SUN SEEBEYOND

eGATE™ INTEGRATOR TUTORIAL

Release 5.1.1

Sun microsystems

Version 20060623093725

Contents

Chapter 5

Chapter 6

Chapter 7

List of Figures

Chapter 1

Introduction

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

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

What's in This Chapter

1.1 About this Document

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

1.1.1 What's in this Document

- Chapter 1 **"Introduction"** overviews the contents of the *Sun SeeBeyond eGate Integrator Tutorial*, describes the writing conventions used in this document, and provides a complete list of related eGate Integrator documentation.
- Chapter 2 **"Features of the Enterprise Designer"** introduces the Enterprise Designer, Menu Bar, Enterprise Explorer, and Project Editor.
- Chapter 3 **"Building a Project - Scenario1"** provides a step-by-step, end-to-end scenario where you perform a data concatenation. Scenario1 uses an input DTD (OTD) to create an output XSD (OTD).

- **Chapter 4** **"Building a Project - Scenario2"** provides a step-by-step, end-to-end scenario where you perform data transformation, involving a multiplication operation. Scenario2 uses a user-defined OTD.
- **Chapter 5** **"Building a Project - Scenario3"** provides a step-by-step, end-to-end scenario involving a concatenation and a multiplication operation where you create an eGate Project using XSD schemas to create OTDs.
- **Chapter 6** **"Web Services - Scenario4"** provides a step-by-step, end-to-end scenario that exposes a (JCD) Java Collaboration Definition as a Web service.
- **Chapter 7** **"Enterprise Manager Overview"** provides an overview of the Enterprise Manager which is used to view and monitor projects dynamically.
- **"Glossary"**.
- **"Index"**

1.1.2 Scope

This eGate Integrator Tutorial provides four scenarios to lead you through all the steps required to successfully build and run an end-to-end Project, including bringing data in using OTDs, manipulating data using Collaborations, deploying the Project, and viewing the results. Sample data and the expected output are provided for each scenario.

1.1.3 Intended Audience

This document is intended for eGate Integrator users who want to create and deploy a sample Project, end-to-end, using the sample data provided by Sun Microsystems. This document also serves as a hands-on introduction to eGate Integrator to help new users get up to speed quickly.

1.1.4 Text Conventions

The following conventions are observed throughout this document.

Table 1 Text Conventions

Text Convention	Used For	Examples
Bold	Names of buttons, files, icons, parameters, variables, methods, menus, and objects	- Click **OK**. - On the **File** menu, click **Exit**. - Select the **eGate.sar** file.
`Monospaced`	Command line arguments, code samples; variables are shown in ***bold italic***	`java -jar` ***`filename`***`.jar`
Blue bold	Hypertext links within document	See **Text Conventions** on page 15

Table 1 Text Conventions (Continued)

Text Convention	Used For	Examples
Blue underlined	Hypertext links for Web addresses (URLs) or email addresses	http://www.sun.com

1.1.5 Screenshots Used in this Document

Depending on what products you have installed, and how they are configured, the screenshots in this document may differ from what you see on your system.

1.2 Related Documents

The following Sun Microsystems documents provide additional information about eGate Integrator and the Composite Application Platform Suite:

- *Composite Application Platform Suite Installation Guide*
- *eGate Integrator System Administration Guide*
- *eGate Integrator Release Notes*
- *eGate Integrator User's Guide*
- *Composite Application Platform Suite Primer*

1.3 Suite Installation Requirements

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

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

Refer to the *Installation Guide* for system requirements and installation instructions. See also: **Chapter 3**, **Files Required for the Tutorials** on page 22.

1.4 Sun Microsystems, Inc. Web Site

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

http://www.sun.com

1.5 Documentation Feedback

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

CAPS_docsfeedback@sun.com

Chapter 2

Features of the Enterprise Designer

The Enterprise Designer is the graphical user interface (GUI) used to design and implement Sun Java Composite Application Platform Suite (Java CAPS) projects. This chapter overviews the features and interface of the Enterprise Designer window.

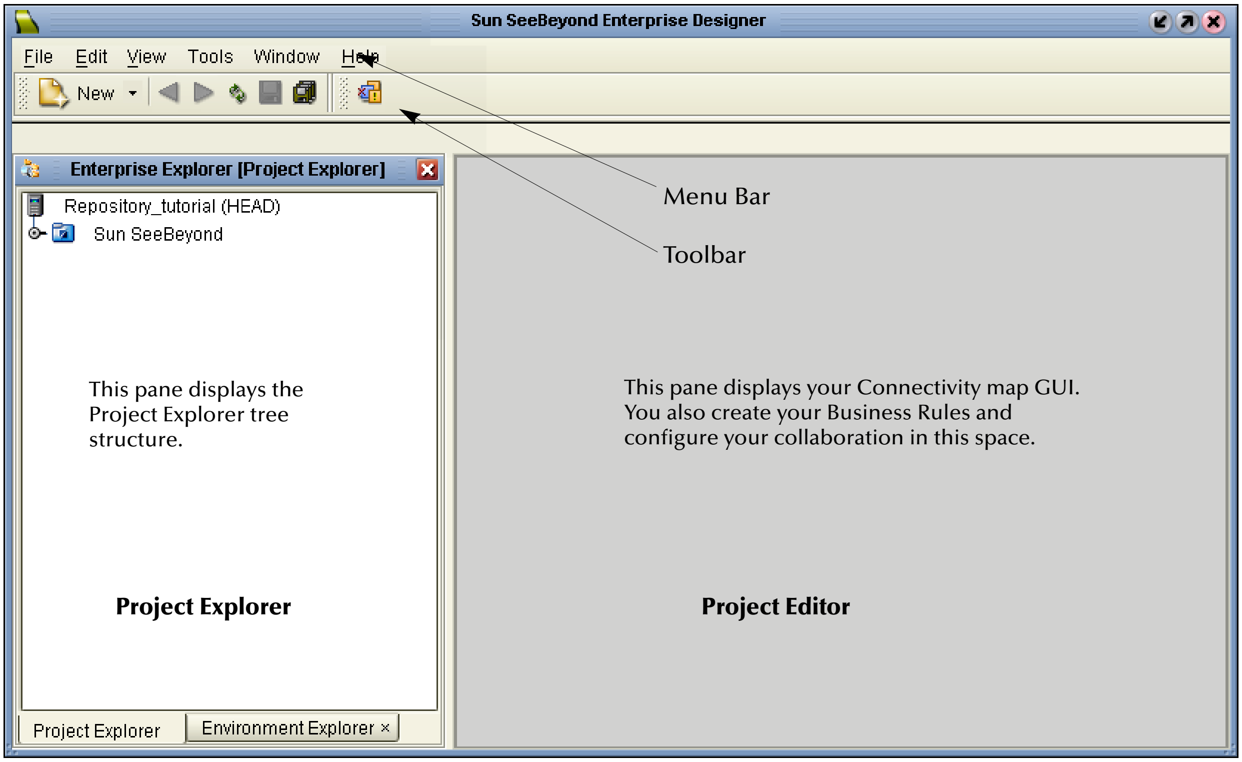
What's in This Chapter

2.1 Enterprise Designer Objects and Components

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

An object is a more generic term and also refers to eWays. A component usually describes a runnable "thing," such as a Java Collaboration.

Figure 1 Enterprise Designer


Sun SeeBeyond Enterprise Designer
File Edit View Tools Window Help
New
Enterprise Explorer [Project Explorer]
Repository_tutorial (HEAD)
Sun SeeBeyond
Menu Bar
Toolbar
This pane displays the Project Explorer tree structure.
This pane displays your Connectivity map GUI. You also create your Business Rules and configure your collaboration in this space.
Project Explorer
Project Editor
Project Explorer
Environment 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 provides access to a series of menu commands used to build and manage a Project.

The function of each menu is briefly described below:

- **File**—Lists options for Save, Save All, and Exit.
- **Tools**—Lists options for Impact Analyzer; Options: Options Setup, Language, Build Options, and Preferences; Update Center (options).
- **View**—Lists options to view the Environment Explorer and Project Explorer.
- **Window**—Lists options for the open windows and various window layouts.
- **Help**—online help for all installed modules.

2.3 Enterprise Explorer

The Enterprise Explorer organizes all of the objects and 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 objects and 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 objects and components and information about external systems, which may be involved with an eGate configuration. You will go into the Environment Explorer when you are ready to deploy and run your project.

2.4 Project Editor

The Project Editor is where you use GUI tools (or even enter code directly) to create a Project. This part of the Enterprise Designer is empty when you start a new Project. However, as you work through the tutorial, the Project Editor quickly fills with objects 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 objects and components, such as Services, Topics, Queues, and eWays, that you include in the design 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. A Deployment Profile contains information about how Project components are mapped and deployed within an Environment.
- **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 an eGate Project. This Project addresses a very simple business challenge where an input TimeCard File is read and a workfile is output. The input schema is a DTD and the output schema is an XSD. The input data is an XML File.

What's in This Chapter

3.1 Files Required for the Tutorials

Please refer to the *Sun Composite Application Platform Suite Installation Guide* for instructions about uploading files to the Repository. The following files/products are required to run the tutorials:

- eGate.sar
- Enterprise_manager-win32 (recommended)
- logicalhost-win32
- FileeWay.sar
- stcuddi.Win32 (for web service Project)
- eGateDocs.sar (PDF and sample files)

The Tutorial assumes the software is installed in **`C:\JavaCAPS51`**. Your installation may be different.

Although not required for the Enterprise Manager, you will have more features for GUI displays if you use the SVG Viewer. Use the **Enterprise_Manager_SVGPlugin-win32.sar** (required for the Adobe SVG Viewer plug-in for Windows). See Chapter 7 for an overview of Enterprise Manager.

3.1.1 Sample Files Used in the Tutorial

The input file names are different in each of the four scenarios. The extensions and prefixes are different.

Scenario1 uses a DTD and an XSD schema. The input is XML.

Scenario2 uses an input text file to generate a user defined OTD.

Scenario3 uses imported XML schemas (XSDs) to create XSD Nodes. The input is XML.

Scenario4 exposes a Java Collaboration as a web service.

3.2 Sample Data

Scenario1 uses the sample files found in the Project1 folder of the **eGate Tutorial Sample** that you download using the steps below. The files/schemas used in Scenario1 are:

- Schemas: **s1_input.dtd**, **s1_output.xsd**
- Input File: **s1_TimeCard.xml**

  (The “s1” in the file names stands for Scenario1.)

3.2.1 Download the Sample files

Download the eGate Tutorial **Sample Projects**.

The sample zip file contain the following folders: Project1, Project2, and Project3, and Project4. Each folder contains the input file and DTDs or XSDs needed for your sample Project. The output file is generated for each Project and the output File schema is included for each Project. Project2 only includes an input file and does not use a schema because the OTD is "user defined."

Note: *In order for the eGate Integrator Tutorial PDF File and the sample files to be available, make sure you have uploaded the* ***eGateDocs.sar*** *File to your Repository.*

Use the Sample files

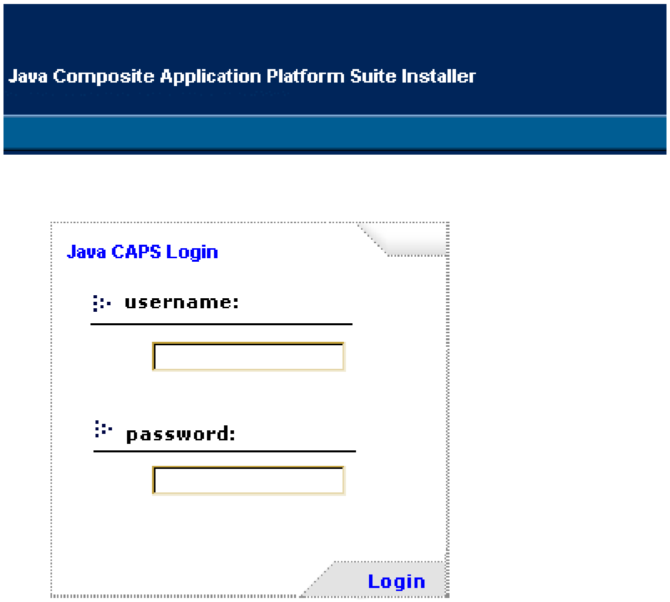
To obtain the sample files use the port number for your Repository Server, and make sure your Repository server is running.

1 Bring up the **Suite Installer**.

For example:

`http://localhost:12000/`.

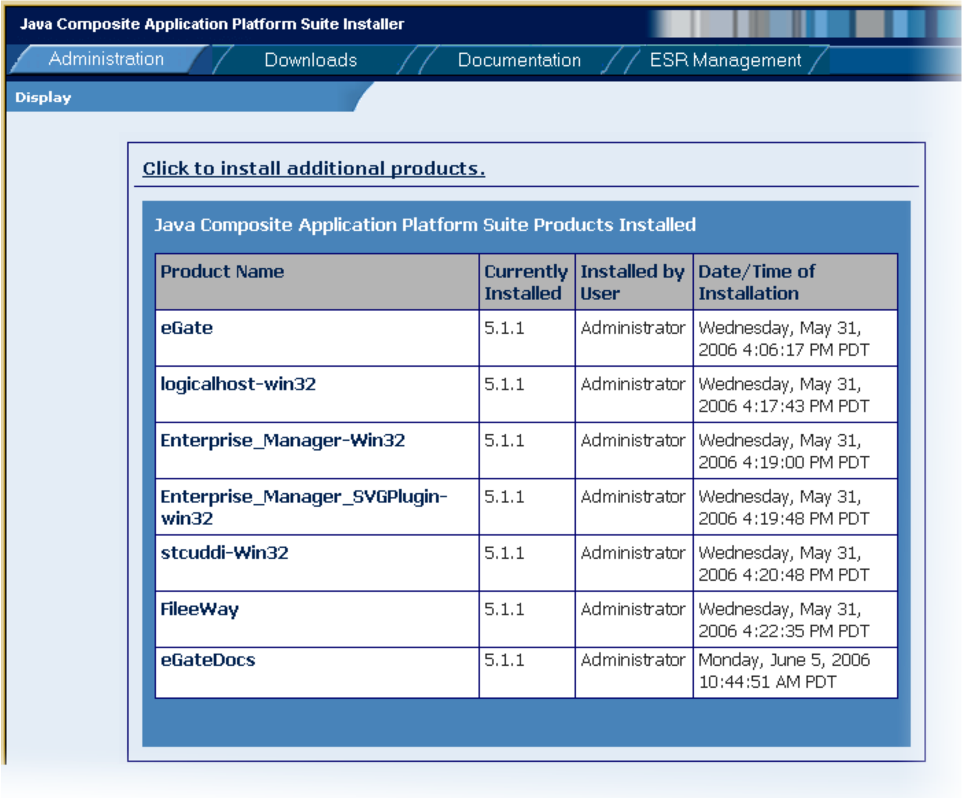
Figure 2 Enterprise Manager Login



2 Enter your **username** and **password** and login.

Note: *Before starting the scenarios in the Tutorial, make sure you have installed the products and documentation shown in the following figure.*

Figure 3 Documentation Tab



Click to install additional products.

Java Composite Application Platform Suite Products Installed

Product Name	Currently Installed	Installed by User	Date/Time of Installation
eGate	5.1.1	Administrator	Wednesday, May 31, 2006 4:06:17 PM PDT
logicalhost-win32	5.1.1	Administrator	Wednesday, May 31, 2006 4:17:43 PM PDT
Enterprise_Manager-Win32	5.1.1	Administrator	Wednesday, May 31, 2006 4:19:00 PM PDT
Enterprise_Manager_SVGPlugin-win32	5.1.1	Administrator	Wednesday, May 31, 2006 4:19:48 PM PDT
stcuddi-Win32	5.1.1	Administrator	Wednesday, May 31, 2006 4:20:48 PM PDT
FileeWay	5.1.1	Administrator	Wednesday, May 31, 2006 4:22:35 PM PDT
eGateDocs	5.1.1	Administrator	Monday, June 5, 2006 10:44:51 AM PDT

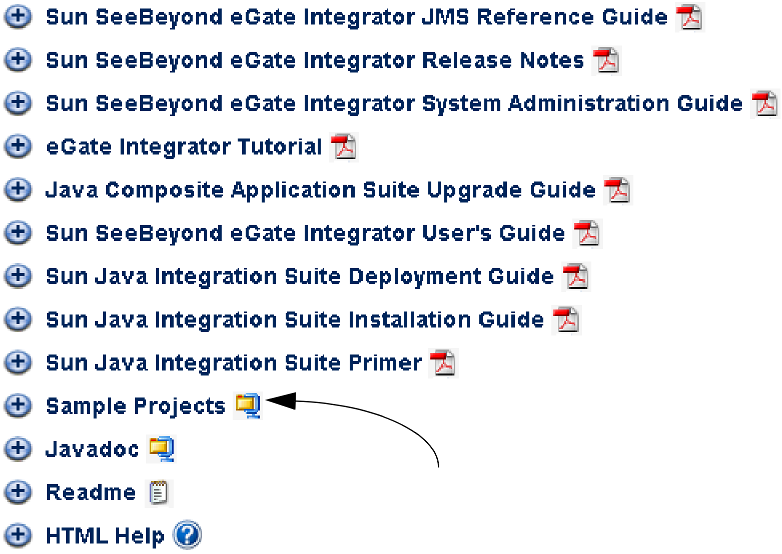
3 Click the **Documentation** tab.

4 Click **Core Products**.

5 Click **eGate Integrator**.

6 Click the **Sample Projects** zip icon and save the files to a location of your choice.

Figure 4 Download the Sample zip File



- Sun SeeBeyond eGate Integrator JMS Reference Guide
- Sun SeeBeyond eGate Integrator Release Notes
- Sun SeeBeyond eGate Integrator System Administration Guide
- eGate Integrator Tutorial
- Java Composite Application Suite Upgrade Guide
- Sun SeeBeyond eGate Integrator User's Guide
- Sun Java Integration Suite Deployment Guide
- Sun Java Integration Suite Installation Guide
- Sun Java Integration Suite Primer
- Sample Projects
- Javadoc
- Readme
- HTML Help

The zip File contains four zip folders: **Project1.zip**, **Project2.zip**, **Project3.zip**, and **Project4.zip**. After unzipping the eGate_Sample.zip, unzip the four zipped Project folders (The Project folders may appear as empty folders before each Project is unzipped).

3.2.2 Sample Input XML File

This section shows a sample of the input data used in Scenario1. Your XML File should look similar to the structure and syntax below. In Scenario1 of the Tutorial, there are four records. These files are included in the Project1 folder and do not have to be created.

Scenario1, Project1 XML File

Your (Timecard) input XML File should look similar to the structure and syntax below. In scenario1 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>
```

3.2.3 Input (DTD)

The **s1_input.dtd File** is used as a representation of the input data in the TimeCard system. The **s1_input.dtd** File uses the following elements:

The **s1_input.dtd** File uses the following elements:

```
<?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) >
```

3.2.4 Output (XSD)

The **s1_output.xsd** File is used as a representation of the data in the output file. The sample data for the output file is generated and does not have to exist ahead of time.

The Output XSD, which represents the output data, is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Output Employee Payroll File eGate Tutorial Sample XML data
for Scenario1 -->

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:complexType name="ComplexType2">
    <xs:sequence><xs:element name="EmployeeNumber"
     type="xs:string"/>
            <xs:element name="FullName" type="xs:string"/>
            <xs:element name="EmpNumber" type="xs:string"/>
            <xs:element name="PayAmount" type="xs:int"/>
        </xs:sequence>
    </xs:complexType><xs:complexType
name="ComplexType1"><xs:sequence><xs:element name="Employee"
type="ComplexType2" maxOccurs="unbounded"/>
    </xs:sequence>
    </xs:complexType><xs:annotation><xs:documentation>Root
element</xs:documentation>
    </xs:annotation>
    <xs:element name="Employees" type="ComplexType1"/>
</xs:schema>
```

3.2.5 Actual output file

See Figure 60.

3.3 Business Challenge

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

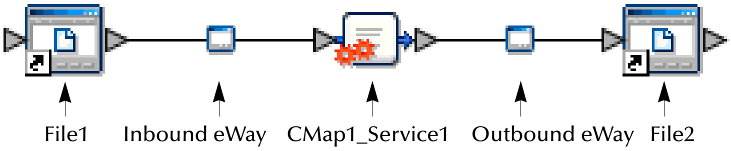
In this very simple scenario **FirstName** and **LastName** from the input file are concatenated into the field **FullName** in the output file. Also, **HoursWorked** and **Rate** are multiplied to produce **PayAmount** in the output file.

- 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**, **EmpNumber**, and **PayAmount**.

3.4 Project Description

The finished eGate Project contains objects used to process data within the Project and publish the data in the appropriate format in an output file.

Figure 5 Project Connectivity Map

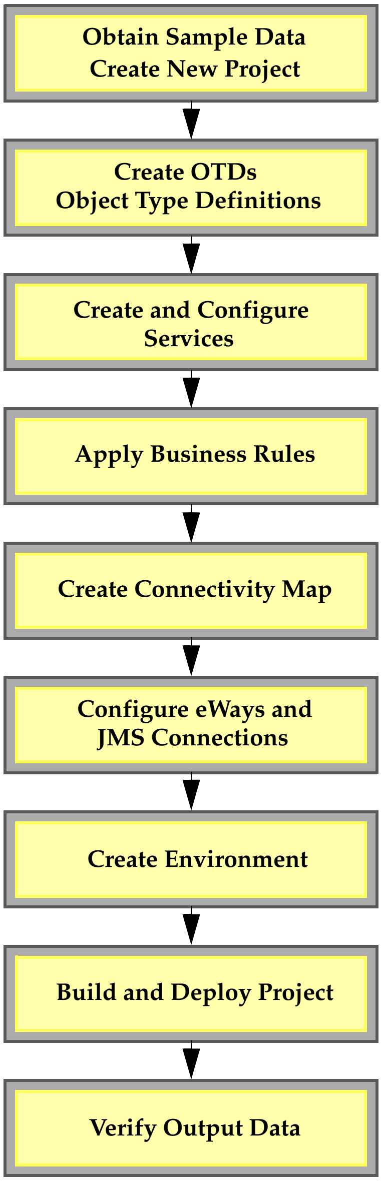


These objects (and the runnable Service component) perform the following functions:

- **File1**: The Input File *External Application* contains the Input XML data.
- **Inbound eWay**: The inbound eWay polls a specified location on the File system for the input XML data file.
- **Service1**: This Service will contain a Collaboration that processes the data.
- **Outbound eWay**: This eWay publishes the processed output data to the File system in XML format.
- **File2**: This is the output work file.

3.4.1 Project Flow Diagram - Scenario1

Figure 6 Project Flowchart - Scenario1



3.5 Start Repository Server and Enterprise Designer

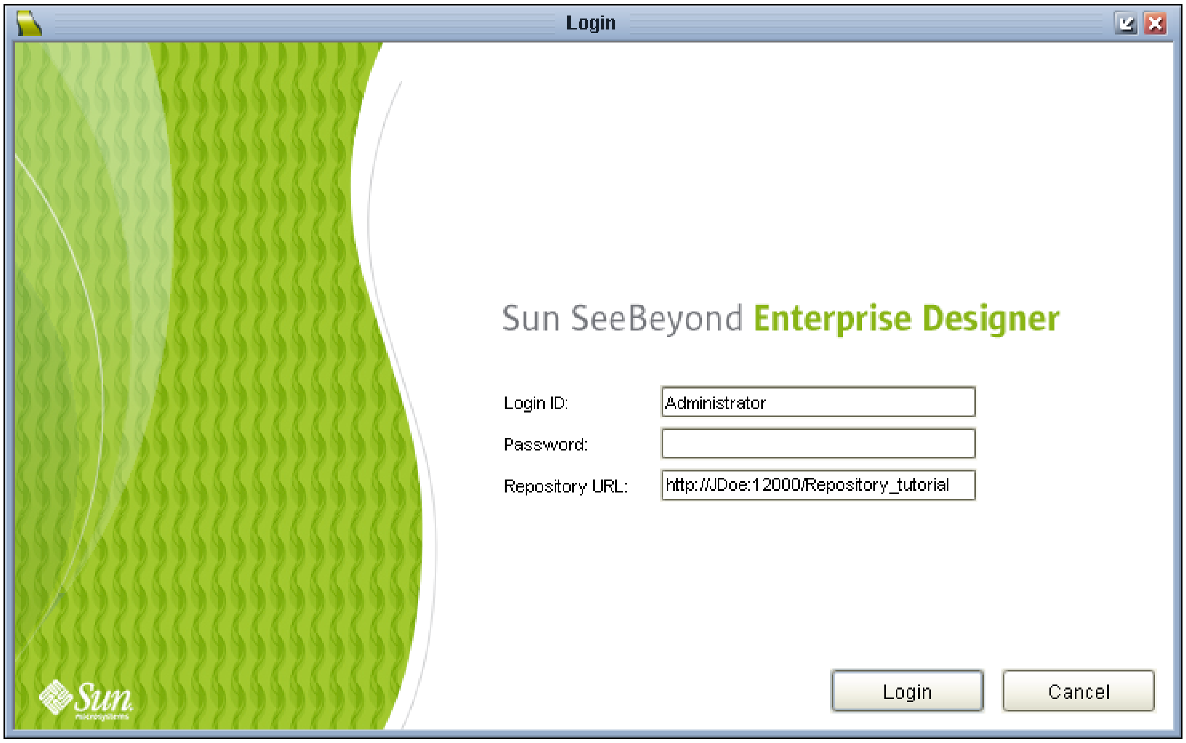
Start the Enterprise Designer according to your organization's instructions.

1 Start the Repository server (if not already running).

2 Start the Enterprise Designer using `runed.bat`. This is located in your edesigner\bin. For example: **`C:\JavaCAPS51\edesigner\bin`**.

The Enterprise Designer Login dialog box appears.

Figure 7 Enterprise Designer Login



Login ID: Administrator

Password:

Repository URL: http://JDoe:12000/Repository_tutorial

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

`http://<hostname>:<port_number>/<repository name>`.

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.*

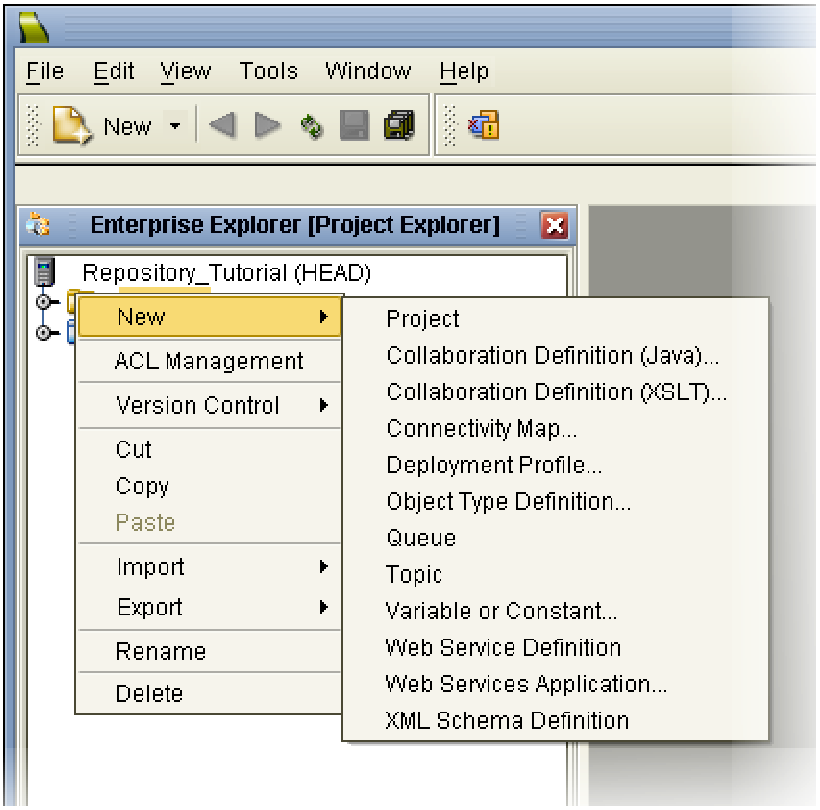
3.6 Create a New Project

Begin by creating and naming a Project in the Enterprise Designer.

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

1 In the Project Explorer tree of the Enterprise Designer, right-click the Repository name (computer icon), and click **New Project**. (You can also select New Project from the drop-down menu.)

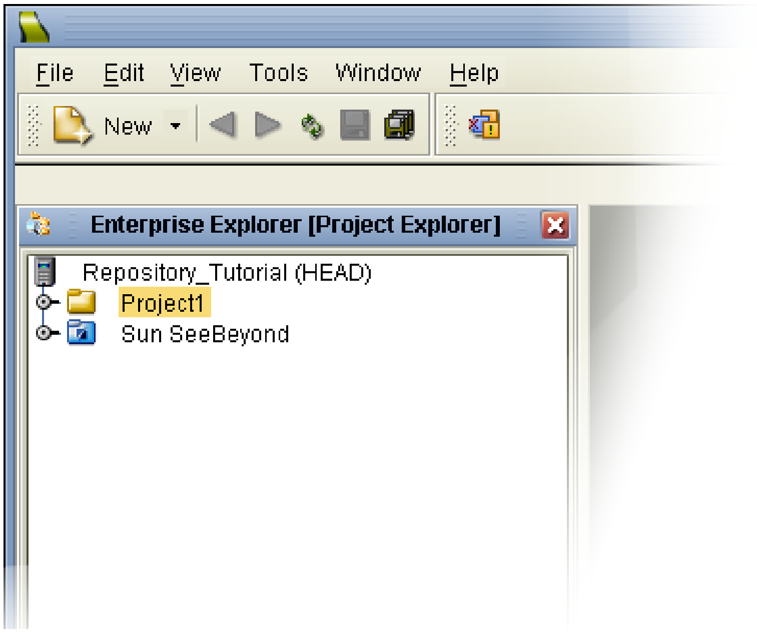
Figure 8 Create a Project



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

The **Project1** appears in the Project Explorer tree on the left side of the window.

Figure 9 Project Folder: Project1


File
Edit
View
Tools
Window
Help
New
Enterprise Explorer [Project Explorer]
Repository_Tutorial (HEAD)
Project1
Sun SeeBeyond

3.7 Create a New Object Type Definition

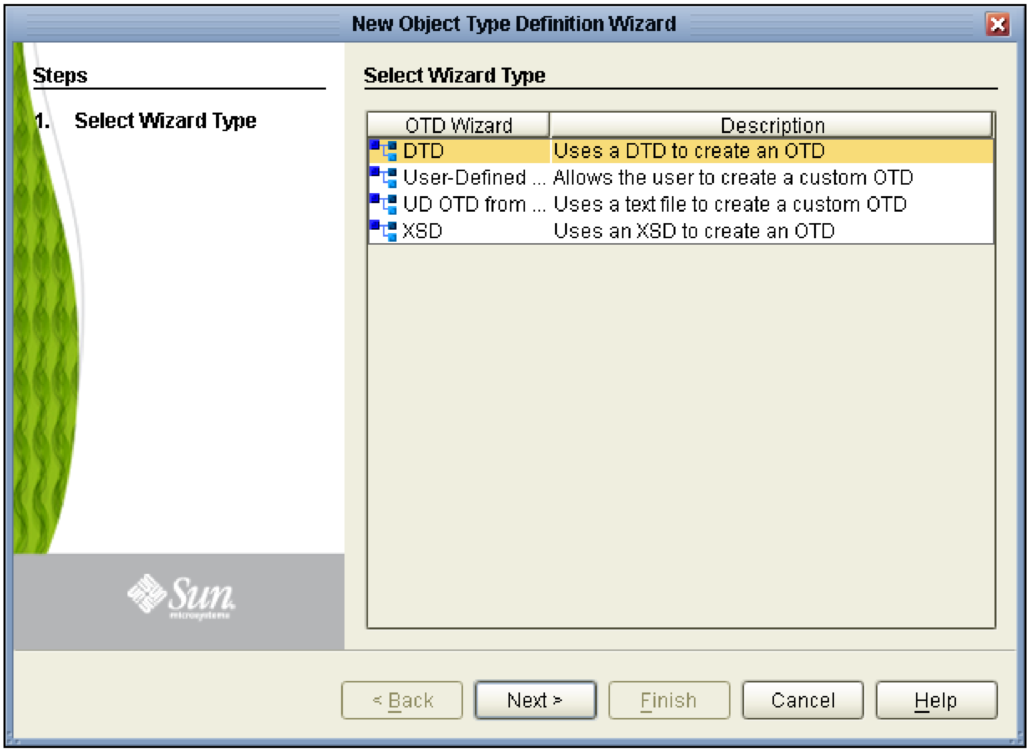
Create an object type definition (OTD) to define your data object.

Select DTD Wizard

An OTD contains the structure and rules that define an object. In this scenario the XML input is defined with a DTD.

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

Figure 10 OTD Wizard (DTD)



New Object Type Definition Wizard

Steps

1. Select Wizard Type

Select Wizard Type

OTD Wizard	Description
DTD	Uses a DTD to create an OTD
User-Defined ...	Allows the user to create a custom OTD
UD OTD from ...	Uses a text file to create a custom OTD
XSD	Uses an XSD to create an OTD

3 Select the **DTD** item.

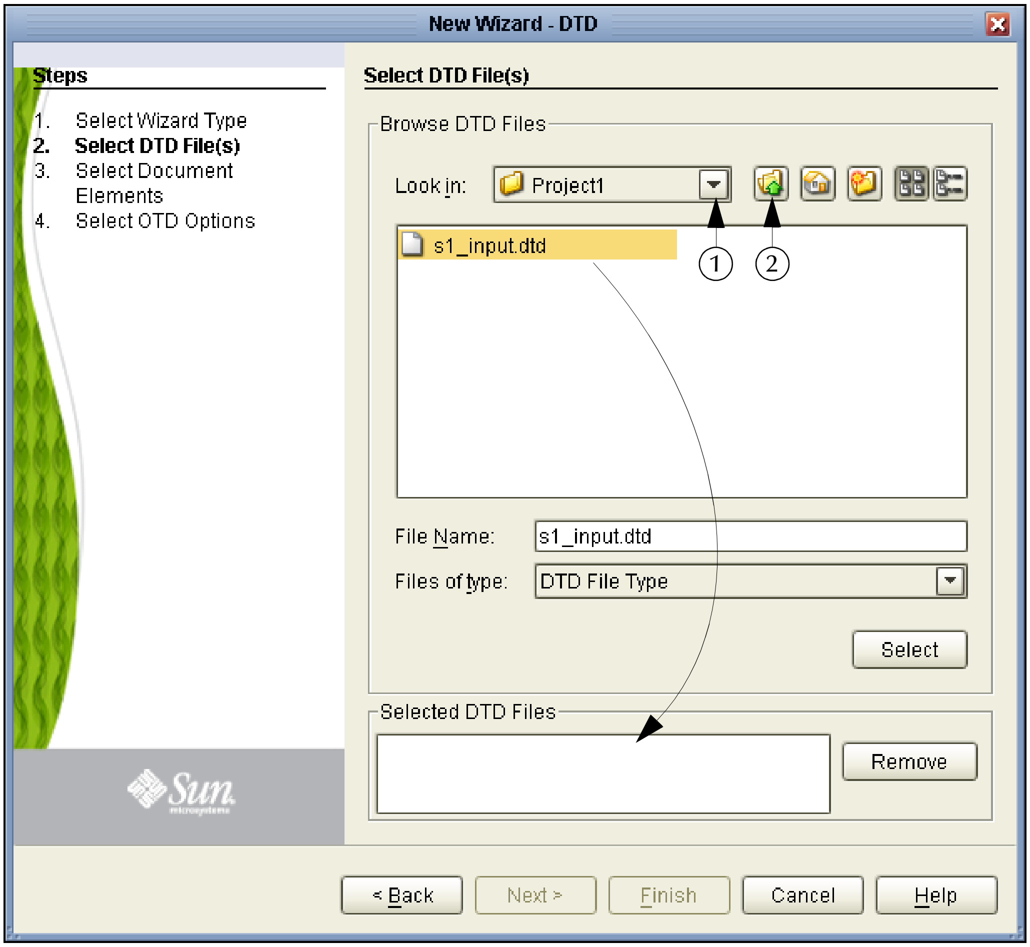
4 Click **Next**.

5 Click the **drop-down** arrow. See number 1 in Figure 11.

6 Navigate to the folder that contains your sample files. For example, **C:\eGateData\Project1**.

See the following figure.

Figure 11 Select DTD File



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)

Browse DTD Files

Look in: Project1

s1_input.dtd

File Name: s1_input.dtd

Files of type: DTD File Type

Selected DTD Files

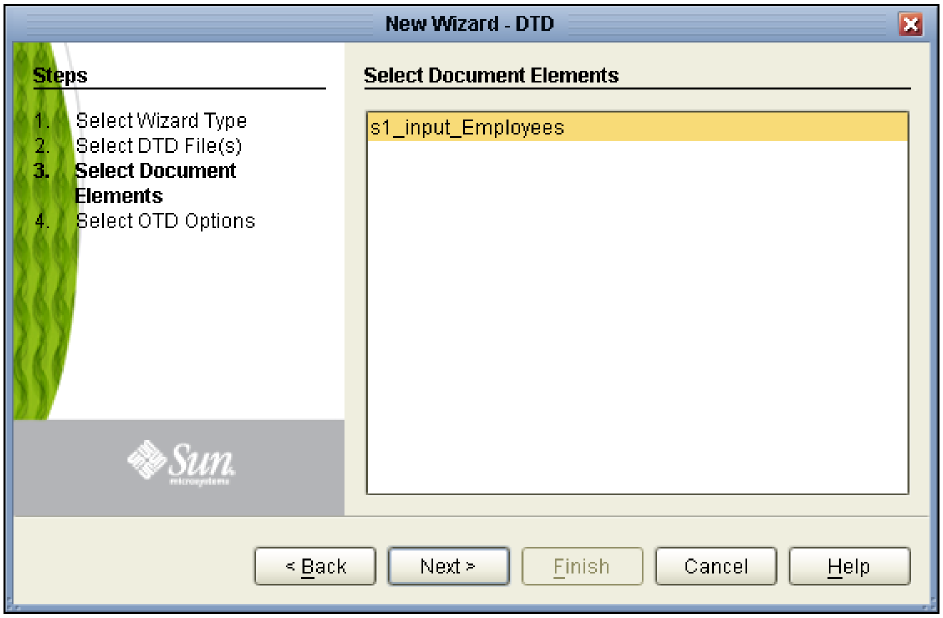
7 Highlight **s1_input.dtd**. Click **Select** (or double-click).

8 When **s1_input.dtd** appears in the Selected DTDs, click **Next**.

Note: *The **Up one Level** button (number 2 in Figure 11 above) is used to navigate one directory level up; this is useful if you have made an incorrect selection.*

Note: *As you progress through the steps, notice that the highlighted text in the left side of the New Wizard DTD window changes. In the previous step **Select DTD File(s)** was bolded. Now **Select Document Elements** is bolded, See Figure 12.*

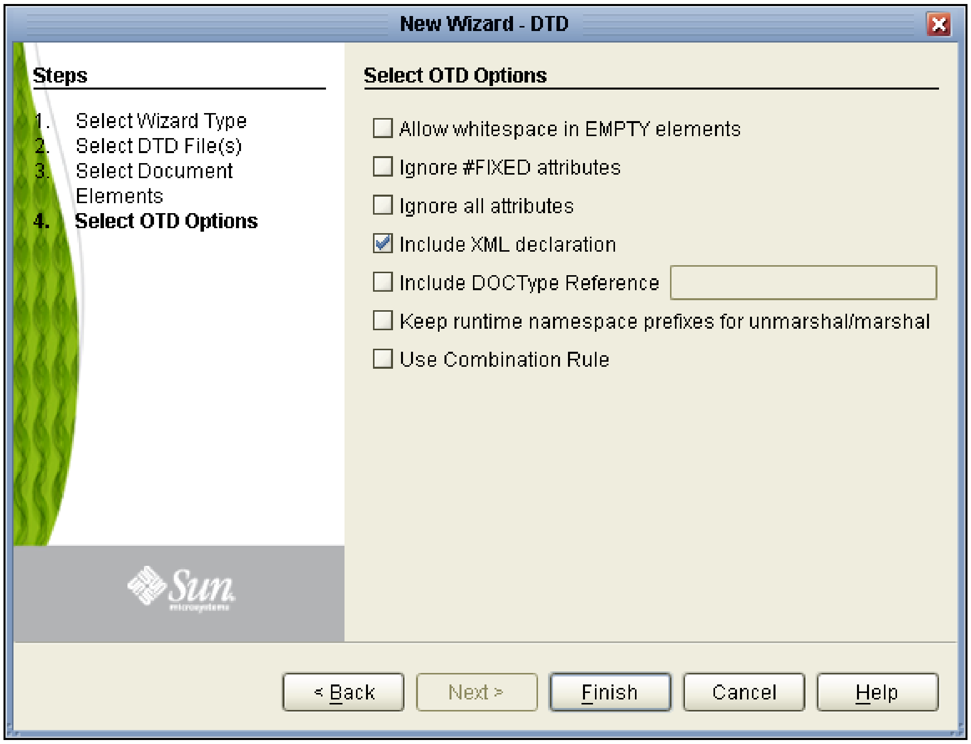
Figure 12 Select Document Elements



Select Document Elements

1. Select **s1_input_Employees**.
2. Click **Next**.

Figure 13 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

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

The OTD Editor window opens. See Figure 17.

4 On the **File** toolbar, 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.*

The OTD Editor provides the developer with a graphical representation of the structure of the data.

Select XSD Wizard

In this scenario the XML output is defined with an XML schema definition (XSD).

1 Right-click **Project1**.

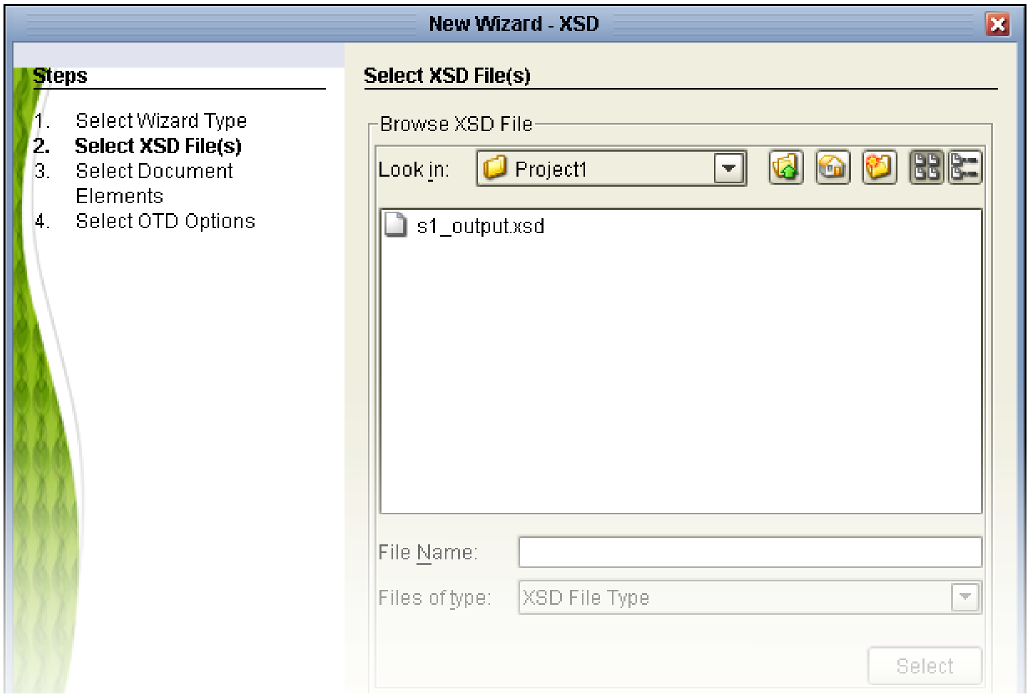
2 Click **New**, **Object Type Definition**.

3 This time select the XSD wizard and click **Next**.

The XSD wizard appears.

4 Navigate to and select **s1_output.xsd**.

Figure 14 OTD Wizard (XSD)



New Wizard - XSD

Steps

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

Select XSD File(s)

Browse XSD File

Look in: Project1

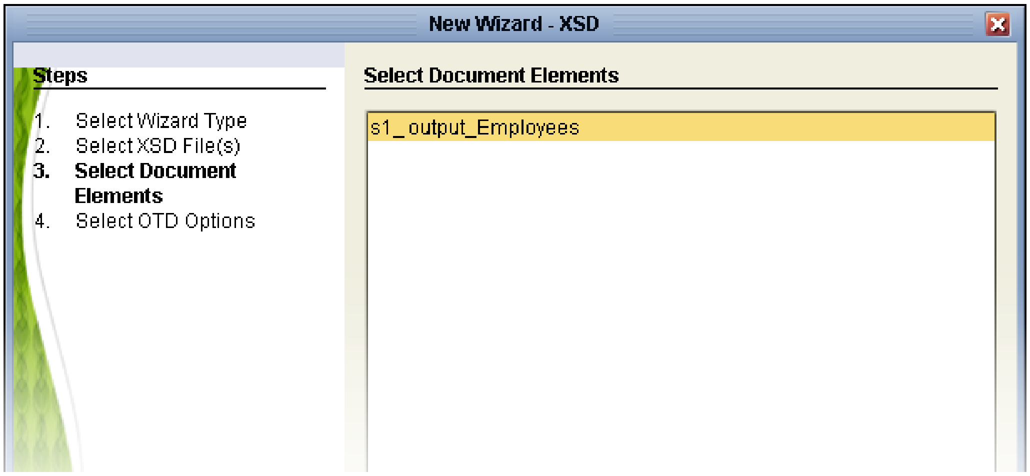
s1_output.xsd

File Name:

Files of type: XSD File Type

5 Double-click the output schema and click **Next**.

Figure 15 Select Output Elements

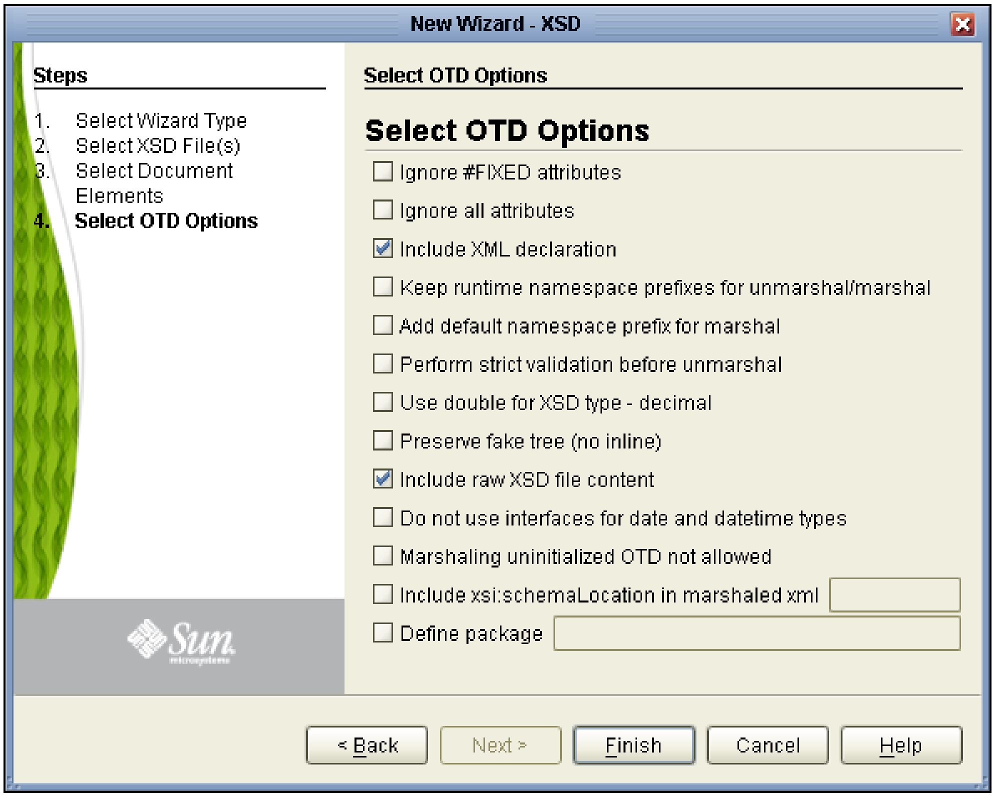


6 Select **s1_output_Employees**.

7 Click **Next**.

The "Select OTD Options" dialog box appears.

Figure 16 Select OTD Options

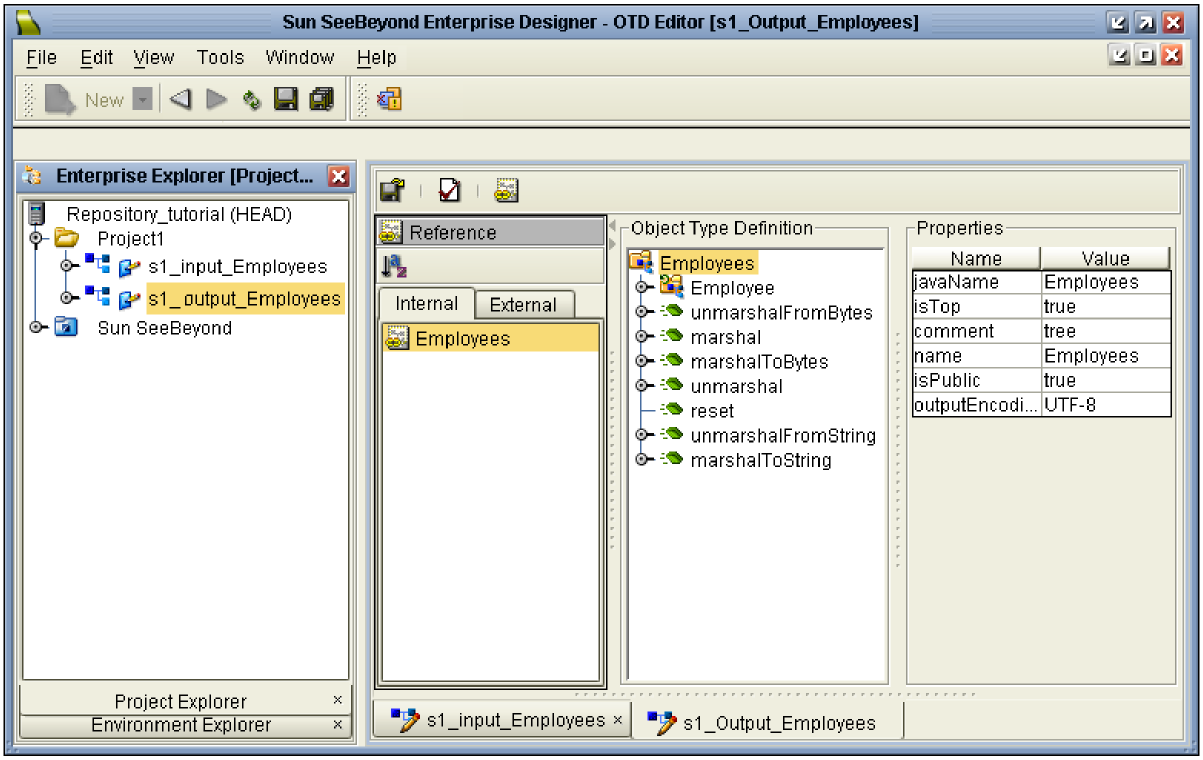


Select OTD Options

- □ Ignore #FIXED attributes
- □ Ignore all attributes
- ☑ Include XML declaration
- □ Keep runtime namespace prefixes for unmarshal/marshal
- □ Add default namespace prefix for marshal
- □ Perform strict validation before unmarshal
- □ Use double for XSD type - decimal
- □ Preserve fake tree (no inline)
- ☑ Include raw XSD file content
- □ Do not use interfaces for date and datetime types
- □ Marshaling uninitialized OTD not allowed
- □ Include xsi:schemaLocation in marshaled xml
- □ Define package

8 Accept the defaults "Include XML declaration" and "Include raw XSD File content" and click **Finish**.

Figure 17 OTD Editor Input



Name	Value
javaName	Employees
isTop	true
comment	tree
name	Employees
isPublic	true
outputEncodi...	UTF-8

9 Click **Close All** from the Window menu bar. (This closes all open windows. Do not close the Enterprise Designer.)

10 Click **Save All**.

3.8 Configure the Services

Use Service wizards in the following steps to configure your Collaboration. A Collaboration describes your data and connections and contains information about message routing and transformation logic.

3.8.1 Configure Service1

Configure a Collaboration (Java) definition.

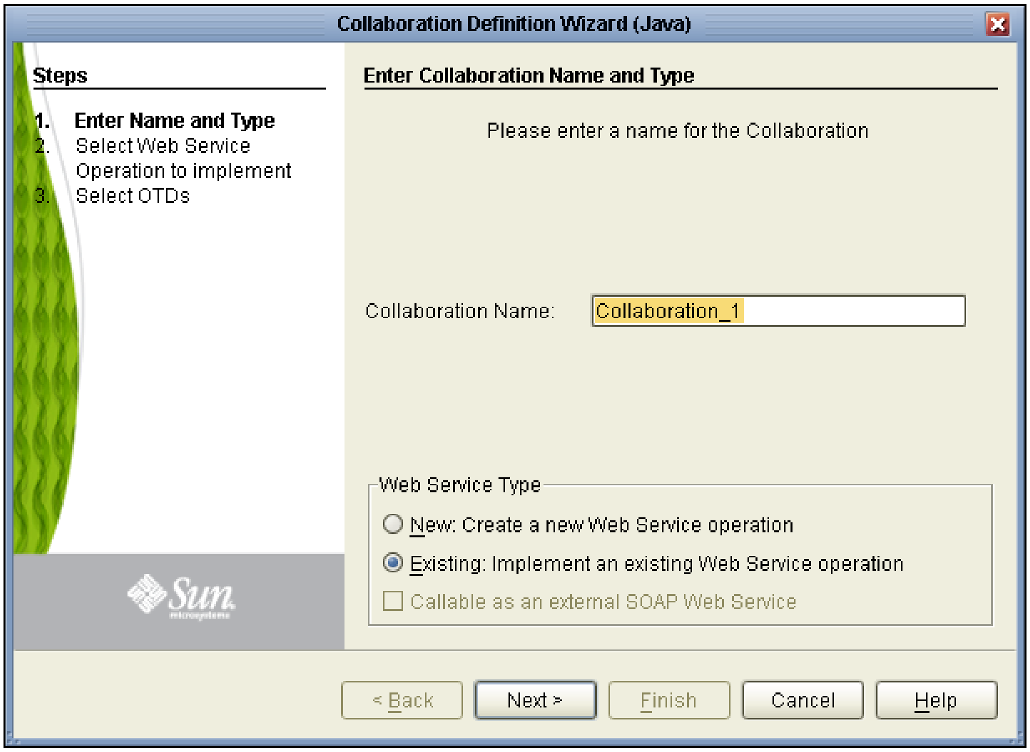
Enter a Collaboration Name

1 Right-click **Project1**.

2 Click **New, 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

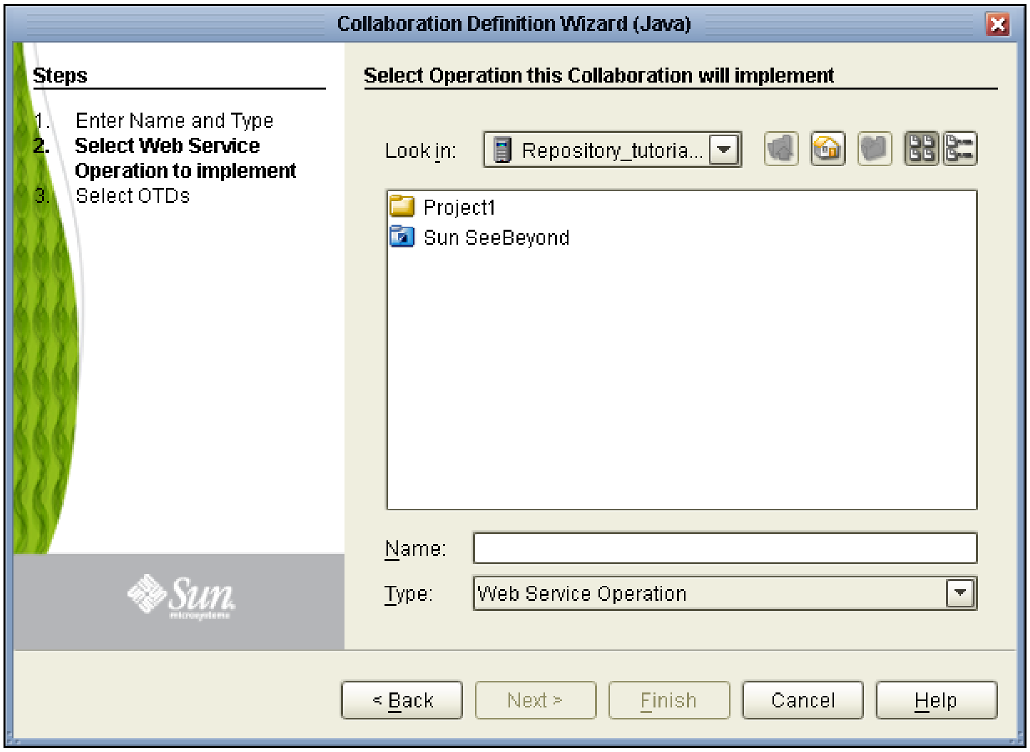
● Existing: Implement an existing Web Service operation

□ Callable as an external SOAP Web Service

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

3 Click **Next**.

Figure 19 Select Web Service File Receive



Collaboration Definition Wizard (Java)

Steps

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

Select Operation this Collaboration will implement

Look in: Repository_tutoria...

Project1
Sun SeeBeyond

Name:

Type: Web Service Operation

Create a File Read Web Service

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

1 Double-click **Sun SeeBeyond**.

2 Double-click **eWays.** eWays are message end points that connect to an external service provider to enable the sending and receiving of messages.

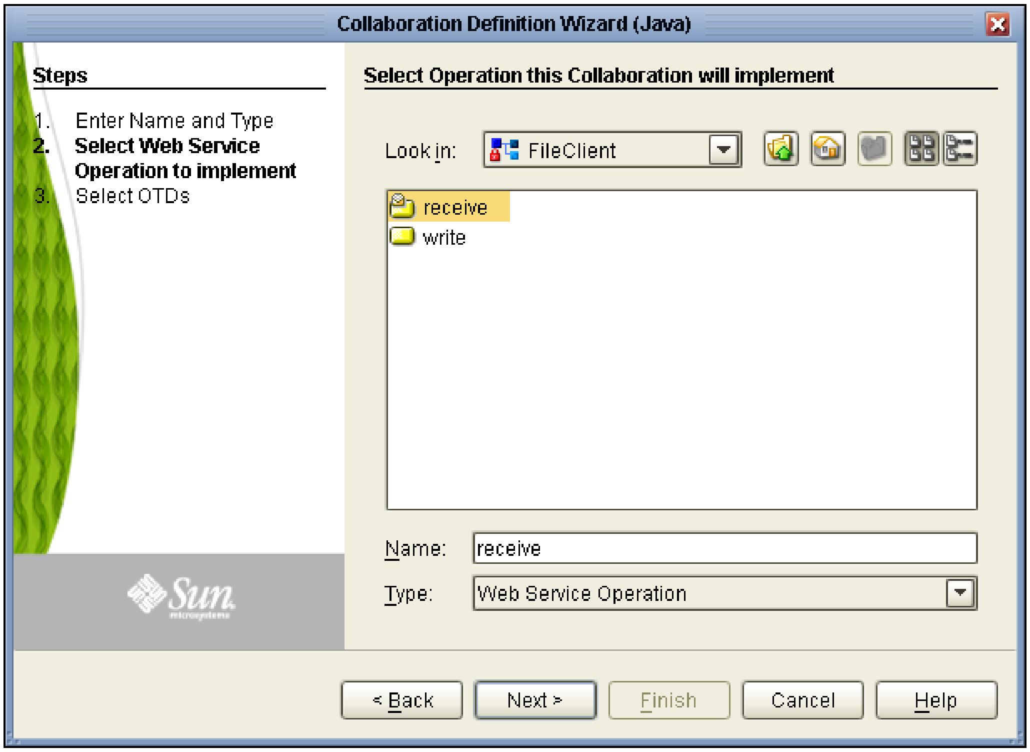
3 Double-click **File**.

4 Double-click **FileClient**.

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

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

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

Name: receive

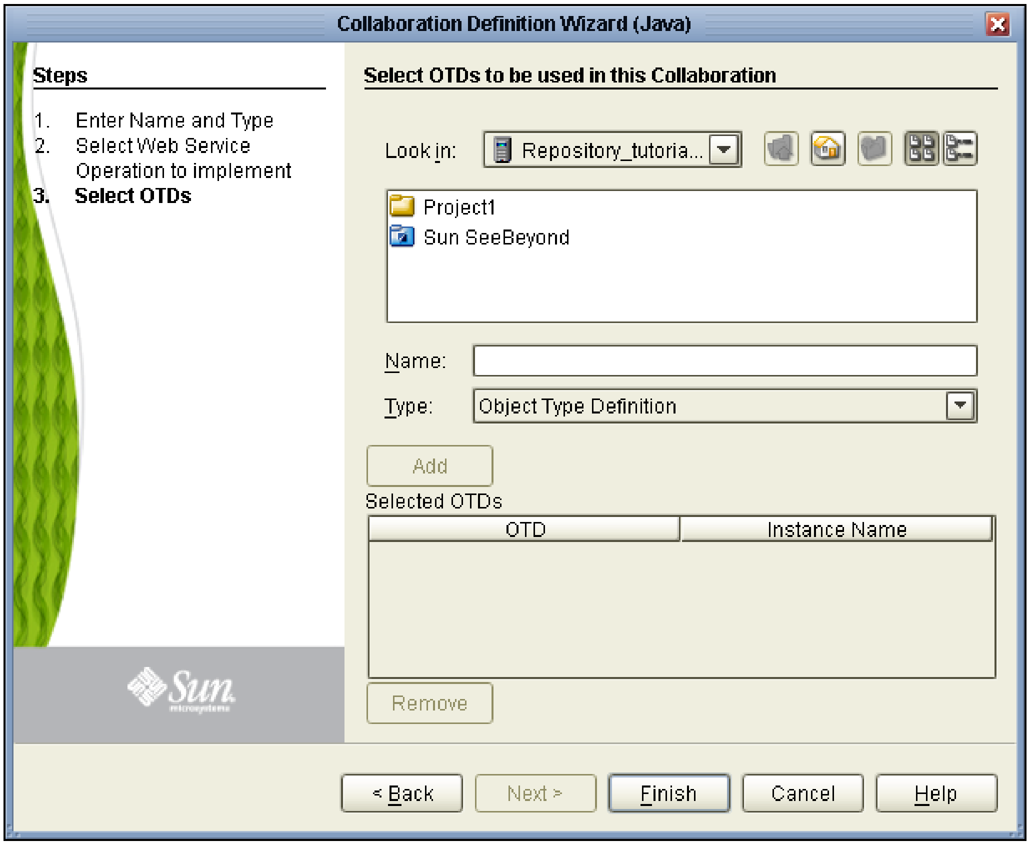
Type: Web Service Operation

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

Select an OTD for the outbound FileClient

First select the output **FileClient** OTD. (Its parameters are automatically defined.) Then, select the XML OTDs.

Figure 21 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: Repository_tutoria...

Project1
Sun SeeBeyond

Name:

Type: Object Type Definition

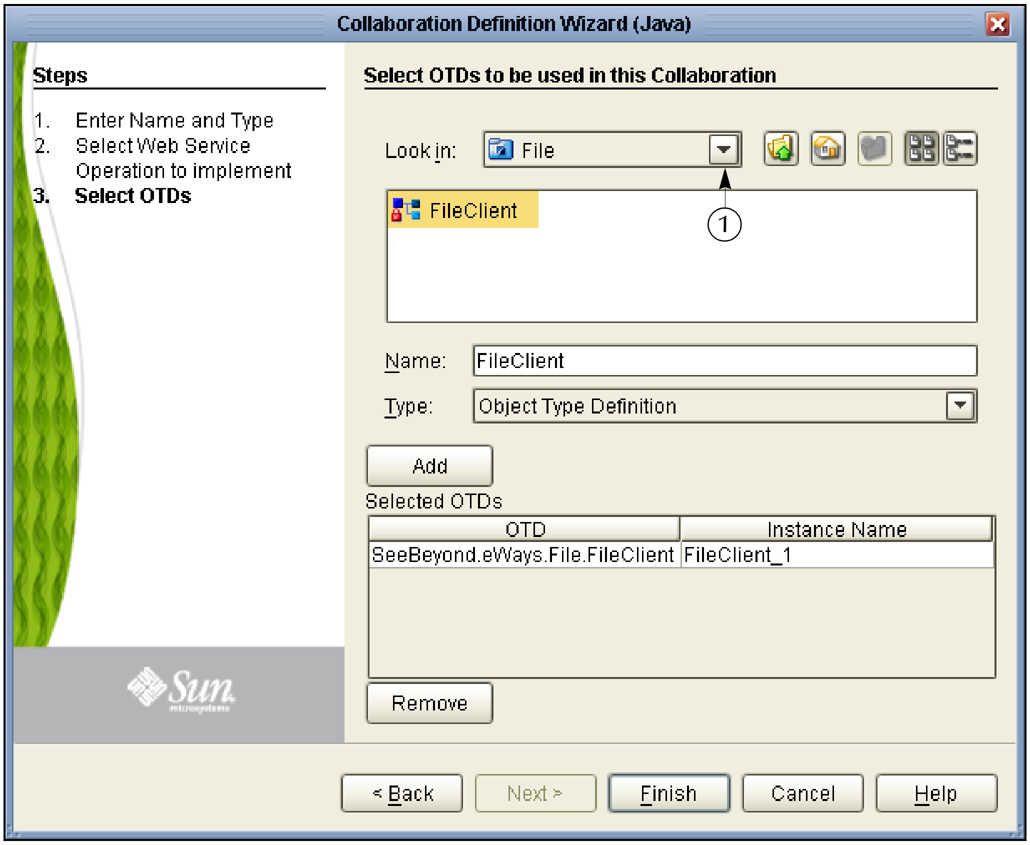
Selected OTDs

OTD	Instance Name

1. Double-click **Sun SeeBeyond**.
2. Double-click **eWays**.
3. Double-click **File**.
4. Double-click **FileClient**.

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

Figure 22 Select Employee 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:
File
FileClient
1
Add
Selected OTDs
Remove
< Back
Next >
Finish
Cancel
Help

Name: FileClient

Type: Object Type Definition

OTD	Instance Name
SeeBeyond.eWays.File.FileClient	FileClient_1

Note: *Before clicking Finish, install the XML OTDs.*

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

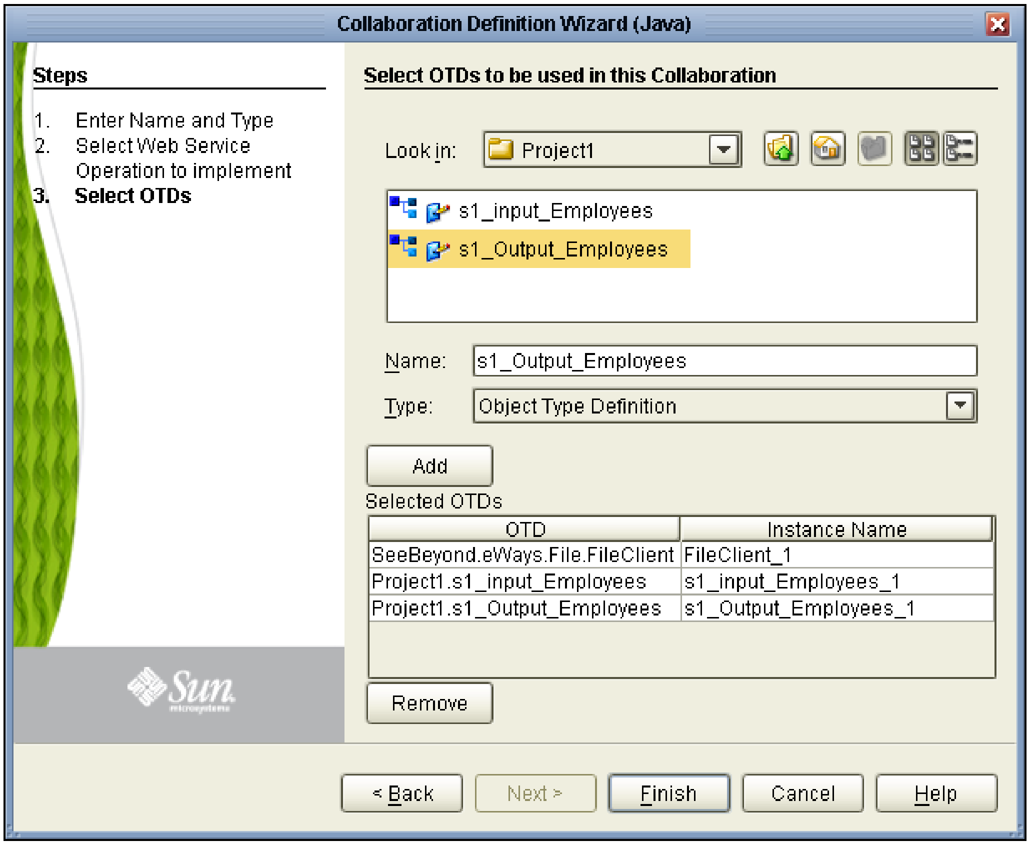
6 Click **Project1** to display the File OTDs.

7 Double-click **s1_input_Employees**.

8 Double-click **s1_output_Employees**.

See the following figure.

