SeeBeyond ICAN Suite

eGate Integrator Tutorial

Release 5.0.5

SEEBEYOND®

SeeBeyond Proprietary and Confidential

Version 20041118160439.

Contents

Chapter 5

List of Figures

Chapter 1

Introduction

Welcome to the *eGate Integrator Tutorial*. The Tutorial must be run on a Windows system. The Repository and Logical Host may be installed on any supported platform, but the Enterprise Designer must be installed and run on a Windows computer.

This document contains tutorials to help you get started with eGate Integrator. These tutorials are very basic and are intended for new eGate users. Sample files are provided along with the documentation. These tutorials are "end-to-end" scenarios. In doing these tutorials the user creates, deploys, and runs Projects to produce output which can be verified.

What's in This Chapter

1.1 About this Document

This document contains the procedures to complete two scenarios that demonstrate some of the basics of the eGate Integrator product. You can use the sample data provided to go through these two scenarios and learn the basics fast.

1.1.1 What's in this Document

- **Chapter 1 "Introduction"** overviews the contents of the *eGate Integrator Tutorial*, describes the writing conventions used in this document, and provides a complete list of related eGate Integrator documentation.
- **Chapter 2 "Features of the Enterprise Designer"** introduces the Enterprise Designer Components, Menu Bar, Enterprise Explorer, and Project Editor.
- **Chapter 3 "Building a Project - Scenario1"** provides a simple "end-to-end" scenario where you perform a data concatenation.
- **Chapter 4 "Building a Project - Scenario2"** provides an "end-to-end" scenario where you perform data transformation, involving a multiplication operation.

- Chapter 5 **"Enterprise Manager Overview"** provides an overview of the Enterprise Manager which is used to view and monitor projects dynamically.
- **"Glossary"**.
- **"Index"**.

1.1.2 Intended Audience

This guide is intended for experienced computer users who have the responsibility of helping to set up and maintain a fully functioning ICAN Suite system. This person must also understand any operating systems on which will be installed , and must be thoroughly familiar with Windows-style GUI operations.

1.1.3 Document Conventions

The following conventions are observed throughout this document.

Table 1 Document Conventions

Text	Convention	Example
Names of buttons, files, icons, parameters, variables, methods, menus, and objects	**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, code samples	`Fixed` font. Variables are shown in ***bold italic***.	`bootstrap -p` ***password***
Hypertext links	**Blue** text	See **"Document Conventions" on page 11**
Hypertext links for Web addresses (URLs) or email addresses	**Blue underlined** text	**http://www.seebeyond.com** **docfeedback@seebeyond.com**

1.1.4 Screenshots Used in this Document

The screenshots in this document should be current with the product release shown on the title page of this document. Depending on what you have installed, it is possible that some of your screens could look slightly different than some screenshots in this document.

1.2 Related Documents

The following SeeBeyond documents provide additional information about eGate Integrator:

- *SeeBeyond ICAN Suite Installation Guide*
- *eGate Integrator System Administration Guide*
- *eGate Integrator Release Notes*
- *eGate Integrator User's Guide*
- *SeeBeyond ICAN Suite Primer*

1.3 ICAN Suite Installation Requirements

To simplify these examples, this tutorial assumes you have all of the following eGate components installed on a single Windows system:

- eGate Repository
- Logical Host
- Enterprise Designer
- eGate product
- File eWay product

Refer to the *SeeBeyond ICAN Suite Installation Guide* for system requirements and installation instructions. See also: **Chapter 3**, **ICAN Files Required for the Tutorials** on page 20.

1.4 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

1.5 SeeBeyond Documentation Feedback

We appreciate your feedback. Please send any comments or suggestions regarding this document to:

docfeedback@seebeyond.com

Chapter 2

Features of the Enterprise Designer

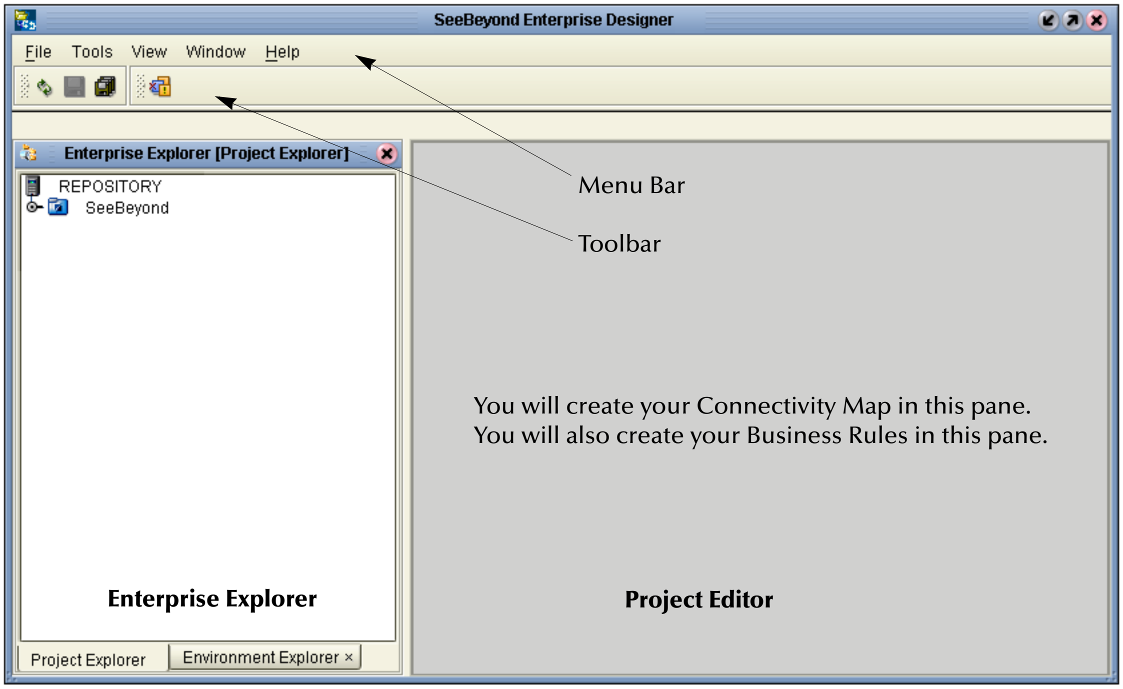
The Enterprise Designer is the graphical user interface (GUI) used to design and implement ICAN 5.0 projects. This chapter overviews the features and interface of the Enterprise Designer window.

What's in This Chapter

2.1 Enterprise Designer Components

The Enterprise Designer is used to create and configure the components of an eGate Project. Each component of this interface is identified in Figure 1.

Figure 1 SeeBeyond Enterprise Designer



You will create your Connectivity Map in this pane.
You will also create your Business Rules in this pane.

Note: *This chapter provides a high-level overview of the Enterprise Designer GUI. Refer to the eGate Integrator User's Guide for a more detailed description of the menu bar, toolbar, Enterprise Explorer, and Enterprise Designer.*

2.2 Menu Bar

The menu bar shown below in Figure 2 provides access to a series of menu commands used to build and manage a Project.

Figure 2 Enterprise Designer Menu Bar

The function of each menu is briefly described below:

- **File**—Lists options for save, save all, and exit.
- **Tools**—Lists options for Impact Analyzer, Update Center and Setup Options.
- **View**—Lists options to view the Environment Explorer and Project Explorer.
- **Window**—Lists options for the various window displays, cascade, tile, etc.
- **Help**—online help for all installed modules

2.3 Enterprise Explorer

The Enterprise Explorer organizes all of the components of a Project into a series of folders and contains the following two tabs:

- **Project Explorer**—Logical configurations designed to help solve a business problem. This branch includes all the components of an Enterprise Designer Project, including Connectivity Maps, Services, Object Type Definitions (OTD), and Deployment Profiles.
- **Environment Explorer**—Collections of logical hosts and external systems capable of hosting eGate components and information about external systems, which may be involved with an eGate configuration. You will go into the Environment Explorer when you are ready to deploy and run your project.

2.4 Project Editor

The Project Editor is where you use GUI tools (or even enter code directly) to create a Project. This part of the Enterprise Designer is empty when you start a new Project. However, as you work through the tutorial, the Project Editor quickly fills with components and graphical structures representing the various stages of the Project. The types of windows in the Project Editor area include:

- **Connectivity Map**—Contains business logic components, such as Services, Topics, queues, and eWays, that you include in the structure of a Project.
- **Collaboration Editor (Java)**—Contains the business rules defined in Java.
- **Collaboration Editor (XSLT)**—Contains the business rules defined in XSLT.
- **OTD Editor**—Edits and tests the OTDs (Object Type Definitions)
- **Deployment Profile Editor**—Edits the deployment profile
- **Environment**—Collection of physical resources (logical hosts and external systems)

Chapter 3

Building a Project - Scenario1

This Tutorial provides step-by-step procedures for creating and testing an eGate Project. This Project addresses a very simple business challenge where an input TimeCard file is read and a workfile is output.

What's in This Chapter

3.1 Business Challenge

The Project described in this chapter provides a solution to the following business challenge:

In this very simple scenario **FirstName** and **LastName** from the input file are concatenated into the field **FullName** in the output file. **JobTitle**, **HoursWorked** and **Rate** are disregarded in Scenario1.

- The input data in this system is in XML format with six fields: **EmployeeNumber**, **LastName**, **FirstName**, **JobTitle**, **HoursWorked**, and **Rate**.

- The output workfile in XML format contains the following fields: **FullName**, and **EmpNo**.

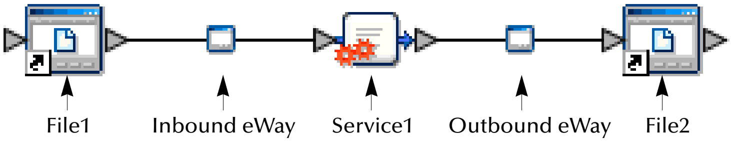
3.2 Project Overview

Create a basic eGate Project to meet the business challenge described in **"Business Challenge" on page 16**.

3.2.1 Project Description

The finished eGate Project contains components used to move the data throughout the Project and publish the data in the appropriate format for the output workfile.

Figure 3 Project Connectivity Map



These components perform the following functions:

- **File1**: The Input file *External Application* contains the Input XML data file.
- **Inbound eWay**: The inbound eWay polls a specified location on the file system for the input XML data file.
- **Service1**: This Service performs a concatenation of the name fields.
- **Outbound eWay**: This eWay publishes the finished output data to the file system in the format appropriate for the output work file.
- **File2**: This is the output work file.

3.2.2 Process Overview

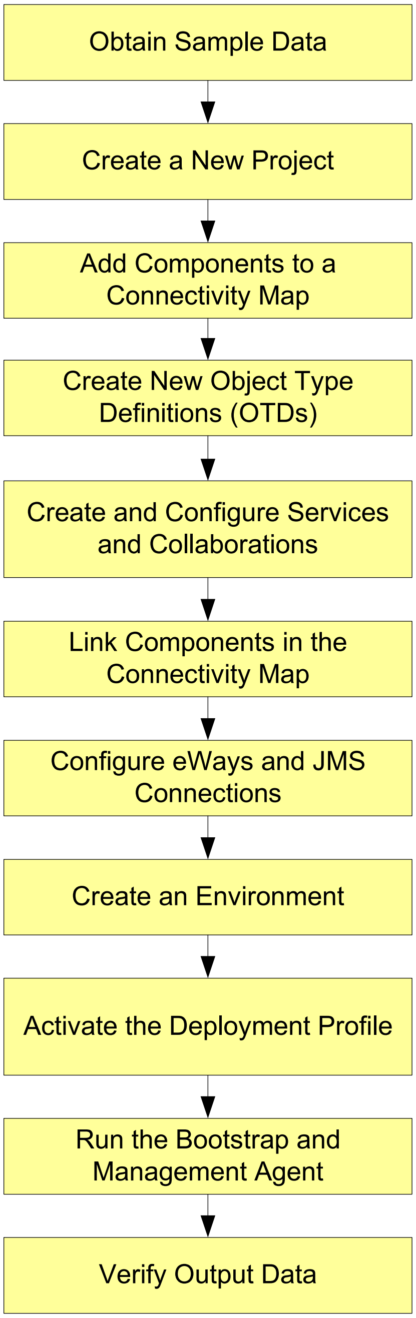
Complete the following general steps:

1. **"Sample Data" on page 20**
2. **"Create New Project" on page 23**
3. **"Create Connectivity Map" on page 25**
4. **"Add Components to the Connectivity Map" on page 26**
5. **"Create New Object Type Definition" on page 28**
6. **"Configure the Services" on page 32**

7. **“Apply the Business Rules” on page 39**
8. **“Apply the Collaboration” on page 46**
9. **“Configure the eWays” on page 48**
10. **“Create Environment” on page 52**
11. **“Run the Bootstrap and Management Agent” on page 57**
12. **“Verify Output Data” on page 61**

3.2.3 Project Flow Diagram - Scenario1

Figure 4 Project Flowchart - Scenario1


Obtain Sample Data
Create a New Project
Add Components to a Connectivity Map
Create New Object Type Definitions (OTDs)
Create and Configure Services and Collaborations
Link Components in the Connectivity Map
Configure eWays and JMS Connections
Create an Environment
Activate the Deployment Profile
Run the Bootstrap and Management Agent
Verify Output Data

3.3 ICAN Files Required for the Tutorials

Please refer to the *SeeBeyond ICAN Suite Installation Guide* for instructions about uploading files to the Repository. The following files are required to run the tutorials:

- License.sar
- eGate.sar
- ProductManifest.xml
- FileeWay.sar
- eGateDocs.sar (PDF and sample files)

The Tutorial assumes the software is installed in C:\ican50. Your installation may be different.

Although not required for the Enterprise Manager, you will have more features for GUI displays if you use the SVG Viewer. Use the **EnterpriseManager-Plugin-Win98ToXP.sar** (required for the Adobe SVG Viewer plug-in for Windows). Enterprise Manager is not required for the tutorials.

3.3.1 Repository, Logicalhost and Environment

You may use the same Repository, Logicalhost and Environment for both tutorials, Scenario1 and Scenario2. You should use different names for the deployments. Scenario1 should be deployed as Deployment1 and Scenario2 should be deployed as Deployment2.

3.3.2 File Names

The file names used in Scenario1 (Project1) should be different than the file names used in Scenario2 (Project2). See the following section, Sample Data.

3.4 Sample Data

This Project uses the sample files found in the Project1 folder of the **eGate Tutorial Sample** that you download using the steps below. The files used in Scenario1 are: **s1input_multiple.dtd**, **s1output_multiple.dtd**, and **TimeCard-Multiple.xml**. (The “s1” in the file names stands for Scenario1.)

3.4.1 Download the Sample Files

Use the Sample Files

1. Logon to the **Enterprise Manager**.
2. Click the **Documentation** tab.

3 Click **eGate Integrator**.

4 Double-click and save the **eGate Tutorial Sample Files**.

Note: *You can also create the files you need. See sections 3.4.2, 3.4.3, and 3.4.4 below, but this shouldn't be necessary.*

3.4.2 Create Sample Input XML File

Use a text editor such as Microsoft Notepad to create the input XML file. You can also copy and paste from this document (PDF) using the text select tool from within Adobe Acrobat.

Create XML file

Your XML file should look similar to the structure and syntax below. In this scenario of the Tutorial there are four records.

```
<?xml version="1.0" encoding="UTF-8" ?>
        <Employees>
                <Employee>
                        <EmployeeNumber>100</EmployeeNumber>
                        <LastName>Smith</LastName>
                        <FirstName>Jane</FirstName>
                        <JobTitle>Manager</JobTitle>
                        <HoursWorked>40</HoursWorked>
                        <Rate>55</Rate>
                </Employee>
        <Employee>
                        <EmployeeNumber>123</EmployeeNumber>
                        <LastName>Chang</LastName>
                        <FirstName>Ling</FirstName>
                        <JobTitle>Manager</JobTitle>
                        <HoursWorked>40</HoursWorked>
                        <Rate>60</Rate>
                </Employee>
        <Employee>
                        <EmployeeNumber>118</EmployeeNumber>
                        <LastName>Jamison</LastName>
                        <FirstName>Dick</FirstName>
                        <JobTitle>Clerk</JobTitle>
                        <HoursWorked>40</HoursWorked>
                        <Rate>22</Rate>
                </Employee>
        <Employee>
                        <EmployeeNumber>144</EmployeeNumber>
                        <LastName>Nakayama</LastName>
                        <FirstName>Takeshi</FirstName>
                        <JobTitle>Supervisor</JobTitle>
                        <HoursWorked>40</HoursWorked>
                        <Rate>35</Rate>
                </Employee>
        </Employees>
```

1 In Windows Explorer, create a directory for the data files. For this scenario, use **c:\eGateData\Project1**.

2 In Notepad, create an XML file containing the data shown above.

3 Save the file as **TimeCard_Multiple.xml** in the **c:\eGateData\Project1** directory.

3.4.3 Create Input (DTD) File

The **s1input_multiple.dtd** file is used as a representation of the data in the TimeCard system.

The **s1input_multiple.dtd** file uses the following elements:

- **EmployeeNumber**
- **LastName**
- **FirstName**
- **JobTitle**
- **HoursWorked**
- **Rate**

Create Input (DTD) File

Your input (DTD) should look similar to the structure and syntax below.

1 In Notepad, create an XML file containing the following information:

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Employees(Employee+)>
<!ELEMENT Employee(EmployeeNumber, LastName, FirstName, JobTitle,
HoursWorked, Rate)>
<!ELEMENT EmployeeNumber(#PCDATA)>
<!ELEMENT LastName(#PCDATA)>
<!ELEMENT FirstName(#PCDATA)>
<!ELEMENT JobTitle(#PCDATA)>
<!ELEMENT HoursWorked(#PCDATA)>
<!ELEMENT Rate (#PCDATA) >
```

2 Save the file as **s1input_multiple**.**dtd** in the **c:\eGateData\Project1** directory.

3.4.4 Create Output (DTD) File

The **s1output_multiple.dtd** file is used as a representation of the data in the output workfile.

The **s1output_multiple.dtd** file uses the following elements:

- **FullName**
- **EmpNo**

Create Output (DTD) file

Your output (DTD) should look similar to the structure and syntax below.

1 In Notepad, create an XML file containing the following information:

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Employees(Employee+)>
<!ELEMENT Employee(FullName, EmpNo)>
<!ELEMENT FullName(#PCDATA)>
<!ELEMENT EmpNo(#PCDATA)>
```

2 Save the file as **s1output_multiple**.**dtd** in the **c:\eGateData\Project1** directory.

3.5 Create New Project

There are many ways to create a Project and link the modules. This Tutorial presents one scenario. Begin by creating and naming a Project in the Enterprise Designer.

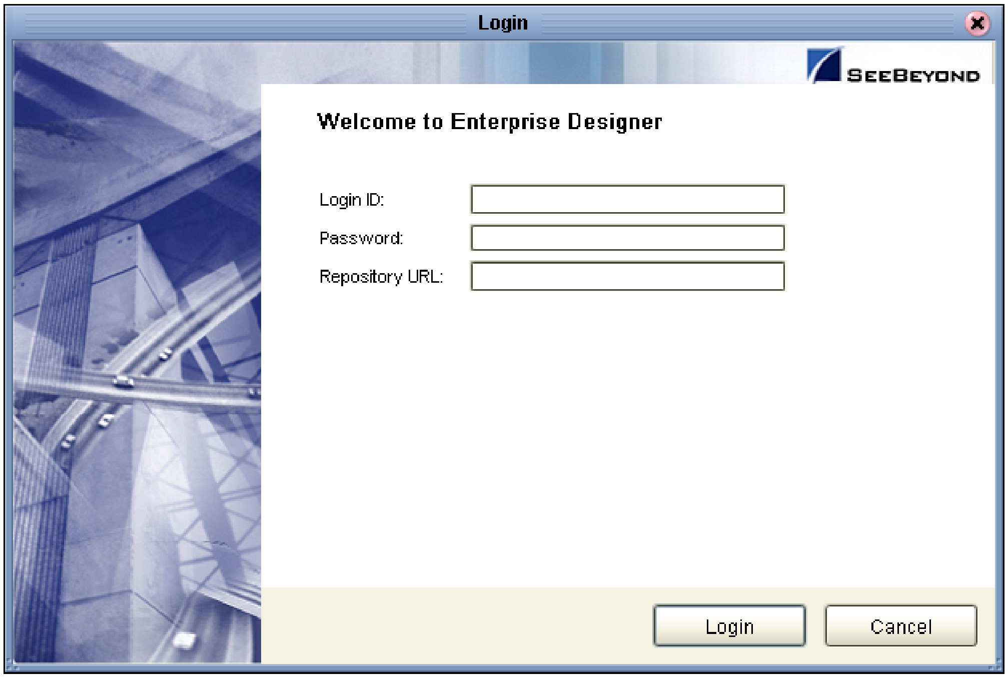
Start Enterprise Designer

Start the Enterprise Designer according to your organization's instructions.

1 Start the Enterprise Designer by executing **runed.bat** in your ican50 folder, for example: (**c:\ican50\edesigner\bin**).

The Enterprise Designer Login dialog box appears.

Figure 5 Enterprise Designer Login



2 Type your **Login ID** and **Password**. Click **Login** to start the Enterprise Designer.

The **Repository URL** value will default to the localhost information you supplied when you downloaded the Enterprise Designer, i.e. http://<localhost>:12000/Repository_tutorial.

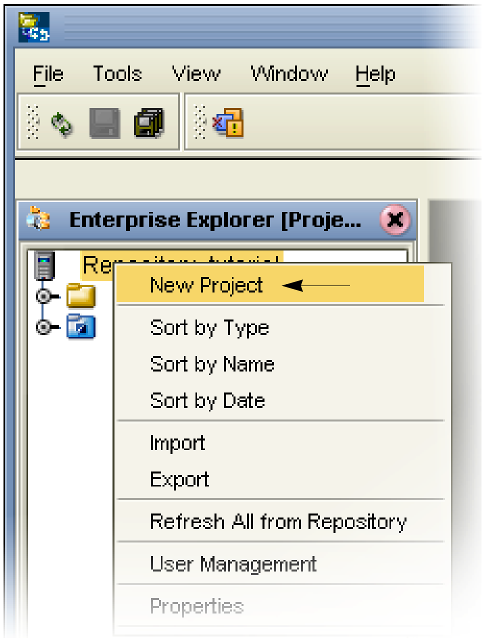
Note: *If your login window doesn't appear or if you get an invalid Username or Password error, make sure your Repository is running. Also, be aware that the Username and Password are case sensitive.*

Create Project

A Project is a collection of logical components, configurations, and files that are used to solve a business problem.

1 In the Enterprise Explorer pane of the Enterprise Designer, right-click the Repository name (computer icon). Click **New Project**.

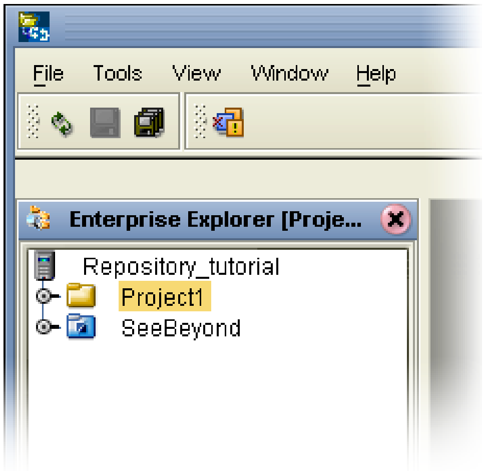
Figure 6 Create a Project



2 Accept the default **Project1** as the file name and press **Enter**.

3 The **Project1** appears in the Explorer pane on the left side of the window.

Figure 7 Project Folder: Project1



3.6 Create Connectivity Map

Project1 needs a Connectivity Map to define the flow between the TimeCard system and the work file output.

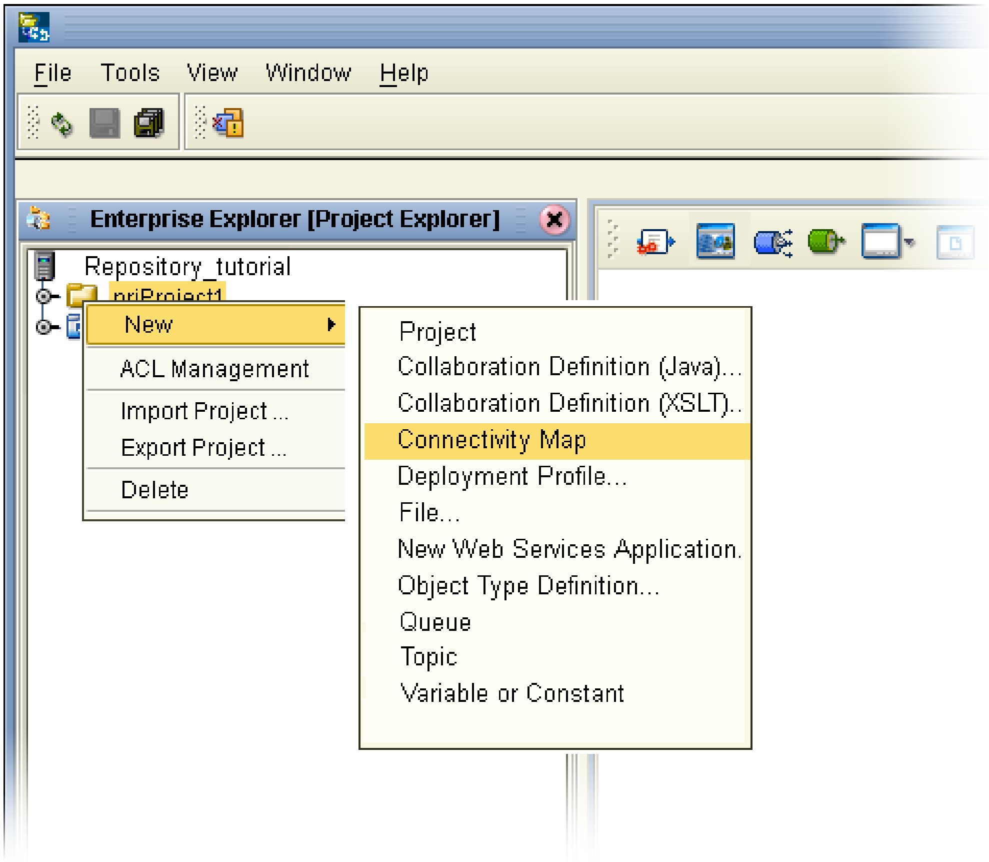
This section describes how to create the Connectivity Map for **Project1**.

Name Connectivity Map

A Connectivity Map contains business logic and routing information about the data transmission. The Connectivity Map is a GUI (graphical representation) of the project.

1 In the Enterprise Explorer, right-click the **Project1** icon. Click **New** and **Connectivity Map**.

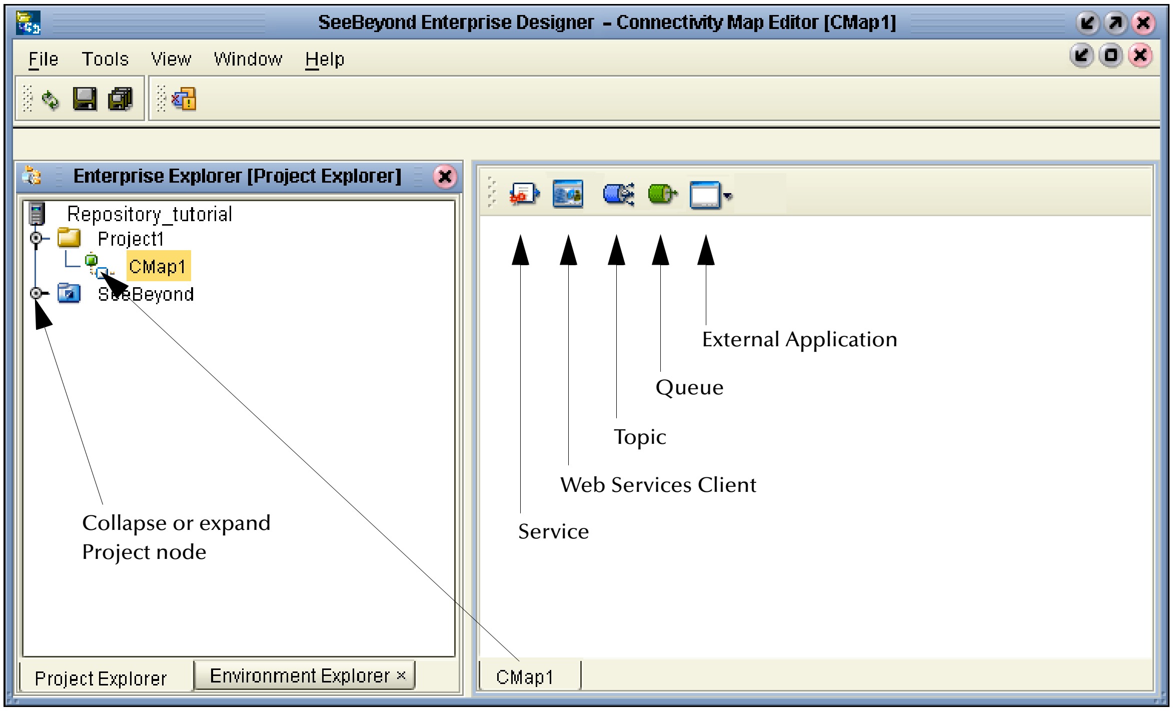
Figure 8 Create a Connectivity Map



2 You may give your Connectivity Map a name. In this Tutorial, accept the default name, **CMap1**.

CMap1 is added to your **Project1**, as shown in the following figure.

Figure 9 Project1 with a Connectivity Map (CMap1)



Note: *The **CMap1** Connectivity Map appears as an icon in the Project Explorer. A **CMap1** tab is also added at the bottom of the Connectivity Map.*

3.6.1 Add Components to the Connectivity Map

In this section, add the components to the Connectivity Map. Later link the components as shown in **Figure 3 on page 17**.

In this section, add the following components to the Connectivity Map:

- **File1** (File External Application)
- **Service1**
- **File2** (File External Application)

Populate Connectivity Map

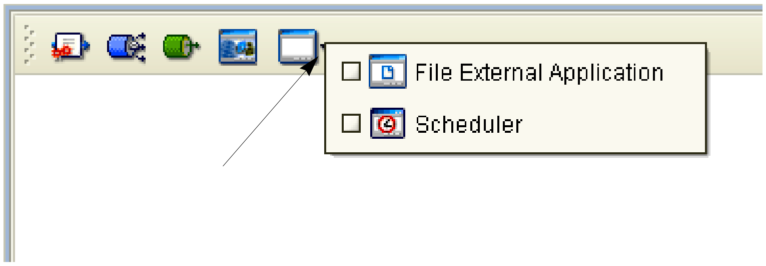
Place an input file (External Application) in your Connectivity Map.

1. On the Connectivity Map toolbar, click the External Application arrow (shown in Figure 10) to display the list.
2. Click **File External Application**.

Note: *If **File External Application** doesn't appear in the drop-down list, you may need to install **FileeWay.sar**.*

This adds a new **File** icon to the toolbar.

Figure 10 External Application Selection



3 Drag the **File** icon from the toolbar into the left side of the Connectivity Map.

This creates a new File External Application called **File1**.

Note: *In this scenario accept the default names. (In Scenario2 you will name the components.)*

Create Service1

Create a Service to contain binding information about connecting the input to the output.

1 On the Connectivity Map toolbar, drag the **Service** icon into the Connectivity Map to the right of the **File1** icon.

This creates a new Service called **Service1**.

Note: *To reposition a component in the Connectivity Map, click the icon and drag it to a new location while holding down the mouse button.*

Create Output

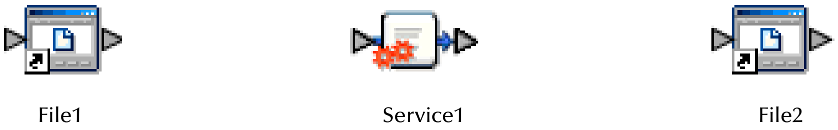
Place an output file (External Application) in your Connectivity Map.

1 Drag another **File** icon into the Connectivity Map, and place it to the right of the **Service1** icon.

This creates a new External Application file.

2 Click **Save** (CTRL+S).

Figure 11 Connectivity Map



Note: *Link these objects later.*

3.7 Create New Object Type Definition

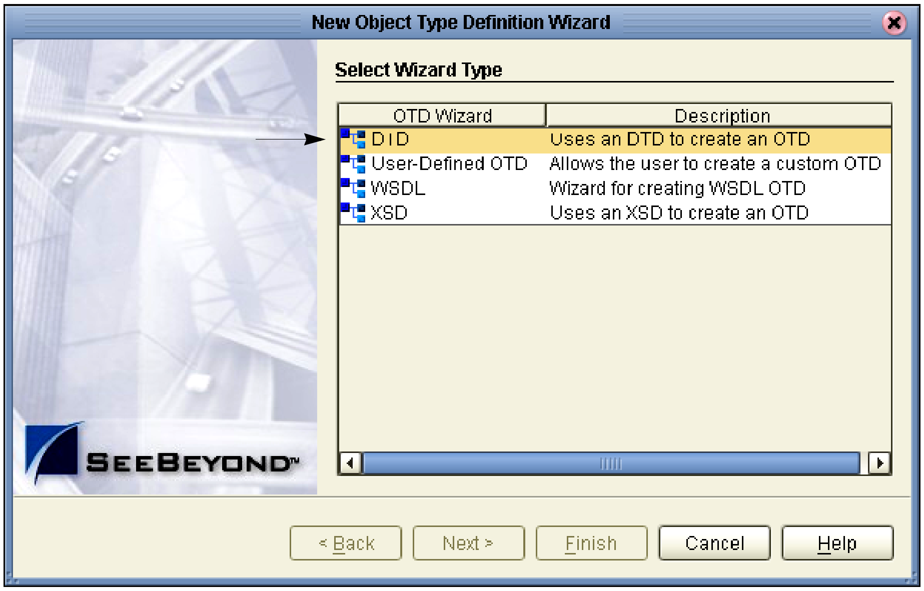
Create an object type definition (OTD) to define your data object.

Create Object Type Definition

An OTD contains the structure and rules that define an object.

1. Right-click **Project1**.
2. Click **New** and **Object Type Definition**.

Figure 12 OTD Wizard Selection



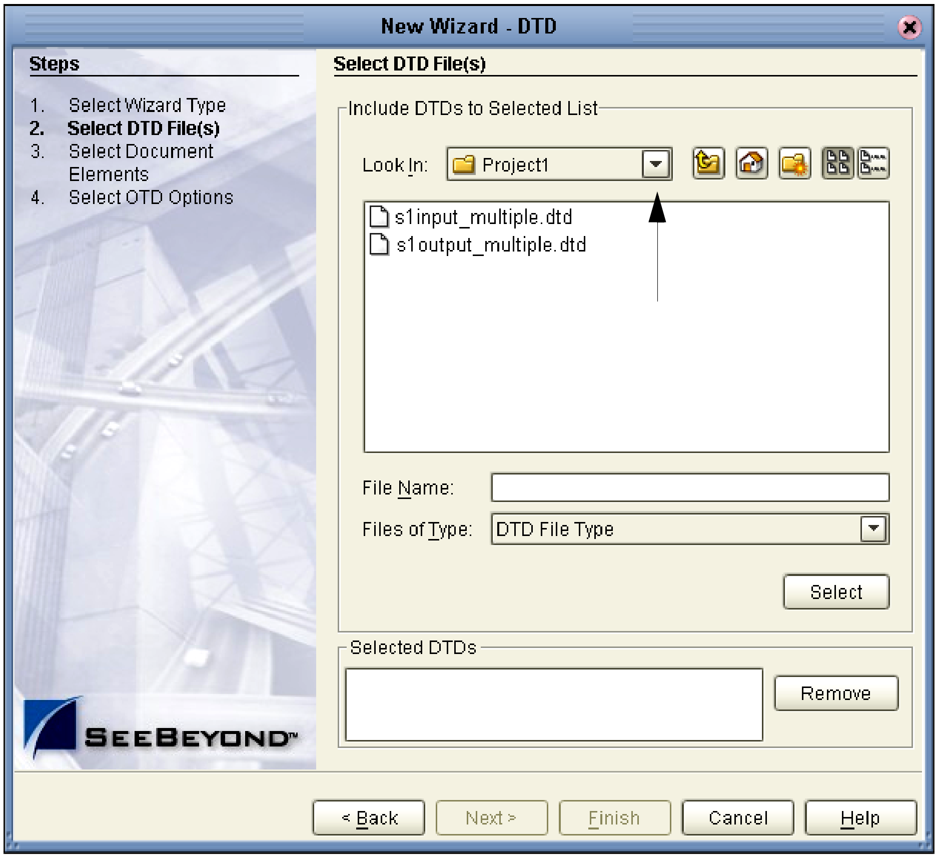
Select Wizard Type

OTD Wizard	Description
DID	Uses an DTD to create an OTD
User-Defined OTD	Allows the user to create a custom OTD
WSDL	Wizard for creating WSDL OTD
XSD	Uses an XSD to create an OTD

3. Select the **DTD** item.
4. Click **Next**.
5. Click the **drop-down** arrow.
6. See the following figure
7. Navigate to the sample data folder. For example, **c:\eGateData\Project1**.

   See the following figure.

Figure 13 Select DTD Files



8 Highlight **s1input_multiple.dtd**. Click **Select** (or double-click). The **s1** in the name stands for Scenario1.

9 Highlight **s1output_multiple.dtd**. Click **Select** (or double-click).

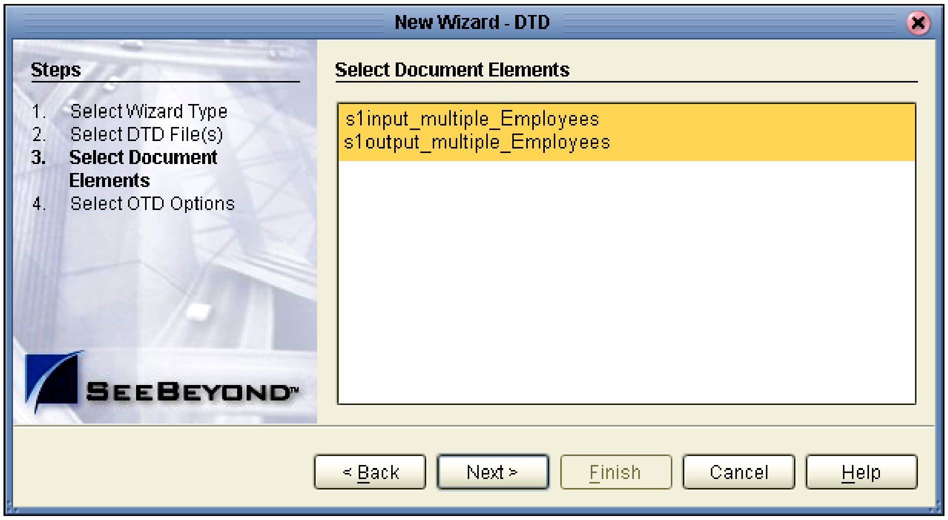
The **s1input_multiple.dtd** and **s1output_multiple.dtd** should appear in the Selected DTDs.

