

## **Oracle® Fusion Middleware Application Adapters**

Application Adapter for PeopleSoft User's Guide for

12c (12.2.1.2.0)

**E84212-01**

December 2016

Provides information on how to integrate with PeopleSoft systems and develop applications.

Oracle Fusion Middleware Application Adapter 12c (12.2.1.2.0) for PeopleSoft User's Guide for Oracle WebLogic Server, 12c (12.2.1.2.0)

E84212-01

Copyright © 2001, 2016, Oracle and/or its affiliates. All rights reserved.

Primary Author: Stefan Kostial

Contributors: Vikas Anand, Marian Jones, Sunil Gopal, Bo Stern

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

---

---

# Contents

<b>Preface</b> .....	xi
Audience .....	xi
Documentation Accessibility .....	xi
Related Documents .....	xi
Conventions .....	xii
<b>1 Introduction</b>	
<b>Adapter Features</b> .....	1-1
<b>PeopleSoft Concepts</b> .....	1-2
PeopleSoft Component Interface .....	1-2
PeopleSoft Application Messaging / Integration Broker .....	1-3
<b>Integration with PeopleSoft</b> .....	1-3
<b>Adapter Architecture</b> .....	1-4
Oracle WebLogic Server Adapter Business Services Engine (BSE) Architecture .....	1-4
Oracle WebLogic Server Adapter J2CA Architecture .....	1-5
<b>BSE Versus Oracle Adapter J2CA Deployment</b> .....	1-6
<b>Sample Projects</b> .....	1-6
<b>Quick Start Guide</b> .....	1-8
Installation .....	1-8
Copying Third-Party Library Files .....	1-8
Configuration .....	1-9
WebLogic Server Deployment and Integration .....	1-10
Creating Configurations, Targets, and Channels in Application Explorer .....	1-10
Working With Service Components in the SOA Suite .....	1-10
Working With Oracle Service Bus .....	1-11
Other Features .....	1-11
<b>2 Configuring Oracle Application Adapter for PeopleSoft</b>	
<b>Starting Application Explorer</b> .....	2-1
<b>Configuring Repository Settings</b> .....	2-2
<b>Creating a Repository Configuration</b> .....	2-2
Creating a Configuration for BSE .....	2-2
Creating a Configuration for J2CA .....	2-3
Connecting to a BSE or J2CA Configuration .....	2-4
<b>Establishing a Connection (Target) for PeopleSoft</b> .....	2-5

Defining a Target to PeopleSoft .....	2-5
Connecting to a Defined PeopleSoft Target .....	2-7
Managing a Target .....	2-7
Disconnecting from a Connection to PeopleSoft .....	2-8
Modifying Connection Parameters .....	2-8
Deleting a Connection to PeopleSoft.....	2-9
<b>Viewing Application System Objects .....</b>	<b>2-9</b>
<b>Creating XML Schemas .....</b>	<b>2-9</b>
Creating XML Request and Response Schemas Against BSE.....	2-10
Creating XML Request and Response Schemas Against the Oracle Adapter J2CA .....	2-10
<b>Generating a WSDL (J2CA Configurations Only).....</b>	<b>2-10</b>
<b>Creating and Testing a Web Service (BSE Configurations Only) .....</b>	<b>2-12</b>
Creating a Web Service .....	2-12
Testing a Web Service.....	2-13
Identity Propagation.....	2-13
<b>Configuring an Event Adapter .....</b>	<b>2-14</b>
Creating and Editing a Channel.....	2-14
Creating an HTTP Channel .....	2-15
Creating a TCP Channel.....	2-17
Creating a File Channel.....	2-20
Editing a Channel.....	2-22
Deleting a Channel .....	2-22
Schema Validation .....	2-22

### 3 Oracle WebLogic Server Deployment and Integration

Adapter Integration with Oracle WebLogic Server .....	3-1
Deployment of Adapter .....	3-1
Updating Adapter Configuration.....	3-2
Creating a Managed Connector Factory Object.....	3-2
Creating Multiple Managed Connector Factory Objects.....	3-4
Modifying WSDL Files for Additional Connection Factory Values.....	3-6

### 4 Integration With BPEL Service Components in the Oracle SOA Suite

Overview .....	4-1
Deployment of Adapter .....	4-2
Configuring a New Application Server Connection.....	4-2
Designing an Outbound BPEL Process for Service Integration (J2CA Configuration) .....	4-7
Generating WSDL for Request/Response Service .....	4-8
Creating an Empty Composite for SOA .....	4-9
Defining a BPEL Outbound Process.....	4-11
Configuring a Third Party Adapter Service Component.....	4-11
Configuring an Outbound BPEL Process Component.....	4-16
Adjusting for Known Deployment Issues With 12c .....	4-26
Deploying the BPEL Outbound Process .....	4-28
Invoking the Input XML Document in the Oracle Enterprise Manager Console.....	4-31
Testing Outbound BPEL and Mediator Processes .....	4-33
Designing an Inbound BPEL Process for Event Integration (J2CA Configuration) .....	4-34

Generating WSDL for Event Integration .....	4-34
Creating a Channel .....	4-35
Generating WSDL for Event Notification .....	4-36
Creating an Empty Composite for SOA .....	4-38
Defining a BPEL Inbound Process .....	4-38
Creating a Third Party Adapter Service Component .....	4-38
Creating an Inbound BPEL Process Component.....	4-42
Adjusting for Known Deployment Issues With 12c .....	4-44
Deploying the BPEL Inbound Process .....	4-44
Triggering an Event in PeopleSoft.....	4-45
Triggering an Event in PeopleSoft.....	4-45
Verifying the Results .....	4-51
<b>Designing an Outbound BPEL Process for Service Integration (BSE Configuration) .....</b>	<b>4-52</b>
Generating a WSDL File for Request and Response Services Using a Web Service .....	4-53
Creating an Empty Composite for SOA .....	4-53
Defining a BPEL Outbound Process.....	4-54
Creating a Partner Link.....	4-57
Creating BPEL Activities and Mappings With the Created Partner Link .....	4-60
<b>5 Integration With Mediator Service Components in the Oracle SOA Suite</b>	
<b>Configuring a New Application Server Connection.....</b>	<b>5-2</b>
<b>Configuring a Mediator Outbound Process (J2CA Configuration) .....</b>	<b>5-2</b>
Creating an Empty Composite for SOA .....	5-2
Defining a Mediator Outbound Process .....	5-3
Configuring a Third Party Adapter Service Component.....	5-3
Configuring an Outbound Mediator Process Component .....	5-4
Configuring the Routing Rules .....	5-8
Adjusting for Known Deployment Issues With 12c .....	5-11
Deploying the Mediator Outbound Process .....	5-11
Invoking the Input XML Document in the Oracle Enterprise Manager Console.....	5-11
<b>Configuring a Mediator Inbound Process (J2CA Configuration).....</b>	<b>5-11</b>
Creating an Empty Composite for SOA .....	5-12
Defining a Mediator Inbound Process .....	5-12
Configuring a Third Party Adapter Service Component.....	5-12
Configuring an Inbound Mediator Process Component With a File Adapter .....	5-13
Configuring the Routing Rules .....	5-17
Adjusting for Known Deployment Issues With 12c .....	5-18
<b>Configuring a Mediator Outbound Process (BSE Configuration) .....</b>	<b>5-18</b>
Creating an Empty Composite for SOA .....	5-19
Defining a Mediator Outbound Process .....	5-19
Configuring a SOAP Service .....	5-19
Creating a Mediator Component.....	5-19
Configuring the Routing Rules.....	5-22

## 6 Integration With BPM Service Components in the Oracle SOA Suite

Overview.....	6-1
---------------	-----

<b>Deployment of Adapter .....</b>	<b>6-1</b>
<b>Configuring a New Application Server Connection.....</b>	<b>6-2</b>
<b>Designing an Outbound BPM Process Using Transformations for Service Integration (J2CA Configuration).....</b>	<b>6-2</b>
Creating an Empty Composite for BPM .....	6-2
Defining a BPM Outbound Process.....	6-3
Configuring a Third Party Adapter Service Component.....	6-3
Configuring an Outbound BPM Process Component.....	6-4
Creating a File Adapter for the Write Operation .....	6-16
Adjusting for Known Deployment Issues With 12c .....	6-26
Deploying the BPM Outbound Process .....	6-26
Invoking the Input XML Document in the Oracle Enterprise Manager Console.....	6-27
<b>Designing an Inbound BPM Process Using Transformations for Event Integration (J2CA Configuration).....</b>	<b>6-28</b>
Creating an Empty Composite for BPM .....	6-28
Defining a BPM Inbound Process.....	6-29
Configuring a Third Party Adapter Service Component.....	6-29
Configuring an Inbound BPM Process Component .....	6-30
Creating a File Adapter for the Write Operation .....	6-37
Adjusting for Known Deployment Issues With 12c .....	6-43
<b>Designing an Outbound BPM Process Using Transformations for Service Integration (BSE Configuration).....</b>	<b>6-44</b>
Creating an Empty Composite for BPM .....	6-44
Defining a BPM Outbound Process.....	6-44
Configuring a Web Service Component.....	6-44
Configuring a BPM Process Component.....	6-46
Creating a File Adapter for the Write Operation .....	6-55

## **7 Configuring an Outbound and Inbound Process for Oracle Service Bus Using sbconsole**

<b>Overview of Application Adapter Integration with Oracle Service Bus .....</b>	<b>7-1</b>
<b>Configuring an Outbound Process Using sbconsole (J2CA Configuration).....</b>	<b>7-2</b>
Starting Oracle Service Bus and Creating Project Folders .....	7-2
Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus.....	7-6
Publishing a WSDL From Application Explorer to Oracle Service Bus.....	7-6
Configuring a WSDL-based Business Service.....	7-7
Configuring a File Type Business Service .....	7-9
Configuring a Pipeline With Proxy Service .....	7-13
<b>Configuring an Inbound Process Using sbconsole (J2CA Configuration) .....</b>	<b>7-22</b>
Starting Oracle Service Bus and Creating Project Folders .....	7-22
Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus.....	7-22
Publishing a WSDL From Application Explorer to Oracle Service Bus.....	7-23
Configuring a WSDL-based Proxy Service .....	7-24
Configuring a File Type Business Service .....	7-25
Configuring a Pipeline .....	7-29
<b>Configuring an Outbound Process Using sbconsole (BSE Configuration) .....</b>	<b>7-36</b>
Starting Oracle Service Bus and Creating Project Folders .....	7-36

Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus.....	7-36
Publishing a WSDL From Application Explorer to Oracle Service Bus.....	7-37
Configuring a File Type Business Service .....	7-38
Configuring a WSDL-based Business Service.....	7-38
Configuring a Pipeline With Proxy Service .....	7-40
<b>Configuring JMS Proxy Services Using Oracle Service Bus (J2CA Configuration) .....</b>	<b>7-49</b>
<b>Configuring HTTP Proxy Services Using Oracle Service Bus (J2CA Configuration) .....</b>	<b>7-64</b>

## **8 Configuring an Outbound and Inbound Process for Oracle Service Bus Using JDeveloper**

<b>Configuring an OSB Outbound Process Using JDeveloper (J2CA Configuration).....</b>	<b>8-1</b>
Creating a Service Bus Application for OSB .....	8-2
Defining an OSB Outbound Process .....	8-3
Configuring a Third-Party Adapter Service Component .....	8-3
Configuring a File Transport Type Business Service.....	8-6
Creating a Proxy Service With Pipeline.....	8-8
Configuring the Routing Rules.....	8-11
Deploying the OSB Outbound Process .....	8-15
<b>Configuring an OSB Inbound Process Using JDeveloper (J2CA Configuration) .....</b>	<b>8-18</b>
Creating a Service Bus Application for OSB .....	8-18
Defining an OSB Inbound Process.....	8-18
Configuring a Third-Party Adapter Service Component .....	8-19
Creating a Pipeline.....	8-21
Configuring a File Transport Type Business Service.....	8-23
Configuring the Routing Rules.....	8-25
Deploying the OSB Inbound Process .....	8-26
<b>Configuring an OSB Outbound Process Using JDeveloper (BSE Configuration).....</b>	<b>8-27</b>
Creating a Service Bus Application for OSB .....	8-27
Defining an OSB Outbound Process .....	8-27
Configuring a WSDL-based Business Service .....	8-27
Creating a Proxy Service With Pipeline.....	8-32
Configuring a File Transport Type Business Service.....	8-35
Configuring the Routing Rules.....	8-38
Deploying the OSB Outbound Process .....	8-41
<b>Configuring a JMS Inbound Process Using JDeveloper (J2CA Configuration) .....</b>	<b>8-41</b>
<b>Configuring a JMS Outbound Process Using JDeveloper (J2CA Configuration).....</b>	<b>8-47</b>
<b>Configuring an HTTP Outbound Process Using JDeveloper (J2CA Configuration) .....</b>	<b>8-54</b>

## **9 Key Features**

<b>Configuring the Logging Feature.....</b>	<b>9-1</b>
Configuring Log File Management for the J2CA Connector Application.....	9-1
Configuring Log File Management for Business Services Engine (BSE) .....	9-8
<b>Configuring the Diagnosibility Feature .....</b>	<b>9-11</b>
Supporting Protocols .....	9-12
Oracle Fusion Middleware Application Adapter for PeopleSoft Endpoint Status.....	9-12
<b>Configuring the SOA Debugging Feature .....</b>	<b>9-12</b>

Guidelines for Using the SOA Debugger .....	9-13
Prerequisite .....	9-13
Debugging a BPEL Process in Oracle JDeveloper .....	9-13
Debugging an Outbound BPEL Process in Oracle JDeveloper .....	9-13
Debugging an Inbound BPEL Process in Oracle JDeveloper .....	9-22
Debugging an OSB Process in Oracle JDeveloper .....	9-26
Prerequisites .....	9-26
Debugging an Outbound OSB Process in Oracle JDeveloper .....	9-29
Debugging an Inbound OSB Process in Oracle JDeveloper .....	9-36
<b>Exception Filter</b> .....	9-44
Configuring the Exception Filter .....	9-44
Generating a WSDL File.....	9-44
Creating a BPEL process With Exception Filter Functionality .....	9-44
Creating Fault Policies and Fault Binding Files .....	9-46
Adjusting for Known Deployment Issues With 12c .....	9-55
Deploying and Testing the BPEL Process With Exception Filter Functionality .....	9-55
<b>Credential Mapping for Oracle SOA Suite (BPEL, Mediator, or BPM)</b> .....	9-55
Configuring Credential Mapping .....	9-56
Associating Oracle WebLogic Server Credentials With EIS Credentials.....	9-56
Generating a WSDL File.....	9-59
Creating and Deploying an Outbound Process.....	9-59
Verifying the EIS Credentials.....	9-60
<b>Credential Mapping for Oracle Service Bus (OSB)</b> .....	9-60
Configuring Credential Mapping .....	9-61
Associating Oracle WebLogic Server Credentials With EIS Credentials.....	9-61
Generating a WSDL File.....	9-64
Creating an Oracle Service Bus (OSB) Outbound Process.....	9-64

## 10 Troubleshooting and Error Messages

<b>Troubleshooting</b> .....	10-1
Application Explorer .....	10-1
PeopleSoft.....	10-3
Oracle Adapter J2CA .....	10-4
<b>BSE Error Messages</b> .....	10-4
General Error Handling in BSE.....	10-5
Adapter-Specific Error Handling.....	10-5
Oracle Application Adapter for PeopleSoft Invalid SOAP Request .....	10-6
Empty Result From PeopleSoft Request.....	10-6
Failure to Connect to PeopleSoft .....	10-6
Invalid SOAP Request.....	10-7
Empty Result From Oracle WebLogic Server Application Adapter Request .....	10-7

## A Generating Component Interface APIs

<b>Building the PeopleSoft API Java Programs</b> .....	A-1
<b>Compiling the PeopleSoft API Java Programs</b> .....	A-5



<b>B</b>	<b>Configuring the PeopleSoft Message Router</b>	
	Configuring the TCP/IP or HTTP Target Connector for PeopleSoft 8.4.....	B-1
	Configuring the TCP/IP Target Connector.....	B-2
	Configuring the Node for the TCP/IP 8.4 Connector.....	B-3
	Configuring the HTTP Target Connector.....	B-6
	Configuring the HTTP Connector.....	B-8
	Configuring the TCP/IP Handler for PeopleSoft 8.1.....	B-12
	Configuring the HTTP Handler for PeopleSoft 8.1.....	B-15
	Testing Your PeopleSoft Configuration.....	B-15
<b>C</b>	<b>Using Component Interfaces</b>	
	Working With Properties.....	C-1
	Creating a New Component Interface.....	C-2
	Standard Methods.....	C-4
	Viewing or Changing Available Methods.....	C-4
	Securing a Component Interface.....	C-5
	Configuring Component Interface Security for PeopleSoft Version 8.1x.....	C-5
	Configuring Component Interface Security for PeopleSoft Version 8.4 or Higher.....	C-8
	Testing a Component Interface.....	C-14
<b>D</b>	<b>Using PeopleSoft Integration Broker</b>	
	PeopleSoft Integration Broker.....	D-1
	Configuring Integration Broker in PeopleSoft 8.4.....	D-2
	Ensuring the Message Is Active and Is Routed Correctly.....	D-3
	Configuring the IntegrationGateway.properties File.....	D-4
	Configuring the Integration Gateway and the File output Connector.....	D-5
	Creating and Configuring a New Gateway Node.....	D-7
	Configuring Application Messaging in PeopleSoft Release 8.1.....	D-11
	Creating and Configuring a New Message Node.....	D-12
	Ensuring the Message is Active and Is Routed Correctly.....	D-15
	Configuring the Message Channel.....	D-16
	Configuring the Simple File Handler in the Gateway.....	D-19
	Viewing the PeopleCode for a Message.....	D-21
	Testing the Integration Broker.....	D-22
	Using Outbound Synchronous Messages.....	D-24
	Configuring Outbound Synchronous Messages.....	D-24
	Configuring an Outbound Synchronous Message.....	D-25
	Viewing the PeopleCode for a Synchronous Message.....	D-26
	Viewing the PeopleCode for a Financials Synchronous Outbound Message.....	D-26
	Using PeopleTools Version 8.48 and Higher to Generate Events.....	D-27
	Configuring PeopleSoft Services.....	D-27
	Adding a Remote Node.....	D-28
	Selecting the Service.....	D-29
	Activating the Gateway.....	D-32
	Activating the Domain.....	D-33
	Activating the Service.....	D-34

Activating the Service Operation (Queue and Message Node) .....	D-34
Triggering the Event in PeopleSoft.....	D-37
Triggering the Event.....	D-37
Verifying the Event Results .....	D-38
Verifying the Results .....	D-39

**Glossary**

**Index**

---

---

# Preface

Welcome to *Oracle Fusion Middleware Application Adapter for PeopleSoft User's Guide for Oracle WebLogic Server*. This document provides information on how to integrate with PeopleSoft systems and develop applications.

## Audience

This document is intended for system administrators and developers who integrate with PeopleSoft systems and develop applications.

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Related Documents

For more information, see the following documents in the Oracle Enterprise Repository 12c (12.2.1.2.0) documentation set:

- *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter Upgrade Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter Best Practices Guide for Oracle WebLogic Server*
- Oracle's Unified Method (OUM)

A wealth of additional Governance information can be found within Oracle's Unified Method (OUM). OUM can be used by Oracle employees, Oracle Partner Network Certified Partners or Certified Advantage Partners, and Clients who either participate in the OUM Customer Program or are engaged on projects

where Oracle provides consulting services. OUM is a web-deployed toolkit for planning, executing and controlling software development and implementation projects.

For more information about OUM, see the OUM FAQ at

[http://my.oracle.com/portal/page/myo/ROOTCORNER/KNOWLEDGEAREAS1/BUSINESS\\_PRACTICE/Methods/Learn\\_about\\_OUM.html](http://my.oracle.com/portal/page/myo/ROOTCORNER/KNOWLEDGEAREAS1/BUSINESS_PRACTICE/Methods/Learn_about_OUM.html)

## Conventions

The following text conventions are used in this document:

<b>Convention</b>	<b>Meaning</b>
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

---

---

# Introduction

Oracle WebLogic Server connects to a PeopleSoft system through Oracle Application Adapter for PeopleSoft, which provides connectivity and carries out interactions on a PeopleSoft system. This chapter contains the following topics:

---

---

**Note:** Throughout this document, `<ORACLE_HOME>` refers to the 12c (12.2.1.0.0) SOA/OSB installed home location.

`<ADAPTER_HOME>` refers to the following:

- For SOA:

`<ORACLE_HOME>\soa\soa\thirdparty\ApplicationAdapters`

- For OSB:

`<ORACLE_HOME>\osb\3rdparty\ApplicationAdapters`

---

---

- [Section 1.1, "Adapter Features"](#)
- [Section 1.2, "PeopleSoft Concepts"](#)
- [Section 1.3, "Integration with PeopleSoft"](#)
- [Section 1.4, "Adapter Architecture"](#)
- [Section 1.5, "BSE Versus Oracle Adapter J2CA Deployment"](#)
- [Section 1.6, "Sample Projects"](#)
- [Section 1.7, "Quick Start Guide"](#)

## 1.1 Adapter Features

Oracle Application Adapter for PeopleSoft provides a means to exchange real-time business data between PeopleSoft systems and other applications, databases, and external business partner systems. The adapter enables inbound and outbound processing with PeopleSoft. Oracle Application Adapter for PeopleSoft can be deployed as a J2EE Connector Architecture (J2CA) version 1.0 resource adapter. This deployment is referred to as Oracle Adapter J2CA. It can also be deployed as a Web services servlet and is referred to as Oracle Adapter Business Services Engine (BSE).

Oracle Application Adapter for PeopleSoft uses XML messages to enable non-PeopleSoft applications to communicate and exchange transactions with

PeopleSoft using services and events. The roles of services and events are outlined as follows:

- Services: Enable applications to initiate a PeopleSoft business event.
- Events: Enable applications to access PeopleSoft data only when a PeopleSoft business event occurs.

To support event functionality, channels are supported. A **channel** represents configured connections to particular instances of back-end or other types of systems.

The channel is the adapter component that receives events in real time from the enterprise information system (EIS) application. The channel component can be a File reader, an HTTP listener, a TCP/IP listener, or an FTP listener. A channel is always EIS specific. The adapter supports multiple channels for a particular EIS, which enables the user to choose the optimal channel component based on deployment requirements.

Oracle Application Adapter for PeopleSoft provides:

- Synchronous and asynchronous, bidirectional message interactions for PeopleSoft component interfaces and PeopleSoft Application Messaging / Integration Broker.
- Oracle WebLogic Server Adapter Application Explorer (Application Explorer), a GUI tool that explores PeopleSoft metadata and builds XML schemas or Web services.
- XML schemas and WSDL files for the J2CA 1.0 and J2CA 1.5 resource adapter.
- Web services for BSE.

The adapter connects to the PeopleSoft Application Server by accessing APIs for the component interfaces that correspond to its supported business objects. Every component interface contains data and business logic for the business component, thus alleviating a requirement for the adapter to duplicate the processes defined within the business component.

**See Also:** *Oracle Application Server Adapter Concepts Guide*

## 1.2 PeopleSoft Concepts

PeopleSoft enables integration with other applications and systems through its component interface framework and its Integration Broker (in release 8.4) or Application Messaging (in release 8.1) facility. Oracle Application Adapter for PeopleSoft uses the PeopleSoft framework and leverages various integration access methods to provide the greatest amount of flexibility and functionality. Integration access methods supported by Oracle Application Adapter for PeopleSoft include:

- PeopleSoft Java API using component interfaces
- PeopleSoft messages using PeopleSoft Application Messaging / Integration Broker

This section contains the following topics:

- [Section 1.2.1, "PeopleSoft Component Interface"](#)
- [Section 1.2.2, "PeopleSoft Application Messaging / Integration Broker"](#)

### 1.2.1 PeopleSoft Component Interface

In the PeopleSoft environment, a component interface is a container for distributing PeopleSoft application data among PeopleSoft logical systems and for exchanging PeopleSoft application data with non-PeopleSoft systems.

The component interface is based on an existing business process within PeopleSoft. An example is a purchase order entry, which can be a PeopleSoft-delivered process or a user-developed process. The component interface also inherits its methods (Add, Update, and so on) and its business logic from the underlying business process.

PeopleSoft delivers generic component interfaces with each of its applications. These are called Enterprise Integration Points (EIP). Customers can also develop their own custom component interfaces, or they can modify EIP as required. Oracle Application Adapter for PeopleSoft supports both types of component interfaces.

## 1.2.2 PeopleSoft Application Messaging / Integration Broker

PeopleSoft Application Messaging / Integration Broker facilitates the integration of PeopleSoft XML with PeopleSoft. Oracle Application Adapter for PeopleSoft provides a handler that must be configured within the PeopleSoft application gateway using TCP/IP transport services.

## 1.3 Integration with PeopleSoft

Oracle Application Adapter for PeopleSoft enables you to:

- Access a PeopleSoft component using a component interface. Component methods are referred to as services.
- Receive messages from a PeopleSoft application. Messages are referred to as events.

When you access a PeopleSoft component from another application, you work with:

- Component interfaces. If a component interface does not exist, then create, secure, and test one. If the component interface exists but you modified it, then secure and test it. For more information, see your PeopleSoft documentation. Alternatively, you can secure and test the component interface and create the component interface API after you generate schemas or Web services.
- Component interface APIs. Create an API for the component interface.
- Schemas and Web services. Create schemas or Web services for the component methods.

For more information, see [Chapter 2, "Configuring Oracle Application Adapter for PeopleSoft"](#).

To receive a message from PeopleSoft, you work with:

- The Integration environment. Configure and test your PeopleSoft Integration Broker (release 8.4) or Application Messaging environment (release 8.1). To ensure that the environment is properly configured, see your PeopleSoft documentation.
- Message routing. Configure TCP/IP Target Connector (release 8.4), HTTP Target Connector (release 8.4), or TCP/IP Handler (release 8.1).

### See Also:

- *Oracle Application Server Adapter Concepts Guide*
- *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server*

## 1.4 Adapter Architecture

Oracle Application Adapter for PeopleSoft uses Application Explorer with one of the following components:

- Oracle WebLogic Server Adapter Business Services Engine (BSE)
- Enterprise Connector for J2EE Connector Architecture (J2CA)

Application Explorer (used to configure PeopleSoft connections and create Web services and events) can be configured to work in a Web services environment with BSE. When working in a J2CA environment, the connector uses the Common Client Interface (CCI) to provide integration services using adapters instead of Web services.

This section contains the following topics:

- [Section 1.4.1, "Oracle WebLogic Server Adapter Business Services Engine \(BSE\) Architecture"](#)
- [Section 1.4.2, "Oracle WebLogic Server Adapter J2CA Architecture"](#)

### 1.4.1 Oracle WebLogic Server Adapter Business Services Engine (BSE) Architecture

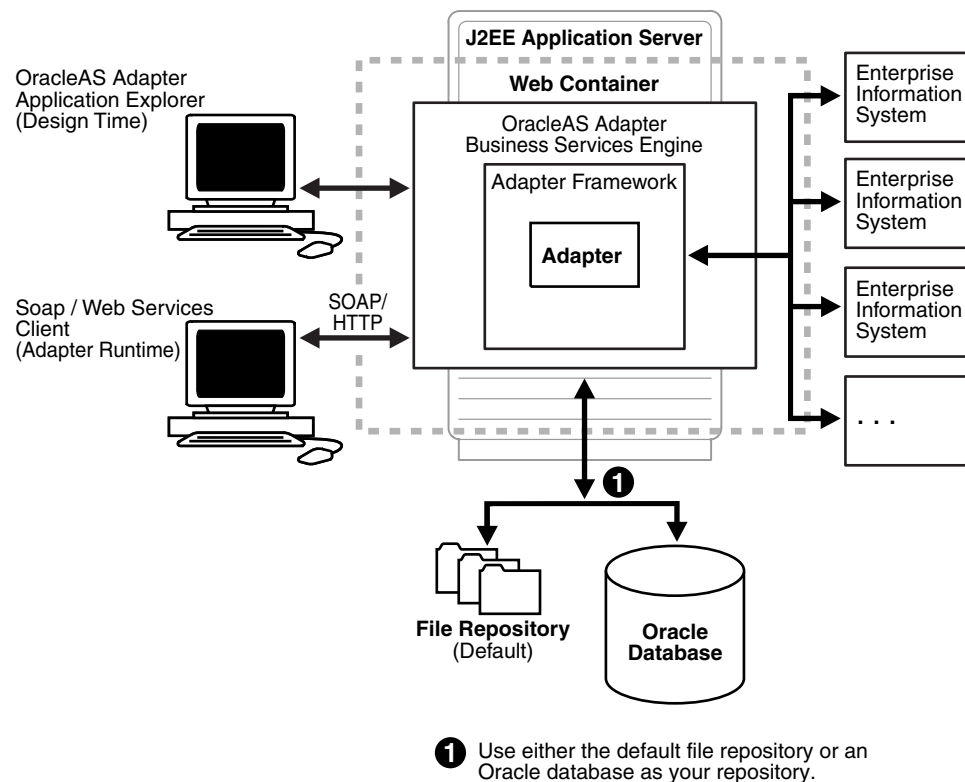
[Figure 1–1](#) shows the generic architecture for the Oracle Web service adapter for packaged applications. The adapter works with BSE, as deployed to a Web container in a J2EE application server.

Application Explorer, a design-time tool deployed along with BSE, is used to configure adapter connections, browse EIS objects, configure services, and configure listeners to listen for EIS events. Metadata created while you perform these operations are stored in the repository by BSE.

BSE uses SOAP as a protocol for receiving requests from clients, interacting with the EIS, and sending responses from the EIS back to clients.



**Figure 1–1 Oracle WebLogic Server Adapter Business Services Engine (BSE) Generic Architecture**




---

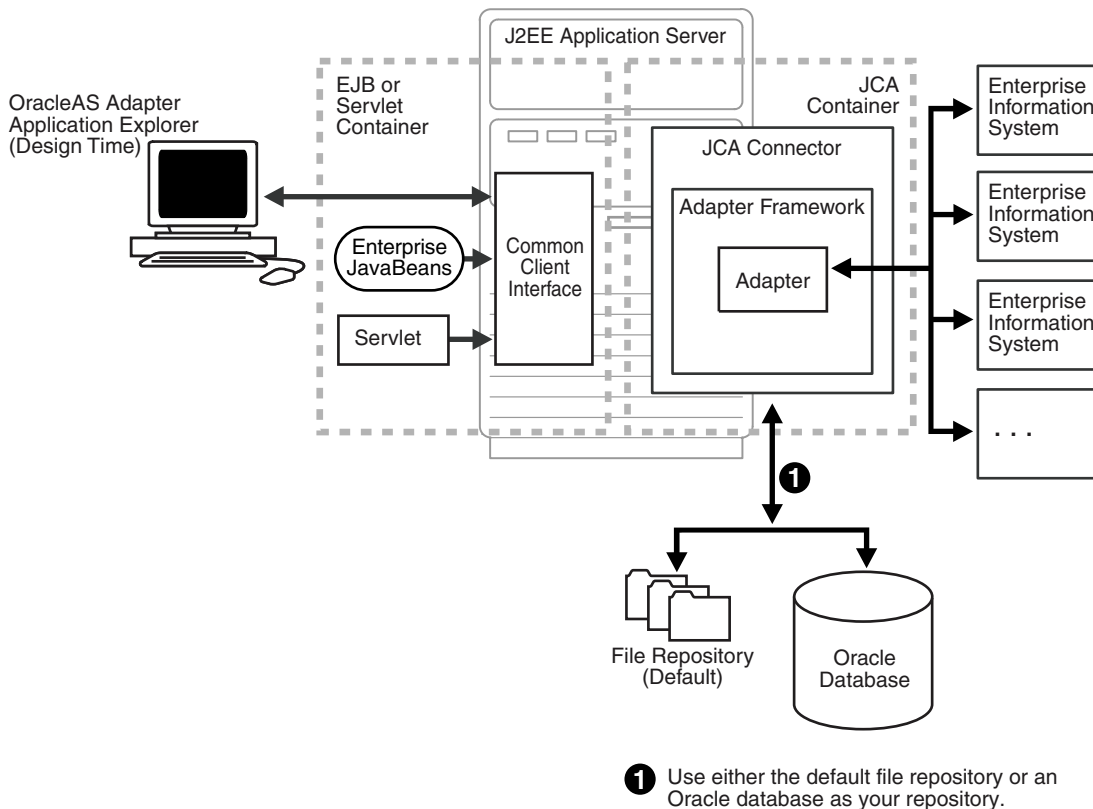
**Note:** Do not use a file repository for BSE in production environments.

---

## 1.4.2 Oracle WebLogic Server Adapter J2CA Architecture

Figure 1–2 shows the generic architecture for Oracle Adapter J2CA for packaged applications. Oracle Adapter J2CA is deployed to a standard J2CA Container and serves as a host container to the adapters. The connector is configured with a repository. The repository can be a file system or an Oracle database. It is deployed as a RAR file and has an associated deployment descriptor called `ra.xml`. You can create multiple connector factories by editing the Oracle WebLogic Server deployment descriptor `ra.xml`. For more information, see [Chapter 3, "Oracle WebLogic Server Deployment and Integration"](#).

Application Explorer, a design tool that works with the connector, is used to configure adapter connections, browse EIS objects, configure services, and configure listeners to listen for EIS events. Metadata created while you perform these operations are stored in the repository by the connector.

**Figure 1–2 Oracle WebLogic Server Adapter J2CA Generic Architecture**

## 1.5 BSE Versus Oracle Adapter J2CA Deployment

If you are using Oracle Application Adapter for PeopleSoft with Oracle SOA Suite components (for example, BPEL, Mediator, BPM, or OSB), then note that:

- Only Oracle Adapter J2CA deployment supports inbound integration (event notification) with Oracle SOA Suite components.
- Oracle Adapter J2CA and BSE deployments support outbound integration (request-response service) with Oracle SOA Suite components.

The following two factors explain the differences between deploying BSE and Oracle Adapter J2CA. Understanding these factors can help in selecting a deployment option.

1. BSE has the following advantages:
  - Can be deployed in a separate instance of Oracle WebLogic Server.
  - Provides better distribution of load.
  - Conforms more closely to the Service Oriented Architecture (SOA) model for building applications.
2. Oracle Adapter J2CA does provide slightly better performance than BSE.

## 1.6 Sample Projects

Sample projects for the Oracle Application Adapter for PeopleSoft that demonstrate outbound and inbound integration scenarios using Oracle BPEL, Mediator, BPM, and

OSB tools are packaged with the Application Adapters installation. The following table lists the locations of the sample projects:

<b>Sample Project</b>	<b>Location</b>
Outbound BPEL Process (J2CA)	<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\BPEL\J2CA\Outbound_Project
Inbound BPEL Process (J2CA)	<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\BPEL\J2CA\Inbound_Project
Outbound BPEL Process (BSE)	<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\BPEL\BSE\Outbound_Project
Outbound Mediator Process (J2CA)	<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\Mediator\J2CA\Outbound_Project
Inbound Mediator Process (J2CA)	<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\Mediator\J2CA\Inbound_Project
Outbound Mediator Process (BSE)	<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\Mediator\BSE\Outbound_Project
Outbound BPM Process (J2CA)	<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\BPM\J2CA\Outbound_Project
Inbound BPM Process (J2CA)	<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\BPM\J2CA\Inbound_Project
Outbound BPM Process (BSE)	<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\BPM\BSE\Outbound_Project
Outbound OSB sbconsole Process (J2CA)	<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\OSB\J2CA\PeopleSoft_Sample_J2CA_OSB_Outbound_Project
Inbound OSB sbconsole Process (J2CA)	<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\OSB\J2CA\PeopleSoft_Sample_J2CA_OSB_Inbound_Project
Outbound OSB sbconsole Process (BSE)	<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\OSB\BSE\PeopleSoft_Sample_BSE_OSB_Outbound_Project

Sample Project	Location
Outbound OSB Jdeveloper Process (J2CA)	<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\OSB_Jdeveloper\J2CA\PeopleSoft_Sample_J2CA_OSB_Outbound_Project
Inbound OSB Jdeveloper Process (J2CA)	<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\OSB_Jdeveloper\J2CA\PeopleSoft_Sample_J2CA_OSB_Inbound_Project
Outbound OSB Jdeveloper Process (BSE)	<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\OSB_Jdeveloper\BSE\PeopleSoft_Sample_BSE_OSB_Outbound_Project

## 1.7 Quick Start Guide

This section enables you to quickly learn the basic steps to install and configure Oracle Application Adapter for PeopleSoft and to use it immediately. It includes the following topics:

- [Section 1.7.1, "Installation"](#)
- [Section 1.7.2, "Copying Third-Party Library Files"](#)
- [Section 1.7.3, "Configuration"](#)
- [Section 1.7.4, "WebLogic Server Deployment and Integration"](#)
- [Section 1.7.5, "Creating Configurations, Targets, and Channels in Application Explorer"](#)
- [Section 1.7.6, "Working With Service Components in the SOA Suite"](#)
- [Section 1.7.7, "Working With Oracle Service Bus"](#)
- [Section 1.7.8, "Other Features"](#)

### 1.7.1 Installation

To install Oracle Application Adapter for PeopleSoft, download the Oracle Fusion Middleware Application Adapters installer and complete the installation for SOA/OSB.

For more information on installing the Oracle Fusion Middleware Application Adapters, see the *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server*.

### 1.7.2 Copying Third-Party Library Files

Once the adapter installation is completed, copy the required third-party library files for PeopleSoft to the following directories:

```
<ADAPTER_HOME>\lib
```

```
<ORACLE_HOME>\user_projects\domains\base_domain\lib
```

For more information on installing Oracle Fusion Middleware Application Adapters, see the *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server*.

### 1.7.3 Configuration

Navigate to `<ADAPTER_HOME>` and make the following changes:

1. Open `iwafjca.rar\META-INF\ra.xml` and add the following values under the specified config-property-name parameters, as shown in [Table 1-1](#).

**Table 1-1**

Config-Property-Name	Config-Property-Value
<code>IWayHome</code>	<code>&lt;ADAPTER_HOME&gt;</code> For example: <ul style="list-style-type: none"> <li>■ <b>For SOA:</b>  <code>C:\12c_soa\soa\soa\thirdparty\ApplicationAdapters</code></li> <li>■ <b>For OSB:</b>  <code>C:\12c_OSB\osb\3rdparty\ApplicationAdapters</code></li> </ul>
<code>IWayConfig</code>	The name of the configuration. For example: <code>jca_sample</code>

2. Open `ibse.war\WEB-INF\web.xml` and add the following values under the specified param-name parameters, as shown in [Table 1-2](#).

**Table 1-2**

Param-Name	Param-Value
<code>ibseroot</code>	<code>&lt;ADAPTER_HOME&gt;\ibse.war</code> For example: <ul style="list-style-type: none"> <li>■ <b>For SOA:</b>  <code>C:\12c_soa\soa\soa\thirdparty\ApplicationAdapters\ibse.war</code></li> <li>■ <b>For OSB:</b>  <code>C:\12c_OSB\osb\3rdparty\ApplicationAdapters\ibse.war</code></li> </ul>
<code>IWay.home</code>	<code>&lt;ADAPTER_HOME&gt;</code> For example: <ul style="list-style-type: none"> <li>■ <b>For SOA:</b>  <code>C:\12c_soa\soa\soa\thirdparty\ApplicationAdapters</code></li> <li>■ <b>For OSB:</b>  <code>C:\12c_OSB\osb\3rdparty\ApplicationAdapters</code></li> </ul>

**Table 1–2 (Cont.)**

Param-Name	Param-Value
<i>Iway.config</i>	The name of the configuration. For example: IBSE

---

**Note:** These steps are provided only when configuring a File repository. For more information about configuring a database repository and general configuration information, see [Chapter 2, "Configuring Oracle Application Adapter for PeopleSoft"](#) and [Chapter 3, "Oracle WebLogic Server Deployment and Integration"](#).

---

### 1.7.4 WebLogic Server Deployment and Integration

1. Start the WebLogic server and open the WebLogic console.
2. Deploy the adapter components (ibse.war, iwafjca.war, and iwafjca.rar files) and start the deployed adapter components.

For more information on deployment, integration, and target creation, see [Chapter 3, "Oracle WebLogic Server Deployment and Integration"](#).

### 1.7.5 Creating Configurations, Targets, and Channels in Application Explorer

For more information on creating configurations, targets, and channels in Application Explorer, see the following sections in this user's guide:

- Starting Application Explorer: [Section 2.1, "Starting Application Explorer"](#)
- Creating a BSE Configuration: [Section 2.3.1, "Creating a Configuration for BSE"](#)
- Creating a J2CA Configuration: [Section 2.3.2, "Creating a Configuration for J2CA"](#)
- Connecting the Created Configurations: [Section 2.3.3, "Connecting to a BSE or J2CA Configuration"](#)
- Creating and Connecting to Targets: [Section 2.4, "Establishing a Connection \(Target\) for PeopleSoft"](#)
- Creating and Testing Web Services: [Section 2.8, "Creating and Testing a Web Service \(BSE Configurations Only\)"](#)
- Generating WSDL Files: [Section 2.7, "Generating a WSDL \(J2CA Configurations Only\)"](#)
- Creating and Working With Channels: [Section 2.9, "Configuring an Event Adapter"](#)

### 1.7.6 Working With Service Components in the SOA Suite

Oracle Application Adapter for PeopleSoft integrates with service components in SOA suite such as BPEL, Mediator, and BPM. Required processes are created in JDeveloper and then deployed to the SOA server.

For more information on working with BPEL, Mediator, and BPM service components, see:

- [Chapter 4, "Integration With BPEL Service Components in the Oracle SOA Suite"](#)

- [Chapter 5, "Integration With Mediator Service Components in the Oracle SOA Suite"](#)
- [Chapter 6, "Integration With BPM Service Components in the Oracle SOA Suite"](#)

### 1.7.7 Working With Oracle Service Bus

Oracle Application Adapter for PeopleSoft integrates with Oracle Service Bus (OSB) to facilitate Web service integration. Required processes are created in the Oracle Service Bus Console. The process can also be created in JDeveloper and then deployed to the SOA server.

For more information on working with OSB Console, see [Chapter 7, "Configuring an Outbound and Inbound Process for Oracle Service Bus Using sbconsole"](#).

For more information on working with OSB Jdeveloper, see [Chapter 8, "Configuring an Outbound and Inbound Process for Oracle Service Bus Using JDeveloper"](#).

### 1.7.8 Other Features

The following is list of other features and their relevant sections in this user's guide:

- [Configuring the Exception Filter: Section 9.4, "Exception Filter"](#)
- [Configuring Credential Mapping:](#)
  - [Section 9.5, "Credential Mapping for Oracle SOA Suite \(BPEL, Mediator, or BPM\)"](#)
  - [Section 9.6, "Credential Mapping for Oracle Service Bus \(OSB\)"](#)





---

# Configuring Oracle Application Adapter for PeopleSoft

Oracle Adapter Application Explorer (Application Explorer) enables the processing of Component Interfaces and Messages.

External applications that access PeopleSoft through the adapter use either XML schemas or Web services to pass data between the external application and the adapter. You can use Application Explorer to create the required XML schemas and Web services.

This chapter contains the following topics:

- [Section 2.1, "Starting Application Explorer"](#)
- [Section 2.2, "Configuring Repository Settings"](#)
- [Section 2.3, "Creating a Repository Configuration"](#)
- [Section 2.4, "Establishing a Connection \(Target\) for PeopleSoft"](#)
- [Section 2.5, "Viewing Application System Objects"](#)
- [Section 2.6, "Creating XML Schemas"](#)
- [Section 2.7, "Generating a WSDL \(J2CA Configurations Only\)"](#)
- [Section 2.8, "Creating and Testing a Web Service \(BSE Configurations Only\)"](#)
- [Section 2.9, "Configuring an Event Adapter"](#)

## 2.1 Starting Application Explorer

To start Application Explorer:

1. Ensure that Oracle WebLogic Server is started, which is where Application Explorer is deployed.
2. Open the command prompt.
3. Navigate to the following directory:

```
<ADAPTER_HOME>\user_projects\domains\base_domain\bin
```

4. Execute *setDomainEnv.cmd* (Windows) or *./setDomainEnv.sh* (UNIX/Linux).

This command sets the class path and other environment variables for Application Explorer in the Oracle WebLogic Server environment. In addition, it allows Application Explorer to access the Oracle WebLogic Server APIs to publish WSDL files to the Oracle Service Bus (OSB) Console.

5. Do not close the command prompt.
6. Navigate to the following directory:  

```
<ADAPTER_HOME>\tools\iwaeb\bin
```
7. Execute `ae.bat` (Windows) or `iwaeb.sh` (UNIX/Linux) to start Application Explorer.

Application Explorer starts. You are ready to define new targets to your PeopleSoft system.

---

---

**Note:** Before you run the `iwaeb.sh` file on UNIX or Linux platforms, the permissions must be changed. For example:

```
chmod +x iwaeb.sh
```

---

---

## 2.2 Configuring Repository Settings

A repository holds information about configuration details, adapter targets, channels, and other configuration information. For more information on how to configure BSE and J2CA repository settings, see the *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server* (Section 2.7.4 "Configuring the Oracle Database Repository").

## 2.3 Creating a Repository Configuration

Before you use Application Explorer with Oracle Application Adapter for PeopleSoft, you must create a repository configuration. You can create two kinds of repository configurations, Web services and J2CA, depending on the container to which the adapter is deployed.

During design time, the repository is used to store metadata created when using Application Explorer to configure adapter connections, browse EIS objects, configure services, and configure listeners to listen for EIS events. The information in the repository is also referenced at run-time.

Web services and BSE refer to the same type of deployment. For more information, see ["Adapter Features"](#) on page 1-1.

This section contains the following topics:

- [Section 2.3.1, "Creating a Configuration for BSE"](#)
- [Section 2.3.2, "Creating a Configuration for J2CA"](#)
- [Section 2.3.3, "Connecting to a BSE or J2CA Configuration"](#)

### 2.3.1 Creating a Configuration for BSE

To create a repository configuration for BSE using Application Explorer, you must first define a new configuration.

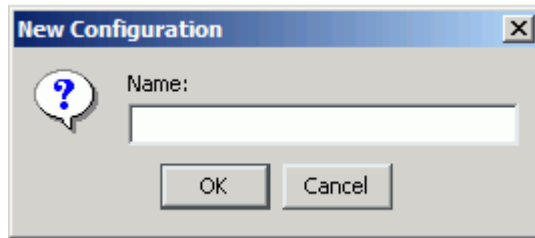
#### Defining a New Configuration for BSE

To define a new configuration for BSE:

1. Start the Application Explorer.
2. Right-click **Configurations** and select **New**.

The New Configuration dialog is displayed, as shown in [Figure 2–1](#).

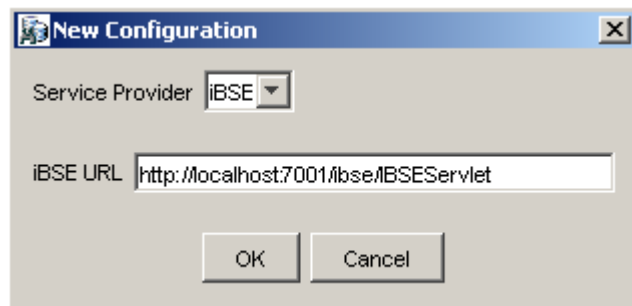
**Figure 2–1 New Configuration Dialog**



3. Enter a name for the new configuration (for example, myConfig) and click **OK**.

The New Configuration dialog is displayed, as shown in [Figure 2–2](#).

**Figure 2–2 New Configuration Dialog - Service Provider List and iBSE URL**



4. From the **Service Provider** list, select **iBSE**.
5. In the **BSE URL** field, accept the default URL or replace it with a different URL with the following format:

```
http://host name:port/ibse/IBSEServlet
```

Where *host name* is the system where Oracle WebLogic Server resides and *port* is the HTTP port for a managed Oracle WebLogic Server (for example, soa\_server1).

6. Click **OK**.

A node representing the new configuration appears beneath the root Configurations node, as shown in [Figure 2–3](#).

**Figure 2–3 New Configuration Node - myConfig**



## 2.3.2 Creating a Configuration for J2CA

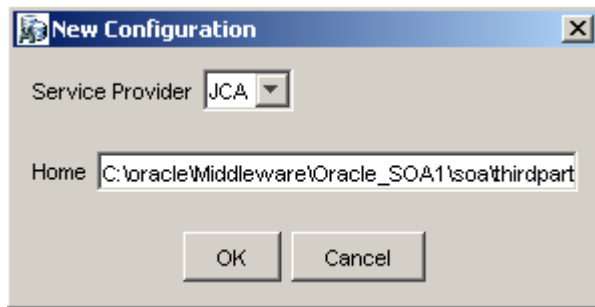
To create a configuration for Oracle Adapter J2CA using Application Explorer, you must first define a new configuration.

To define a new configuration for J2CA:

1. Start the Application Explorer.

2. Right-click **Configurations** and select **New**.  
The New Configuration dialog is displayed, as shown in [Figure 2-4](#).

**Figure 2-4 New Configuration Dialog**



3. Enter a name for the new configuration (for example, myConfig) and click **OK**.
4. From the **Service Provider** list, select **JCA**.
5. Click **OK**.

A node representing the new configuration appears beneath the root Configurations node, as shown in [Figure 2-5](#).

**Figure 2-5 New Configuration Node - myConfig**



The Oracle Adapter J2CA configuration folder is stored in a location based on your adapter installation:

```
<ADAPTER_HOME>\config\configuration_name
```

The *configuration\_name* is the name of the configuration you created (for example, SampleConfig).

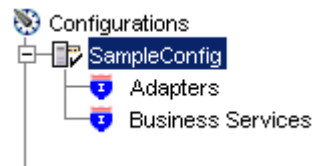
### 2.3.3 Connecting to a BSE or J2CA Configuration

To connect to a new configuration:

1. Right-click the configuration to which you want to connect, for example, SampleConfig.
2. Select **Connect**.

Nodes appear for Adapters, Events, and Business Services (also known as Web services). The Business Services node is only available for BSE configurations. If you are connected to a J2CA configuration, then the Business Services node is not shown. As shown in [Figure 2-6](#), the following is an example of a BSE configuration named SampleConfig:

**Figure 2–6 The New SampleConfig Configuration That Appears Under The Configurations Node**



- Use the **Adapters** node to create inbound interaction with PeopleSoft. For example, you use the PeopleSoft node in the Adapters node to configure a service that updates PeopleSoft.
- Use the **Events** node (available for J2CA configurations only) to configure listeners that listen for events in PeopleSoft.
- Use the **Business Services** node (available for BSE configurations only) to test Web services created in the Adapters node. You can also control security settings for the Web services by using the security features of the Business Services node.

You can now define new targets to PeopleSoft.

## 2.4 Establishing a Connection (Target) for PeopleSoft

Part of the application definition includes adding a target for Oracle Application Adapter for PeopleSoft. Setting up the target in Application Explorer requires information that is specific to the target.

To browse PeopleSoft business objects, you must create a target for the system you intend to use. The target serves as your connection point and is automatically saved after you create it. You must establish a connection to this system every time you start Application Explorer or after you disconnect from the system.

When you launch Application Explorer, the left pane displays (as nodes) the application systems supported by Application Explorer, based on the adapters that are installed.

This section contains the following topics:

- [Section 2.4.1, "Defining a Target to PeopleSoft"](#)
- [Section 2.4.2, "Connecting to a Defined PeopleSoft Target"](#)
- [Section 2.4.3, "Managing a Target"](#)

### 2.4.1 Defining a Target to PeopleSoft

To connect to PeopleSoft for the first time, you must define a new target. Oracle Application Adapter for PeopleSoft supports PeopleSoft standard security, in addition to component interface-level security. Once connected to the PeopleSoft application server, application security is managed by user ID, roles and privileges. For more information on PeopleSoft application security, see the appropriate PeopleSoft documentation.

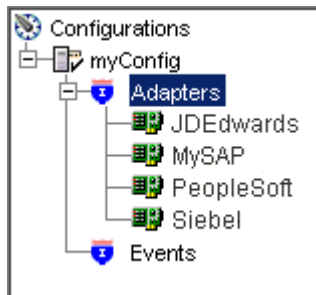
When you define a new target, you must restart the Oracle WebLogic Server to update the repository for run time purposes.

To define a target:

1. In the left pane, expand the **Adapters** node.

The applications systems supported by Application Explorer appear as nodes based on the adapters that are installed, as shown in [Figure 2-7](#).

**Figure 2-7 Adapters List**



2. Right-click the **PeopleSoft** node and select **Add Target**.

The Add Target dialog is displayed. Provide the following information:

- a. In the **Name** field, enter a descriptive name for the target, for example, **PSoftTarget**.
- b. In the **Description** field, enter a description for the target (optional).
- c. From the **Target Type** list, select **Application Server**.

This is the only possible value for target type.

3. Click **OK**.

The Application Server dialog is displayed, as shown in [Figure 2-8](#). You must specify connection information for PeopleSoft and the application server that is hosting PeopleSoft.

**Figure 2-8 Application Server Dialog**

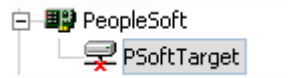
Provide the following information:

- a. In the **Application Server** field, enter the host name or IP address for the computer that is hosting the PeopleSoft application.
- b. In the **Port** field, enter the port number where the PeopleSoft application is listening.
- c. In the **User** field, enter a valid user ID for the PeopleSoft application.

- d. In the **Password** field, enter a valid password for the PeopleSoft application.
4. Click **OK**.

In the left pane, the new target (**PSoffTarget**) appears the PeopleSoft node, as shown in [Figure 2-9](#).

**Figure 2-9** *Disconnected PeopleSoft Target Node*



You are ready to connect to your PeopleSoft target.

## 2.4.2 Connecting to a Defined PeopleSoft Target

To connect to an existing target:

1. In the left pane, expand the **Adapters** node.
2. Expand the **PeopleSoft** node.
3. Click the target name (for example, **PSoffTarget**) under the PeopleSoft node.  
The Connection dialog displays the values you entered for connection parameters.
4. Verify your connection parameters.
5. Right-click the target name and select **Connect**.

The x icon disappears, indicating that the node is connected. A list of PeopleSoft business objects is displayed, as shown in [Figure 2-10](#).

**Figure 2-10** *PeopleSoft Business Objects*



## 2.4.3 Managing a Target

Although you can maintain multiple open connections to different transaction processing systems, it is recommended that you disconnect from connections not in use. After you disconnect, you can modify an existing target.

You can modify the connection parameters when your system properties change. You also can delete a target. The following procedures describe how to disconnect from a target, edit a target, and delete a target.

This section contains the following topics:

- [Section 2.4.3.1, "Disconnecting from a Connection to PeopleSoft"](#)
- [Section 2.4.3.2, "Modifying Connection Parameters"](#)
- [Section 2.4.3.3, "Deleting a Connection to PeopleSoft"](#)

### 2.4.3.1 Disconnecting from a Connection to PeopleSoft


To disconnect from a connection to PeopleSoft:

1. Expand the **Adapters** node.
2. Expand the **PeopleSoft** node.
3. Right-click the target to which you are connected (for example, PSoftTarget), and select **Disconnect**, as shown in [Figure 2–11](#).

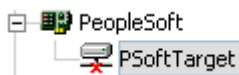
**Figure 2–11 Disconnect**



Disconnecting from PeopleSoft drops the connection with PeopleSoft, but the node remains.

The  icon appears, indicating that the node is disconnected, as shown in [Figure 2–12](#).

**Figure 2–12 Disconnected PeopleSoft Target Node**



### 2.4.3.2 Modifying Connection Parameters

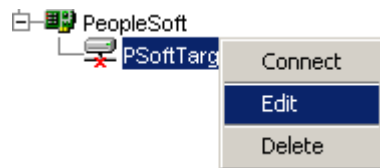
After you create a target for PeopleSoft using Application Explorer, you can edit any of the information that you provided previously.

When modify connection parameters for a defined target, you must restart the Oracle WebLogic Server to update the repository for run time purposes.

To edit a target:

1. Verify that the target you want to edit is disconnected.
2. Right-click the target and select **Edit**, as shown in [Figure 2–13](#).

**Figure 2–13 PSoftTarget Node Menu**



The Application Server dialog displays the target connection information.

3. Change the properties in the dialog as required and click **OK**.



### 2.4.3.3 Deleting a Connection to PeopleSoft

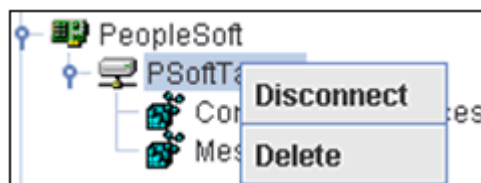
You can delete a connection, rather than just disconnecting and closing it. When you delete the connection, the node disappears from the list of PeopleSoft connections in the left pane of Application Explorer.

When you delete a connection, you must restart the Oracle WebLogic Server to update the repository for run time purposes.

To delete a connection to PeopleSoft:

1. Locate the target you want to delete.
2. Right-click the target (for example, PsoftTarget), and select **Delete**, as shown in Figure 2-14.

**Figure 2-14 PSoftTarget Node - Delete Option**



The node disappears from the list of available connections.

## 2.5 Viewing Application System Objects

After you are connected to PeopleSoft, Application Explorer enables you to explore and browse business object metadata. For example, Application Explorer enables you to view PeopleSoft Component Interface and Message metadata stored in the PeopleSoft business object repository.

To view application system objects:

1. Click the icon to the left of the target name, for example, PSoftTarget.

The target expands and the available system objects are displayed, as shown in Figure 2-15.

**Figure 2-15 PSoftTarget Node Objects**



2. To expand the desired PeopleSoft repository node, click the icon to the left of the repository name, for example, Component Interfaces.

A list of PeopleSoft Component Interfaces appears. You can now generate schemas.

## 2.6 Creating XML Schemas

After you browse the PeopleSoft business object repository, you can generate XML request and response schemas for the object you want to use with your adapter.

This section contains the following topics:

- [Section 2.6.1, "Creating XML Request and Response Schemas Against BSE"](#)
- [Section 2.6.2, "Creating XML Request and Response Schemas Against the Oracle Adapter J2CA"](#)

## 2.6.1 Creating XML Request and Response Schemas Against BSE

To create XML request and response schemas for a PeopleSoft Component Interface against a BSE implementation:

1. Expand the PeopleSoft node and then, expand the **Component Interfaces** node.
2. From the list of **Component Interfaces**, select **LOCATION**.
3. Navigate to and right-click an operation.
4. Click the **Request Schema** or **Response Schema** tab to view the request or response schema information.

The schema you selected is displayed.

After you browse the Component Interfaces and make a selection, the request and response XML schemas are automatically created for that Component Interface and stored in the repository you created.

## 2.6.2 Creating XML Request and Response Schemas Against the Oracle Adapter J2CA

To create XML request and response schemas for a PeopleSoft Component Interface against an Oracle Adapter J2CA implementation:

1. Expand the **PeopleSoft** node and then, expand the **Component Interfaces** node.
2. From the list of **Component Interfaces**, select **LOCATION**.
3. Navigate to and right-click an operation.
4. Click the **Request Schema** or **Response Schema** tab to view the request or response schema information.

The schema you selected is displayed.

After you browse the Component Interfaces and make a selection, the request and response XML schemas are automatically created for that Component Interface and stored in the repository you created.

## 2.7 Generating a WSDL (J2CA Configurations Only)

The Web Service Definition Language (WSDL) description of a service enables you to make the service available to other services within a host server. You use Application Explorer to create both request-response (outbound) and event notification (inbound) JCA services of the adapter.

---

---

**Note:** The **Create Inbound JCA Service (Event)** option is only available when the selected node supports events.

---

---

---

---

**Note:** PeopleSoft Component Interfaces only support services. As a result, only outbound WSDL files can be generated.

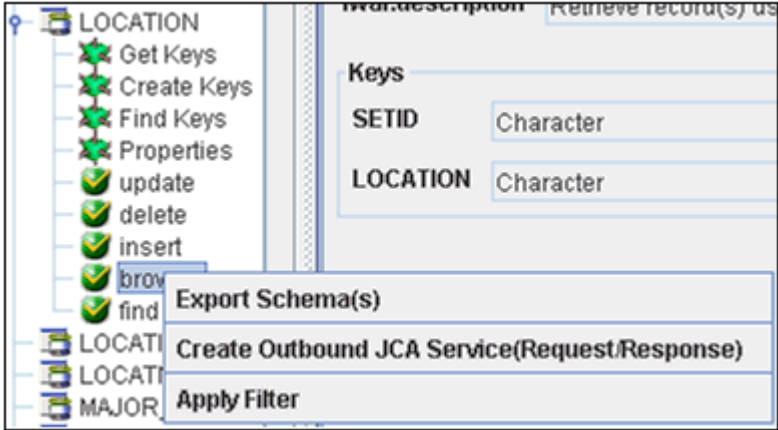
---

---

To generate a WSDL file for request-response service:

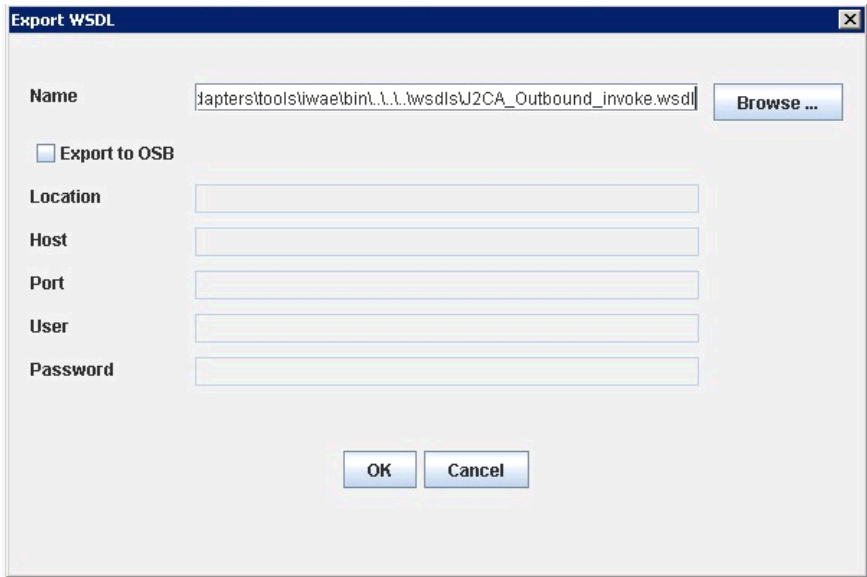
1. After you create a schema, right-click the respective object.  
The options are displayed, as shown in [Figure 2-16](#).

**Figure 2-16 Create Outbound JCA Service (Request/Response) Option**



2. Select **Create Outbound JCA Service (Request/Response)**.  
The Export WSDL dialog is displayed, as shown in [Figure 2-17](#).

**Figure 2-17 Export WSDL Dialog**



3. Accept the default name for the file.  
The **.wsdl** file extension is added automatically. By default, the names of WSDL files generated for request-response services end with **\_invoke**, while those generated for event notification end with **\_receive**.
4. Click **OK**.  
The WSDL file is saved in the specified location.

The procedure for generating WSDL for event notification is similar to request-response. To generate WSDL for event notification, you must first create a channel for every event.

## 2.8 Creating and Testing a Web Service (BSE Configurations Only)

You can create Web services (also known as a **business service**) using Application Explorer. The PeopleSoft Component Interface called LOCATION is used as an example in the following procedure.

---



---

**Note:** In a J2EE Connector Architecture (J2CA) implementation of adapters, Web services are not available. When the adapters are deployed to use Oracle Adapter J2CA, the Common Client Interface provides integration services using the adapters.

---



---

This section contains the following topics:

- [Section 2.8.1, "Creating a Web Service"](#)
- [Section 2.8.2, "Testing a Web Service"](#)
- [Section 2.8.3, "Identity Propagation"](#)

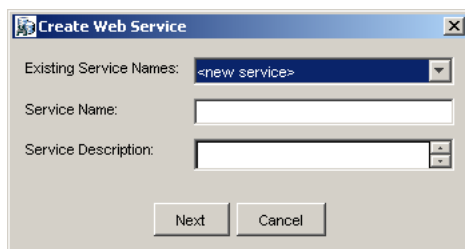
### 2.8.1 Creating a Web Service

To create a Web service, perform the following steps:

1. Connect the created PeopleSoft target under the BSE configuration and expand the **Component Interfaces** node.
2. From the list of **Component Interfaces**, select **LOCATION**.
3. Right-click the node from which you want to create a **business service** and select **Create Web Service**.

The Create Web Service dialog is displayed, as shown in [Figure 2-18](#).

**Figure 2-18** Create Web Service Dialog



You can add the business function as a method for a new Web service or as a method for an existing one. Perform the following steps:

- a. From the **Existing Service Names** list, select either **<new service>** or an existing service.
- b. If you are creating a new service, then specify a service name. This name identifies the Web service in the list of services under the **Business Services** node.
- c. Enter a brief description for the service (optional).

#### 4. Click **Next**.

The license and method dialog is displayed.

Perform the following steps:

- a. In the **License Name** field, select one or more license codes to assign to the Web service. To select more than one, hold down the **Ctrl** key and click the licenses.
- b. In the **Method Name** field, leave the default method name.
- c. In the **Method Description** field, enter a brief description of the method (optional).
- d. The **DTD Directory** field specifies a destination for your Web service. You may click **Browse** to change the default location.

#### 5. Click **OK**.

Application Explorer switches the view to the **Business Services** node, and the new Web service appears in the left pane.

#### 6. Right-click the new Web service and select **Save WSDL** from the menu.

The Save dialog is displayed.

#### 7. Provide a name for the WSDL file and a location to save the WSDL file on your file system.

#### 8. Click **Save**.

## 2.8.2 Testing a Web Service

After a Web service is created, you can test it to ensure that it functions properly. A test tool is provided for testing the Web service.

To test a business service:

1. Click the **Business Services** node to access your Web services.
2. Expand the **Services** node.
3. Select the name of the business service you want to test.

The business service name appears as a link in the right pane.

#### 4. In the right pane, click the named business services link.

The test option appears in a separate BSE Servlet page. If you are testing a Web service that requires XML input, then an input field appears.

#### 5. Enter the appropriate input.

#### 6. Click **Invoke**.

The BSE Servlet page displays the results.