Figure 23 Selected 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: Project1

- s1_input_Employees
- s1_Output_Employees

Name: s1_Output_Employees

Type: Object Type Definition

Selected OTDs

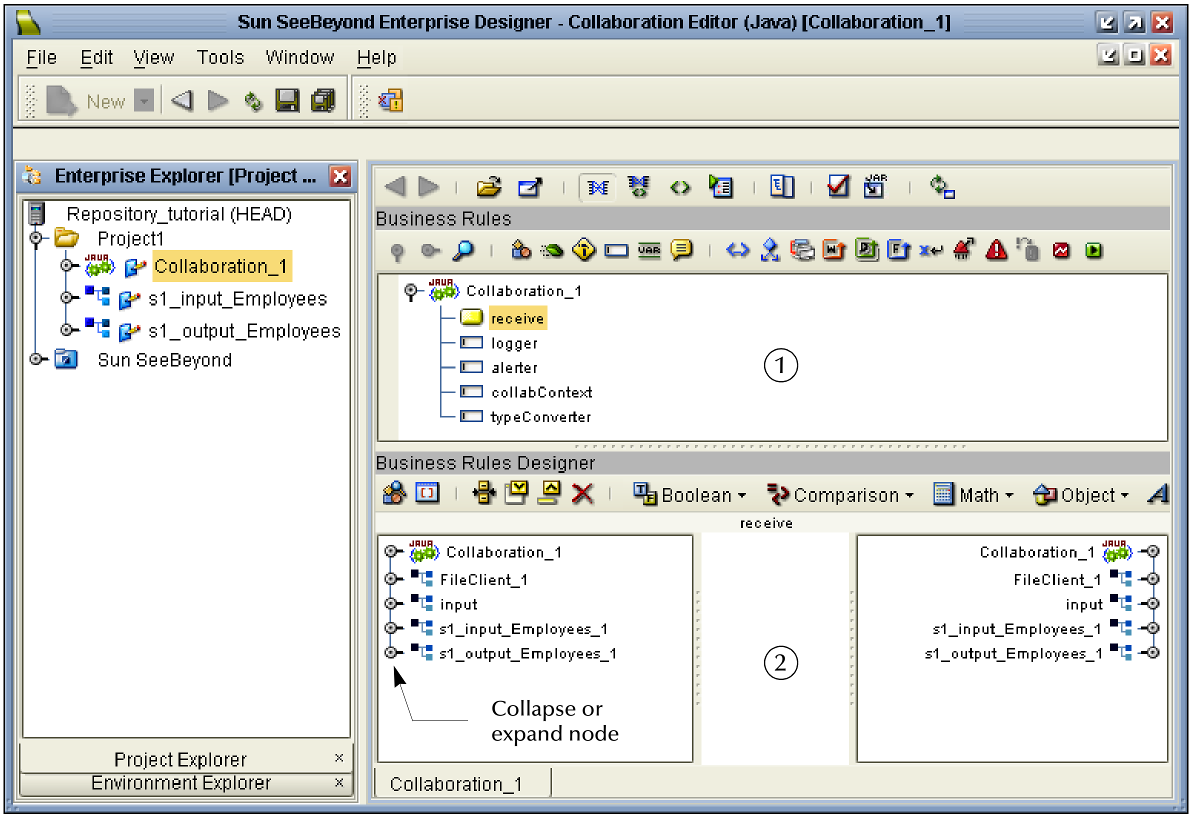
OTD	Instance Name
SeeBeyond.eWays.File.FileClient	FileClient_1
Project1.s1_input_Employees	s1_input_Employees_1
Project1.s1_Output_Employees	s1_Output_Employees_1

All three OTDs appear in the list of Selected OTDs.

9 Click **Finish**.

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

Figure 24 Collaboration Editor (Java)



3.9 Apply Business Rules

The Collaboration Editor (Java), Figure 24, is a full-featured Java source editor that assists the developer in creating Java Collaboration Definitions using a step-by-step wizard.

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

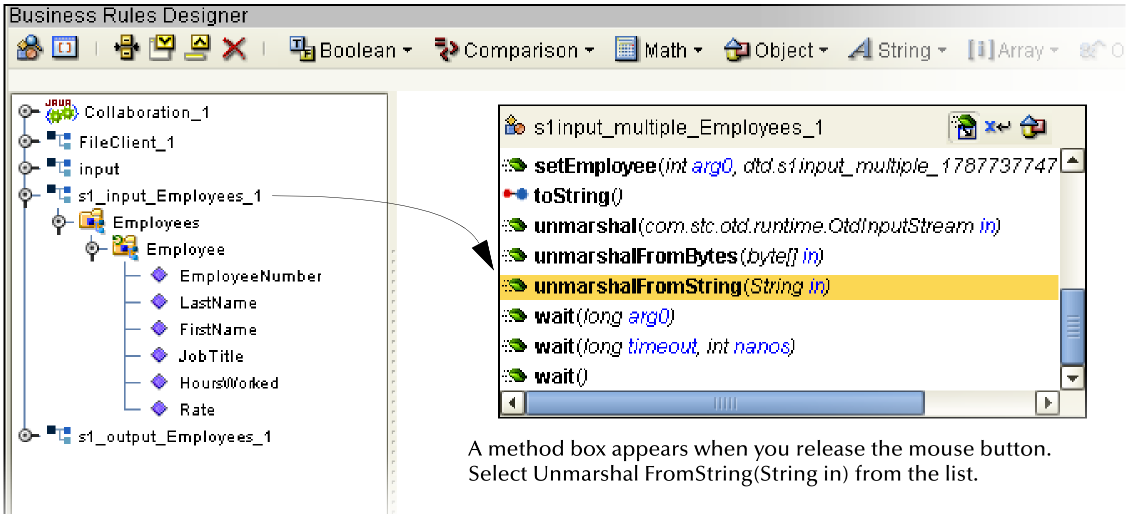
Unmarshal Text

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

1. Click the expansion node or double-click **input** to expand the view of **input**.
2. Expand the view for **s1_input_Employees_1** in the left pane of the Business Rules Designer, and expand the view for **Employee**.
3. Click and then drag **s1_input_Employees_1** from the left pane to the center pane of the Business Rules Designer (or right-click **s1_input_Employees_1** and click **Select Method to Call**).

   A method list appears.

Figure 25 Unmarshal Text



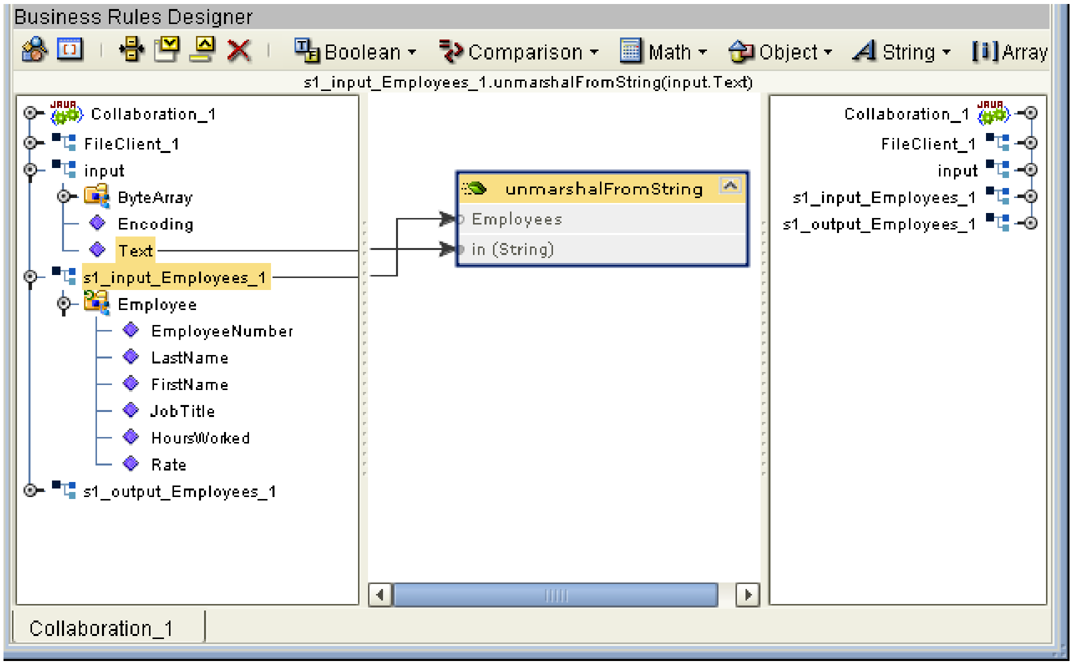
A method box appears when you release the mouse button.
Select Unmarshal FromString(String in) from the list.

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

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

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

Figure 26 Business Designer - Unmarshal Text



Connect Input and Output OTD Nodes

1 Expand the view of **s1_output_Employees_1** and **Employee**, in the right pane.

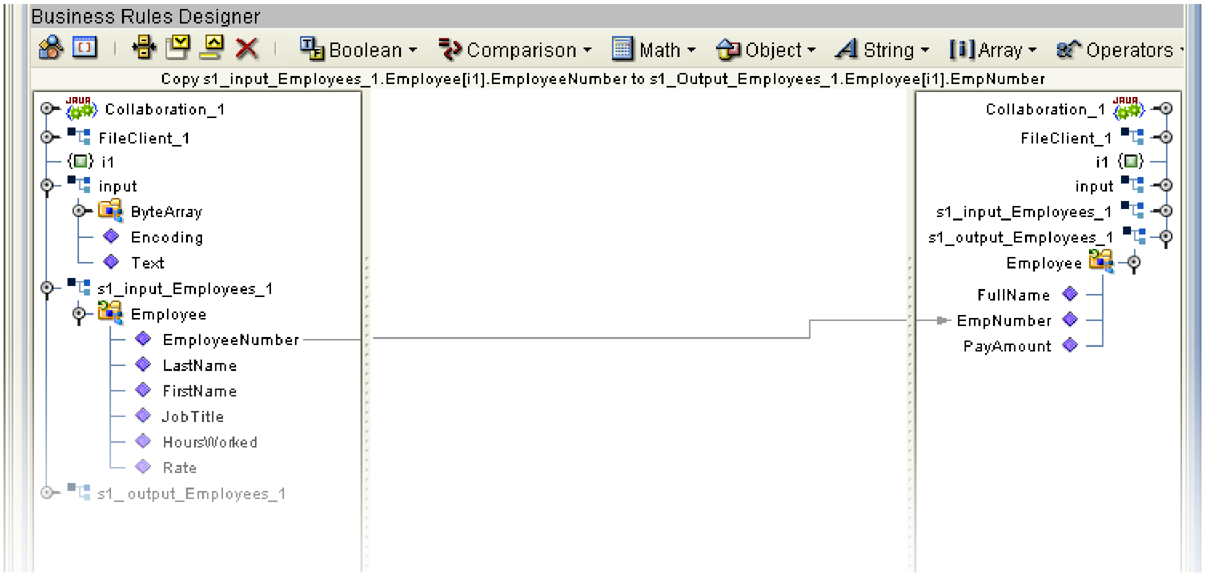
Note: *Notice that the objects in both the left and right pane of the Business Rules Designer window have the same name. Keep in mind that the OTDs (**FileClient_1**, **s1_input_Employees_1**, and **s1_output_Employees_1** for example) refer to the same object whether they are in the left or right pane.*

2 Connect **EmployeeNumber**, from **s1_input_Employees_1** in the left pane to **EmpNumber** of **s1_output_Employees_1** in the right pane.

This action connects the nodes and also automatically sets up the loop logic to process multiple records. See Figure 27.

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

Figure 27 Employee Number Node


Copy s1_input_Employees_1.Employee[i1].EmployeeNumber to s1_Output_Employees_1.Employee[i1].EmpNumber

3 Drag the **Concat** icon down to your Business Rules Designer workspace. (**Concat** is on the **String** drop-down menu. See Figure 28.)

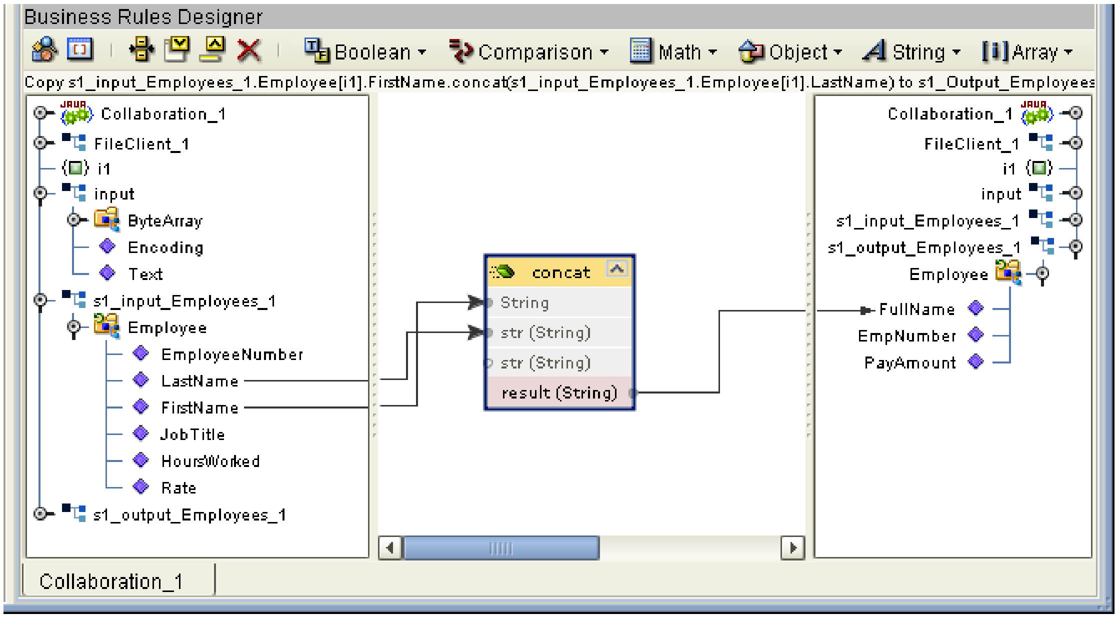
4 Connect **FirstName** to **String** in the Concat box. (Connect from **s1_input_Employee_1** in the left pane.)

5 Connect **LastName** to **str (String)** in the Concat box. (Connect from **s1_input_Employee_1** in the left pane.)

6 Connect **result (String)** to **FullName** in **Employee** of the root element **s1_output_employees_1** in the right pane.

See the following figure.

Figure 28 Concatenation Logic



Copy s1_input_Employees_1.Employee[i1].FirstName.concat(s1_input_Employees_1.Employee[i1].LastName) to s1_Output_Employees

3.9.1 Multiplication Logic

In this section set up the multiplication logic (Business Rules) to obtain an amount (**PayAmount**) for the output file.

1 Click the Math drop down list and select the Multiply operation.

The Multiply box appears in the Business Rules Designer pane.

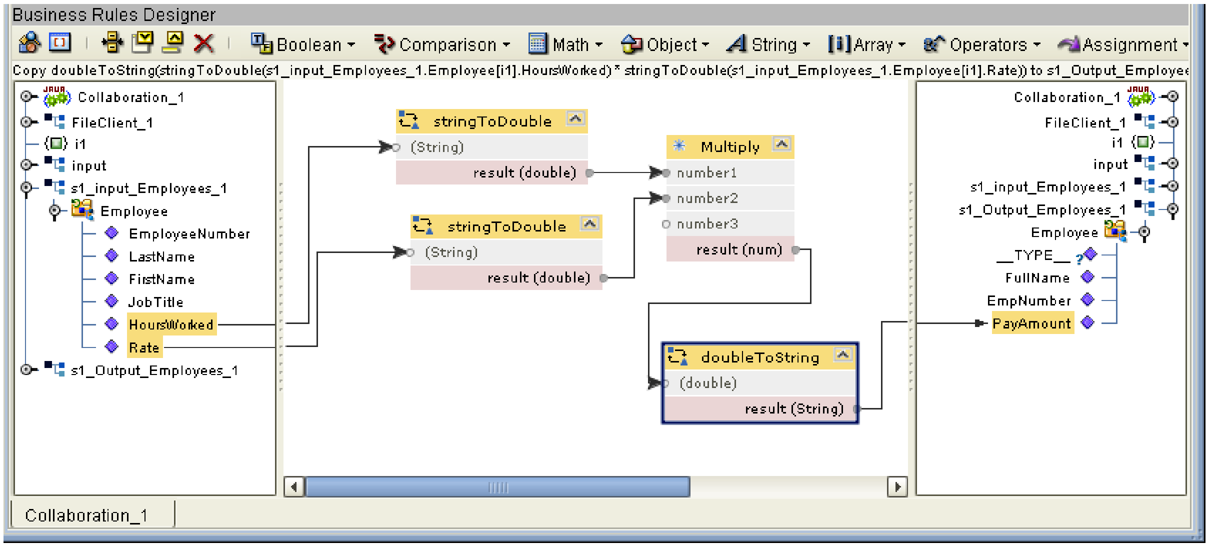
2 Map **HoursWorked** to **number1** and **Rate** to **number2** in the dialog box.

Note: *Accept the defaults when the String to Number Conversion dialog boxes appear.*

3 Map **result(num)** to **PayAmount** in the output OTD on the right.

See the following figure.

Figure 29 Multiplication Logic



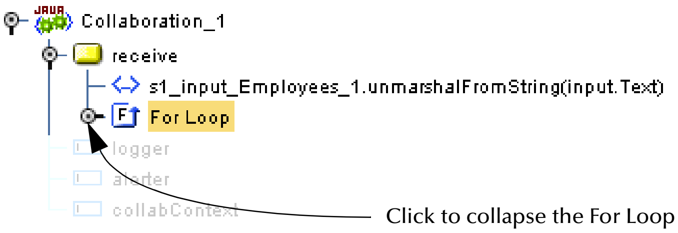
Copy doubleToString(stringToDouble(s1_input_Employees_1.Employee[i1].HoursWorked) * stringToDouble(s1_input_Employees_1.Employee[i1].Rate)) to s1_Output_Employee

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

Closing the “For Loop” sets the context to the same level as the for loop, instead of inside the loop. All new rules are generated below the context. This will soon include the **marshalToString** and **write** methods.

See the following figure.

Figure 30 Collapse For Loop

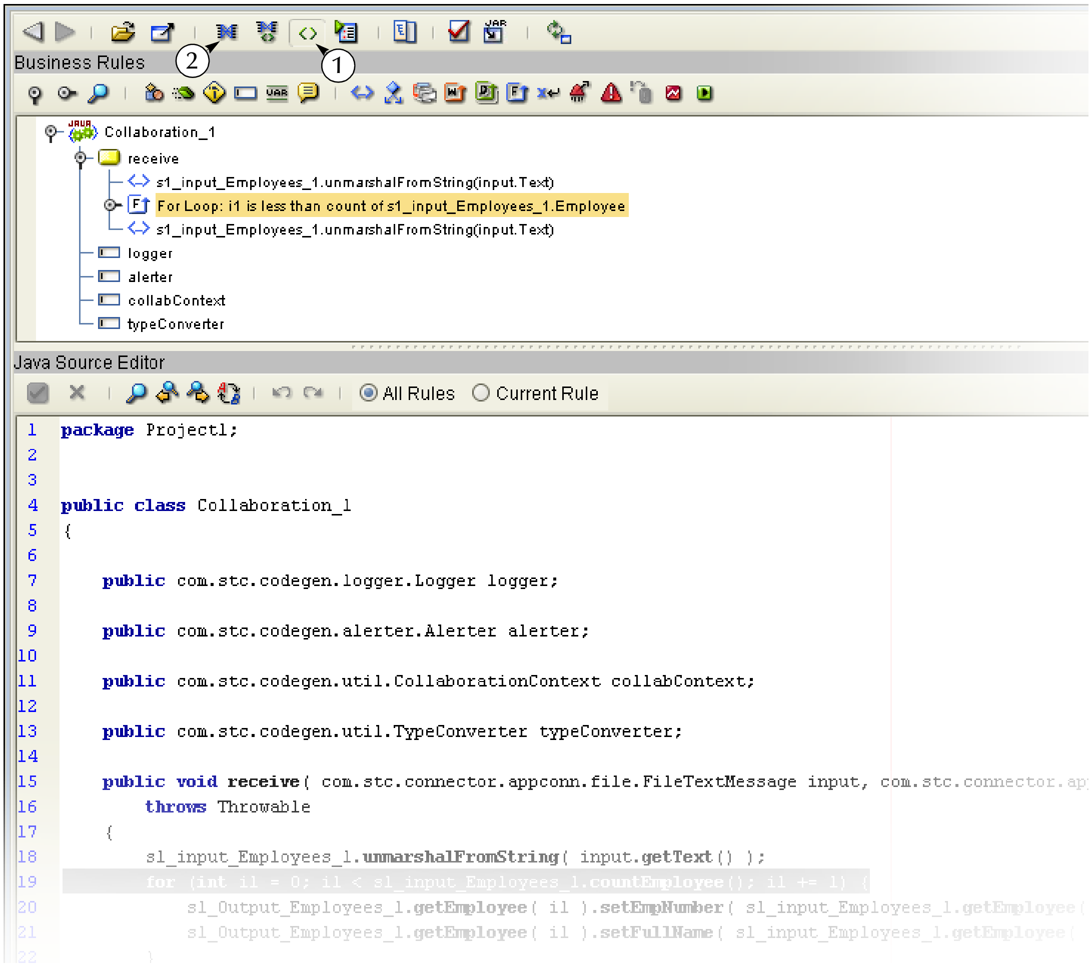


Click to collapse the For Loop

View Source Code

You may view or modify the source code by clicking the **Source code mode** button (number 1 in Figure 31. You may toggle back to Standard mode by clicking the **Standard mode** button (number 2 in Figure 31).

Figure 31 Source Code Mode


2
1
Business Rules
Collaboration_1
receive
s1_input_Employees_1.unmarshalFromString(input.Text)
For Loop: i1 is less than count of s1_input_Employees_1.Employee
s1_input_Employees_1.unmarshalFromString(input.Text)
logger
alerter
collabContext
typeConverter
Java Source Editor
● All Rules
○ Current Rule

```
package Project1;


public class Collaboration_1
{

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

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

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

    public com.stc.codegen.util.TypeConverter typeConverter;

    public void receive( com.stc.connector.appconn.file.FileTextMessage input, com.stc.connector.ap
        throws Throwable
    {
        s1_input_Employees_1.unmarshalFromString( input.getText() );
        for (int i1 = 0; i1 < s1_input_Employees_1.countEmployee(); i1 += 1) {
            s1_Output_Employees_1.getEmployee( i1 ).setEmpNumber( s1_input_Employees_1.getEmployee(
            s1_Output_Employees_1.getEmployee( i1 ).setFullName( s1_input_Employees_1.getEmployee(
        }
```

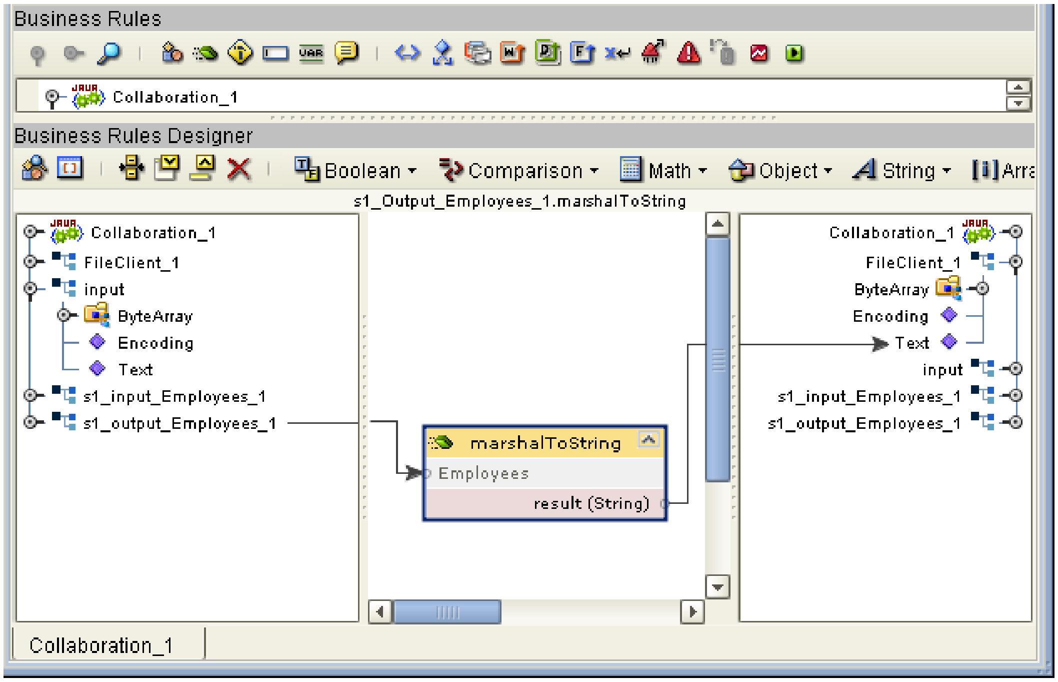
Marshal Text

In the following steps you prepare to “write” your output data.

1. Expand the view of **FileClient_1** in the right pane.
2. Click and then drag **s1_output_Employees_1** from the left to the center pane of the Business Rules Designer (or *right-click **s1_output_Employees_1** and **Select method to call***).
3. Scroll down then double-click the method **marshalToString()** from the pop up list.

   The **marshalToString** method box appears.
4. Connect a node from **result (String)** to **Text** under **FileClient_1** in the right pane.

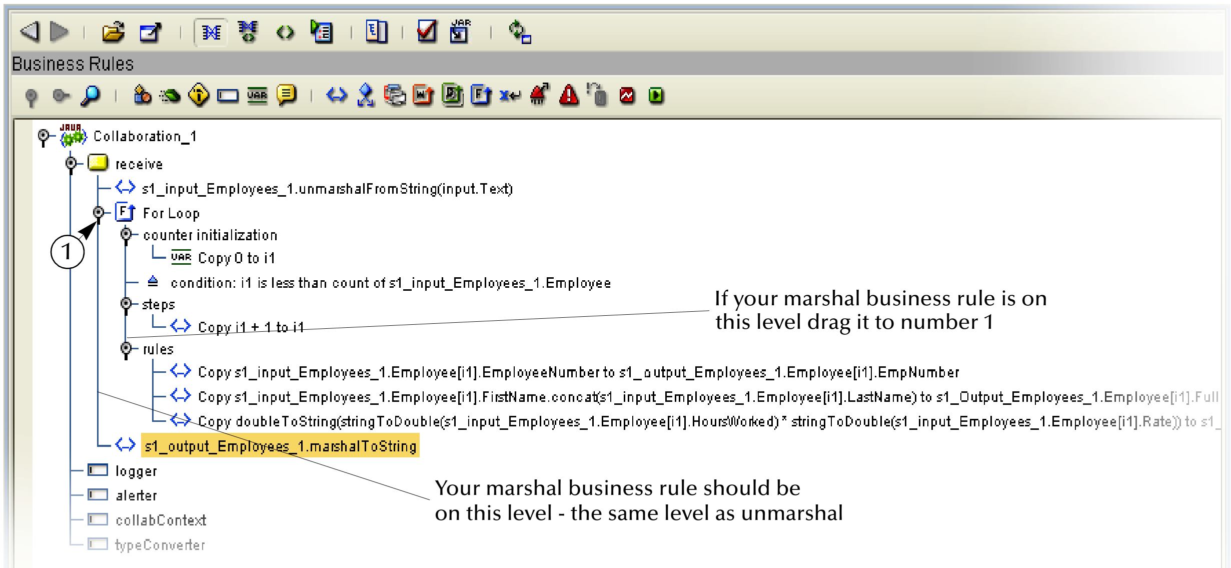
Figure 32 Marshal to String



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

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

Figure 33 View Business Rules



```
Collaboration_1
  receive
    s1_input_Employees_1.unmarshalFromString(input.Text)
    For Loop
      counter initialization
        Copy 0 to i1
      condition: i1 is less than count of s1_input_Employees_1.Employee
      steps
        Copy i1 + 1 to i1
      rules
        Copy s1_input_Employees_1.Employee[i1].EmployeeNumber to s1_output_Employees_1.Employee[i1].EmpNumber
        Copy s1_input_Employees_1.Employee[i1].FirstName.concat(s1_input_Employees_1.Employee[i1].LastName) to s1_Output_Employees_1.Employee[i1].Full
        Copy doubleToString(stringToDouble(s1_input_Employees_1.Employee[i1].HoursWorked) * stringToDouble(s1_input_Employees_1.Employee[i1].Rate)) to s1_
    s1_output_Employees_1.marshalToString
  logger
  alerter
  collabContext
  typeConverter
```

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

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

Write FileClient_1

Write to the output file.

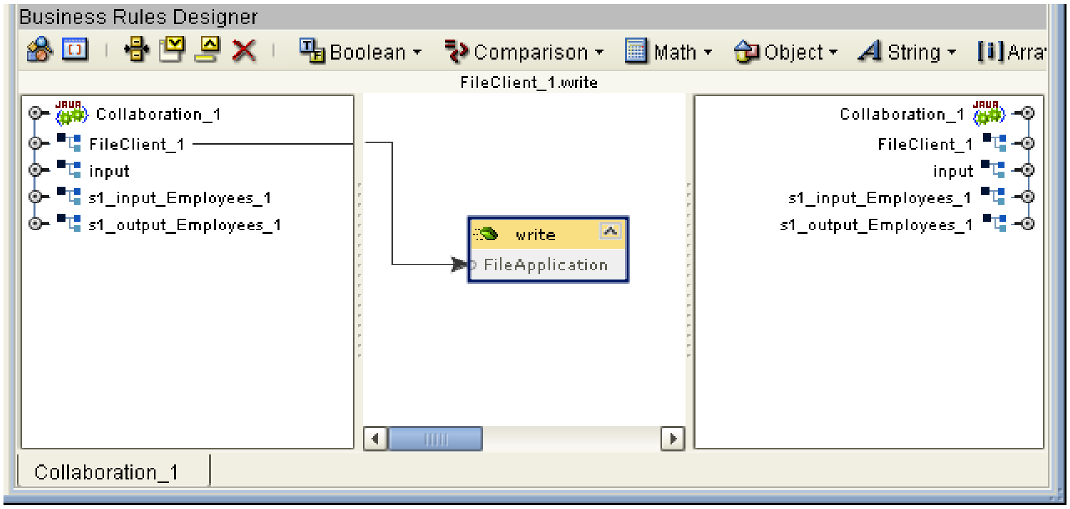
1 Click and then drag **FileClient_1** from the left pane to the center pane of the Business Rules designer.

  A list of methods appears.

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

  See the following figure.

Figure 34 Write Output File



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

  This completes the Collaboration. In the following section you will use the drag-and-drop method to move the Collaboration into your Service.

3.10 Create a Connectivity Map

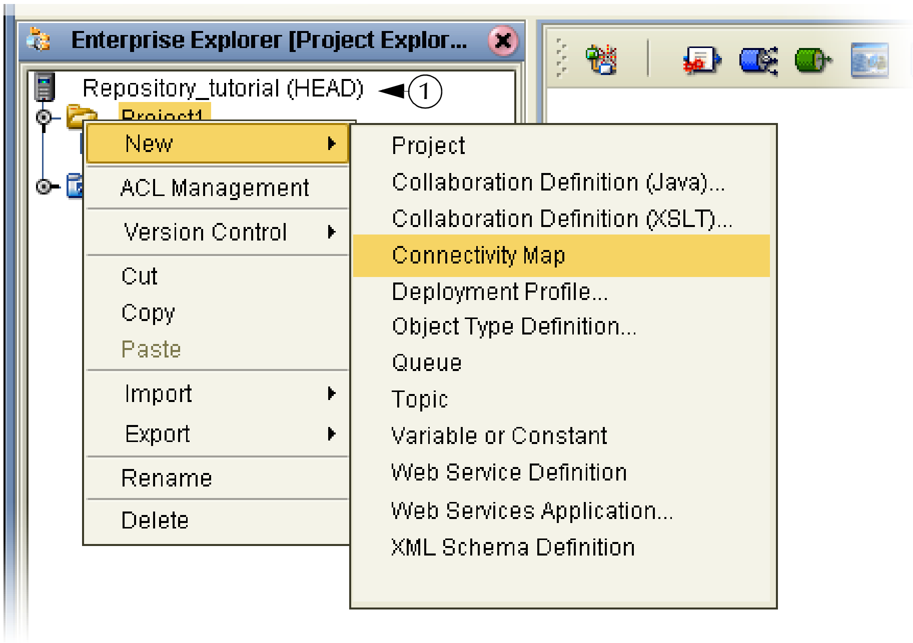
Project1 needs a Connectivity Map to define the flow of data between the TimeCard system and the work file output. The Connectivity Map is a graphical representation (GUI) of the data connectivity, including the publish/subscribe information for data routing.

Name Connectivity Map

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

1 In the Enterprise Explorer, right-click the **Project1** icon. Click **New, Connectivity Map**. (You can also select New Connectivity Map from the drop-down menu.)

Figure 35 Create a Connectivity Map

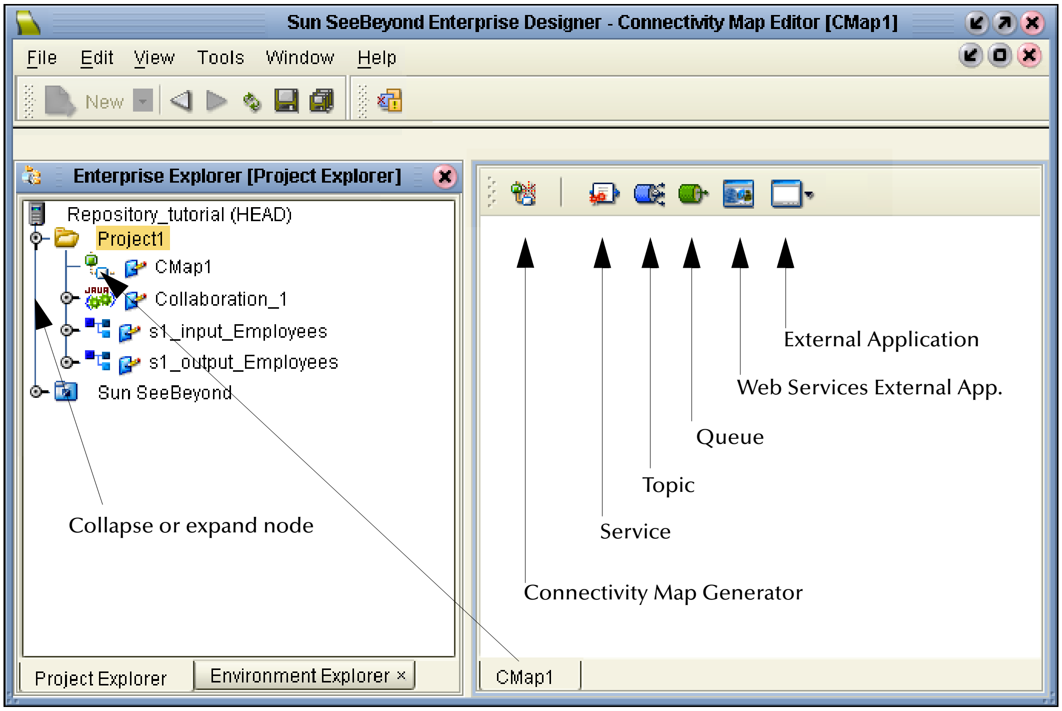


You may give your Connectivity Map a name. In this Tutorial, accept the default name, **CMap1**.

CMap1 is added to your Project Explorer tree as shown in Figure 36.

Note: *The word HEAD (number 1 in Figure 35) is appended to the Repository name. This is the default branch and is used for version control when developing and testing your project. For this scenario the default version control name is OK.*

Figure 36 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.*

3.10.1 Populate the Connectivity Map

In this section, add the objects (and runnable components such as a Java Collaboration) to the Connectivity Map.

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

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

Get Input Object

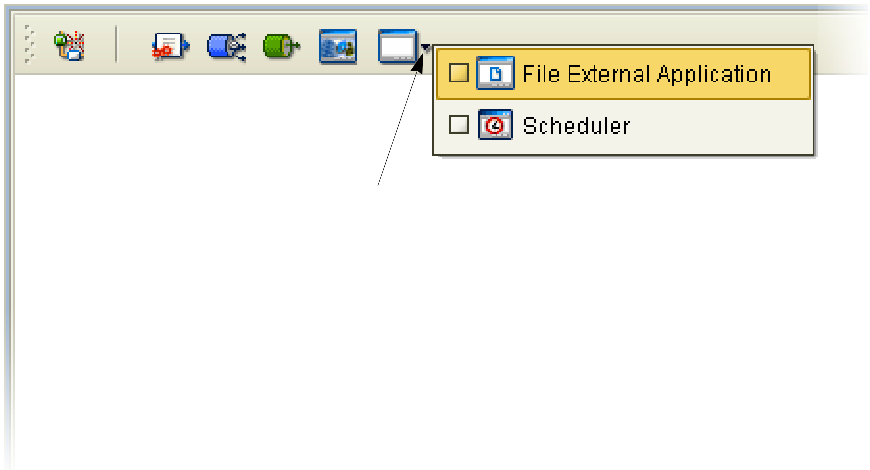
Place an input file (External Application) in your Connectivity Map.

1. On the Connectivity Map toolbar, click the External Application (shown in Figure 37) to display the list. (This is the icon with the “drop-down” arrow next to it.)
2. Select the **File External Application** from the list.

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

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

Figure 37 External Application Selection



3 Click and drag the **File** icon from the toolbar into the left side of the Connectivity Map.

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

Note: *In this scenario accept the default names.*

Get Service Component

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

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

This enables the use of a new Service called **CMap1_Service1**.

Note: *To reposition an object or component in the Connectivity Map, click the icon and drag it to a new location.*

Get Output Object

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

1 Drag another **File** icon into the Connectivity Map, and place it to the right of the **CMap1_Service1** icon.

This adds a new External Application File.

2 Click **Save** or press CTRL+S.

3.11 Apply the Collaboration

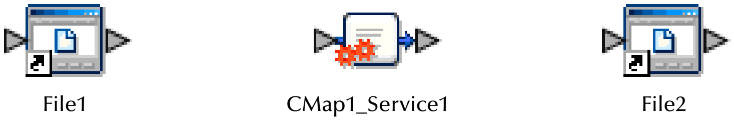
You have mapped your Business Rules and are now ready to bind your Java Collaboration Definition (JCD) Service, using the drag-and-drop method.

3.11.1 Link Objects in the Connectivity Map

Linking the objects and components in your Connectivity Map creates a logical flow of data through your Project. This linking of files and Services also adds eWays to the Connectivity Map.

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

Figure 38 Objects in Connectivity Map

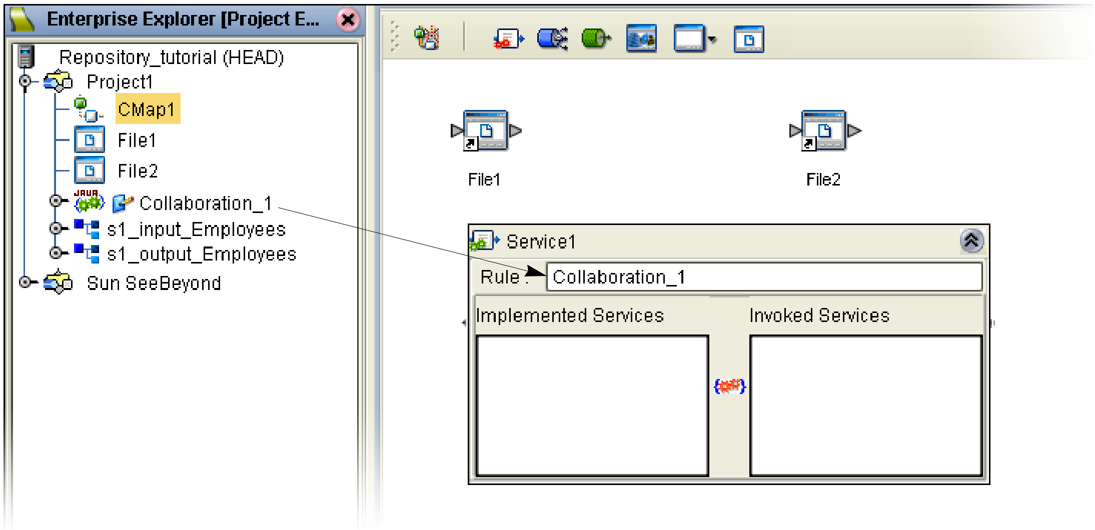


Connect Nodes

1. Double-click **CMap1_Service1** in the Connectivity Map to open the Service box. (You can also reposition the box after it is open.)
2. Click and drag the **Collaboration_1** definition from the Project Explorer tree to the **Rule** field in the open Service box as shown in the following Figure 39.

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 Service1 component without first opening the Service1 box.*

Figure 39 Using the Drag and Drop method



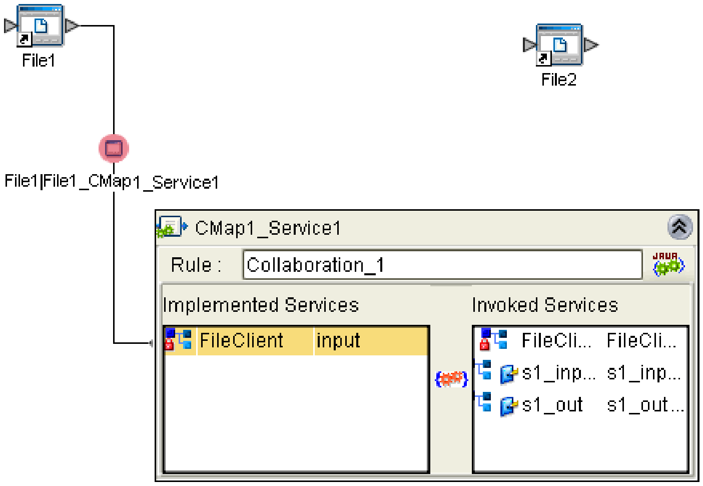
Rule : Collaboration_1

Notice that the gear icons change from red to green to signify that the Collaboration is bound to the Service (See Figure 40).

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

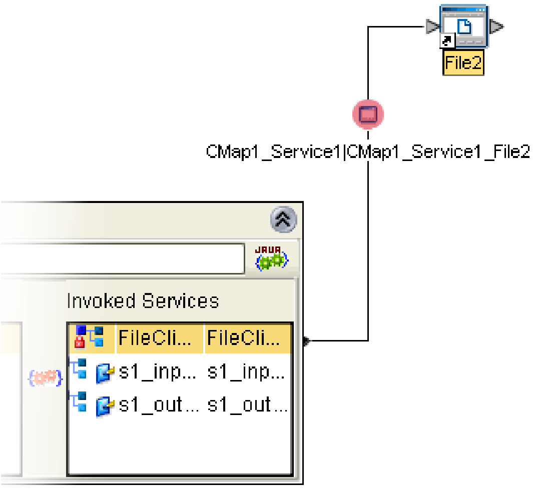
3 Drag a node to connect **FileClient** input to **File1** as shown in the following figure.

Figure 40 Connect FileClient to File1



4 Drag a node to connect **FileClient_1** to **File2** as shown in the following figure.

Figure 41 Connect FileClient to File2



5 Click the **Minimize** button to close the expanded view of the Service1 box.

6 Click **Save All**.

3.12 Configure the eWays

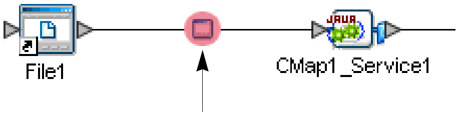
Scenario1 requires the following eWay configurations:

- 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 in the Project1 folder. For example: **C:\eGateData\Project1**.

Configure Inbound eWay

Set the properties to identify the file and its location.

Figure 42 Inbound eWay

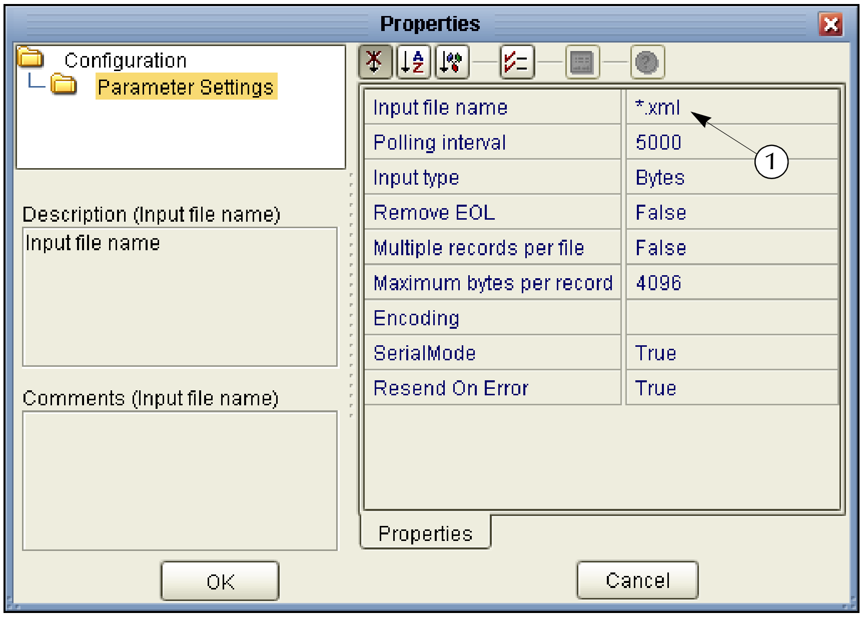


1. Double-click the first eWay (shown in the figure above).

   The input file name (number 1 in Figure 43) is an editable field.

Note: *In the following properties configurations for the inbound and outbound files, accept the default settings for everything except the file names.*

Figure 43 Properties Configuration (Inbound eWay)



Description (Input file name)
Input file name

Comments (Input file name)

Input file name	*.xml
Polling interval	5000
Input type	Bytes
Remove EOL	False
Multiple records per file	False
Maximum bytes per record	4096
Encoding	
SerialMode	True
Resend On Error	True

2. Click the **Input File Name** field and enter ***.xml**. The eWay reads any files with the **.xml** extension.
3. Click **OK** to close the Properties Dialog Box for the inbound eWay.

After the inbound eWay is set, the red circle in the graphic disappears.

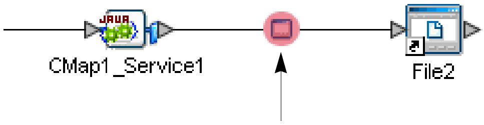
Note: *If the* **Polling interval** *value is left unchanged, the eWay polls the directory location every five seconds (5000 milliseconds).*

Configure Outbound eWay

Set the properties to identify the file and its location.

1 Double-click the second eWay.

Figure 44 Outbound eWay

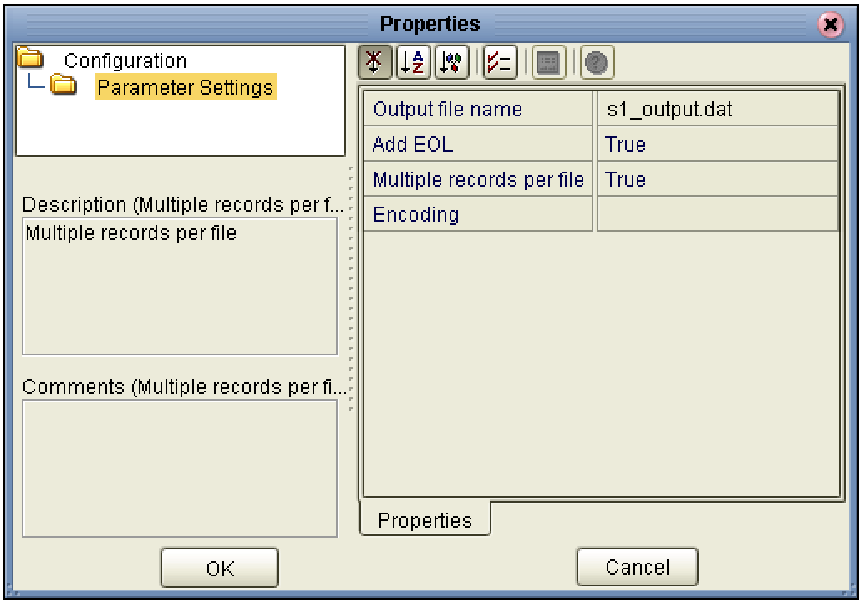


The Properties dialog box appears.

2 Name the output File: **s1_output.dat** (Scenario1 output).

After the outbound eWay is set, the link (red circle) in the graphic disappears.

Figure 45 Properties Configuration (Outbound eWay)


Properties
Configuration
Parameter Settings
Description (Multiple records per f...
Multiple records per file
Comments (Multiple records per fi...
Properties
OK
Cancel

Output file name	s1_output.dat
Add EOL	True
Multiple records per file	True
Encoding	

Note: *Refer to* **"Multiple records per File (True or False)" on page 59**.

3 Click **OK** to close the Properties Dialog Box for the outbound eWay and accept the default settings for the remaining properties.

4 On the **File** toolbar, click **Save All**.

You have now configured both the input and output eWays and are ready to deploy your Project. You may skip to **"Create an Environment" on page 59**.

Multiple records per File (True or False)

This setting (refer to Figure 43 and Figure 45) specifies if multiple records are to be obtained or sent per file. Multiple records (messages) are generated per line up to the maximum bytes per record.

Inbound eWay

A **True** setting means that each separate line in the input file is treated as a separate message. Messages are generated up to the number specified in the Maximum bytes per record property.

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

Outbound eWay

This setting determines whether records are concatenated in a file. If no incrementer is used, the output file is overwritten with each message.

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

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

3.13 Create an Environment

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

Create a Logical Host and an External File System using Environment Explorer. First create an Environment. The Environment Explorer deploys resources required to implement a project and includes information about external systems that interact with eGate.

Add Logical Host and Servers

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

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

Add an Integration Server

The Integration Server is a J2EE software platform that houses the business logic container used to run Collaborations and JCA connectors (eWays).

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

   IntegrationSvr1 appears in the **LogicalHost1** box. Your Collaborations will be bound to this server.

Add a JMS IQ Manager

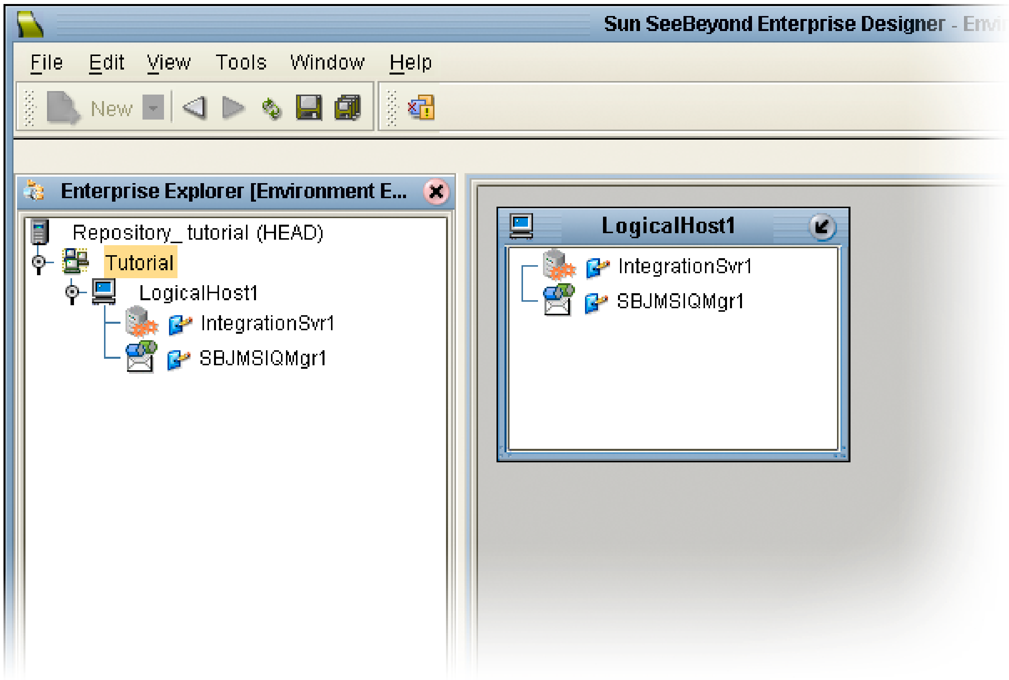
The JMS IQ Manager is a JMS-compliant, guaranteed delivery store, forwarding, and queueing Service.

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

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

   SBJMSIQMgr1 appears in the **LogicalHost1** box.

Figure 46 Environment with Logicalhost



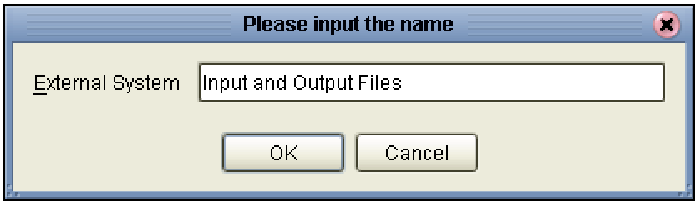
Add an External File System for the eWays

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

   This creates a container in the Environment Editor to hold your File eWays. You will place both input and output files in this container.

3 Name the External System **Input and Output files**.

Figure 47 External files



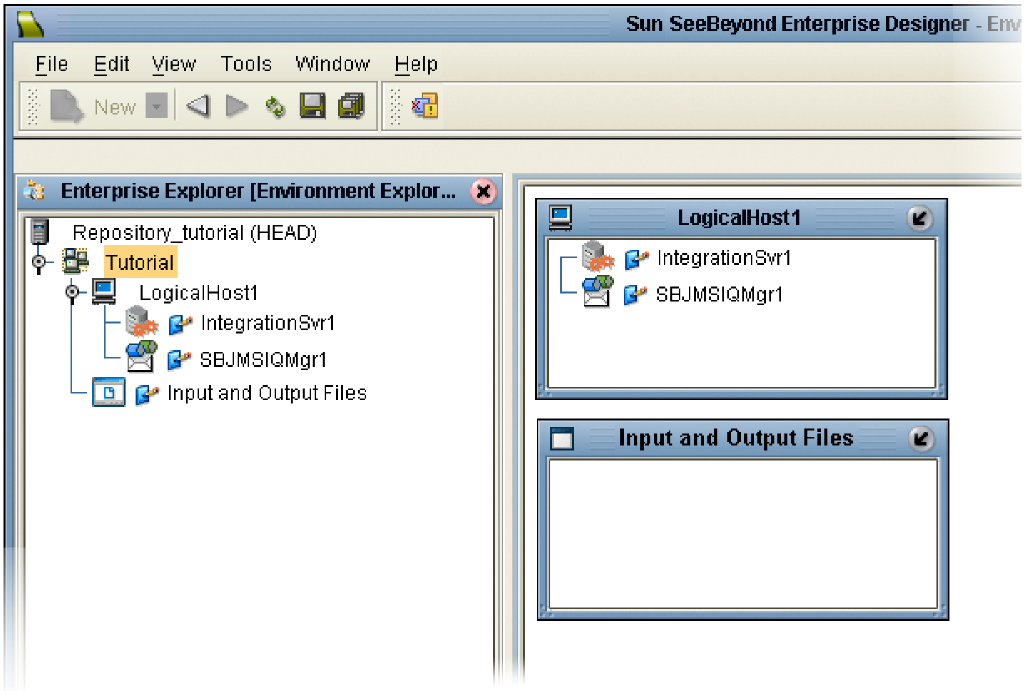
External System: Input and Output Files

4 Click **OK**.

Note: *You may have to move a dialog (GUI) if it is blocking an underlying dialog in the designer canvas.*

Your Environment Editor pane should look similar the following figure.

Figure 48 Environment Editor

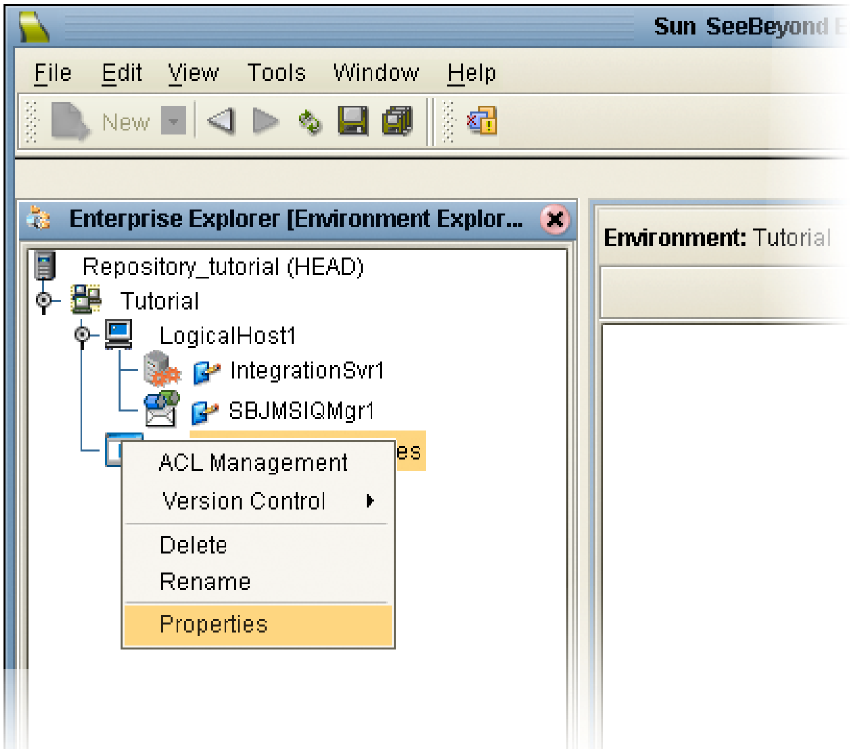


Set Properties for External File System

Set the properties and parameter settings to show the directory data path.

1 Right-click **Input and Output files** in the left pane.

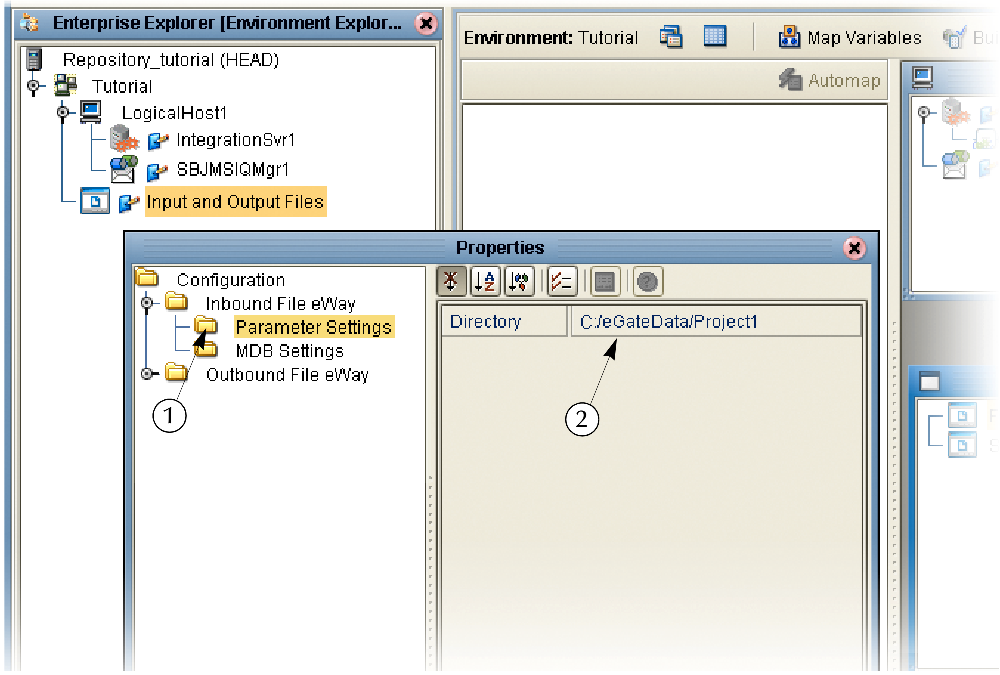
Figure 49 Properties Pop-up Box



2 Select **Properties**.

The Properties dialog box appears.

Figure 50 Properties, Parameter Settings



Directory	C:/eGateData/Project1

3. Expand the view under **Inbound File eWay**.
4. Select **Parameter Settings** (number 1 in the previous figure).
5. Set the Directory path to point to your input data File (number 2 in the previous figure).
6. Click **OK**.
7. Repeat these steps for your **Outbound File eWay**.
8. Click **Save**.

Note: *Enter the directory paths only, not including the file name. The output data file does not have to already exist. It will be created.*

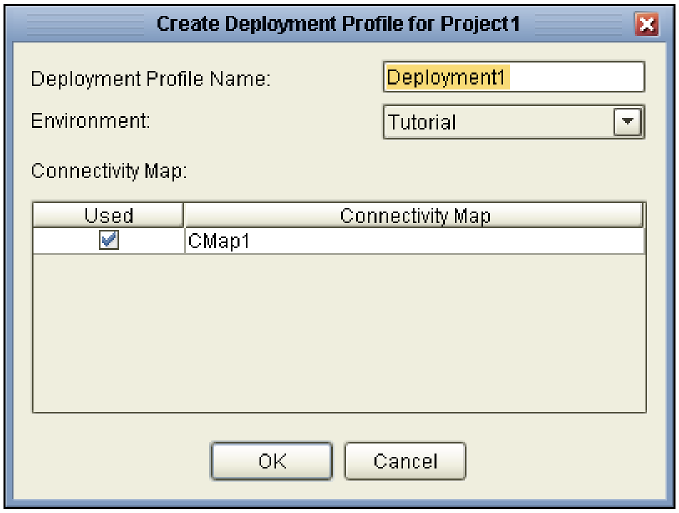
3.14 Create the Deployment Profile

A Deployment Profile contains information about how Project components are mapped and deployed within an Environment.

Create a Deployment Profile

1. Click the **Project Explorer** tab to return to the Project Explorer pane of your Project.
2. Right-click **Project1**, and then click **New**, **Deployment Profile**.

Figure 51 Deployment Profile



Deployment Profile Name: Deployment1

Environment: Tutorial

Used	Connectivity Map
☑	CMap1

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

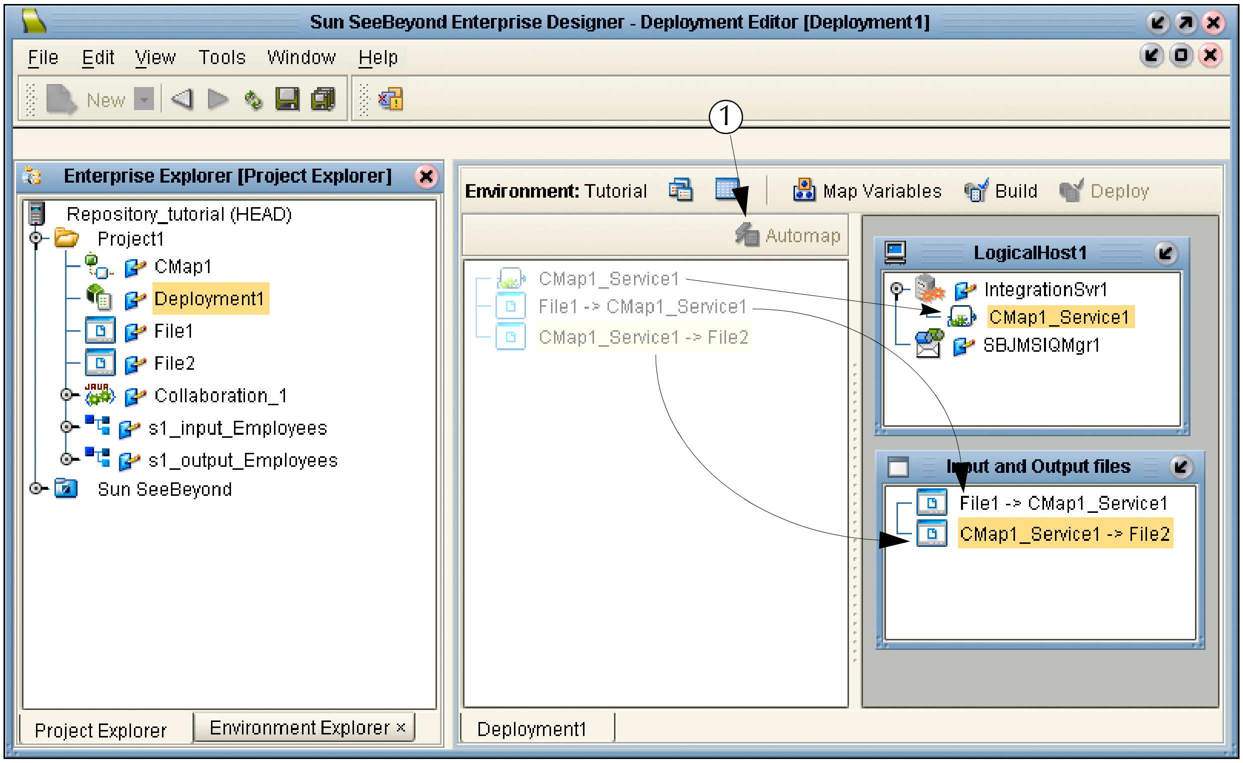
   Your files and Service1 appear in the center pane of the Environment editor. Deploy these objects and components using the drag and drop method.

4. Drag-and-drop your Service, **CMap1_Service1**, into your Integration Server, **IntegrationSvr1**. See the following figure.

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

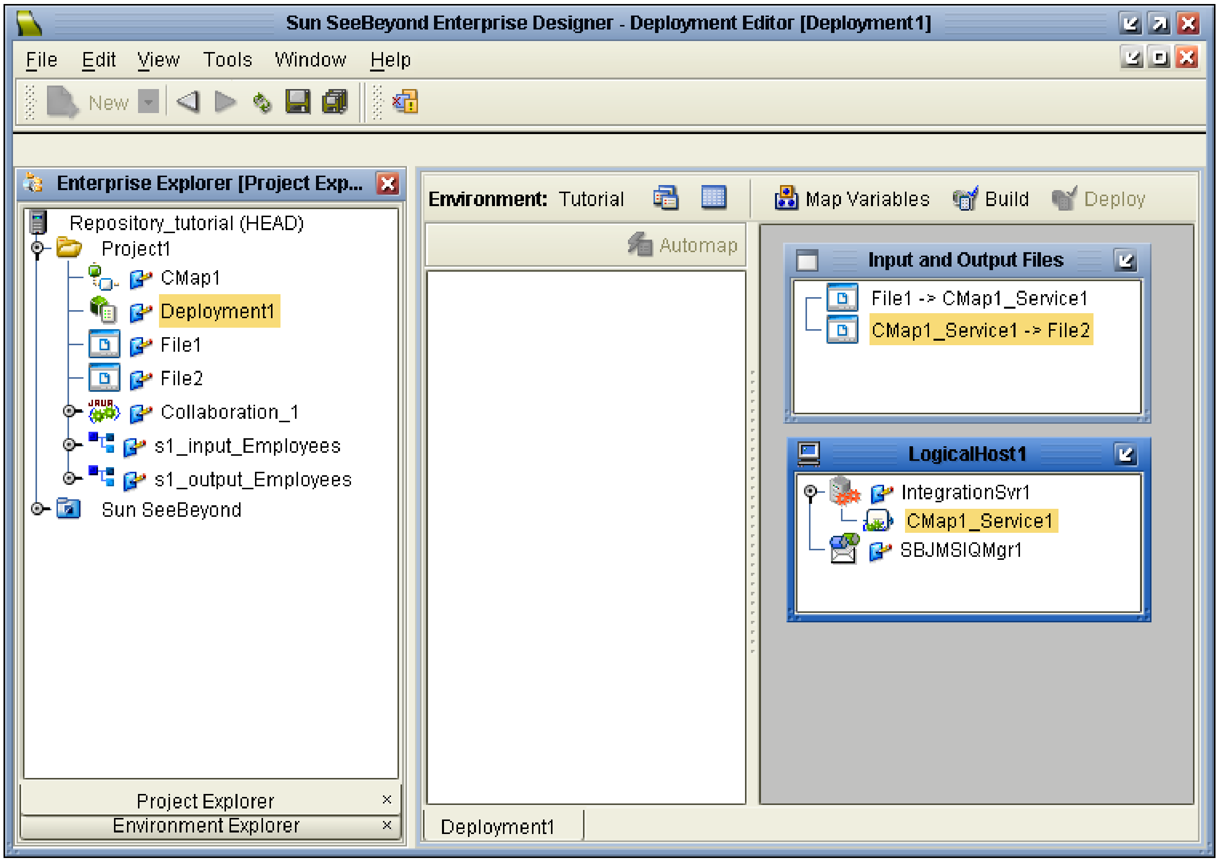
Note: *You can also use **Automap**, (number 1 in the following figure).*

Figure 52 Environment, Deployment Editor



The center pane of the Environment Editor should be empty after all available objects and components have been mapped to the Environment, as shown in Figure 53.

Figure 53 Populated Environment



You are now ready to build and deploy your project.

6 Save your Project.

3.15 Build and Deploy the Project

In this section you will run your Project.

3.15.1 Create a Domain

Make sure an instance of the Logical Host is running (domain) before you deploy your project. You can start the domain with the **domainmgr.bat** script that is located in your Java CAPS folder, **logicalhost** directory.

The start script is only present after you have created/installed the domain. The Domain Manager will prompt you to create a domain if no domain is present.

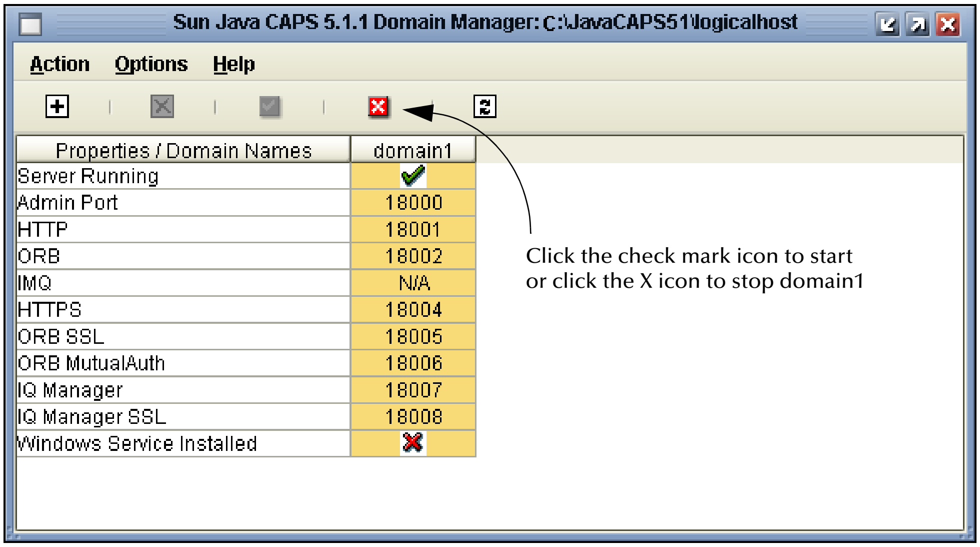
See the Deploying Applications section of the *Sun SeeBeyond eGate Integrator System Administration Guide*.

1 Double-click **domainmgr.bat** to launch the Domain Manager.

Use the Domain Manager GUI to start your Domain, or double-click **start_domain1.bat** (in the same directory) to run domain1.

2 Accept the default Properties/Domain Names.

Figure 54 Start Domain Manager


Sun Java CAPS 5.1.1 Domain Manager: C:\JavaCAPS51\logicalhost
Action
Options
Help

Properties / Domain Names	domain1
Server Running	✓
Admin Port	18000
HTTP	18001
ORB	18002
IMQ	N/A
HTTPS	18004
ORB SSL	18005
ORB MutualAuth	18006
IQ Manager	18007
IQ Manager SSL	18008
Windows Service Installed	✗

Click the check mark icon to start
or click the X icon to stop domain1

See the *Deploying Applications* section of the *System Administration Guide*.

3.15.2 Enter Passwords and Set URLs

Make sure you have entered your user name and password (see the steps below) for the Integration Server (and the JMS IQ Manager), and make sure your Domain server is running.