10 When both **s1input_multiple.dtd** and **s1output_multiple.dtd** appear in the Selected DTDs, click **Next**.

Note: *The **Up one Level** icon (number 1 in Figure 13 above) is used to navigate to a previous screen; this is useful if you have made an incorrect selection.*

Note: *As you progress through the steps, notice that the highlighted text in the left side of the New Wizard DTD window changes. In the previous step **Select DTD File(s)** was bolded. Now **Select Document Elements** is bolded, See Figure 14.*

Figure 14 Select Document Elements



Steps

1. Select Wizard Type
2. Select DTD File(s)
3. **Select Document Elements**
4. Select OTD Options

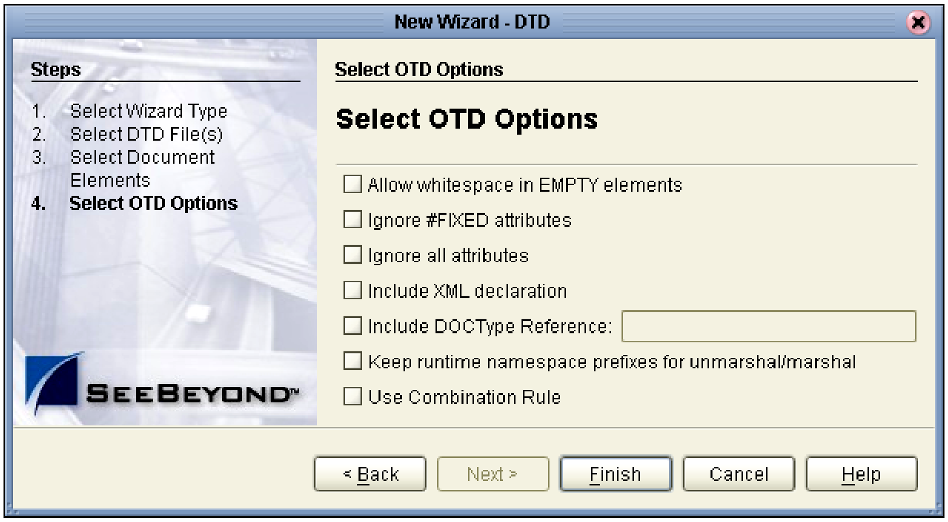
Select Document Elements

s1input_multiple_Employees
s1output_multiple_Employees

11 Select both **s1input_multiple_Employees** and **s1output_multiple_Employees** (hold down the CTRL key while selecting).

12 Click **Next**.

Figure 15 Select OTD Options



Steps

1. Select Wizard Type
2. Select DTD File(s)
3. Select Document Elements
4. **Select OTD Options**

Select OTD Options

Select OTD Options

☐ Allow whitespace in EMPTY elements
☐ Ignore #FIXED attributes
☐ Ignore all attributes
☐ Include XML declaration
☐ Include DOCType Reference:
☐ Keep runtime namespace prefixes for unmarshal/marshal
☐ Use Combination Rule

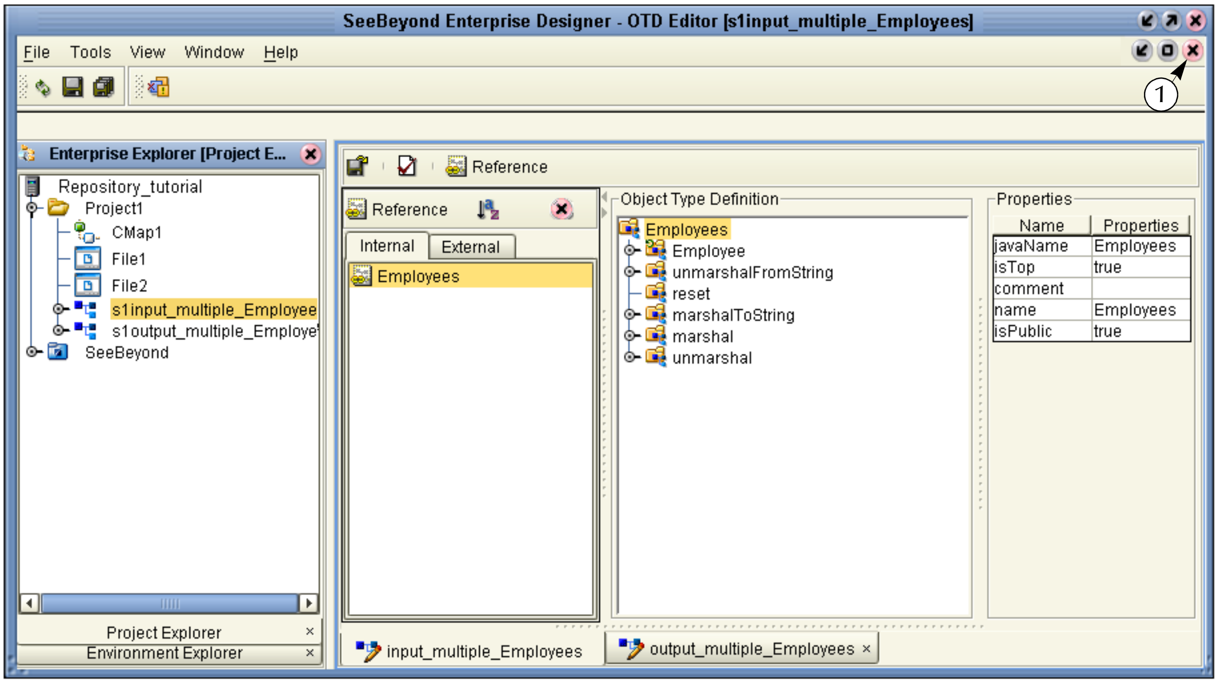
13 Click **Finish** (accepting the defaults).

The OTD Editor window opens. See the following Figure 16.

14 On the **File** menu, click **Save**.

Note: ***Save*** *saves all your changes in your current open editor while* ***Save All*** *saves all changes across all editors, including changes in the environments.*

Figure 16 OTD Editor Input

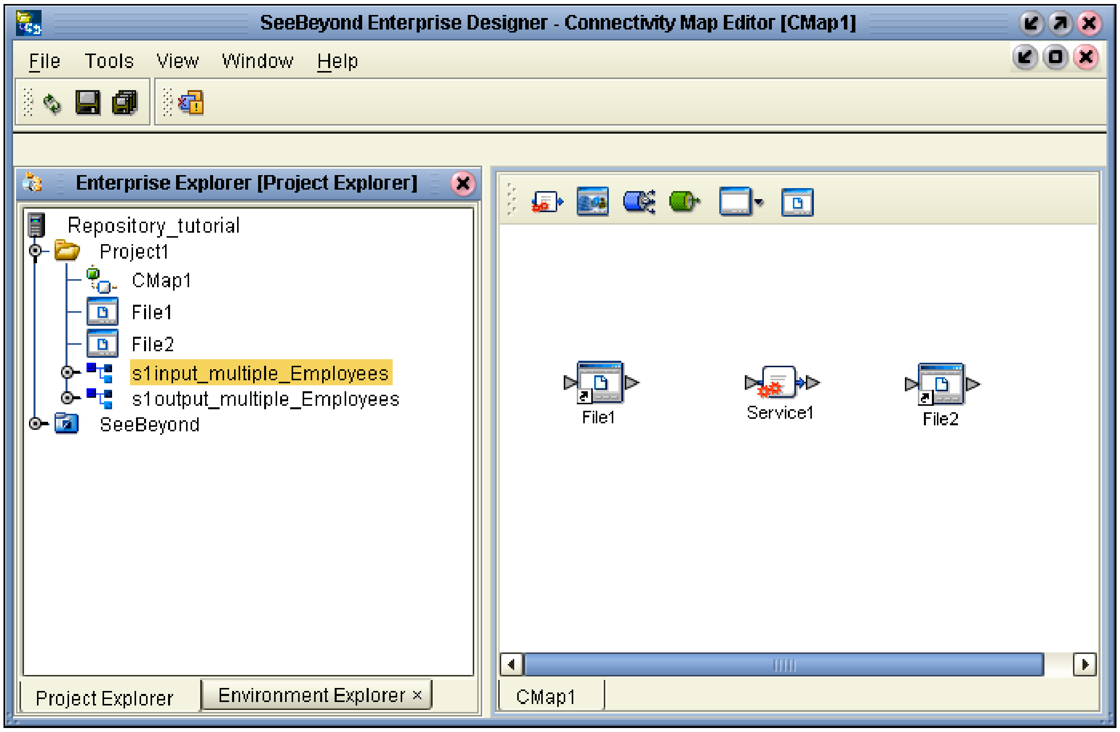


Name	Properties
javaName	Employees
isTop	true
comment	
name	Employees
isPublic	true

15 Click the close window icon (number 1 in Figure 16) to redisplay the Connectivity map. (Or you may double-click the **CMap1** icon in the Project Explorer pane.)

The Connectivity map window appears. See the following figure.

Figure 17 Connectivity Map Window


SeeBeyond Enterprise Designer - Connectivity Map Editor [CMap1]
File
Tools
View
Window
Help
Enterprise Explorer [Project Explorer]
Repository_tutorial
Project1
CMap1
File1
File2
s1input_multiple_Employees
s1output_multiple_Employees
SeeBeyond
Project Explorer
Environment Explorer
File1
Service1
File2
CMap1

3.8 Configure the Services

Use Service wizards in the following steps to configure your Collaboration. A Collaboration describes your data and connections.

3.8.1 Create Service1

Configure a Collaboration (Java) definition.

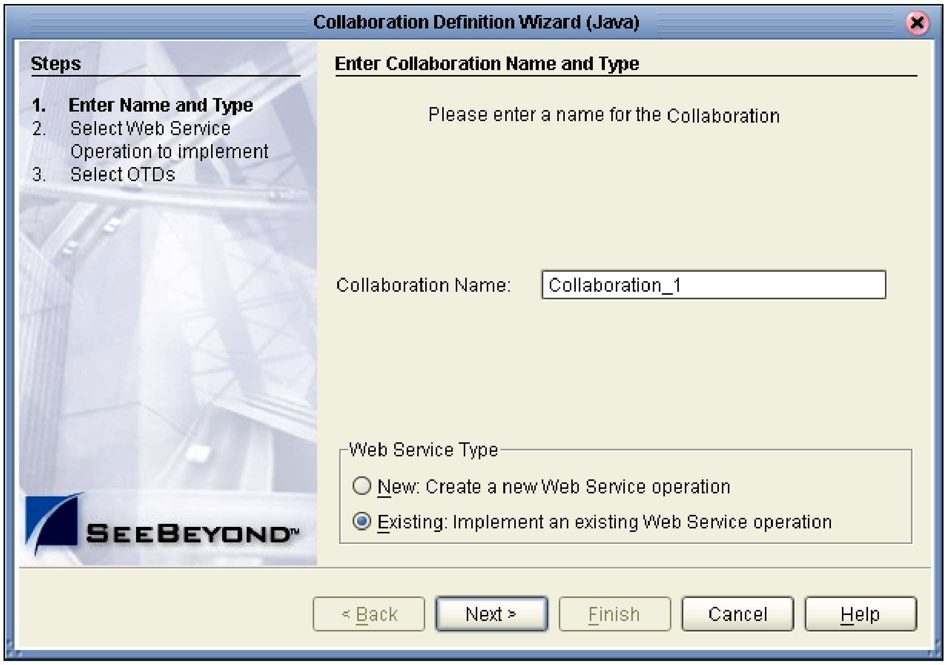
Enter a Collaboration Name

1 Right-click **Project1**.

2 Click **New** and **Collaboration Definition (Java)**.

The Collaboration Definition Wizard (Java) appears.

Figure 18 New Collaboration Definition (Java) Name



Collaboration Definition Wizard (Java)

Steps

1. **Enter Name and Type**
2. Select Web Service Operation to implement
3. Select OTDs

Enter Collaboration Name and Type

Please enter a name for the Collaboration

Collaboration Name: Collaboration_1

Web Service Type

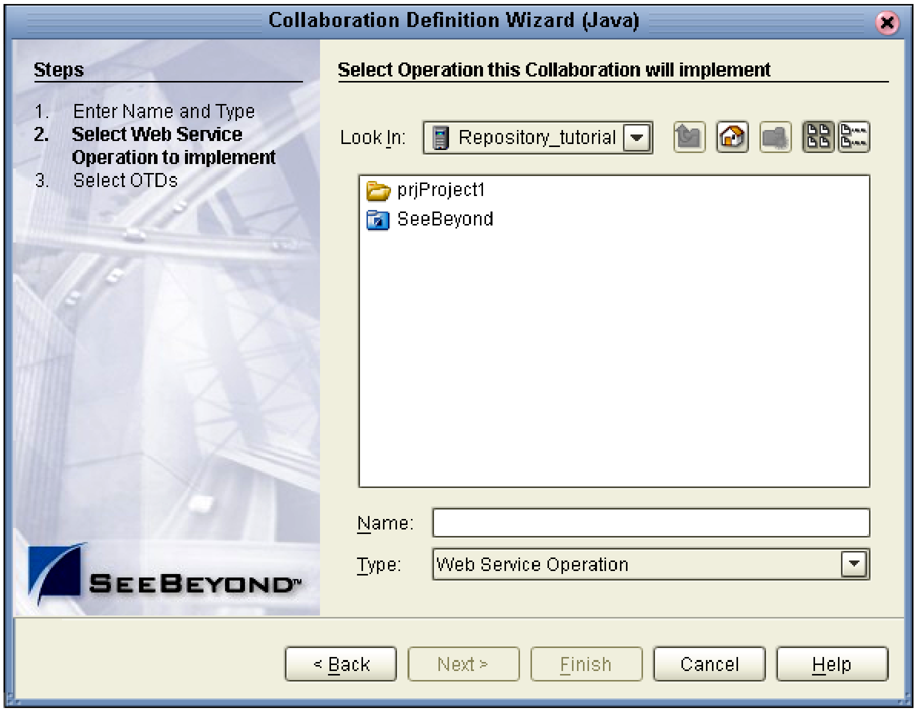
○ New: Create a new Web Service operation

● Existing: Implement an existing Web Service operation

3 Accept the default name for the Collaboration Definition, **Collaboration_1**. (Also accept the Web Service Type: Existing: Implement an existing Web Service operation.)

4 Click **Next**.

Figure 19 New Collaboration Definition Wizard (Java)



Collaboration Definition Wizard (Java)

Steps

1. Enter Name and Type
2. **Select Web Service Operation to implement**
3. Select OTDs

Select Operation this Collaboration will implement

Look In: Repository_tutorial

prjProject1

SeeBeyond

Name:

Type: Web Service Operation

Create Web Service Interface

Web Services enable communication between diverse applications using the Internet. Select a Web Service to “implement” a file-read. Refer to *eGate Integrator User’s Guide* for information about Web Services.

1 Double-click SeeBeyond.

2 Double-click **eWays.** eWays are message end points that connect to an external service provider (system, person, or organization) to enable the sending and receiving of messages.

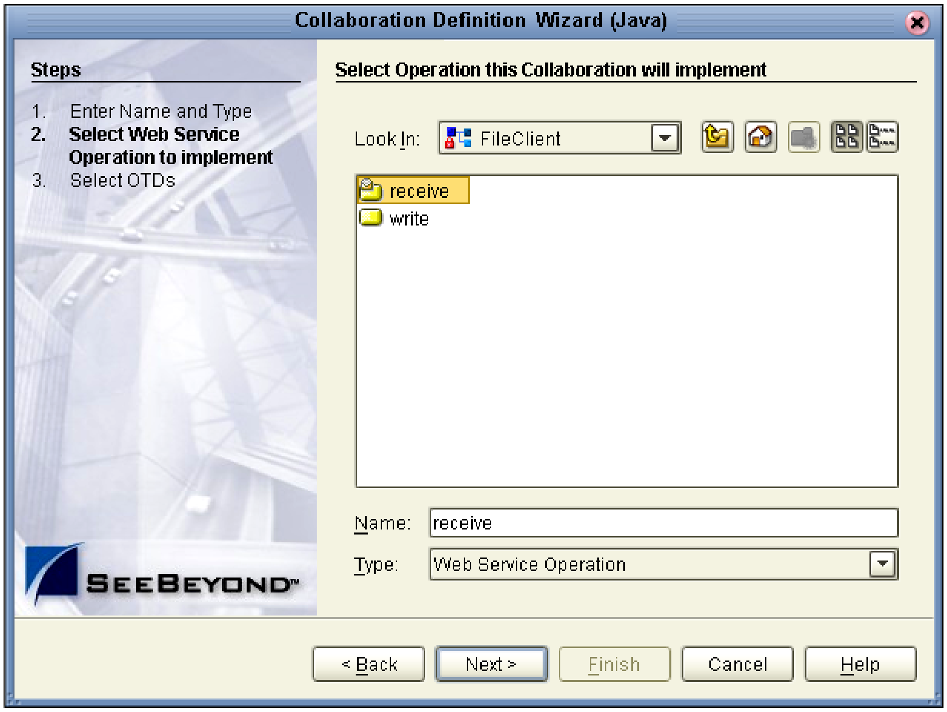
3 Double-click **File**.

4 Double-click **FileClient**.

5 Click **receive**. Notice that **receive** appears in the Name field as shown in the following Figure 20.

(**FileClient.receive** is used for file-based eWays to bring data in or send data out.)

Figure 20 New Collaboration Web Service Interface



Collaboration Definition Wizard (Java)
Steps
1. Enter Name and Type
2. Select Web Service Operation to implement
3. Select OTDs
SeeBeyond
< Back
Next >
Finish
Cancel
Help

Select Operation this Collaboration will implement

Look In: FileClient

receive
write

Name: receive

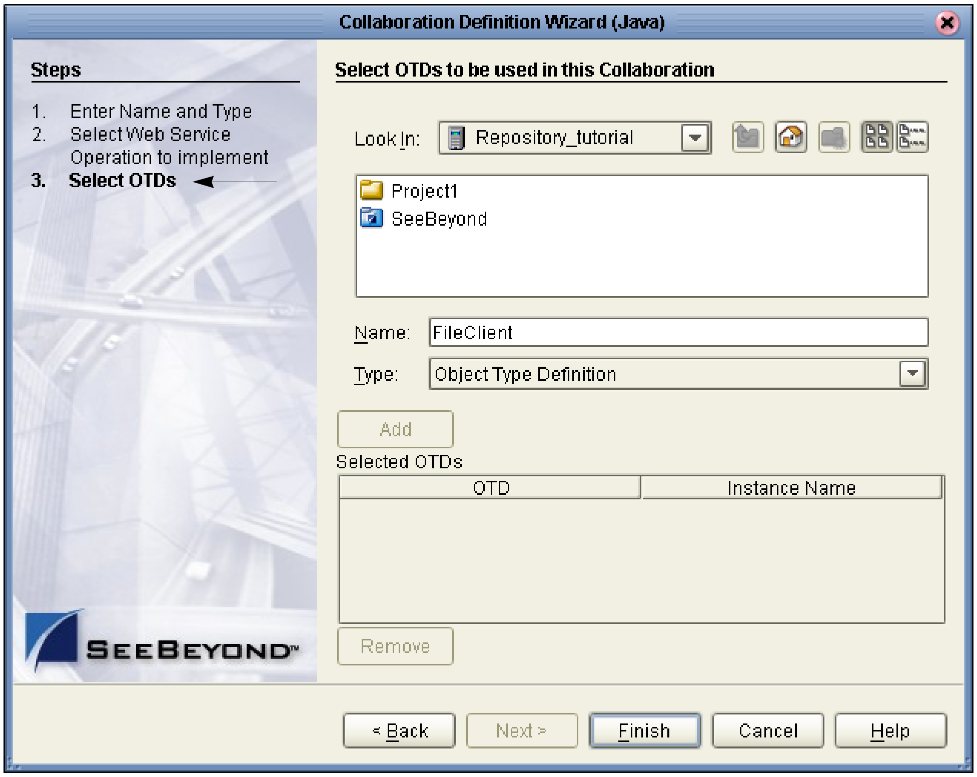
Type: Web Service Operation

6 Click **Next** when "receive" appears in the File Name field.

Select OTDs

In the following steps select the **FileClient.receive** OTD (its parameters are automatically defined). Also, select the user defined OTDs.

Figure 21 Select OTDs



Select OTDs to be used in this Collaboration

Look In: Repository_tutorial

Project1
SeeBeyond

Name: FileClient

Type: Object Type Definition

Selected OTDs

OTD	Instance Name

1 Double-click **SeeBeyond**.

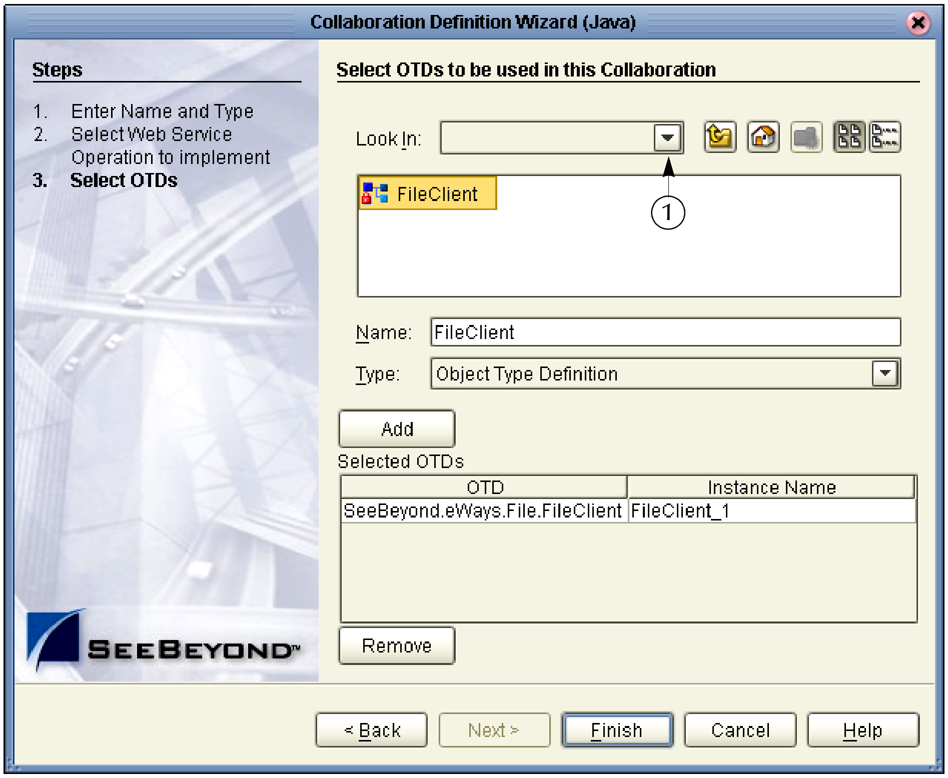
2 Double-click **eWays**.

3 Double-click **File**.

4 Double-click **FileClient**.

This adds the **FileClient_1** OTD to the list. See the following Figure 22.

Figure 22 Select FileClient OTD



Select OTDs to be used in this Collaboration

FileClient

Name: FileClient

Type: Object Type Definition

Selected OTDs

OTD	Instance Name
SeeBeyond.eWays.File.FileClient	FileClient_1

Note: *Before clicking Finish, install the user defined OTDs.*

5 Click the drop-down arrow (number 1 in Figure 22).

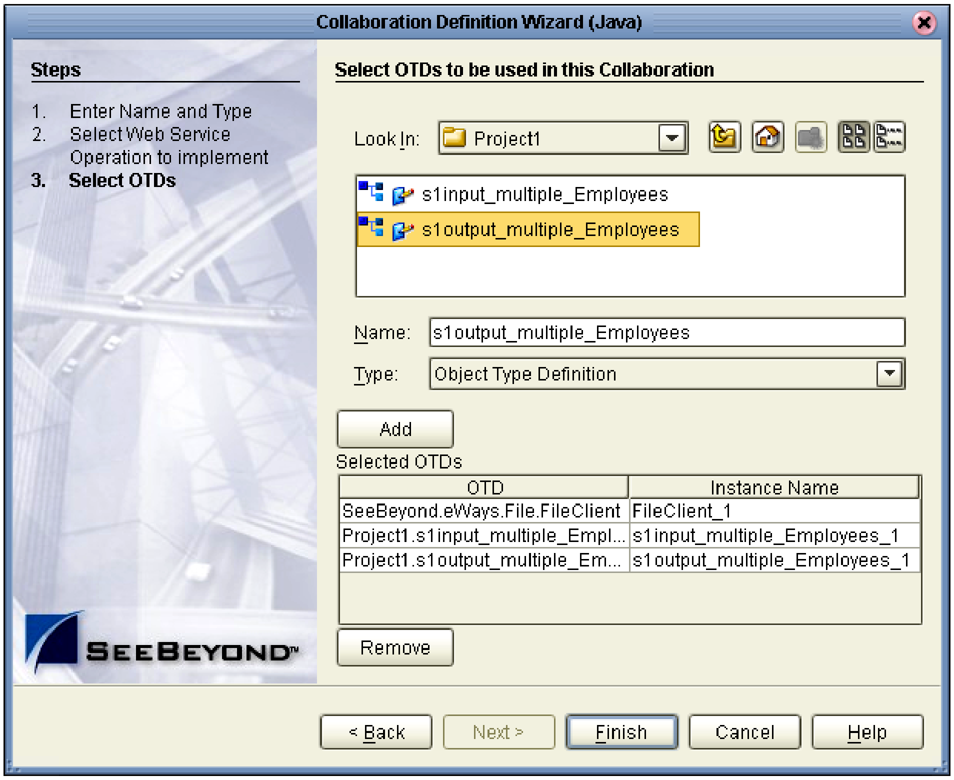
6 Click **Project1** to display the file OTDs.

7 Double-click **s1input_multiple_Employees**.

8 Double-click **s1output_multiple_Employees**.

See the following figure.

Figure 23 Selected OTDs



Steps

1. Enter Name and Type
2. Select Web Service Operation to implement
3. **Select OTDs**

Select OTDs to be used in this Collaboration

Look In: Project1

- s1input_multiple_Employees
- s1output_multiple_Employees

Name: s1output_multiple_Employees

Type: Object Type Definition

Selected OTDs

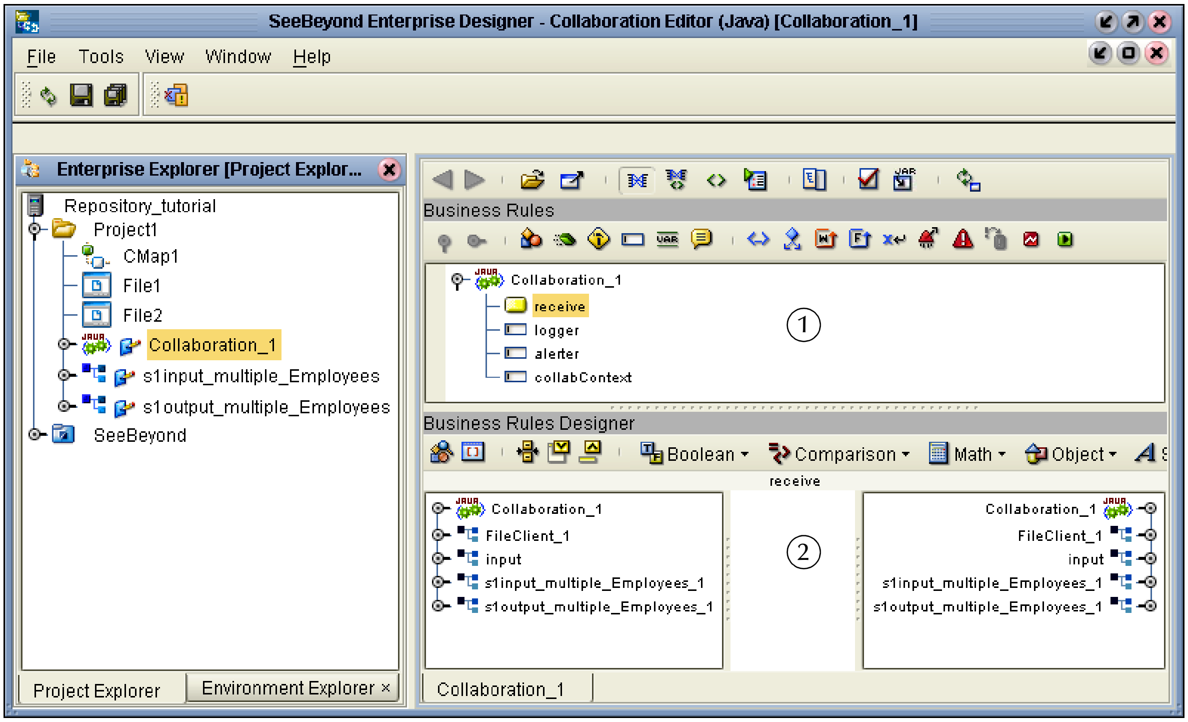
OTD	Instance Name
SeeBeyond.eWays.File.FileClient	FileClient_1
Project1.s1input_multiple_Empl...	s1input_multiple_Employees_1
Project1.s1output_multiple_Em...	s1output_multiple_Employees_1

All three OTDs appear in the list of OTDs.

9 Click **Finish**.

The Collaboration Editor (Java) appears. See the following figure.

Figure 24 Collaboration Editor (Java)



3.9 Apply the Business Rules

Business Rules display in the Business Rules pane (number 1 in Figure 24). Use the Business Rules Designer GUI (number 2 in Figure 24) to map your rules.

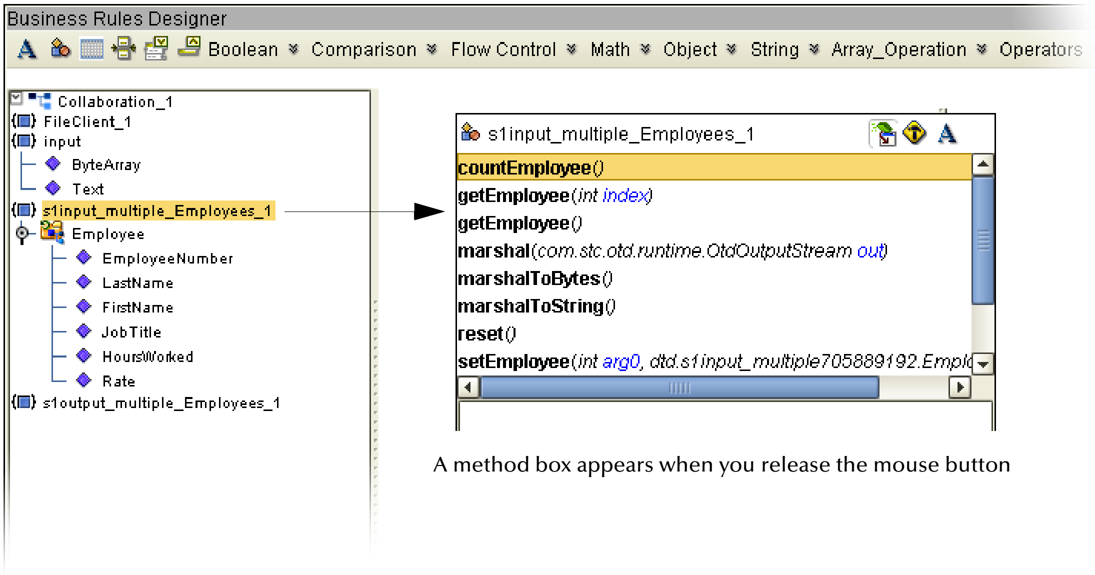
Unmarshal Text

First unmarshal the text input to prepare for a “read,” then concatenate the elements, as explained in the following steps.

1. Double-click **input** in the left pane of the Business Rules Designer to expand the view, (or click on the expansion node icon).
2. Double-click **s1input_multiple_Employees_1** in the left pane of the Business Rules Designer to expand the view (and double-click **Employee**).
3. Click and drag **s1input_multiple_Employees_1** from the left pane to the center pane of the Business Rules Designer (or right-click **s1input_multiple_Employees_1** and click **Select a Method to Call**).

   A method list appears.

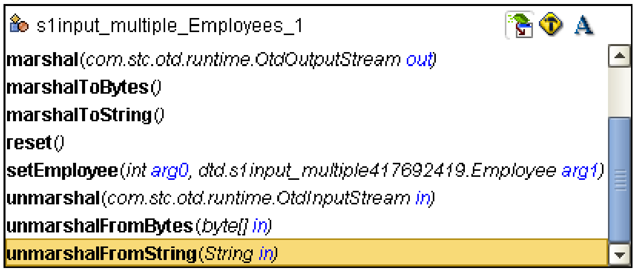
Figure 25 Unmarshal Text



countEmployee()
getEmployee(int index)
getEmployee()
marshal(com.stc.otd.runtime.OtdOutputStream out)
marshalToBytes()
marshalToString()
reset()
setEmployee(int arg0, dtd.s1input_multiple705889192.Emplo

A method box appears when you release the mouse button

Figure 26 Select Unmarshal



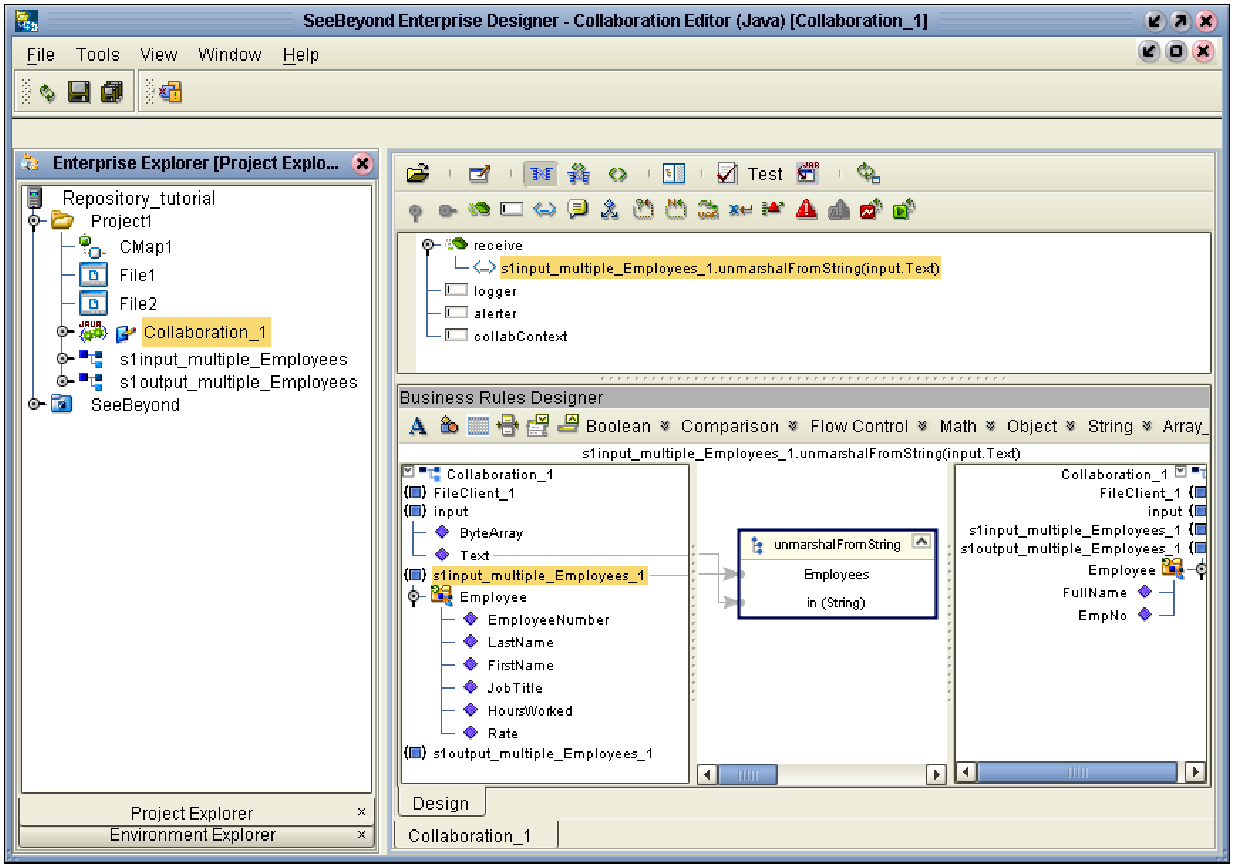
marshal(com.stc.otd.runtime.OtdOutputStream out)
marshalToBytes()
marshalToString()
reset()
setEmployee(int arg0, dtd.s1input_multiple417692419.Employee arg1)
unmarshal(com.stc.otd.runtime.OtdInputStream in)
unmarshalFromBytes(byte[] in)
unmarshalFromString(String in)

4 Scroll down then double-click the method **unmarshalFromString(String in)** in the list.

This action places a method box into the center pane of the Business Rules Designer canvas. The unmarshal method reads your employee input XML file.

5 Connect a node from **input**, **Text** to **in(String)**. See the following figure.

Figure 27 Business Designer - Unmarshal Text


SeeBeyond Enterprise Designer - Collaboration Editor (Java) [Collaboration_1]
File Tools View Window Help
Enterprise Explorer [Project Explo...
Repository_tutorial
Project1
CMap1
File1
File2
Collaboration_1
s1input_multiple_Employees
s1output_multiple_Employees
SeeBeyond
Project Explorer
Environment Explorer
Test
receive
s1input_multiple_Employees_1.unmarshalFromString(input.Text)
logger
alerter
collabContext
Business Rules Designer
Boolean Comparison Flow Control Math Object String Array
s1input_multiple_Employees_1.unmarshalFromString(input.Text)
Collaboration_1
FileClient_1
input
ByteArray
Text
s1input_multiple_Employees_1
Employee
EmployeeNumber
LastName
FirstName
JobTitle
HoursWorked
Rate
s1output_multiple_Employees_1
unmarshalFromString
Employees
in (String)
Collaboration_1
FileClient_1
input
s1input_multiple_Employees_1
s1output_multiple_Employees_1
Employee
FullName
EmpNo
Design
Collaboration_1

Connect Input and Output OTD Nodes

In the following steps connect **FirstName** and **LastName** in **s1input_multiple_Employees_1** in the left pane with **FullName** in **s1output_multiple_Employees_1** in the right pane (using **concat**).

1 Double-click to expand the view of **s1output_multiple_Employees_1** and **Employee** in the right pane.

Note: *Notice that the objects in both the left and right pane of the Business Rules Designer window have the same name. Keep in mind that the GUIs (**FileClient_1**, **s1input_multiple_Employee_1**, and **s1output_multiple_Employee_1** for example) refer to the same object whether they are in the left or right pane.*

2 Connect **EmployeeNumber**, from **s1input_multiple_Employees_1** in the left pane to **EmpNo** of **s1output_multiple_Employees_1** in the right pane.

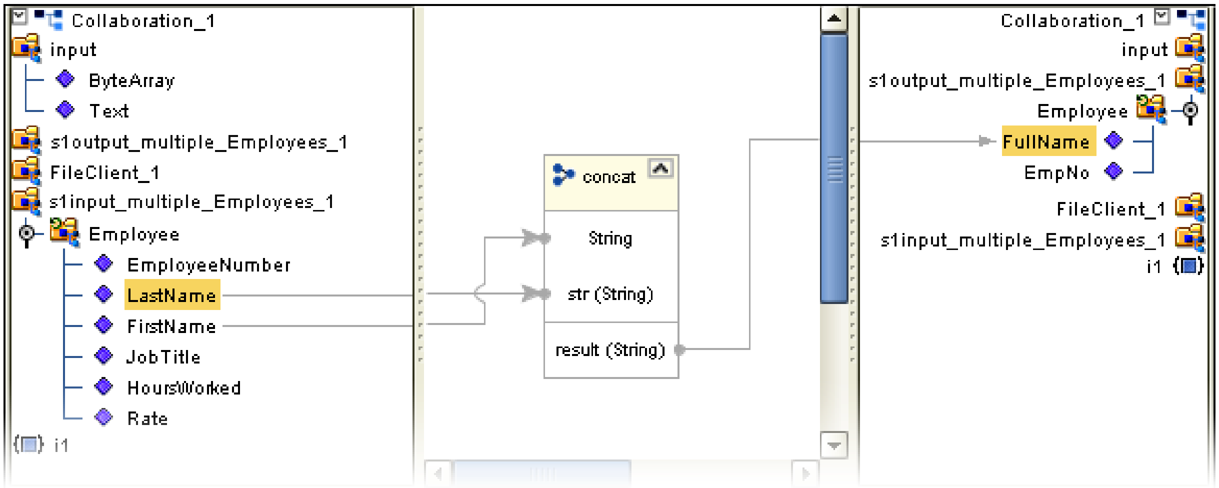
This action connects the nodes and also sets up the loop logic to process multiple records.

