SeeBeyond ICAN Suite

eGate Integrator Tutorial

Release 5.0.3

SEEBEYOND®

Version 20040302102224.

Contents

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 1

Introduction

Welcome to the *eGate Integrator Tutorial*. The Tutorial must be run on a Windows system.

This document includes an overview of the Enterprise Designer graphical user interface (GUI) and tutorials to help you learn how to create and run a Project in eGate Integrator.

1.1 Organization of Information

This document includes the following chapters:

- **Chapter 1 "Introduction" on page 6** 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" on page 9** introduces the Enterprise Designer Components, Menu Bar, Enterprise Explorer, and Project Editor.
- **Chapter 3 "Building a Project - Scenario1" on page 12** provides a simple "end-to-end" scenario where you will perform a data concatenation.
- **Chapter 4 "Building a Project - Scenario2" on page 53** provides an "end-to-end" scenario where you will perform data transformation, involving a multiplication operation.
- **Chapter 5 "Enterprise Manager Overview" on page 100** provides an overview of the Enterprise Manager which is used to view and monitor projects dynamically.
- **"Glossary" on page 106**.
- **"Index" on page 111**.

1.2 Supporting 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*
- *Message Server Reference Guide*
- *SeeBeyond ICAN Suite Deployment Guide*
- *SeeBeyond ICAN Suite Primer*

See the *SeeBeyond ICAN Suite Primer* for a complete list of eGate Integrator documentation. You can also refer to the appropriate Windows or UNIX documents, if necessary.

1.3 Writing Conventions

The following writing conventions are observed throughout this document.

Table 1 Writing Conventions

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

Additional Conventions

Windows Systems

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

Path Name Separator

This guide uses the backslash ("\") as the separator within path names. If you are working on a UNIX system, please make the appropriate substitutions.

1.4 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: **"Files Required for the Tutorials" on page 16**.

1.5 The SeeBeyond Web Site

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

http://www.seebeyond.com

Chapter 2

Features of the Enterprise Designer

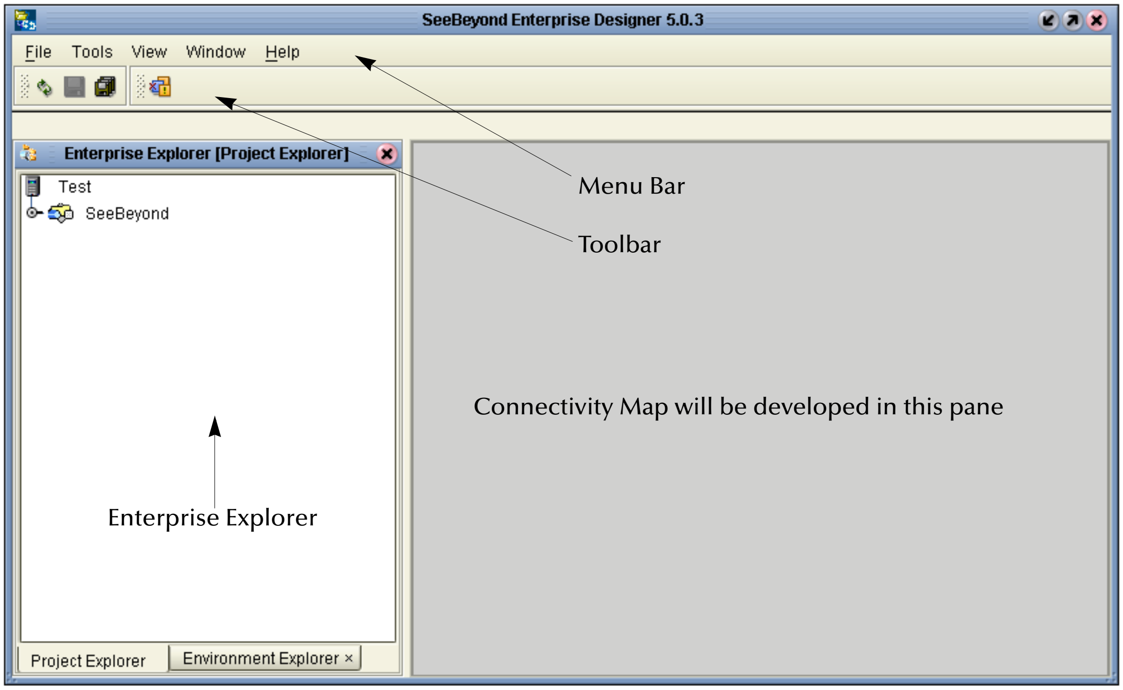
The Enterprise Designer is the graphical user interface (GUI) used to design and implement ICAN 5.0 projects. This chapter overviews the features and interface of the Enterprise Designer window.

This chapter includes

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 Window



Menu Bar

Toolbar

Connectivity Map will be developed in this pane

Enterprise Explorer

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.

2.4 Project Editor

The Project Editor contains the "nuts-and-bolts" of 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 will quickly fill 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.

This chapter includes

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 will be concatenated into the field **FullName** in the output file. **JobTitle**, **HoursWorked** and **Rate** will be 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**.

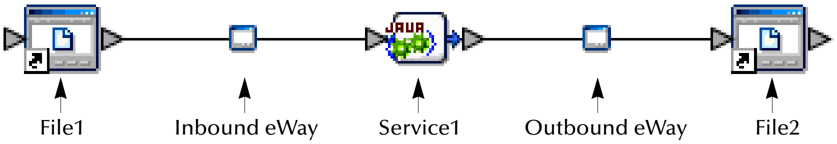
3.2 Project Overview

You will create a basic eGate Project to meet the business challenge described in **“Business Challenge” on page 12**.

3.2.1 Project Description

The finished eGate Project will contain 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 16**
2. **“Create a New Project” on page 18**
3. **“Create a Connectivity Map” on page 20**
4. **“Add Components to the Connectivity Map” on page 22**
5. **“Create a New Object Type Definition” on page 24**
6. **“Configure the Services” on page 28**
7. **“Apply the Business Rules” on page 33**

8 **"Apply the Collaborations" on page 38**

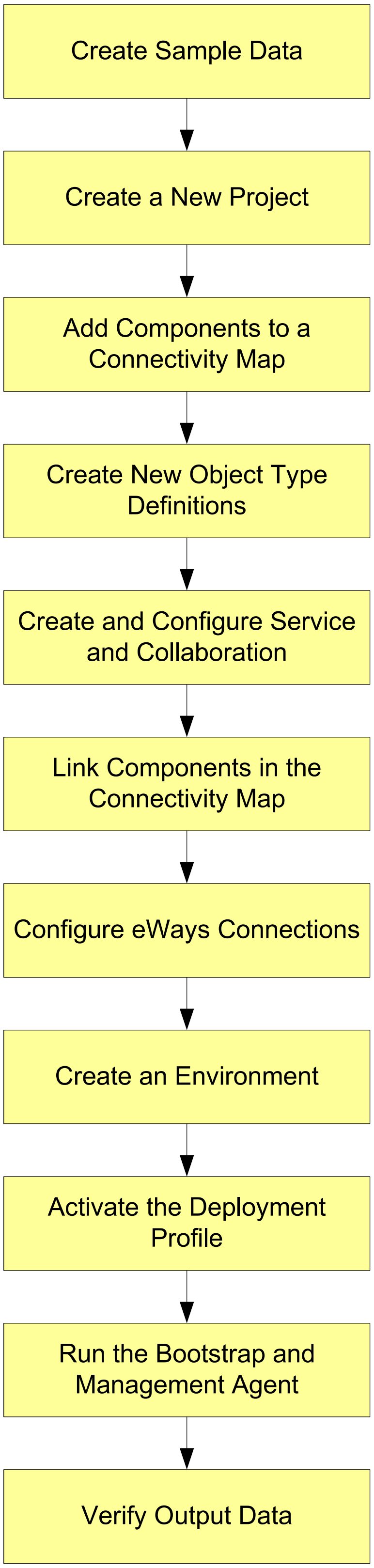
9 **"Configure the eWays" on page 40**

10 **"Create an Environment and Activate the Deployment Profile" on page 43**

11 **"Run the Bootstrap and Management Agent" on page 48**

12 **"Verify the Output Data" on page 50**

Figure 4 Project Diagram


Create Sample Data
Create a New Project
Add Components to a Connectivity Map
Create New Object Type Definitions
Create and Configure Service and Collaboration
Link Components in the Connectivity Map
Configure eWays Connections
Create an Environment
Activate the Deployment Profile
Run the Bootstrap and Management Agent
Verify Output Data

3.3 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

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

In order to use the Enterprise Manager you will also need the **EnterpriseManager-Plugin-Win98ToXP.sar** (required for the Adobe SVG Viewer plug-in for Windows), although you can do the tutorials without using the Enterprise Manager.

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. Scenairo1 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).

3.4 Sample Data

This Project uses the sample files found in the Project1 folder of the **eGateTutorialSample**. The files used in Scenario1 are: **s1input_multiple.dtd**, **s1output_multiple.dtd**, and **TimeCard-Multiple.xml**.

You can also create the files you need. See sections 3.4.1, 3.4.2, and 3.4.3 below.

3.4.1 Create the 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 the 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.

3.4.2 Create the 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 the 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.

3.4.3 Create the 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 the 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 a 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.

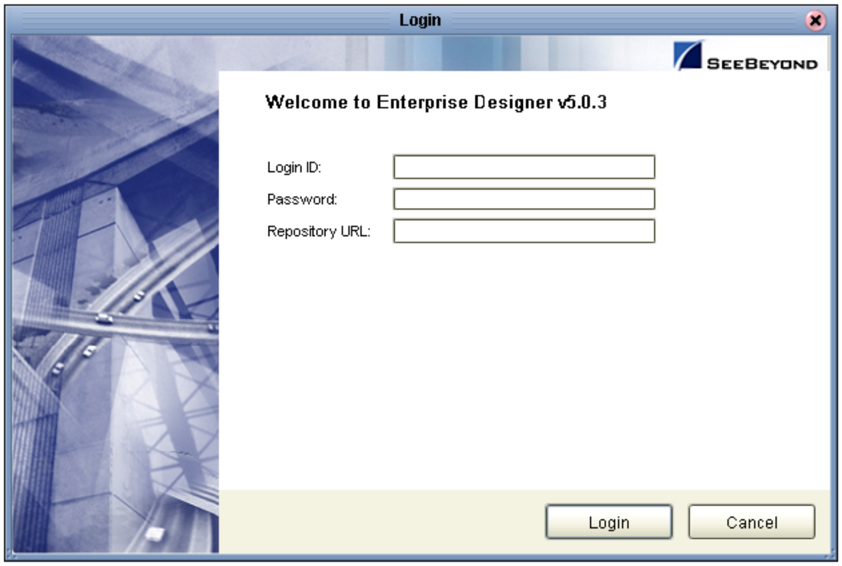
Start the Enterprise Designer

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 5 Enterprise Designer Login



Login

Welcome to Enterprise Designer v5.0.3

Login ID:

Password:

Repository URL:

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

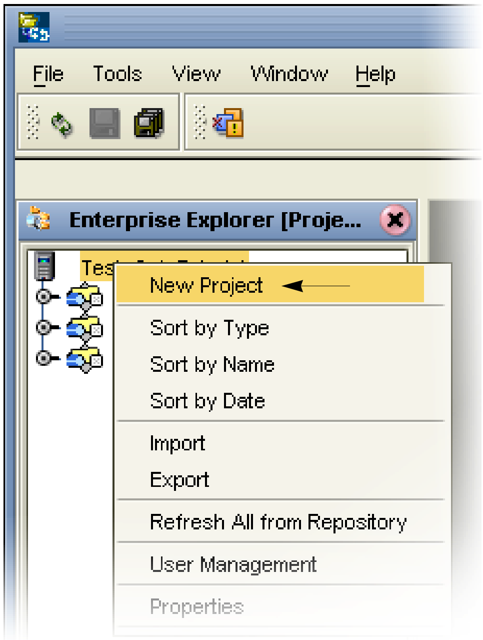
Note: *If your login window doesn't appear or if you get an invalid Username or Password error, make sure your repository is running. Also, be aware that the Username and Password are case sensitive.*

Create and name a 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**.

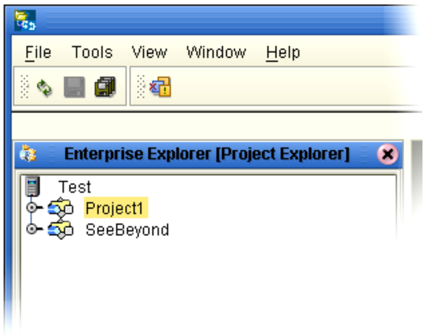
Figure 6 Create a Project



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

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

Figure 7 Project Folder: Project1



3.6 Create a Connectivity Map

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

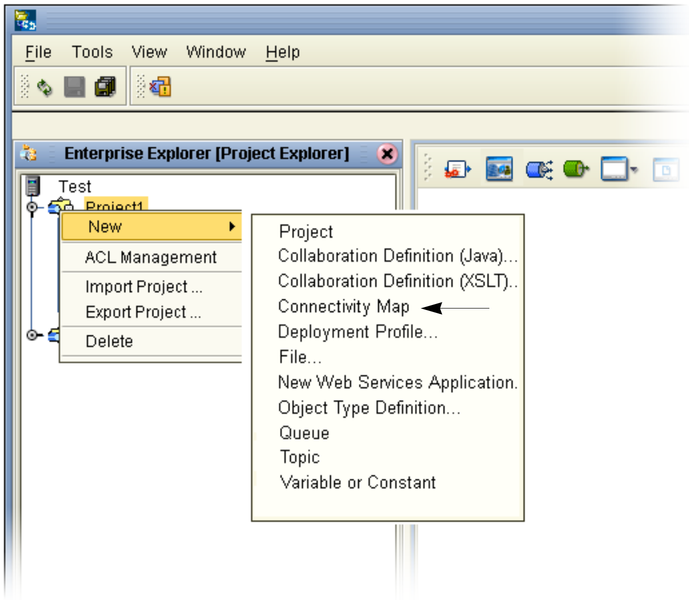
This section describes how to create the Connectivity Map for **Project1**.

Create the 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**.

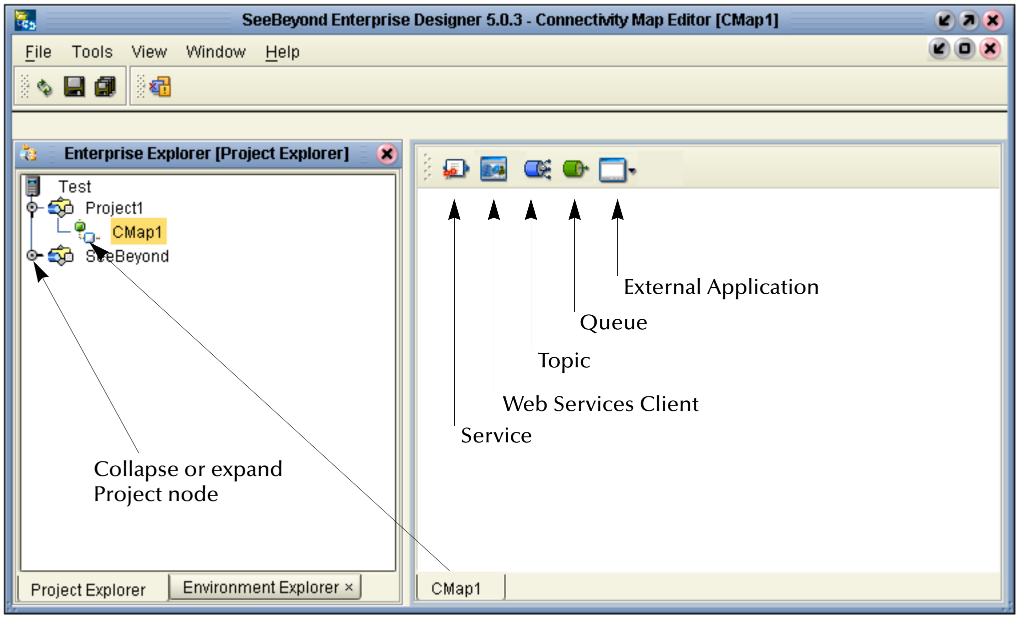
Figure 8 Create a Connectivity Map



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

3 **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.*

3.7 Add Components to the Connectivity Map

In this section, you will add the components to the Connectivity Map. Later you will link the components as shown in **Figure 3 on page 13**.

3.7.1 Add Components to the Connectivity Map

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

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

Create the InputFS

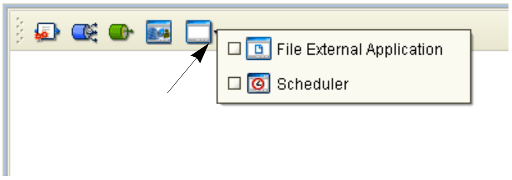
Place an input file (External Application) in your Connectivity Map.

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 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 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**.

4 Click **Save** (Cntl+S).

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

Create Service1

Create a Service which will eventually 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**.

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

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

Create the Output

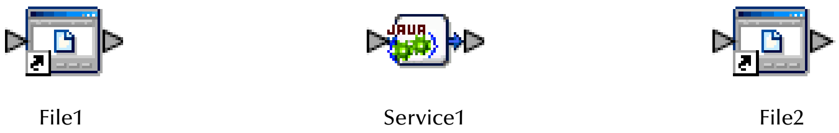
Place an output file (External Application) in your Connectivity Map.

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

This creates a new External Application file.

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

Figure 11 Connectivity Map



Note: *You will link these objects later.*

3.8 Create a New Object Type Definition

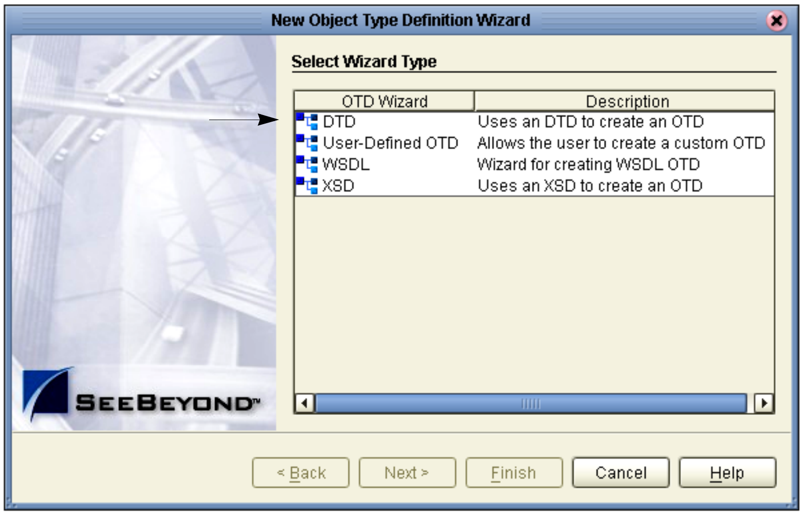
You must create an object type definition before you can configure your Services.

Create an 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**.

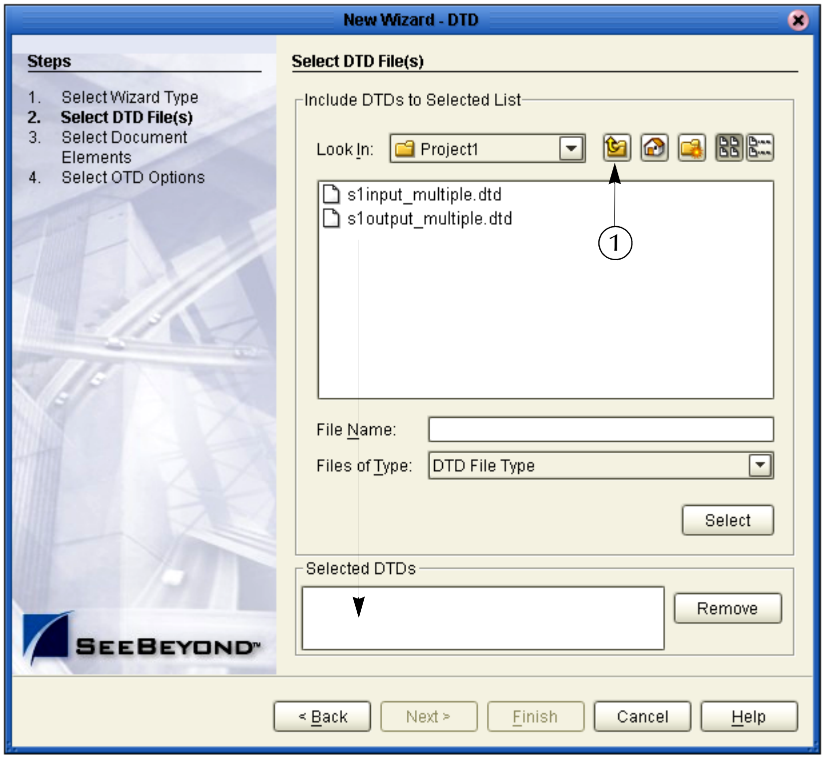
Figure 12 OTD Wizard Selection



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

3. Click the **DTD** item.
4. Click **Next**.
5. Navigate to the sample data folder. For example, **c:\eGateData\Project1**.

Figure 13 Select DTD Files



Look In: Project1

File Name:

Files of Type: DTD File Type

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

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

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

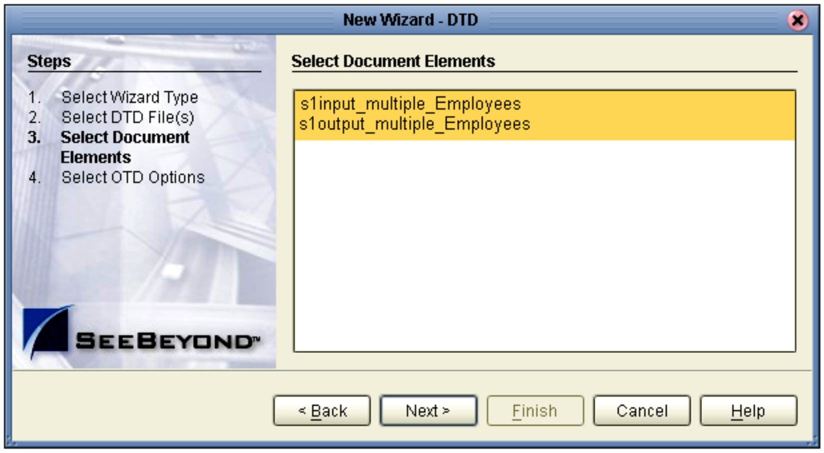
(You can also select both at the same time by holding down the shift key.)

8 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 you will* ***Select Document Elements****, See Figure 14.*

Figure 14 Select Document Elements



New Wizard - DTD

Steps

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

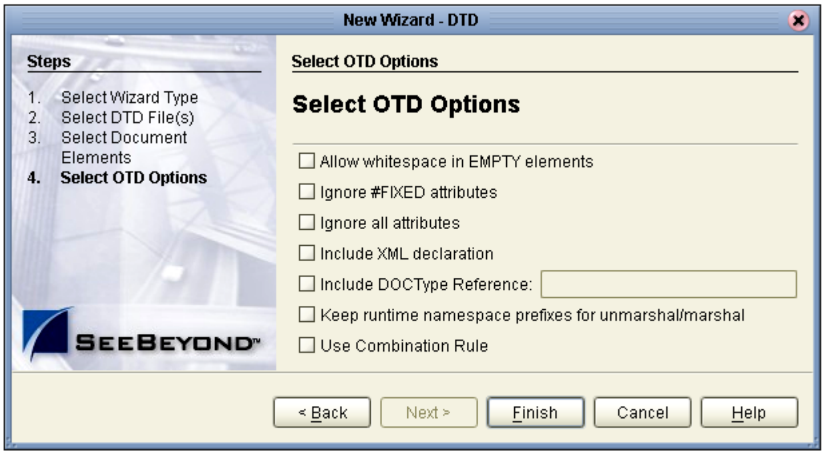
Select Document Elements

s1input_multiple_Employees
s1output_multiple_Employees

9 Select both **s1input_Employees_multiple** and **s1output_Employee_multiple** (hold down the CTRL key while selecting).

10 Click **Next**.

Figure 15 Select OTD Options



New Wizard - DTD

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

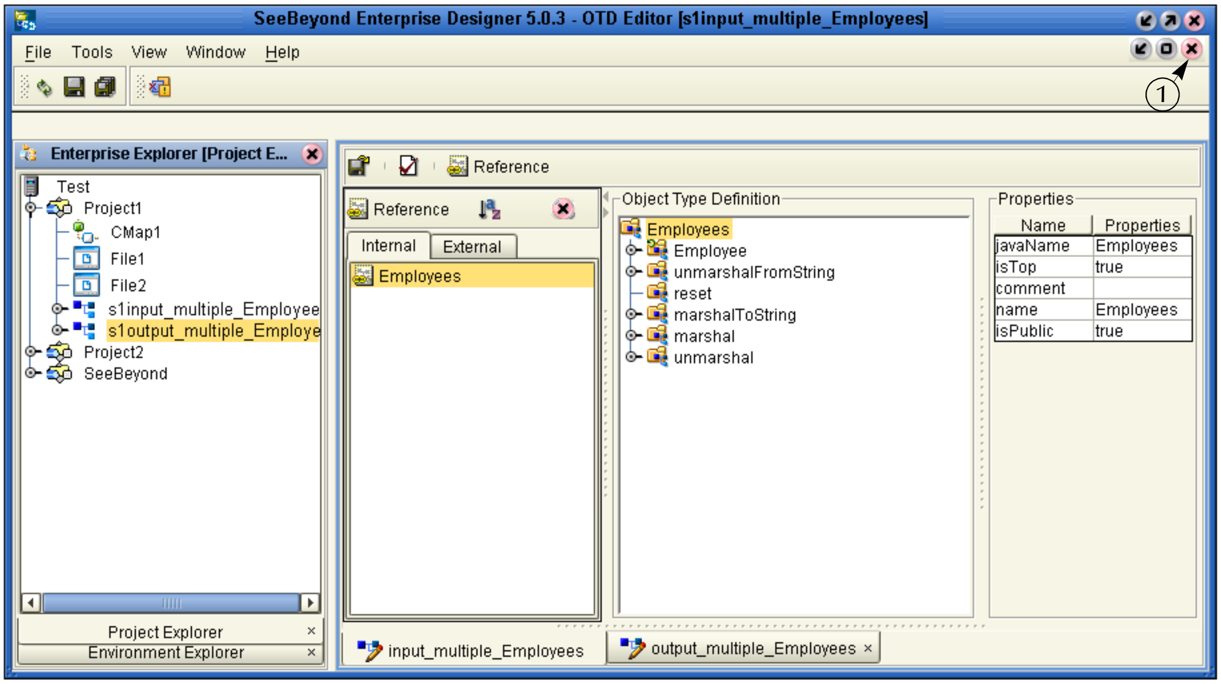
11 Click **Finish** (accepting the defaults).