## 2.8.3 Identity Propagation

If you test or execute a Web service using a third party XML editor, for example XMLSPY, then the user name and password values that you specify in the SOAP header must be valid and are used to connect to PeopleSoft. The user name and password values that you provided for PeopleSoft during target creation using Application Explorer are overwritten for this Web service request. The following is a sample SOAP header that is included in the WSDL file for a Web service:

```
<SOAP-ENV:Header>
  <m:ibsinfo xmlns:m="urn:schemas-iwaysoftware-com:iwse">
    <m:service>String</m:service>
    <m:method>String</m:method>
    <m:license>String</m:license>
    <m:disposition>String</m:disposition>
    <m:Username>String</m:Username>
    <m>Password>String</m>Password>
    <m:language>String</m:language>
  </m:ibsinfo>
</SOAP-ENV:Header>
```

You can remove the `<m:disposition>` and `<m:language>` tags from the SOAP header, since they are not required.

## 2.9 Configuring an Event Adapter

Events are generated by activity in an application system. You can use events to trigger an action in your application. For example, PeopleSoft may generate an event when customer information is updated. If your application performs an action when this happens, then your application is a consumer of this event.

---

---

**Note:** BSE configurations do not support events.

---

---

After you create a connection to your application system, you can add events using Application Explorer. To create an event, you must create a channel.

---

---

**Note:** If you are using a J2CA configuration, then you must create a new channel for every event object and select this channel when you generate WSDL. Creating a channel is required for J2CA configurations only.

---

---

A **channel** represents configured connections to particular instances of back-end systems. For more information, see ["Creating and Editing a Channel"](#) on page 2-14.

This section contains the following topics:

- [Section 2.9.1, "Creating and Editing a Channel"](#)
- [Section 2.9.2, "Schema Validation"](#)

### 2.9.1 Creating and Editing a Channel

This section describes how to create a channel for your event. All defined event ports must be associated with a channel.

When you create, modify, or delete a channel, you must restart the Oracle WebLogic Server to recognize the change and update the repository for run time purposes.

---

---

**Note:** If using a J2CA configuration, then you must create a new channel for every event and select this channel when creating an inbound service. Creating a channel is required for J2CA configurations only.

---

---

You can create the following types of channels using Application Explorer:

- HTTP

It is recommended to use HTTP channels with PeopleTools Version 8.4.

- TCP

It is recommended to use TCP channels with PeopleTools Version 8.1.

- File

It is recommended to use File channels for testing purposes only. Do not use File channels in a production environment.

---

---

**Note:** Channels can be configured and started only on the system where the Oracle Application Adapter for PeopleSoft is installed. Configuring and starting a channel for a remote host is not supported.

---

---

This section contains the following topics:

- [Section 2.9.1.1, "Creating an HTTP Channel"](#)
- [Section 2.9.1.2, "Creating a TCP Channel"](#)
- [Section 2.9.1.3, "Creating a File Channel"](#)
- [Section 2.9.1.4, "Editing a Channel"](#)
- [Section 2.9.1.5, "Deleting a Channel"](#)

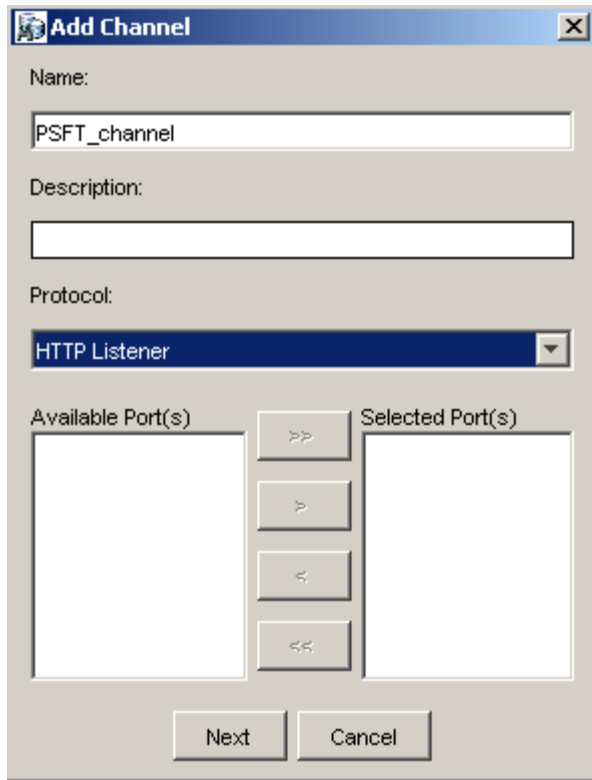
### 2.9.1.1 Creating an HTTP Channel

To create an HTTP Channel:

1. Click the **Events** node.
2. Expand the **PeopleSoft** node.  
The ports and channels nodes appear in the left pane.
3. Right-click **Channels** and select **Add Channel**.

The Add Channel dialog is displayed, as shown in [Figure 2–19](#).

**Figure 2–19 Add Channel Dialog**

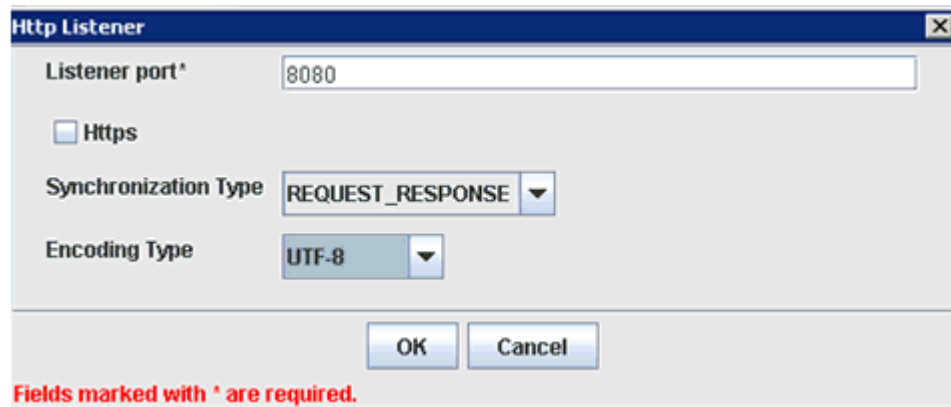


Provide the following information:

- a. Enter a name for the channel, for example, **PSFT\_channel**.
  - b. Enter a brief description.
  - c. From the **Protocol** list, select **HTTP Listener**.
4. Click **Next**.

The **Http Listener** dialog is displayed, as shown in [Figure 2–20](#).

**Figure 2–20 Http Listener Dialog**



5. Enter the system information as specified in the following table:



Parameter	Description
Listener port	Port on which to listen for PeopleSoft event data.
Https	For a secure HTTP connection, select the <b>Https</b> check box. This option is currently not supported.
Synchronization Type	Choose from the following synchronization options: <ul style="list-style-type: none"> <li>■ REQUEST_RESPONSE</li> <li>■ REQUEST_ACK</li> </ul> <p><b>Important:</b> The PeopleSoft channel does not work if the synchronization type is set to REQUEST.</p>
Encoding Type	Choose an encoding type to be used from the list. By default, ASCII is selected.

**6. Click OK.**

A summary pane is displayed, providing the channel description, channel status, and available ports. All the information is associated with the channel you created.

The channel appears under the channels node in the left pane.

An X over the icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.

---

**Note:** If you are planning to integrate Oracle Application Adapter for PeopleSoft with BPM, BPEL, Mediator, or OSB inbound process components, then do not start the channel. The channel is managed by the run-time server after the BPM, BPEL, Mediator, or OSB process component is deployed. If you start the channel from Application Explorer for testing and debugging purposes, then stop it before run-time (when working with BPM, BPEL, Mediator, or OSB process components).

---

**7. Right-click the channel and select **Start**.**

The channel you created becomes active. The X over the icon in the left pane disappears.

**8. To stop the channel, right-click the channel and select **Stop**.**

### 2.9.1.2 Creating a TCP Channel

**1. Click the **Events** node.**

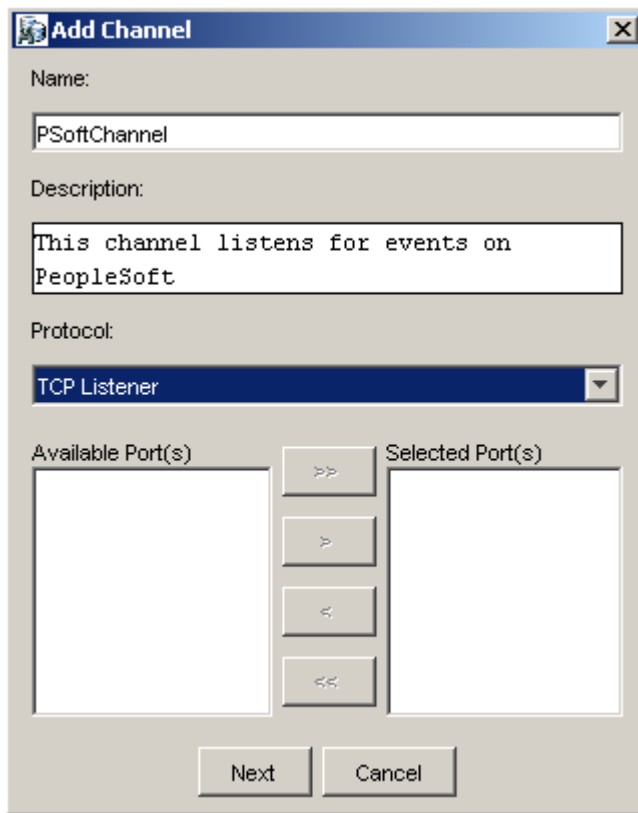
**2. Expand the **PeopleSoft** node.**

The ports and channels nodes appear in the left pane.

**3. Right-click **Channels** and select **Add Channel**.**

The Add Channel dialog is displayed, as shown in [Figure 2-21](#).

**Figure 2–21 Add Channel Dialog**

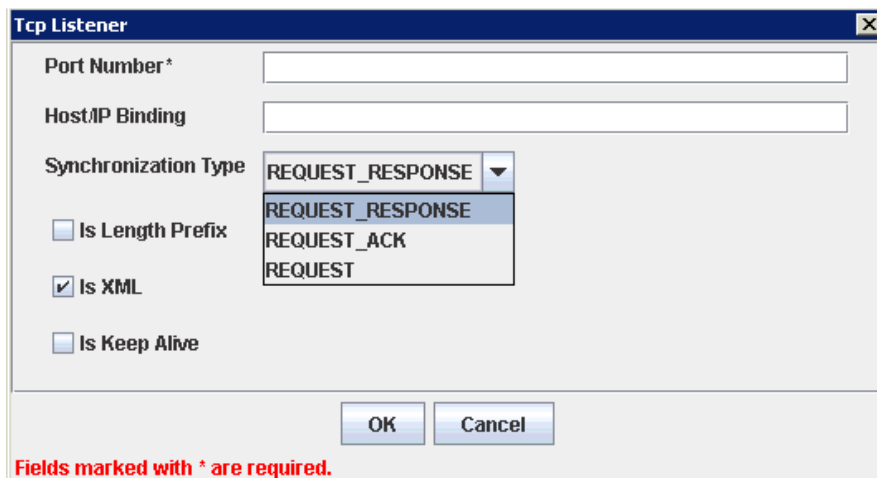


Perform the following steps:

- a. Enter a name for the channel, for example, PSoftChannel.
  - b. Enter a brief description.
  - c. From the **Protocol** list, select **TCP Listener**.
4. Click **Next**.

The Tcp Listener dialog is displayed, as shown in [Figure 2–22](#).

**Figure 2–22 TCP Listener Dialog**



Provide the following information:

- a. In the **Port Number** field, specify the port number for your TCP listener (required).
- b. In the **Host/IP Binding** field, specify the host IP for your TCP listener (optional).

---

**Note:** Channels can be configured and started only on the system where the Oracle Application Adapter for PeopleSoft is installed. Configuring and starting a channel for a remote host is not supported.

---

- c. From the **Synchronization Type** list, select from the following synchronization type options:

**REQUEST\_RESPONSE**

**REQUEST\_ACK**

**REQUEST**

**Important:** The PeopleSoft channel only works with one of these synchronization types.

- d. **Is Length Prefix** check box: For PeopleSoft events that return data that is not in XML format. The TCP/IP event application must prefix the data with a 4-byte binary length field when writing data to the TCP/IP port.
  - e. **Is XML** check box: For PeopleSoft events that return data in XML format. No preparser is required.
  - f. **Is Keep Alive** check box: Maintains continuous communication between the event transaction and the channel.
5. Click **OK**.

The channel appears under the Channels node in the left pane, as shown in [Figure 2–23](#).

**Figure 2–23 Inactive PeopleSoft Channel Node**



An X over the icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.

---

**Note:** If you are planning to integrate Oracle Application Adapter for PeopleSoft with BPM, BPEL, OSB inbound or Mediator inbound process components, then do not start the channel. The channel is managed by the run-time server after the BPM, BPEL, OSB, or Mediator process component is deployed. If you start the channel from Application Explorer for testing and debugging purposes, then stop it before run-time (when working with BPM, BPEL, OSB, or Mediator process components).

---

6. Right-click the channel node and select **Start**.  
The channel becomes active, as shown in [Figure 2–24](#).

**Figure 2–24 Active PeopleSoft Channel Node**



The X over the icon disappears.

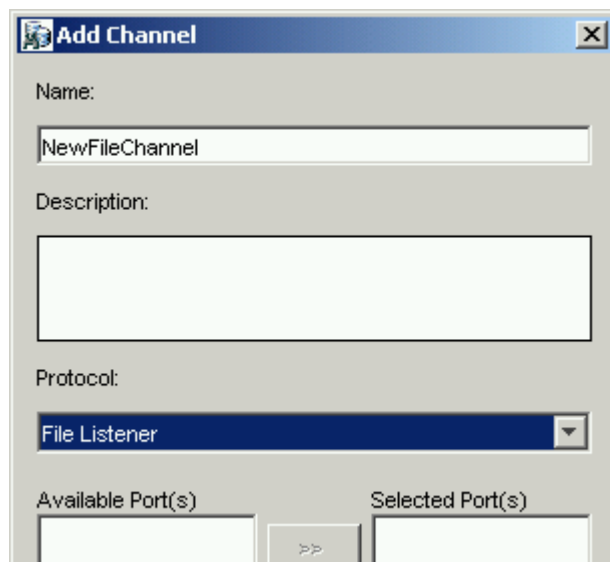
7. To stop the channel, right-click the connected channel node and select **Stop**.  
The channel becomes inactive and an X appears over the icon.

### 2.9.1.3 Creating a File Channel

To create a channel for the File listener:

1. Click the **Events** node.
2. In the left pane, expand the **PeopleSoft** node.  
The ports and channels nodes appear.
3. Right-click **Channels** and select **Add Channel**.  
The Add Channel dialog is displayed, as shown in [Figure 2–25](#).

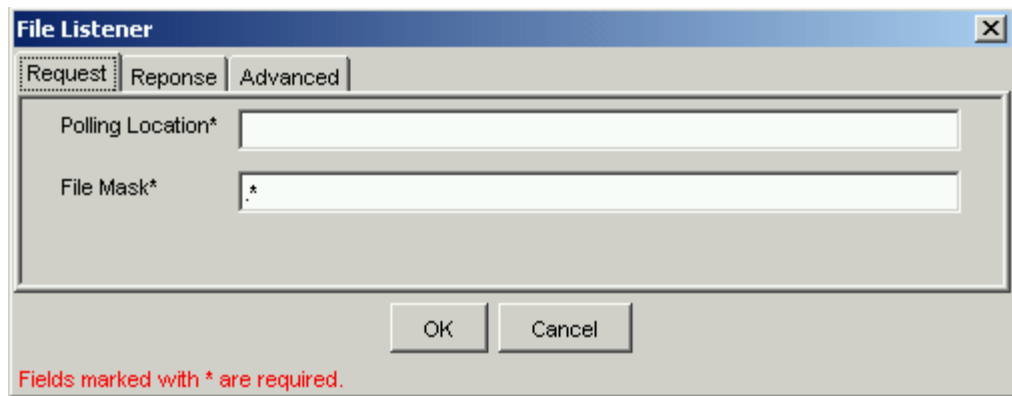
**Figure 2–25 Add Channel Dialog**



Perform the following steps:

- a. Enter a name for the channel, for example, **NewFileChannel**.
- b. Enter a brief description (optional).
- c. From the **Protocol** list, select **File Listener**.
4. Click **Next**.
5. The File Listener dialog is displayed, as shown in [Figure 2–26](#).

Figure 2–26 File Listener Dialog



Enter the system information as follows:

- a. In the **Request** tab, enter values for the following parameters:

Parameter	Description
Polling Location	Target file system location for the PeopleSoft XML file.
File Mask	File name to be used for the output file generated by the operation.

- b. In the **Response** tab, enter values for the following parameters:

Parameter	Description
Synchronization Type	Choose from the following synchronization options: <ul style="list-style-type: none"> <li>▪ REQUEST_RESPONSE</li> <li>▪ REQUEST_ACK</li> </ul> <p><b>Important:</b> The PeopleSoft channel does not work if the synchronization type is set to REQUEST.</p>
Response/Ack Directory	Target file system location for the PeopleSoft XML file.

- c. In the **Advanced** tab, enter values for the following parameters:

Parameter	Description
Error directory	Directory to which documents with errors are written.
Poll interval (msec)	Interval (in milliseconds) when to check for new input. The default is three seconds. Optional.
Processing Mode	<b>Sequential</b> indicates single processing of requests. <b>Threaded</b> indicates processing of multiple requests simultaneously.
Thread limit	If you selected threaded processing, then indicate the maximum number of requests that can be processed simultaneously.

6. Click **OK**.

A summary pane is displayed, providing the channel description and channel status. All the information is associated with the channel you created.

The channel appears under the channels node in the left pane. An X over the icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.

---

---

**Note:** If you are planning to integrate Oracle Application Adapter for PeopleSoft with BPEL inbound or Mediator inbound process components, then do not start the channel. The channel is managed by the run-time server after the BPEL or Mediator process component is deployed. If you start the channel from Application Explorer for testing and debugging purposes, then stop it before run-time (when working with BPEL or Mediator process components).

---

---

7. Right-click the channel and select **Start**.

The channel you created becomes active. The X over the icon in the left pane disappears.

8. To stop the channel, right-click the channel and select **Stop**.

#### 2.9.1.4 Editing a Channel

You must stop the channel before you can edit it. When you edit the channel, you must also restart the Oracle WebLogic Server to update the repository for run time purposes. To edit a channel:

1. In the left pane, locate the channel you want to edit.
2. Right-click the channel and select **Edit**.

The Edit channels pane is displayed.

3. Make the required changes to the channel configuration and click **Finish**.

#### 2.9.1.5 Deleting a Channel

You must stop the channel before you can delete it. When you delete the channel, you must also restart the Oracle WebLogic Server to update the repository for run time purposes. To delete a channel:

1. In the left pane, locate the channel you want to delete.
2. Right-click the channel and select **Delete**.

The channel disappears from the list in the left pane.

## 2.9.2 Schema Validation

Root validation, namespace validation, and schema validation for inbound processing (events) are supported for the Oracle Application Adapter for PeopleSoft.

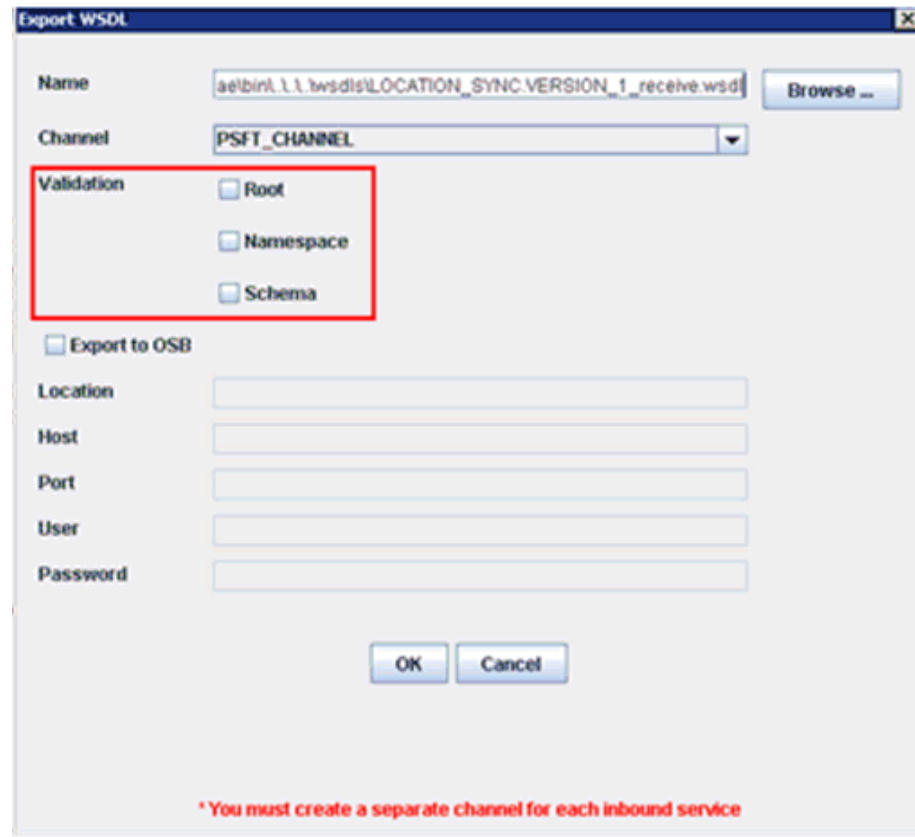
To validate inbound processing using the Oracle Application Adapter for PeopleSoft, perform the following steps. This procedure uses the LOCATION\_SYNC.VERSION\_1 PeopleSoft Message as an example for inbound processing.

1. Start Application Explorer.
2. Connect to the PeopleSoft target.
3. Expand the **Messages** node.
4. Verify that you have already created a channel for the PeopleSoft adapter.

5. Right-click the `LOCATION_SYNC.VERSION_1` node and select **Create Inbound JCA Service (event)**.

The Export WSDL dialog opens and includes three check boxes for Root, Namespace, and Schema validation, as shown in [Figure 2–27](#).

**Figure 2–27** *Export WSDL Dialog*



- Selection of multiple validation options is allowed.
- Root validation is used to validate the root element in the inbound XML document.
- Namespace validation is used to validate the namespace in the inbound XML document.
- Schema validation is used to validate the inbound XML document with the schema in the WSDL document.
- During run time, validation is processed based on the validation options that are selected.
- If more than one validation option is selected, during run time if the first validation option fails, then the remaining validation options are not processed.
- Root and namespace validations are considered modest levels of validation. Schema validation is a stricter validation level.
- It is recommended to use root and namespace validation options together, unless the root element and namespace are different between the Messages in the PeopleSoft environment.

6. Generate the WSDL document and create the BPEL, Mediator, BPM, or OSB process.
7. Trigger the transactions (Messages) from the PeopleTools environment.



---

---

# Oracle WebLogic Server Deployment and Integration

This chapter describes Oracle WebLogic Server (OracleWLS) deployment and integration with Oracle Application Adapter for PeopleSoft.

This chapter contains the following topics:

- [Section 3.1, "Adapter Integration with Oracle WebLogic Server"](#)
- [Section 3.2, "Deployment of Adapter"](#)
- [Section 3.3, "Updating Adapter Configuration"](#)

**See Also:**

- *Oracle Application Server Adapter Concepts Guide*

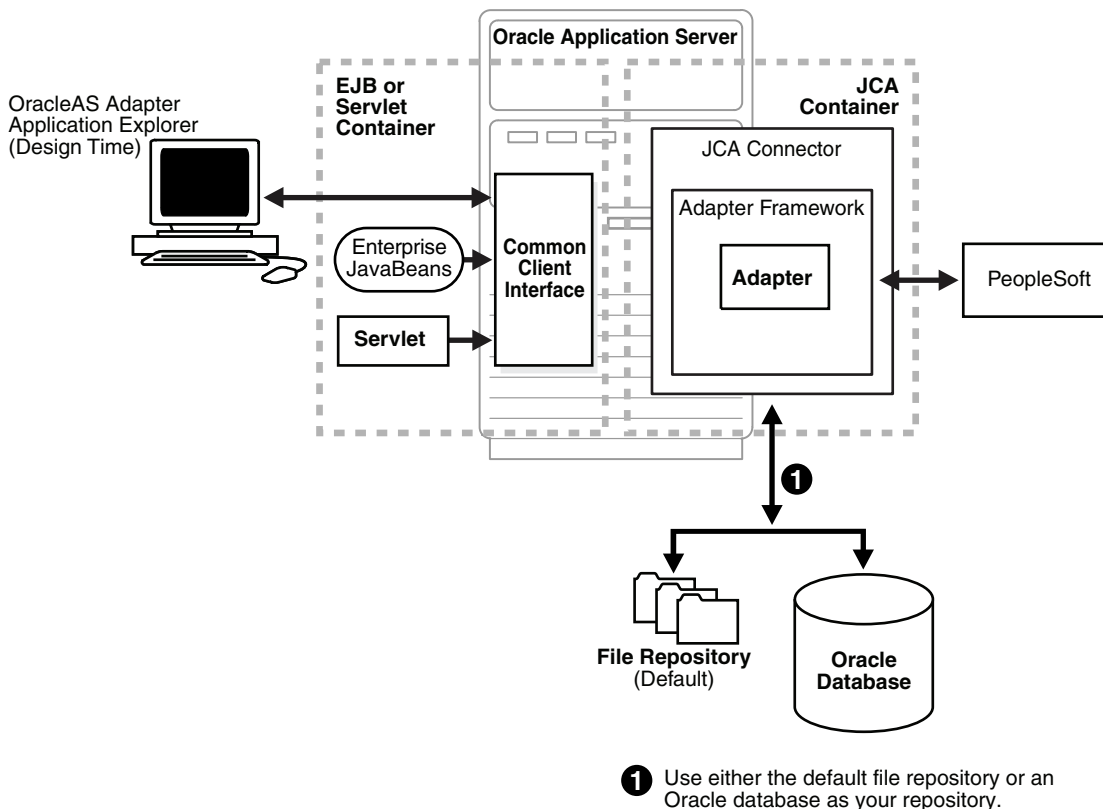
## 3.1 Adapter Integration with Oracle WebLogic Server

Oracle Application Adapter for PeopleSoft is deployed within an OracleWLS container during installation. All client applications run within the OracleWLS environment. In J2CA deployment, the Common Client Interface (CCI) integrates an OracleWLS client application with a resource adapter.

## 3.2 Deployment of Adapter

[Figure 3–1](#) shows deployment of the J2CA Connector to the Oracle Application Server. In a run-time service scenario, an Enterprise Java Bean, servlet, or Java program client makes CCI calls to J2CA resource adapters. The adapters process the calls as requests and send them to the EIS. The EIS response is then sent back to the client.

Figure 3–1 Oracle Application Server J2CA Architecture



### 3.3 Updating Adapter Configuration

During the J2CA deployment of Oracle Application Adapter for PeopleSoft, OracleWLS generates a deployment descriptor called `ra.xml`, located in:

```
<ADAPTER_HOME>\iwafjca.rar\META-INF
```

Your installation contains more than one file named `ra.xml`. The OracleWLS deployment descriptor that is described in this section is located in the directory specified above.

---

**Note:** Multiple managed connection factories are supported only for outbound processing (services).

---

This topic contains the following sections:

- [Section 3.3.1, "Creating a Managed Connector Factory Object"](#)
- [Section 3.3.2, "Creating Multiple Managed Connector Factory Objects"](#)
- [Section 3.3.3, "Modifying WSDL Files for Additional Connection Factory Values"](#)

#### 3.3.1 Creating a Managed Connector Factory Object

The `ra.xml` descriptor provides OracleWLS-specific deployment information for resource adapters. For example, the `jca_sample` configuration in Application Explorer is represented in the `ra.xml` file as follows:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE connector PUBLIC "-//Sun Microsystems, Inc.//DTD Connector 1.0//EN"
'http://java.sun.com/dtd/connector_1_0.dtd'>
<connector>
  <display-name>IWAYFJCA10</display-name>
  <vendor-name>IWAY Software</vendor-name>
  <spec-version>1.0</spec-version>
  <eis-type>IWAY</eis-type>
  <version>1.0</version>
  <license>
    <license-required>>false</license-required>
  </license>
  <resourceadapter>

  <managedconnectionfactory-class>com.ibi.afjca.spi.IWAYManagedConnectionFactory</ma
nagedconnectionfactory-class>

  <connectionfactory-interface>javax.resource.cci.ConnectionFactory</connectionfacto
ry-interface>

  <connectionfactory-impl-class>com.ibi.afjca.cci.IWAYConnectionFactory</connectionf
actory-impl-class>
    <connection-interface>javax.resource.cci.Connection</connection-interface>

  <connection-impl-class>com.ibi.afjca.cci.IWAYConnection</connection-impl-class>
    <transaction-support>NoTransaction</transaction-support>
    <config-property>
      <config-property-name>AdapterName</config-property-name>
      <config-property-type>java.lang.String</config-property-type>
      <config-property-value></config-property-value>
    </config-property>
    <config-property>
      <config-property-name>Config</config-property-name>
      <config-property-type>java.lang.String</config-property-type>
      <config-property-value></config-property-value>
    </config-property>
    <config-property>
      <config-property-name>IWayHome</config-property-name>
      <config-property-type>java.lang.String</config-property-type>
      <config-property-value>C:\oracle\Middleware\Oracle_
SOA1\soa\thirdparty\ApplicationAdapters</config-property-value>
    </config-property>
    <config-property>
      <config-property-name>IWayConfig</config-property-name>
      <config-property-type>java.lang.String</config-property-type>
      <config-property-value>jca_sample</config-property-value>
    </config-property>
    <config-property>
      <config-property-name>IWayRepoDriver</config-property-name>
      <config-property-type>java.lang.String</config-property-type>
      <config-property-value></config-property-value>
    </config-property>
    <config-property>
      <config-property-name>IWayRepoURL</config-property-name>
      <config-property-type>java.lang.String</config-property-type>
      <config-property-value></config-property-value>
    </config-property>
    <config-property>
      <config-property-name>IWayRepoUser</config-property-name>
      <config-property-type>java.lang.String</config-property-type>

```

```

        <config-property-value></config-property-value>
    </config-property>
    <config-property>
        <config-property-name>IWayRepoPassword</config-property-name>
        <config-property-type>java.lang.String</config-property-type>
        <config-property-value></config-property-value>
    </config-property>
    <config-property>
        <config-property-name>LogLevel</config-property-name>
        <config-property-type>java.lang.String</config-property-type>
        <config-property-value>DEBUG</config-property-value>
    </config-property>
    <authentication-mechanism>
        <authentication-mechanism-type>BasicPassword</authentication-mechanism-type>

    <credential-interface>javax.resource.spi.security.PasswordCredential</credential-i
nterface>
    </authentication-mechanism>
    <reauthentication-support>true</reauthentication-support>
    </resourceadapter>
</connector>

```

The parameters defined in the ra.xml file are described in the following table:

Parameter Name	Description
IWayHome	The base installation directory for the OracleWLS packaged application adapter.
IWayConfig	The adapter configuration name as defined in Application Explorer. For example, Oracle Application Adapter for PeopleSoft has a preconfigured <code>jca_sample</code> configuration in Application Explorer.
IWayRepoURL	The URL to use when opening a connection to the database. This is necessary only when using an Oracle database as the repository.
IWayRepoUser	User name to use when connecting to the database. This is necessary only when using an Oracle database as the repository.
IWayRepoPassword	Password. If provided, it overwrites configuration. This is necessary only when using an Oracle database as the repository.
loglevel	It overwrites the level set by the <code>ManagedConnectionFactory</code> property.

### 3.3.2 Creating Multiple Managed Connector Factory Objects

To establish multiple managed connector factory objects, you must edit the `weblogic-ra.xml` file and add more `<connection-instance>` nodes. This file is located in:

```
<ADAPTER_HOME>\iwafjca.rar\META-INF
```

For example, the first `jca_configuration` in Application Explorer is represented in the `weblogic-ra.xml` file as follows:

```

<?xml version="1.0"?>
<weblogic-connector xmlns="http://www.bea.com/ns/weblogic/90">

```

```

<enable-access-outside-app>true</enable-access-outside-app>
<enable-global-access-to-classes>true</enable-global-access-to-classes>
<outbound-resource-adapter>
  <default-connection-properties>
    <pool-params>
      <initial-capacity>0</initial-capacity>
    </pool-params>
    <transaction-support>LocalTransaction</transaction-support>
  </default-connection-properties>
  <connection-definition-group>

<connection-factory-interface>javax.resource.cci.ConnectionFactory</connection-factory-interface>
  <connection-instance>
    <jndi-name>eis/OracleJCAAdapter/DefaultConnection</jndi-name>
  </connection-instance>
</connection-definition-group>
</outbound-resource-adapter>
</weblogic-connector>

```

To create multiple managed connector factory objects, you must add new `<connection-instance>` nodes in the file. For example:

```

<?xml version="1.0"?>
<weblogic-connector xmlns="http://www.bea.com/ns/weblogic/90">

  <enable-access-outside-app>true</enable-access-outside-app>
  <enable-global-access-to-classes>true</enable-global-access-to-classes>

  <outbound-resource-adapter>
    <default-connection-properties>
      <pool-params>
        <initial-capacity>0</initial-capacity>
      </pool-params>
      <transaction-support>LocalTransaction</transaction-support>
    </default-connection-properties>
    <connection-definition-group>

<connection-factory-interface>javax.resource.cci.ConnectionFactory</connection-factory-interface>
  <connection-instance>
    <jndi-name>eis/OracleJCAAdapter/DefaultConnection</jndi-name>
  </connection-instance>
  <connection-instance>
    <jndi-name>eis/OracleJCAAdapter/DefaultConnection1</jndi-name>
    <connection-properties>
      <properties>
        <property>
<name>IWayHome</name>
<value>C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters</value>
        </property>
        <property>
<name>IWayConfig</name>
<value>jca_sample2</value>
        </property>
        <property>
<name>IWayRepoURL</name>
<value></value>
        </property>
        <property>
<name>IWayRepoUser</name>

```

```

<value></value>
  </property>
  <property>
<name>IWayRepoPassword</name>
<value></value>
  </property>
  <property>
<name>LogLevel</name>
<value>Debug</value>
  </property>
</properties>
</connection-properties>
</connection-instance>
</connection-definition-group>
</outbound-resource-adapter>
</weblogic-connector>

```

If you do not specify a `<property>` element in the `<connection-instance>` section, the value is taken from the `ra.xml` file. You can specify the default properties in the `ra.xml` file and then override them as required in the `weblogic-ra.xml` file. In addition, note that the J2CA configuration (for example, `jca_sample2`) must already be created in Application Explorer.

---



---

**Note:** When you modify the `ra.xml` and `weblogic-ra.xml` files, the application server must be restarted. If the application server is already running, stop the application server and then restart it.

In addition, the `iwafjca.rar` file must be redeployed in the Oracle WebLogic Administration Console to activate these changes.

---



---

### 3.3.3 Modifying WSDL Files for Additional Connection Factory Values

Application Explorer generates the J2CA properties file using the default connection factory name `eis/OracleJCAAdapter/DefaultConnection`. If you created additional connection factories, the WSDLs generated for the additional configuration and connection factory should be changed to reflect the location field of the `jca:address` section in the J2CA properties file. The default J2CA properties file for the Oracle Application Adapter for PeopleSoft with a configuration of `isdsrv2_conn2` is shown in the following example.

Notice that the J2CA properties file has the following default connection factory:  
`eis/OracleJCAAdapter/DefaultConnection`

```

<jca:address location="eis/OracleJCAAdapter/DefaultConnection"
  ConnectionSpec="com.ibi.afjca.cci.IWAFConnectionSpec"
  cs.AdapterName="PeopleSoft" cs.Config="isdsrv2_conn2"
  UIConnectionName="Connection1"/>

```

The connection factory value must be changed to the following:  
`eis/OracleJCAAdapter/DefaultConnection1`

For example:

```

<jca:address location="eis/OracleJCAAdapter/DefaultConnection1"
  ConnectionSpec="com.ibi.afjca.cci.IWAFConnectionSpec"
  cs.AdapterName="PeopleSoft" cs.Config="isdsrv2_conn2"
  UIConnectionName="Connection1"/>

```

Note that only the value for the location field in the `jca:address` section should be modified. Do not modify any other field or section.





---

---

## Integration With BPEL Service Components in the Oracle SOA Suite

Oracle Application Adapter for PeopleSoft integrates seamlessly with Oracle Business Process Execution Language (BPEL) Process Manager to facilitate Web service integration. Oracle BPEL Process Manager is based on the Service-Oriented Architecture (SOA). It consumes adapter services exposed as Web Service Definition Language (WSDL) documents.

This chapter contains the following topics:

- [Section 4.1, "Overview"](#)
- [Section 4.2, "Deployment of Adapter"](#)
- [Section 4.3, "Configuring a New Application Server Connection"](#)
- [Section 4.4, "Designing an Outbound BPEL Process for Service Integration \(J2CA Configuration\)"](#)
- [Section 4.5, "Designing an Inbound BPEL Process for Event Integration \(J2CA Configuration\)"](#)
- [Section 4.6, "Designing an Outbound BPEL Process for Service Integration \(BSE Configuration\)"](#)

### 4.1 Overview

To integrate with Oracle SOA Suite, Oracle Application Adapter for PeopleSoft must be deployed in the same WLS container as Oracle BPEL Process Manager. The underlying adapter services must be exposed as WSDL files, which are generated during design time in Oracle Application Adapter Application Explorer (Application Explorer) for both request-response (outbound) and event notification (inbound) services of the adapter. For more information, see [Chapter 2, "Configuring Oracle Application Adapter for PeopleSoft"](#).

The generated WSDL files are used to design the appropriate BPEL processes for inbound or outbound adapter services. A completed BPEL process must be successfully compiled in JDeveloper and deployed to a SOA server. Upon deployment to the SOA server, every newly built process is automatically deployed to the Oracle Enterprise Manager Console, where you run, monitor, and administer BPEL processes, and listen to adapter events.

## 4.2 Deployment of Adapter

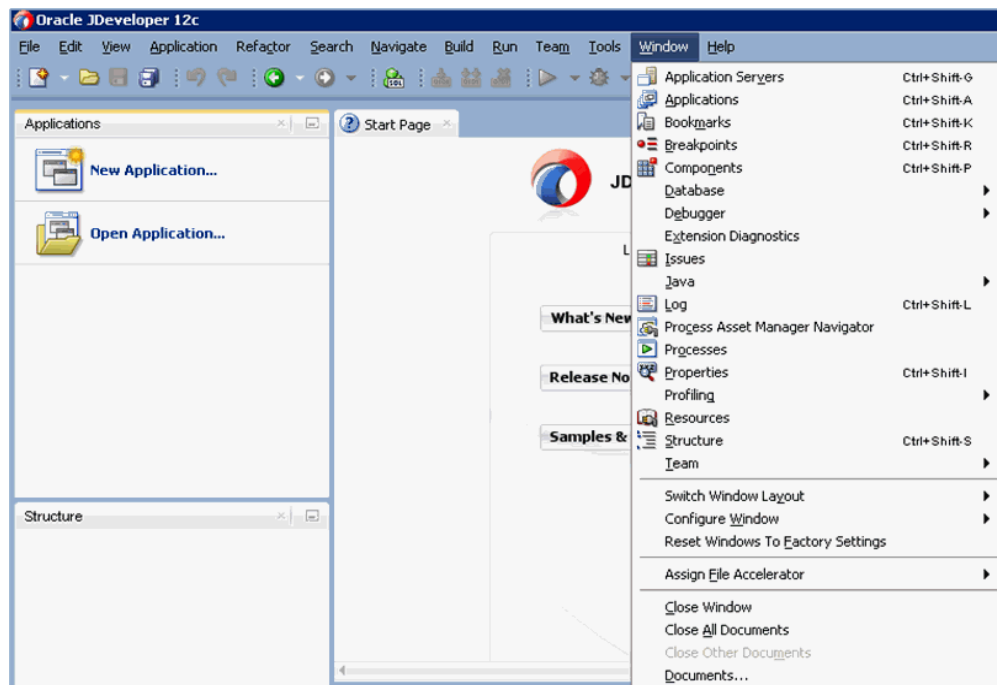
During installation, Oracle Application Adapter for PeopleSoft is deployed as a J2CA 1.0 resource adapter within the WLS container. The adapter must be deployed in the same WLS container as Oracle BPEL Process Manager.

## 4.3 Configuring a New Application Server Connection

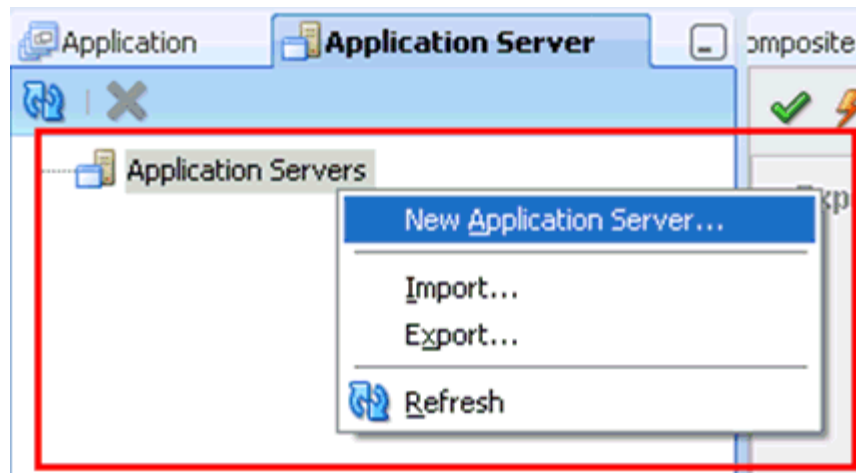
To configure a new Application Server connection in Oracle JDeveloper:

1. Open **Oracle JDeveloper** on your system.
2. From the menu bar, click **Window** and select **Application Server Navigator**, as shown in [Figure 4-1](#).

**Figure 4-1 Application Server Navigator**

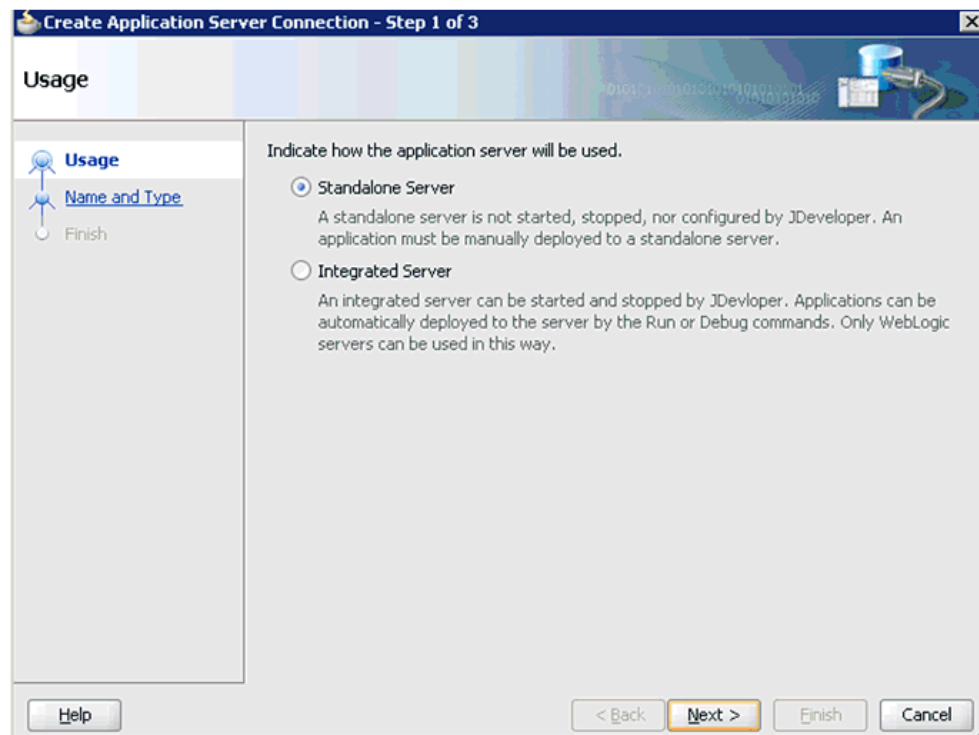


The Application Server tab is displayed, as shown in [Figure 4-2](#).

**Figure 4–2 Application Server Tab**

3. Right-click **Application Servers** and select **New Application Server**.

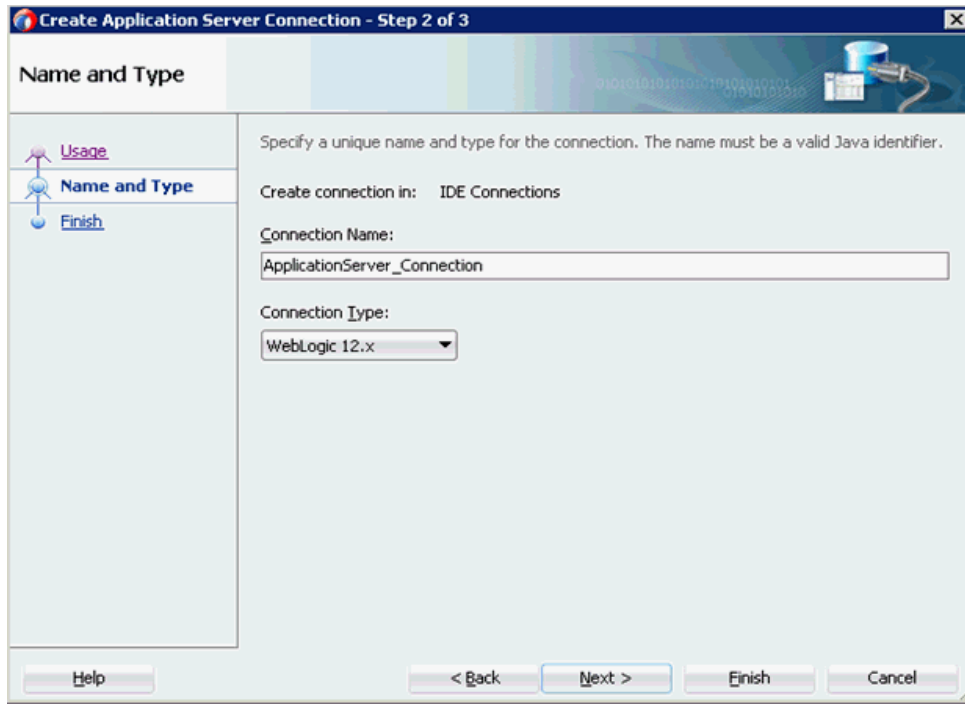
The Create Application Server Connection Wizard is displayed, as shown in [Figure 4–3](#).

**Figure 4–3 Create Application Server Connection Wizard**

4. Accept the default selection (Standalone Server) and click **Next**.

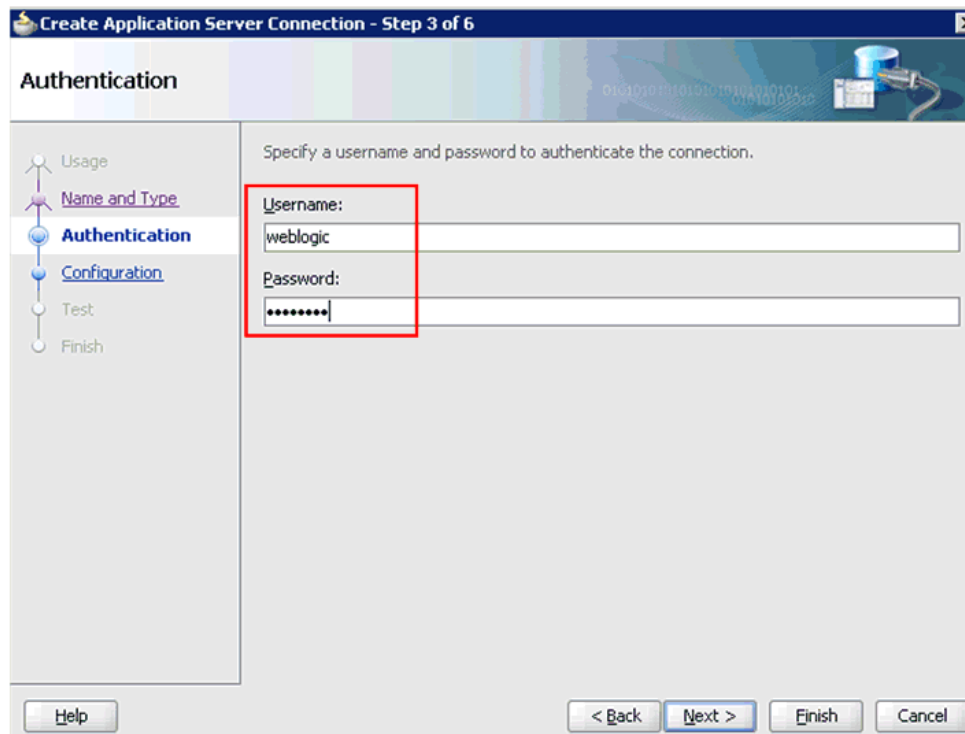
The Name and Type page is displayed, as shown in [Figure 4–4](#).

**Figure 4–4 Name and Type Page**



5. Specify a new name for the Application Server connection and click **Next**.  
The Authentication page is displayed, as shown in [Figure 4-5](#).

**Figure 4–5 Authentication Page**



6. Specify a valid user name (for example, weblogic) and a password (for example, welcome1) for your new connection.
7. Click **Next**.

The Configuration page is displayed, as shown in [Figure 4-6](#).

**Figure 4-6 Configuration Page**

**Create Application Server Connection - Step 4 of 6**

**Configuration**

Usage  
Name and Type  
Authentication  
**Configuration**  
Test  
Finish

WebLogic Server connections use a host name and port to establish a connection. The Domain of the target will be verified

WebLogic Hostname (Administration Server):  
localhost

Port: 7001 SSL Port: 7002

Always use SSL

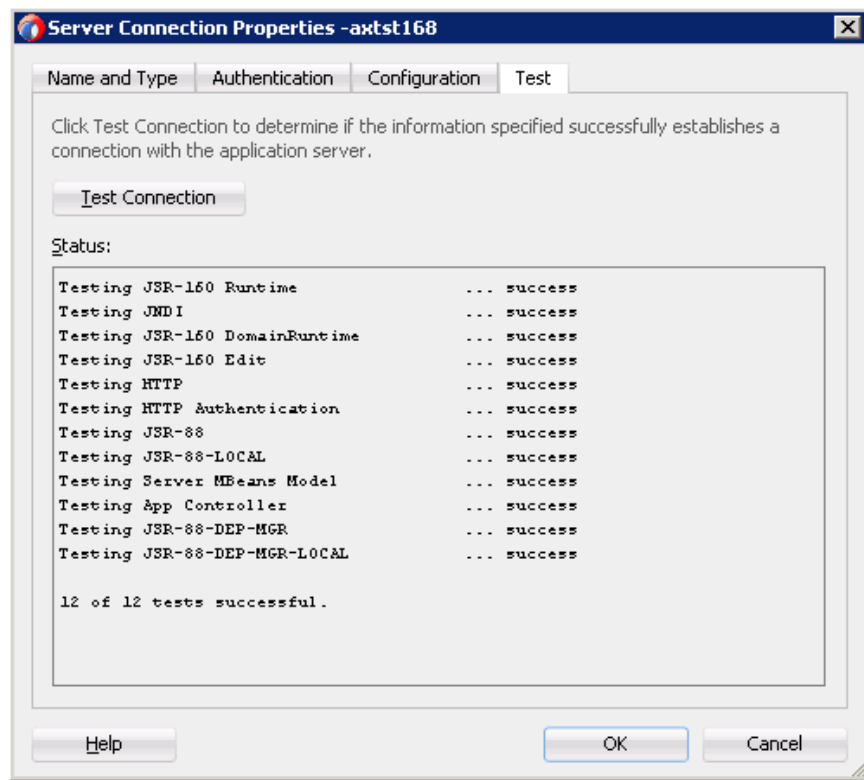
WebLogic Domain:  
base\_domain

Help < Back Next > Finish Cancel

8. Specify the Oracle WebLogic host name (for example, localhost), which is the system IP where the process must deploy and Oracle WebLogic domain (for example, base\_domain).
9. Click **Next**.

The Test page is displayed, as shown in [Figure 4-7](#).

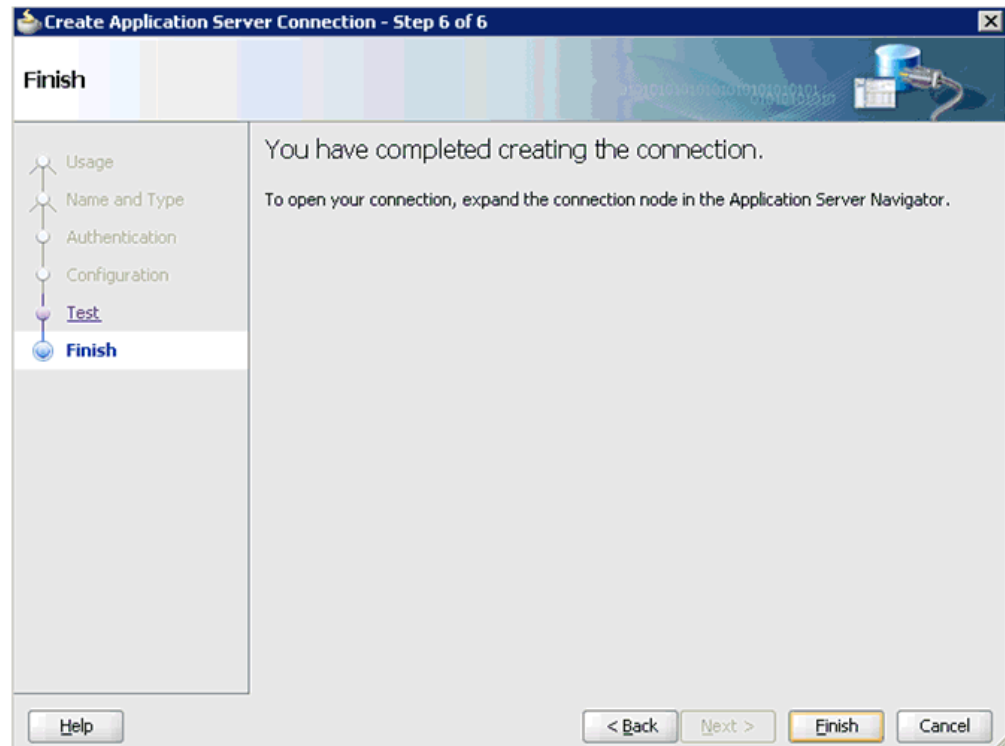
**Figure 4–7 Test Page**



10. Click **Test Connection**.
11. Make sure that the test status is successful.
12. Click **Next**.

The Finish page is displayed, as shown in [Figure 4–8](#).

Figure 4–8 Finish Page



13. Click **Finish**.

The new Application Server connection is listed in the left pane (Application Server tab).

## 4.4 Designing an Outbound BPEL Process for Service Integration (J2CA Configuration)

This section demonstrates how to design an outbound BPEL process for service integration.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\BPEL\J2CA\Outbound_Project
```

The following tools are required to complete your adapter design-time configuration:

- Oracle Adapter Application Explorer (Application Explorer)
- Oracle JDeveloper BPEL Designer (JDeveloper)

---

**Note:** The examples in this chapter demonstrate the use of JDeveloper.

---

Before you design a BPEL process, you must create a schema and generate the respective WSDL file using Application Explorer. For more information, see "[Generating WSDL for Request/Response Service](#)" on page 4-8.

This section contains the following topics:

- [Section 4.4.1, "Generating WSDL for Request/Response Service"](#)
- [Section 4.4.2, "Creating an Empty Composite for SOA"](#)
- [Section 4.4.3, "Defining a BPEL Outbound Process"](#)
- [Section 4.4.4, "Deploying the BPEL Outbound Process"](#)
- [Section 4.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console"](#)
- [Section 4.4.6, "Testing Outbound BPEL and Mediator Processes"](#)

### 4.4.1 Generating WSDL for Request/Response Service

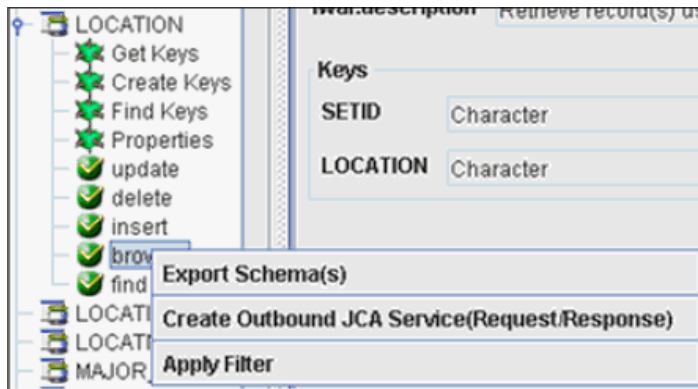
To generate WSDL for outbound interaction in Application Explorer:

1. Start **Application Explorer** and connect to a defined PeopleSoft target or create a new target.

For more information, see ["Defining a Target to PeopleSoft"](#) on page 2-5.

2. Expand **Component Interfaces** and select **LOCATION**.
3. Navigate to and right-click an operation, and then select **Create Outbound JCA Service (Request/Response)**, as shown in [Figure 4-9](#).

**Figure 4-9 Create Outbound JCA Service (Request/Response) Option**



The Export WSDL dialog is displayed, as shown in [Figure 4-10](#).



**Figure 4–10 Export WSDL Dialog**

4. Specify an export location on your file system or accept the default path.  
The **.wsdl** file extension is added automatically. By default, the names of WSDL files generated for request-response services end with **\_invoke**.
5. Click **OK**.

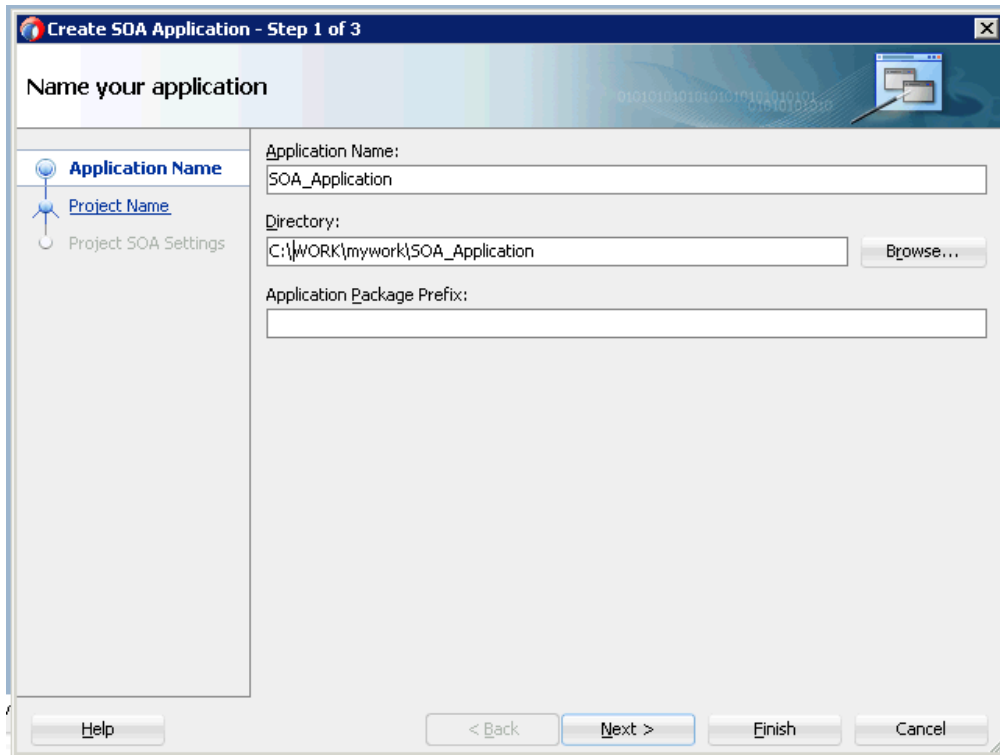
You can now create a new SOA application, which is the first step that is required to define a BPEL outbound process in JDeveloper.

#### 4.4.2 Creating an Empty Composite for SOA

Perform the following steps to create an empty composite for SOA:

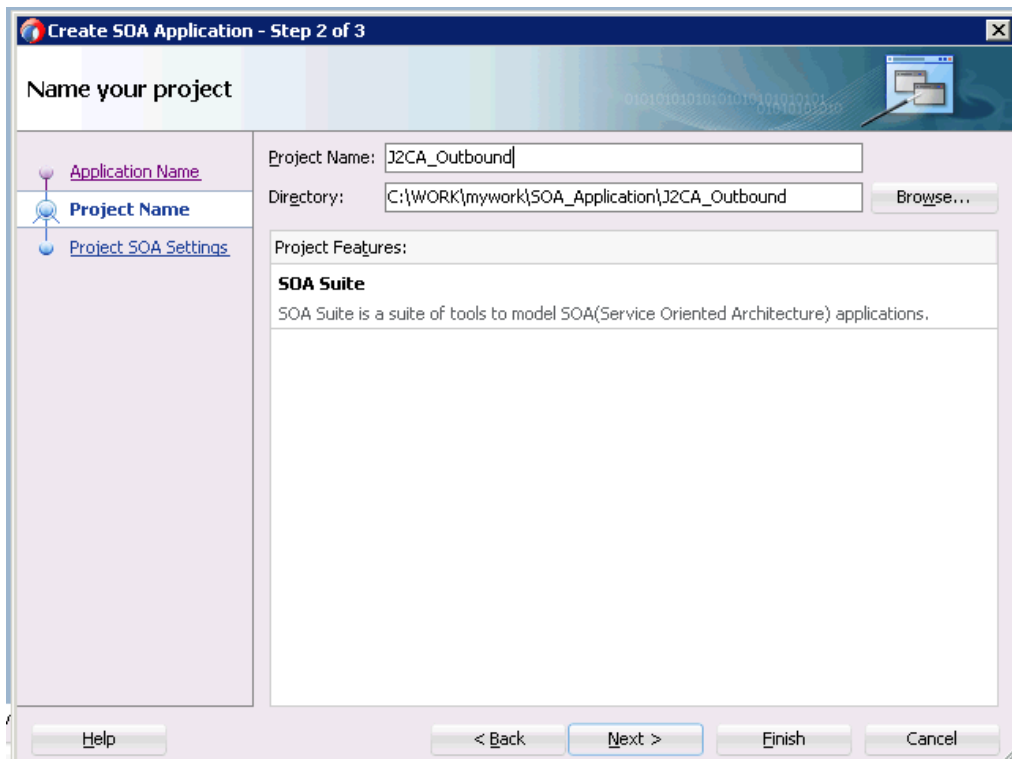
1. Create a new SOA application.
2. Enter a name for the new SOA Application and click **Next**, as shown in [Figure 4–11](#).

Figure 4–11 Name Your Application Page



The Name your project page is displayed, as shown in Figure 4–12.

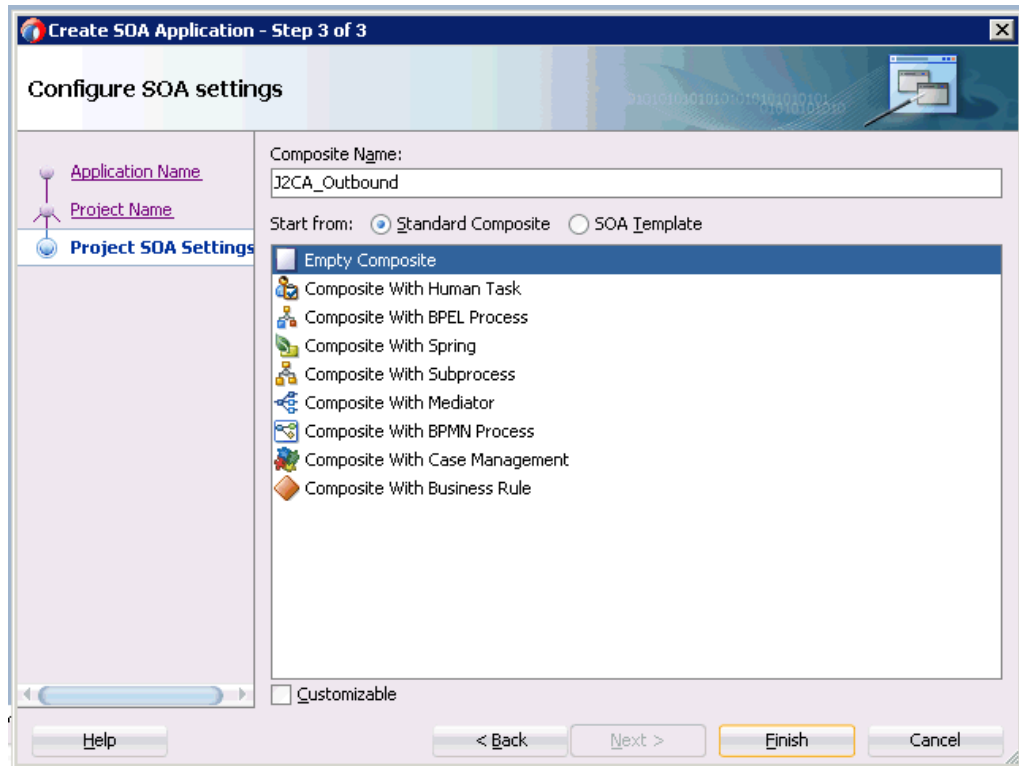
Figure 4–12 Name Your Project Page



3. Enter a project name and click **Next**.

The Configure SOA settings page is displayed, as shown in [Figure 4–13](#).

