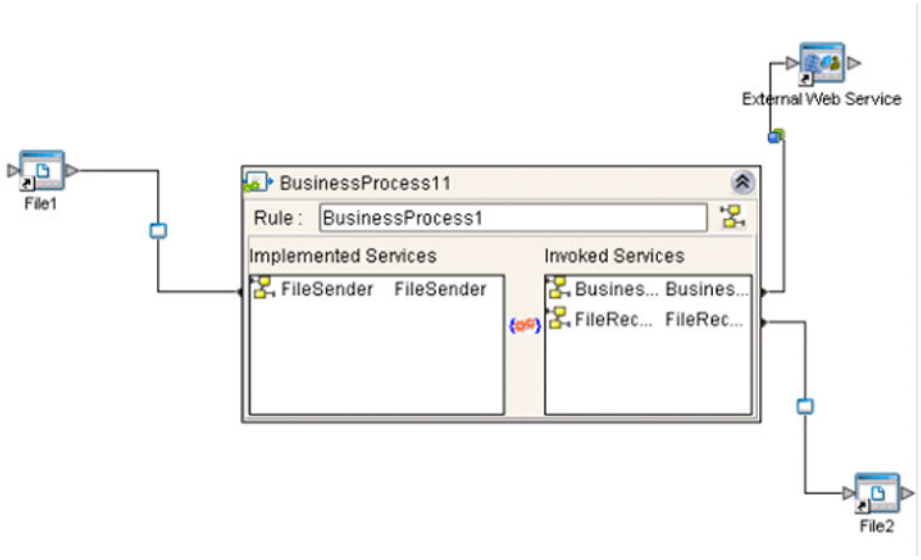
9.4.3 eGate Project

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

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

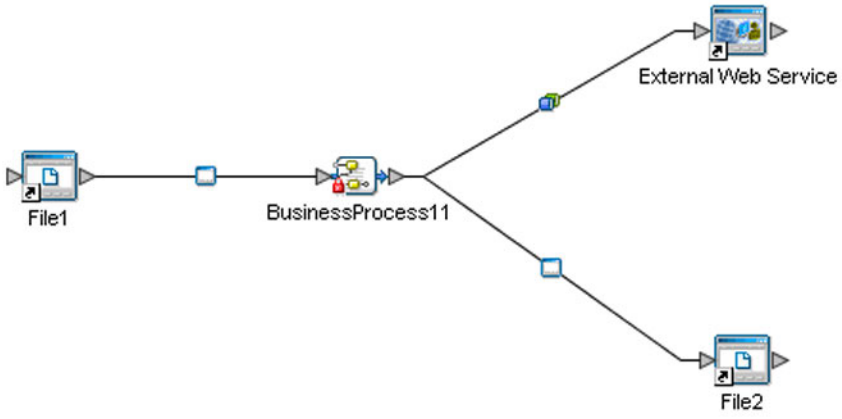
The business process is connected as shown in Figure 128.

Figure 128 Map Business Process



The completed Connectivity Map for the example Project is shown in Figure 129.