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

12 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

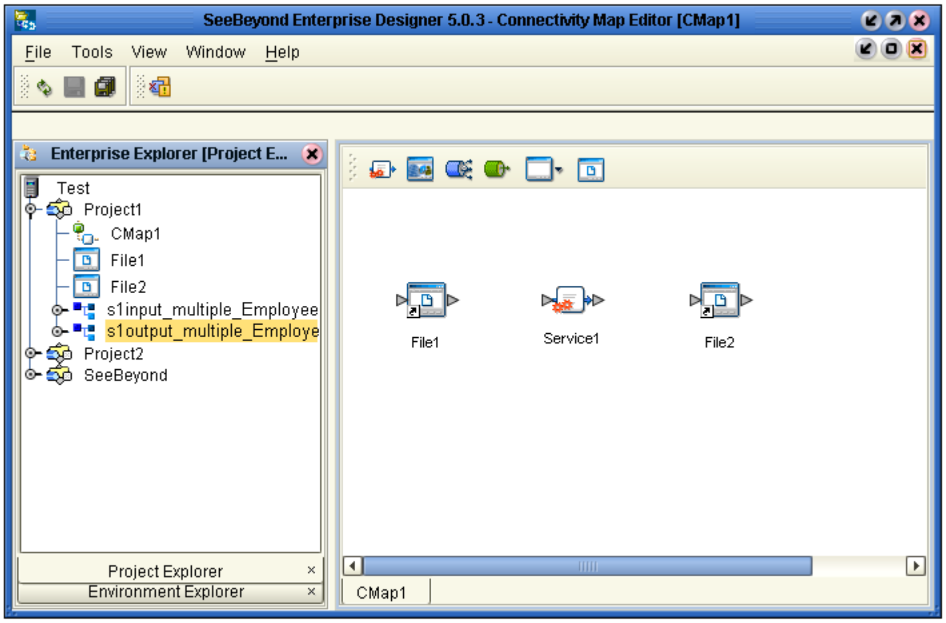


Name	Properties
javaName	Employees
isTop	true
comment	
name	Employees
isPublic	true

13 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 17.

Figure 17 Connectivity Map Window


SeeBeyond Enterprise Designer 5.0.3 - Connectivity Map Editor [CMap1]
File
Tools
View
Window
Help
Enterprise Explorer [Project E...
Test
Project1
CMap1
File1
File2
s1input_multiple_Employee
s1output_multiple_Employe
Project2
SeeBeyond
Project Explorer
Environment Explorer
File1
Service1
File2
CMap1

3.9 Configure the Services

You will use Service wizards in the following steps to configure your Collaboration.

3.9.1 Configure Service1

Configure a Collaboration (Java) definition.

Select Web Service Interface

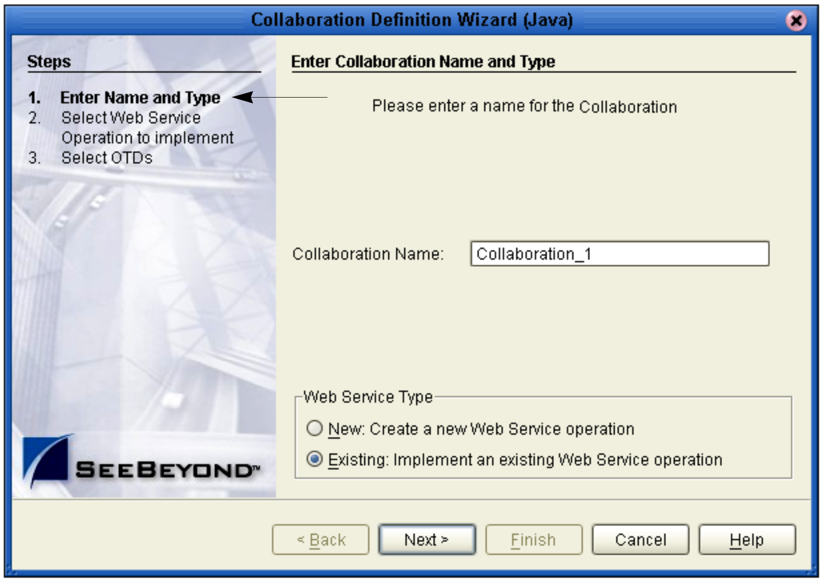
Web Services enable communication between diverse applications using the Internet. Refer to *eGate Integrator User's Guide*, Chapter 6 for information about Web Services.

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

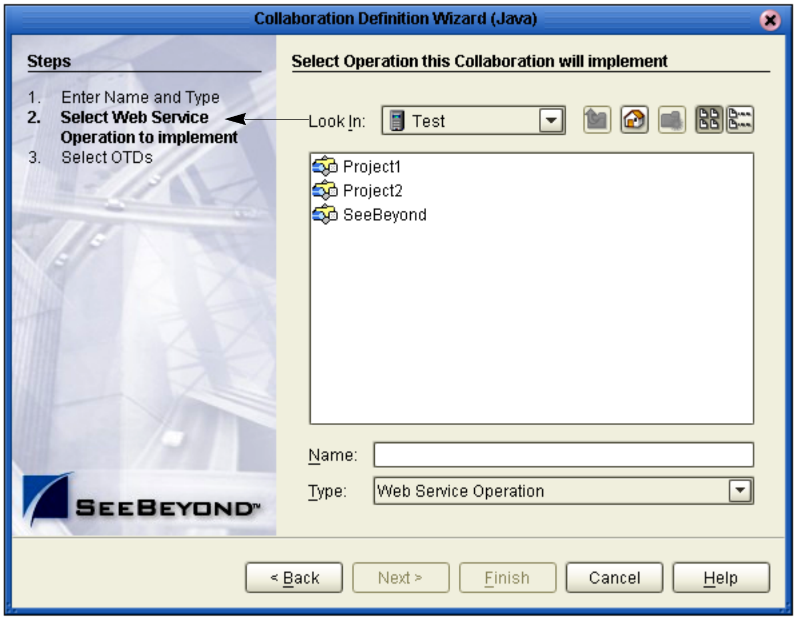
● Existing: Implement an existing Web Service operation

Note: *Accept the default radio button,* ***Existing Web Service****, for this Tutorial.*

3 Accept the default name for the Collaboration Definition, **Collaboration_1**.

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: Test

- Project1
- Project2
- SeeBeyond

Name:

Type: Web Service Operation

You will now select a Web Service that will "implement" a file-read.

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.)

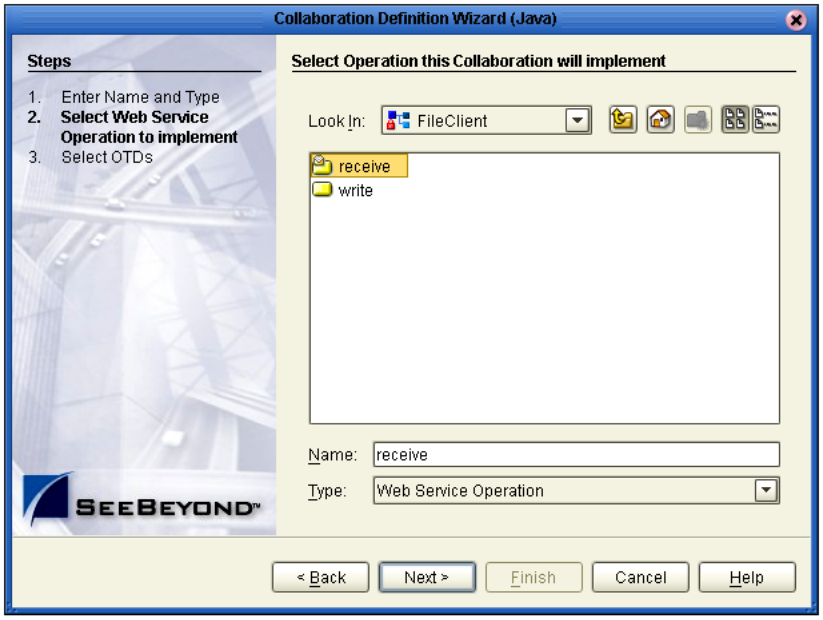
7 Double-click **File**.

8 Double-click **FileClient**

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

Select Operation this Collaboration will implement

Look In: FileClient

- receive
- write

Name: receive

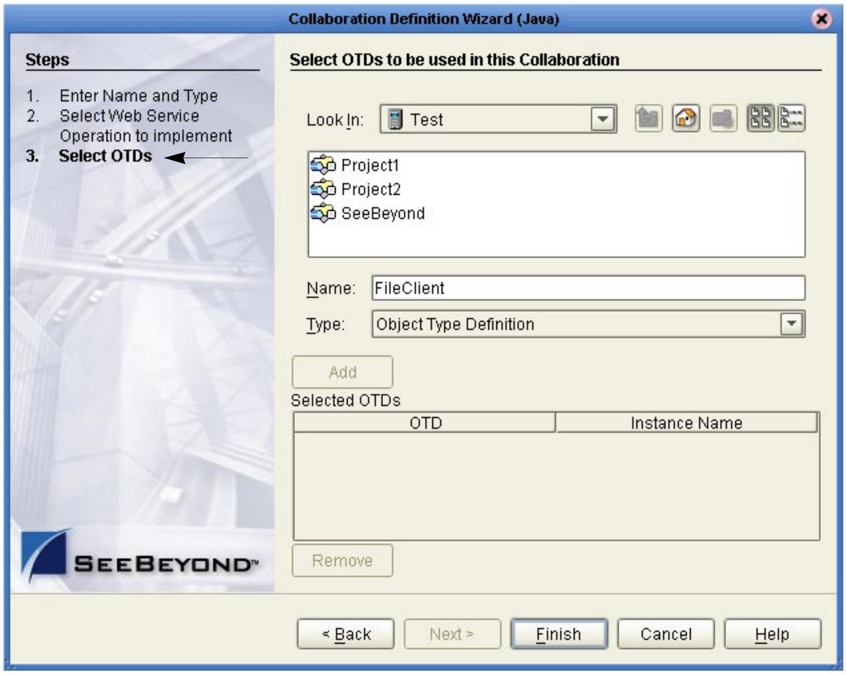
Type: Web Service Operation

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

Select OTDs

In the following steps you will select the **FileClient.receive** OTD (its parameters are automatically defined). You will also select the user defined OTDs.

Figure 21 Select OTDs



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: Test

- Project1
- Project2
- SeeBeyond

Name: FileClient

Type: Object Type Definition

Selected OTDs

OTD	Instance Name

11 Double-click **SeeBeyond**.

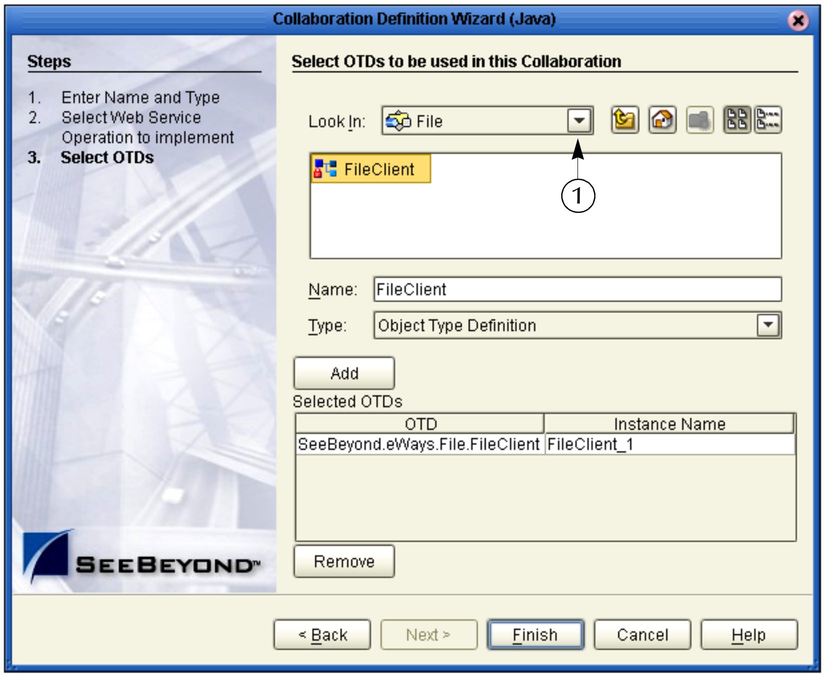
12 Double-click **eWays**.

13 Double-click **File**.

14 Double-click **FileClient**.

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

Figure 22 Select FileClient OTD



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: File

Name: FileClient

Type: Object Type Definition

Selected OTDs

OTD	Instance Name
SeeBeyond.eWays.File.FileClient	FileClient_1

Don't Click Finish yet. You need to install the user defined OTDs.

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

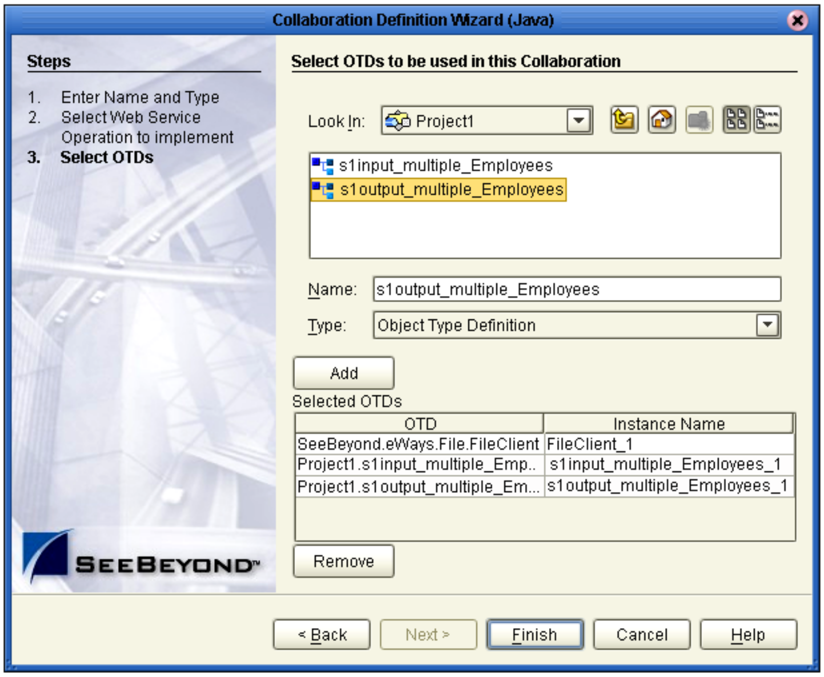
16 Click **Project1** to display the file OTDs.

17 Double-click **s1input_multiple_Employees**.

18 Double-click **s1output_multiple_Employees**.

See the following Figure 23.

Figure 23 Selected OTDs


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:
Project1
s1input_multiple_Employees
s1output_multiple_Employees
Add
Remove
< Back
Next >
Finish
Cancel
Help

Name: s1output_multiple_Employees

Type: Object Type Definition

Selected OTDs

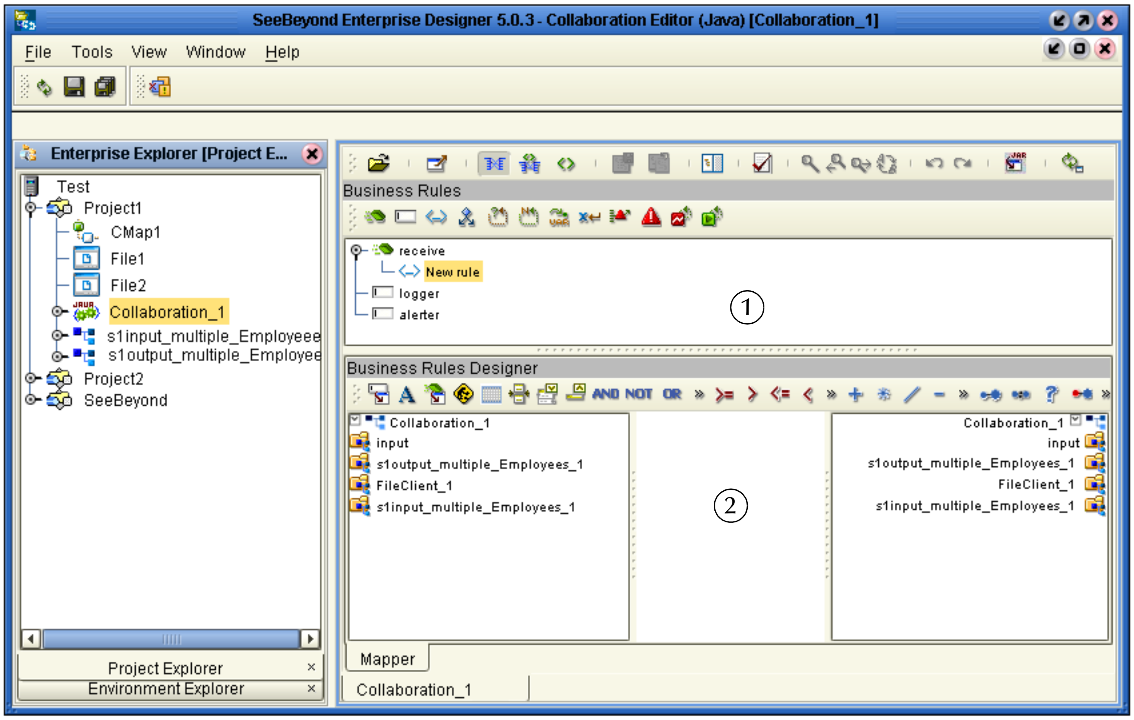
OTD	Instance Name
SeeBeyond.eWays.File.FileClient	FileClient_1
Project1.s1input_multiple_Emp..	s1input_multiple_Employees_1
Project1.s1output_multiple_Em...	s1output_multiple_Employees_1

All three OTDs appear in the list of OTDs, as shown in the previous Figure 23.

19 Click **Finish**.

The Collaboration Editor (Java) appears. See the following Figure 24.

Figure 24 Collaboration Editor (Java)



3.10 Apply the Business Rules

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

Unmarshal Text

You must first unmarshal the text input before you can concatenate the elements.

1 Double-click **input** in the left pane of the Business Rules Designer to expand the view.

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 Right-click **s1input_multiple_Employees_1** in the left pane of the Business Rules Designer.

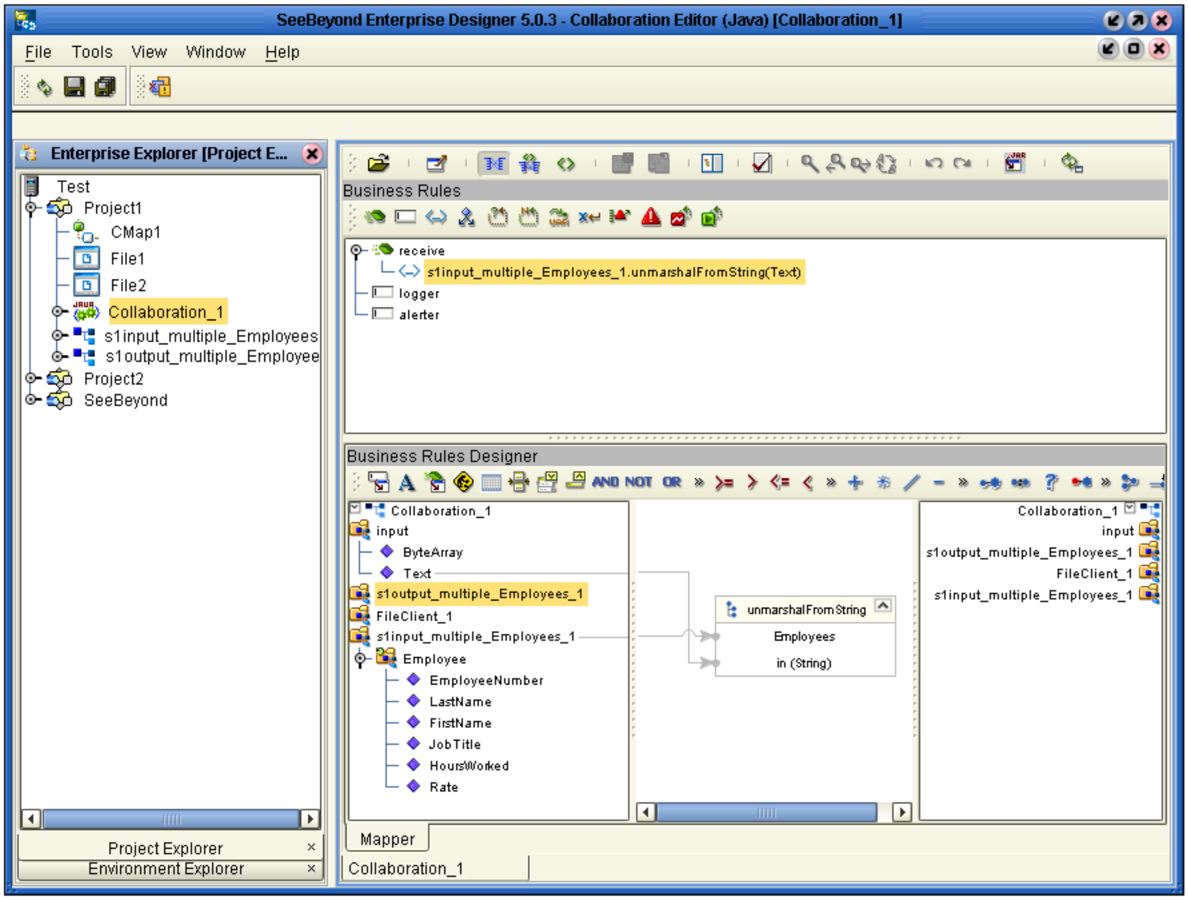
4 Click **Select a method to call**.

5 Select **unmarshalFromString(java.lang.String in)** from the list.

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

6 Drag Text from **input** into **in(String)**. See the following Figure 25.

Figure 25 Unmarshal Text



Connect Input and Output OTD Nodes

In the following steps you will 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 Figure 24 and Figure 25 reside in both the right and left pane of the Business Rules Designer window. Keep in mind that the GUIs (**FileClient_1** for example) refer to the same object.*

Note: *As you begin mapping, the Connectivity map graphics disappear. Click on any business rule in the top pane to redisplay the Connectivity Map graphic and the business rule.*

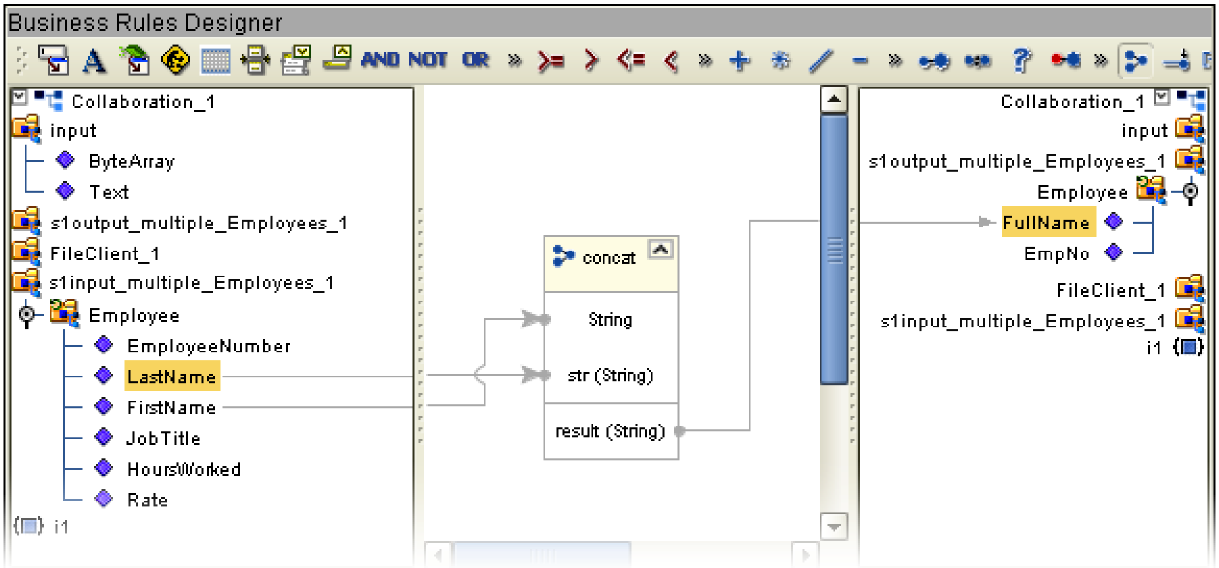
2. Connect **EmployeeNumber**, from **s1input_multiple_Employees_1** in the left pane to **EmpNo** of **s1output_multiple_Employees_1** in the right pane.