**Figure 4–13 Configure SOA Settings Page**



4. From the Composite Template list, select **Empty Composite** and click **Finish**.

### 4.4.3 Defining a BPEL Outbound Process

This section describes how to define a BPEL outbound process, which consists of the following topics:

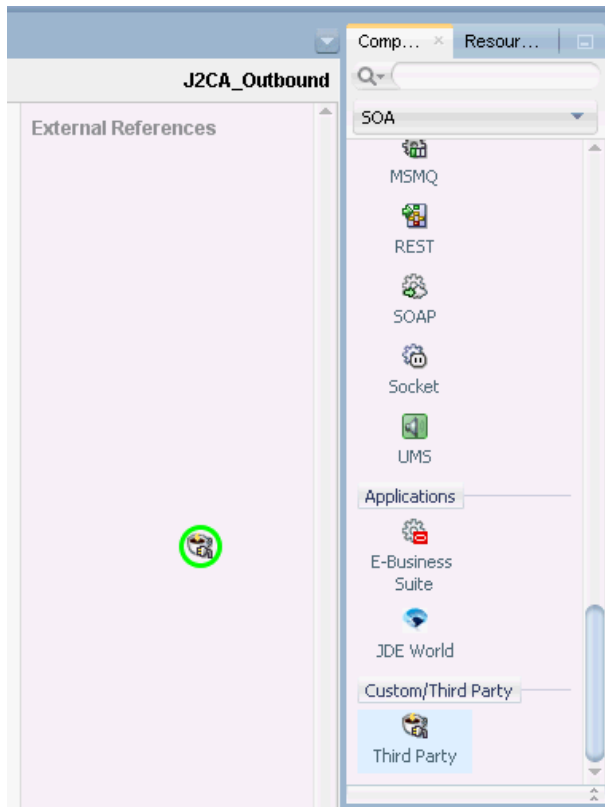
- [Section 4.4.3.1, "Configuring a Third Party Adapter Service Component"](#)
- [Section 4.4.3.2, "Configuring an Outbound BPEL Process Component"](#)
- [Section 4.4.3.3, "Adjusting for Known Deployment Issues With 12c"](#)

#### 4.4.3.1 Configuring a Third Party Adapter Service Component

Perform the following steps to create a third party adapter service component:

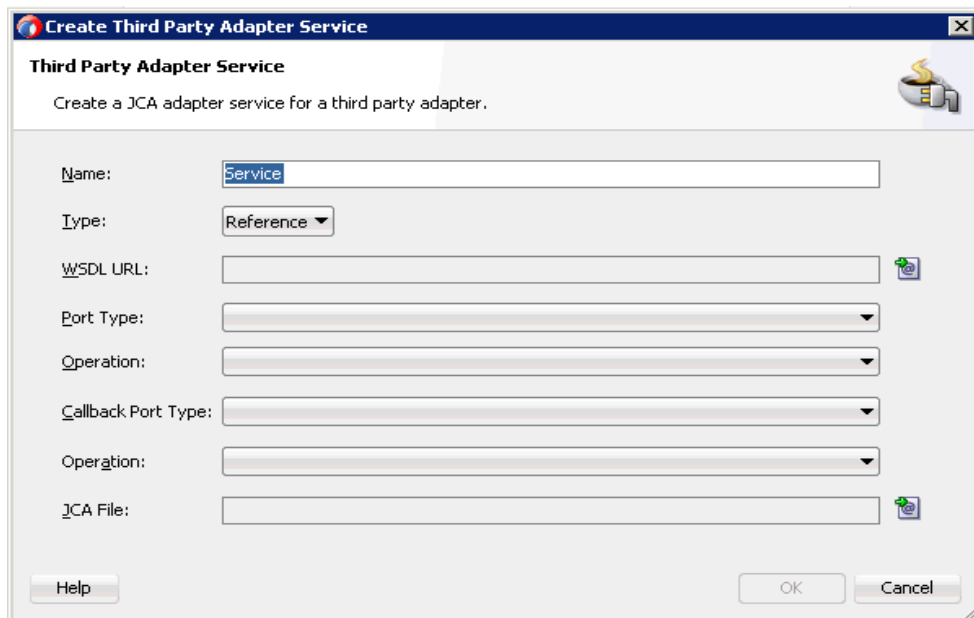
1. Drag and drop the **Third Party Adapter** component from the Service Adapters pane to the External References pane, as shown in [Figure 4–14](#).

**Figure 4–14 Third Party Adapter Component**



The Create Third Party Adapter Service dialog is displayed, as shown in [Figure 4–15](#).

**Figure 4–15 Create Third Party Adapter Service Dialog**

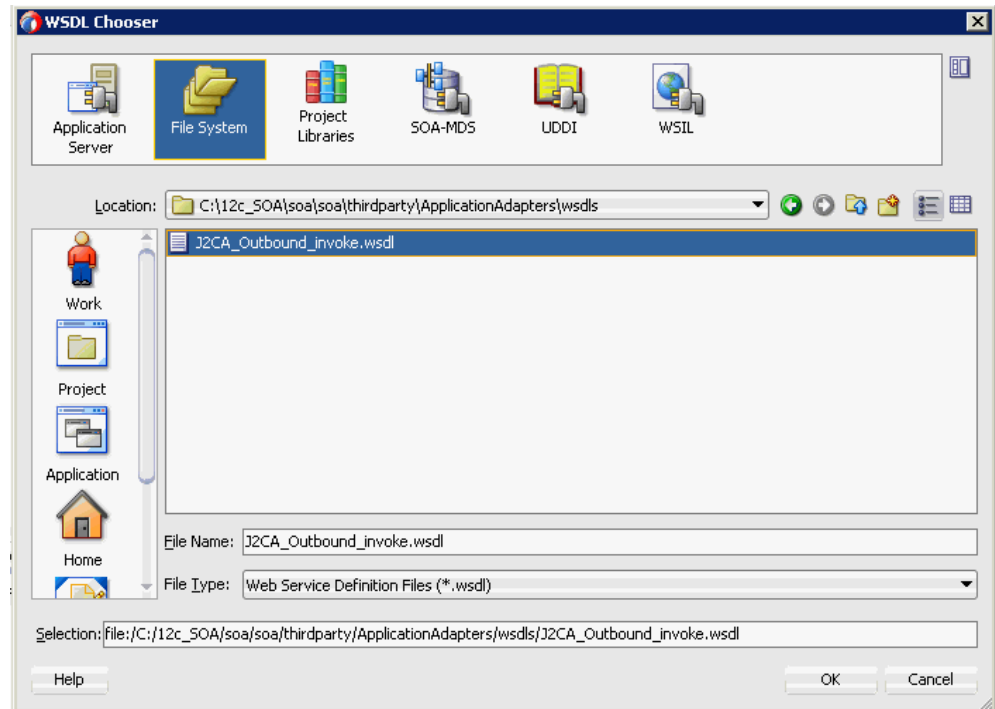


2. Ensure that **Reference** is selected from the Type list (default).

3. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.

The WSDL Chooser dialog is displayed, as shown in [Figure 4-16](#).

**Figure 4-16 WSDL Chooser Dialog**



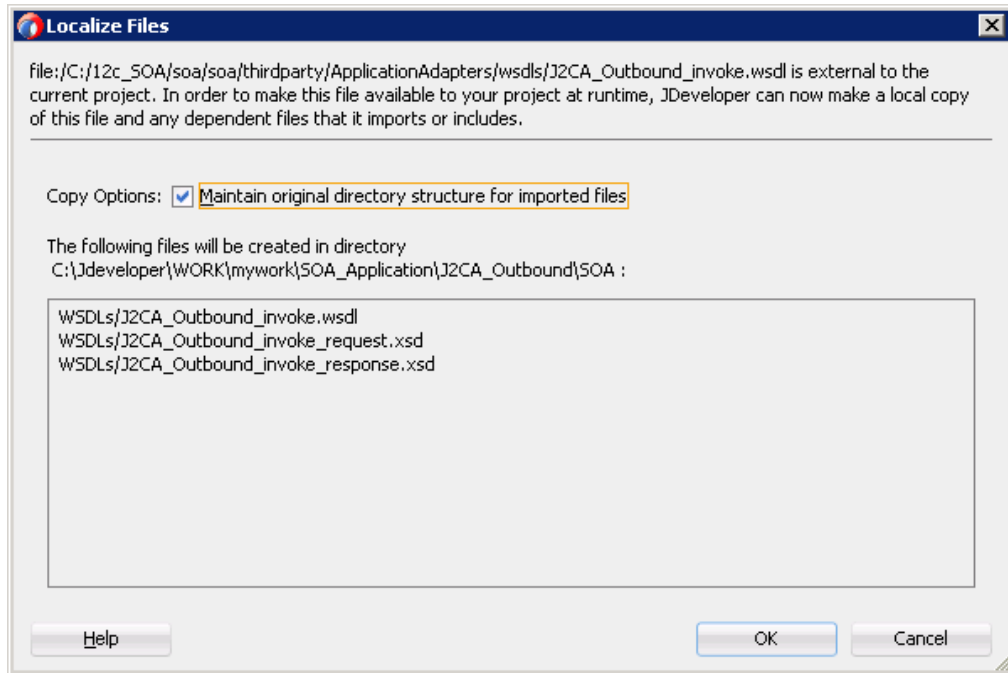
4. Browse and select an outbound WSDL file from the following directory:

<ADAPTER\_HOME>\wsdls

5. Click **OK**.

The Localize Files dialog is displayed, as shown in [Figure 4-17](#).

**Figure 4–17 Localize Files Dialog**

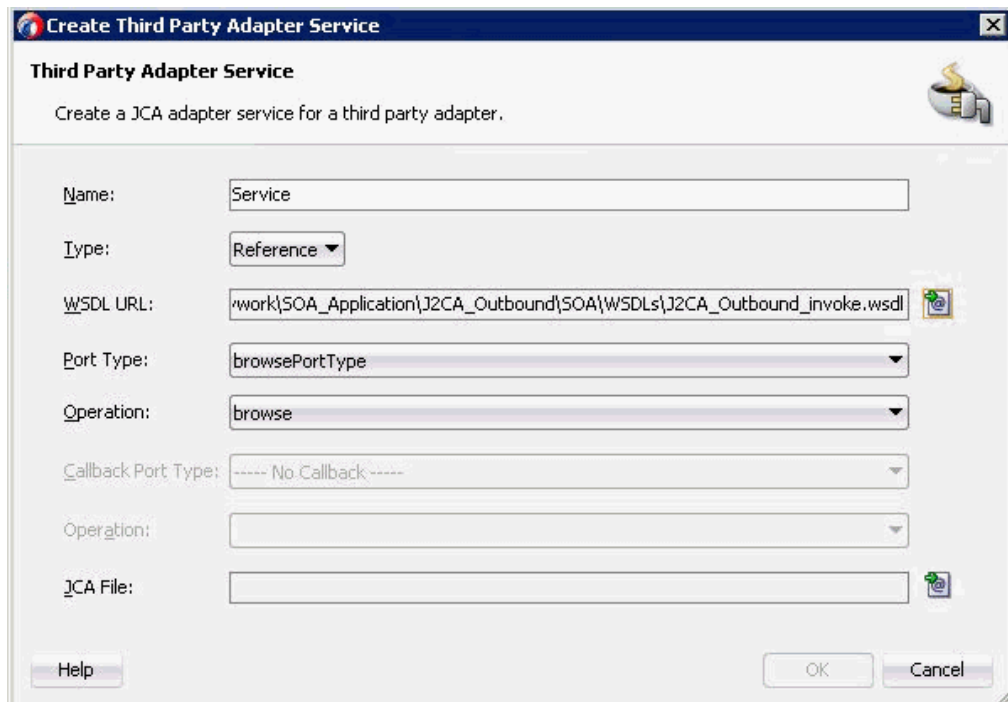


6. Click **OK**.

The outbound WSDL file and associated request and response XML schema files (.xsd) are imported to the project folder that has been created.

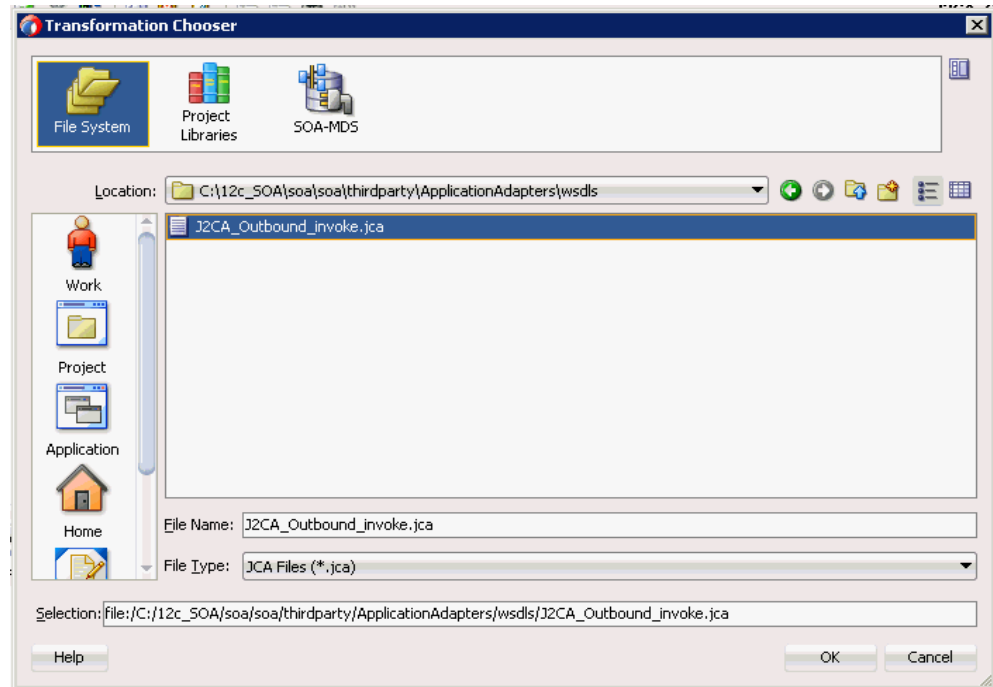
You are returned to the Create Third Party Adapter Service dialog, as shown in [Figure 4–18](#).

**Figure 4–18 Create Third Party Adapter Service Dialog**



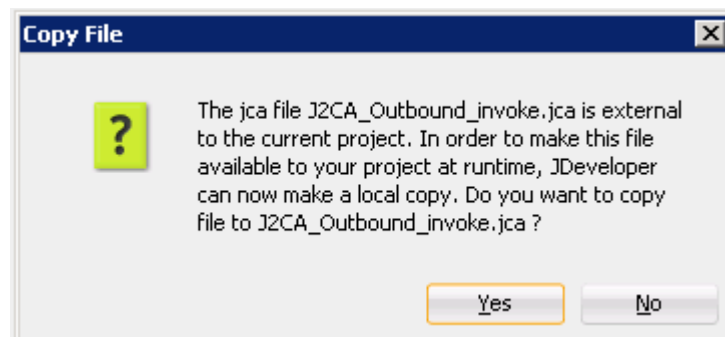
- Click the **Find JCA file** icon, which is located to the right of the JCA File field.  
The Transformation Chooser dialog is displayed, as shown in [Figure 4-19](#).

**Figure 4-19 Transformation Chooser Dialog**



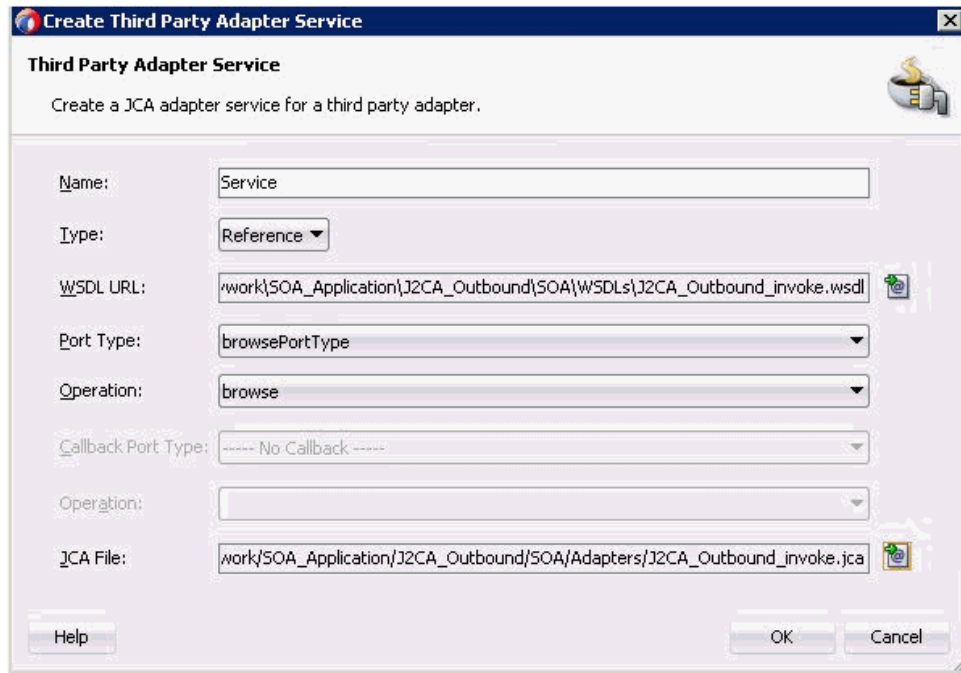
- Browse and select the JCA properties file from the following directory:  
`<ADAPTER_HOME>\wsdl\`
- Click **OK**.  
The Copy File message is displayed, as shown in [Figure 4-20](#).

**Figure 4-20 Copy File Message**



- Click **Yes**.  
A copy of the JCA properties file is made in the project folder.  
You are returned to the Create Third Party Adapter Service dialog, as shown in [Figure 4-21](#).

**Figure 4–21 Create Third Party Adapter Service Dialog**



**11. Click OK.**

The third party adapter service component is created and displayed in the External References pane.

You are now ready to configure an outbound BPEL process component.

**4.4.3.2 Configuring an Outbound BPEL Process Component**

Perform the following steps to configure an outbound BPEL process component:

1. Drag and drop the **BPEL Process** component from the Components pane to the Components pane.

The Create BPEL Process dialog is displayed, as shown in [Figure 4–22](#).



**Figure 4–22 Create BPEL Process Dialog**

**Create BPEL Process**

A BPEL process is a service orchestration, based on the BPEL specification, used to describe/execute a business process (or large grained service), which is implemented as a stateful service.

BPEL 2.0 Specification  BPEL 1.1 Specification

Name: BPELProcess1

Namespace: http://xmlns.oracle.com/SOA\_Application/J2CA\_Outbound/BPELProcess1

Directory: C:\Jdeveloper\WORK\mywork\SOA\_Application\J2CA\_Outbound\SOA\BPEL

Template: Synchronous BPEL Process

Service Name: bpelprocess1\_client

Expose as a SOAP service

Transaction: required

Input: {http://xmlns.oracle.com/SOA\_Application/J2CA\_Outbound/BPELProcess1}process

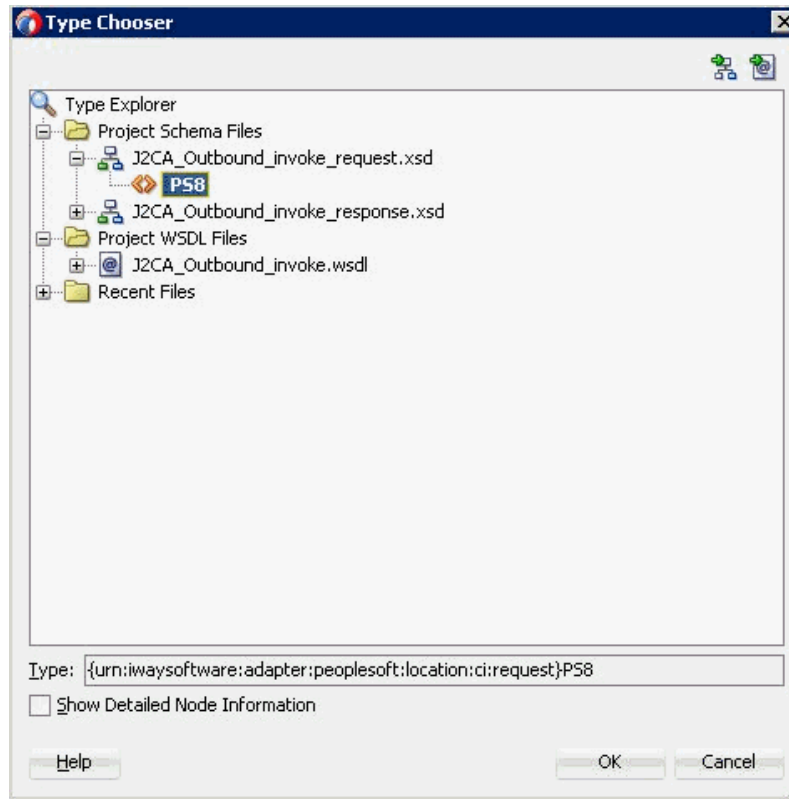
Output: :/xmlns.oracle.com/SOA\_Application/J2CA\_Outbound/BPELProcess1}processResponse

Help OK Cancel

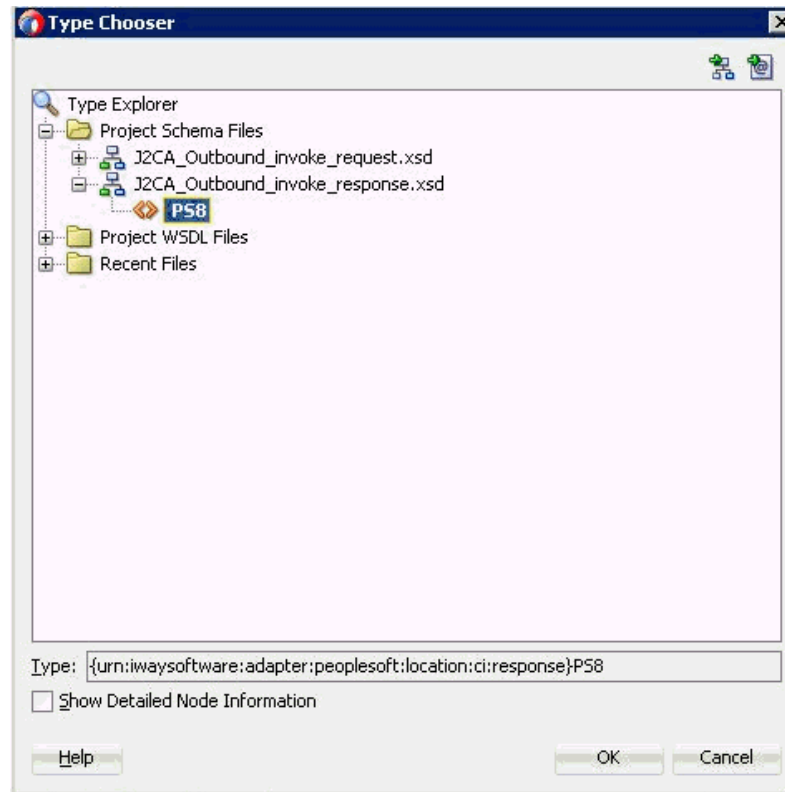
2. In the Name field, enter a name to identify the new outbound BPEL process component or leave it to the default value.  
By default, the BPEL 2.0 Specification option is selected.
3. From the Template list, select **Synchronous BPEL Process**.
4. Click the **Browse** icon, which is located to the right of the Input field to select the associated XML request schema file.

The Type Chooser dialog is displayed, as shown in [Figure 4–23](#).

**Figure 4–23 Type Chooser Dialog**

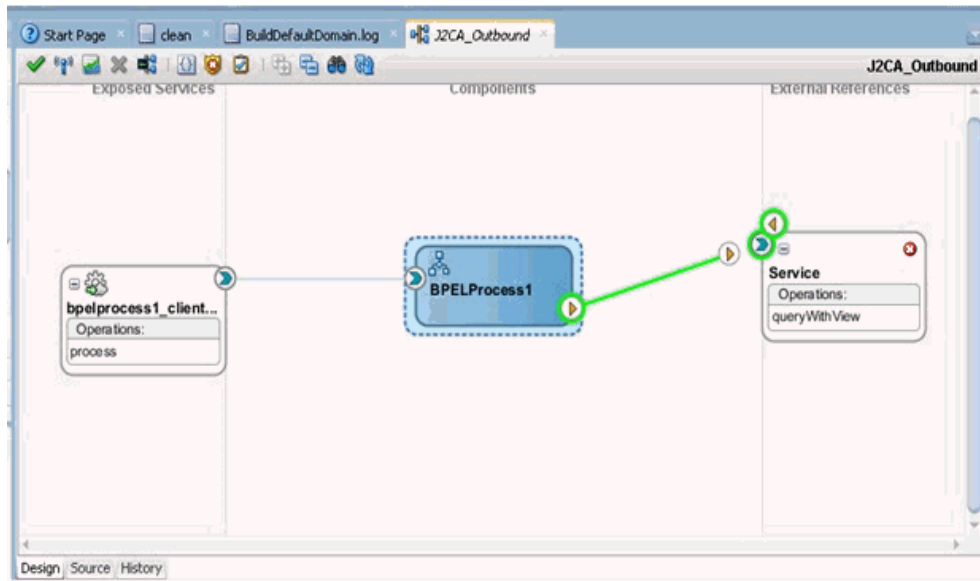


5. Expand **Project Schema Files**, **J2CA\_Outbound\_invoke\_request.xsd**, and select **PS8**.
6. Click **OK**.  
You are returned to the Create BPEL Process dialog.
7. Click the **Browse** icon, which is located to the right of the Output field to select the associated XML response schema file.  
The Type Chooser dialog is displayed, as shown in [Figure 4–24](#).

**Figure 4–24 Type Chooser Dialog**

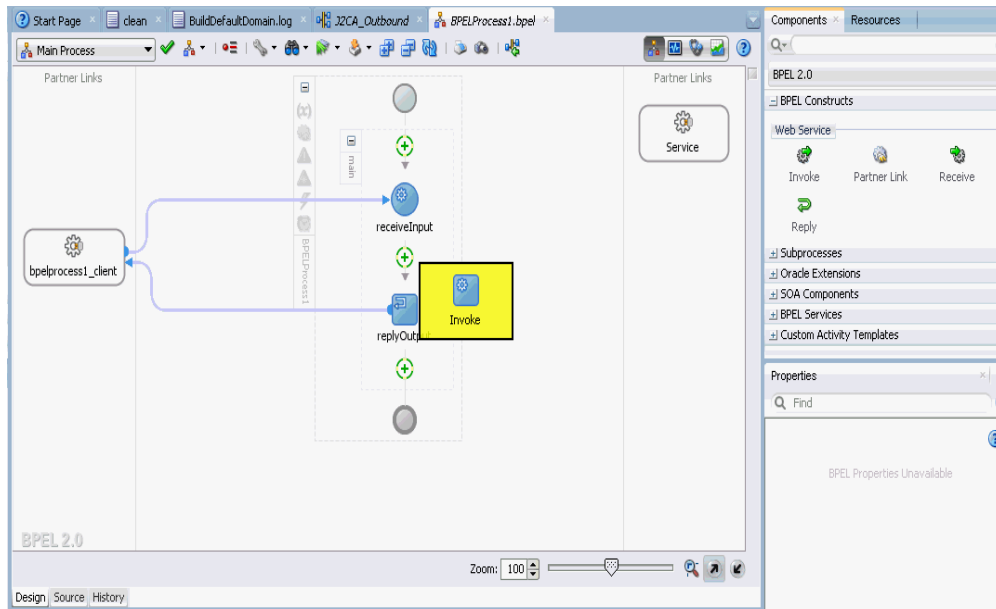
8. Expand **Project Schema Files**, **J2CA\_Outbound\_invoke\_response.xsd**, and select **PS8**.
9. Click **OK**.  
You are returned to the Create BPEL Process dialog.
10. Click **OK**.
11. Create a connection between the outbound BPEL process component and the third party adapter service component, as shown in [Figure 4–25](#).

**Figure 4–25 Created Connection**



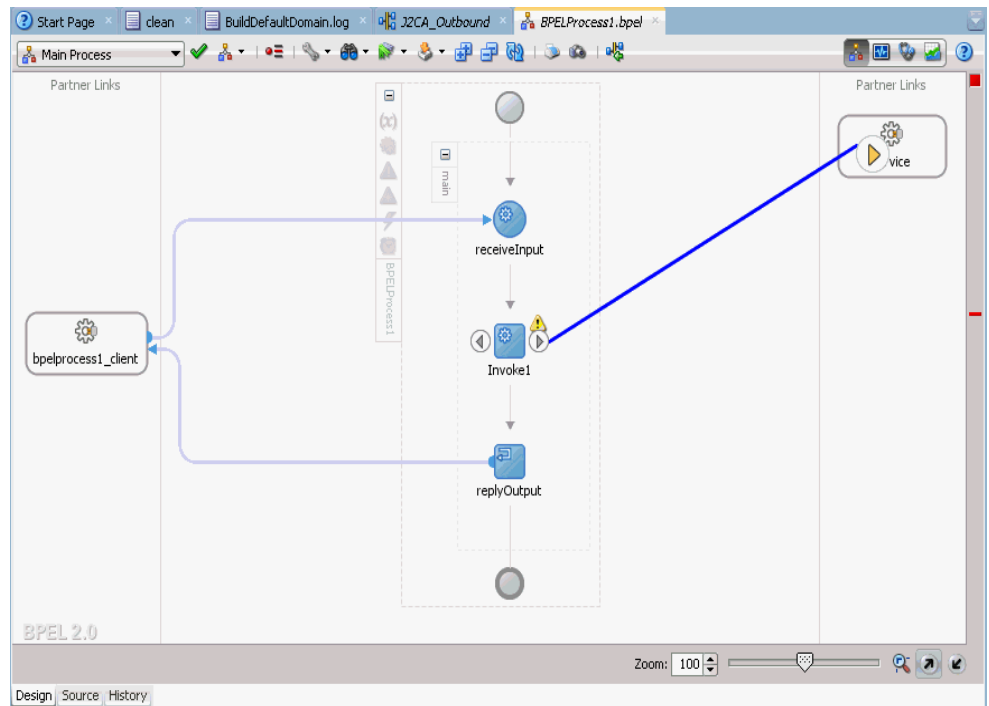
12. Double-click the outbound BPEL process component in the Components pane.
13. Drag and drop the **Invoke** activity component under BPEL Constructs - Web Service, to the Components pane and place it between the **receiveInput** activity component and the **replyOutput** activity component, as shown in [Figure 4–26](#).

**Figure 4–26 Invoke Activity Component**



14. Create a connection between the new Invoke activity component and the third party adapter service component (Service), as shown in [Figure 4–27](#).

Figure 4–27 Created Connection



The Edit Invoke dialog is displayed.

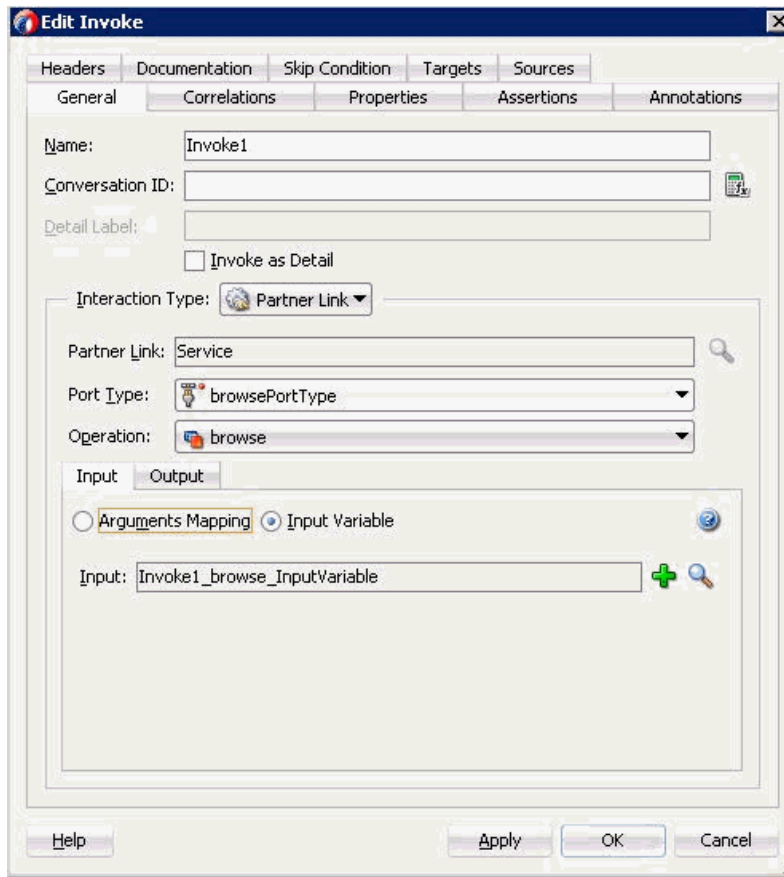
15. Click the **Plus sign** icon, which is located to the right of the Input field to configure a new input variable.

The Create Variable dialog is displayed.

16. Accept the default values that are provided for the new input variable and click **OK**.

You are returned to the Edit Invoke dialog, as shown in [Figure 4–28](#).

**Figure 4–28 Edit Invoke Dialog**



17. Select the **Output** tab and click the **Plus sign** icon, which is located to the right of the Output field to configure a new output variable.

The Create Variable dialog is displayed.

18. Accept the default values that are provided for the new output variable and click **OK**.

You are returned to the Edit Invoke dialog, as shown in [Figure 4–29](#).

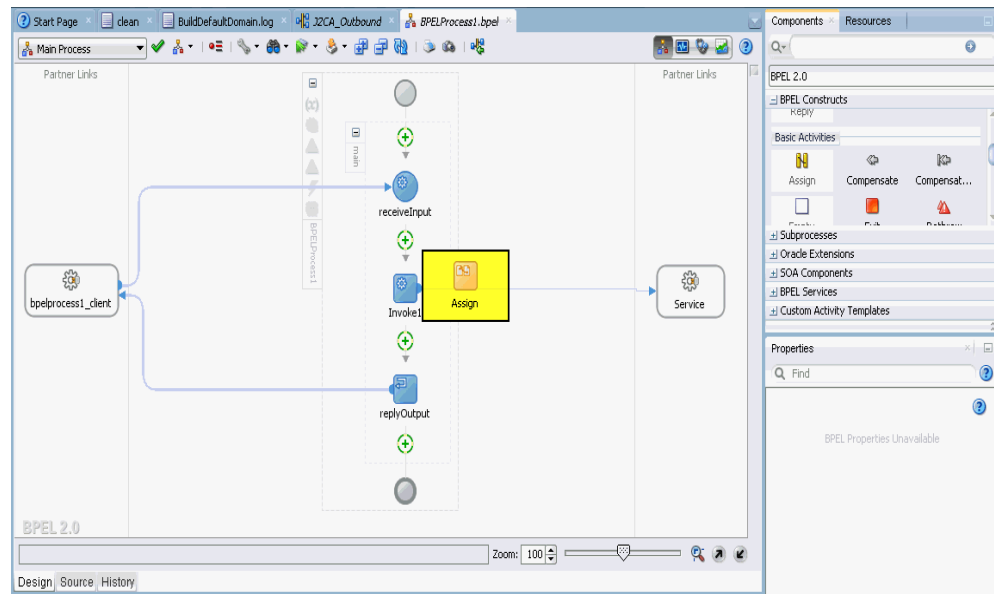
**Figure 4–29 Edit Invoke Dialog**

The screenshot shows the 'Edit Invoke' dialog box with the following configuration:

- Name:** Invoke1
- Conversation ID:** (empty)
- Detail Label:** (empty)
- Invoke as Detail
- Interaction Type:** Partner Link
- Partner Link:** Service
- Port Type:** browsePortType
- Operation:** browse
- Output:** Invoke1\_browse\_OutputVariable

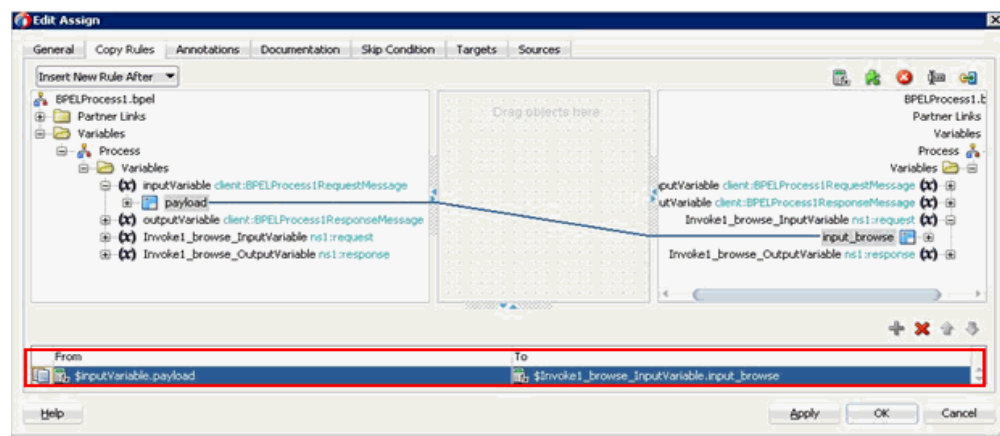
19. Click **Apply** and then **OK**.
20. Drag and drop the **Assign** activity under BPEL Constructs - Basic Activities component, to the Components pane and place it between the Receive activity component (receiveInput) and the Invoke activity component (Invoke1), as shown in [Figure 4–30](#).

**Figure 4–30 Assign Activity Component**



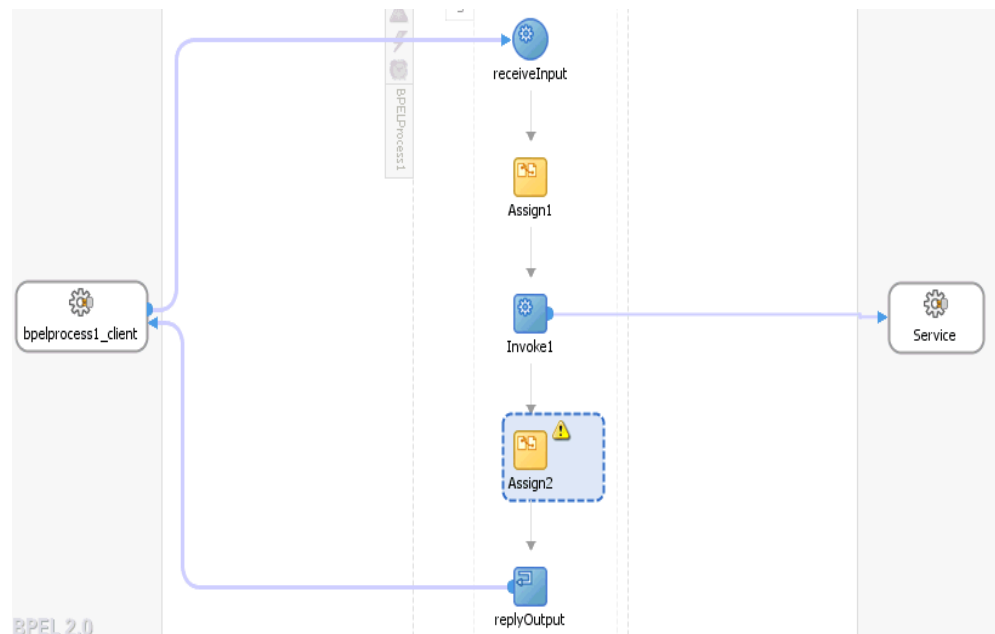
21. Double-click the new Assign activity component (**Assign1**).  
The Edit Assign dialog is displayed.
22. In the left pane, under Variables, expand **InputVariable**, and then select **payload**.
23. In the right pane, under Variables, expand **Invoke1\_browse\_InputVariable**, and then select **input\_browse**.
24. Drag and map the **payload** variable to the **input\_browse** variable.  
The mapped variables are populated in the highlighted area as shown in [Figure 4–31](#).

**Figure 4–31 Edit Assign Dialog**



25. Click **Apply** and then **OK**.
26. Drag and drop the **Assign** activity component to the Components pane and place it between the Invoke activity (Invoke1) and the Reply activity (replyOutput).
27. Double-click the new Assign activity component (**Assign2**), as shown in [Figure 4–32](#).

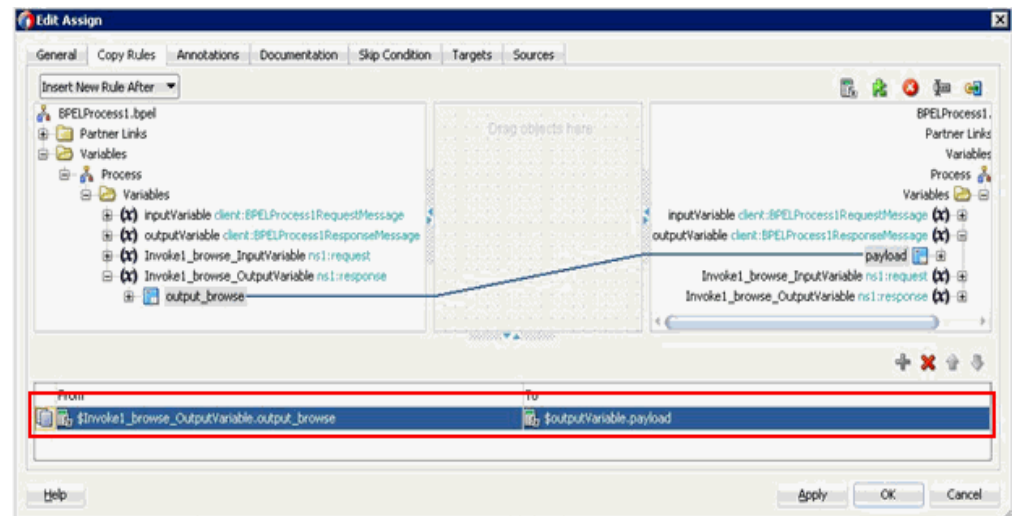


**Figure 4–32 New Assign Activity Component**


The Edit Assign dialog is displayed.

28. In the left pane, under Variables, expand **Invoke1\_browse\_OutputVariable**, and then select **output\_browse**.
29. In the right pane, under Variables, expand **outputVariable** and select **payload**.
30. Drag and map the **output\_browse** variable to the **payload** variable.

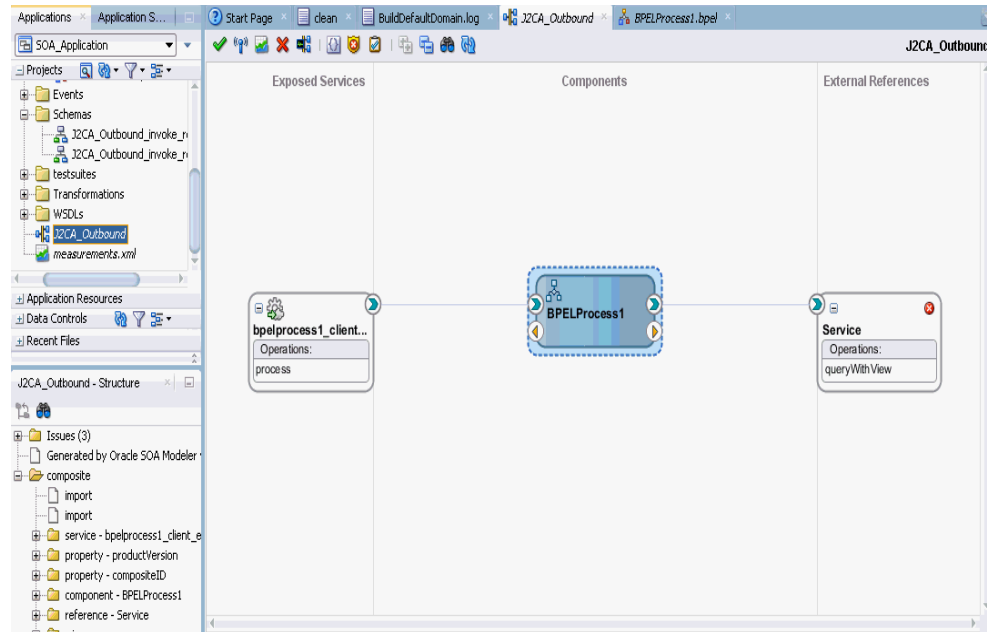
The mapped variables are populated in the highlighted area as shown in [Figure 4–33](#).

**Figure 4–33 Edit Assign Dialog**


31. Click **Apply** and then **OK**.

You are returned to the Activity component pane, as shown in [Figure 4–34](#).

**Figure 4–34 Activity Component Pane**



32. Click the **Save All** icon in the menu bar to save the new outbound BPEL process component that was configured.

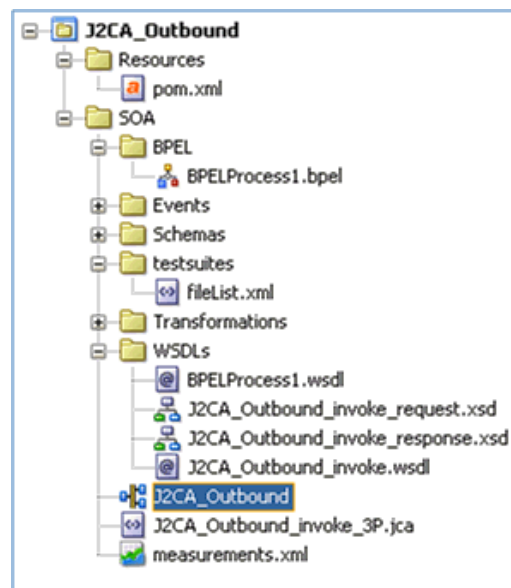
You are now ready to deploy the BPEL outbound process.

#### 4.4.3.3 Adjusting for Known Deployment Issues With 12c

Perform the following steps to adjust for known deployment issues with 12c.

1. Double-click **J2CA\_Outbound** (created BPEL process) of the created process, as shown in [Figure 4–35](#).

**Figure 4–35 J2CA\_Outbound Node**



2. Click the **Source** tab below the opened process, as shown in [Figure 4–36](#).

Figure 4–36 Source Tab

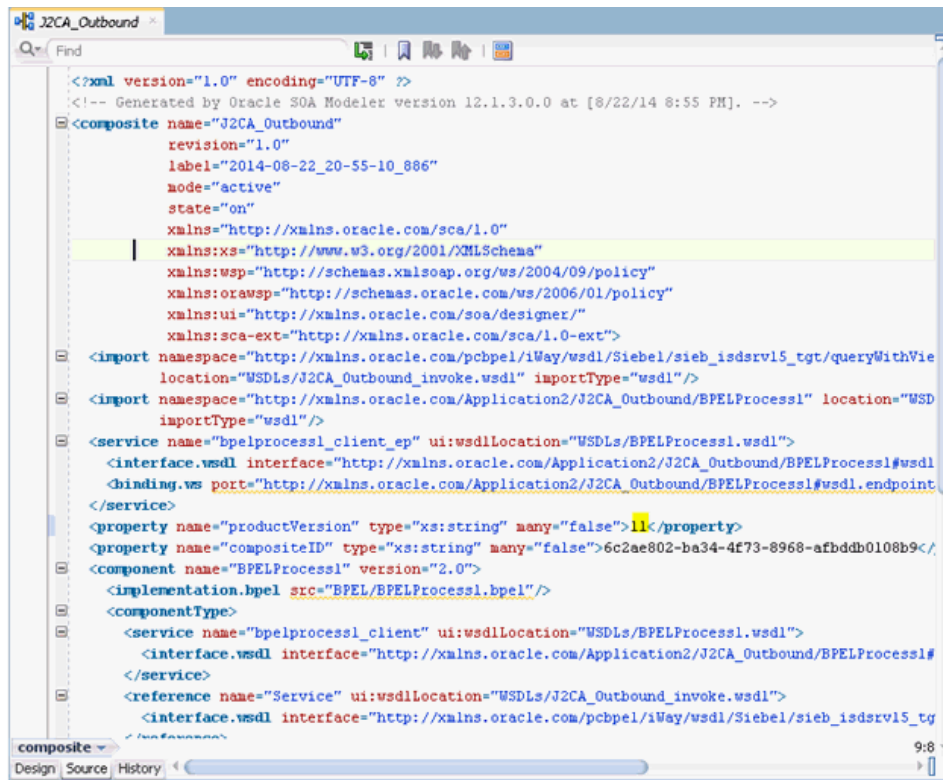
```

<?xml version="1.0" encoding="UTF-8" ?>
<!-- Generated by Oracle SOA Modeler version 12.1.3.0.0 at [8/22/14 8:55 PM]. -->
<composite name="J2CA_Outbound"
  revision="1.0"
  label="2014-08-22_20-55-10_886"
  mode="active"
  state="on"
  xmlns="http://xmlns.oracle.com/sca/1.0"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
  xmlns:orawsp="http://schemas.oracle.com/ws/2006/01/policy"
  xmlns:ui="http://xmlns.oracle.com/soa/designer/"
  xmlns:sca-ext="http://xmlns.oracle.com/sca/1.0-ext">
  <import namespace="http://xmlns.oracle.com/pcbpel/iWay/wsd1/Siebel/sieb_isdsrv15_tgt/queryWithVie
    location="WSDLs/J2CA_Outbound_invoke.wsd1" importType="wsdl"/>
  <import namespace="http://xmlns.oracle.com/Application2/J2CA_Outbound/BPELProcess1" location="WSD
    importType="wsdl"/>
  <service name="bpelprocess1_client_ep" ui:wsdLocation="WSDLs/BPELProcess1.wsd1">
    <interface.wsdl interface="http://xmlns.oracle.com/Application2/J2CA_Outbound/BPELProcess1#wsd1"
    <binding.ws port="http://xmlns.oracle.com/Application2/J2CA_Outbound/BPELProcess1#wsd1.endpoint
  </service>
  <property name="productVersion" type="xs:string" many="false">12.1.3.0.0</property>
  <property name="compositeID" type="xs:string" many="false">6c2ae802-ba34-4f73-8968-afbddb0108b9</
  <component name="BPELProcess1" version="2.0">
    <implementation.bpel src="BPEL/BPELProcess1.bpel"/>
    <componentType>
      <service name="bpelprocess1_client" ui:wsdLocation="WSDLs/BPELProcess1.wsd1">
        <interface.wsdl interface="http://xmlns.oracle.com/Application2/J2CA_Outbound/BPELProcess1#
      </service>
      <reference name="Service" ui:wsdLocation="WSDLs/J2CA_Outbound_invoke.wsd1">
        <interface.wsdl interface="http://xmlns.oracle.com/pcbpel/iWay/wsd1/Siebel/sieb_isdsrv15_tg
    </reference>
  </componentType>
</composite>

```

3. Change the productVersion property value from 12.1.3.0.0 to 11, as shown in Figure 4–37.

Figure 4-37 Property Value



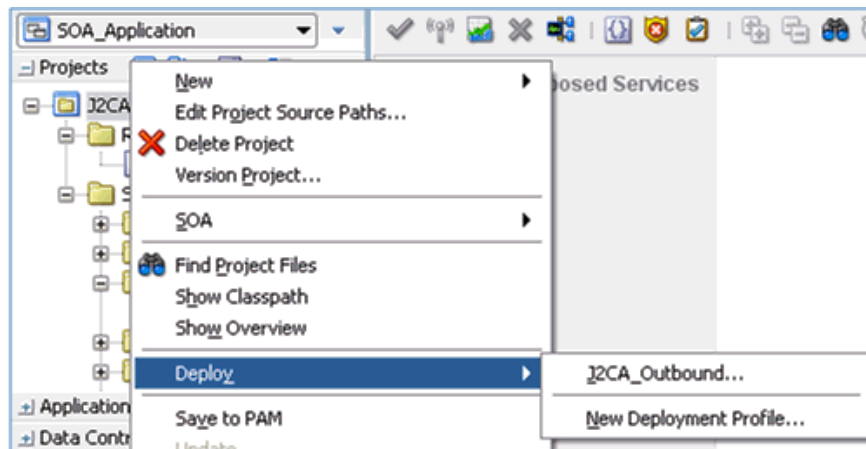
4. Save the changes and proceed to deploy the project.

#### 4.4.4 Deploying the BPEL Outbound Process

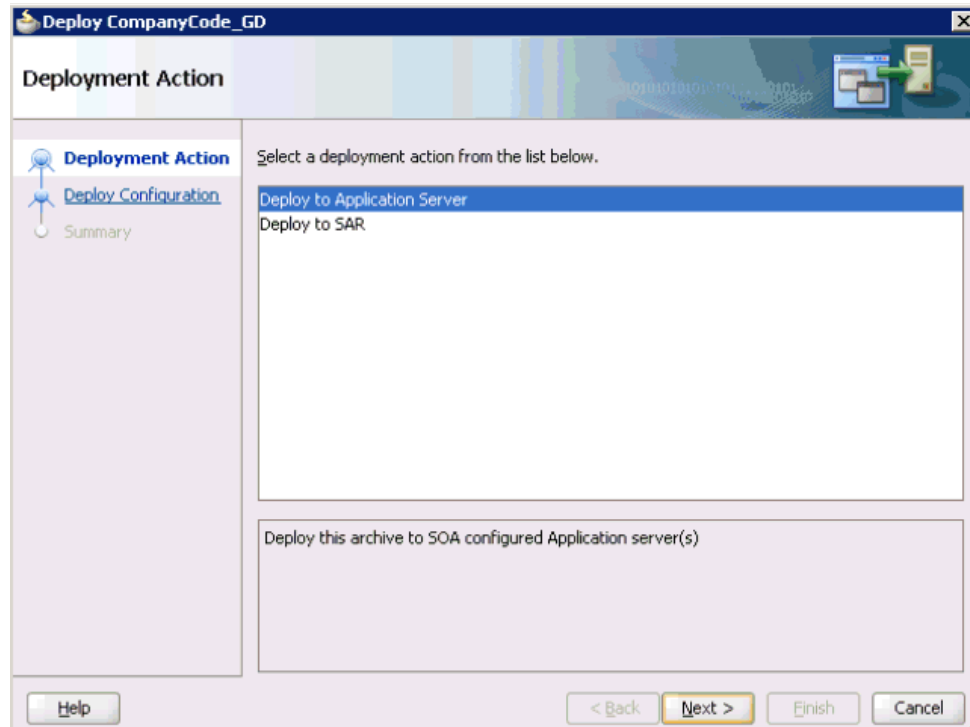
Perform the following steps to deploy the BPEL outbound process.

1. Right-click the project name in the left pane, select **Deploy**, and then click **J2CA\_Outbound**, as shown in Figure 4-38.

Figure 4-38 J2CA\_Outbound Option



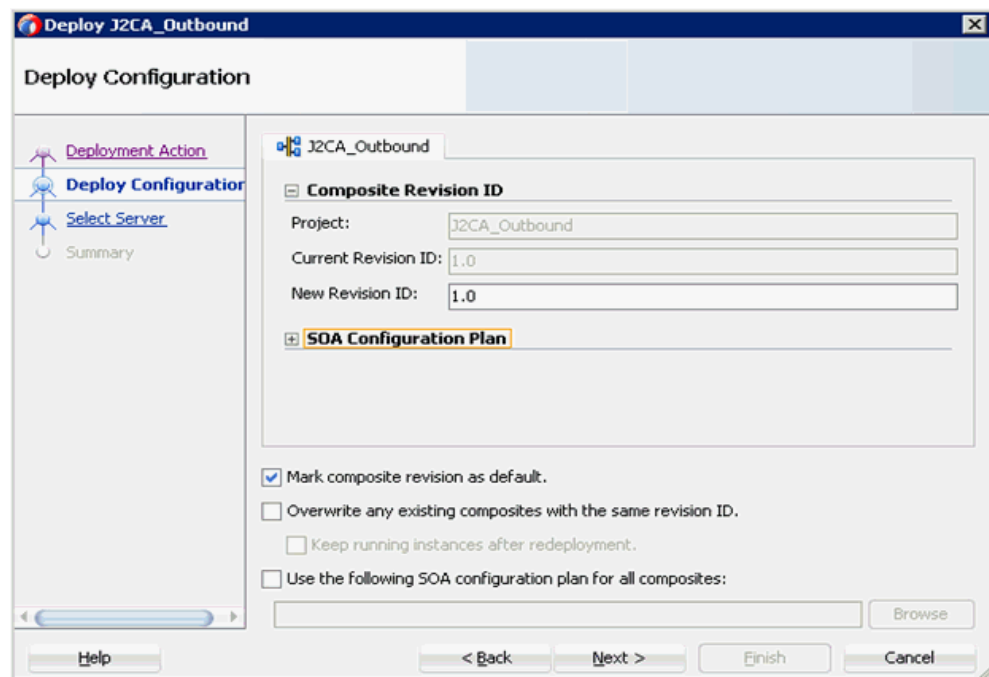
The Deployment Action page is displayed, as shown in Figure 4-39.

**Figure 4–39 Deployment Action Page**

2. Ensure that **Deploy to Application Server** is selected.

3. Click **Next**.

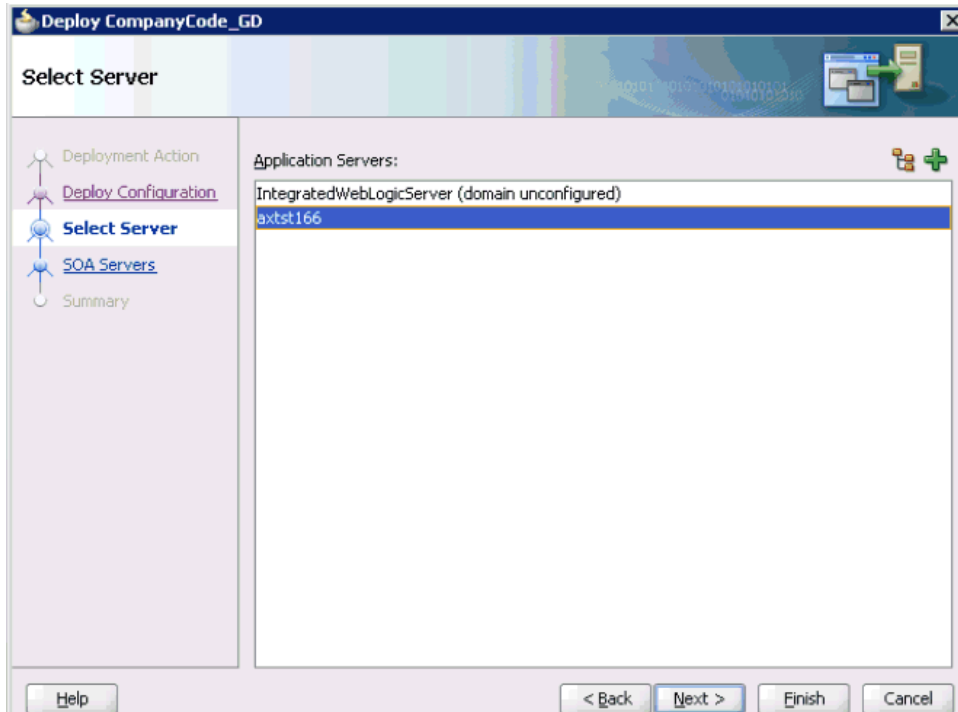
The **Deploy Configuration** page is displayed, as shown in [Figure 4–40](#).

**Figure 4–40 Deploy Configurations Page**

4. Leave the default values selected and click **Next**.

The Select Server page is displayed, as shown in [Figure 4-41](#).

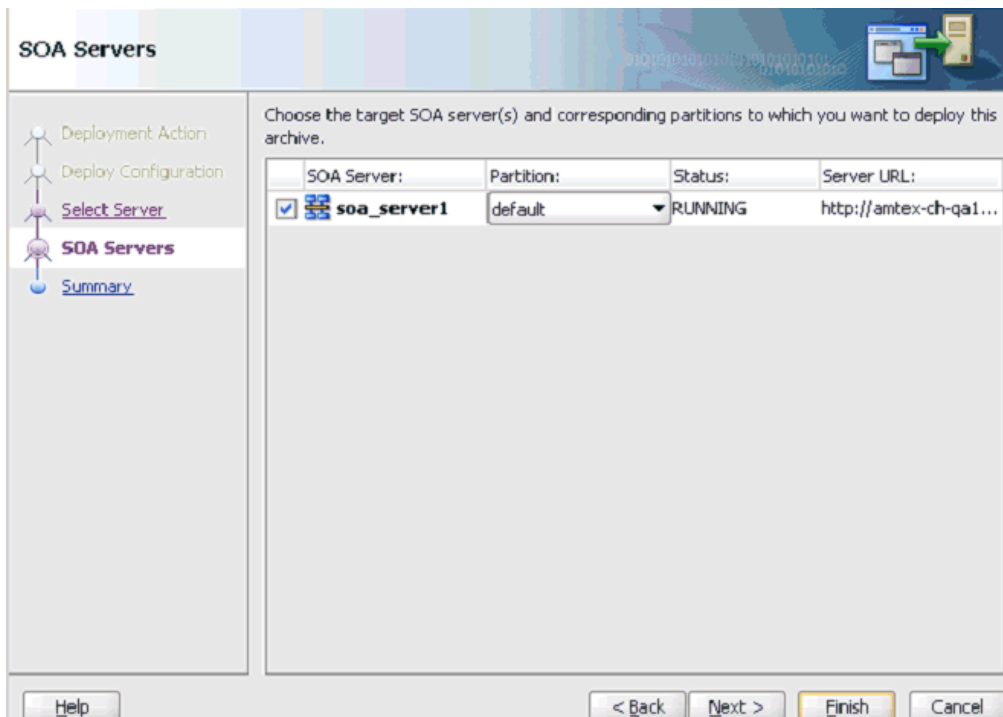
**Figure 4-41 Select Server Page**



5. Select an available application server that was configured and click **Next**.

The SOA Servers page is displayed, as shown in [Figure 4-42](#).

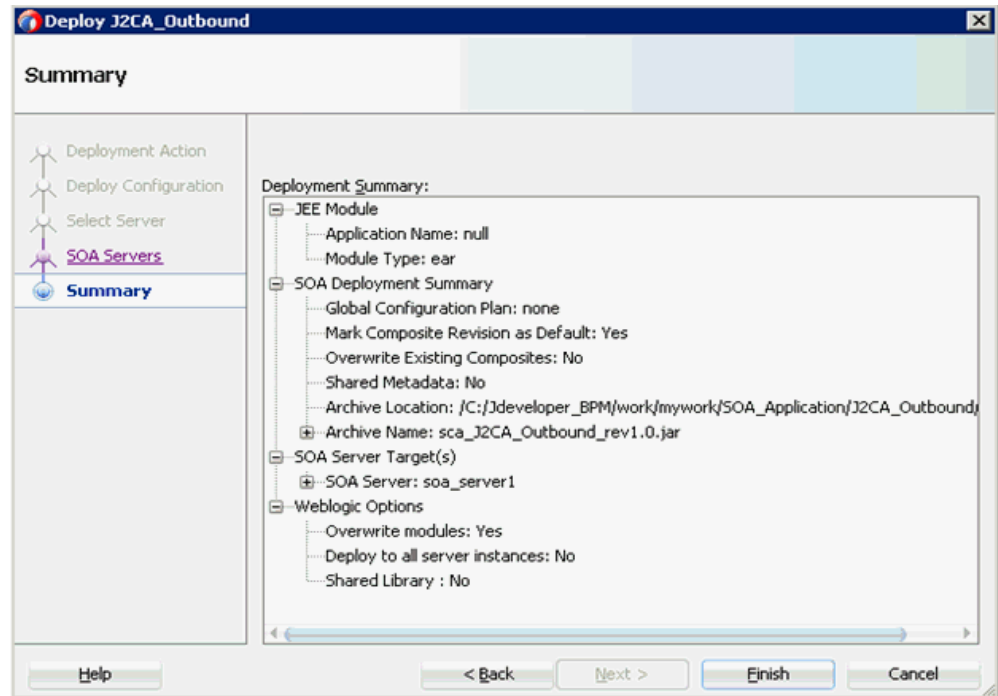
**Figure 4-42 SOA Servers Page**



6. Select a target SOA server and click **Next**.

The Summary page is displayed, as shown in [Figure 4–43](#).

**Figure 4–43 Summary Page**



7. Review and verify all the available deployment information for your project and click **Finish**.

The process is deployed successfully, as shown in [Figure 4–44](#).

**Figure 4–44 Successful Deployment Message**



#### 4.4.5 Invoking the Input XML Document in the Oracle Enterprise Manager Console

Perform the following steps to invoke the input XML document in the Oracle Enterprise Manager console.

1. Logon to the Oracle Enterprise Manager console.

**Note:** For customers using 12c (12.2.1.1.0) and 12c (12.2.1.2.0), perform the following steps:

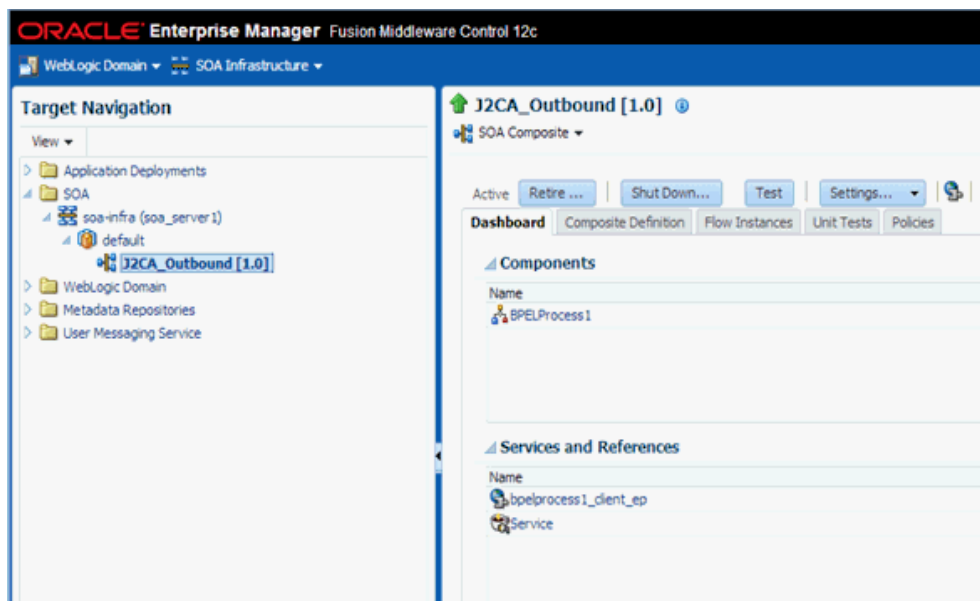
2. Click **Target Navigation** in the left pane, expand **SOA**, and then select **soa-infra (soa\_server1)**.

3. Click the **Deployed Composites** tab, which will list all of the deployed composites. Click on the available project (for example, J2CA\_Outbound).

Skip to **Step 4** in this procedure.

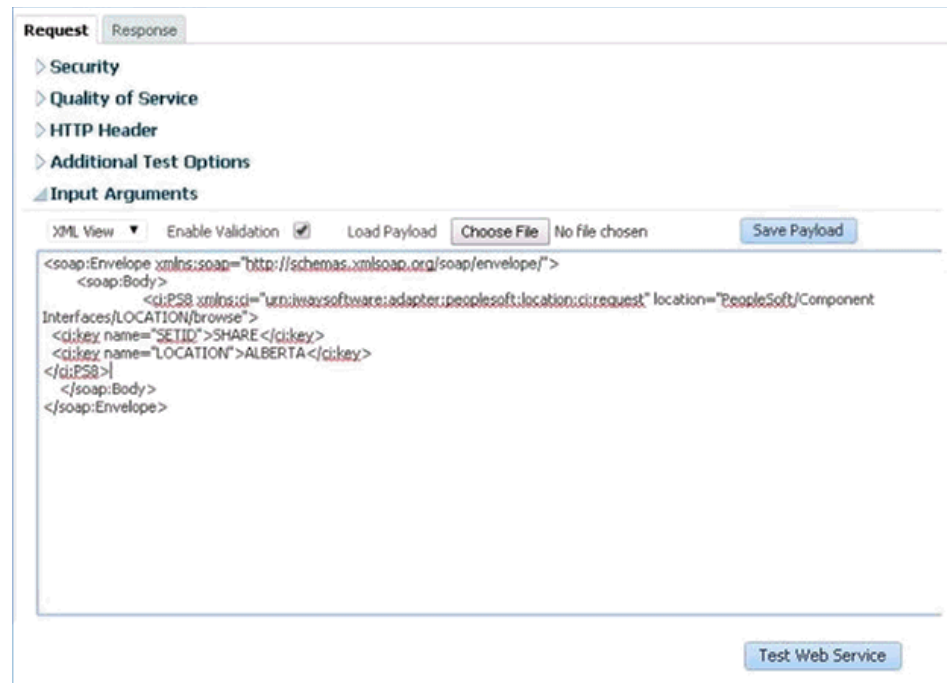
2. Expand **SOA**, select **soa-infra (soa\_server1)**, and then click **Default**.
3. Select an available project (for example, J2CA\_Outbound) and click **Test** as shown in [Figure 4-45](#).