Figure 129 Web Client Connectivity Map

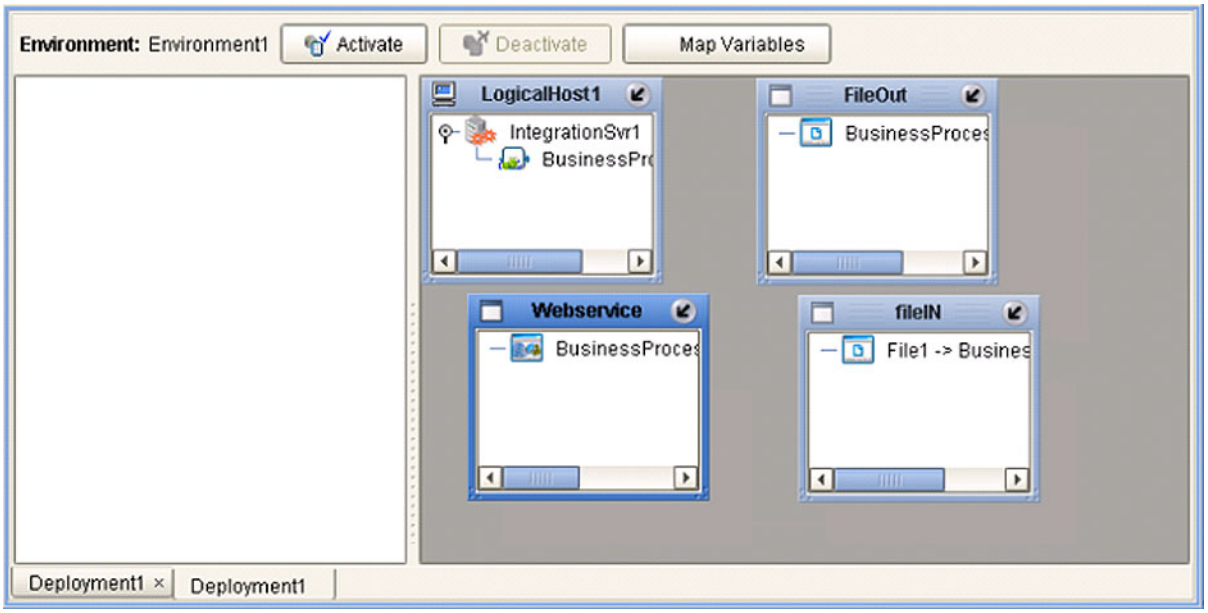


The Web client example Project appears in the Project Explorer as shown in Figure 130.

Figure 130 Web Client Example Project

The example Project is deployed as shown in Figure 131.

Figure 131 Project Deployment



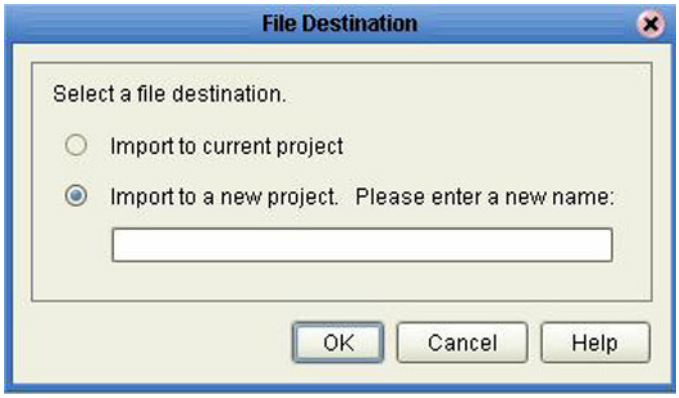
9.5 Building a Web Server

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 119), select **Import to a new Project**, enter the name of the Project, and click **OK**.