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

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

   See the following Figure 26.

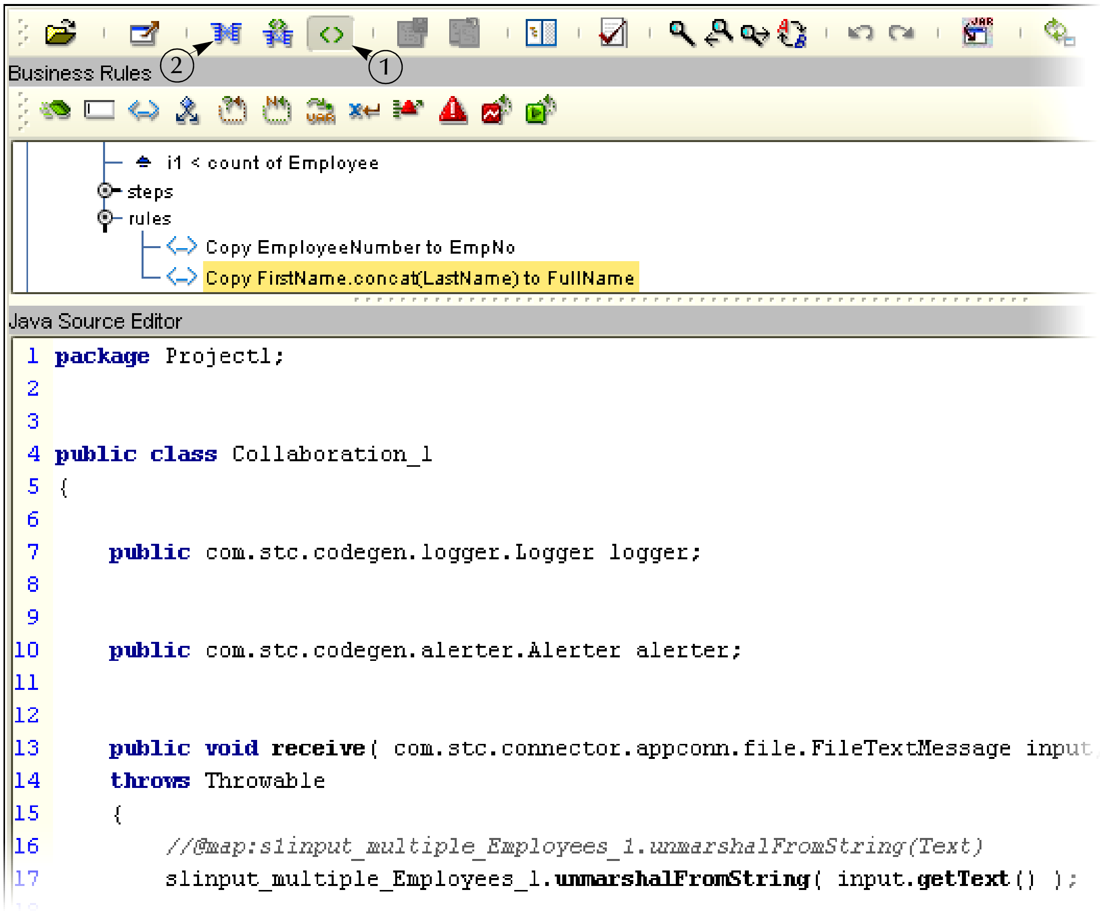
Figure 26 Concatenation Logic



View Source Code

You may view (or modify) the source code by clicking the **Source code mode** icon (number 1 in Figure 27. You may toggle back to Standard mode by clicking the **Standard mode** icon (number 2 in Figure 27).

Figure 27 Source Code Mode



```
i1 < count of Employee
steps
rules
    Copy EmployeeNumber to EmpNo
    Copy FirstName.concat(LastName) to FullName
```

```
package Project1;


public class Collaboration_1
{

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


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


    public void receive( com.stc.connector.appconn.file.FileTextMessage input,
    throws Throwable
    {
        //@map:s1input_multiple_Employees_1.unmarshalFromString(Text)
        s1input_multiple_Employees_1.unmarshalFromString( input.getText() );
```

Marshal Text

Since you began by unmarshalling data from string, you must now marshal the data to string. The marshal method prepares for a write method.

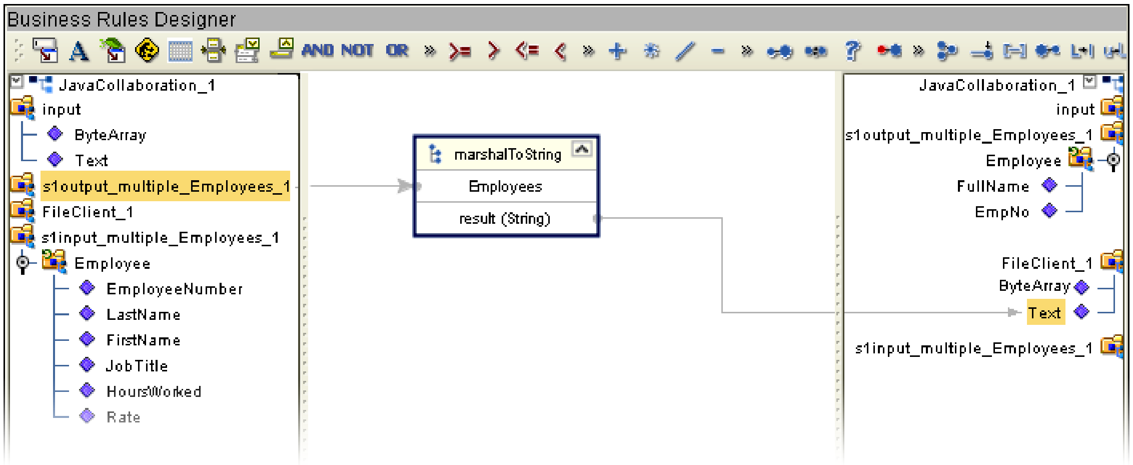
1. Double-click **FileClient_1** in the right pane to expand the view.
2. Right-click **s1output_multiple_Employee_1** in the left pane, and click **Select a method to call**.
3. Click **marshalToString()**.

   The **marshalToString** method box appears.

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

   See the following Figure 28.

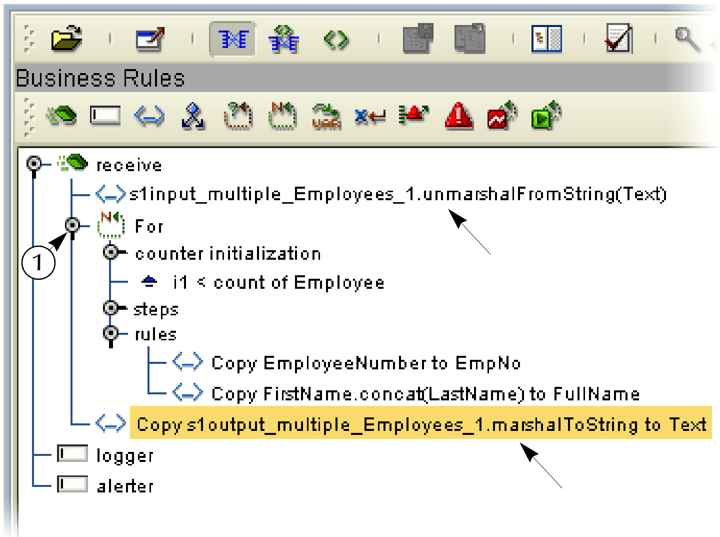
Figure 28 Marshal to String



Your Business Rules should look like Figure 29. The marshal and unmarshal actions should be on the same level (and “marshal” should not be within the loop because that would cause a write after each individual employee record).

(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 29.)

Figure 29 View Business Rules


Business Rules
receive
s1input_multiple_Employees_1.unmarshalFromString(Text)
1
For
counter initialization
i1 < count of Employee
steps
rules
Copy EmployeeNumber to EmpNo
Copy FirstName.concat(LastName) to FullName
Copy s1output_multiple_Employees_1.marshalToString to Text
logger
alerter

Write FileClient_1

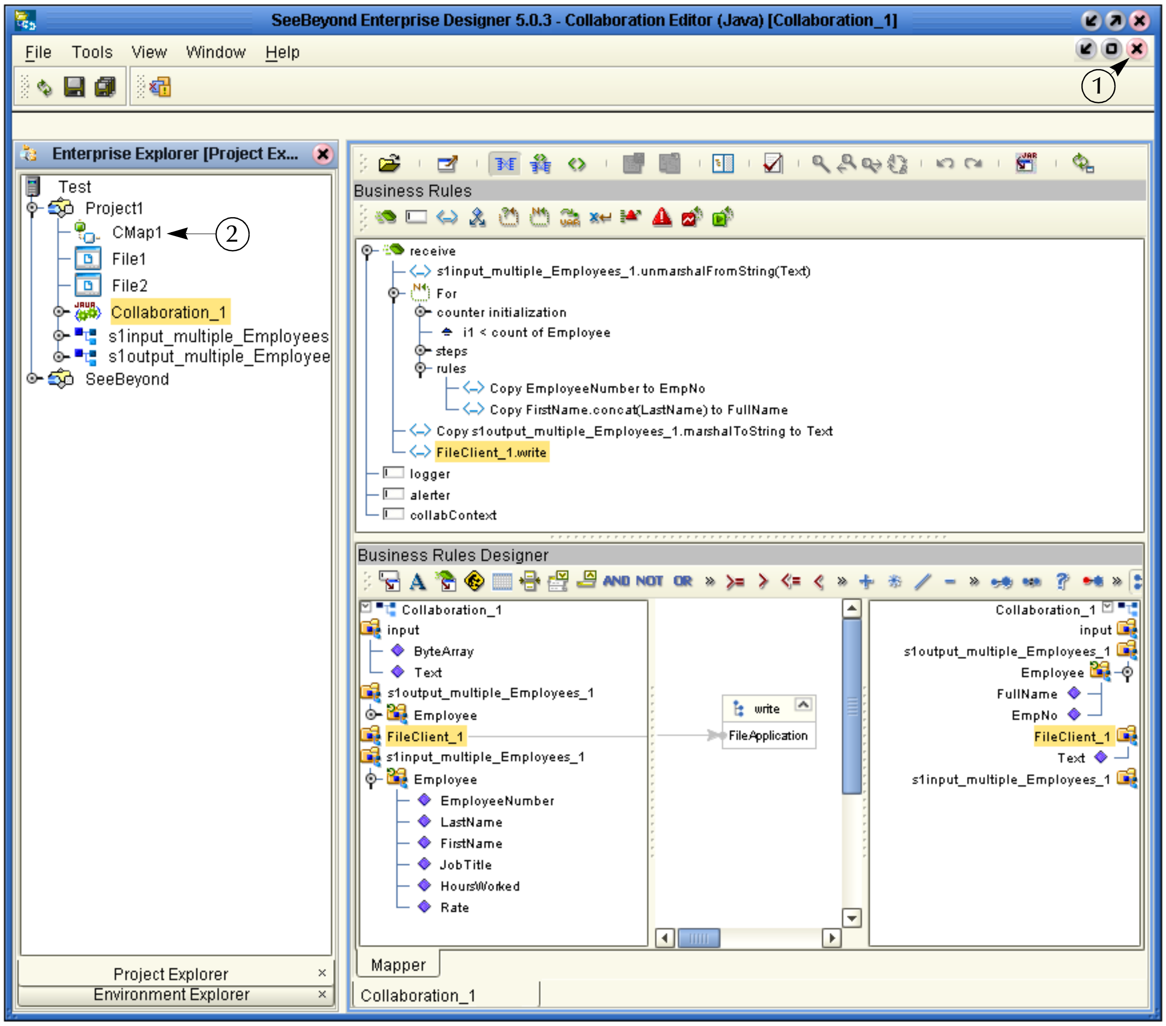
Write to the output file.

1 Right-click **FileClient_1** in the left pane, and click **Select a method to call**.

2 Click **write()**.

See the following Figure 30.

Figure 30 Write Output File


SeeBeyond Enterprise Designer 5.0.3 - Collaboration Editor (Java) [Collaboration_1]
File Tools View Window Help
1
Enterprise Explorer [Project Ex...
Test
Project1
CMap1
2
File1
File2
Collaboration_1
s1input_multiple_Employees
s1output_multiple_Employee
SeeBeyond
Project Explorer
Environment Explorer
Business Rules
Business Rules Designer
Mapper
Collaboration_1

```
receive
  s1input_multiple_Employees_1.unmarshalFromString(Text)
  For
    counter initialization
      i1 < count of Employee
    steps
    rules
      Copy EmployeeNumber to EmpNo
      Copy FirstName.concat(LastName) to FullName
  Copy s1output_multiple_Employees_1.marshalToString to Text
  FileClient_1.write
logger
alerter
collabContext
```

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

This completes the Collaboration. You will now 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 30 above) or by double-clicking the Connectivity Map (number 2 in Figure 30 above) name, **CMap1**.

3.11 Apply the Collaborations

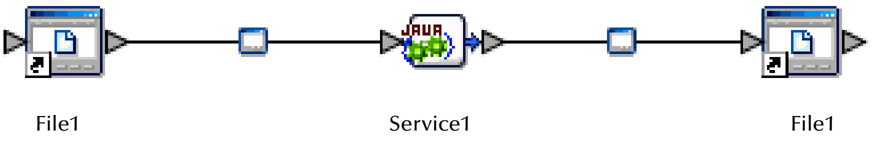
You have configured your Collaboration Definition (business rules) in the previous section. You are now ready to bind your Collaboration Definition with the Service using the drag-and-drop method.

3.11.1 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 three unlinked objects. You will connect nodes between the modules as shown in the following Figure 31.

Figure 31 Linked Objects in Connectivity Map

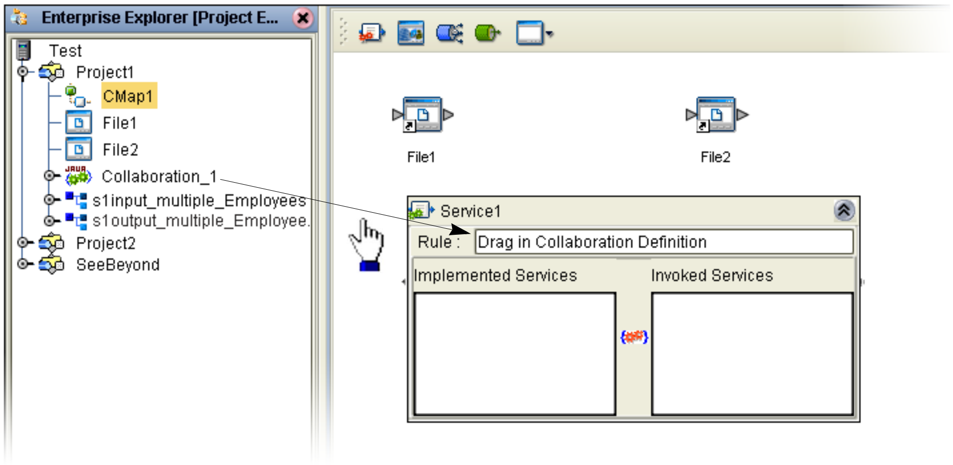


1 Double-click **Service1** in the Connectivity Map to open the Service window. (You can also reposition the window after it is open.)

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 32.

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 32 Using the Drag and Drop method



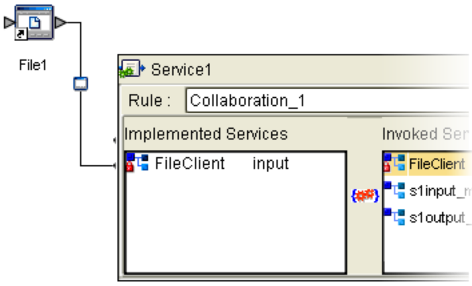
Rule : Drag in Collaboration Definition

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 33. Since there is only one option the application automatically makes the connection.

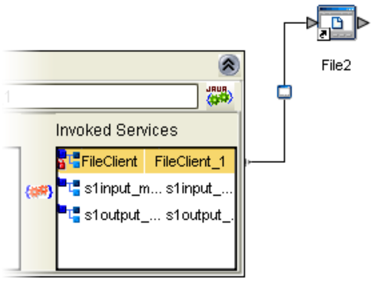
Figure 33 Connect FileClient to File1



Rule : Collaboration_1

4. Connect **FileClient** to **File2** as shown in the following Figure 34.

Figure 34 Connect FileClient to File2



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

6. Click **Save All**.

3.12 Configure the eWays

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

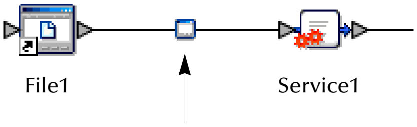
- The input eWay polls a directory on the local file system. For example: **c:\eGateData\Project1**, for any XML file (***.xml**).
- The Output eWay writes the results to an output file (**.dat**) on the local file system. For example: **c:\eGateData\Project1**.

Configure the Inbound eWay

You will identify the file eWay shown in the following Figure 35 as an inbound file and then set the properties to identify the file and its location.

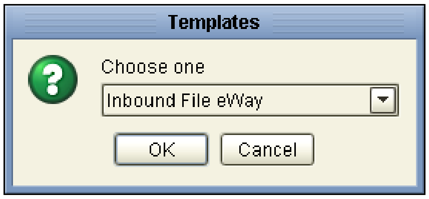
1. Double-click the first eWay -□- . Select **Inbound File eWay** from the drop-down list.

Figure 35 Inbound eWay



2 Click **OK**.

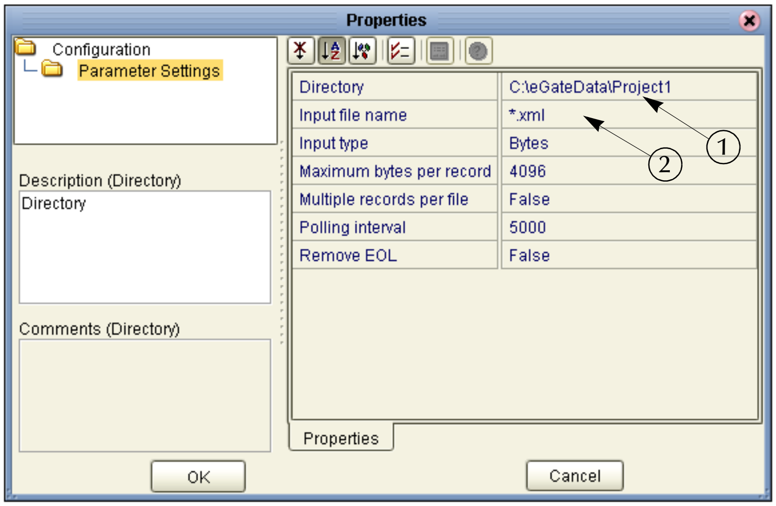
Figure 36 External Type Dialog Box



Choose one
Inbound File eWay

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

Figure 37 Properties Configuration (Inbound eWay)



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 arrows in the previous Figure 37. These are the fields you will edit.

3 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 37 above).

4 Click once on the current **Input File Name** field and enter ***.xml**. The eWay will read any files with the .xml extension (number 2 in Figure 37 above).

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

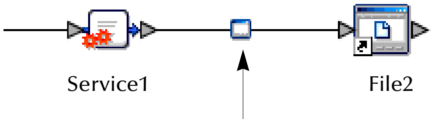
Note: *If the **Polling interval** value is left unchanged, the eWay polls the directory location every five seconds (5000 milliseconds).*

Configure the Outbound eWay

You will identify the file eWay shown in the following Figure 38 as an outbound file and then set the properties to identify the file and its location.

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

Figure 38 Outbound eWay



2 Click **OK**.

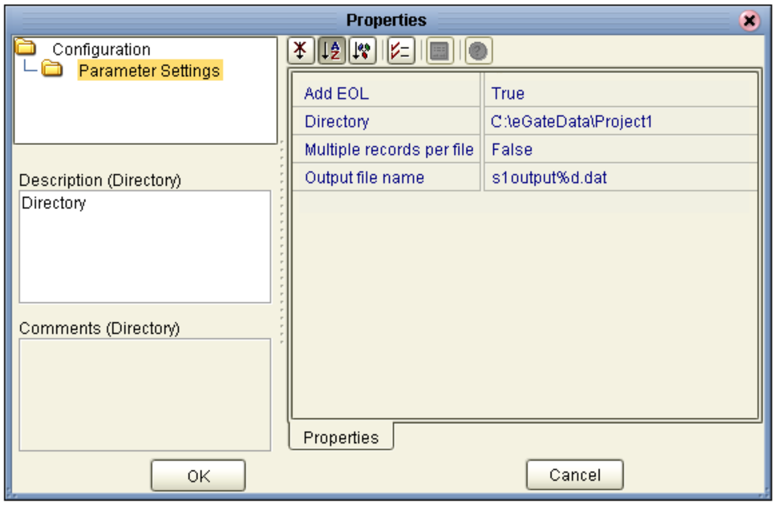
The Properties dialog box appears.

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

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

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

Figure 39 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 43**.*

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

You have now configured both the input and output eWays and are ready to deploy your Project components in an Environment containing a Logical Host and External Applications. You may skip to **"Create an Environment and Activate the Deployment Profile" on page 43**.

Multiple records per file (True or False)

This setting (refer to Figure 37 and Figure 39) specifies if multiple records will be obtained or sent per file. Multiple records (messages) will be generated per line up to the maximum bytes per record.

Input 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 in 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.

Output eWay

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

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

A **False** setting means that only one message will be written in one file. New messages will be 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.

3.13 Create an Environment and Activate the Deployment Profile

In this section, you will create and activate the Deployment Profile. Before you can create the profile, you must create an Environment.

3.13.1 Create an 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 an 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 will appear in the **LogicalHost1** window. You will bind your Collaboration to this server.

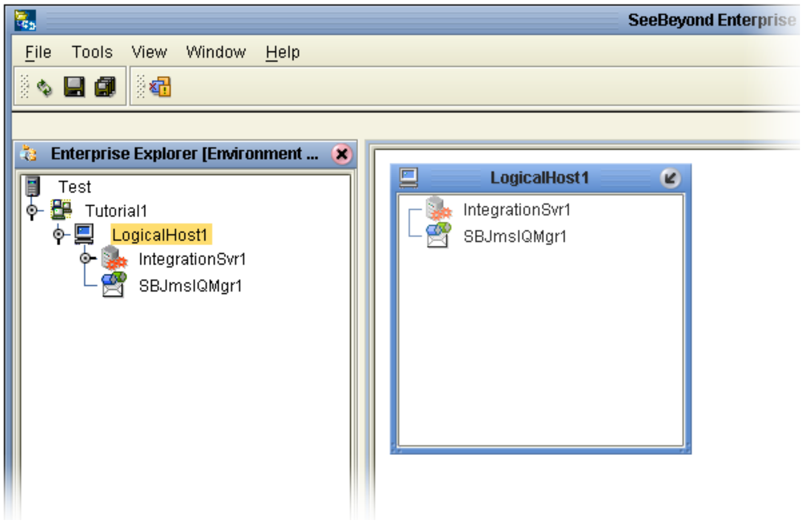
Create a JMS IQ Manager

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

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

   SBJmsIQMgr1 will appear in the **LogicalHost1** window.

Figure 40 Environment with Logicalhost

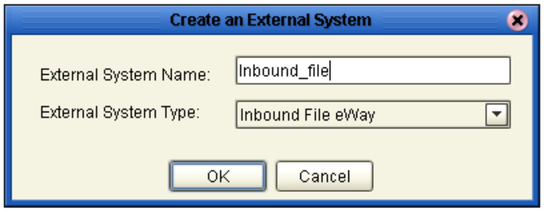


These servers automatically appear in the **LogicalHost1** dialog box.

Create an External File System for the Inbound eWay

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

Figure 41 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.

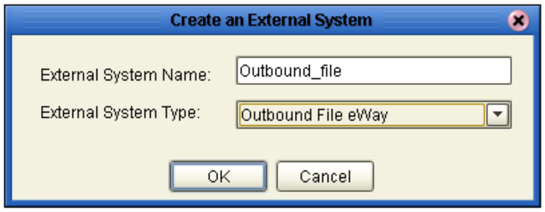
3 Select **Inbound File eWay**. Click **OK**.

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

Create an External File System for the Outbound eWay

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

Figure 42 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.

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."*

3.13.2 Create a New Deployment Profile

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

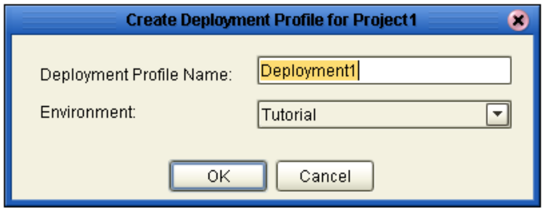
Deployment Profile

A Deployment Profile contains information about how Project components will be 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**. Click **New** and **Deployment Profile**.