Note: *As you begin mapping, the graphic (GUI) disappears, but the Business Rules continue to appear in the top pane. The active Business Rule also appears at the top of the Business Rules Designer pane.*

3. Drag the **Concat** icon down to your designer workspace. (**Concat** is on the **String** drop-down menu.)
4. Connect **FirstName** to **String** in the Concat box. (Connect from **s1input_multiple_Employee_1** in the left pane.)
5. Connect **LastName** to **str (String)** in the Concat box. (Connect from **s1input_multiple_Employee_1** in the left pane.)
6. Connect **result (String)** to **FullName** in **Employee** of the root element **s1output_multiple_employees_1** in the right pane.

   See the following figure.

Figure 28 Concatenation Logic

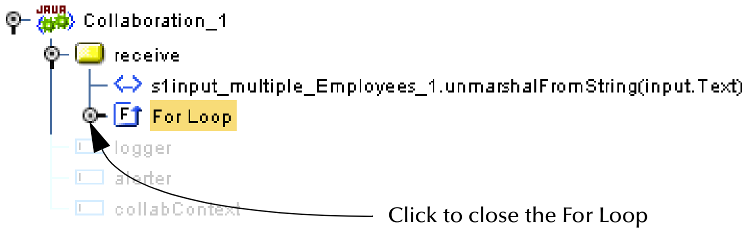


7. Click the “For Loop” node expansion control (in the top pane - Business Rules) to collapse the loop.

   The reason for this is so the **marshal** and **write** are only performed once after the file has been processed.

   See the following figure.

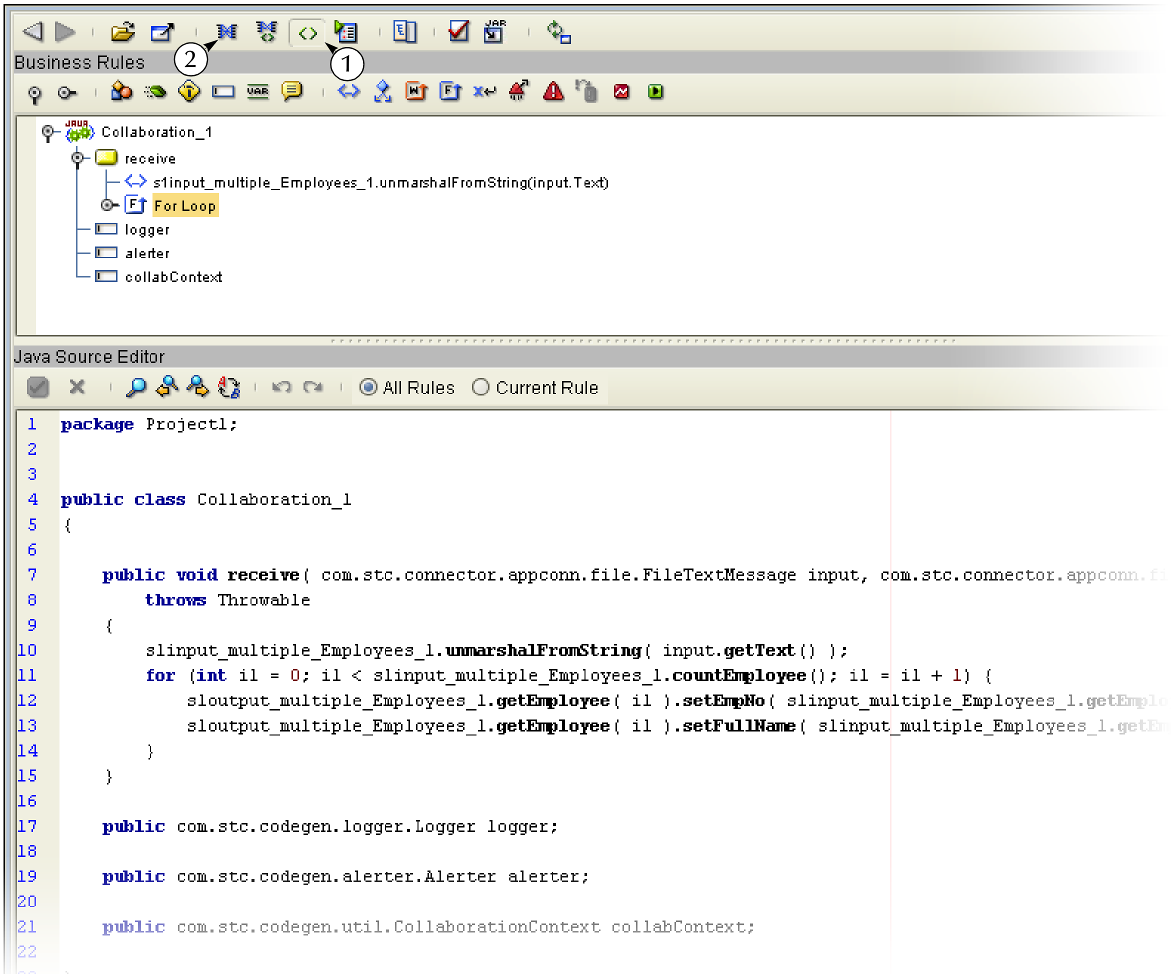
Figure 29 Collapse For Loop


Collaboration_1
receive
s1input_multiple_Employees_1.unmarshalFromString(input.Text)
For Loop
logger
alerter
collabContext
Click to close the For Loop

View Source Code

You may view or modify the source code by clicking the **Source code mode** icon (number 1 in Figure 30. You may toggle back to Standard mode by clicking the **Standard mode** icon (number 2 in Figure 30).

Figure 30 Source Code Mode



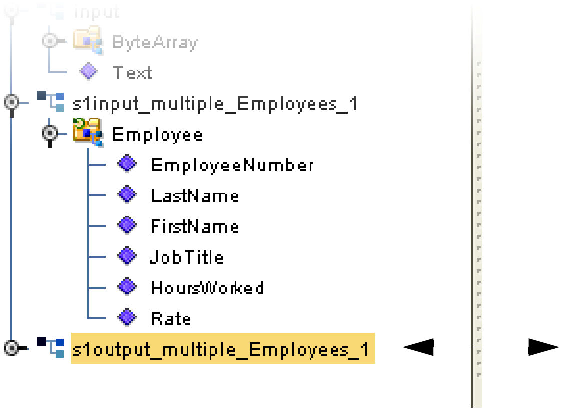
```
package Project1;


public class Collaboration_1
{

    public void receive( com.stc.connector.appconn.file.FileTextMessage input, com.stc.connector.appconn.fi
        throws Throwable
    {
        s1input_multiple_Employees_1.unmarshalFromString( input.getText() );
        for (int i1 = 0; i1 < s1input_multiple_Employees_1.countEmployee(); i1 = i1 + 1) {
            s1output_multiple_Employees_1.getEmployee( i1 ).setEmpNo( s1input_multiple_Employees_1.getEmplo
            s1output_multiple_Employees_1.getEmployee( i1 ).setFullName( s1input_multiple_Employees_1.getEm
        }
    }

    public com.stc.codegen.logger.Logger logger;

    public com.stc.codegen.alerter.Alerter alerter;

    public com.stc.codegen.util.CollaborationContext collabContext;
```

Marshal Text

You began by unmarshalling data from string to prepare to “read.” In the following steps marshal the data to string to prepare to “write.”

1. Double-click **FileClient_1** in the right pane to expand the view.
2. Click and drag **s1output_multiple_Employees_1** from the left to the center pane of the Business Rules Designer.

Figure 31 Click and Drag s1output



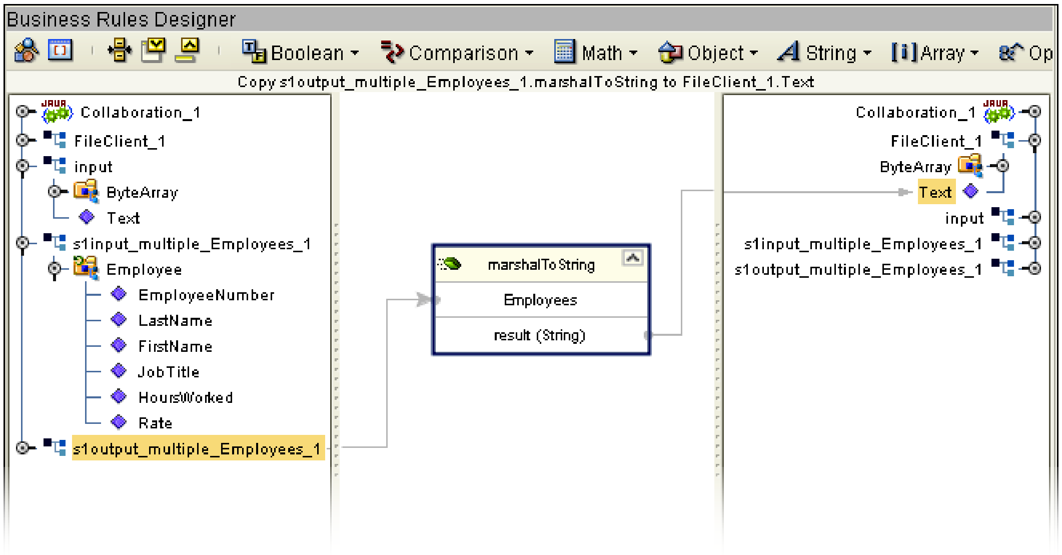
Note: *You may also right-click **s1output_multiple_Employees_1** and **Select a method to call**.)*

3 Scroll down then double-click the method **marshalToString()**.

The **marshalToString** method box appears.

4 Connect a node from **result (String)** to **Text** under **FileClient_1** in the right pane.

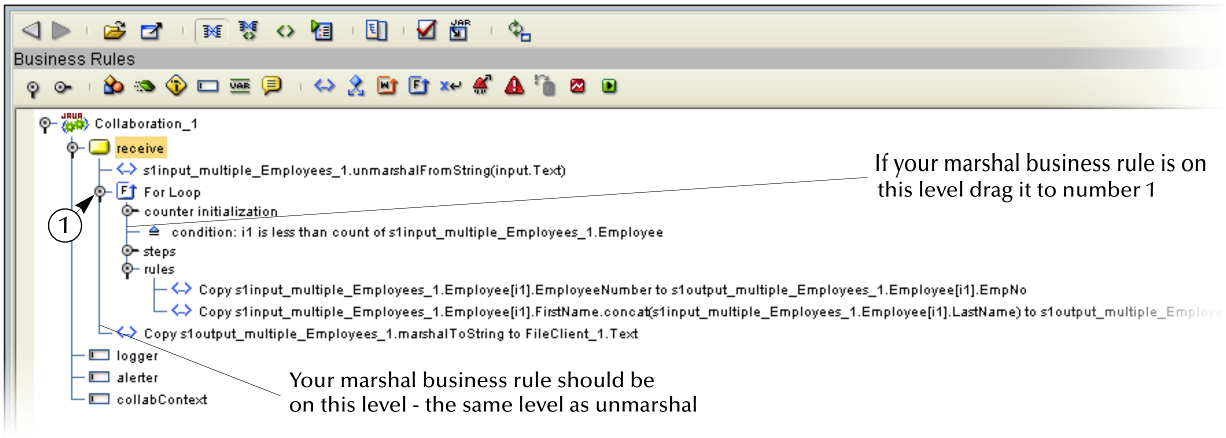
Figure 32 Marshal to String


Copy s1output_multiple_Employees_1.marshalToString to FileClient_1.Text

Your Business Rules (in the top pane) should look like Figure 33 (after you expand the nodes). The marshal and unmarshal actions should be on the same level. The "marshal" (and "write" which you will add next) should not be within the loop because that would cause a write after each individual employee record.

Note: *If your "marshal" rule is on the same level as the concat rule, click and drag your "marshal" outside the loop to number 1 in Figure 33.*

Figure 33 View Business Rules



```
Collaboration_1
  receive
    s1input_multiple_Employees_1.unmarshalFromString(input.Text)
    For Loop
      counter initialization
      condition: i1 is less than count of s1input_multiple_Employees_1.Employee
      steps
      rules
        Copy s1input_multiple_Employees_1.Employee[i1].EmployeeNumber to s1output_multiple_Employees_1.Employee[i1].EmpNo
        Copy s1input_multiple_Employees_1.Employee[i1].FirstName.concat(s1input_multiple_Employees_1.Employee[i1].LastName) to s1output_multiple_Employe
    Copy s1output_multiple_Employees_1.marshalToString to FileClient_1.Text
  logger
  alerter
  collabContext
```

If your marshal business rule is on this level drag it to number 1

Your marshal business rule should be on this level - the same level as unmarshal

Write FileClient_1

In these steps prepare to write to the output file.

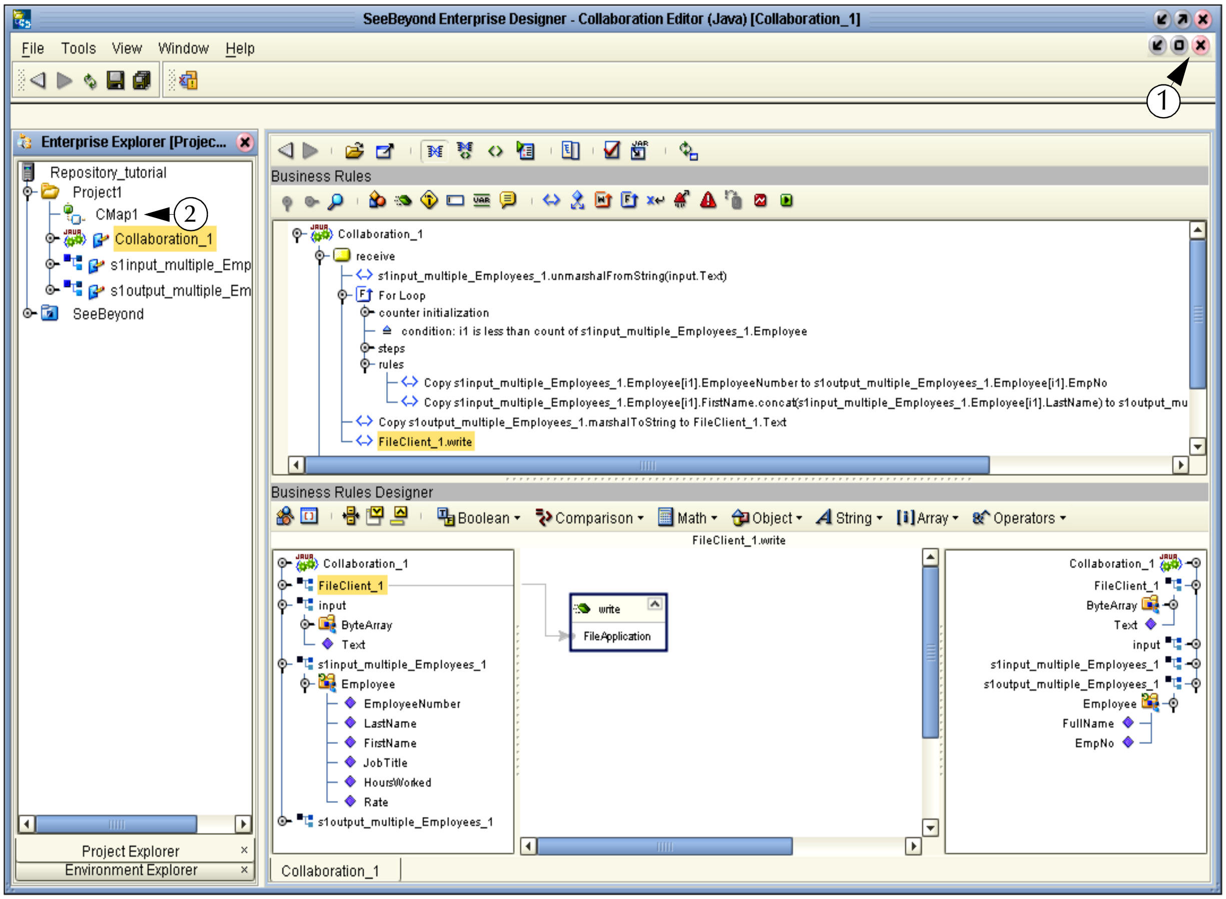
1 Click and drag **FileClient_1** from the left pane to the center pane of the Business Rules designer.

A list of methods appears.

2 Double-click **write()** from the pop-up dialog box.

See the following figure.

Figure 34 Write Output File



```
Collaboration_1
  receive
    s1input_multiple_Employees_1.unmarshalFromString(input.Text)
    For Loop
      counter initialization
        condition: i1 is less than count of s1input_multiple_Employees_1.Employee
      steps
      rules
        Copy s1input_multiple_Employees_1.Employee[i1].EmployeeNumber to s1output_multiple_Employees_1.Employee[i1].EmpNo
        Copy s1input_multiple_Employees_1.Employee[i1].FirstName.concat(s1input_multiple_Employees_1.Employee[i1].LastName) to s1output_mu
    Copy s1output_multiple_Employees_1.marshalToString to FileClient_1.Text
    FileClient_1.write
```

3 On the **File** menu, click **Save All**.

This completes the Collaboration. Use the drag-and-drop method to move the Collaboration into your Service.

4 Navigate back to the Connectivity Map by closing the current window (number 1 in Figure 34 above) or by double-clicking the Connectivity Map (number 2 in Figure 34 above) name, **CMap1**.

3.10 Apply the Collaboration

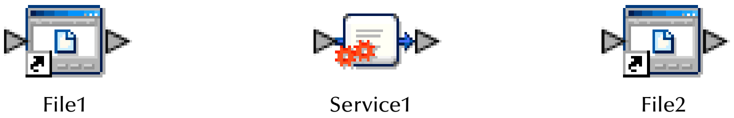
You have configured your Collaboration Definition (business rules) in the previous section. You are now ready to bind your Collaboration Definition with the Service using the drag-and-drop method.

3.10.1 Link Components in the Connectivity Map

Linking the components creates the logical flow of data throughout the Project. Additionally, creating a link between the files and the Service adds eWays to the Connectivity Map.

At this point, the Connectivity Map contains three unlinked objects.

Figure 35 Objects in Connectivity Map



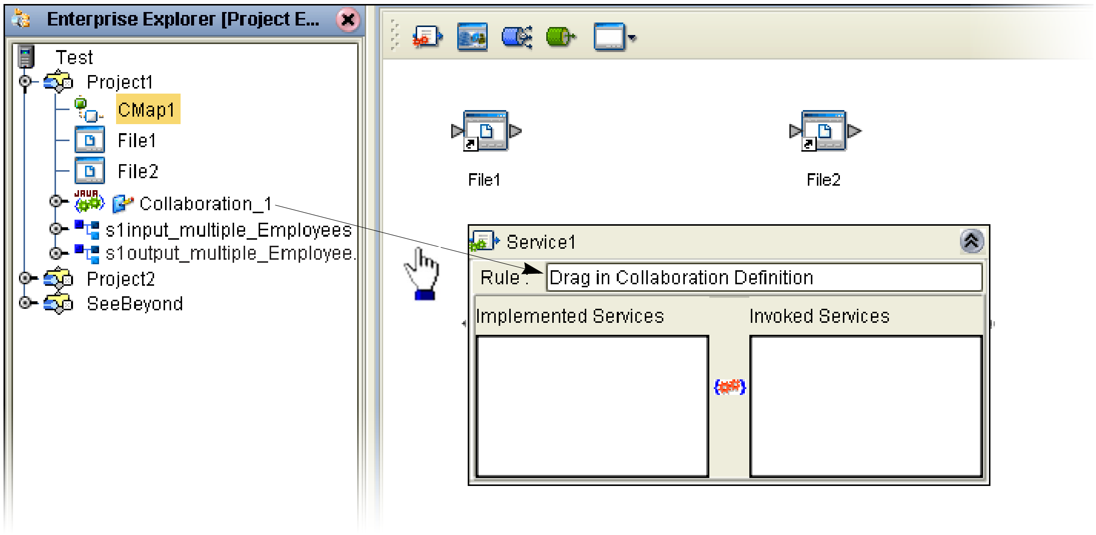
Connect Nodes

1 Double-click **Service1** in the Connectivity Map to open the Service window. (You can also reposition the window after it is open.)

2 Drag the **Collaboration_1** definition from the Project Explorer pane to the **Rule** field in the open window as shown in the following Figure 36.

Note: *Click once on* ***Collaboration_1*** *to activate Project Explorer in the left pane. Then click* ***Collaboration_1*** *once again to select and drag.You may also drag* ***Collaboration_1*** *into the icon without first opening the Collaboration window.*

Figure 36 Using the Drag and Drop method

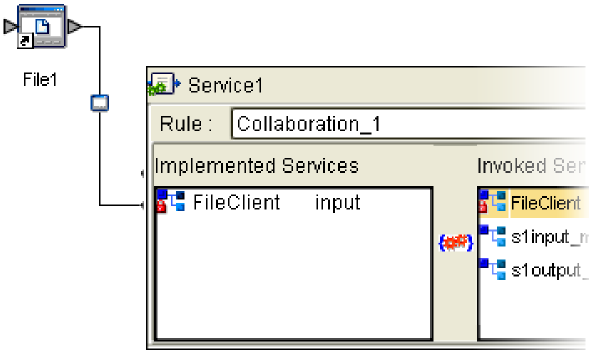


Notice that the gear icons change from red to green to signify that the Collaboration is bound to the Service.

Note: *You may connect nodes forward or backward. However, it may be easier to connect from the Service dialog box backward to the module icon, especially when there are multiple Services in more complex projects.*

3 Connect **FileClient** input to **File1** as shown in the following Figure 37. Since there is only one option the application automatically makes the connection.

Figure 37 Connect FileClient to File1



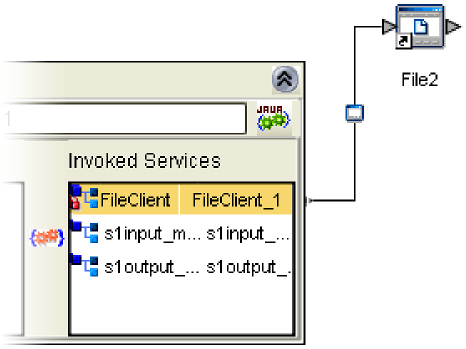
Rule : Collaboration_1

FileClient input

FileClient

4 Connect **FileClient** to **File2** as shown in the following Figure 38.

Figure 38 Connect FileClient to File2



FileClient FileClient_1

s1input_m... s1input_...

s1output_... s1output_

5 Click the **Close** button to close the binding box.

6 Click **Save All**.

3.11 Configure the eWays

Your Connectivity Map contains two eWays. These must be configured for deployment.

- The input eWay polls a directory on the local file system. For example: **c:\eGateData\Project1**, for any XML file (***.xml**).

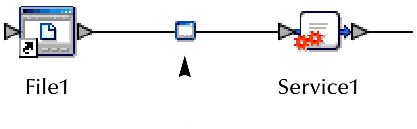
- The Output eWay writes the results to an output file (**.dat**) on the local file system. For example: **c:\eGateData\Project1**.

Configure Inbound eWay

Identify the file eWay shown in Figure 39 as an inbound file and then set the properties to identify the file and its location.

1 Double-click the first eWay —□— .

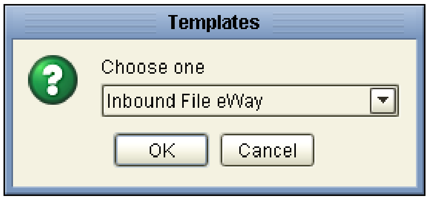
Figure 39 Inbound eWay



2 Select **Inbound File eWay** from the drop-down list.

3 Click **OK**.

Figure 40 External Type Dialog Box



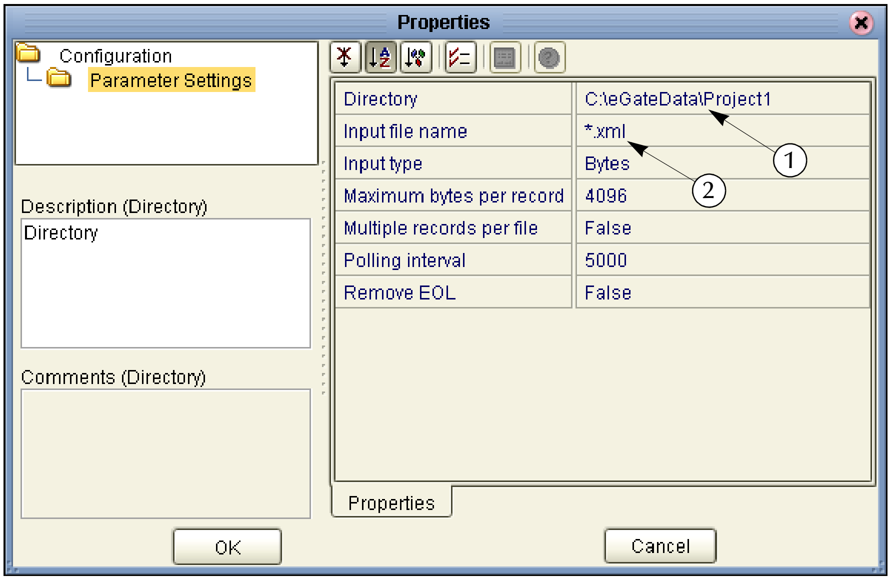
Templates

Choose one

Inbound File eWay

This creates a new inbound eWay and the Properties Dialog Box appears.

Figure 41 Properties Configuration (Inbound eWay)



Properties

Configuration
Parameter Settings

Description (Directory)

Directory

Comments (Directory)

Directory	C:\eGateData\Project1
Input file name	*.xml
Input type	Bytes
Maximum bytes per record	4096
Multiple records per file	False
Polling interval	5000
Remove EOL	False

Refer to the numbered arrows in the previous figure. These are the editable fields you configure.

4 Click once on the current path to select and type a correct directory path. In this Tutorial the path is **c:\eGateData\Project1** (number 1 in Figure 41 above).

5 Click once on the current **Input File Name** field and enter ***.xml**. The eWay reads any files with the .xml extension (number 2 in Figure 41 above).

6 Click **OK** to close the Properties Dialog Box for the inbound eWay.

Note: *If the **Polling interval** value is left unchanged, the eWay polls the directory location every five seconds (5000 milliseconds).*

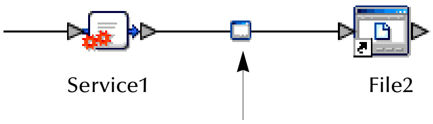
Configure Outbound eWay

Identify the file eWay shown in Figure 42 as an outbound file and then set the properties to identify the file and its location.

1 Double-click the second eWay.

2 Select **Outbound File eWay** from the drop down list.

Figure 42 Outbound eWay



3 Click **OK**.

The Properties dialog box appears.

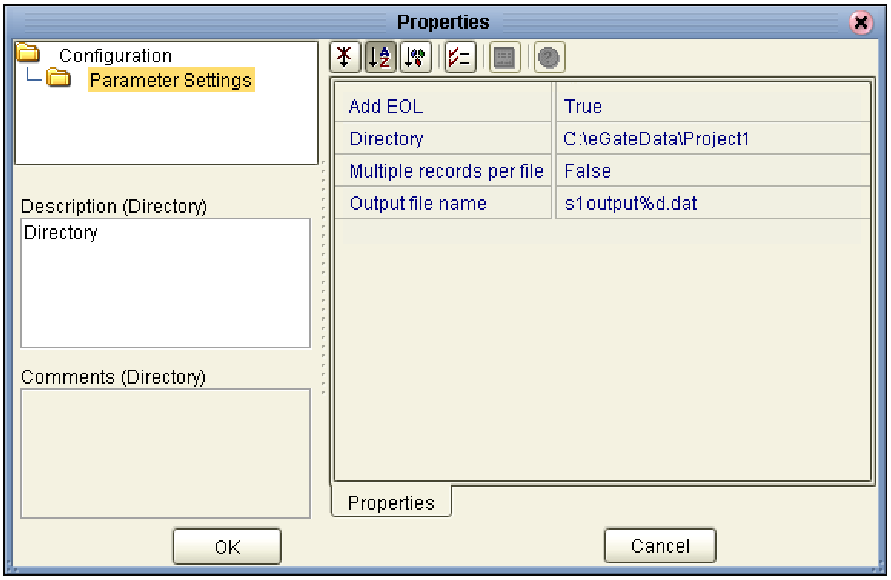
Use the same path you used for the inbound eWay (**c:\eGateData\Project1**).

4 Name the output file: **s1output%d.dat** (Scenario1 output).

5 Click **OK** to close the Properties Dialog Box for the outbound eWay.

Note: *The output is a text file and can have any name, and the extension can also be .txt. The %d makes the file a "generation" file and the number will increment each time the file is written.*

Figure 43 Properties Configuration (Outbound eWay)



Add EOL	True
Directory	C:\eGateData\Project1
Multiple records per file	False
Output file name	s1output%d.dat

Note: *The **Multiple records per file** setting can be confusing. You may use either True or False in the tutorials. Refer to* **"Multiple records per file (True or False)" on page 51**.

6 On the **File** menu, click **Save All**.

You have now configured both the input and output eWays and are ready to deploy your Project. You may skip to **"Create Environment" on page 52**.

Multiple records per file (True or False)

This setting (refer to Figure 41 and Figure 43) specifies if multiple records are to be obtained or sent per file. Multiple records (messages) are generated per line up to the maximum bytes per record.

Inbound eWay

A **True** setting means that multiple messages are generated per line up to the number specified in the Maximum bytes per record property.

A **False** setting, as in the Tutorial, means that the XML file is read as a block and represents a single message. This message can be treated as one "record" because our business rule contains the logic to loop through each record in the XML message.

Outbound eWay

For the output eWay, this setting determines whether records are concatenated in a file.

A **True** setting means that multiple messages are written to the same file. New messages are appended (concatenated) to the output file with each **File.send()**.

A **False** setting means that only one message is written in one file. New messages are written to a different file (with the file number incremented if you use %d) with each **File.send()**.

You may want to experiment with this setting and observe the different results.

3.12 Create Environment

Create and activate the Deployment Profile. Before you can create the Deployment Profile, you must create an Environment.

3.12.1 Create Environment and Logical Host

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

1. Click **View** from the Menu bar. Click **Environment Explorer** (or click the **Environment Explorer** tab).
2. Right-click the Repository name ( ). Click **New Environment**.
3. Right-click **Environment1** and rename **Tutorial**. Press **Enter**.
4. Right-click **Tutorial**. Click **New Logical Host**.

   This creates a **LogicalHost1** window in the right pane.

Create Integration Server

An integration server is a J2EE software platform that houses the business logic container used to run Collaborations and JCA connectors (eWays).

1. Right-click **LogicalHost1** in the **Enterprise Explorer** window. Click **New SeeBeyond Integration Server**.

   IntegrationSvr1 appears in the **LogicalHost1** window. Your Collaboration is bound to this server.

Create JMS IQ Manager

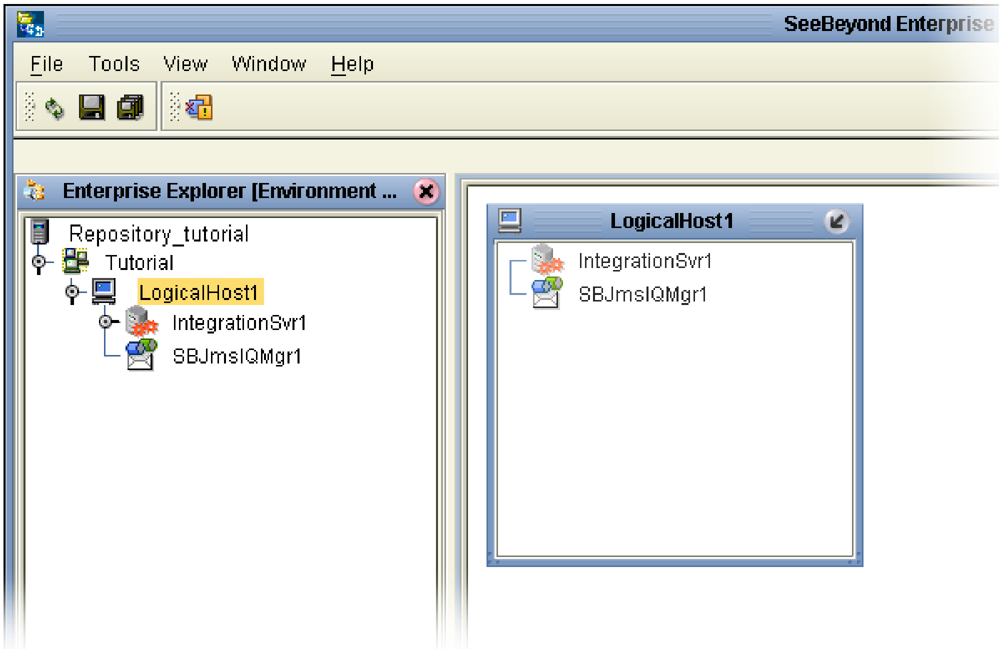
The JMS IQ Manager is a JMS-compliant, guaranteed delivery store, forwarding, and queueing service.

Note: *You would bind your Topics and Queues to this server if you had them. In Scenario1 there are no Topics or Queues; in Scenario2 you will use a Topic.*

1. Right-click **LogicalHost1** in the **Enterprise Explorer** window. Click **New SeeBeyond JMS IQ Manager**.

   SBJmsIQMgr1 appears in the **LogicalHost1** window.

Figure 44 Environment with Logicalhost

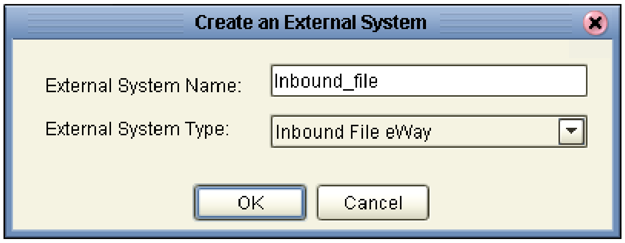


These servers automatically appear in the **LogicalHost1** dialog box.

Create External File System for Inbound eWay

1. Right-click **Tutorial** Environment).
2. Click **New File External System**.

Figure 45 External File Inbound



Create an External System

External System Name: Inbound_file

External System Type: Inbound File eWay

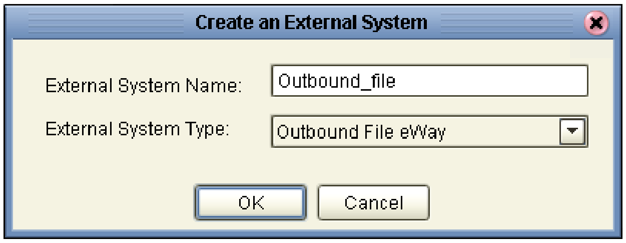
3. Select **Inbound File eWay**.
4. Type **Inbound_file** in the name field.
5. Click **OK**.

This places the inbound external system, **Inbound_file**, in the Environment Editor pane.

Create External File System for Outbound eWay

1. Right-click **Tutorial**.
2. Click **New File External System**.

Figure 46 External File Outbound



Create an External System

External System Name: Outbound_file

External System Type: Outbound File eWay

3 Select **Outbound File eWay**.

4 Type **Outbound_file** in the name field.

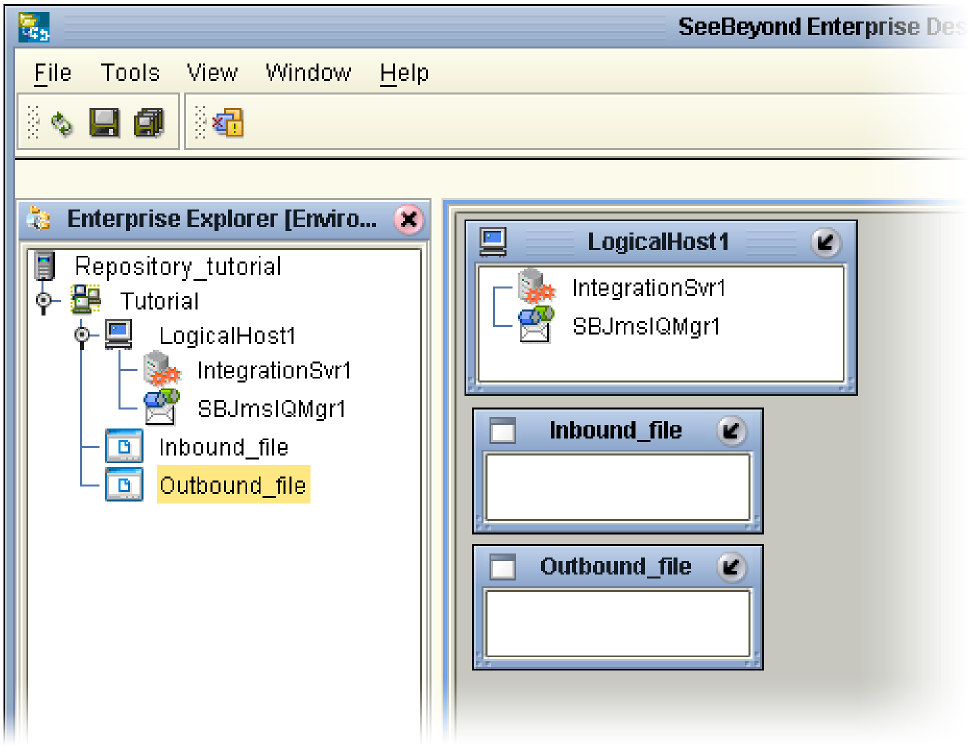
5 Click **OK**.

This places the outbound external system, **Outbound_file**, in the Environment Editor pane.

Note: *If the "Create an External System" (or any Java pop-up) window doesn't appear, it may be hidden behind your Enterprise Explorer window. Press the ALT+TAB keys to see if you have any hidden pop-ups. Use this keystroke combination if your Enterprise Explorer GUI appears to be "frozen."*

Your Environment Editor pane should look similar the following figure.

Figure 47 Environment Editor



3.13 Create and Activate Deployment Profile

First create a Deployment Profile, and then deploy (map) the components.

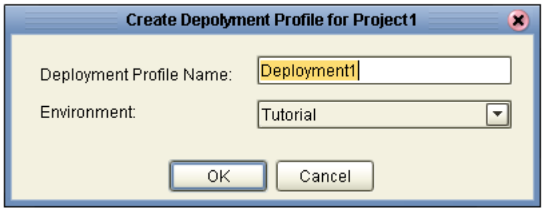
Create Deployment Profile

A Deployment Profile contains information about how Project components are deployed in an Environment, and it also maps components to the Environment.

1 Click the **Project Explorer** tab to return to the Project Explorer pane of your Project.

2 Right-click **Project1**, and then click **New** and **Deployment Profile**.

Figure 48 Deployment Profile



Create Depolyment Profile for Project1

Deployment Profile Name: Deployment1

Environment: Tutorial

3 Accept the default name **Deployment1** (and **Tutorial**). Click **OK**.

Your files and Service1 appear in the center pane of the Environment editor. Deploy these components using the drag and drop method.