Figure 132 File Destination Dialog Box



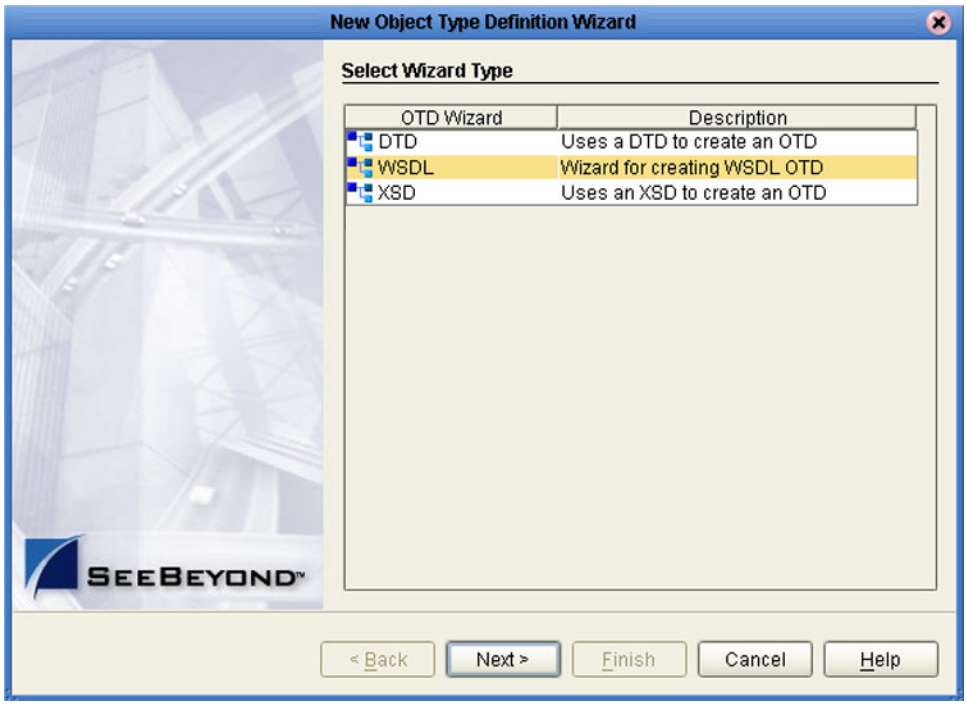
5. After the import has successfully completed, select the **Repository** in the Project Explorer and click **Refresh All from Repository**.

To build a Web Server Using the ICAN Suite

1 In Project Explorer, create a new OTD.

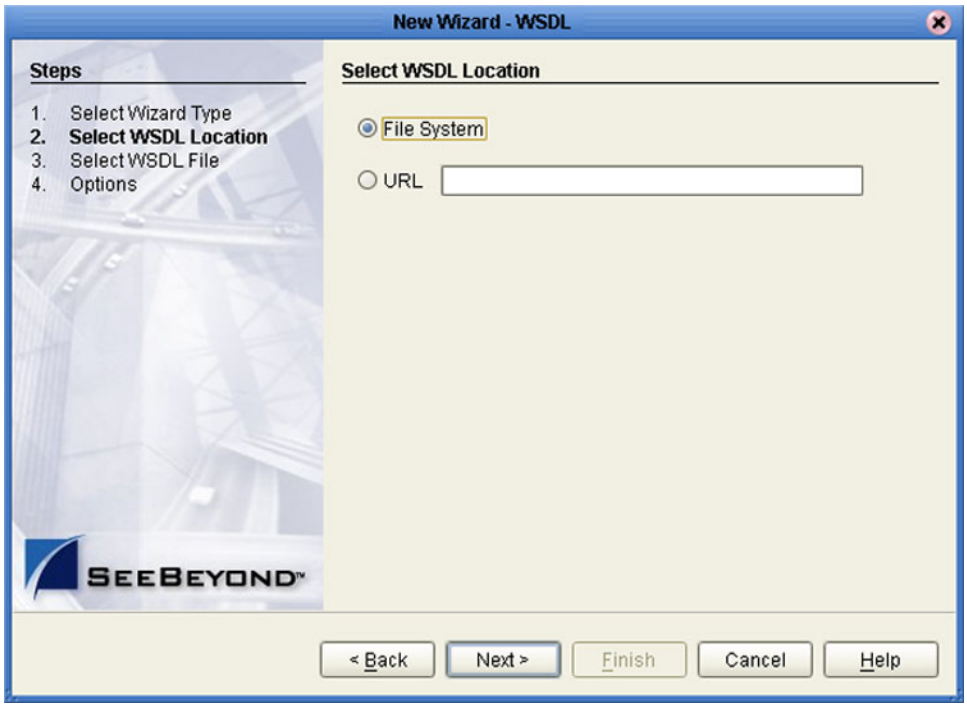
2 Select the WSDL OTD Wizard (see Figure 120).