Figure 43 Deployment Profile



Create Deployment Profile for Project1

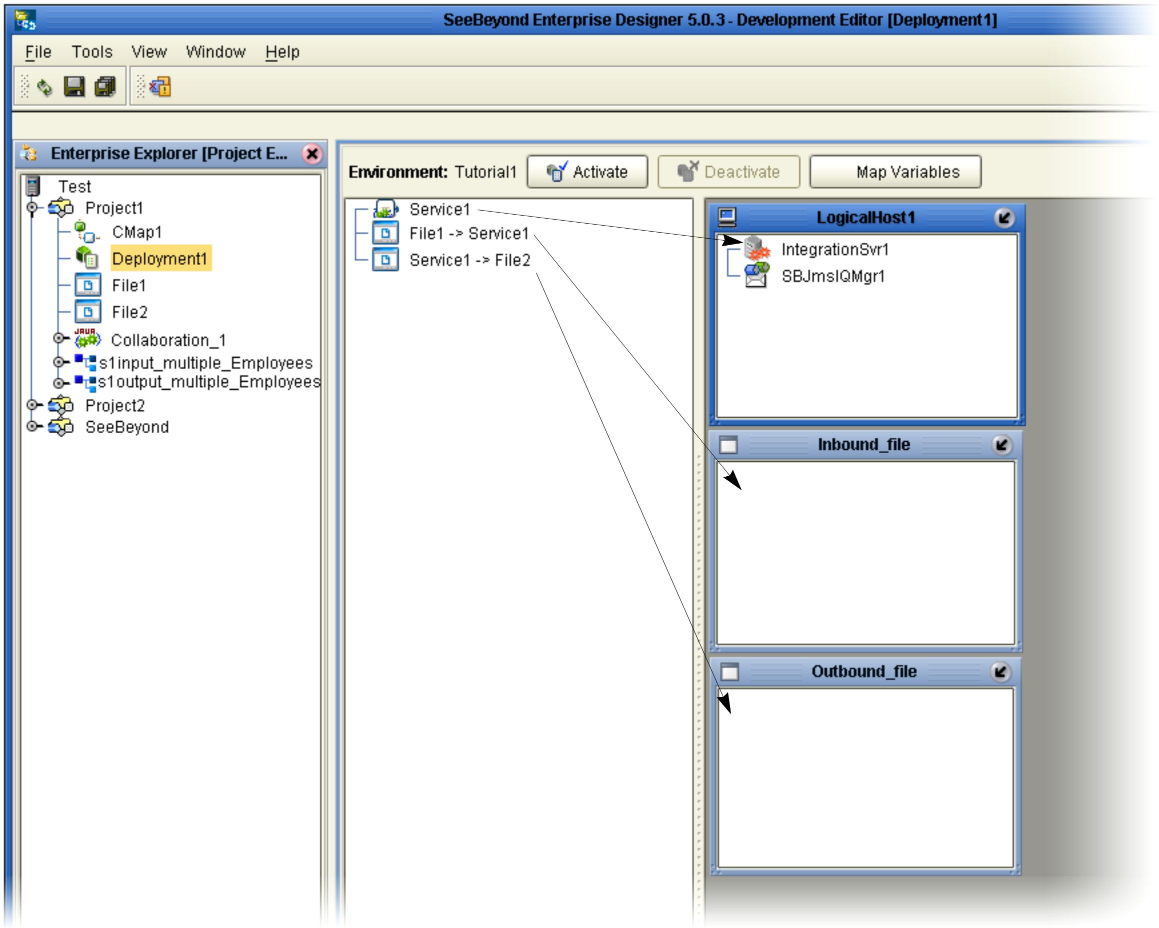
Deployment Profile Name: Deployment1

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. You will deploy these components using the drag and drop method.

Figure 44 Environment, Drag and Drop Components



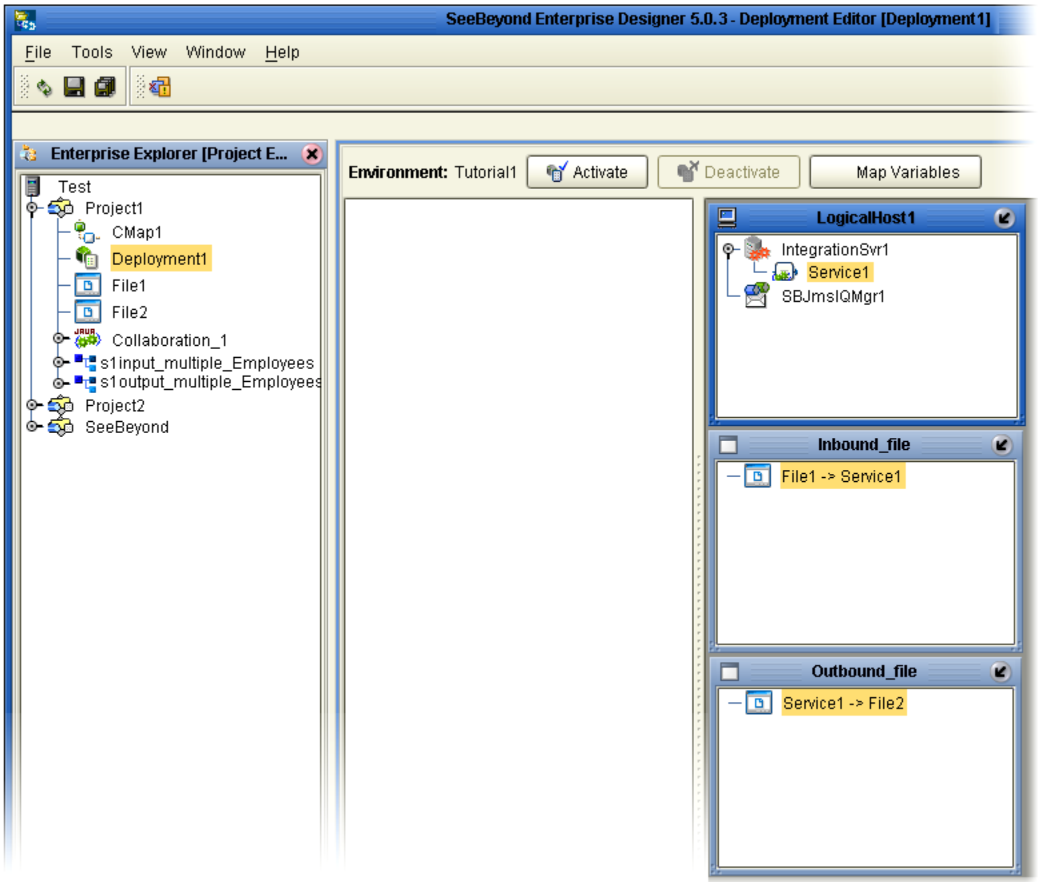
3.13.3 Deploy your Project

Drag and drop components into the server dialog boxes.

1 Drag-and-drop your Service, **Service1,** into your Integration Server, **IntegrationSvr1**. (See the previous Figure 44.)

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

Figure 45 Populated Environment



After populating your Environment with all the available components, the center pane of the Environment editor will be empty. All your components reside in the servers, as shown in Figure 45 above.

3.13.4 Run your Project

When you deploy **Project1**, a deployment file is created and will later be picked up by the Logical Host. The Project you created with Enterprise Designer will be “code generated” into a package that will be passed to the Repository. The Logical Host will pick up this package from the Repository.

If you want to do a “hot deploy,” which means that the Logical Host will pick up the current package in real time after the bootstrap has been started, you could click **YES** in the Activate Dialog box when prompted “Do you wish to apply Logical host(s) immediately?” (See below).

Activate Environment

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

2 Click the **Activate** button. The Activation in Progress message appears.

Figure 46 Activation in Progress Message

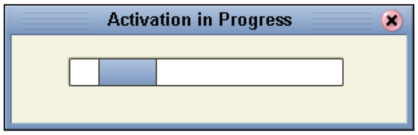
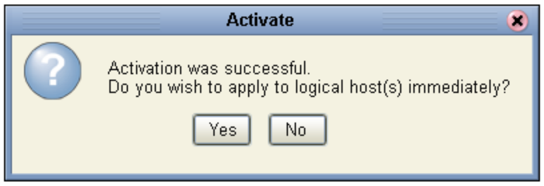


Figure 47 Successful Activation

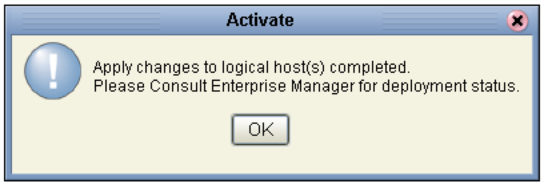


Activation was successful.
Do you wish to apply to logical host(s) immediately?

Activating the Deployment Profile may take a few minutes. When the activation is complete, the message above appears.

3 Click **No** to Finish. (If your bootstrap has already been started, click **Yes**.)

Figure 48 Changes Applied to Logical Host



Apply changes to logical host(s) completed.
Please Consult Enterprise Manager for deployment status.

If you click **Yes** in step 3 above, this message verifies that changes were applied to the Logical Host.

4 Click **OK**.

Note: *If you want to make changes to your project Deactivate your deployment first. You can then Reactivate your deployment to apply your changes.*

3.14 Run the Bootstrap and Management Agent

The Bootstrap process executes your Project and begins the process of polling your input data. The Bootstrap process is run from a command prompt or by executing **bootstrap.bat**. Bootstrap will pick up the deployment profile the first time it runs; after that you would redeploy and click **Yes** (in Figure 47) to apply the most recent changes to the logical host(s).

Logical Host Overview

At run time, 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).

3.14.1 Run the Bootstrap

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

Two ways to run the bootstrap are explained below:

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 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/Test**

   localhost:12000 the default port number. Test is the repository name in this example.

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

   Enter the Administrator name.

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

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

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

   Enter an environment name that will be 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**.

   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.*

Run Bootstrap 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 49. (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*),

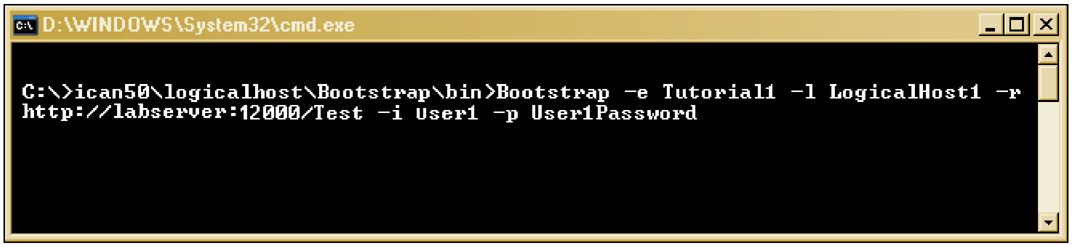
logicalhost_name is the name of your Logical Host (for example, *LogicalHost1*),

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

username is your user name, and

password is your password.

Figure 49 Bootstrap Command Example



```
C:\>ican50\logicalhost\Bootstrap\bin>Bootstrap -e Tutorial1 -l LogicalHost1 -r
http://labserver:12000/Test -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.

3.15 Verify the Output Data

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

View the output

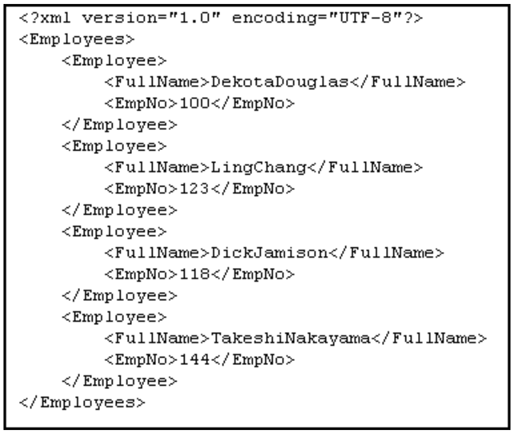
1 Use Windows Explorer to navigate to **c:\eGateData\Project1**.

2 See that the Input eWay, after processing the input file, renamed the input file to **TimeCard_Multiple.~in**.

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.

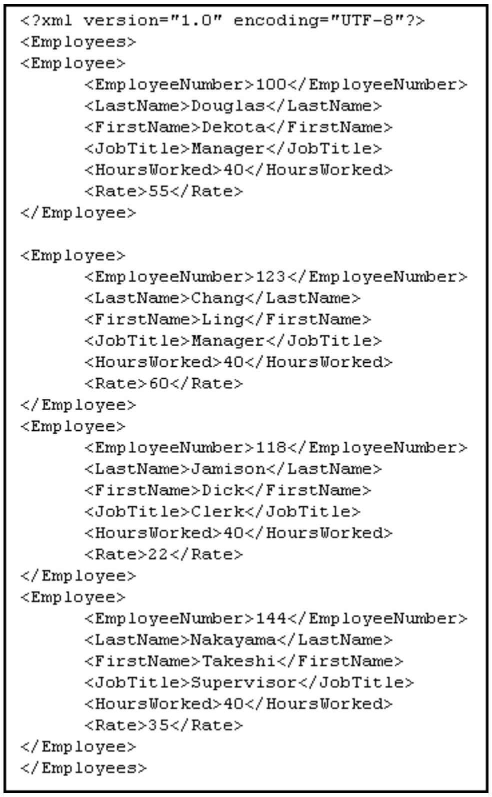
Figure 50 Output File



```
<?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 50. (The test files are editable and you can experiment with the input. Your data may not match the samples shown in the Tutorial.)