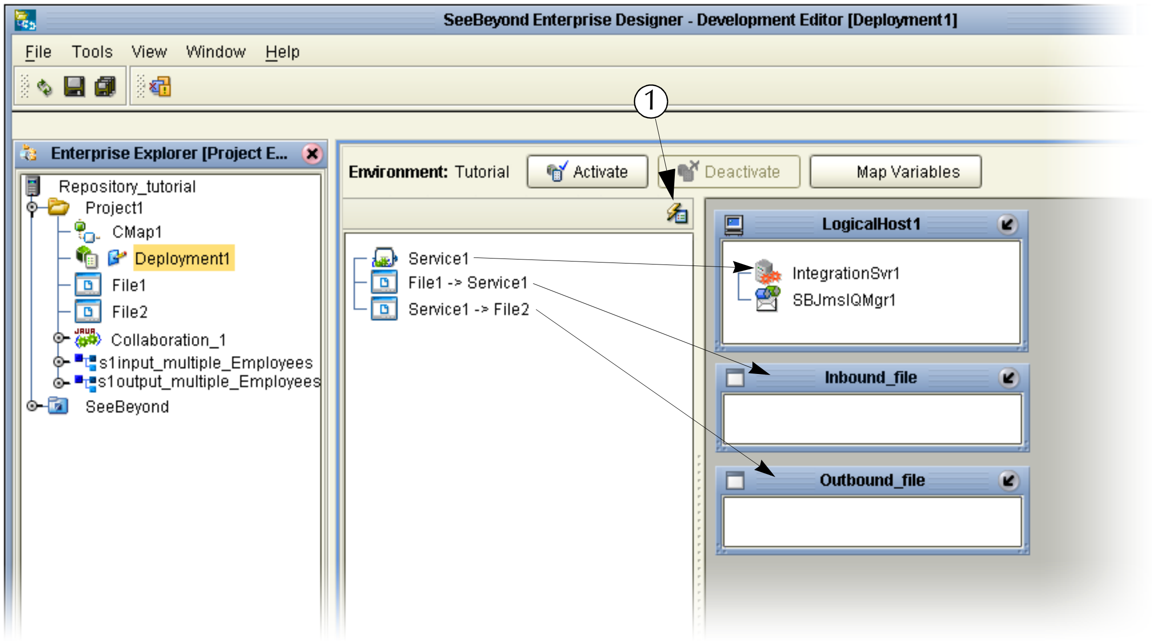
4 Double-click **Deployment1** in the Project Explorer to open the Environment editor (if it is not already open).

5 Drag-and-drop your Service, **Service1,** into your Integration Server, **IntegrationSvr1**. (See the previous Figure 49.)

6 Drag-and-drop your external files into your External File Servers. (File1 is your inbound file and File2 is your outbound file.)

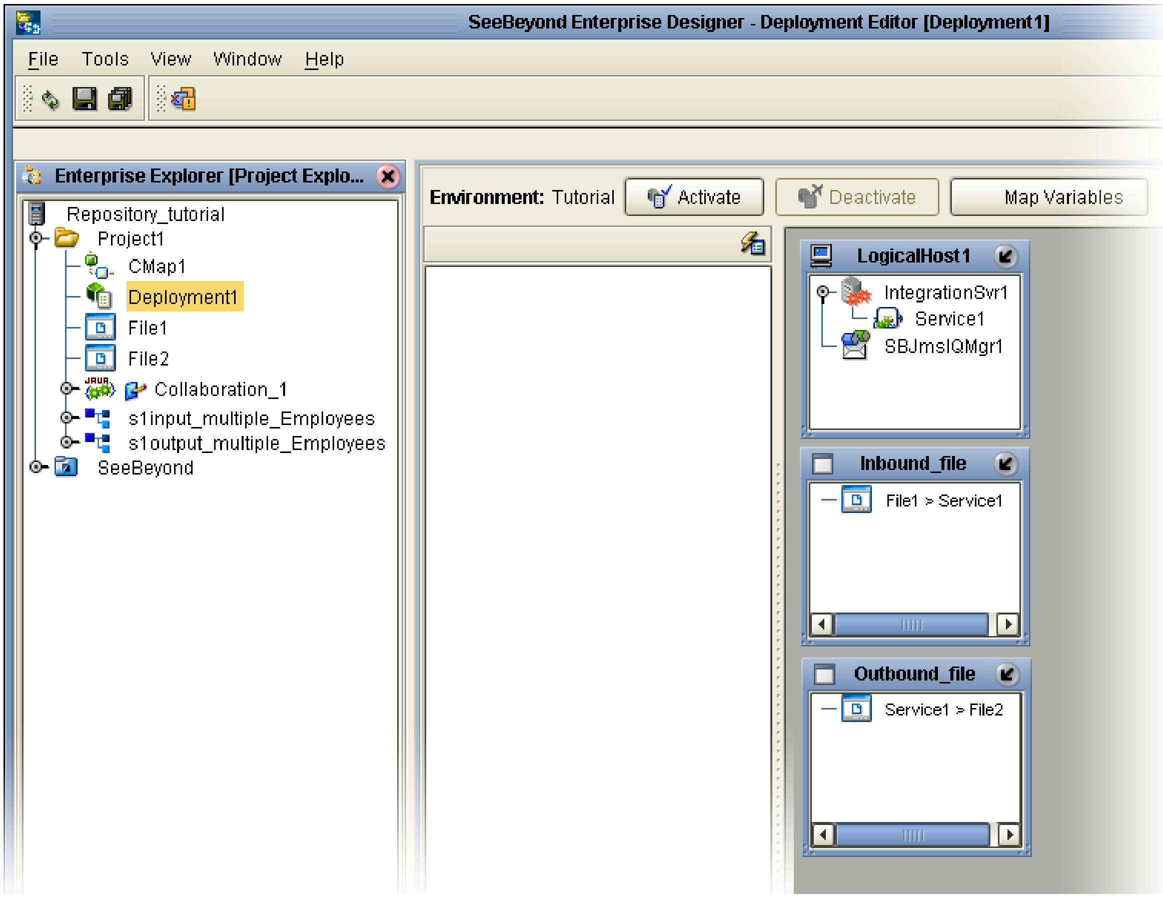
See the following figure.

Figure 49 Environment, Drag and Drop Components



Note: *Number 1 in the figure above is the Auto Map. You can click on this icon to map components to the LogicalHost1. In this Tutorial, we will manually map the components.*

Figure 50 Populated Environment



After populating your Environment with all the available components, the center pane of the Environment editor is empty. All your components reside in the servers, as shown in the previous Figure 50.

7 Save your Project.

3.14 Run the Bootstrap and Management Agent

The Bootstrap command starts the Logical Host and begins the process of polling your input data. The bootstrap process is run from a command prompt or by executing **bootstrap.bat**. Bootstrap picks up the deployment profile the first time it runs; after that reactivate and check the box in Figure 53 to apply the most recent changes to the logical host(s).

Logical Host Overview

At runtime, the Logical Host bootstrap script starts the bootstrap Java program that downloads the Management Agent, the Message Server, and the Integration Server from the Repository. The Management Agent is then started, which in turn starts the Message Server(s) and Integration Server(s).

On subsequent bootstraps of the Logical Host, the configuration will not automatically update and you can continue running the previous version of the deployed components. You can force a refresh of the Logical Host configuration by using the **-f** option.

Enter the command: **bootstrap -f**

You can also refresh the Logical Host using the option on the Activate dialog box. See Figure 53.

3.15 Run the Bootstrap

Note: *The Bootstrap command is case sensitive on Windows platforms.*

Two ways to run the bootstrap are explained below:

3.15.1 Edit Properties and run bootstrap.bat (method 1)

You can edit the configuration file: ***Logical-host.properties***. Ensure that the Logical Host is not running. The required fields with sample input are explained below:

1. Navigate to the following file (depending your installation's setup): Ican50>logicalhost>bootstrap>config>**logical_host.properties**.

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

2. Edit: **repository.url=http://localhost:12000/Repository_tutorial**

   localhost:12000 is the default port number. Repository_tutorial is the Repository name in this example.

3. Edit: **repository.username=Administrator**

   Enter the Administrator name.

4. Enter the repository password: **repository.password=XXX**

   repository.password: (USER CONFIGURABLE). Plain text form of password used for connecting to the Repository. Any value provided here are cleared out by the system and written in encrypted form to the **repository.password.encrypted** field.

5. Edit: **logical.host.environment.name=Tutorial**

   Enter an environment name that is used when you deploy your project.

6. Edit: **logical.host.name=LogicalHost1**

   logical.host.environment.name: (USER CONFIGURABLE). Specifies the name of the environment containing the current logical host.

   After you have configured this file you can execute **bootstrap.bat**. You can double-click on **Bootstrap.bat** which may be in the following folder path:

logicalhost>bootstrap>bin. You could also enter the following command from a command prompt in a DOS shell:

Example: C:\*Ican50**logicalhost**bootstrap**bin****bootstrap.bat***

Note: *Do not enter spaces before or after the equal sign (=) and the property values. Spaces are allowed only in the value itself. Also, port 12000 is used in the example; your port number could be different.*

3.15.2 Run Bootstrap.bat from a DOS command (method 2)

You can also enter the bootstrap commands in a **cmd** shell as explained in the following steps.

1. Open a Windows command prompt as shown in the following Figure 51. (Click **Start**; click **Run;** type **cmd**.)
2. Navigate to where you installed the logicalhost. For example, ***C:\ican50\logicalhost\bootstrap\bin***
3. To start the bootstrap process, type the following command:

```
bootstrap -e environment_name -l logicalhost_name
-r repository_URL -i username -p password
```

Where ***environment_name*** is the name of your environment (for example, *Tutorial*),

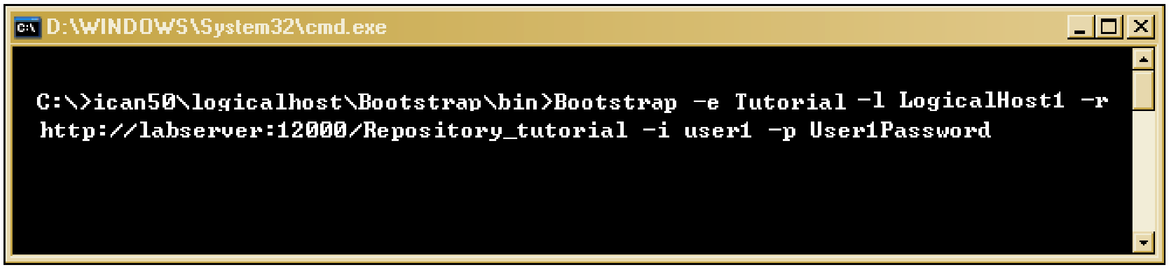
logicalhost_name is the name of your Logical Host (for example, *LogicalHost1*),

repository_URL is the full URL of your Repository including the Repository name (for example, *http://labserver:12000/Repository_tutorial*),

username is your user name, and

password is your password.

Figure 51 Bootstrap Command Example



```
C:\>ican50\logicalhost\Bootstrap\bin>Bootstrap -e Tutorial -l LogicalHost1 -r
http://labserver:12000/Repository_tutorial -i user1 -p User1Password
```

4. Press **Enter**.

The bootstrap process takes a few minutes to execute. The Management Agent starts the components in the Project; the Input eWay processes the input XML message.

Note: *To force a refresh of the Logical Host see* **"Logical Host Overview" on page 57**.

After you have successfully started the Bootstrap, you are ready do deploy and run your project.

3.16 Run Project1

When you create and deploy **Project1**, a deployment file is downloaded by the Logical Host. The Project is then packaged into an .ear file that is stored in the Repository. The Logical Host downloads this package from the Repository.

If you want to do a "hot deploy," which means that the Logical Host picks up the current package in real time after the bootstrap has been started, you could click **YES** when prompted. See Figure 53.

Activate Environment

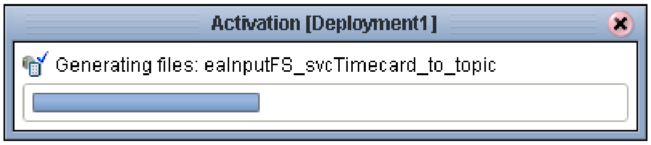
1 On the **File** menu, click **Save All**.

2 Click the **Activate** button.

When you click the Activate button, the project is packaged into an .ear file that is then copied to and stored in the Repository.

The Activation in Progress message appears.

Figure 52 Activation in Progress Message



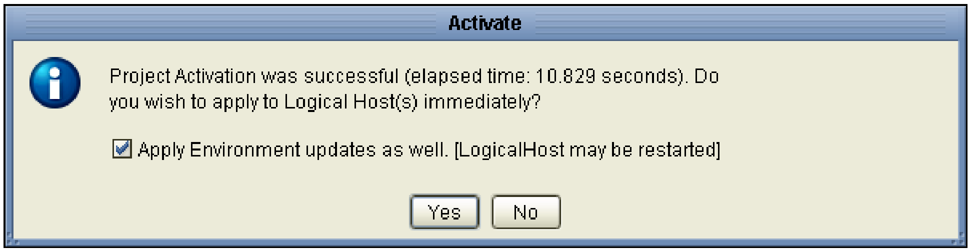
Activation [Deployment1]

Generating files: eaInputFS_svcTimecard_to_topic

3 Check the box, Apply Environment Updates, the first time you execute the bootstrap, then click **Yes**. See the following figure.

Note: *If you do not check the box, the Management Agent will not pick up the current Environment changes but will only pick up the runable project file.*

Figure 53 Activation Options



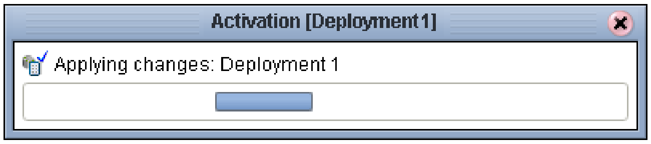
Activate

Project Activation was successful (elapsed time: 10.829 seconds). Do you wish to apply to Logical Host(s) immediately?

☑ Apply Environment updates as well. [LogicalHost may be restarted]

Assuming your project is already running, your project deploys when you click **Yes**. This means that the processes of the Logical Host are started up (Integration Server, Management Agent) and the .ear file is copied into the Logical Host.

Figure 54 Apply Changes in Progress

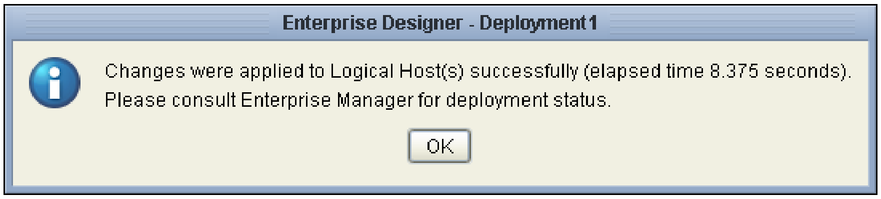


Activation [Deployment1]

Applying changes: Deployment 1

Applying changes to the Deployment Profile may take a few minutes. If your changes are successful the following message appears.

Figure 55 Changes Applied to Logical Host



Enterprise Designer - Deployment1

Changes were applied to Logical Host(s) successfully (elapsed time 8.375 seconds).
Please consult Enterprise Manager for deployment status.

4 Click **OK**.

Note: *You can make changes while the Project is running and then Reactivate to get the latest changes.*

Note: *When you Deactivate a Project, the .ear file that was copied to the Logical Host during deployment is removed. In this case, the Integration Server and Management Agent are still running (the Logical Host is still running) but there is no project for the Logical Host to run. The .ear file in the Repository, created at Activation time, remains untouched, so it can be available for another Logical Host to deploy.*

3.17 Verify Output Data

The Management Agent processes your input file, **TimeCard_Multiple.xml**, and writes the results to **s1output1.dat**.

View Output

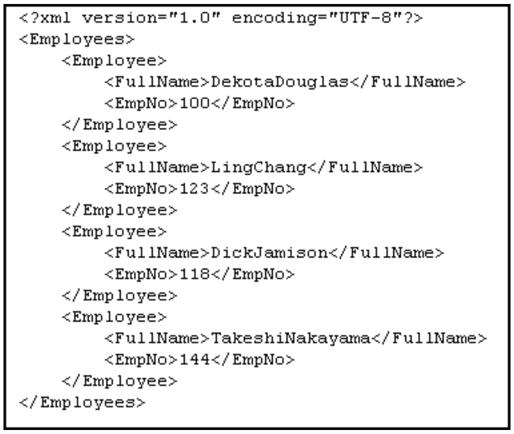
1 Use Windows Explorer to navigate to **c:\eGateData\Project1**.

2 See that the Input eWay, after processing the input file, renamed the input file to **TimeCard_Multiple.~in**. You can make changes in this file using a text editor and then rename the file back to **TimeCard_Multiple.xml** and immediately see the results of your changes after the file is processed again.

3 See that the Output eWay generated the output file, **s1output1.dat**.

4 Use Windows Wordpad or a similar text edit application to view the contents of the output file.

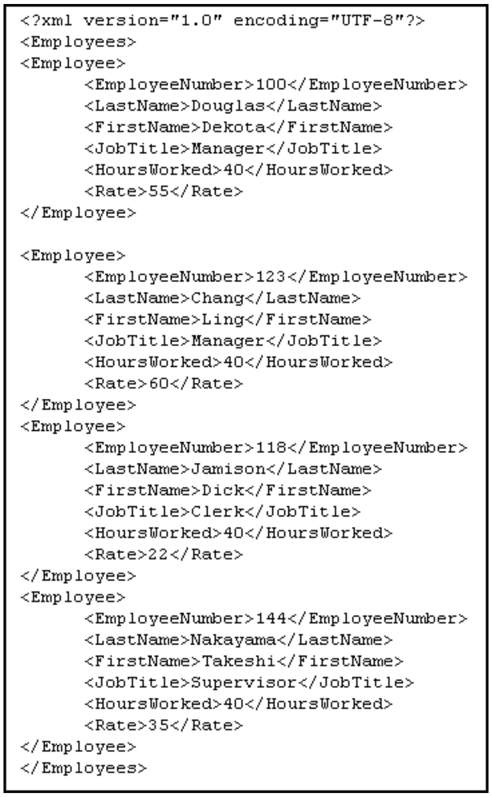
Figure 56 Output File



```
<?xml version="1.0" encoding="UTF-8"?>
<Employees>
    <Employee>
        <FullName>DekotaDouglas</FullName>
        <EmpNo>100</EmpNo>
    </Employee>
    <Employee>
        <FullName>LingChang</FullName>
        <EmpNo>123</EmpNo>
    </Employee>
    <Employee>
        <FullName>DickJamison</FullName>
        <EmpNo>118</EmpNo>
    </Employee>
    <Employee>
        <FullName>TakeshiNakayama</FullName>
        <EmpNo>144</EmpNo>
    </Employee>
</Employees>
```

In this sample, the first and last names are concatenated in the field, **FullName** as shown in Figure 56. (The test files are editable and you can experiment with the input. Your data may not match the samples shown in the Tutorial.)

Figure 57 Input File



```
<?xml version="1.0" encoding="UTF-8"?>
<Employees>
<Employee>
      <EmployeeNumber>100</EmployeeNumber>
      <LastName>Douglas</LastName>
      <FirstName>Dekota</FirstName>
      <JobTitle>Manager</JobTitle>
      <HoursWorked>40</HoursWorked>
      <Rate>55</Rate>
</Employee>

<Employee>
      <EmployeeNumber>123</EmployeeNumber>
      <LastName>Chang</LastName>
      <FirstName>Ling</FirstName>
      <JobTitle>Manager</JobTitle>
      <HoursWorked>40</HoursWorked>
      <Rate>60</Rate>
</Employee>
<Employee>
      <EmployeeNumber>118</EmployeeNumber>
      <LastName>Jamison</LastName>
      <FirstName>Dick</FirstName>
      <JobTitle>Clerk</JobTitle>
      <HoursWorked>40</HoursWorked>
      <Rate>22</Rate>
</Employee>
<Employee>
      <EmployeeNumber>144</EmployeeNumber>
      <LastName>Nakayama</LastName>
      <FirstName>Takeshi</FirstName>
      <JobTitle>Supervisor</JobTitle>
      <HoursWorked>40</HoursWorked>
      <Rate>35</Rate>
</Employee>
</Employees>
```

3.17.1 Edit Input File in Real Time

With your Repository server running, you may edit the input, Time Card, file in real time and change the data names or values in that file.

1 Edit the input file using a text edit application such as WordPad. As a test you can change one or two names in the file.

2 Rename the input file whose name has changed to **TimeCard_Multiple.~in**. Change the file extension name **.~in** back to **.xml**.

Within a few seconds the File eWay processes the input file again, **TimeCard_Multiple**.**xml**, and writes the results to **s1output1**.**dat** (**s1output%d.dat** where "d" is incremented with each write, depending on your Multiple Records per File setting. See Figure 43).

3 Open the output file with a text edit program and view the data to see your changes.

Chapter 4

Building a Project - Scenario2

This Tutorial provides step-by-step procedures for creating and testing an eGate Project. This Project2 addresses a business challenge in which a company uses two systems to handle time tracking and payroll distribution. As the systems use different formats, one system must use data from the other system to accomplish its tasks. This chapter explains how to structure your Project to allow the two systems to work together effectively.

What's in This Chapter

4.1 Business Challenge

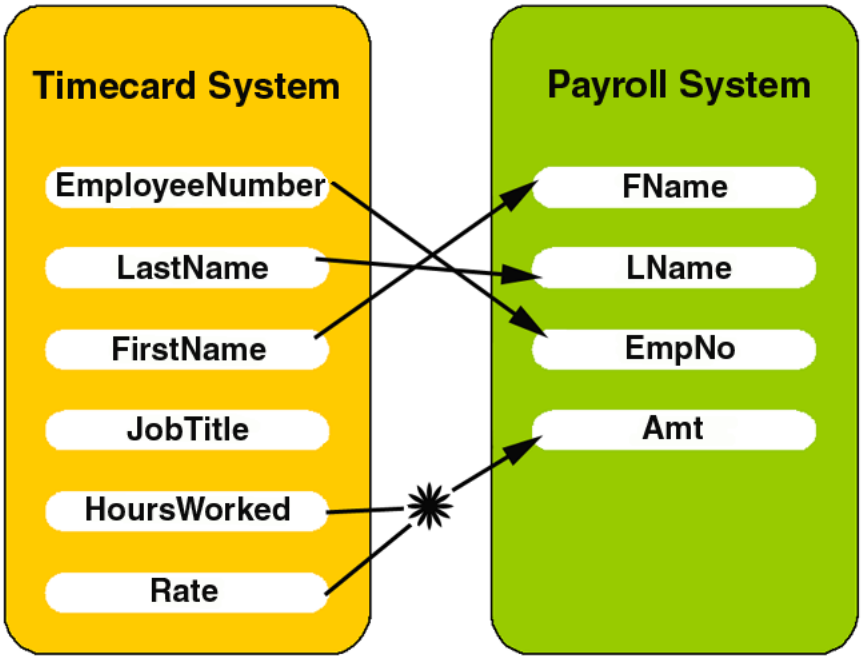
The Project described in this chapter provides a solution to the following business challenge:

- A timecard system tracks the weekly hours worked by employees. The data in this system is in XML format with six fields: **EmployeeNumber**, **LastName**, **FirstName**, **JobTitle**, **HoursWorked**, and **Rate**.
- A Payroll system is responsible for employee payment. The data in this system is in XML format with four fields: **FName**, **LName**, **EmpNo**, and **Amt**. The last element, **Amt**, is calculated by multiplying the **HoursWorked** by the **Rate**.

As shown in Figure 58 below, the following issues must be addressed to successfully integrate the systems:

- The value for the Amt field in the Payroll system must be calculated from data in the Timecard system.
- The timecard data must be formatted in such a way that it is recognized by the Payroll system.

Figure 58 The Relationship Between the Two Systems



4.2 Project Overview

Create a basic eGate Project to meet the business challenge described in **"Business Challenge" on page 65**. If you did the Scenario1 Tutorial, you may use the same Repository for Scenario2.

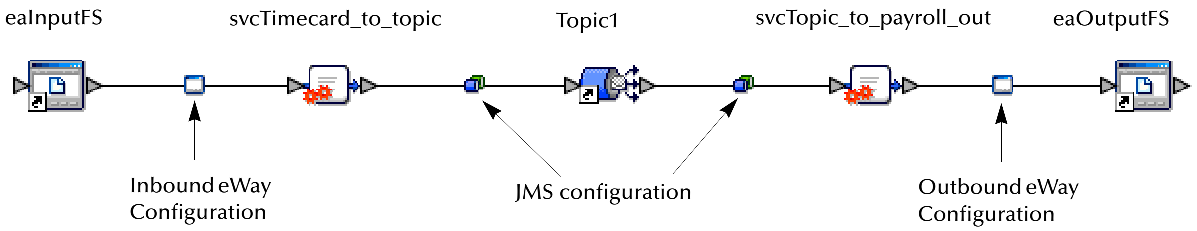
4.2.1 Required Resources

See **Chapter 3**, **ICAN Files Required for the Tutorials** on page 20 and **Chapter 3**, **Sample Data** on page 20 of this Tutorial.

4.2.2 Project Description

The finished eGate Project contains components used to process data in Project2 and publish the data in the appropriate format for the Payroll system.

Figure 59 Project Connectivity Map



These components perform the following functions:

- **eaInputFS**: The eaInputFS *External Application* represents the Timecard system containing the Input XML data file. (**eaInputFS** stands for external application input file system.)
- **Inbound eWay**: The inbound eWay polls a specified location on the file system for the input XML data file.
- **svcTimecard_to_topic**: This Service copies the input data to the **Topic1** without making any changes to the data.
- **Topic1**: This topic publishes the message from the Service **svcTimecard_to_topic** to **svcTopic_to_Payroll_out**.
- **svcTopic_to_Payroll_out**: This Service copies the input timecard elements to the appropriate output Payroll system elements. It also calculates the payroll amount.
- **Outbound eWay**: This eWay publishes the finished output data to the file system in the format appropriate for the Payroll system.
- **eaOutputFS**: This represents the Payroll system—the output data file.

4.2.3 Naming Conventions used in this Scenario

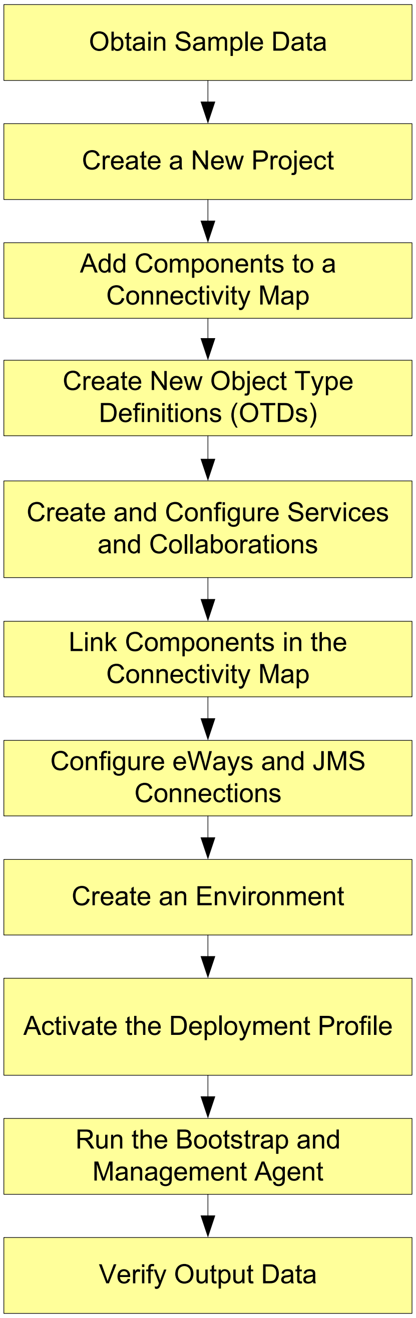
The following naming conventions are used in Scenario2 and are used in SeeBeyond's training classes.

Table 2 Naming Conventions

Component Type	Naming Convention Prefix
Repository_<additional info>	**Repository_tutorial**
Project	**prj** (e.g. **prjProject2**)
Connectivity Map	**cm** (e.g. **cmCMap2**)
Java Collaboration Definition	**jcd** (e.g. **jcdTimecardToTopic_collab**)
External Application	**ea** (e.g. **eaInputFS**)
Service	**svc** (e.g. **svcTimecard_to_topic**)

4.2.4 Project Flow Diagram - Scenario2

Figure 60 Project Flowchart - Scenario2


Obtain Sample Data
Create a New Project
Add Components to a Connectivity Map
Create New Object Type Definitions (OTDs)
Create and Configure Services and Collaborations
Link Components in the Connectivity Map
Configure eWays and JMS Connections
Create an Environment
Activate the Deployment Profile
Run the Bootstrap and Management Agent
Verify Output Data

4.3 Sample Data

This Project uses the sample files found in the Project2 folder of the **eGate Tutorial Sample**. The files used in Scenario2 are: **s2input.dtd**, **s2output.dtd**, and **Timecard.xml** (S2 in the input and output file names stands for Scenario2.) To keep it simple, the input file contains only one record. Only one record appears in the output file after you successfully run your project. See **Chapter 3**, **Sample Data** on page 20.

4.3.1 Download the Sample Files

Use the Sample Files

1. Log on to the **Enterprise Manager**.
2. Click the **Documentation** tab.
3. Click **eGate Integrator**.
4. Double-click and save the **eGate Tutorial Sample Files**.

   If you have successfully downloaded the sample files, which includes the DTDs, you can skip to **"Create New Project" on page 71**.

Create Sample Input XML File

Although it is easier to download the sample files, you can also use a text editor such as Microsoft Notepad to create the input XML file. You can also copy and paste from this document (PDF) using the text select tool from within Adobe Acrobat.

Create XML file

Your XML file should look similar to the structure and syntax below. In this scenario of the Tutorial there is only one record.

1. In Windows Explorer, create a directory for the data files. For this sample scenario, use **c:\eGateData\Project2**.
2. In Notepad, create an XML file containing the following information:

```
<?xml version="1.0" encoding="UTF-8"?>
   <Employee>
    <EmployeeNumber>100</EmployeeNumber>
    <LastName>Smith</LastName>
    <FirstName>Jane</FirstName>
    <JobTitle>Manager</JobTitle>
    <HoursWorked>40</HoursWorked>
    <Rate>50</Rate>
  </Employee>
```

3. Save the file as **Timecard.xml** in the **c:\eGateData\Project2** directory.

Create Input (DTD) File

The **s2input.dtd** file is used as a representation of the data in the Timecard system.

The **s2input.dtd** file uses the following elements:

- **EmployeeNumber**
- **LastName**
- **FirstName**
- **JobTitle**
- **HoursWorked**
- **Rate**

Create input (DTD) File

Your input (DTD) should look similar to the structure and syntax below.

1 In Notepad, create an XML file containing the following information:

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Employee (EmployeeNumber, LastName, FirstName, JobTitle,
HoursWorked, Rate)>
<!ELEMENT EmployeeNumber(#PCDATA)>
<!ELEMENT LastName(#PCDATA)>
<!ELEMENT FirstName(#PCDATA)>
<!ELEMENT JobTitle(#PCDATA)>
<!ELEMENT HoursWorked(#PCDATA)>
<!ELEMENT Rate(#PCDATA)>
```

2 Save the file as **s2input.dtd** in the **c:\eGateData\Project2** directory.

Create Output (DTD) File

The **s2output.dtd** file is used as a representation of the data in the Payroll system.

The **s2output.dtd** file uses the following elements:

- **FName**
- **LName**
- **EmpNo**
- **Amt**

Create Output (DTD) file

Your output (DTD) should look similar to the structure and syntax below.

1 In Notepad, create an XML file containing the following information:

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT Employee (FName, LName, EmpNo, Amt)>
<!ELEMENT FName(#PCDATA)>
<!ELEMENT LName(#PCDATA)>
<!ELEMENT EmpNo(#PCDATA)>
<!ELEMENT Amt(#PCDATA)>
```

2 Save the file as **s2output.dtd** in the **c:\eGateData\Project2** directory.

4.4 Create New Project

This scenario presents one way to create a Project and link the modules. Begin by creating and naming a Project in the Enterprise Designer. The sample Timecard input file used in this scenario contains only one record. The output file contains only one record.

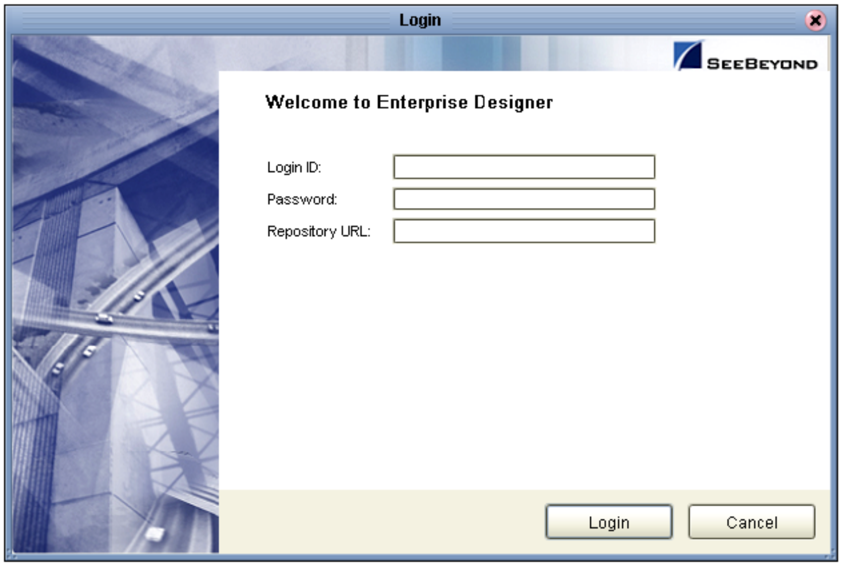
Start Enterprise Designer

Start the Enterprise Designer according to your organization's instructions.

1 Start the Enterprise Designer by executing **runed.bat** in your ican50 folder (**c:\ican50\edesigner\bin**).

The Enterprise Designer Login dialog box appears.

Figure 61 Enterprise Designer Login Dialog Box



2 Type your **Username** and **Password**. Click **Log in** to start the Enterprise Designer.

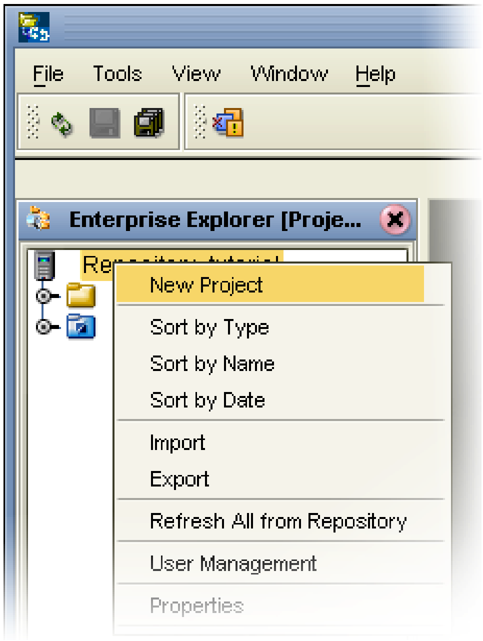
Note: *If your login dialog box doesn't appear or if you get an invalid Username or Password error, make sure your Repository is running. Also, be aware that the Username and Password are case sensitive.*

Create prjProject2

A Project is a collection of logical components, configurations, and files that are used to solve a business problem. If you have already done Scenario1, this project can be prjProject2. The "prj" is an optional preface that stands for "project."

1 In the Enterprise Explorer pane of the Enterprise Designer, right-click the Repository name (computer icon). Click **New Project**.

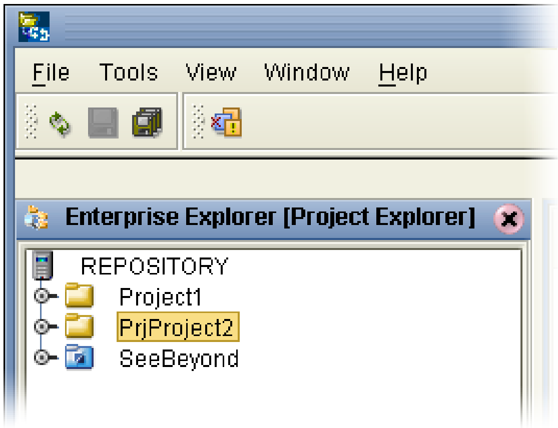
Figure 62 Create a Project



2 Name your Project, **prjProject2** and press **Enter**.

3 Your **prjProject2** appears in the Explorer pane on the left side of the window.

Figure 63 Project Folder: prjProject2



4.5 Create Connectivity Map

prjProject2 needs a Connectivity Map to define the flow between the Timecard system and the Payroll system.

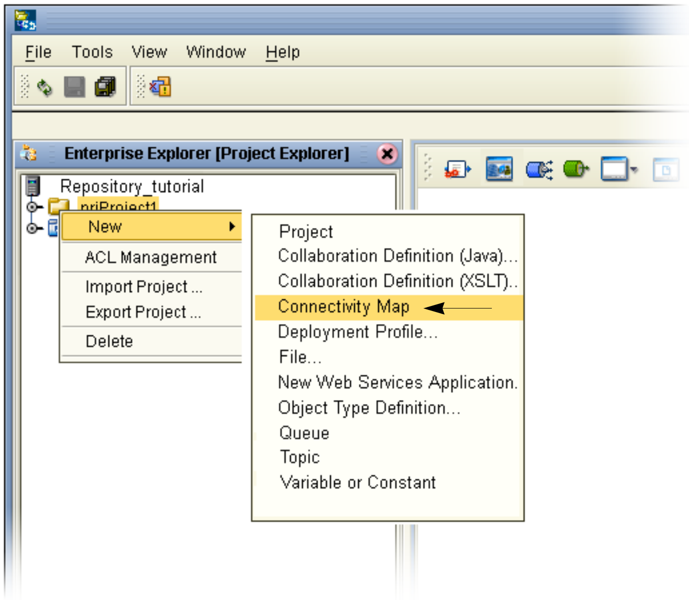
This section describes how to create the Connectivity Map for **prjProject2**.

Name Connectivity Map

A Connectivity Map contains business logic and routing information about the data transmission. The Connectivity Map is a GUI (graphical representation) of the project.