Integration Server Password

1 Click the **Environment Explorer** tab.

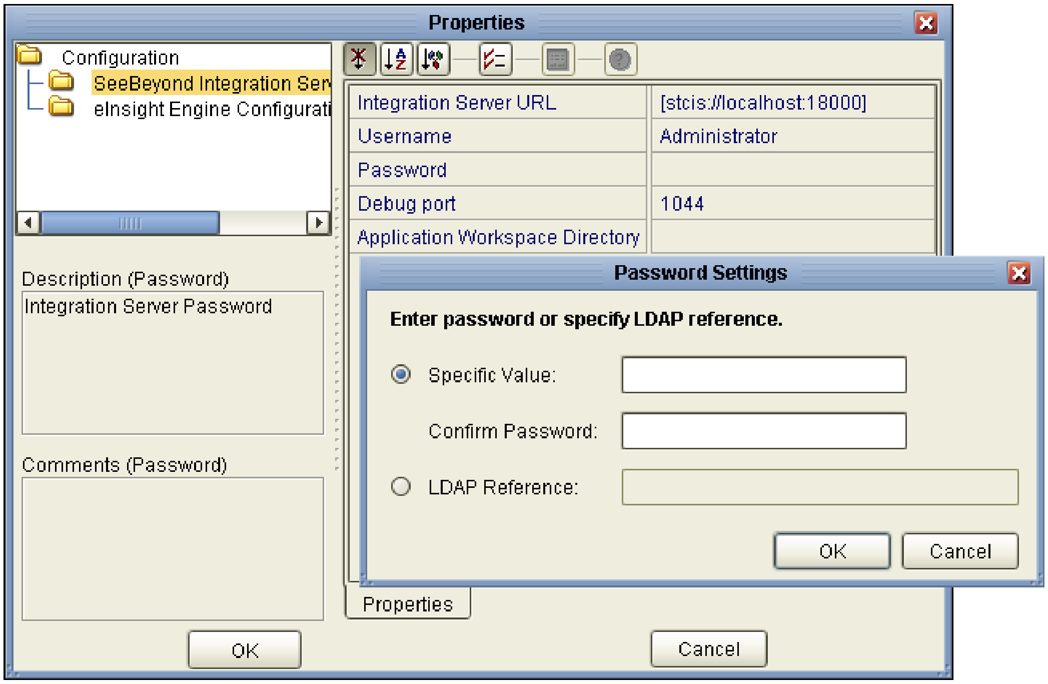
2 Right-click **IntegrationSvr1** in the Environment Explorer tree.

3 Click **Properties**.

4 Enter and confirm **Password**.

See the following figure.

Figure 55 Integration Server Password



Integration Server URL	[stcis://localhost:18000]
Username	Administrator
Password	
Debug port	1044
Application Workspace Directory	

Description (Password)
Integration Server Password

Comments (Password)

Enter password or specify LDAP reference.

● Specific Value:

Confirm Password:

○ LDAP Reference:

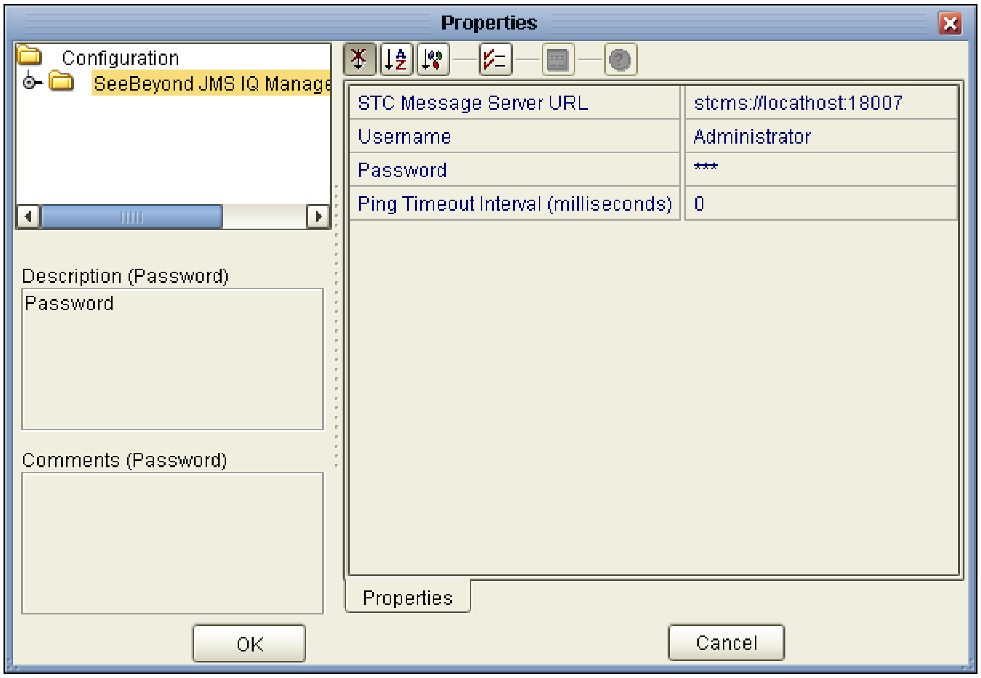
5. Also, verify the Username and Integration Server URL host and port number.
6. Click **OK**.

JMS IQ Manager Password

1. Similar to the previous steps for the Integration Server, enter a password for the JMS IQ Manager.
2. Set the URL for the message server for the IQ manager.

   See the following figure.

Figure 56 IQ Manager



Properties

Configuration
SeeBeyond JMS IQ Manage

STC Message Server URL	stcms://localhost:18007
Username	Administrator
Password	***
Ping Timeout Interval (milliseconds)	0

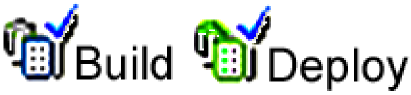
Description (Password)
Password

Comments (Password)

3.15.3 Build

The Build compiles the Service(s) and Java files, then creates the Project EAR File.

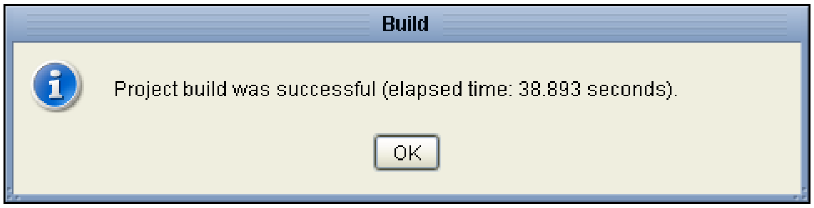
Figure 57 Build Button



1. Click the Build button in the Environment toolbar.

   After the Project EAR File is created the following message appears:

Figure 58 Project Build Successful



Build

Project build was successful (elapsed time: 38.893 seconds).

3.15.4 Deploy

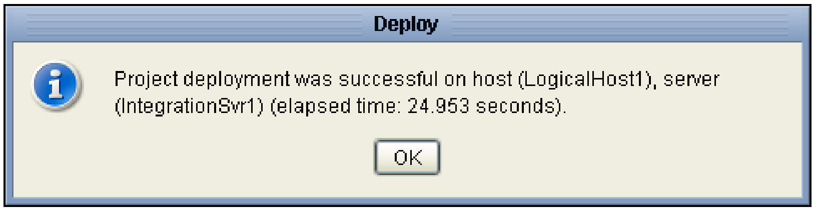
In this section, deploy your Project to start the Integration Server and start the deployment.

Note: *Before you can deploy your Project, an instance of the Logical Host (domain) must be running.*

1. Click the Deploy button.
2. Click **Yes**.

   You will see a message similar to the following, when the deployment is successful.

Figure 59 Deployment Successful



Project deployment was successful on host (LogicalHost1), server (IntegrationSvr1) (elapsed time: 24.953 seconds).

3. Click **OK**.

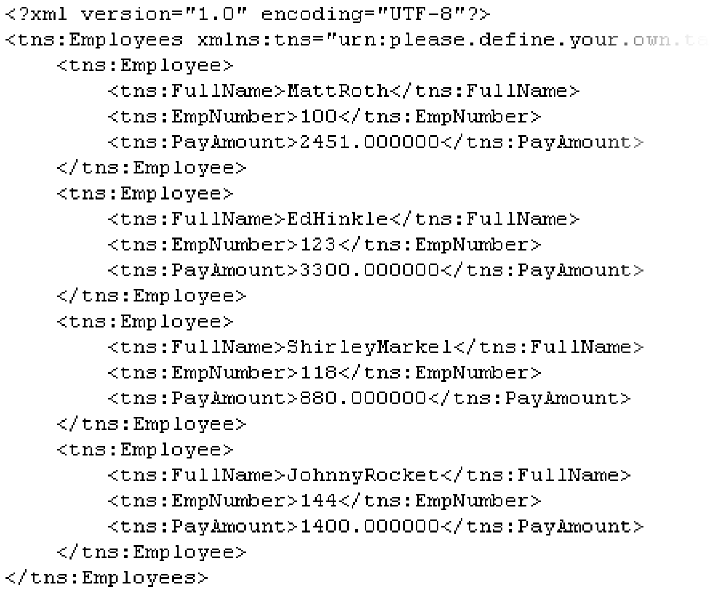
3.16 Verify the Output Data

The Project processes your input File, **TimeCard.xml**, and writes the results to **s1_output1.dat**.

View Output

1. Use Windows Explorer to navigate to **C:\eGateData\Project1**.
2. See that the inbound File eWay, after processing the input file, renamed the input file to **TimeCard.xml.~in**. You can make changes in this file using a text editor and then rename the file back to **TimeCard.xml** and immediately see the results of your changes after the file is processed again.
3. See that the outbound File eWay generated the output File, **s1_output1.dat**.
4. Use a text editor to view the contents of the output file. (In this scenario you accepted the default decimal positions so your actual data will show more zeroes to the right of the decimal point.)

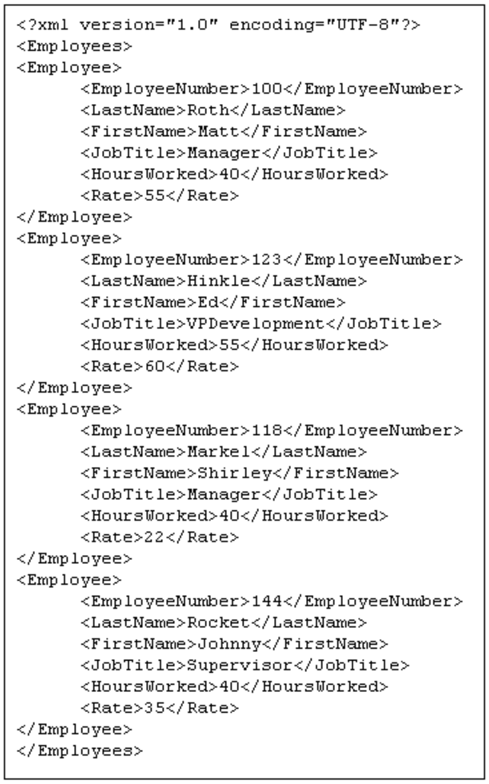
Figure 60 Output File



```
<?xml version="1.0" encoding="UTF-8"?>
<tns:Employees xmlns:tns="urn:please.define.your.own.t
    <tns:Employee>
        <tns:FullName>MattRoth</tns:FullName>
        <tns:EmpNumber>100</tns:EmpNumber>
        <tns:PayAmount>2451.000000</tns:PayAmount>
    </tns:Employee>
    <tns:Employee>
        <tns:FullName>EdHinkle</tns:FullName>
        <tns:EmpNumber>123</tns:EmpNumber>
        <tns:PayAmount>3300.000000</tns:PayAmount>
    </tns:Employee>
    <tns:Employee>
        <tns:FullName>ShirleyMarkel</tns:FullName>
        <tns:EmpNumber>118</tns:EmpNumber>
        <tns:PayAmount>880.000000</tns:PayAmount>
    </tns:Employee>
    <tns:Employee>
        <tns:FullName>JohnnyRocket</tns:FullName>
        <tns:EmpNumber>144</tns:EmpNumber>
        <tns:PayAmount>1400.000000</tns:PayAmount>
    </tns:Employee>
</tns:Employees>
```

In this sample, the first and last names are concatenated in the **FullName** field, as shown in Figure 60.

Figure 61 Input File



```
<?xml version="1.0" encoding="UTF-8"?>
<Employees>
<Employee>
      <EmployeeNumber>100</EmployeeNumber>
      <LastName>Roth</LastName>
      <FirstName>Matt</FirstName>
      <JobTitle>Manager</JobTitle>
      <HoursWorked>40</HoursWorked>
      <Rate>55</Rate>
</Employee>
<Employee>
      <EmployeeNumber>123</EmployeeNumber>
      <LastName>Hinkle</LastName>
      <FirstName>Ed</FirstName>
      <JobTitle>VPDevelopment</JobTitle>
      <HoursWorked>55</HoursWorked>
      <Rate>60</Rate>
</Employee>
<Employee>
      <EmployeeNumber>118</EmployeeNumber>
      <LastName>Markel</LastName>
      <FirstName>Shirley</FirstName>
      <JobTitle>Manager</JobTitle>
      <HoursWorked>40</HoursWorked>
      <Rate>22</Rate>
</Employee>
<Employee>
      <EmployeeNumber>144</EmployeeNumber>
      <LastName>Rocket</LastName>
      <FirstName>Johnny</FirstName>
      <JobTitle>Supervisor</JobTitle>
      <HoursWorked>40</HoursWorked>
      <Rate>35</Rate>
</Employee>
</Employees>
```

3.16.1 Text Editor

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.

You can 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.

After your input data is processed, the file name is changed to **TimeCard.xml.~in**. When you delete the file extension name **.~in**, the **TimeCard.xml** File is processed again.

Note: *If the ".~in" does not appear in your folder view you may need to change the Windows setting under "Folder Options." Go to the View tab and uncheck "Hide extensions for known file types."*

Within a few seconds after deleting the extension **~in**, the File eWay processes the input file again and writes the results to **s1_output.dat**. If your Multiple Records per file setting is "False," a new output file will be created each time the input data is processed. Open the output file with a text edit program and view the data to see your changes.

Chapter 4

Building a Project - Scenario2

This Tutorial provides step-by-step procedures for creating and testing an eGate Project, using sample data. Project2 demonstrates how to use an input payroll File containing hours worked and rate to calculate gross pay and then output a payroll File with gross pay data.

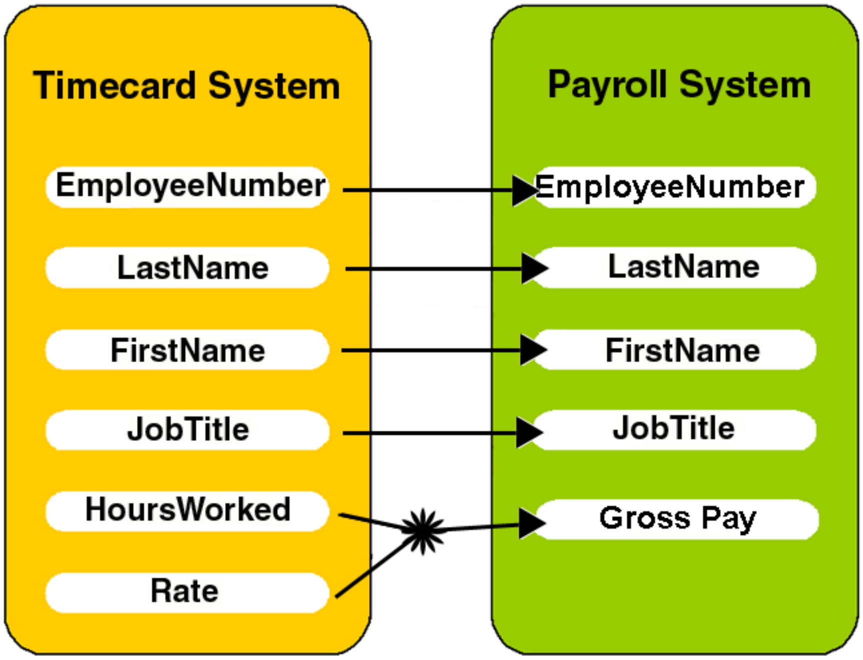
What's in This Chapter

4.1 Business Challenge

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

- A timecard system tracks the weekly hours worked by employees. The data in this system is in text format with six fields: **EmployeeNumber**, **LastName**, **FirstName**, **JobTitle**, **HoursWorked**, and **Rate**.
- A very simple payroll File is created. The output fields are **EmployeeNumber**, **LastName**, **FirstName**, **JobTitle**, and **GrossPay**. This output file is also in text format.
- The value for the **GrossPay** field in the Payroll system must be calculated from data in the timecard system.

Figure 62 The Relationship Between the Two Systems



4.2 Project Overview

Create a basic eGate Project to meet the business challenge described in **"Business Challenge" on page 72**. If you did the Scenario1 Tutorial, you may use the same Repository and Environment for Scenario2.

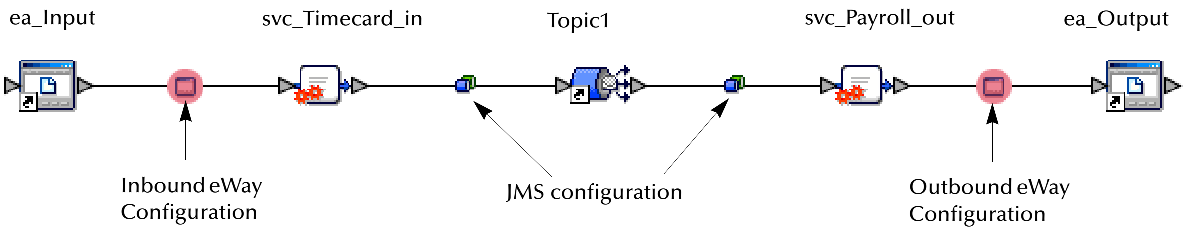
4.2.1 Required Resources

See **Chapter 3**, **Sample Data** on page 22.

4.2.2 Project Description

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

Figure 63 Project Connectivity Map



These components perform the following functions:

- **ea_Input**: The ea_Input *External Application* represents the Timecard system containing the Input data file. (**ea_Input** stands for external application input file.)
- **Inbound eWay**: The inbound eWay polls a specified location on the File system for the input data file.
- **svc_Timecard_in**: This Service copies the input data to **Topic1** without making any changes to the data.
- **Topic1**: This Topic publishes the message from the Service **svc_Timecard_in** to **svc_Payroll_out**.
- **svc_Payroll_out**: This Service copies the input timecard elements to the appropriate output Payroll system elements.
- **Outbound eWay**: This eWay publishes the finished output data to the Payroll system.
- **ea_Output**: This represents the Payroll system—the output data file.

4.2.3 Naming Conventions used in this Scenario

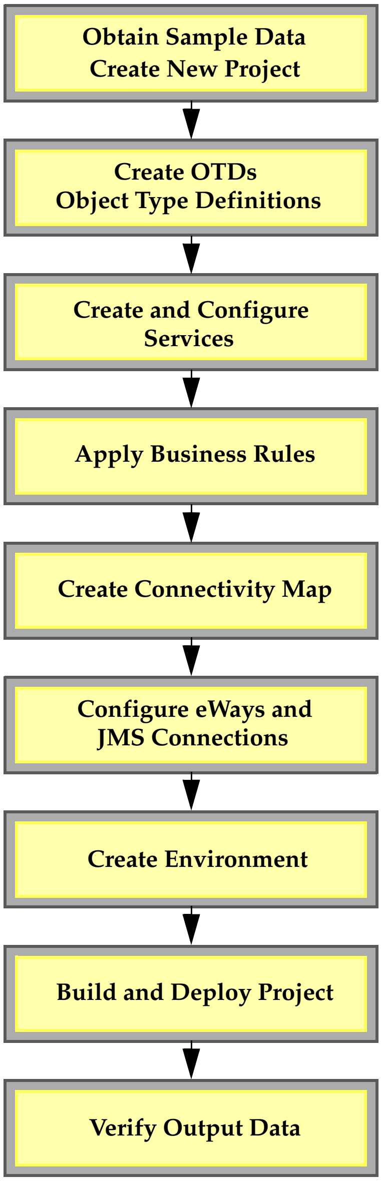
The following naming conventions are used in Scenario2 and are used in Sun Microsystems' training classes.

Table 2 Naming Conventions

Component Type	Naming Convention Prefix
Repository_<additional info>	**Repository_tutorial**
Project	**prj** (e.g. **prj_2-UD**)
Connectivity Map	**cm** (e.g. **cm_2_UD**)
Object Type Definition (OTD)	**otd** (e.g. **otd_input**)
Java Collaboration Definition	**jcd** (e.g. **jcd_Timecard_in**)
External Application	**ea** (e.g. **ea_Input**)
Service	**svc** (e.g. **svc_Timecard_in**)

4.2.4 Project Flow Diagram - Scenario2

Figure 64 Project Flowchart - Scenario2



4.3 Sample Data for Project2

This Project uses the sample files found in the Project2 folder of the **eGate Tutorial Sample**. The input file used in Scenario2 is: **Timecard.txt**.

4.3.1 Download the Sample File

See **“Download the Sample files” on page 23**.

Sample Input Text Data/File

This section shows a sample of the input data used in Scenario2. Your text file should look similar to the structure and syntax below. The input file is in the Project2 folder and does not have to be created.

```
200~Hazelwood~Johnny~Manager~40~33
210~Smith~Chuck~Clerk~40~25
220~Jones~Terry~Manager~37~45
230~Chang~Judy~Manager~40~45
240~Nakamura~Jim~Manager~35~45
```

Sample Output Text Data/File

```
200~Hazelwood~Johnny~Manager~1320
210~Smith~Chuck~Clerk~1000
220~Jones~Terry~Manager~1665
230~Chang~Judy~Manager~1800
240~Nakamura~Jim~Manager~1575
```

4.4 Create a New Project

Begin by creating and naming a Project in the Enterprise Designer.

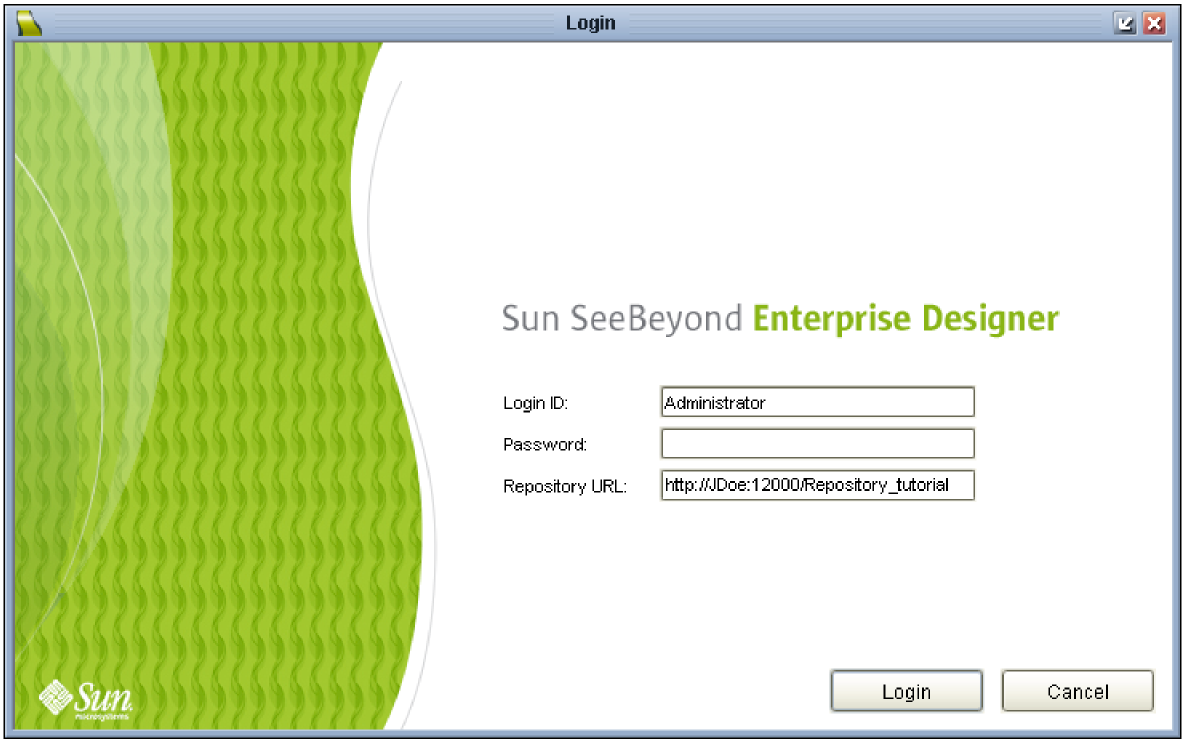
Start Enterprise Designer

Start the Enterprise Designer. The following is an example:

1 Start the Enterprise Designer by executing **runed.bat** in your JavaCAPS51 folder: **`C:\JavaCAPS51\edesigner\bin`**.

  The Enterprise Designer Login dialog box appears.

Figure 65 Enterprise Designer Login Dialog Box



Login ID: Administrator

Password:

Repository URL: http://JDoe:12000/Repository_tutorial

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

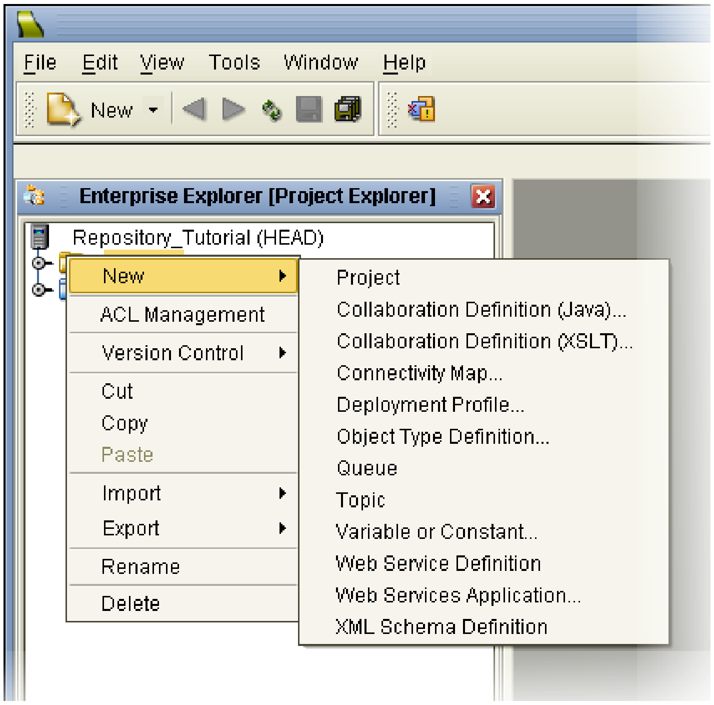
Note: *If your login dialog box doesn't appear or if you get an invalid Username or Password error, make sure your Repository is running. Also, be aware that the Username and Password are case sensitive.*

Create prj_2_UD

A Project is a collection of logical components, configurations, and files that are used to solve a business problem. If you have already done Scenario1, this project can be **prj_2_UD**. The "prj" is an optional prefix that stands for "project."

1 In the Project Explorer tree of the Enterprise Designer, right-click the Repository name (computer icon). Click **New Project**. (You can also select "Project" from the drop-down list.)

Figure 66 Create a Project

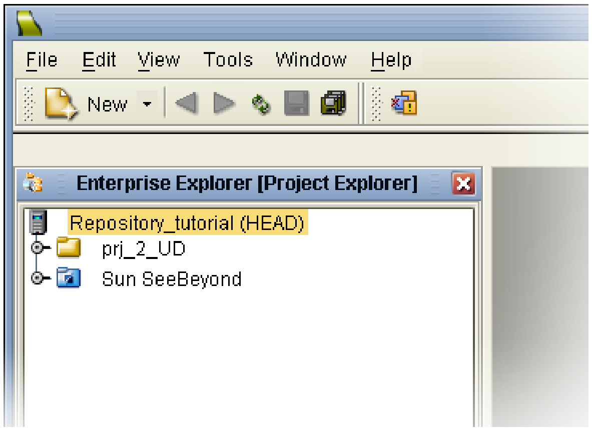


2 Name your Project, **prj_2_UD**. (This stands for Project 2 User-Defined OTD.)

3 Press **Enter**.

Your **prj_2_UD** appears in the Project Explorer tree.

Figure 67 Project Folder: prj_2_UD


File
Edit
View
Tools
Window
Help
New
Enterprise Explorer [Project Explorer]
Repository_tutorial (HEAD)
prj_2_UD
Sun SeeBeyond

4.5 Create a New Object Type Definition

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

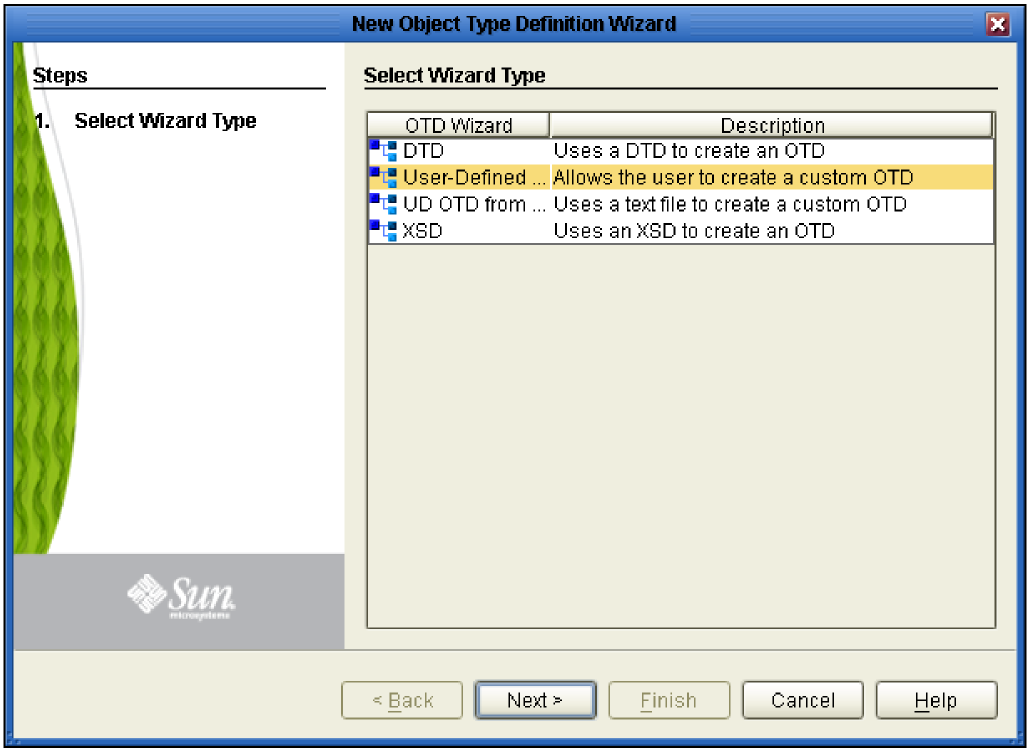
4.5.1 Create a User-Defined OTD for the Input File

The following steps explain how to create a User-Defined OTD for your input file.

OTD for Input

1. Right-click **prj_2_UD**.
2. Click **New**, then **Object Type Definition**.
3. Click the **User-Defined OTD** item. (See the following figure.)

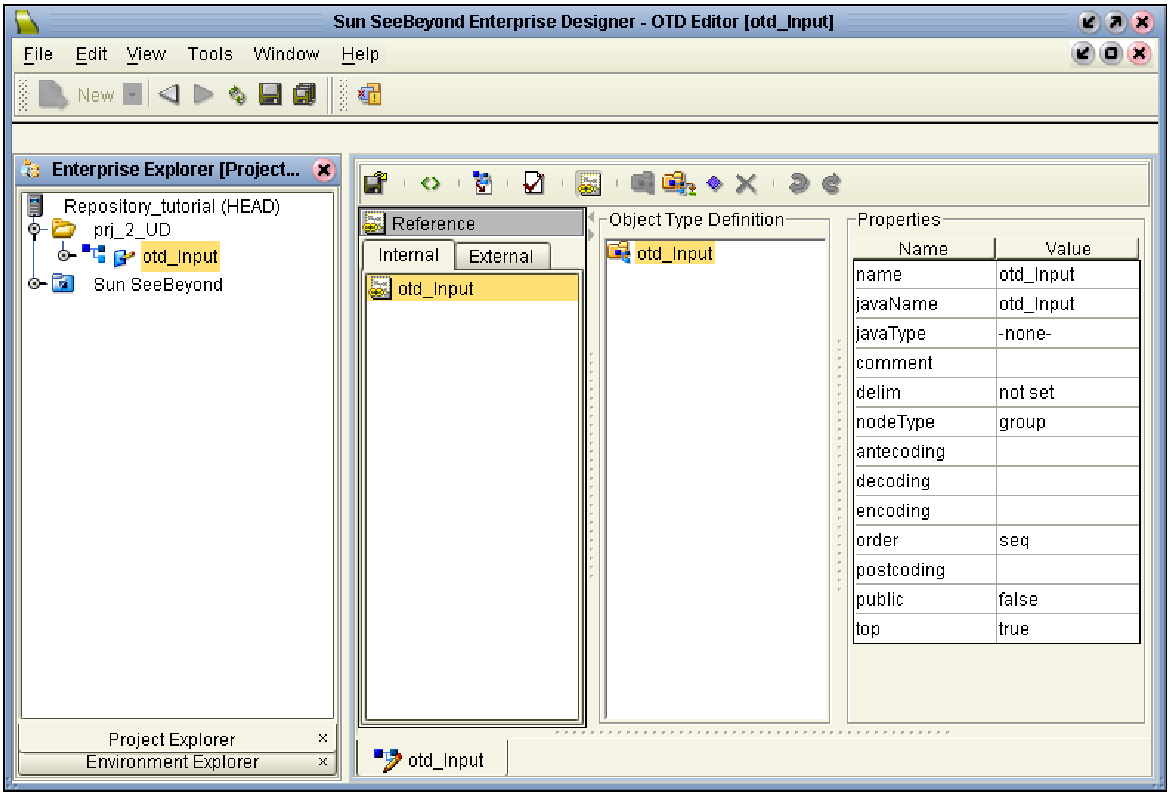
Figure 68 OTD Wizard Selection



OTD Wizard	Description
DTD	Uses a DTD to create an OTD
User-Defined ...	Allows the user to create a custom OTD
UD OTD from ...	Uses a text file to create a custom OTD
XSD	Uses an XSD to create an OTD

4. Click **Next**.
5. Name your OTD **otd_input**.
6. Click **Finish**. The OTD Editor appears.

Figure 69 OTD Editor for Input



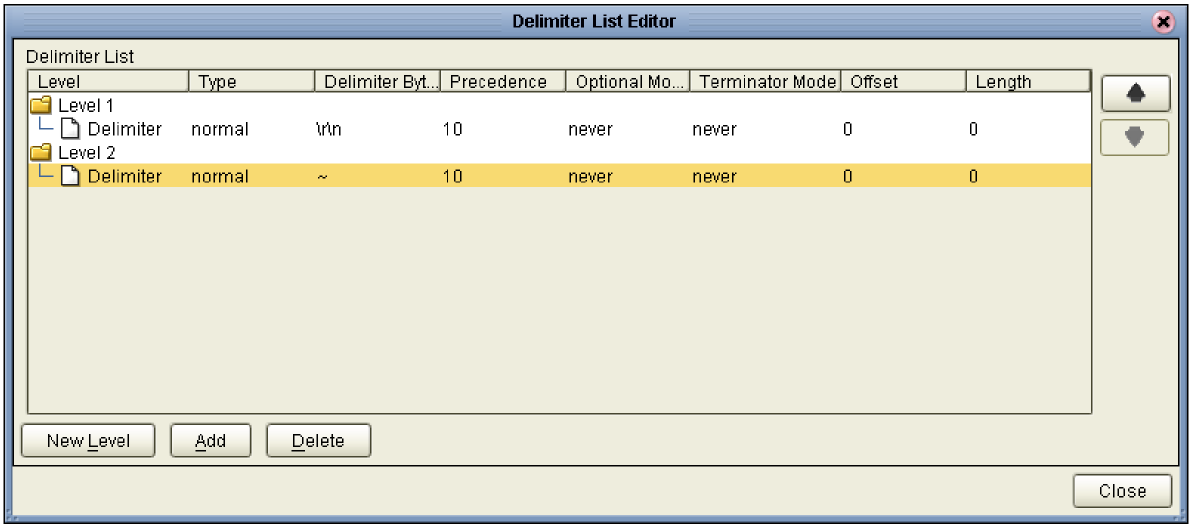
Name	Value
name	otd_Input
javaName	otd_Input
javaType	-none-
comment	
delim	not set
nodeType	group
antecoding	
decoding	
encoding	
order	seq
postcoding	
public	false
top	true

User-Defined Fields

You are now ready to define the properties for your input fields. You will have a total of 6 fields to define. Refer to Figure 71 as you go through the steps.

1. Select the root node (**otd_input**) (number 1 in Figure 71) and make sure Properties is set: **nodeType = group**.
2. Next to **delim**, click "not set" and then click the **ellipses** button (...).
3. Click **NewLevel** then click the **Add** button to add a new line character (**\r\n**) and press Enter.
4. Click **NewLevel** then click the **Add** button to add the tilde (~) delimiter and press Enter.

Figure 70 Delimiter Settings



Level	Type	Delimiter Byt...	Precedence	Optional Mo...	Terminator Mode	Offset	Length
Level 1							
Delimiter	normal	\r\n	10	never	never	0	0
Level 2							
Delimiter	normal	~	10	never	never	0	0

5 Close the **Delimiter List Editor**.

The **delim** property shows the value “**specified**.”

6 Right-click **otd_input** (see number 2 in Figure 71), then select **Add**, **Element**.

7 Set the **element** property **repeat** to **true**. This is a repeating node.

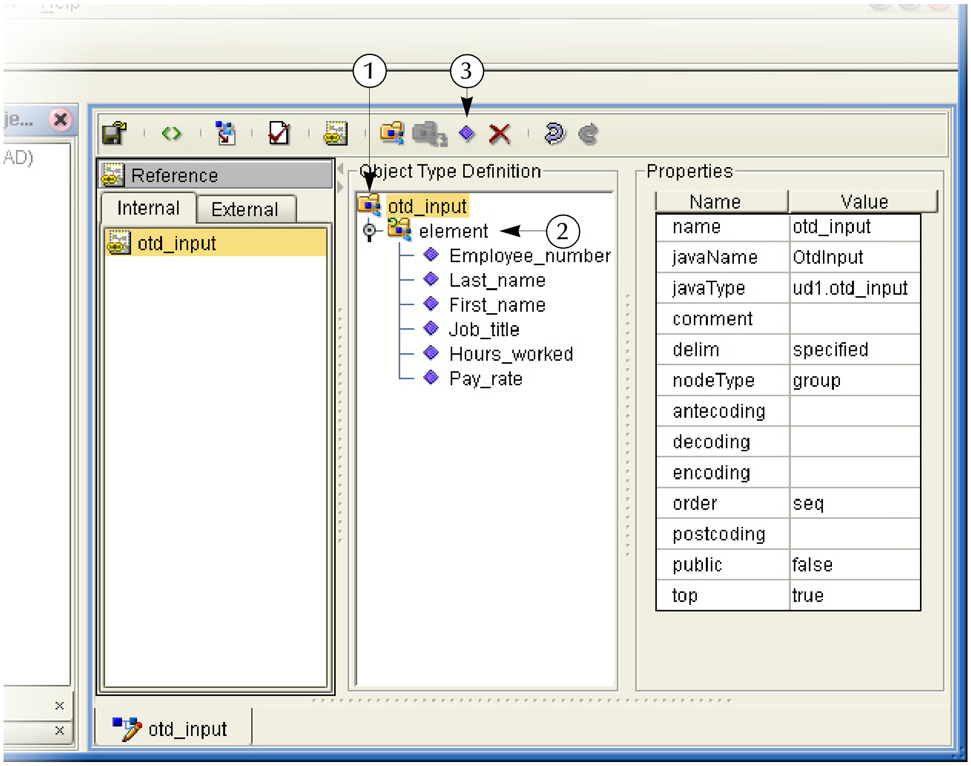
8 Verify the default **element** property **nodeType** is delim.

9 Press **Enter**.

Next you add fields. You can add fields using the **Field** button (number 3 in Figure 71).

10 Add Fields until you have a total of six fields (field through field_5).

Figure 71 User-Defined Input OTD Editor

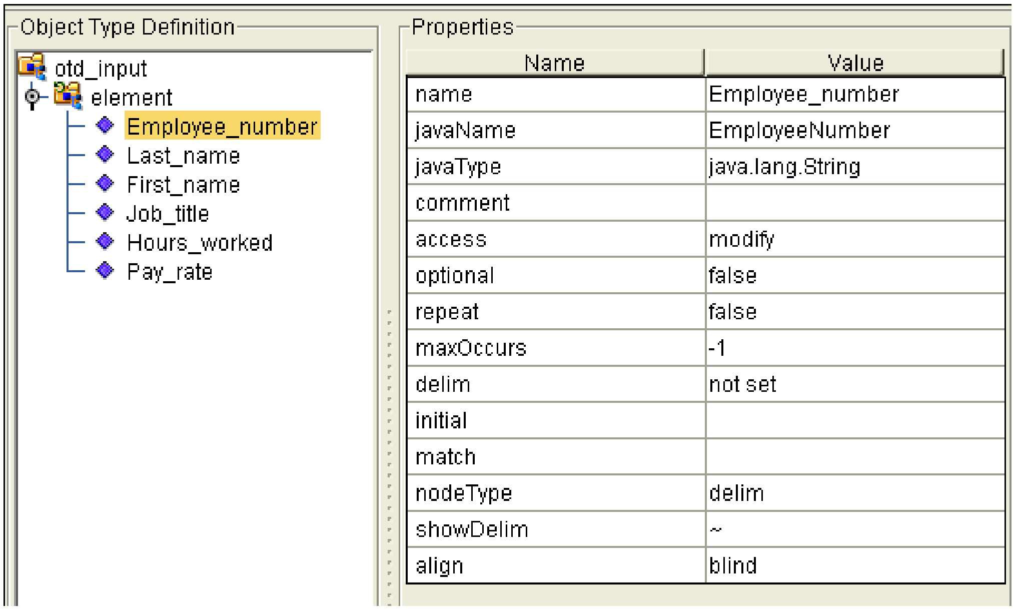


Name	Value
name	otd_input
javaName	OtdInput
javaType	ud1.otd_input
comment	
delim	specified
nodeType	group
antecoding	
decoding	
encoding	
order	seq
postcoding	
public	false
top	true

11 Right-click each field to rename it. The field names should match those shown in Figure 71.

Note: *Be sure to press **Enter** after setting the Properties for each field. A unique javaName is set/resolved when the OTD is saved.*

Figure 72 Set Field Names and Delimiters



Name	Value
name	Employee_number
javaName	EmployeeNumber
javaType	java.lang.String
comment	
access	modify
optional	false
repeat	false
maxOccurs	-1
delim	not set
initial	
match	
nodeType	delim
showDelim	~
align	blind

12 Save your Project after setting all the input fields.

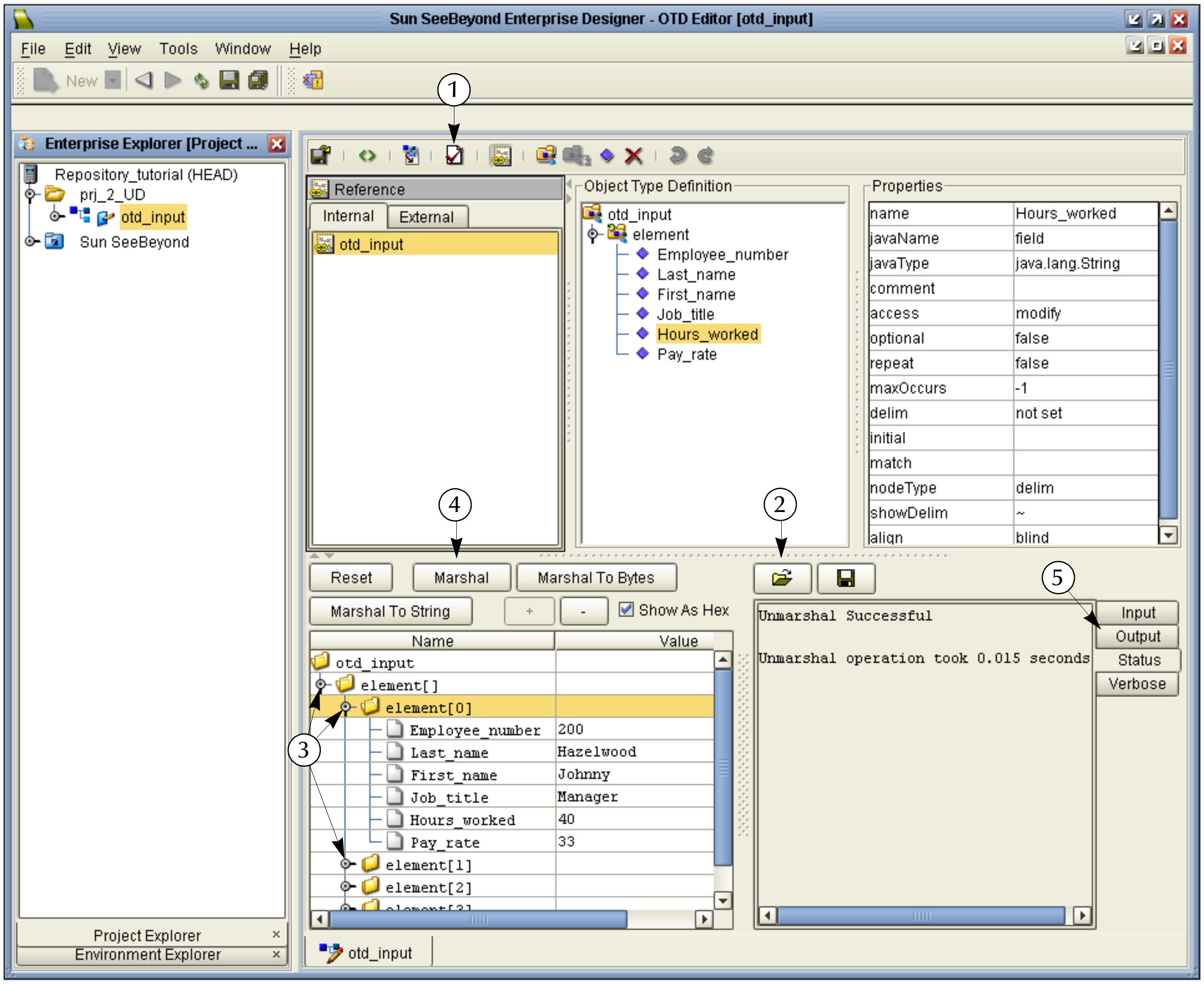
4.5.2 Test Input Data

Run a test to ensure that the Properties are set correctly and there are no problems with your input data file.

Run Test

1 Click the **Run Test** button on the toolbar (number 1 in Figure 73).

Figure 73 Test Input Data



Properties

name	Hours_worked
javaName	field
javaType	java.lang.String
comment	
access	modify
optional	false
repeat	false
maxOccurs	-1
delim	not set
initial	
match	
nodeType	delim
showDelim	~
align	blind

☑ Show As Hex

Name	Value
otd_input	
element[]	
element[0]	
Employee_number	200
Last_name	Hazelwood
First_name	Johnny
Job_title	Manager
Hours_worked	40
Pay_rate	33
element[1]	
element[2]	

```
Unmarshal Successful

Unmarshal operation took 0.015 seconds
```

2 Click the **Open a File** button (number 2 in Figure 73) and then navigate to your input data file.

3 Double-click your input file.

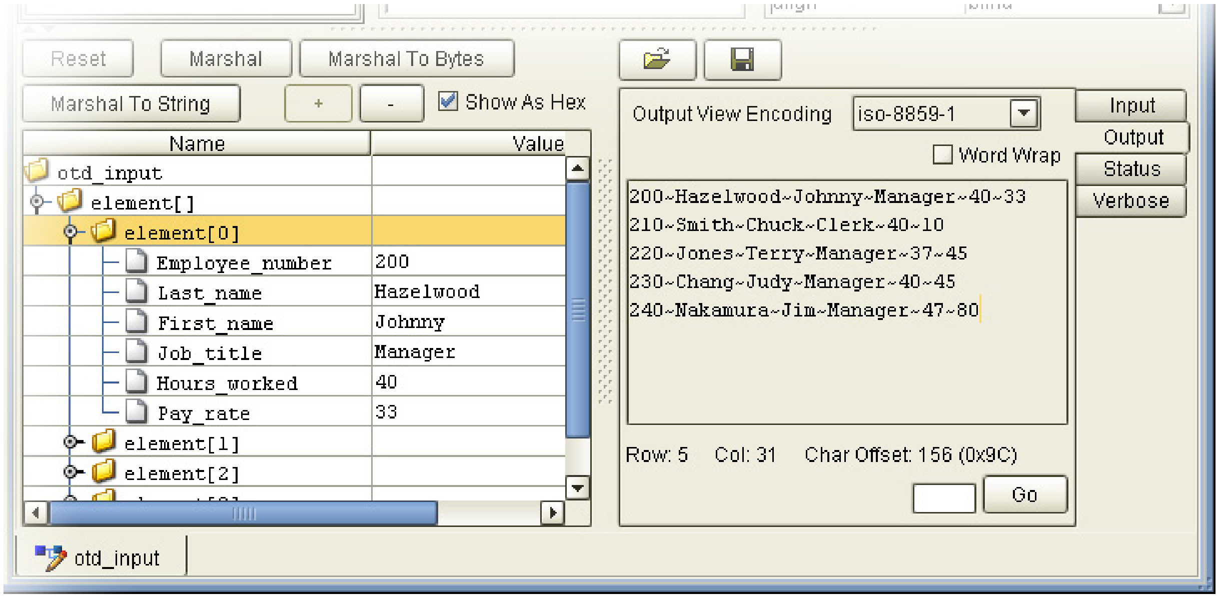
The words “Unmarshal Successful” should appear in the pane.

4 Open the element nodes (number 3 in Figure 73.)

5 Click the **Marshal** button (number 4 in Figure 73).

6 Click the **Output** button (number 5 in Figure 73).

Figure 74 Marshal to String



☑ Show As Hex

Name	Value
otd_input	
element[]	
element[0]	
Employee_number	200
Last_name	Hazelwood
First_name	Johnny
Job_title	Manager
Hours_worked	40
Pay_rate	33
element[1]	
element[2]	

Output View Encoding iso-8859-1

□ Word Wrap

```
200~Hazelwood~Johnny~Manager~40~33
210~Smith~Chuck~Clerk~40~10
220~Jones~Terry~Manager~37~45
230~Chang~Judy~Manager~40~45
240~Nakamura~Jim~Manager~47~80
```

Row: 5 Col: 31 Char Offset: 156 (0x9C)

Your output results should look similar to Figure 74.

7 Close the Tester (Click the same **Run Test** button again).

8 Save your Project.

4.5.3 Create a User-Defined OTD for the Output File

The following steps explain how to create a User-Defined OTD for your output file.

OTD for Output

1 Right-click **prj_2_UD**.

2 Click **New**, then **Object Type Definition**.

3 Select **User-Defined OTD** from the list.

4 Click **Next**.

5 Name your OTD **otd_Output**.

6 Click **Finish**.

The OTD Editor appears.

You are now ready to define the properties for your output fields. You will have a total of five fields to define. Refer to Figure 76 as you go through the steps.

7 Select the root node (**otd_output**) (number 1 in Figure 76), and make sure **nodeType** is set to **group**.

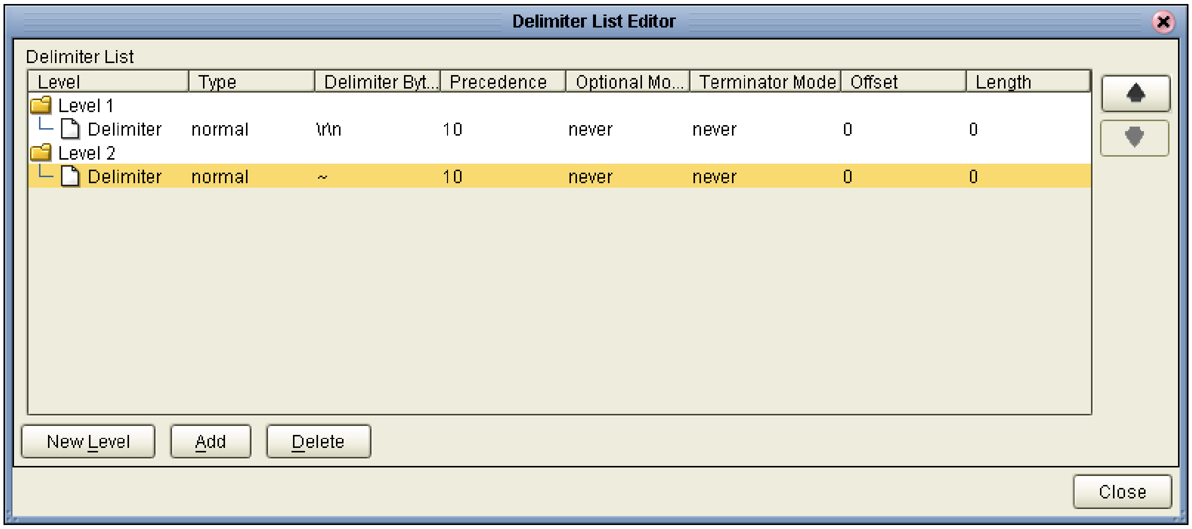
With the root node selected, set the delimiters.

8 Next to **delim**, click “not set” and then click the **ellipses** button (...).

9 Click **NewLevel** then click the **Add** button to add a new line character (**\r\n**) and press Enter.

10 Click **NewLevel** then click the **Add** button to add the tilde (~) delimiter and press Enter.

Figure 75 Delimiter Settings



Delimiter List Editor

Delimiter List

Level	Type	Delimiter Byt...	Precedence	Optional Mo...	Terminator Mode	Offset	Length
Level 1							
Delimiter	normal	\r\n	10	never	never	0	0
Level 2							
Delimiter	normal	~	10	never	never	0	0

11 Close the **Delimiter List Editor**.

12 Right-click **otd_output** then select **Add** and **Element** (see number 2 in Figure 76).

13 Set the **element** property **repeat** to **true**. This is a repeating node.

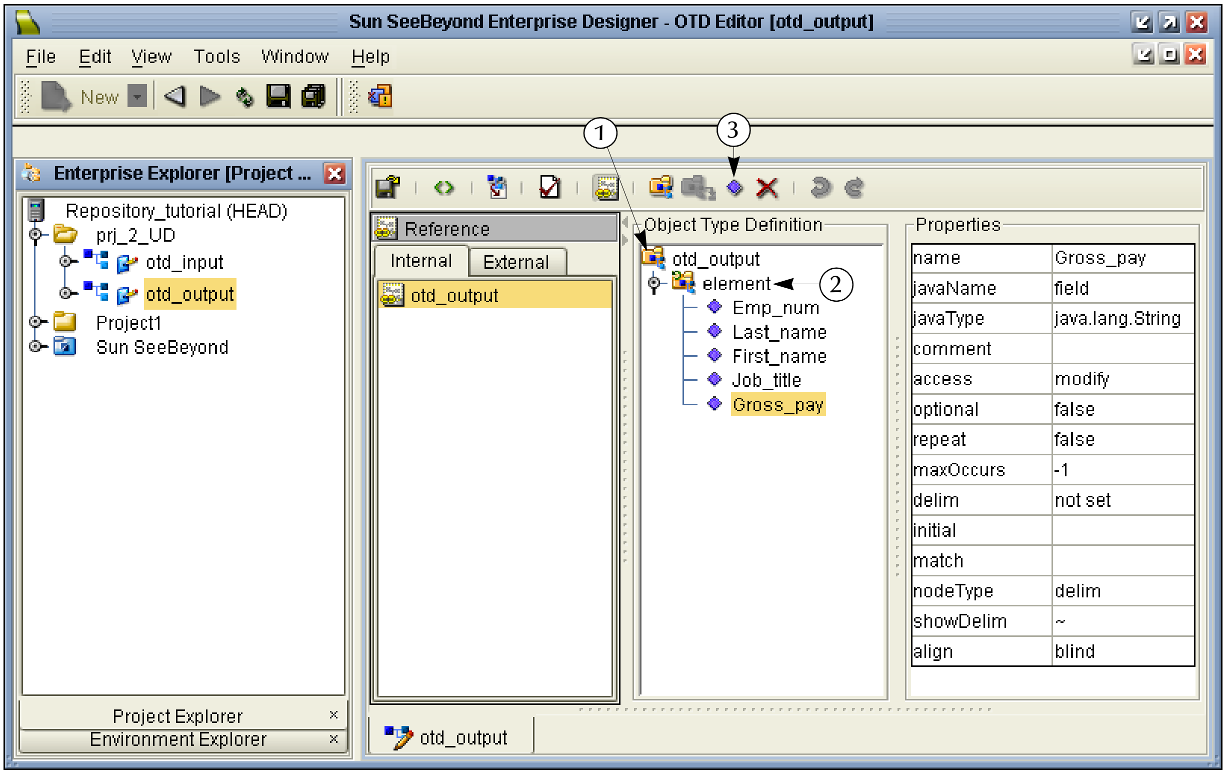
14 Verify the **element** property **nodeType** is delim.

15 Press **Enter**.

Next you add fields. You can add fields using the **Field** button (number 3 in Figure 76).

16 Add Fields until you have a total of five fields.

Figure 76 User-Defined Output OTD Editor



name	Gross_pay
javaName	field
javaType	java.lang.String
comment	
access	modify
optional	false
repeat	false
maxOccurs	-1
delim	not set
initial	
match	
nodeType	delim
showDelim	~
align	blind

17 Right-click each field to rename it. The field names should match those shown in Figure 76, and make sure **nodeType** is set to **delim** for each field.

Note: *Be sure to press* ***Enter*** *after setting the Properties for each field. A unique javaName is set/resolved when the OTD is saved.*

18 Save your Project after setting all the output fields.

4.6 Configure a Service for Input

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

4.6.1 Configure svc_Timecard_in

The first Collaboration (Service), **jcd_Timecard_in**, serves only to receive data from the input file and pass it to the next Service. The data is not processed in this Collaboration.

Timecard_in Collaboration

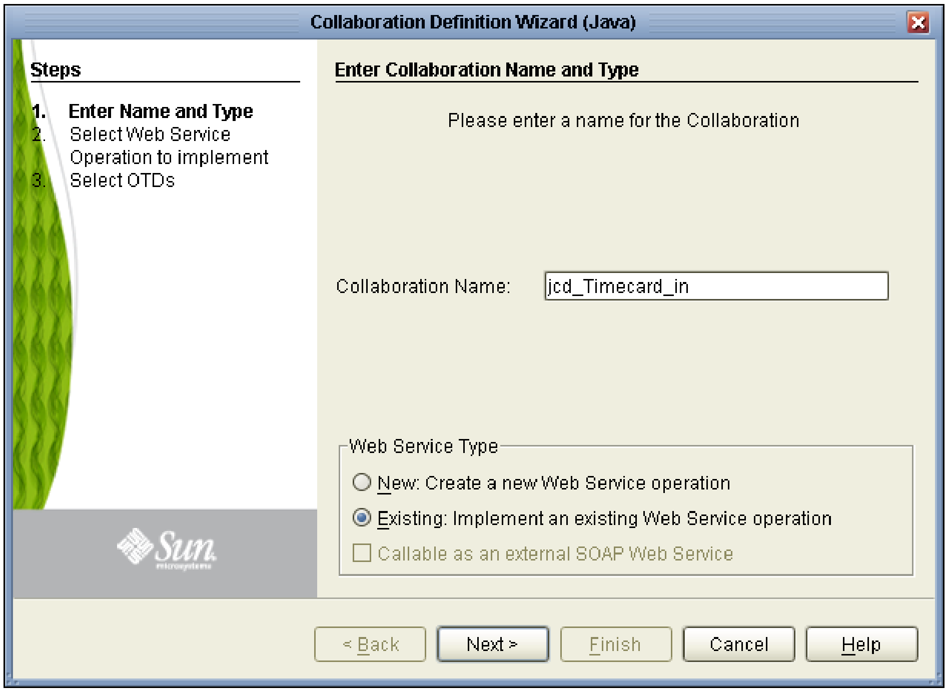
1 Right-click **prj_2_UD**.

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

The Collaboration Definition Wizard (Java) appears.

3 Type **jcd_Timecard_in** as the name for the Java Collaboration Definition. (The optional prefix “jcd” stands for Java Collaboration Definition.)

Figure 77 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
Web Service Type
< Back
Next >
Finish
Cancel
Help

Collaboration Name: jcd_Timecard_in

○ New: Create a new Web Service operation

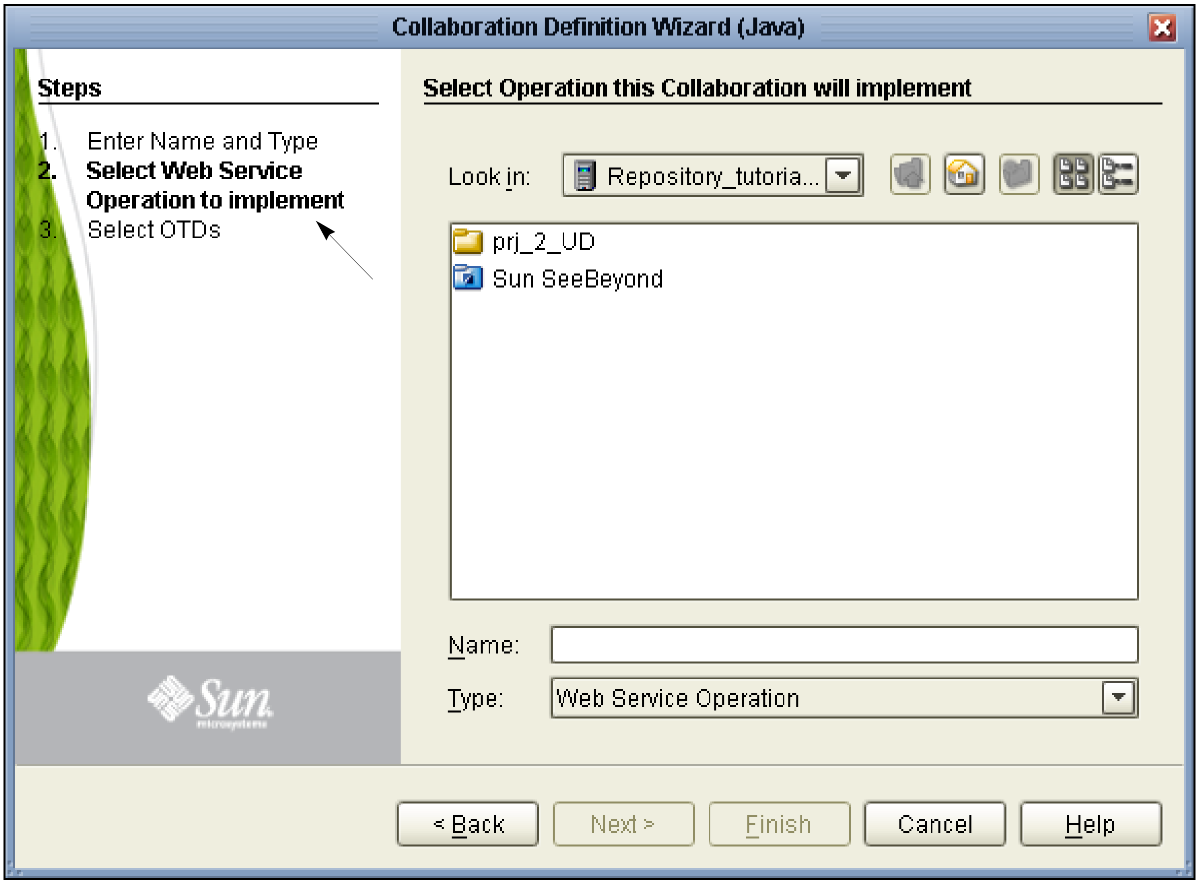
● Existing: Implement an existing Web Service operation

□ Callable as an external SOAP Web Service

Note: *Accept the Web Service Type default, “Existing: Implement an existing web service operation.”*

4 Click **Next**.

Figure 78 New Collaboration Definition Wizard (Java)



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

5 Double-click **Sun SeeBeyond**.

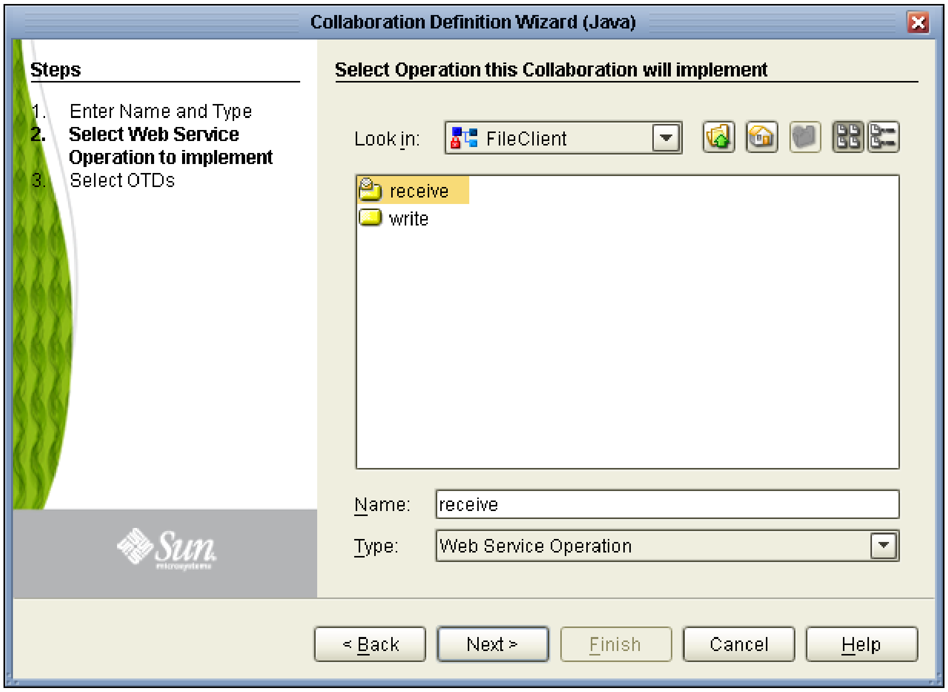
6 Double-click **eWays**. eWays are message end points that connect to an external service providers to enable the sending and receiving of messages.

7 Double-click **File**.

8 Double-click **FileClient**.

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

Figure 79 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

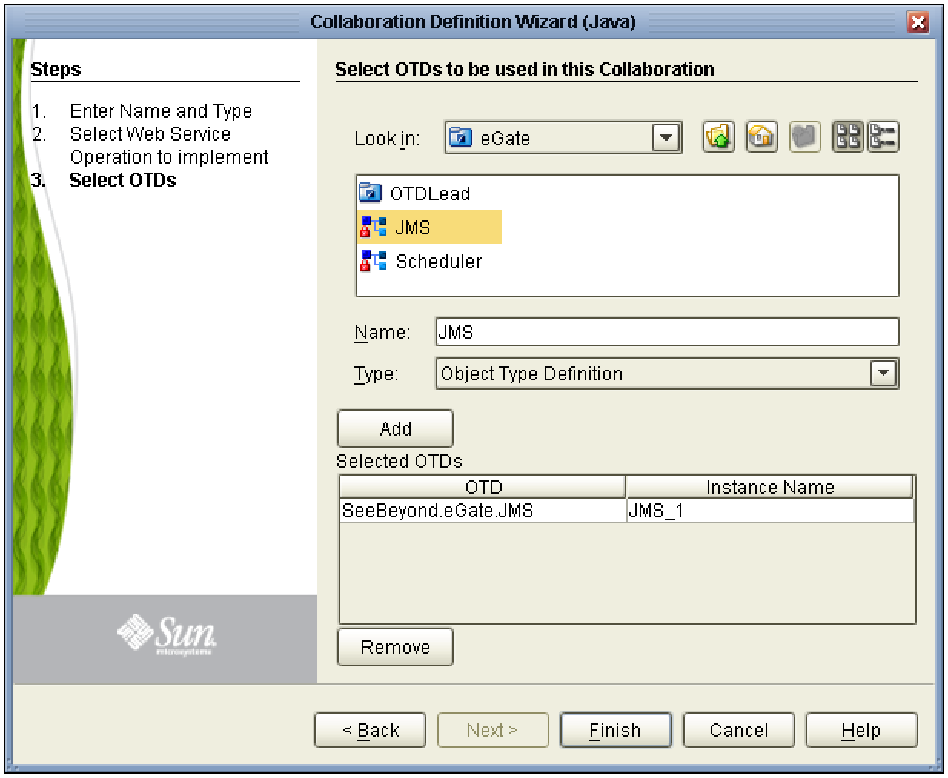
The following steps set up the OTD for the existing web service to write to JMS. The JMS OTD is used to route data between Topics (and Queues). You are still using the Collaboration Definition Wizard (Java).

1 Double-click **Sun SeeBeyond**.

2 Double-click **eGate**.

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

Figure 80 New Collaboration Select 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:
eGate
OTDLead
JMS
Scheduler
Add
Remove
< Back
Next >
Finish
Cancel
Help

Name: JMS

Type: Object Type Definition

Selected OTDs

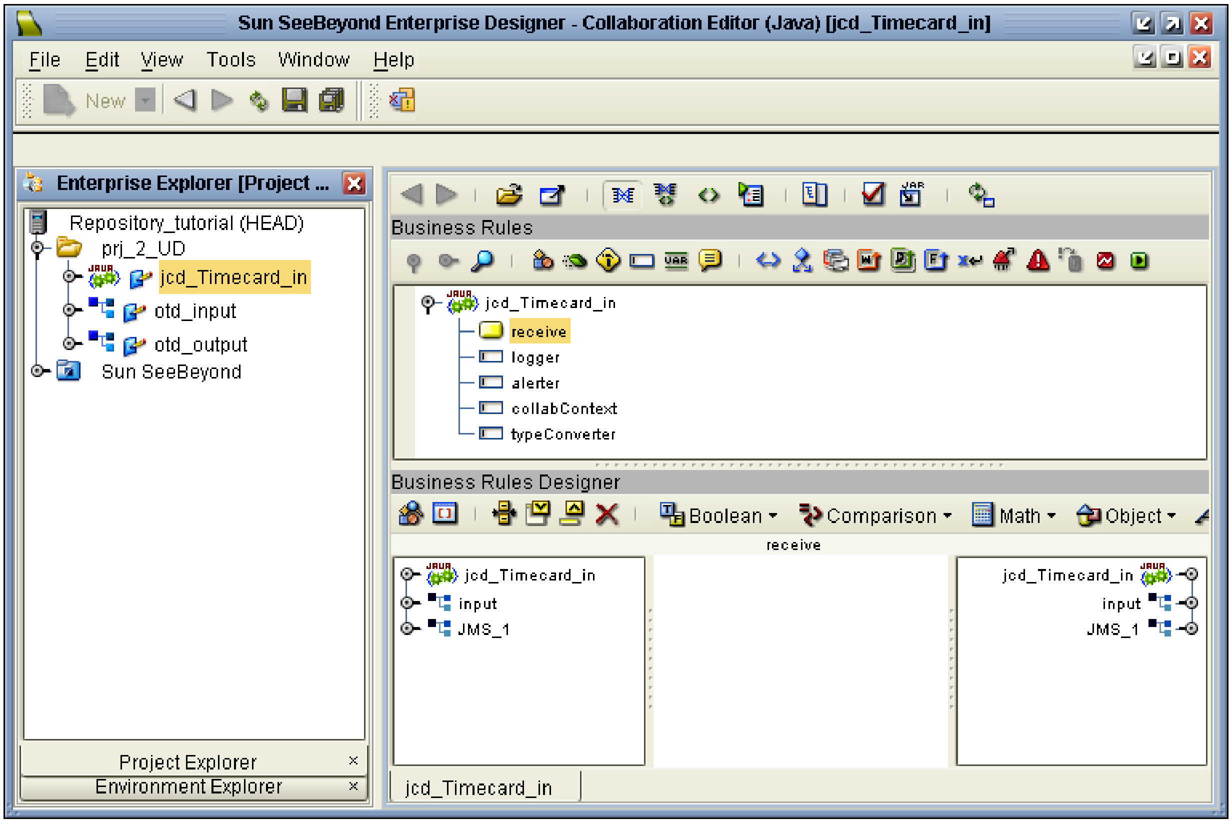
OTD	Instance Name
SeeBeyond.eGate.JMS	JMS_1

4 Click **Finish**.

After a few seconds, the **jcd_Timecard_in** icon appears in the Project Explorer tree in the left pane. Later, you will bind this Collaboration to the Service, **svc_Timecard_in**, in the Connectivity Map.