Figure 51 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>
```

3.15.1 Edit Input File in Real Time

With your Repository server running, you may edit the input, TimeCard, 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 39).

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

Chapter 4

Building a Project - Scenario2

This tutorial provides step-by-step procedures for creating and testing an eGate Project with This Project 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.

4.0.1 Required Resources

See **"Files Required for the Tutorials" on page 16** in Scenario1 and **"Sample Data" on page 57** of this tutorial.

This chapter includes

4.1 Business Challenge

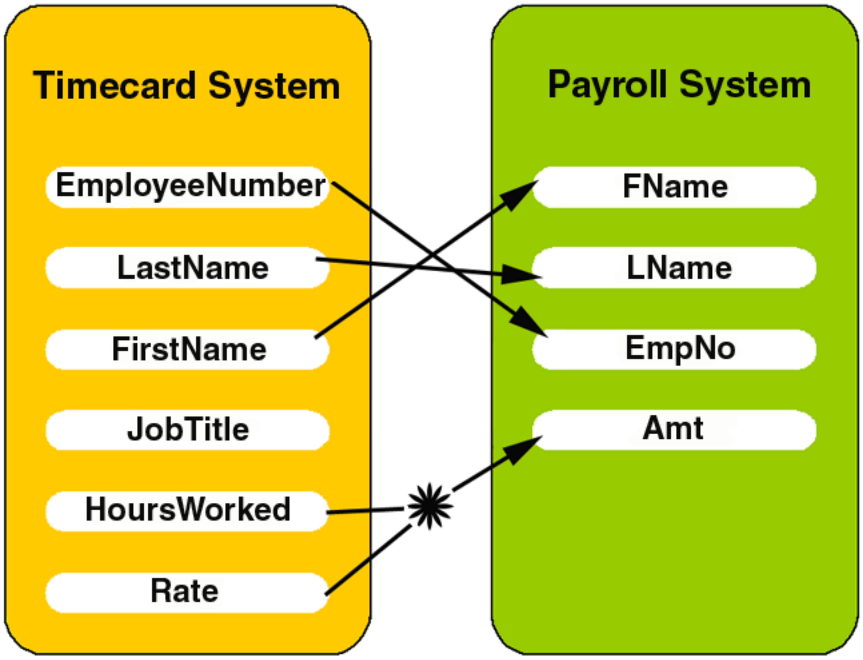
The Project described in this chapter provides a solution to the following business challenge:

- 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 52 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 52 The Relationship Between the Two Systems



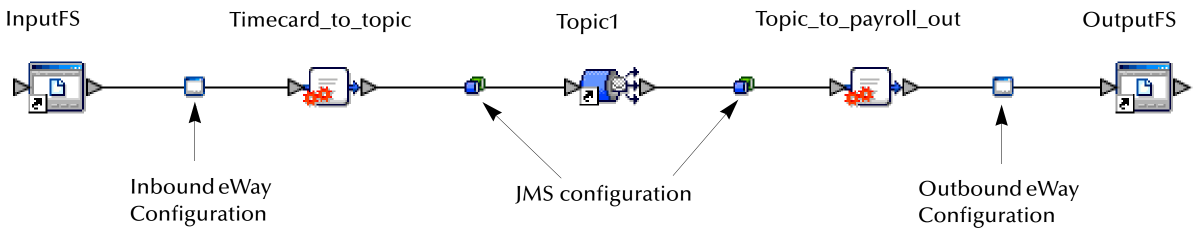
4.2 Project Overview

You will create a basic eGate Project to meet the business challenge described in **"Business Challenge" on page 54**. If you did the Scenario1 tutorial, you may use the same Repository for Scenario2.

4.2.1 Project Description

The finished eGate Project will contain components used to move the data throughout the Project and publish the data in the appropriate format for the Payroll system.

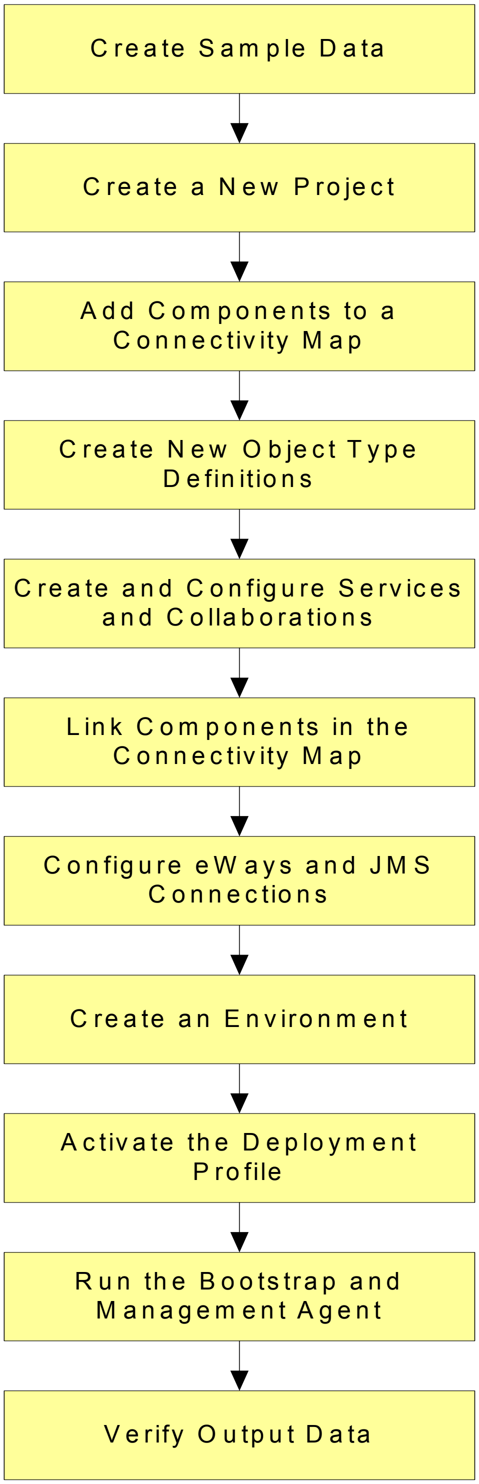
Figure 53 Project Connectivity Map



These components perform the following functions:

- **InputFS**: The InputFS *External Application* represents the Timecard system containing the Input XML data file. (InputFS stands for Input File System.)
- **Inbound eWay**: The inbound eWay polls a specified location on the file system for the input XML data file.
- **Timecard_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 **Timecard_to_topic** to **Topic_to_Payroll_out**.
- **Topic_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.
- **OutputFS**: This represents the Payroll system—the output data file.

Figure 54 Project Diagram



Create Sample Data
Create a New Project
Add Components to a Connectivity Map
Create New Object Type Definitions
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 **eGateTutorialSample**. The files used in Scenario2 are: **s2input.dtd**, **s2output.dtd**, and **Timecard.xml**. To keep it simple, the input file contains only one record. Only one record will appear in the output file after you successfully run your project.

You can also create the files you need. See sections 4.3.1, 4.3.2, and 4.3.3 below.

4.3.1 Create the 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 the 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.

4.3.2 Create the 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 the 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.

4.3.3 Create the 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 the 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 a 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. The sample Timecard input file used in this scenario contains only one record. The output file will contain only one record.

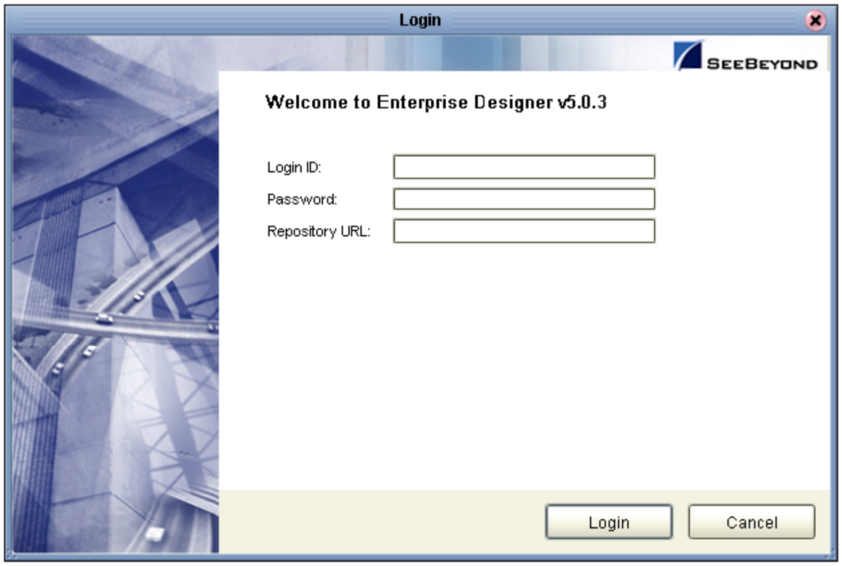
Start the Enterprise Designer

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 55 Enterprise Designer Login



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

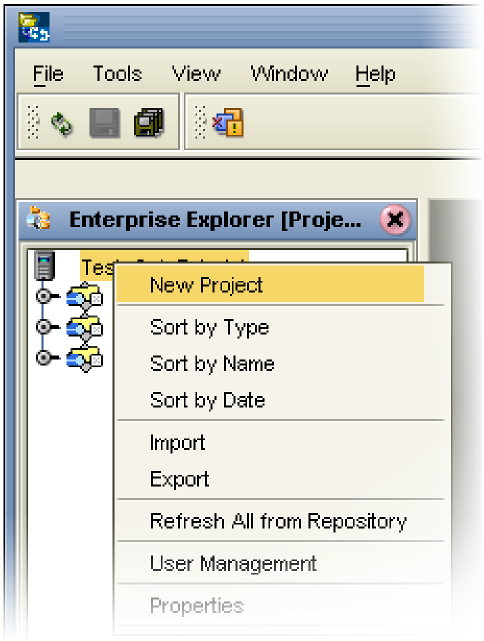
Note: *If your login window doesn't appear or if you get an invalid Username or Password error, make sure your repository is running. Also, be aware that the Username and Password are case sensitive.*

Create and name a Project

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 could be Project2; otherwise, you could name this project Project1.

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

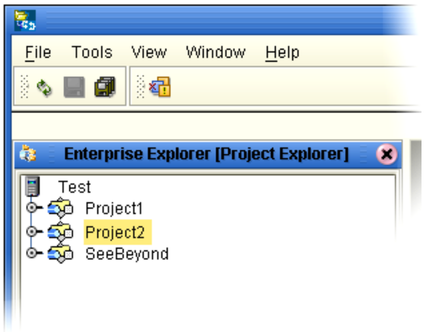
Figure 56 Create a Project



2 Accept the default **Project2** (if you have already done Project1) as the file name and press **Enter**.

3 The **Project2** structure appears in the Explorer pane on the left side of the window.

Figure 57 Project Folder: Project2



4.5 Create a Connectivity Map

Project2 needs a Connectivity Map to define the flow between the Timecard system and the Payroll system.

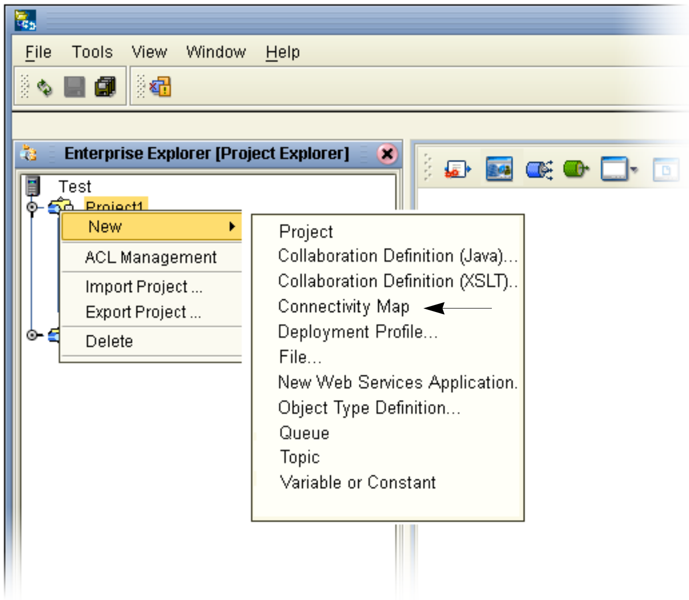
This section describes how to create the Connectivity Map for **Project2**.

Create the 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 **Project2** icon. Click **New** and **Connectivity Map**.

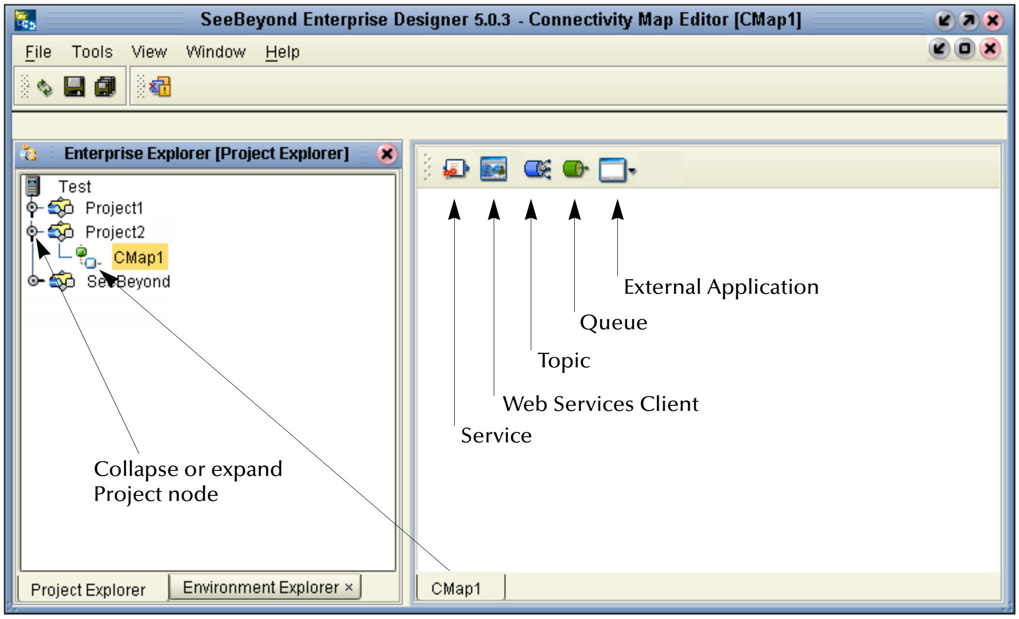
Figure 58 Create a Connectivity Map



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

3 **CMap1** is added to your **Project2**, as shown in the following figure.

Figure 59 Project2 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.*

4.6 Add Components to the Connectivity Map

In this section, you will add the components to the Connectivity Map. Later you will link the components as shown in **Figure 53 on page 55**.

4.6.1 Add Components to the Connectivity Map

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

- **InputFS** (File External Application)
- **Timecard_to_topic**
- **Topic1**
- **Topic_to_Payroll_out**
- **OutputFS** (File External Application)

Create the InputFS

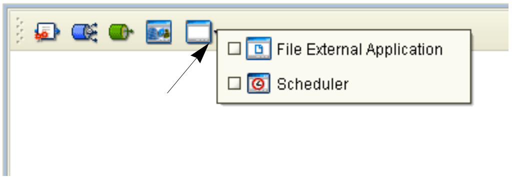
Place an input file (External Application) in your Connectivity Map.

1 On the Connectivity Map toolbar, click the External Application arrow (shown in Figure 60) to display the list. 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 60 External Application Selection



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

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

3 Type a new file name, **InputFS** (Input File System). You also have the option to rename files by Right-clicking on the label.

4 Press **Enter**.

Note: *Do not type a space between Input and FS. (Input and output are reserved words.)*

Create the Service, Timecard_to_topic

Create a Service which will eventually 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 **InputFS icon**.

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

2 Right-click the Service1 label and rename **Timecard_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 to a new location while holding down the mouse button.*

Create a Topic

The JMS topic conforms to the publish-and-subscribe messaging domain, where one publisher broadcasts messages to potentially many subscribers. In this tutorial, scenario2, the topic is not used to its full potential.

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

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

Create the Service, Topic_to_payroll_out

Create a Service which will eventually 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 a new file name, **Topic_to_payroll_out**.
3. Press **Enter**.

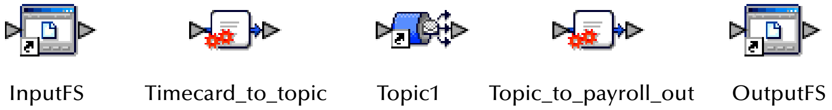
Create the OutputFS

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 **Topic_to_payroll_out** icon.

   This creates a new External Application file.
2. Right-click the **File** label and rename **OutputFS**.
3. Press **Enter**.

Figure 61 Connectivity Map



Note: *These objects will first be configured, and later they will be linked.*

4.7 Create a New Object Type Definition

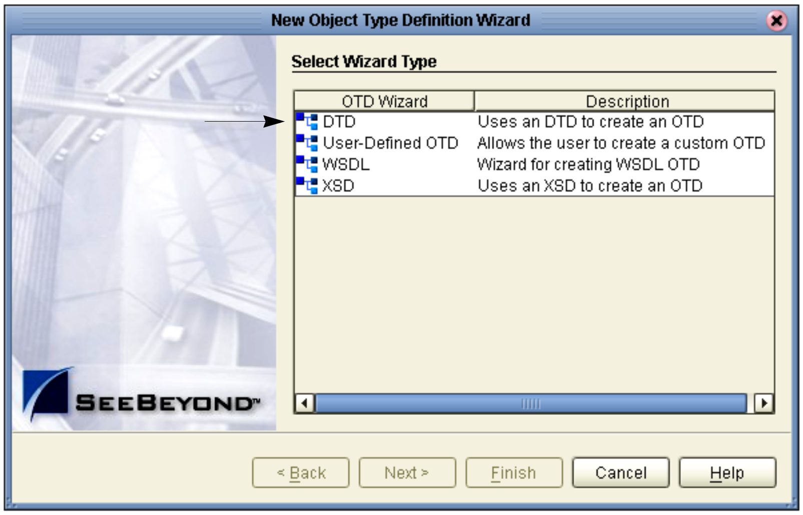
You will create an Object Type Definition before you configure your Services. OTDs represent the structure of the data and are used for message parsing; they contain send/receive methods. You will use an OTD wizard to make the OTD creation easy.

Create an Object Type Definition

First you will use DTD to create an OTD from a Document Type Definition.

1. Right-click **Project2**.
2. Click **New** and **Object Type Definition**.
3. Click the **DTD** item. (See the following Figure 62.)

Figure 62 OTD Wizard Selection



New Object Type Definition Wizard

Select Wizard Type

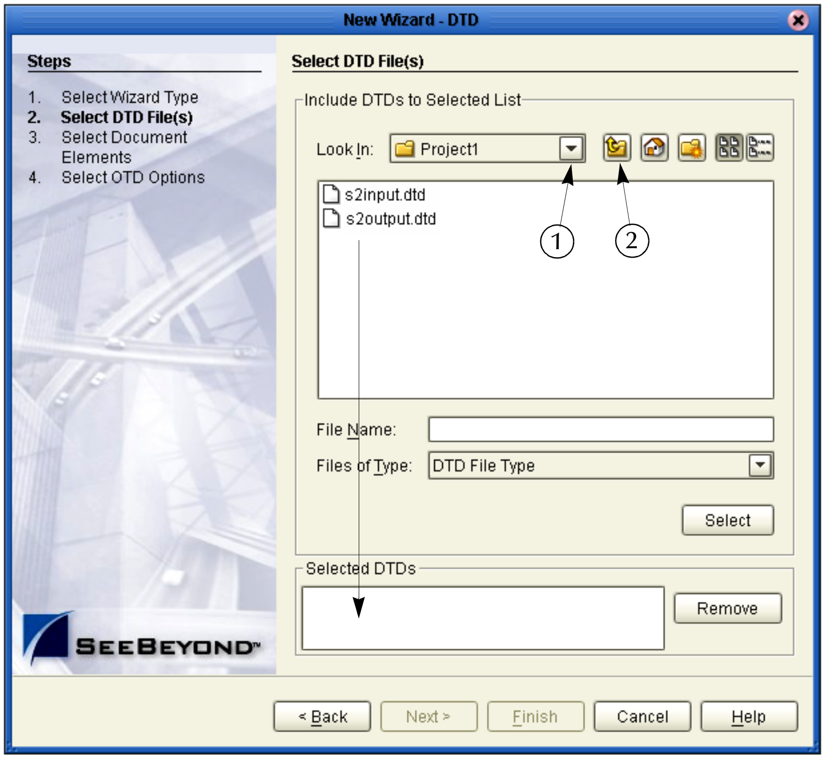
OTD Wizard	Description
DTD	Uses an DTD to create an OTD
User-Defined OTD	Allows the user to create a custom OTD
WSDL	Wizard for creating WSDL OTD
XSD	Uses an XSD to create an OTD

4 Click **Next**.

5 Click the drop-down arrow in the Look In dialog box (number 1 in the following Figure 63).

6 Navigate to the sample data folder. For example, **c:\eGateData\Project2**.

Figure 63 Select DTD Files



New Wizard - DTD

Steps

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

Select DTD File(s)

Include DTDs to Selected List

Look In: Project1

s2input.dtd
s2output.dtd

File Name:

Files of Type: DTD File Type

Selected DTDs

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

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

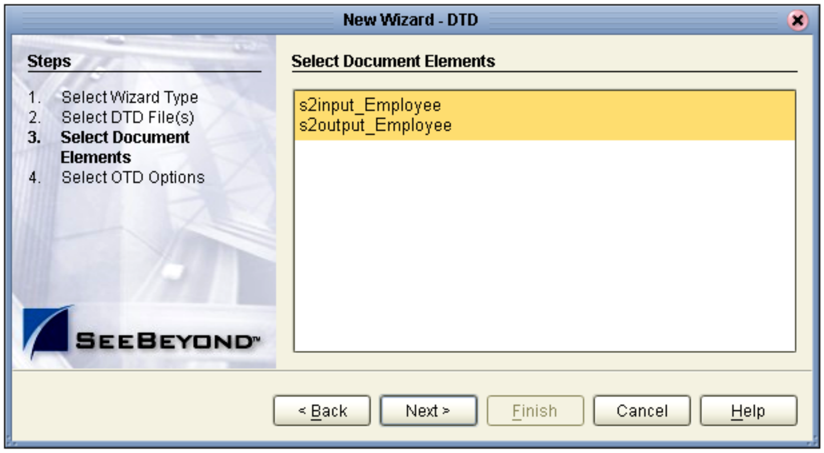
The **s2input.dtd** and **s2output.dtd** should appear in the Selected DTDs.

(You can also select both at the same time by holding down the shift key.)

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 63 above) is used to navigate to a previous screen; this is useful if you have made an incorrect selection.*

Figure 64 Select Document Elements



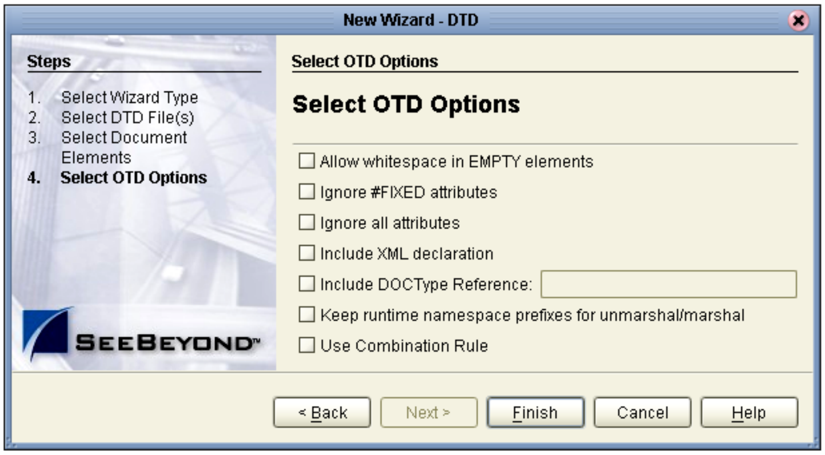
Select Document Elements

s2input_Employee
s2output_Employee

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

11 Click **Next**.

Figure 65 Select OTD Options



Select OTD Options

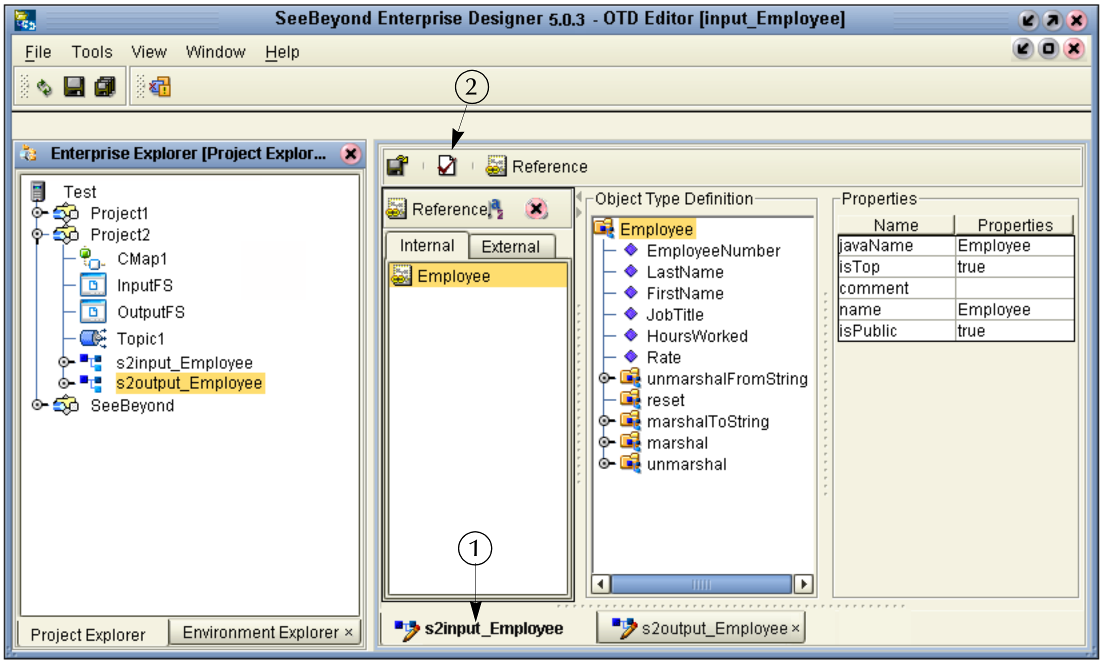
Select OTD Options

☐ 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 66.

Figure 66 OTD Editor Input



Name	Properties
javaName	Employee
isTop	true
comment	
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 the Input Data

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

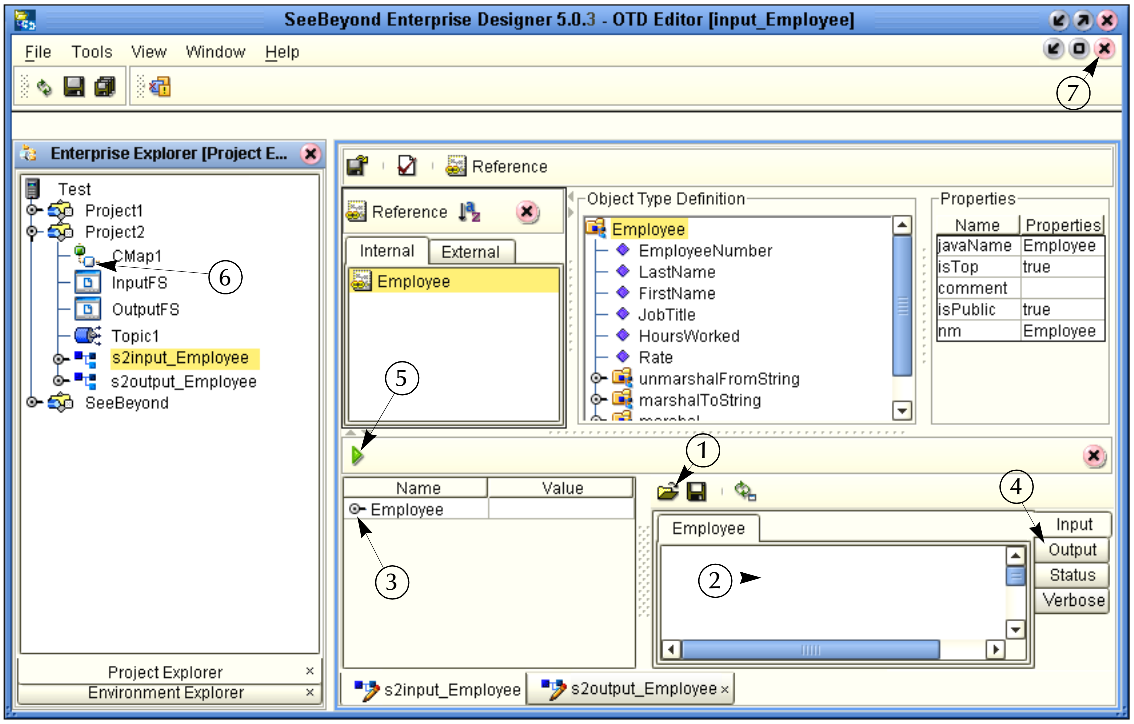
***Important:** Make sure the s2input_Employee tab is selected (number 1 in Figure 66).*

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

2 Click the **Tester** icon (number 2 in Figure 66 above).

An expanded window appears. See the following Figure 67.

Figure 67 Test Input Data File



Name	Properties
javaName	Employee
isTop	true
comment	
isPublic	true
nm	Employee

3 Click the **Open File** icon (number 1 in Figure 67 above) and navigate to your input data (**Timecard.xml**).

4 Double-click your sample data file (Timecard.xml) to display the XML data in the window. If the file is read successfully the Status “Unmarshal Successful” will appear in the Employee window (number 2 in Figure 67).

5 Click the expand view icon (number 3 in Figure 67) to expand the view of the root element **Employee**, showing the child elements (fields).

6 Click the Output button (number 4 in Figure 67). Output is empty before the test is run.

7 Click the green **Tester** icon (number 5 in Figure 67) to execute the test. If the test is successful the Status “Marshal Successful” will appear in the Employee window (number 2 in Figure 67).

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 will be populated with error messages. The message “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 **CMap1** icon (number 6 in Figure 67**)** to redisplay the Connectivity Map (or close the window, number 7 in Figure 67).

4.8 Configure the Services for Input

You will use Service wizards in the following steps to configure your Collaborations.

4.8.1 Configure Timecard_to_topic

Configure the connectivity of the input Timecard file to the topic.

Select Web Service Interface

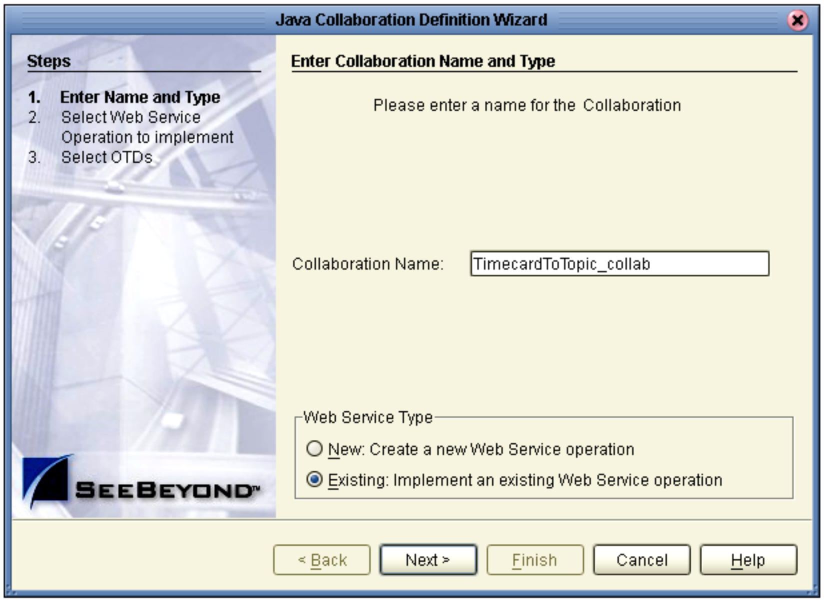
Web Services enable communication between diverse applications using the Internet. Refer to *eGate Integrator User's Guide* for information about Web Services.

1 Right-click **Project2**.

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

The Collaboration Definition Wizard (Java) appears.

Figure 68 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: TimecardToTopic_collab

Web Service Type

○ New: Create a new Web Service operation

● Existing: Implement an existing Web Service operation

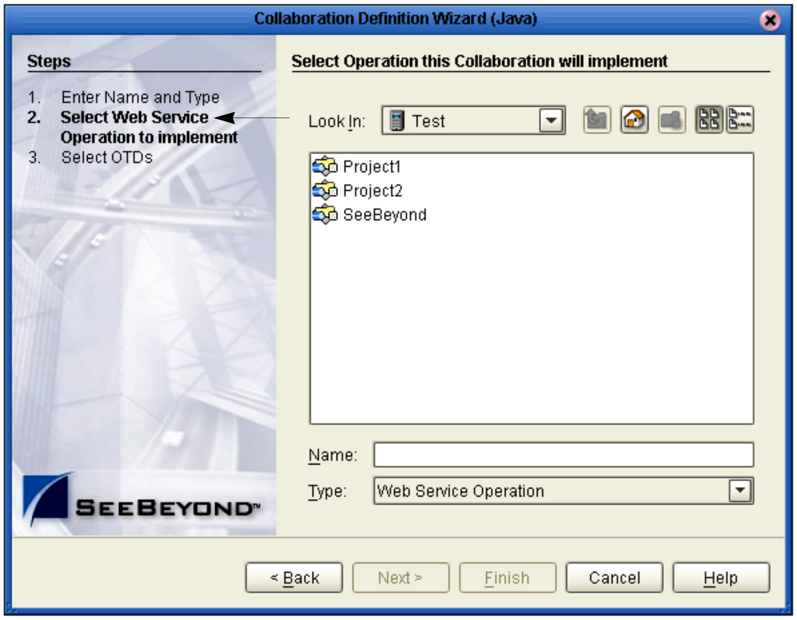
Note: *Accept the default radio button, **Existing Web Service**, for this Tutorial.*

3 Type **TimecardToTopic_collab** as the name for the Collaboration.

4 Click **Next**.

Note: *You must give the collaboration a unique name, not **Timecard_to_topic,** which will be the Service name in the Connectivity Map. **TimecardToTopic_collab** is the Java Collaboration which will appear in the Project Explorer tree. The Service and the Collaboration will be bound together later.*

Figure 69 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: Test

- Project1
- Project2
- SeeBeyond

Name:

Type: Web Service Operation

You will select a web service to “implement” a file-read Service. Then you will select an OTD.

5 Double-click **SeeBeyond**.

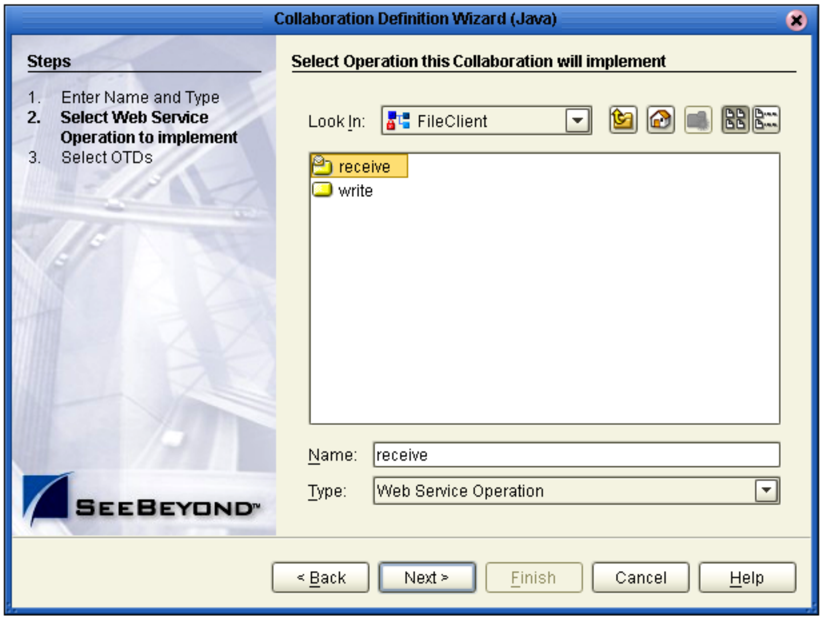
6 Double-click **eWays**. (eWays are message end points that connect to an external service provider (system, person, or organization) to enable the sending and receiving of messages.)

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 70.

Figure 70 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

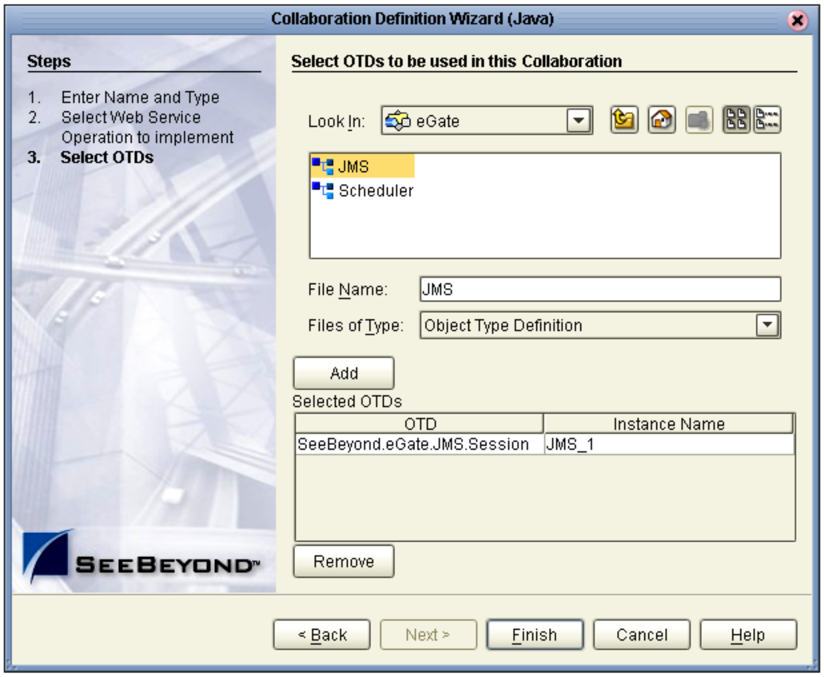
The following steps set up the OTD for the existing web Service to write to JMS. The **JMS.receive** is used to rout data between Topics (and Queues). You are still using the Collaboration Definition Wizard (Java).

1 Double-click **SeeBeyond**.

2 Double-click **eGate**.

3 Click **JMS**. Click **Add** (or double-click). **JMS** will appear in the **Selected OTDs** field as shown in the following Figure 71.

Figure 71 New Collaboration Select OTD



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
JMS
Scheduler
Add
Remove
< Back
Next >
Finish
Cancel
Help

Look In: eGate

File Name: JMS

Files of Type: Object Type Definition

Selected OTDs

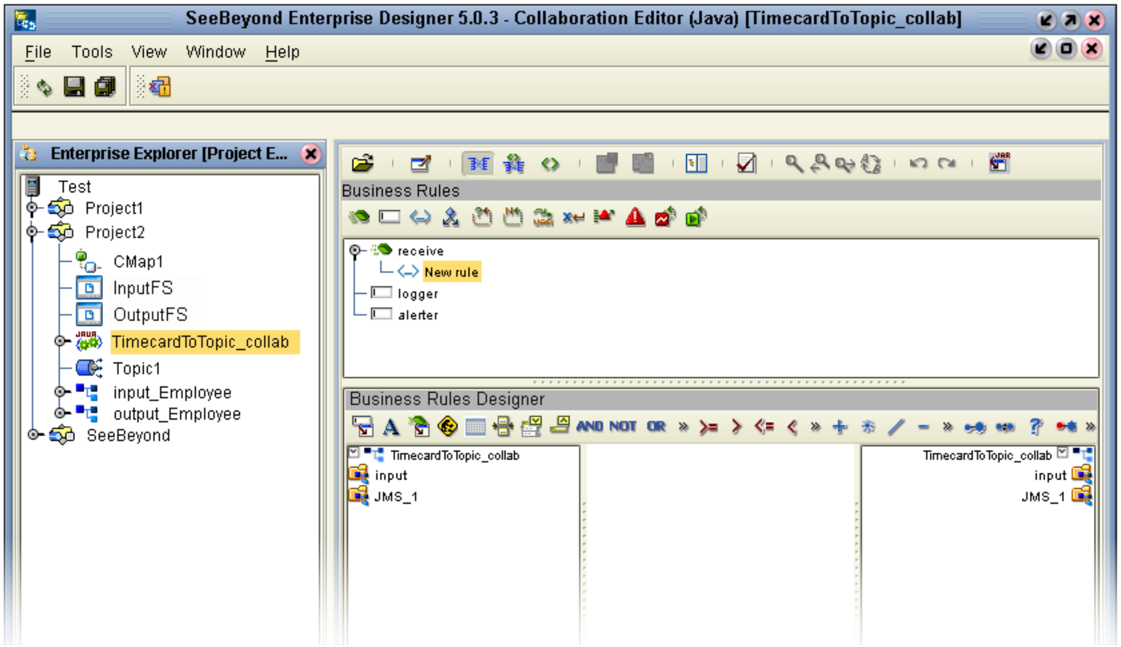
OTD	Instance Name
SeeBeyond.eGate.JMS.Session	JMS_1

4 Click **Finish**.

After a few seconds, the **TimecardToTopic_collab** icon appears in the Project Explorer tree in the left pane. This Collaboration will later be bound to the Service, **Timecard_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 72.

Figure 72 Business Rules Designer



4.9 Apply Business Rules for TimecardToTopic_collab

In **TimecardToTopic_collab** you will connect the input from the Timecard to the JMS OTD. The JMS will pass the message through to the next Service. The processing of the message will occur in the next service **TopicToPayrollOut_collab**.

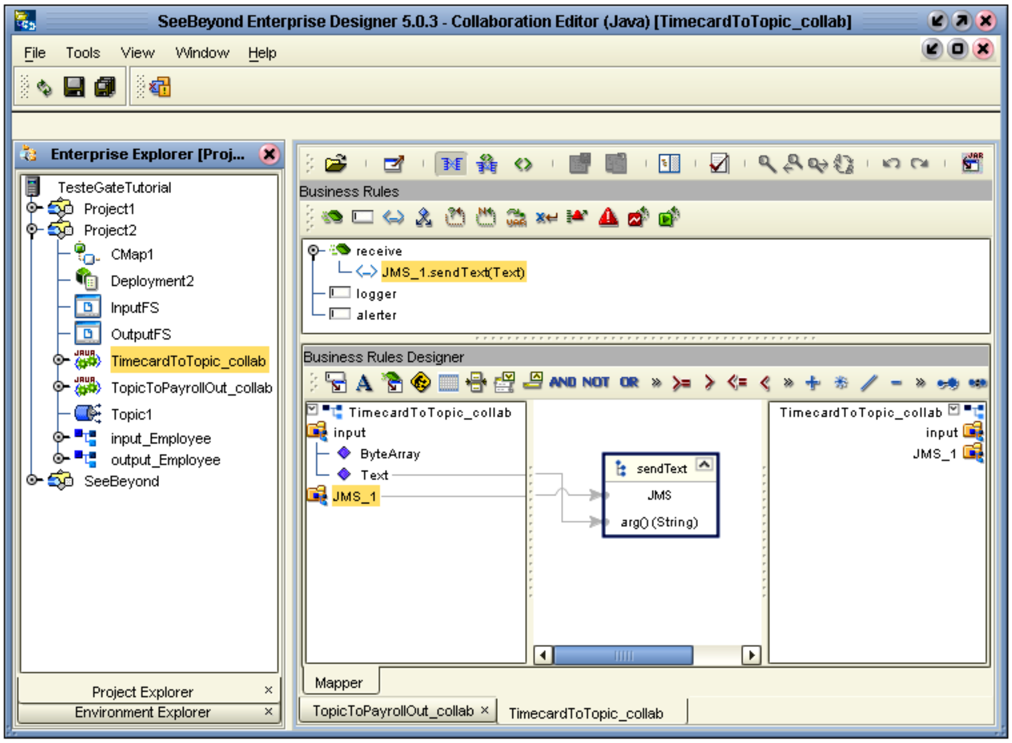
Send Input to Next Service

Select the Send method so the message can be passed on to the next service.

1. Right-click **JMS_1** in the left pane.
2. Click **Select a method to call**. Select **sendText(java.lang.string arg())**.
3. A send **sendText** box appears in the center pane.
4. Double-click **input** in the left pane.
5. Connect **Text** from **input** to **arg() (String)** in the **SendText** box.

   See the following Figure 73.

Figure 73 Send Method Box



Note: *Notice that the objects in Figure 73 reside in both the right and left pane of the Business Rules Designer window. Keep in mind that the GUIs (**JMS_1** for example) refer to the same object.*

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

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

8 Double-click **CMap1** to redisplay the Connectivity Map.

This completes the setup for the **TimecardToTopic_collab** collaboration definition. You are now ready to drag-and-drop the Service configuration into your Connectivity Map, but first you will configure **Topic_to_payroll_out**.

4.10 Configure the Services for Output

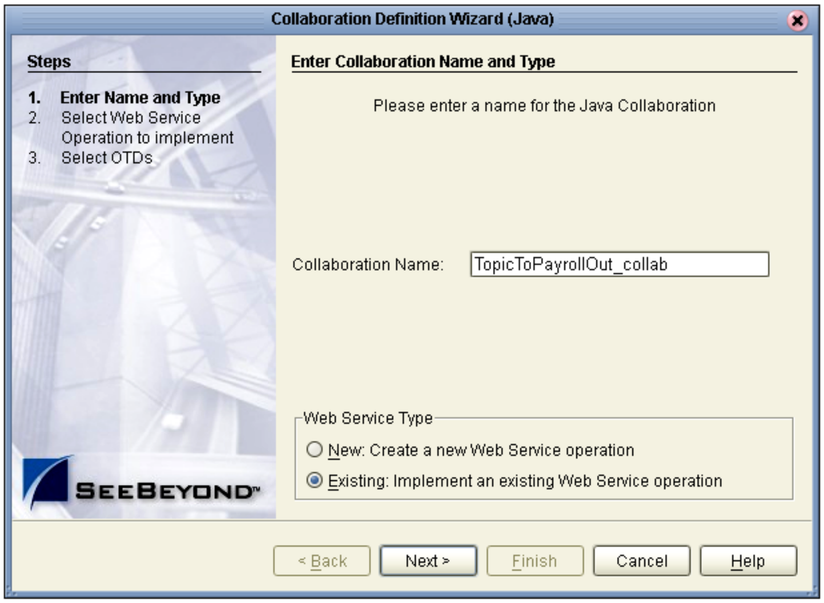
Select a Web Service Operation

First you will configure your Web Service interface.

1 Right-click **Project2**.

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

Figure 74 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

Please enter a name for the Java Collaboration

Collaboration Name: TopicToPayrollOut_collab

Web Service Type

○ New: Create a new Web Service operation

● Existing: Implement an existing Web Service operation

Note: *Accept the default radio button **Existing Web Service** for this Tutorial.*

3 Type **TopicToPayrollOut_collab** as the name for the Collaboration Definition (Java).

4 Click **Next**.

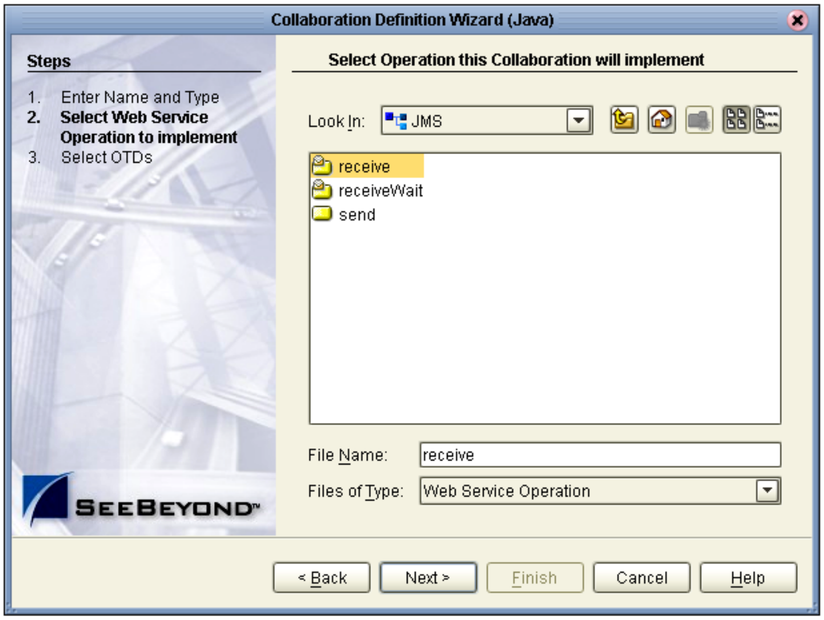
5 From the Collaboration Definition Wizard (Java), double-click **SeeBeyond**.

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 will implement.)

Figure 75 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**.

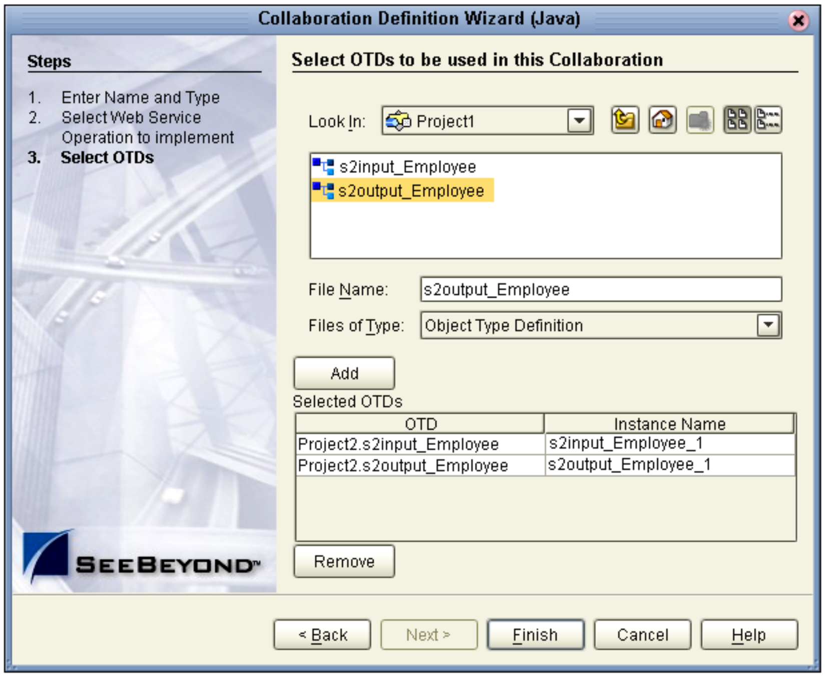
The Collaboration Definition Wizard (Java) refreshes. Now you will select input and output OTDs.

Select OTDs

1 Double-click **Project2** in the Collaboration Definition Wizard (Java).

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

Figure 76 Input/Output OTDs for TopicToPayrollOut_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: Project1

s2input_Employee
s2output_Employee

File Name: s2output_Employee

Files of Type: Object Type Definition

Selected OTDs

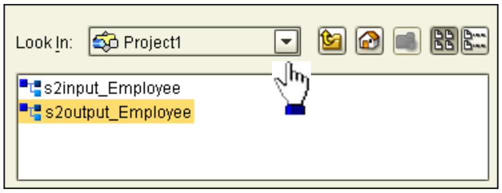
OTD	Instance Name
Project2.s2input_Employee	s2input_Employee_1
Project2.s2output_Employee	s2output_Employee_1

2 Double-click **s2input_Employee**. Double-click **s2output_Employee** to move these selections to the list of Selected OTDs.

Before clicking Finish, you must first select FileClient which represents your output to an external file.

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

Figure 77 Drop Down Arrow button



Look In: Project1

s2input_Employee
s2output_Employee

4 Click **SeeBeyond**.

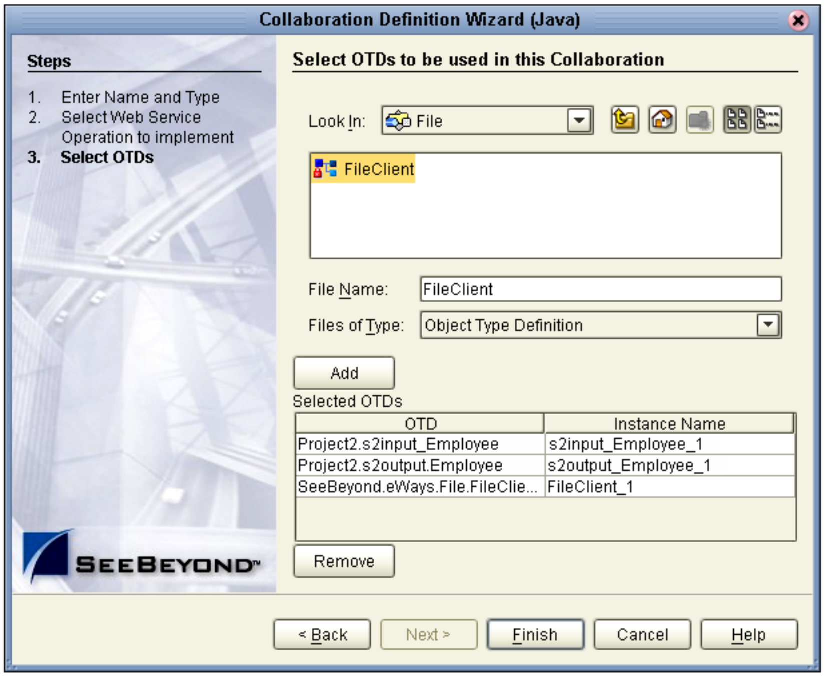