**Figure 4-45 Test Button**



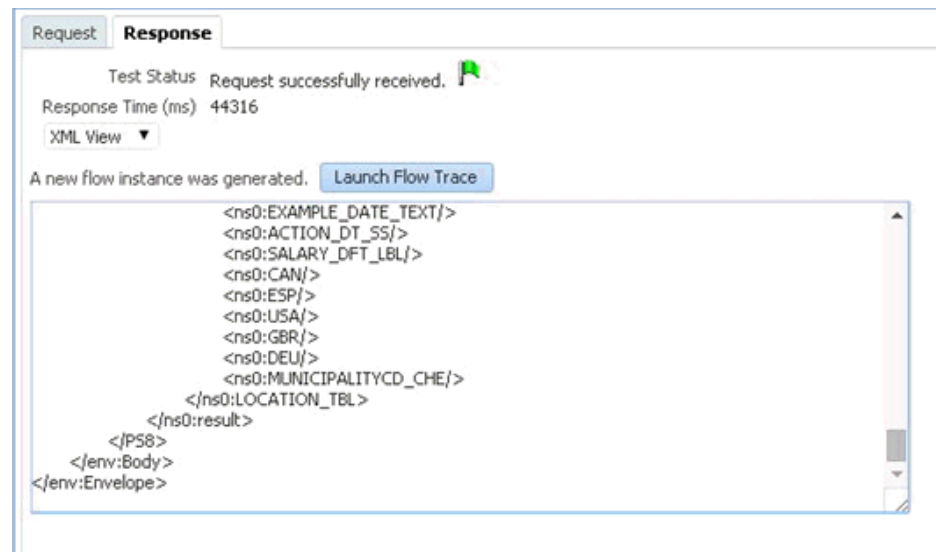
4. Click the **Request** tab.
5. Select **XML View** from the list, as shown in [Figure 4-46](#).



**Figure 4–46 Input Arguments List**

6. Provide an appropriate input XML document in the Input Arguments area and click **Test Web Service**.

The output response is received in the Oracle Enterprise Manager console, as shown in [Figure 4–47](#).

**Figure 4–47 Received Output Response**

#### 4.4.6 Testing Outbound BPEL and Mediator Processes

When testing an outbound BPEL process or an outbound Mediator process from the Oracle Enterprise Manager console, do not use the XML envelopes that are generated

by these consoles. Instead, remove them and use the XML payloads that are generated from the schemas, which conform to the WSDLs for namespace qualifications.

The Mediator data flows can be tested using the Enterprise Manager console. When creating a Mediator data flow and interactions, the Web services are created and registered with the Oracle Application Server. For more information on creating a Mediator outbound process, see [Chapter 5, "Integration With Mediator Service Components in the Oracle SOA Suite"](#).

## 4.5 Designing an Inbound BPEL Process for Event Integration (J2CA Configuration)

This section demonstrates how Oracle Application Adapter for PeopleSoft integrates with PeopleSoft to receive event data. In this example, a PeopleSoft event occurs when a customer record is added to a PeopleSoft system.

A sample project has been provided for this inbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\BPEL\J2CA\Inbound_Project
```

The following tools are required to complete your adapter design-time configuration:

- Oracle Adapter Application Explorer (Application Explorer)
- Oracle JDeveloper BPEL Designer (JDeveloper)

---

---

**Note:** The examples in this chapter demonstrate the use of JDeveloper.

---

---

Before you design a BPEL process, you must generate the respective WSDL file using Application Explorer. For more information, see ["Generating WSDL for Event Integration"](#) on page 4-34.

This section contains the following topics:

- [Section 4.5.1, "Generating WSDL for Event Integration"](#)
- [Section 4.5.2, "Creating an Empty Composite for SOA"](#)
- [Section 4.5.3, "Defining a BPEL Inbound Process"](#)
- [Section 4.5.4, "Deploying the BPEL Inbound Process"](#)
- [Section 4.5.5, "Triggering an Event in PeopleSoft"](#)

### 4.5.1 Generating WSDL for Event Integration

You must create a separate channel for every event and select that channel when you generate WSDL for inbound interaction using Application Explorer.

---

---

**Note:** If two or more events share the same channel, event messages may not be delivered to the right BPEL process.

---

---

This section contains the following topics:

- [Section 4.5.1.1, "Creating a Channel"](#)

- [Section 4.5.1.2, "Generating WSDL for Event Notification"](#)

### 4.5.1.1 Creating a Channel

To create a channel:

1. Click the **Events** node.
2. Expand the **PeopleSoft** node.

The ports and channels nodes appear in the left pane, as shown in [Figure 4-48](#).

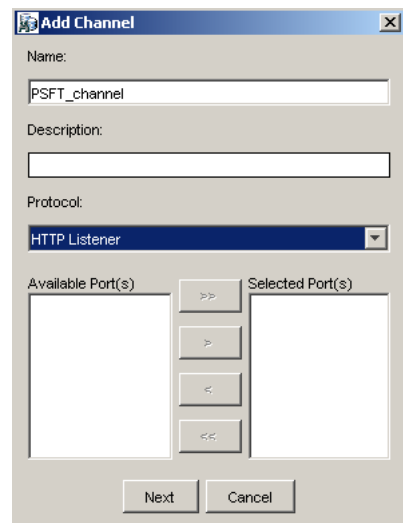
**Figure 4-48 Expanded PeopleSoft Events Node**



3. Right-click **Channels** and select **Add Channel**.

The Add Channel dialog is displayed, as shown in [Figure 4-49](#).

**Figure 4-49 Add Channel Dialog**



Provide the following information:

- a. Enter a name for the channel, for example, **PSFT\_Channel**.
  - b. Enter a brief description (optional).
  - c. From the Protocol list, select **HTTP Listener**.
4. Click **Next**.
  5. When the HTTP listener dialog is displayed, enter the system information as specified in the following table:

Parameter	Description
Listener port	Port on which to listen for PeopleSoft event data.
Https	For a secure HTTP connection, select the <b>Https</b> check box.
Synchronization Type	Choose from the following synchronization options: <ul style="list-style-type: none"> <li>■ REQUEST_RESPONSE</li> <li>■ REQUEST_ACK</li> </ul> <p><b>Important:</b> The PeopleSoft channel does not work if the synchronization type is set to REQUEST.</p>

6. Click **OK**.

The channel is displayed under the channels node in the left pane. An X over the icon indicates that the channel is currently disconnected.

---

**Note:** Do not start the channel, as it is managed by BPEL PM Server. If you start the channel for testing and debugging purposes, stop it before run-time.

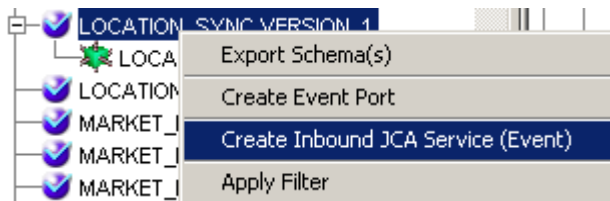
---

#### 4.5.1.2 Generating WSDL for Event Notification

After you create a channel and verify that it is not started, you must generate WSDL for the event using Application Explorer.

1. Start Application Explorer.
2. Expand the **Adapters** node.  
A list of all available adapters is displayed.
3. Expand **PeopleSoft**.
4. Click a target name under the **PeopleSoft** node, for example, **PSFTtarget**.  
The Logon pane on the right displays the saved parameters.
5. Verify your connection parameters.
6. Right-click the target name and select **Connect**.  
The x icon disappears, indicating that the target is connected.
7. Expand **Messages** and select **LOCATION\_SYNC.VERSION\_1**.
8. Right-click **LOCATION\_SYNC.VERSION\_1**.

**Figure 4–50 Create Inbound JCA Service (Event) Option**



9. Select **Create Inbound JCA Service (Event)**, as shown in [Figure 4–50](#).  
The Export WSDL dialog is displayed, as shown in [Figure 4–51](#).

Figure 4–51 Export WSDL Dialog

The screenshot shows the 'Export WSDL' dialog box with the following details:

- Name:** A text field containing the path `...adapters\tools\iwaet\bin\1.1.1\wsdl\J2CA_Inbound_receive.wsdl` and a 'Browse...' button.
- Channel:** A dropdown menu currently set to 'Psft\_Channel'.
- Validation:** Three unchecked checkboxes labeled 'Root', 'Namespace', and 'Schema'.
- Export to OSB:** An unchecked checkbox.
- Location, Host, Port, User, Password:** Five empty text input fields.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom center.
- Note:** A red text note at the bottom reads: '\* You must create a separate channel for each inbound service'.

Perform the following steps:

- a. In the **Name** field, specify the name of the WSDL file.

The **.wsdl** file extension is added automatically. By default, the names of WSDL files generated for request-response services end with **\_receive**.

- b. From the **Channel** list, select the channel you created for this inbound service.

**Important:** You must create a separate channel for every event. Verify that the channel is stopped before run-time.

- c. Three check boxes for Root, Namespace, and Schema validation are also available. Selection of multiple validation options is allowed.

- Root validation is used to validate the root element in the inbound XML document.

- Namespace validation is used to validate the namespace in the inbound XML document.

- Schema validation is used to validate the inbound XML document with the schema in the WSDL document.

During run time, validation is processed based on the validation options that are selected. If more than one validation option is selected, during run time if the first validation option fails, the remaining validation options are not processed. Root and namespace validations are considered modest levels of validation. Schema validation is a stricter validation level. It is recommended to use root and namespace validation options together, unless the root element

and namespace are different between the Messages in the PeopleSoft environment.

**10. Click OK.**

You can now create a new SOA application, which is the first step that is required to define a BPEL inbound process in JDeveloper.

## 4.5.2 Creating an Empty Composite for SOA

Perform the following steps to create an empty composite for SOA:

1. Create a new SOA application.
2. Enter a name for the new SOA Application and click **Next**.  
The Name your project page is displayed.
3. Enter a project name and click **Next**.  
The Configure SOA settings page is displayed.
4. From the Composite Template list, select **Empty Composite** and click **Finish**.  
For more information, see [Section 4.4.2, "Creating an Empty Composite for SOA"](#) on page 4-9.

## 4.5.3 Defining a BPEL Inbound Process

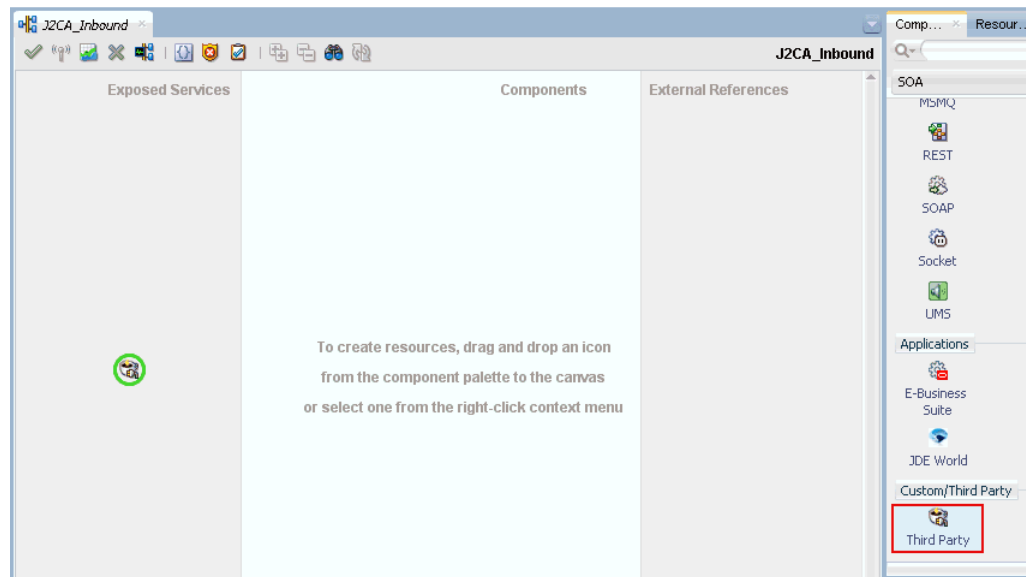
This section describes how to define a BPEL inbound process, which consists of the following topics:

- [Section 4.5.3.1, "Creating a Third Party Adapter Service Component"](#)
- [Section 4.5.3.2, "Creating an Inbound BPEL Process Component"](#)

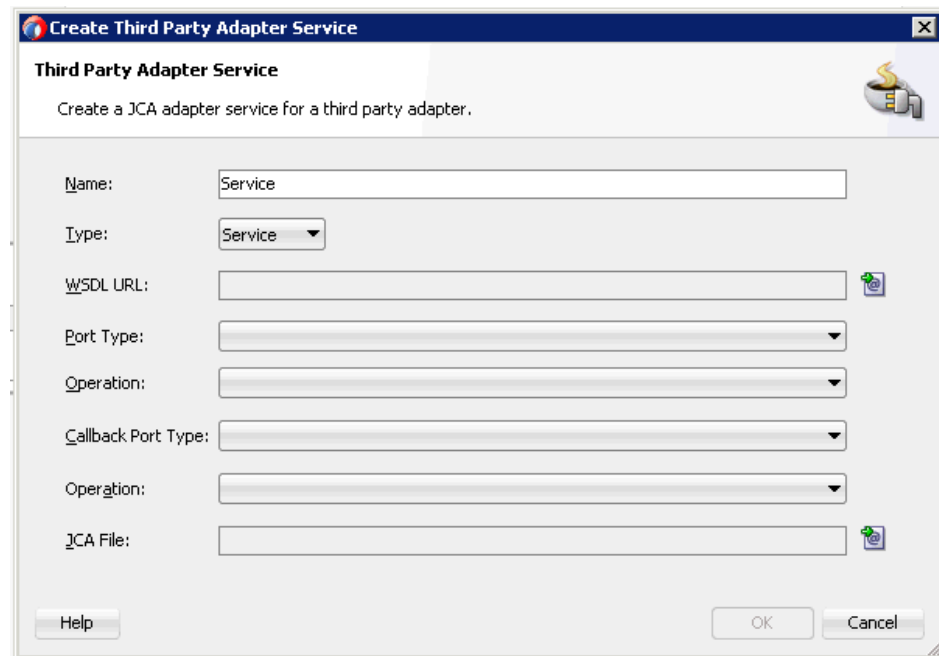
### 4.5.3.1 Creating a Third Party Adapter Service Component

Perform the following steps to create a third party adapter service component:

1. Drag and drop the **Third Party Adapter** component from the Service Adapters pane to the Exposed Services pane, as shown in [Figure 4-52](#).

**Figure 4–52 Third Party Adapter Component**

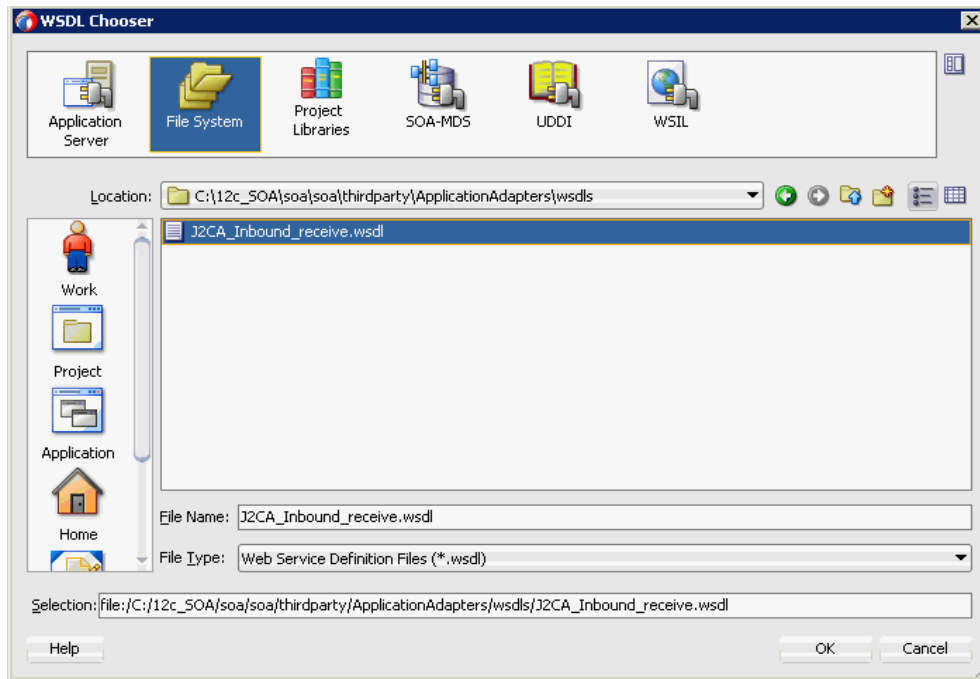
The Create Third Party Adapter Service dialog is displayed, as shown in [Figure 4–53](#).

**Figure 4–53 Create Third Party Adapter Service Dialog**

2. Ensure that **Service** is selected from the Type list (default).
3. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.

The WSDL Chooser dialog is displayed, as shown in [Figure 4–54](#).

**Figure 4–54 WSDL Chooser Dialog**



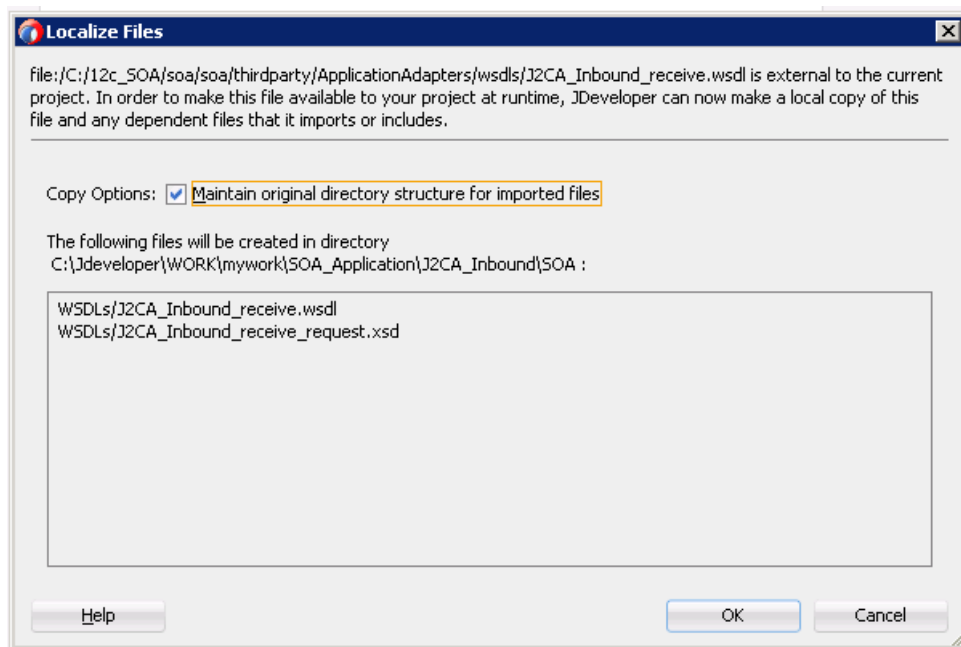
4. Browse and select an inbound WSDL file from the following directory:

<ADAPTER\_HOME>\wsdls

5. Click **OK**.

The Localize Files dialog is displayed, as shown in [Figure 4–55](#).

**Figure 4–55 Localize Files Dialog**



6. Click **OK**.



The inbound WSDL file and associated receive/request XML schema file (.xsd) are imported to the project folder that has been created.

You are returned to the Create Third Party Adapter Service dialog.

7. Click the **Find JCA file** icon, which is located to the right of the JCA File field. The Transformation Chooser dialog is displayed.

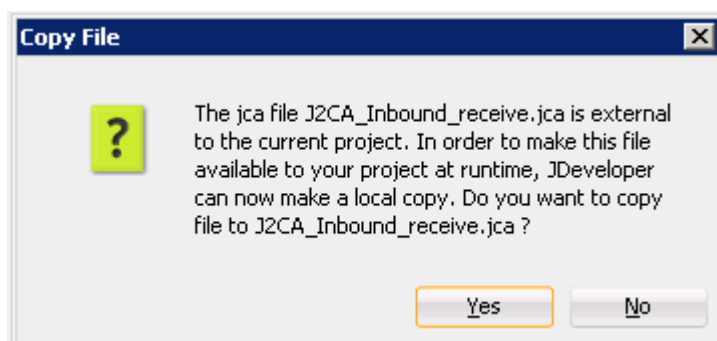
8. Browse and select the JCA properties file from the following directory:

```
<ADAPTER_HOME>\wsdl.s
```

9. Click **OK**.

A Copy File message is displayed, as shown in [Figure 4-56](#).

**Figure 4-56 Copy File Confirmation Message**

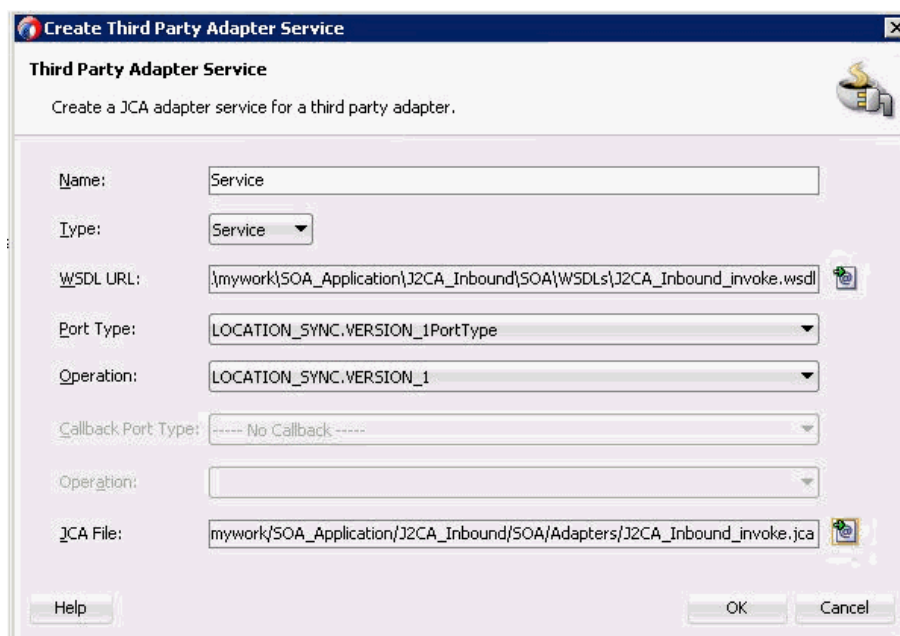


10. Click **Yes**.

A copy of the JCA properties file is made in the project folder.

You are returned to the Create Third Party Adapter Service dialog, as shown in [Figure 4-57](#).

**Figure 4-57 Create Third Party Adapter Service Dialog**



11. Click **OK**.

The third party adapter service component is created and displayed in the Exposed Services pane.

You are now ready to configure an inbound BPEL process component.

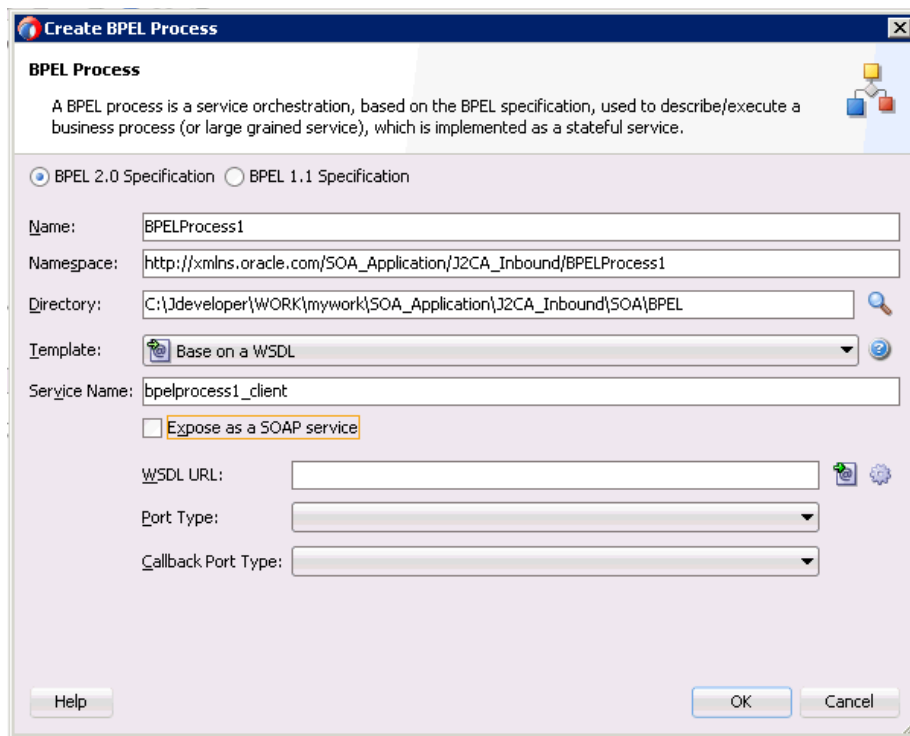
### 4.5.3.2 Creating an Inbound BPEL Process Component

Perform the following steps to create an inbound BPEL process component:

1. Drag and drop the **BPEL Process** component from the Service Components pane to the Components pane.

The Create BPEL Process dialog is displayed, as shown in [Figure 4-58](#).

**Figure 4-58 Create BPEL Process Dialog**



2. In the Name field, enter a name to identify the new inbound BPEL process component or leave to default.

By default, the BPEL 2.0 Specification option is selected.

3. From the Template list, select **Base on a WSDL**.
4. Uncheck the **Expose as SOAP service** check box.
5. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.

The WSDL Chooser dialog is displayed.

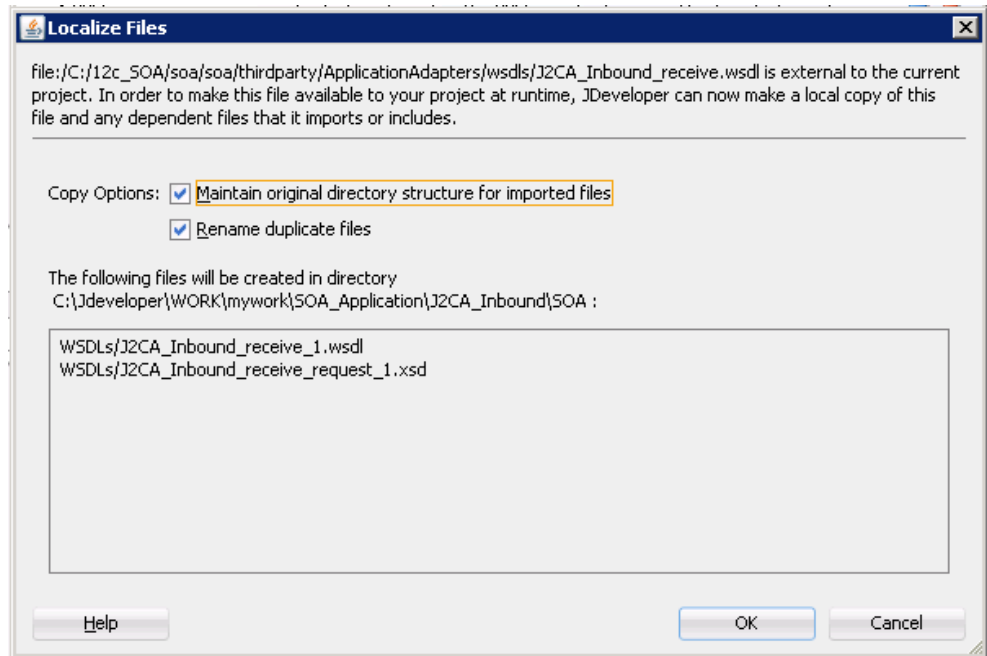
6. Select an inbound WSDL file from the following directory:

`<ADAPTER_HOME>\wsdl\`

7. Click **OK**.

The Localize Files dialog is displayed, as shown in [Figure 4-59](#).

**Figure 4-59 Localize Files Dialog**



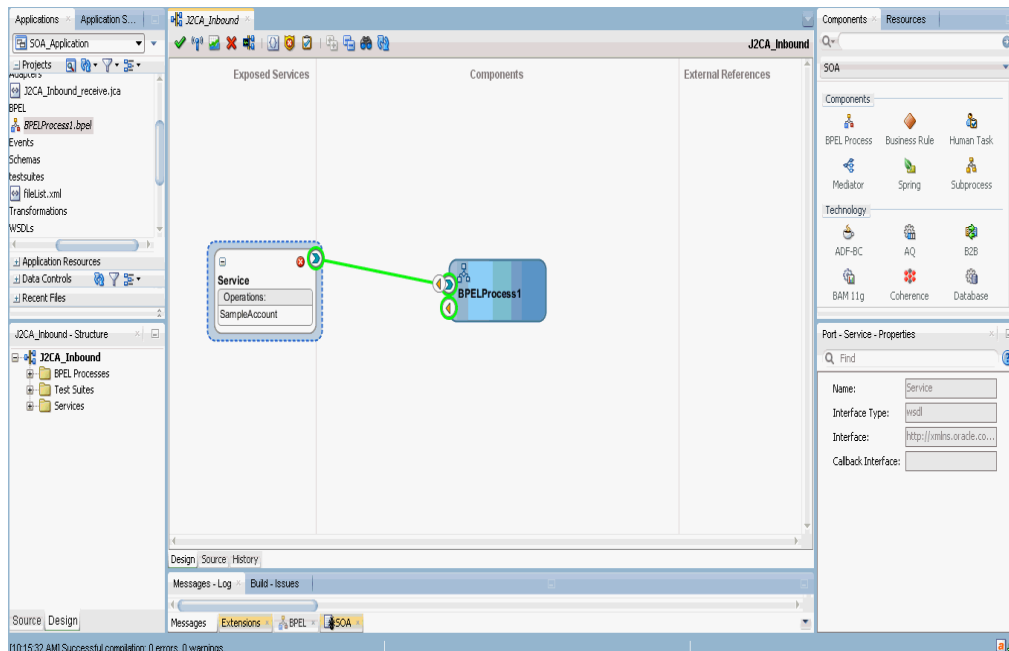
8. Uncheck the **Rename duplicate files** option.

9. Click **OK**.

You are returned to the Create BPEL Process dialog.

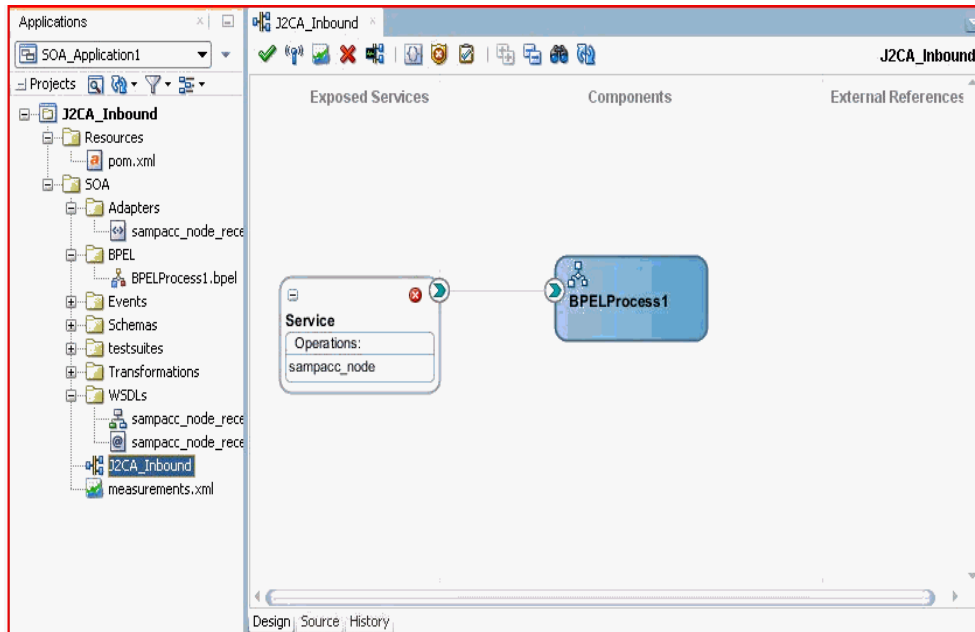
10. Click **OK**.

**Figure 4-60 Created Connection**



11. Create a connection between the third party adapter service component and the inbound BPEL process component, as shown in [Figure 4-60](#).
12. Double-click **J2CA\_Outbound** in the left pane.

**Figure 4-61 Save All Icon**



13. Click the **Save All** icon in the menu bar to save the new inbound BPEL process component that was configured, as shown in [Figure 4-61](#).

You are now ready to deploy the BPEL inbound process.

#### 4.5.3.3 Adjusting for Known Deployment Issues With 12c

For more information on how to adjust for known deployment issues with 12c, see [Section 4.4.3.3, "Adjusting for Known Deployment Issues With 12c"](#) on page 4-26.

### 4.5.4 Deploying the BPEL Inbound Process

Perform the following steps to deploy the BPEL inbound process.

1. Right-click the project name in the left pane, select **Deploy**, and click **J2CA\_Inbound**.  
The Deployment Action page is displayed.
2. Ensure that **Deploy to Application Server** is selected.
3. Click **Next**.  
The Deploy Configuration page is displayed.
4. Leave the default values selected and click **Next**.  
The Select Server page is displayed.
5. Select an available application server that was configured and click **Next**.  
The SOA Servers page is displayed.
6. Select a target SOA server and click **Next**.

The Summary page is displayed.

7. Review and verify all the available deployment information for your project and click **Finish**.

The process is deployed successfully.

For more information, see [Section 4.4.4, "Deploying the BPEL Outbound Process"](#) on page 4-28.

Once event messages are triggered through PeopleSoft, successful instances are received in the Oracle Enterprise Manager console, as shown in [Figure 4-62](#).

**Figure 4-62** Received Instances

The screenshot shows the Oracle Enterprise Manager console interface. On the left is the 'Target Navigation' tree with 'J2CA\_Inbound [1.0]' selected. The main area shows the 'Flow Instances' tab for the 'J2CA\_Inbound [1.0]' composite. A table displays the search results for instances created in the last 24 hours. All three instances are in a 'Completed' state.

Flow ID	Initiating Composite	Flow State	Created	Last Updated
80038	J2CA_Inbound [1.0]	Completed	Aug 25, 2014 10:08:24 AM	Aug 25, 2014 10:08:24 AM
80037	J2CA_Inbound [1.0]	Completed	Aug 25, 2014 10:08:22 AM	Aug 25, 2014 10:08:22 AM
80036	J2CA_Inbound [1.0]	Completed	Aug 25, 2014 10:08:20 AM	Aug 25, 2014 10:08:20 AM

## 4.5.5 Triggering an Event in PeopleSoft

Events are generated by activity in an application system. For example, PeopleSoft may generate an event as customer information is updated in the system. The following topics describe how to trigger an event in PeopleSoft and verify the event using Oracle Application Adapter for PeopleSoft.

This section consists of the following topic:

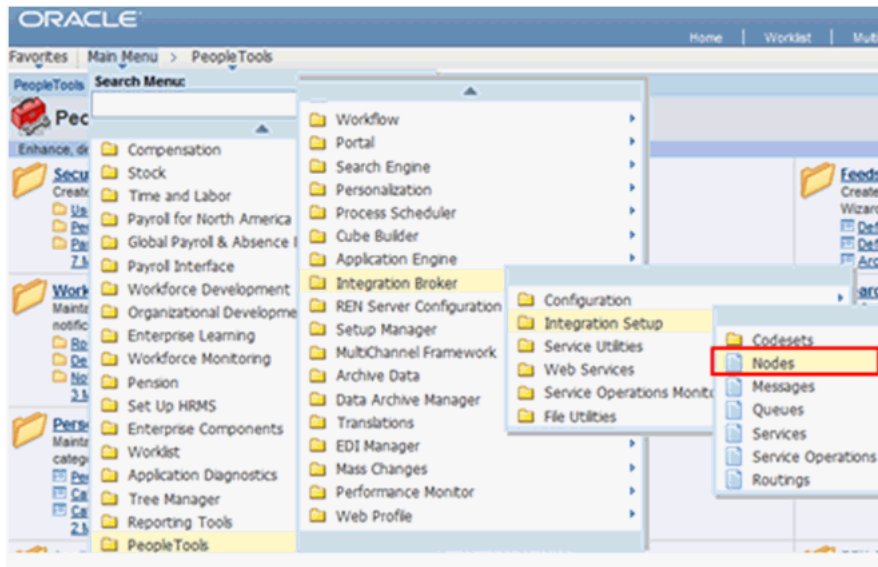
- [Section 4.5.5.1, "Triggering an Event in PeopleSoft"](#)
- [Section 4.5.5.2, "Verifying the Results"](#)

### 4.5.5.1 Triggering an Event in PeopleSoft

To trigger an event in PeopleSoft:

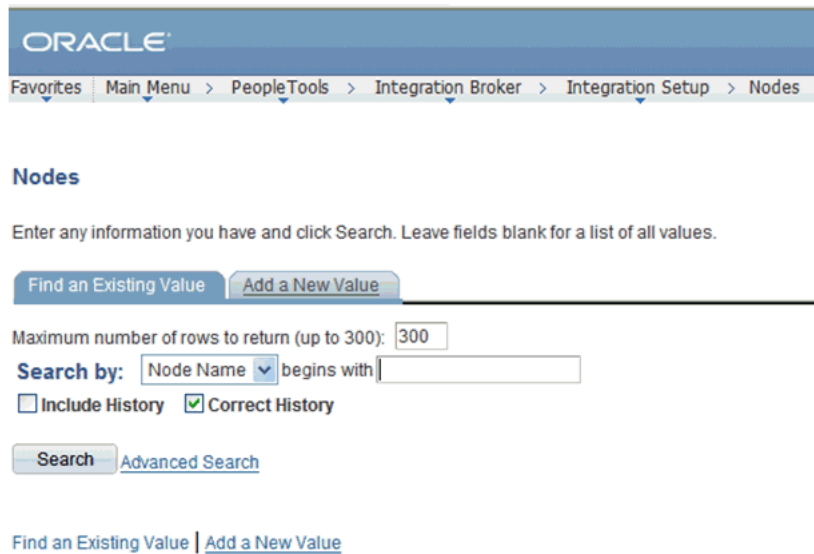
1. Log in to PeopleSoft, select **PeopleTools**, **Integration Broker**, **Integration Setup** and then **Nodes**, as shown in [Figure 4-63](#).

**Figure 4–63 PeopleTools Node Definitions**



2. Enter the node name you are using, for example, EXTERNAL, then click **Search**, as shown in Figure 4–64.

**Figure 4–64 External Node Name**



3. Click the **Connectors** tab, as shown in Figure 4–65.

Figure 4–65 Connectors Tab

ORACLE

Home | Worklist

Favorites | Main Menu > PeopleTools > Integration Broker > Integration Setup > Nodes

Node Definitions **Connectors** Portal WS Security Routings

Node Name: EXTERNAL Ping Node

Details

Gateway ID: LOCAL

Connector ID: HTTPTARGET

Properties

Property ID	Property Name	Required	Value
1 HEADER	sendUncompressed	<input checked="" type="checkbox"/>	Y
2 HTTPPROPERTY	Method	<input checked="" type="checkbox"/>	POST
3 PRIMARYURL	URL	<input checked="" type="checkbox"/>	http://172.19.20.242:1837

4. Enter **HTTPTARGET** in the **Connector ID** field; then enter the **listener URL** and its port in the **PRIMARYURL** field.
5. To save the configuration, click **Save**.

A successful dialog window is displayed.

To check the listener status, click on the **Ping Node** button and continue the triggering if a window is displayed, as shown in Figure 4–66.

Figure 4–66 Ping Node

Ping Node Results

Integration Gateway ID	Connector ID	Connector URL	Message Text
LOCAL	HTTPTARGET	http://172.19.20.242:1837	Success (117,73)

Return

6. Select **Integration Broker**, **Integration Setup**, and then **Services** from the main menu.
7. Enter **LOCATION\_SYNC** in the Services field, and then click **Search**, as shown in Figure 4–67.

Figure 4-67 Services

ORACLE

Favorites Main Menu > PeopleTools > Integration Broker > Integration Setup > Services

### Services

Enter any information you have and click Search. Leave fields blank for a list of all values.

Find an Existing Value Add a New Value

Maximum number of rows to return (up to 300): 300

Service: begins with LOCATION\_SYNC

Service Alias: begins with

Object owner identifier: =

Case Sensitive

Search Clear Basic Search Save Search Criteria

- Click LOCATION\_SYNC.VERSION\_1 in the Operation tab, as shown in Figure 4-68.

Figure 4-68 Operation Tab

### Services

Service: LOCATION\_SYNC

Description: Location Table - Sync Message

Comments: Incremental (Component Publish) Message.

Service Alias:

Object Owner ID: HR Core Objects

Namespace: http://xmlns.oracle.com/EnterpriseHCM/services

[Link Existing Operations](#) [View WSDL](#)

**Service Operations**

Service Operation:

Operation Type:

**Existing Operations** Customize | Find | View All | First 1 of 1 Last

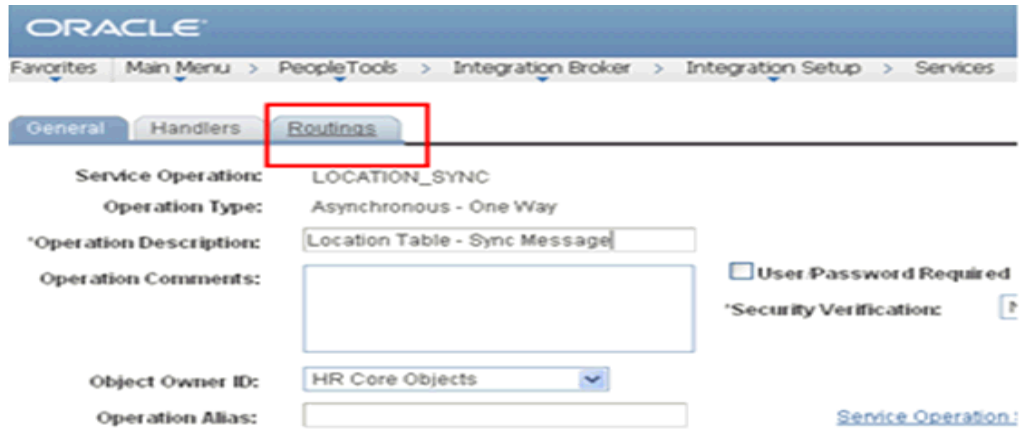
Operation Message Links

Operation	Default Version	Description	Active	Operation Type
LOCATION_SYNC.VERSION_1		Location Table - Sync Message	<input checked="" type="checkbox"/>	Asynch

- Click the **Routings** tab, as shown in Figure 4-69.

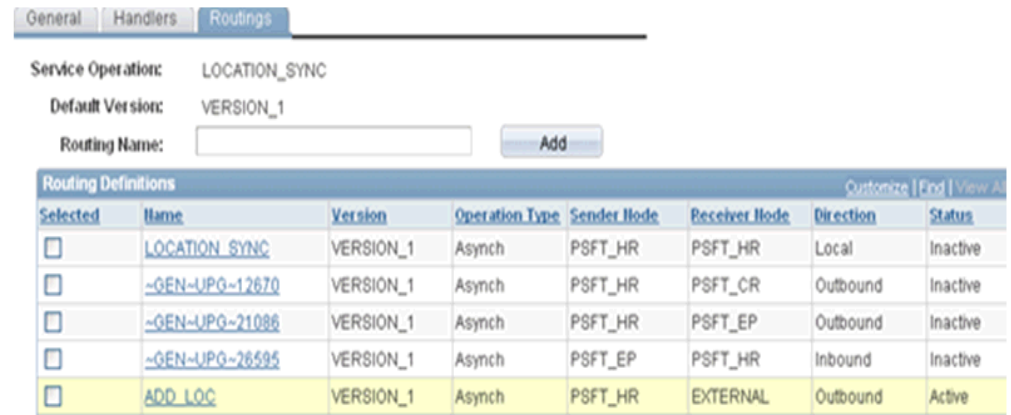


Figure 4–69 **Routings Tab**



10. Click ADD\_LOC in the Routing Definitions section, as shown in Figure 4–70.

Figure 4–70 **Routing Definitions Section**



11. Click the **Connector Properties** tab, as shown in Figure 4–71.

**Figure 4–71 Connector Properties Tab**

ORACLE

Favorites Main Menu > PeopleTools > Integration Broker > Integration Setup > Services

Routing Definitions Parameters **Connector Properties** Routing Properties

Routing Name: ADD\_LOC  Active

\*Service Operation: LOCATION\_SYNC  System Generated

Version: VERSION\_1

\*Description: ADD\_LOC [Graphical View](#)

Comments:

\*Sender Node: PSFT\_HR

\*Receiver Node: EXTERNAL

- Enter the URL and the port number of the HTTP listener in the Connector Properties section and click **Save**, as shown in [Figure 4–72](#).

**Figure 4–72 Connector Properties Section**

Routing Definitions Parameters **Connector Properties** Routing Properties

Routing Name: ADD\_LOC

Service Operation: LOCATION\_SYNC

Service Operation Version: VERSION\_1

Gateway ID: LOCAL

Connector ID: HTTPTARGET

Property ID	Property Name	Value
HEADER	sendUncompressed	Y
HTTPPROPERTY	Method	POST
PRIMARYURL	URL	http://172.19.20.242:1837 No Match Data

Save Return

- From the **Main Menu** in the left pane, select **Set Up HRMS, Foundation Tables, Organization**, and then **Location**, as shown in [Figure 4–73](#).

**Figure 4–73 PeopleSoft Integration Broker Connectors Tab**

The screenshot shows the Oracle PeopleSoft interface for the Location search. The breadcrumb trail is: Favorites > Main Menu > Set Up HRMS > Foundation Tables > Organization > Location. The page title is "Location". Below the title, there is a search instruction: "Enter any information you have and click Search. Leave fields blank for a list of all values." There are two buttons: "Find an Existing Value" and "Add a New Value". Below these buttons, there is a field for "Maximum number of rows to return (up to 300):" with the value "300". The search criteria are as follows:

- Set ID: = [dropdown] SHARE
- Location Code: begins with [dropdown] [text field]
- Description: begins with [dropdown] [text field]
- Salary Set ID: begins with [dropdown] [text field]
- Salary Administration Plan: begins with [dropdown] [text field]

- To find the location record you want to update, click **Search**; then make the changes and click **Save**, as shown in [Figure 4–74](#).

**Figure 4–74 Location Definition Pane**

The screenshot shows the Oracle PeopleSoft Location Definition Pane. The tabs are "Location Definition" and "Location Detail". The location is identified by SetID: SHARE and Location Code: ALBERTA. The "Location Definition" pane shows the following details:

- Eff Date: 05/04/2004 (calendar icon) Active (dropdown) Descr: Alberta - Canada (+/-)
- Country: CAN (dropdown) Canada
- Address 1: 600 Maple Street
- Address 2: Suite 9000
- Address 3: A11dress 3 Changed
- Address 4: [text field]
- City: Massapequa
- County: Kent
- Province: NF (dropdown) Newfoundland
- Building #: [text field]
- Floor #: [text field]
- Postal: A9A 9B9
- Jurisdiction: [text field]
- Sector: [text field]
- Prefix: [text field]
- Phone: [text field]
- Ext: [text field]
- Fax: [text field]
- In City Limit

At the bottom of the pane, there are several buttons: Save, Return to Search, Notify, Add, Update/Display, Include History, and Correct History. Below the buttons, there are links for "Location Definition" and "Location Detail".

#### 4.5.5.2 Verifying the Results

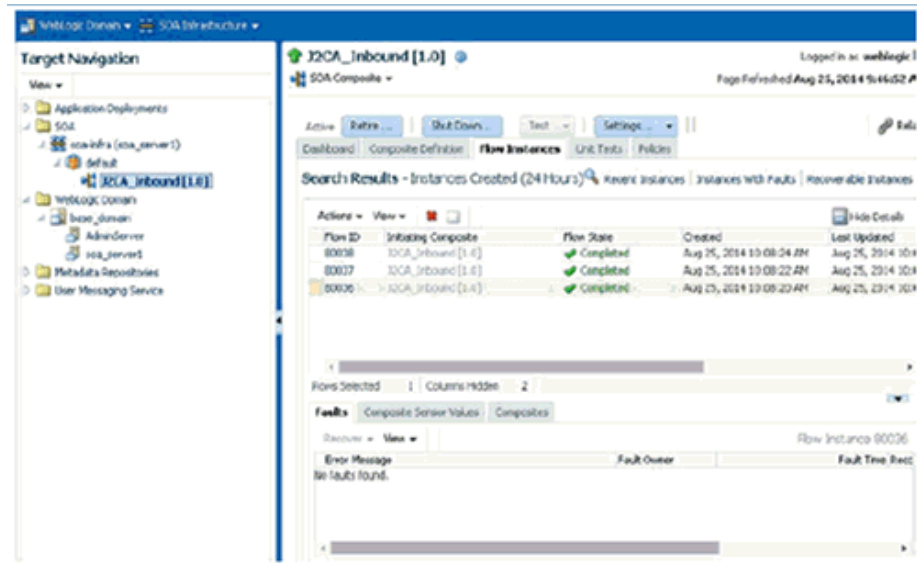
To verify your results:

- Log in to the Oracle Enterprise Manager console by using the following URL:

`http://localhost:7001/em`

2. Click **SOA**, select **soa-infra (soa\_server1)**, **default**, and then click **J2CA\_Inbound**.
3. Click **Flow Instances**.  
Instances will be received, as shown in [Figure 4–75](#).

**Figure 4–75 Flow Instances Tab**



## 4.6 Designing an Outbound BPEL Process for Service Integration (BSE Configuration)

This section describes how to design an outbound BPEL process for service integration.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\BPEL\BSE\Outbound_Project
```

The following tools are required to complete your outbound design-time configuration:

- Oracle Adapter Application Explorer (Application Explorer)
- Oracle JDeveloper BPEL Designer (JDeveloper)

This section includes the following topics:

- [Section 4.6.1, "Generating a WSDL File for Request and Response Services Using a Web Service"](#)
- [Section 4.6.2, "Creating an Empty Composite for SOA"](#)
- [Section 4.6.3, "Defining a BPEL Outbound Process"](#)

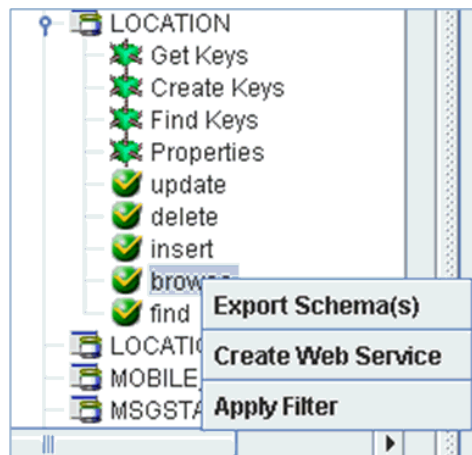
Before you design a BPEL process, you must generate the respective WSDL file using Application Explorer. For more information, see [Section 4.6.1, "Generating a WSDL File for Request and Response Services Using a Web Service"](#).

### 4.6.1 Generating a WSDL File for Request and Response Services Using a Web Service

To generate a WSDL file for request and response services using a Web service:

1. Start Application Explorer and connect to a defined PeopleSoft target (BSE configuration).  
For more information on defining a target and connecting to PeopleSoft, see [Section 2.4.1, "Defining a Target to PeopleSoft"](#).
2. Expand the PeopleSoft target to which you are connected.
3. Expand **Component Interfaces** and select **LOCATION**.
4. Navigate to and right-click an operation, and select **Create Web Service** from the menu, as shown in [Figure 4-76](#).

**Figure 4-76 Create Web Service Option**



The Create Web Service dialog is displayed.

5. Enter a service name, and click **Next**.
6. Click **OK** on the next dialog that is displayed.

Application Explorer switches the view to the Business Services node, and the new Web service is displayed in the left pane.

7. Right-click the new Web service and select **Save WSDL** from the menu.
8. Save the WSDL in the wsdl folder and click **Save**.

You can now create an empty composite for SOA, which is the first step that is required to define a BPEL outbound process in JDeveloper.

### 4.6.2 Creating an Empty Composite for SOA

To create an empty composite for SOA:

1. Create a new SOA application.
2. Enter a name for the SOA Application and click **Next**.

The Name your project page is displayed.

3. Enter a project name and click **Next**.

The Configure SOA settings page is displayed.

4. From the Composite Template list, select **Empty Composite** and click **Finish**.  
For more information, see [Section 4.4.2, "Creating an Empty Composite for SOA,"](#) on page 4-9.

### 4.6.3 Defining a BPEL Outbound Process

This section describes how to configure a BPEL outbound process component.

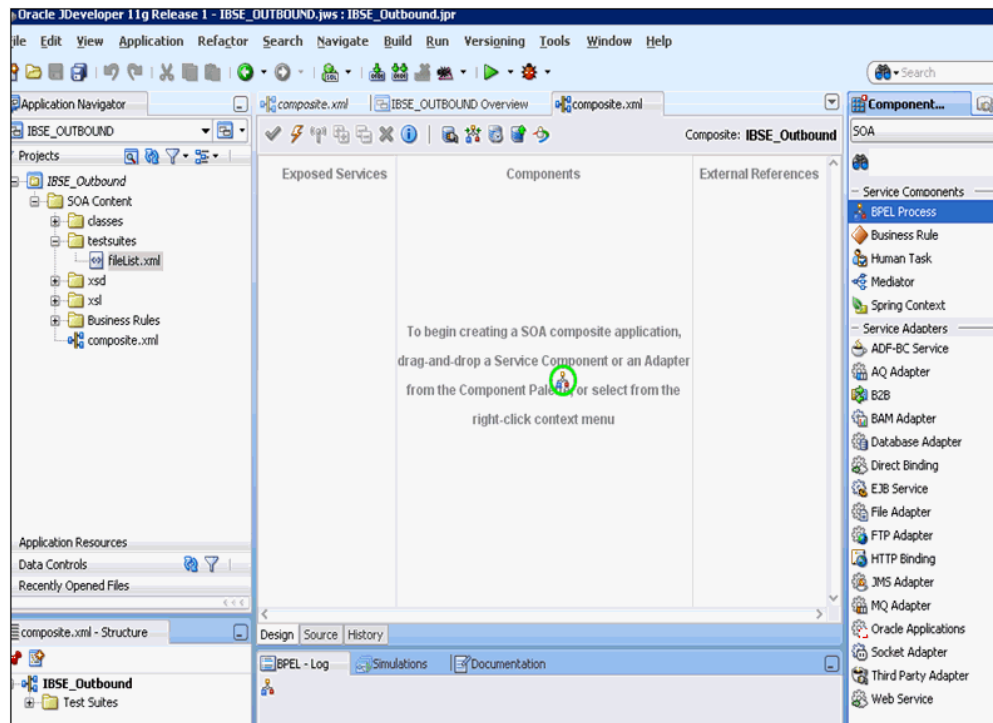
This section includes the following topics:

- [Section 4.6.3.1, "Creating a Partner Link"](#)
- [Section 4.6.3.2, "Creating BPEL Activities and Mappings With the Created Partner Link"](#)

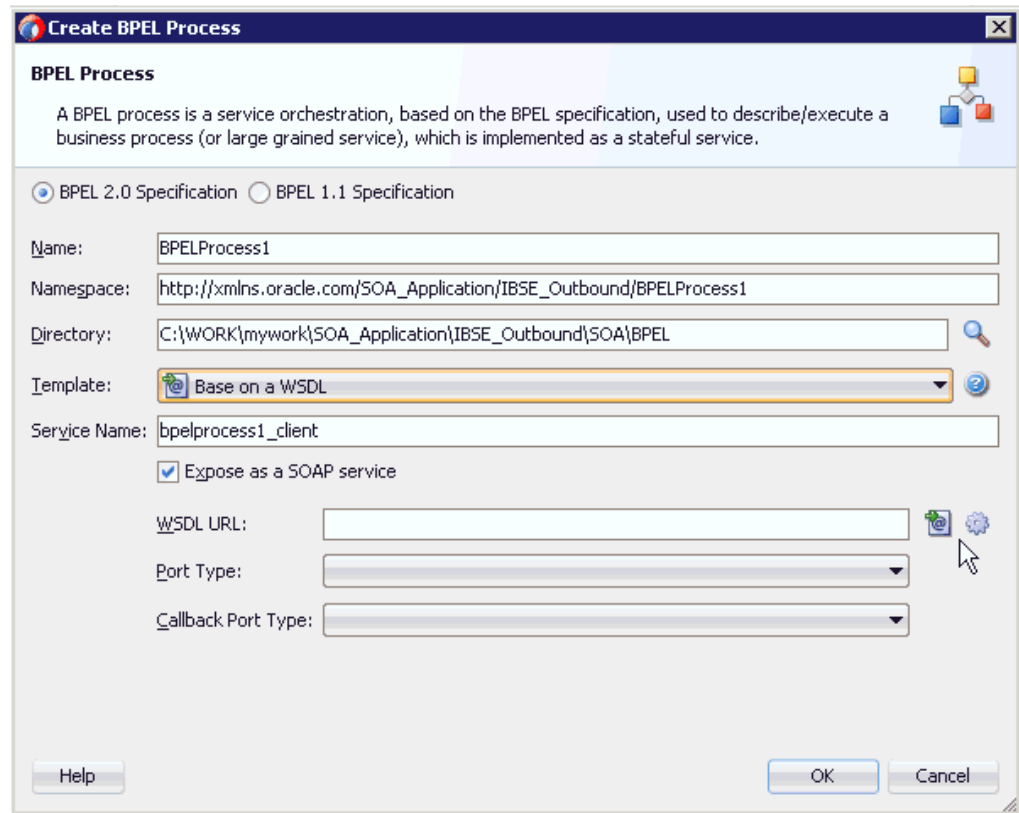
To define a BPEL outbound process:

1. Drag and drop the **BPEL Process** component from the Service Components pane to the Components pane, as shown in [Figure 4-77](#).

**Figure 4-77 BPEL Process Component**



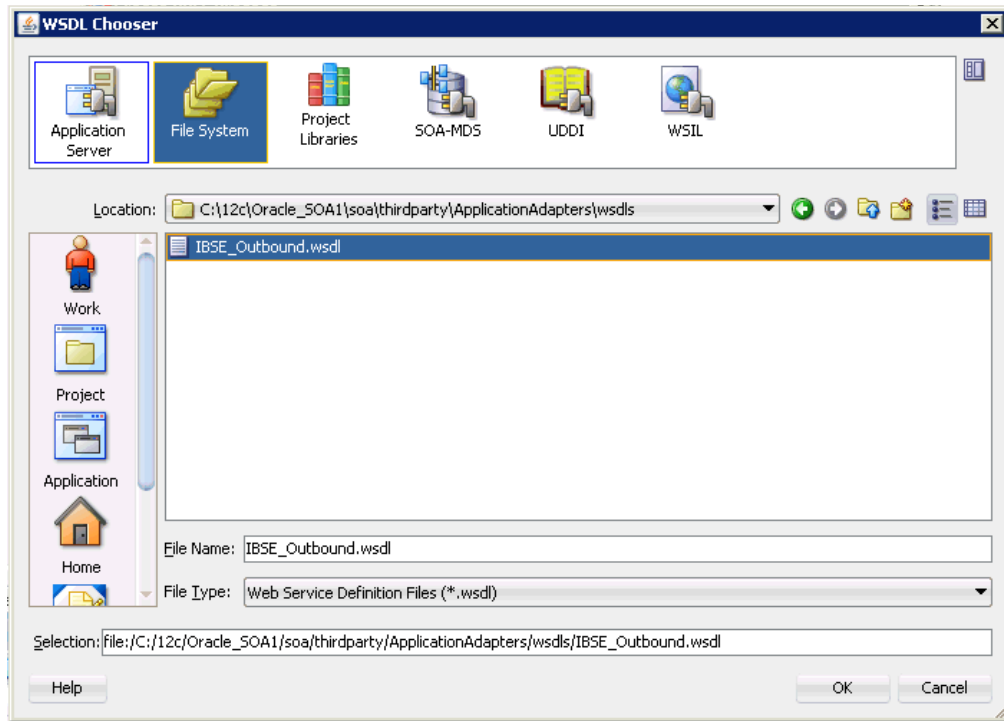
2. In the Name field, enter a name to identify the new outbound BPEL process component or leave it to the default value.  
By default, the BPEL 2.0 Specification option is selected.
3. From the Template drop-down list, select **Base on a WSDL**.
4. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field, as shown in [Figure 4-78](#).

**Figure 4–78 Find Existing WSDLs Icon**

The WSDL Chooser dialog is displayed.

5. Navigate to the location where the WSDL is exported from Application Explorer, select the WSDL, and click **OK**, as shown in [Figure 4–79](#).

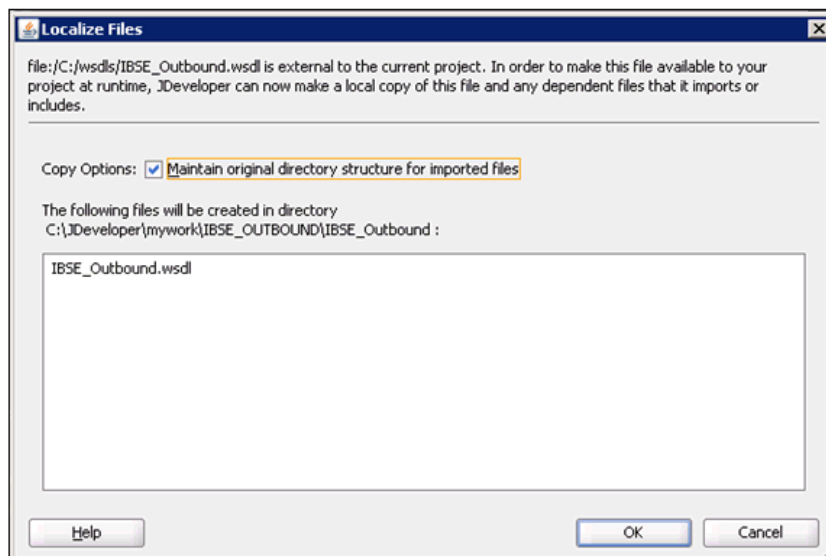
**Figure 4–79 WSDL Chooser Dialog**



The Localize Files window is displayed.

6. In the displayed Localize Files window, click **OK**. This imports the WSDL file to the project folder, as shown in [Figure 4–80](#).

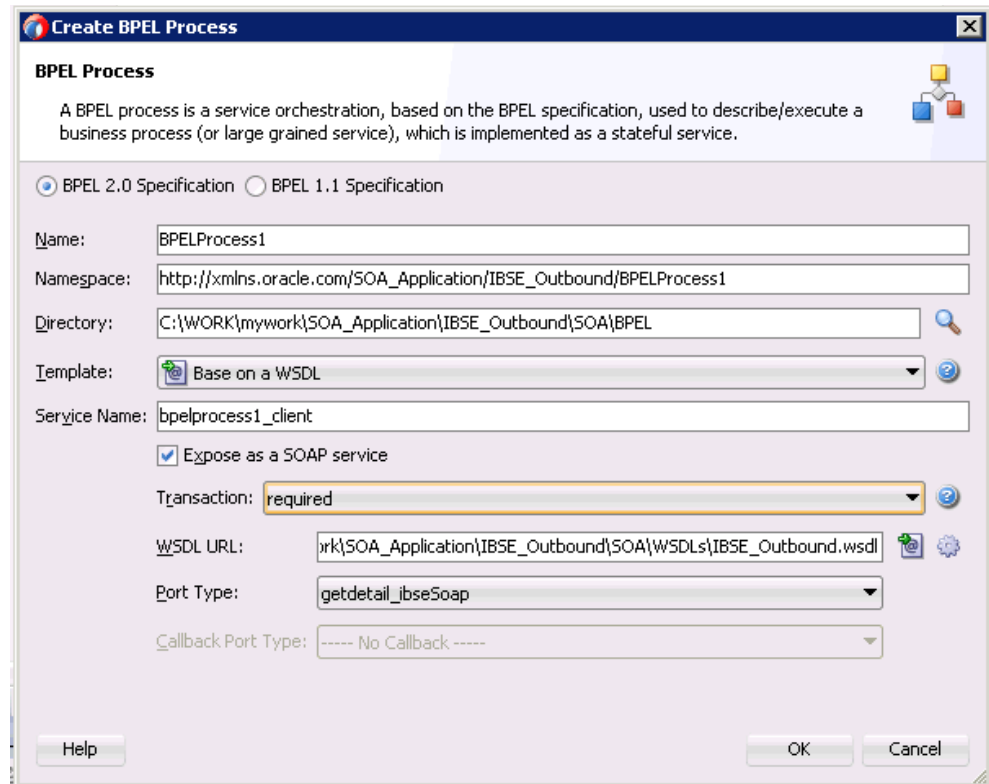
**Figure 4–80 Localize Files Window**



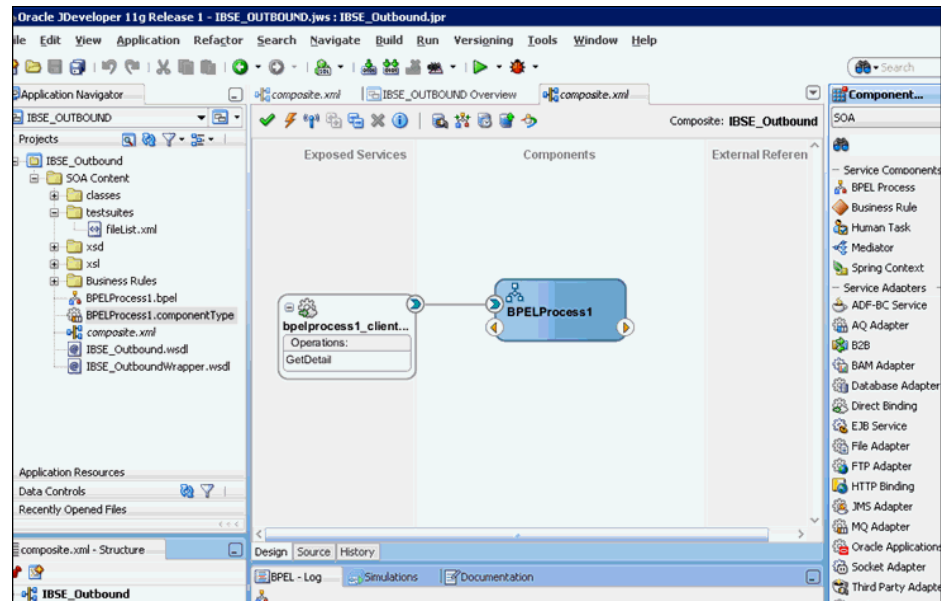
The Create BPEL Process window is displayed.