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

Figure 81 Business Rules Designer



Note: *Notice that the objects in Figure 81 reside in both the left and right pane of the Business Rules Designer window. Keep in mind that the OTD instances (**JMS_1** for example) refer to the same object although they are shown in the GUI in two different places at the same time.*

4.7 Apply Business Rules

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

Send Input to the Next Service

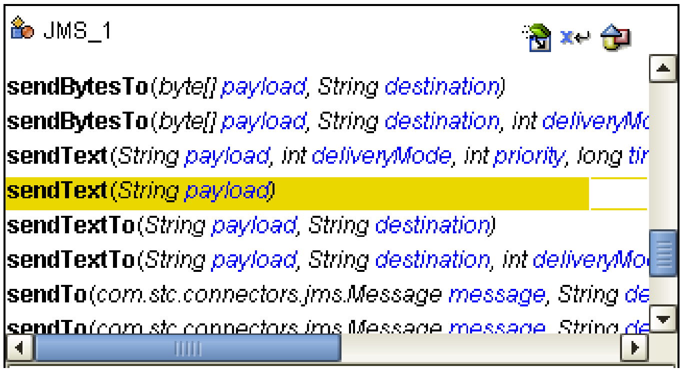
In these steps send the message to the next Service.

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

   This action causes the methods dialog box to appear.

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

Figure 82 Methods Dialog Box


JMS_1
sendBytesTo(byte[] payload, String destination)
sendBytesTo(byte[] payload, String destination, int deliveryMc
sendText(String payload, int deliveryMode, int priority, long tir
sendText(String payload)
sendTextTo(String payload, String destination)
sendTextTo(String payload, String destination, int deliveryMo
sendTo(com.stc.connectors.jms.Message message, String de
sendTo(com.stc.connectors.jms.Message message, String de

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

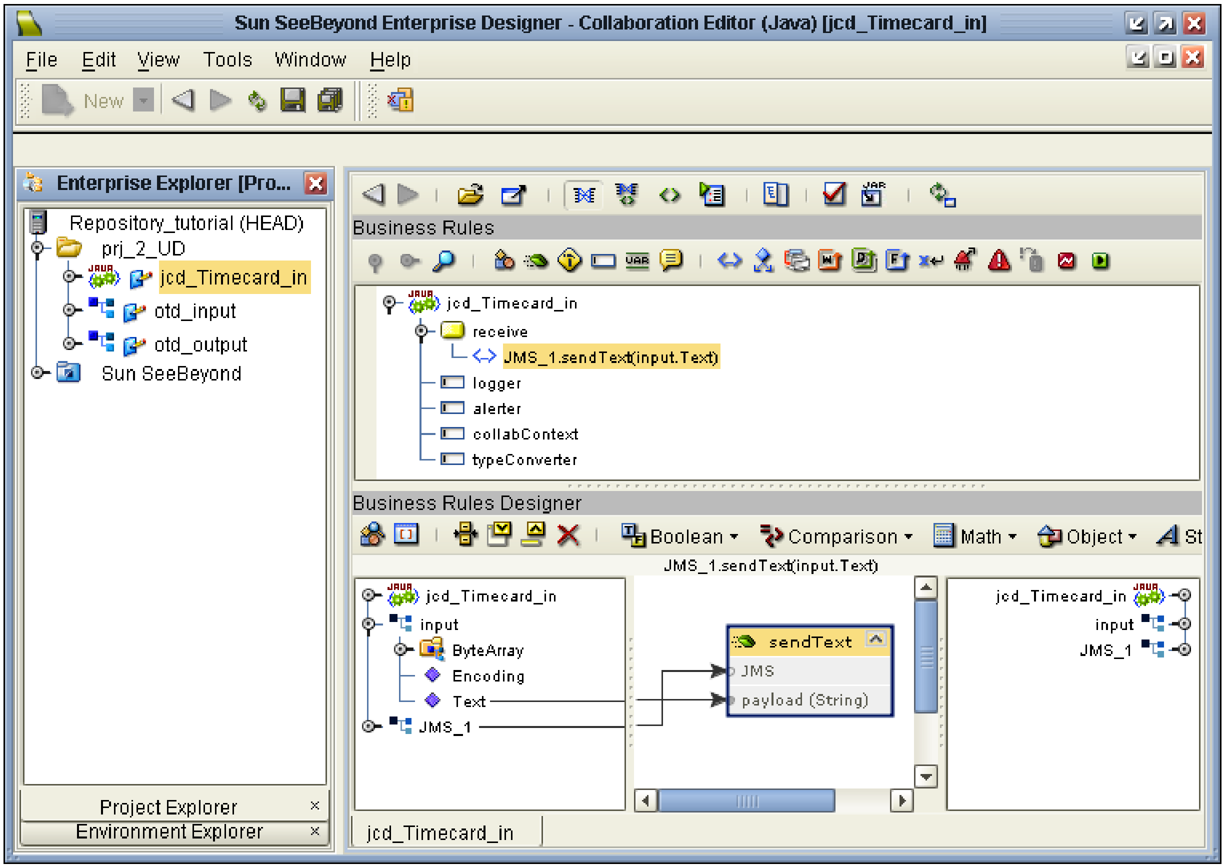
A **sendText** box appears in the center pane.

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

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

See the following figure.

Figure 83 Send Text Payload



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

This completes the setup for the **jcd_Timecard_in** Collaboration definition. The Collaboration (Service) passes the data to the Topic, which is then passed along to the next Service.

4.8 Configure a Service for Output

Select a Web Service Operation

Configure your web service interface.

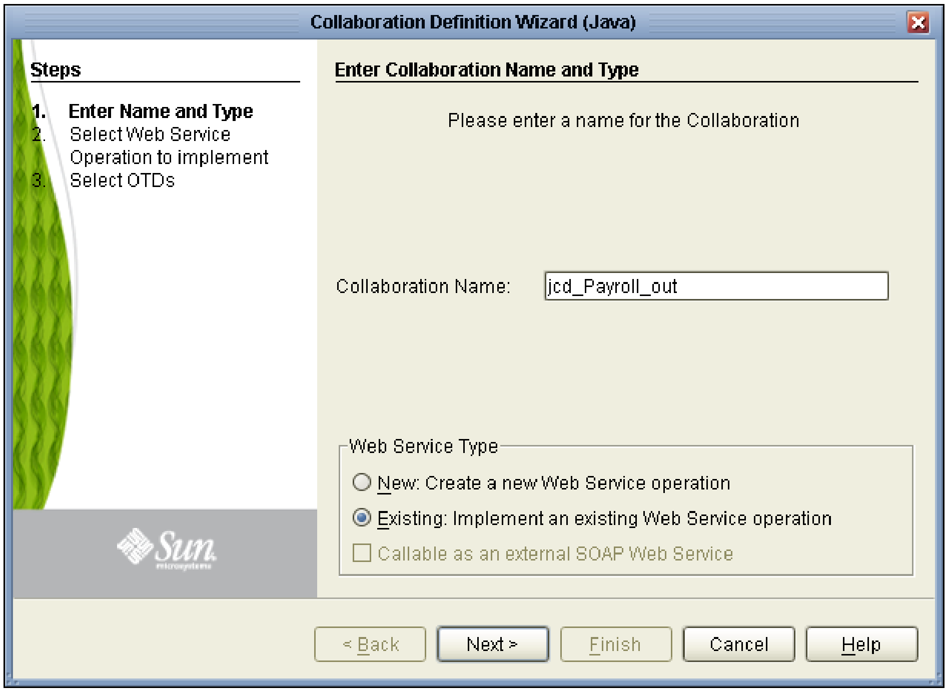
1 Right-click **prj_2_UD**.

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

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

See the following figure.

Figure 84 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 Collaboration

Collaboration Name: jcd_Payroll_out

Web Service Type

○ New: Create a new Web Service operation

● Existing: Implement an existing Web Service operation

□ Callable as an external SOAP Web Service

4 Click **Next**.

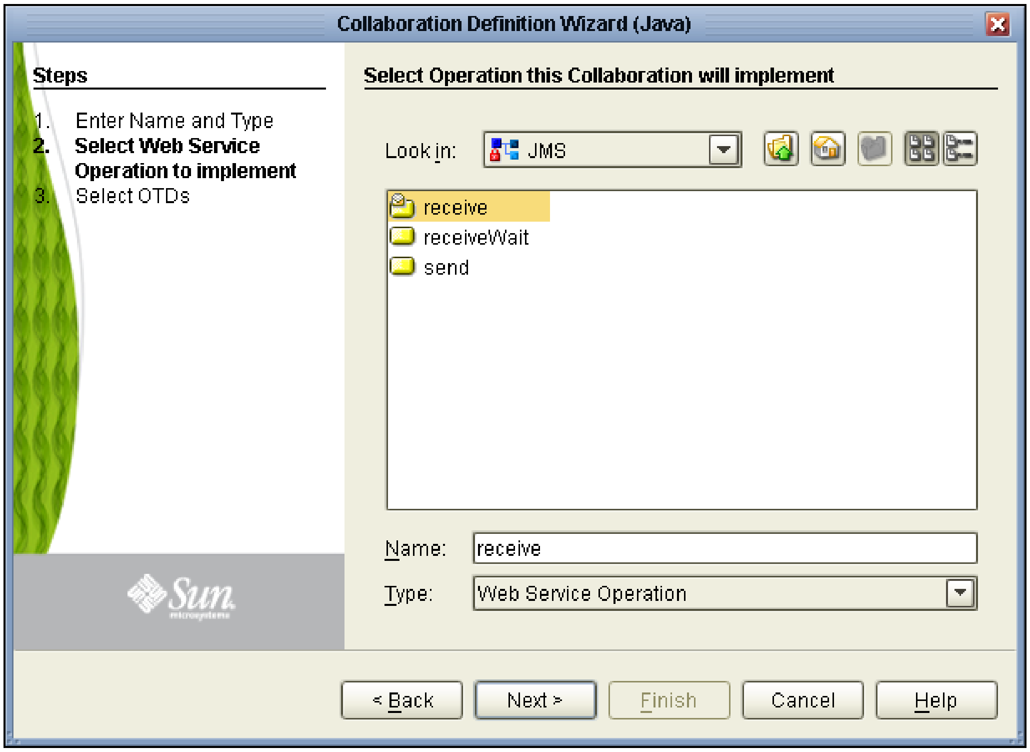
5 Using the Collaboration Definition Wizard (Java), double-click **Sun SeeBeyond** to "Select Operation this Collaboration will implement"

6 Double-click **eGate**.

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

8 Click **receive**. (Receive is a method/web service used to read a message from a Topic that your Collaboration implements.)

Figure 85 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

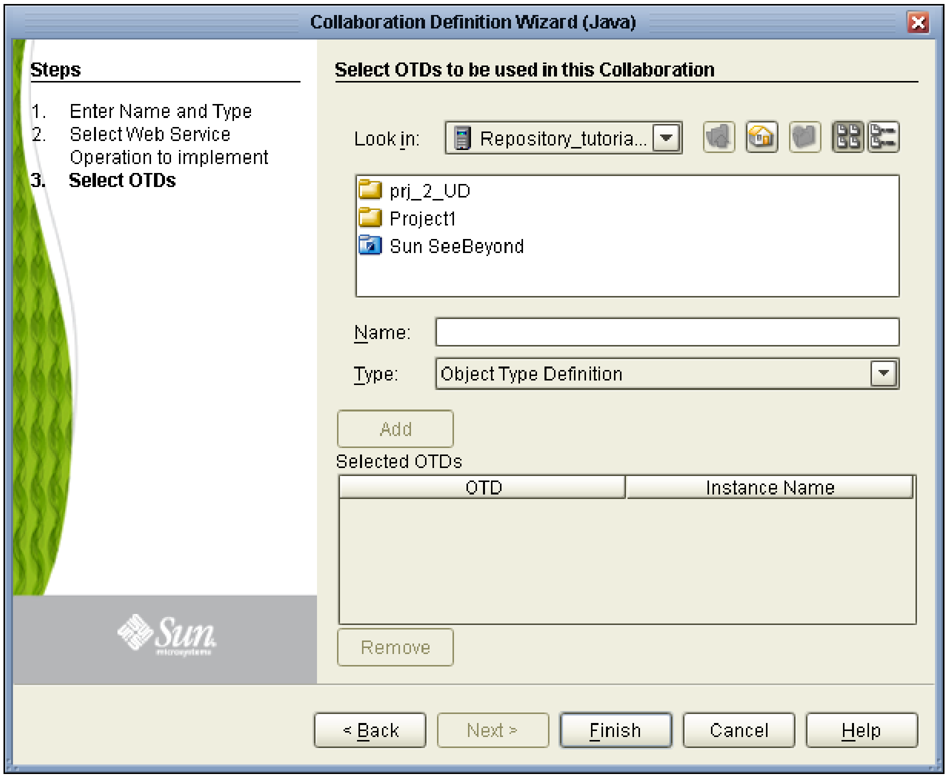
Name: receive

Type: Web Service Operation

9 Click **Next**.

The Collaboration Definition Wizard (Java) refreshes. Select OTDs in the following steps.

Figure 86 Select OTDs for jcd_Payroll_out



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: Repository_tutoria...

- prj_2_UD
- Project1
- Sun SeeBeyond

Name:

Type: Object Type Definition

Selected OTDs

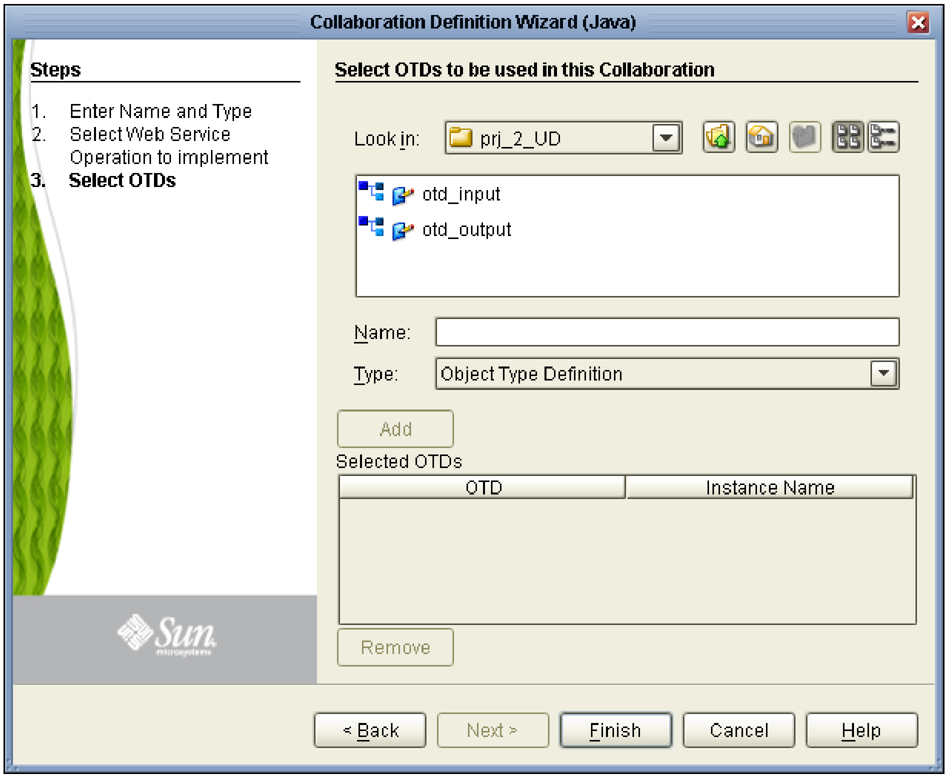
OTD	Instance Name

Select OTDs

1 Double-click **prj_2_UD** in the Collaboration Definition Wizard (Java).

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

Figure 87 Input/Output 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: prj_2_UD

otd_input
otd_output

Name:

Type: Object Type Definition

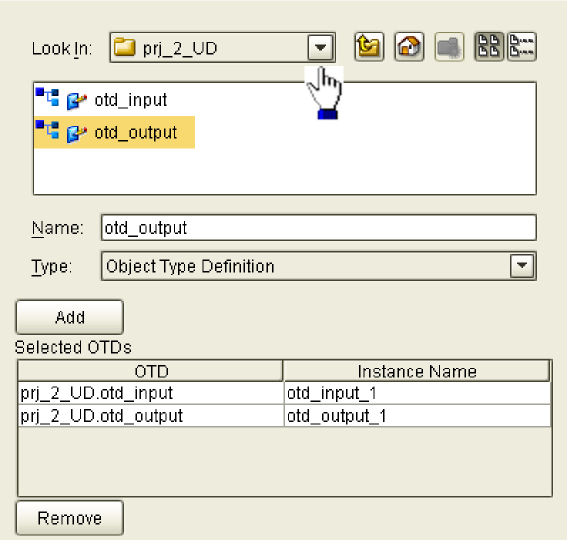
2 Double-click **otd_input** to move the selection to the list of Selected OTDs.

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

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

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

Figure 88 Drop Down Arrow button



Look In: prj_2_UD

Name: otd_output

Type: Object Type Definition

Selected OTDs

OTD	Instance Name
prj_2_UD.otd_input	otd_input_1
prj_2_UD.otd_output	otd_output_1

5 Click **Sun SeeBeyond**.

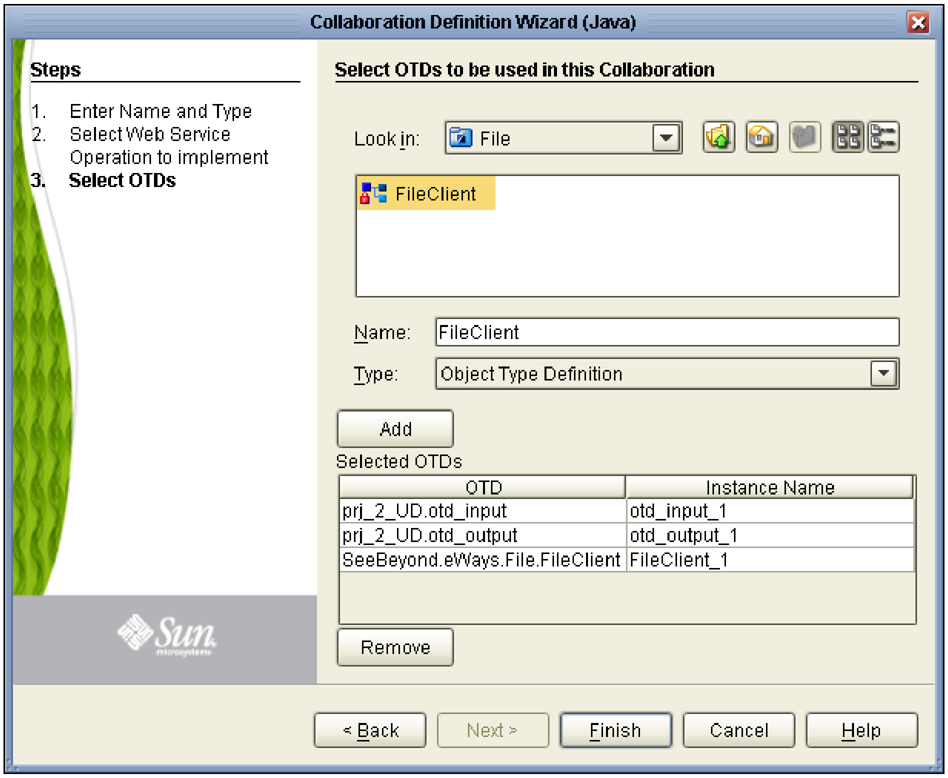
6 Double-click **eWays**.

7 Double-click **File**.

8 Double-click **FileClient**.

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

Figure 89 File Client 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
FileClient
Add
Remove
< Back
Next >
Finish
Cancel
Help

Name: FileClient

Type: Object Type Definition

Selected OTDs

OTD	Instance Name
prj_2_UD.otd_input	otd_input_1
prj_2_UD.otd_output	otd_output_1
SeeBeyond.eWays.File.FileClient	FileClient_1

9 Click **Finish**.

After a few seconds, **jcd_Payroll_out** appears in the Project Explorer tree in the left pane. The Business Rules Designer Collaboration Editor also appears. See Figure 90.

4.9 Apply Business Rules for jcd_Payroll_out

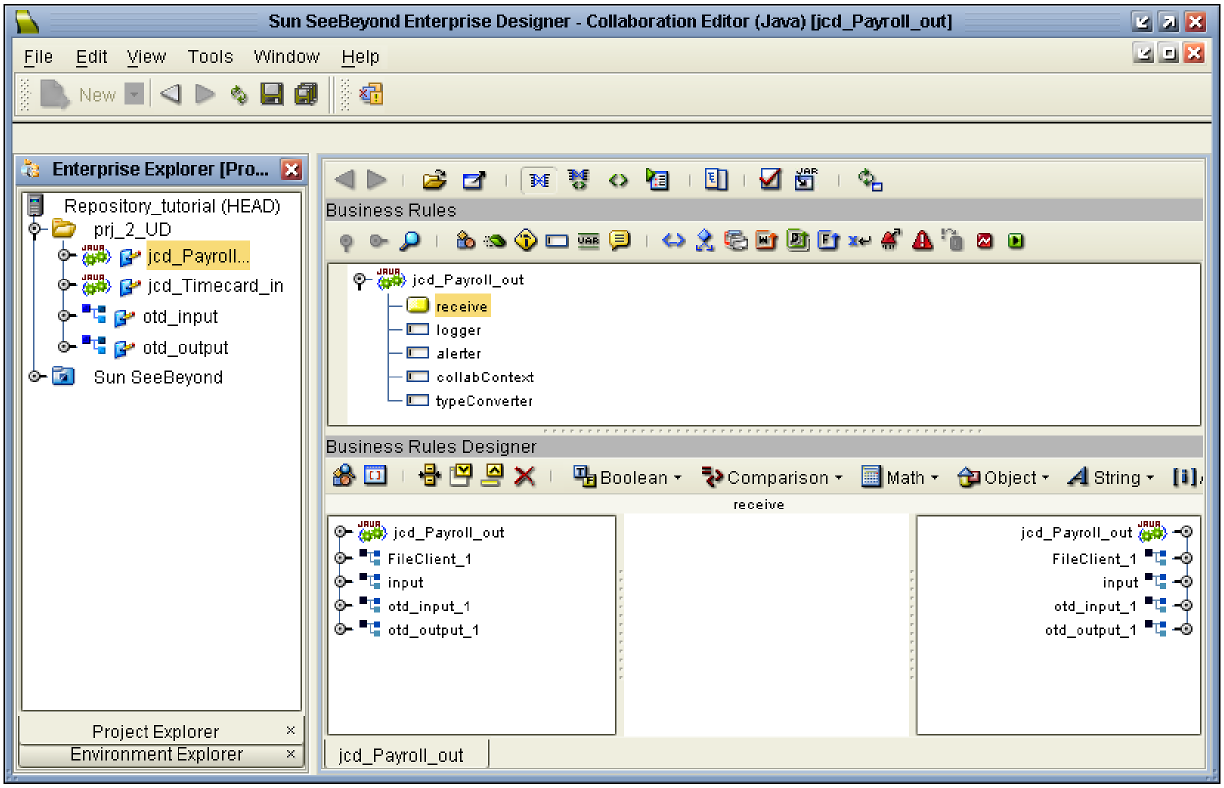
The following is a summary of what you will do in this section.

Overview

- Receive a message from **JMS_1** and unmarshal the string into OTD **otd_input_1**.
- Copy data from the input nodes, **EmployeeNumber**, **LastName**, **FirstName**, and **JobTitle** to the corresponding nodes in **otd_output_1**.
- Set up the multiplication logic (Business Rules) to obtain an amount (**Gross_pay**) for **otd_output_1**.
- Marshal the data from **otd_output_1** to string and write it to a file.

See the following Figure 90.

Figure 90 Business Rules Designer for jcd_Payroll_out



Unmarshal From String

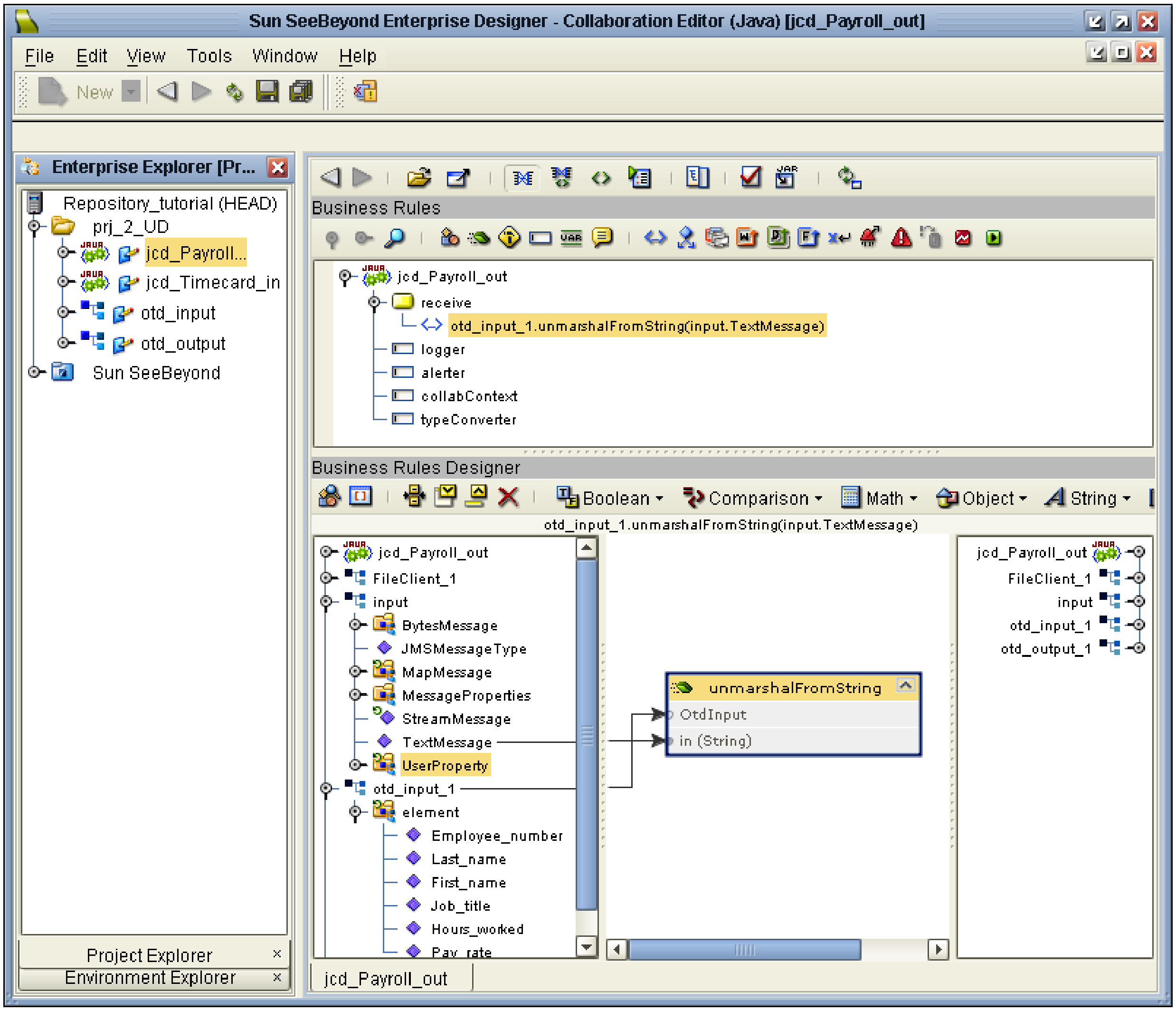
Unmarshal the text input from the Topic into the output OTD.

1. Click the **input** OTD expansion node in the left pane of the Business Rules Designer.
2. Click the **otd_input_1** expansion node, and also expand the element.
3. Click and then drag **otd_input_1** from the left to the center pane of the Business Rules Designer. (You can also right-click **otd_input_1** and select a method to call.)

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

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

Figure 91 Unmarshal From String



```
otd_input_1.unmarshalFromString(input.TextMessage)
```

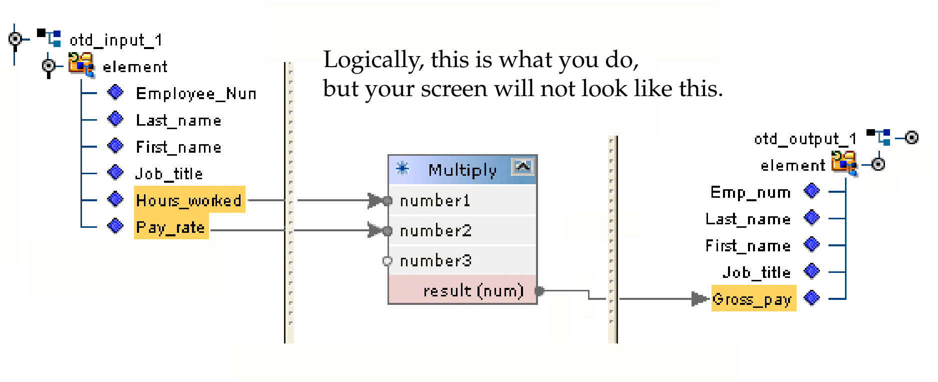
Map Input Elements to Output

In the following steps, map the nodes from **otd_input_1** in the left pane to **otd_output_1** in the right pane, with the elements expanded. Click the **otd_output_1** expansion node in the right pane, and also expand the element.

1. Map **Employee_number** to **Emp_num**.
2. Map **Last_name** to **Last_name**.
3. Map **First_name** to **First_name**.
4. Map **Job_title** to **Job_title**.
5. Click the **Math** drop down list and select the **Multiply** operation.

   The Multiply box appears in the Business Rules Designer pane.
6. Attempt to map the Multiplication operation as shown in the following Figure 92.

Figure 92 Multiplication - Logical



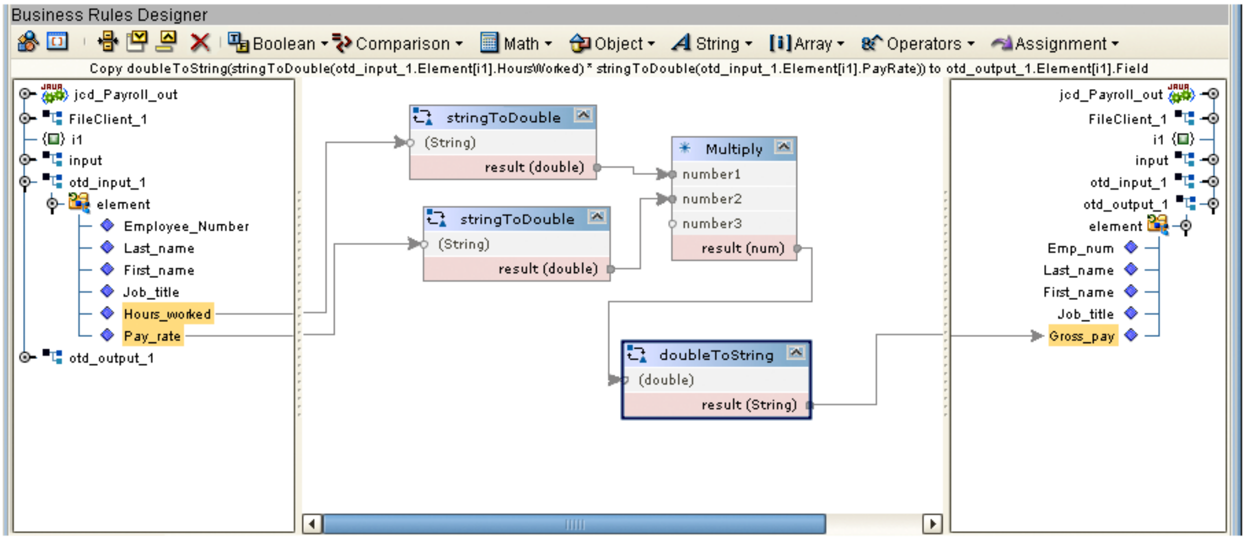
Logically, this is what you do,
but your screen will not look like this.

The system will automatically apply the logic and method boxes to convert **stringToDouble** and **doubleToString**.

7 Click **OK** to accept the Number to String Conversion defaults when the dialog boxes appear.

See the mapped multiplication operation in the following Figure 93.

Figure 93 Multiplication - Actual



Business Rules Designer

Copy doubleToString(stringToDouble(otd_input_1.Element[i1].HoursWorked) * stringToDouble(otd_input_1.Element[i1].PayRate)) to otd_output_1.Element[i1].Field

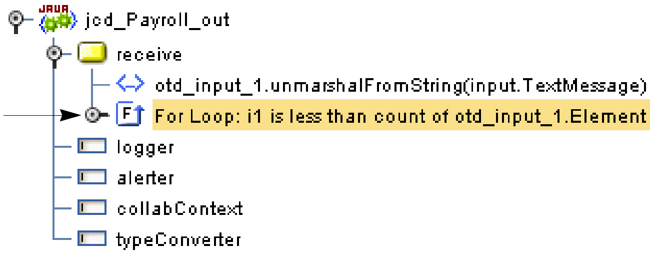
Marshal Text and Write FileClient_1

Close the For loop.

1 In the Business Rules window in the top pane, click the Loop expansion node as shown in the following figure.

This prevents a marshal and write after each record by setting the context to the same level as the For loop. All new rules are generated below the current context, which is now the For loop.

Figure 94 Close Loop


jcd_Payroll_out
receive
otd_input_1.unmarshalFromString(input.TextMessage)
For Loop: i1 is less than count of otd_input_1.Element
logger
alerter
collabContext
typeConverter

2 Expand the view of **FileClient_1** in the right pane.

3 Click and drag **otd_Output_1** from the left to the center pane of the Business Rules Designer (or right-click **otd_Output_1**).

A list of methods appears.

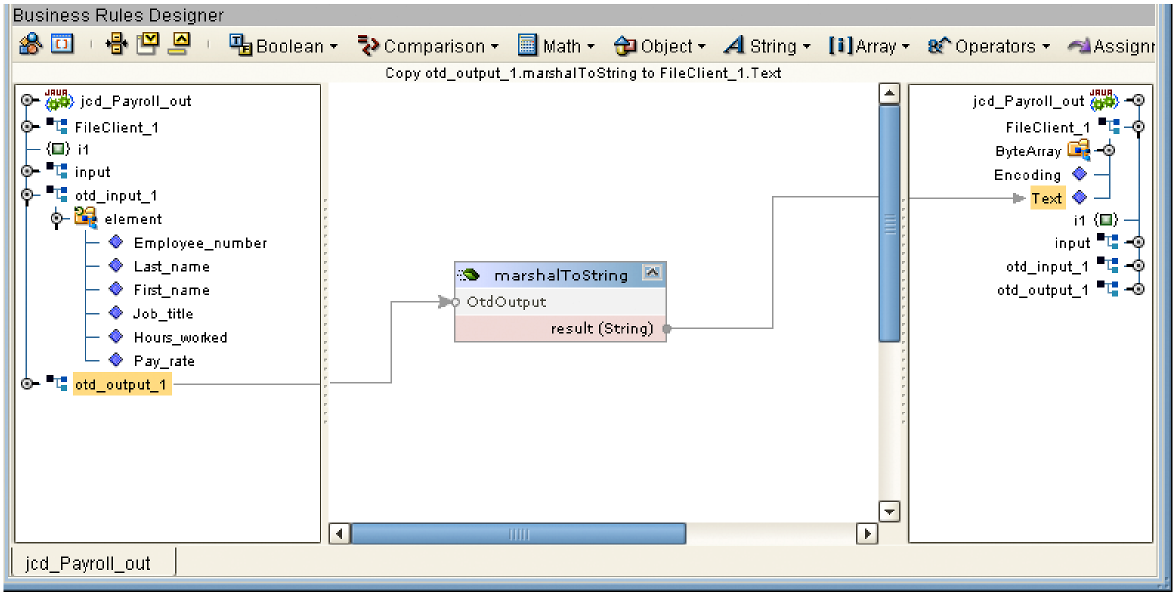
4 Double-click the method **marshalToString()** from the pop-up list.

The **marshalToString** method box appears in the Business Rules Designer.

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

See the following figure.

Figure 95 Marshal Text


Copy otd_output_1.marshalToString to FileClient_1.Text

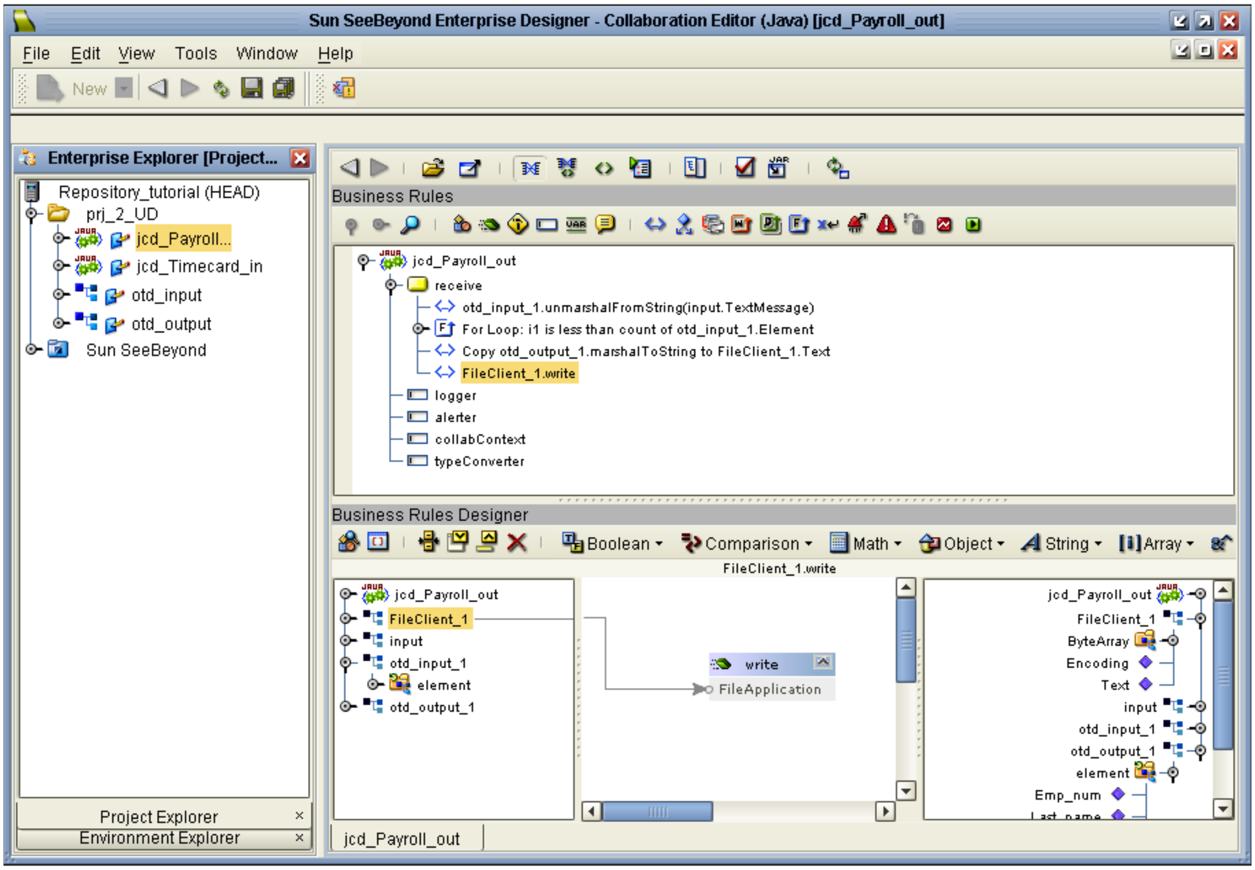
As a final step use the OTD **FileClient_1** to write your output file.

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

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

See the following figure.

Figure 96 Write File Application



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

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

This completes **jcd_Payroll_out** using the Business Rules Designer.

4.10 Create a Connectivity Map

Create a Connectivity Map to define the data flow between the Timecard system and the Payroll system. In this section, bind your Collaboration Definitions with Services.

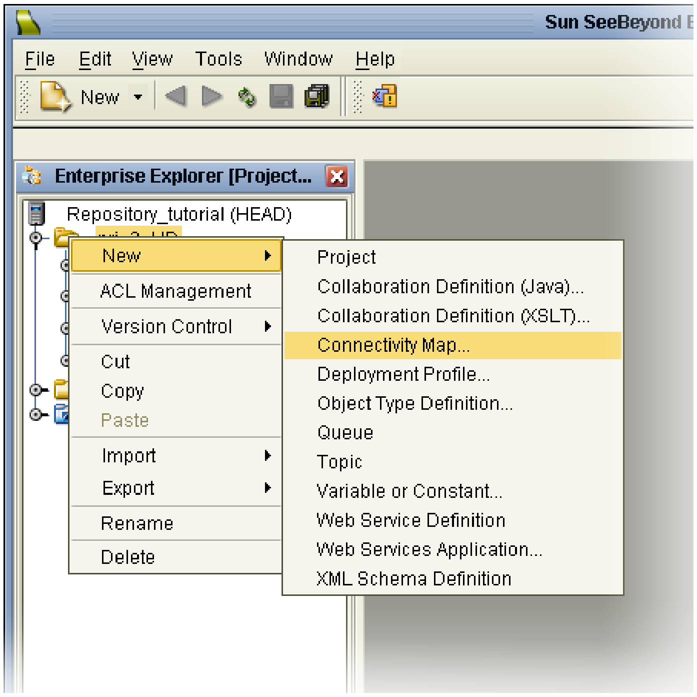
This section describes how to create the Connectivity Map for **prj_2_UD**.

Name Connectivity Map

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

1 In the Project Explorer tree, right-click the **prj_2_UD** icon, then click **New**, and **Connectivity Map**.

Figure 97 Create a Connectivity Map


Sun SeeBeyond
File Edit View Tools Window Help
New
Enterprise Explorer [Project...
Repository_tutorial (HEAD)
New
ACL Management
Version Control
Cut
Copy
Paste
Import
Export
Rename
Delete
Project
Collaboration Definition (Java)...
Collaboration Definition (XSLT)...
Connectivity Map...
Deployment Profile...
Object Type Definition...
Queue
Topic
Variable or Constant...
Web Service Definition
Web Services Application...
XML Schema Definition

2 Name or rename your Connectivity Map, **cm_2_UD** (The optional prefix cm stands for Connectivity Map.)

3 Press **Enter**.

cm_2_UD is added to your Project Explorer tree, as shown in the following figure.

Figure 98 Connectivity Map Icon

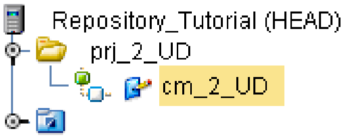
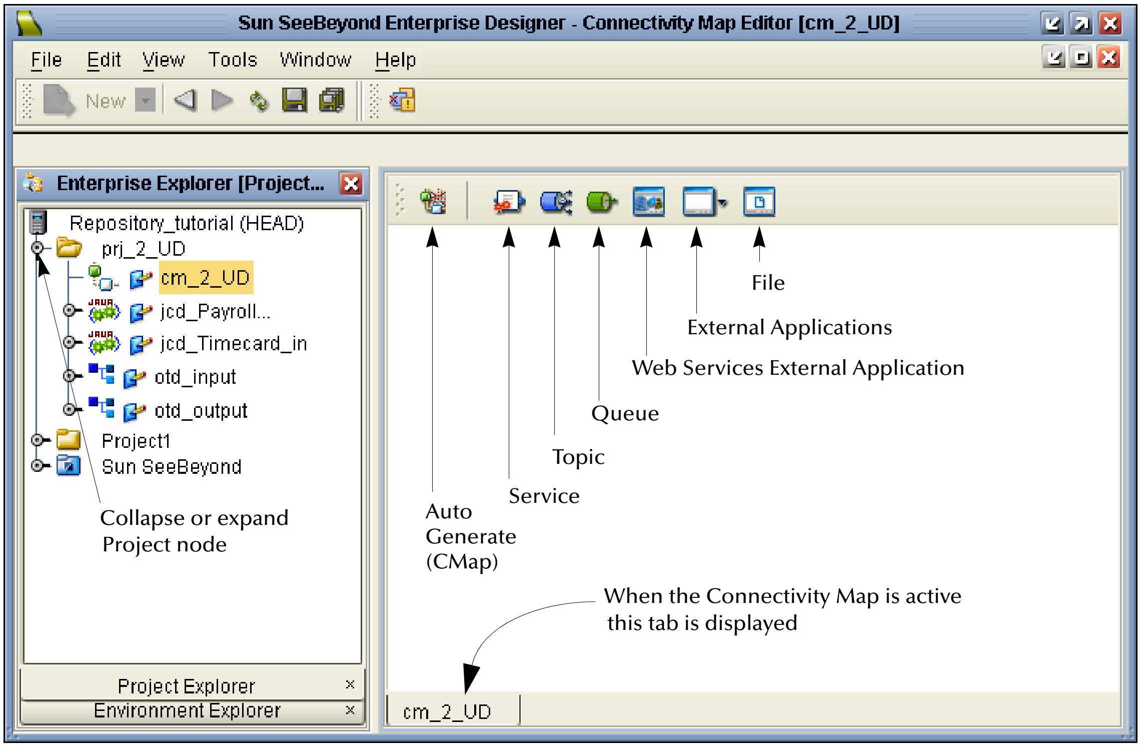


Figure 99 prj_2_UD with a Connectivity Map (cm_2_UD)



4.11 Add Objects to the Connectivity Map

In this section you will add components to the Connectivity Map. Later you will link the components as shown in Figure 63 on page 74.

4.11.1 Populate the Connectivity Map

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

- **ea_Input** (external application File system)
- **svc_Timecard_in**
- **Topic1**
- **svc_Payroll_out**
- **ea_Output**

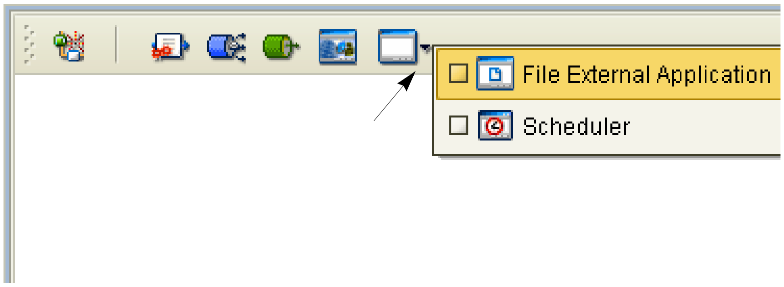
Get Input Object

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

1. On the Connectivity Map toolbar, click the External Application arrow (shown in Figure 100) to display the list.

2 Click **File External Application**.

Figure 100 External Application Selection



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

Note: *If the Service options do not display, or if the File External Application option does not appear in the drop-down list, you may need to install FileeWay.sar. Also, make sure the Connectivity map is active. Double-click the Connectivity map icon if necessary.*

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

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

4 Name the File icon, **ea_Input** (external application input file). Rename files by Right-clicking the label. The name should not contain any spaces.

5 Press **Enter**.

Add a Service Component for Input

Add a Service to contain binding information about connecting the input Timecard to a JMS Topic.

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

This adds a new Service called **cm_2_UD_Service1**.

2 Right-click the **cm_2_UD_Service1** icon and rename it to **svc_Timecard_in**. (The optional prefix svc stands for Service.)

3 Press **Enter**.

Services are components that interact through messages.

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

Get Topic

The JMS Topic conforms to the publish-and-subscribe messaging domain, where one publisher broadcasts messages to potentially many subscribers.

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

  This adds a new Topic called **Topic1**. Accept the default name, **Topic1**.

Get Service Component for Output

Add a Service to contain binding information about connecting the Topic to the output payroll File.

1 On the Connectivity Map toolbar, drag another Service icon into the Connectivity Map to the right of the **Topic1** icon.

2 Rename the **Service** as **svc_Payroll_out**.

3 Press **Enter**.

Get Output Object

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

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

  This icon represents your external application (output) file.

2 Name the file, **ea_Output** (external application output file). Rename the file by right-clicking the icon. (The name should not contain any spaces.)

3 Press **Enter**.

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

4.11.2 Link Components in the Connectivity Map

Linking the components creates a logical flow of data between External Applications via eWays. In this scenario use the **Connectivity Map Generator** button on the tool bar.

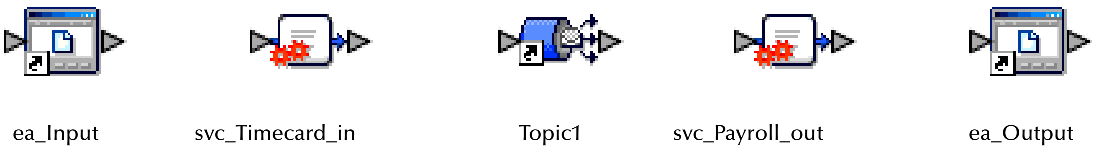
At this point, the Connectivity Map contains five unlinked objects. You are now ready to link them.

Define the Input Service

You define a Service by dragging a Collaboration Definition into a Service component. In the following steps you will bind a Collaboration to the **svc_Timecard_in** Service.

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

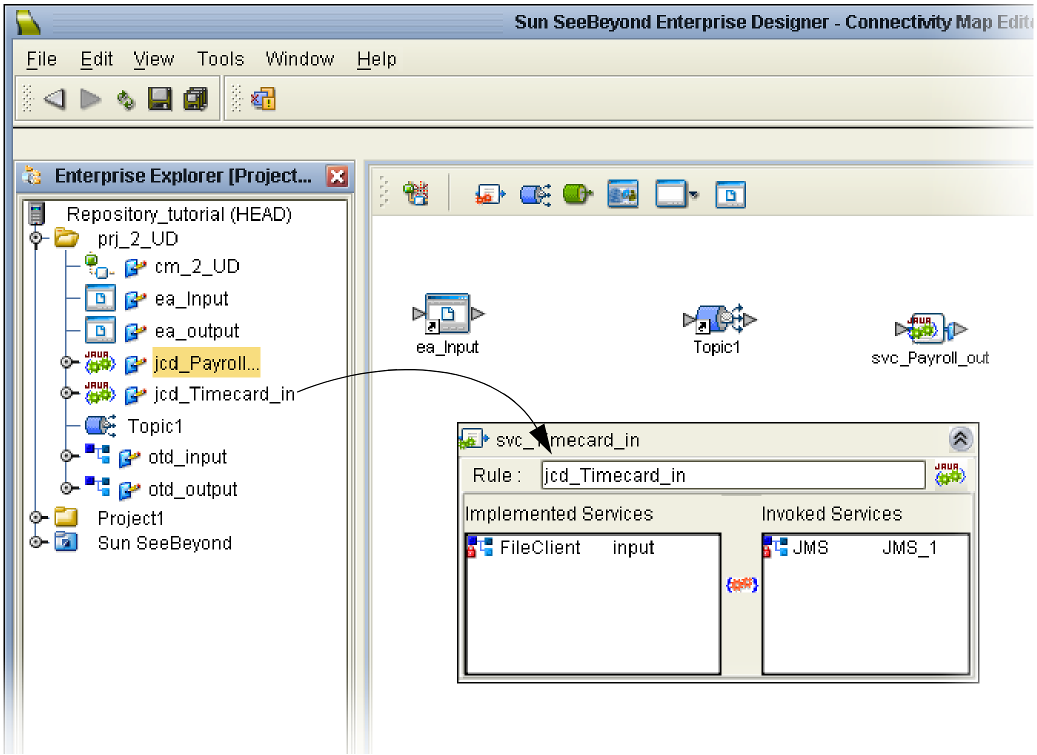
Figure 101 Unlinked Objects in Connectivity Map



2 Double-click the Service icon, **svc_Timecard_in**, in the Connectivity Map to open the Service box.

3 Click and then drag the Collaboration **jcd_Timecard_in** from the Project Explorer tree to the **Rule** field in the open Service box as shown in the following figure.

Figure 102 Using the Drag-and-Drop Method



Note: *Click once in the Project Explorer tree to activate the window, then click the Collaboration definition icon. When a + sign appears next to your cursor you can drag the icon. Notice that after you drag the Collaboration definition to the Rule field, the gear icons change from red to green to signify that the Service is defined.*

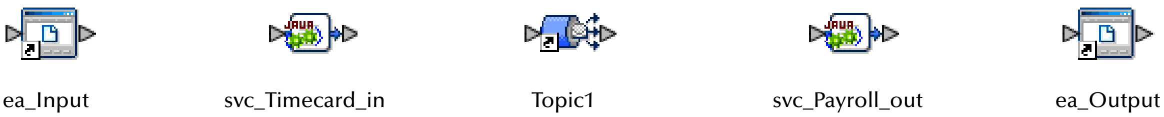
Note: *You may also drag the **jcd_Timecard_in** definition into the Service icon without opening the Service dialog box.*

4 Click the **close** button to close the Service dialog box.

5 Similarly, configure the **svc_Payroll_out** Service using the **jcd_Payroll_out** Collaboration definition.

6 Click **Save All**.

Figure 103 Unlinked Connectivity Map

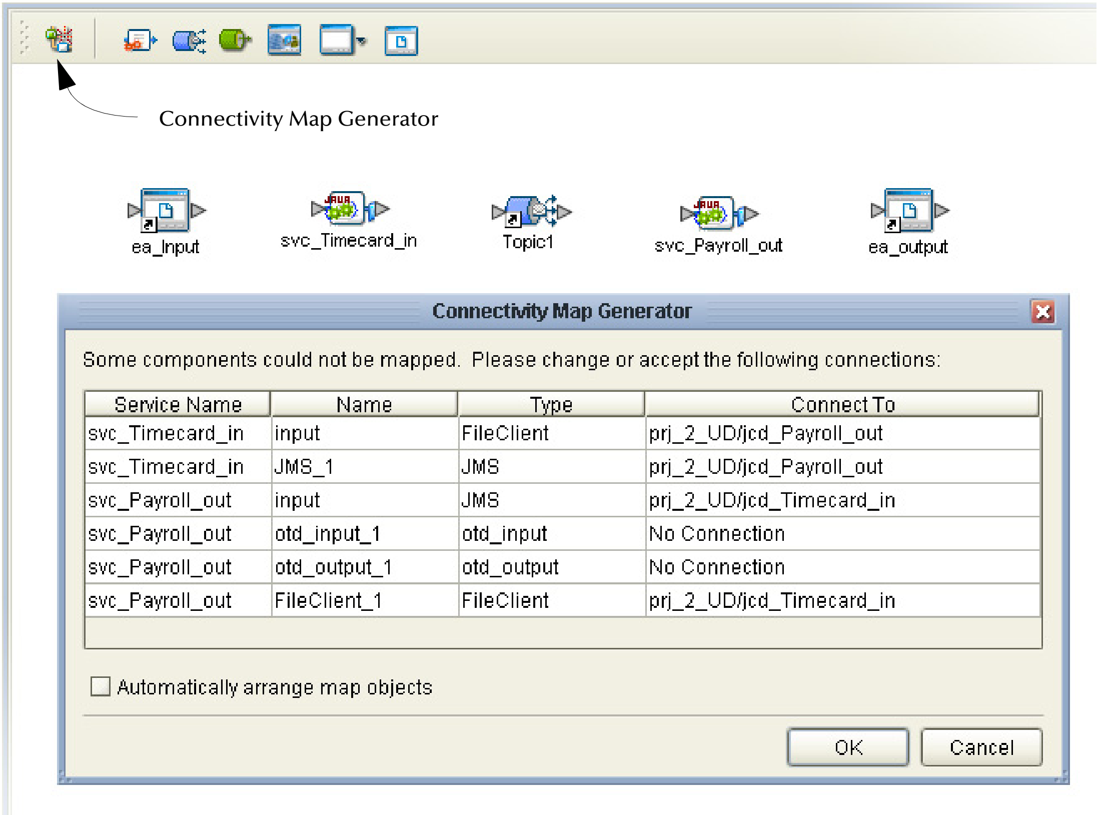


Connectivity Map Generator

1 Click the **Connectivity Map Generator** button on the toolbar.

The Connectivity Map Generator appears with mappable components.

Figure 104 CMap Generator Options



Some components could not be mapped. Please change or accept the following connections:

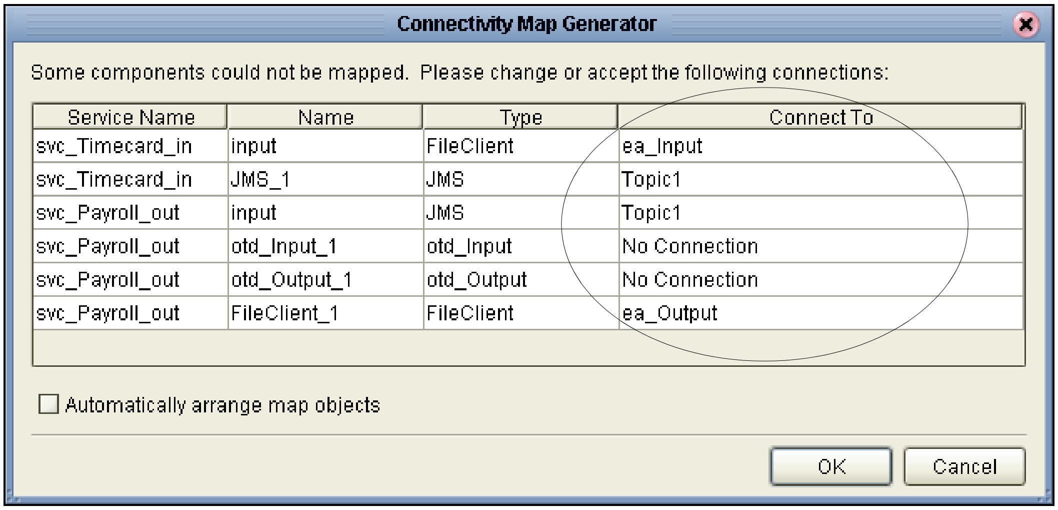
Service Name	Name	Type	Connect To
svc_Timecard_in	input	FileClient	prj_2_UD/jcd_Payroll_out
svc_Timecard_in	JMS_1	JMS	prj_2_UD/jcd_Payroll_out
svc_Payroll_out	input	JMS	prj_2_UD/jcd_Timecard_in
svc_Payroll_out	otd_input_1	otd_input	No Connection
svc_Payroll_out	otd_output_1	otd_output	No Connection
svc_Payroll_out	FileClient_1	FileClient	prj_2_UD/jcd_Timecard_in

☐ Automatically arrange map objects

2 Click inside the **Connect To** field of each component, and then select the component.

Refer to the following figure. **Do Not** click **OK** until all the components are selected.

Figure 105 Select CMap Generator Components



Connectivity Map Generator

Some components could not be mapped. Please change or accept the following connections:

Service Name	Name	Type	Connect To
svc_Timecard_in	input	FileClient	ea_Input
svc_Timecard_in	JMS_1	JMS	Topic1
svc_Payroll_out	input	JMS	Topic1
svc_Payroll_out	otd_Input_1	otd_Input	No Connection
svc_Payroll_out	otd_Output_1	otd_Output	No Connection
svc_Payroll_out	FileClient_1	FileClient	ea_Output

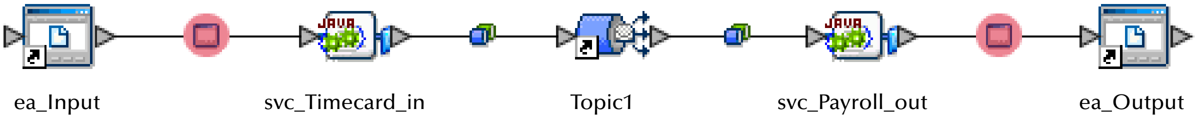
☐ Automatically arrange map objects

Note: *otd_input_1* *and* *otd_Output_1* *represent your input and output OTDs. They do not need to be connected to any other objects on the Connectivity Map.*

3 When your **Connect To** components match those shown in Figure 105 click **OK**.

Note: *The components in your Connectivity Map auto-connect. The red circles around the eWays indicate that they have to be configured; you do that in the following steps.*

Figure 106 Connected CMap Components



4.12 Configure eWays and JMS Connections

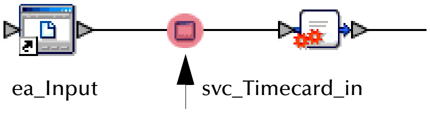
Your Connectivity Map contains two eWays and JMS connections. The eWays must be configured for deployment.

- **Inbound eWay**: The inbound eWay polls a directory on the local File system, for example: **C:\eGateData\Project2**. In this scenario the **eWay** reads the text input File (***.txt**).
- **Outbound eWay**: The outbound eWay writes the results to an output File (.txt) on the local File system, **C:\eGateData\Project2**.

Configure the Inbound eWay

Identify the inbound file.

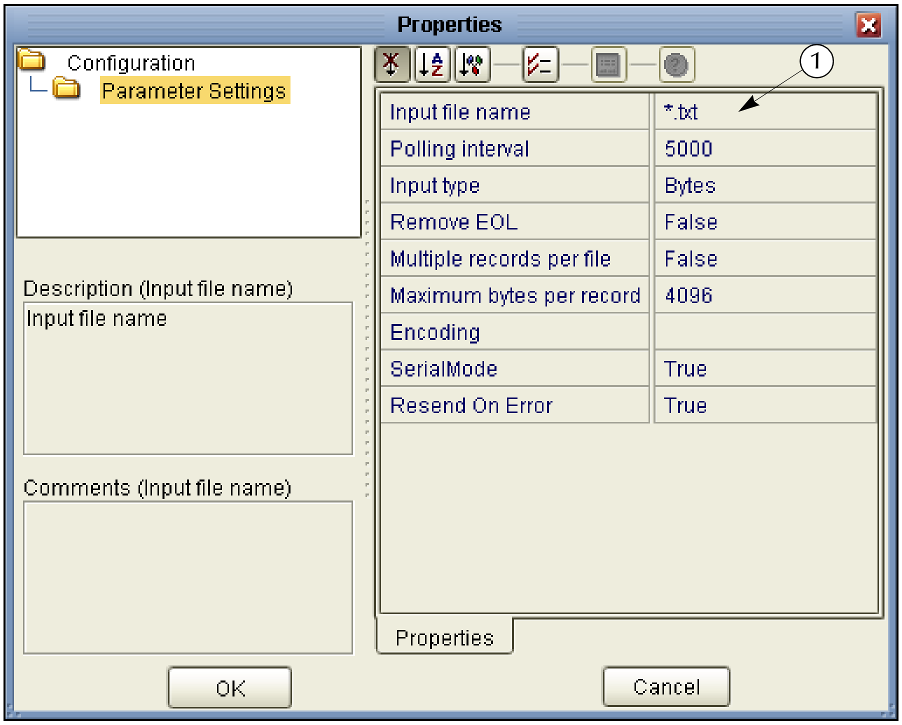
Figure 107 Inbound eWay



1 Double-click the first eWay.

The Properties dialog box appears.

Figure 108 Properties Configuration (Inbound eWay)



Input file name

Input file name	*.txt
Polling interval	5000
Input type	Bytes
Remove EOL	False
Multiple records per file	False
Maximum bytes per record	4096
Encoding	
SerialMode	True
Resend On Error	True

2 Enter ***.txt** in the **Input File name** field (number 1 in Figure 108).

The eWay reads any files with the **.txt** extension.

Accept the other default settings.

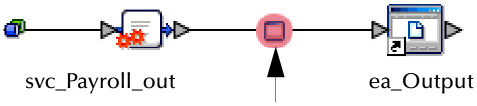
After the eWay is configured, the red circle around the eWay disappears.

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

Configure the Outbound eWay

Identify the outbound file.

Figure 109 Outbound eWay



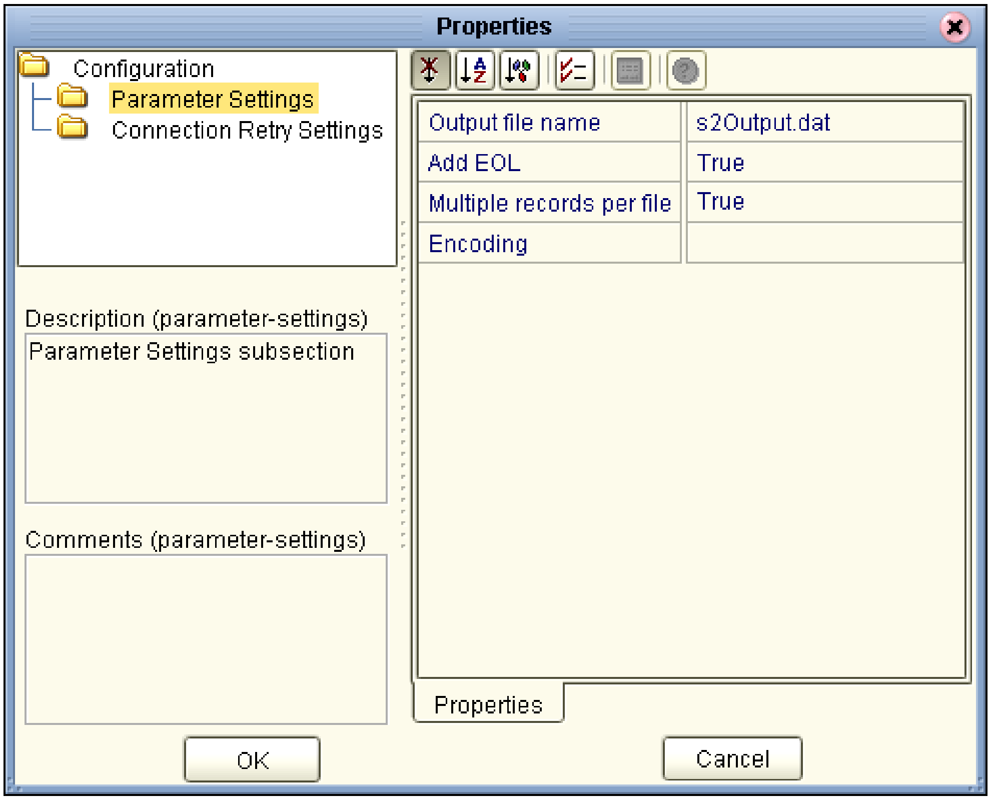
1 Double-click the second eWay. Select **Outbound File eWay**.

The Properties dialog box appears.

2 Enter the name of your output File: **s2Output.dat**.

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

Figure 110 Properties Configuration (Outbound eWay)



Description (parameter-settings)

Parameter Settings subsection

Comments (parameter-settings)

Output file name	s2Output.dat
Add EOL	True
Multiple records per file	True
Encoding	

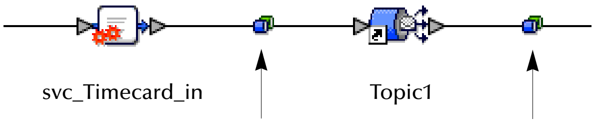
Note: *__Add EOL__ This is the line.separator property, and it defaults to "\n." A "True" setting means an end of line character(s) is added when the record is written. The EOL character(s) used is based on the Java system property, line.separator. The File eWay grabs the system property (which is OS-specific) and appends that, if __Add EOL__ is True. This setting has no effect when the input file is read.*

Note: *The __Multiple records per File__ setting determines whether records are concatenated in the output file. A True setting (default) means that multiple records are written* *in the same file. A False setting means that only one record is written to one file. See* **Multiple records per File (True or False)** *on page 59 in Scenario1.*

JMS Client Configuration

The JMS Client Configuration icons appear on both sides of the **Topic1** icon, as shown by the arrows in the figure below.

Figure 111 JMS Client Properties Icon



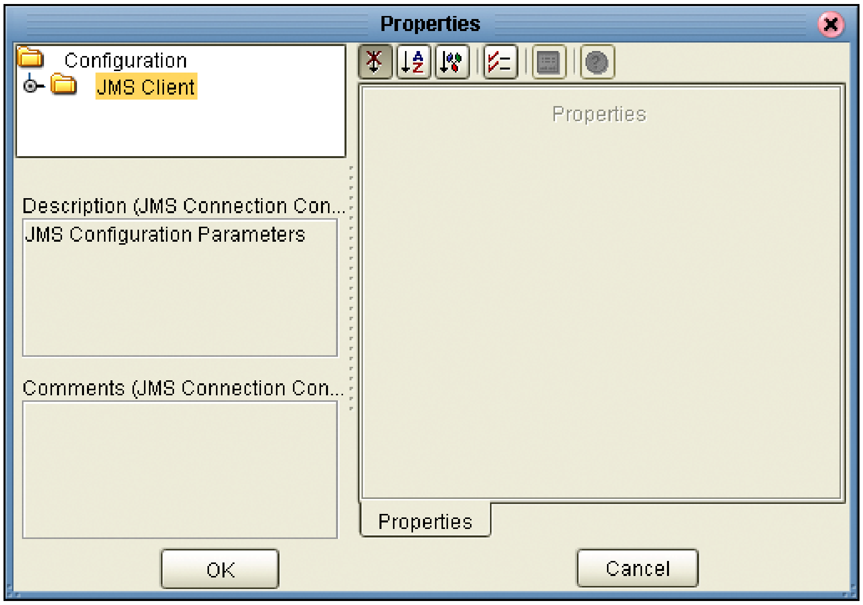
Select JMS Client Configuration

For this Tutorial accept the default configurations.

1. Double-click the first of the two JMS icons (see previous Figure 111).

   The JMS Client properties dialog box appears. You can set the basic configuration and the connection security parameters here. In this tutorial, accept the defaults.

Figure 112 JMS Client Properties


Properties
Configuration
JMS Client
Description (JMS Connection Con...
JMS Configuration Parameters
Comments (JMS Connection Con...
Properties
Properties
OK
Cancel

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

   (Since you accepted the defaults, the previous steps were not required and are for illustrative purposes only.)

You are now ready to deploy your Project 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.

4.13 Create an Environment

Create a Logical Host and an External File System using Environment Explorer. First create an Environment. The Environment Explorer deploys resources required to implement a project and includes information about external systems that interact with eGate. If you already created an Environment in one of the Tutorial scenarios, you may use that same Environment.

Note: *If you are using an existing Environment, you can also run this project in the same instance of the Logical Host (domain1 for example). But make sure you are using the correct input data, and make sure you are using the correct path to the data, as explained in this section.*

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.

Add Logical Host and Servers

1 Click **View** on the Menu bar. Click **Environment Explorer** (or click the **Environment Explorer** tab).

2 Right-click the Repository name (computer icon). Click **New Environment**.

3 Right-click **Environment1** and rename it to **Tutorial**. Press **Enter**.

4 Right-click **Tutorial**. Click **New**, **Logical Host**.

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

Add an Integration Server

The Integration Server is a J2EE software platform that houses the business logic container used to run Collaborations and JCA connectors (eWays).

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

IntegrationSvr1 appears in the **LogicalHost1** box. Your Collaboration will be bound to this server.