5 Double-click **eWays**.

6 Double-click **File**.

7 Double-click **FileClient**.

Notice that the **FileClient_1** has been added to the list of **Selected OTDs**. See the following Figure 78.

Figure 78 OTDs for TopicToPayrollOut_collab



Select OTDs to be used in this Collaboration

Look In: File

FileClient

File Name: FileClient

Files of Type: Object Type Definition

Selected OTDs

OTD	Instance Name
Project2.s2input_Employee	s2input_Employee_1
Project2.s2output.Employee	s2output_Employee_1
SeeBeyond.eWays.File.FileClie...	FileClient_1

8 Click **Finish**.

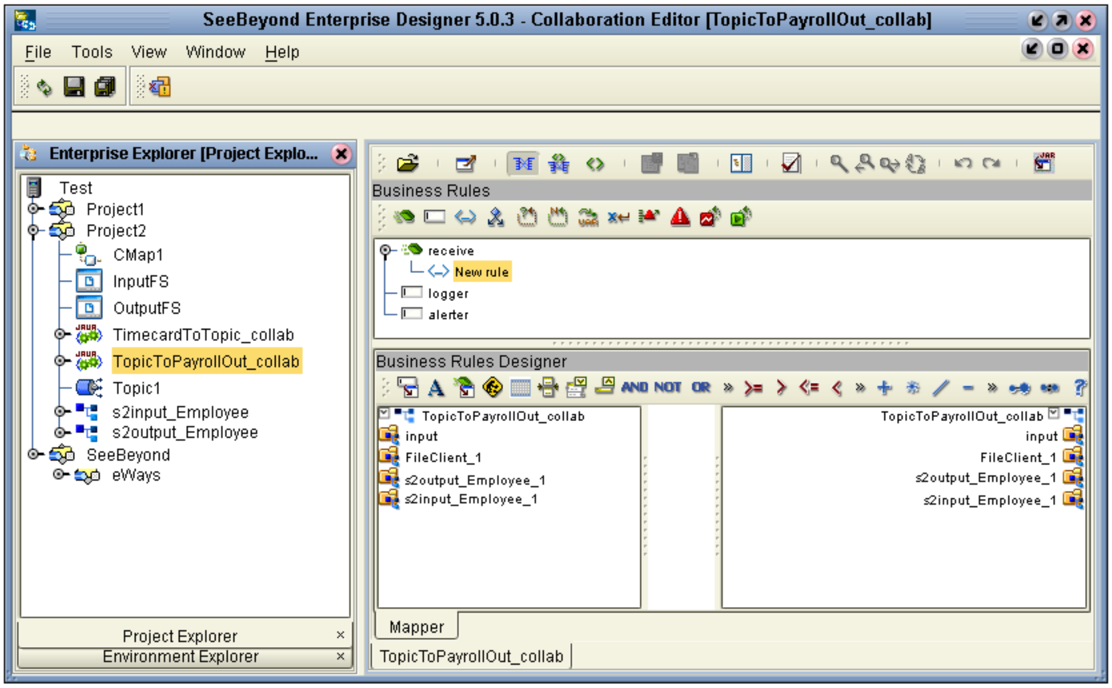
After a few seconds, the **TopicToPayrollOut_collab** appears in the Enterprise Explorer tree in the left pane. The Business Rules Designer window also appears, with the Business Rules window above. (See the following Figure 79.)

4.11 Apply Business Rules for TopicToPayrollOut_collab

In the following section you will receive a message from **JMS_1** and unmarshal the string into OTD **s2input_Employee_1**. You will then copy data from the nodes, **EmployeeNumber**, **LastName** and **FirstName** to the corresponding nodes in **s2output_Employee_1** OTD. Then you will set up the multiplication logic (business rules) for execution, to obtain an amount (**Amt**) for **s2output_Employee_1**, requiring the conversion from string to integer. Finally, you will marshal the data from **s2output_Employee_1** to a string and write it to a file.

See the following Figure 79.

Figure 79 Business Rules Designer for TopicToPayrollOut_collab



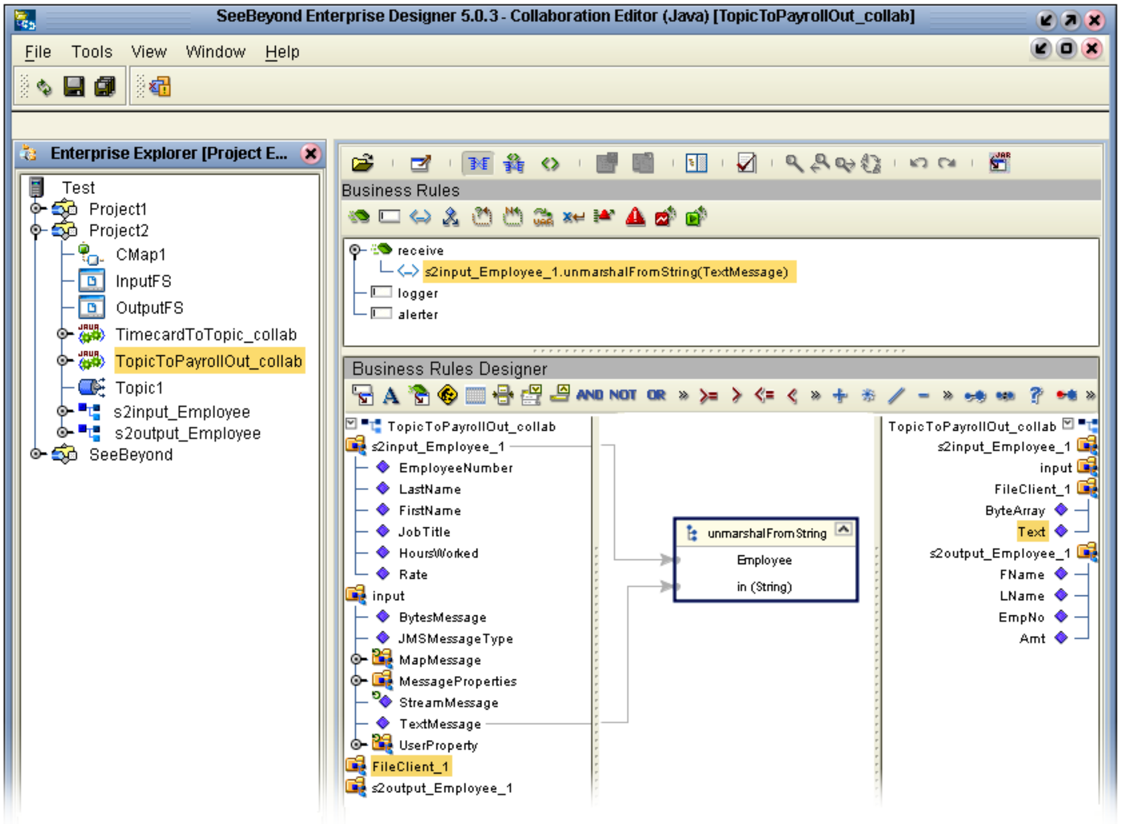
Unmarshal Text

You must first unmarshal the text input from the topic into the input OTD.

1. Double-click **input** in the left pane of the Business Rules Designer to expand the view.
2. Double-click **s2input_Employee_1** in the left pane of the Business Rules Designer to expand the view.
3. Right-click **s2input_Employee_1** in the left pane of the Business Rules Designer.
4. Click **Select a method to call**.
5. Click the method **unmarshalFromString(java.lang.String in)**.

   This action places a method box into the center pane of the Business Rules Designer window.
6. Drag **TextMessage** from **input** into **in (String)**. See the following Figure 80.

Figure 80 Collaboration Editor (Java) - Unmarshal From String



Connect Input and Output OTD Nodes

Make sure the views of **s2input_Employee_1** in the left pane and **s2output_Employee_1** in the right pane of the Business Rules Designer are expanded.

1. Double-click **s2output_Employee_1** in the right pane to expand the view.
2. Connect the nodes between the **s2input_Employee_1** OTD in the left pane with the **s2output_Employee_1** OTD in the right pane of the Business Rules Designer:

Note: *When you begin to connect the nodes, the previously displayed rule will disappear but the record of your action appears in the Business Rules pane which is just above the Business Rules Designer pane.*

Note: *You can delete a rule, method box or other GUI objects by right-clicking on the object and selecting Delete, or by selecting the item and pressing the Delete key.*

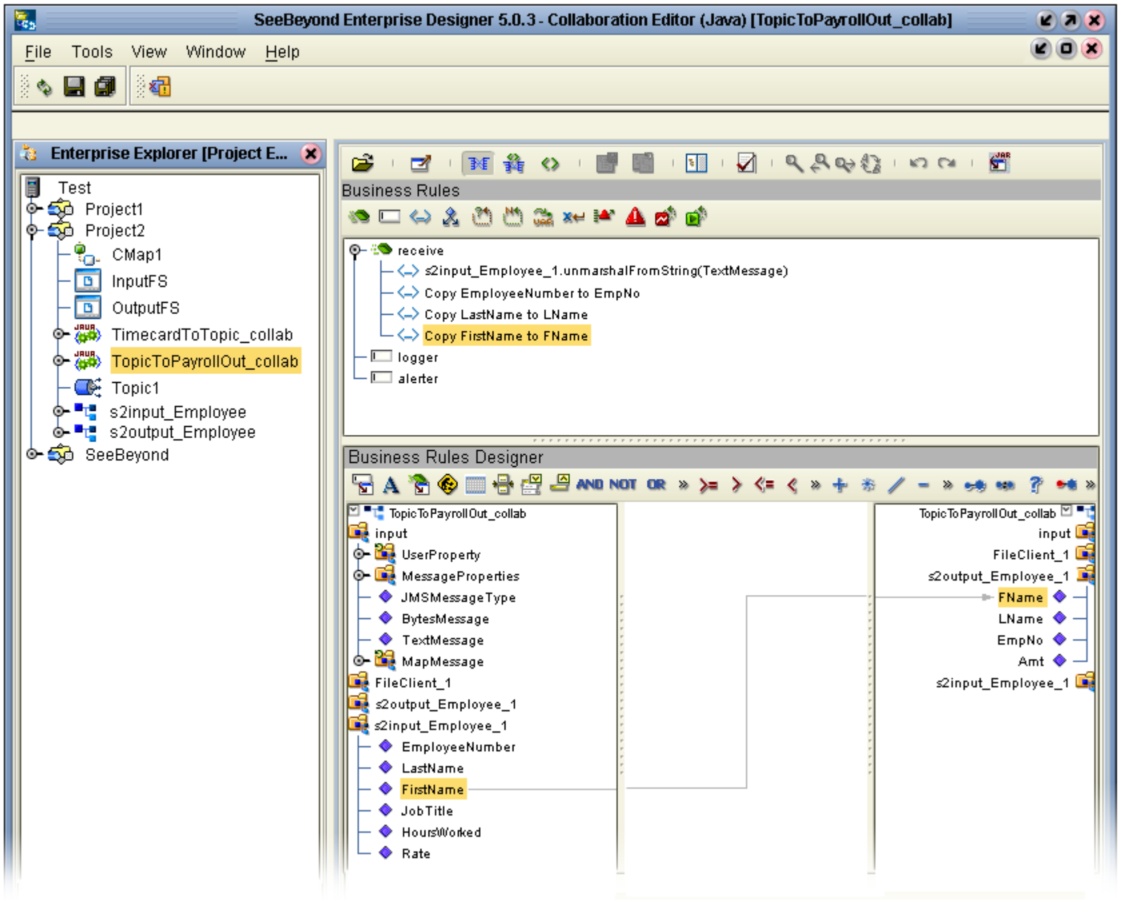
Connect the following nodes:

Connect **EmployeeNumber** to **EmpNo**.

Connect **LastName** to **LName**.

Connect **FirstName** to **FName**.

Figure 81 Connect Input/Output OTD Nodes



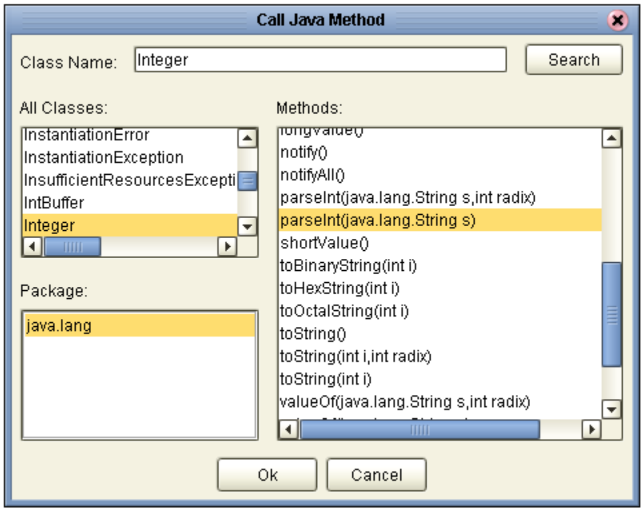
Prepare for the Multiplication Operation

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

1. Click the **Call Java Method** icon to display a method box.

   The method box will appear in your center pane. See the following Figure 82.

Figure 82 Integer Java Method



Class Name: Integer

2 Type **Integer** (be sure to use capital I) into the Class Name box. Click **Search**.

3 Scroll down. Click **parseInt(java.lang.String s)** from the methods list. Click **OK**.

4 Repeat the previous steps because you will need two method boxes.

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 84.

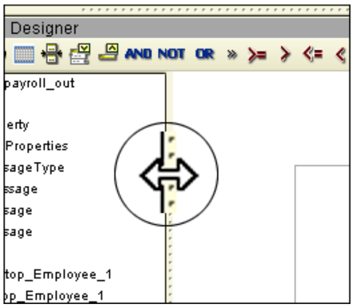
Place these two boxes on the left side of the workspace in the center pane.

5 Click the Multiplication Operator ✳ on the toolbar and drag the method box to the right of the two parsing boxes.

The Multiplication Method box appears.

Note: *You can adjust the size of the window panes by clicking on the frame and adjusting. Refer to the following Figure 83.*

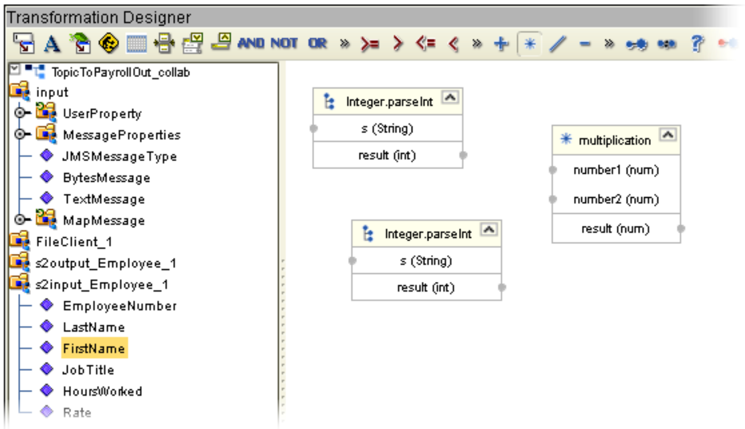
Figure 83 Window Adjust



Use Methods to Prepare the Data for String to Integer Conversion

Convert int primitive to Integer Object.

Figure 84 Prepare for Multiplication



1 Click the **Call New Constructor** icon.

2 Type **Integer** (be sure to use capital I) and press **Enter**.

3 Click **java.lang.Integer(int value)** in the list. Click **OK**.

A new method box appears. Place this method box under, or to the right of the Multiplication Method box. This method is necessary to convert the int primitive to an Integer Object.

4 Click the **Call Java Method** icon once more to display a method box.

5 Type **Integer** (if not already displayed). Click **toString()** in the list.

6 Click **OK**. (Place this method to the right and slightly above the **java.lang.Integer**.)

This method will convert the Integer Object to a String.

Connect the Nodes

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.

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

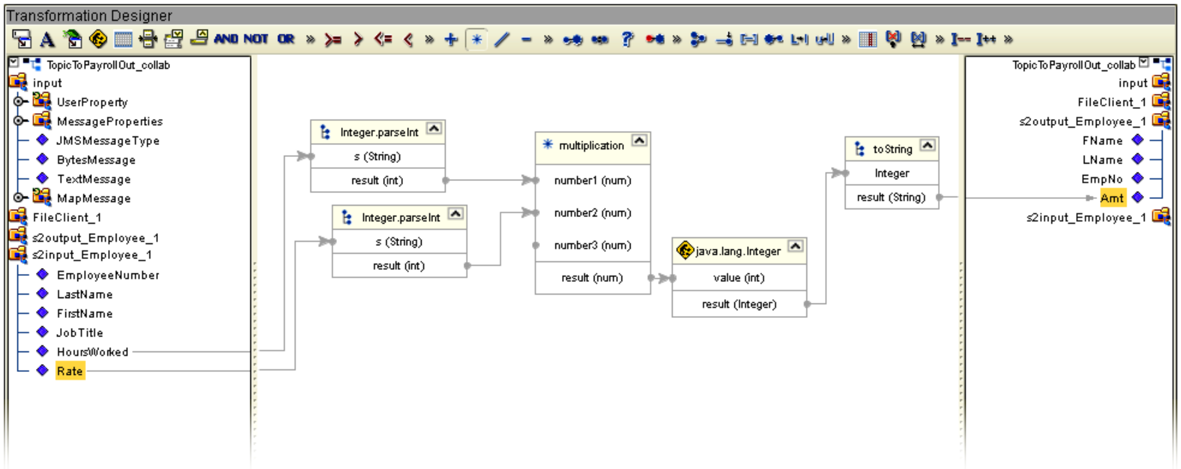
6 Connect the **result(integer)** to **Integer** in the **toString** method box.

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 multiplication. See the following Figure 85.

Figure 85 Multiplication Operation



Marshal Text

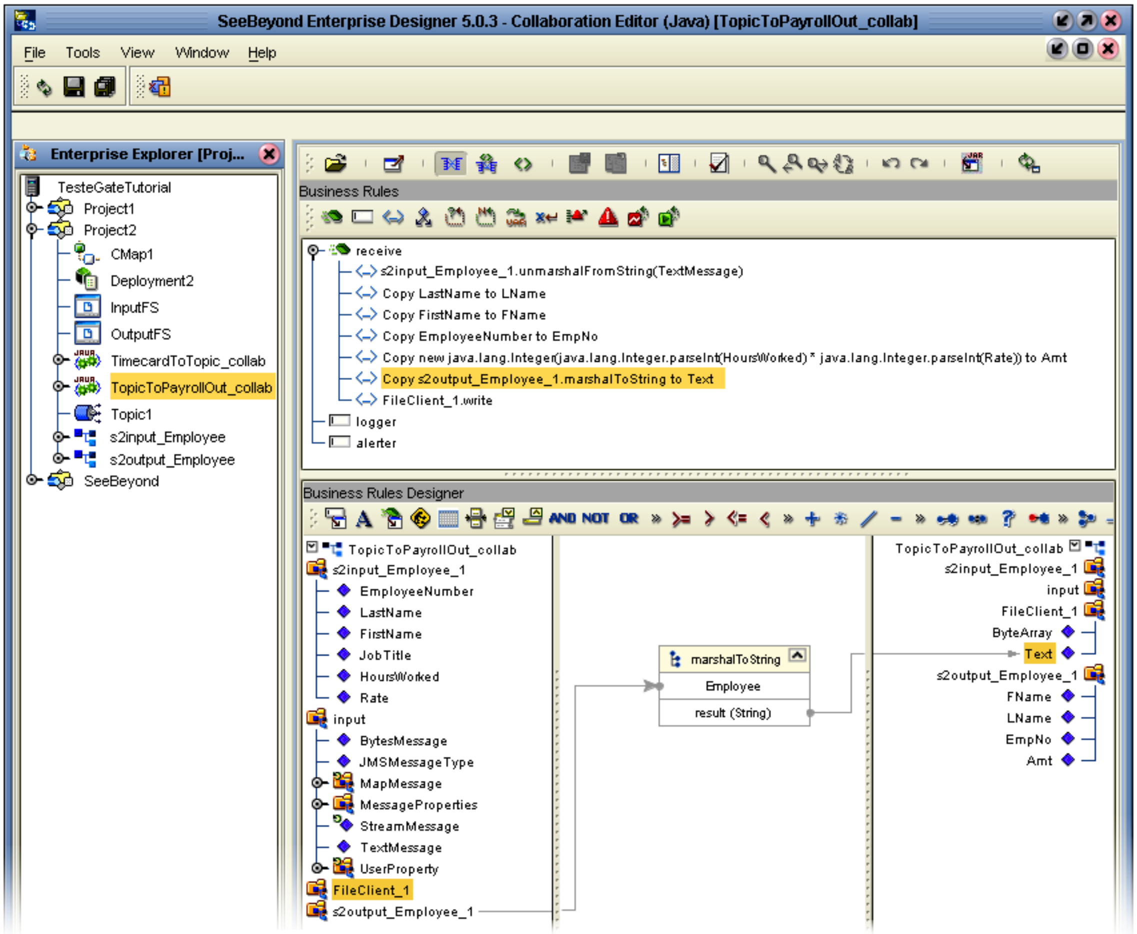
Since you began the data manipulation process by unmarshalling data from string, you must now marshal the data from output into a string.

Make sure the view of **FileClient_1** in the right pane is expanded.

1. Right-click **s2output_Employee_1** in the left pane. Click **Select a method to call**.
2. Click **marshalToString()**.

   The **marshalToString** method box appears.
3. Connect a node from **result (String)** to Text under **FileClient_1** in the right pane. See the following Figure 86.

Figure 86 Marshal Text



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

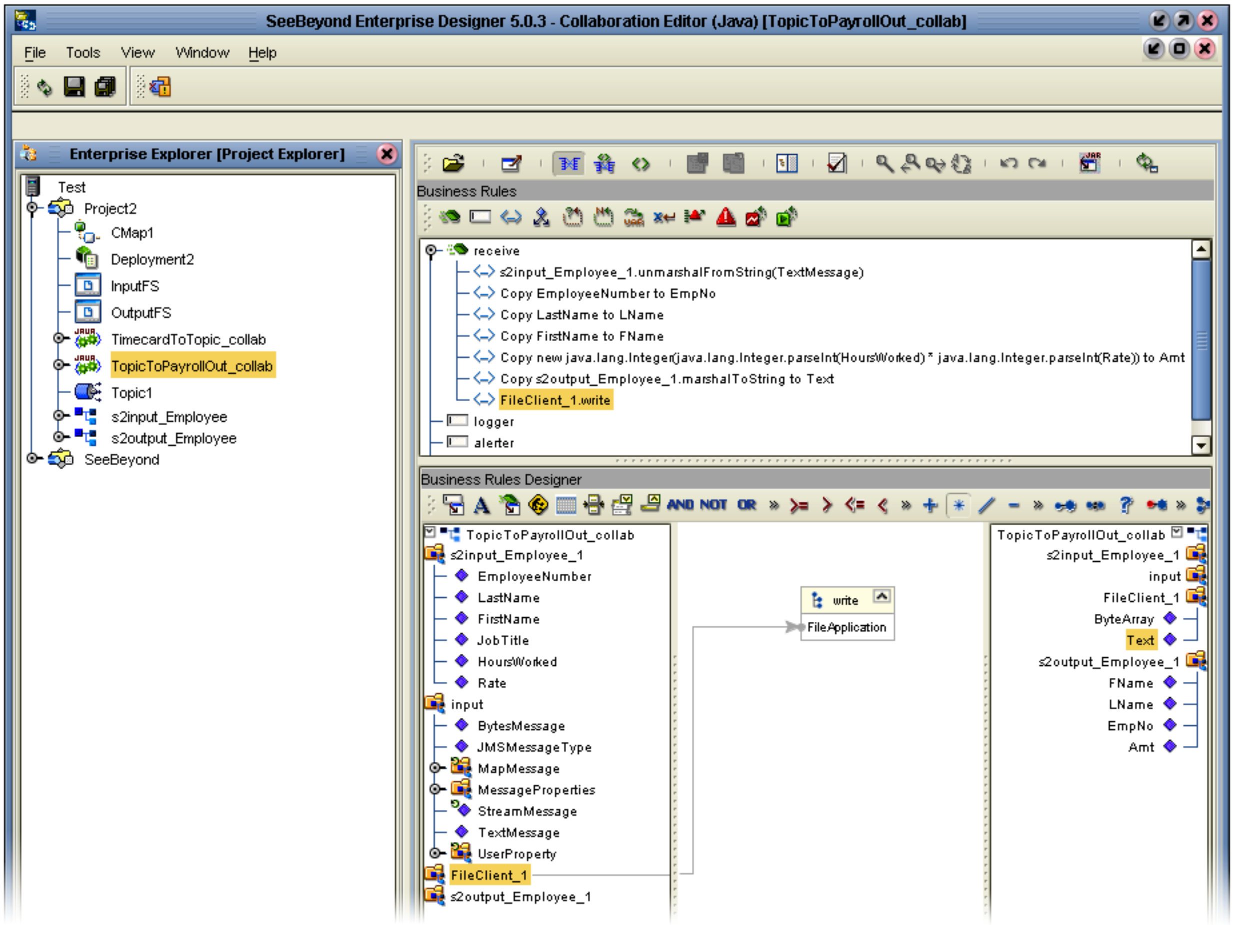
Write FileClient_1

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

1. Right-click **FileClient_1** in the left pane. Click **Select a method to call...**
2. Click **write ()**.

   See the following Figure 87.

Figure 87 Write File Application



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

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

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

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

You will use the drag-and-drop method to move the configurations into your Services on the Connectivity map.

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, **CMap1**.

4.12 Apply the 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.

Drag-and-drop TimecardToTopic_collab

Refer to Figure 89 for the following steps.

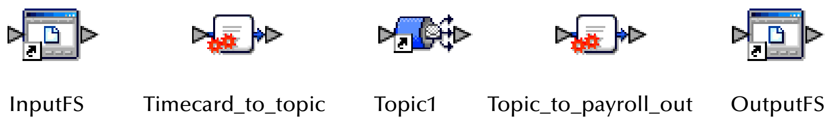
- Double-click your Connectivity Map icon (if the Connectivity Map is not already displayed) to return to the Connectivity Map.

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.

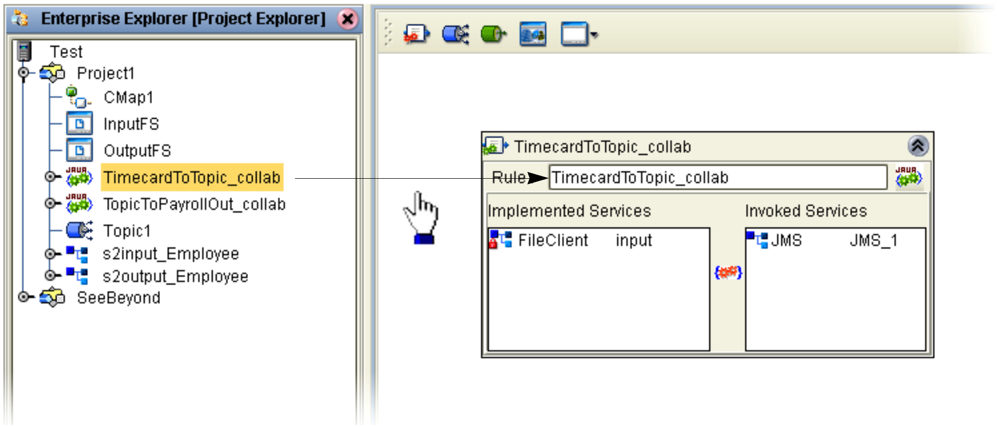
At this point, the Connectivity Map contains five unlinked objects. You are now ready to link them.

Figure 88 Unlinked Objects in Connectivity Map



1. Double-click **Timecard_to_topic** in the Connectivity Map to open the Service Window.
2. Drag the **TimecardToTopic_collab** Collaboration Definition from the Project Explorer pane to the **Rule** field in the open window as shown in the following Figure 89.

Figure 89 Using the Drag and Drop Method

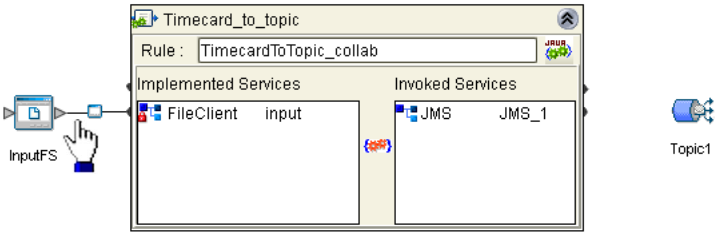


Notice that the gear icons change from red to green to signify that the Collaboration is complete.

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

3 Connect the **InputFS** to **FileClient**.

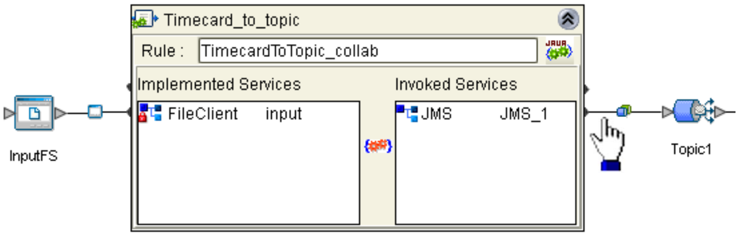
Figure 90 Connect Input to Client



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 where there are multiple Services.

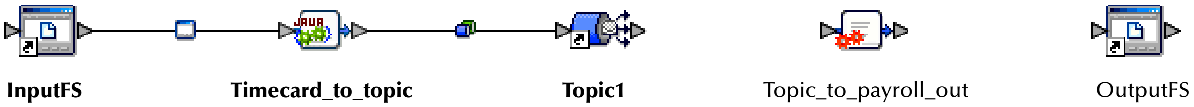
4 Connect **JMS_1** to **Topic1**. (Reposition binding box as necessary.)

Figure 91 Connect JMS to Topic



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

Figure 92 Timecard_to_topic Linked



Drag-and-drop TopicToPayrollOut_collab

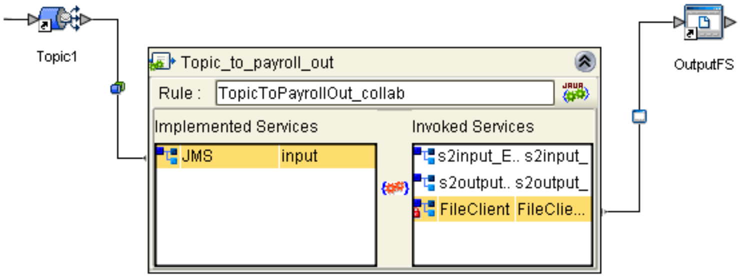
1 Drag the **TopicToPayrollOut_collab** Collaboration Definition from the Project Explorer pane into the **Topic_to_payroll_out** Service icon.

2 Double-click the **Topic_to_payroll_out** Service icon to open the binding box.

3 Connect **Topic1** to **JMS**.

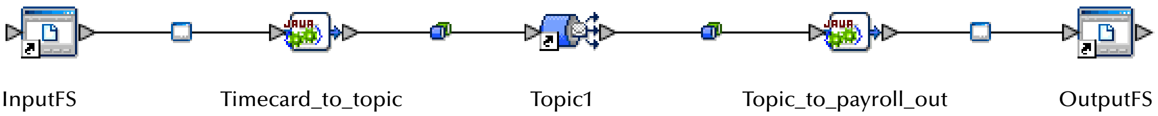
4 Connect **FileClient_1** to **OutputFS**. (See the following Figure 93.)

Figure 93 Connect Topic to Output



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

Figure 94 Linked Connectivity Map



4.13 Configure the eWays and JMS Connections

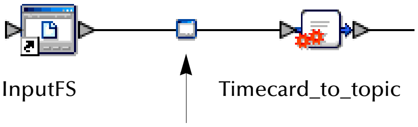
Your Connectivity Map contains two eWays and JMS connections. These components must be configured for deployment. Each of these components performs a different function:

- **Input eWay**: The input eWay polls a directory on the local file system, **c:\eGateData\Project2**, for any XML file (***.xml**).
- **Timecard_to_topic**: This Collaboration Service copies the input XML data without making any changes to it.
- **Topic_to_payroll_out**: This Collaboration Service performs the calculation and data mapping function.
- **Output eWay**: The Output eWay writes the results to an output file (**.dat**) on the local file system, **c:\eGateData\Project2**.

Configure the Inbound eWay

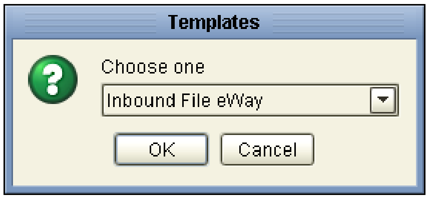
Choose inbound, and identify the inbound file and location.

Figure 95 Inbound eWay



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

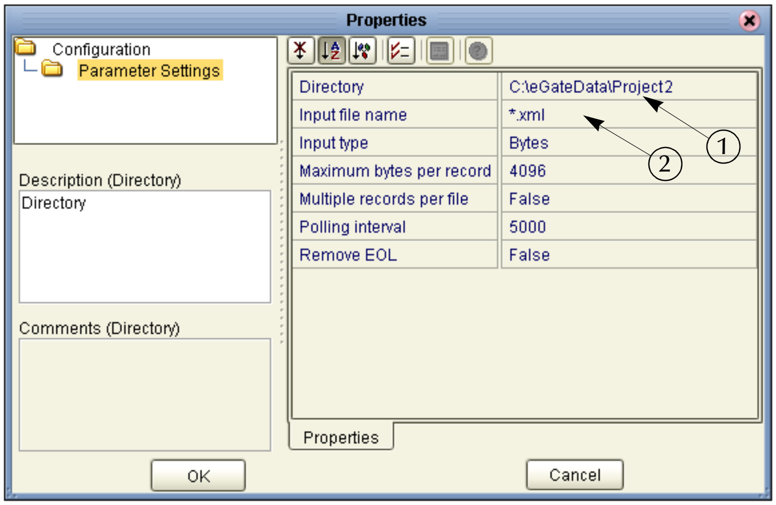
Figure 96 External Type Dialog Box



Choose one
Inbound File eWay

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

Figure 97 Properties Configuration (Inbound eWay)



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 97. These are the fields you will edit.

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 97 above).

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 will read any files with the .xml extension (number 2 in Figure 97 above).

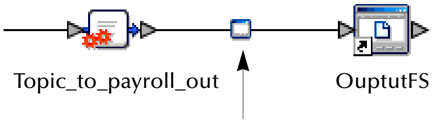
4 Click **OK** to close the Properties Dialog Box for the inbound eWay.

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

Configure the Outbound eWay

Choose outbound, and identify the outbound file and location.

Figure 98 Outbound eWay



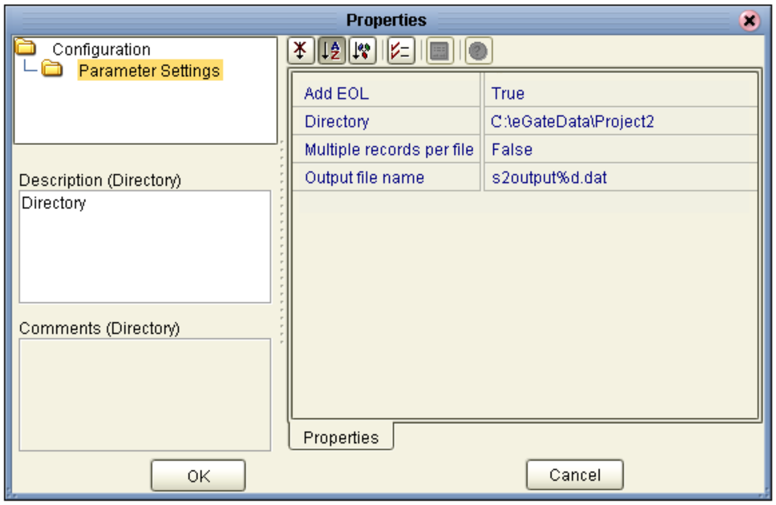
1 Double-click the second eWay. Select **Outbound File eWay**. Click **OK**.

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 99 Properties Configuration (Outbound eWay)



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

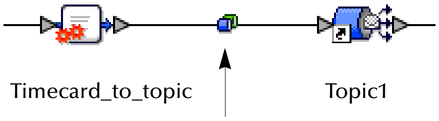
Note: *The **Multiple records per file** setting determines whether records will be concatenated in the output file. A True setting (default) means that multiple records will be written in the same file. A False setting, as shown in Figure 99 above, means that only one record will be written to one file. See* **"Multiple records per file (True or False)" on page 43** *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 100 JMS Client Properties Icon



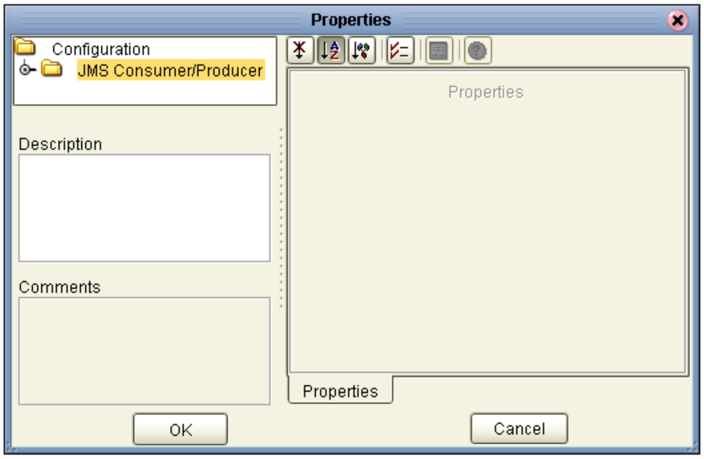
Select JMS Client Configuration

For this tutorial accept the default configurations.

1 Double-click the first of the two JMS icons ⊸ (see previous Figure 100).

The Message Server properties dialog box appears.

Figure 101 Message Server Configuration



2 Click **OK** to accept the defaults.

3 Repeat this step for the other Message Server connection.

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 an Environment and Activate the Deployment Profile

In this section, you will create and activate the Deployment Profile. Before you can create the profile, you must create an Environment.

Create an 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**.

Note: *If you have already created an Environment called Tutorial, you may use that Environment and logicalhost. Continue with* **"Create an External File System for the Inbound eWay" on page 94***. 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 an 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 will appear in the **LogicalHost1** window.

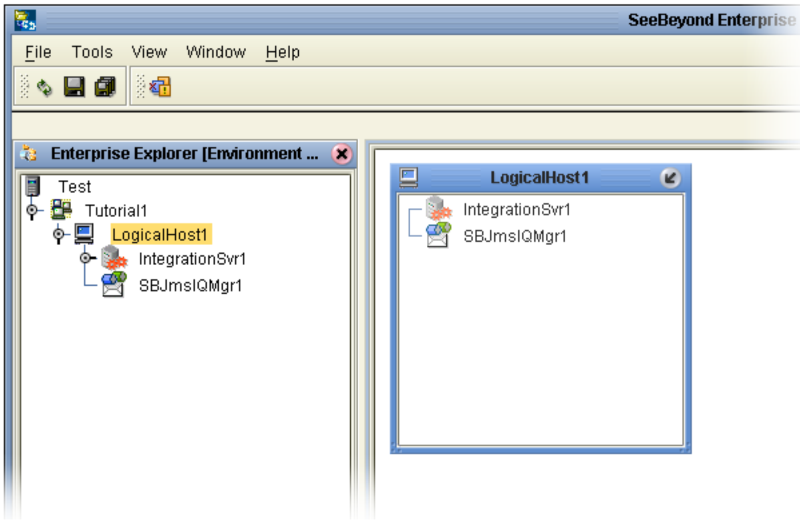
Create a JMS IQ Manager

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 will appear in the **LogicalHost1** window.

Figure 102 Create Logical Host Servers

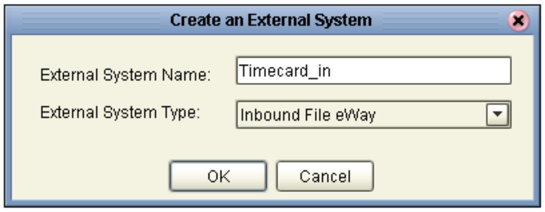


These servers automatically appear in the **LogicalHost1** dialog box.

Create an External File System for the Inbound eWay

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

Figure 103 External File Inbound



External System Name: Timecard_in

External System Type: Inbound File eWay

2 Type **Timecard_in** in the name field.

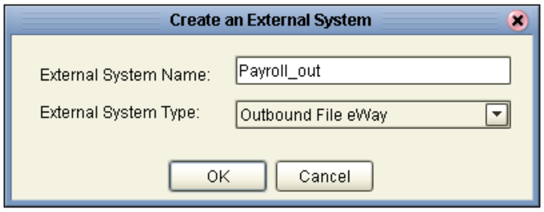
3 Select **Inbound File eWay**. Click **OK**.

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

Create an External File System for the Outbound eWay

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

Figure 104 External File Outbound



External System Name: Payroll_out

External System Type: Outbound File eWay

2 Type **Payroll_out** in the name field.

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

This places the outbound external system, **Payroll_out** 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 a New Deployment Profile

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

Note: *If you already created Deployment1 for Scenario1, you need a different Deployment profile for Scenario2.*

Deployment Profile

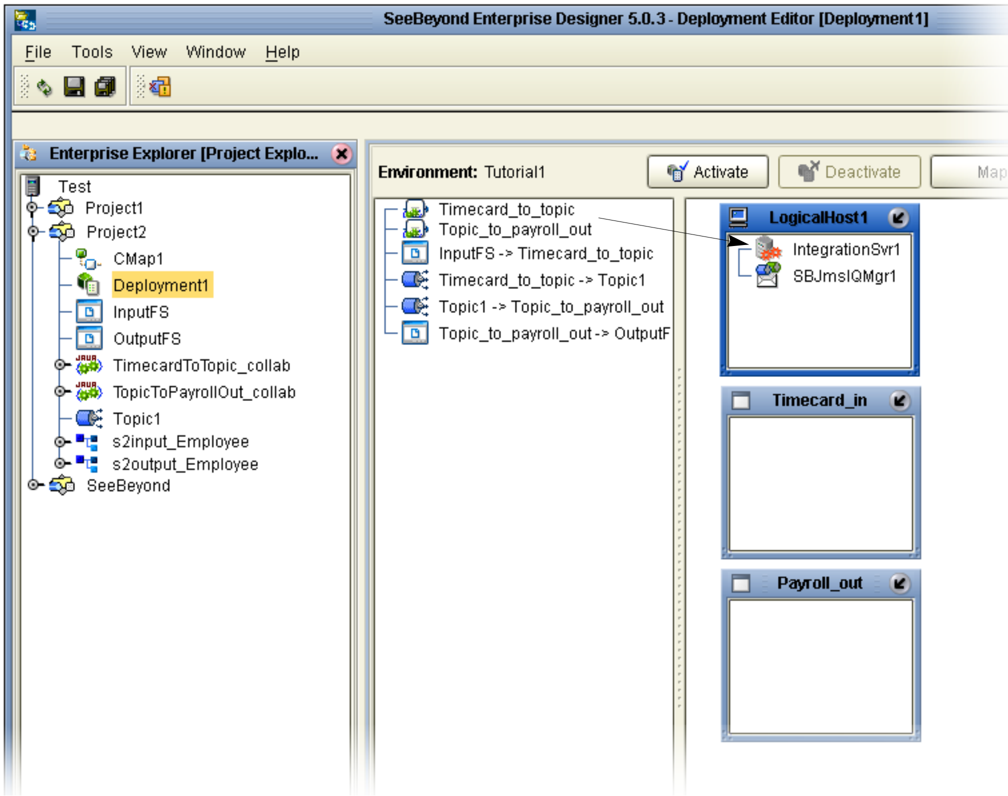
A Deployment Profile contains information about how Project components will be deployed in an Environment, and it also maps components to the Environment.

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

2 Right-click **Project2**. Click **New** and **Deployment Profile**.

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

Figure 105 Environment, Drag and Drop Components



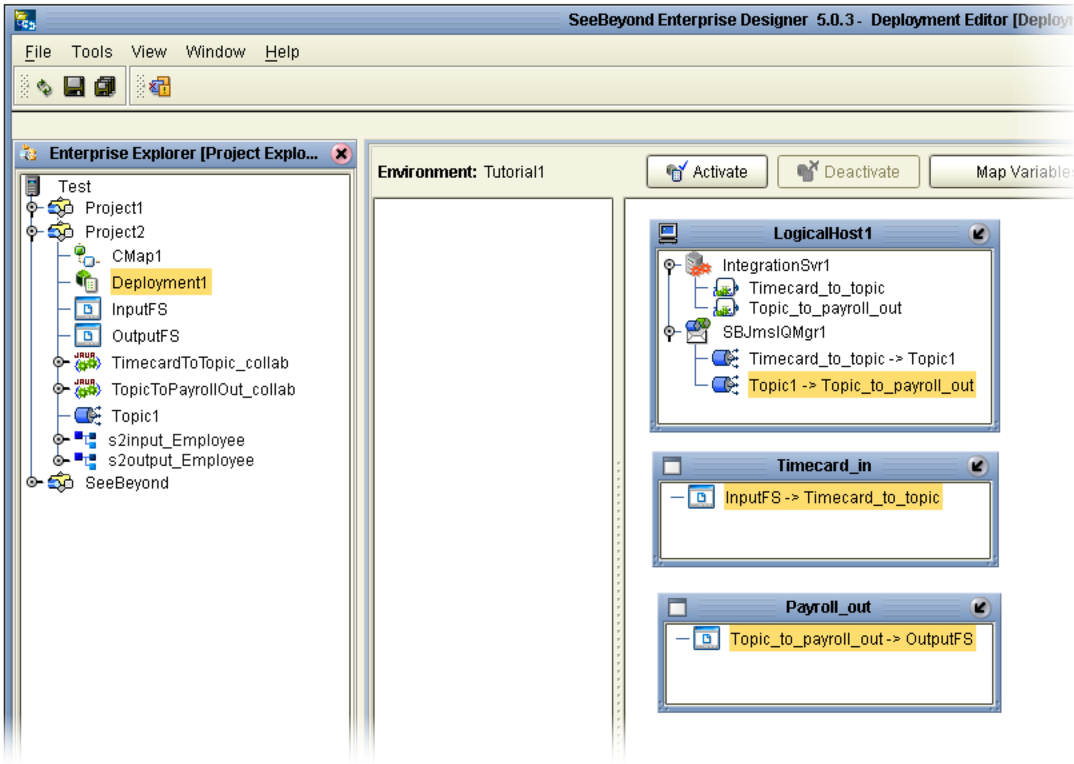
After creating your deployment profile, drag and drop components into the server dialog boxes.

Drag and Drop Collaborations

1. Drag-and-drop your Services, **Timecard_to_topic** and **Topic_to_payroll_out**, into your Integration Server, **IntegrationSvr1**. (See the previous Figure 105.)
2. Drag-and-drop your Topics into your Message Server, **SBJmsIQMgr1**.
3. Drag-and-drop your external files into your External File Servers, as shown in the following Figure 106.

Note: *If you are using the same Environment you used for Scenario1 (Tutorial), be sure to place the External files in the correct deployment boxes. For Scenario2 the InputFS goes into Timecard_in and OutputFS goes into Payroll_out.*

Figure 106 Populated Environment



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 Figure 106 above.

4.15 Run your Project

When you deploy **Project2**, a deployment file will be created that will later be picked up by the Logical Host. The Project you created with Enterprise Designer will be “code generated” into a package that will be passed to the Repository. The Logical Host will pick up this package from the Repository.

If you want to do a “hot deploy,” which means that the Logical Host will pick up the current package in real time while it is running, you could click **YES** in the Activate Dialog box when prompted “Do you wish to apply Logical host(s) immediately?” (See below).

Activate Environment

1. On the **File** menu, click **Save All**.
2. Click the **Activate** button. The Activation in Progress message appears.

Figure 107 Activation in Progress Message

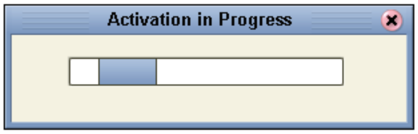
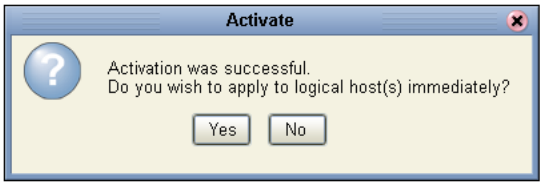


Figure 108 Successful Activation

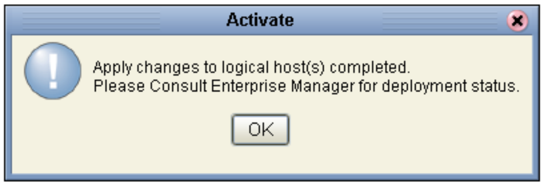


Activation was successful.
Do you wish to apply to logical host(s) immediately?

Activating the Deployment Profile may take a few minutes. When the activation is complete, the message above appears.

3 Click **No** to Finish. (If your bootstrap is already running, click **Yes**.)

Figure 109 Changes Applied to Logical Host



Apply changes to logical host(s) completed.
Please Consult Enterprise Manager for deployment status.

When you click **Yes** in step 3 above, this message verifies that changes were applied to the Logical Host.

4 Click **OK**.

4.16 Run the Bootstrap

Refer to **"Run the Bootstrap and Management Agent" on page 48** in Scenario1.

Note: *You can use the same Environment (Tutorial) and same logicalhost that you used in Scenario1 for Scenario2, but the external file names must be different and the file names in each project must be different. Also, the deployment profile (name) must be different.*

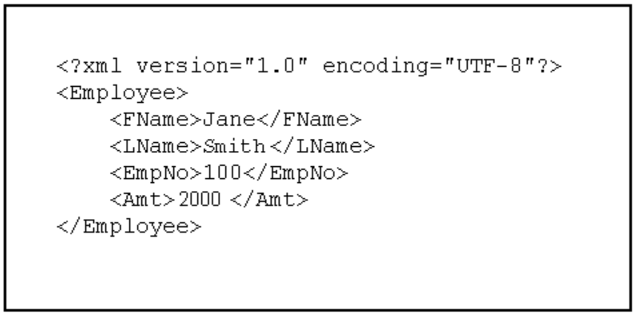
4.17 Verify the Output Data

In **"Sample Data" on page 57**, you created a sample input XML file and set up the DTDs. The Management Agent processes your input file, **Timecard.xml**, and writes the results to **s2output1.dat**.

View the 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.

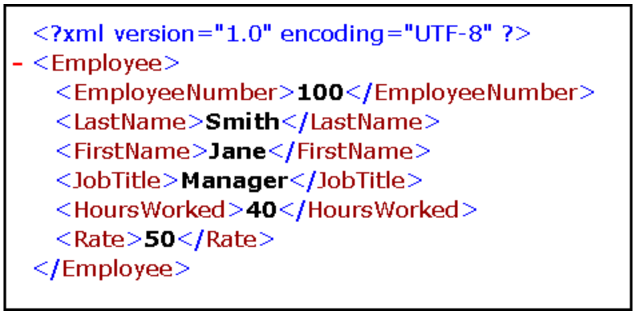
Figure 110 Output File