1 In the Enterprise Explorer, right-click the **prjProject2** icon, then click **New**, and **Connectivity Map**.

Figure 64 Create a Connectivity Map

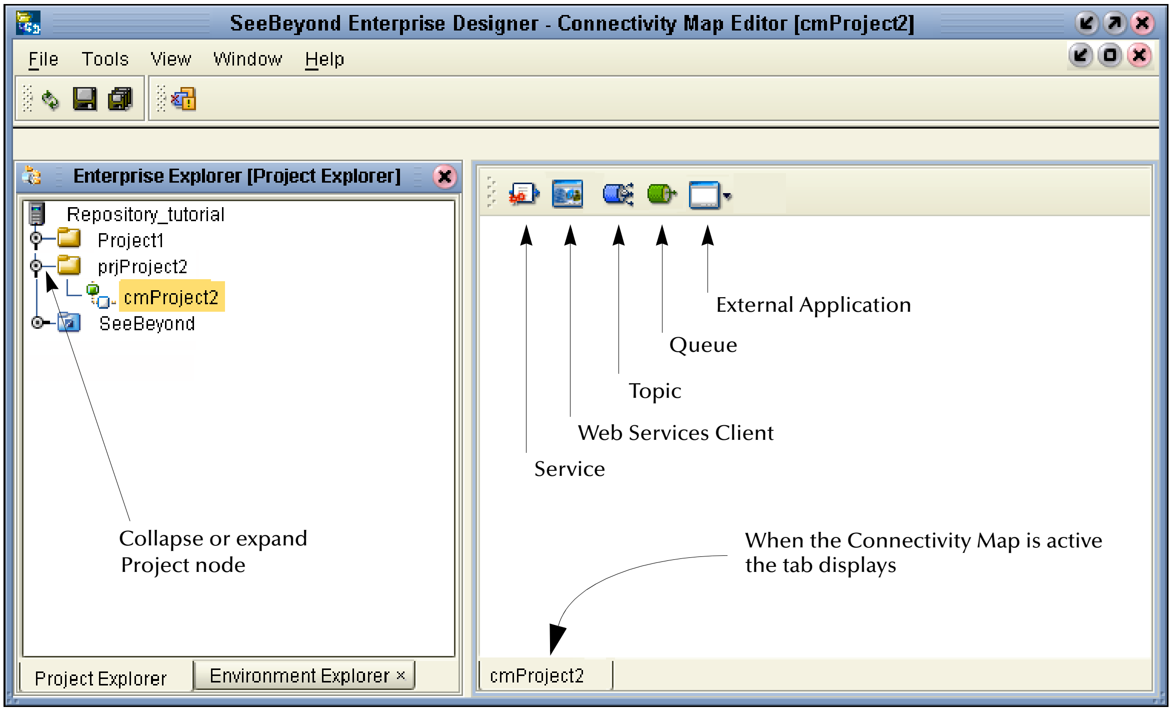


2 Name or rename your Connectivity Map, **cmProject2**.

3 Press **Enter**.

4 **cmProject2** is added to your **prjProject2**, as shown in the following figure.

Figure 65 prjProject2 with a Connectivity Map (cmCMap2)



Note: *The* ***cmProject2*** *Connectivity Map appears as an icon in the Project Explorer. A* ***cmProject2*** *tab is also added at the bottom of the Connectivity Map.*

4.6 Add Components to the Connectivity Map

In this section, add the components to the Connectivity Map. Later link the components as shown in **Figure 59 on page 66**.

4.6.1 Add Components to the Connectivity Map

In this section, add the following components to the Connectivity Map:

- **eaInputFS** (external application file system)
- **svcTimecard_to_topic**
- **Topic1**
- **svcTopic_to_Payroll_out**
- **eaOutputFS**

Create Input

In the following steps place an input file (External Application) in your Connectivity Map.

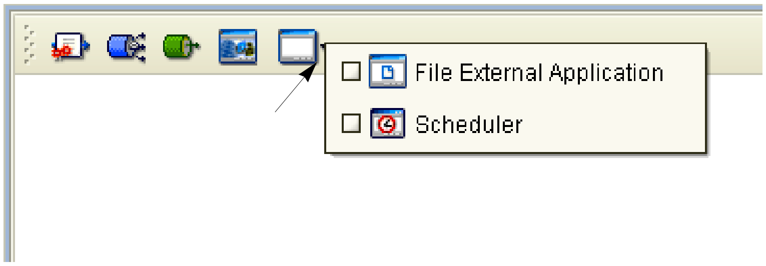
1 On the Connectivity Map toolbar, click the External Application arrow (shown in Figure 66) to display the list.

2 Click **File External Application**.

Note: *If the file option doesn't display in the drop-down list, you may need to install FileeWay.sar.*

This adds a new **File** icon to the toolbar.

Figure 66 External Application Selection



3 Drag the **File** icon from the toolbar into the left side of the Connectivity Map.

This creates a new external application called **File1**, and also adds the **File1** icon to the Project Explorer pane for your Project.

4 Name the file, **eaInputFS** (external application input file system). Start typing to name the file. Or rename files by Right-clicking on the label. (Don't leave any spaces in the file name.)

5 Press **Enter**.

Create Service, Timecard_to_topic

Create a Service to contain binding information about connecting the input Timecard to a JMS topic.

1 On the Connectivity Map toolbar, drag the **Service** icon into the Connectivity Map to the right of the **eaInputFS** icon.

This creates a new Service called **Service1**.

2 Right-click the **Service1** label and rename **svcTimecard_to_topic**.

3 Press **Enter**.

Services are components that interact through messages.

Note: *To reposition a component in the Connectivity Map, click the icon and drag it to a new location while holding down the mouse button.*

Create Topic

The JMS topic conforms to the publish-and-subscribe messaging domain, where one publisher broadcasts messages to potentially many subscribers. (In this Tutorial, which is intended to get you started quickly with eGate, topics are not used to their full potential.)

1. On the Connectivity Map toolbar, drag the Topic icon into the Connectivity Map to the right of the **svcTimecard_to_topic** icon.

   This creates a new topic called **Topic1**. Accept the default name, **Topic1**.

Create Service, svcTopic_to_payroll_out

Create a Service to contain binding information about connecting the topic to the output payroll file.

1. On the Connectivity Map toolbar, drag another **Service** icon into the Connectivity Map to the right of the **Topic1** icon.
2. Type or rename, **svcTopic_to_payroll_out**.
3. Press **Enter**.

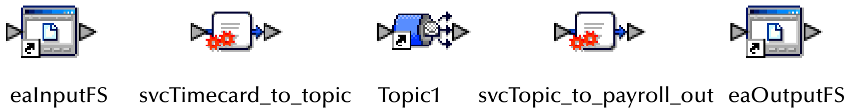
Create Output

Place an output file (external application) in your Connectivity Map.

1. Drag the **File** icon into the Connectivity Map, and place it to the right of the **svcTopic_to_payroll_out** icon.

   This creates a new external application file.
2. Name the file, **eaOutputFS** (external application output file system). Start typing to name the file. Or rename files by Right-clicking on the label. (Don't leave any spaces in the file name.)
3. Press **Enter**.
4. On the **File** menu, click **Save**.

Figure 67 Connectivity Map



Note: *In the following sections you will first configure these objects and later link them.*

4.7 Create New Object Type Definition

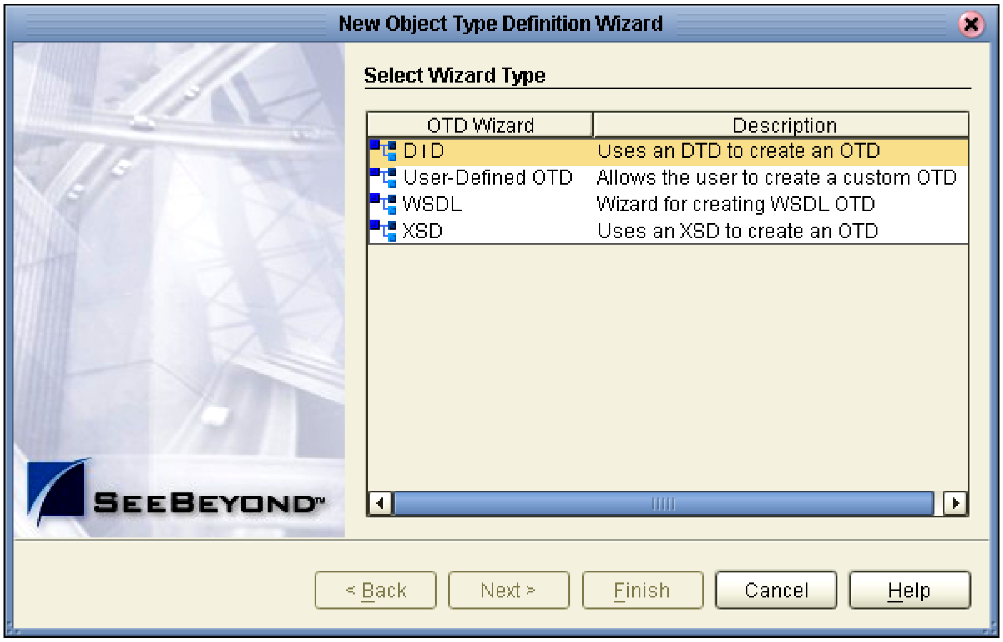
Create an Object Type Definition before you configure your Services. OTDs represent the structure of the data and are used for message parsing; they contain send/receive methods. Use the OTD wizard to make the OTD creation easy.

Create Object Type Definition

The following steps explain how to use a DTD (Document Type Definition) to create an OTD (Object Type Definition.

1. Right-click **prjProject2**.
2. Click **New**, then **Object Type Definition**.
3. Click the **DTD** item. (See the following figure.)

Figure 68 OTD Wizard Selection

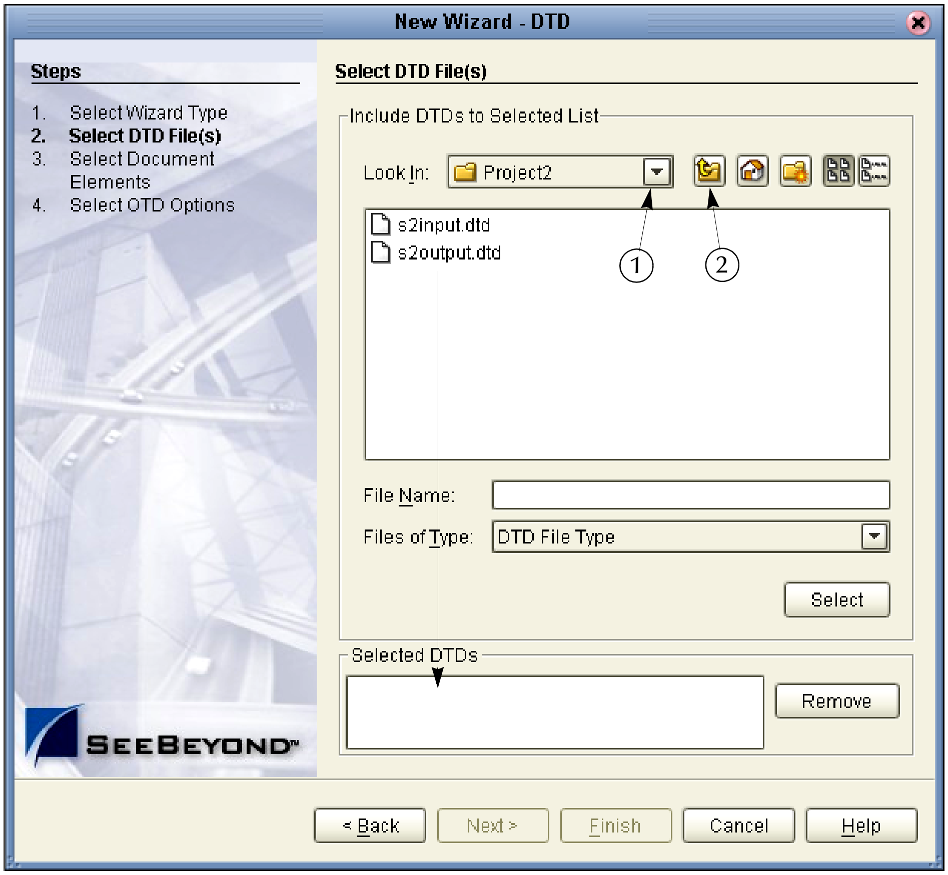


Select Wizard Type

OTD Wizard	Description
DTD	Uses an DTD to create an OTD
User-Defined OTD	Allows the user to create a custom OTD
WSDL	Wizard for creating WSDL OTD
XSD	Uses an XSD to create an OTD

4. Click **Next**.
5. Click the drop-down arrow in the Look-In dialog box (number 1 in the following figure).
6. Navigate to the sample data folder. For example, **c:\eGateData\Project2**.

Figure 69 Select DTD Files



7 Highlight **s2input.dtd** and click **Select** (or double-click).

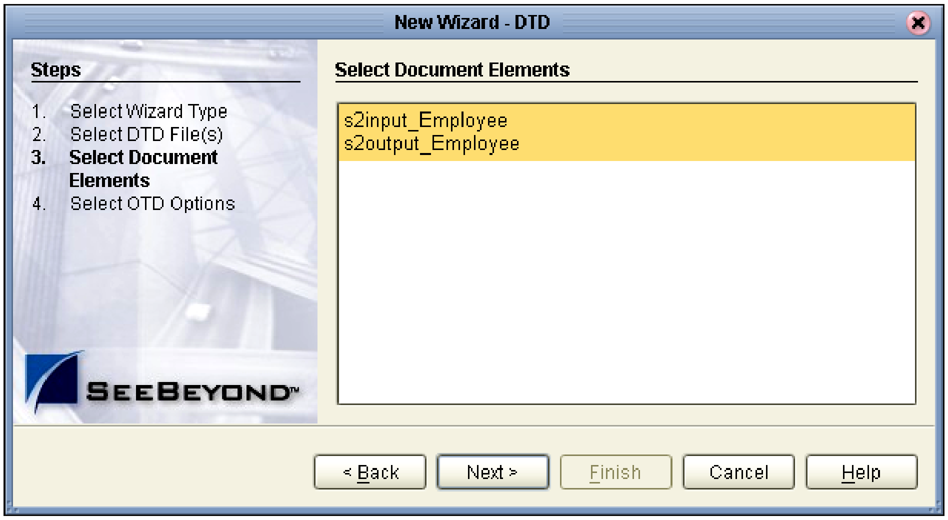
8 Highlight **s2output.dtd** and click **Select** (or double-click).

The **s2input.dtd** and **s2output.dtd** should appear in the Selected DTDs.

9 When both **s2input.dtd** and **s2output.dtd** appear in the Selected DTDs, click **Next**.

Note: *The **Up one Level** icon (number 2 in Figure 69) is used to navigate to a previous dialog box; this is useful if you have made an incorrect selection.*

Figure 70 Select Document Elements



Steps

1. Select Wizard Type
2. Select DTD File(s)
3. **Select Document Elements**
4. Select OTD Options

Select Document Elements

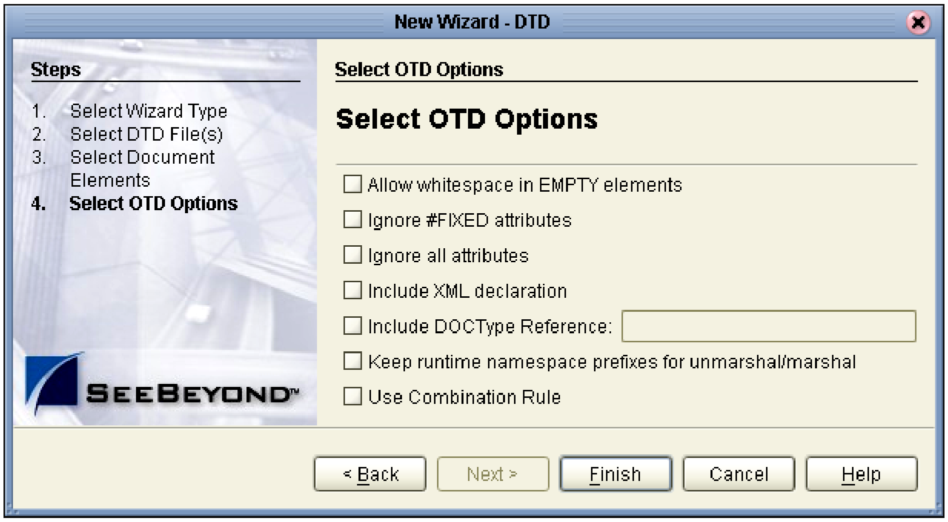
s2input_Employee
s2output_Employee

Select the document elements within the dtd files.

10 Select both **s2input_Employee** and **s2output_Employee** (hold down the CTRL key while selecting).

11 Click **Next**.

Figure 71 Select OTD Options



Steps

1. Select Wizard Type
2. Select DTD File(s)
3. Select Document Elements
4. **Select OTD Options**

Select OTD Options

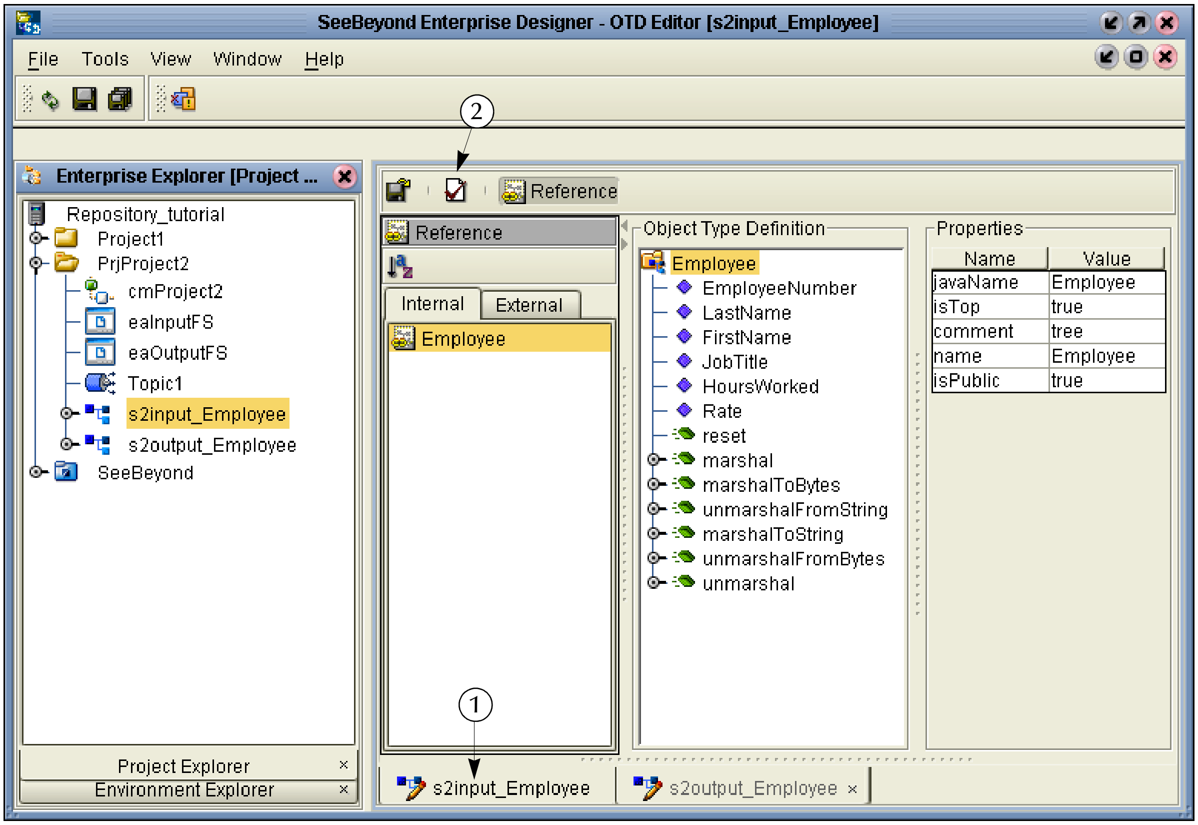
Select OTD Options

□ Allow whitespace in EMPTY elements
□ Ignore #FIXED attributes
□ Ignore all attributes
□ Include XML declaration
□ Include DOCType Reference:
□ Keep runtime namespace prefixes for unmarshal/marshal
□ Use Combination Rule

12 Click **Finish** (accept the defaults).

The OTD Editor window opens. See the following figure.

Figure 72 OTD Editor Input



Name	Value
javaName	Employee
isTop	true
comment	tree
name	Employee
isPublic	true

13 On the **File** menu, click **Save**.

Note: *__Save__ saves all your changes in your current open editor while __Save All__ saves all changes across all editors, including changes in the environments.*

Validate Input Data

Use the following steps to validate the syntax and run a test to check your DTD files. In the following steps test the input **Timecard.xml** file.

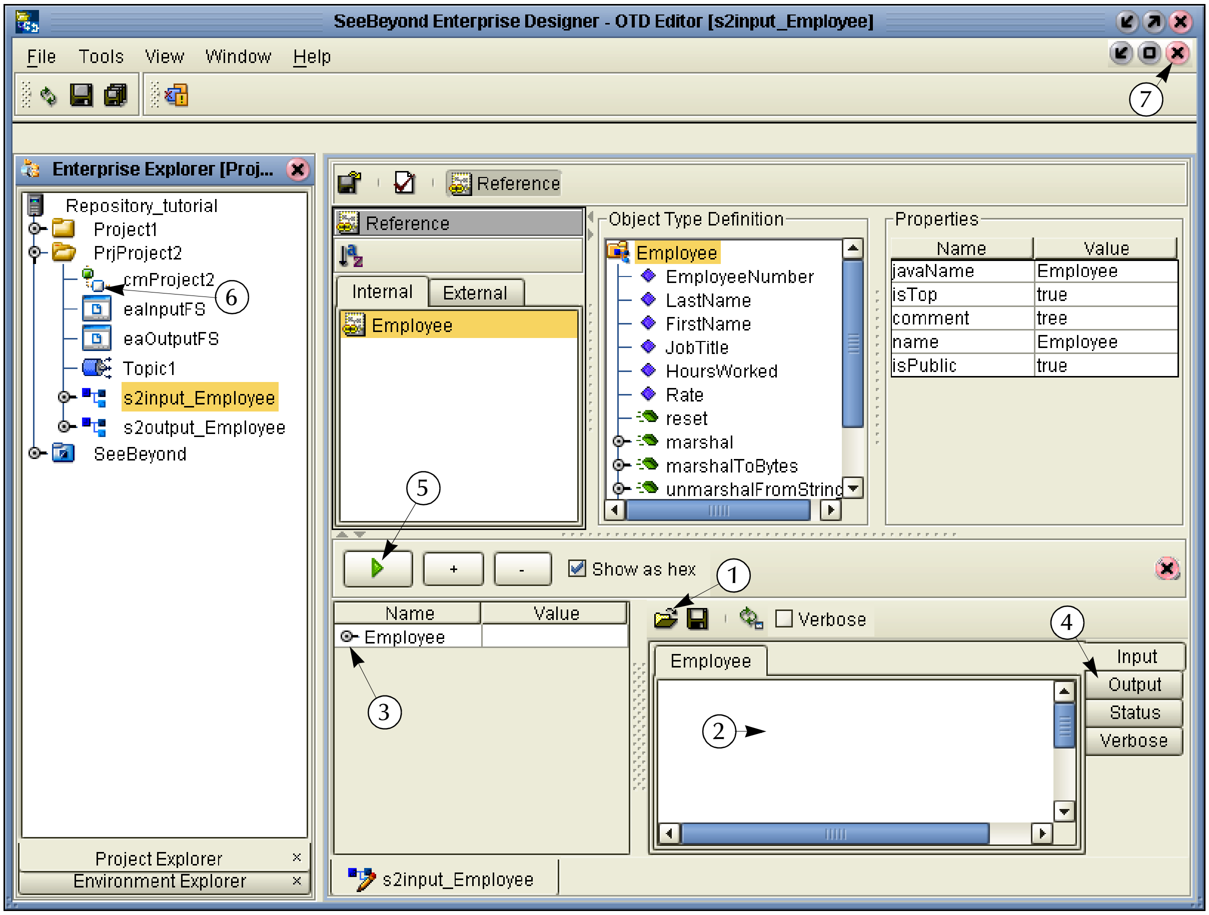
Important: *Make sure the s2input_Employee tab is selected (number 1 in Figure 72). The tab that is not selected will contain a small "x." This tab can be closed by clicking on the "x" although it is not necessary to close the tab.*

1 Click the **s2input_Employee** tab, if it is not already selected.

2 Click the **Tester** icon (number 2 in Figure 72 above).

An expanded window appears. See the following figure.

Figure 73 Test Input Data File



Properties

Name	Value
javaName	Employee
isTop	true
comment	tree
name	Employee
isPublic	true

☑ Show as hex

□ Verbose

Name	Value
Employee	

3 Click the **Open File** icon (number 1 in Figure 73 above) and navigate to your input data (**Timecard.xml**).

4 Double-click your sample input data file (Timecard.xml) to display the XML data in the window. If the file is read successfully the Status “Unmarshal Successful” appears in the Employee window (number 2 in Figure 73).

5 Click the expand view icon (number 3 in Figure 73) to expand the view of the root element **Employee**, showing the child elements (fields).

6 Click the Output button (number 4 in Figure 73). Output is empty before the test is run.

7 Click the green **Tester** icon (number 5 in Figure 73) to execute the test. If the test is successful the Status “Marshal Successful” appears in the Employee window (number 2 in Figure 73).

8 Click the Output button again to display the test results. This is not the output file but merely the “output,” meaning the results of the test of the input DTD.

If there are any errors in your file, the Status tab is populated with error messages. The status “Unmarshal Successful” means that the file can be read. (unmarshalled). The message “Marshal Successful” means that the file can be written (marshalled).

9 Double-click the **cmProject2** icon (number 6 in Figure 73**)** to redisplay the Connectivity Map (or close the window, number 7 in Figure 73).

4.8 Configure Services for Input

Use Service wizards in the following steps to configure your Collaborations.

4.8.1 Configure svcTimecard_to_topic

Configure the connectivity of the input Timecard file to the topic. The first Service, **jcdTimecardToTopic_collab**, serves only to receive data from the input file and pass it to the next service. The data is not processed in this Collaboration.

Create Web Service Interface

Web Services enable communication between diverse applications using the Internet. Refer to *eGate Integrator User's Guide* for information about Web Services.

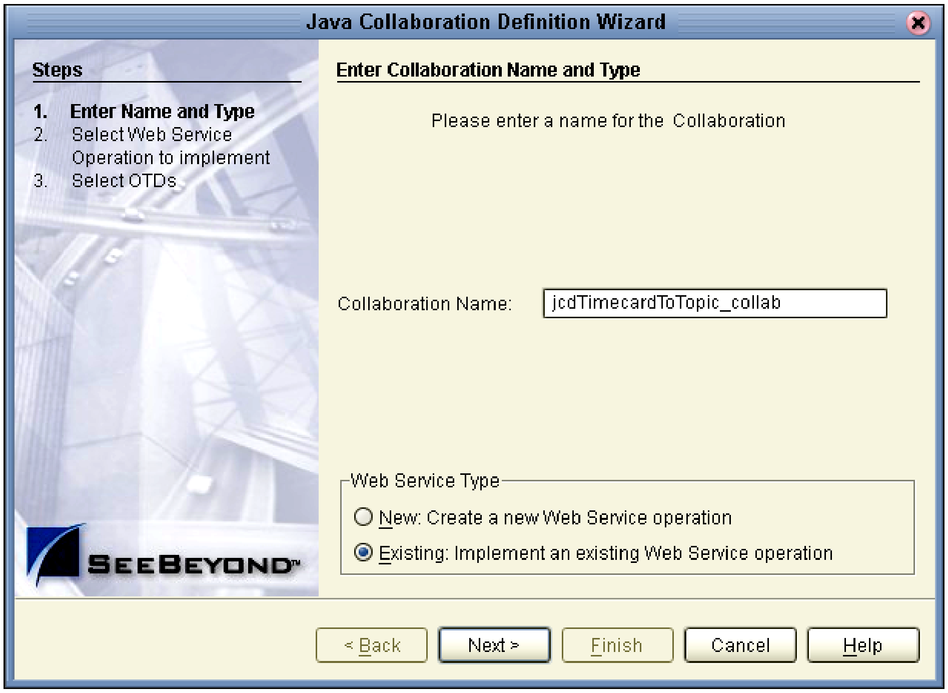
1 Right-click **prjProject2**.

2 Click **New**, then **Collaboration Definition (Java)**.

The Collaboration Definition Wizard (Java) appears.

3 Type **jcdTimecardToTopic_collab** as the name for the Java Collaboration Definition. (The prefix jcd stands for Java Collaboration Definition.)

Figure 74 New Collaboration Definition (Java) Name



Java Collaboration Definition Wizard
Steps
1. Enter Name and Type
2. Select Web Service Operation to implement
3. Select OTDs
Enter Collaboration Name and Type

Please enter a name for the Collaboration

Collaboration Name: jcdTimecardToTopic_collab

Web Service Type

○ New: Create a new Web Service operation

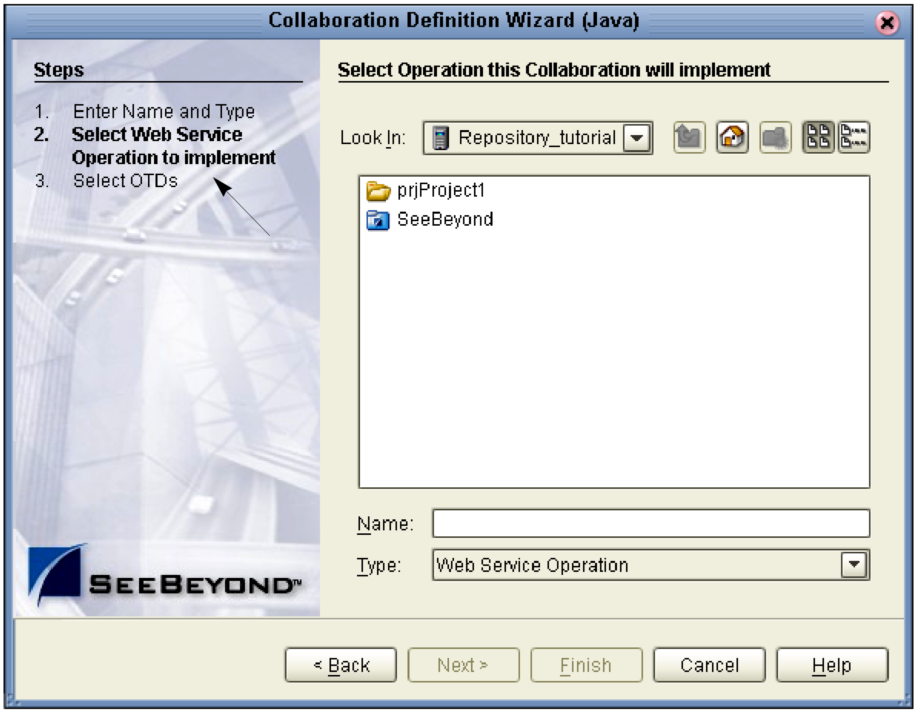
● Existing: Implement an existing Web Service operation

SEEBEYOND
< Back
Next >
Finish
Cancel
Help

Note: *Accept the Web Serice Type default, Existing: Implement an existing Web Service operation.*

4 Click **Next**.

Figure 75 New Collaboration Definition Wizard (Java)



Select a web service to “implement” a file-read Service, then select an OTD.

5 Double-click **SeeBeyond**.

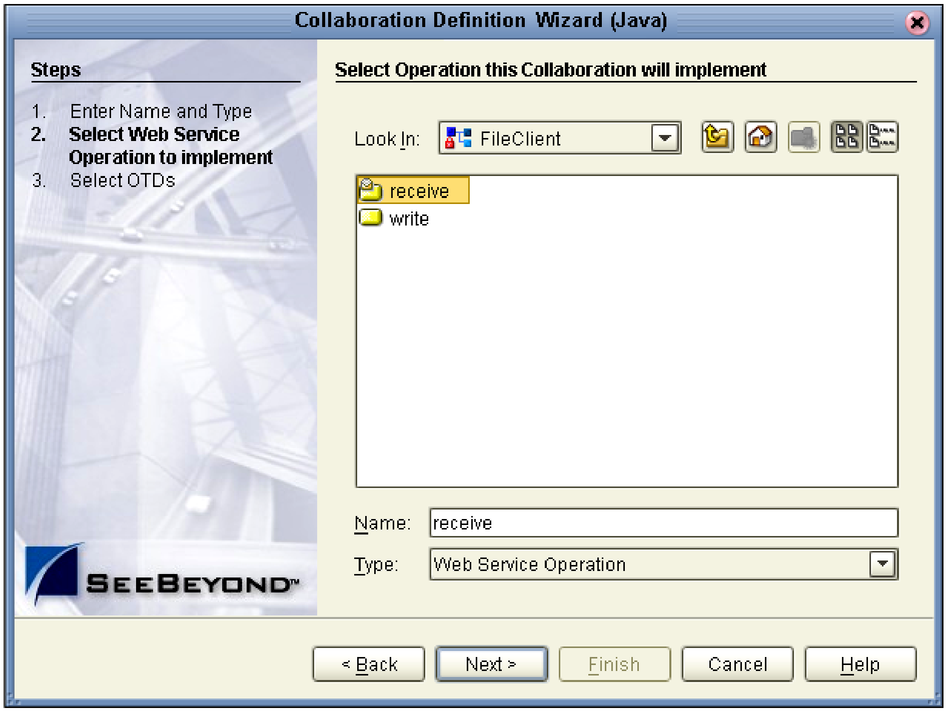
6 Double-click **eWays**. eWays are message end points that connect to an external service provider (system, person, or organization) to enable the sending and receiving of messages.

7 Double-click **File**.

8 Double-click **FileClient**.

9 Click **receive**. Notice that **receive** appears in the Name field as shown in the following figure.

Figure 76 New Collaboration Web Service Interface


Collaboration Definition Wizard (Java)

Steps

1. Enter Name and Type
2. **Select Web Service Operation to implement**
3. Select OTDs

Select Operation this Collaboration will implement

Look In: FileClient

- receive
- write

Name: receive

Type: Web Service Operation

10 Click **Next** when “receive” appears in the File Name field.

The New Collaboration Definition Wizard (Java) refreshes, and you can select OTDs. The existing web service **FileClient.receive** is used to bring data in.

Select OTDs

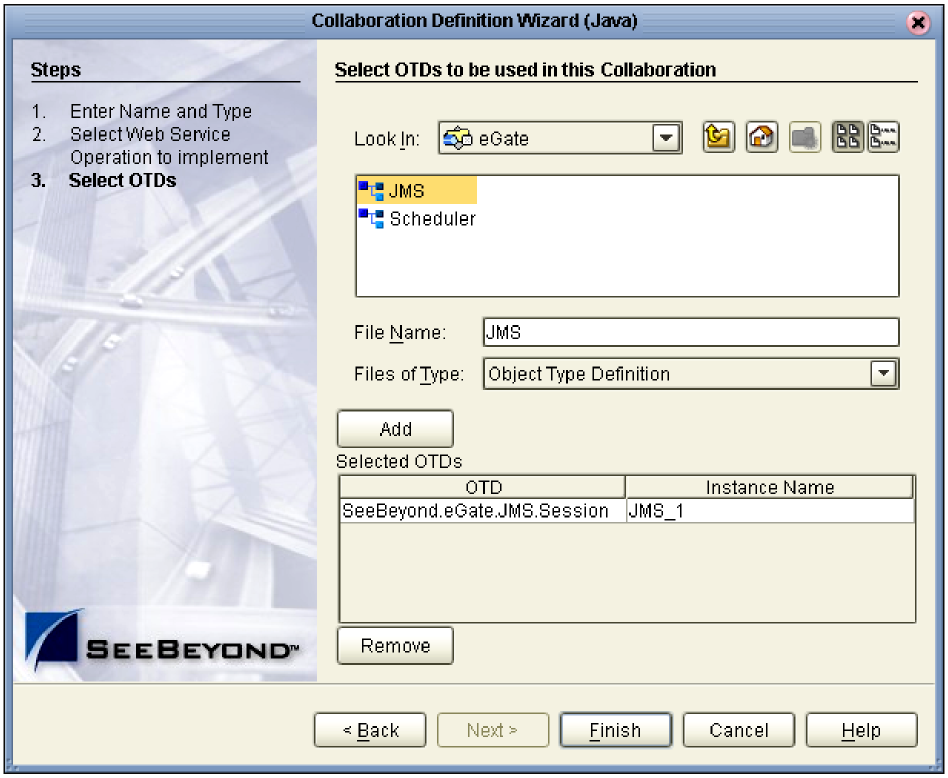
The following steps set up the OTD for the existing web Service to write to JMS. The **JMS.receive** is used to route data between Topics (and Queues). You are still using the Collaboration Definition Wizard (Java).

1 Double-click **SeeBeyond**.

2 Double-click **eGate**.

3 Double-click **JMS**. **JMS** appears in the **Selected OTDs** field as shown in the following figure.

Figure 77 New Collaboration Select OTD



Steps

1. Enter Name and Type
2. Select Web Service Operation to implement
3. **Select OTDs**

Select OTDs to be used in this Collaboration

Look In: eGate

- JMS
- Scheduler

File Name: JMS

Files of Type: Object Type Definition

Selected OTDs

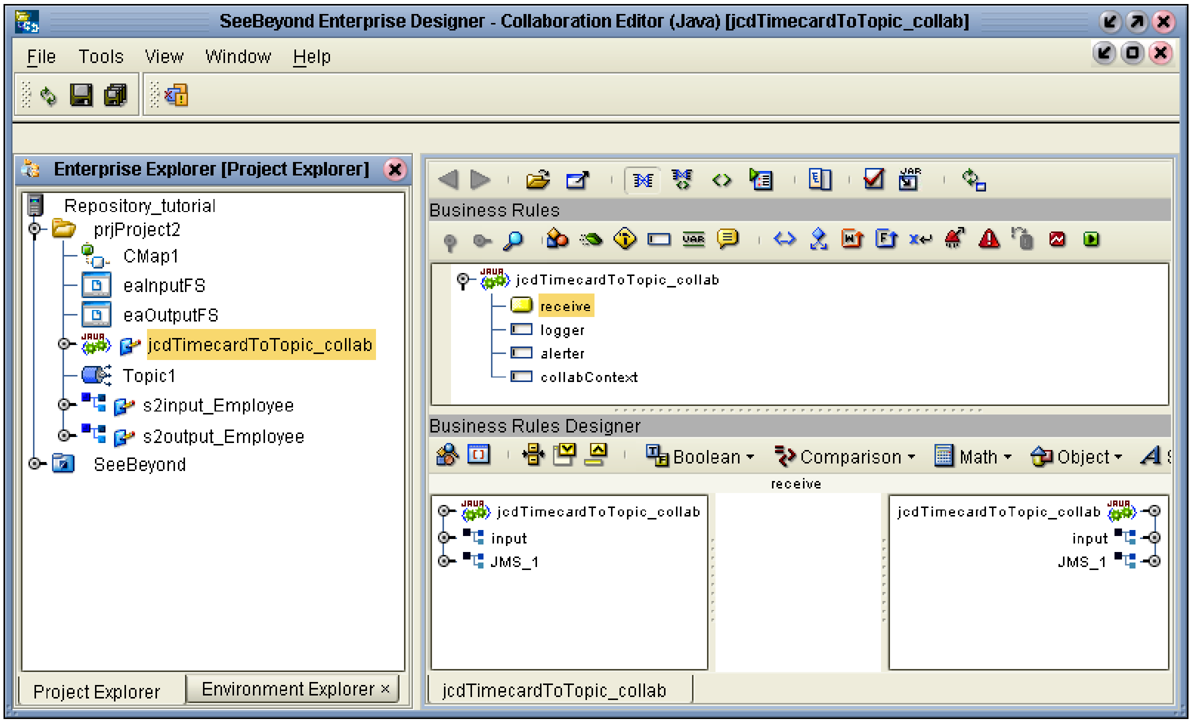
OTD	Instance Name
SeeBeyond.eGate.JMS.Session	JMS_1

4 Click **Finish**.

After a few seconds, the **jcdTimecardToTopic_collab** icon appears in the Project Explorer tree in the left pane. Later, this Collaboration is bound to the Service, **svcTimecard_to_topic**, in the Connectivity Map, using the drag and drop method.

The Business Rules Designer window appears in the top pane. See the following figure.

Figure 78 Business Rules Designer



Note: *Notice that the objects in Figure 78 reside in both the left and right pane of the Business Rules Designer window. Keep in mind that the GUIs (**JMS_1** for example) refer to the same object although they are shown in the GUI in two different places at the same time.*

4.9 Apply Business Rules for jcdTimecardToTopic_collab

In **jcdTimecardToTopic_collab** connect the input from the Timecard to the JMS OTD. The JMS passes the message through to the next Service. The processing of the message occurs in the next service, **jcdTopicToPayrollOut_collab**.