7. In the BPEL Process pane, click **OK**, as shown in [Figure 4–81](#).



**Figure 4–81 BPEL Process Pane**

The BPEL Process component is created and displayed, as shown in [Figure 4–82](#).

**Figure 4–82 BPEL Process Component**

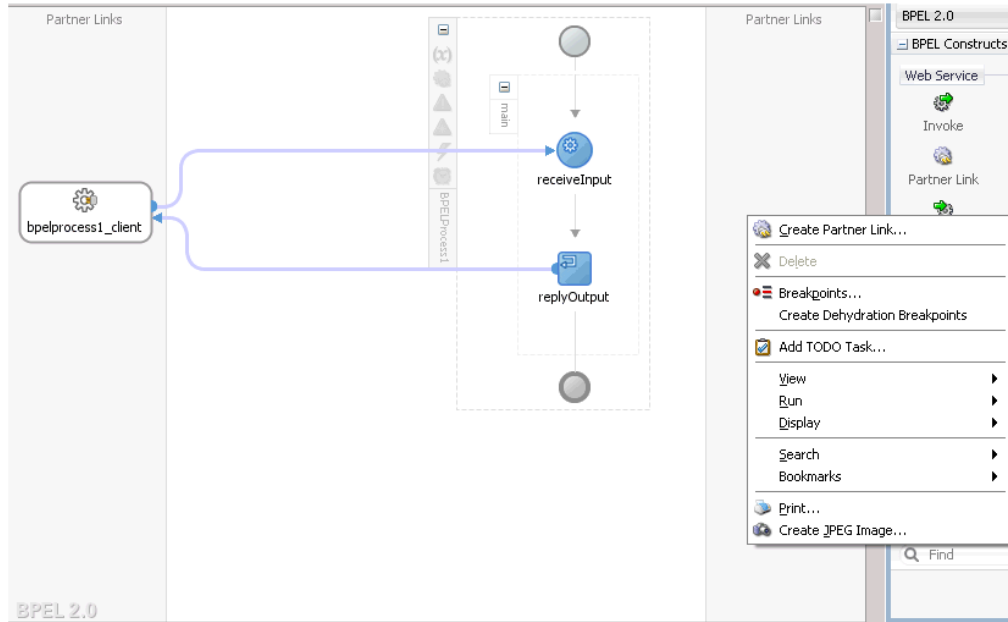
#### 4.6.3.1 Creating a Partner Link

This section describes how to create a partner link.

To create a partner link:

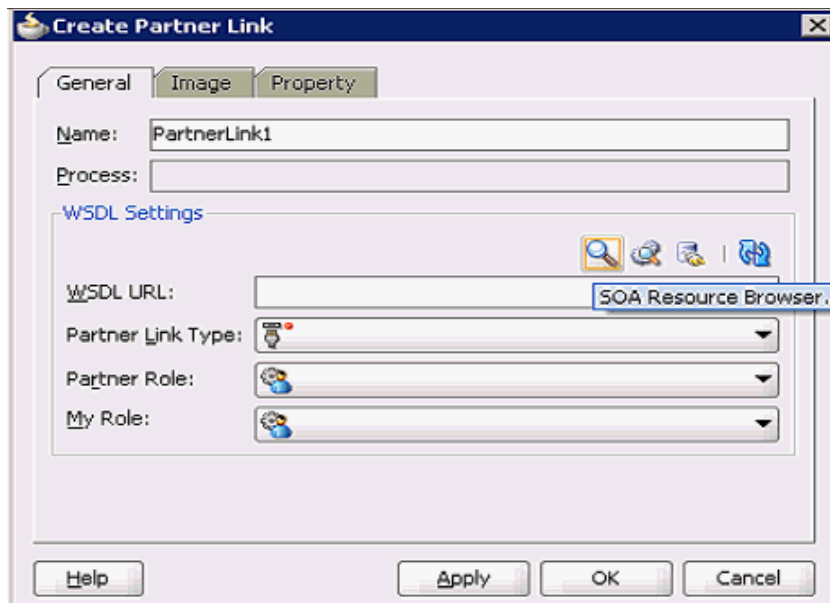
1. Double-click the outbound BPEL process component in the Components pane.
2. Right-click on the **Partner Links** pane and select **Create Partner Link**, as shown in [Figure 4-83](#).

**Figure 4-83 Create Partner Link**

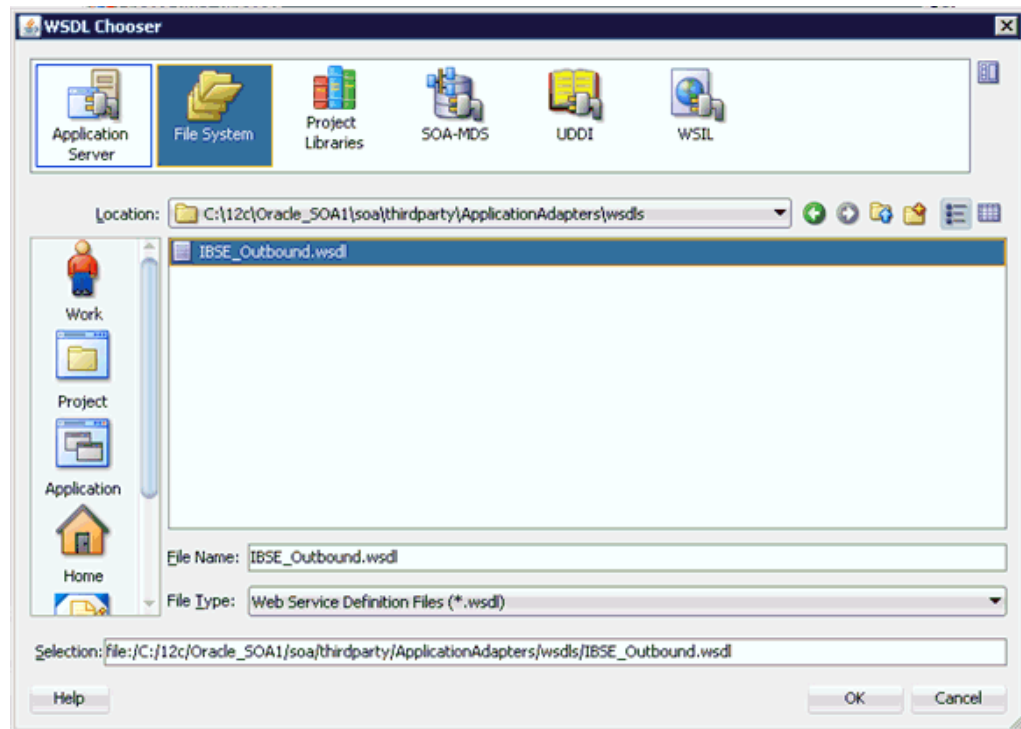


3. In the displayed Create Partner Link window, provide an appropriate name and click on the SOA Resource Browser tool, as shown in [Figure 4-84](#).

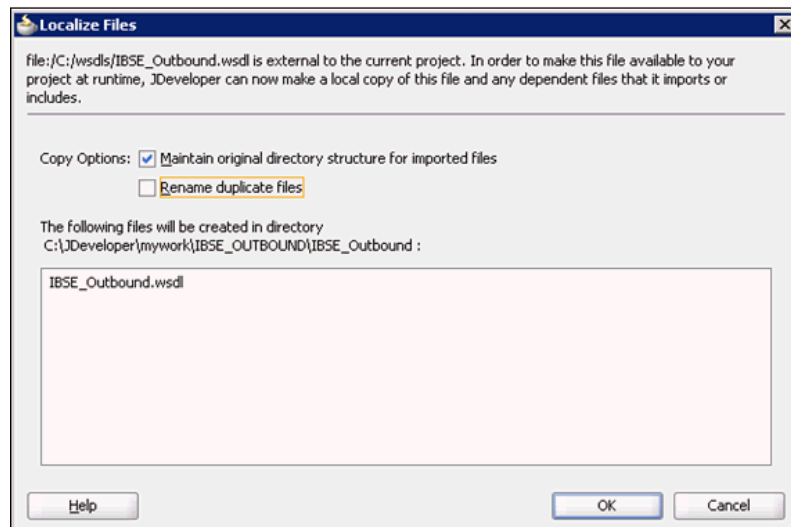
**Figure 4-84 SOA Resource Browser Tool**



4. In the WSDL Chooser dialog that is displayed, navigate to the location where the WSDL is exported from Application Explorer, select the WSDL, and click **OK**, as shown in [Figure 4-85](#).

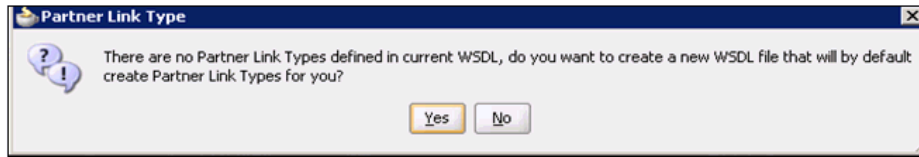
**Figure 4–85 WSDL Chooser Dialog**

5. In the displayed Localize Files window, uncheck the **Rename duplicate files** check box and click **OK**, as shown in [Figure 4–86](#).

**Figure 4–86 Localize Files Window**

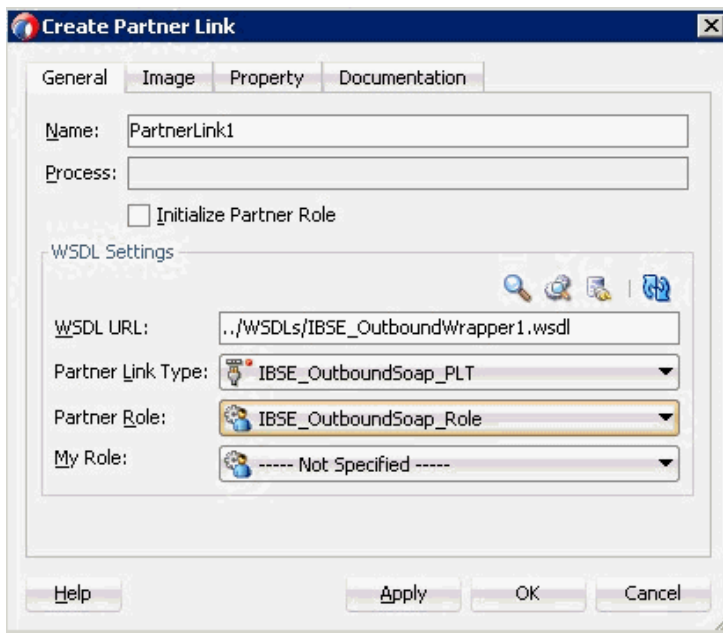
6. Click **Yes** in the displayed Partner Link Type window, as shown in [Figure 4–87](#).

**Figure 4–87 Partner Link Type**



7. In the displayed Create Partner Link window, expand the **Partner Role** drop-down list and select the available partner role.
8. Click **Apply**, and then **OK**, as shown in [Figure 4–88](#).

**Figure 4–88 Create Partner Link**

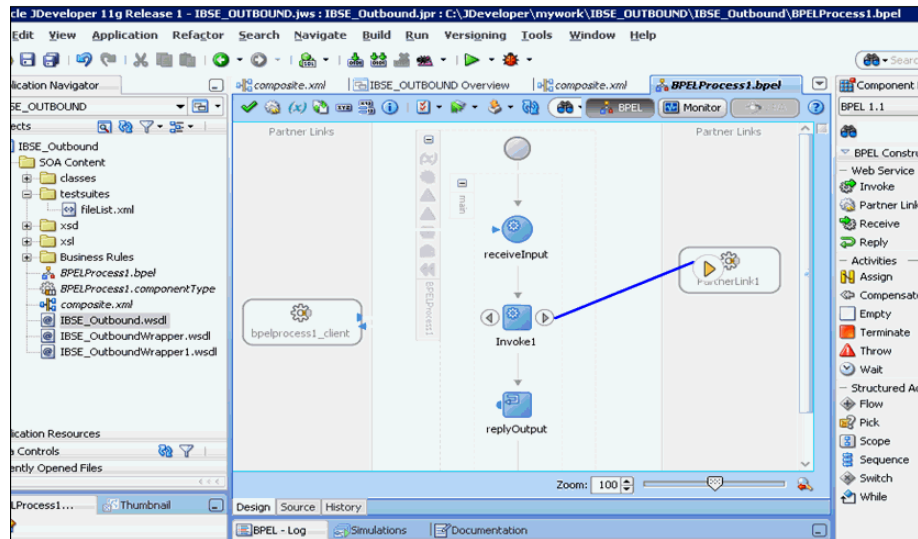


#### 4.6.3.2 Creating BPEL Activities and Mappings With the Created Partner Link

This section describes how to create BPEL activities and mappings with the created partner link.

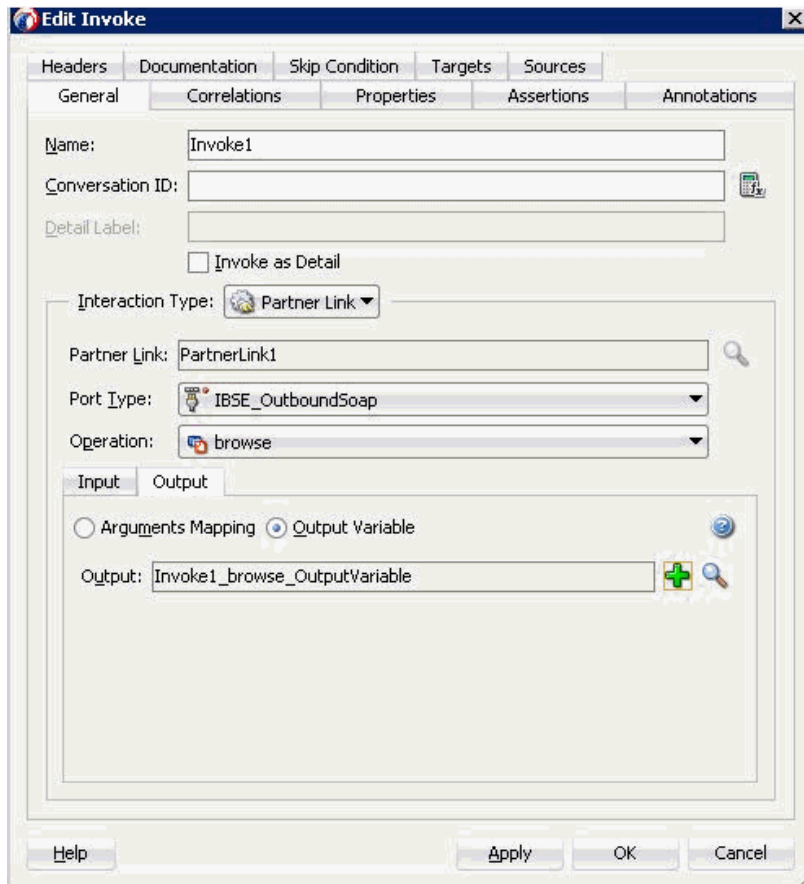
To create BPEL Activities and map with the created partner link:

1. Drag and drop the **Invoke** activity component from BPEL Constructs to the Components pane. Place it between the **receiveInput** activity component and the **replyOutput** activity component.
2. Create a connection between the new **Invoke** activity component (Invoke1) and the **Partner Link** component (Partner link1), as shown in [Figure 4–89](#).

**Figure 4–89 Partner Link Component**

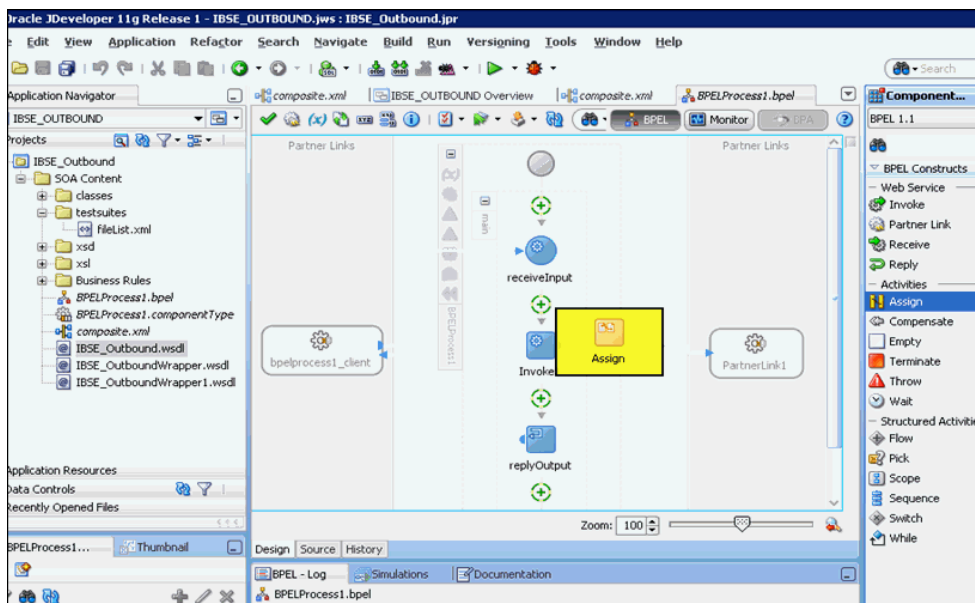
3. In the displayed Edit Invoke window, click the Plus (+) icon, located to the right of the Input field, to configure a new input variable.
4. Accept the default values that are provided for the new input variable and click **OK**.
5. Click the Plus (+) icon, which is located to the right of the Output field, to configure a new output variable.
6. Accept the default values that are provided for the new output variable and click **OK**.
7. Click **Apply** and then **OK**, as shown in [Figure 4–90](#).

**Figure 4–90 Edit Invoke Window**



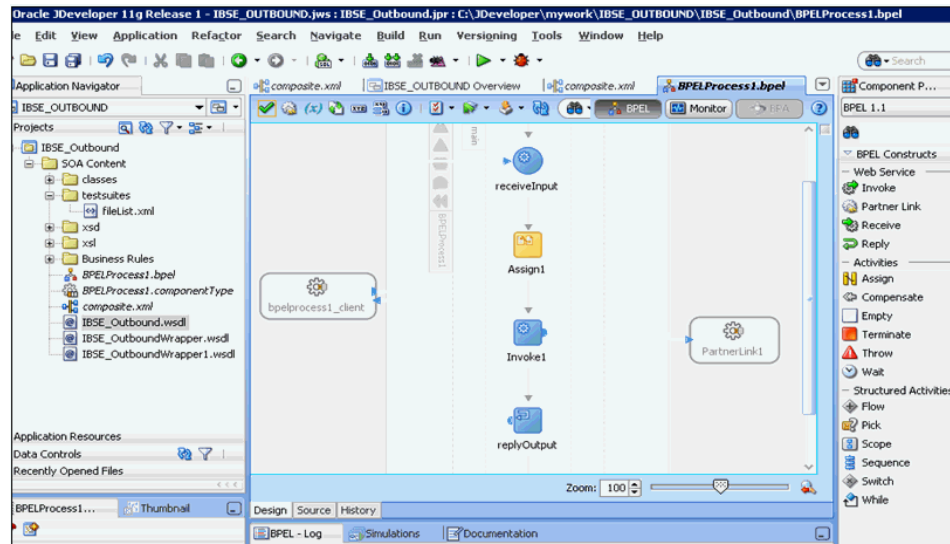
8. Drag and drop the **Assign** activity component from BPEL Constructs to the Components pane. Place it between the **Receive** activity component (receiveInput) and the **Invoke** activity component (Invoke1), as shown in Figure 4-91.

**Figure 4–91 Assign Activity Component**



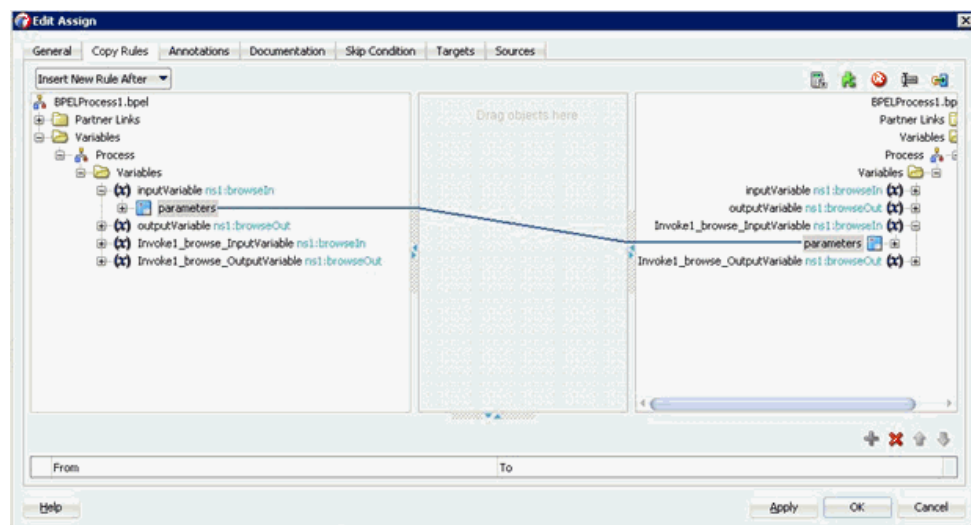
- Double-click the new **Assign** activity component (Assign1), as shown in Figure 4–92.

**Figure 4–92 Assign Activity Component**



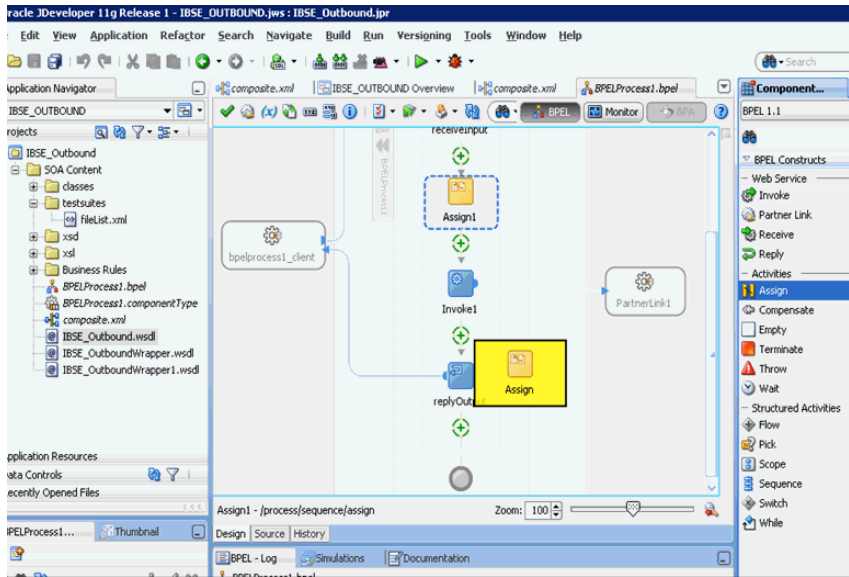
- In the left pane, under Variables, expand **InputVariable**, and then select **parameters**.
- In the right pane, under Variables, expand **Invoke1\_browse\_InputVariable**, and then select **parameters**.
- Drag and map the **InputVariable** parameters to the **Invoke1\_browse\_InputVariable** parameters, as shown in Figure 4–93.

**Figure 4–93 InputVariable Parameters**



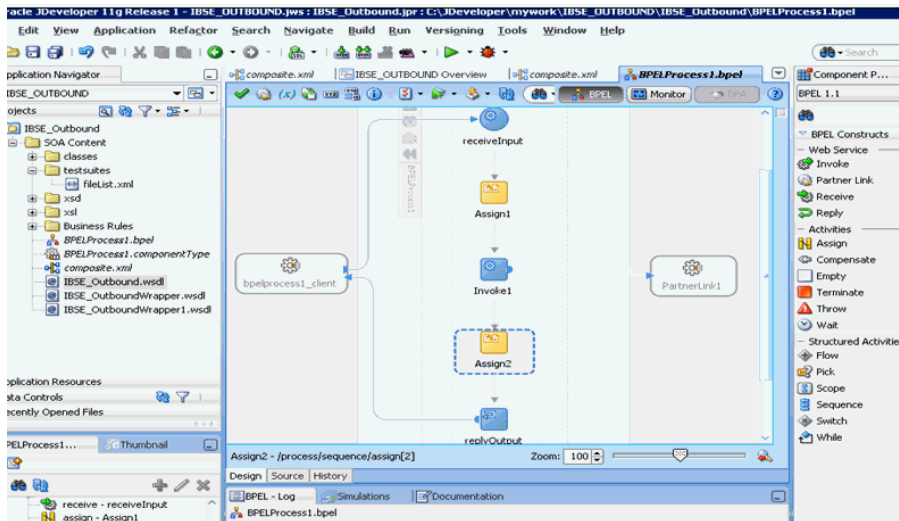
- Click **Apply** and then **OK**.
- Drag and drop the **Assign** activity component to the Components pane and place it between the **Invoke** activity (Invoke1) and the **Reply** activity (replyOutput), as shown in Figure 4–94.

**Figure 4–94 Assign Activity Component**



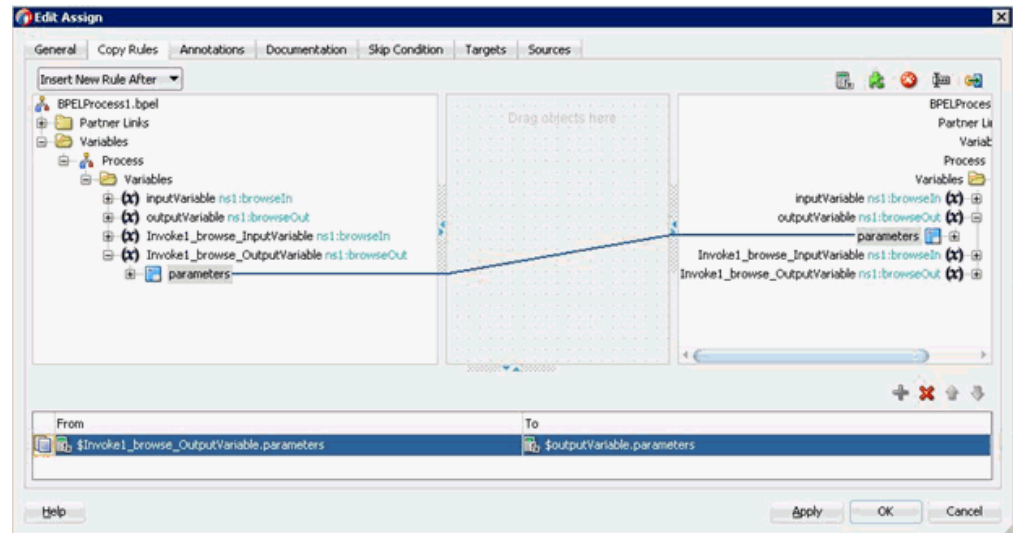
15. Double-click the new **Assign** activity component (Assign2), as shown in Figure 4–95.

**Figure 4–95 New Assign Activity Component**



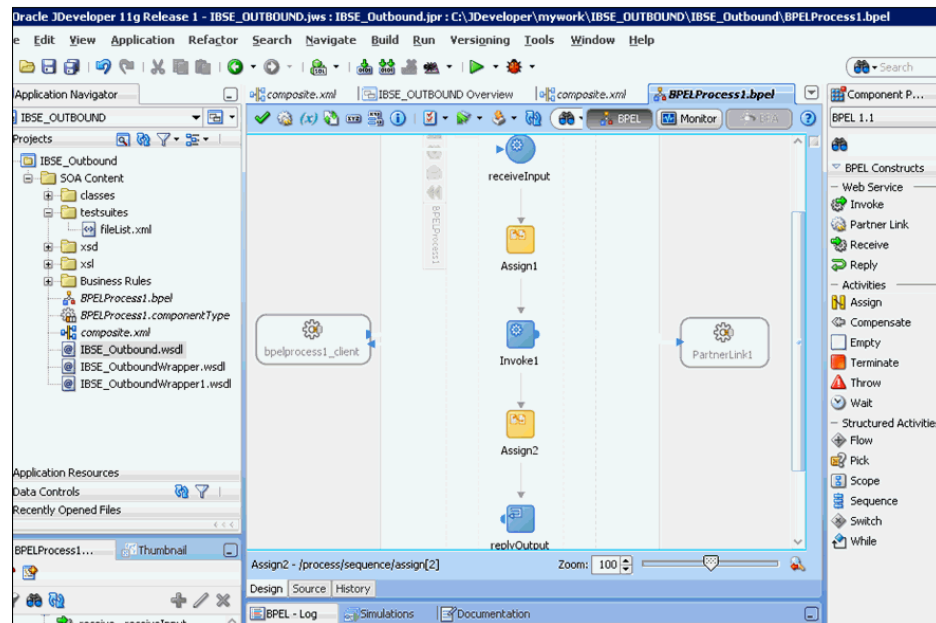
16. In the left pane, under Variables, expand **Invoke1\_browse\_OutputVariable**, and then select **parameters**.
17. In the right pane, under Variables, expand **outputVariable**, and then select **parameters**.
18. Drag and map the **Invoke1\_browse\_OutputVariable** parameters to the **outputVariable** parameters, as shown in Figure 4–96.



**Figure 4–96** *outputVariable Parameters*

19. Click **Apply** and then **OK**.

You are returned to the component pane, as shown in [Figure 4–97](#).

**Figure 4–97** *Component Pane*

20. Click the **Save All** icon in the menu bar to save the new outbound BPEL process component that was configured.

You are now ready to deploy the BPEL Outbound process. You can follow the same procedure as in [Section 4.4.4, "Deploying the BPEL Outbound Process"](#) on page 4-28.

Once deployed you can invoke the input XML, as defined in [Section 4.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console"](#) on page 4-31.



---

---

## Integration With Mediator Service Components in the Oracle SOA Suite

This chapter describes integration with Mediator service components in the Oracle SOA Suite. It contains the following topics:

- [Section 5.1, "Configuring a New Application Server Connection"](#)
- [Section 5.2, "Configuring a Mediator Outbound Process \(J2CA Configuration\)"](#)
- [Section 5.3, "Configuring a Mediator Inbound Process \(J2CA Configuration\)"](#)
- [Section 5.4, "Configuring a Mediator Outbound Process \(BSE Configuration\)"](#)

The scenarios shown in this chapter require the following prerequisites.

### Prerequisites

The following are installation and configuration requirements:

- Oracle Application Adapter for PeopleSoft must be installed on Oracle WebLogic Server.
- PeopleSoft must be configured for inbound and outbound processing.

**See Also:** *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server*

The examples in this chapter present the configuration steps necessary for demonstrating service and event integration with PeopleSoft. Prior to using this material, you must be familiar with the following:

- How to configure Oracle Application Adapter for PeopleSoft for services and events. For more information, see [Chapter 2, "Configuring Oracle Application Adapter for PeopleSoft"](#).
- How to configure a new Application Server and Integration Server connection in Oracle JDeveloper. For more information, see [Chapter 4, "Integration With BPEL Service Components in the Oracle SOA Suite"](#).

### Overview of Mediator Integration

Mediator provides a comprehensive application integration framework. Oracle Application Adapter for PeopleSoft used with Mediator enables you to seamlessly integrate enterprise software, eliminating the need to write custom code. Functional modeling, as opposed to custom coding solutions, allows for software reuse and reduces the complexity and management challenges that arise over the software lifecycle. This integration model consists of two components—high-level integration logic and low-level platform services.

Adapter integration with Oracle WebLogic Server, Mediator is a two-step process:

1. **Design Time:** Oracle Application Adapter for PeopleSoft is configured in Application Explorer for services and events, as described in [Chapter 2, "Configuring Oracle Application Adapter for PeopleSoft"](#). Integration logic is modeled in Oracle JDeveloper. Metadata are stored in repositories.
2. **Runtime:** The underlying platform treats this metadata as run-time instructions to enable the communication between participating applications.

## 5.1 Configuring a New Application Server Connection

For more information on how to configure a new Application Server connection in Oracle JDeveloper, see [Section 4.3, "Configuring a New Application Server Connection"](#).

## 5.2 Configuring a Mediator Outbound Process (J2CA Configuration)

This section describes how to configure a Mediator outbound process to your PeopleSoft system, using a Mediator project in Oracle JDeveloper.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\Mediator\J2CA\Outbound_Project
```

This section contains the following topics:

- [Section 5.2.1, "Creating an Empty Composite for SOA"](#)
- [Section 5.2.2, "Defining a Mediator Outbound Process"](#)
- [Section 5.2.3, "Deploying the Mediator Outbound Process"](#)
- [Section 5.2.4, "Invoking the Input XML Document in the Oracle Enterprise Manager Console"](#)

### Prerequisites

Before you design a Mediator outbound process, you must generate the respective WSDL file using Application Explorer. For more information, see [Section 4.4.1, "Generating WSDL for Request/Response Service"](#) on page 4-8.

### 5.2.1 Creating an Empty Composite for SOA

Perform the following steps to create an empty composite for SOA:

1. Create a new SOA application.
2. Enter a name for the new SOA Application and click **Next**.  
The Name your project page is displayed.
3. Enter a project name and click **Next**.  
The Configure SOA settings page is displayed.
4. From the Composite Template list, select **Empty Composite** and click **Finish**.

For more information, see [Section 4.4.2, "Creating an Empty Composite for SOA"](#) on page 4-9.

## 5.2.2 Defining a Mediator Outbound Process

This section describes how to define a Mediator outbound process, which consists of the following topics:

- [Section 5.2.2.1, "Configuring a Third Party Adapter Service Component"](#)
- [Section 5.2.2.2, "Configuring an Outbound Mediator Process Component"](#)
- [Section 5.2.2.3, "Configuring the Routing Rules"](#)
- [Section 5.2.2.4, "Adjusting for Known Deployment Issues With 12c"](#)

### 5.2.2.1 Configuring a Third Party Adapter Service Component

Perform the following steps to create a third party adapter service component:

1. Drag and drop the **Third Party Adapter** component from the Service Adapters pane to the External References pane.
2. Enter a name for the third party adapter service.
3. Ensure that **Reference** is selected from the Type drop-down list (default).
4. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.
5. Browse and select an outbound WSDL file from the following directory:

`<ADAPTER_HOME>\wsdl.s`

6. Click **OK**.
7. Click **OK**.

The outbound WSDL file and associated request and response XML schema files (.xsd) are imported to the project folder that has been created.

8. Click the **Find JCA file** icon, which is located to the right of the JCA File field.
9. Browse and select the JCA properties file from the following directory:

`<ADAPTER_HOME>\wsdl.s`

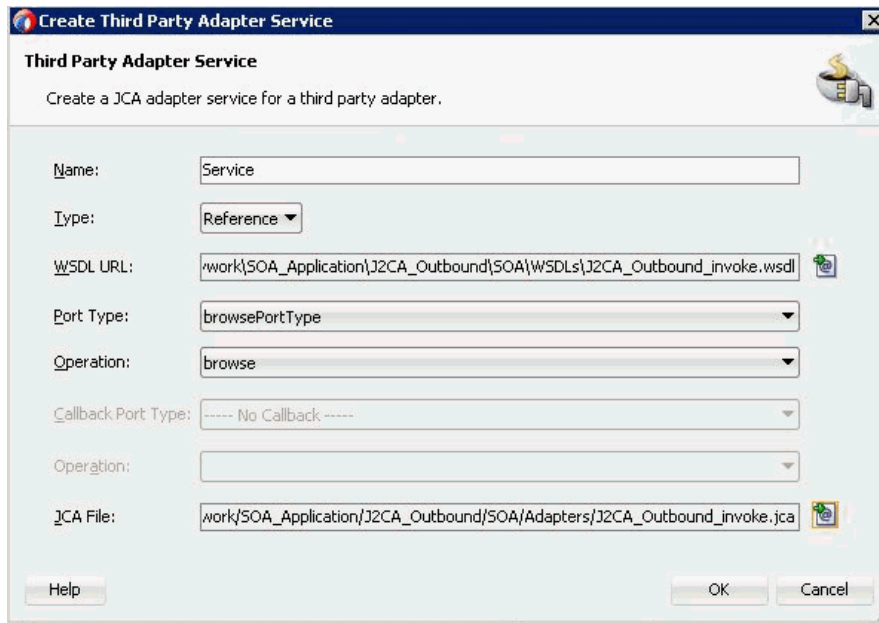
10. Click **OK**.

A Copy File confirmation message is displayed.

11. Click **Yes**.

A copy of the JCA properties file is made in the project folder.

**Figure 5–1 Create Third Party Adapter Service Dialog**



12. Click **OK**.

The third party adapter service component (GetDetail) is created in the External References pane.

You are now ready to configure an outbound Mediator process component.

For more information, see [Section 6.4.3.1, "Configuring a Third Party Adapter Service Component"](#) on page 6-11.

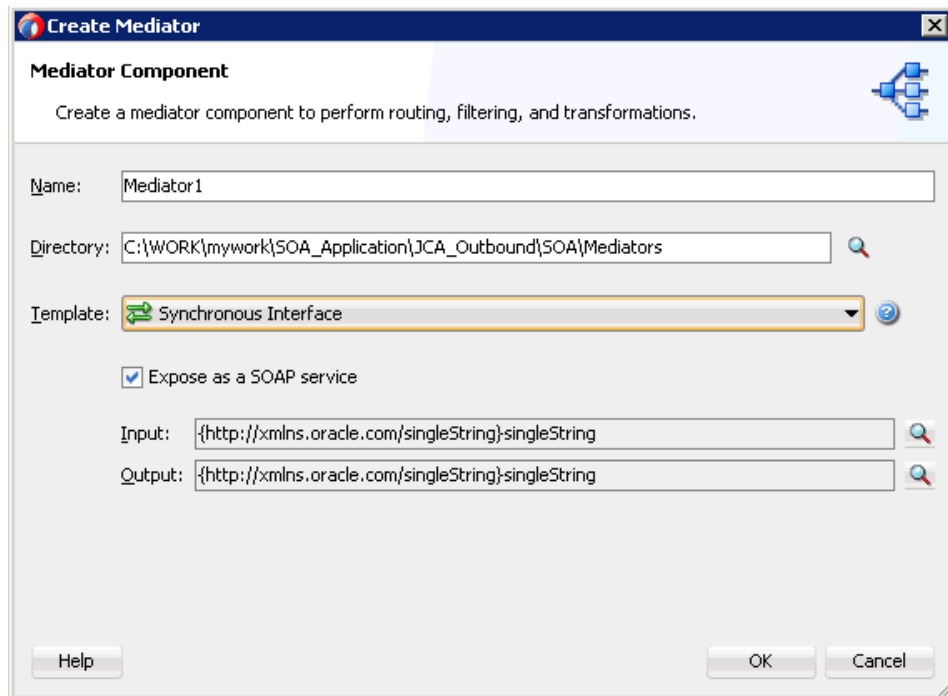
### 5.2.2.2 Configuring an Outbound Mediator Process Component

Perform the following steps to configure an outbound Mediator process component:

1. Drag and drop the **Mediator Process** component from the Components pane to the Components pane.

The Create Mediator dialog is displayed, as shown in [Figure 5–2](#).

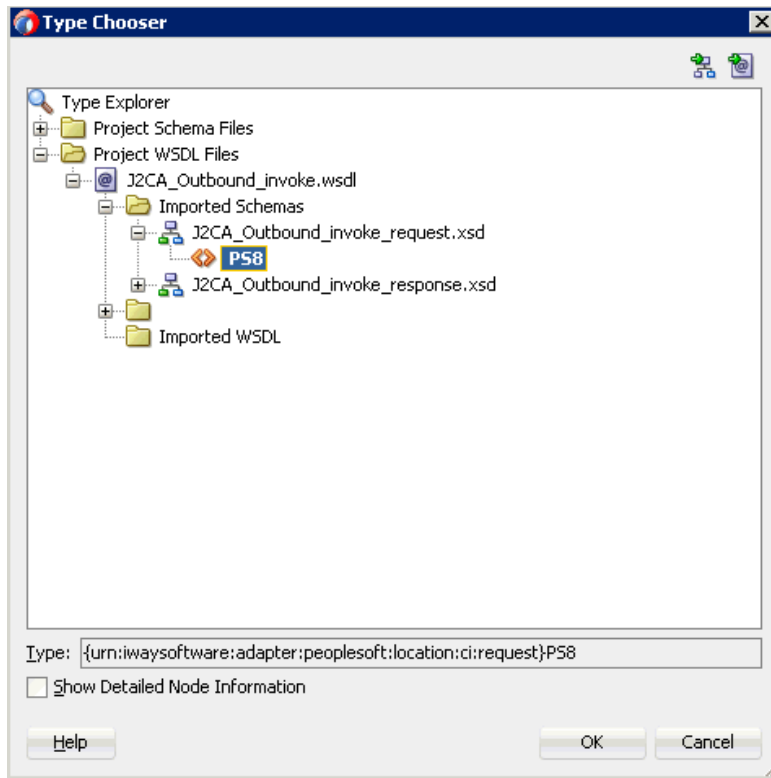
Figure 5–2 Create Mediator Dialog



2. In the Name field, enter a name to identify the new outbound Mediator process component or leave it to the default value.
3. From the Template drop-down list, select **Synchronous Interface**.
4. Click the **Browse** icon, which is located to the right of the Input field to select the associated XML request schema file.

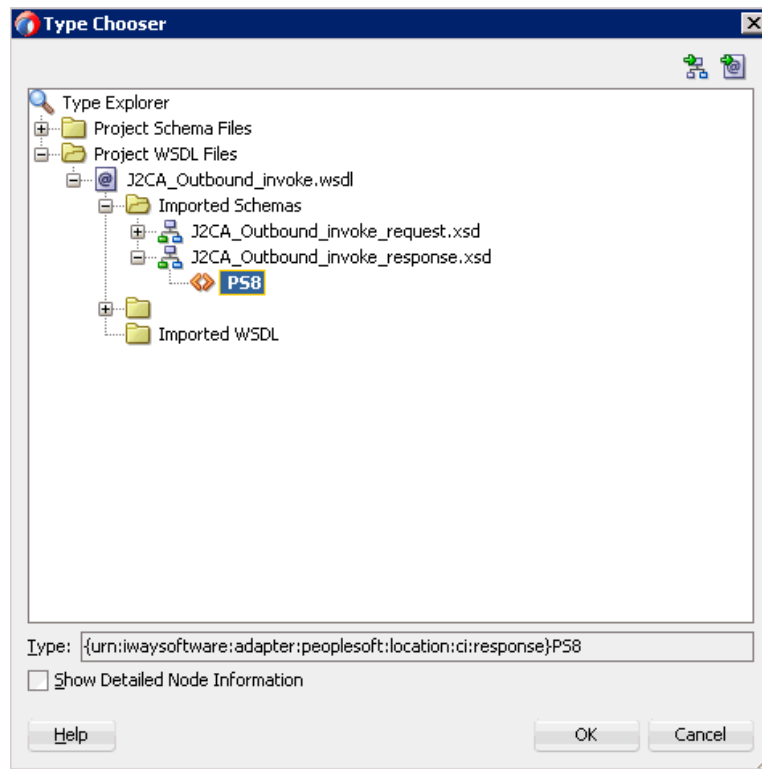
The Type Chooser dialog is displayed, as shown in [Figure 5–3](#).

**Figure 5–3 Type Chooser Dialog**



5. Expand **Project WSDL Files**, **J2CA\_Outbound\_invoke.wsdl**, **Imported Schemas**, **J2CA\_Outbound\_invoke\_request.xsd**, and select **PS8**.
6. Click **OK**.  
You are returned to the Create Mediator dialog.
7. Click the **Browse** icon, which is located to the right of the Output field to select the associated XML response schema file.  
The Type Chooser dialog is displayed, as shown in [Figure 5–4](#).

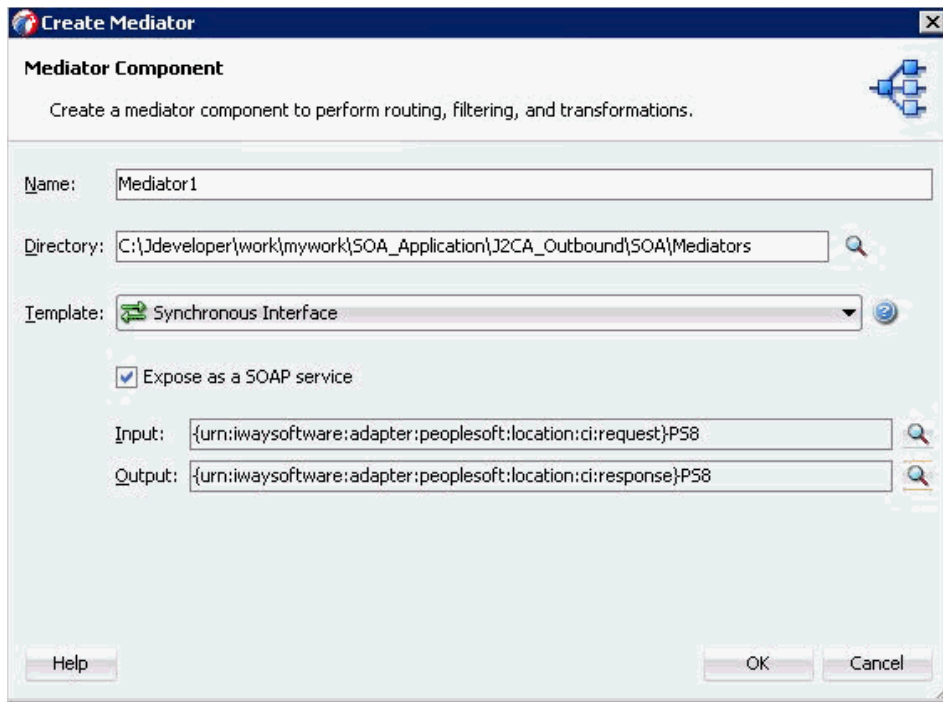


**Figure 5–4 Type Chooser Dialog**

8. Expand **Project WSDL Files**, **J2CA\_Outbound\_invoke.wSDL**, **Imported Schemas**, **J2CA\_Outbound\_invoke\_response.xsd**, and select **PS8**.
9. Click **OK**.

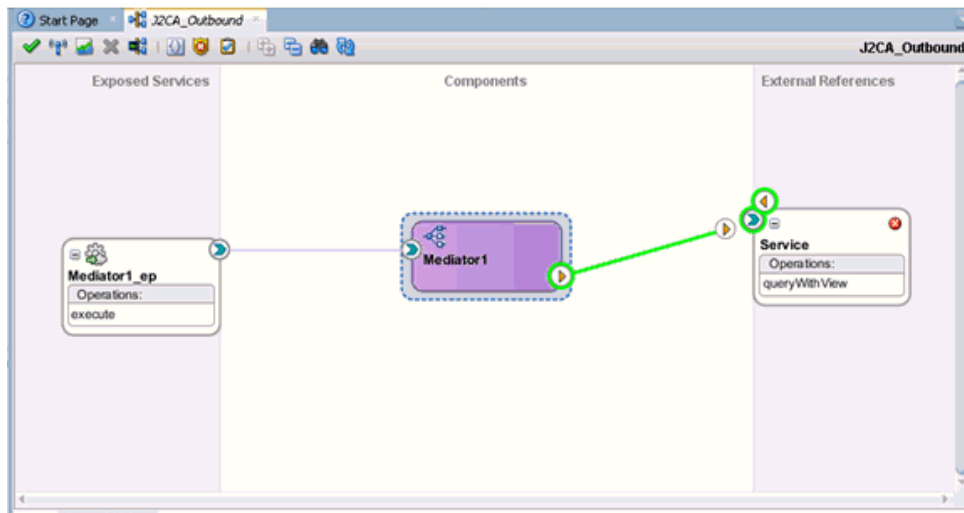
You are returned to the Create Mediator dialog, as shown in [Figure 5–5](#).

**Figure 5–5 Create Mediator Dialog**



10. Click OK.
11. Create a connection between the outbound Mediator process component and the third party adapter service component, as shown in [Figure 5–6](#).

**Figure 5–6 Created Connection**



You are now ready to configure the routing rules.

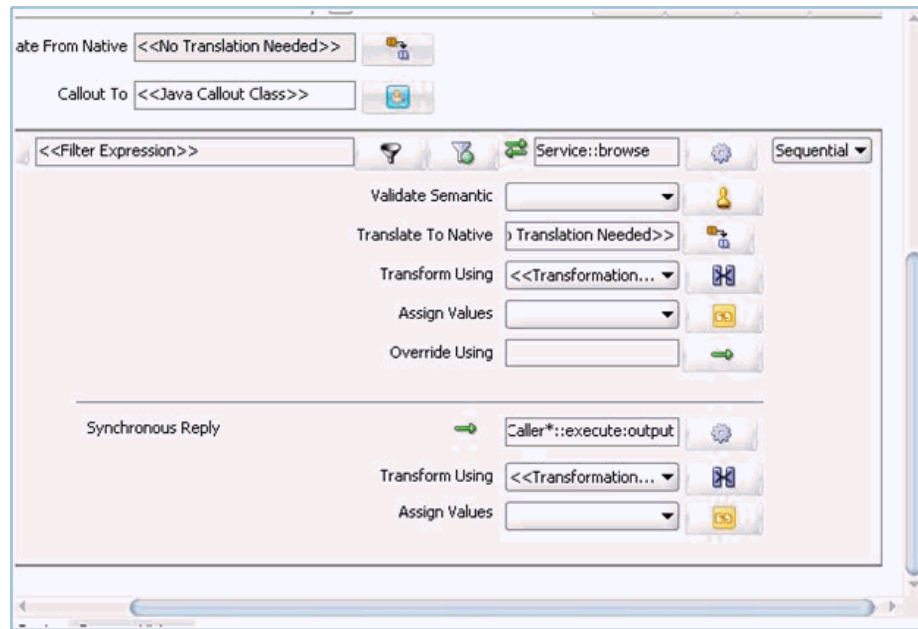
### 5.2.2.3 Configuring the Routing Rules

Perform the following steps to configure routing rules for the Mediator outbound process component:

1. Double-click the outbound Mediator process component in the Components pane.

The Routing Rules dialog is displayed, as shown in [Figure 5-7](#).

**Figure 5-7 Routing Rules Dialog**



2. In the <<Filter Expression>> area, click the icon to the right of the Transform Using field.

The Request Transformation Map dialog is displayed, as shown in [Figure 5-8](#).

**Figure 5-8 Request Transformation Map Dialog**

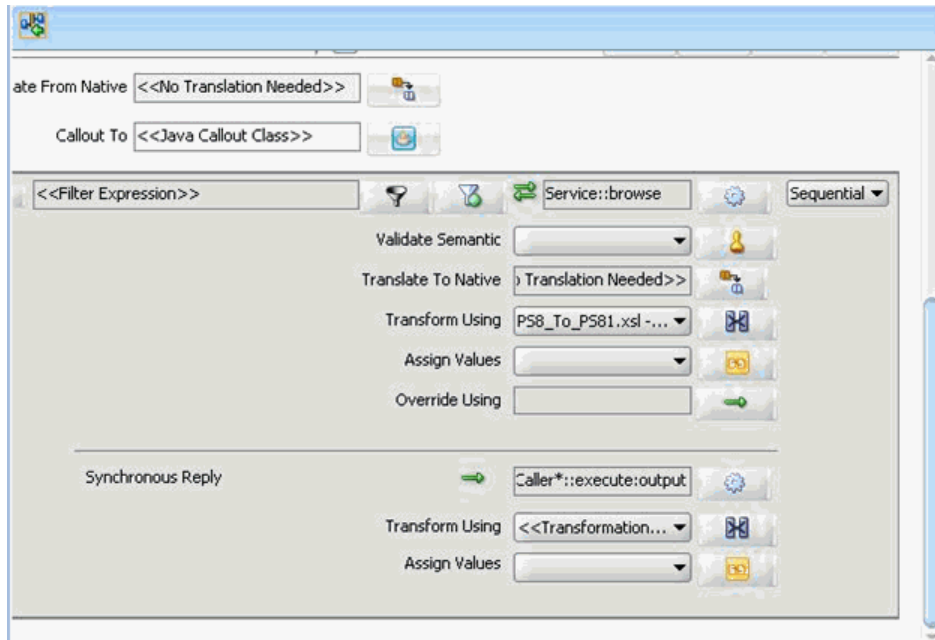


3. Click the Add (+) icon.
 

The Create Transformation Map page is displayed.
4. Make sure the Type is selected as **XSLT** and click **OK**.
5. Click **OK**.
6. Map the **ns0:PS8** source element to the **ns0:PS8** target element.
 

The Auto Map Preferences dialog is displayed.
7. Retain the default values and click **OK**.
8. Return to the Routing Rules dialog, as shown in [Figure 5-9](#).

**Figure 5–9 Routing Rules Dialog**



9. In the Synchronous Reply area, click the icon to the right of the Transform Using field.

The Reply Transformation Map dialog is displayed.

10. Click the Add (+) icon.

The create Transformation Page is displayed.

11. Make sure the type is selected as **XSLT** and click **OK**.

A mapping page is displayed.

12. Click **OK**.

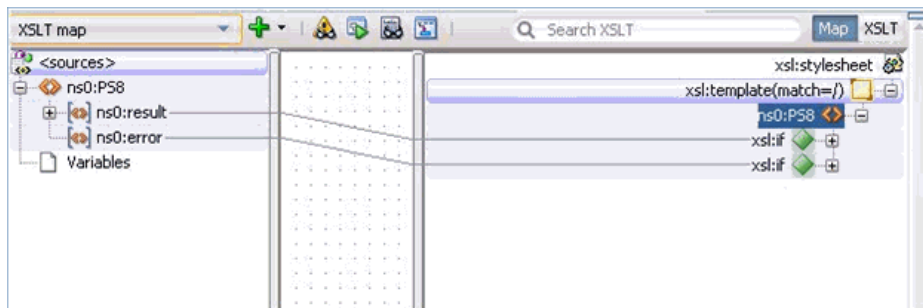
13. Map the **ns0:PS8** source element to the **ns0:PS8** target element.

The Auto Map Preferences dialog is displayed.

14. Retain the default values and click **OK**.

The mapping is completed, as shown in [Figure 5–10](#).

**Figure 5–10 Completed Mapping**



15. Click the **Save All** icon in the menu bar to save the new outbound Mediator process component that was configured.

### 5.2.2.4 Adjusting for Known Deployment Issues With 12c

For more information on how to adjust for known deployment issues with 12c, see [Section 4.4.3.3, "Adjusting for Known Deployment Issues With 12c"](#) on page 4-26.

## 5.2.3 Deploying the Mediator Outbound Process

Perform the following steps to deploy the Mediator outbound process.

1. Right-click the project name in the left pane, select **Deploy**, and then click **J2CA\_Outbound**.

The Deployment Action page is displayed.

2. Ensure that **Deploy to Application Server** is selected.
3. Click **Next**.

The Deploy Configuration page is displayed.

4. Leave the default values selected and click **Next**.

The Select Server page is displayed.

5. Select an available application server that was configured and click **Next**.

The SOA Servers page is displayed.

6. Select a target SOA server and click **Next**.

The Summary page is displayed.

7. Review and verify all the available deployment information for your project and click **Finish**.

For more information, see [Section 4.4.4, "Deploying the BPEL Outbound Process"](#) on page 4-28.

## 5.2.4 Invoking the Input XML Document in the Oracle Enterprise Manager Console

For more information, see [Section 4.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console"](#) on page 4-31.

## 5.3 Configuring a Mediator Inbound Process (J2CA Configuration)

This section describes how to configure a Mediator inbound process to your PeopleSoft system, using a Mediator project in Oracle JDeveloper.

A sample project has been provided for this inbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\Mediator\J2CA\Inbound_Project
```

This section contains the following topics:

- [Section 5.3.1, "Creating an Empty Composite for SOA"](#)
- [Section 5.3.2, "Defining a Mediator Inbound Process"](#)

### Prerequisites

Before you design a Mediator inbound process, you must generate the respective WSDL file using Application Explorer. For more information, see [Section 4.5.1, "Generating WSDL for Event Integration"](#) on page 4-34.

### 5.3.1 Creating an Empty Composite for SOA

Perform the following steps to create an empty composite for SOA:

1. Create a new SOA application.
2. Enter a name for the new SOA Application and click **Next**.  
The Name your project page is displayed.
3. Enter a project name and click **Next**.  
The Configure SOA settings page is displayed.
4. From the Composite Template list, select **Empty Composite** and click **Finish**.  
For more information, see [Section 4.4.2, "Creating an Empty Composite for SOA"](#) on page 4-9.

### 5.3.2 Defining a Mediator Inbound Process

This section describes how to define a Mediator inbound process, which contains the following topics:

- [Section 5.3.2.1, "Configuring a Third Party Adapter Service Component"](#)
- [Section 5.3.2.2, "Configuring an Inbound Mediator Process Component With a File Adapter"](#)
- [Section 5.3.2.3, "Configuring the Routing Rules"](#)
- [Section 5.3.2.4, "Adjusting for Known Deployment Issues With 12c"](#)

#### 5.3.2.1 Configuring a Third Party Adapter Service Component

Perform the following steps to create a third party adapter service component:

1. Drag and drop the **Third Party Adapter** component from the Service Adapters pane to the Exposed Services pane.  
The Create Third Party Adapter Service dialog is displayed.
2. Enter a name for the third party adapter service.
3. Ensure that **Service** is selected from the Type drop-down list (default).
4. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.  
The WSDL Chooser dialog is displayed.
5. Browse and select an inbound WSDL file from the following directory:  
`<ADAPTER_HOME>\wsdl.s`
6. Click **OK**.  
The Localize Files dialog is displayed.
7. Click **OK**.  
The inbound WSDL file and associated receive/request schema file (.xsd) are imported to the project folder that has been created.  
You are returned to the Create Third Party Adapter Service dialog.
8. Click the **Find JCA file** icon, which is located to the right of the JCA File field.  
The Transformation Chooser dialog is displayed.

9. Browse and select the JCA properties file from the following directory:

<ADAPTER\_HOME>\wsdl.s

10. Click **OK**.

The Copy File Confirmation message is displayed.

11. Click **Yes**.

A copy of the JCA properties file is made in the project folder.

You are returned to the Create Third Party Adapter Service dialog.

12. Click **OK**.

The third party adapter service component is created in the Exposed Services pane.

You are now ready to configure an inbound Mediator process component.

For more information, see [Section 4.5.3.1, "Creating a Third Party Adapter Service Component"](#) on page 4-38.

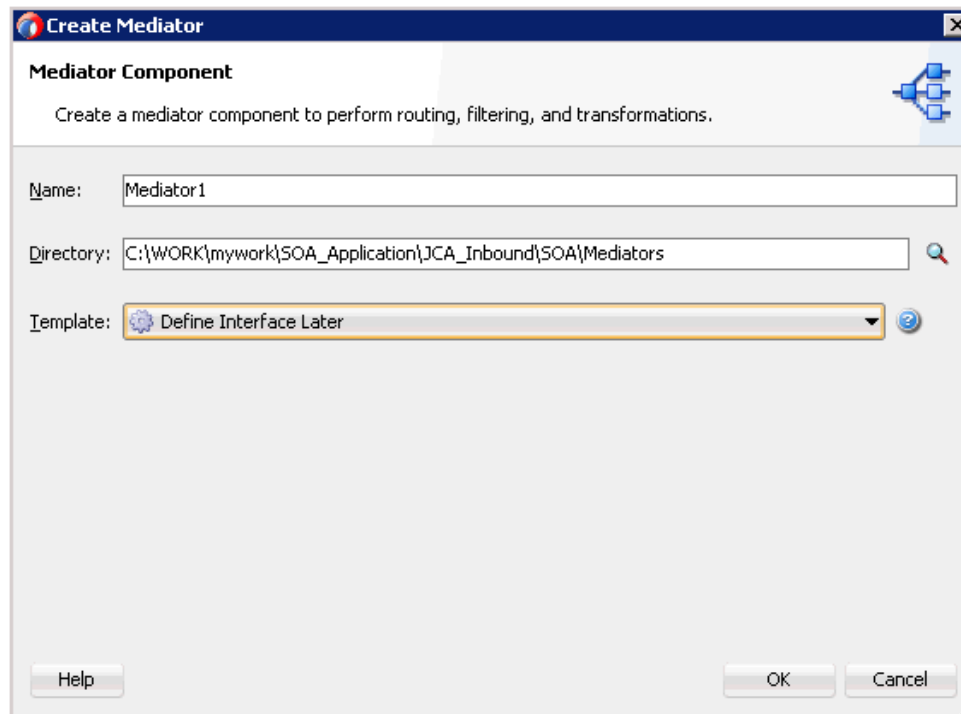
### 5.3.2.2 Configuring an Inbound Mediator Process Component With a File Adapter

Perform the following steps to configure an inbound Mediator process component with a File adapter.

1. Drag and drop the **Mediator Process** component from the Service Components pane to the Components pane.

The Create Mediator dialog is displayed, as shown in [Figure 5–11](#).

**Figure 5–11** Create Mediator Dialog



2. In the Name field, enter a name to identify the new inbound Mediator process component.

3. From the Template drop-down list, select **Define Interface Later**.
4. Click the **OK**.

The new Mediator process component is added to the Components pane.
5. Drag and drop the **File** component from the Technology Adapters pane to the External References pane.

The File Adapter Configuration Wizard is displayed.
6. Type a name for the new File adapter and click **Next**.

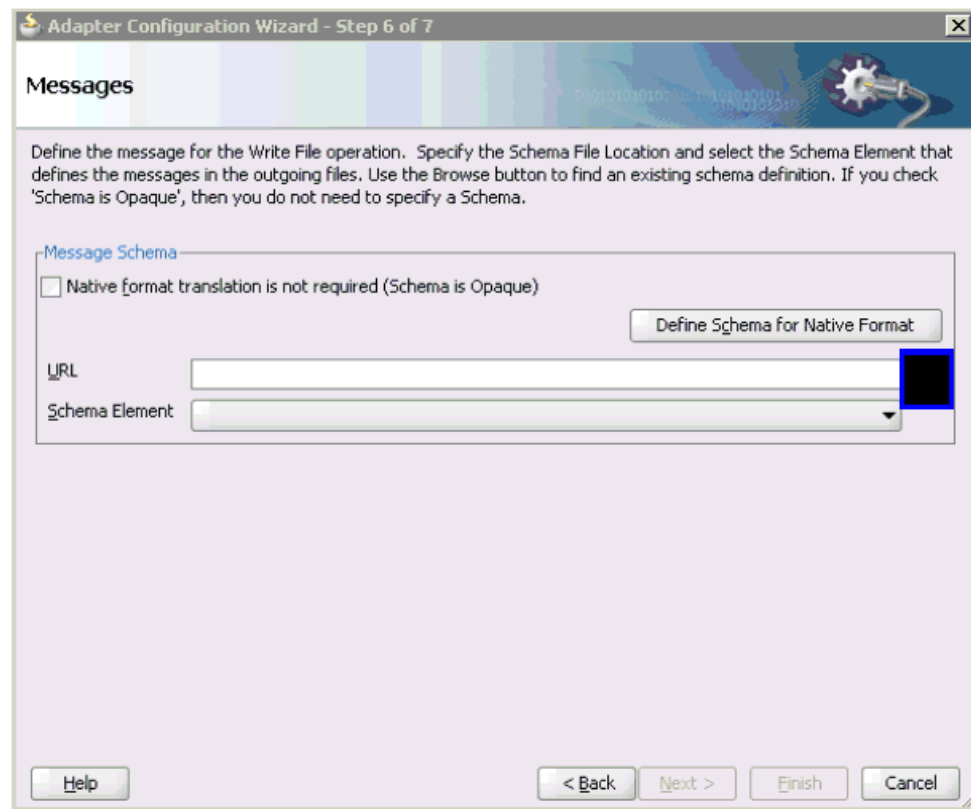
The Adapter Interface page is displayed.
7. Ensure that the **Define from operation and schema (specified later)** option is selected.
8. Click **Next**.

The Operation page is displayed.
9. Click **Next**.
10. Select **Write File** from the list of Operation Type options and specify an Operation Name (for example, Write).
11. Click **Next**.

The File Configuration page is displayed.
12. Specify a location on your file system where the output file is written.
13. In the File Naming Convention field, specify a name for the output file.
14. Click **Next**.

The Messages page is displayed, as shown in [Figure 5-12](#).