```
<?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 111.

Figure 111 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>
```

To 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 99).

Chapter 5

Enterprise Manager Overview

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

For more detailed information about the Enterprise Manager see the *eGate Integrator System Administrator Guide*.

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.

- 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)

5.1.1 When to Use the Monitor

Enterprise Manager will notify you by flashing the affected component node structure when:

- There is a problem within the Project
- You have stopped a server within a Project or Environment, or stopped an object in a Connectivity map

5.1.2 Using Project View

Project view allows you to perform run-time monitoring and management of Deployments and Connectivity Maps.

5.1.3 Using Environment View

Environment view allows you to perform run-time monitoring and management of eGate and eInsight System components.

5.1.4 Locating the Problem Node

When a problem occurs, the Project node is decorated with a flashing highlight, which is propagated upward from the problem node. When you expand the tree, the problem node will be highlighted and will also have a flashing exclamation point appended to it. The exclamation point association visually calls out the target node (or nodes) for troubleshooting the problem.

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*.

1. Start Enterprise Manager and type your username and password.
2. Click **Login**.
3. Click the **Home** tab.
4. Double-click **Monitor**.

You can click the Project 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.

5.2.1 Monitor a Collaboration in your Connectivity Map

Enterprise Manager Graphic Tab

With the Enterprise Manager open, perform the following steps:

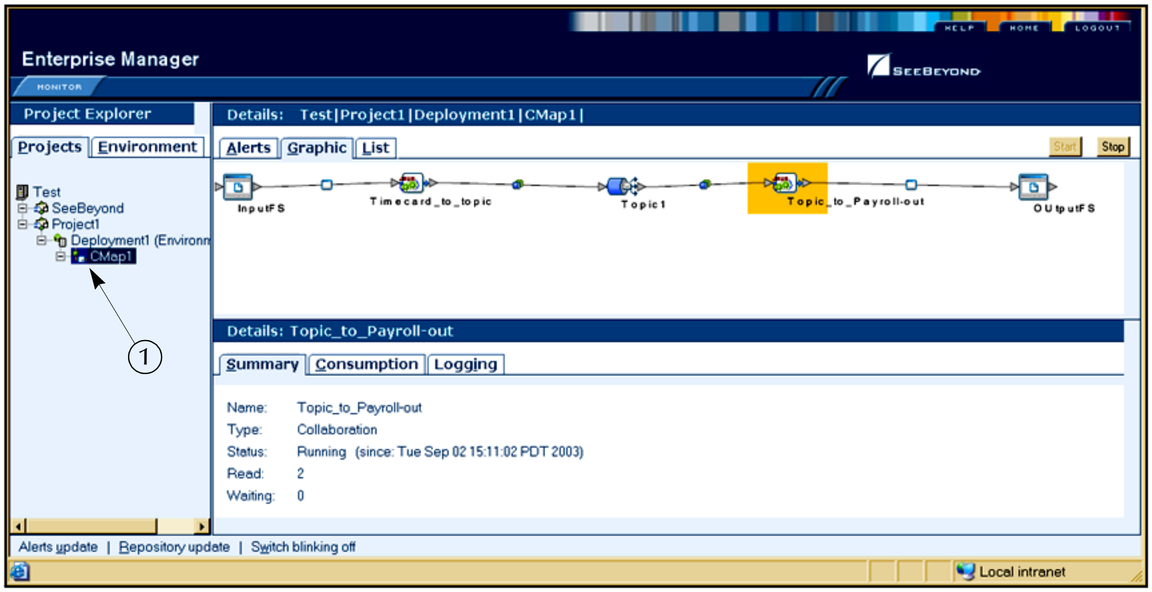
1. Click on the Project tab.
2. Click on one of the tabs to display information.
   - **List** shows deployment status
   - **Graphic** shows the Connectivity Map
   - **Alerts** shows message types and severity of the message
3. Click on the Graphic tab. (If this tab is not available, you may need to download and install the SVG Viewer.)

4. Expand all the views by double-clicking the repository name (Test).
5. Click on the CMap1 icon (number 1 in Figure 112).

   (The Connectivity Map will appear.)
6. Double-click on **Topic_to_payroll_out**.

   Refer to the following Figure 112.

Figure 112 Enterprise Manager CMap1



Name: Topic_to_Payroll-out
Type: Collaboration
Status: Running (since: Tue Sep 02 15:11:02 PDT 2003)
Read: 2
Waiting: 0

The Enterprise Manager shows that the collaboration Service is open for **Topic_to_payroll_out**. The status shows that the application is running and there have been two reads of the file.

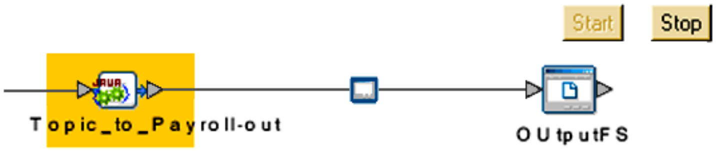
5.2.2 Stop Service

Using the example above, illustrated in Figure 112, you can also start and stop services. For example, if you want to stop the **Topic_to_payroll_out** Service (stop processing the messages coming from Topic1 and instead hold them in a queue) you would perform the following steps:

Stop Topic_to_payroll_out Service

1. Click the **Topic_to_payroll_out** Service (icon)

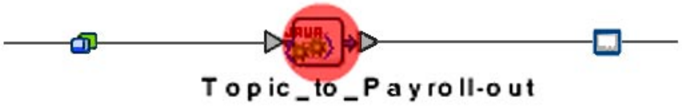
Figure 113 Stop Service Button



2 Click the **Stop** button

A red disk appears over the icon to indicate that the Service is stopped.

Figure 114 Stop Service Indicator



Keep in mind that although the **Topic_to_payroll_out** Service will not process messages, those messages will be held in a queue and will be processed immediately when the Service is started, by clicking on the Start button.

5.2.3 Monitor Alerts

Enterprise Manager Project Tab

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

1 Click on the Project tab.

2 Expand all the views by double-clicking the repository name (Test).

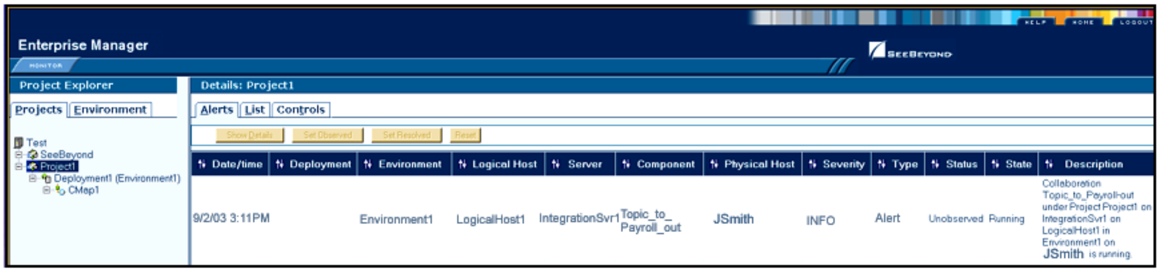
3 Click on the word Project (will be underlined) next to the Project2 icon.

4 Click on the Alerts tab.

(The alert messages will display with a description in the far right column.)

Refer to the following Figure 115.

Figure 115 Enterprise Manager CMap1



Date/time	Deployment	Environment	Logical Host	Server	Component	Physical Host	Severity	Type	Status	State	Description
9/2/03 3:11PM		Environment1	LogicalHost1	IntegrationSvr1	Topic_to_ Payroll_out	JSmith	INFO	Alert	Unobserved	Running	Collaboration Topic_to_Payroll-out under Project Project1 on IntegrationSvr1 on LogicalHost1 in Environment1 on JSmith is running.

5.2.4 Check Log Messages

You can troubleshoot errors by viewing the Logged messages.

In this tutorial, for example, the input timecard 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 $42.50, he or she would have to input the integer as 4250, 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 42.50 (timecard input), the decimal would generate an error message and the output file would not be updated.

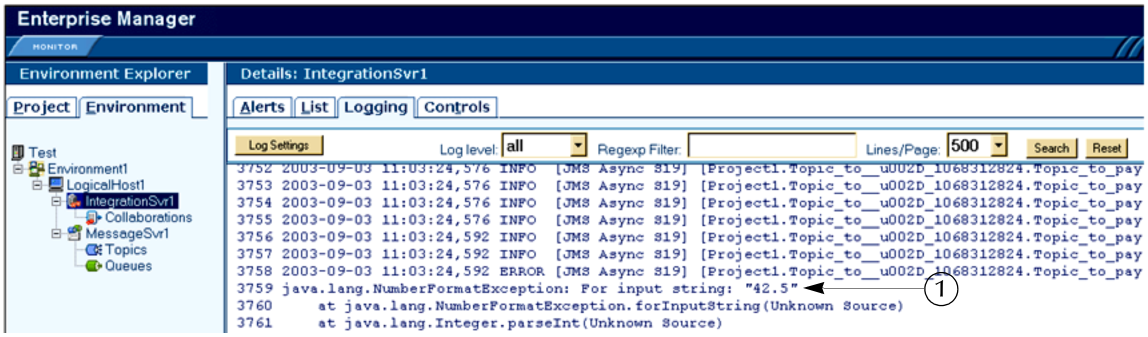
View Logged Messages

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 (Test).
3. Click **IntegrationSvr1**.
4. Click the **Logging** tab.
5. Click the **Search** button.
6. Scroll down to find error messages.

   The error, **NumberFormatException** is recorded in the log (number 1 in the following Figure 116). Use the “search on page for” feature to find “exception.”

Figure 116 Check Logged Messages



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