Figure 133 Select WSDL Wizard



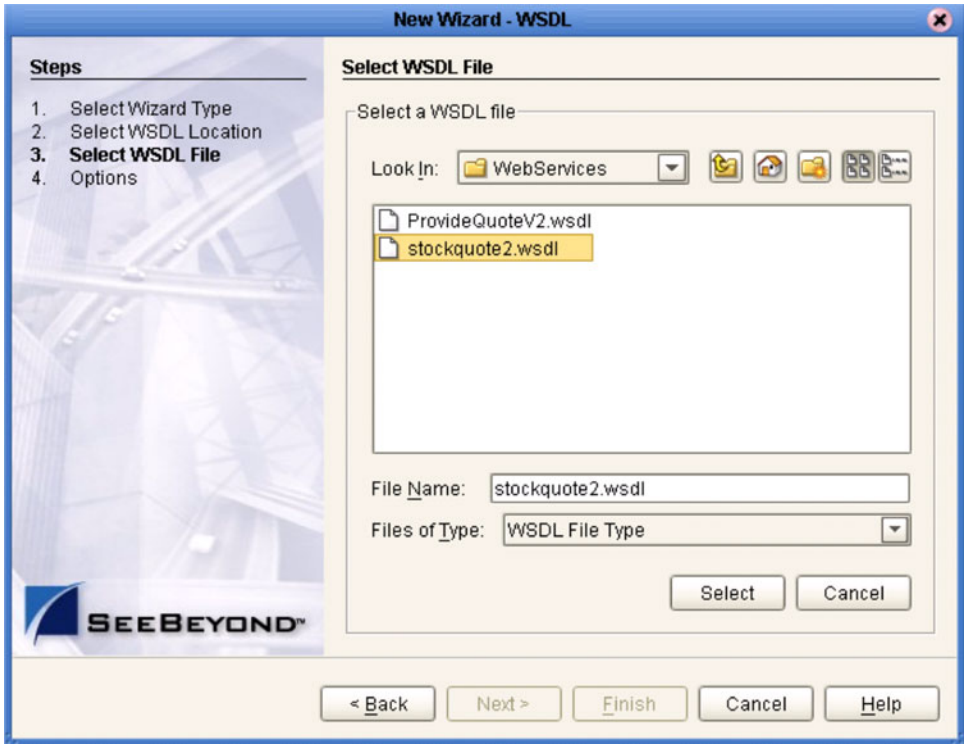
3 Select the WSDL file location (see Figure 134).

Figure 134 Select File Location



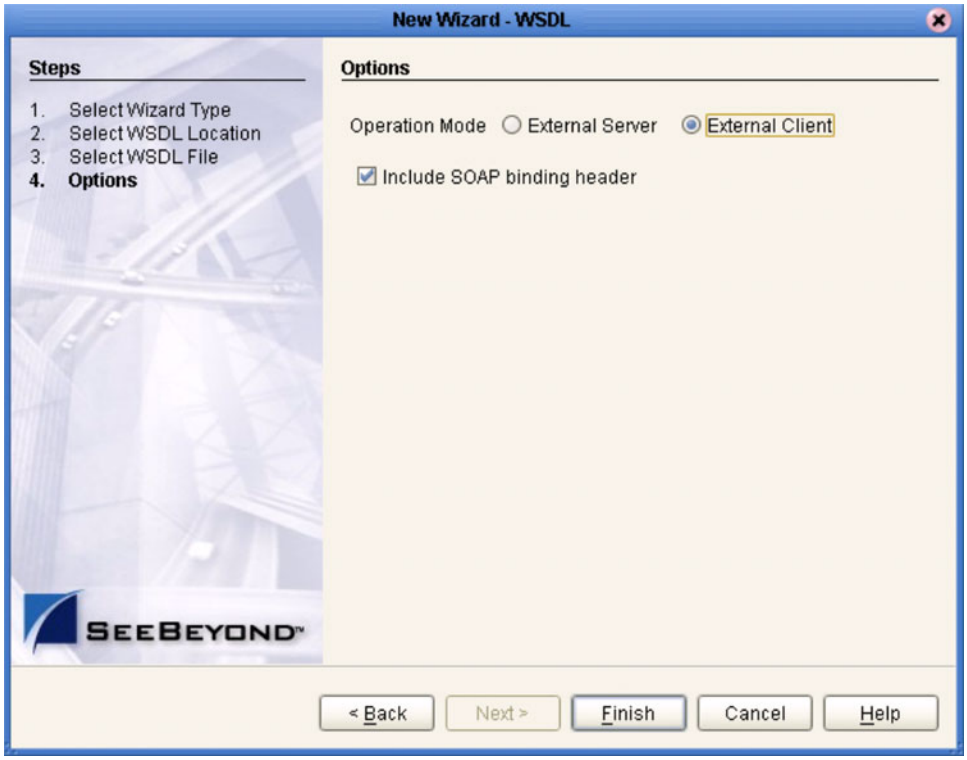
4 Select the WSDL file you want to use for the OTD (see Figure 135).