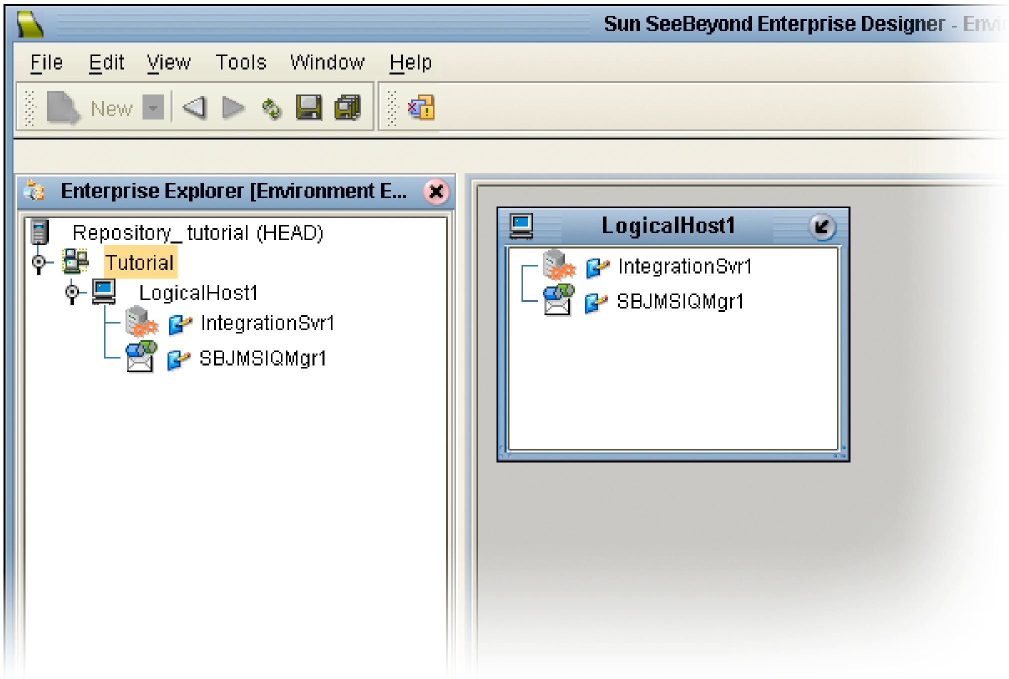
Add a JMS IQ Manager

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 appears in the **LogicalHost1** box.

Figure 113 Environment with Logicalhost


Sun SeeBeyond Enterprise Designer - Envi
File Edit View Tools Window Help
New
Enterprise Explorer [Environment E...
Repository_ tutorial (HEAD)
Tutorial
LogicalHost1
IntegrationSvr1
SBJMSIQMgr1
LogicalHost1
IntegrationSvr1
SBJMSIQMgr1

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

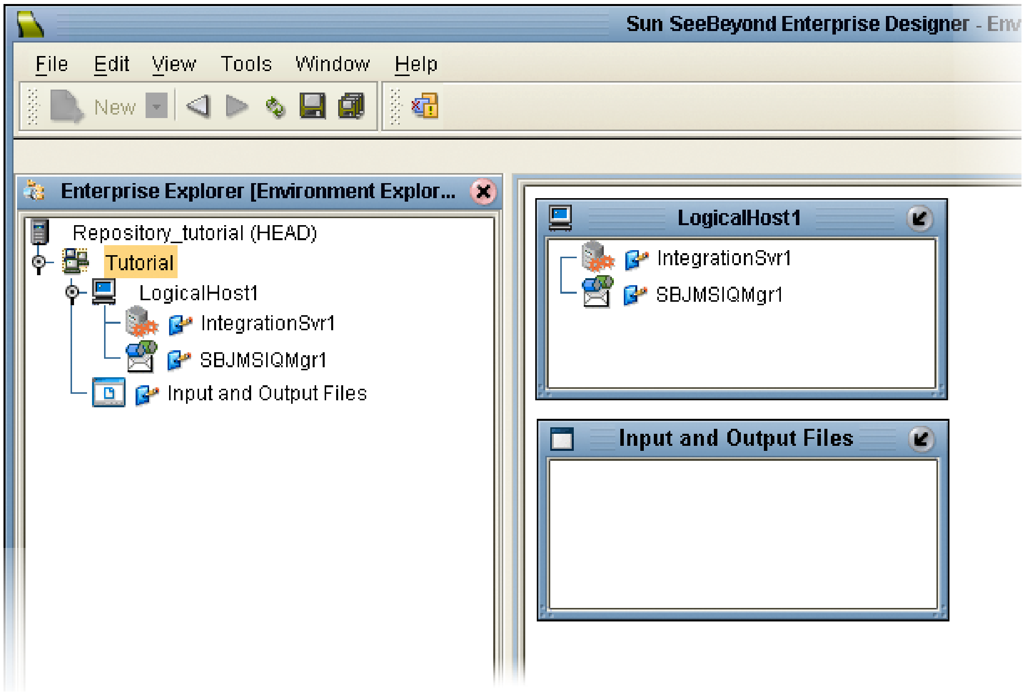
Add an External File System for the eWays

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

   This creates a container in the Environment Editor to hold your File eWays. You will place both input and output files in this container.
3. Name the External System **Input and Output Files**.
4. Click **OK**. (You may have to move a GUI if it is blocking an underlying GUI.)

   Your Environment Editor pane should look similar the following figure. (You may have to move the **Input and Output Files** box if it is blocking the **LogicalHost1** box.)

Figure 114 Environment Editor


Sun SeeBeyond Enterprise Designer - Env
File Edit View Tools Window Help
New
Enterprise Explorer [Environment Explor...
Repository_tutorial (HEAD)
Tutorial
LogicalHost1
IntegrationSvr1
SBJMSIQMgr1
Input and Output Files
LogicalHost1
IntegrationSvr1
SBJMSIQMgr1
Input and Output Files

4.13.1 Set Properties for Servers and External Files

Set the properties for your servers and external files. Make sure you have entered your user name and password for the Integration Server, and the JMS IQ Manager, and make sure an instance of the Logical Host (domain) is running before you deploy your project.

Set Properties for External Files

Make sure you are in the Environment Explorer.

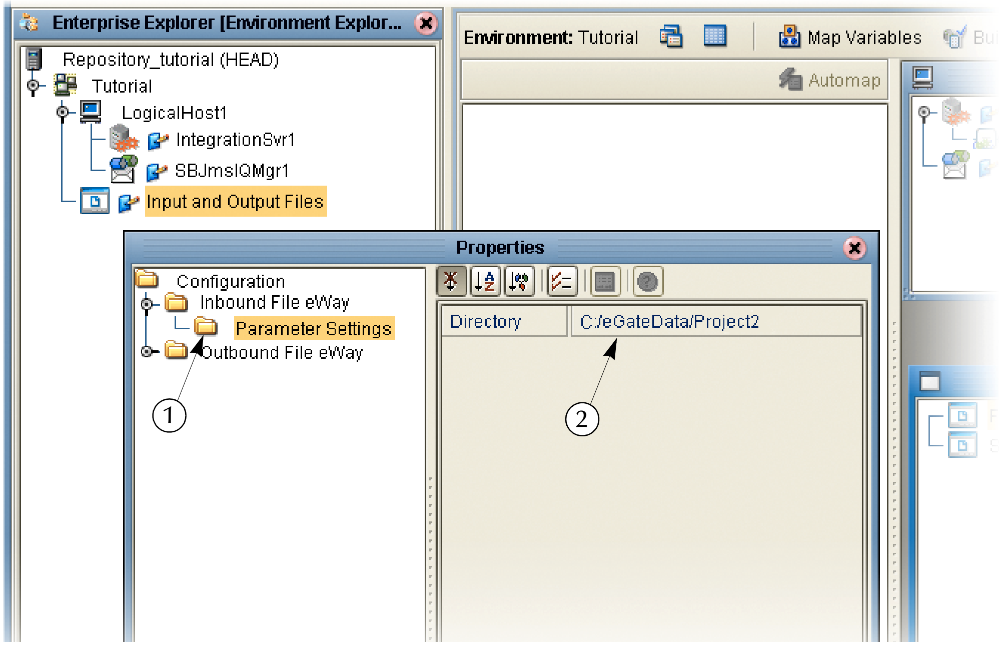
1 Right-click **Input and Output Files** in the left pane.

A pop-up box appears.

2 Select **Properties**.

The Properties dialog box appears. Your Directory path should point to **Project2** for scenario2.

Figure 115 Properties, Parameter Settings



Directory C:/eGateData/Project2

3 Expand the view under **Inbound File eWay** and select **Parameter Settings** (number 1 in Figure 115).

4 Set the Directory path to point to your input data file (number 2 in Figure 115).

5 Repeat these steps for your **Outbound File eWay**.

Set Properties for the Integration Server

Make sure you are in the Environment Explorer.

1 Right-click **IntegrationSvr1** in the Environment Explorer tree.

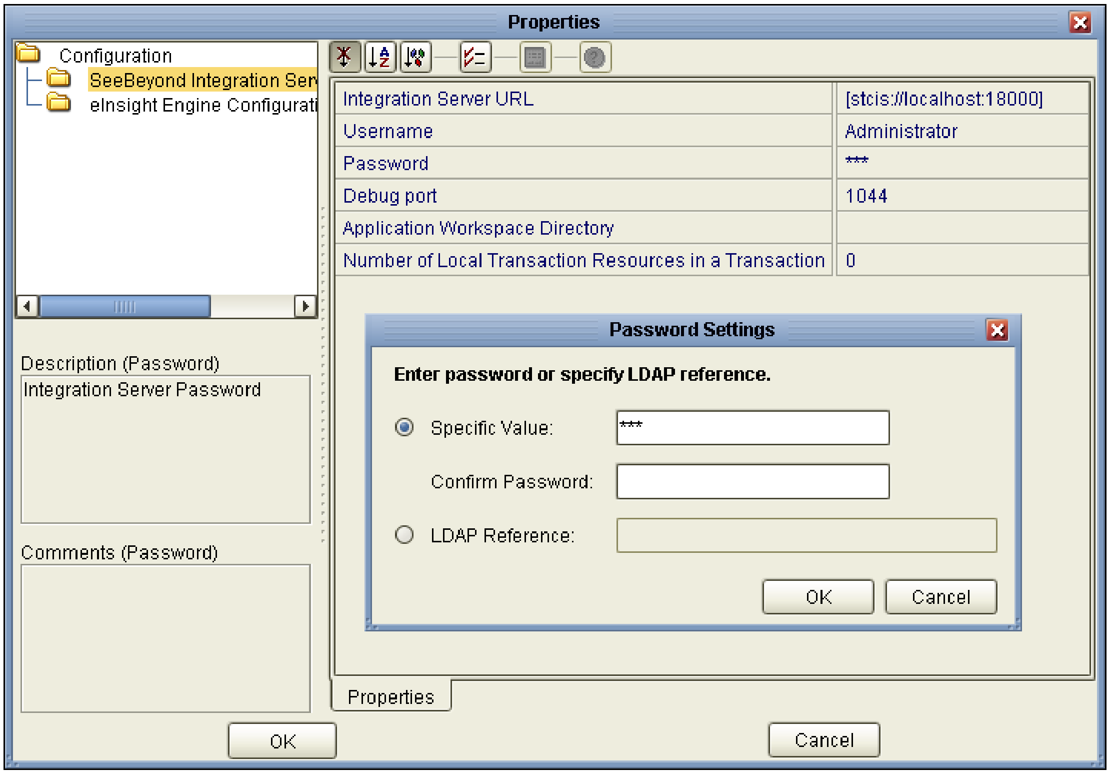
2 Click **Properties**.

3 Enter and confirm **Password**.

Note: *Make sure the port for your Integration Server (shown in the figure below with the default 18000) is your domain port.*

See the following figure.

Figure 116 Integration Server Password



Integration Server URL	[stcis://localhost:18000]
Username	Administrator
Password	***
Debug port	1044
Application Workspace Directory	
Number of Local Transaction Resources in a Transaction	0

Description (Password)
Integration Server Password

Comments (Password)

Password Settings

Enter password or specify LDAP reference.

● Specific Value: ***

Confirm Password:

○ LDAP Reference:

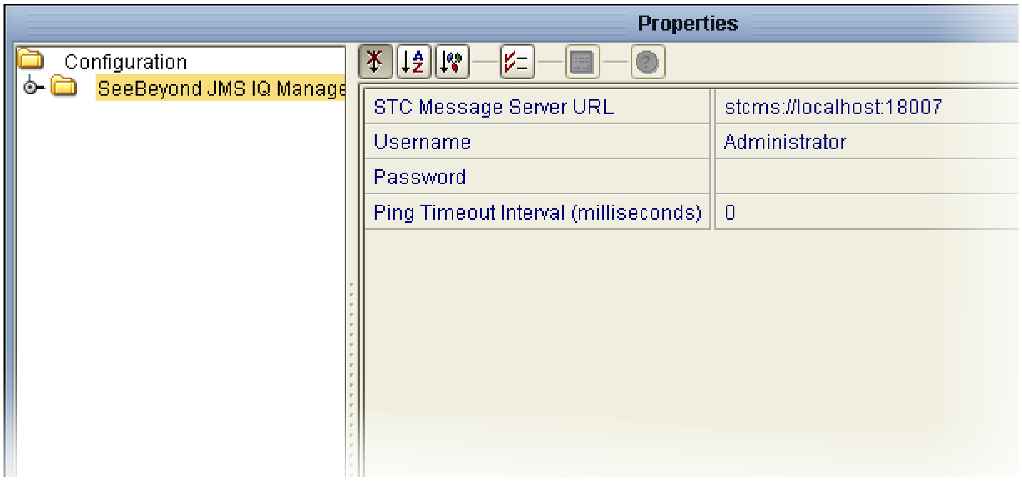
4 Click **OK**.

Set Properties for the JMS IQ Manager

Make sure you are in the Environment Explorer.

Similar to the previous four steps for the Integration Server, enter a password for the JMS IQ Manager, and verify the port, which should match the running domain’s IQ Manager.

Figure 117 Message Server



STC Message Server URL	stcms://localhost:18007
Username	Administrator
Password	
Ping Timeout Interval (milliseconds)	0

5 Click **OK** and **Save All**.

4.14 Create and Activate the Deployment Profile

First create a Deployment Profile, and then deploy (map) the components. **prj_2_UD** can run on the same instance of the Logical Host (domain1 for example) as other Tutorial Projects, and it can run at the same time on the Integration Server. You still must create a Deployment Profile for **prj_2_UD**.

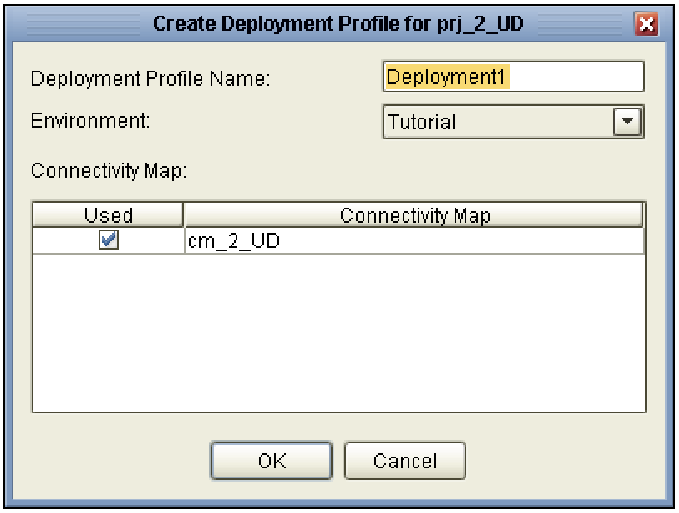
Create a Deployment Profile

A Deployment Profile contains information about how Project components are mapped and deployed within an Environment.

1 Click the **Project Explorer** tab to return to Project Explorer in your Project.

2 Right-click **prj_2_UD**, and then click **New**, **Deployment Profile**.

Figure 118 Deployment Profile



Create Deployment Profile for prj_2_UD

Deployment Profile Name: Deployment1

Environment: Tutorial

Connectivity Map:

Used	Connectivity Map
☑	cm_2_UD

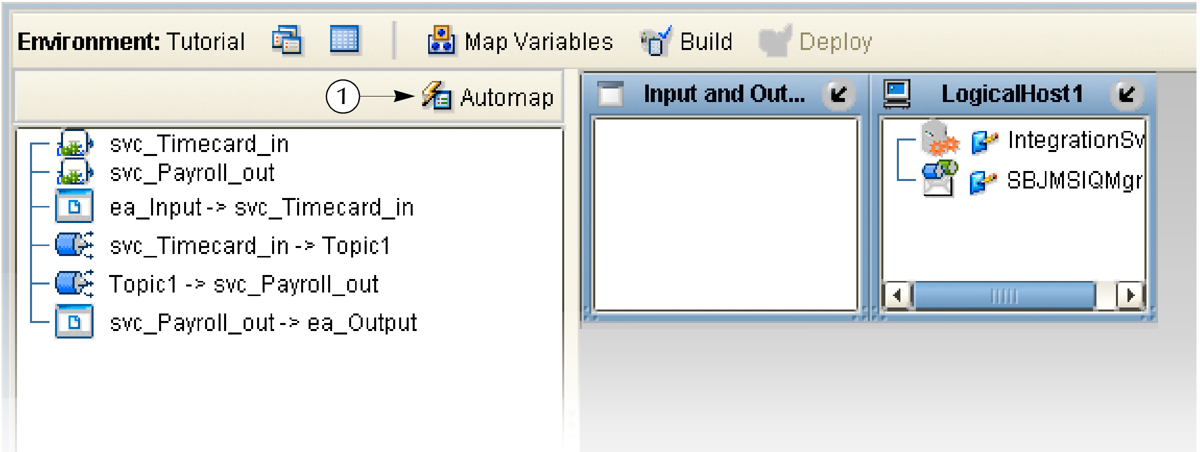
3 Accept the default name **Deployment1** (and Environment **Tutorial**). Click **OK**.

Your files and Services appear in the center pane of the Environment editor. Deploy these components using the **Automap** feature.

4 Click the **Automap** button in the Environment tool bar (number 1 in Figure 119).

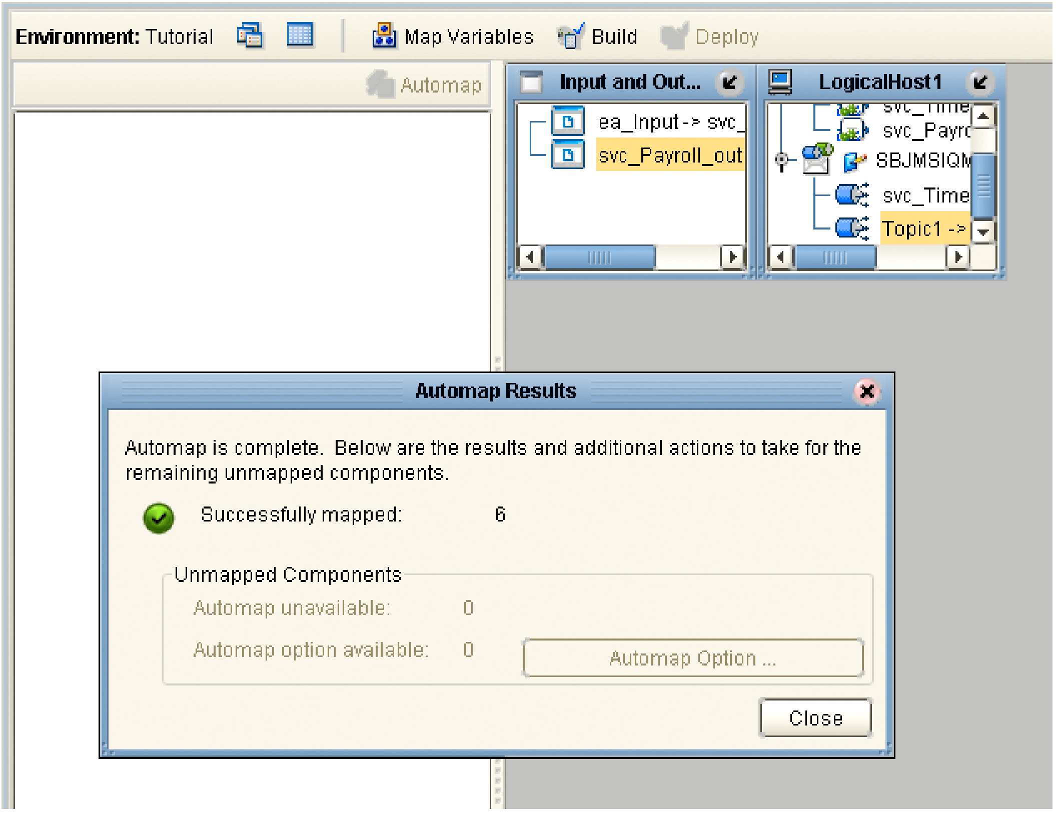
Your components automatically map.

Figure 119 Environment, Drag and Drop Components



5 Close the Automap Results dialog box.

Figure 120 Automap Results



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

✓ Successfully mapped: 6

Unmapped Components

Automap unavailable: 0

Automap option available: 0

The center pane of the Environment Editor should be empty after all the available objects and components have been mapped to the Environment. You are now ready to build and deploy your Project.

6 Save your Project.

4.15 Build and Deploy the Project

In this section you deploy and run your Project. This involves a Build to compile files and create a project EAR File and a Deploy to start the Integration Server to run your project.

4.15.1 Domain Manager

Make sure an instance of the Logical Host is running (domain) before you deploy your project. You can start the domain with the **domainmgr.bat** script that is located in your Java CAPS folder, **logicalhost** directory.

The start script is only present after you have created/installed the domain. The Domain Manager will prompt you to create a domain if no domain is present.

See the Deploying Applications section of the *Sun SeeBeyond eGate Integrator System Administration Guide*.

- Run **domainmgr.bat** then start your domain from the dialog box.

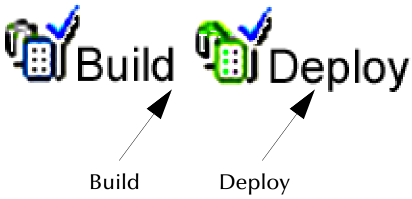
or

- Double-click **start_domain1.bat** to run domain1.

4.15.2 Build

The Build compiles the Service(s) and Java files, then creates the Project EAR File.

Figure 121 Environment Toolbar Buttons



1. Click the **Build** button in the Environment toolbar.

   After the Project EAR File is created the following message appears: “Project build was successful (elapsed time: xx:xxx seconds).”

Note: *A successful build places a new application in your Enterprise Designer:*

```
<Enterprise Designer/builds/<ProjectNameDPName>/<LH>/<IS> directory
```

For example:

```
C:\JavaCAPS51\edesigner\builds\prj_2_UDDeployment1\LogicalHost1\Integ
rationSvr1\prj_2_UDDeployment1.ear).
```

This is the project that you will deploy in the following sections. You may also deploy, enable, disable, or undeploy the project using the Enterprise Manager.

4.15.3 Deploy

Deploy your Project with an instance of the Logical Host (domain) running. The executable domain files are in your Logicalhost folder.

1. Click the Deploy button in the Environment toolbar.

   You will receive the following message when the deployment is successful: "Project deployment was successful on host (LogicalHost1), server (IntegrationSv1) (elapsed time: xx.xxx seconds)."

2. Click **OK** to deploy your project.

Note: *If your current deployment is enabled (not disabled or undeployed) your Project will continue to execute until you disable or undeploy the Project. You can use the Enterprise Manager to disable or undeploy your Project.*

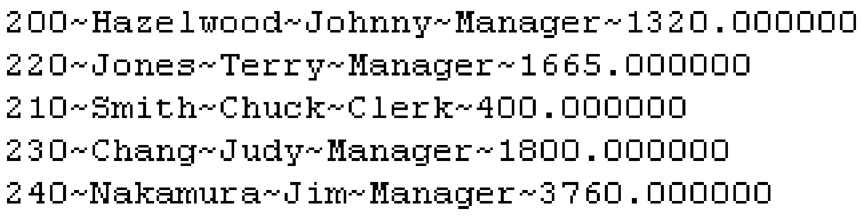
4.16 Verify Output Data

After the Management Agent processes your input File, **Timecard_Data.txt**, and writes the results to **s2Output.dat**, you can check the output and reprocess the input data as often as you wish while your project continues to run.

View Output

1. Use Windows Explorer to navigate to your project data directory; for example: **C:\eGateData\Project2**.
2. See that the Input eWay, after processing the input file, renamed the input file to **Timecard_Data.txt.~in**.
3. See that the Output eWay generated the output File, **s2Output.dat**.
4. Use a text editor to view the contents of the output file.

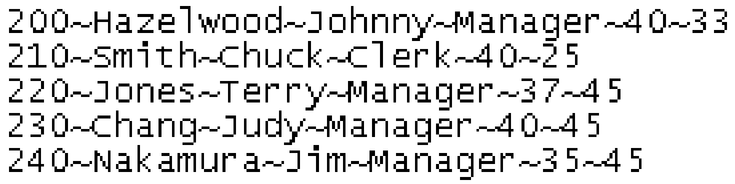
Figure 122 Output File



```
200~Hazelwood~Johnny~Manager~1320.000000
220~Jones~Terry~Manager~1665.000000
210~Smith~Chuck~Clerk~400.000000
230~Chang~Judy~Manager~1800.000000
240~Nakamura~Jim~Manager~3760.000000
```

In this sample, the output file shows the gross pay for each employee (with extra decimal places). See input file, Figure 123.

Figure 123 Input File



```
200~Hazelwood~Johnny~Manager~40~33
210~Smith~Chuck~Clerk~40~25
220~Jones~Terry~Manager~37~45
230~Chang~Judy~Manager~40~45
240~Nakamura~Jim~Manager~35~45
```

You can change the data in your input file and process the input file again – rename **Timecard_Data.txt.~in** to **Timecard_Data.txt**.

The File eWay picks up the input file and writes the results to **s2Output.dat**. The output is written to the same file, since you left "Multiple records per file" set to True.

Chapter 5

Building a Project - Scenario3

5.0.1 Using XSD Objects to Generate XSD Nodes

This Tutorial provides step-by-step procedures for creating and testing an eGate Project using XSDs (XML Schema Definition). In this scenario you import XSDs as XSD Objects which are then exported to XSD Nodes that are used in this Project.

This Project addresses a simple business challenge where an input timecard File is read and a payroll workfile is output.

What's in This Chapter

5.1 Business Challenge

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

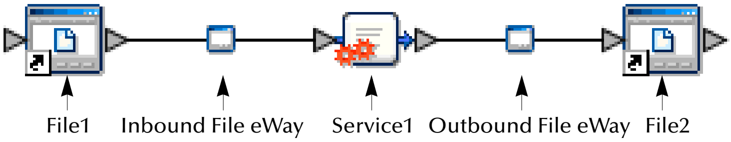
In this very simple scenario **FirstName** and **LastName** from the input file are concatenated into the field **FullName** in the output file, and **PayAmount** is calculated based on multiplying **HoursWorked** times **Rate**.

- The input data in this system is in XML format with six fields: **EmployeeNumber**, **LastName**, **FirstName**, **JobTitle**, **HoursWorked**, and **Rate**.
- The output file in XML format contains the following fields: **FullName**, **EmpNumber** and **PayAmount**.

5.1.1 Project Description

The finished eGate Project contains components used to move the data through the Project and publish the data in an XML output workfile.

Figure 124 Project Connectivity Map

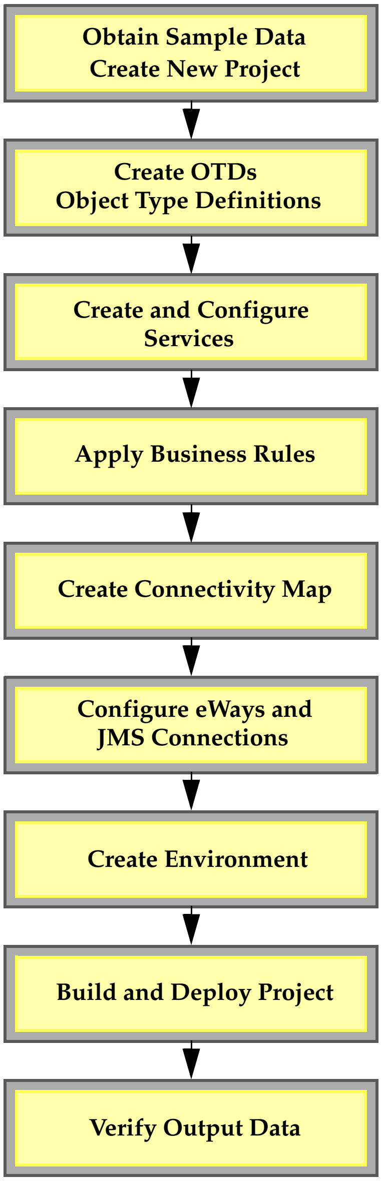


These components perform the following functions:

- **File1**: The Input File *External Application* contains the Input XML data.
- **Inbound File eWay**: The inbound File 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 and performs the multiplication operation.
- **Outbound File eWay**: This eWay publishes the processed data and creates an output file.
- **File2**: This is the output file.

5.1.2 Project Flow Diagram - Scenario3

Figure 125 Project Flowchart - Scenario3



5.2 Sample Data for Project3

This Project uses the sample files found in the Project3 folder of the **eGate Tutorial Sample**. The files used in Scenario3 are: **s3_TimeCard.xml**, **s3_InputSchema**, and **s3_OutputSchema**. (The “s3” in the file names stands for Scenario3.)

5.2.1 Download the Sample files

See **“Download the Sample files” on page 23**.

5.2.2 Sample files

This section shows a sample of the input data used in Scenario3. Your XML File should look similar to the structure and syntax below. In Scenario3 the input file contains two records.

These files and schemas are already created and are included in your sample files.

5.2.3 Input XML File

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2005 sp1 U (http://
www.xmlspy.com)-->
<Employees xmlns="urn:please.define.your.own.target.namespace1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
    <Employee>
        <EmployeeNumber>1234</EmployeeNumber>
        <LastName>Yu</LastName>
        <FirstName>Jen</FirstName>
        <JobTitle>SysEngineer</JobTitle>
        <HoursWorked>40</HoursWorked>
        <Rate>65</Rate>
    </Employee>
      <Employee>
        <EmployeeNumber>4321</EmployeeNumber>
        <LastName>McMurphy</LastName>
        <FirstName>Randall</FirstName>
        <JobTitle>Programmer</JobTitle>
        <HoursWorked>45</HoursWorked>
        <Rate>55</Rate>
    </Employee>
</Employees>
```

5.2.4 Input XML Schema

The purpose of an XML Schema is to define the building blocks of an XML document. It is similar to a DTD. The Input XSD, which represents the input data, is shown below.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="urn:please.define.your.own.target.namespace1"
xmlns:tns="urn:please.define.your.own.target.namespace1"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
<xs:complexType name="ComplexType2">
        <xs:sequence>
            <xs:element name="EmployeeNumber" type="xs:string"/>
            <xs:element name="LastName" type="xs:string"/>
            <xs:element name="FirstName" type="xs:string"/>
            <xs:element name="JobTitle" type="xs:string"/>
            <xs:element name="HoursWorked" type="xs:int"/>
            <xs:element name="Rate" type="xs:int"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="ComplexType1">
```

```
            <xs:sequence>
                <xs:element name="Employee" type="ComplexType2"
                 maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
        <xs:annotation>
            <xs:documentation>Root element</xs:documentation>
        </xs:annotation>
        <xs:element name="Employees" type="ComplexType1"/>
    </xs:schema>
```

5.2.5 Output XML Schema (XSD)

The Output XSD, which represents the output data, is shown below. The actual data for the output file is generated as a text file and does not have to exist ahead of time.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
targetNamespace="urn:please.define.your.own.target.namespace2"
xmlns:tns="urn:please.define.your.own.target.namespace2"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:complexType name="ComplexType2">
    <xs:sequence><xs:element name="EmployeeNumber"
     type="xs:string"/>
            <xs:element name="FullName" type="xs:string"/>
            <xs:element name="EmpNumber" type="xs:string"/>
            <xs:element name="PayAmount" type="xs:int"/>
        </xs:sequence>
    </xs:complexType><xs:complexType
name="ComplexType1"><xs:sequence><xs:element name="Employee"
type="ComplexType2" maxOccurs="unbounded"/>
    </xs:sequence>
    </xs:complexType><xs:annotation><xs:documentation>Root
element</xs:documentation>
    </xs:annotation>
    <xs:element name="Employees" type="ComplexType1"/>
</xs:schema>
```

5.3 Create a New Project

Begin by creating and naming a Project in the Enterprise Designer.

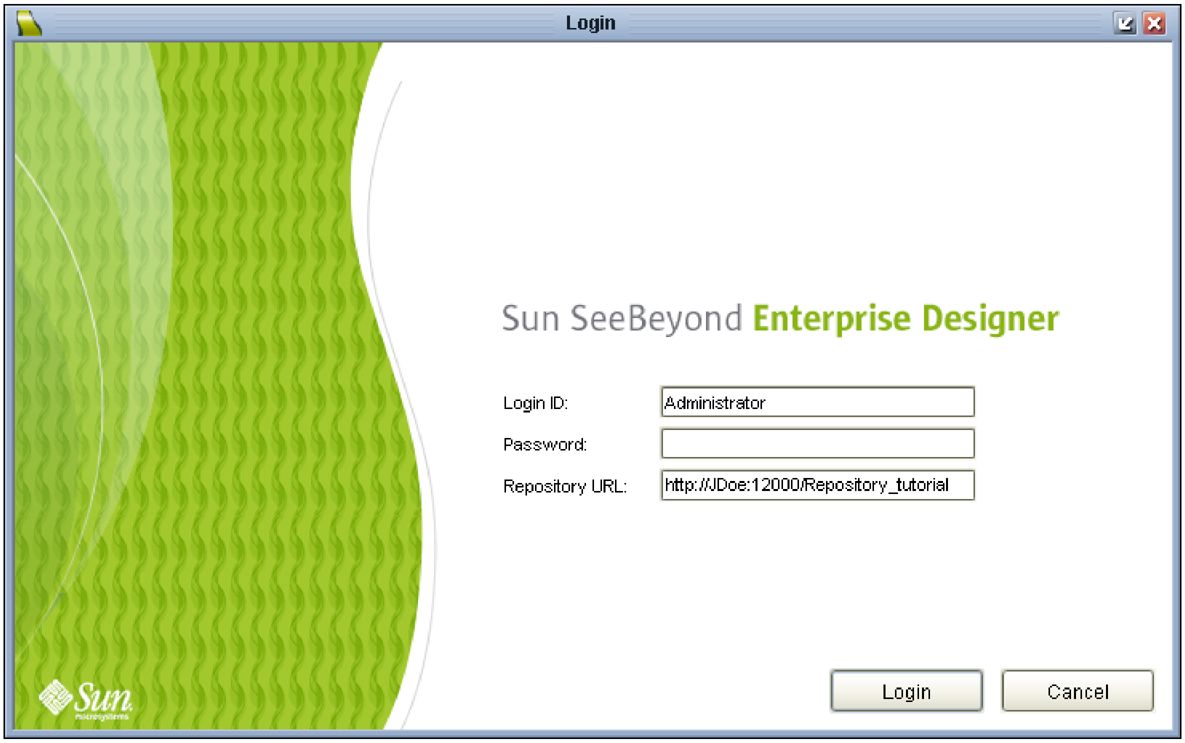
Start Enterprise Designer

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

1 Start the Enterprise Designer by executing **runed.bat** in the following location: **`C:\JavaCAPS51\edesigner\bin`**.

The Enterprise Designer Login dialog box appears.

Figure 126 Enterprise Designer Login



Login ID: Administrator

Password:

Repository URL: http://JDoe:12000/Repository_tutorial

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

The **Repository URL** value will default to the localhost information you supplied when you downloaded the Enterprise Designer.

`http://<localhost>:<port_number>/Repository_tutorial`.

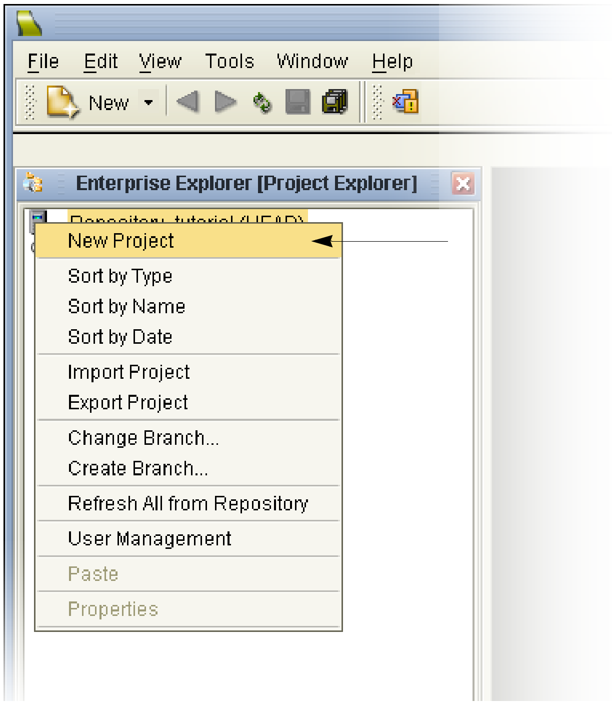
Note: *If your login window doesn't appear or if you get an invalid Login ID or Password error, make sure your Repository is running. Also, be aware that the Login ID and Password are case sensitive.*

Create 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** (or select "Project" from the drop-down list).

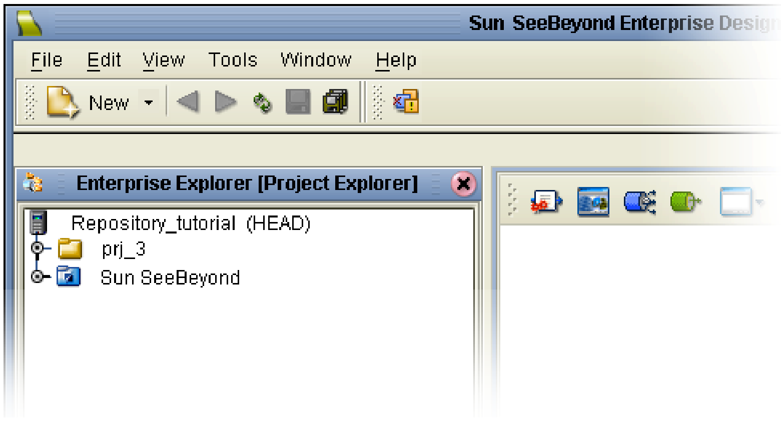
Figure 127 Create a Project



2 Name your Project **prj_3** and press **Enter**.

The Project, **prj_3**, appears in the Project Explorer tree on the left side of the window.

Figure 128 Project3


Sun SeeBeyond Enterprise Desig
File
Edit
View
Tools
Window
Help
New
Enterprise Explorer [Project Explorer]
Repository_tutorial (HEAD)
prj_3
Sun SeeBeyond

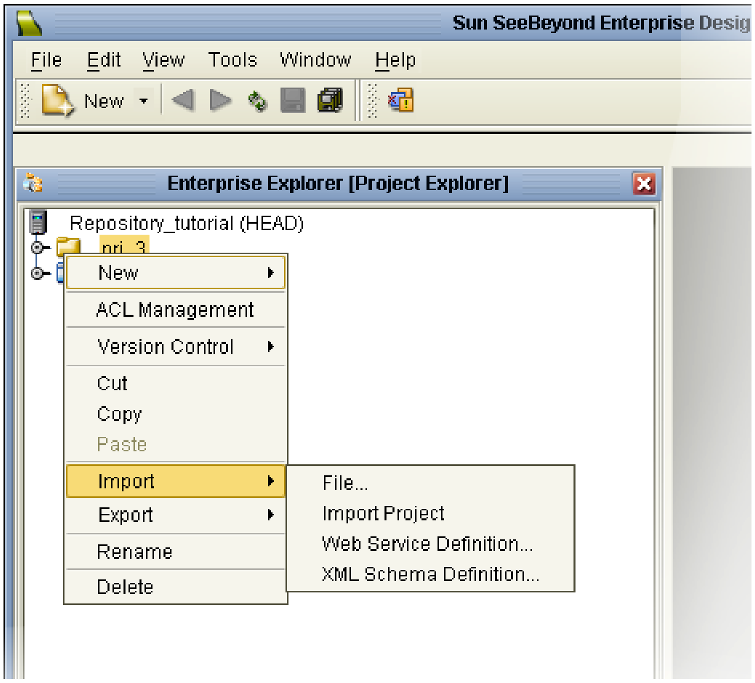
5.4 Import a Schema Definition

Import a schema definition to create an XSD Node to be used in your Java Collaboration later.

Import Schema Definition

1 Right-click your Project, **prj_3**, in the Project Explorer tree.

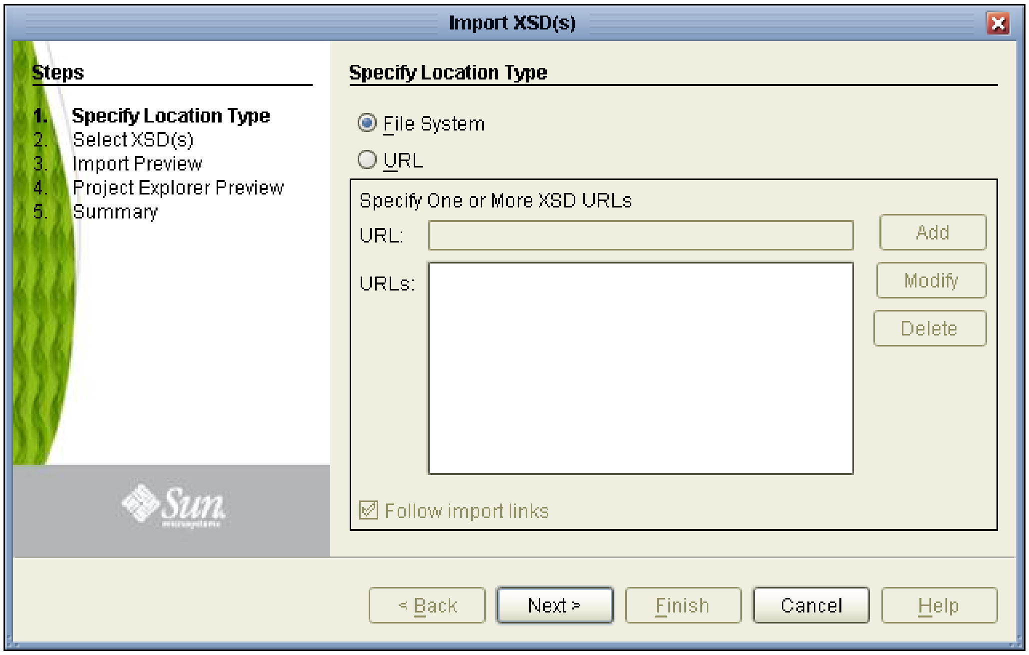
Figure 129 Select Schema Definition



2 Select **Import**, **XML Schema Definition**.

The Specify Location Type dialog appears.

Figure 130 Import From File Location



Import XSD(s)

Steps

1. **Specify Location Type**
2. Select XSD(s)
3. Import Preview
4. Project Explorer Preview
5. Summary

Specify Location Type

● File System
○ URL

Specify One or More XSD URLs

URL:

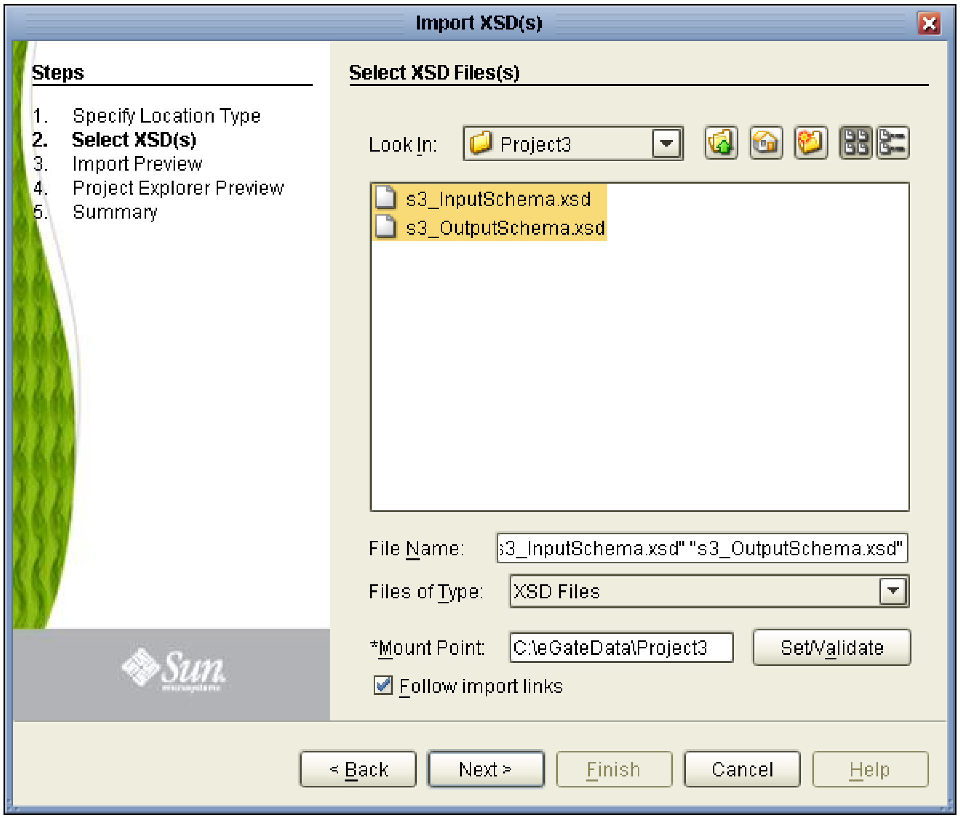
URLs:

☑ Follow import links

3 Accept the “File System” default. Click **Next**.

4 Navigate to your XML schemas in your sample Project3 folder.

Figure 131 XML Schemas (XSDs)



Import XSD(s)

Steps

1. Specify Location Type
2. **Select XSD(s)**
3. Import Preview
4. Project Explorer Preview
5. Summary

Select XSD Files(s)

Look In: Project3

s3_InputSchema.xsd
s3_OutputSchema.xsd

File Name: ;3_InputSchema.xsd" "s3_OutputSchema.xsd"

Files of Type: XSD Files

*Mount Point: C:\eGateData\Project3

☑ Follow import links

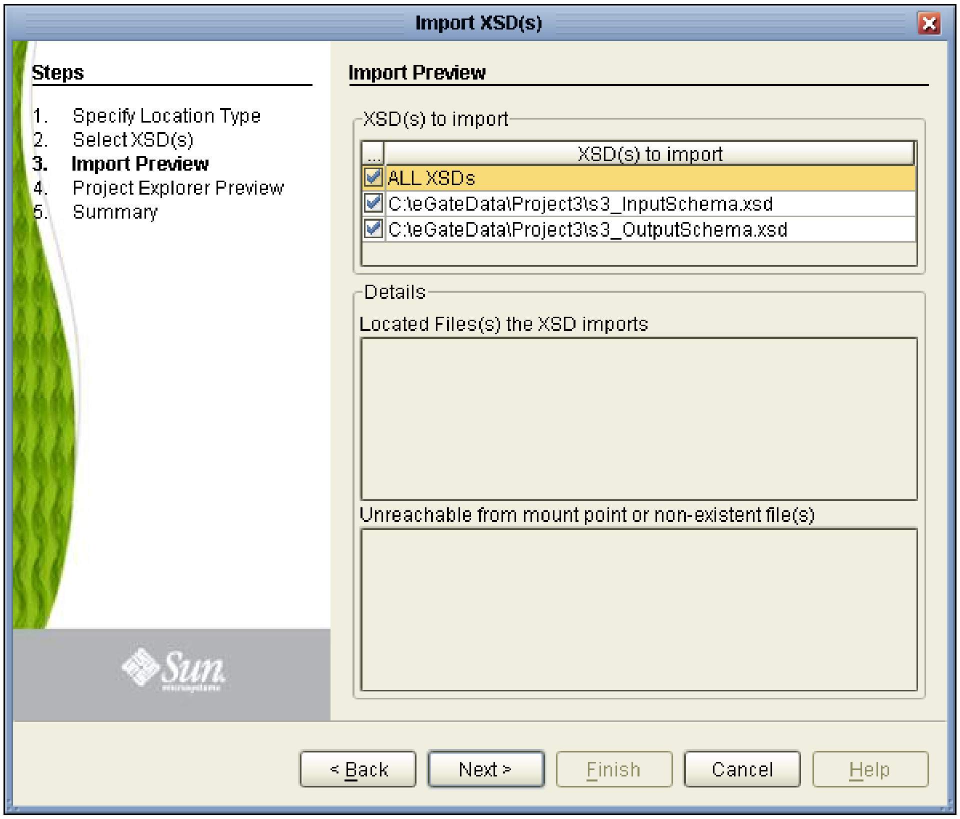
5 Select the input schema. Hold down the shift key and also select the output schema.

Both schemas appear in the **File Name** field.

6 Click **Next**.

The Import Preview screen appears.

Figure 132 Import Preview



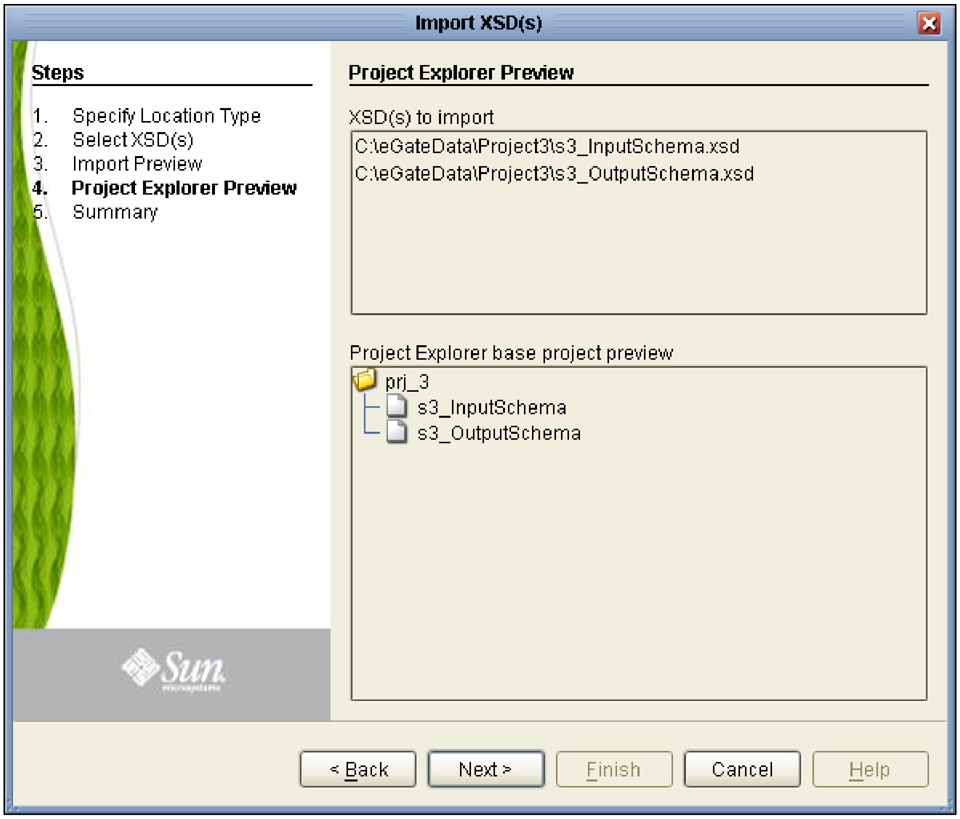
☑ ALL XSDs
☑ C:\eGateData\Project3\s3_InputSchema.xsd
☑ C:\eGateData\Project3\s3_OutputSchema.xsd

Details
Located Files(s) the XSD imports
Unreachable from mount point or non-existent file(s)
< Back
Next >
Finish
Cancel
Help

7 Click **Next**.

The Project Explorer Preview screen appears.

Figure 133 Project Explorer Preview



Import XSD(s)

Steps

1. Specify Location Type
2. Select XSD(s)
3. Import Preview
4. **Project Explorer Preview**
5. Summary

Project Explorer Preview

XSD(s) to import

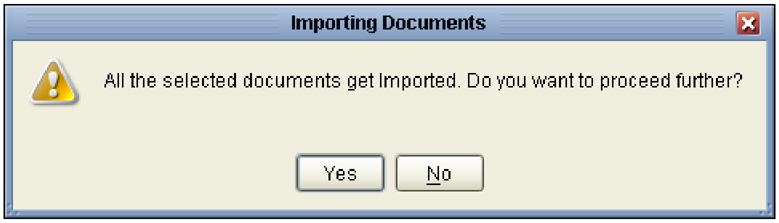
C:\eGateData\Project3\s3_InputSchema.xsd
C:\eGateData\Project3\s3_OutputSchema.xsd

Project Explorer base project preview

prj_3
s3_InputSchema
s3_OutputSchema

8 Click **Next**.

Figure 134 Import All

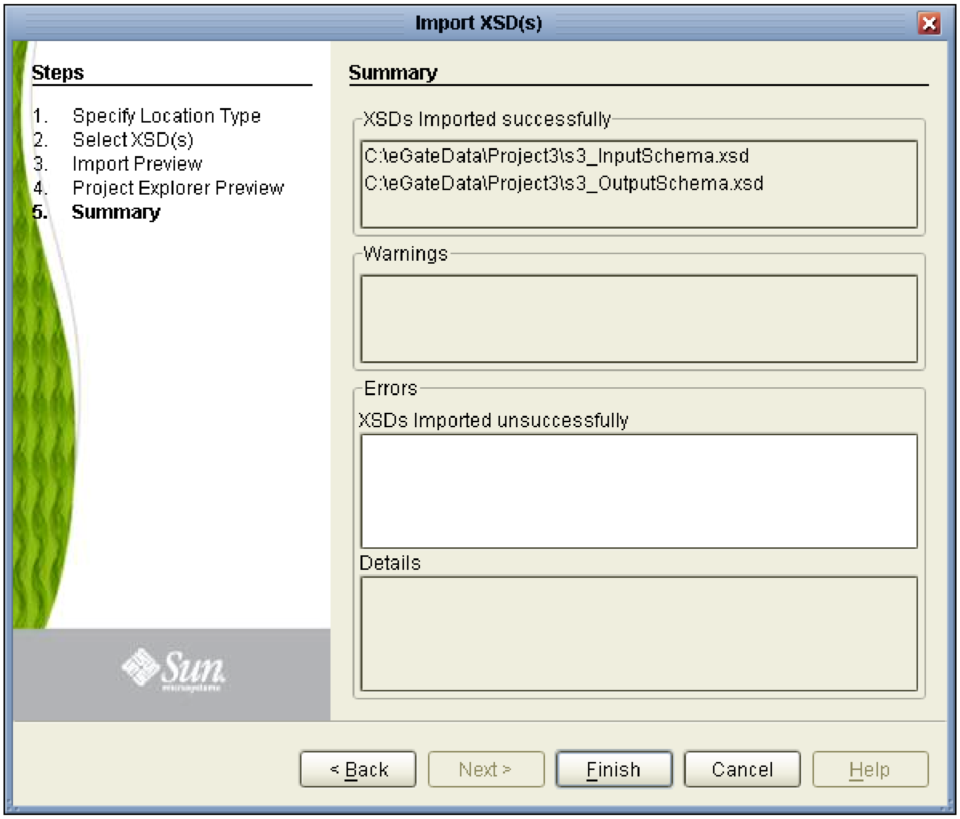


Importing Documents

All the selected documents get Imported. Do you want to proceed further?

9 Click **Yes** to get all (both) schemas.

Figure 135 Imported Successfully



XSDs Imported successfully
C:\eGateData\Project3\s3_InputSchema.xsd
C:\eGateData\Project3\s3_OutputSchema.xsd

XSDs Imported unsuccessfully

The Summary screen should show that your schemas were imported successfully.

10 Click **Finish**

Open Schema Definition and Verify

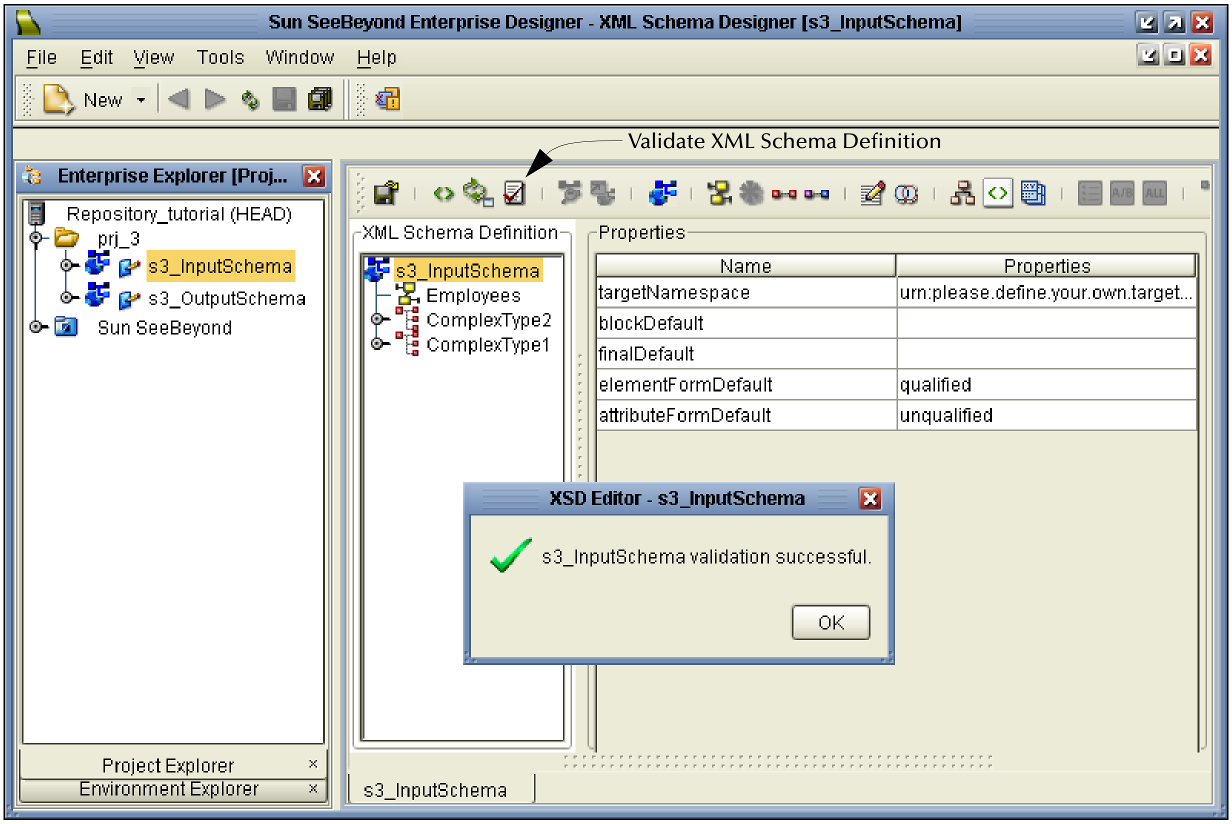
1 Right-click each schema definition (XSD) and Open.

2 Validate your input XML Schema Definition - click the Icon.

Note: *If your schema validates successfully, the validation dialog box pops up.*

See the following figure which shows validation for the input schema. Validate both schemas.

Figure 136 Verify XML Definition



Validate XML Schema Definition

Sun SeeBeyond Enterprise Designer - XML Schema Designer [s3_InputSchema]

XSD Editor - s3_InputSchema

s3_InputSchema validation successful.

Name	Properties
targetNamespace	urn:please.define.your.own.target...
blockDefault	
finalDefault	
elementFormDefault	qualified
attributeFormDefault	unqualified

After you have validated both XSD objects, you are ready to export the XSD Nodes.

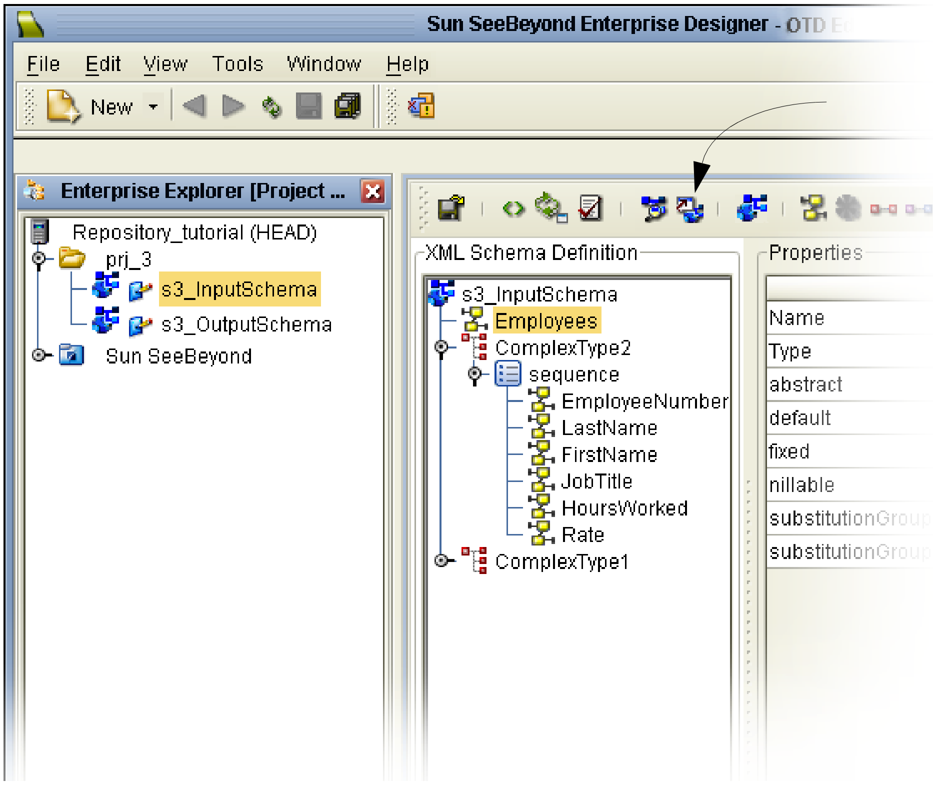
5.5 Export Schema Definitions to XSD Nodes

Use the Export XSD Node button on the toolbar to export your XML schema definition to an OTD.

Create an XSD Node from the Input Schema

1. Right-click the **s3_InputSchema** definition and click **Open**.
2. Make sure the **Employees** node is highlighted.

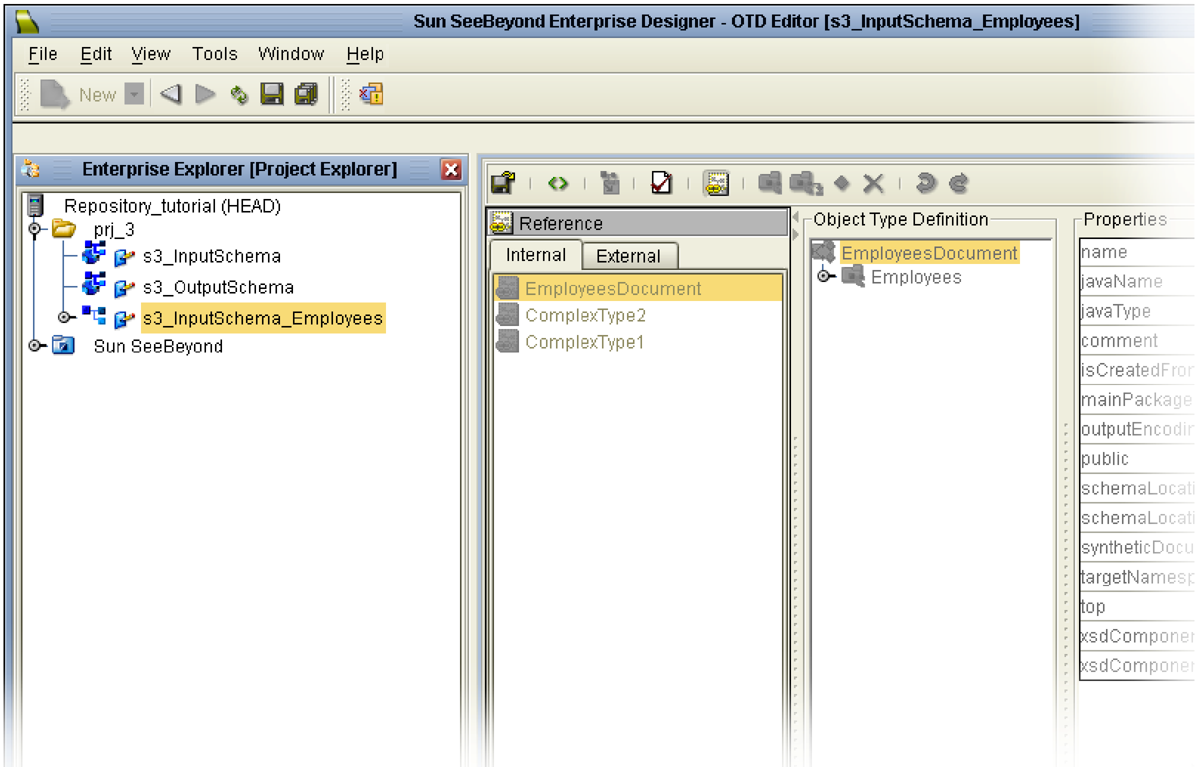
Figure 137 Export to OTD



3 Click the **Export XSD Node** button on the toolbar.

The XSD Node appears in your Project Explorer tree. See the following figure.

Figure 138 Exported OTD



The elements are grayed-out because you cannot modify an XSD Node that is exported into your project. You would typically modify the XML schema outside of Java CAPS.

Create an XSD Node from the Output Schema

1. Right-click the **s3_OutputSchema** definition and click **Open**.
2. Make sure **Employees** is highlighted.
3. Click the **Export XSD Node** button on the toolbar.

   Your XSD Nodes are now ready to be used in a Java Collaboration.

5.5.1 Create a Java Collaboration

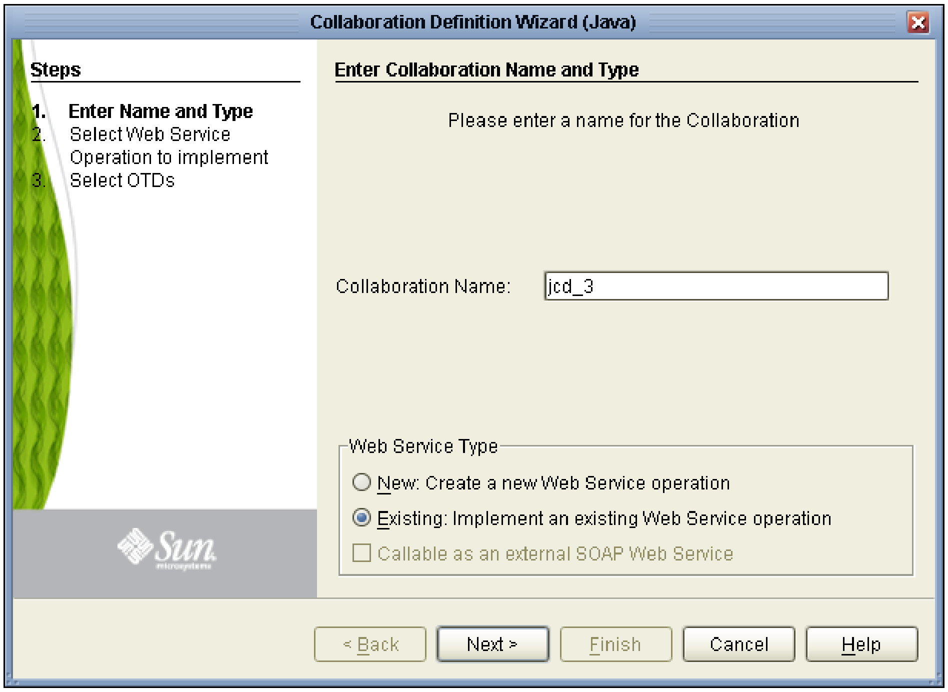
Use Service wizards in the following steps to configure your Collaboration. A Collaboration describes your data and destination connectivity. Configure a Collaboration (Java) definition. This service will process the input timecard.

Enter a Collaboration Name

1. Right-click **prj_3**.
2. Click **New, Collaboration Definition (Java)**.

   The Collaboration Definition Wizard (Java) appears.
3. Name your Collaboration **jcd_3**.

Figure 139 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: jcd_3

Web Service Type

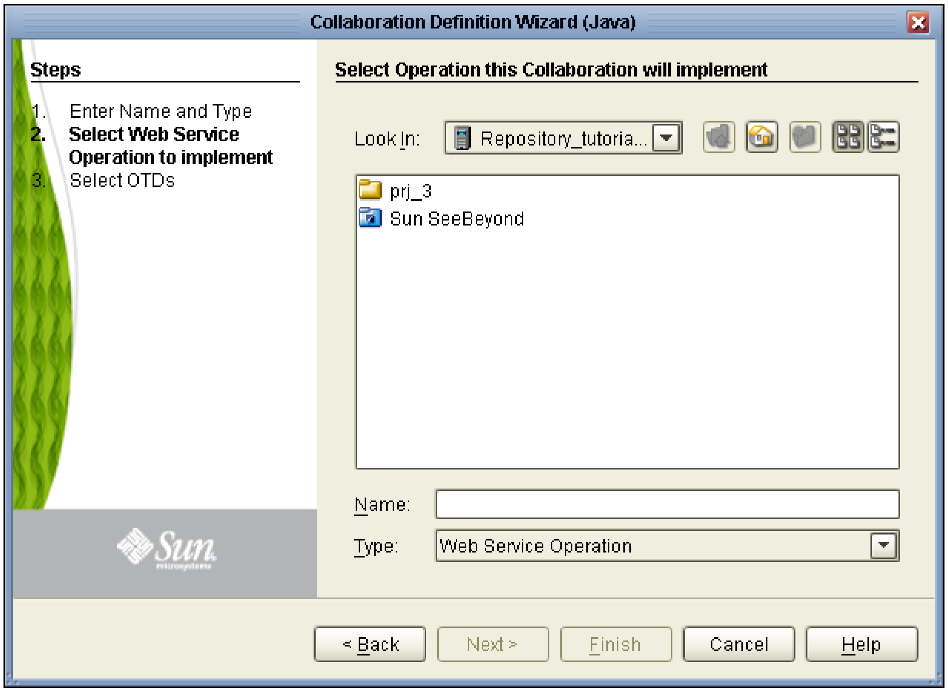
○ New: Create a new Web Service operation

● Existing: Implement an existing Web Service operation

□ Callable as an external SOAP Web Service

4 Click **Next**.

Figure 140 Collaboration Definition Wizard (Java)



Collaboration Definition Wizard (Java)

Steps

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

Select Operation this Collaboration will implement

Look In: Repository_tutoria...

prj_3

Sun SeeBeyond

Name:

Type: Web Service Operation

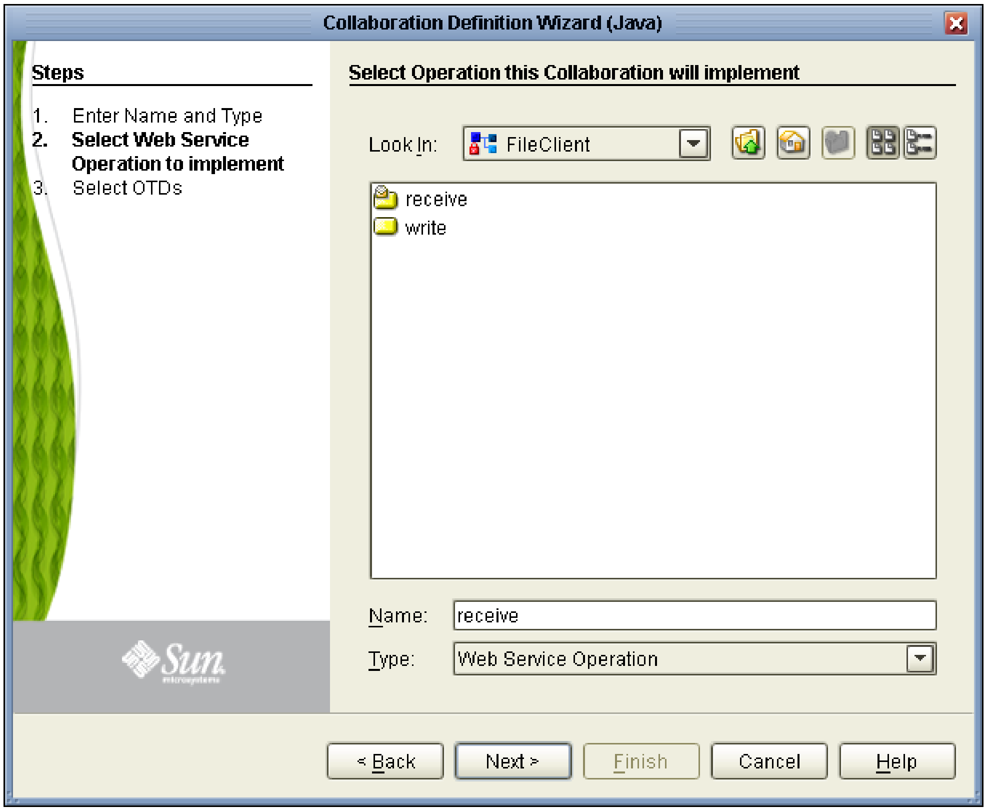
Create a Web Service Interface

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

1. Double-click Sun SeeBeyond.
2. Double-click **eWays.** (eWays to enable the sending and receiving of messages.)
3. Double-click **File**.
4. Double-click **FileClient**.
5. Click **receive**. Notice that **receive** appears in the Name field as shown in the following Figure 141.

   (**FileClient.receive** is the web service used to bring in file-based eWays.)

Figure 141 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

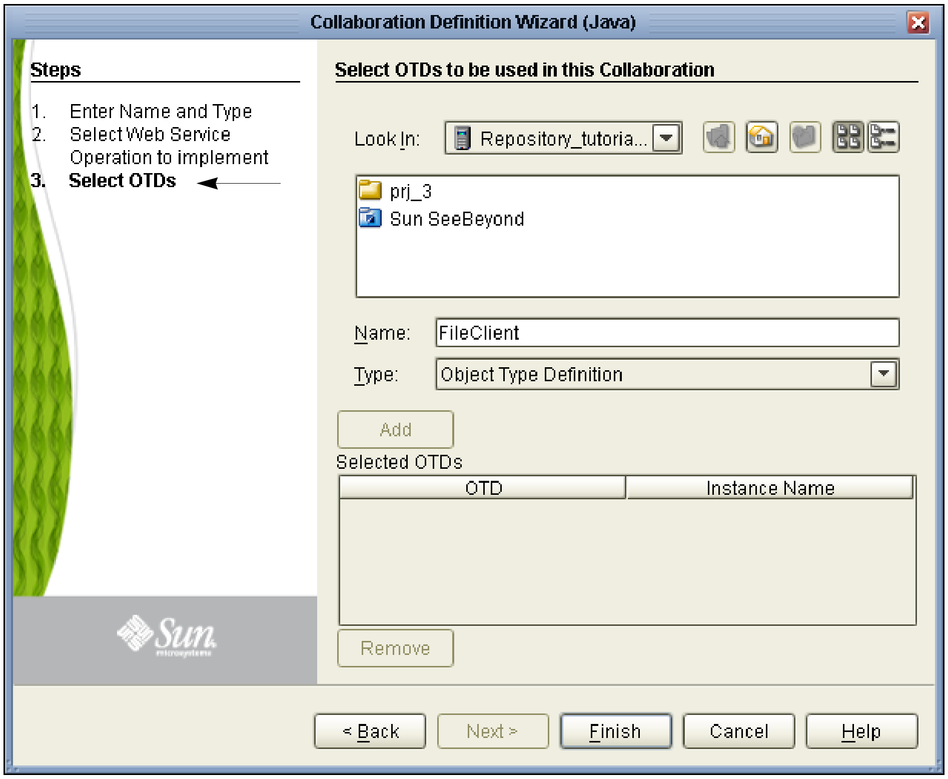
Type: Web Service Operation

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

Select OTDs

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

Figure 142 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: Repository_tutoria...

prj_3
Sun SeeBeyond

Name: FileClient

Type: Object Type Definition

Selected OTDs

OTD	Instance Name

1 Double-click **Sun SeeBeyond**.

2 Double-click **eWays**.

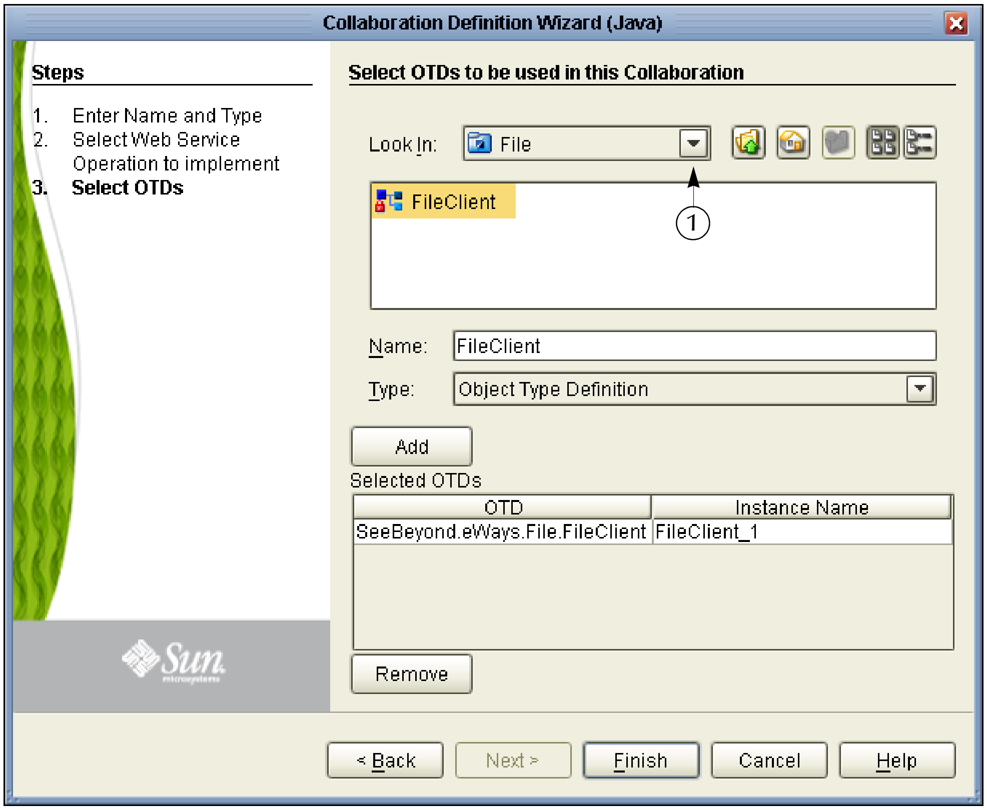
3 Double-click **File**.

4 Double-click **FileClient**.

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

This adds the **FileClient_1** OTD to the list. See the following Figure 143.

Figure 143 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
FileClient
1
Add
Remove
< Back
Next >
Finish
Cancel
Help
Sun microsystems

Name: FileClient

Type: Object Type Definition

Selected OTDs

OTD	Instance Name
SeeBeyond.eWays.File.FileClient	FileClient_1

Note: *Before clicking Finish, select the XSD Nodes.*

6 Click the drop-down arrow (number 1 in Figure 143).

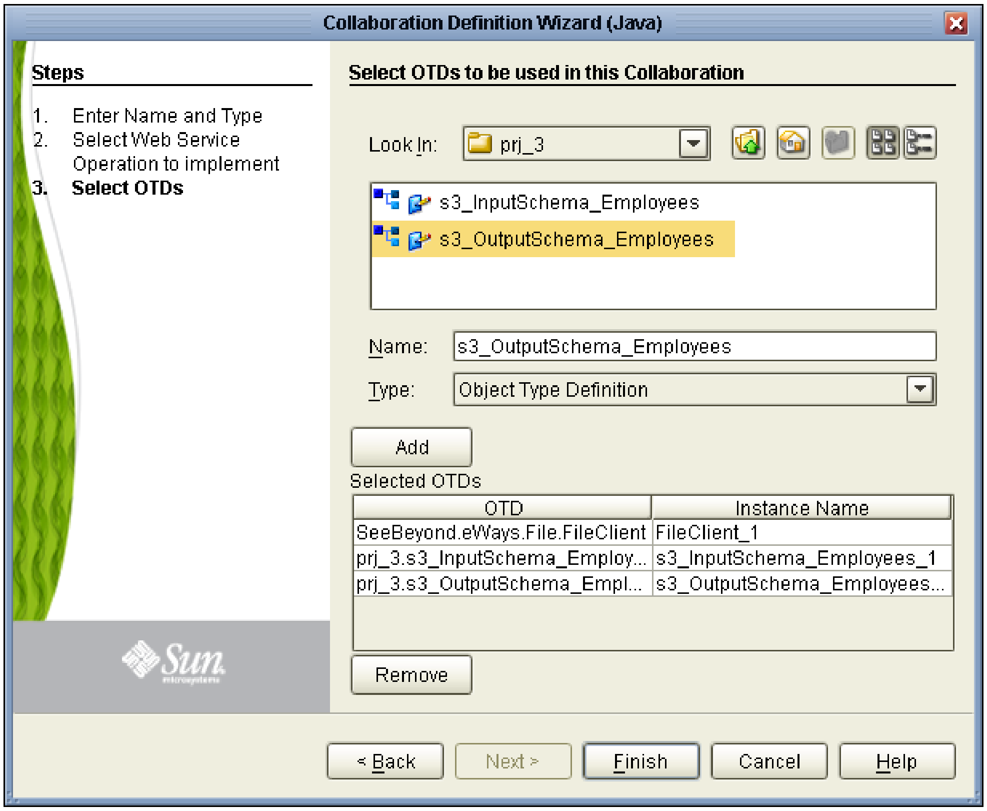
7 Click **prj_3** to display the XSD Nodes.

8 Double-click **s3_InputSchema_Employees**.

9 Double-click **s3_OutputSchema_Employees**.

See the following figure.

Figure 144 Selected 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: prj_3

s3_InputSchema_Employees
s3_OutputSchema_Employees

Name: s3_OutputSchema_Employees

Type: Object Type Definition

Selected OTDs

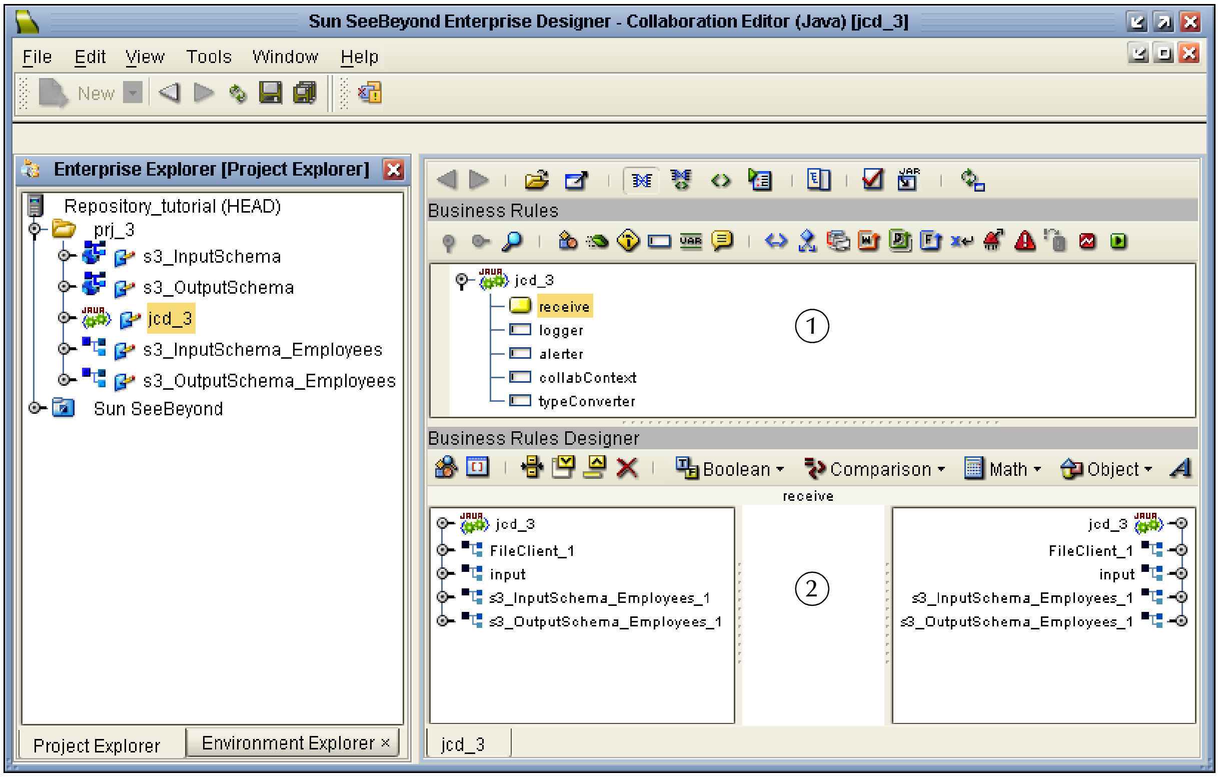
OTD	Instance Name
SeeBeyond.eWays.File.FileClient	FileClient_1
prj_3.s3_InputSchema_Employ...	s3_InputSchema_Employees_1
prj_3.s3_OutputSchema_Empl...	s3_OutputSchema_Employees...

All three objects appear in the list of OTDs.

10 Click **Finish**.

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

Figure 145 Collaboration Editor (Java)



5.6 Apply Business Rules

Business Rules display in the Business Rules pane (number 1 in Figure 145). Use the Business Rules Designer GUI (number 2 in Figure 145) to map your rules.

5.6.1 Concatenation Logic

In this section concatenate **FirstName** and **LastName** in the input file and create **FullName** in the output file.

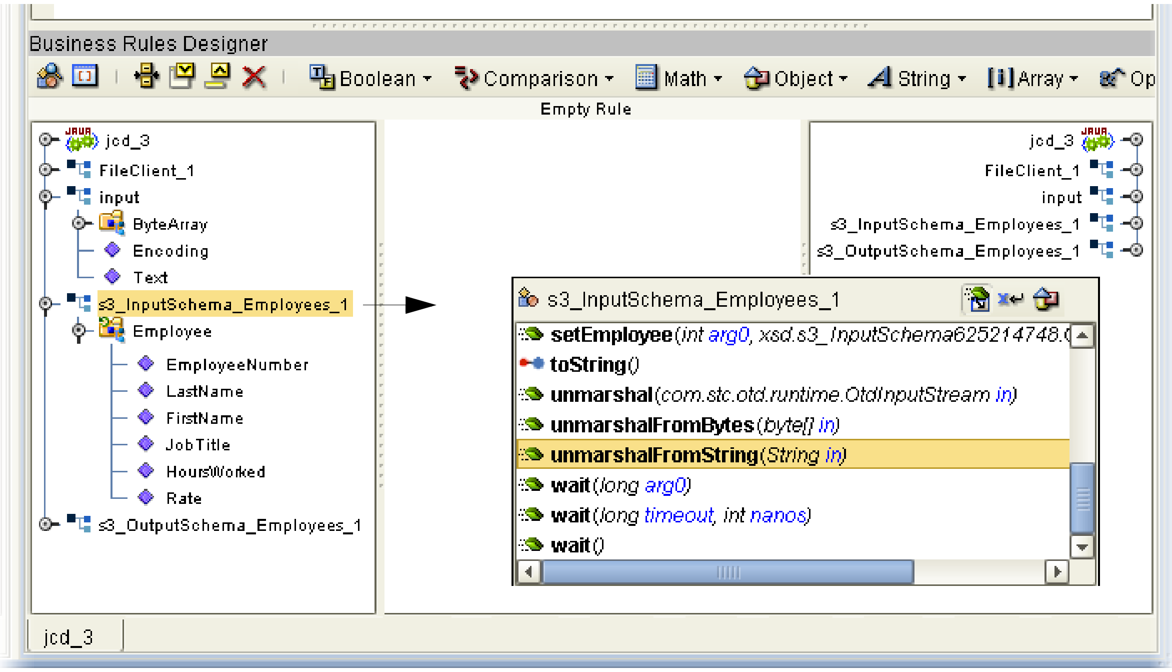
Unmarshal Text

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

1. Click the expansion node of **input** in the left pane of the Business Rules Designer, or double-click the **input** icon.
2. Click the expansion node of **s3_InputSchema_Employees_1** in the left pane of the Business Rules Designer to expand the view, and also open the view of **Employee**.
3. Click and drag **s3_InputSchema_Employees_1** from the left pane to the center pane of the Business Rules Designer and select **unmarshalFromString(String in)** from the methods list by double-clicking.

See the following figure.

Figure 146 Unmarshal Text



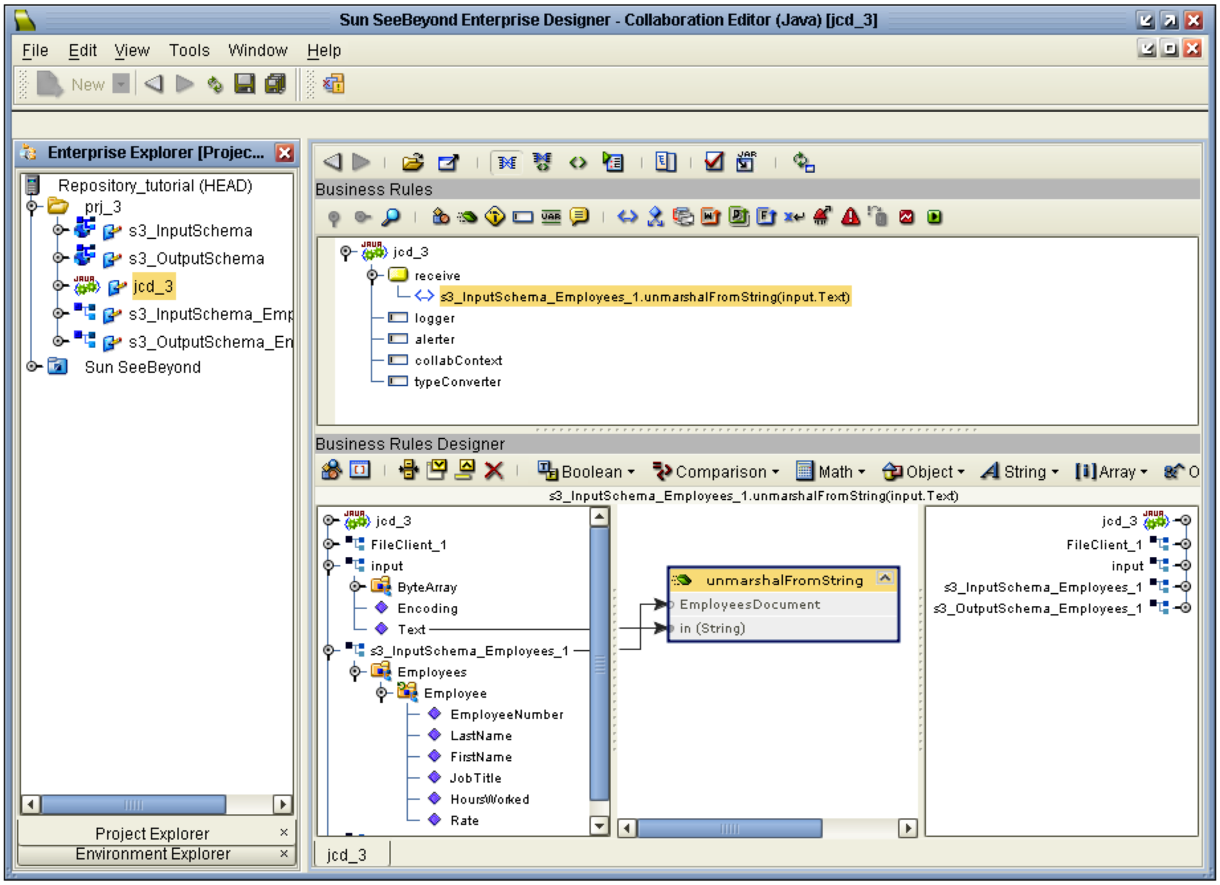
Business Rules Designer
Boolean
Comparison
Math
Object
String
Array
Op
Empty Rule
jcd_3
FileClient_1
input
ByteArray
Encoding
Text
s3_InputSchema_Employees_1
Employee
EmployeeNumber
LastName
FirstName
JobTitle
HoursWorked
Rate
s3_OutputSchema_Employees_1
jcd_3
FileClient_1
input
s3_InputSchema_Employees_1
s3_OutputSchema_Employees_1
s3_InputSchema_Employees_1
setEmployee(int arg0, xsd.s3_InputSchema625214748.(
toString()
unmarshal(com.stc.otd.runtime.OtdInputStream in)
unmarshalFromBytes(byte[] in)
unmarshalFromString(String in)
wait(long arg0)
wait(long timeout, int nanos)
wait()
jcd_3

(You can also right-click **s3_InputSchema_Employees_1** and select a method.)

The unmarshal method reads your employee input XML File.

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

Figure 147 Business Designer - Unmarshal Text



Connect Input and Output OTD Nodes

In the following steps concatenate names and perform a multiplication operation.

1 Click the expansion nodes to open the views of **s3_OutputSchema_Employees_1** and **Employee**, in the right pane.

Note: *Notice that the objects in both the left and right pane of the Business Rules Designer window have the same name. Keep in mind that the objects (**FileClient_1**, **s3_InputSchema_Employees_1**, and **s3_OutputSchema_Employees_1,** for example,) refer to the same object whether they are in the left or right pane.*

2 Connect **EmployeeNumber**, from **s3_InputSchema_Employees_1** in the left pane to **EmpNumber** of **s3_OutputSchema_Employees_1** in the right pane.

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

Note: *As you map in the center pane, the Business Rules continue to appear in the top pane. The active Business Rule also appears at the top of the Business Rules Designer pane.*

3 Drag the **Concat** icon down to your designer workspace. (**Concat** is on the **String** drop-down menu.)

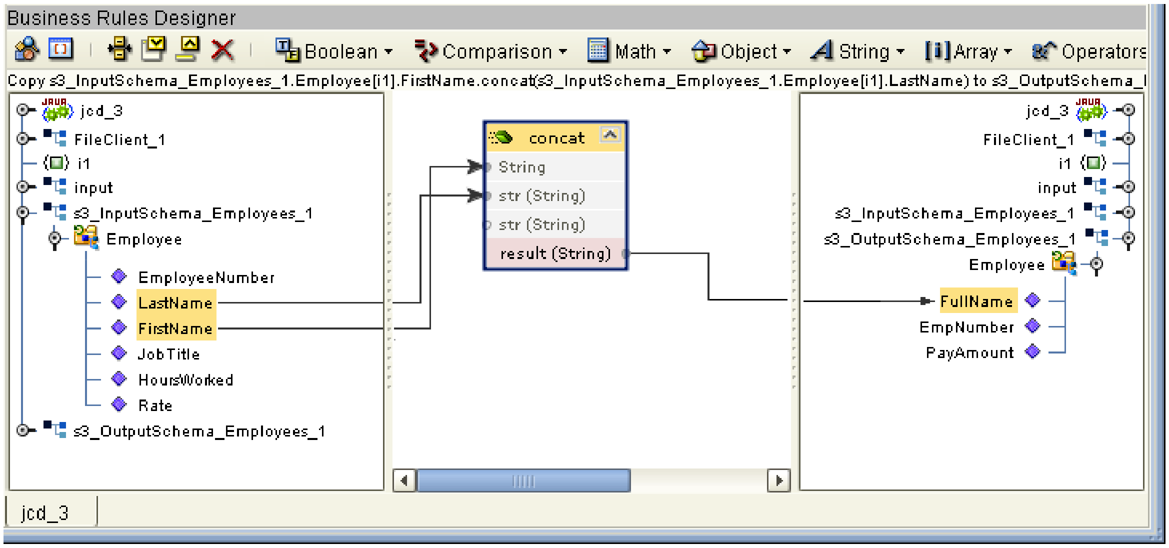
4 Connect **FirstName** to **String** in the Concat box. (Connect from **s3_InputSchema_Employee_1** in the left pane.)

5 Connect **LastName** to **str (String)** in the Concat box. (Connect from **s3_InputSchema_1** in the left pane.)

6 Connect **result (String)** to **FullName** in **Employee** of the root element **s3_OutputSchema_Employees_1** in the right pane.

See the following figure.

Figure 148 Concatenation Logic



Copy s3_InputSchema_Employees_1.Employee[i1].FirstName.concat(s3_InputSchema_Employees_1.Employee[i1].LastName) to s3_OutputSchema_I

5.6.2 Multiplication Logic

In this section set up the multiplication logic (Business Rules) to obtain an amount (**PayAmount**) for the output file.

1 Click the Math drop down list and select the Multiply operation.

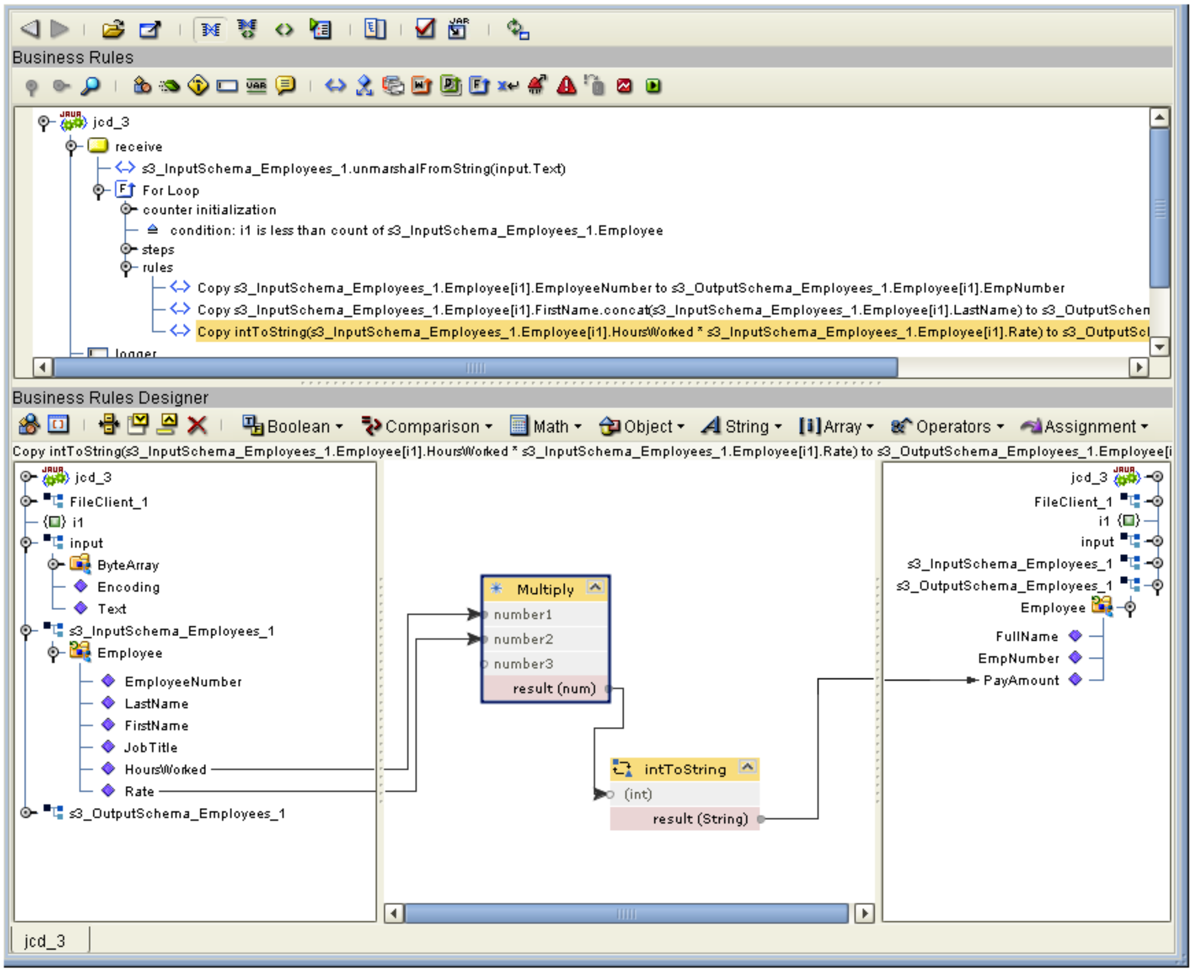
The Multiply box appears in the Business Rules Designer pane.

2 Map **HoursWorked** to **number1** and **Rate** to **number2** in the dialog box.

3 Map **result(num)** to **PayAmount** in the output schema on the right.

4 Click **OK** in the **Number to String Conversion** pop-up dialog box, to accept the formatting defaults.

Figure 149 Mapped Multiplication Operation