```
3752 2003-09-03 11:03:24,576 INFO  [JMS Async S19] [Project1.Topic_to__u002D_1068312824.Topic_to_pay
3753 2003-09-03 11:03:24,576 INFO  [JMS Async S19] [Project1.Topic_to__u002D_1068312824.Topic_to_pay
3754 2003-09-03 11:03:24,576 INFO  [JMS Async S19] [Project1.Topic_to__u002D_1068312824.Topic_to_pay
3755 2003-09-03 11:03:24,576 INFO  [JMS Async S19] [Project1.Topic_to__u002D_1068312824.Topic_to_pay
3756 2003-09-03 11:03:24,592 INFO  [JMS Async S19] [Project1.Topic_to__u002D_1068312824.Topic_to_pay
3757 2003-09-03 11:03:24,592 INFO  [JMS Async S19] [Project1.Topic_to__u002D_1068312824.Topic_to_pay
3758 2003-09-03 11:03:24,592 ERROR [JMS Async S19] [Project1.Topic_to__u002D_1068312824.Topic_to_pay
3759 java.lang.NumberFormatException: For input string: "42.5"
3760      at java.lang.NumberFormatException.forInputString(Unknown Source)
3761      at java.lang.Integer.parseInt(Unknown Source)
```

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, Passthrough Collaborations, topics, queues, and eWays. A Connectivity Map is created under a Project. A Project may have multiple Connectivity Maps.

Constants

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

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.

"Data Warehouse".

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