Figure 135 Select WSDL File



5 For a Web server, select External Client (see Figure 136).

Figure 136 Select External Client

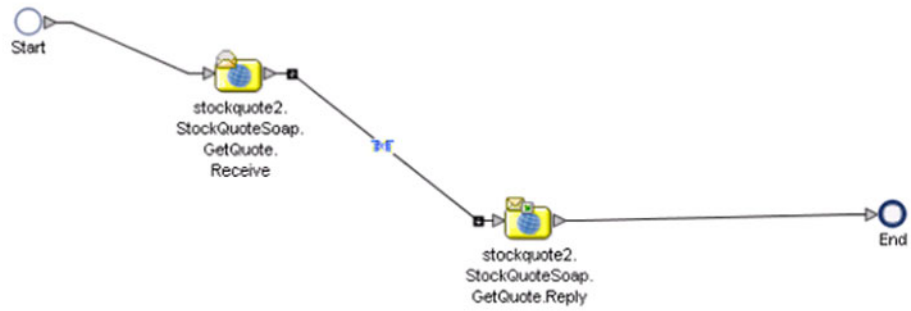


6 Click **Finish**.

9.5.1 eInsight Business Process

The example business process, developed in eInsight Business Process Manager, is shown in Figure 137 (see the *eInsight Business Process Manager User's Guide* for details).

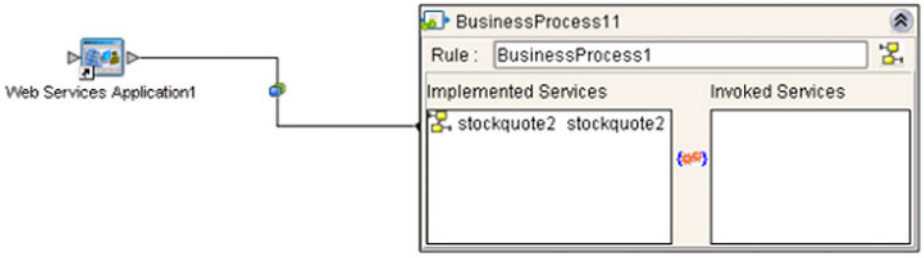
Figure 137 Web Server Business Process



9.5.2 eGate Project

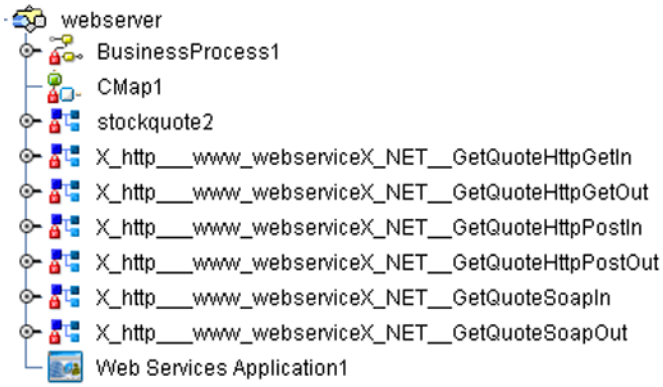
The business process is connected as shown in Figure 138, using the Enterprise Designer Connectivity Map Editor.