Send Input to Next Service

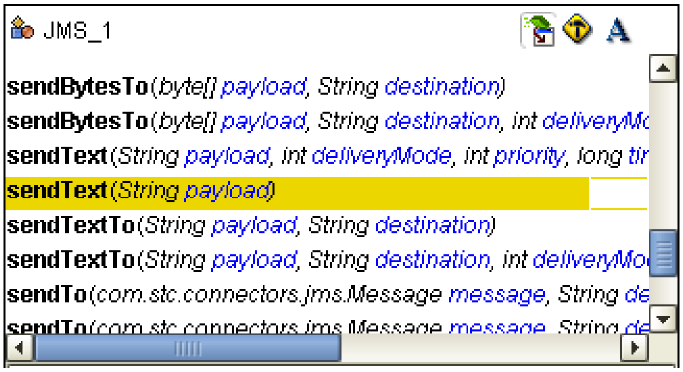
In these steps select the a Send method to pass the message on to the next service.

1 Click and drag **JMS_1** from the left pane to the center pane of the Business Rules Designer.

This action causes the methods dialog box to appear.

Note: *You can also right-click **JMS_1** and click **Select a method to call**.*

Figure 79 Methods Dialog Box



JMS_1

sendBytesTo(byte[] payload, String destination)
sendBytesTo(byte[] payload, String destination, int deliveryMc
sendText(String payload, int deliveryMode, int priority, long tir
sendText(String payload)
sendTextTo(String payload, String destination)
sendTextTo(String payload, String destination, int deliveryMo
sendTo(com.stc.connectors.jms.Message message, String de
sendTo(com.stc.connectors.jms.Message message, String de

2 Scroll down to and double-click the method **sendText(String payload))** from the pop-up dialog box.

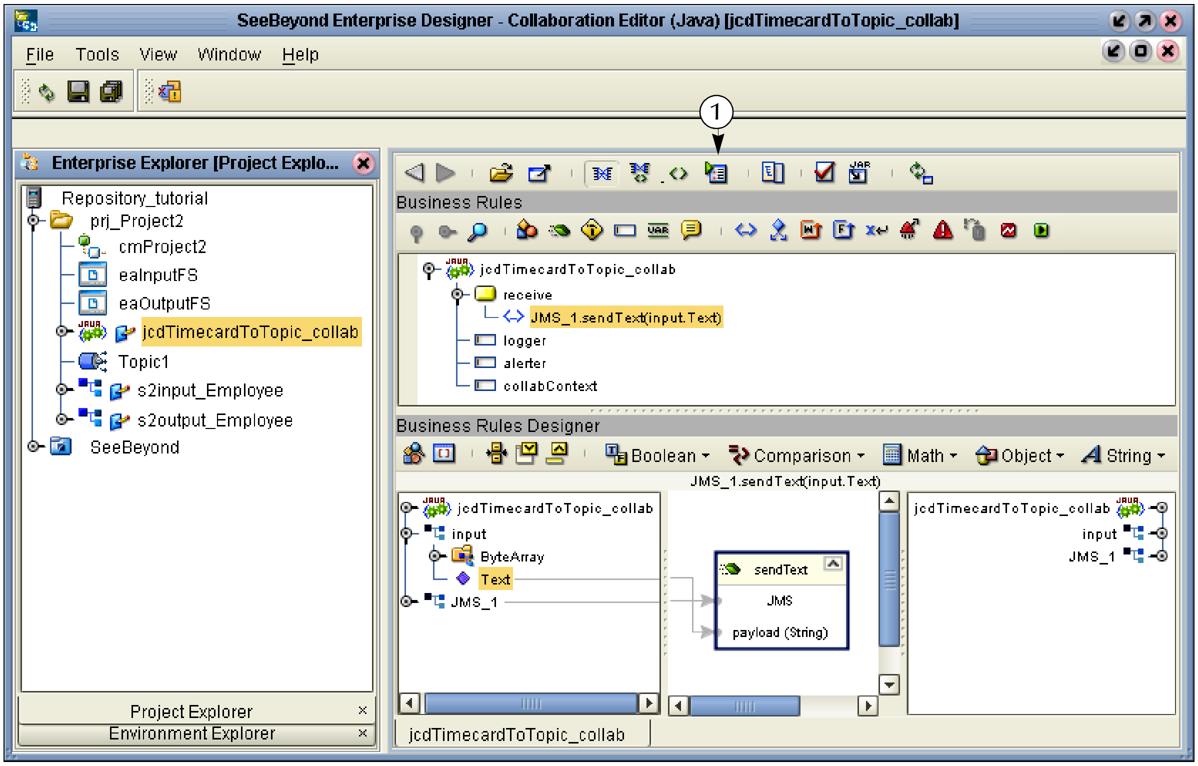
A **sendText** box appears in the center pane.

3 Double-click **input** (or expand **input** to expose the **Text** icon) in the left pane.

4 Connect **Text** from **input** to **payload(String)** in the **SendText** box.

See the following figure.

Figure 80 Send Text Payload



Note: *Refer to number 1 Figure 80: This is the JCE tester. If it is depressed, the tester GUI appears to the left of the Business Rules pane. With the tester you can debug your Java collaboration, and you can step through the JCE one line at a time and see the value of your variables update. The tester is not used in the tutorials.*

5 On the **File** menu, click **Save All**.

6 Click the **Window** menu. Click **Close All**.

7 Double-click **cmProject2** to redisplay the Connectivity Map.

This completes the setup for the **jcdTimecardToTopic_collab** collaboration definition; this Service only passes the input data to the next Service where the real processing is done.

You are now ready to drag-and-drop the Collaboration into your Connectivity Map Service, but first configure **svcTopic_to_payroll_out**, in the next section.

4.10 Configure Services for Output

Select a Web Service Operation

Configure your Web Service interface.

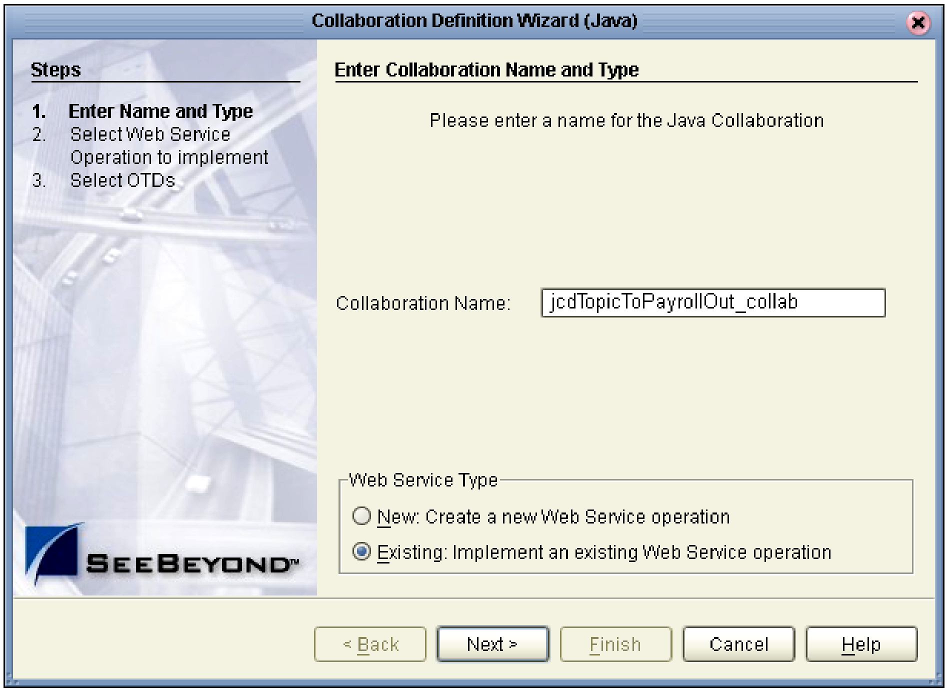
1 Right-click **prjProject2**.

2 Click **New**, then **Collaboration Definition (Java)**.

3 Type **jcdTopicToPayrollOut_collab** as the name for the Collaboration Definition (Java). (Also accept the Web Service Type: Existing: Implement an existing Web Service operation.)

See the following figure.

Figure 81 New Collaboration Definition Editor (Java)



Collaboration Definition Wizard (Java)
Steps
1. Enter Name and Type
2. Select Web Service Operation to implement
3. Select OTDs
Enter Collaboration Name and Type
SEEBEYOND™
< Back
Next >
Finish
Cancel
Help

Please enter a name for the Java Collaboration

Collaboration Name: jcdTopicToPayrollOut_collab

Web Service Type

○ New: Create a new Web Service operation

● Existing: Implement an existing Web Service operation

4 Click **Next**.

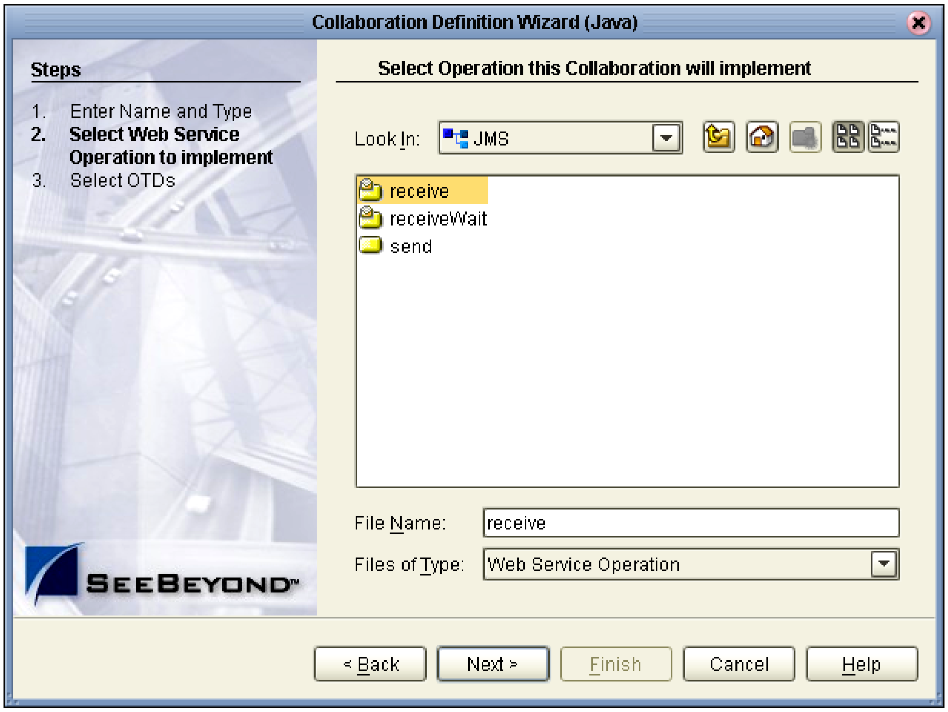
5 From the Collaboration Definition Wizard (Java), double-click **SeeBeyond**.

6 Double-click **eGate**.

7 Double-click **JMS**. This is your source for this Collaboration.

8 Click **receive**. (receive is a method/Web Service used to read a message from a topic that your collaboration implements.)

Figure 82 Receive from Topic



Collaboration Definition Wizard (Java)

Steps

1. Enter Name and Type
2. Select Web Service Operation to implement
3. Select OTDs

Select Operation this Collaboration will implement

Look In: JMS

receive
receiveWait
send

File Name: receive

Files of Type: Web Service Operation

9 Click **Next**.

The Collaboration Definition Wizard (Java) refreshes. Now select input and output OTDs.

Select OTDs

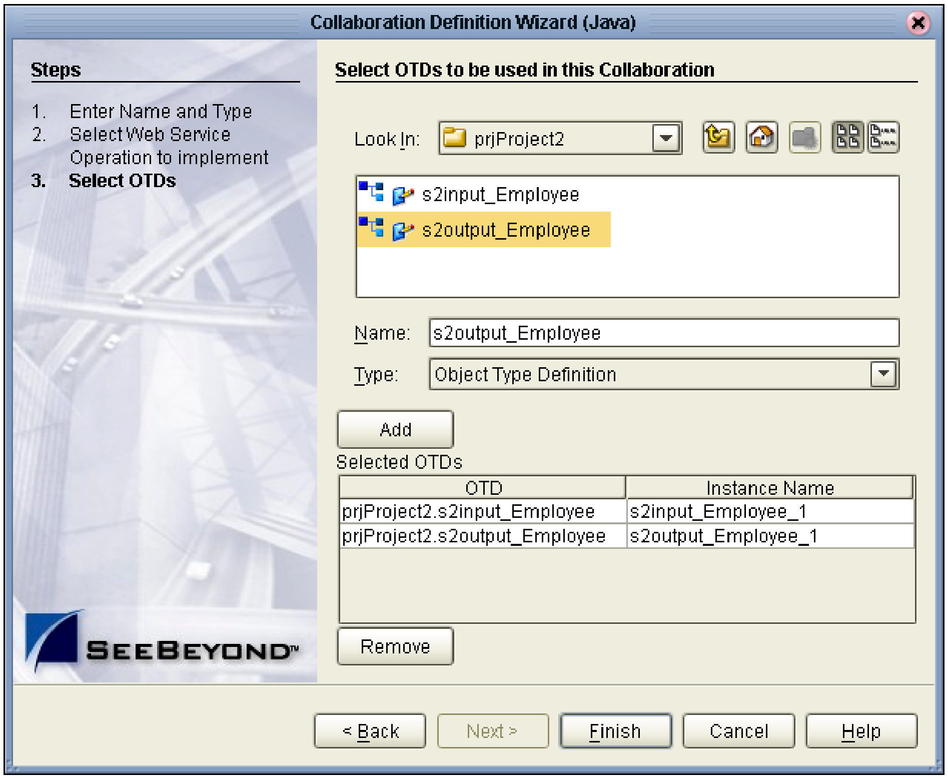
1 Double-click **prjProject2** in the Collaboration Definition Wizard (Java).

The Collaboration Definition Wizard (Java) refreshes, showing the input and output options.

2 Double-click **s2input_Employee** to move the selection to the list of Selected OTDs.

3 Double-click **s2output_Employee** to move the selection to the list of Selected OTDs.

Figure 83 Input/Output OTDs for jcdTopicToPayrollOut_collab



Collaboration Definition Wizard (Java)

Steps

1. Enter Name and Type
2. Select Web Service Operation to implement
3. **Select OTDs**

Select OTDs to be used in this Collaboration

Look In: prjProject2

s2input_Employee
s2output_Employee

Name: s2output_Employee

Type: Object Type Definition

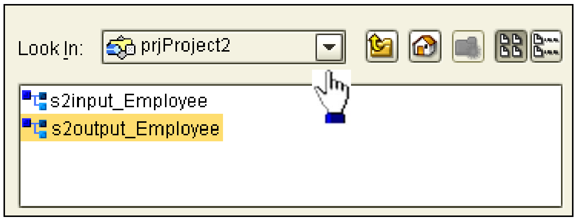
Selected OTDs

OTD	Instance Name
prjProject2.s2input_Employee	s2input_Employee_1
prjProject2.s2output_Employee	s2output_Employee_1

Note: Before clicking Finish, select ***FileClient*** *which represents your output to an external file. This is explained in the following steps.*

4 Click the **drop-down arrow** for the Look In list.

Figure 84 Drop Down Arrow button



Look In: prjProject2

s2input_Employee
s2output_Employee

5 Click **SeeBeyond**.

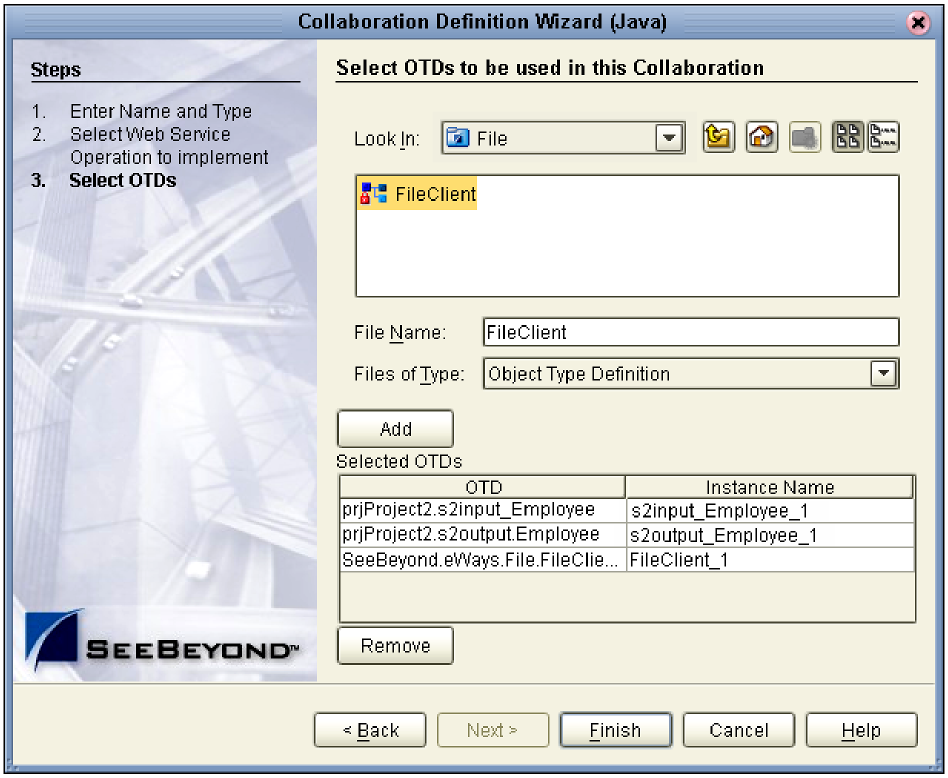
6 Double-click **eWays**.

7 Double-click **File**.

8 Double-click **FileClient**.

Notice that the **FileClient_1** is added to the list of **Selected OTDs**. See the following figure.

Figure 85 OTDs for jcdTopicToPayrollOut_collab


Collaboration Definition Wizard (Java)
Steps
1. Enter Name and Type
2. Select Web Service Operation to implement
3. Select OTDs
SEEBEYOND
Select OTDs to be used in this Collaboration
Look In:
File
FileClient
Add
Remove
< Back
Next >
Finish
Cancel
Help

File Name: FileClient

Files of Type: Object Type Definition

Selected OTDs

OTD	Instance Name
prjProject2.s2input_Employee	s2input_Employee_1
prjProject2.s2output.Employee	s2output_Employee_1
SeeBeyond.eWays.File.FileClie...	FileClient_1

9 Click **Finish**.

After a few seconds, the **jcdTopicToPayrollOut_collab** appears in the Enterprise Explorer tree in the left pane. The Business Rules Designer Collaboration Editor also appears. See Figure 86.

4.11 Apply Business Rules for jcdTopicToPayrollOut_collab

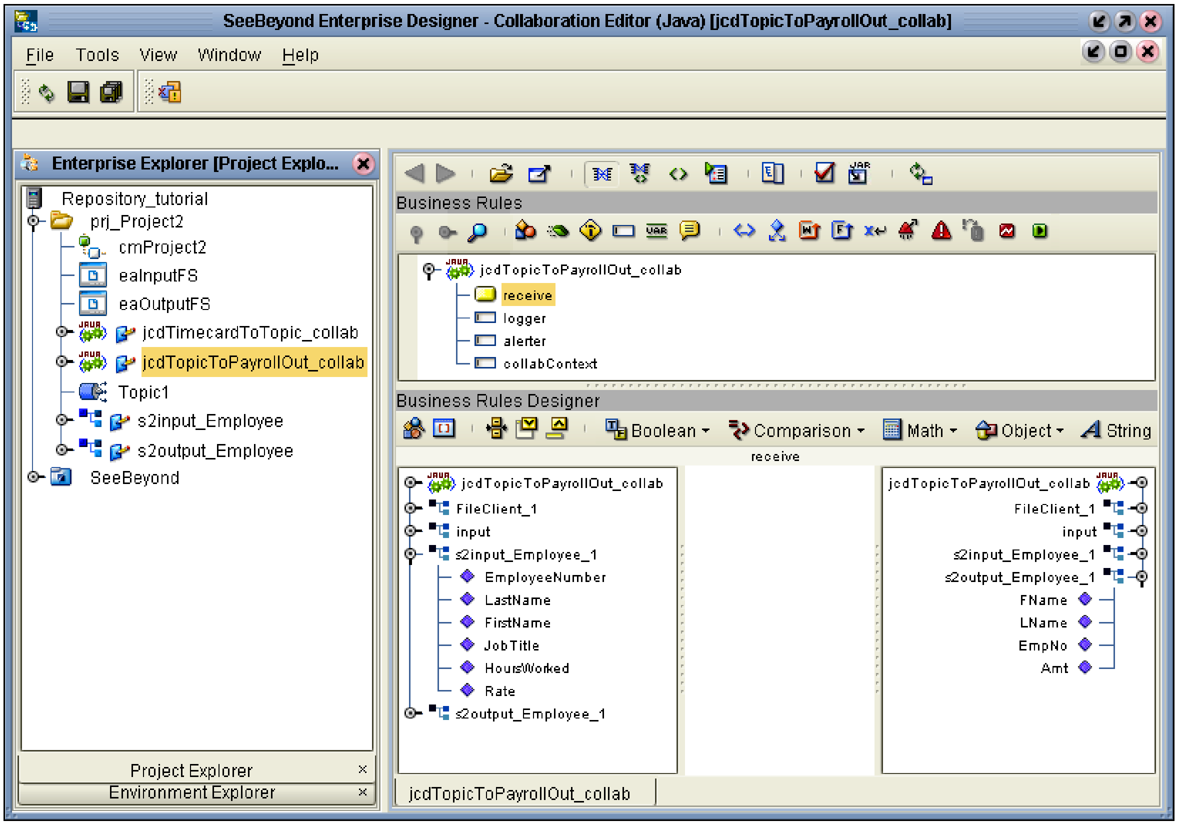
The following is a summary of what you will do in this section.

Overview

- Receive a message from **JMS_1** and unmarshal the string into OTD **s2input_Employee_1**.
- Copy data from the nodes, **EmployeeNumber**, **LastName** and **FirstName** to the corresponding nodes in **s2output_Employee_1** OTD.
- Set up the multiplication logic (Business Rules) to obtain an amount (**Amt**) for **s2output_Employee_1**.
- Marshal the data from **s2output_Employee_1** to a string and write it to a file.

See the following Figure 86, of the Collaboration Editor dialog box, showing an expanded view of the input and output Employee files.

Figure 86 Business Rules Designer for jcdTopicToPayrollOut_collab



Unmarshal Text

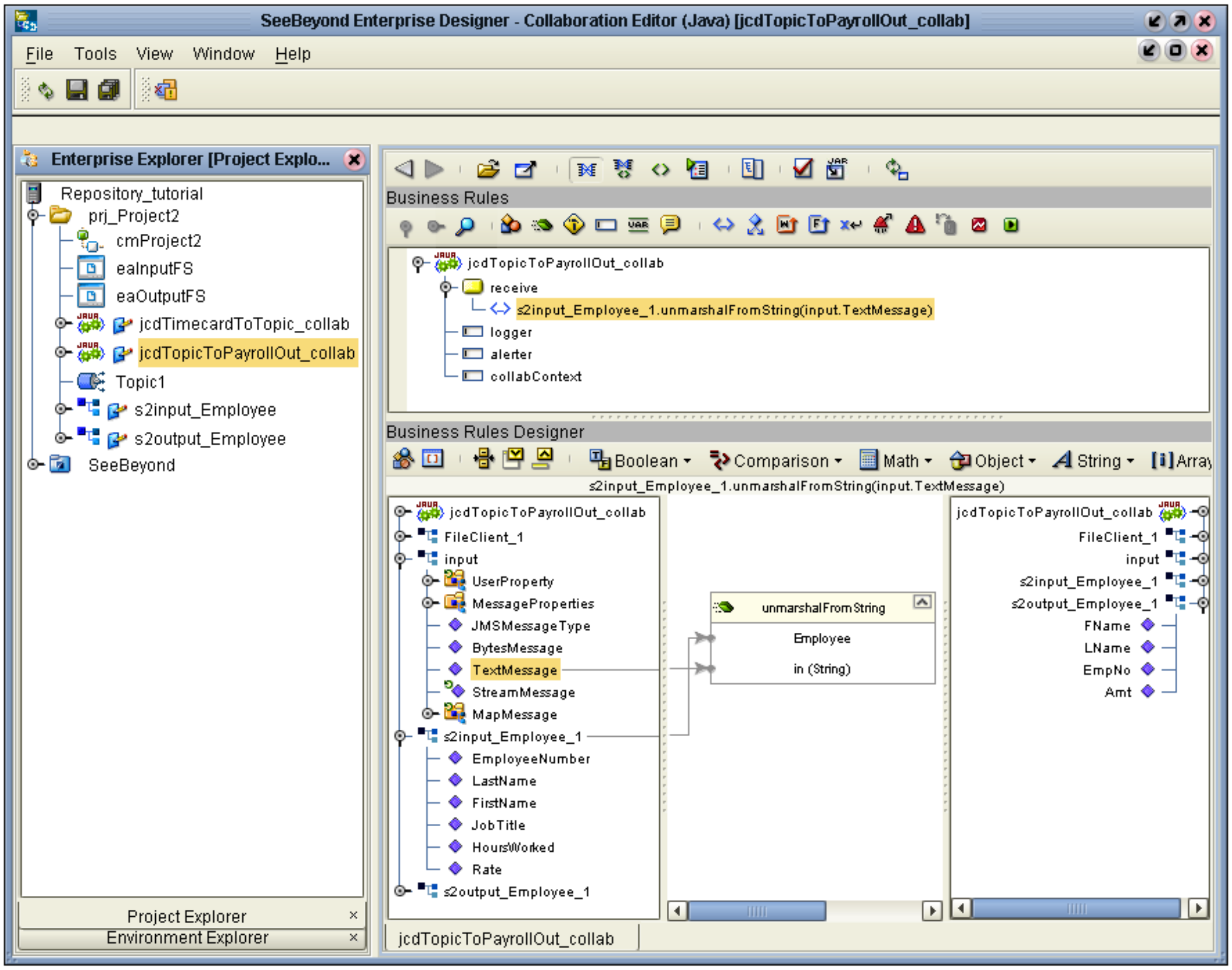
First unmarshal the text input from the topic into the input OTD, as explained in the following steps.

1. Double-click **input** in the left pane of the Business Rules Designer to expand the view, (or click on the expansion node icon ⊙).
2. Double-click **s2input_Employee_1** in the left pane of the Business Rules Designer to expand the view, as shown in the figure above.
3. Click and drag **s2input_Employee_1** from the left to the center pane of the Business Rules Designer. (You can also right-click **s2input_Employee_1** and select a method to call.)

   A method list appears.
4. Double-click the method **unmarshalFromString(String in)** from the pop-up list.

   This action places a method box into the center pane of the Business Rules Designer window.
5. Connect a node from **input**, **TextMessage** to the method box **UnmarshalFromString , in (String)**. See the following figure.

Figure 87 Collaboration Editor (Java) - Unmarshal From String



```
s2input_Employee_1.unmarshalFromString(input.TextMessage)
```

Connect Input and Output OTD Nodes

As you connect nodes in the following steps, the GUI rule disappears, but the record of your action appears in the Business Rules pane that is just above the Business Rules Designer pane.

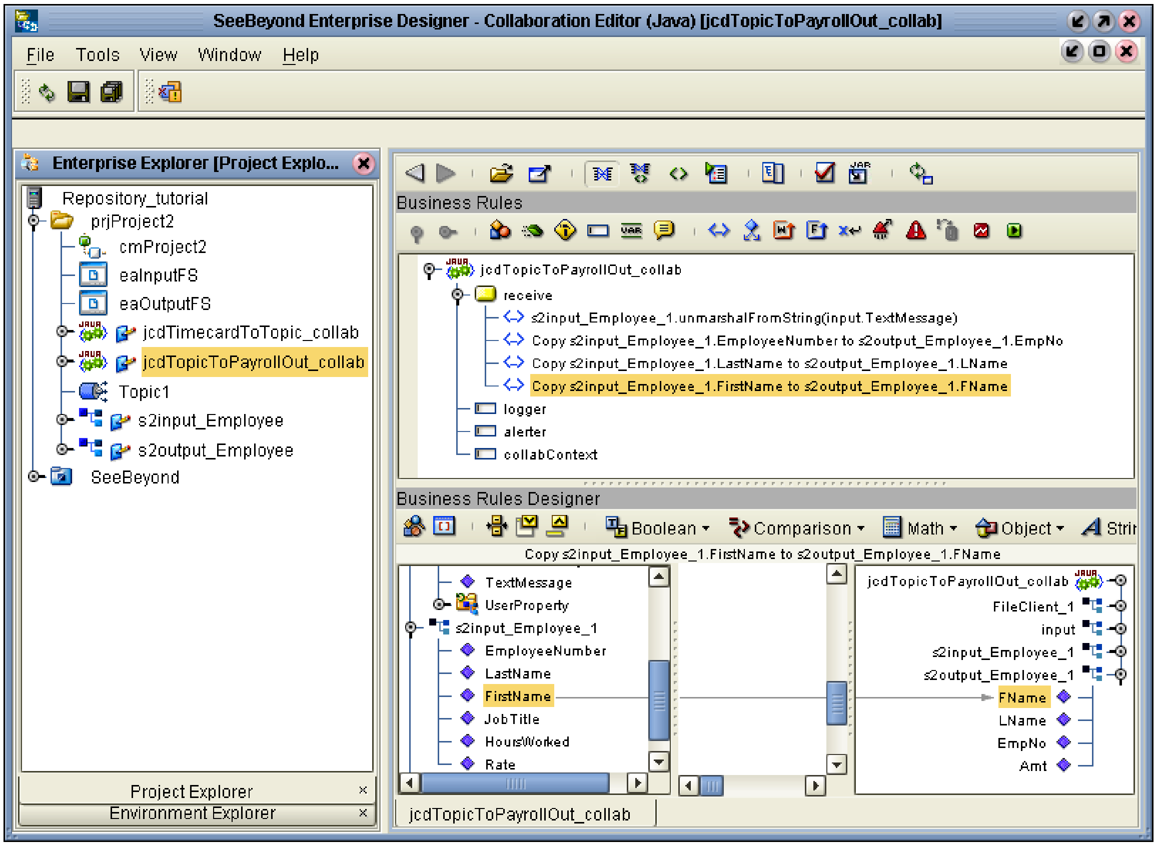
1. Expand the view of **s2input_Employee_1** in the left pane (if not already expanded).
2. Expand the view of **s2output_Employee_1** in the right pane (if not already expanded).

   In the following steps, connect nodes from the left to the right pane of the Business Rules Designer.
3. Connect **EmployeeNumber** to **EmpNo**.
4. Connect **LastName** to **LName**.
5. Connect **FirstName** to **FName**.

   Job Title is not used.

Note: *To delete an object, right-click the object and select Delete. You can delete a rule, method box or other GUI object.*

Figure 88 Connect Input/Output OTD Nodes



s2input_Employee_1.unmarshalFromString(input.TextMessage)
Copy s2input_Employee_1.EmployeeNumber to s2output_Employee_1.EmpNo
Copy s2input_Employee_1.LastName to s2output_Employee_1.LName
Copy s2input_Employee_1.FirstName to s2output_Employee_1.FName

Copy s2input_Employee_1.FirstName to s2output_Employee_1.FName

Create Multiplication Operation

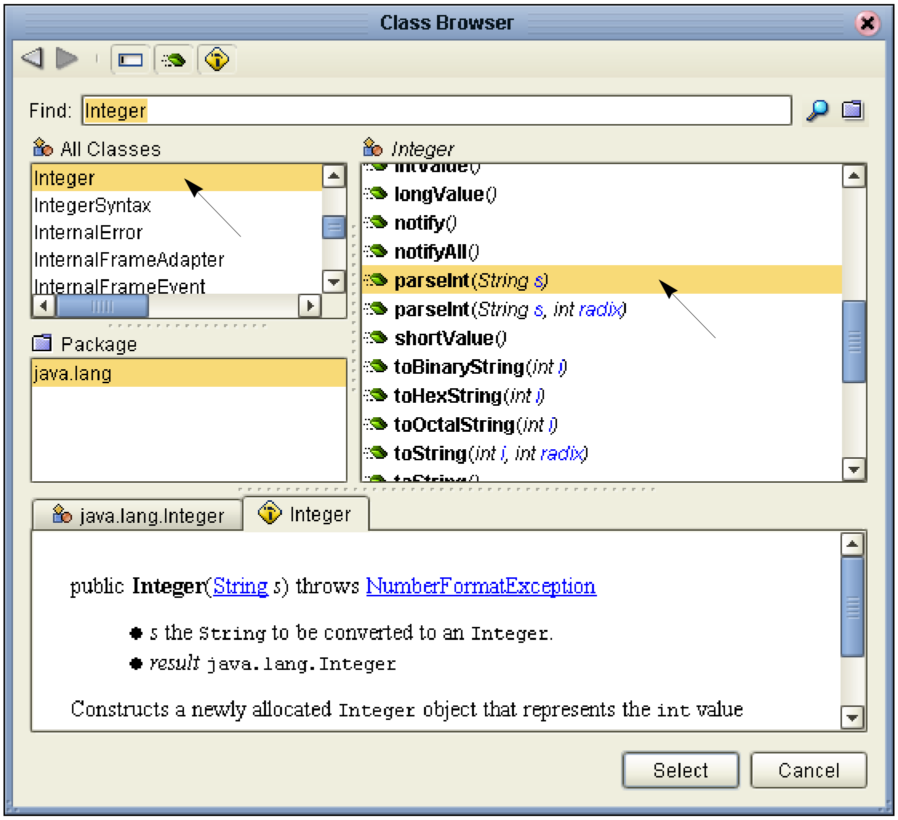
In the following steps, convert string values to Integer so the values can be used in the multiplication operation.

1. Click the **Class Browser** icon to display a list of Classes.

2. Type the Class name, **Integer**, and click the **Search** icon .

   The Class Browser dialog box appears.

Figure 89 Class Browser - parseInt(String s)



Find: Integer

public **Integer**(String s) throws NumberFormatException

- s the String to be converted to an Integer.
- *result* java.lang.Integer

Constructs a newly allocated Integer object that represents the int value

3 Click **Integer** in the left pane, then select **parseInt(String s)**in the list of methods.

4 Click **Select**.

5 Repeat the previous steps because you will convert two integers to string (hours worked and rate).

The method boxes may overlap in your workspace. Be sure to locate these boxes and drag them to a location where they do not overlap. See the positioning of the boxes in Figure 91.

Place these two boxes on the left side of the workspace in the center pane.

6 Click the **Math** drop-down menu and click the Multiplication Operator ✳ .

The Multiplication Method box appears. See Figure 91.

Note: *You can adjust the size of the window panes by clicking on the frame and adjusting. Refer to the following figure.*

Figure 90 Window Adjust

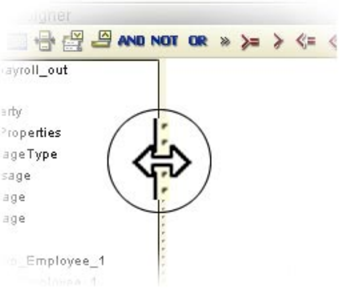
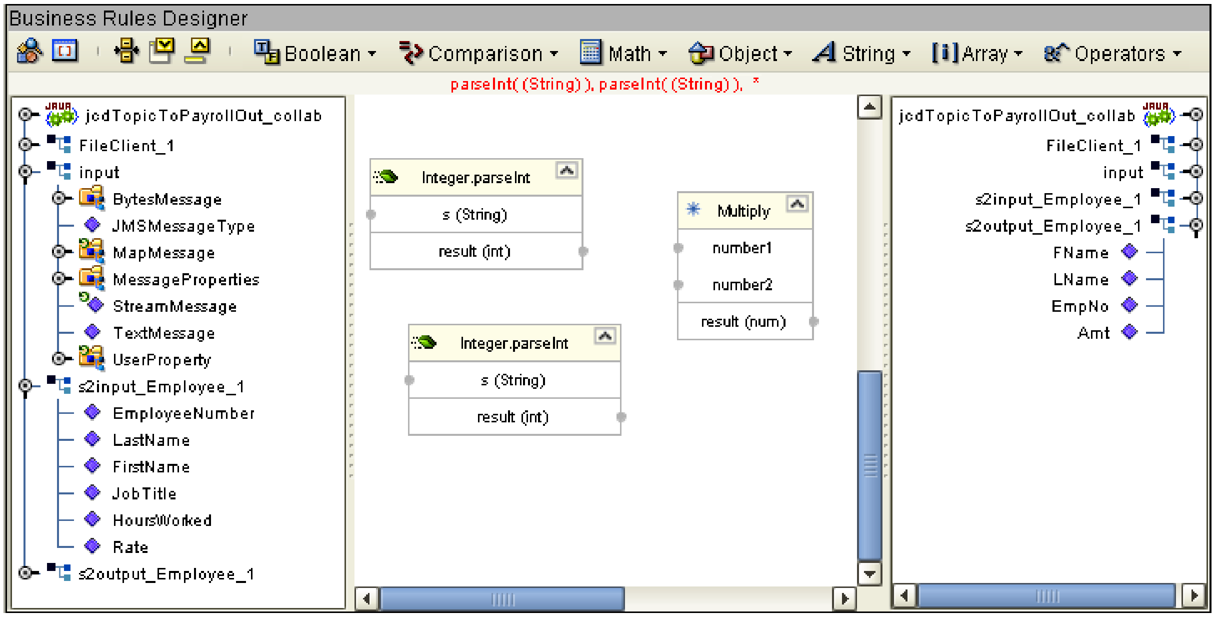


Figure 91 Prepare for Multiplication



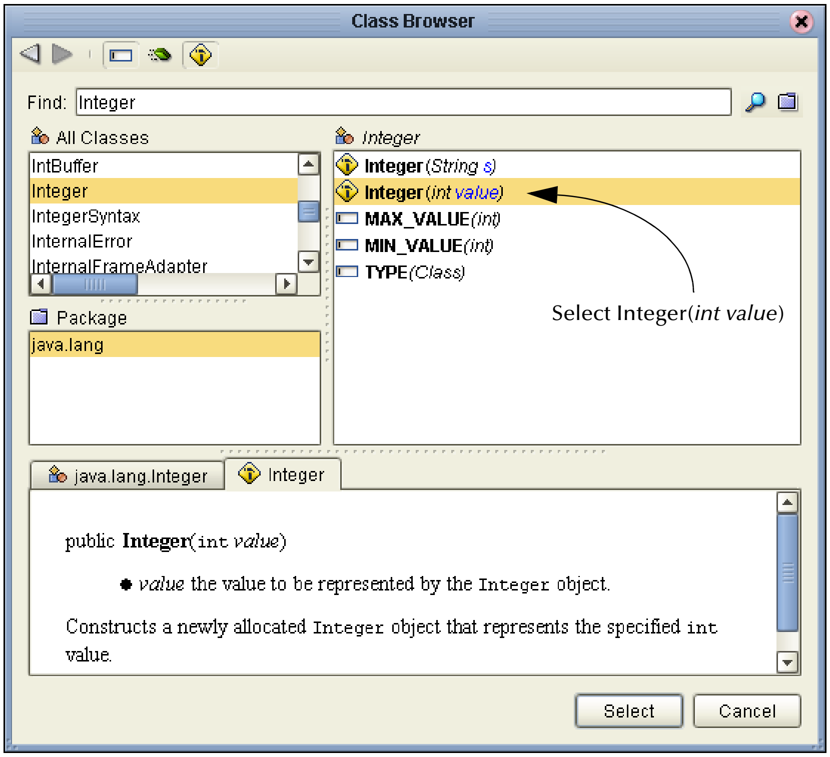
7 Click the **Class Browser** icon again .