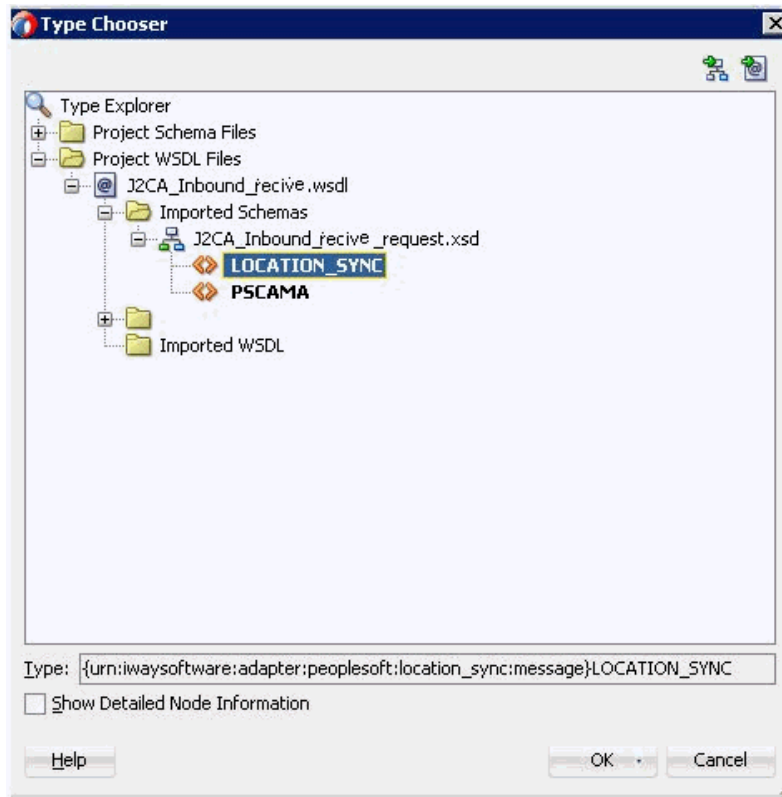


**Figure 5–12 Messages Page**

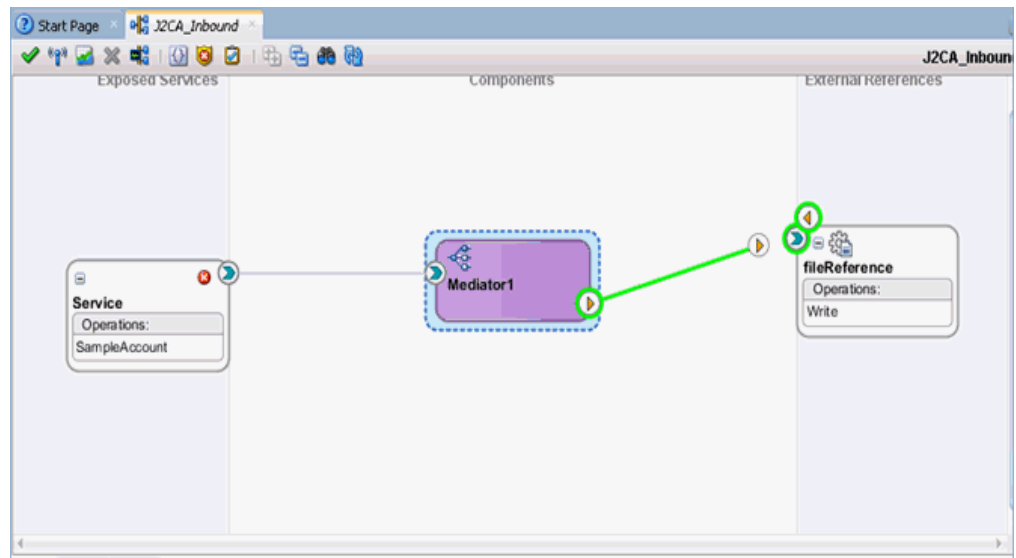
15. Click **Browse**, which is located to the right of the URL field.

The Type Chooser dialog is displayed, as shown in [Figure 5–13](#).

**Figure 5–13 Type Chooser Dialog**



16. Expand **Project WSDL Files**, **J2CA\_Inbound\_receive.wsdl**, **Imported Schemas**, **J2CA\_Inbound\_receive\_request.xsd**, and select **LOCATION\_SYNC**.
17. Click **OK**.  
You are returned to the Messages page.
18. Click **Next**.  
The Finish page is displayed.
19. Click **Finish**.
20. Create a connection between the inbound Mediator process component and the third party adapter service component.
21. Create a connection between the inbound Mediator process component and the File adapter component, as shown in [Figure 5–14](#).

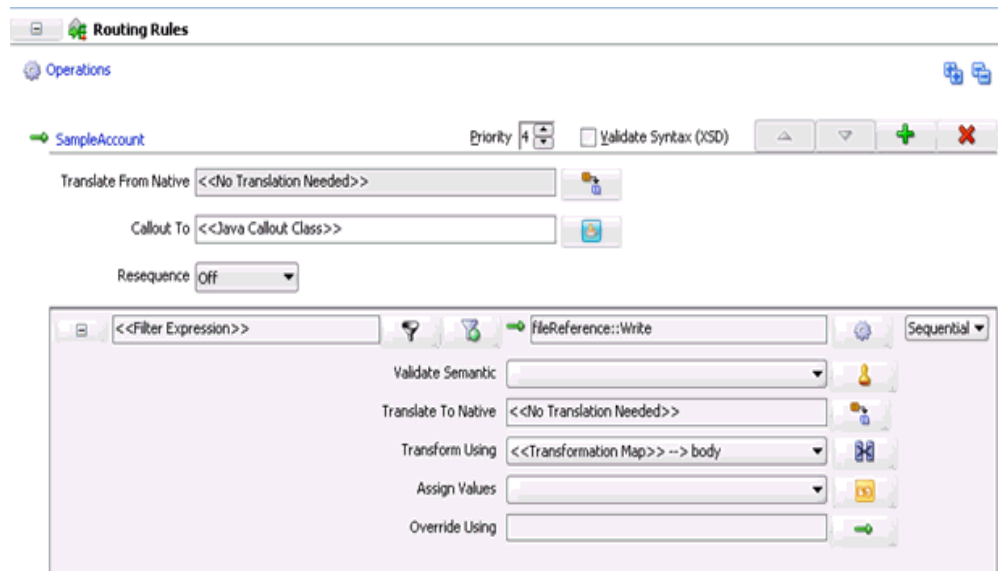
**Figure 5–14 Created Connection**

You are now ready to configure the routing rules.

### 5.3.2.3 Configuring the Routing Rules

Perform the following steps to configure routing rules for the Mediator inbound process component:

1. Double-click the inbound Mediator process component in the Components page.  
The Routing Rules dialog is displayed, as shown in [Figure 5–15](#).

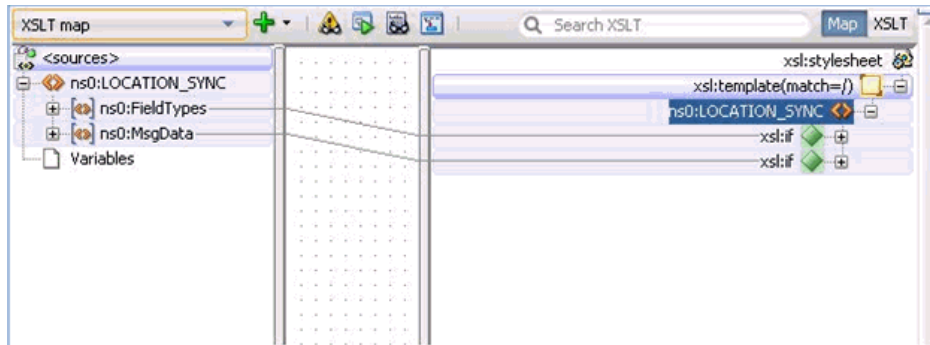
**Figure 5–15 Routing Rules Dialog**

2. In the '<<Filter Expression>>' area, click the icon to the right of the Transform Using field.  
The Request Transformation Map dialog is displayed.
3. Click the Add (+) icon and ensure that the selected Type is **XSLT**, then click **OK**.

4. Click **OK**.

The mapping page is displayed, as shown in [Figure 5–16](#).

**Figure 5–16 Mapping Page**



5. Click **OK**.
6. Map the `ns0:LOCATION_SYNC` source element to the `ns0:LOCATION_SYNC` target element.  
The Auto Map Preferences dialog is displayed.
7. Retain the default values and click **OK**.  
The mapping is now complete.
8. Click the **Save All** icon in the menu bar to save the new inbound Mediator process component that was configured.

#### 5.3.2.4 Adjusting for Known Deployment Issues With 12c

For more information on how to adjust for known deployment issues with 12c, see [Section 4.4.3.3, "Adjusting for Known Deployment Issues With 12c"](#) on page 4-26.

You are now ready to deploy the Mediator inbound process. You can follow the same procedure in [Section 4.5.4, "Deploying the BPEL Inbound Process,"](#) on page 4-44.

Once event messages are triggered through PeopleSoft, output XML is received in the location that was specified for the File adapter component. For more information on triggering events in PeopleSoft, see [Section 4.5.5, "Triggering an Event in PeopleSoft"](#) on page 4-45.

## 5.4 Configuring a Mediator Outbound Process (BSE Configuration)

This section describes how to configure a Mediator outbound process to your PeopleSoft system, using a Mediator project in Oracle JDeveloper.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\Mediator\BSE\Outbound_Project
```

This section contains the following topics:

- [Section 5.4.1, "Creating an Empty Composite for SOA"](#)
- [Section 5.4.2, "Defining a Mediator Outbound Process"](#)

### Prerequisites

Before you design a Mediator outbound process, you must generate the respective WSDL file using Application Explorer. For more information, see [Section 4.6.1, "Generating a WSDL File for Request and Response Services Using a Web Service"](#) on page 4-53.

## 5.4.1 Creating an Empty Composite for SOA

Perform the following steps to create an empty composite for SOA:

1. Create a new SOA application.
2. Enter a name for the SOA application, and click **Next**.
3. Enter a project name (for example, IBSE\_Outbound), and click **Next**.
4. From the Composite Template list, select **Empty Composite** and click **Finish**.

For more information, see [Section 4.4.2, "Creating an Empty Composite for SOA"](#) on page 4-38.

## 5.4.2 Defining a Mediator Outbound Process

This section describes how to define a Mediator outbound process. The following topics are included:

- [Section 5.4.2.1, "Configuring a SOAP Service"](#)
- [Section 5.4.2.2, "Creating a Mediator Component"](#)
- [Section 5.4.2.3, "Configuring the Routing Rules"](#)

### 5.4.2.1 Configuring a SOAP Service

Perform the following steps to configure a SOAP Service:

1. Drag and drop the **SOAP** node from the Technology Adapters pane to the External References pane.
2. Enter an appropriate name for the SOAP Service and click on the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.
3. In the displayed SOA Resource Browser window, select the File system tab and navigate to the location where the WSDL is exported from the Application Explorer, select the WSDL, and click **OK**.
4. In the Create Web Service Window, click **OK**.
5. In the displayed Localize Files window, click **OK**. This imports the WSDL file to the project folder.

The Web Service is created and displayed.

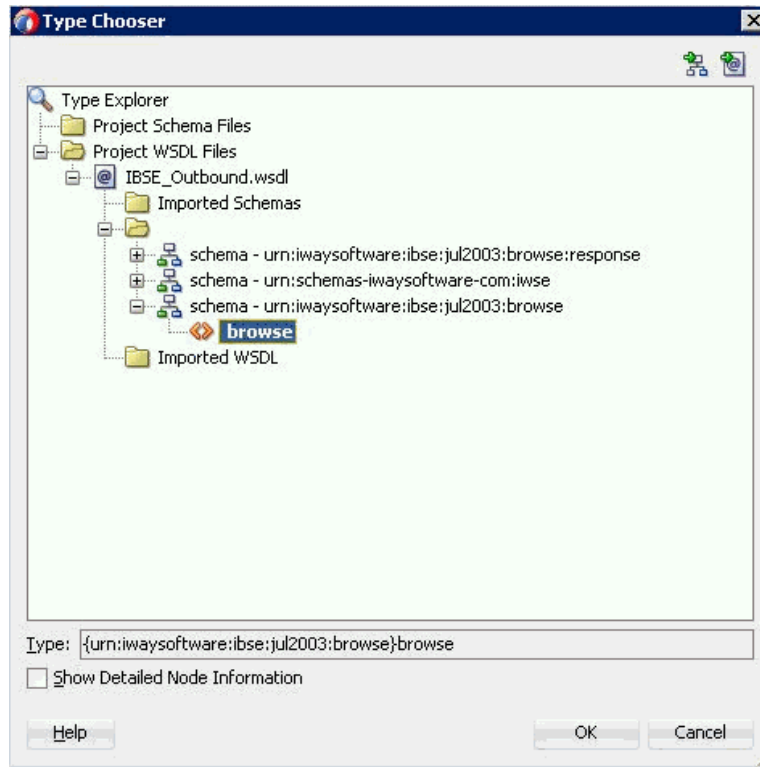
### 5.4.2.2 Creating a Mediator Component

Perform the following steps to create a Mediator component:

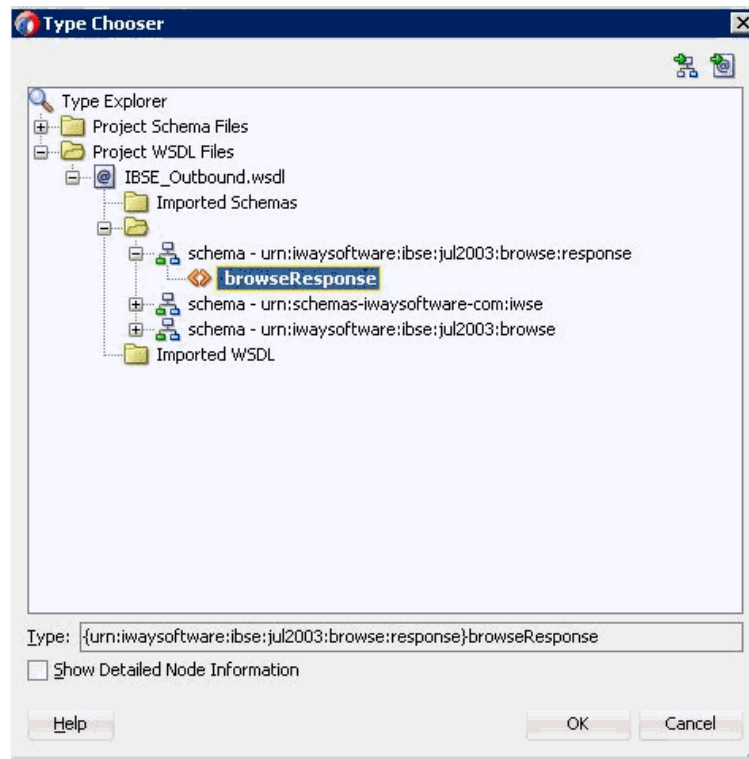
1. Drag and drop the **Mediator** component from the Components pane in to the Components pane.
2. In the Name field, enter a name to identify the new outbound Mediator process component.
3. From the Template drop-down list, select **Synchronous Interface**.

4. Click the **Browse** icon, which is located to the right of the Input field, to select the associated XML request schema file.
5. In the Type Chooser dialog, expand **Project WSDL Files**, select **IBSE\_Outbound.wsdl**, and click **browse**, as shown in [Figure 5–17](#).

**Figure 5–17** Type Chooser Dialog



6. Click **OK**.
7. Click the **Browse** icon, which is located to the right of the Output field, to select the associated XML response schema file.
8. In The Type Chooser dialog, expand **Project WSDL Files**, select **IBSE\_Outbound.wsdl**, and click **browseResponse**, as shown in [Figure 5–18](#).

**Figure 5–18** Type Chooser Dialog

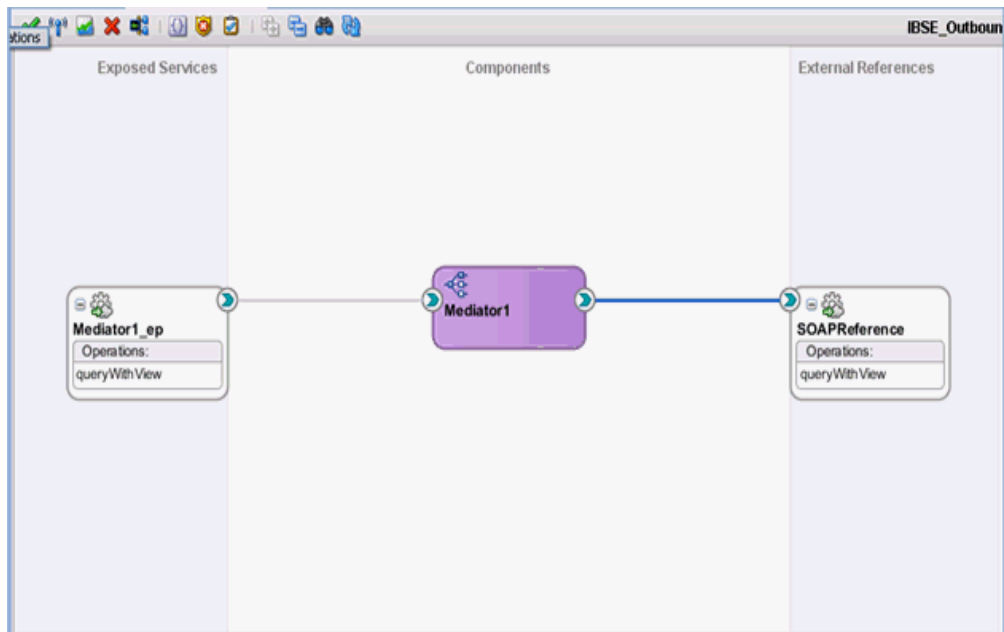
9. Click **OK**.

10. Click **OK**.

The Mediator component is created and displayed.

11. Create a connection between the **Mediator** component and the **SOAP service** component, as shown in [Figure 5–19](#).

**Figure 5–19 Created Connection**

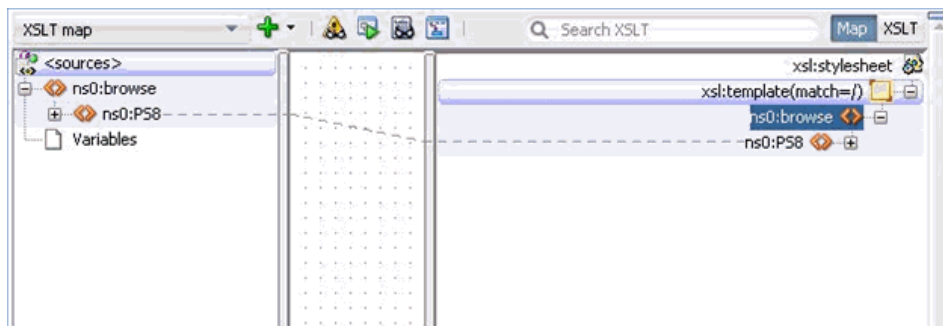


### 5.4.2.3 Configuring the Routing Rules

Perform the following steps to configure the routing rules:

1. Double-click the **Mediator** component in the Components pane.
2. In the <<Filter Expression>> area of the Static Routing section, click the icon to the right of the Transform Using field.
3. In the displayed Request Transformation Map window, click the Add (+) icon and make sure the selected Type is **XSLT** in the Create Transformation Map dialog box and click **OK**.
4. Click **OK**.
5. Map the **ns0:browse** source element to the **ns0:browse** target element, as shown in [Figure 5–20](#).

**Figure 5–20 GetDetail\_To\_GetDetail.xsl Tab**

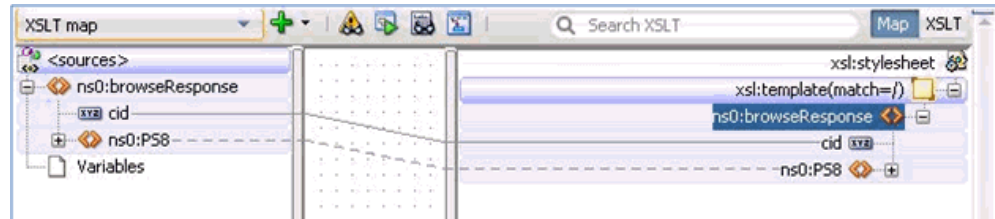


6. In the displayed Auto Map Preferences window, retain the default values and click **OK**.
7. In the Synchronous Reply area, click the icon to the right of the Transform Using field.



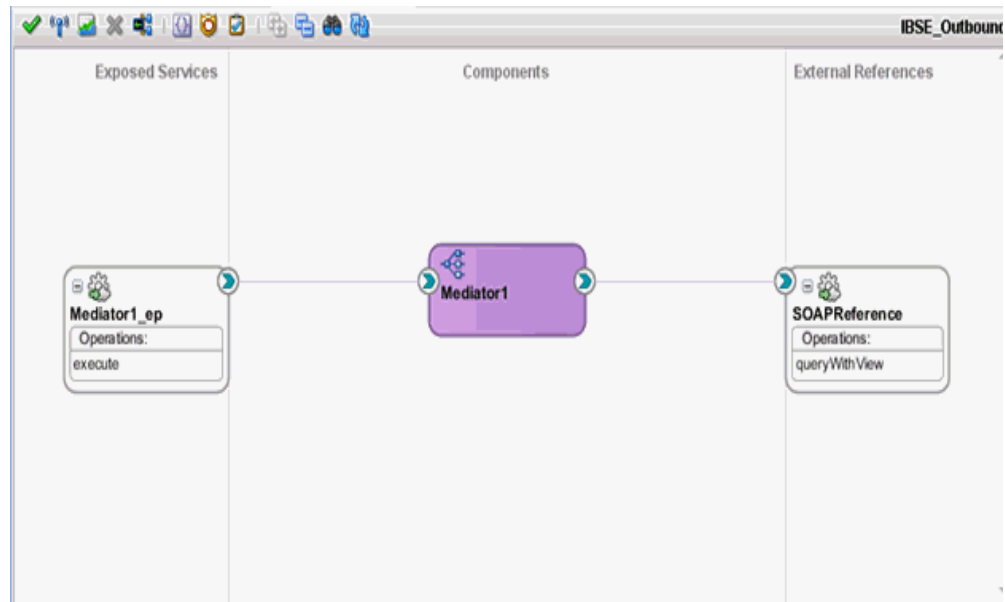
8. In the displayed Reply Transformation Map window, click the Add (+) icon and make sure the Type is selected as **XSLT** in the Create Transformation Map dialog box, and then click **OK**.
9. Map the **ns0:browseResponse** source element to the **ns0:browseResponse** target element, as shown in [Figure 5–21](#).

**Figure 5–21 Source and Target Elements**



10. In the displayed Auto Map Preferences window, retain the default values and click **OK**.
11. Double-click **composite.xml** in the left pane.
12. Click the **Save All** icon in the menu bar to save the new outbound Mediator component that was configured, as shown in [Figure 5–22](#).

**Figure 5–22 Save All Icon**



You are now ready to deploy the Mediator IBSE outbound process. You can follow the same procedure found in [Section 5.2.3, "Deploying the Mediator Outbound Process"](#) on page 5-11.

Once deployed, you can invoke the input XML, as defined in [Section 4.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console"](#) on page 4-31.



---

---

## Integration With BPM Service Components in the Oracle SOA Suite

Oracle Application Adapter for PeopleSoft integrates seamlessly with Oracle Business Process Management (BPM) to facilitate Web service integration. Oracle BPM is based on the Service-Oriented Architecture (SOA). It consumes adapter services exposed as Web Service Definition Language (WSDL) documents.

This chapter contains the following topics:

- [Section 6.1, "Overview"](#)
- [Section 6.2, "Deployment of Adapter"](#)
- [Section 6.3, "Configuring a New Application Server Connection"](#)
- [Section 6.4, "Designing an Outbound BPM Process Using Transformations for Service Integration \(J2CA Configuration\)"](#)
- [Section 6.5, "Designing an Inbound BPM Process Using Transformations for Event Integration \(J2CA Configuration\)"](#)
- [Section 6.6, "Designing an Outbound BPM Process Using Transformations for Service Integration \(BSE Configuration\)"](#)

### 6.1 Overview

To integrate with Oracle BPM, Oracle Application Adapter for PeopleSoft must be deployed in the same WLS container as Oracle BPM. The underlying adapter services must be exposed as WSDL files, which are generated during design time in Oracle Adapter Application Explorer (Application Explorer) for both request-response (outbound) and event notification (inbound) services of the adapter. For more information, see ["Generating a WSDL \(J2CA Configurations Only\)"](#) on page 2-10.

The generated WSDL files are used to design the appropriate BPM processes for inbound or outbound adapter services. A completed BPM process must be successfully compiled in JDeveloper and deployed to a BPM server. Upon deployment to the BPM server, every newly built process is automatically deployed to the Oracle Enterprise Manager console, where you run, monitor, and administer BPM processes, and listen to adapter events.

### 6.2 Deployment of Adapter

During installation, Oracle Application Adapter for PeopleSoft is deployed as a J2CA 1.0 resource adapter within the WLS container. The adapter must be deployed in the same WLS container as Oracle BPM.

## 6.3 Configuring a New Application Server Connection

For more information on how to configure a new Application Server connection in Oracle JDeveloper, see [Section 4.3, "Configuring a New Application Server Connection"](#) on page 4-2.

## 6.4 Designing an Outbound BPM Process Using Transformations for Service Integration (J2CA Configuration)

This section describes how to design an outbound BPM process using transformations for service integration.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\BPM\J2CA\Outbound_Project
```

The following tools are required to complete your outbound design-time configuration:

- Oracle Adapter Application Explorer (Application Explorer)
- Oracle JDeveloper BPM Designer (JDeveloper)

---

---

**Note:** The examples in this chapter demonstrate the use of JDeveloper.

---

---

This section contains the following topics:

- [Section 6.4.1, "Creating an Empty Composite for BPM"](#)
- [Section 6.4.2, "Defining a BPM Outbound Process"](#)
- [Section 6.4.3, "Adjusting for Known Deployment Issues With 12c"](#)
- [Section 6.4.4, "Deploying the BPM Outbound Process"](#)
- [Section 6.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console"](#)

Before you design a BPM process, you must generate the respective WSDL file using Application Explorer. For more information, see [Section 4.4.1, "Generating WSDL for Request/Response Service"](#) on page 4-8.

### 6.4.1 Creating an Empty Composite for BPM

Perform the following steps to create an empty composite for BPM:

1. Create a new BPM application.
2. Enter a name for the new BPM application and click **Next**.  
The Name your project page is displayed.
3. Enter a project name, in the project features select **BPM**, and then click **Next**.  
The Configure SOA settings page is displayed.
4. From the Composite Template list, select **Empty Composite** and click **Finish**.

## 6.4.2 Defining a BPM Outbound Process

This section describes how to define a BPM outbound process, which contains the following topics:

- [Section 6.4.2.1, "Configuring a Third Party Adapter Service Component"](#)
- [Section 6.4.2.2, "Configuring an Outbound BPM Process Component"](#)
- [Section 6.4.2.3, "Creating a File Adapter for the Write Operation"](#)

### 6.4.2.1 Configuring a Third Party Adapter Service Component

Perform the following steps to create a third party adapter service component:

1. Double-click the created project to load the components.
2. Drag and drop the **Third Party Adapter** component from the Custom/Thirdparty pane to the External References pane.

The Create Third Party Adapter Service dialog is displayed.

3. Enter a name for the third party adapter service.
4. Ensure that **Reference** is selected from the Type list (default).
5. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.

The WSDL Chooser dialog is displayed.

6. Browse and select an outbound WSDL file from the following directory:

```
<ADAPTER_HOME>\wsdl.s
```

7. Click **OK**.

The Localize Files dialog is displayed.

8. Click **OK**.

The outbound WSDL file and associated request and response XML schema files (.xsd) are imported to the project folder that has been created.

You are returned to the Create Third Party Adapter Service dialog.

9. Click the **Find JCA file** icon, which is located to the right of the JCA File field.

The Transformation Map dialog is displayed.

10. Browse and select the JCA properties file from the following directory:

```
<ADAPTER_HOME>\wsdl.s
```

11. Click **OK**.

The Copy File message is displayed.

12. Click **Yes**.

A copy of the JCA properties file is made in the project folder.

You are returned to the Create Third Party Adapter Service dialog.

13. Click **OK**.

The third party adapter service component is created and displayed in the External References pane.

You are now ready to configure an outbound BPM process component.

For more detailed information, including screen shots, see [Section 4.4.3.1, "Configuring a Third Party Adapter Service Component"](#) on page 4-11.

### 6.4.2.2 Configuring an Outbound BPM Process Component

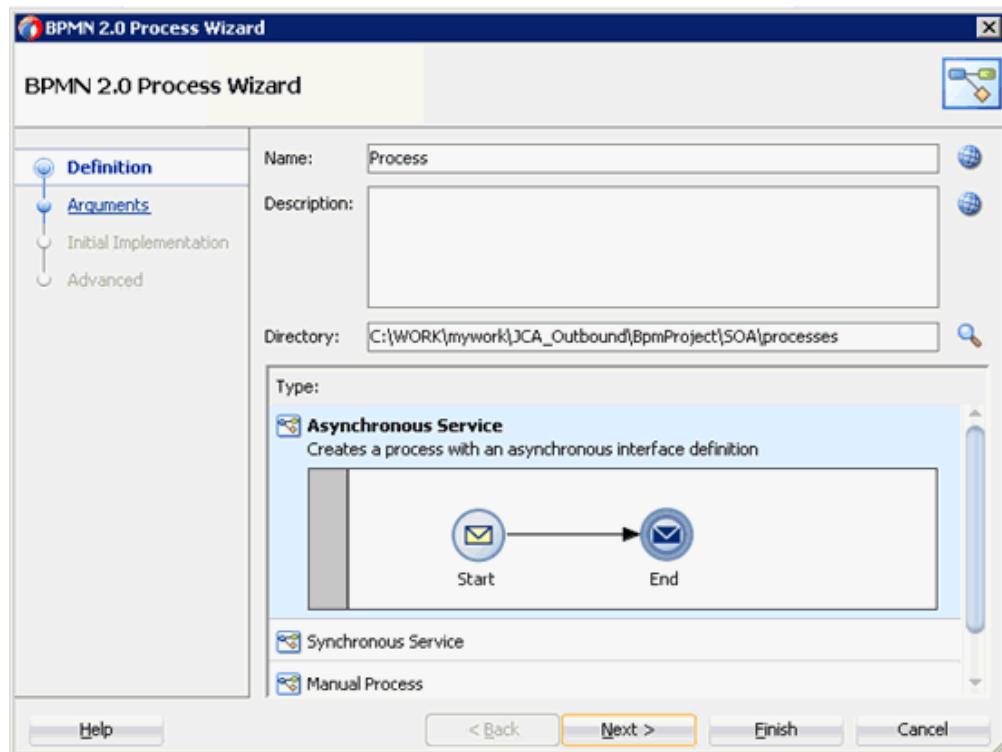
This section describes how to configure an outbound BPM process component.

Perform the following steps to configure an outbound BPM process component:

1. Drag and drop the **BPMN Process** component from the Components pane to the Components pane.

The Create BPMN Process dialog is displayed, as shown in [Figure 6-1](#).

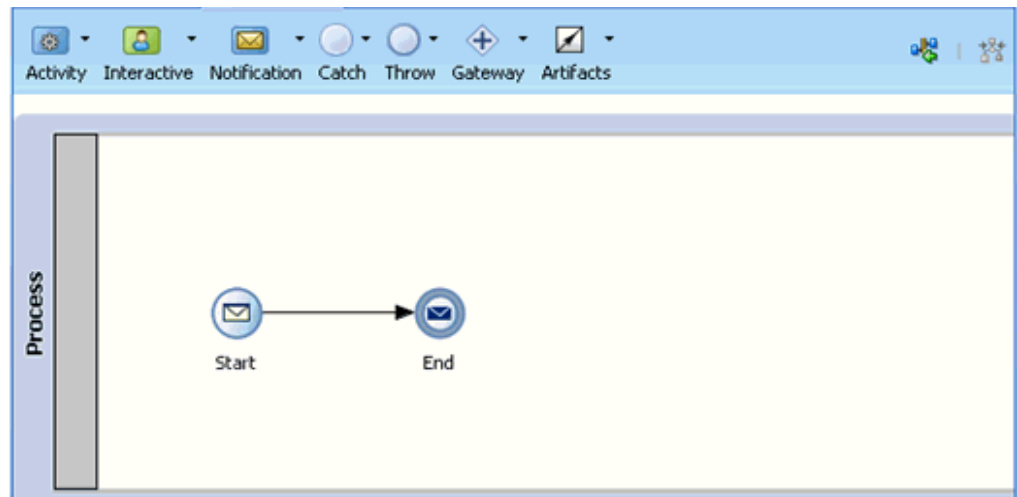
**Figure 6-1** Create BPMN Process Dialog



2. Accept the default option that is selected under the Type area (Asynchronous Service) and click **Finish**.

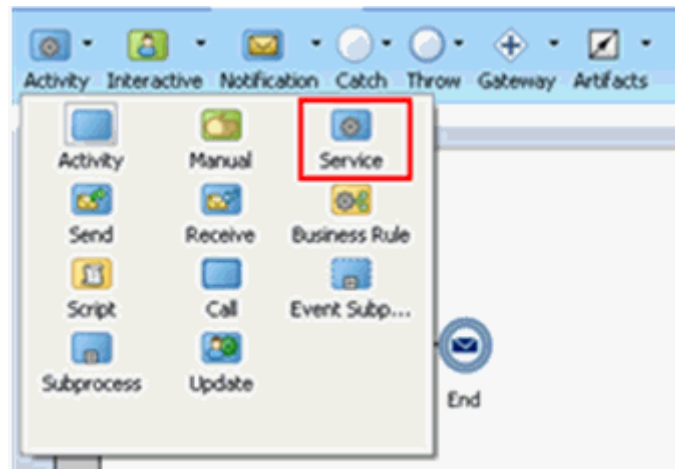
The BPMN process is displayed, as shown in [Figure 6-2](#).

**Figure 6–2 BPMN Process**



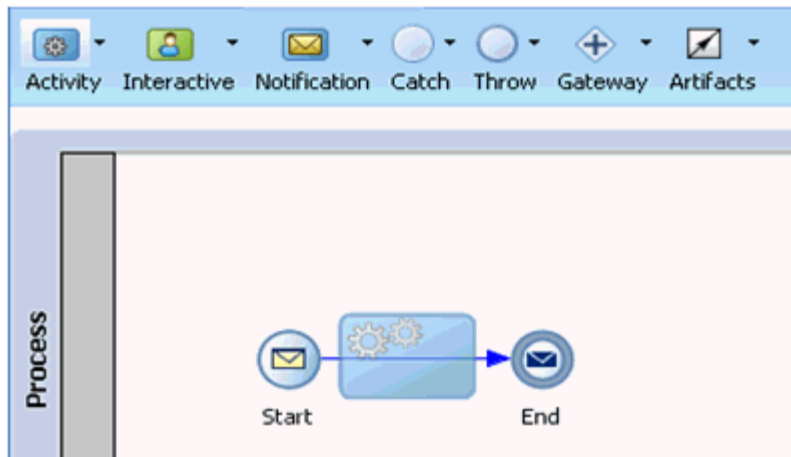
3. Click the **Activity** drop-down menu and select **Service**, as shown in [Figure 6–3](#).

**Figure 6–3 Activity Drop-down Menu**



4. Drop the **Service** icon on the wire between the Start and End event components, as shown in [Figure 6–4](#).

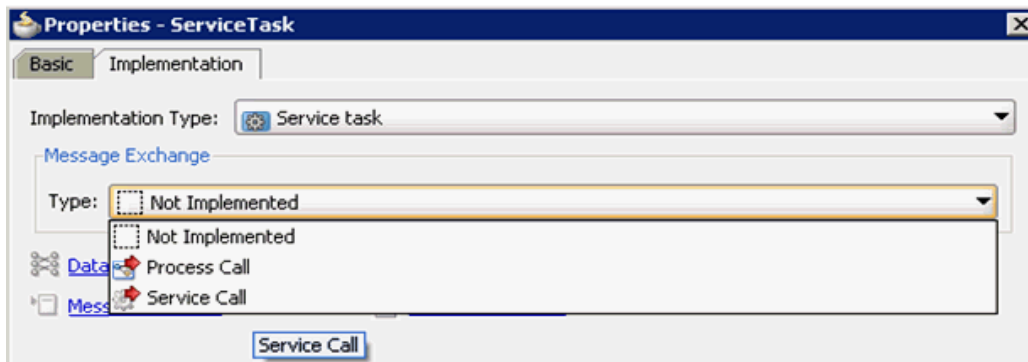
**Figure 6–4 Activity Icon**



The Properties - ServiceTask window is displayed.

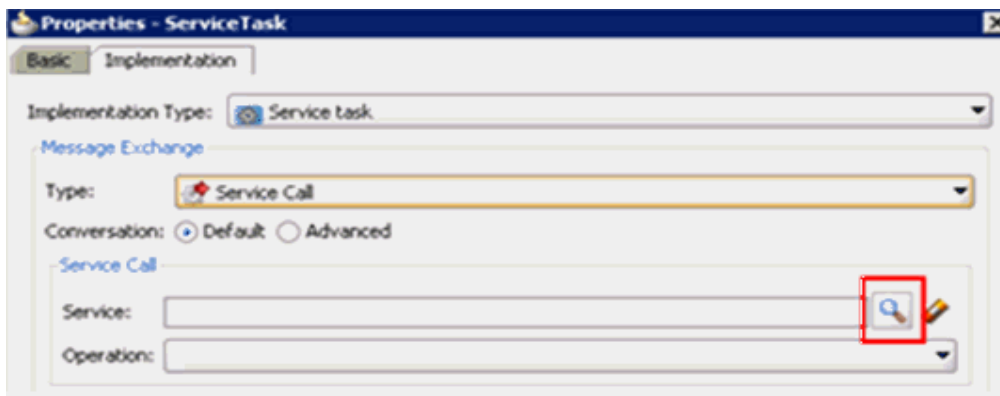
5. Click the **Implementation** tab.
6. Select **Service Call** from the Message Exchange Type list, as shown in [Figure 6–5](#).

**Figure 6–5 Service Call**



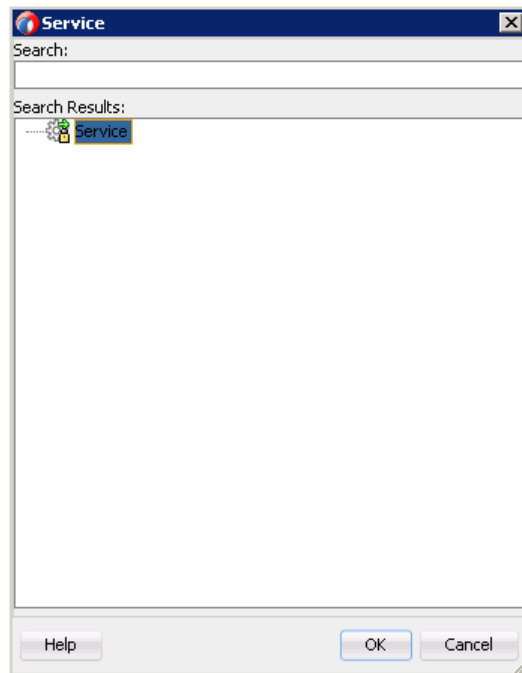
7. Click the Browse icon to the right of the Service field, as shown in [Figure 6–6](#).

**Figure 6–6 Browse Icon**



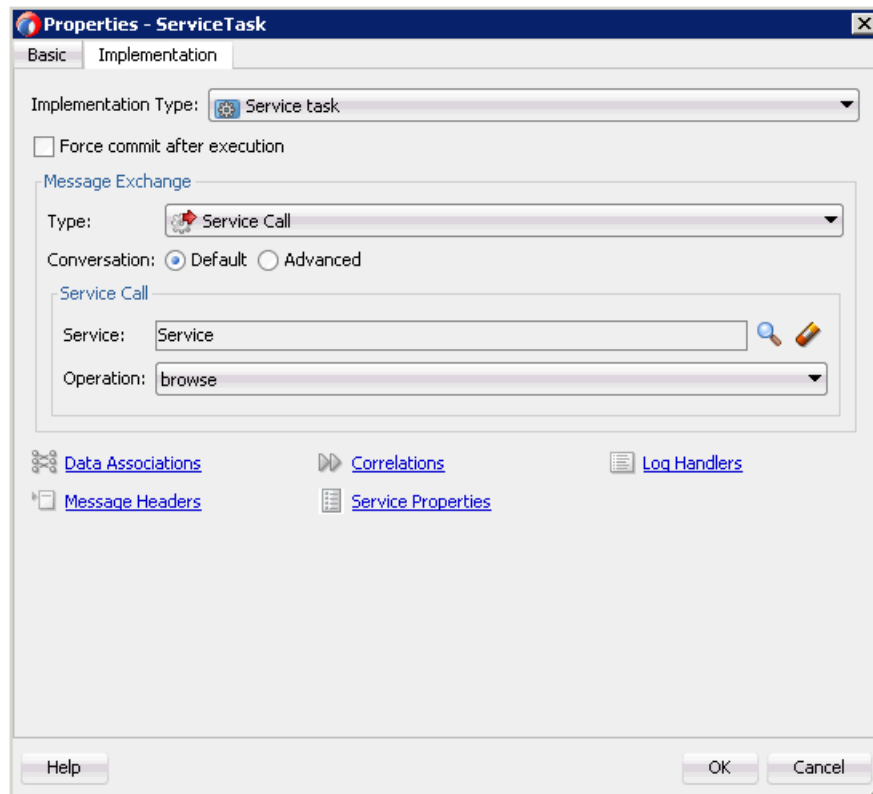
The Service dialog is displayed, as shown in [Figure 6–7](#).



**Figure 6–7 Service Dialog**

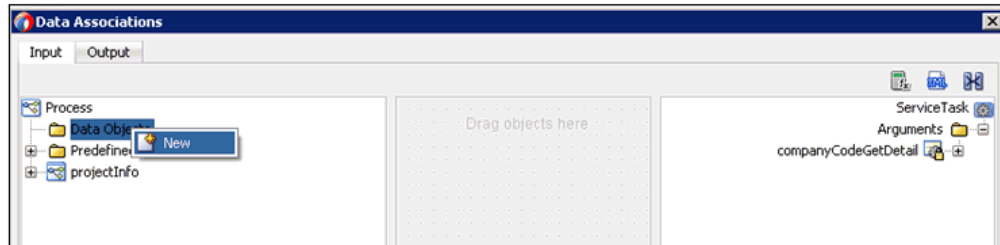
8. Select the Third Party Service that has been created and click **OK**.

You are returned to the Properties - ServiceTask dialog, as shown in [Figure 6–8](#).

**Figure 6–8 Properties - ServiceTask Dialog**

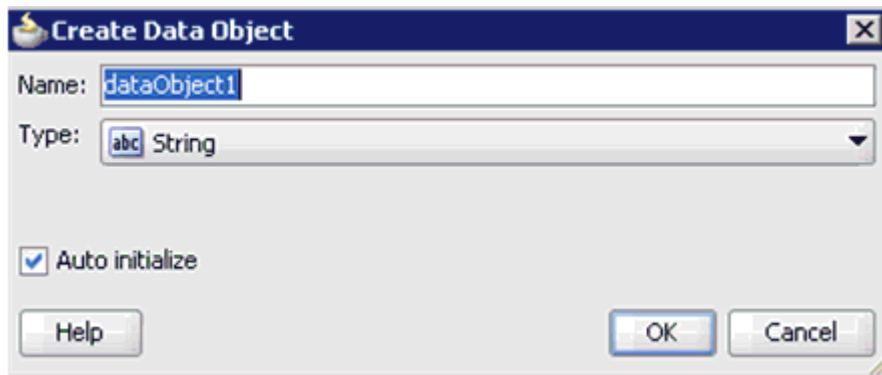
9. Click the **Data Associations** hyperlink.  
The Data Associations dialog is displayed.
10. Right-click the **Data Objects** node in the left pane under Process, and select **New** as shown in [Figure 6-9](#).

**Figure 6-9 New Option**



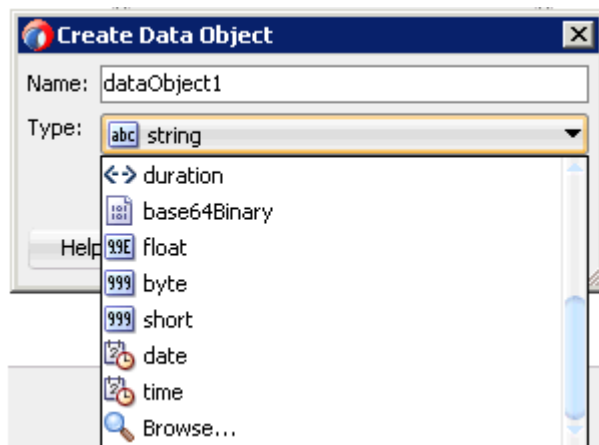
The Create Data Object dialog is displayed, as shown in [Figure 6-10](#).

**Figure 6-10 Create Data Object Dialog**

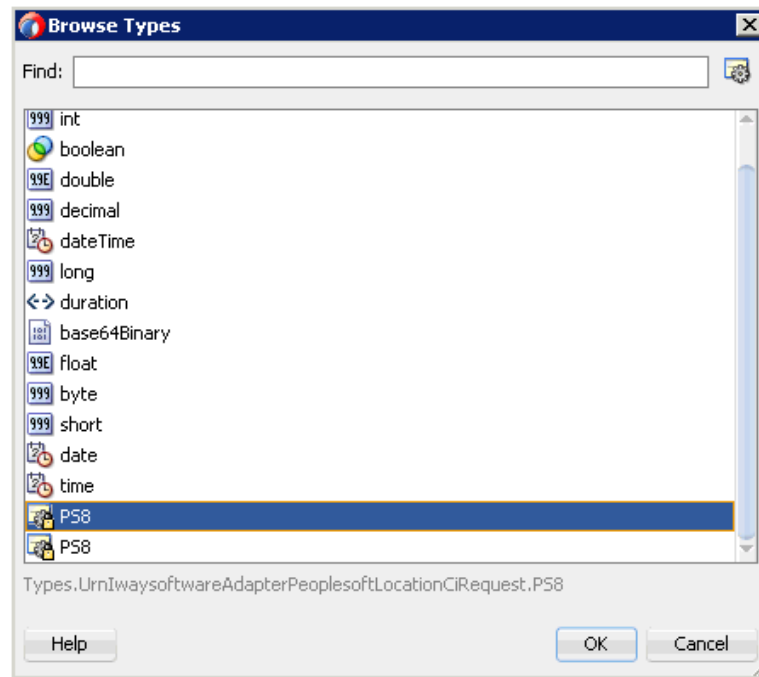


11. Enter a name in the Name field (for example, Request), click the drop-down button in the Type field, and select **Browse** from the list, as shown in [Figure 6-11](#).

**Figure 6-11 Create Data Object Dialog**



The Browse Types dialog is displayed, as shown in [Figure 6-12](#).

**Figure 6–12 Browse Types Dialog**

12. Select the first component (for example, PS8) and click **OK**.

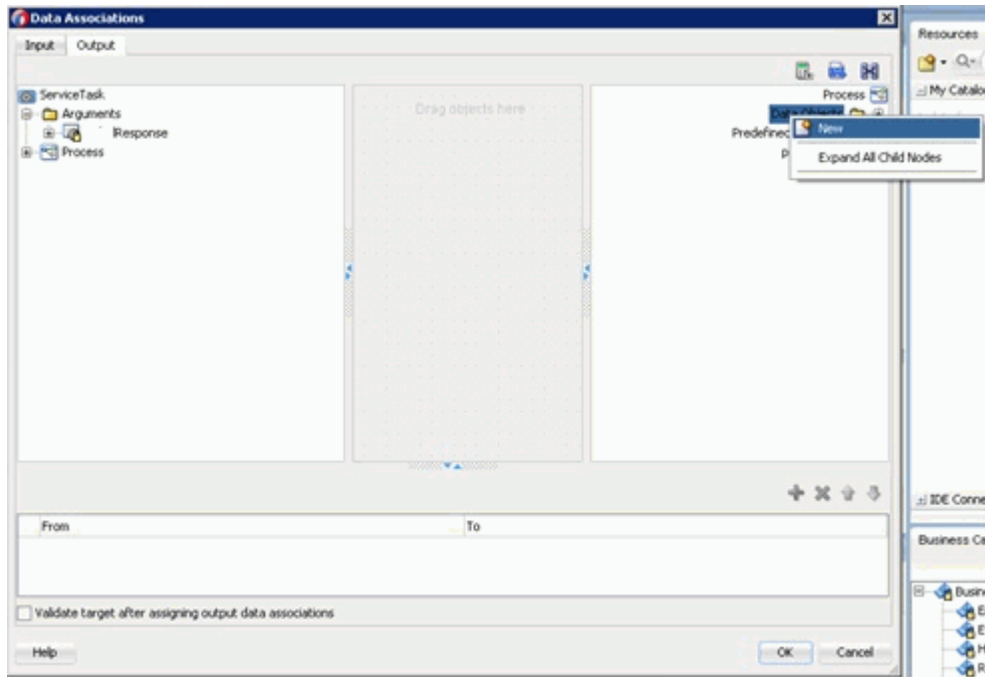
You are returned to the Create Data Object dialog.

13. Click **OK**.

The Data Object (for example, Request) that has been created is displayed under the Data Objects node in the Data Associations dialog.

14. Create another Data Object by right-clicking the **Data Objects** node in the right pane of the Output tab and selecting **New**, as shown in [Figure 6–13](#).

**Figure 6–13 Data Associations Dialog**

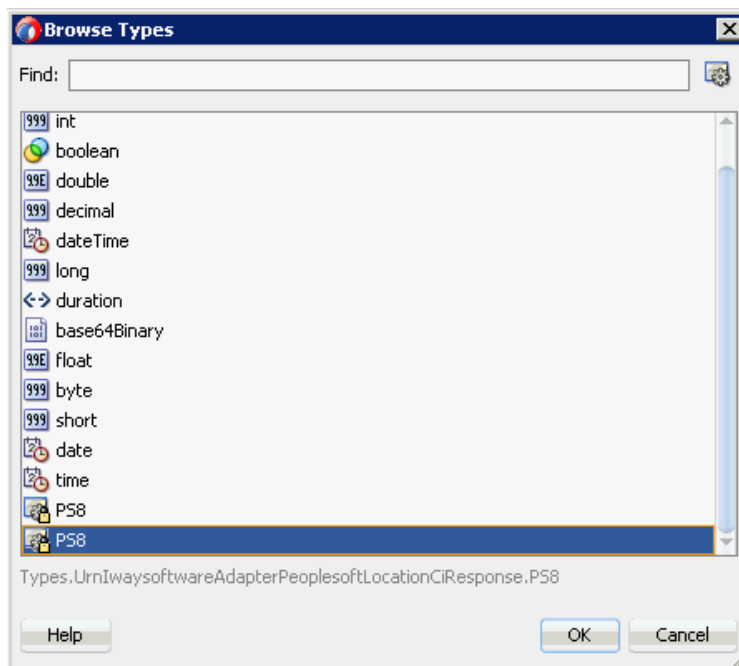


The Create Data Object dialog is displayed.

15. Enter a name in the Name field (for example, Response), and then click the drop-down button in the Type field and select **Browse** from the list.

The Browse Types dialog is displayed, as shown in [Figure 6–14](#).

**Figure 6–14 Browse Types Dialog**



16. Select the second component (for example, PS8) and click **OK**.

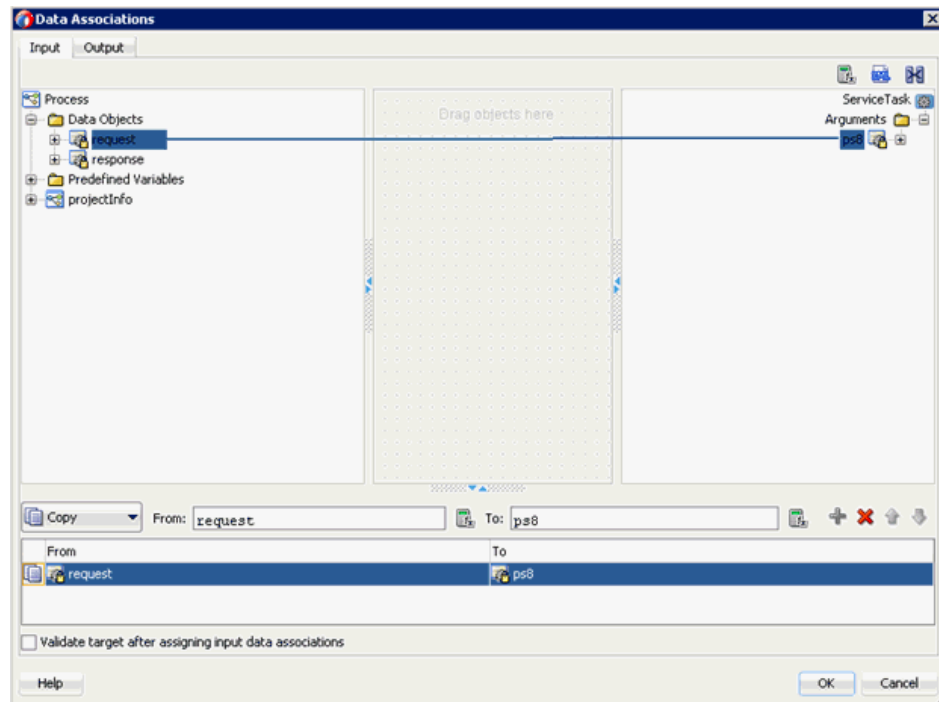
You are returned to the Create Data Object dialog.

**17. Click OK.**

The Data Object (for example, Response) that has been created is displayed under the Process node in the Data Associations dialog.

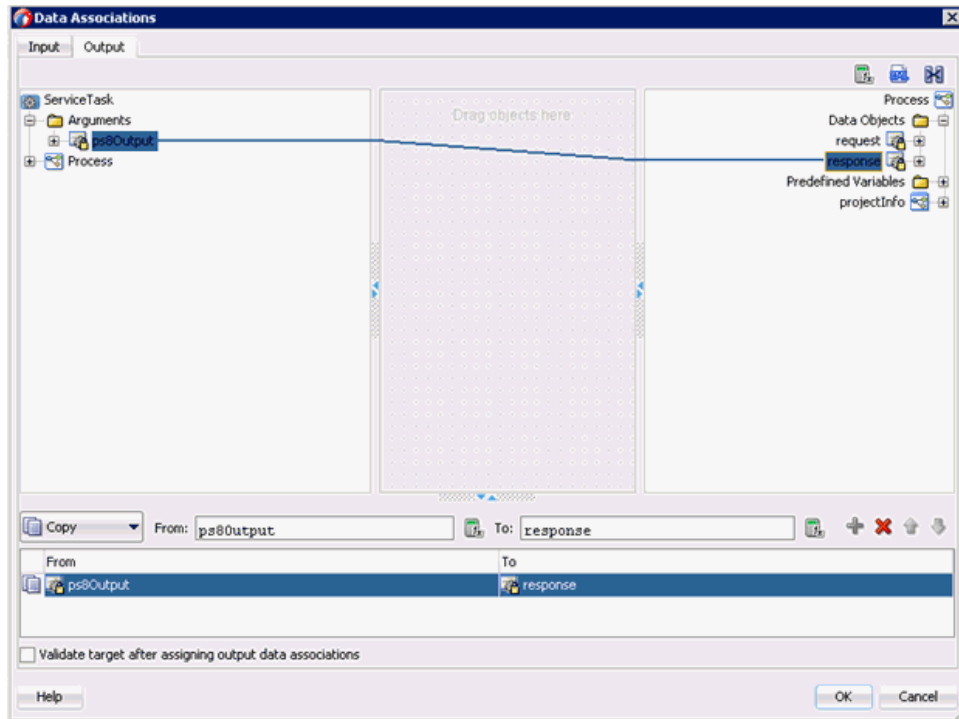
**18. Select the **Request** Data Object under the Data Objects node in the left pane of the Input tab and drag and connect it to PS8 under the Arguments node in the right pane, as shown in [Figure 6–15](#).**

**Figure 6–15 Request Data Object**



**19. Click on the **Output** tab and select **ps8Output** under the Arguments node in the left pane and drag and connect it to the Response Data Object under the Data Objects node, as shown in [Figure 6–16](#).**

**Figure 6–16 Response Data Object**



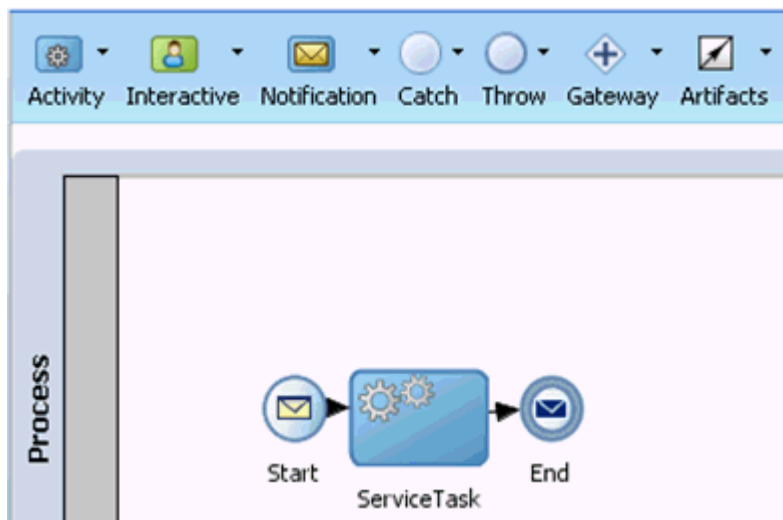
20. Click **OK**.

You are returned to the Properties - ServiceTask dialog.

21. Click **OK**.

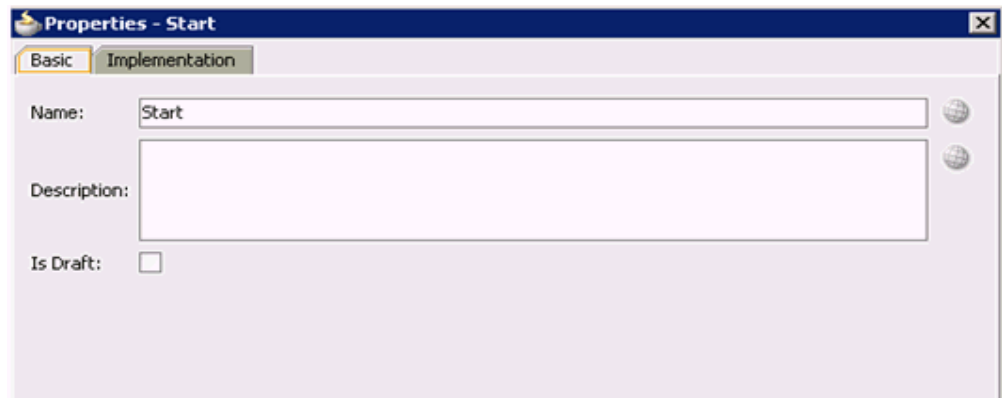
The Service Task is created between the Start and End Event components, as shown in [Figure 6–17](#).

**Figure 6–17 Service Task**

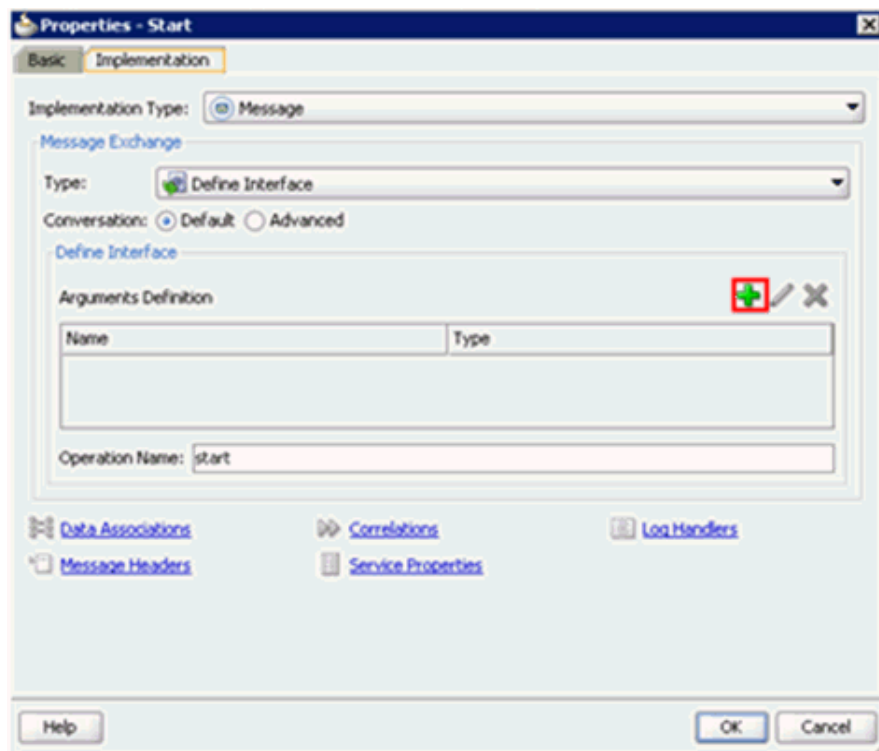


22. Save the process and double-click the Start event component.

The Properties - Start dialog is displayed, as shown in [Figure 6–18](#).

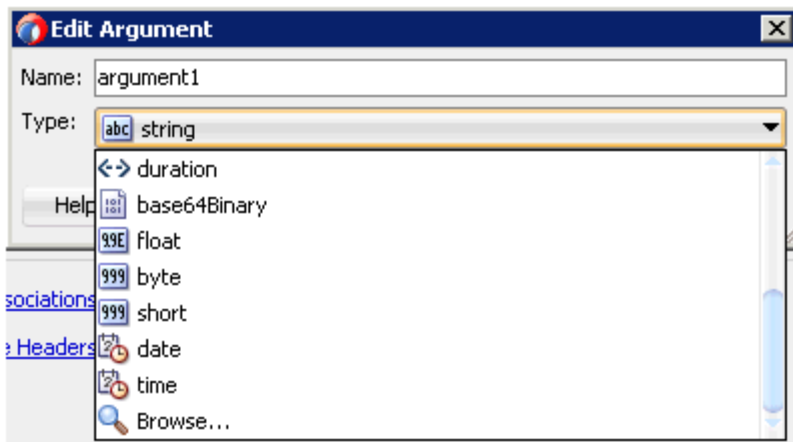
**Figure 6–18 Properties - Start Dialog**

23. Click the **Implementation** tab, as shown in [Figure 6–19](#).

**Figure 6–19 Implementation Tab**

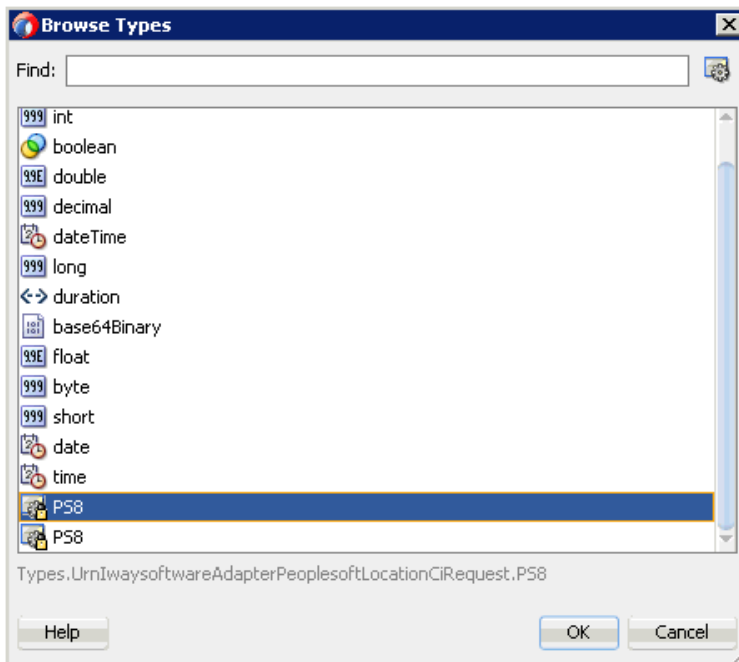
24. Click the **Plus** icon to the right of the Arguments Definition field.  
The Create Argument dialog is displayed.
25. Enter a name in the Name field (by default, argument1), and then click the drop-down button in the Type field and select **Browse** from the list, as shown in [Figure 6–20](#).

**Figure 6–20 Create Argument Dialog**



The Browse Types dialog is displayed, as shown in Figure 6–21.

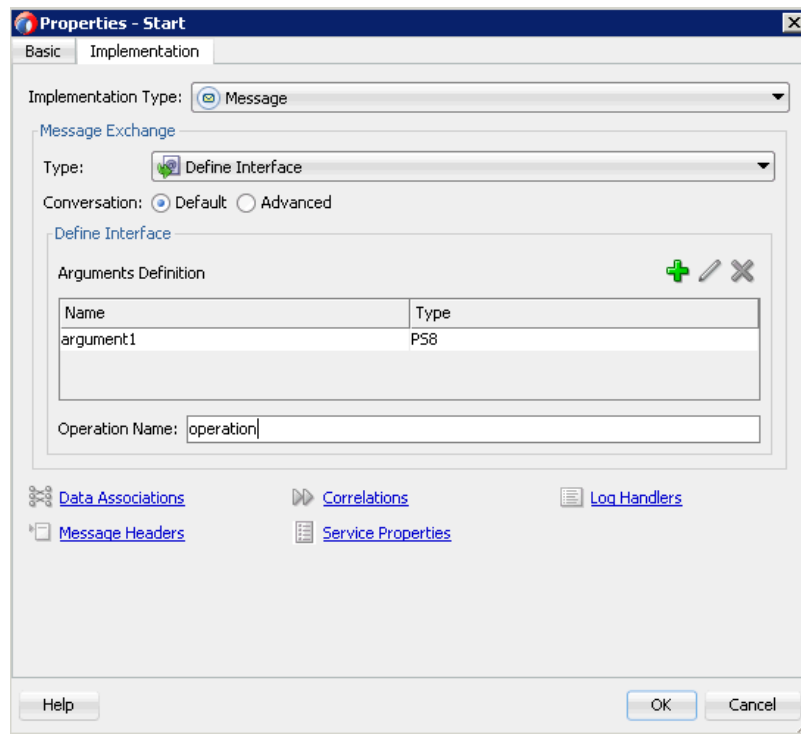
**Figure 6–21 Browse Types Dialog**



26. Select the first component (for example, PS8) and click **OK**.  
You are returned to the Create Argument dialog.
27. Click **OK**.  
You are returned to the Properties - Start dialog.
28. In the Operation Name field, change **start** (default) to **operation** as shown in Figure 6–22.

**Note:** This change is necessary to work with old BPM payloads.



**Figure 6–22 Operation Name Field**

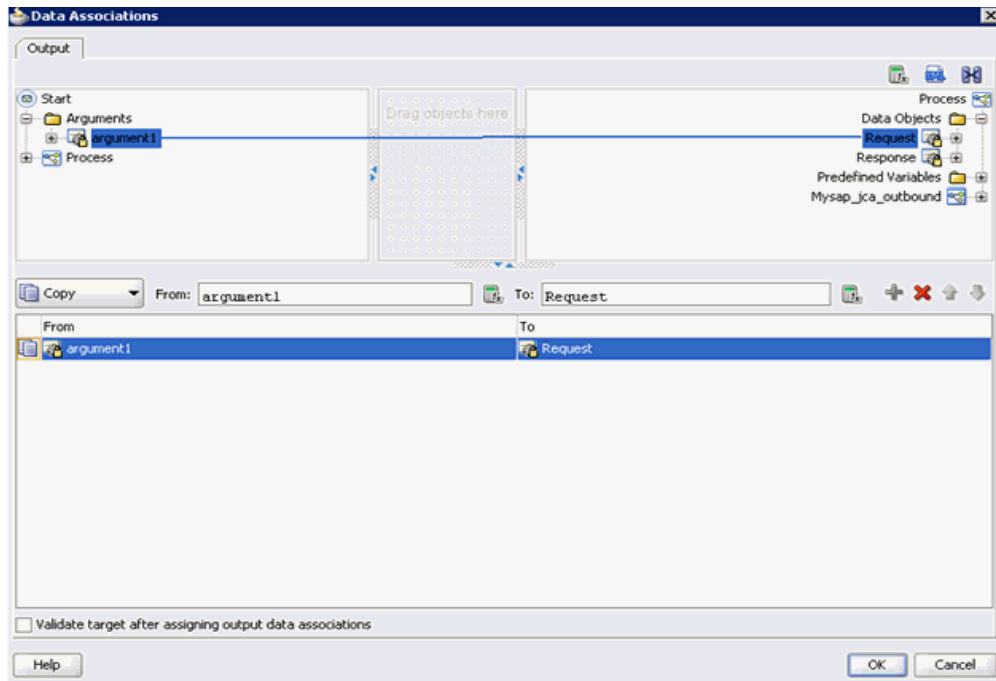
29. Click the **Data Associations** hyperlink.