```
jcd_3
  receive
    s3_InputSchema_Employees_1.unmarshalFromString(input.Text)
    For Loop
      counter initialization
        condition: i1 is less than count of s3_InputSchema_Employees_1.Employee
      steps
      rules
        Copy s3_InputSchema_Employees_1.Employee[i1].EmployeeNumber to s3_OutputSchema_Employees_1.Employee[i1].EmpNumber
        Copy s3_InputSchema_Employees_1.Employee[i1].FirstName.concat(s3_InputSchema_Employees_1.Employee[i1].LastName) to s3_OutputSchen
        Copy intToString(s3_InputSchema_Employees_1.Employee[i1].HoursWorked * s3_InputSchema_Employees_1.Employee[i1].Rate) to s3_OutputScl
  logger
```

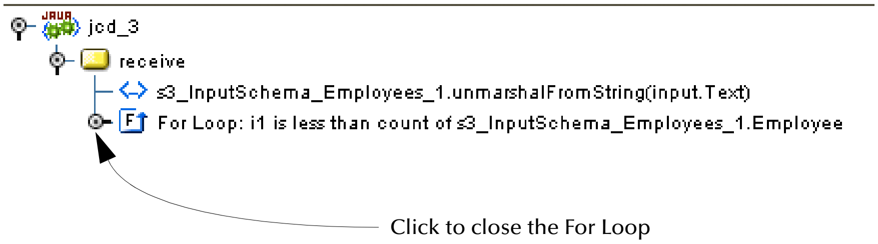
Copy intToString(s3_InputSchema_Employees_1.Employee[i1].HoursWorked * s3_InputSchema_Employees_1.Employee[i1].Rate) to s3_OutputSchema_Employees_1.Employee[i

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

Closing the “For Loop” sets the context to the same level as the for loop, instead of inside the loop. All new rules are generated below the context. This will soon include **marshalToString** and **write**.

See the following figure.

Figure 150 Collapse For Loop



```
jcd_3
  receive
    s3_InputSchema_Employees_1.unmarshalFromString(input.Text)
    For Loop: i1 is less than count of s3_InputSchema_Employees_1.Employee
```

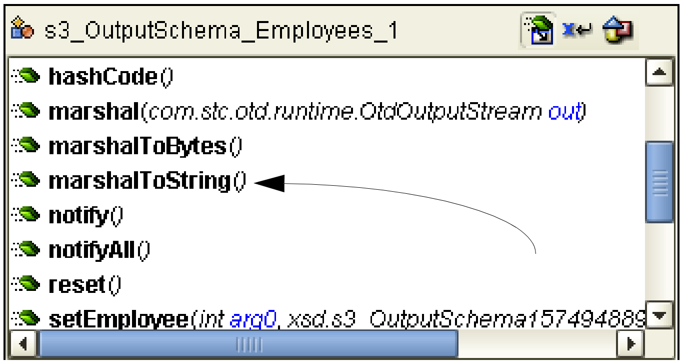
Click to close the For Loop

Marshal Text

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

1. Expand the view of **FileClient_1** in the right pane to expose the **Text** node.
2. Click and then drag **s3_OutputSchema_Employees_1** from the left to the center pane of the Business Rules Designer.

Figure 151 Marshal to String


s3_OutputSchema_Employees_1
hashCode()
marshal(com.stc.otd.runtime.OtdOutputStream out)
marshalToBytes()
marshalToString()
notify()
notifyAll()
reset()
setEmployee(int arg0, xsd.s3_OutputSchema157494889

A list of methods appears.

Note: *You can also right-click* ***s3_OutputSchema_Employees_1*** *and click* ***Select method to call****.*

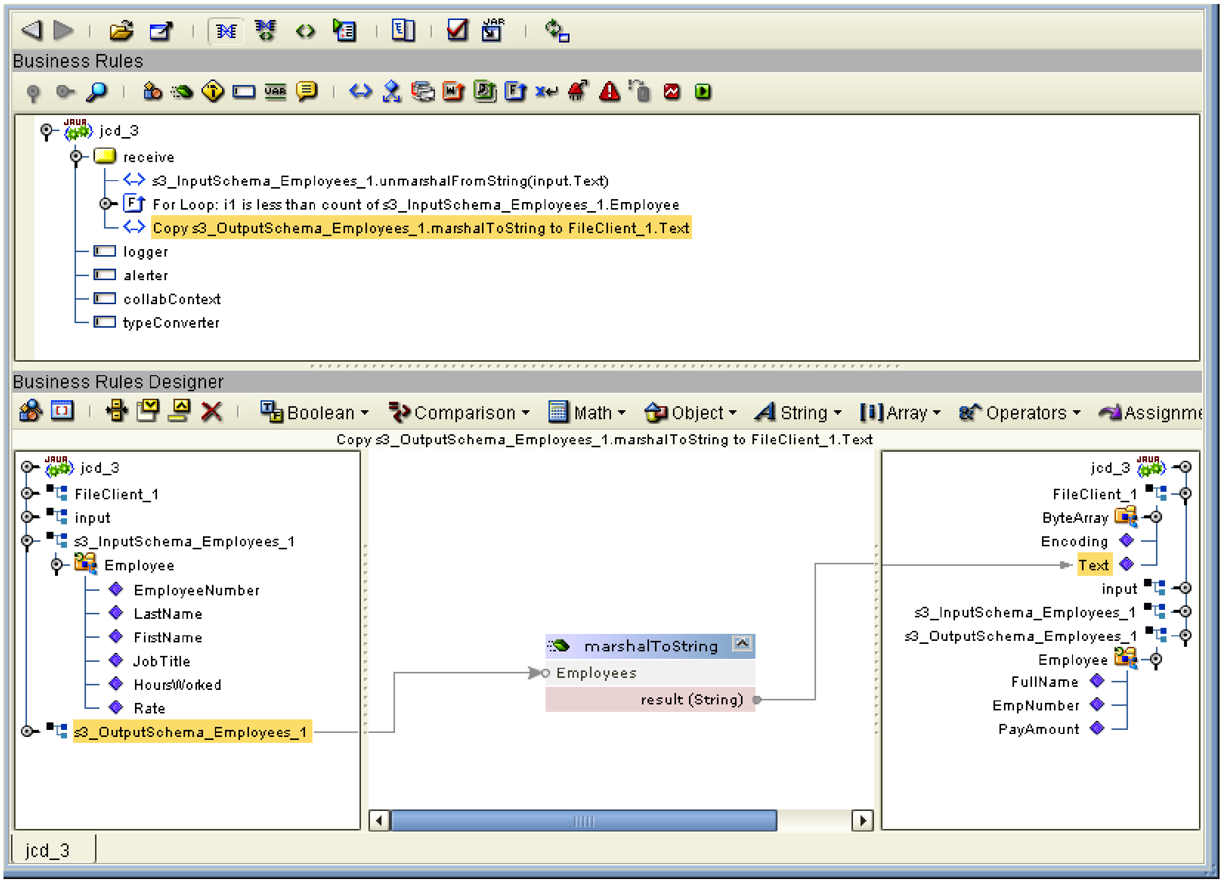
3. Double-click the method **marshalToString()** from the pop-up list.

   The **marshalToString** method box appears.

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

   See the following figure.

Figure 152 Marshal Text



```
jcd_3
  receive
    s3_InputSchema_Employees_1.unmarshalFromString(input.Text)
    For Loop: i1 is less than count of s3_InputSchema_Employees_1.Employee
    Copy s3_OutputSchema_Employees_1.marshalToString to FileClient_1.Text
  logger
  alerter
  collabContext
  typeConverter
```

Copy s3_OutputSchema_Employees_1.marshalToString to FileClient_1.Text

```
jcd_3
FileClient_1
input
s3_InputSchema_Employees_1
  Employee
    EmployeeNumber
    LastName
    FirstName
    JobTitle
    HoursWorked
    Rate
s3_OutputSchema_Employees_1
```

marshalToString
Employees
result (String)

```
                             jcd_3
                      FileClient_1
                         ByteArray
                          Encoding
                              Text
                             input
        s3_InputSchema_Employees_1
       s3_OutputSchema_Employees_1
                          Employee
                          FullName
                         EmpNumber
                         PayAmount
```

Write FileClient_1

In these steps prepare to write to the output file.

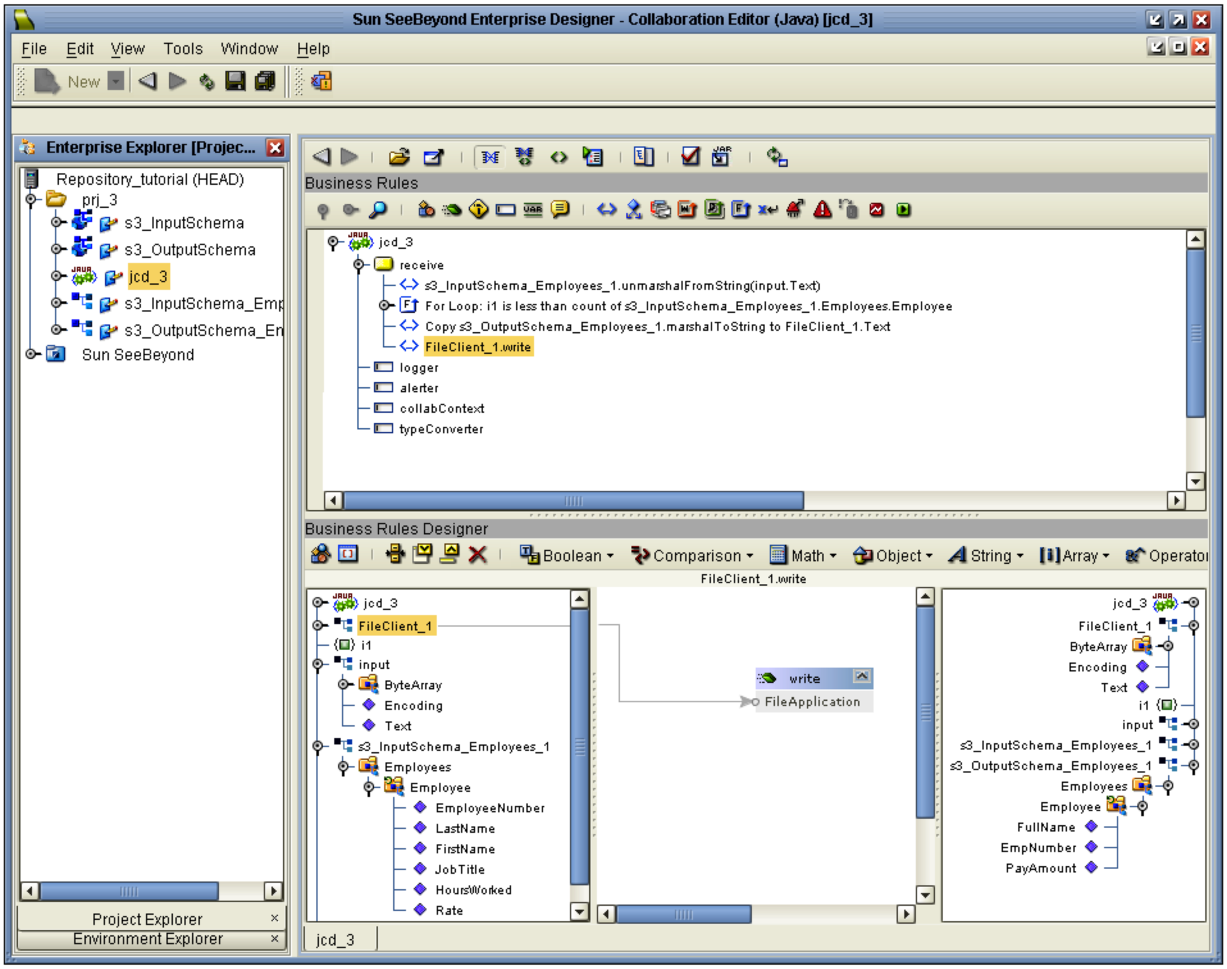
1. Click and drag **FileClient_1** from the left pane to the center pane of the Business Rules designer.

   A list of methods appears.

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

   See the following figure.

Figure 153 Write Output File



jcd_3
receive
s3_InputSchema_Employees_1.unmarshalFromString(input.Text)
For Loop: i1 is less than count of s3_InputSchema_Employees_1.Employees.Employee
Copy s3_OutputSchema_Employees_1.marshalToString to FileClient_1.Text
FileClient_1.write

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

5.7 Create a Connectivity Map

Your Project, **prj_3**, needs a Connectivity Map to define the data flow between the timecard system and the work file output.

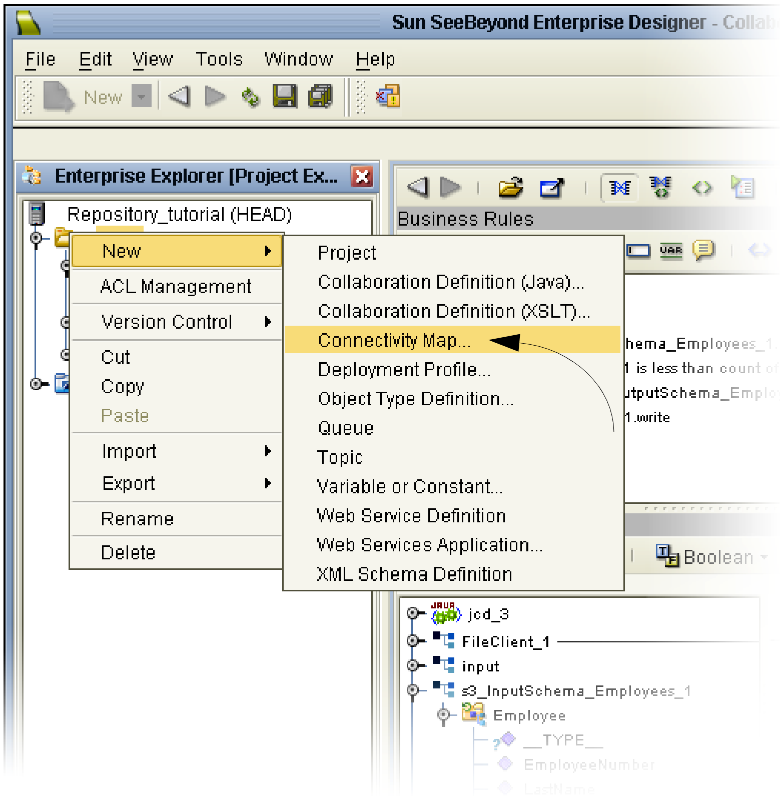
Name Connectivity Map

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

1 In the Project Explorer, right-click **prj_3**.

2 Click **New**, **Connectivity Map**.

Figure 154 Create a Connectivity Map

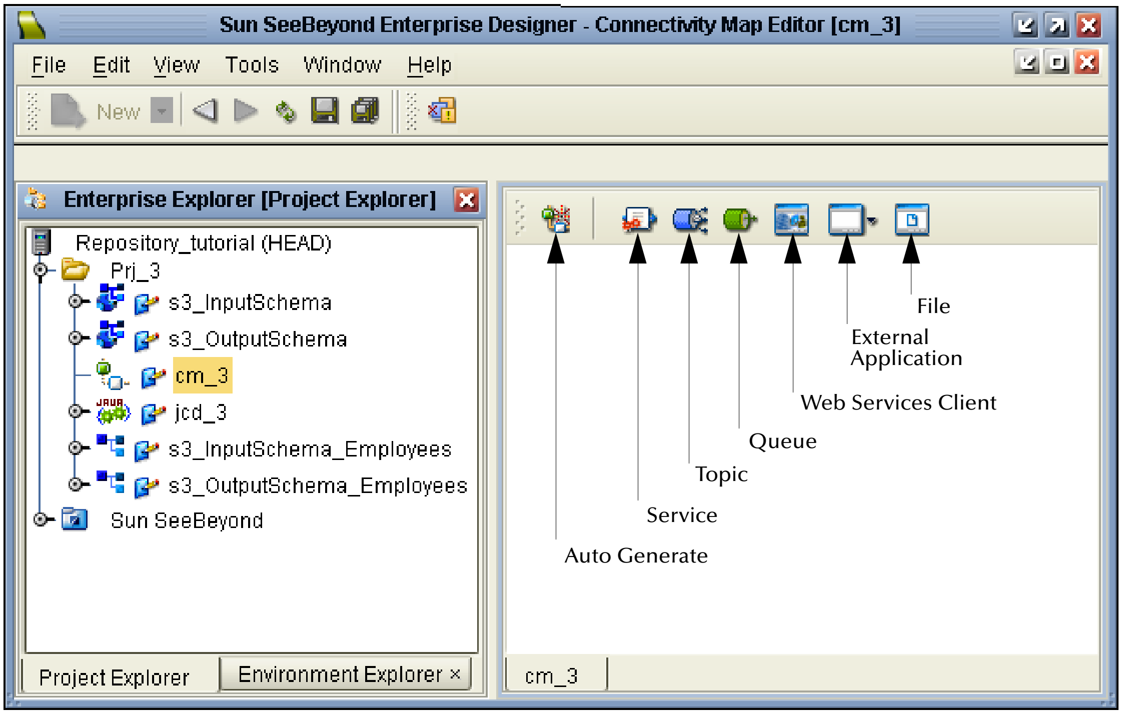


3 Right-click and rename your Connectivity Map **cm_3**, and press **Enter**.

4 Click **Yes** to confirm.

cm_3 is added to your Project Explorer tree, as shown in the following figure.

Figure 155 Connectivity Map Active



Note: *The* **cm_3** *Connectivity Map appears as an icon in the Project Explorer tree. A* **cm_3** *tab is also added at the bottom of the Connectivity Map.*

5.7.1 Populate the Connectivity Map

In this section, add the components to the Connectivity Map. Later you will link the components.

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

- **input_file** (File External Application)
- **svc_process_timecard** (Service)
- **output_file** (File External Application)

Add Input and Output Objects

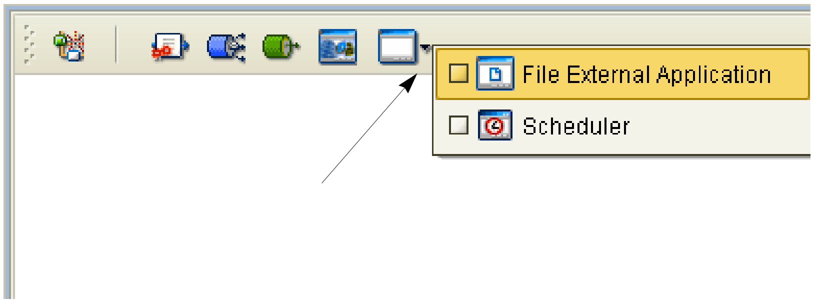
Place an input file (File External Application) in your Connectivity Map.

1. On the Connectivity Map toolbar, click the External Application arrow (shown in Figure 156) to display the list.
2. Click **File External Application**.

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

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

Figure 156 External Application Selection



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

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

4 Name the File **input_file** and press **Enter**.

5 On the Connectivity Map toolbar, drag the Service object into the Connectivity Map to the right of the **input_file**.

This Service will eventually contain the binding information about connecting the input to the output.

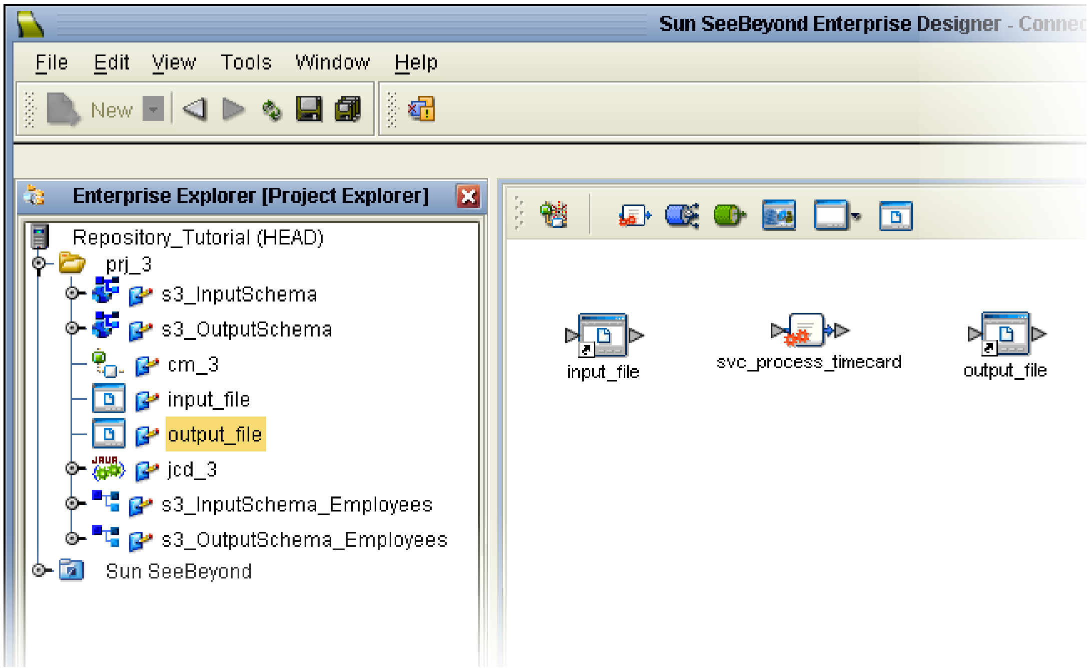
6 Rename this Service **svc_process_timecard** and press **Enter**.

7 Drag another **File** object into the Connectivity Map, and place it to the right of **svc_process_timecard**.

8 Name the File **output_file** and press **Enter**.

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

Figure 157 Connectivity Map



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.

5.7.2 Link Components in the Connectivity Map

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

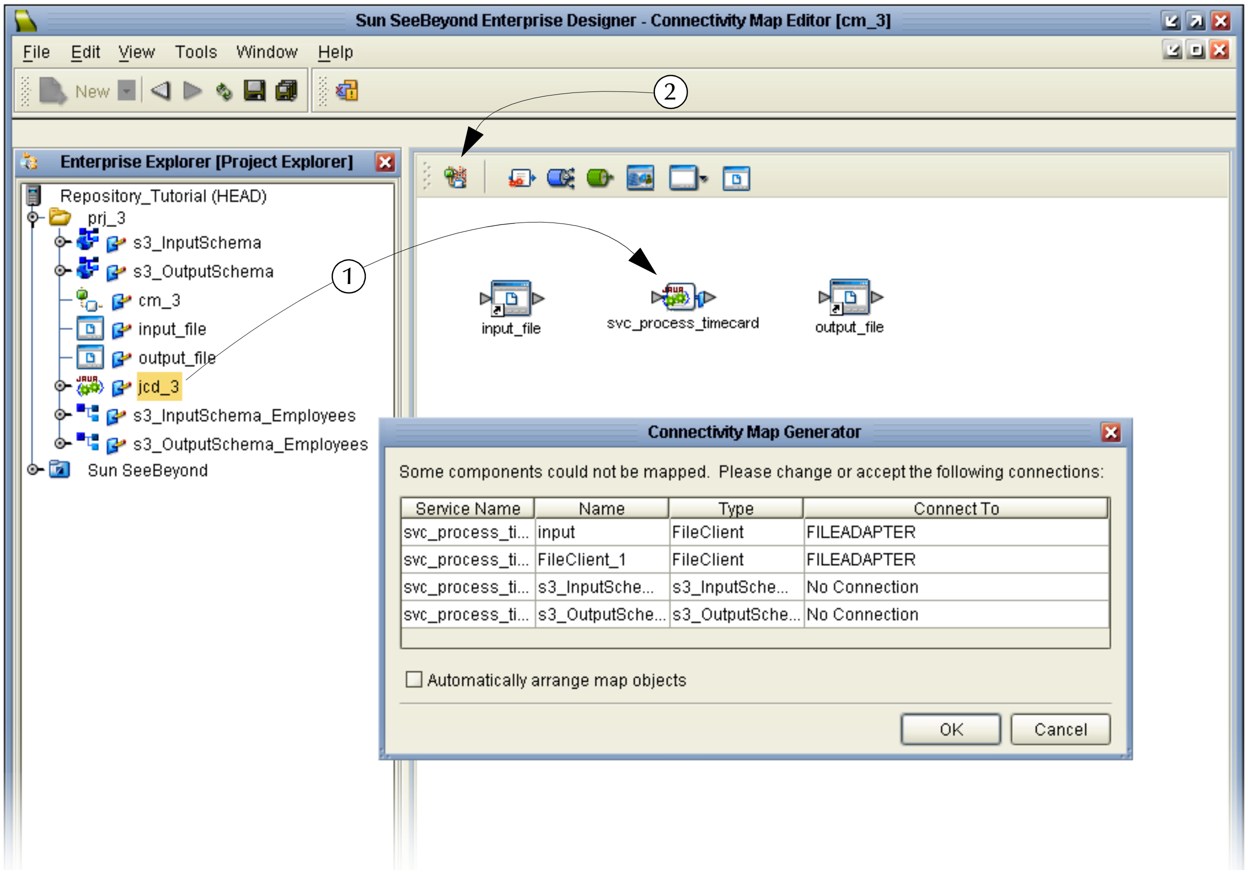
Connect Nodes

Use the Connectivity Map Generator to link the components.

1. Click and drag the **jcd_3** definition from the Project Explorer tree to the Service object **svc_process_timecard**.

   See the following figure.

Figure 158 Automap Components


Sun SeeBeyond Enterprise Designer - Connectivity Map Editor [cm_3]
File
Edit
View
Tools
Window
Help
New
2
Enterprise Explorer [Project Explorer]
Repository_Tutorial (HEAD)
prj_3
s3_InputSchema
s3_OutputSchema
cm_3
input_file
output_file
jcd_3
s3_InputSchema_Employees
s3_OutputSchema_Employees
Sun SeeBeyond
1
input_file
svc_process_timecard
output_file
Connectivity Map Generator

Some components could not be mapped. Please change or accept the following connections:

Service Name	Name	Type	Connect To
svc_process_ti...	input	FileClient	FILEADAPTER
svc_process_ti...	FileClient_1	FileClient	FILEADAPTER
svc_process_ti...	s3_InputSche...	s3_InputSche...	No Connection
svc_process_ti...	s3_OutputSche...	s3_OutputSche...	No Connection

☐ Automatically arrange map objects

2. Click the **Connectivity Map Generator** button.

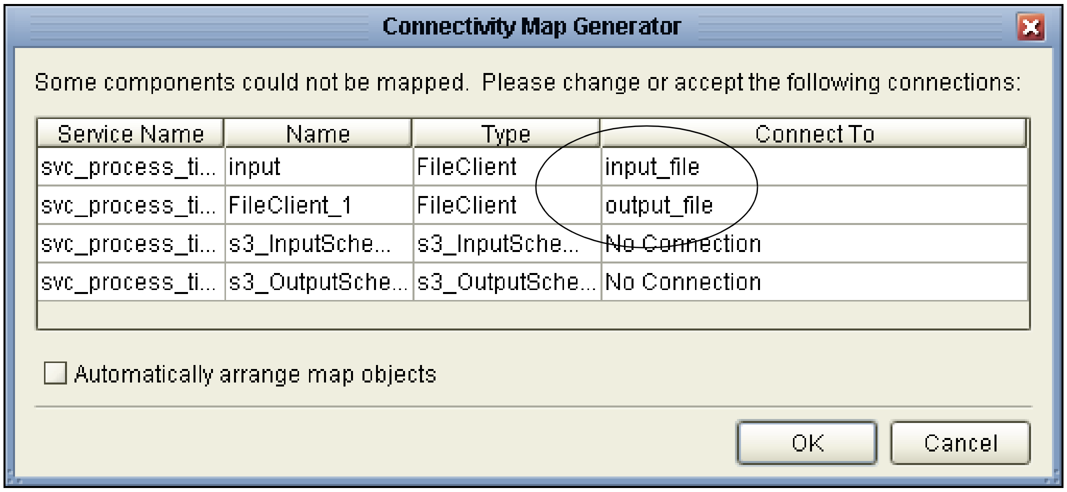
   In the Connectivity Map Generator dialog, the first two rows represent files.

3. Click in the Connect To, FILEADAPTER, field and set the first row to “input.”

4 Set the second row to “output.”

In the Connectivity Map Generator dialog, the third and fourth rows represent schemas and are not mappable objects.

Figure 159 Connectivity Map Generator



Connectivity Map Generator

Some components could not be mapped. Please change or accept the following connections:

Service Name	Name	Type	Connect To
svc_process_ti...	input	FileClient	input_file
svc_process_ti...	FileClient_1	FileClient	output_file
svc_process_ti...	s3_InputSche...	s3_InputSche...	No Connection
svc_process_ti...	s3_OutputSche...	s3_OutputSche...	No Connection

□ Automatically arrange map objects

5 Click **OK**.

6 Click **Save All**.

5.8 Configure the eWays

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

- The input eWay polls a directory on the local File system. For example: **C:\eGateData\Project3**.
- The Outbound File eWay writes the results to an output file on the local File system. For example: **C:\eGateData\Project3**.

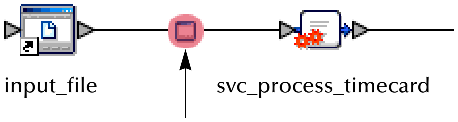
Set these directory paths later in the section **Create an Environment** on page 161.

Configure Inbound File eWay

Identify the inbound File eWay shown in Figure 160 and then set the properties.

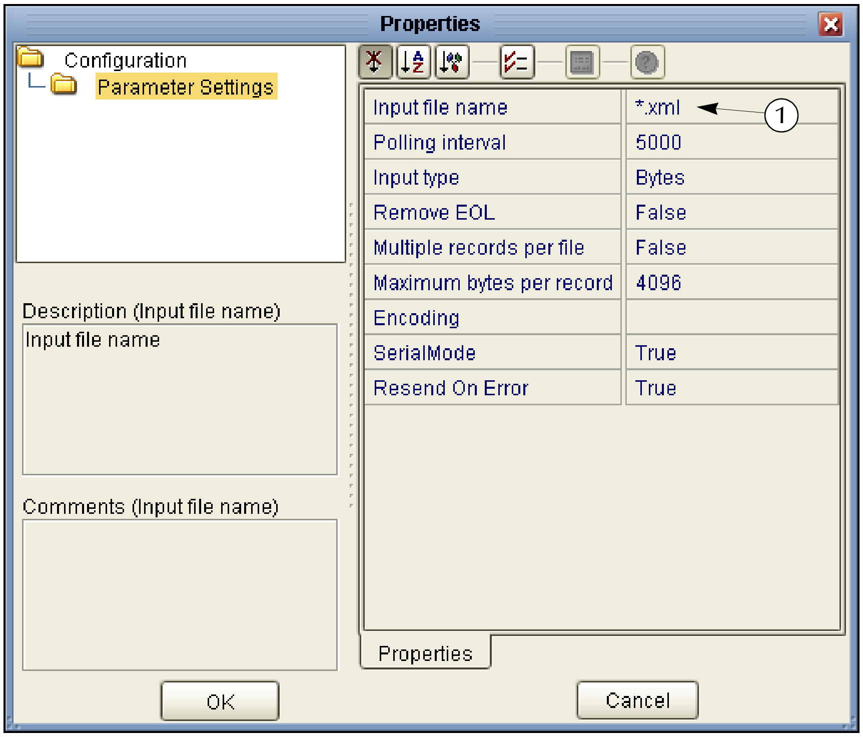
1 Double-click the first eWay.

Figure 160 Inbound File eWay



The Properties dialog box appears.

Figure 161 Properties Configuration (Inbound File eWay)



Description (Input file name)
Input file name

Comments (Input file name)

Input file name	*.xml
Polling interval	5000
Input type	Bytes
Remove EOL	False
Multiple records per file	False
Maximum bytes per record	4096
Encoding	
SerialMode	True
Resend On Error	True

2 Enter the input file name, or use a “wildcard” as shown in Figure 161. Be sure to include the asterisk and period: ***.xml** in the Input file name field.

Note: *Just put in the file name. You will enter the directory path later when you set up your Environment.*

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

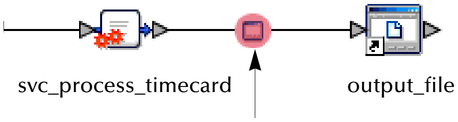
Note: *If the* **Polling interval** *value is left unchanged, the eWay polls the directory location every five seconds (5000 milliseconds).*

Configure Outbound File eWay

Identify the outbound File eWay and then set the properties.

1 Double-click the second eWay.

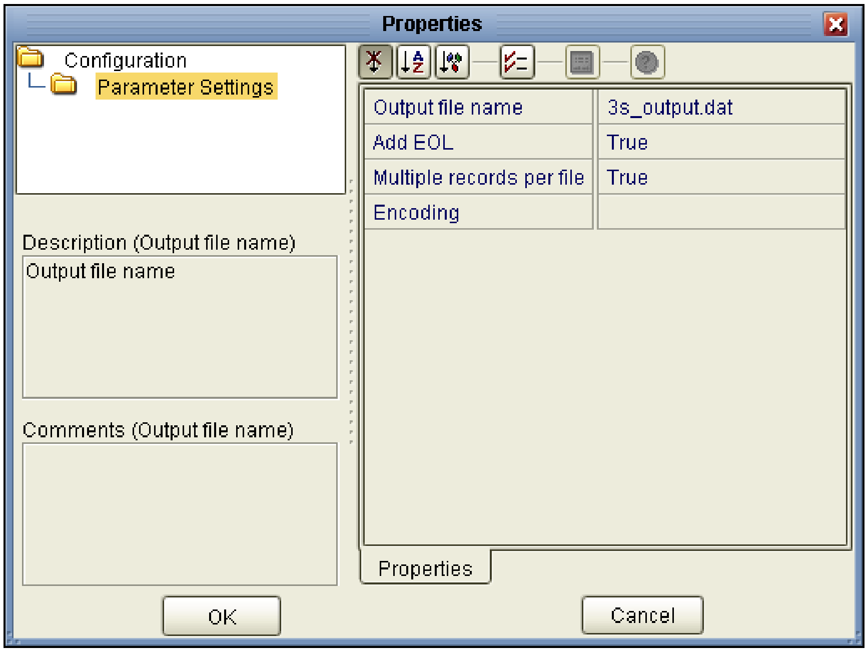
Figure 162 Outbound File eWay



The Properties dialog box appears.

2 Name the output File: **s3_output.dat**.

Figure 163 Properties Configuration (Outbound File eWay)



Output file name	3s_output.dat
Add EOL	True
Multiple records per file	True
Encoding	

In this scenario, accept the default settings (see Figure 163) for the output file. Additional records will be appended to the end of the file each time the input timecard File is processed.

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

Note: *See* **"Multiple records per File (True or False)" on page 59**.

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

You have now configured both the Inbound and Outbound File eWays and are ready to deploy your Project.

5.9 Create an Environment

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

Create a Logical Host and an External File System using Environment Explorer. First create an Environment. The Environment Explorer deploys resources required to implement a project and includes information about external systems that interact with eGate.

Note: *If you are using an existing Environment, you can also run this project in the same instance of the Logical Host (domain1 for example). But make sure you are using the correct input data, and make sure you are using the correct path to the data, as explained in this section.*

You can skip to **Create Deployment Profile** on page 164 if you have already created an Environment and want to use that same setup.

Add Logical Host and Servers

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

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

Add an Integration Server

The Integration Server is a J2EE software platform that houses the business logic container used to run Collaborations and JCA connectors (eWays).

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

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

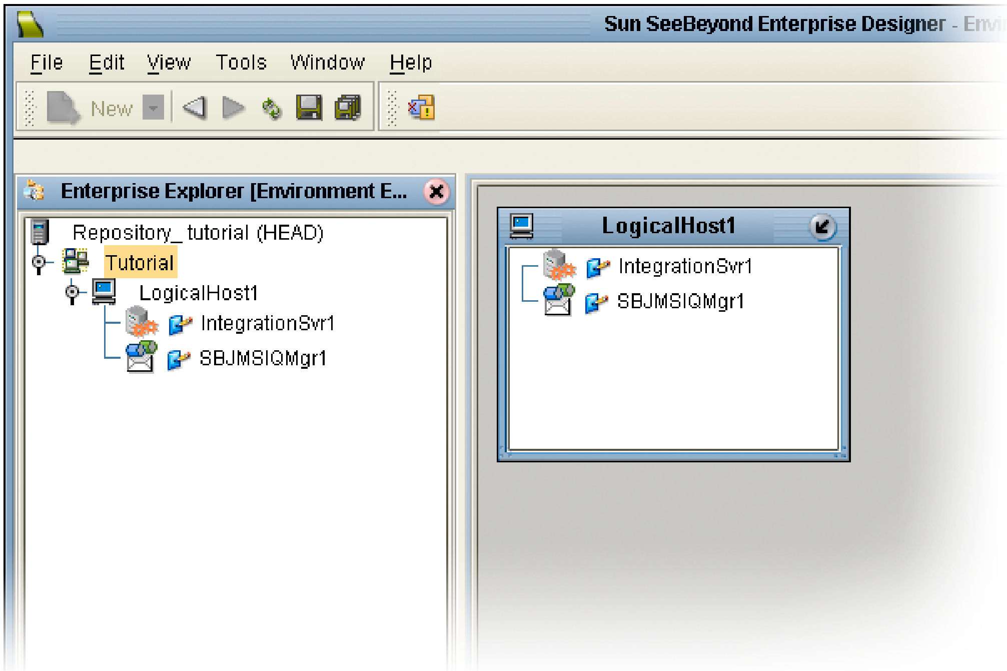
Add a JMS IQ Manager

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 appears in the **LogicalHost1** window.

Figure 164 Environment with Logicalhost


Sun SeeBeyond Enterprise Designer - Envi
File Edit View Tools Window Help
New
Enterprise Explorer [Environment E...
Repository_tutorial (HEAD)
Tutorial
LogicalHost1
IntegrationSvr1
SBJMSIQMgr1
LogicalHost1
IntegrationSvr1
SBJMSIQMgr1

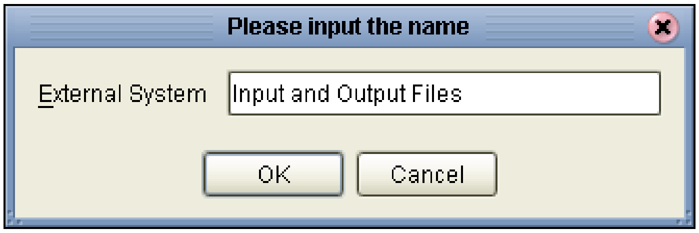
Add an External File System

Create a container to hold your input and output files.

1 Right-click **Tutorial** Environment.

2 Click **New**, **File External System**.

Figure 165 External files



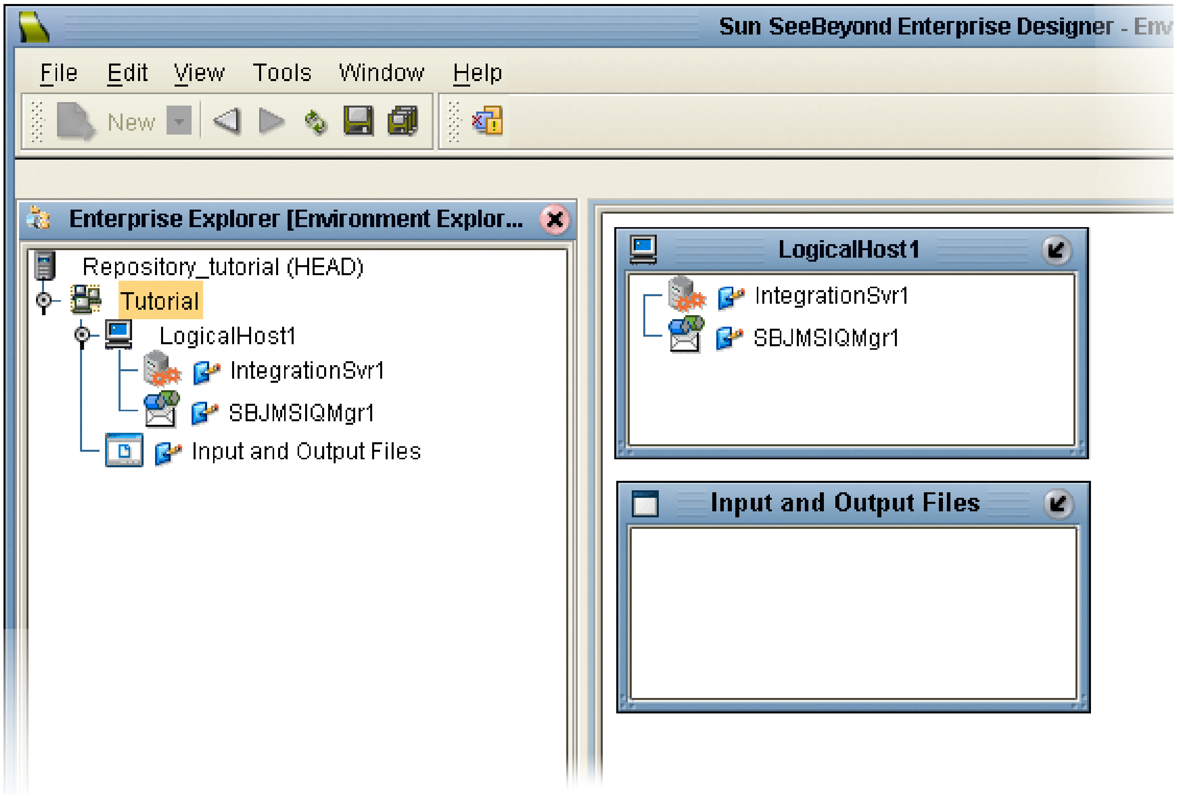
External System: Input and Output Files

3 Enter the External System name, **Input and Output Files**.

4 Click **OK**.

This places the External File component in the Enterprise Explorer tree. Your Environment Editor pane should look similar the following figure – if not – make sure your LogicalHost1 object isn't masked behind the newly created Input and Output Files object.

Figure 166 Environment Editor



5.10 Create Deployment Profile

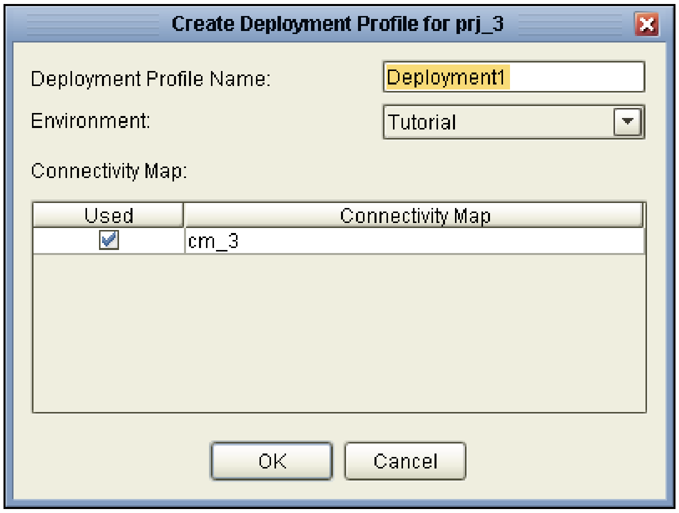
First create a Deployment Profile, and then deploy (map) the components. **prj_3** can run in the same domain (Integration Server and STCMS - JMS message server) as Project1 and/or Project2, and this Project can run at the same time. You still must create a Deployment Profile for **prj_3**.

Create a Deployment Profile

A Deployment Profile contains information about how Project components are mapped and deployed within an Environment.

1. Click the **Project Explorer** tab to return to Project Explorer in your Project.
2. Right-click **prj_3**, and then click **New**, **Deployment Profile**.

Figure 167 Deployment Profile



Create Deployment Profile for prj_3

Deployment Profile Name: Deployment1

Environment: Tutorial

Connectivity Map:

Used	Connectivity Map
☑	cm_3

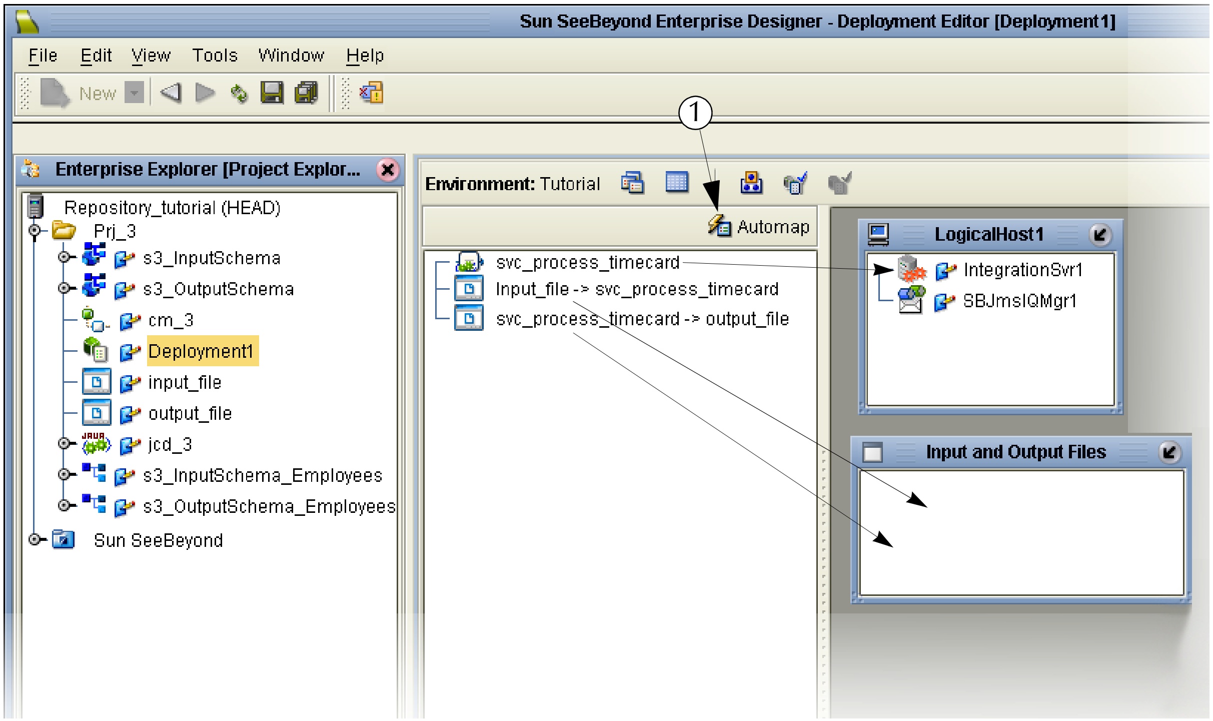
3 Accept the default name **Deployment1** (and **Tutorial**). Click **OK**.

Your files and **svc_process_timecard** appear in the center pane of the Environment editor. Deploy these components using the Automap feature.

If not already open, double-click **Deployment1** in the Project Explorer to open the Environment editor.

4 Click the **Automap** button (number 1 in the following figure) in the Environment toolbar. (Refer to Figure 168.)

Figure 168 Environment Automap Components



The center pane of the Environment Editor should be empty after all the available objects and components have been mapped to the Environment. You are now ready to build and deploy your Project.

5 Save your Project.

5.11 Build and Deploy the Project

Make sure an instance of the Logical Host is running (domain) before you deploy your project. You can start the domain with the **domainmgr.bat** script that is located in your, **logicalhost** directory.

Enter Passwords

Make sure you have entered your user name and password (see the steps below) for the Integration Server (and the JMS IQ Manager), and make sure your domain (Integration Server and STCMS - JMS server) is running.

Integration Server Password

Enter a password and verify the URL port for the Integration Server.

1 Click the **Environment Explorer** tab.

2 Right-click **IntegrationSvr1** in the Environment Explorer tree.

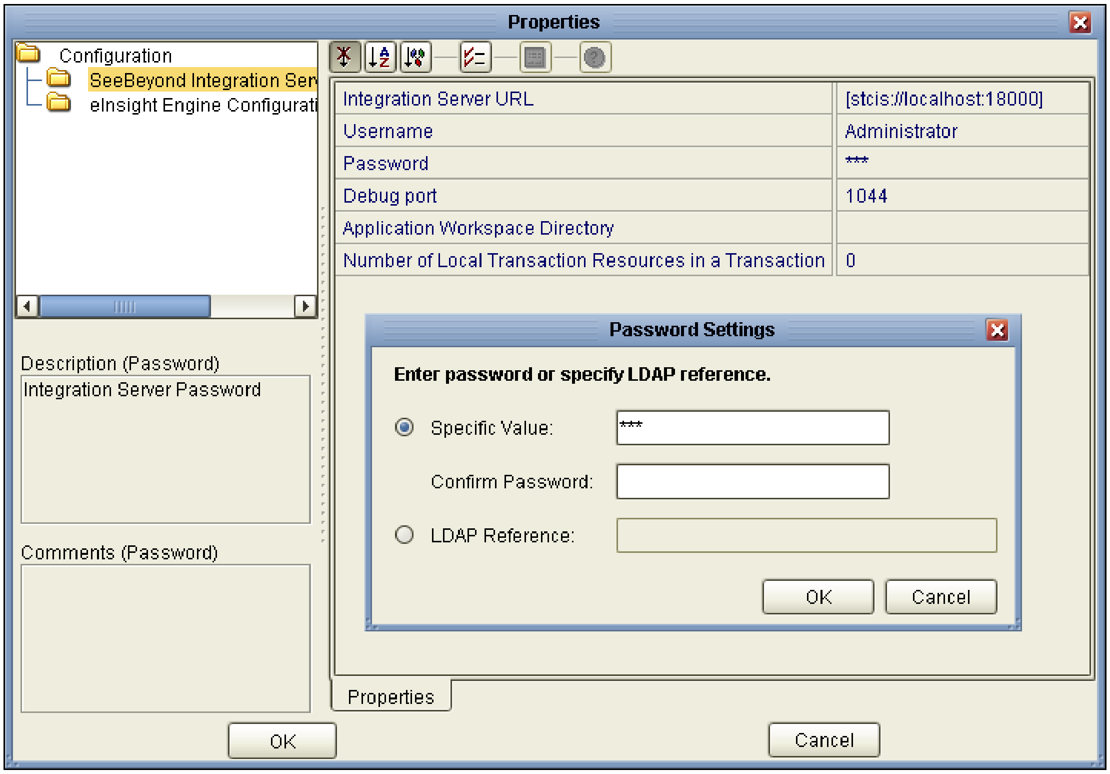
3 Click **Properties**.

4 Enter and confirm **Password**.

5 Verify the Integration Server URL, Admin port.

See the following figure.

Figure 169 Integration Server Password



Integration Server URL	[stcis://localhost:18000]
Username	Administrator
Password	***
Debug port	1044
Application Workspace Directory	
Number of Local Transaction Resources in a Transaction	0

Description (Password)
Integration Server Password

Comments (Password)

Enter password or specify LDAP reference.

● Specific Value: ***

Confirm Password:

○ LDAP Reference:

6 Click **OK**.

JMS IQ Manager Password

Enter a password and verify the URL port for the message server.

1 Click the **Environment Explorer** tab.

2 Right-click **SBJMSIQMgr1** in the Environment Explorer tree.

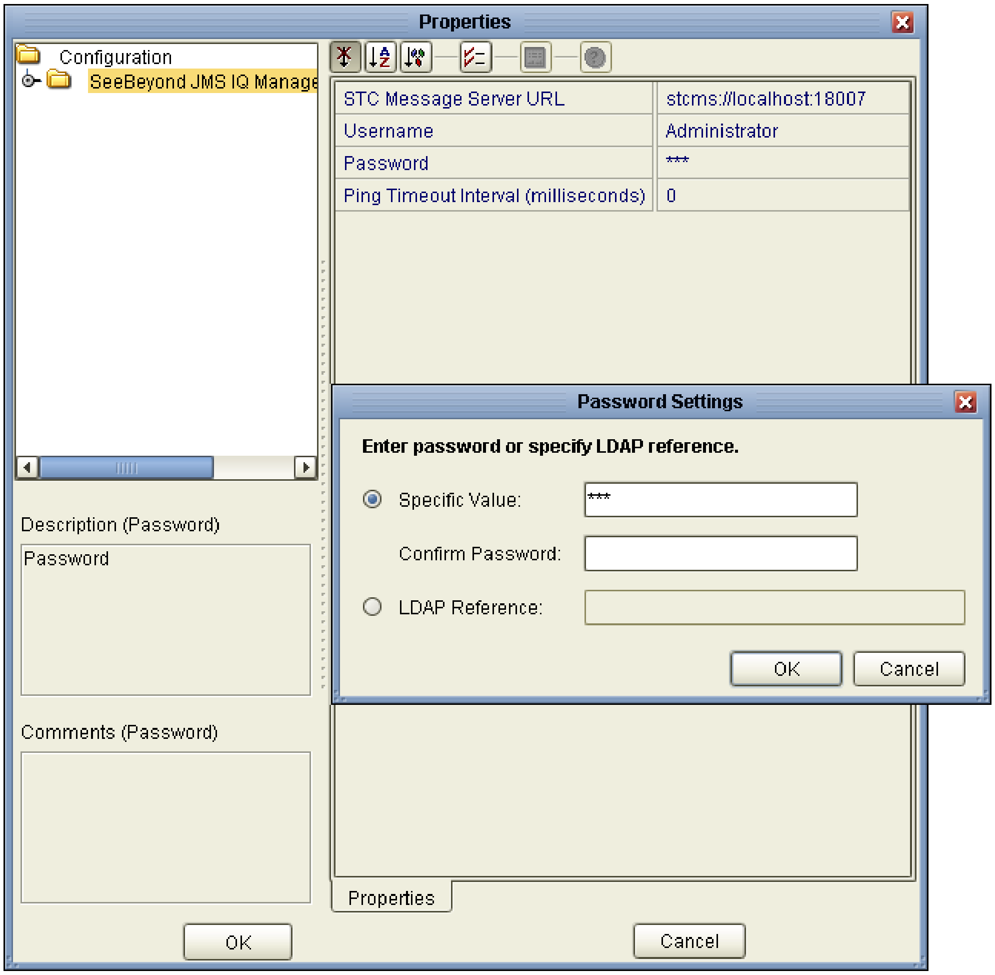
3 Click **Properties**.

4 Enter and confirm **Password**.

5 Verify your domain **IQ Manager** URL port number. (You may have to enter the localhost and port number - for example - **localhost:18007**, as shown in the following figure.)

See the following figure.

Figure 170 JMS IQ Manager



STC Message Server URL	stcms://localhost:18007
Username	Administrator
Password	***
Ping Timeout Interval (milliseconds)	0

Password

Enter password or specify LDAP reference.

● Specific Value: ***

Confirm Password:

○ LDAP Reference:

6 Click **OK**.

5.11.1 Set the Properties for the File eWays

In the Environment Explorer, set the Properties for your Inbound and Outbound File eWays.

1 Right-click the **Input and Output Files** object in your Environment Explorer tree.

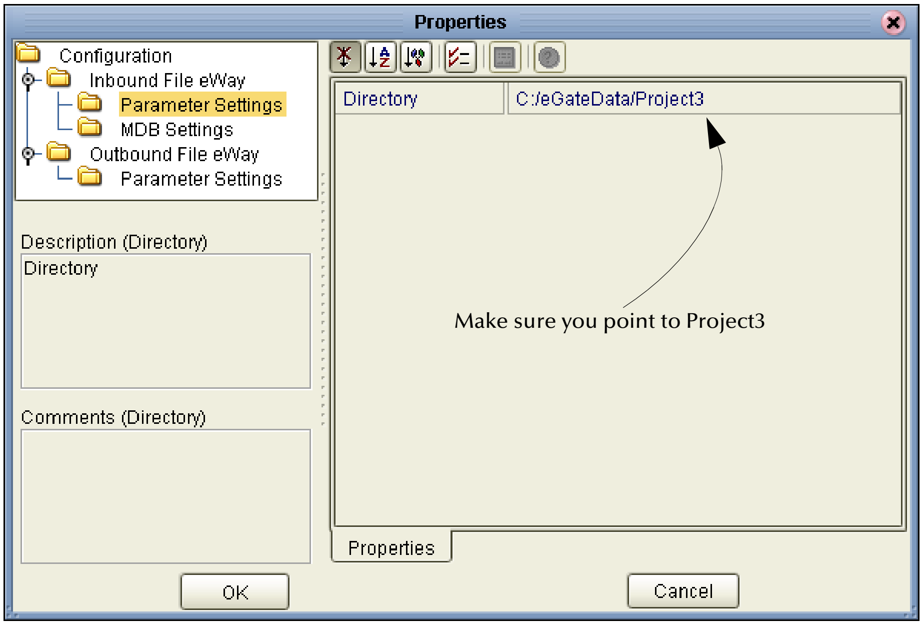
2 Click **Properties**.

3 Open the nodes for your Inbound and Outbound File eWays.

4 Click **Parameter Settings**. (Set the path for both the Inbound and Outbound File eWays. The path will be the same for both.)

5 Enter the path. (Do not put the file name in the path.)

Figure 171 Properties Setting for the Inbound File eWay



Properties

Description (Directory)

Directory

Comments (Directory)

Directory	C:/eGateData/Project3

Make sure you point to Project3

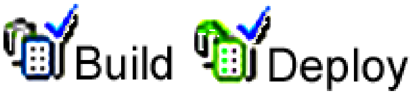
6 Click **OK** and **Save All**.

Important: *Be sure to set the Properties for the Outbound File eWay also. The directory path is the same.*

5.11.2 Build

The Build compiles the Service(s) and Java files, then creates the Project EAR File (application).

Figure 172 Build Button



1 Click the Build button in the Environment toolbar.

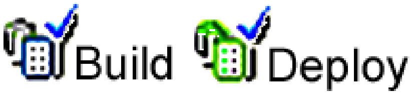
After the Project EAR File is created the following message appears: “Project build was successful (elapsed time: xx.xxx seconds).”

You may now deploy your project.

5.11.3 Deploy

Deploy your Project with an instance of the Logical Host (domain) running. The executable domain files are in your Logicalhost folder.

Figure 173 Deploy Button



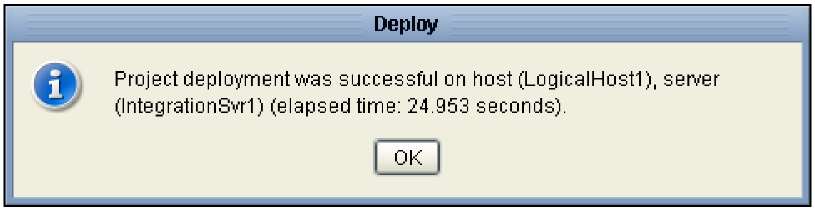
1 Click the Deploy button.

2 Click **Yes** when you see the message "Are you sure you want to deploy the current project deployment?"

A message displays: "Deploying the application..."

You will receive a message similar to the following figure when the deployment is successful.

Figure 174 Deployment Successful



Project deployment was successful on host (LogicalHost1), server (IntegrationSvr1) (elapsed time: 24.953 seconds).

3 Click **OK**.

Your Project (application) has been deployed.

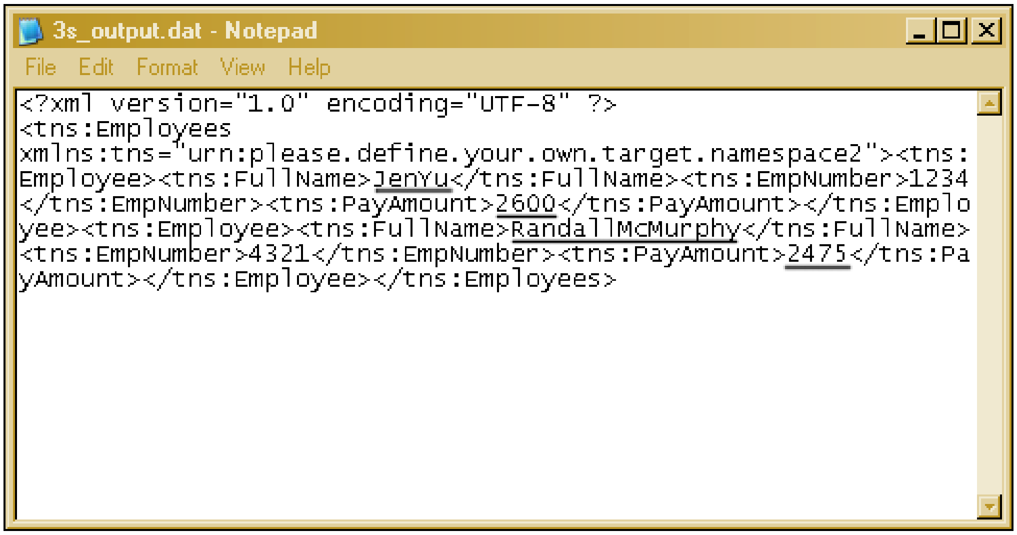
5.12 Verify Output Data

The Management Agent processes your input File, **s3_TimeCard.xml**, and writes the results to **s3_output.dat**.

View Output

- Use Windows Explorer to navigate to **C:\eGateData\Project3**.
- See that the Inbound File eWay, after processing the inbound file, renamed the file to **s3_TimeCard.xml~in**. You can make changes in this file using a text editor and then rename the file back to **s3_TimeCard.xml** and immediately see the results of your changes after the file is processed again.
- See that the Output eWay generated the output File, **s3_output.dat**.
- Use Notepad or a similar text editor to view the contents of the output file.

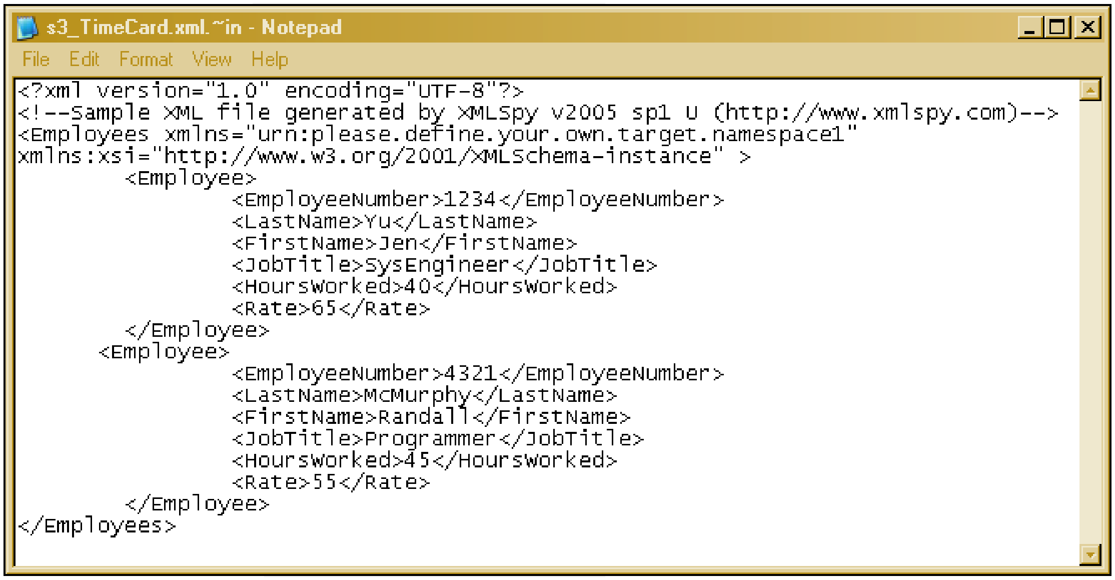
Figure 175 Output File