8 Type **Integer** (if not already displayed), then click the **Show Constructors** icon on the top of the menu bar of the **Class Browser** window.

9 Select **Integer(int value)** in the list.

See the following Figure 92.

Figure 92 Class Browser Constructor



Select Integer(*int value*)

public **Integer**(int *value*)

- *value* the value to be represented by the `Integer` object.

Constructs a newly allocated `Integer` object that represents the specified `int` value.

10 Click **Select**.

This method is necessary to convert the int primitive to an Integer Object.

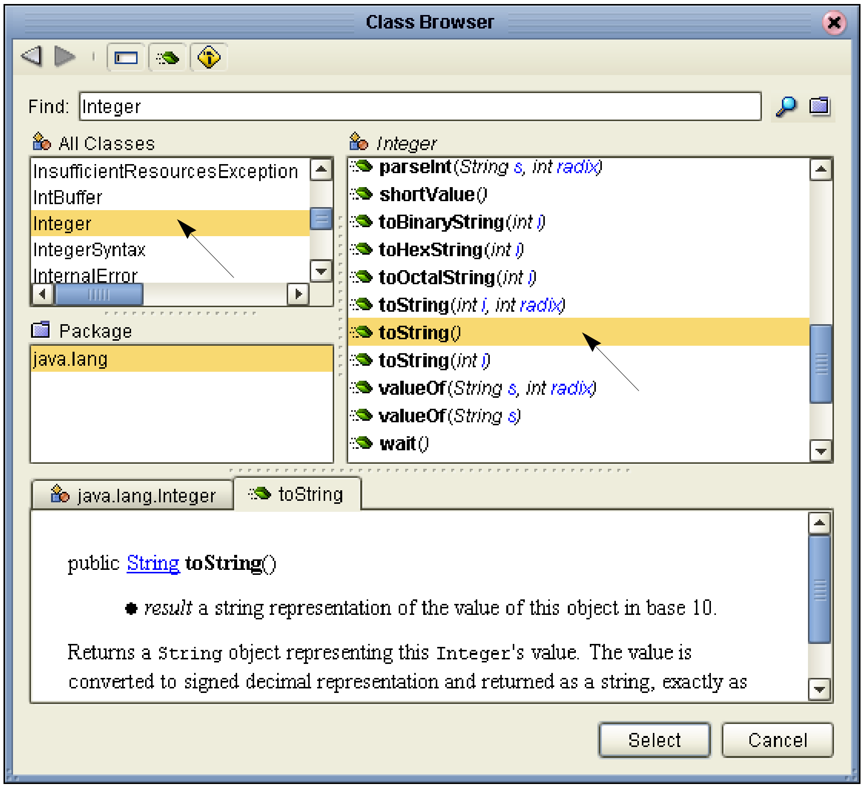
11 Click the **Class Browser** icon again.

12 Click the **Show Methods** icon on the top menu bar of the **Class Browser** window.

13 Type **Integer** (if not already displayed).

14 Click **toString()** in the list of methods.

Figure 93 Class Browser - toString



Find: Integer

InsufficientResourcesException
IntBuffer
Integer
IntegerSyntax
InternalError

java.lang

Integer

- parseInt(*String s, int radix*)
- shortValue()
- toBinaryString(*int i*)
- toHexString(*int i*)
- toOctalString(*int i*)
- toString(*int i, int radix*)
- toString()
- toString(*int i*)
- valueOf(*String s, int radix*)
- valueOf(*String s*)
- wait()

java.lang.Integer | toString

public String **toString**()

- *result* a string representation of the value of this object in base 10.

Returns a `String` object representing this `Integer`'s value. The value is converted to signed decimal representation and returned as a string, exactly as

15 Click **Select**.

This method converts the Integer Object to a String.

Connect Nodes

Connect the nodes as shown in Figure 94 to remove the temporary Error condition (Java statement) shown in red.

1 Connect **HoursWorked** in the **s2input_Employee_1** OTD to **s(string)** in the first **Integer.parseInt** method box.

2 Connect **Rate** in the **s2input_Employee_1** OTD to **s(string)** in the second **Integer.parseInt** method box.

The numbers are now set for conversion to int and ready for multiplication as integers.

3 Connect **HoursWorked** from result to **number1** in the Multiplication method box.

4 Connect **Rate** from result to **number2** in the Multiplication method box.

Number3 in the Multiplication box is not used.

5 Connect result from the multiplication to **value(int)** in the **java.lang.Integer**.

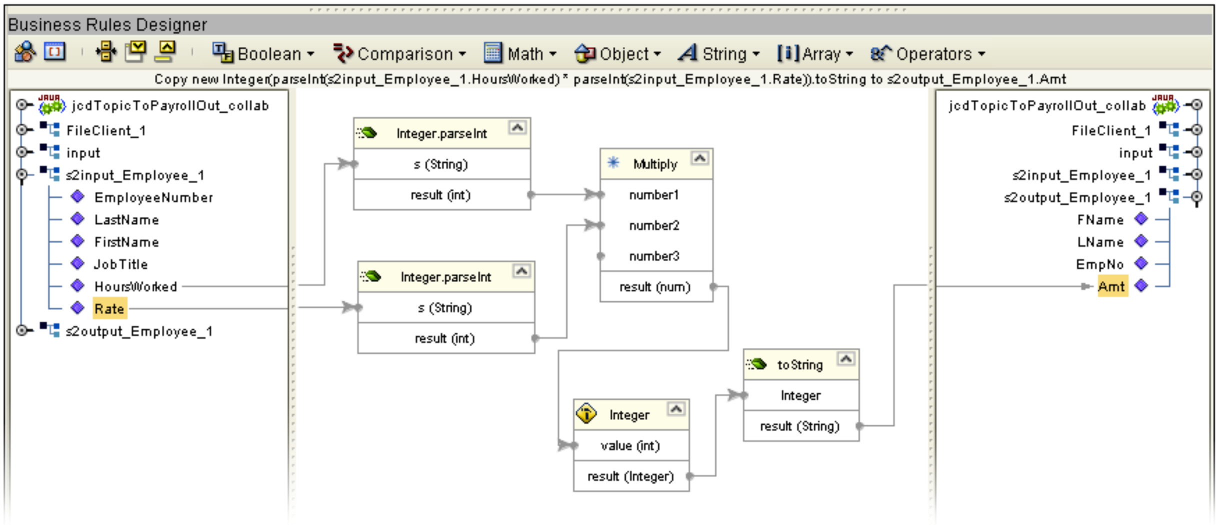
6 Connect the **result(integer)** to **Integer** in the **toString** method box.

7 Connect the **result(String)** to **Amt** in **s2output_Employee_1** in the right pane.

8 On the **File** menu, click **Save All**.

This completes the setup for the multiplication. See the following figure.

Figure 94 Multiplication Operation

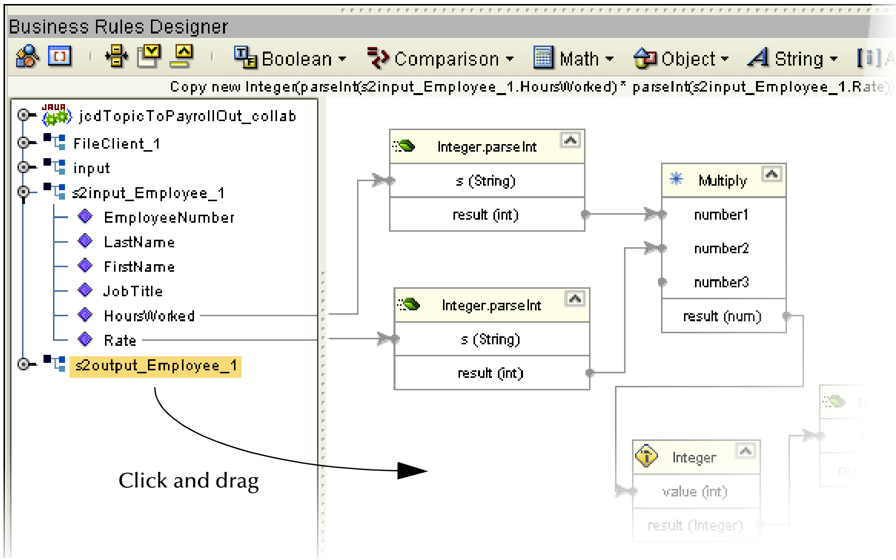


Marshal Text

You began the data manipulation process by unmarshalling data from string. You will now marshal the data from output to string.

1. Expand the view of **FileClient_1** in the right pane.
2. Click and drag **s2output_Employee_1** from the left to the center pane of the Business Rules Designer.

Figure 95 Select a Method



A list of methods appears.

Note: *You can also right click **s2output_Employee_1** and click **Select method to call**.*

3 Double-click the method **marshalToString()** from the pop-up list.

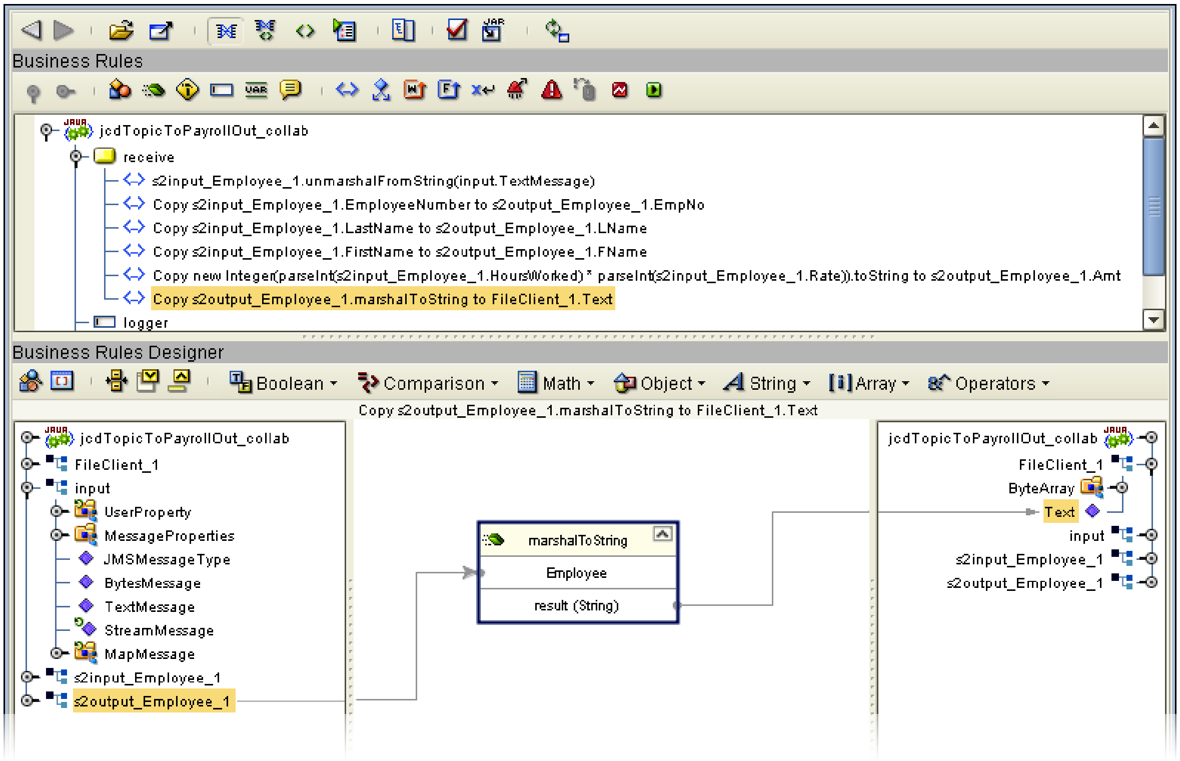
The **marshalToString** method box appears.

Note: *Your designer window may appear cluttered, and the red text implies a temporary error condition but this will clear up when you connect the result string to text in FileClient_1.*

4 Connect a node from the dialog box, **marshalToString**, **result (String)** to **FileClient_1**, **Text**.

See the following figure.

Figure 96 Marshal Text



Business Rules

```
jcdTopicToPayrollOut_collab
  receive
    s2input_Employee_1.unmarshalFromString(input.TextMessage)
    Copy s2input_Employee_1.EmployeeNumber to s2output_Employee_1.EmpNo
    Copy s2input_Employee_1.LastName to s2output_Employee_1.LName
    Copy s2input_Employee_1.FirstName to s2output_Employee_1.FName
    Copy new Integer(parseInt(s2input_Employee_1.HoursWorked) * parseInt(s2input_Employee_1.Rate)).toString to s2output_Employee_1.Amt
    Copy s2output_Employee_1.marshalToString to FileClient_1.Text
  logger
```

Business Rules Designer

Copy s2output_Employee_1.marshalToString to FileClient_1.Text

Write FileClient_1

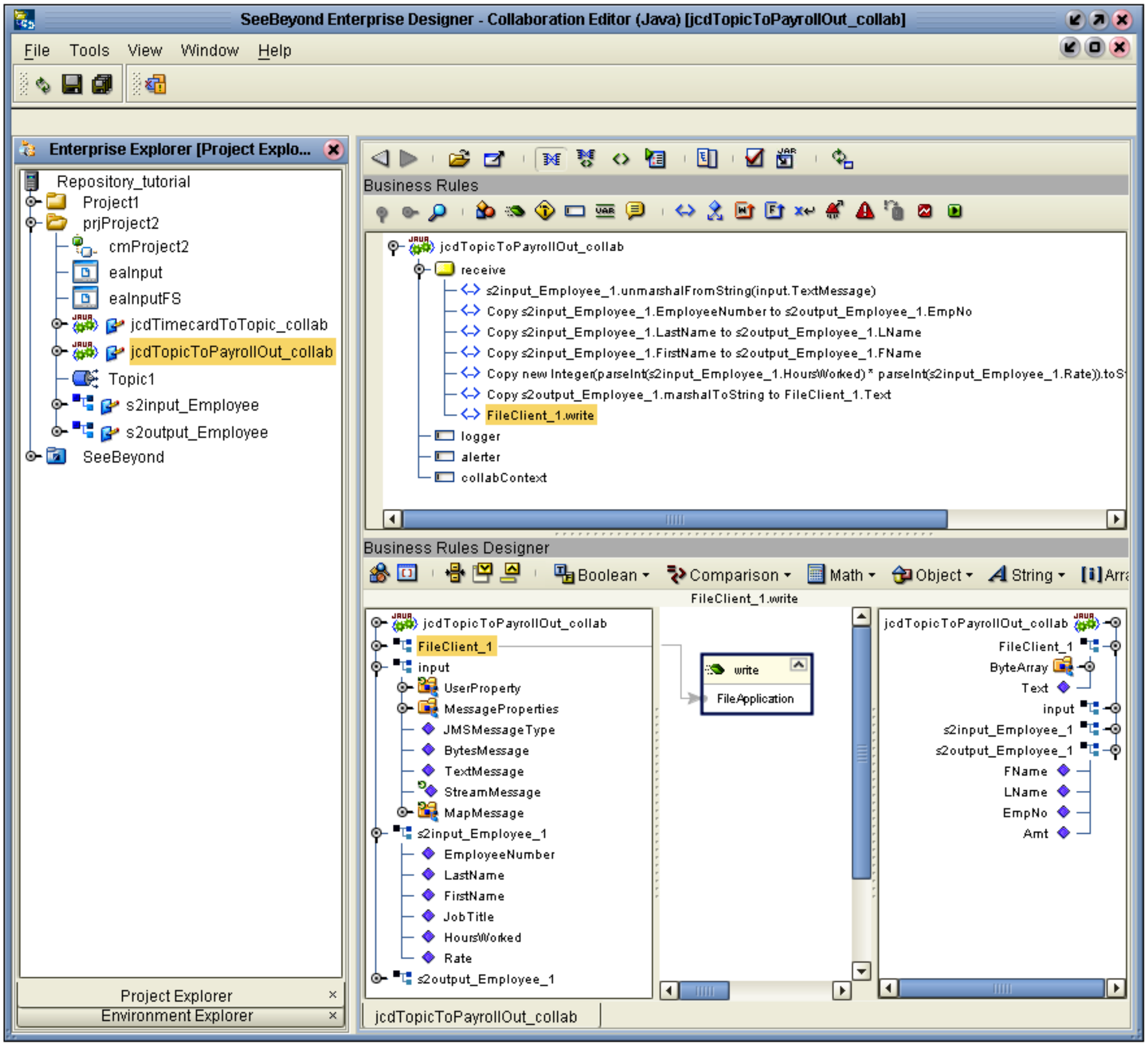
As a final step use the OTD **FileClient_1** to write your output file.

1 Click and drag **FileClient_1** from the left to the center pane of the Business Rules Designer.

2 Double-click **write ()**.

See the following figure.

Figure 97 Write File Application



```
jcdTopicToPayrollOut_collab
  receive
    s2input_Employee_1.unmarshalFromString(input.TextMessage)
    Copy s2input_Employee_1.EmployeeNumber to s2output_Employee_1.EmpNo
    Copy s2input_Employee_1.LastName to s2output_Employee_1.LName
    Copy s2input_Employee_1.FirstName to s2output_Employee_1.FName
    Copy new Integer(parseInt(s2input_Employee_1.HoursWorked) * parseInt(s2input_Employee_1.Rate)).toS
    Copy s2output_Employee_1.marshalToString to FileClient_1.Text
    FileClient_1.write
  logger
  alerter
  collabContext
```

```
jcdTopicToPayrollOut_collab
  FileClient_1
  input
    UserProperty
    MessageProperties
    JMSMessageType
    BytesMessage
    TextMessage
    StreamMessage
    MapMessage
  s2input_Employee_1
    EmployeeNumber
    LastName
    FirstName
    JobTitle
    HoursWorked
    Rate
  s2output_Employee_1
```

```
jcdTopicToPayrollOut_collab
  FileClient_1
    ByteArray
    Text
  input
  s2input_Employee_1
  s2output_Employee_1
    FName
    LName
    EmpNo
    Amt
```

A write method box appears with a node pointing to FileApplication. This instructs the system to write to the output file.

3 On the **File** menu, click **Save All**.

This completes **jcdTopicToPayrollOut_collab** using the Business Rules Designer window.

4 Navigate back to the Connectivity Map if it is not already displayed. You can close the current window or you can double-click the Connectivity Map name, **cmProject2**.

4.12 Apply Collaborations and Link Components

You have configured your Collaboration Definitions (Business Rules) in the previous sections. You are now ready to bind your Collaboration Definitions with Services using the drag-and-drop method.

4.12.1 Link Components in the Connectivity Map

Linking the components creates the logical flow of data throughout the Project. Additionally, creating a link between an External Application and a Service adds an eWay to the Connectivity Map.

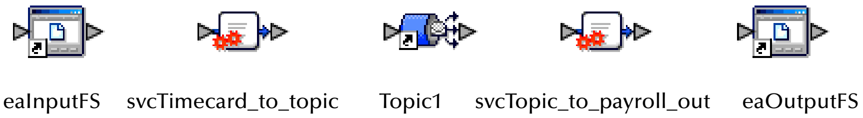
At this point, the Connectivity Map contains five unlinked objects. You are now ready to link them.

Drag-and-drop jcdTimecardToTopic_collab

Refer to 3 for the following steps.

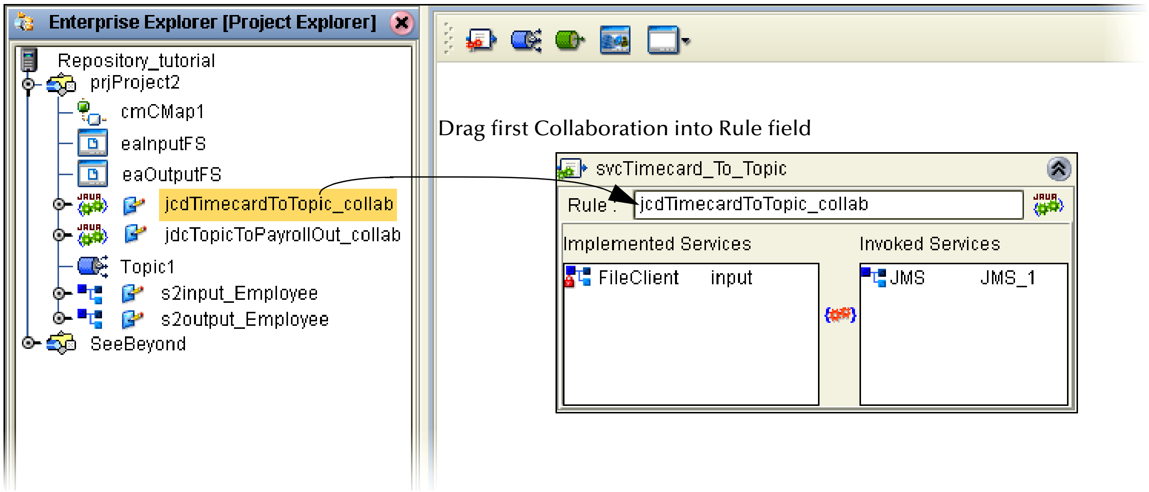
1 Double-click the Connectivity Map icon (if the Connectivity Map is not already displayed).

Figure 98 Unlinked Objects in Connectivity Map



2 Double-click the Service **svcTimecard_to_topic** in the Connectivity Map to open the Service Window.

3 Drag the Collaboration **jcdTimecardToTopic_collab** from the Project Explorer pane to the **Rule** field in the open window as shown in the following figure. Using the Drag and Drop Method



Drag first Collaboration into Rule field

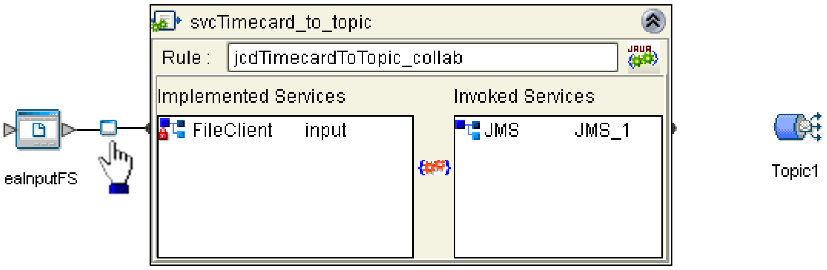
Rule : jcdTimecardToTopic_collab

Note: *Click once in the Project Explorer pane to activate the window, then click the Collaboration icon; when a+ sigh appears next to your cursor you can drag the icon. Notice that after you drag the Collaboration to the Rule field, the gear icons change from red to green to signify that the Collaboration is complete.*

Note: *You may also drag **jcdTimecardToTopic_collab** into the icon without opening the Collaboration window.*

4 Connect the **eaInputFS** to **FileClient**.

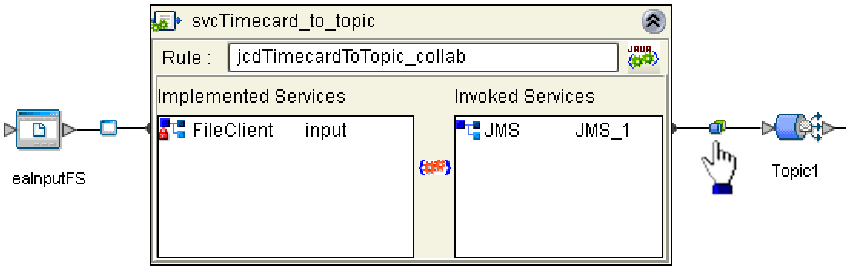
Figure 99 Connect Input to Client



Note: *You may connect nodes forward or backward. As long as the connection is made, it makes no difference; however, it may be easier to connect from the Service dialog box backward to the module icon, especially when there are multiple Services.*

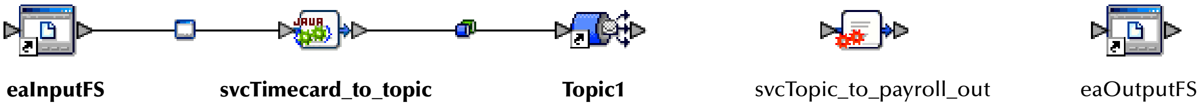
5 Connect **JMS_1** to **Topic1**. (Reposition binding box as necessary.)

Figure 100 Connect JMS to Topic



6 Click the **close** button ⏫ to close the binding box.

Figure 101 svcTimecard_to_topic Linked

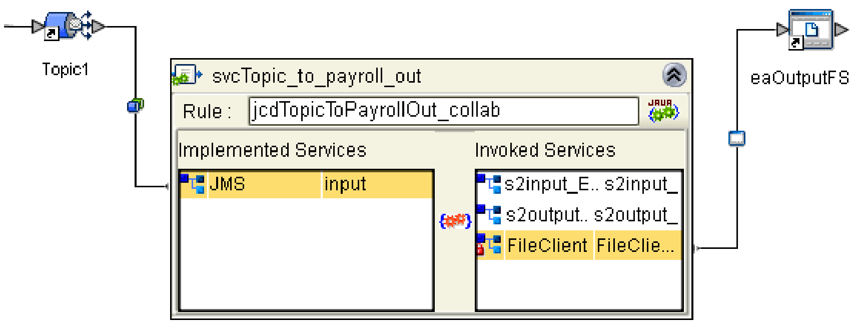


Drag-and-drop jcdTopicToPayrollOut_collab

1. Drag the Collaboration **jcdTopicToPayrollOut_collab** from the Project Explorer pane into the Service **svcTopic_to_payroll_out**. (It is not necessary to open the binding box first.)
2. Double-click the Service **svcTopic_to_payroll_out** to open the binding box.
3. Connect **Topic1** to **JMS**.
4. Connect **FileClient_1** to **eaOutputFS**.

   See the following figure.

Figure 102 Connect Topic to Output



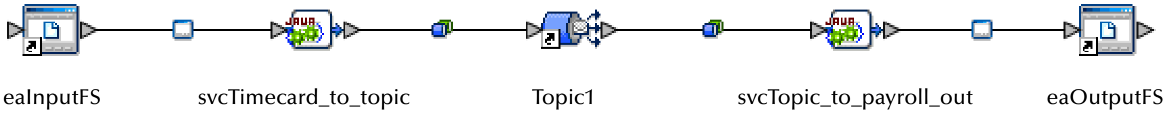
Rule : jcdTopicToPayrollOut_collab

Implemented Services	
JMS	input

Invoked Services	
s2input_E..	s2input_
s2output..	s2output_
FileClient	FileClie...

5. Click the **close** button to close the binding box.
6. Click **Save All**.

Figure 103 Linked Connectivity Map



4.13 Configure eWays and JMS Connections

Your Connectivity Map contains two eWays and JMS connections. These components must be configured for deployment. Each of these components performs a different function:

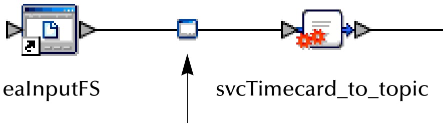
- **Inbound eWay**: The inbound eWay polls a directory on the local file system, for example: **c:\eGateData\Project2**. In this scenario the **eWay** reads the XML file (***.xml**).
- **svcTimecard_to_topic**: This Java Collaboration Service copies the input XML data without making any changes to it.

- **svcTopic_to_payroll_out**: This Java Collaboration Service performs the calculation and data mapping function.
- **Outbound eWay**: The outbound eWay writes the results to an output file (.dat or .txt) on the local file system, **c:\eGateData\Project2**.

Configure Inbound eWay

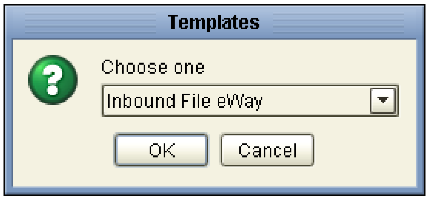
Choose inbound, and identify the inbound file and location.

Figure 104 Inbound eWay



1 Double-click the first eWay –□– . Select **Inbound File eWay** from the drop-down list, then click **OK**.

Figure 105 External Type Dialog Box



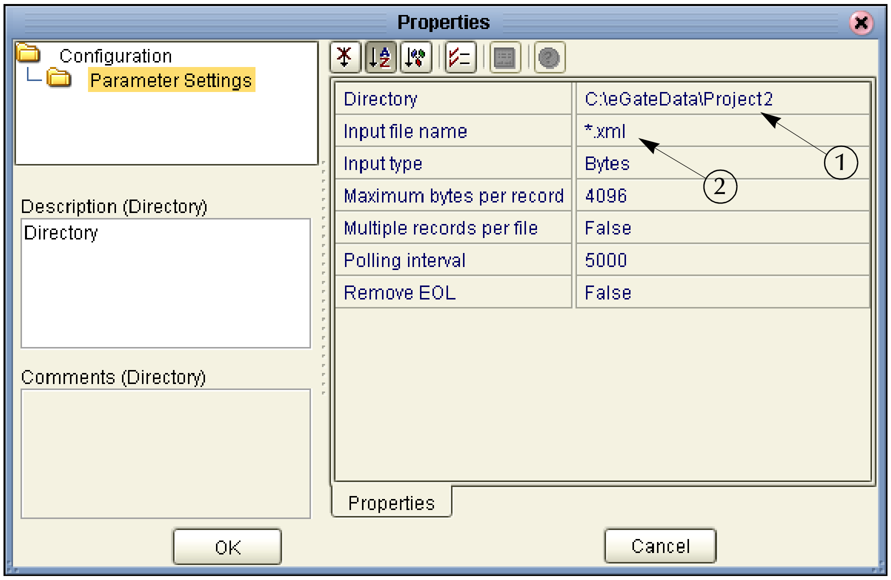
Templates

Choose one

Inbound File eWay

This creates a new inbound eWay and the Properties dialog box appears.

Figure 106 Properties Configuration (Inbound eWay)



Properties

Configuration
Parameter Settings

Description (Directory)
Directory

Comments (Directory)

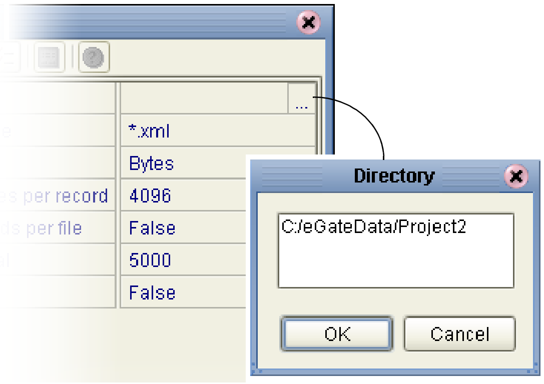
Directory	C:\eGateData\Project2
Input file name	*.xml
Input type	Bytes
Maximum bytes per record	4096
Multiple records per file	False
Polling interval	5000
Remove EOL	False

Refer to the arrows in the previous figure. These are the editable fields that you configure.

2 Click once on the current path to select and type a correct directory path. In this Tutorial the path is **c:\eGateData\Project2** (number 1 in Figure 106 above).

Alternately, you can click the ellipses (...) and enter the data path. See the following figure.

Figure 107 Directory Data Path



C:/eGateData/Project2

Note: *You can use a forward slash (/) or a backward slash (\) for your directory path, but you must be consistent.*

3 Click once on the current **Input File Name** field and enter ***.xml**. The eWay reads any files with the **.xml** extension (number 2 in Figure 106).

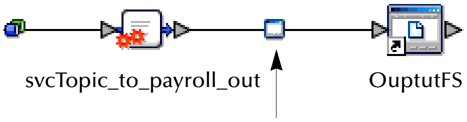
4 Click **OK** to close the Properties Dialog Box for the inbound eWay.

Note: *If the* ***Polling interval*** *value is left unchanged, the eWay polls the directory location every five seconds (5000 milliseconds).*

Configure Outbound eWay

Identify the outbound file and location.

Figure 108 Outbound eWay



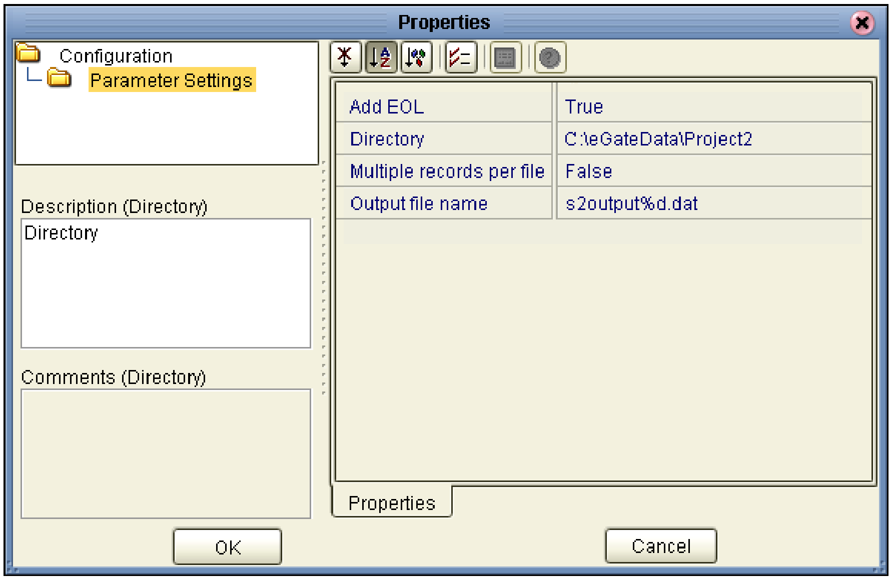
1 Double-click the second eWay. Select **Outbound File eWay**. Click **OK**.

The Properties dialog box appears.

Use the same path you used for the inbound eWay (**c:\eGateData\Project2**). Name the output file name: **s2output%d.dat**.

2 Click **OK** to close the Properties Dialog Box for the outbound eWay.

Figure 109 Properties Configuration (Outbound eWay)



Description (Directory)
Directory

Comments (Directory)

Add EOL	True
Directory	C:\eGateData\Project2
Multiple records per file	False
Output file name	s2output%d.dat

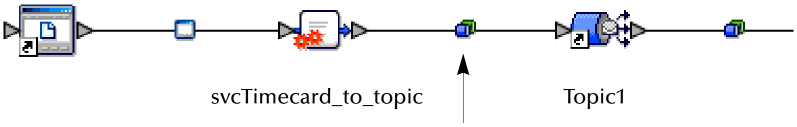
*Note: The **Multiple records per file** setting determines whether records are concatenated in the output file. A True setting (default) means that multiple records are written in the same file. A False setting, as shown in Figure 109 above, means that only one record is written to one file. See* **"Multiple records per file (True or False)" on page 51** *in Scenario1.*

You have now configured both the input and output eWays.

JMS Client Configuration

The JMS Client Configuration icons appear on both sides of the **Topic1** icon.

Figure 110 JMS Client Properties Icon



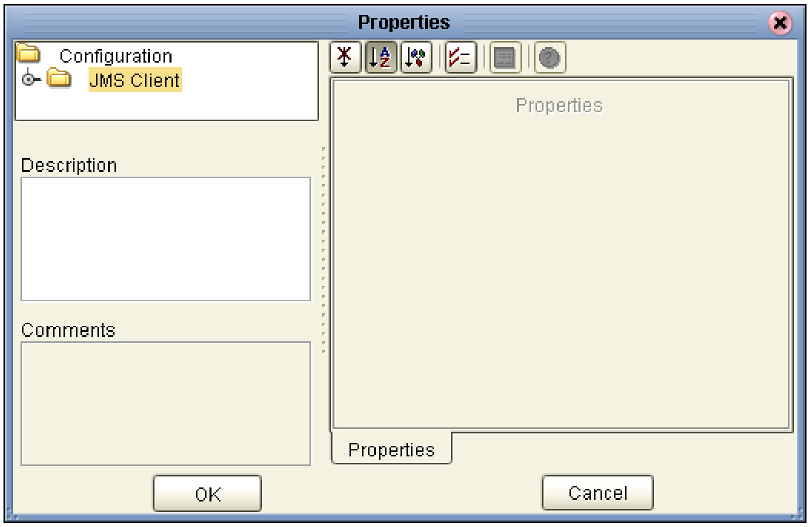
Select JMS Client Configuration

For this Tutorial accept the default configurations.

1 Double-click the first of the two JMS icons (see previous Figure 110).

The Message Server properties dialog box appears.

Figure 111 Message Server Configuration



2. Click **OK** to accept the defaults.
3. Repeat this step for the other JMS client configuration.

   You are now ready to deploy your Project components in an Environment containing a Logical Host and External Applications.
4. On the **File** menu, click **Save All**.

   Your project is now complete and ready for deployment.

4.14 Create Environment

In this section, create and activate the Deployment Profile. First, create an Environment.

Create Environment and Logical Host

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

1. Click **View** from the Menu bar, then click **Environment Explorer**.

Note: *If you have already created an Environment called Tutorial, you may use that Environment and logicalhost. Continue with* **"Create External File System for the Inbound eWay" on page 110***. Otherwise, continue with the steps in this section.*

2. Right-click the Repository name ( ). Click **New Environment**.
3. Right-click **Environment1** and rename **Tutorial**. Press **Enter**.
4. Right-click **Tutorial**. Click **New Logical Host**.

5 Accept the default name **LogicalHost1**.

This creates a **LogicalHost1** window in the right pane.

Create Integration Server

An integration server is a J2EE software platform that houses the business logic container used to run Collaborations and JCA connectors (eWays).

1 Right-click **LogicalHost1** in the **Enterprise Explorer** window. Click **New SeeBeyond Integration Server**.

IntegrationSvr1 appears in the **LogicalHost1** window.

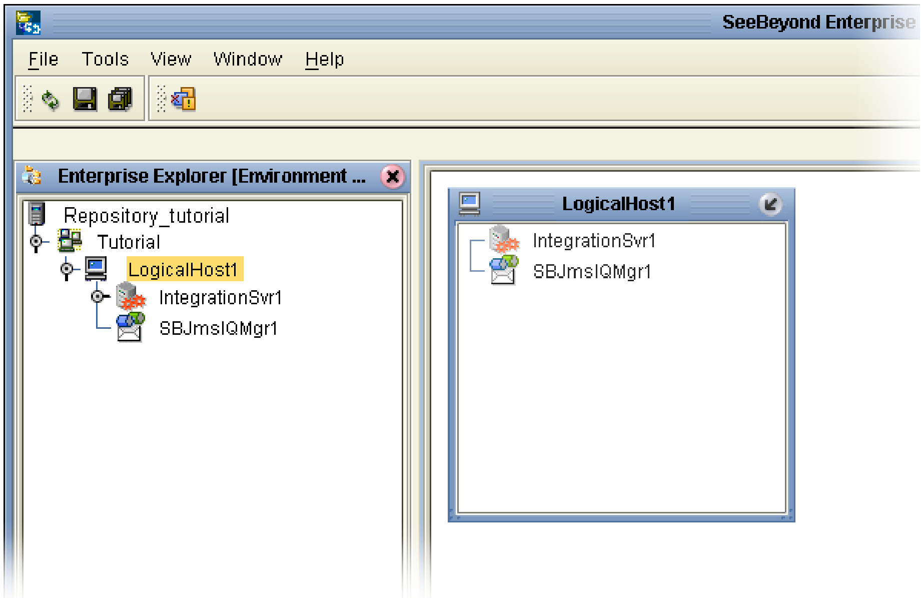
Create JMS IQ Manager

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

1 Right-click **LogicalHost1** in the **Enterprise Explorer** window. Click **New SeeBeyond JMS IQ Manager**.

SBJmsIQMgr1 appears in the **LogicalHost1** window.