The Data Associations dialog is displayed.

30. Select **arguments1** under the Arguments node in the left pane and drag and connect it to the **Request** Data Object under Data Objects in the right pane.

31. Click **OK** as shown in [Figure 6–23](#).

**Figure 6–23 OK Button**

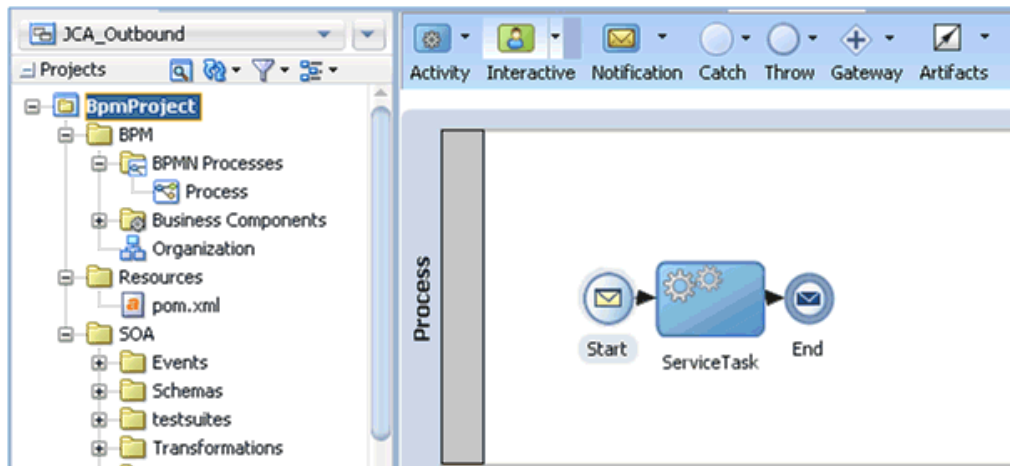


You are returned to the Properties - Start dialog.

32. Click OK.

You are returned to the Process workspace area, as shown in [Figure 6–24](#).

**Figure 6–24 Process Workspace Area**



33. Double-click the created project to load the components.
34. Click the **Save All** icon in the menu bar to save the new outbound BPM process component that was configured.

You are now ready to create a File adapter for the write operation.

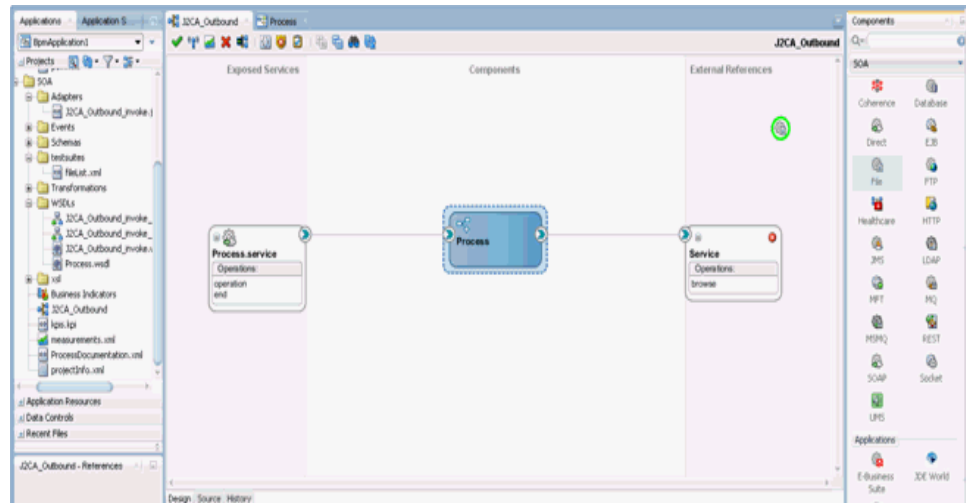
### 6.4.2.3 Creating a File Adapter for the Write Operation

This section describes how to create a File adapter for the write operation.

Perform the following steps to create a File adapter for the write operation:

1. Drag and drop the **File Adapter** component from the Technology Adapters pane to the External References pane, as shown in [Figure 6–25](#).

**Figure 6–25 File Adapter Component**



The Adapter Configuration Wizard is displayed.

2. Provide a Reference Name (for example, FileWrite).
3. Click **Next**.

The Adapter Interface page is displayed.

4. Ensure that the **Define from operation and schema (specified later)** option is selected.
5. Click **Next**.

The File Server Connection page is displayed.

6. Click **Next**.

The Operation page is displayed.

7. Select **Write File** from the list of Operation Type options and specify an Operation Name (for example, Write).
8. Click **Next**.

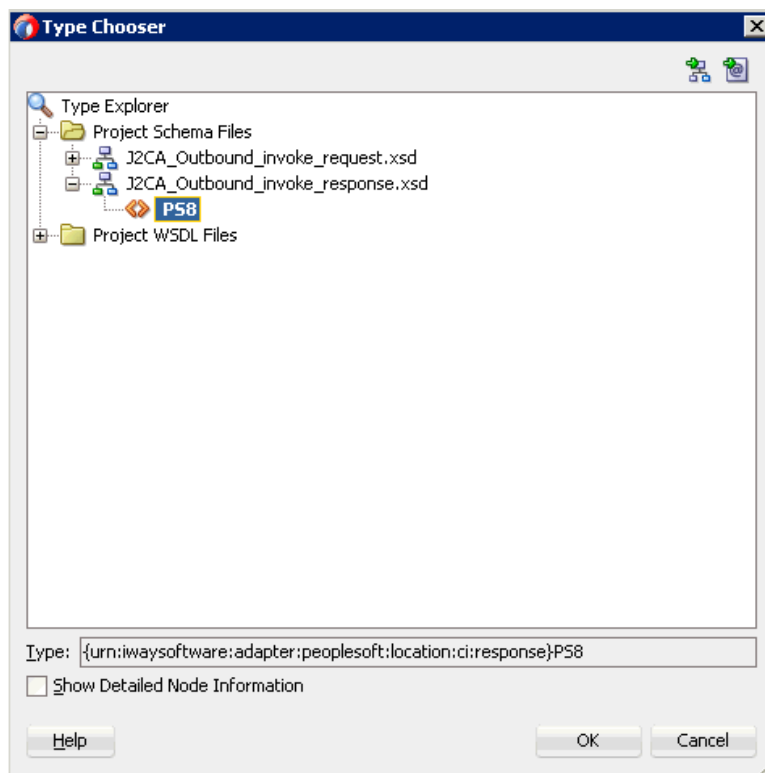
The File Configuration page is displayed.

9. Specify a location on your file system where the output file is written.
10. In the File Naming Convention field, specify a name for the output file.
11. Click **Next**.

The Messages page is displayed.

12. Click **Browse**, which is located to the right of the URL field.

The Type Chooser dialog is displayed, as shown in [Figure 6–26](#).

**Figure 6–26 Type Chooser Dialog**

13. Expand **Project Schema Files** and **J2CA\_Outbound\_invoke\_response.xsd**.

14. Select the available schema (for example, PS8).

15. Click **OK**.

You are returned to the Messages page.

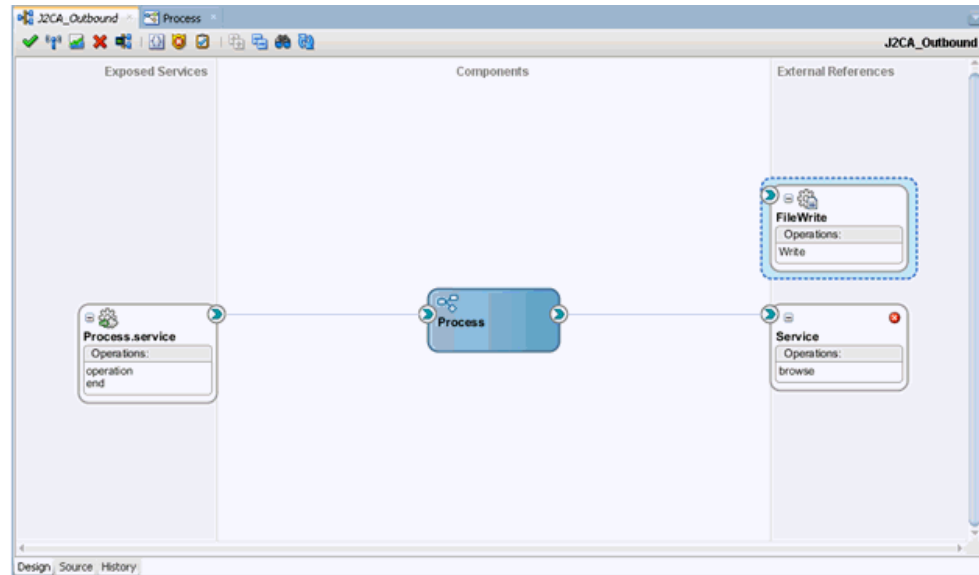
16. Click **Next**.

The Finish page is displayed.

17. Click **Finish**.

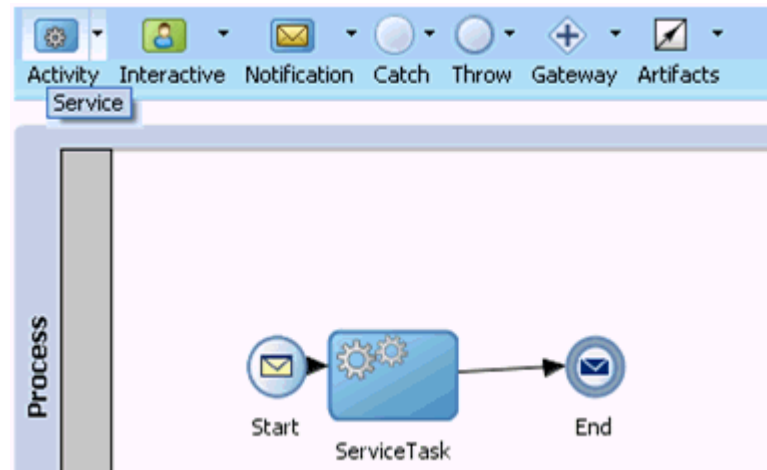
The File Adapter service is created in the External References pane, as shown in [Figure 6–27](#).

**Figure 6–27 File Adapter Service**



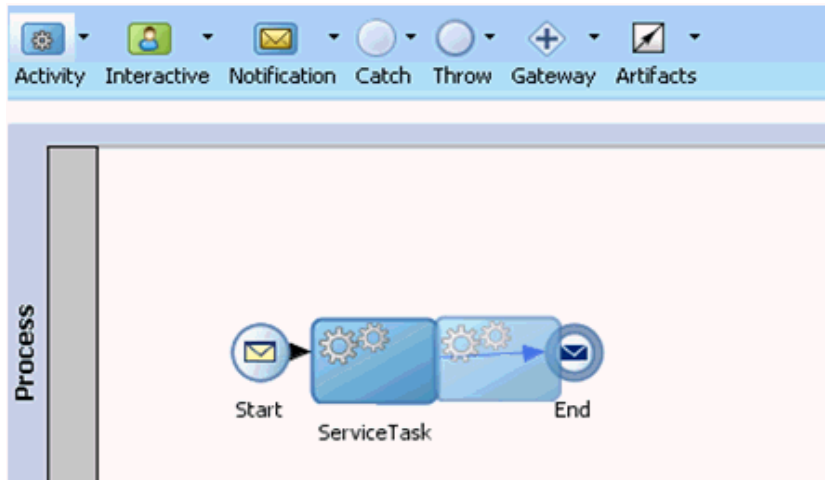
18. Double-click the BPMN Process component.  
The BPMN process is displayed, as shown in [Figure 6–28](#).

**Figure 6–28 BPMN Process**



19. Click the **Activity** icon, and select **Service**.
20. Drop the Service icon on the wire between the Service Task and End event components, as shown in [Figure 6–29](#).

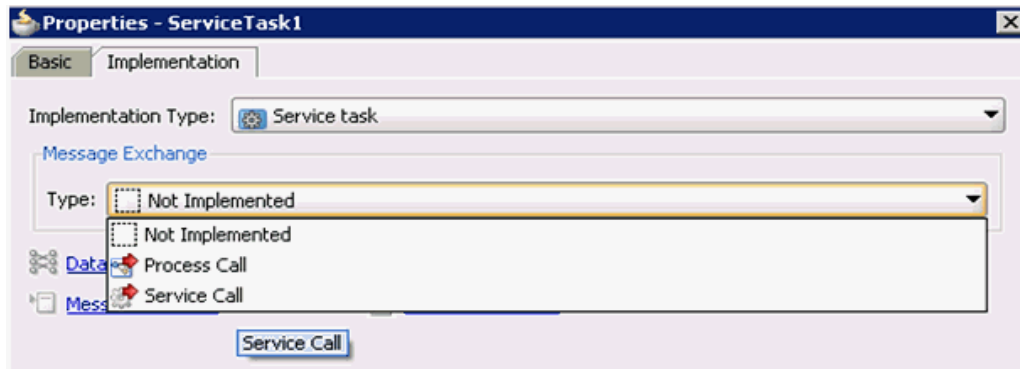
**Figure 6–29 Activity Icon**



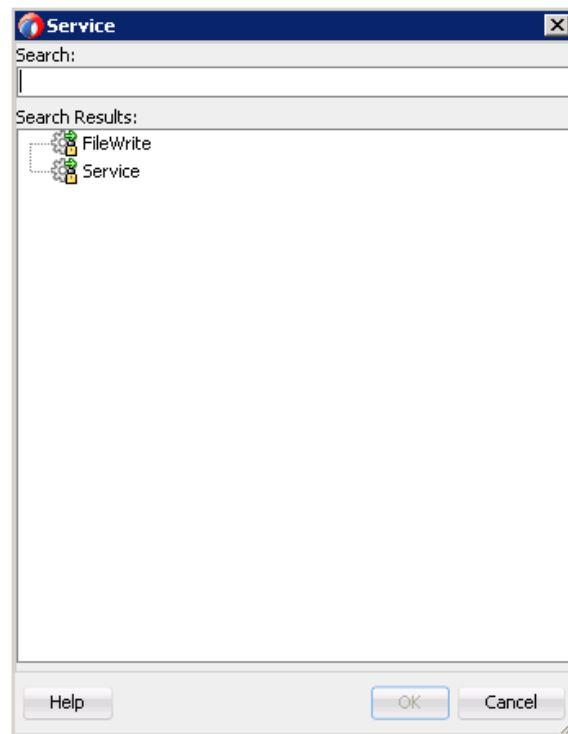
The Properties - ServiceTask1 dialog is displayed.

21. Click the **Implementation** tab.
22. Select **Service Call** from the Type drop-down list in the Message Exchange section, as shown in [Figure 6–30](#).

**Figure 6–30 Service Call**



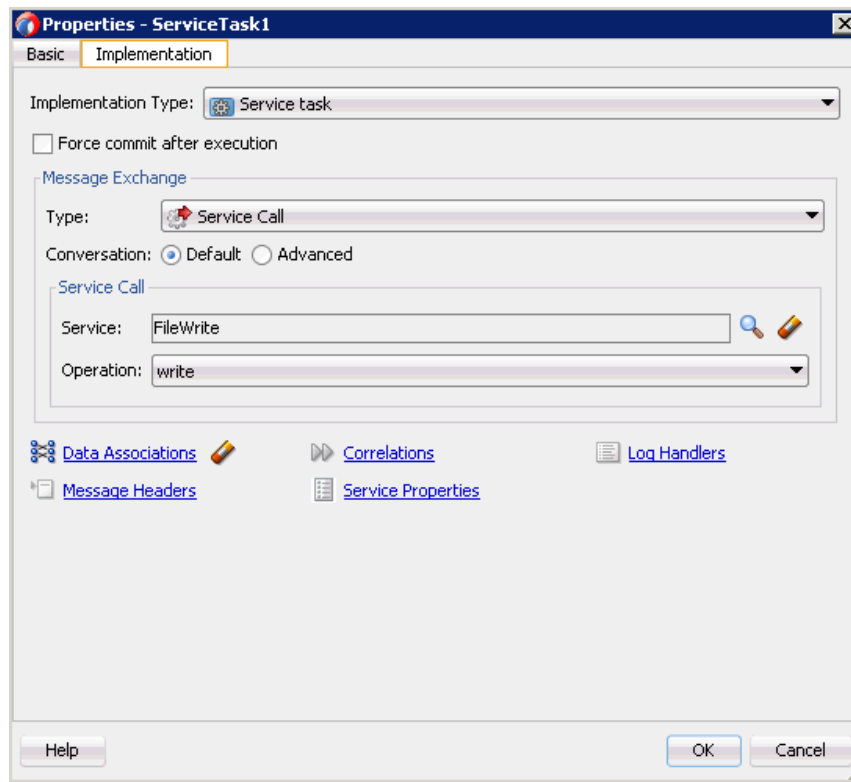
23. Click the **Browse** icon to the right of the Service field.  
The Service dialog is displayed, as shown in [Figure 6–31](#).

**Figure 6–31 Service Dialog**

24. Select the service for write operation that has been created (for example, FileWrite) and click **OK**.

You are returned to the Properties - ServiceTask1 dialog, as shown in [Figure 6–32](#).

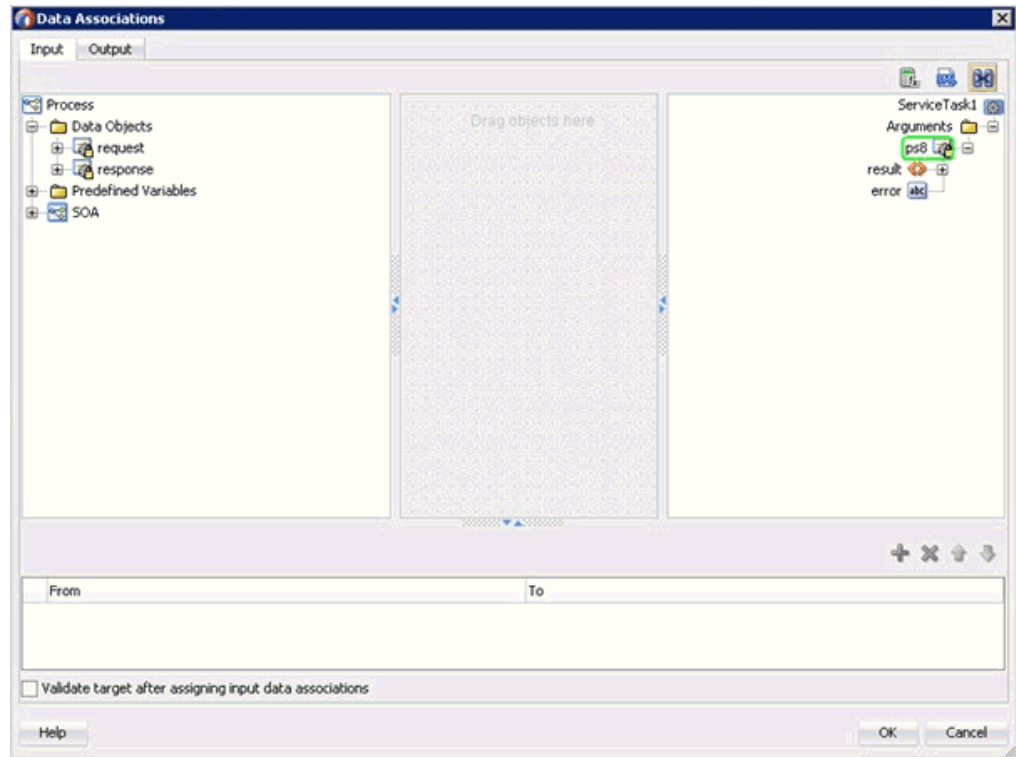
**Figure 6–32 Properties - ServiceTask1 Dialog**



25. Click the **Data Associations** hyperlink.

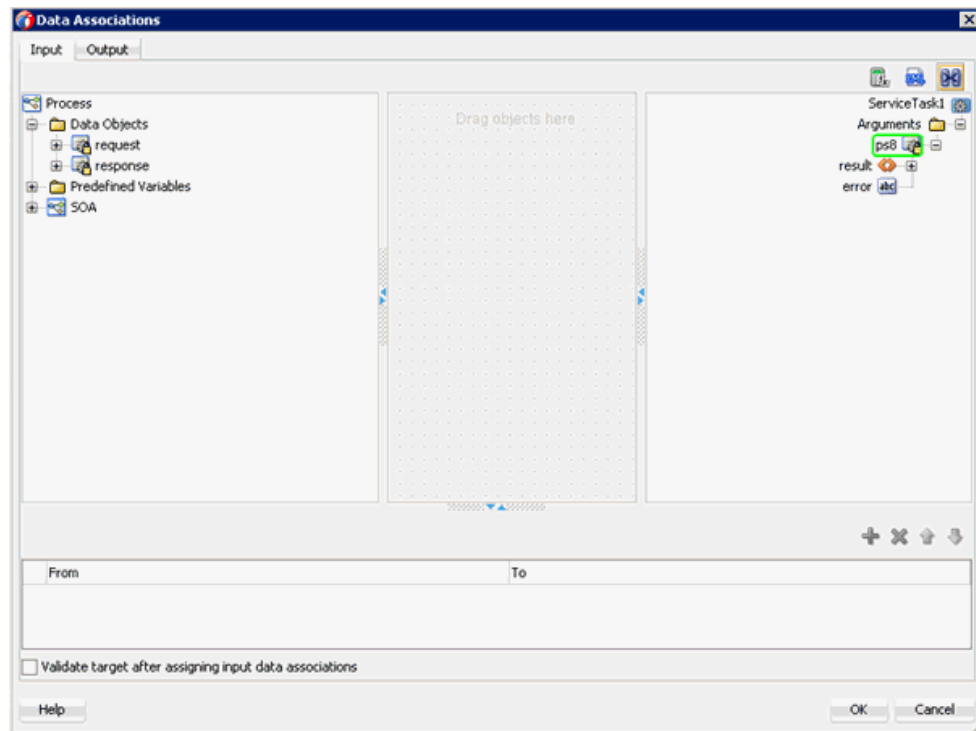
The Data Associations dialog is displayed, as shown in [Figure 6–33](#).



**Figure 6–33 Data Associations Dialog**

26. In the Input tab, click the XSL Transformation icon in the top right pane.
27. Drag and drop the XSL Transformation icon to the **PS8** node, as shown in [Figure 6–34](#).

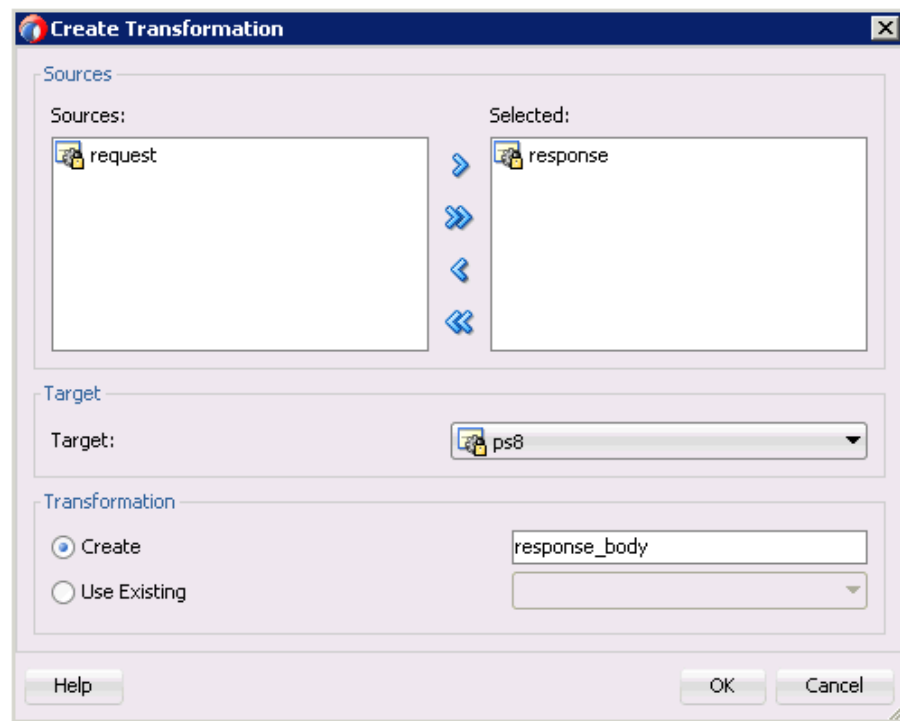
**Figure 6–34** *CompanyCodePeopleSoftResponse Node*



The Create Transformation dialog is displayed.

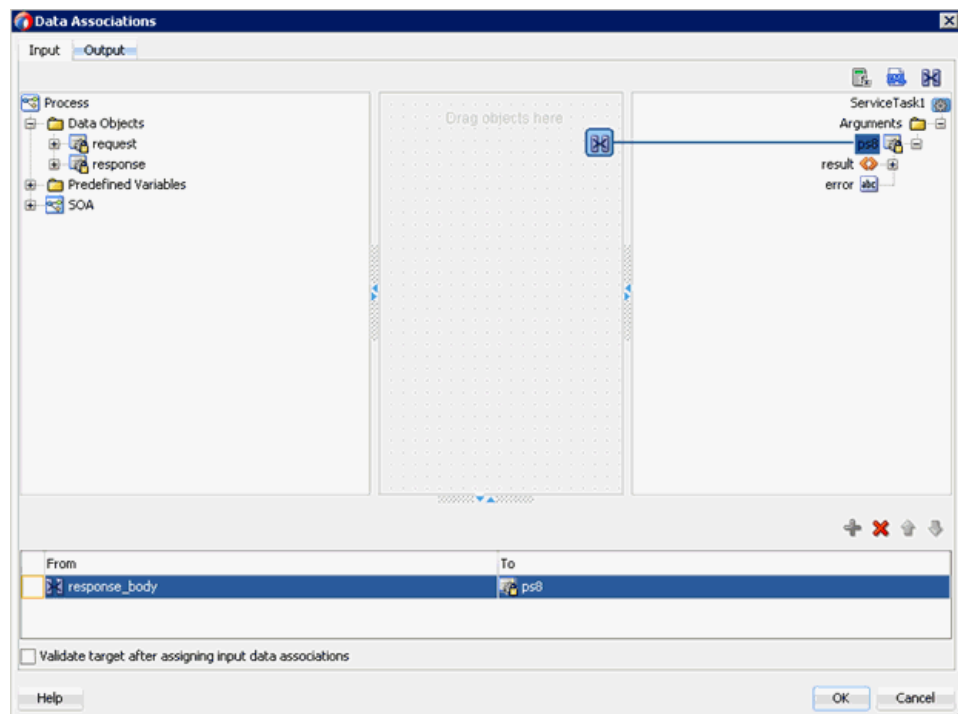
28. Select **Response** in the Sources section and click the right arrow symbol.

The Response object is added to the Selected elements area as shown in [Figure 6–35](#).

**Figure 6–35 Response Object**

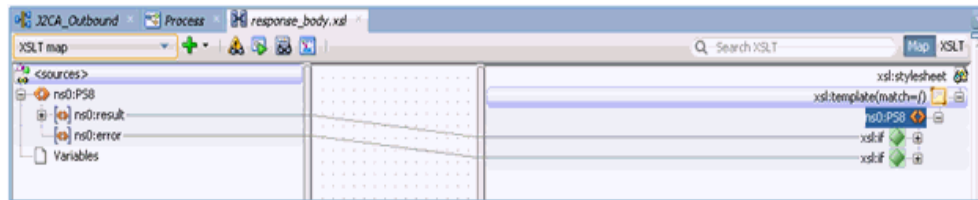
29. Accept the default value selected in the Target drop-down list and the default name in the Create field by clicking **OK**.

You are returned to the Data Associations dialog window with the XSL transformation created, as shown in [Figure 6–36](#).

**Figure 6–36 Data Associations Dialog**

30. Click **OK**.  
You are returned to the Properties - ServiceTask1 dialog.
31. Click **OK**.  
The Response\_body.xml tab is displayed.
32. Automap the Source and Target elements.  
The Auto Map Preferences dialog is displayed.
33. Accept the default values and click **OK**.  
The transformation is completed, as shown in [Figure 6–37](#).

**Figure 6–37 Completed Transformation**



34. Save the transformation.
35. Return to the Process workspace area.  
The ServiceTask1 component is created between the ServiceTask component and the End event component.
36. Click the **Save All** icon in the menu bar to save the new outbound BPM process component that was configured.  
You are now ready to deploy the outbound BPM process.

### 6.4.3 Adjusting for Known Deployment Issues With 12c

For more information on how to adjust for known deployment issues with 12c, see [Section 4.4.3.3, "Adjusting for Known Deployment Issues With 12c"](#) on page 4-26.

### 6.4.4 Deploying the BPM Outbound Process

Perform the following steps to deploy the Mediator outbound process.

1. Right-click the project name in the left pane, select **Deploy**, and then click **J2CA\_Outbound**.  
The Deployment Action page is displayed.
2. Ensure that **Deploy to Application Server** is selected.
3. Click **Next**.  
The Deploy Configuration page is displayed.
4. Leave the default values selected and click **Next**.  
The Select Server page is displayed.
5. Select an available application server that was configured and click **Next**.  
The SOA Servers page is displayed.
6. Select a target SOA server and click **Next**.

The Summary page is displayed.

7. Review and verify all the available deployment information for your project and click **Finish**.

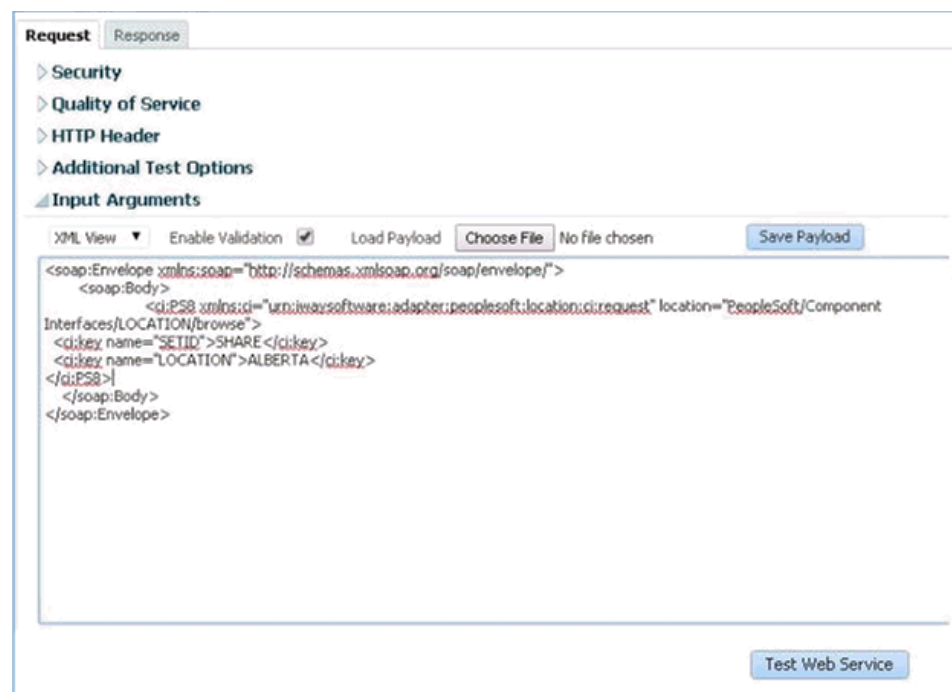
For more information, see [Section 4.4.4, "Deploying the BPEL Outbound Process"](#) on page 4-31.

## 6.4.5 Invoking the Input XML Document in the Oracle Enterprise Manager Console

Perform the following steps to invoke the input XML document in the Oracle Enterprise Manager console.

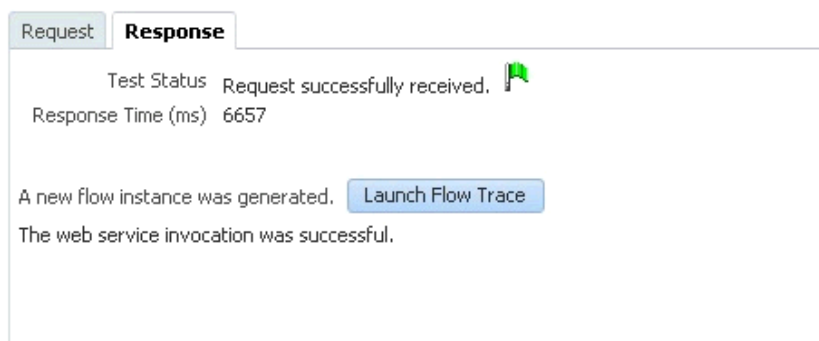
1. Logon to the Oracle Enterprise Manager console.
2. Expand your domain in the left pane followed by the **SOA** folder.
3. Select an available project (for example, J2CA\_Outbound).
4. Click **Test**.
5. Click the **Request** tab.

**Figure 6–38 Request Tab**



6. Provide an appropriate input value in the Value field and click **Test Web Service**, as shown in [Figure 6–38](#).

A response is received in the Response tab to indicate that invocation was successful in the Oracle Enterprise Manager console, as shown in [Figure 6–39](#).

**Figure 6–39 Received Response**

7. Navigate to the defined output directory on your file system and open the XML response document that was received.

The XML response document contains the generated output with values.

## 6.5 Designing an Inbound BPM Process Using Transformations for Event Integration (J2CA Configuration)

This section demonstrates how Oracle Application Adapter for PeopleSoft integrates with PeopleSoft to receive event data.

A sample project has been provided for this inbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\BPM\J2CA\Inbound_Project
```

The following tools are required to complete your adapter design-time configuration:

- Oracle Adapter Application Explorer (Application Explorer)
- Oracle JDeveloper BPM Designer (JDeveloper)

---



---

**Note:** The examples in this chapter demonstrate the use of JDeveloper.

---



---

This section contains the following topics:

- [Section 6.5.1, "Creating an Empty Composite for BPM"](#)
- [Section 6.5.2, "Defining a BPM Inbound Process"](#)

Before you design a BPM process, you must generate the respective WSDL file using Application Explorer. For more information, see [Section 4.5.1, "Generating WSDL for Event Integration"](#) on page 4-34.

### 6.5.1 Creating an Empty Composite for BPM

For more information on how to configure a new Application Server connection in Oracle JDeveloper, see [Section 4.3, "Configuring a New Application Server Connection"](#) on page 4-2.

## 6.5.2 Defining a BPM Inbound Process

This section describes how to define a BPM inbound process, which contains the following topics:

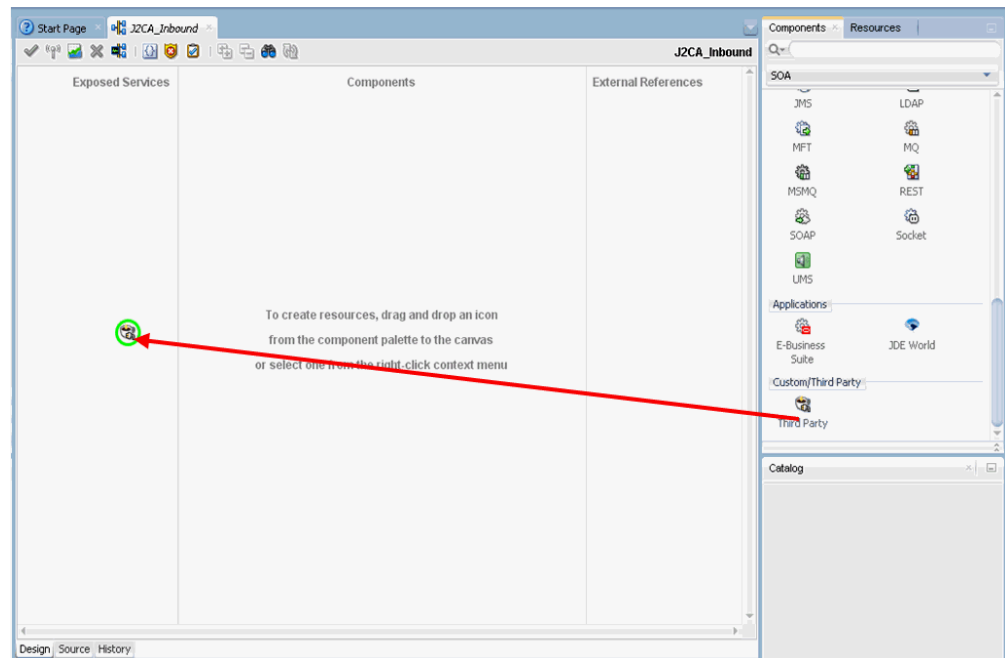
- Section 6.5.2.1, "Configuring a Third Party Adapter Service Component"
- Section 6.5.2.2, "Configuring an Inbound BPM Process Component"
- Section 6.5.2.3, "Creating a File Adapter for the Write Operation"
- Section 6.5.2.4, "Adjusting for Known Deployment Issues With 12c"

### 6.5.2.1 Configuring a Third Party Adapter Service Component

Perform the following steps to create a third party adapter service component:

1. Double-click the created project to load the components.
2. Drag and drop the **Third Party Adapter** component from the Custom/Thirdparty pane to the Exposed References pane, as shown in [Figure 6–40](#).

**Figure 6–40** *Third Party Adapter Component*



The Create Third Party Adapter Service dialog is displayed.

3. Enter a name for the third party adapter service.
4. Ensure that **Service** is selected from the Type list (default).
5. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.

The WSDL Chooser dialog is displayed.

6. Select **File System**, and then browse and select an inbound WSDL file from the following directory:

```
<ADAPTER_HOME>\wsdl.s
```

7. Click **OK**.

The Localize Files dialog is displayed.

8. Click **OK**.

The inbound WSDL file and associated receive\_request XML schema file (.xsd) are imported to the project folder that has been created.

You are returned to the Create Third Party Adapter Service dialog.

9. Click the **Find JCA file** icon, which is located to the right of the JCA File field.

The Transformation Chooser dialog is displayed.

10. Select **File System**, and then browse and select the JCA properties file from the following directory:

```
<ADAPTER_HOME>\wsdl.s
```

11. Click **OK**.

The Copy File message is displayed.

12. Click **Yes**.

A copy of the JCA properties file is made in the project folder.

You are returned to the Create Third Party Adapter Service dialog.

13. Click **OK**.

The third party adapter service component (matmas) is created in the Exposed References pane.

You are now ready to configure an inbound BPM process component.

For more information, see [Section 4.4.3.1, "Configuring a Third Party Adapter Service Component"](#) on page 4-11.

### 6.5.2.2 Configuring an Inbound BPM Process Component

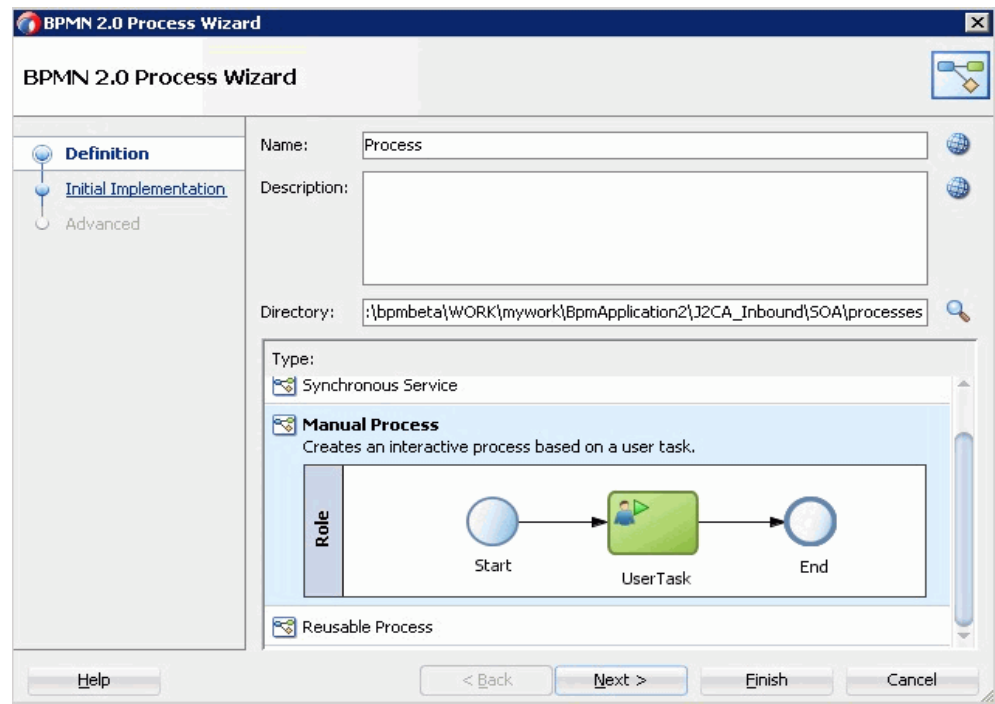
This section describes how to configure an inbound BPM process component.

Perform the following steps to configure an inbound BPM process component:

1. Drag and drop the **BPMN Process** component from the Components pane to the Components pane.

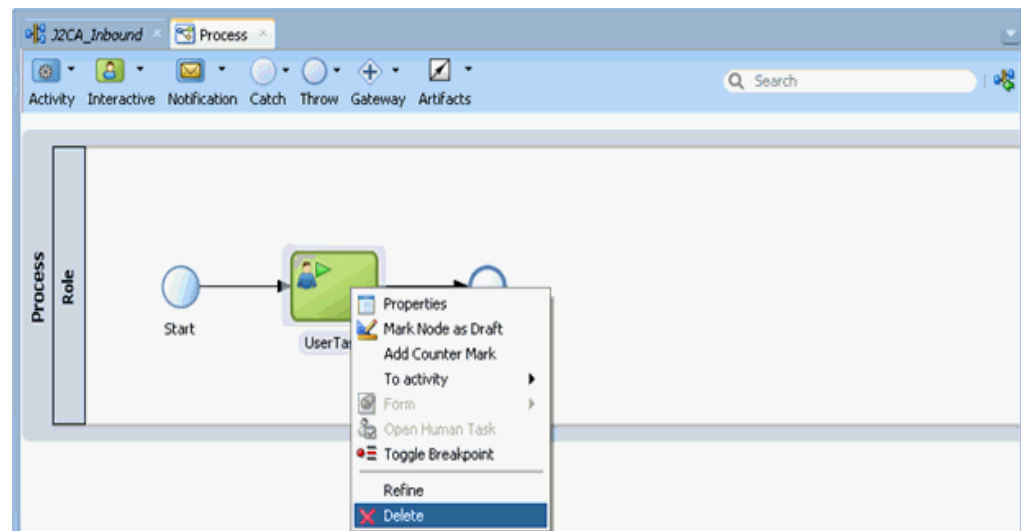
The Create BPMN Process dialog is displayed, as shown in [Figure 6-41](#).



**Figure 6–41 Create BPMN Process Dialog**

2. Select **Manual Process** in the Type section.
3. Click **Finish**.

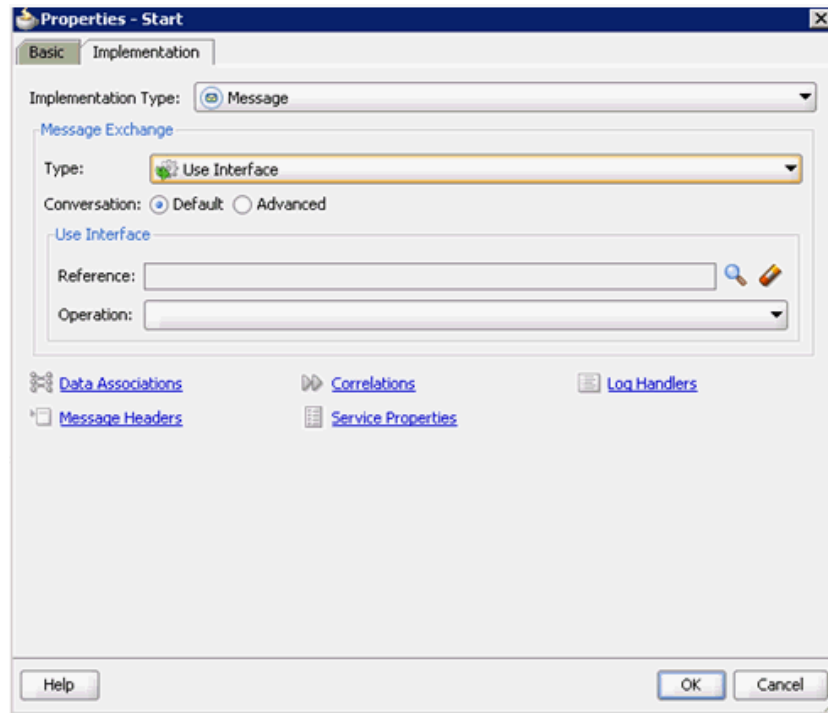
The BPMN process is displayed, as shown in [Figure 6–42](#).

**Figure 6–42 BPMN Process**

4. Right-click **UserTask** and select **Delete** from the menu.
5. Double-click the **Start** event component.  
The Properties - Start dialog is displayed.
6. Click the **Implementation** tab.

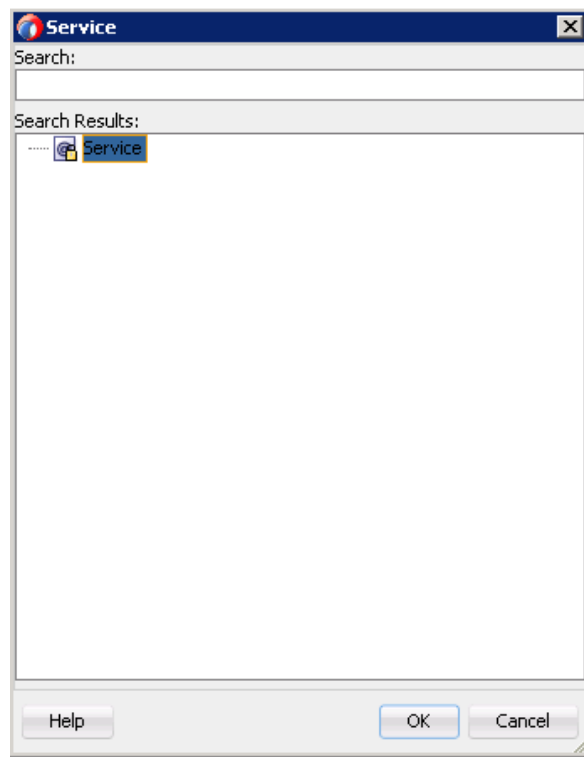
7. Select **Message** from the Implementation Type list.
8. Select **Use Interface** from the Message Exchange Type drop-down list.
9. Click the **Browse** icon to the right of the Reference field, as shown in [Figure 6-43](#).

**Figure 6-43** Browse Icon



The Service dialog is displayed, as shown in [Figure 6-44](#).

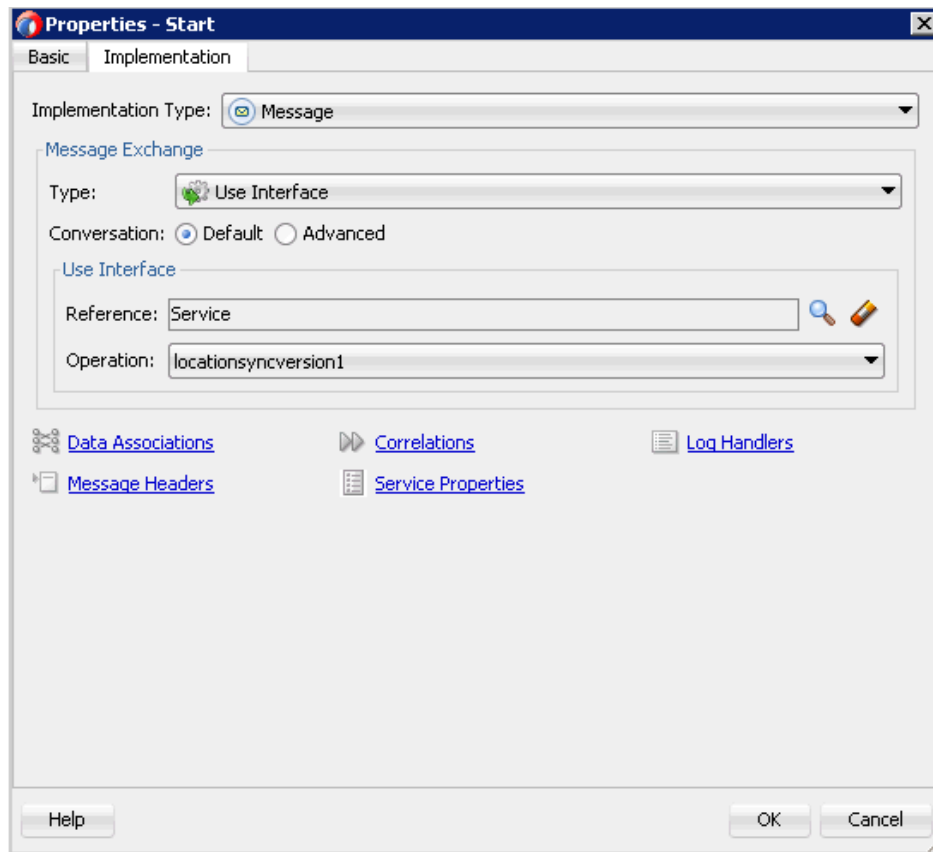
**Figure 6–44** *Service Dialog*



10. Select the Third Party Service that has been created and click **OK**.

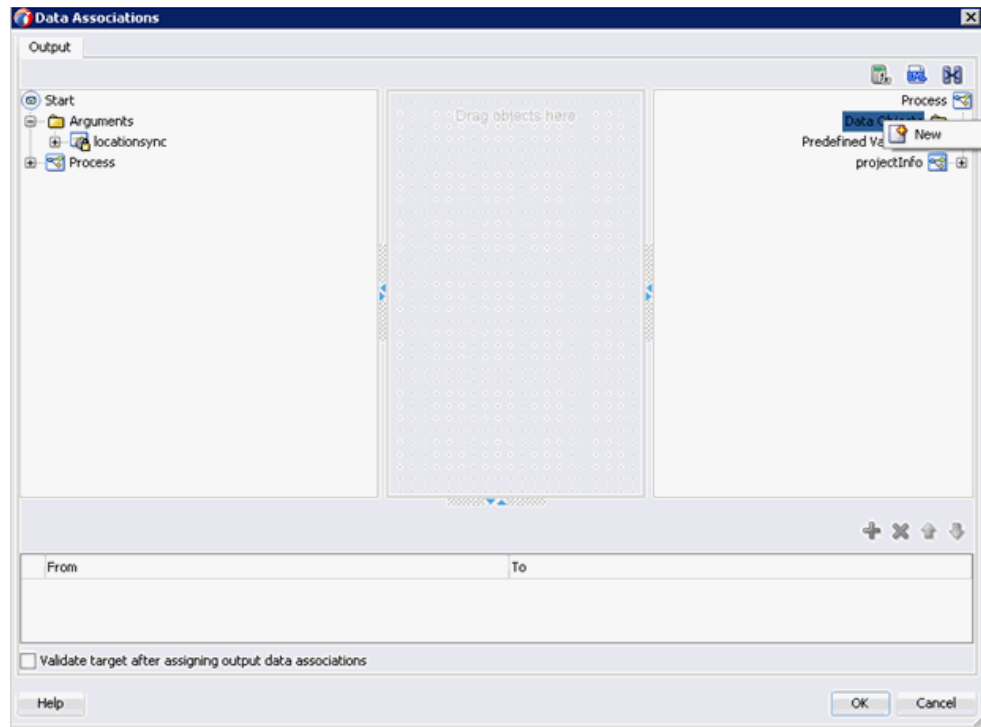
You are returned to the Properties - Start dialog, as shown in [Figure 6–45](#).

**Figure 6–45 Properties - Start Dialog**



11. Click the **Data Associations** icon.

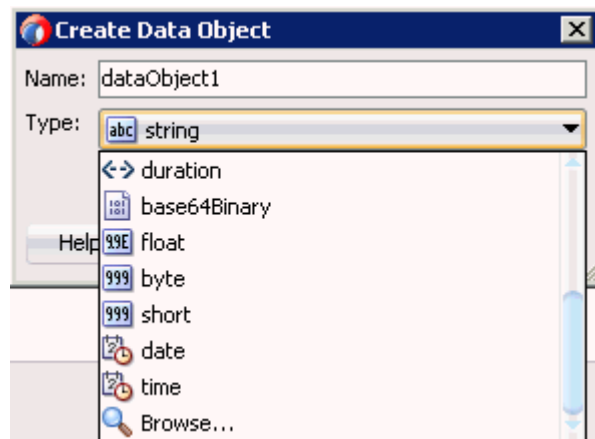
The Data Associations dialog is displayed, as shown in [Figure 6–46](#).

**Figure 6–46 Data Associations Dialog**

12. Right-click the **Data Object** node in the right pane and select **New**.

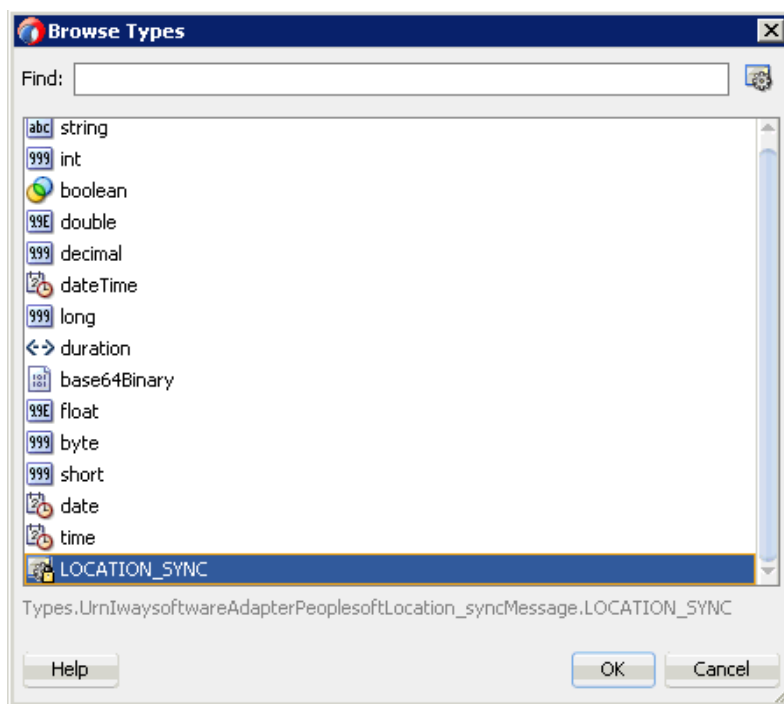
The Create Data Object dialog is displayed.

13. Enter a name in the Name field, and then click the drop-down button in the Type field and select **Browse** from the list, as shown in [Figure 6–47](#).

**Figure 6–47 Create Data Object Dialog**

The Browse Types dialog is displayed, as shown in [Figure 6–48](#).

**Figure 6–48 Browse Types Dialog**

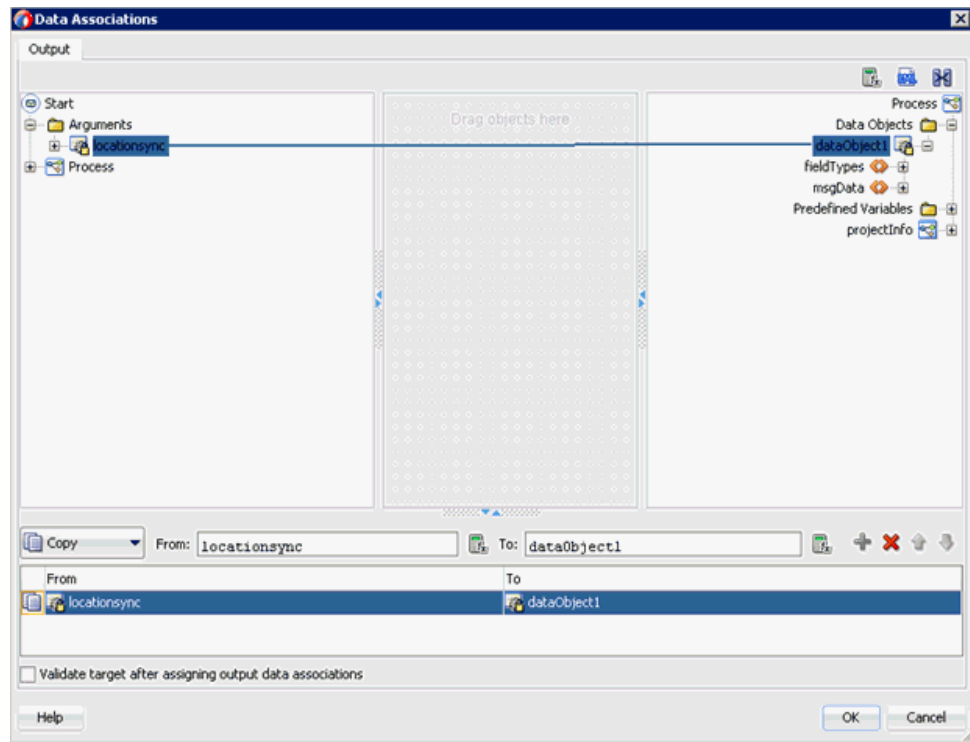


14. Select the component and click **OK**.

You are returned to the Create Data Object dialog.

15. Click **OK**.

The Data Object that has been created is displayed under the Data Objects node in the Data Associations dialog, as shown in [Figure 6–49](#).

**Figure 6–49 Data Associations Dialog**

16. Select and drag the **locationsync** Argument under the Start node in the left pane and drag it to the Data Object in the right pane.

17. Click **OK**.

You are returned to the Properties - Start dialog.

18. Click **OK**.

You are returned to the Process workspace area.

19. Double-click the created project to load the components.

20. Click the **Save All** icon in the menu bar to save the new inbound BPM process component that was configured.

You are now ready to create a File adapter for the write operation.

### 6.5.2.3 Creating a File Adapter for the Write Operation

This section describes how to create a File adapter for the write operation.

Perform the following steps to create a File adapter for the write operation:

1. Drag and drop the **File Adapter** component from the Technology Adapters pane to the External References pane.

The Adapter Configuration Wizard is displayed.

2. Type a name for the new File adapter in the Name field and click **Next**.

The Adapter Interface page is displayed.

3. Ensure that the **Define from operation and schema (specified later)** option is selected.

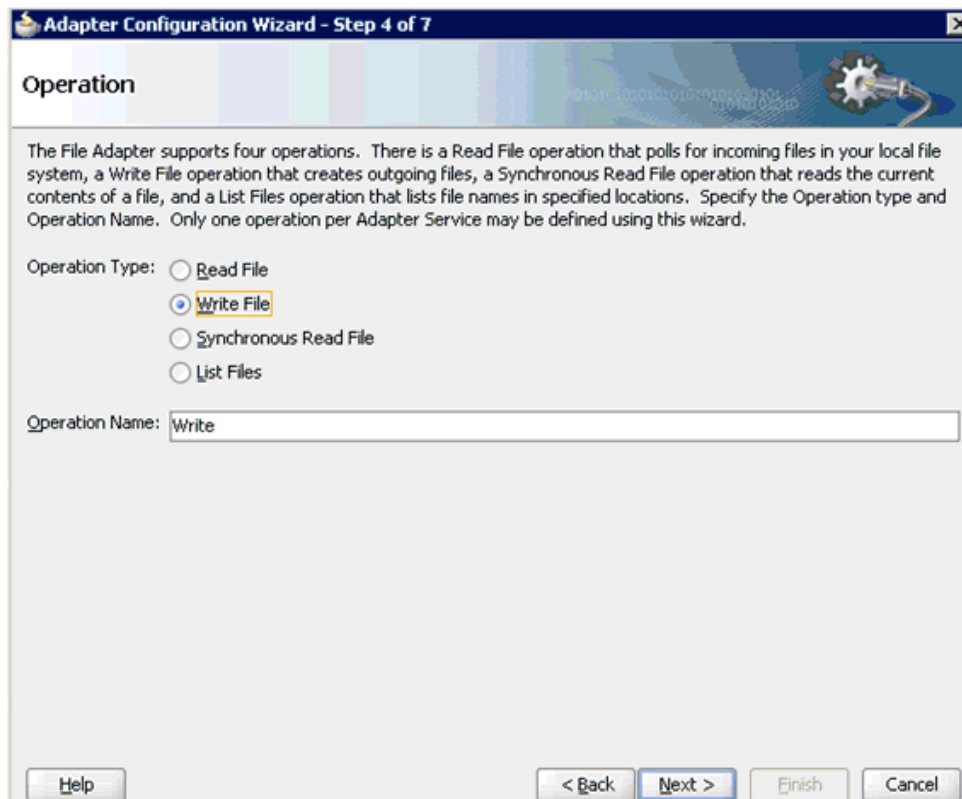
4. Click **Next**.

The File Server Connection page is displayed.

5. Click **Next**.

The Operation page is displayed, as shown in [Figure 6–50](#).

**Figure 6–50 Operation Page**



6. Select **Write File** from the list of Operation Type options and specify an Operation Name (for example, Write).

7. Click **Next**.

The File Configuration page is displayed.

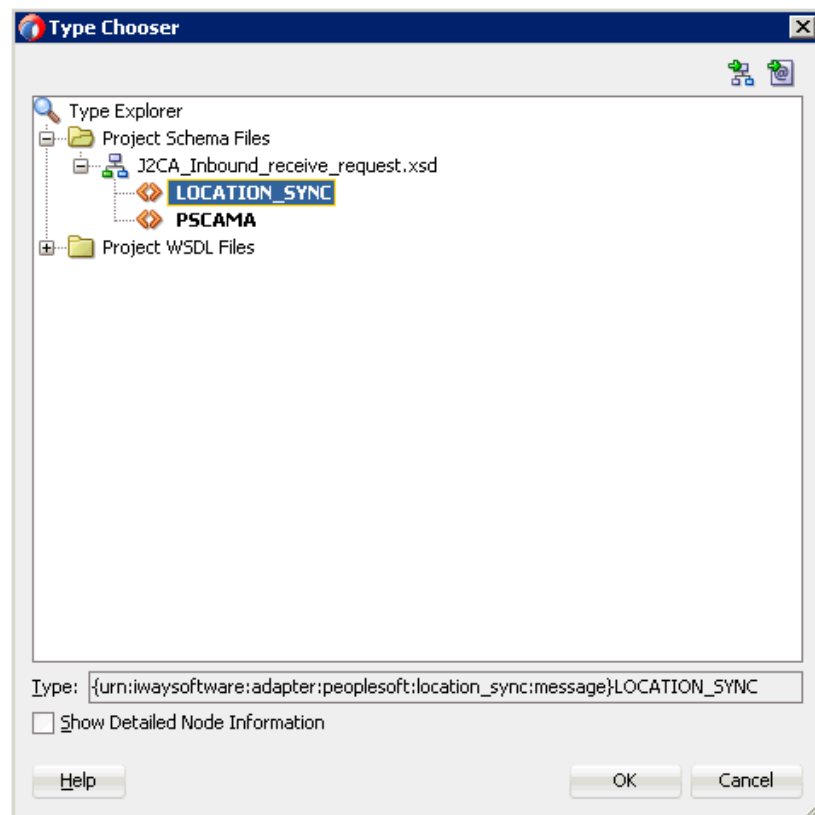
8. Specify a location on your file system where the output file is written.
9. In the File Naming Convention field, specify a name for the output file.
10. Click **Next**.

The Messages page is displayed.

11. Click **Browse**, which is located to the right of the URL field.

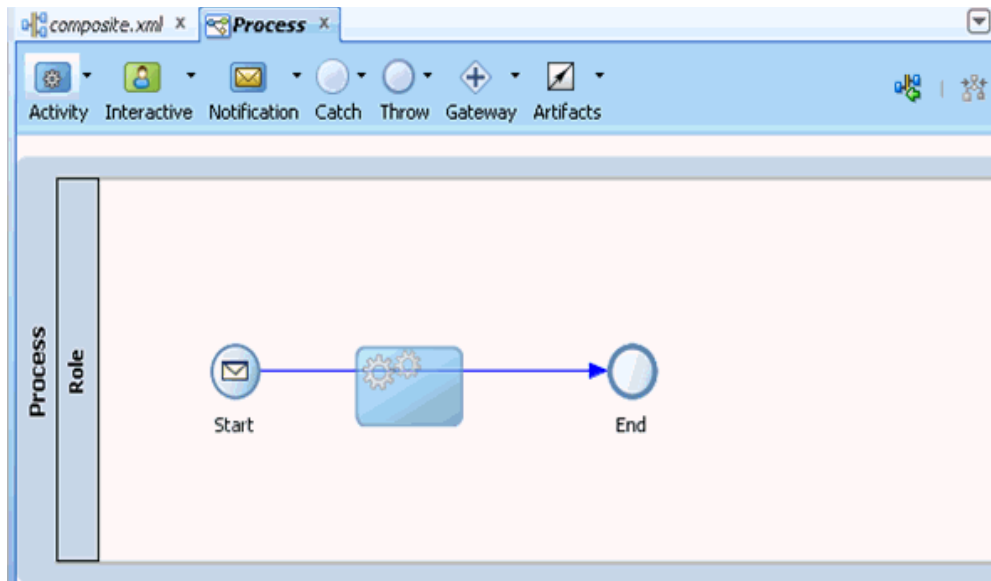
The Type Chooser dialog is displayed, as shown in [Figure 6–51](#).



**Figure 6–51 Type Chooser Dialog**

12. Expand **Project Schema Files** and **J2CA\_Inbound\_receive\_request.xsd**.
13. Select the available schema.
14. Click **OK**.  
You are returned to the Messages page.
15. Click **Next**.  
The Finish page is displayed.
16. Click **Finish**.  
The File Adapter service is created in the External References pane.
17. Double-click the BPMN Process component.  
The BPMN process is displayed.
18. Click the **Activity** icon, and select **Service**.
19. Drop the Service icon on the wire between the Start and End event components, as shown in [Figure 6–52](#).

**Figure 6–52 Activity Icon**

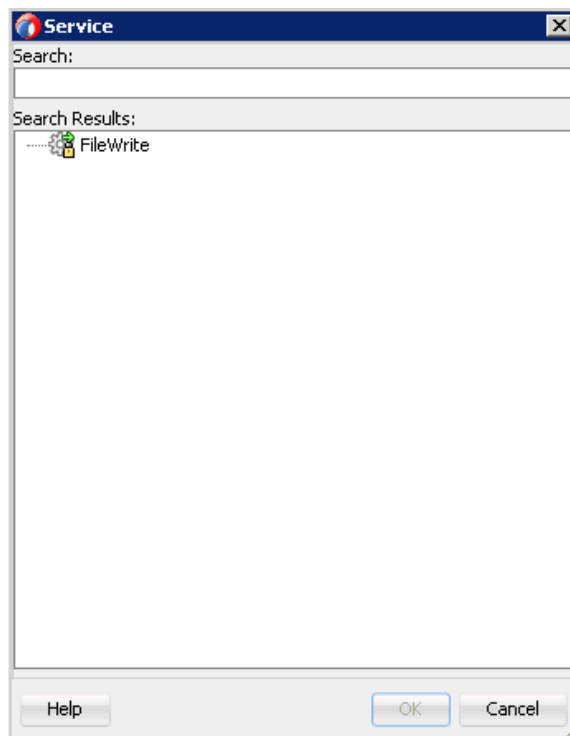


The Properties - ServiceTask dialog is displayed.

20. Click the **Implementation** tab.
21. Select **Service Task** from the Implementation Type list.
22. Select **Service Call** from the Message Exchange Type list.
23. Click the **Browse** icon to the right of the Service field.

The Type dialog is displayed, as shown in [Figure 6–53](#).

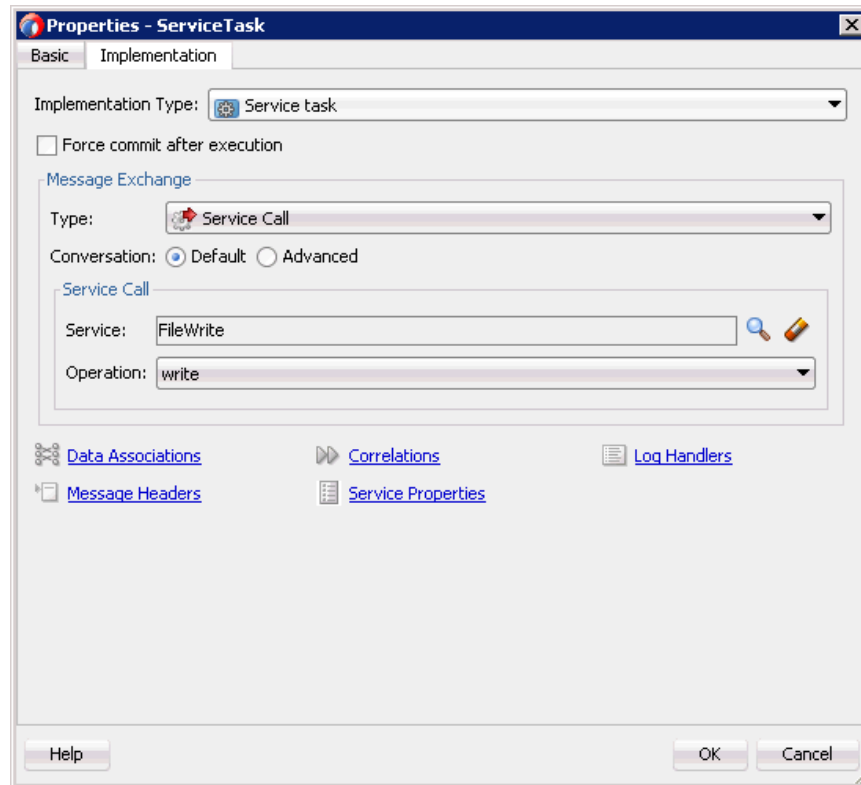
**Figure 6–53 Type Dialog**



24. Select the service for write operation that has been created and click **OK**.

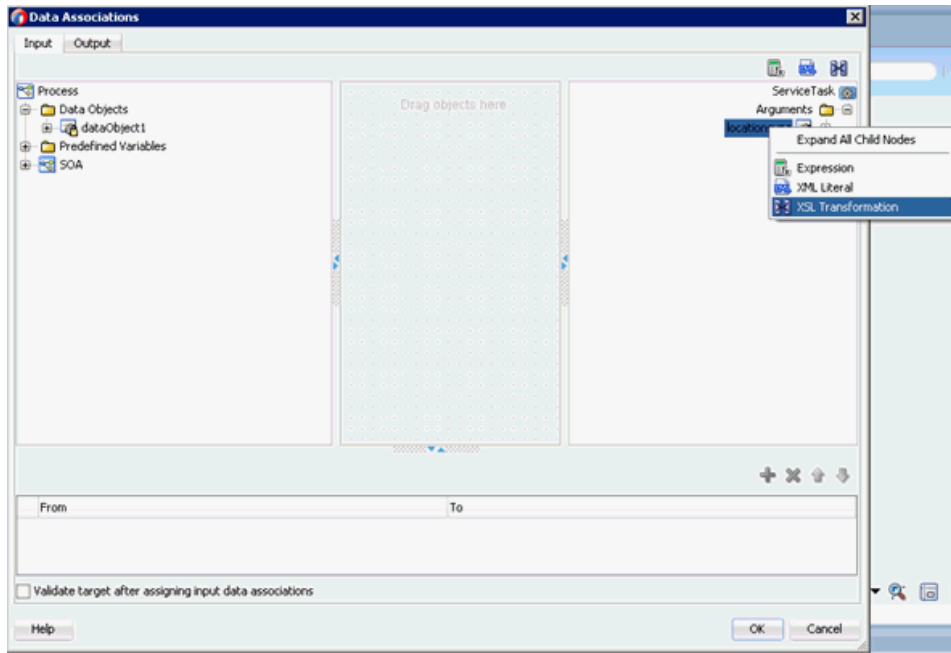
You are returned to the Properties - ServiceTask dialog, as shown in [Figure 6–54](#).

**Figure 6–54** *Properties - ServiceTask Dialog*



25. Click the **Data Associations** hyperlink.  
The Data Associations dialog is displayed.
26. Right-click the **locationsync** argument on the right pane and select **XSL Transformation**, as shown in [Figure 6–55](#).

**Figure 6–55 XSL Transformation**

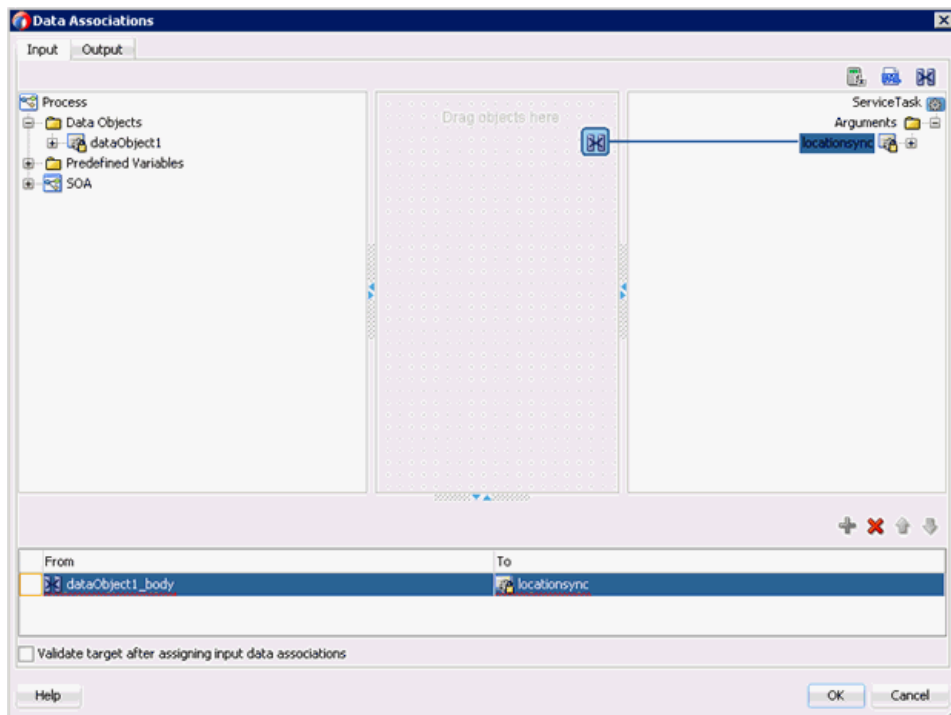


The Create Transformation dialog is displayed.

27. Select the created data object in the Sources area and click the right arrow icon so that the created data object is added to the Selected elements area.
28. Click OK.

You are returned to the Data Associations dialog, as shown in [Figure 6–56](#).

**Figure 6–56 Data Associations Dialog**



29. Click **OK**.

You are returned to the Properties - ServiceTask dialog.

30. Click **OK**.

The dataobject1\_body.xml tab is displayed.

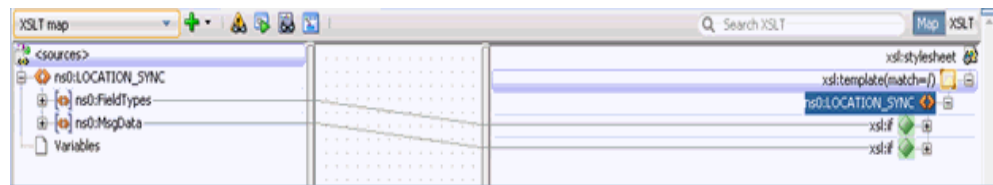
## 31. Automap the Source and Target elements.

The Auto Map Preferences dialog is displayed.

32. Accept the default values and click **OK**.

The transformation is completed, as shown in [Figure 6–57](#).

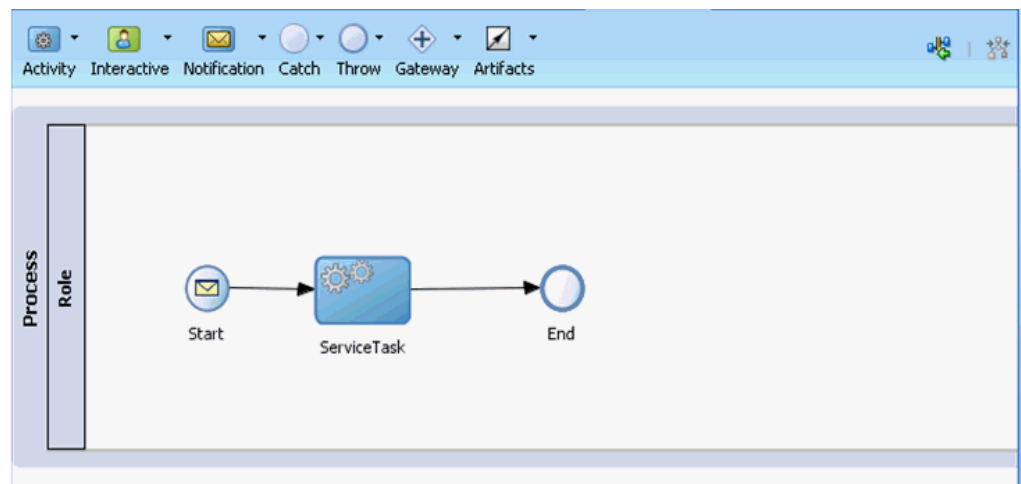
**Figure 6–57 Completed Transformation**



## 33. Save the transformation.

34. Return to the Process workspace area, as shown in [Figure 6–58](#).

**Figure 6–58 Process Workspace Area**



The ServiceTask component is created between the Start event component and the End event component.

35. Click the **Save All** icon in the menu bar to save the new inbound BPM process component that was configured.

#### 6.5.2.4 Adjusting for Known Deployment Issues With 12c

For more information on how to adjust for known deployment issues with 12c, see [Section 4.4.3.3, "Adjusting for Known Deployment Issues With 12c"](#) on page 4-26.

You are now ready to deploy the inbound BPM process. You can follow the same procedure that is described in [Section 4.5.4, "Deploying the BPEL Inbound Process"](#) on

page 4-44. For more information on how to trigger events in PeopleSoft, see [Section 4.5.5, "Triggering an Event in PeopleSoft"](#) on page 4-45.

## 6.6 Designing an Outbound BPM Process Using Transformations for Service Integration (BSE Configuration)

This section describes how to configure a BPM outbound process to your PeopleSoft system, using a BPM project in Oracle JDeveloper.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\BPM\BSE\Outbound_Project
```