```
<?xml version="1.0" encoding="UTF-8" ?>
<tns:Employees
xmlns:tns="urn:please.define.your.own.target.namespace2"><tns:
Employee><tns:FullName>JenYu</tns:FullName><tns:EmpNumber>1234
</tns:EmpNumber><tns:PayAmount>2600</tns:PayAmount></tns:Emplo
yee><tns:Employee><tns:FullName>RandallMcMurphy</tns:FullName>
<tns:EmpNumber>4321</tns:EmpNumber><tns:PayAmount>2475</tns:Pa
yAmount></tns:Employee></tns:Employees>
```

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

Figure 176 Input File



```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSpy v2005 sp1 U (http://www.xmlspy.com)-->
<Employees xmlns="urn:please.define.your.own.target.namespace1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
        <Employee>
                <EmployeeNumber>1234</EmployeeNumber>
                <LastName>Yu</LastName>
                <FirstName>Jen</FirstName>
                <JobTitle>SysEngineer</JobTitle>
                <HoursWorked>40</HoursWorked>
                <Rate>65</Rate>
        </Employee>
      <Employee>
                <EmployeeNumber>4321</EmployeeNumber>
                <LastName>McMurphy</LastName>
                <FirstName>Randall</FirstName>
                <JobTitle>Programmer</JobTitle>
                <HoursWorked>45</HoursWorked>
                <Rate>55</Rate>
        </Employee>
</Employees>
```

5.12.1 Edit Input File in Real Time

With an instance of the Logical Host (domain) running, you may edit the input timecard file and change the data names or values in that file.

1. Edit the input file using a text editor. As a test you can change names or values in the file.

2 Rename the input file from **s3_TimeCard.xml.~in**. to **s3_TimeCard.xml**. Delete the file extension name **~in**.

Within a few seconds the File eWay processes the input file again and writes the results to **s3_output.dat**.

Chapter 6

Web Services - Scenario4

Web services are web-based enterprise applications that use open, XML-based standards and transport protocols to exchange data with calling clients. With the eGate Integrator you can expose a (JCD) Java Collaboration Definition as a web service.

What's in This Chapter

6.1 Expose a JCD as a Web Service

In Scenario4 you create a Project that exposes a JCD as a web service. The sample input and output files (not used in this scenario) show what the data looks like before and after processing (data manipulation). You need a client in order to actually get the results of the processing. It is the intention of this scenario to expose a JCD as a web service using the eGate product, but this scenario does not use a client.

For an example of invoking a web service in a client application, see the **eInsight** user guides.

- A timecard system tracks the weekly hours worked by employees. The data in the system is in text format with six fields: **EmployeeNumber**, **LastName**, **FirstName**, **JobTitle**, **HoursWorked**, and **Rate**.
- A very simple payroll file is created. The output fields are **EmployeeNumber**, **FirstName** and **LastName** concatenated and **GrossPay**. This output file is in XML format.

- The value for the **GrossPay** field in the Payroll system must be calculated from data in the Timecard system.

The input is submitted through a client. And the output is returned to a client.

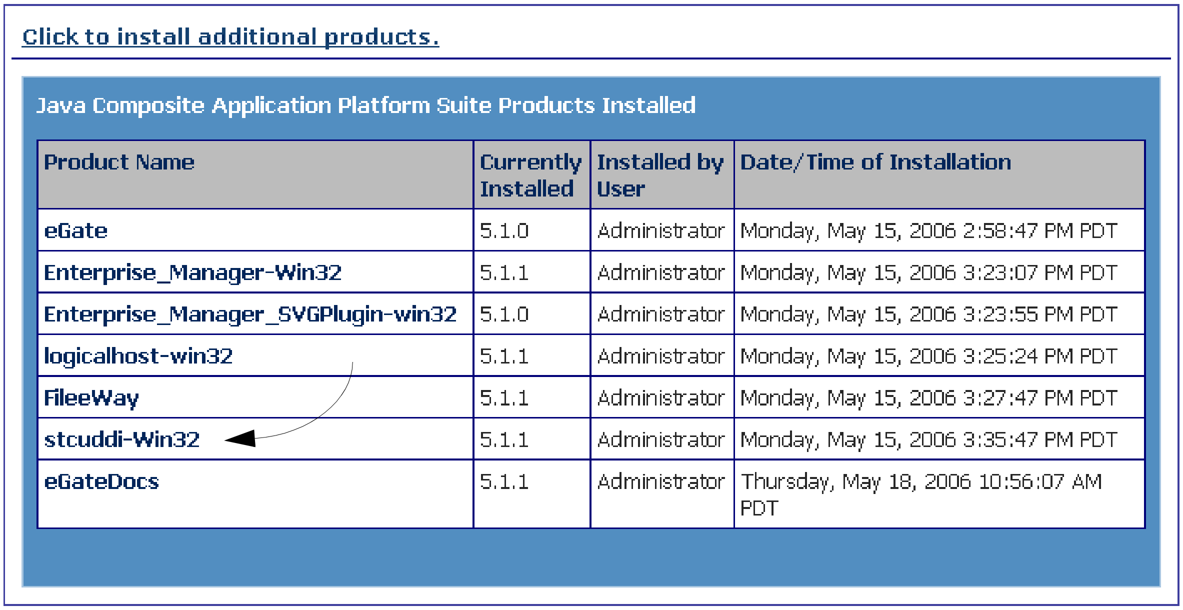
6.2 Required Resources

In addition to the products and resources you need to do the sample scenarios in the Tutorial, you need the **UDDI Server** to run the web services scenario. This feature is part of the eGate Integrator.

Download the UDDI Installer

1. Login to the **Enterprise Manager**, and enter your **User ID** and **Password**.
2. Make sure the UDDI product is listed among the **Products Installed**.

Figure 177 Download



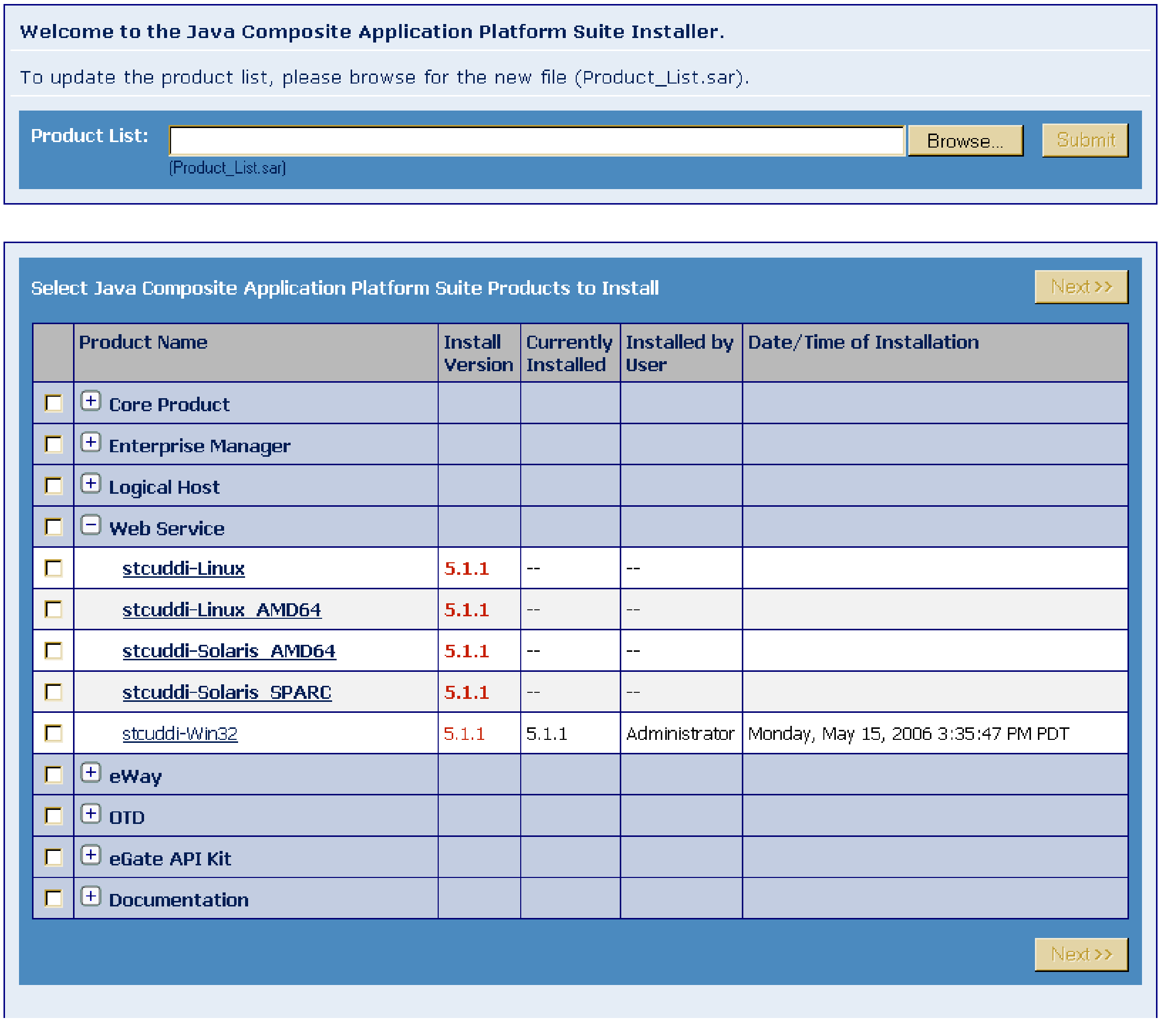
Click to install additional products.

Java Composite Application Platform Suite Products Installed

Product Name	Currently Installed	Installed by User	Date/Time of Installation
eGate	5.1.0	Administrator	Monday, May 15, 2006 2:58:47 PM PDT
Enterprise_Manager-Win32	5.1.1	Administrator	Monday, May 15, 2006 3:23:07 PM PDT
Enterprise_Manager_SVGPlugin-win32	5.1.0	Administrator	Monday, May 15, 2006 3:23:55 PM PDT
logicalhost-win32	5.1.1	Administrator	Monday, May 15, 2006 3:25:24 PM PDT
FileeWay	5.1.1	Administrator	Monday, May 15, 2006 3:27:47 PM PDT
stcuddi-Win32	5.1.1	Administrator	Monday, May 15, 2006 3:35:47 PM PDT
eGateDocs	5.1.1	Administrator	Thursday, May 18, 2006 10:56:07 AM PDT

3. If the UDDI product is not listed, install it from the **Products to Install** list. See the following figure.

Figure 178 Select UDDI Server



Welcome to the Java Composite Application Platform Suite Installer.

To update the product list, please browse for the new file (Product_List.sar).

Product List:

(Product_List.sar)

Select Java Composite Application Platform Suite Products to Install

	Product Name	Install Version	Currently Installed	Installed by User	Date/Time of Installation
□	⊞ Core Product				
□	⊞ Enterprise Manager				
□	⊞ Logical Host				
□	⊟ Web Service				
□	stcuddi-Linux	5.1.1	--	--	
□	stcuddi-Linux_AMD64	5.1.1	--	--	
□	stcuddi-Solaris_AMD64	5.1.1	--	--	
□	stcuddi-Solaris_SPARC	5.1.1	--	--	
□	stcuddi-Win32	5.1.1	5.1.1	Administrator	Monday, May 15, 2006 3:35:47 PM PDT
□	⊞ eWay				
□	⊞ OTD				
□	⊞ eGate API Kit				
□	⊞ Documentation				

4 Download the UDDI win32 Sar file and install it on your local host.

5 Extract the files to a location on your computer. For example:
`C:\JavaCAPS51\``UDDIServer`.

6 Run **Install.bat**.

7 In your **UDDIServer** folder, execute **startup.bat** to start your UDDI Server.

6.2.1 Download the Sample File

See "Download the Sample files" on page 23. The sample files used in the scenario are in the Project4 folder.

Sample Input Schema

`Input_Employee.xsd`

Sample Output Schema

`Output_Employee.xsd`

Sample Input File

`InputWS.xml`

6.3 Create a New Project

Create a new Project to expose a JCD as a web service.

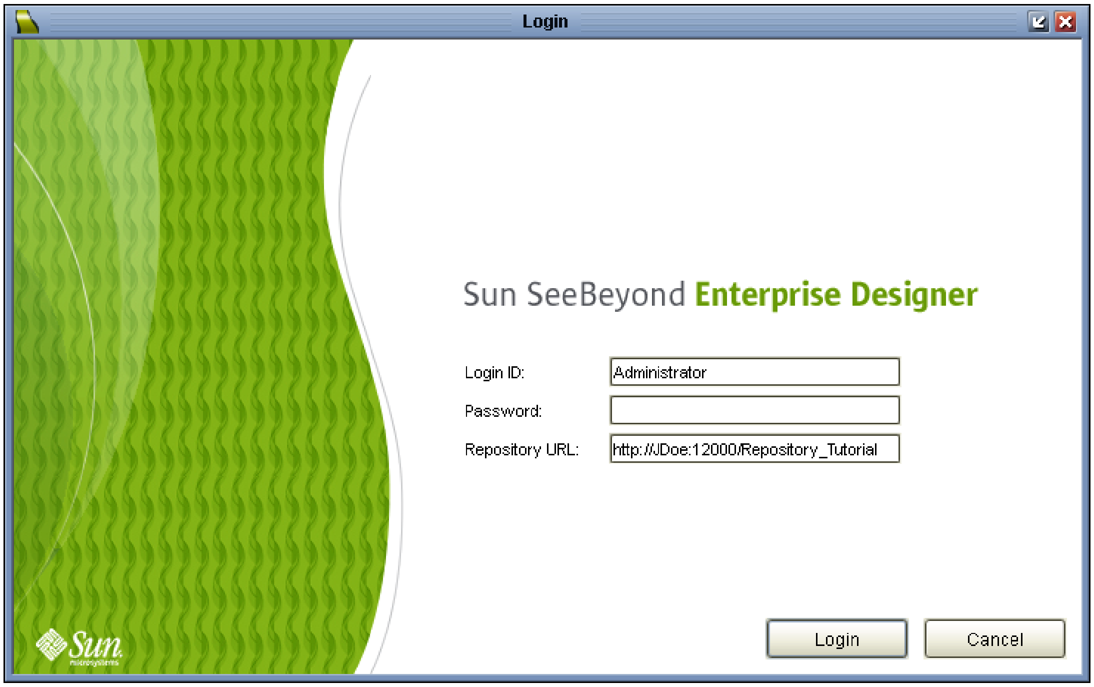
Start Enterprise Designer

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

1 Start the Enterprise Designer by executing **runed.bat** in your JavaCAPS51 folder: **`C:\JavaCAPS51\edesigner\bin`**.

The Enterprise Designer Login dialog box appears.

Figure 179 Enterprise Designer Login Dialog Box



Login ID: Administrator
Password:
Repository URL: http://JDoe:12000/Repository_Tutorial

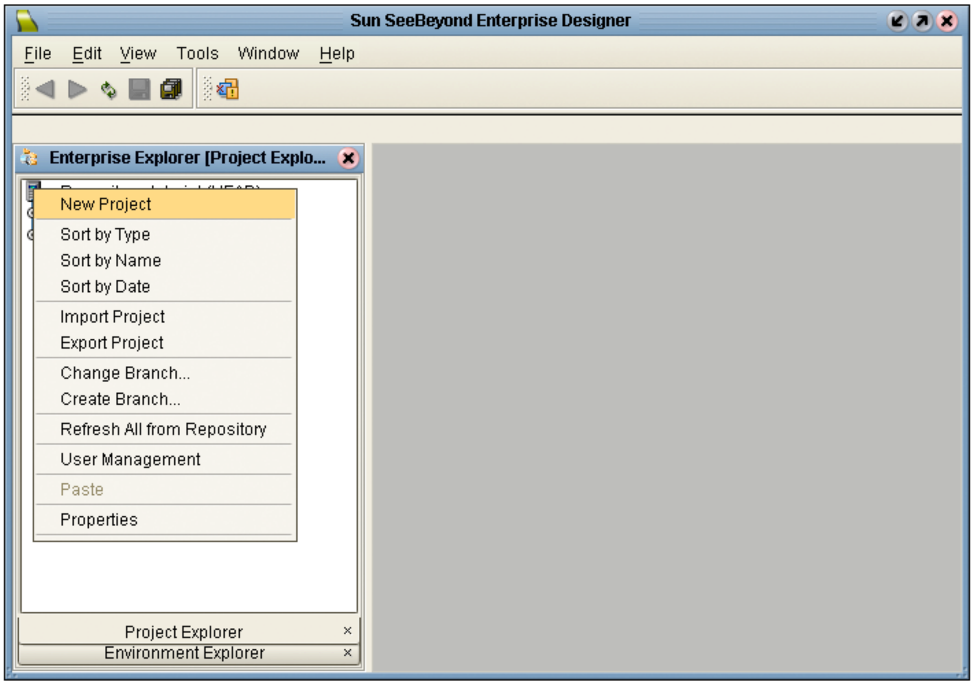
2 Type your **Username** and **Password**. Click **Log in** to start the Enterprise Designer.

Create prj_Web_Ser

Create a new Project called **prj_Web_Ser** and use the sample data in the Scenario4 folder.

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

Figure 180 Create a WS Project

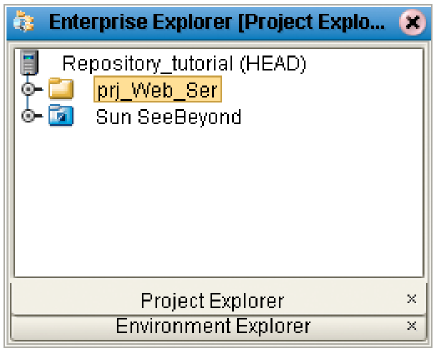


2 Name your Project, **prj_Web_Ser**.

3 Press **Enter**.

Your **prj_Web_Ser** appears in the Project Explorer tree.

Figure 181 Project Folder: prj_Web_Ser


Enterprise Explorer [Project Explo...
Repository_tutorial (HEAD)
prj_Web_Ser
Sun SeeBeyond
Project Explorer
Environment Explorer

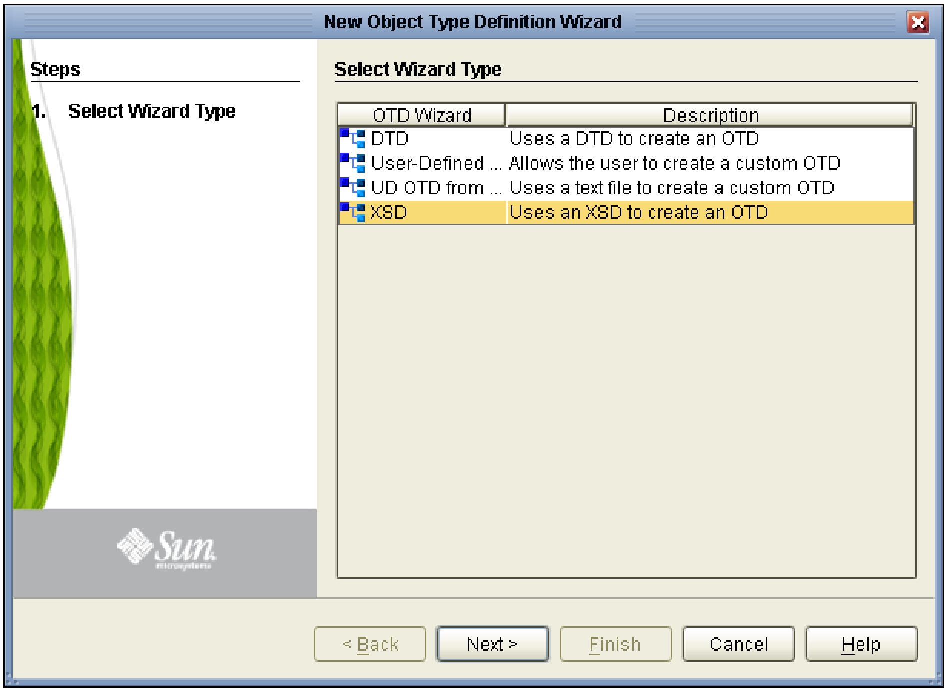
6.4 Create an Object Type Definition

An OTD describes external data formats that characterize the input and output data in a Collaboration Definition.

Select OTD Definition

1 In the Project Explorer tree, right-click the **prj_Web_Ser** icon, then click **New**, **Object Type Definition**.

Figure 182 Select XSD Wizard

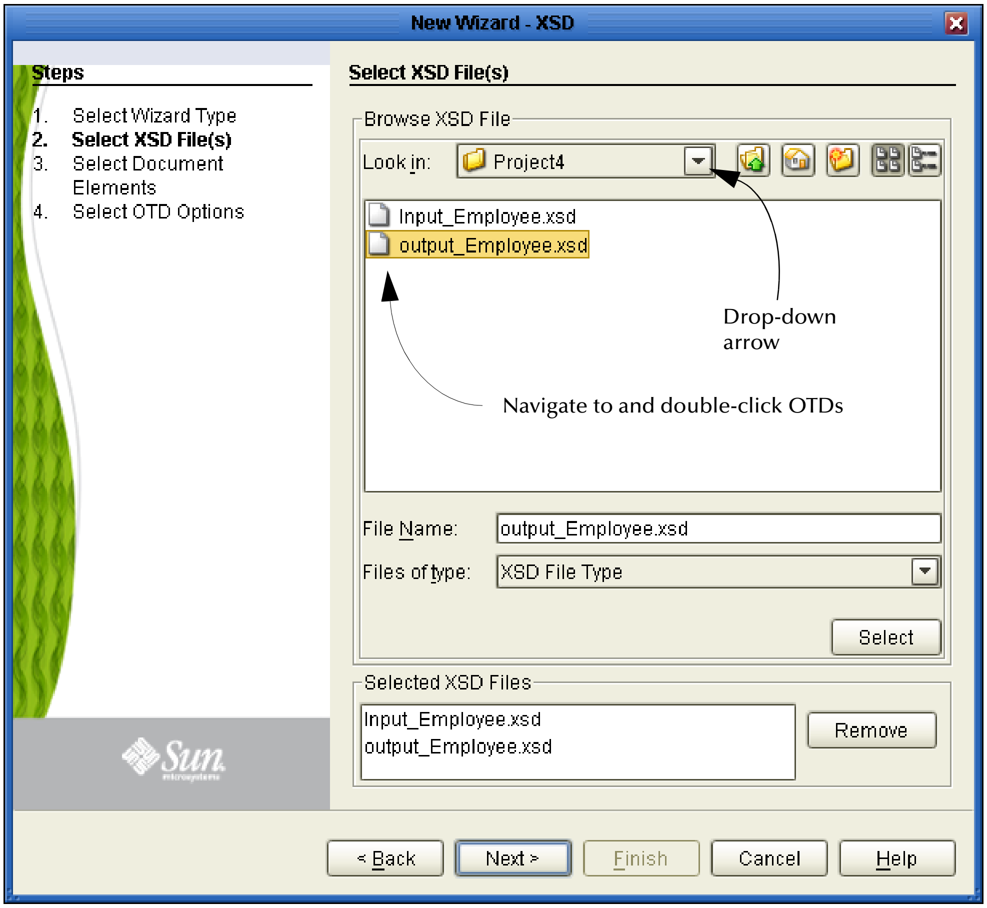


OTD Wizard	Description
DTD	Uses a DTD to create an OTD
User-Defined ...	Allows the user to create a custom OTD
UD OTD from ...	Uses a text file to create a custom OTD
XSD	Uses an XSD to create an OTD

2 Select the **XSD** wizard and click **Next**.

3 Click the **drop-down arrow**, and navigate to your XSD schemas.

Figure 183 Navigate to XSD Schemas



New Wizard - XSD

Steps

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

Select XSD File(s)

Browse XSD File

Look in: Project4

Input_Employee.xsd
output_Employee.xsd

File Name: output_Employee.xsd

Files of type: XSD File Type

Selected XSD Files

Input_Employee.xsd
output_Employee.xsd

Drop-down arrow

Navigate to and double-click OTDs

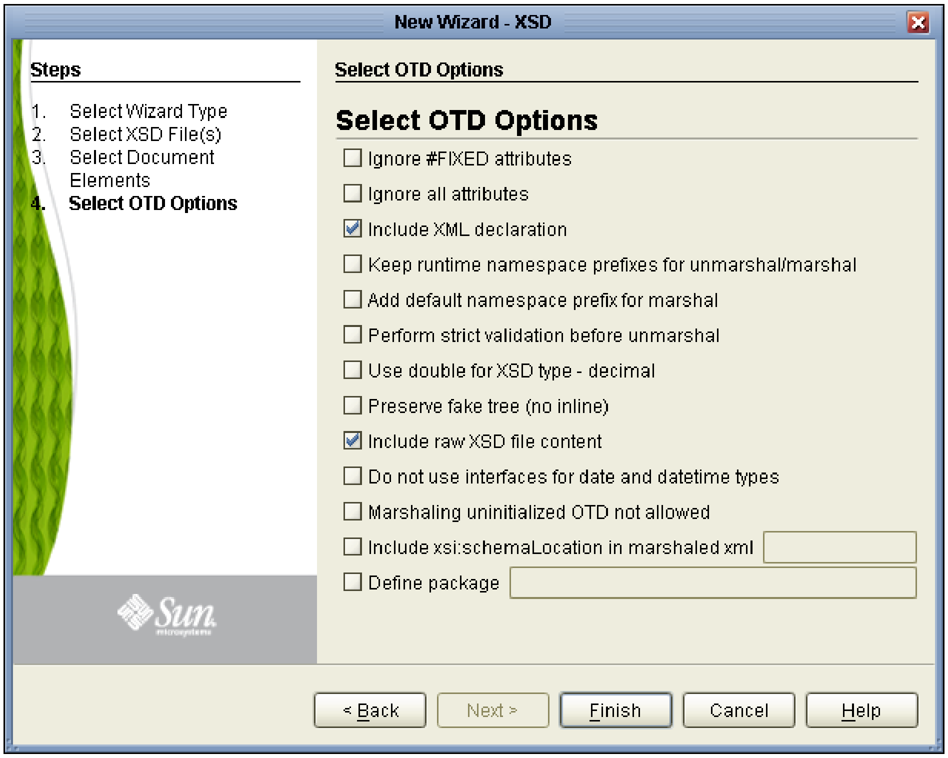
4 Select your input and output XSD schemas (or double-click).

5 When your XSDs appear in the **Selected XSD files** dialog box, click **Next**.

6 When the next dialog box appears, select both document elements and click **Next**.

The Select OTD Options dialog box appears. See the following Figure 184.

Figure 184 Select OTD Options



New Wizard - XSD

Steps

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

Select OTD Options

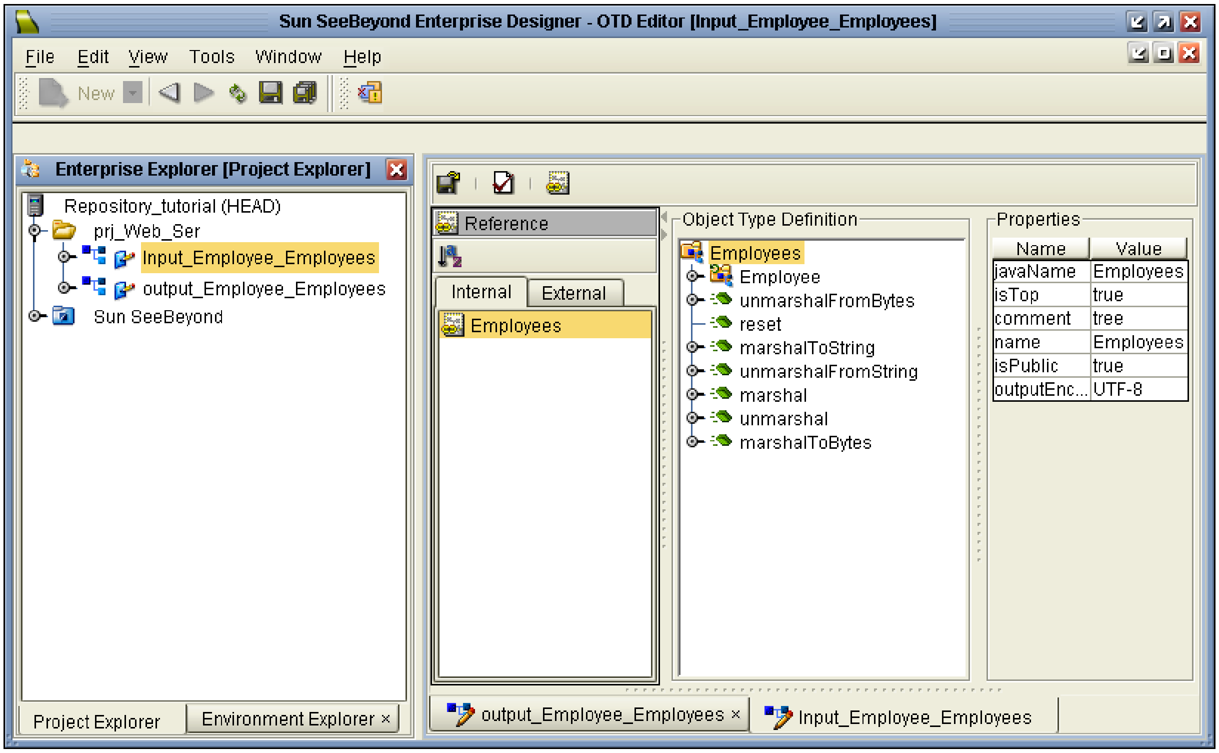
Select OTD Options

- □ Ignore #FIXED attributes
- □ Ignore all attributes
- ☑ Include XML declaration
- □ Keep runtime namespace prefixes for unmarshal/marshal
- □ Add default namespace prefix for marshal
- □ Perform strict validation before unmarshal
- □ Use double for XSD type - decimal
- □ Preserve fake tree (no inline)
- ☑ Include raw XSD file content
- □ Do not use interfaces for date and datetime types
- □ Marshaling uninitialized OTD not allowed
- □ Include xsi:schemaLocation in marshaled xml
- □ Define package

7 Accept the default check boxes, **Include XML declaration** and **Include raw XSD file content**. Click **Finish**.

The input and output OTDs for Employees appears.

Figure 185 Employee OTD



Name	Value
javaName	Employees
isTop	true
comment	tree
name	Employees
isPublic	true
outputEnc...	UTF-8

Click the tabs at the bottom of the OTD editor to toggle between the input and output Employee OTDs.

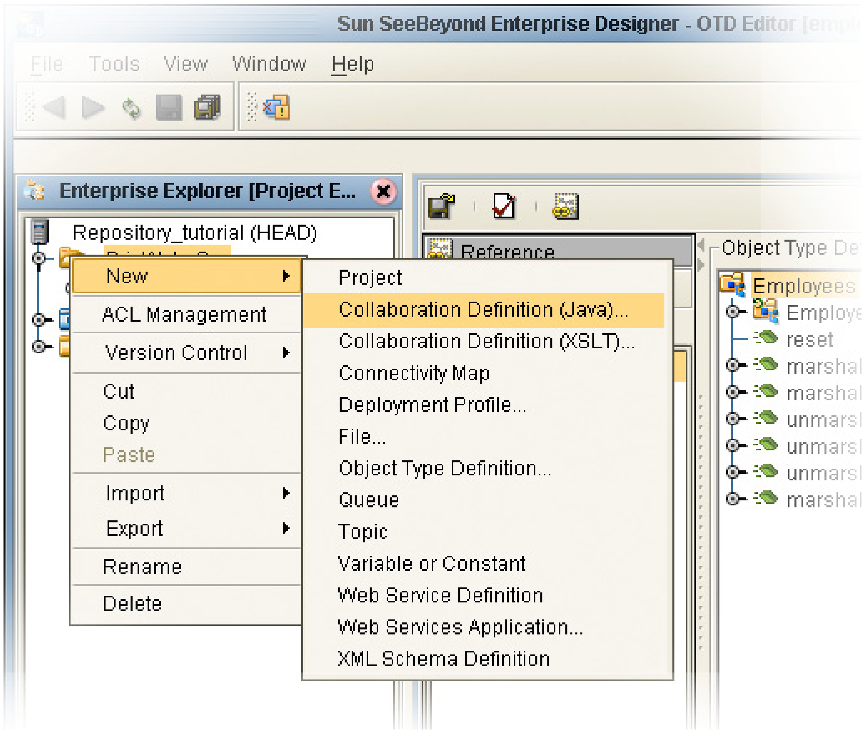
6.5 Create a Java Collaboration Definition

Create a Collaboration that is a new web service operation. Select callable as an external SOAP web service.

Select New Java Collaboration Definition

1 In the Project Explorer tree, right-click the **prj_Web_Ser** icon, then click **New**, **Collaboration Definition (Java)**.

Figure 186 Java Collaboration

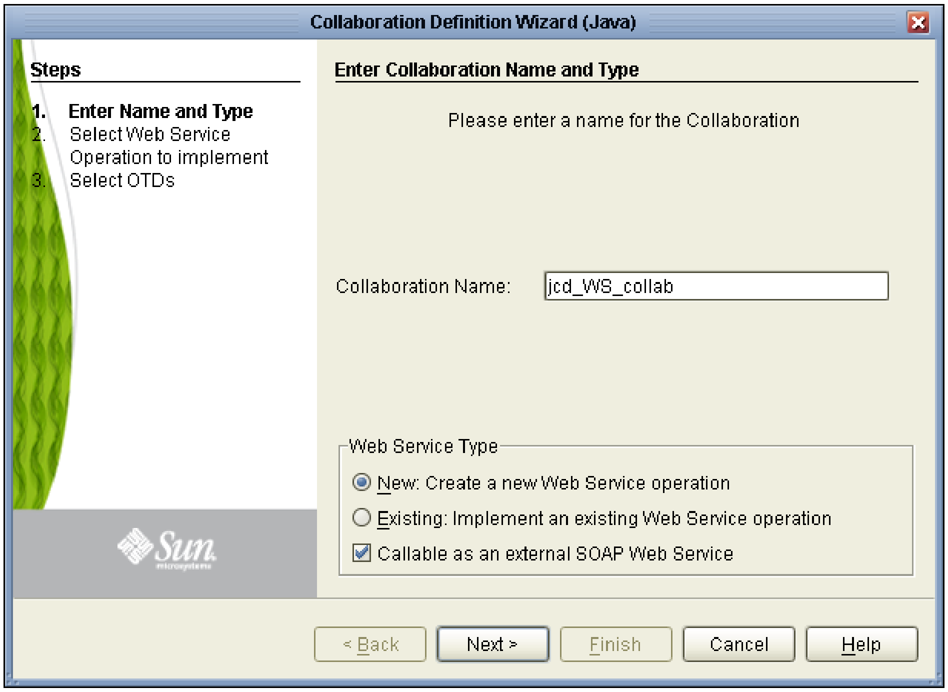


The **Collaboration Definition Wizard** appears.

2 Enter **jcd_WS_collab** as the Collaboration name.

3 Check the options, as shown in the following figure.

Figure 187 Collaboration 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: jcd_WS_collab

Web Service Type

● New: Create a new Web Service operation

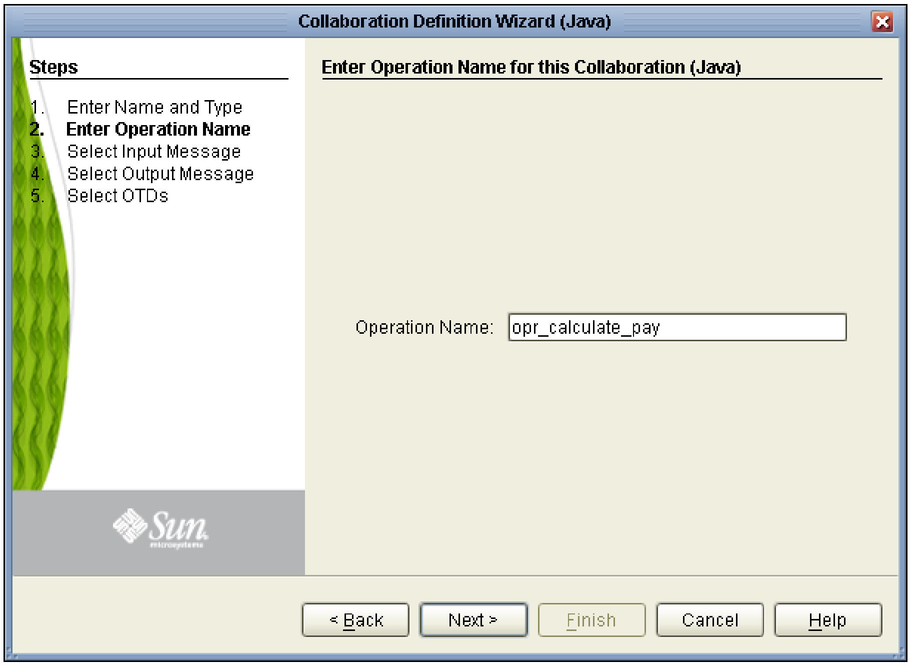
○ Existing: Implement an existing Web Service operation

☑ Callable as an external SOAP Web Service

4 Click **Next**.

The **Operation Name** dialog box appears.

Figure 188 Operation Name



Collaboration Definition Wizard (Java)

Steps

1. Enter Name and Type
2. **Enter Operation Name**
3. Select Input Message
4. Select Output Message
5. Select OTDs

Enter Operation Name for this Collaboration (Java)

Operation Name: opr_calculate_pay

5 Enter the operation name, **WS_calculate_pay**.

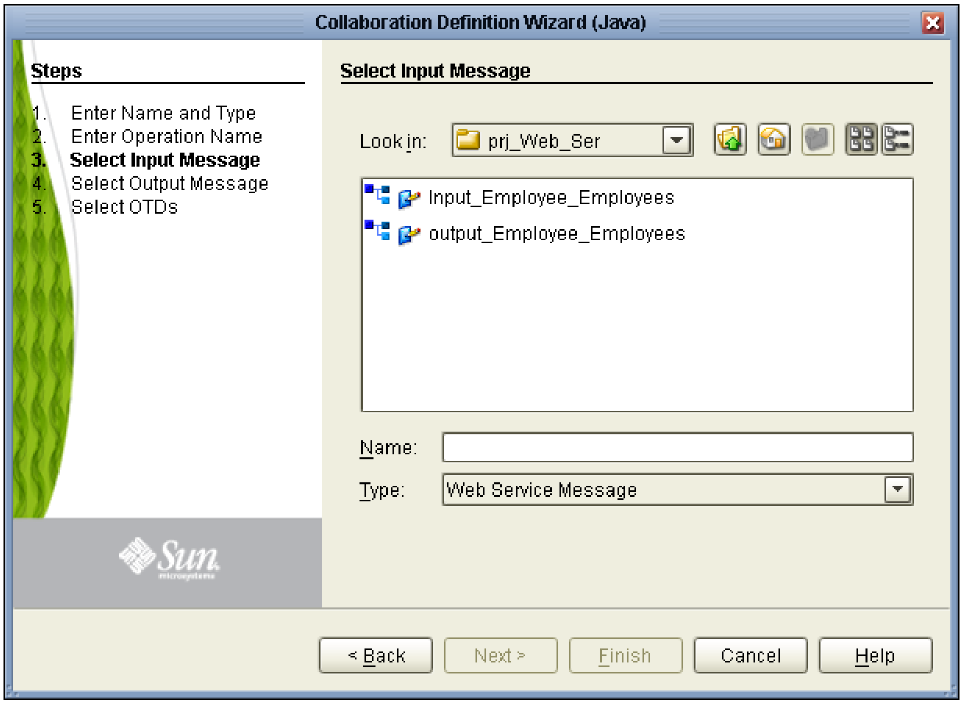
6 Click **Next**.

The **Select Input Message** dialog box appears.

7 Click your Project, **prj_Web_Ser**, then navigate to your input schema (OTD).

8 Select your input schema, and click **Next**.

Figure 189 Select Message Input



Look in: prj_Web_Ser

Input_Employee_Employees
output_Employee_Employees

Type: Web Service Message

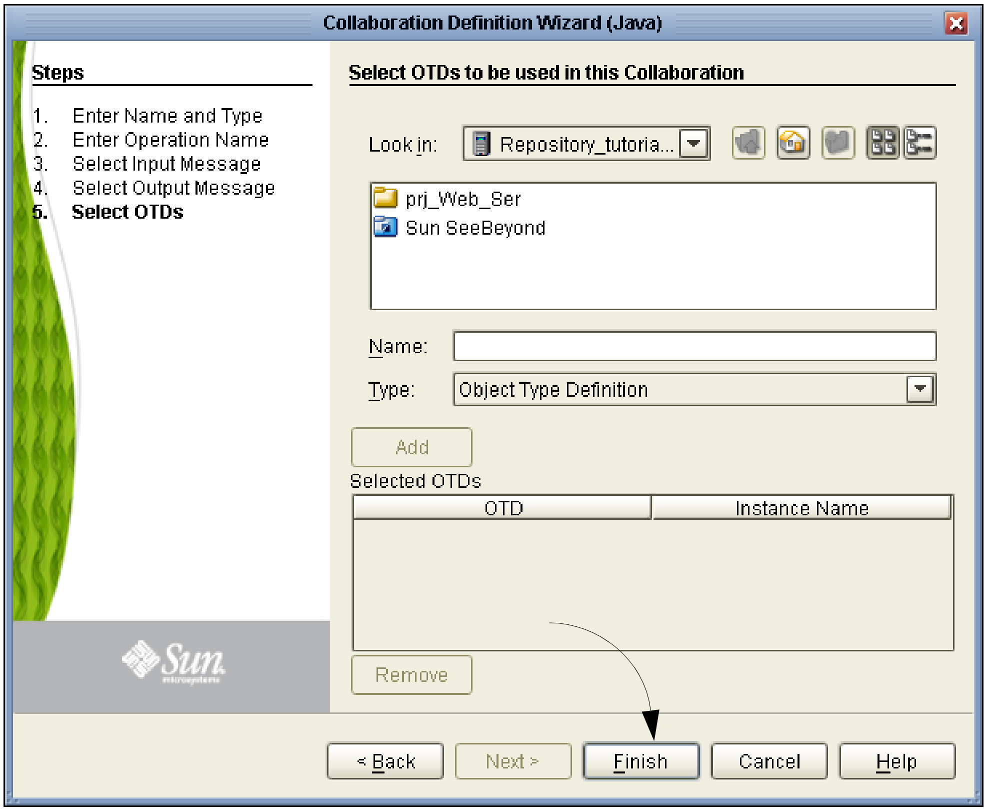
Select the output message.

9 Repeat the previous steps, and select your output schema (OTD).

Note: *After clicking Next, you do not need to select any more OTDs in Step 5 "Select OTDs." This is because you have already selected the input and output messages, which are OTDs.*

10 Click **Finish** without selecting additional OTDs. See the following figure.

Figure 190 Select Additional OTDs



Collaboration Definition Wizard (Java)

Steps

1. Enter Name and Type
2. Enter Operation Name
3. Select Input Message
4. Select Output Message
5. **Select OTDs**

Select OTDs to be used in this Collaboration

Look in: Repository_tutoria...

prj_Web_Ser
Sun SeeBeyond

Name:

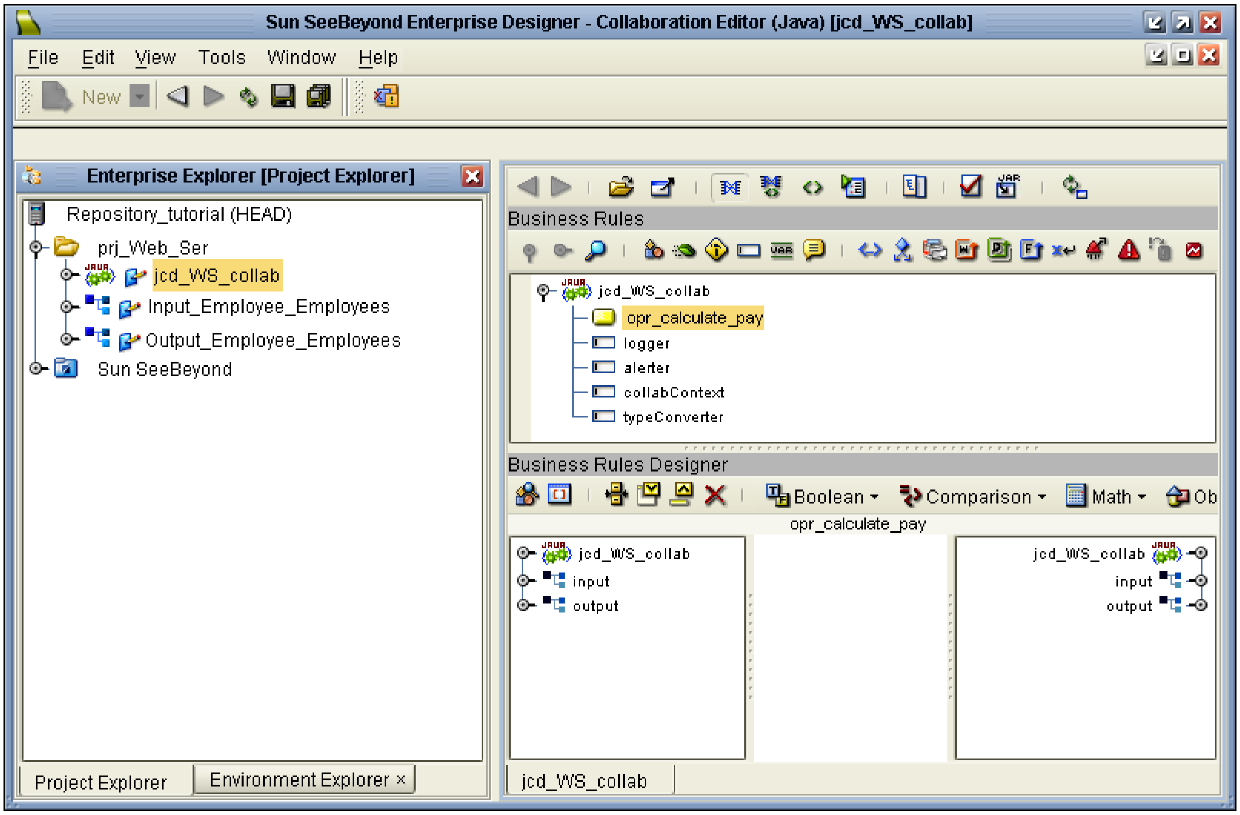
Type: Object Type Definition

Selected OTDs

OTD	Instance Name

A message appears on your screen, “**Loading jcd_WS_collab**.”

Figure 191 Collaboration Editor



6.6 Map Business Rules

Map Business Rules to create the logic that will process the data when the JCD is exposed as a web service. When this web service in invoked with an application client, such as eInsight, the data is processed.

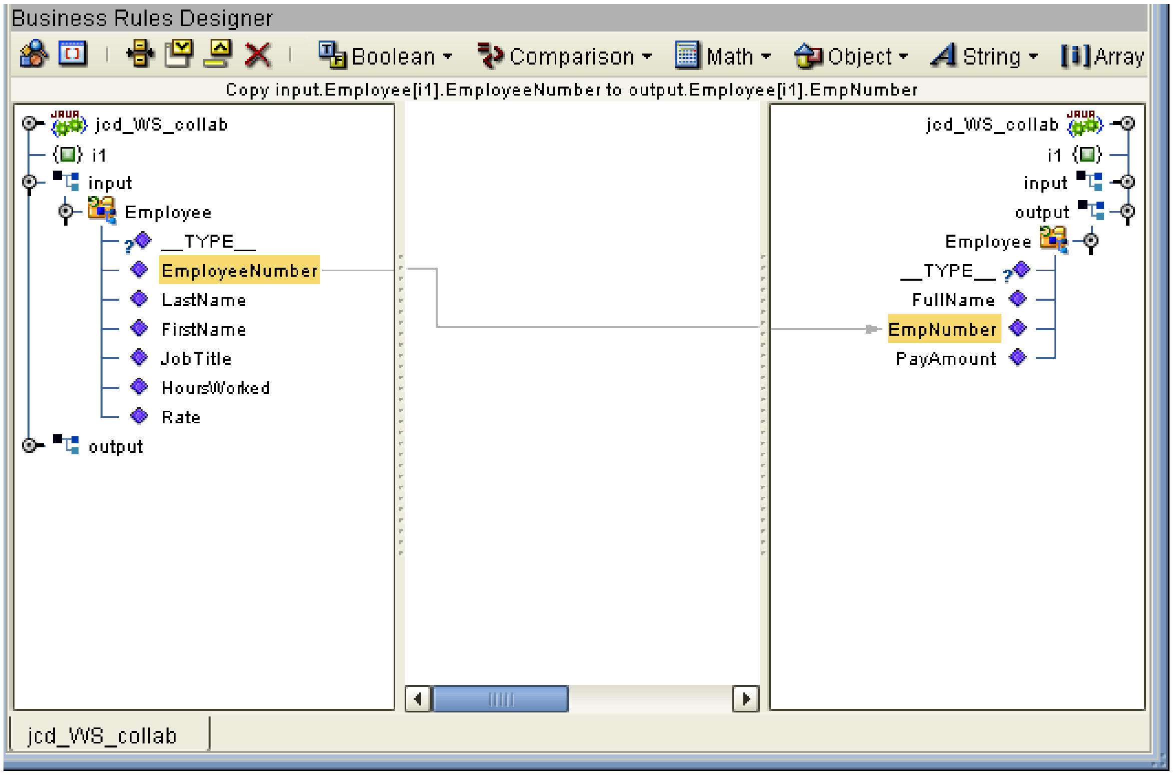
Map the Operation to Calculate Pay

In the Business Rules Designer pane, begin mapping rules.

1. Expand the nodes for **input** and **Employee** in the left pane.
2. Expand the nodes for **output** and **Employee** in the right pane.
3. Map Employee Number.

See the following Figure 192.

Figure 192 Map Employee Number

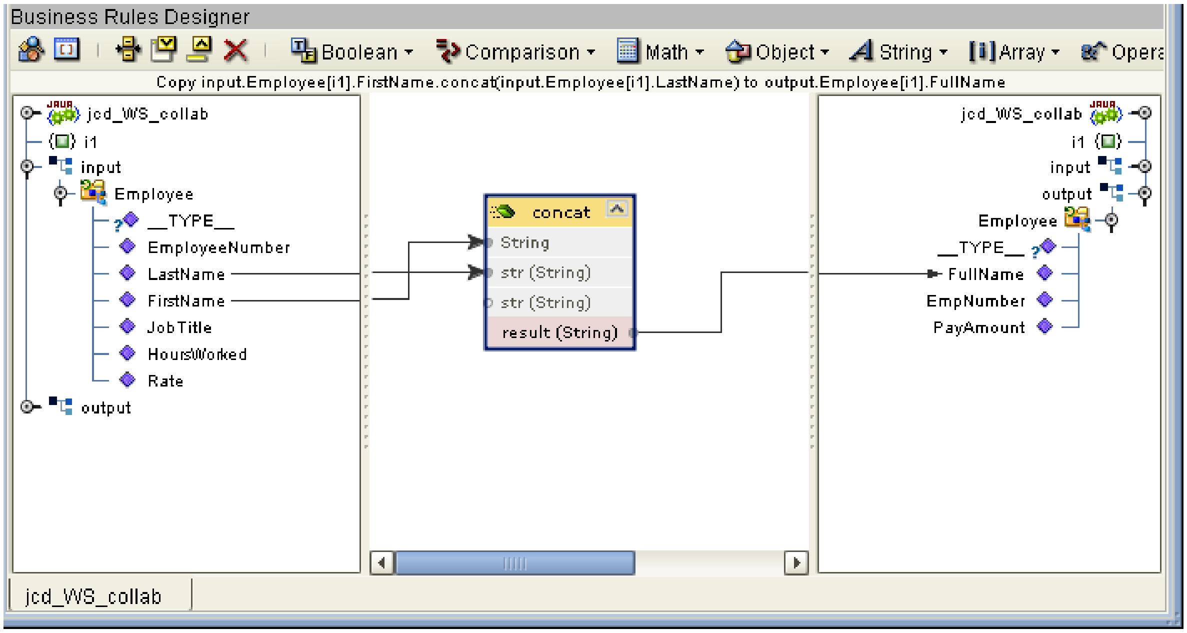


4 Click the **String** button, and then select the **Concat** method from the drop-down list.

The **String** methods appear in your Business Rules Designer tool bar.

5 Concatenate **FirstName** and **LastName** as shown in the following figure, then connect to **FullName** in the right pane.

Figure 193 Concat Method



Copy input.Employee[i1].FirstName.concat(input.Employee[i1].LastName) to output.Employee[i1].FullName

6 Click the **Math** button, and then select the **Multiplication** operation from the drop-down list.

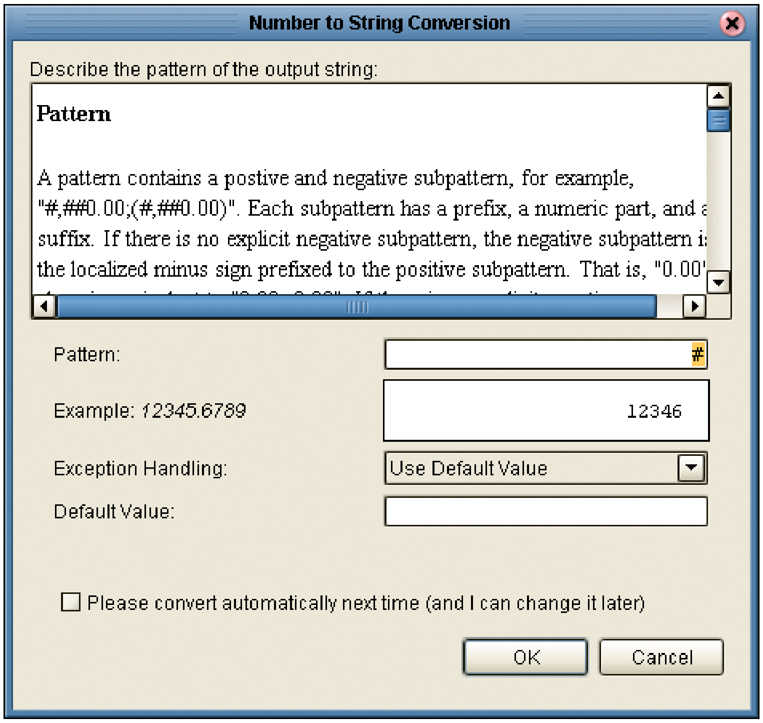
7 Connect **HoursWorked** to number 1 in the method box.

8 Connect **Rate** to number 2 in the method box.

9 Connect **result (num)** to **PayAmount**.

When you attempt to connect the **result (num)** to **PayAmount** in the right pane, the pop-up dialog box **Number to String Conversion** appears.

Figure 194 Number to String Conversion



Number to String Conversion

Describe the pattern of the output string:

Pattern

A pattern contains a postive and negative subpattern, for example, "#,##0.00;(#,##0.00)". Each subpattern has a prefix, a numeric part, and a suffix. If there is no explicit negative subpattern, the negative subpattern i the localized minus sign prefixed to the positive subpattern. That is, "0.00"

Pattern: #

Example: *12345.6789* 12346

Exception Handling: Use Default Value

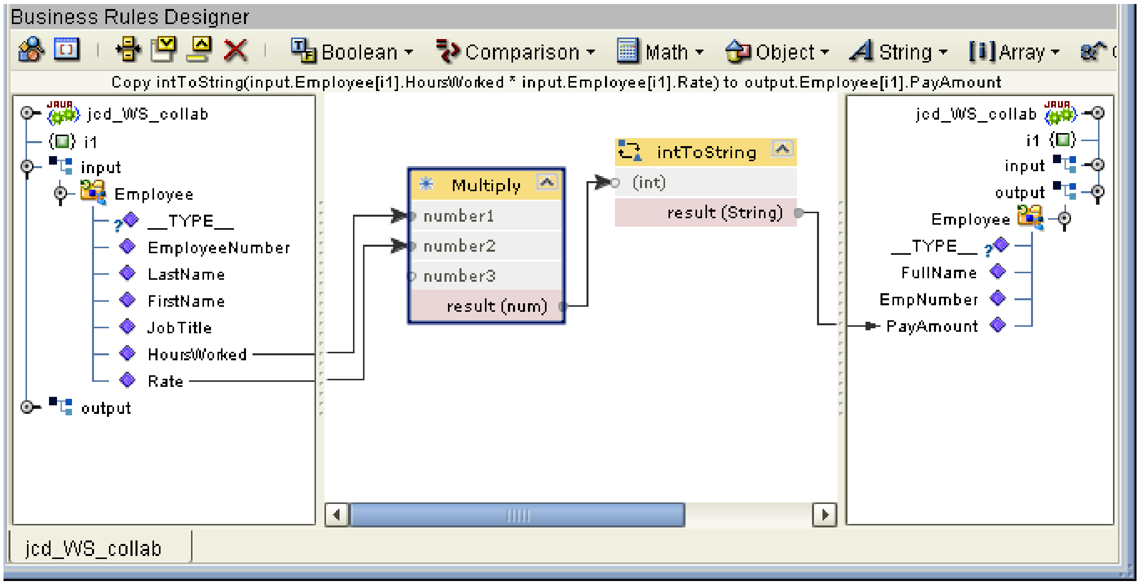
Default Value:

□ Please convert automatically next time (and I can change it later)

10 Click OK in the **Number to String Conversion** dialog box to accept the default, **intToString**.

The resulting mapped multiplication operation should appear as in the following figure.

Figure 195 Mapped Multiplication Rule



Business Rules Designer

Copy intToString(input.Employee[i1].HoursWorked * input.Employee[i1].Rate) to output.Employee[i1].PayAmount

11 Save all.

12 From the Windows menu, select **Close All**.

This closes all the open windows but does not close the Enterprise Designer.

6.7 Create a Connectivity Map

You have configured your Collaboration Definition and are now ready to link the objects using a Connectivity Map.

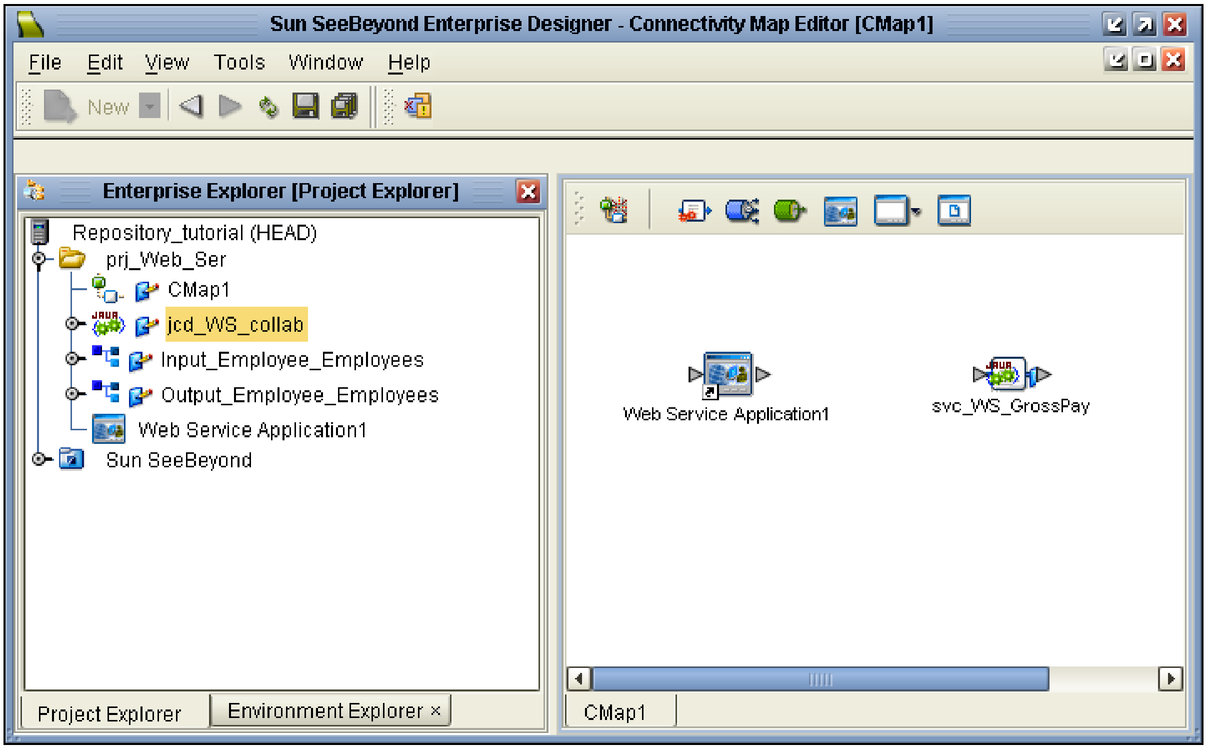
6.7.1 Apply the Collaboration

To make your JCD (Java Collaboration Definition) available as a web service, you first must bind the Collaboration to a Service in the Connectivity Map.

Link Objects in the Connectivity Map

1. Right-click your Project folder in the Project Explorer tree, then select **New**, **Connectivity Map**.
2. In the Enterprise Designer, drag the **Web Service External Application** icon to your canvas.
3. Drag the **Service** icon to your canvas.
4. Rename the Service, **svc_WS_GrossPay**, as shown in the following figure.
5. Press **Enter**.

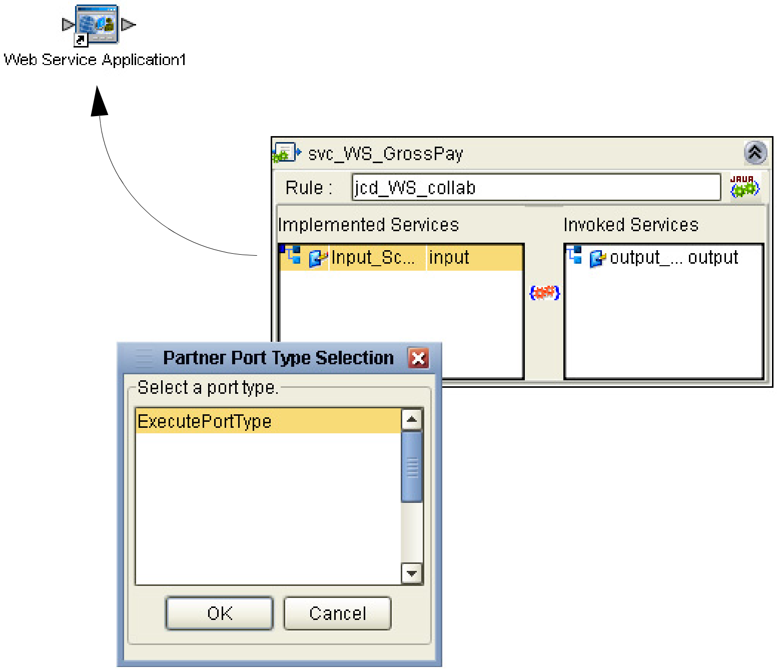
Figure 196 Connectivity Map



6 In the Project Explorer pane, click your Collaboration, **jcd_WS_collab**, then drag the icon into the Service, **svc_WS_GrossPay**.

7 Open the Service dialog box and connect the **input** to your web service.

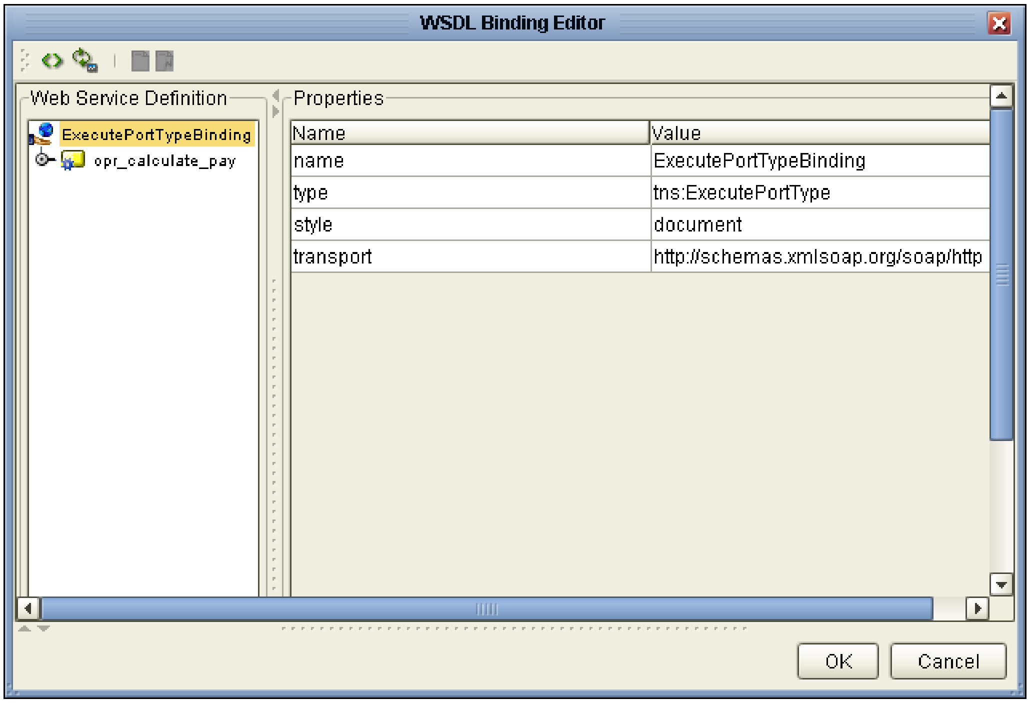
Figure 197 Partner Port type



8 Click **OK** in the Partner Port Type Selection dialog box.

The WSDL Binding Editor appears.

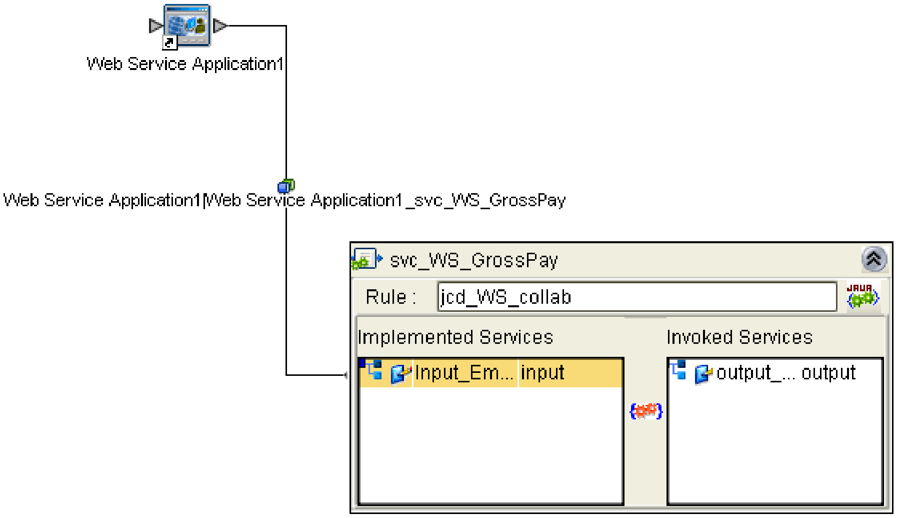
Figure 198 WSDL Binding Editor



Name	Value
name	ExecutePortTypeBinding
type	tns:ExecutePortType
style	document
transport	http://schemas.xmlsoap.org/soap/http

9 Click **OK**.

Figure 199 Map Connection



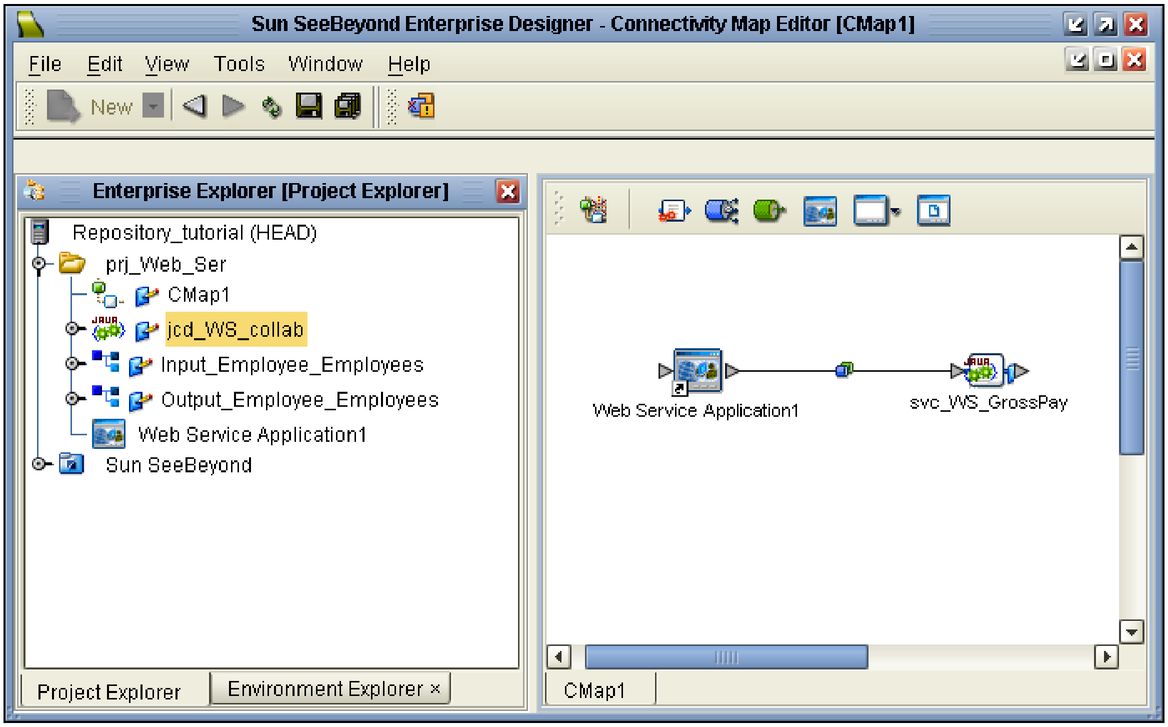
Rule : jcd_WS_collab

Your web service is mapped to the input.

10 Close the dialog box.

11 Save all.

Figure 200 Map Completed


Sun SeeBeyond Enterprise Designer - Connectivity Map Editor [CMap1]
File Edit View Tools Window Help
New
Enterprise Explorer [Project Explorer]
Repository_tutorial (HEAD)
prj_Web_Ser
CMap1
jcd_WS_collab
Input_Employee_Employees
Output_Employee_Employees
Web Service Application1
Sun SeeBeyond
Project Explorer
Environment Explorer
Web Service Application1
svc_WS_GrossPay
CMap1

12 Select **Close all** from the Window menu (but do not close the Enterprise Designer).

6.8 Create an Environment

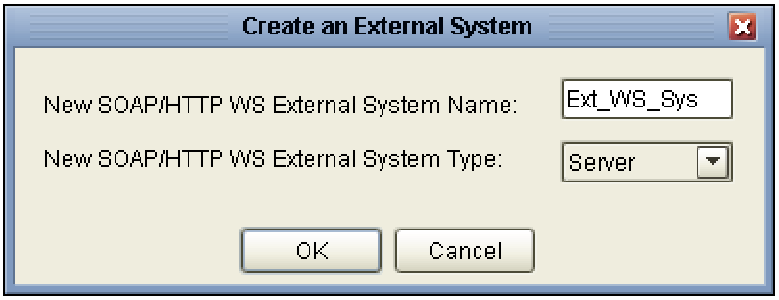
An Environment is a collection of physical resources and their configurations that are used to host Projects. An Environment contains logical hosts and external systems.

In this section create an Environment that includes web service components.

6.8.1 Add logical Host and Services

1. Click the **Environment Explorer** tab.
2. Right-click the **Repository** icon, and select **New Environment**.
3. Rename your Environment **WS_tutorial**. Click **Enter**.
4. Right-click your **WS_tutorial**, and select **New**, **Logical Host**.
5. Right-click your **Logical Host**, and select **New**, **SeeBeyond Integration Server**.
6. Right-click **WS_tutorial** and select New, **SOAP/HTTP Web Service External System**. Name your external system, **Ext_WS_Sys**.

Figure 201 SOAP External System



Create an External System

New SOAP/HTTP WS External System Name: Ext_WS_Sys

New SOAP/HTTP WS External System Type: Server

7 Accept the “Server” default and click **OK**.

Note: *You may have to move your Ext_WS_Sys GUI if it overlays the Logical Host.*

Configure the Properties

After creating the Web Service External System, configure the Properties.

1 Double-click and open Properties in **Ext_WS_Sys**.

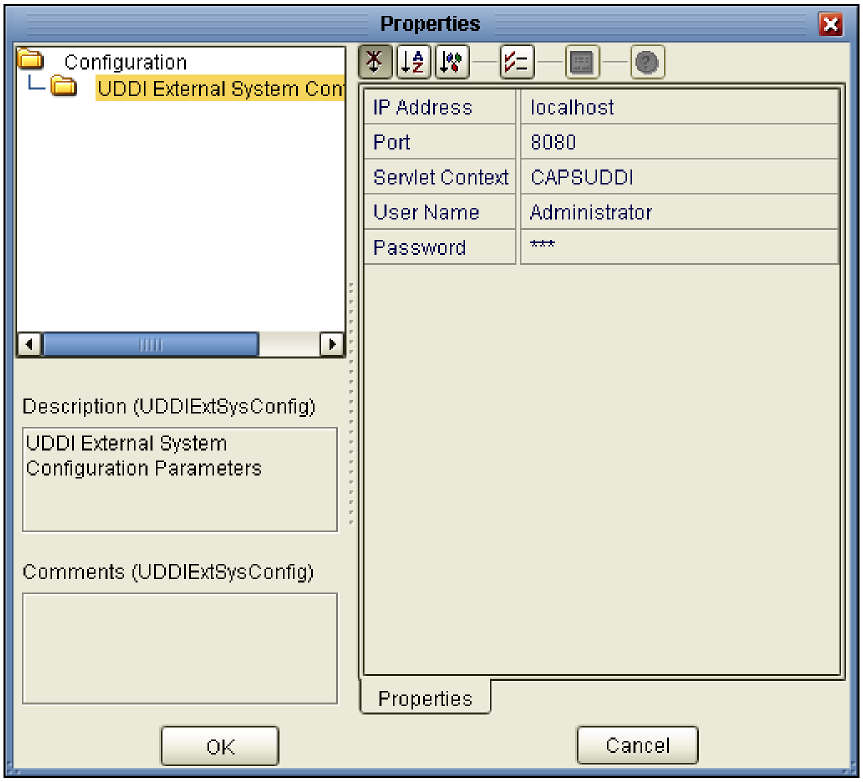
2 Configure as follows: **Port**=18001 (the HTTP admin port); **hostname**=Localhost; **Servlet Context**=WSserver (any string, such as WSserver).

3 Click **OK**.

4 Right-click **WS_tutorial** and select New, **UDDI External System**. Name your UDDI external system, **UDDI_Ext**.

5 Right-click **UDDI_Ext** and click Properties to see the default values.

Figure 202 UDDI_Ext Properties



Description (UDDIExtSysConfig)

UDDI External System Configuration Parameters

Comments (UDDIExtSysConfig)

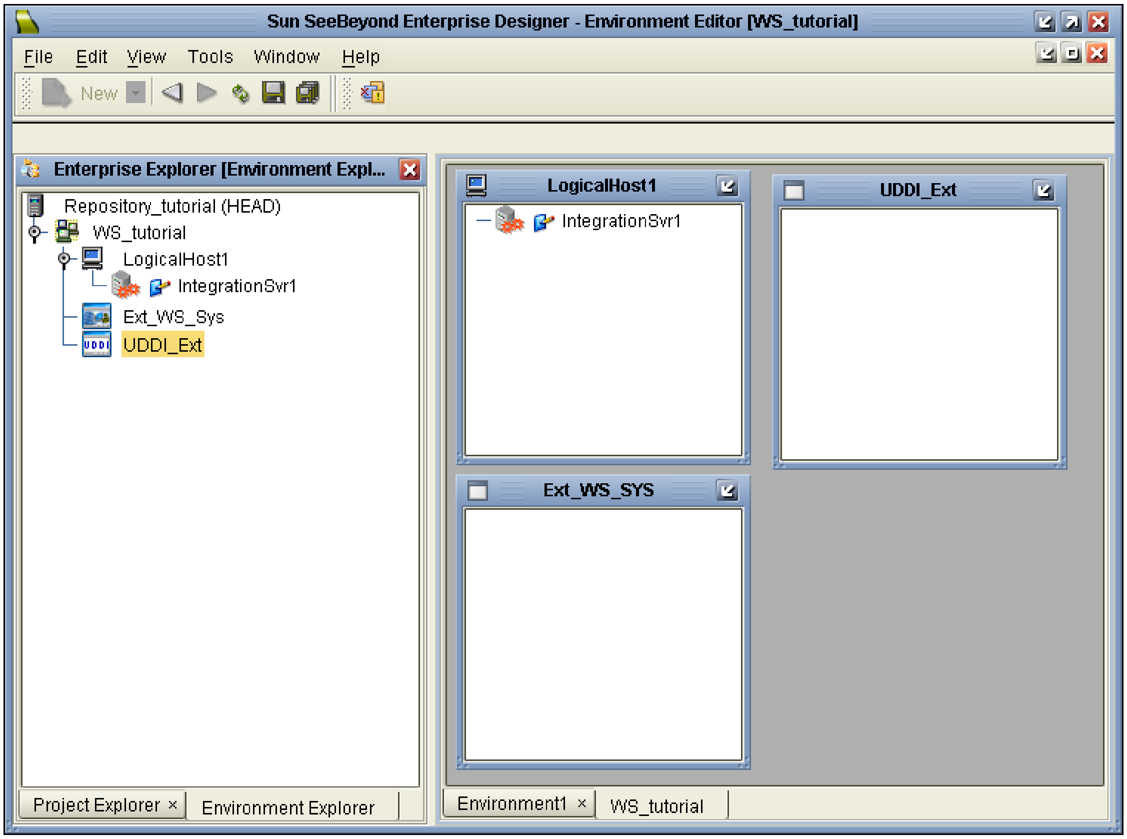
IP Address	localhost
Port	8080
Servlet Context	CAPSUDDI
User Name	Administrator
Password	***

6 Make sure you have entered your Password.

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

Note: *If a UDDI External System is not available at this time, then you can leave the properties blank.*

Figure 203 Web Service Environment

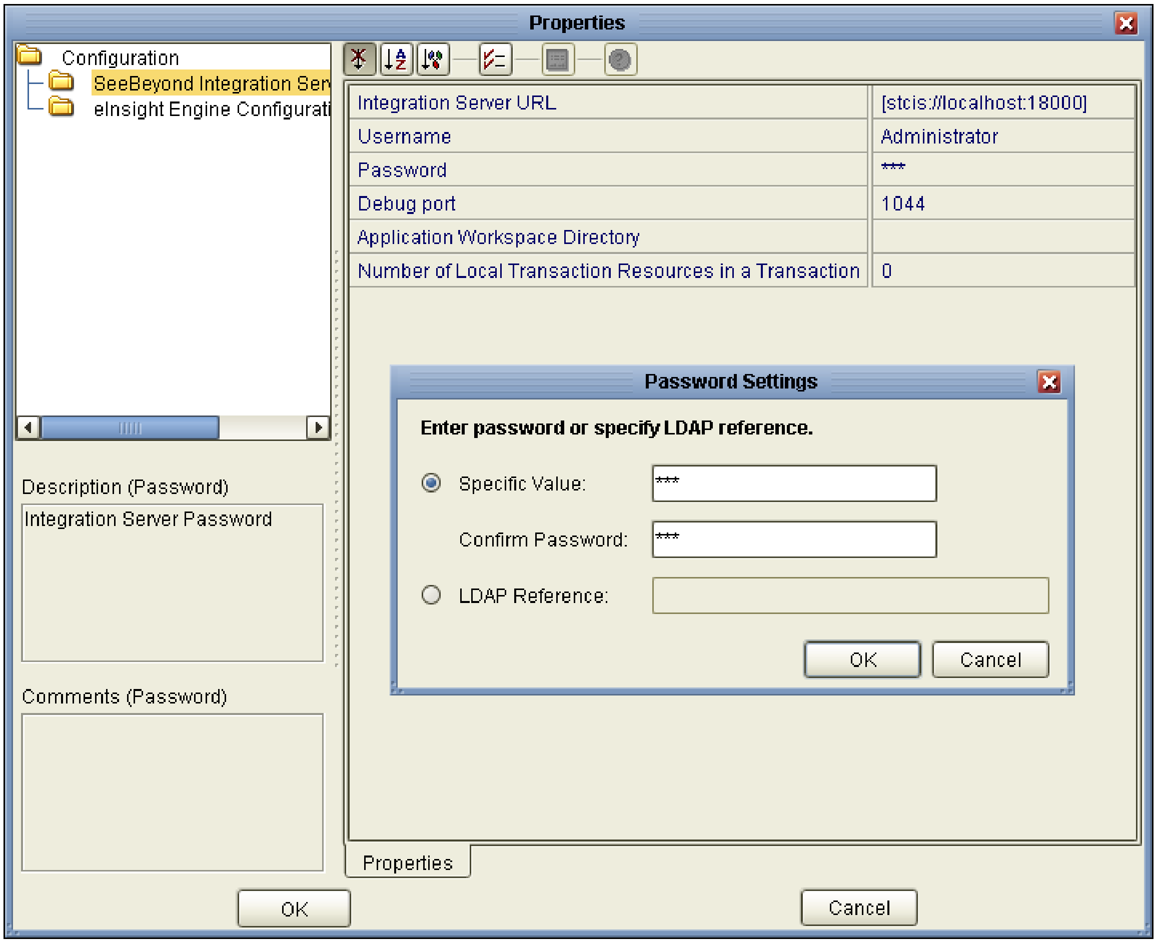


User Authentication

1 Right-click **IntegrationSvr1**, and select **Properties** from the pop-up list.

The Properties dialog box appears.

Figure 204 IS Properties



Integration Server URL	[stcis://localhost:18000]
Username	Administrator
Password	***
Debug port	1044
Application Workspace Directory	
Number of Local Transaction Resources in a Transaction	0

Description (Password)

Integration Server Password

Comments (Password)

Password Settings

Enter password or specify LDAP reference.

● Specific Value: ***

Confirm Password: ***

○ LDAP Reference:

2 Enter and confirm the password, and make sure the Integration Server URL is correct.

3 Save.

6.9 Create and Activate a Deployment Profile

A Deployment Profile contains information about how Project components are mapped and deployed within an Environment.

Create Deployment Profile

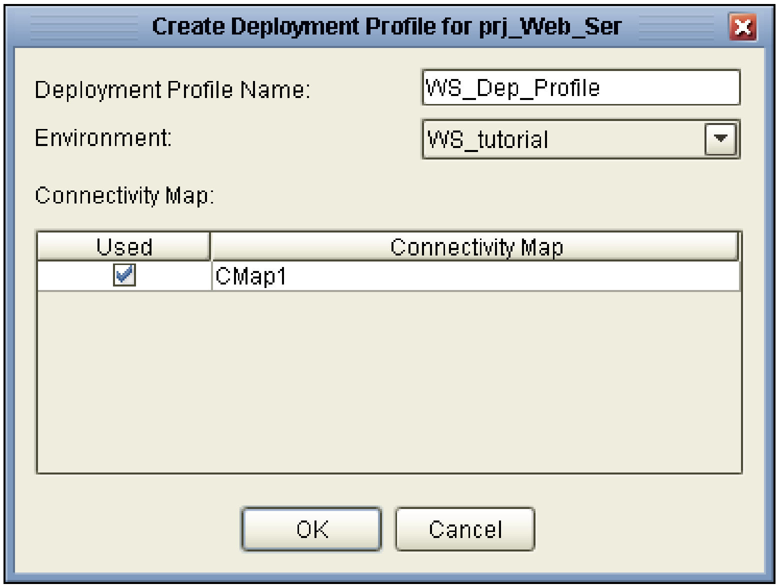
Return to your Project Explorer tree.

1 Click the **Project Explorer** tab.

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

3 Rename your Deployment Profile, **WS_Dep_Profile**.

Figure 205 Deployment Profile



Deployment Profile Name: WS_Dep_Profile

Environment: WS_tutorial

Connectivity Map:

Used	Connectivity Map
☑	CMap1

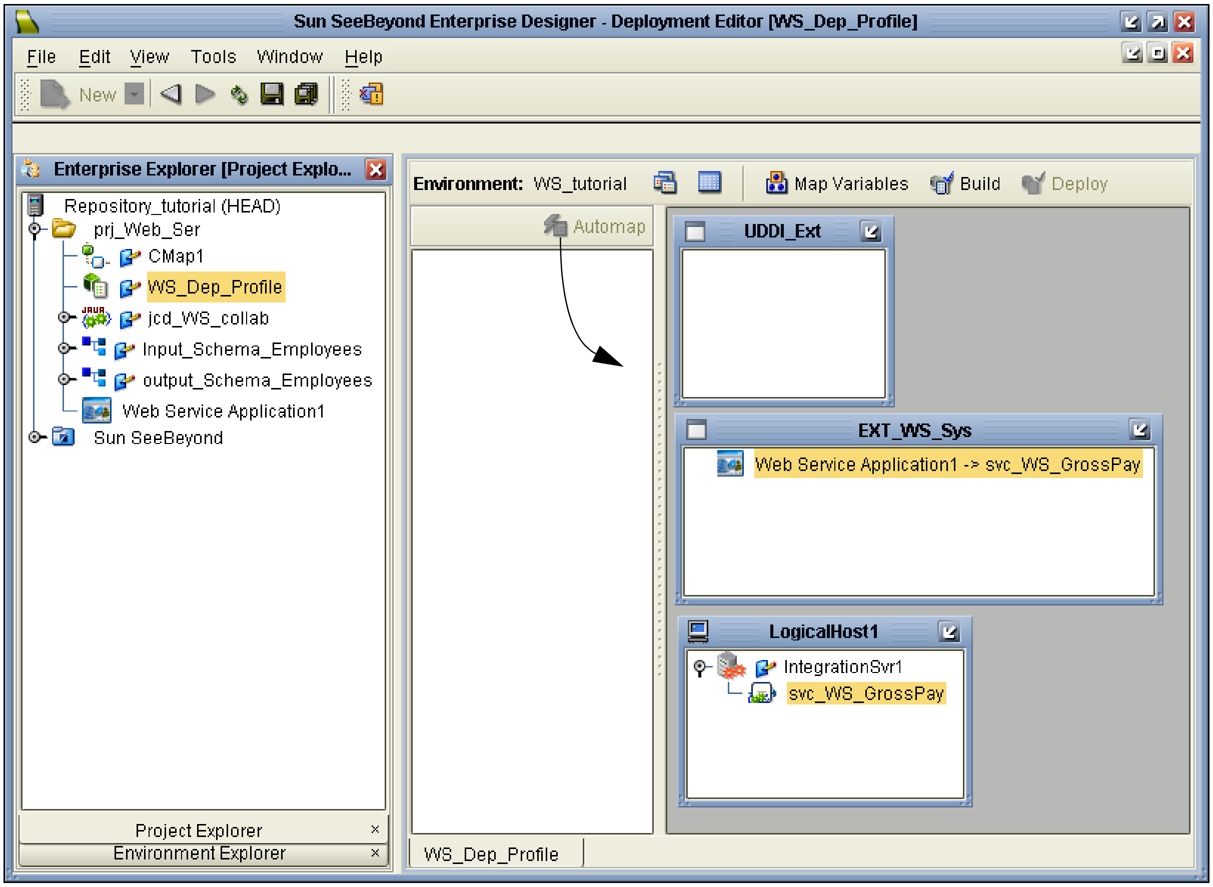
4 Click **OK**.

Your Environment, showing deployable components, appears.

5 Click the **Automap** icon.

Your components automatically map within your Environment.

Figure 206 Deployed Components

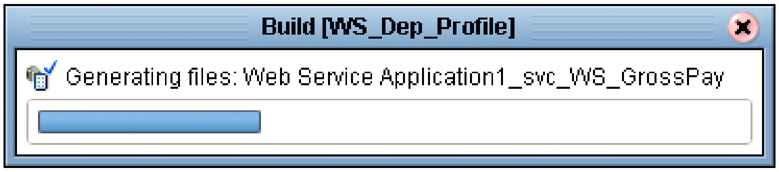


Build and Deploy your Project

1 Click the **Build** button.

The Build message appears.

Figure 207 Build Message



Generating files: Web Service Application1_svc_WS_GrossPay

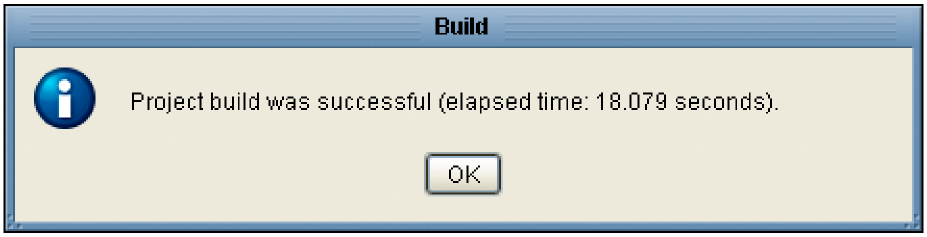
After a brief period of time the UDDI Registry Information dialog appears.

Note: *If the UDDI External System was not configured correctly, then export the WSDL(s) in a .jar file by selecting the option in the window below and proceeding with the File Browse step. If the UDDI External System is configured correctly and if the UDDI Server is running, then you can publish to the UDDI Registry directly.*

2 Accept the default in the dialog "Publish WSDL(s) to default UDDI Registry" and click **OK**.

If your Build is successful, you will see the following message.

Figure 208 Build Successful



Build

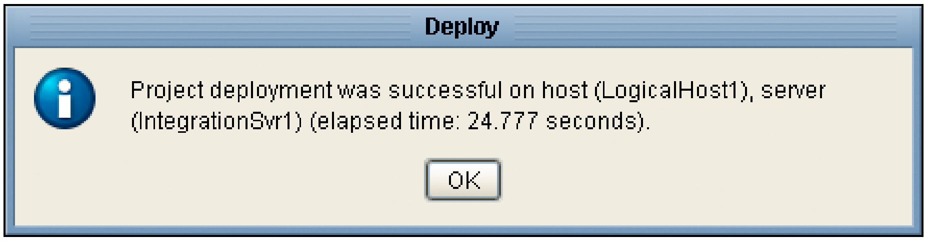
Project build was successful (elapsed time: 18.079 seconds).

3 Click the **Deploy** button, and click **Yes** when you see "Are you sure you want to deploy the current Project deployment?"

A Deployment message appears briefly.

If your Deployment is successful, you will see the following message.

Figure 209 Deploy Successful



Deploy

Project deployment was successful on host (LogicalHost1), server (IntegrationSvr1) (elapsed time: 24.777 seconds).

Your **WSDL** is published to **uddidocs** in your **UDDIServer** folder: `UDDIServer/uddidocs/WS_tutorial/prj_Web_ser/jcd_WS_collab`.

6.10 Sample Input and Output Data

Since you are only exposing a JCD as a web service in this scenario, you are not actually submitting the input from the **eGate** application. Also, the output is generated at run time when the web service is invoked, using a client application. You can use any client application(s) such as Sun's **eInsight** product.

Sample Input Data

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Sample XML file generated by XMLSpy v2005 sp1 U (http://
www.xmlspy.com)-->
<Employees xmlns="urn:please.define.your.own.target.namespace1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<Employee>
<EmployeeNumber>945</EmployeeNumber>
<LastName>Gallery</LastName>
```

```
<FirstName>Robert</FirstName>
<JobTitle>SysEngineer</JobTitle>
<HoursWorked>40</HoursWorked>
<Rate>95</Rate>
</Employee>
</Employees>
```