Figure 138 Connectivity Map



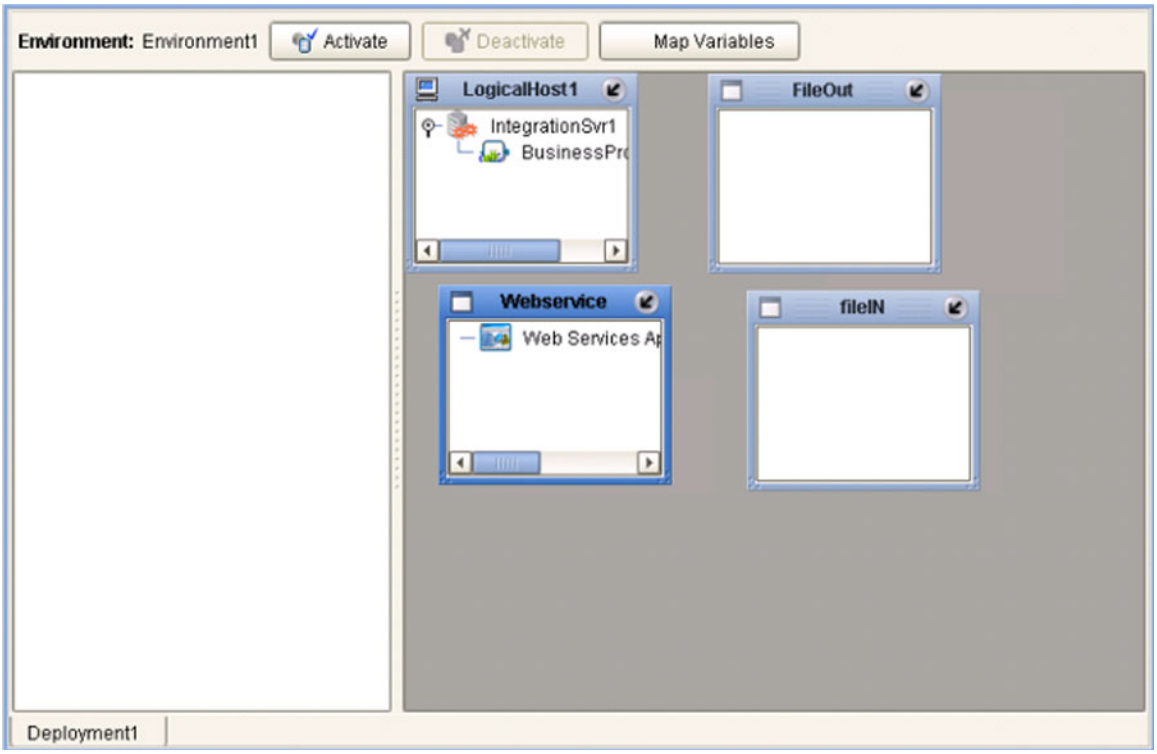
The Web server example Project appears in the Project Explorer as shown in Figure 139.

Figure 139 Web Server Example Project



The Project is deployed as shown in Figure 140.

Figure 140 Project Deployment



Glossary

BI
Business integration (also Business Intelligence).

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

Also 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"**). 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 40 lists terminology that is new with eGate release 5.0 along with equivalent terms from eGate release 4.x, where applicable.

Table 40 Terminology Cross-Reference

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

Index

I

J

L

M

N

O

P

Q

R

S

T

V

W

X