Figure 112 Create Logical Host Servers



These servers automatically appear in the **LogicalHost1** dialog box.

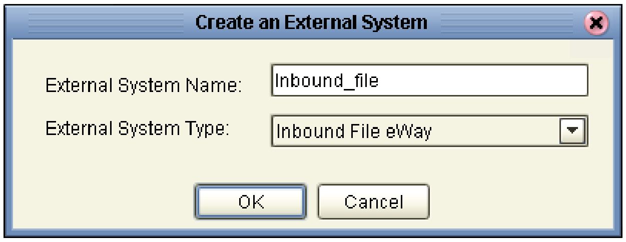
Create External File System for the Inbound eWay

Create the External File System if it does not already exist from Scenario1.

Note: *If you already have external files from Scenario1 in your Environment pane, you can use those same files* ***Inbound_file*** *and* ***Outbound_file****; in that case continue with* **"Create and Activate Deployment Profile" on page 111**.

1 Right-click **Tutorial**. Click **New File External System**.

Figure 113 External File Inbound


Create an External System

External System Name: Inbound_file

External System Type: Inbound File eWay

2 Type **Inbound_file** in the name field (if file doesn't already exist).

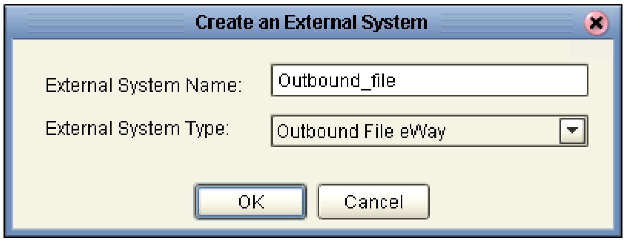
3 Select **Inbound File eWay**. Click **OK**.

This places the inbound external system, **Inbound_file** in the Environment Editor pane.

Create External File System for the Outbound eWay

1 Right-click **Tutorial**. Click **New File External System**.

Figure 114 External File Outbound


Create an External System

External System Name: Outbound_file

External System Type: Outbound File eWay

2 Type **Outbound_file** in the name field (if file doesn't already exist).

3 Select **Outbound File eWay**. Click **OK**.

This places the outbound external system, **Outbound_file** in the Environment Editor pane.

Note: *If the "Create an External System" (or any Java pop-up) window doesn't appear, it may be hidden behind your Enterprise Explorer window. Press the ALT+TAB keys to see if you have any hidden pop-ups. Use this keystroke combination if your Enterprise Explorer GUI appears to be "frozen."*

4.14.1 Create and Activate Deployment Profile

In the following steps, create a Deployment Profile, then deploy (map) the components. If you already created Deployment1 for Scenario1, you need a different Deployment profile for Scenario2, e.g. Deployment2.

Deployment Profile

A Deployment Profile contains information about how Project components are deployed in an Environment, and it also maps components to the Environment.

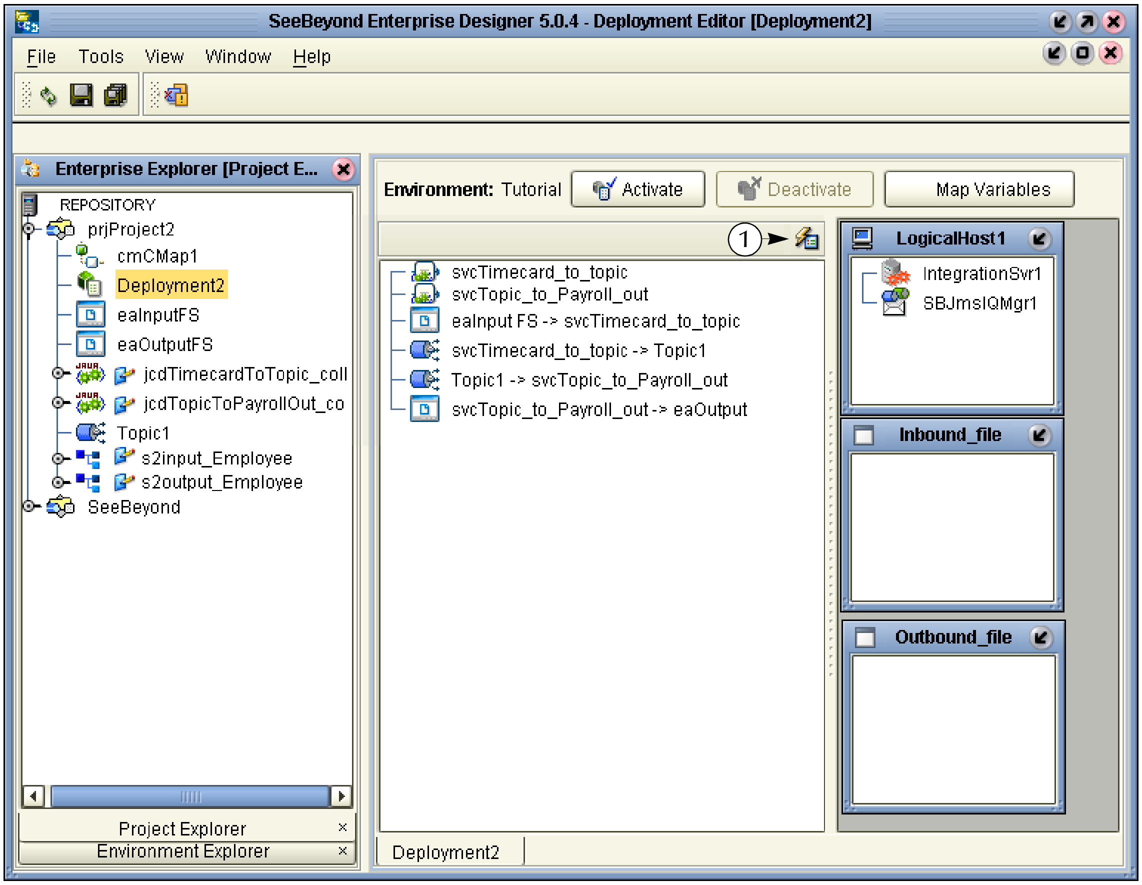
1 Click the Project Explorer tab to return to the Project Explorer pane of your Project.

2 Right-click **prjProject2**, then click **New** and **Deployment Profile**.

3 Use the name **Deployment2** (and Tutorial).

4 Click **OK**.

Figure 115 Deploy Components



After creating your deployment profile, you can auto map components to your LogicalHost1.

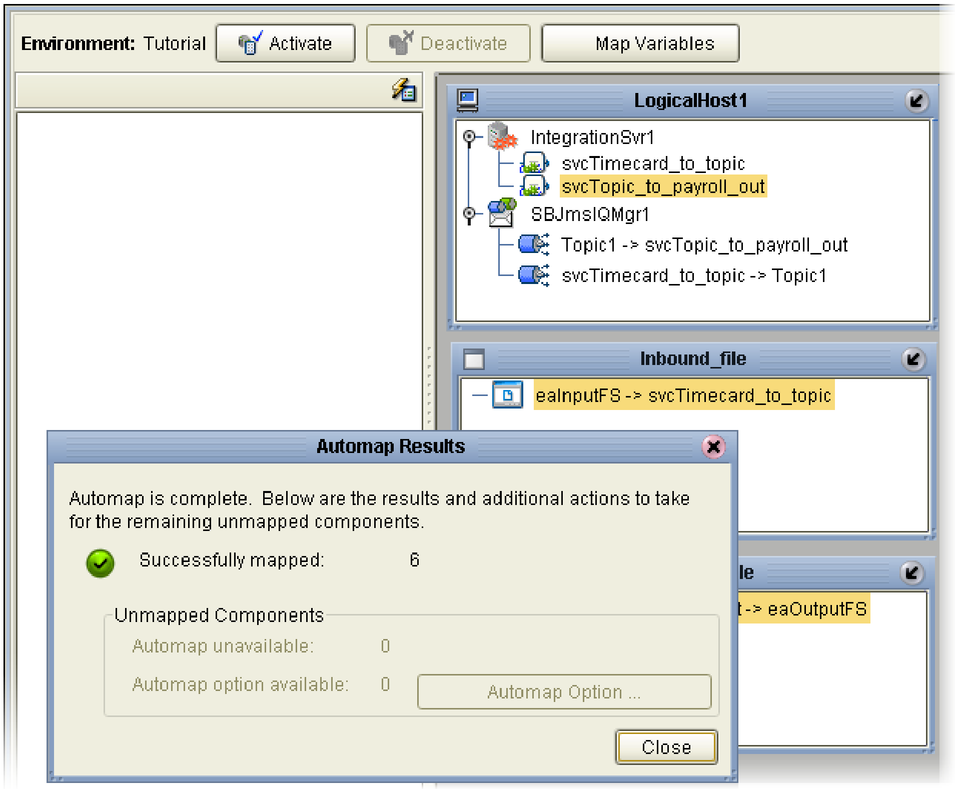
Auto Map and Deployment Components

1 Click the **Auto Map** icon (number 1 in Figure 115).

This automatically maps components to the logical host and the file external systems. You could have also used the drag-and-drop method to map your components.

See the following figure.

Figure 116 Auto Map LogicalHost1



Automap is complete. Below are the results and additional actions to take for the remaining unmapped components.

Successfully mapped: 6

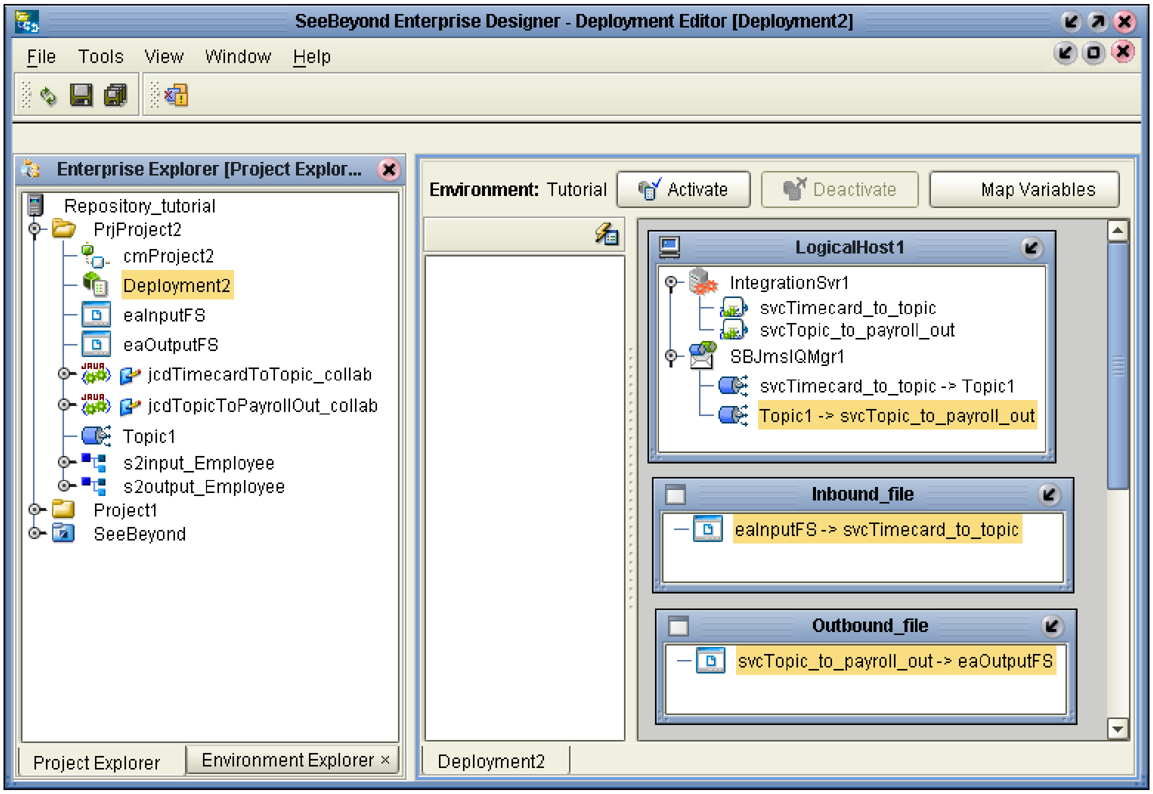
Automap unavailable: 0

Automap option available: 0

After populating your Environment with all the available components, there are no components listed in the middle pane. All your components reside in the servers, as shown in the following figure.

2 Click the **Close** button.

Figure 117 Populated Environment



4.15 Run Bootstrap and Management Agent

Refer to **Chapter 3**, **Run the Bootstrap and Management Agent** on page 57 in Scenario1.

4.16 Run prjProject2

Note: *Refer to* **Chapter 3**, **Run Project1** *on page 60 in Scenario1, except this will be for prjProject2. If you have already run Project1 in the first scenario, you may want to deactivate the deployment profile in Project1 before running prjProject2.*

Note: *You can use the same Environment (Tutorial) and same logicalhost that you used in Scenario1 for Scenario2, but the deployment profile name must be different, Deployment2, for example. The external files used are of course different. (The inbound and outbound names in the Environment editor can be the same.)*

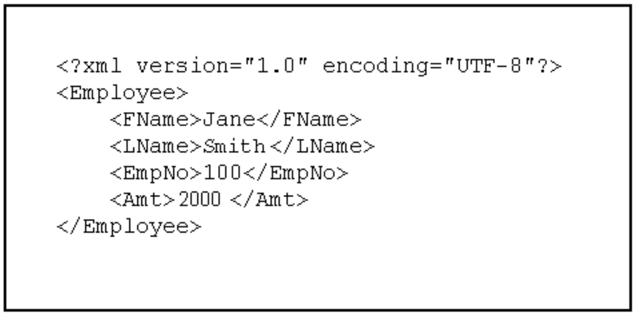
4.17 Verify Output Data

In **"Sample Data" on page 69**, you downloaded a sample input XML file and later set up the DTDs. The Management Agent processes your input file, **Timecard.xml**, and writes the results to **s2output1.dat**.

View Output

1. Use Windows Explorer to navigate to **c:\eGateData\Project2**.
2. See that the Input eWay, after processing the input file, renamed the input file to **Timecard.~in**.
3. See that the Output eWay generated the output file, **s2output1.dat**.
4. Use Windows Wordpad or a similar text edit application to view the contents of the output file.

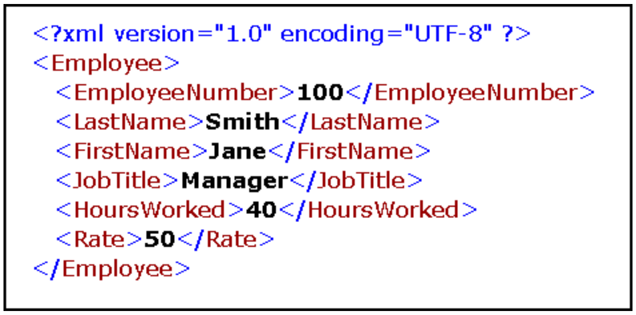
Figure 118 Output File



```
<?xml version="1.0" encoding="UTF-8"?>
<Employee>
    <FName>Jane</FName>
    <LName>Smith</LName>
    <EmpNo>100</EmpNo>
    <Amt>2000 </Amt>
</Employee>
```

In this sample, the amount is 2000 because hours worked was 40 and rate of pay was 50. See input file, Figure 119.

Figure 119 Input File



```
<?xml version="1.0" encoding="UTF-8" ?>
<Employee>
  <EmployeeNumber>100</EmployeeNumber>
  <LastName>Smith</LastName>
  <FirstName>Jane</FirstName>
  <JobTitle>Manager</JobTitle>
  <HoursWorked>40</HoursWorked>
  <Rate>50</Rate>
</Employee>
```

You can change the data in your input file and process the input file again, rename **Timecard.~in** to **Timecard.xml**. The File eWay picks up the input file, **Timecard.xml**, and writes the results to **s2output1.dat** or **s2output%d.dat** (where "d" is incremented with each write), depending on your Multiple Records per File setting (see Figure 109).

Chapter 5

Enterprise Manager Overview

For more detailed information about the Enterprise Manager see the *eGate Integrator System Administration Guide*. This chapter provides a quick overview only.

The Enterprise Manager is a powerful monitoring tool that is useful for troubleshooting problems and viewing system activity at runtime. The Enterprise Manager (although not required for the tutorials) provides alerts, error handling, logs, and graphic views, showing active and inactive Services.

What's in This Chapter

5.1 Browser-based Monitor Console

The Enterprise Manager is a browser-based console that allows you to view and monitor projects dynamically, in real time. Within Enterprise Manager, components on the Connectivity Map flash to signal a problem or warning. Components flash to signal a stopped component or object.

- Monitor, start, and stop Services
- Monitor JMS topics and queries
- View and edit messages
- Monitor system logs
- Monitor and manage system alerts
- Monitor eInsight Business Processes (if licensed)

Locating the Problem Node

When a problem occurs, a yellow question mark icon appears in the Project Explorer pane.

5.1.1 Using Project View

Project view allows you to perform runtime monitoring and management of Deployments and Connectivity Maps.

5.1.2 Using Environment View

Environment view allows you to perform runtime monitoring and management of eGate and eInsight System components.

5.2 Starting and Using Enterprise Manager

Start Enterprise Manager

Use Internet Explorer to point to the URL used by your installation to access the Enterprise Manager. Refer to the *SeeBeyond ICAN Suite Installation Guide*.

Launch the Monitor by clicking on the home page that appears after logging in to the Enterprise Manager, as explained in the steps below.

You can click the Projects tab or the Environment tab to monitor the state of the Logical Host(s) and the Project components deployed in them. You can also view the logs and change the log levels of the Logical Host.

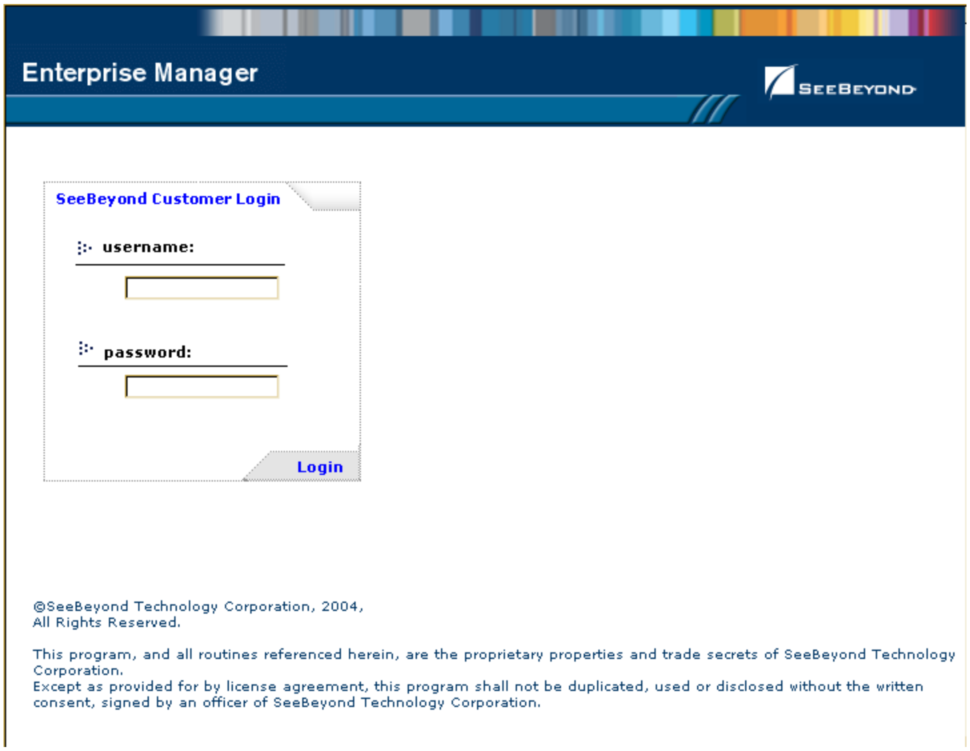
It is beyond the scope of this Tutorial to explain all the capabilities of the Enterprise Manager. The following examples demonstrate a few of its capabilities.

Monitor your Project in Real Time

With the Enterprise Manager running, you can check the runtime status of your project.

1 Start Enterprise Manager and type your username and password.

Figure 120 Enterprise Manager Login



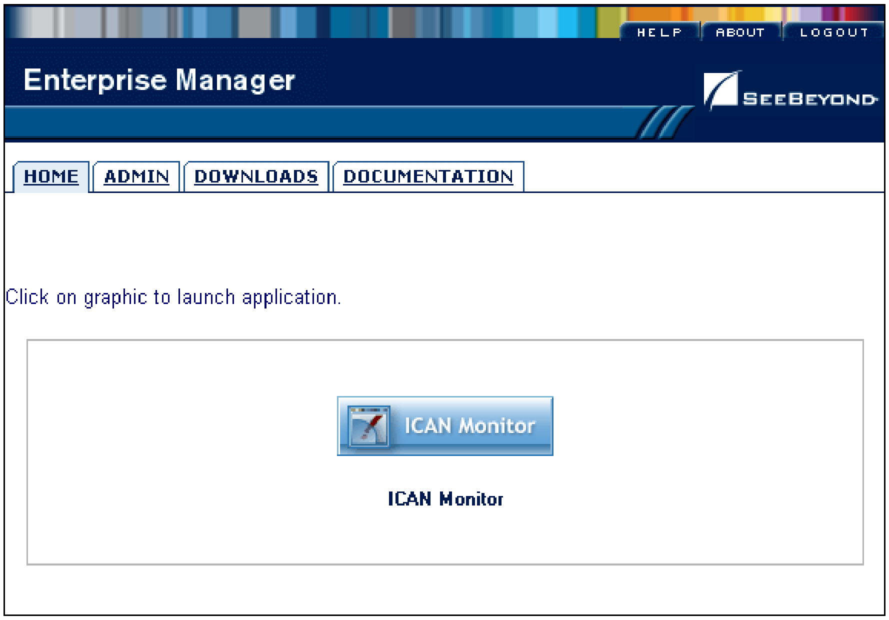
©SeeBeyond Technology Corporation, 2004,
All Rights Reserved.

This program, and all routines referenced herein, are the proprietary properties and trade secrets of SeeBeyond Technology Corporation.
Except as provided for by license agreement, this program shall not be duplicated, used or disclosed without the written consent, signed by an officer of SeeBeyond Technology Corporation.

The Enterprise Manager home page appears.

2 Click the **ICAN Monitor** button.

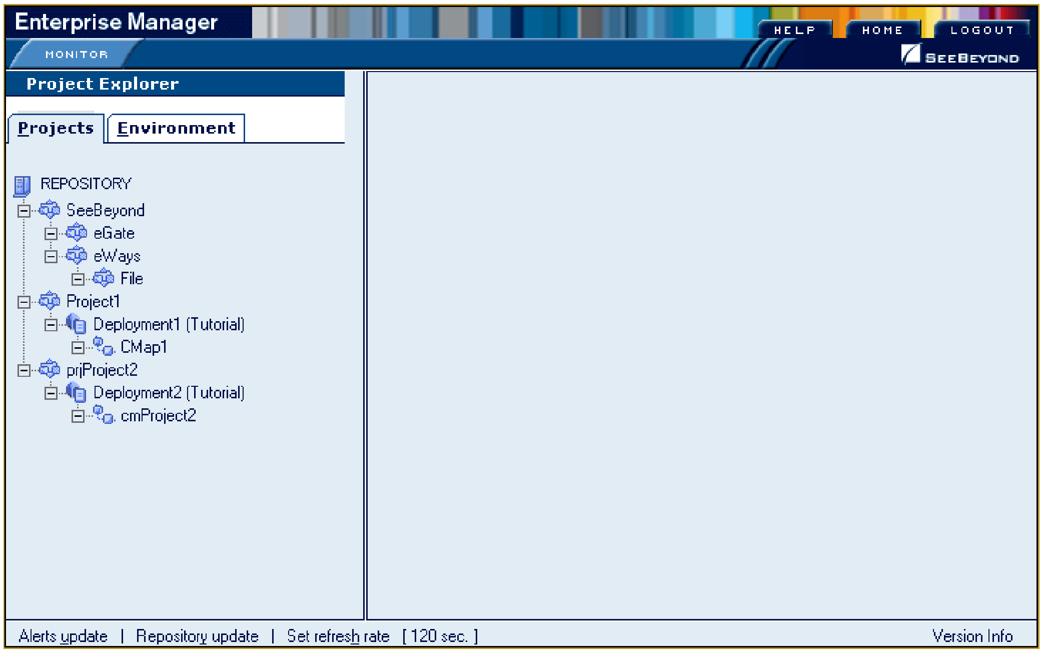
Figure 121 Enterprise Manager Home Page



Click on graphic to launch application.

The Enterprise Manager Project Explorer appears.

Figure 122 Enterprise Manager Project Explorer



3 Double-click **Repository_tutorial** (Repository name) to expand all Project views.

4 Click the **Connectivity Map**.

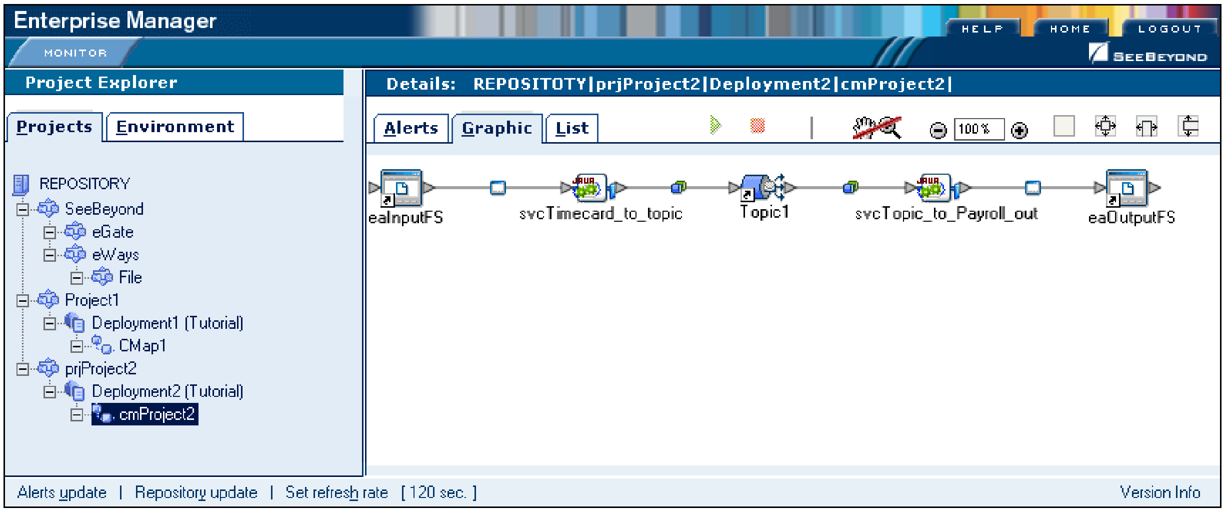
Click on one of the following tabs to view messages or status:

- **Alerts** shows the alert messages, including their types and severity
- **Graphic** shows the Connectivity Map
- **List** shows deployment status

5 Click the **Graphic** tab.

Refer to the following Figure 123.

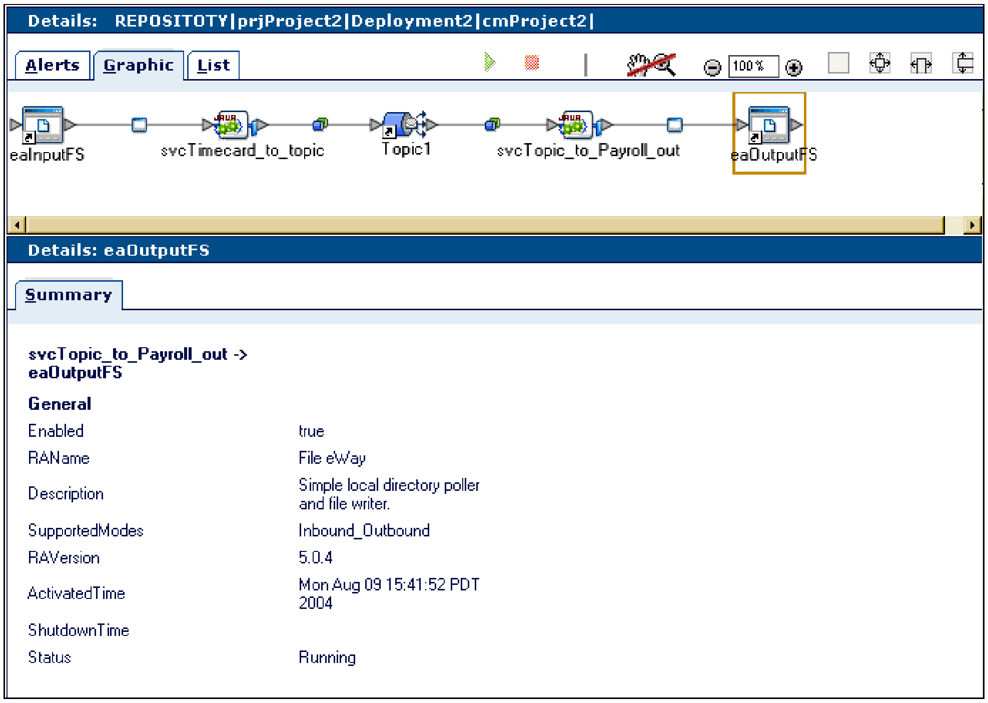
Figure 123 Enterprise Manager Connectivity Map



6 Click on **eaOutputFS** to see a runtime summary for that object.

Refer to the following figure.

Figure 124 Runtime Summary



svcTopic_to_Payroll_out -> eaOutputFS

General

Enabled	true
RAName	File eWay
Description	Simple local directory poller and file writer.
SupportedModes	Inbound_Outbound
RAVersion	5.0.4
ActivatedTime	Mon Aug 09 15:41:52 PDT 2004
ShutdownTime	
Status	Running

5.2.1 Stop Service

Using the example above, illustrated in Figure 123, you can also start and stop services. For example, if you want to stop **svcTopic_to_payroll_out** Service (stop processing the messages and hold them in a Topic) you would perform the following steps:

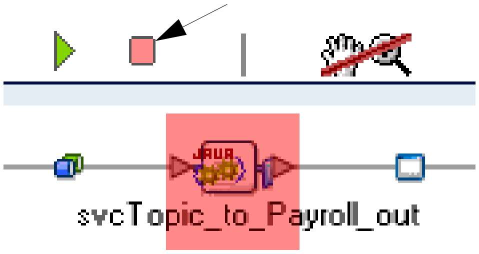
Stop Service

1 Click the Service, **svcTopic_to_Payroll_out**.

2 Click the small red Box.

See the following figure.

Figure 125 Stop Service



A red box around the Service flashes while the service is stopped.

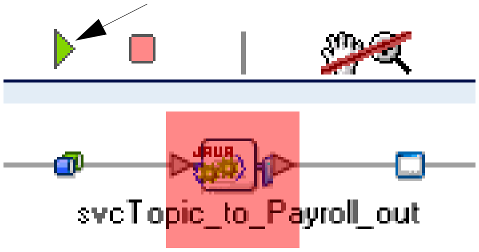
Start Service

3 Click the Service, **svcTopic_to_Payroll_out**.

4 Click the small green arrow.

See the following figure.

Figure 126 Start Service



Keep in mind that although the stopped Service does not process messages, those messages are held in a queue to be processed immediately when the Service is restarted.

5.2.2 Monitor Alerts

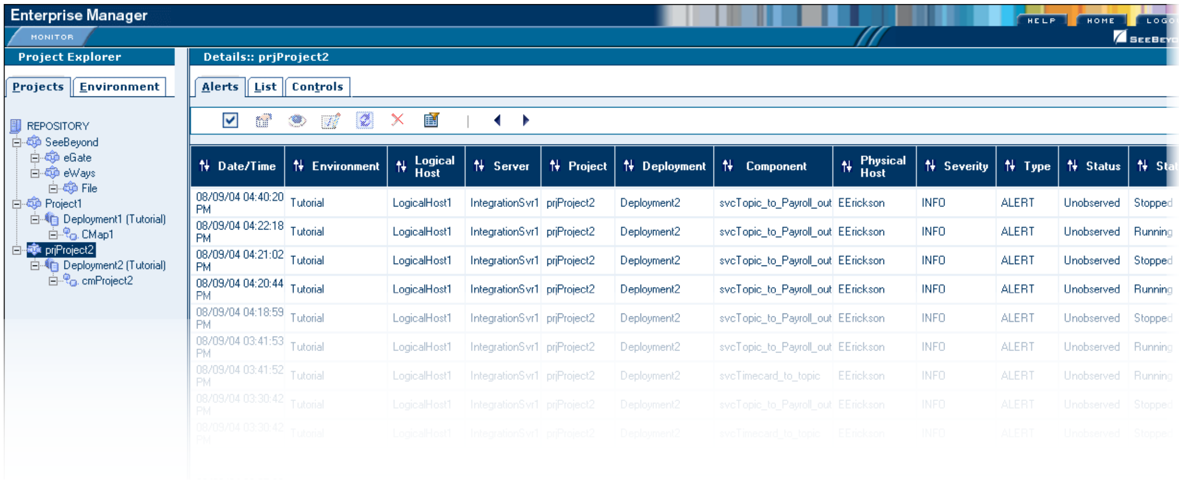
Enterprise Manager Projects Tab

With the Enterprise Manager open and **Topic_to_payroll_out** stopped (causes alert), perform the following steps:

1. Click the Projects tab.
2. Expand all the views by double-clicking the Repository name (Repository_tutorial).
3. Click the Project name.
4. Click the Alerts tab.

   Refer to the following Figure 127.

Figure 127 Enterprise Manager Alert Messages



Date/Time	Environment	Logical Host	Server	Project	Deployment	Component	Physical Host	Severity	Type	Status	Sta
08/09/04 04:40:20 PM	Tutorial	LogicalHost1	IntegrationSvr1	prjProject2	Deployment2	svcTopic_to_Payroll_out	EErickson	INFO	ALERT	Unobserved	Stopped
08/09/04 04:22:18 PM	Tutorial	LogicalHost1	IntegrationSvr1	prjProject2	Deployment2	svcTopic_to_Payroll_out	EErickson	INFO	ALERT	Unobserved	Running
08/09/04 04:21:02 PM	Tutorial	LogicalHost1	IntegrationSvr1	prjProject2	Deployment2	svcTopic_to_Payroll_out	EErickson	INFO	ALERT	Unobserved	Stopped
08/09/04 04:20:44 PM	Tutorial	LogicalHost1	IntegrationSvr1	prjProject2	Deployment2	svcTopic_to_Payroll_out	EErickson	INFO	ALERT	Unobserved	Running
08/09/04 04:18:59 PM	Tutorial	LogicalHost1	IntegrationSvr1	prjProject2	Deployment2	svcTopic_to_Payroll_out	EErickson	INFO	ALERT	Unobserved	Stopped
08/09/04 03:41:53 PM	Tutorial	LogicalHost1	IntegrationSvr1	prjProject2	Deployment2	svcTopic_to_Payroll_out	EErickson	INFO	ALERT	Unobserved	Running
08/09/04 03:41:52 PM	Tutorial	LogicalHost1	IntegrationSvr1	prjProject2	Deployment2	svcTimecard_to_topic	EErickson	INFO	ALERT	Unobserved	Running
08/09/04 03:30:42 PM	Tutorial	LogicalHost1	IntegrationSvr1	prjProject2	Deployment2	svcTopic_to_Payroll_out	EErickson	INFO	ALERT	Unobserved	Stopped
08/09/04 03:30:42 PM	Tutorial	LogicalHost1	IntegrationSvr1	prjProject2	Deployment2	svcTimecard_to_topic	EErickson	INFO	ALERT	Unobserved	Stopped

5.2.3 Check Log Messages

You can troubleshoot errors by viewing the logged messages.

In this example, the timecard input is designed for integers. The business rules are set up to handle integers only. If a data entry person wanted to correctly type a pay rate of $50.00, he or she would have to input the integer as 5000, assuming two decimal places. (The decimal places would have to be reformatted through additional processing.)

If a data entry person incorrectly entered a pay rate as 55.00, the decimal would generate an error message and the output file would not be updated.

View Logged Messages

With the Enterprise Manager open, perform the following steps to check logged messages:

1. Click **Environment**.
2. Expand all the views by double-clicking the Repository name (Repository_tutorial).
3. Click **IntegrationSvr1**.

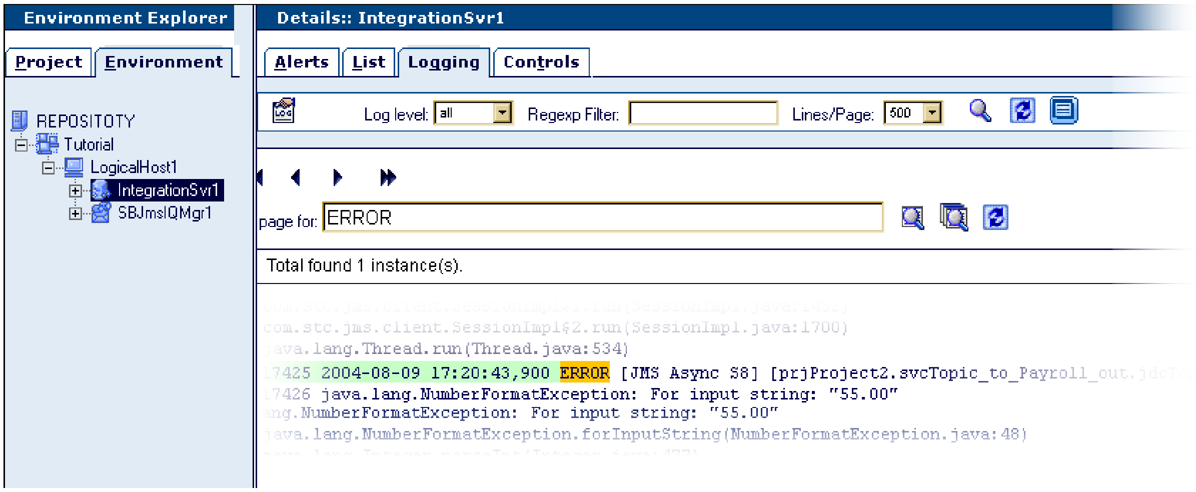
4 Click the **Logging** tab.

5 Do a search on the word **Error** (or the word Exception).

The exception, **NumberFormatException** is recorded in the log.

See the following figure.

Figure 128 Check Logged Messages



Log level: all Regexp Filter: Lines/Page: 500

page for: ERROR

Total found 1 instance(s).

```
com.stc.jms.client.SessionImpl$1.run(SessionImpl.java:1452)
com.stc.jms.client.SessionImpl$2.run(SessionImpl.java:1700)
java.lang.Thread.run(Thread.java:534)
17425 2004-08-09 17:20:43,900 ERROR [JMS Async S8] [prjProject2.svcTopic_to_Payroll_out.jdcTo
17426 java.lang.NumberFormatException: For input string: "55.00"
ang.NumberFormatException: For input string: "55.00"
java.lang.NumberFormatException.forInputString(NumberFormatException.java:48)
java.lang.Integer.parseInt(Integer.java:477)
```

Glossary

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, 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.

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.

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.

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.

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.

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.

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.

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.

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.

Subproject

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

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.

XSLT

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

e*Gate 4.x Terms in eGate 5.0

Table 1 provides definitions for the terms that are new with eGate release 5.0, as well as equivalent terms from eGate release 4.x.

Table 1 eGate 5.0 Terms

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

Table 1 eGate 5.0 Terms (Continued)

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

Index