Sample Output Data

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Sample XML file generated by XMLSpy v2005 sp1 U (http://
www.xmlspy.com)-->
<Employees xmlns="urn:please.define.your.own.target.namespace1"
<Employee>
<EmpNumber>945</EmpNumber>
<FullName>RobertGallery</FullName>
<PayAmount>3800</PayAmount>
</Employee>
</Employees>
```

Chapter 7

Enterprise Manager Overview

For more detailed information about the Enterprise Manager see the *eGate Integrator System Administration Guide* and *Java Composite Application Platform Suite Installation Guide*. This chapter provides a quick overview only.

The Enterprise Manager is a powerful monitoring tool that is useful for troubleshooting and monitoring system activity.

Although the Enterprise Manager Monitor is not required for the tutorials, you may want to have it open so that you can become familiar with its functionality.

What's in This Chapter

7.1 Install and Run the Enterprise Manager

After you have downloaded the Enterprise Manager-Monitoring and Administration component, you can proceed to install and run the Enterprise Manager to monitor your Project in real time.

Install the Enterprise Manager

- Execute **install.bat** from your **emanager** subdirectory of your CAPS directory.

Start the Enterprise Manager

- Execute **startserver.bat** from your CAPS **emanager** folder.

Access the Enterprise Manager as a Web Service

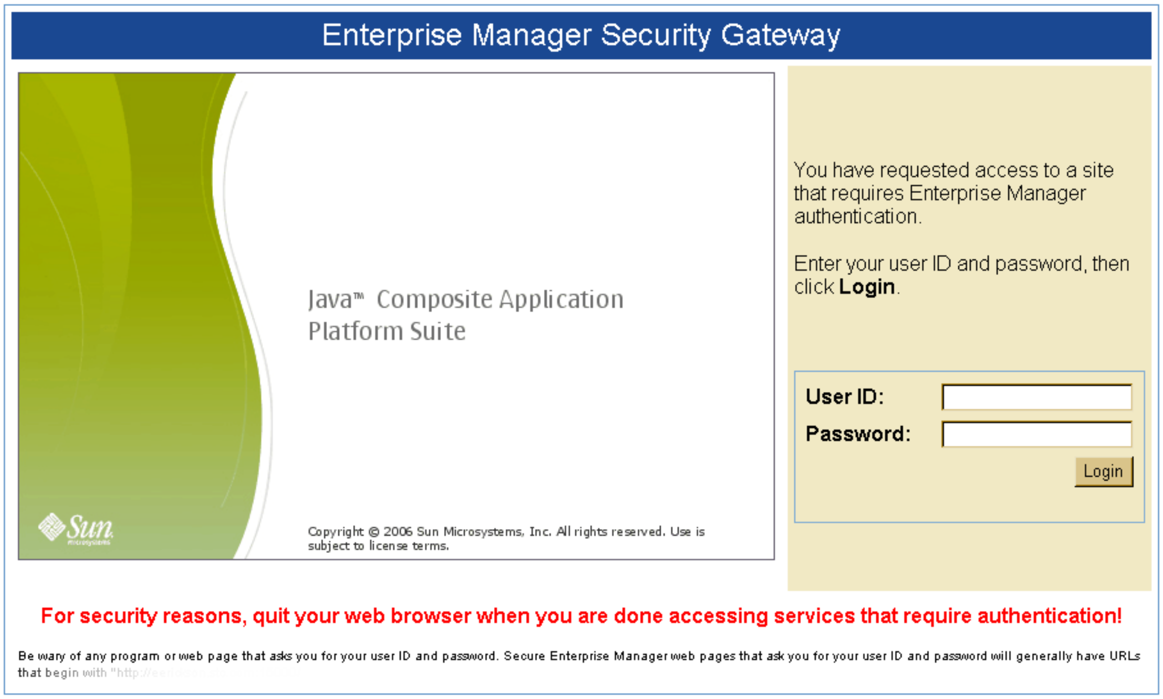
Use the URL **`http://<hostname>:<port_number>`**

where **<hostname>** is the fully qualified (although “localhost” is used in some of the examples), network addressable host name

and **<port_number>** is the Enterprise Manager port number

For example: **`http://johndoe.stc.com:15000`** or **`http://localhost:15000`**.

Figure 210 Enterprise Manager Security Gateway



Enterprise Manager Security Gateway

Java™ Composite Application Platform Suite

Copyright © 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

You have requested access to a site that requires Enterprise Manager authentication.

Enter your user ID and password, then click **Login**.

User ID:

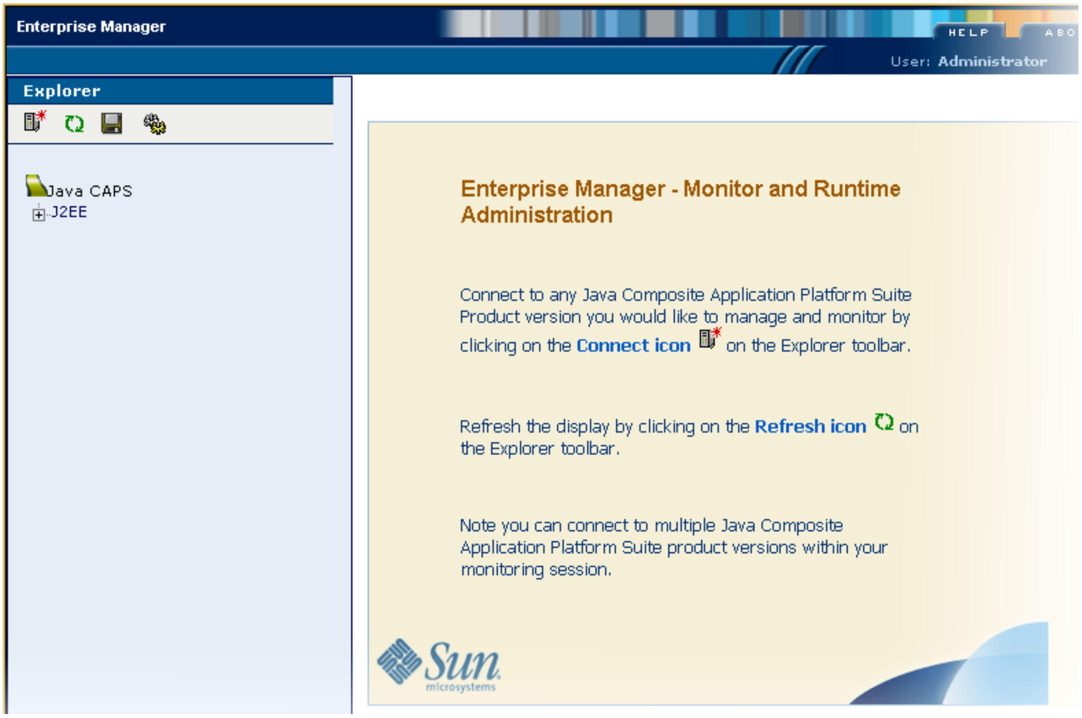
Password:

For security reasons, quit your web browser when you are done accessing services that require authentication!

Be wary of any program or web page that asks you for your user ID and password. Secure Enterprise Manager web pages that ask you for your user ID and password will generally have URLs that begin with "http://[illegible]

- Enter your User ID and Password.

Figure 211 Enterprise Manager Monitor and Runtime Administration



Enterprise Manager - Monitor and Runtime Administration

Connect to any Java Composite Application Platform Suite Product version you would like to manage and monitor by clicking on the **Connect icon** on the Explorer toolbar.

Refresh the display by clicking on the **Refresh icon** on the Explorer toolbar.

Note you can connect to multiple Java Composite Application Platform Suite product versions within your monitoring session.

Add a Runtime Server

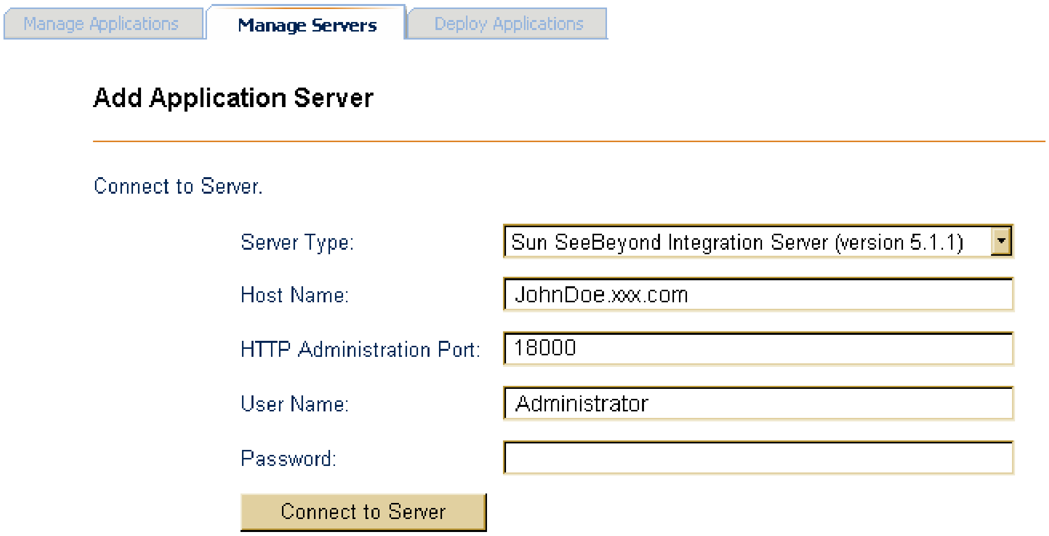
Add a Runtime Server. For the example shown below, make sure your domain\integration server is runnig.

1 Expand the nodes under J2EE and select **Deployer** in the Explorer Tree in the left pane.

2 Click the **Manage Servers** tab.

3 Click **Add New Server** and enter the required information. See the following figure.

Figure 212 Add Runtime Server



Add Application Server

Connect to Server.

Server Type:	Sun SeeBeyond Integration Server (version 5.1.1)
Host Name:	JohnDoe.xxx.com
HTTP Administration Port:	18000
User Name:	Administrator
Password:	

7.2 Install the eWay Monitor and File eWay Monitor

Ensure that both the **File eWay Monitor** and the **eWay Monitor** are installed on the Enterprise Manager server.

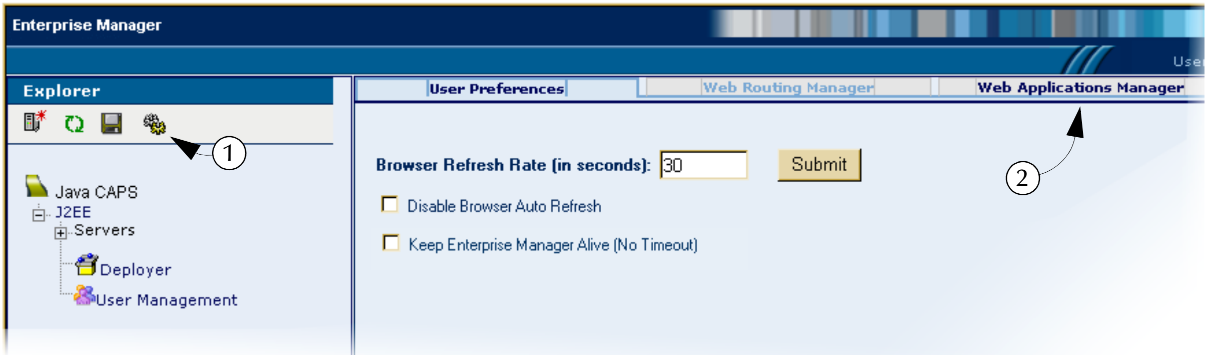
Use the Enterprise Manager to install these applications.

1 Click the **Configuration** Icon on the Explorer tool bar in the Enterprise Manager.

2 Click the **Web Application Manager** tab, then select the **Auto-Install From Repository** tab.

See the following figure.

Figure 213 Web Applications Manager



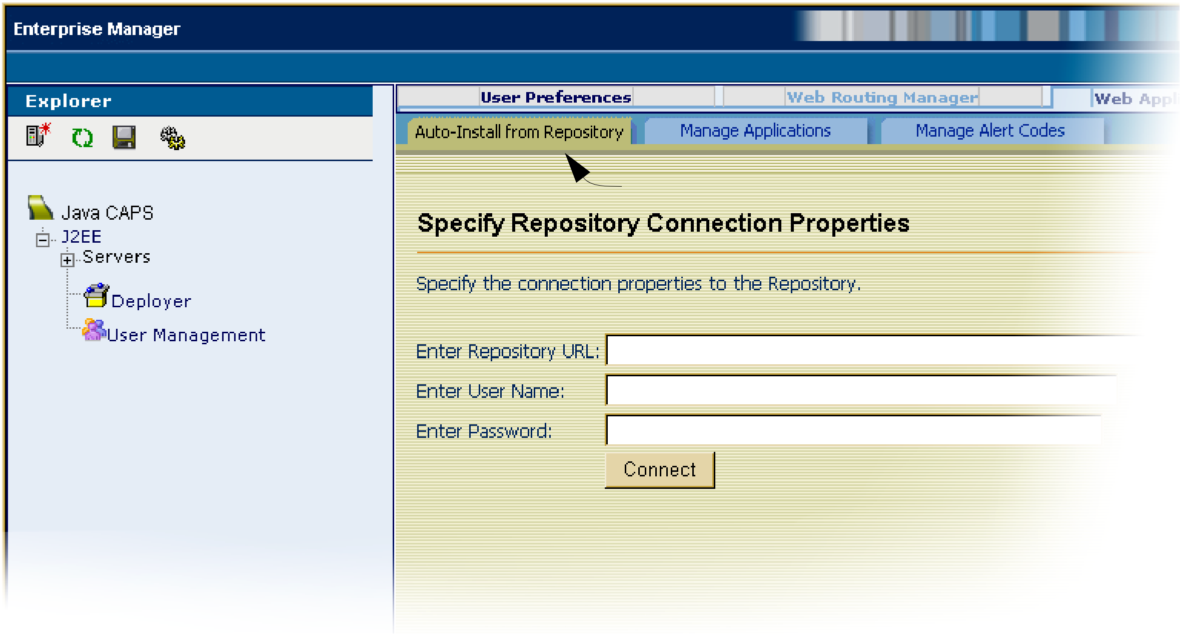
Browser Refresh Rate (in seconds): 30

□ Disable Browser Auto Refresh

□ Keep Enterprise Manager Alive (No Timeout)

3 Click the **Auto_Install** tab.

Figure 214 Auto-install eWay Monitors



Specify Repository Connection Properties

Specify the connection properties to the Repository.

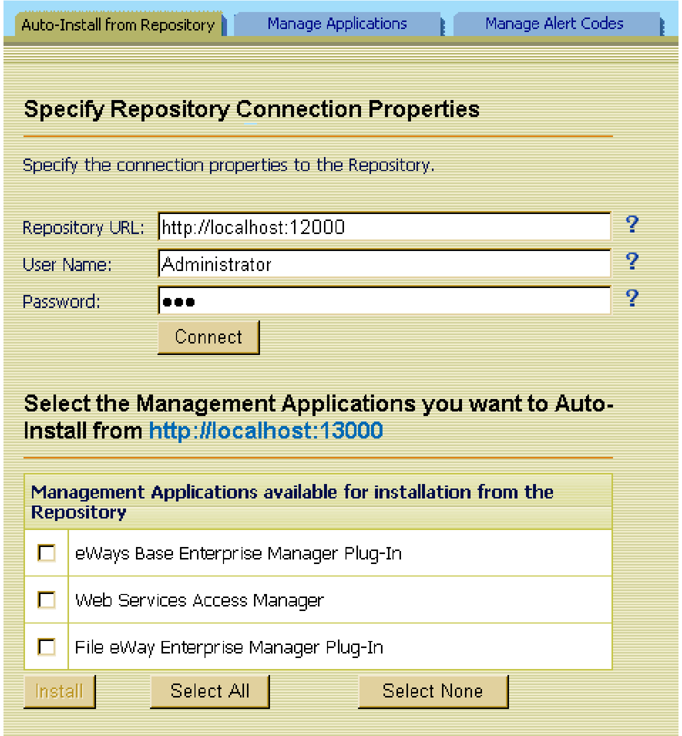
Enter Repository URL:

Enter User Name:

Enter Password:

4 Enter the required information in the Specify Repository Connection Properties fields, as shown in the following figure.

Figure 215 Applications Available for Installation



Specify Repository Connection Properties

Specify the connection properties to the Repository.

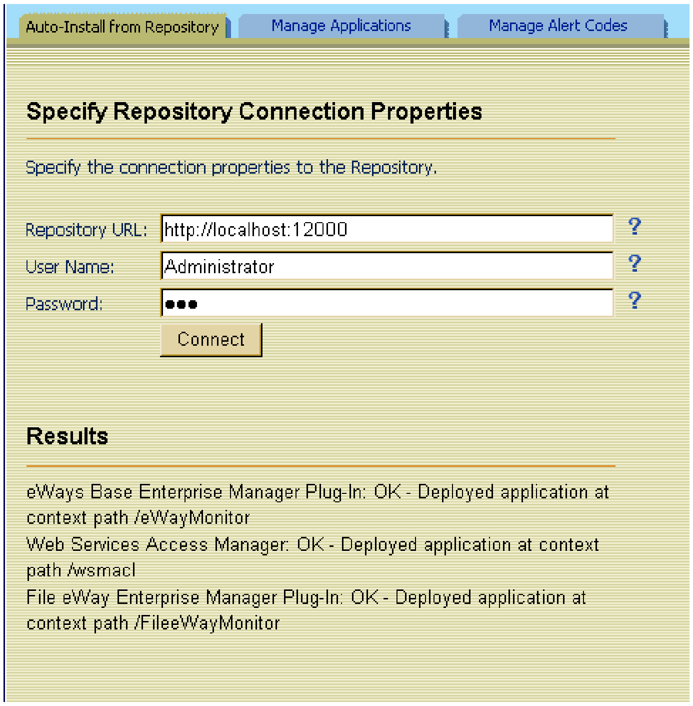
Repository URL:	http://localhost:12000
User Name:	Administrator
Password:	●●●

Select the Management Applications you want to Auto-Install from http://localhost:13000

	Management Applications available for installation from the Repository
□	eWays Base Enterprise Manager Plug-In
□	Web Services Access Manager
□	File eWay Enterprise Manager Plug-In

5 Check the applications you want to install as shown in the previous figure. It may be best to check all the components listed.

Figure 216 View Results of Install



Specify Repository Connection Properties

Specify the connection properties to the Repository.

Repository URL:	http://localhost:12000
User Name:	Administrator
Password:	●●●

Results

eWays Base Enterprise Manager Plug-In: OK - Deployed application at context path /eWayMonitor
Web Services Access Manager: OK - Deployed application at context path /wsmacl
File eWay Enterprise Manager Plug-In: OK - Deployed application at context path /FileeWayMonitor

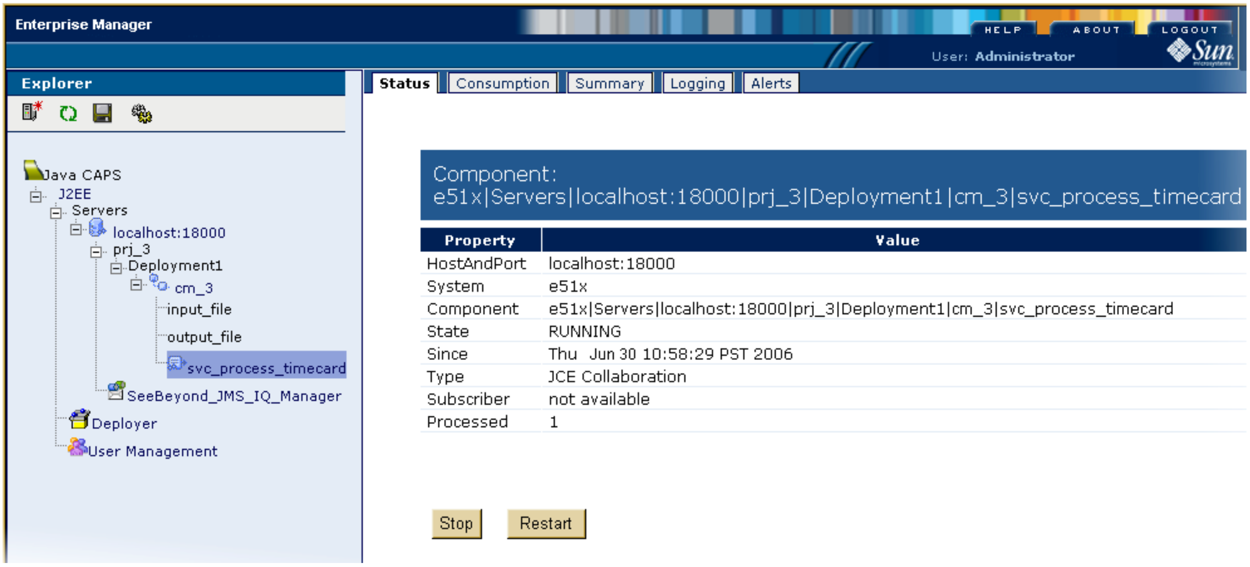
7.3 Monitor your Project

The examples used in this section were taken from Scenario3 in the eGate Tutorial.

Service 1

Click the components in the Explorer Tree to monitor your project. You can start or stop a service. In the figure below, the service is up and running.

Figure 217 Monitor Service



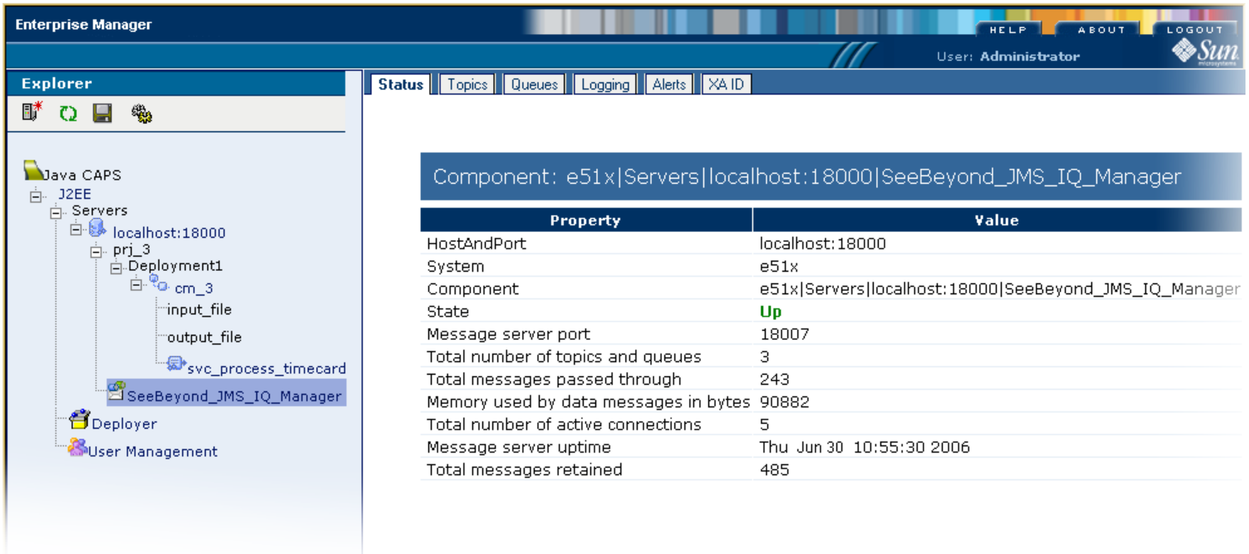
Component:
e51x|Servers|localhost:18000|prj_3|Deployment1|cm_3|svc_process_timecard

Property	Value
HostAndPort	localhost:18000
System	e51x
Component	e51x|Servers|localhost:18000|prj_3|Deployment1|cm_3|svc_process_timecard
State	RUNNING
Since	Thu Jun 30 10:58:29 PST 2006
Type	JCE Collaboration
Subscriber	not available
Processed	1

JMS IQ Manager

Monitor message server information.

Figure 218 JMS IQ Manager



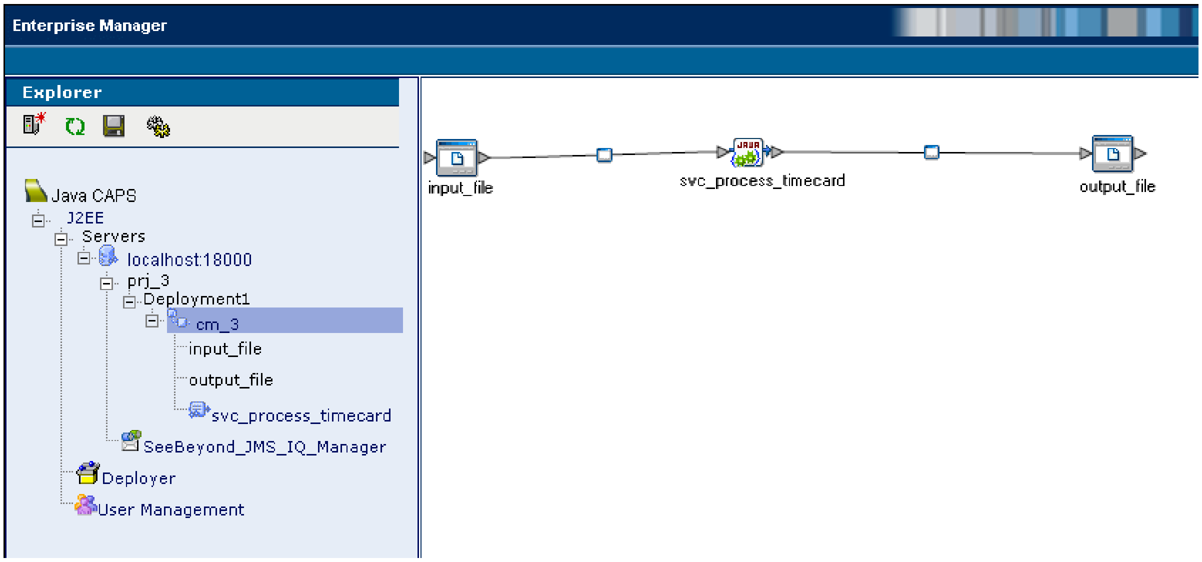
Component: e51x|Servers|localhost:18000|SeeBeyond_JMS_IQ_Manager

Property	Value
HostAndPort	localhost:18000
System	e51x
Component	e51x|Servers|localhost:18000|SeeBeyond_JMS_IQ_Manager
State	Up
Message server port	18007
Total number of topics and queues	3
Total messages passed through	243
Memory used by data messages in bytes	90882
Total number of active connections	5
Message server uptime	Thu Jun 30 10:55:30 2006
Total messages retained	485

Connectivity Map

Display the Connectivity Map GUI. You can start, stop, and monitor services or components (such as the File eWay). Make sure you have installed the eWay Monitor (plug-ins). See Figure 215.

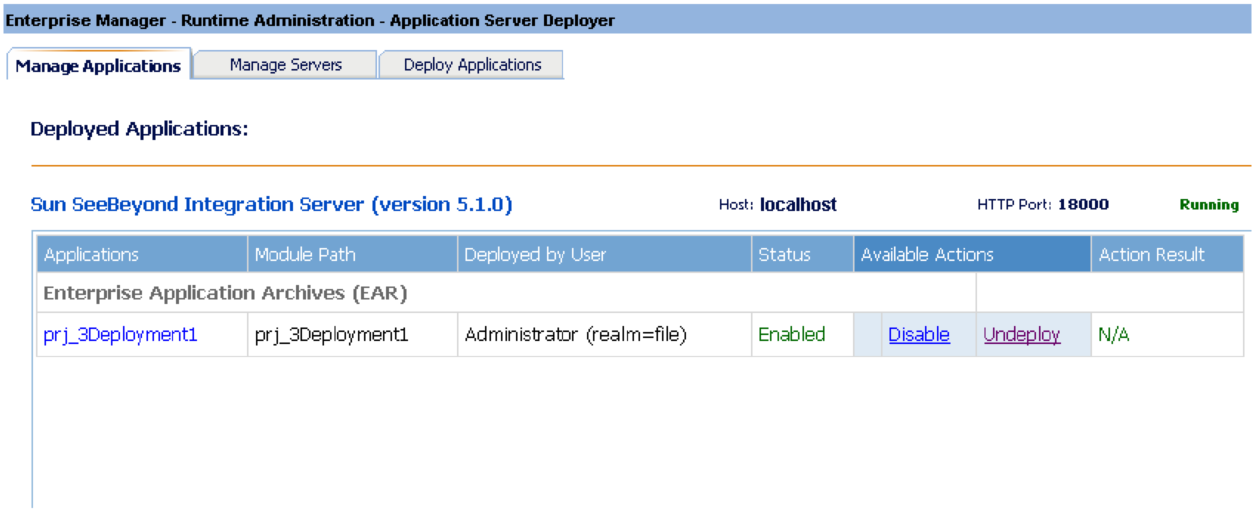
Figure 219 Connectivity Map



Deployed Applications

You can start, stop, or undeploy your applications (Projects). The following figure shows your Project3 deployment running in the Integration Server (in domain1). This application is deployed and enabled. (Click the **Deployer** icon.)

Figure 220 Deployed Applications


Enterprise Manager - Runtime Administration - Application Server Deployer
Manage Applications
Manage Servers
Deploy Applications

Deployed Applications:

Sun SeeBeyond Integration Server (version 5.1.0) Host: localhost HTTP Port: 18000 Running

Applications	Module Path	Deployed by User	Status	Available Actions		Action Result
Enterprise Application Archives (EAR)						
prj_3Deployment1	prj_3Deployment1	Administrator (realm=file)	Enabled	Disable	Undeploy	N/A

Glossary

CAPS

Composite Application Platform Suite.

Collaboration

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

Also see **"Service"** and **"Collaboration Definition"**.

Collaboration Definition

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

Connection

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

Connectivity Map

Contains business logic and routing information about the data transmission. A Connectivity Map usually includes one or more Collaborations, topics, queues, and eWays. A Connectivity Map is created under a Project. A Project may have multiple Connectivity Maps.

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.

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.

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.

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.

Topic

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

XSLT

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

e*Gate 4.x Terms in eGate 5.0

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

Table 1 eGate 5.0 Terms

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

Table 1 eGate 5.0 Terms (Continued)

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

Index