The following tools are required to complete your outbound design-time configuration:

- Oracle Adapter Application Explorer (Application Explorer)
- Oracle JDeveloper BPM Designer (JDeveloper)

This section contains the following topics:

- [Section 6.6.1, "Creating an Empty Composite for BPM"](#)
- [Section 6.6.2, "Defining a BPM Outbound Process"](#)

### Prerequisites

Before you design a BPM outbound process, you must generate the respective WSDL file using Application Explorer. For more information, see [Section 4.6.1, "Generating a WSDL File for Request and Response Services Using a Web Service"](#) on page 4-53.

### 6.6.1 Creating an Empty Composite for BPM

Perform the following steps to create an empty composite for SOA:

1. Create a new BPM application.
2. Enter a name for the BPM Application, and click **Next**.
3. Enter a name in the Project Name field, and click **Next**.
4. From the Composite Template list, select **Empty Composite** and click **Finish**.

For more information, see [Section 4.4.2, "Creating an Empty Composite for SOA"](#) on page 4-9.

### 6.6.2 Defining a BPM Outbound Process

This section describes how to define a BPM outbound process. It contains the following topics:

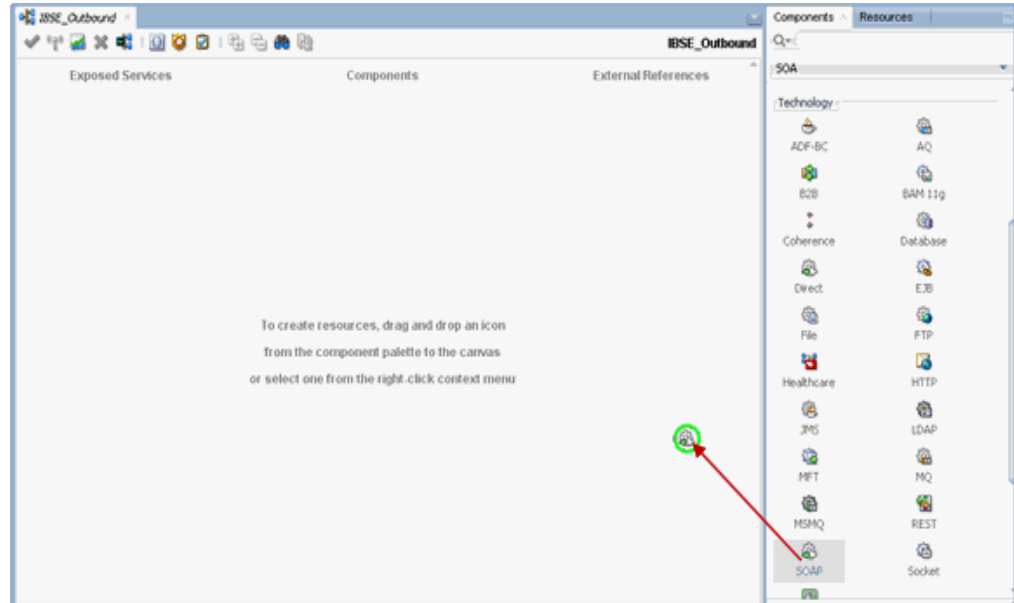
- [Section 6.6.2.1, "Configuring a Web Service Component"](#)
- [Section 6.6.2.2, "Configuring a BPM Process Component"](#)
- [Section 6.6.2.3, "Creating a File Adapter for the Write Operation"](#)

#### 6.6.2.1 Configuring a Web Service Component

Perform the following steps to configure a Web Service component:

1. Double-click the created project to load the components.
2. Drag and drop the **Web Service** node from the Technology Adapters pane to the External References pane, as shown in [Figure 6–59](#).

**Figure 6–59 Web Service Node**



3. Enter an appropriate name for the Web Service and click on the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.
4. In the displayed WSDL Chooser window, navigate to the location where the WSDL is exported from the Application Explorer, and select the WSDL.
5. Click **OK**.
6. In the Web Service pane, click **OK**, as shown in [Figure 6–60](#).

**Figure 6–60 Web Service Pane**

**Create Web Service**

**SOAP**

Create a web service for services external to the SOA composite.

Name: SOAPReference

Type: Reference

WSDL URL: .2c\_SOA\soa\soa\thirdparty\ApplicationAdapters\wsdls\IBSE\_Outbound.wsdl

Port Type: IBSE\_OutboundSoap

Callback Port Type: ----- No Callback -----

copy wsdl and its dependent artifacts into the project.

Transaction Participation: WSDLDriven

Version: DEFAULT

Help OK Cancel

7. In the displayed Localize Files window, click **OK**.  
This will import the WSDL file to the project folder

### 6.6.2.2 Configuring a BPM Process Component

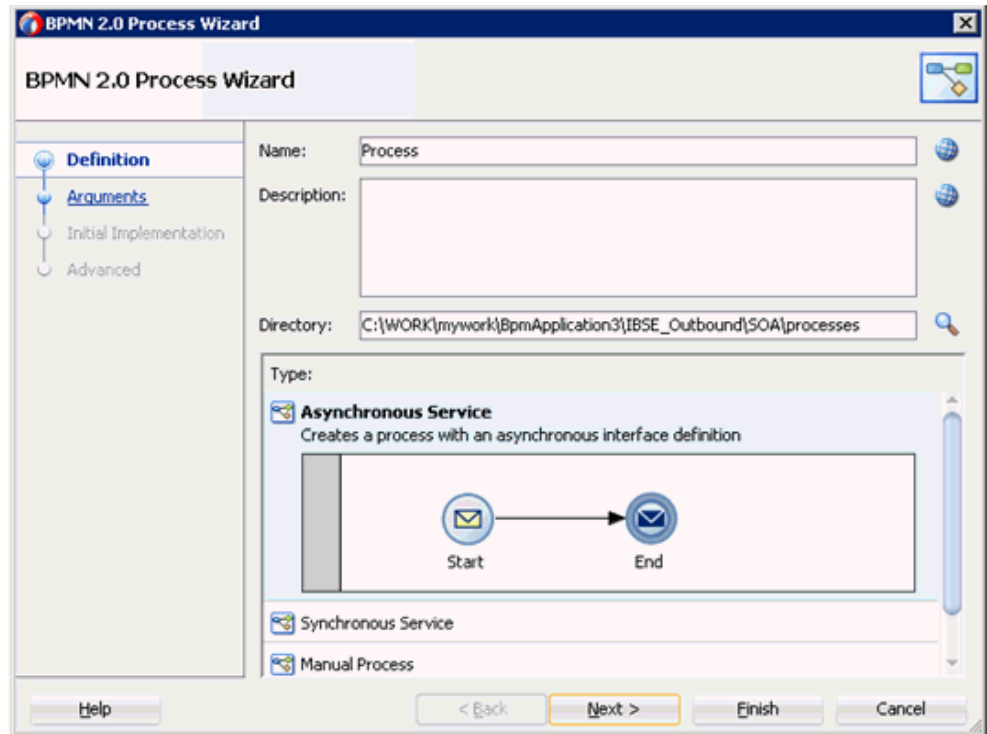
This section describes how to configure an outbound BPM process component.

Perform the following steps to configure a BPM Component:

1. Drag and drop the **BPMN Process** component from the Components pane in to the Components pane.
2. Accept the default option that is selected under the Type area (Asynchronous Service) and click **Finish**, as shown in [Figure 6–61](#).

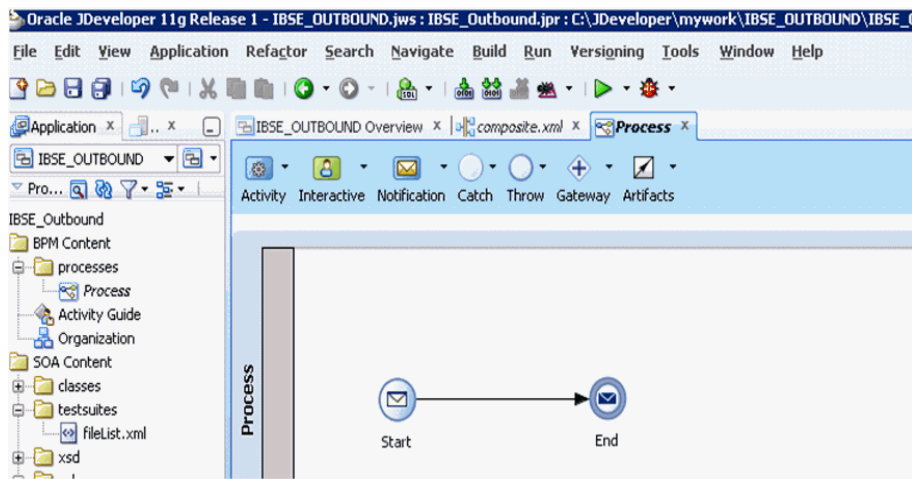


**Figure 6–61 Type Area**



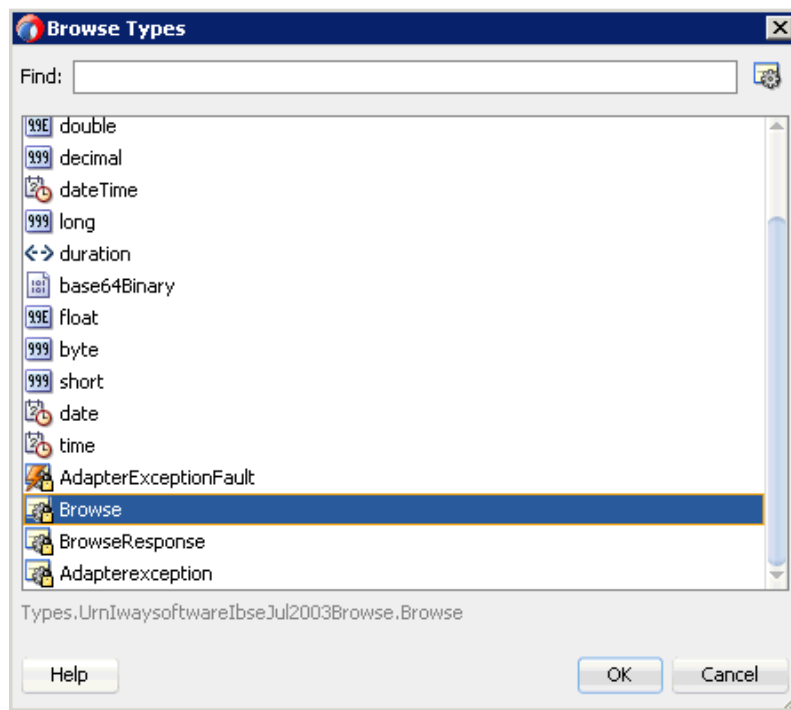
3. Double click on the Start Event component, as shown in Figure 6–62.

**Figure 6–62 Start Event Component**

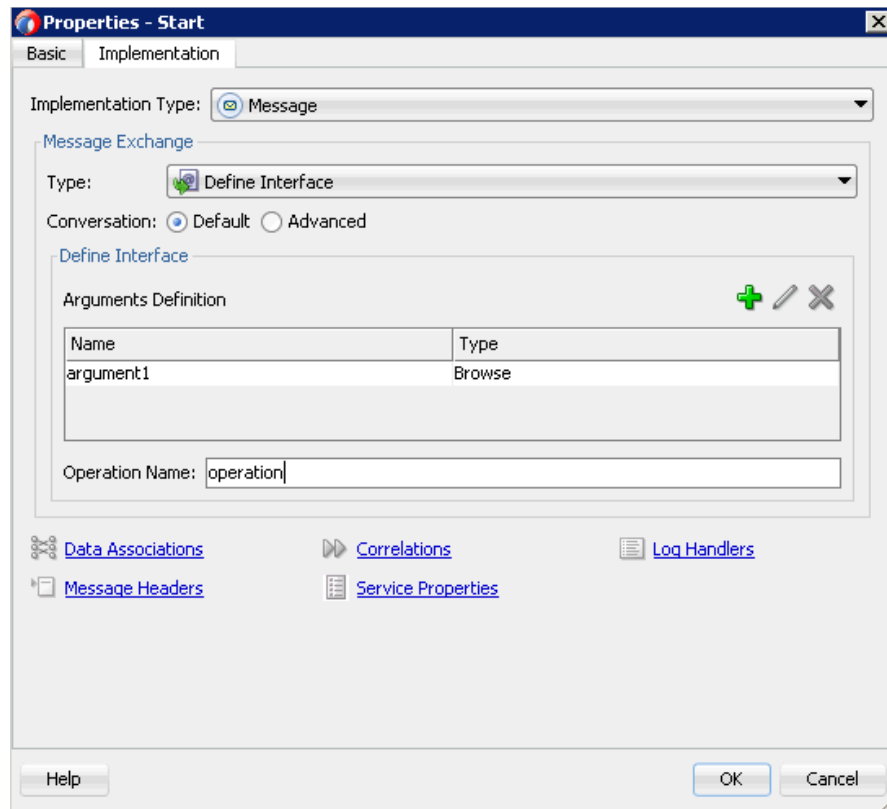


4. In the displayed Properties-start window, click the **Implementation** tab.
5. Click the Plus (+) icon to the right of the Arguments Definition field.  
The Edit Argument window is displayed.
6. Enter a name in the Name field, and then click the Type drop-down list and select **Browse**.
7. Select the **Request** component (for example, browse), and click **OK**, as shown in Figure 6–63.

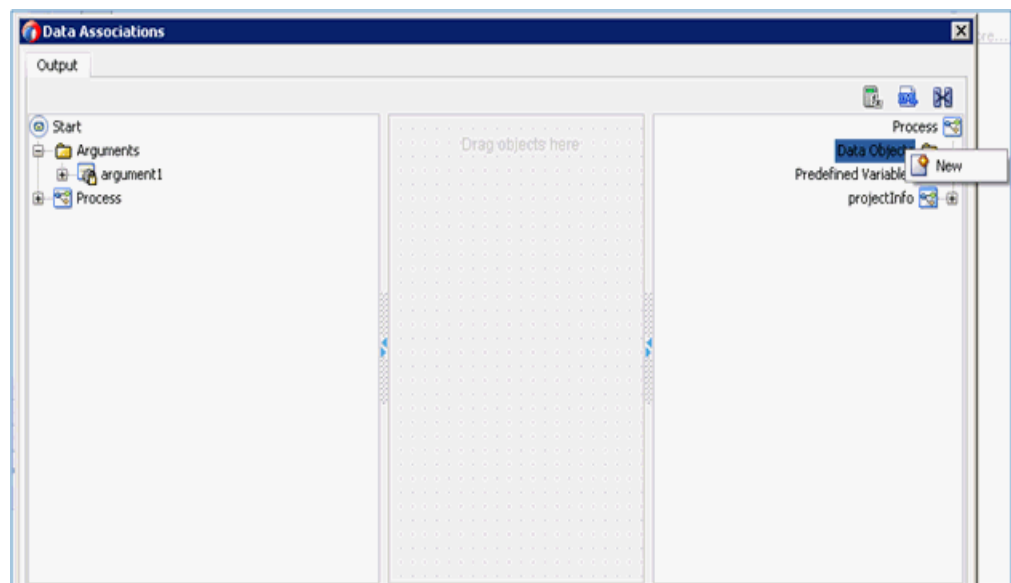
**Figure 6–63 Request Component**



8. In the Edit Argument window that is displayed, click **OK**.  
The Properties - Start window is displayed.
9. In the Operation Name field, change the default entry from **start** to **operation**.
10. Click the **Data Associations** hyperlink, as shown in [Figure 6–64](#).

**Figure 6–64 Properties - Start Window**

- Right-click the **Data Objects** node in the right pane, under **Process**, and select **New**, as shown in [Figure 6–65](#).

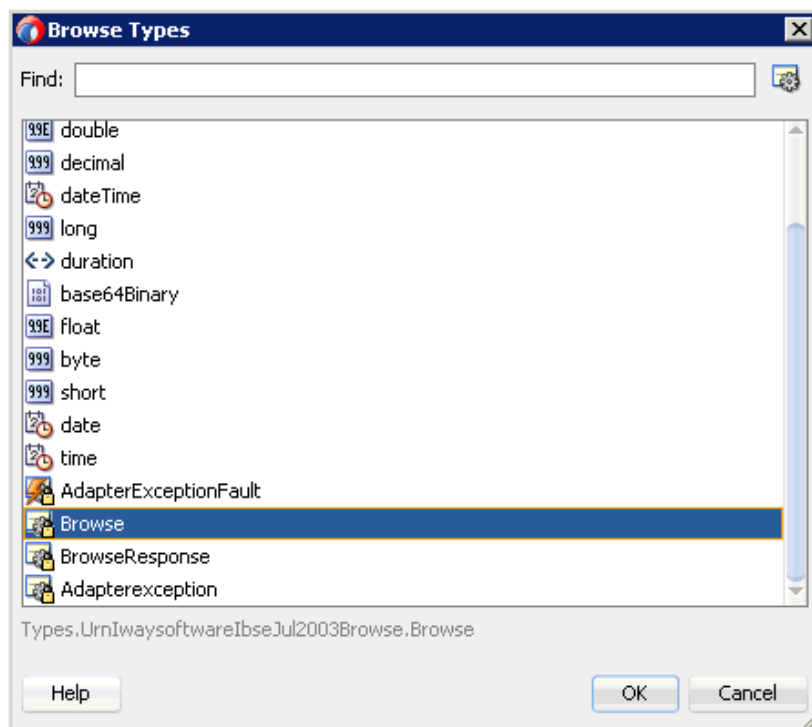
**Figure 6–65 Data Objects Node**

The Create Data Object window is displayed.

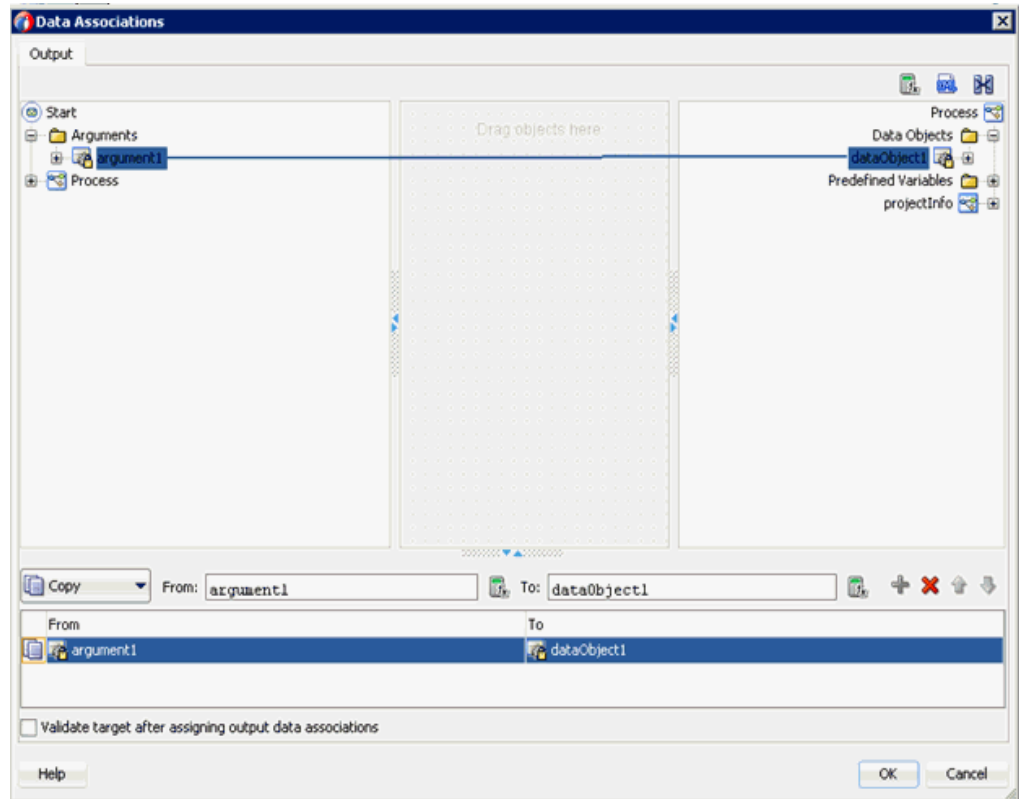
- Enter a name in the Name field, click the Type drop-down list, and select **Browse**.

13. Select the **Request** component (for example, browse) and click **OK**, as shown in [Figure 6–66](#).

**Figure 6–66 Request Component**



14. In the Create Data Object window, click **OK**.  
The Data Associations window is displayed.
15. Select **argument1** under the Arguments node in the left pane and drag and connect it to **dataObject1**, under Data Objects, in the right pane.
16. Click **OK**, as shown in [Figure 6–67](#).

**Figure 6–67 Data Associations**

17. In the Properties - Start window that is displayed, click **OK**.

You are returned to the Process workspace area.

18. Click the **Activity** drop-down menu and select **Service**.

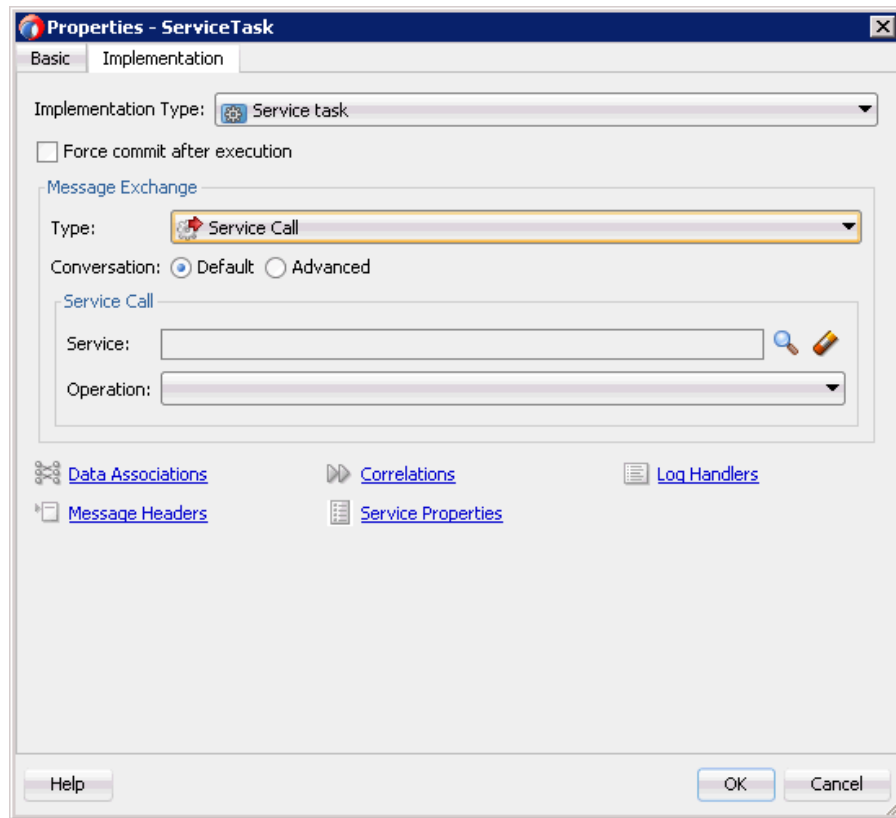
19. Drop the **Service** icon on the wire between the **Start** and **End** event components.

20. In the displayed Properties - ServiceTask window, click the **Implementation** tab.

21. Select **Service Call** from the Message Exchange Type list.

22. Click the **Browse** icon to the right of the Service field, as shown in [Figure 6–68](#).

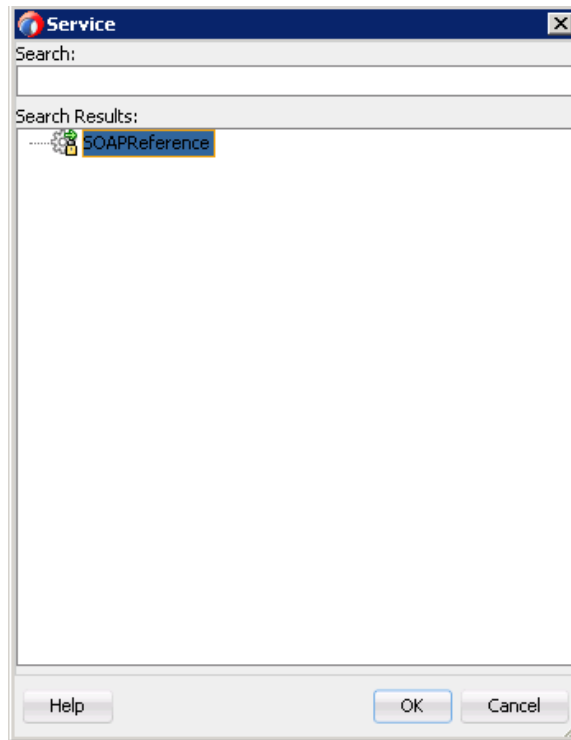
**Figure 6–68 Browse Icon**



The Service window is displayed.

23. Select the Web Service that has been created and click **OK**, as shown in [Figure 6–69](#).

**Figure 6–69 Created Web Service**

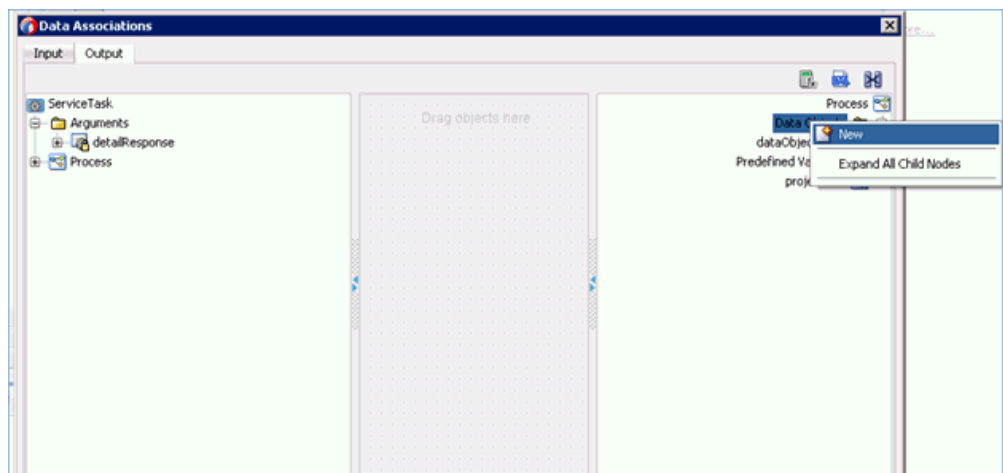


24. In the Properties - ServiceTask window that is displayed, click the **Data Associations** hyperlink.

The Data Associations window is displayed.

25. Create response Data Object by right-clicking the **Data Objects** node in the right pane of the Output tab and selecting **New**, as shown in Figure 6–70.

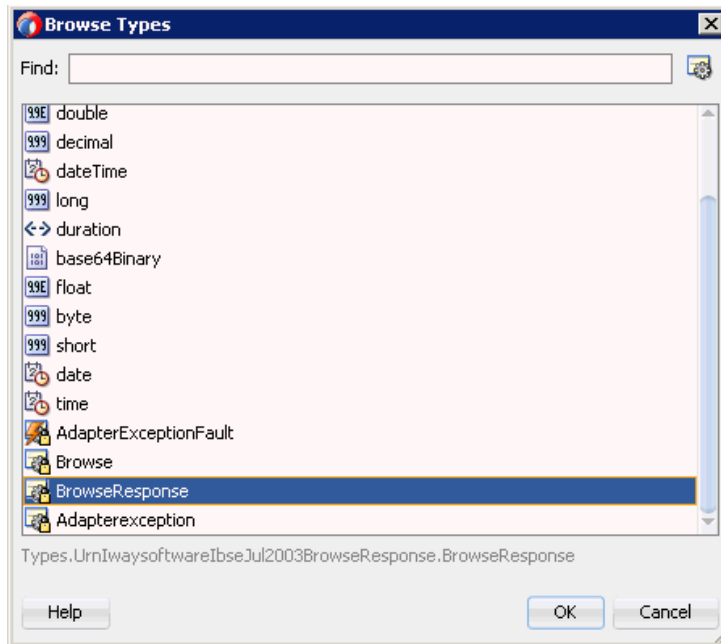
**Figure 6–70 Data Objects Node**



The Create Data Object window is displayed.

26. Enter a name in the Name field, click the Type drop-down list, and select **Browse**.
27. Select the Response component (for example, BrowseResponse) and click **OK**, as shown in Figure 6–71.

**Figure 6–71 Response Component (BrowseResponse)**

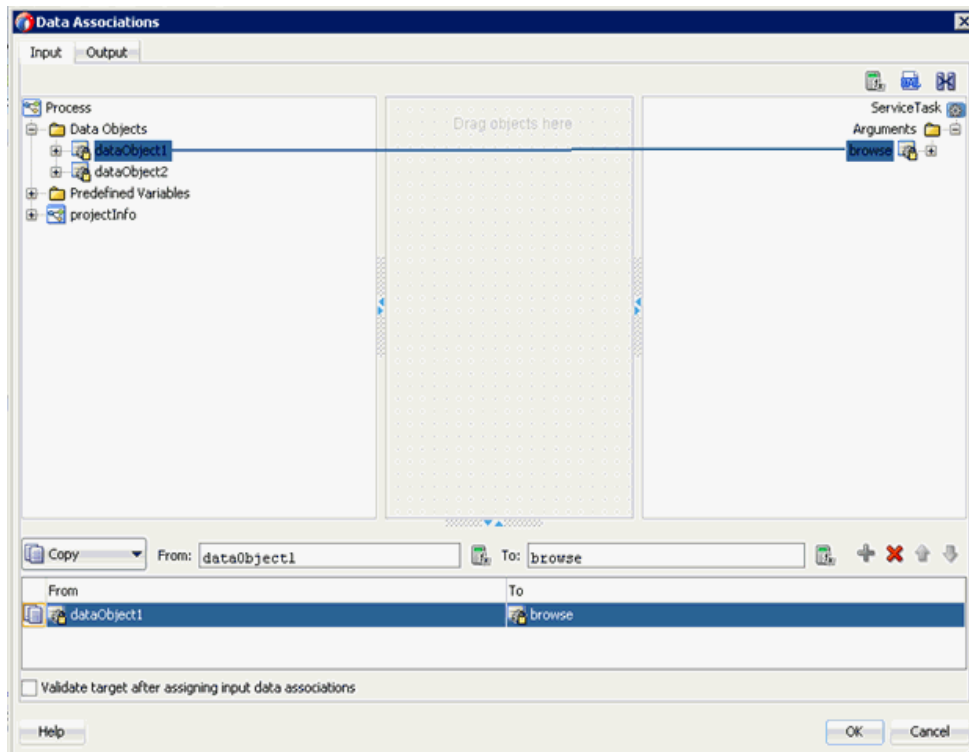


28. In the Create Data Object window, click **OK**.

The Data Associations window is displayed.

29. Select **dataObject1**, under the Data Objects node in the left pane of the Input tab, and drag and connect it to the **browse** node, under the Arguments node in the right pane, as shown in [Figure 6–72](#).

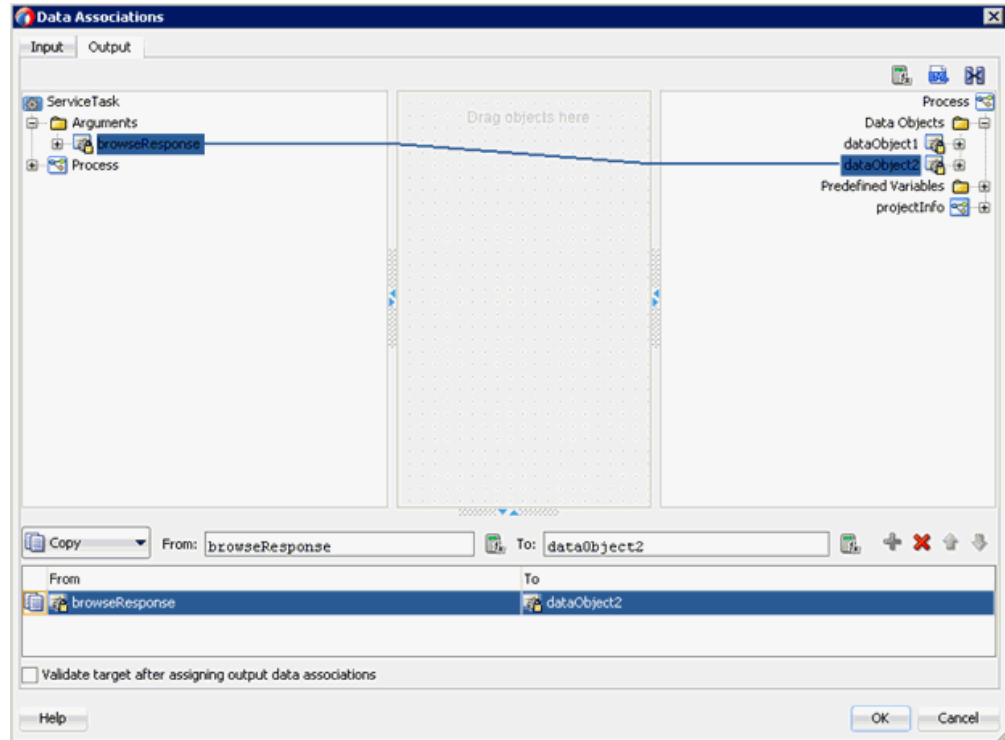
**Figure 6–72 Data Associations**





30. Click on the **Output** tab and select **BrowseResponse** under the Arguments node in the left pane and drag and connect it to **dataObject2** under the Data Objects node.
31. Click **OK**, as shown in [Figure 6-73](#).

**Figure 6-73 Output Tab**



32. In the Properties - ServiceTask window that is displayed, click **OK**.
33. Click the **Save All** icon in the menu bar to save the new outbound BPM process component that was configured.
34. Double-click the **composite.xml** node in the left pane.

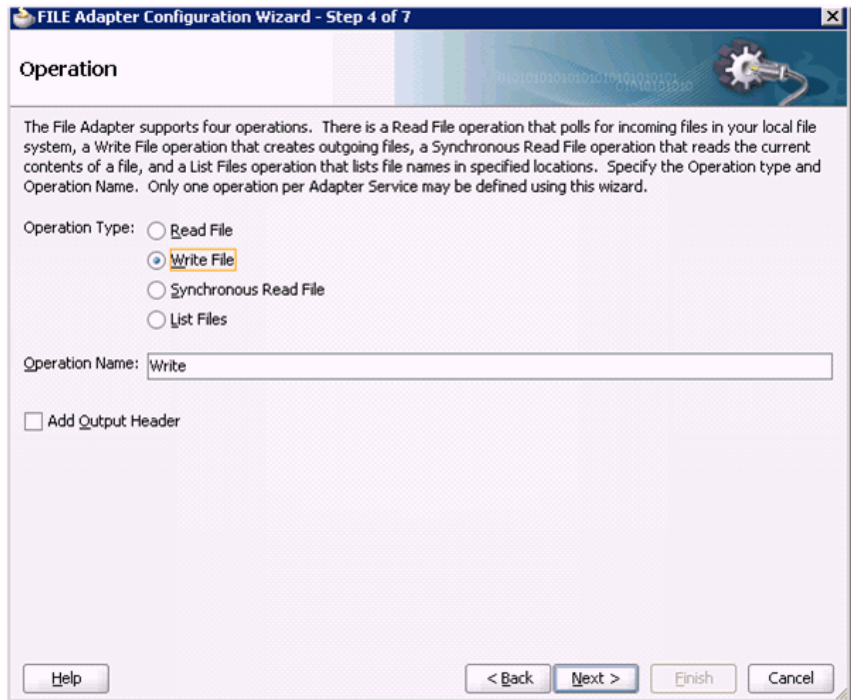
### 6.6.2.3 Creating a File Adapter for the Write Operation

This section describes how to create a File adapter for the write operation.

Perform the following steps to create a File adapter for the write operation:

1. Drag and drop the **File Adapter** component from the Technology Adapters pane to the External References pane, and provide a name for the File Adapter.
2. In the Adapter Interface pane that is displayed, ensure that the **Define from operation and schema (specified later)** option is selected, and click **Next**.
3. Click **Next**.
4. In the Operation pane that is displayed, select **Write File** from the list of Operation Type options, and click **Next**, as shown in [Figure 6-74](#).

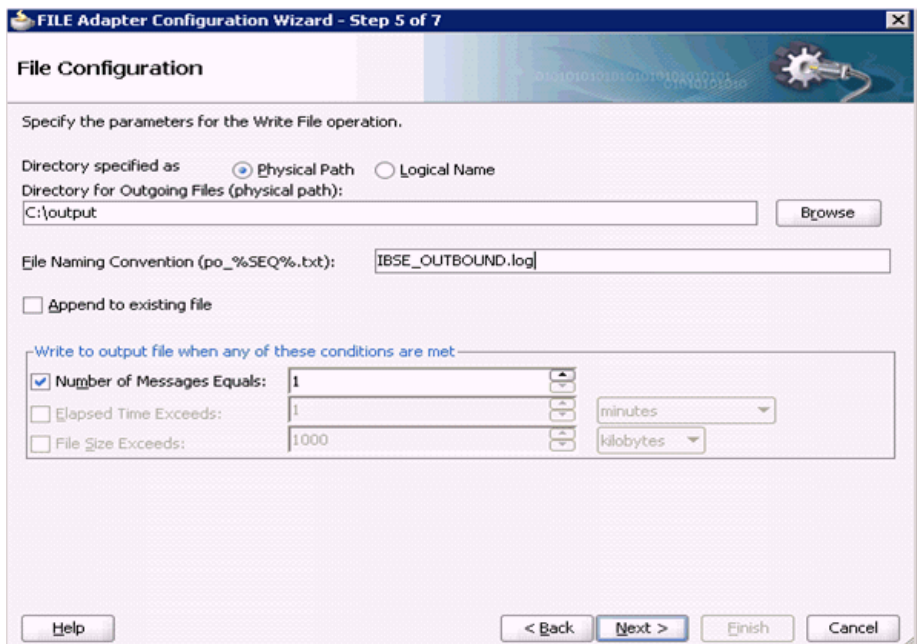
**Figure 6–74 Operation Pane**



The File Configuration pane is displayed.

5. In the Directory for Outgoing Files (physical path) field, specify a location on your file system where the output file is written.
6. In the File Naming Convention field, specify a name for the output file.
7. Click Next, as shown in [Figure 6–75](#).

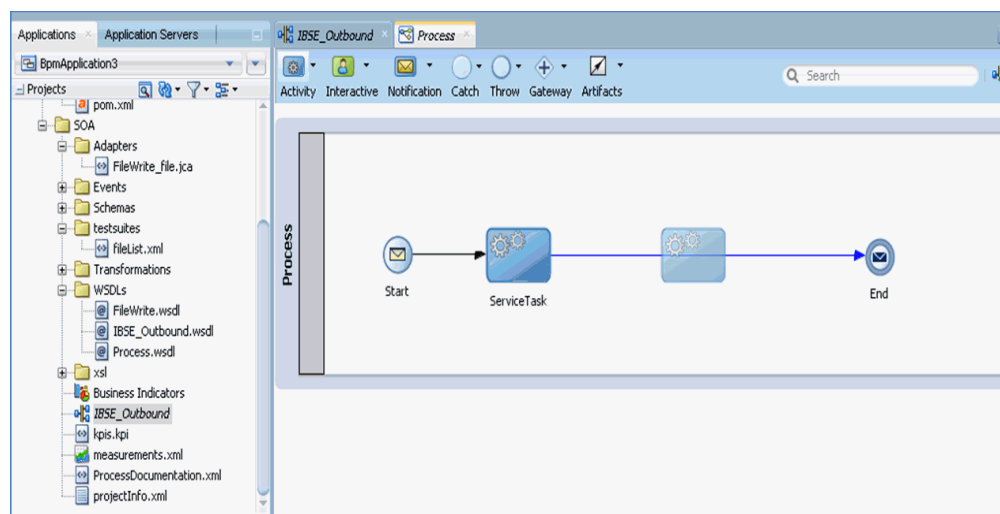
**Figure 6–75 File Configuration Pane**



The Messages pane is displayed.

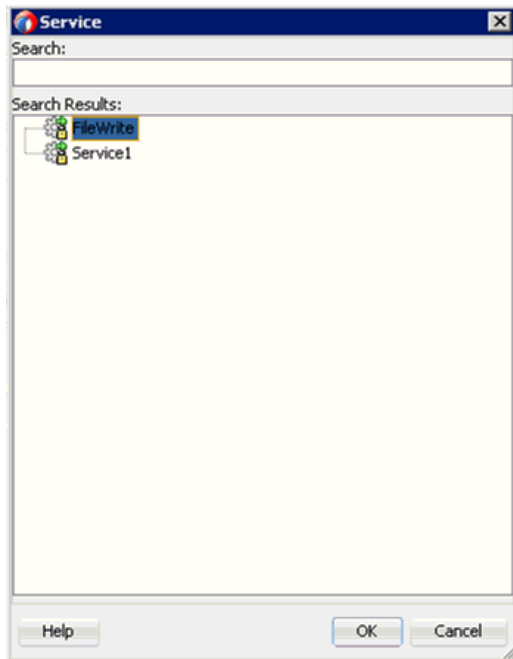
8. Click the **Browse**, which is located to the right of the URL field.
9. In the displayed Type Chooser window, expand **Project WSDL Files, IBSE\_Outbound.wsdl, Inline Schemas** and then select **browseResponse**.
10. Click **OK**.
11. In the Messages pane, click **Next**.
12. In the Finish pane that is displayed, click **Finish**.
13. Click the **Activity** icon.
14. Drop the **Activity** icon on the wire between the **Service Task** and **End** event components, as shown in [Figure 6-76](#).

**Figure 6-76 Activity Icon**



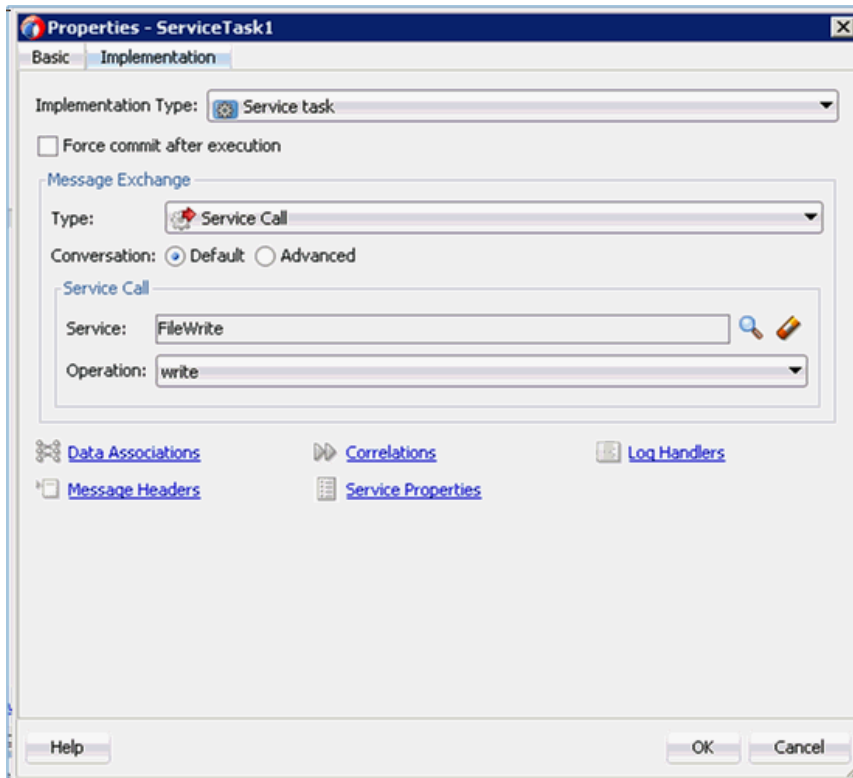
15. In the displayed Properties-ServiceTask1 window, click the **Implementation** tab
16. Select **Service Call** from the Type drop-down list in the Message Exchange section.
17. Click the **Browse** icon to the right of the Service field.
18. Select the service for write operation that has been created and click **OK**, as shown in [Figure 6-77](#).

Figure 6–77 Service Window



19. In the Properties - ServiceTask1 window, click the **Data Associations** hyperlink, as shown in Figure 6–78.

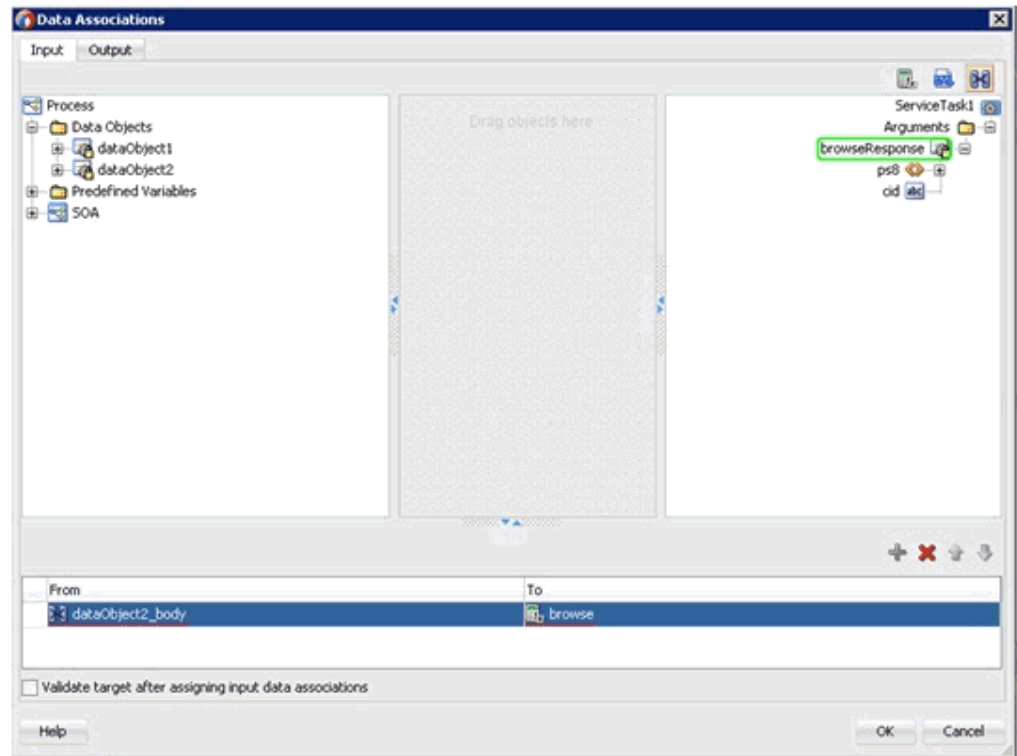
Figure 6–78 Data Associations



20. In the Input tab, click the **XSL Transformation** icon in the top right pane.

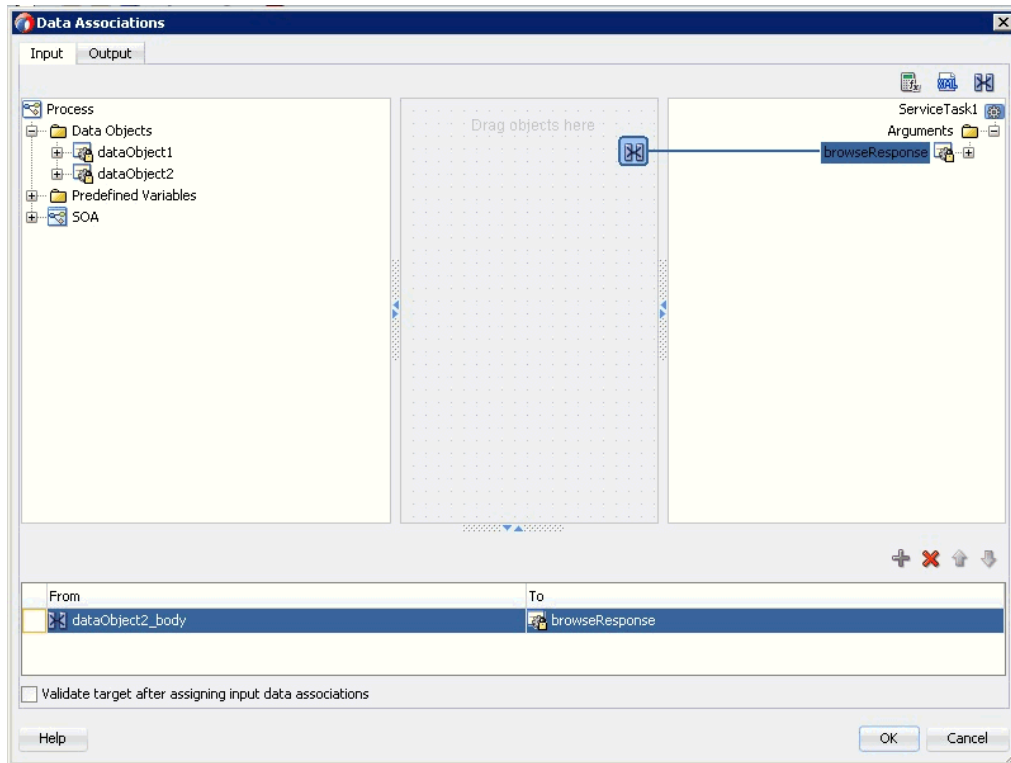
21. Drag and drop the **XSL Transformation** icon to the **browseResponse** node, as shown in [Figure 6-79](#).

**Figure 6-79 Browse Node**



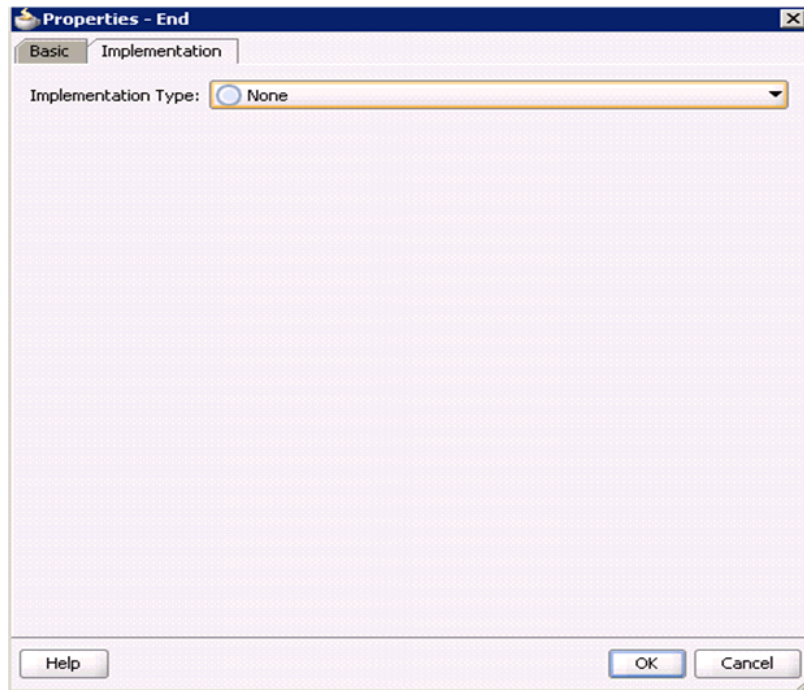
22. In the displayed Create Transformation window, select **dataObject2** in the Sources section and click the right arrow symbol.
23. Accept the default value selected in the Target drop-down list and the default name in the Create field by clicking **OK**.
24. In the Data Associations window, click **OK**, as shown in [Figure 6-80](#).

**Figure 6–80 Data Associations Window**



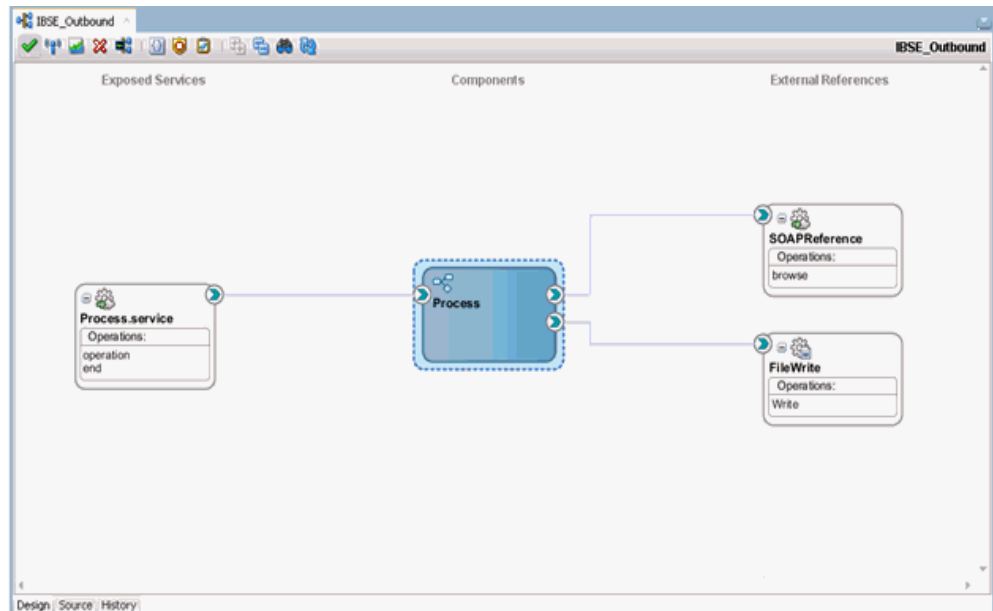
25. In the Properties - ServiceTask1 window, click **OK**.
26. In the response\_body.xml tab, map the **ns0:browseResponse** source element to the **ns0:browseResponse** target element.
27. In the displayed Auto Map Preferences window, retain the default values and click **OK**.
28. Return to the Process workspace area and double-click the **End** event component.
29. In the displayed Properties - End window, click the **Implementation** tab.
30. Select **None** from the Implementation Type drop-down list.
31. Click **OK**, as shown in [Figure 6–81](#).

**Figure 6–81 Implementation Tab**



32. Click the **Save All** icon in the menu bar to save the new outbound BPM component that was configured, as shown in [Figure 6–82](#).

**Figure 6–82 Save All Icon**



You are now ready to deploy the BPM BSE Outbound process. You can follow the same procedure as [Section 6.4.4, "Deploying the BPM Outbound Process"](#) on page 6-26.

Once deployed, you can invoke the input XML as defined in [Section 6.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console"](#) on page 6-27.



---

---

# Configuring an Outbound and Inbound Process for Oracle Service Bus Using sbconsole

---

---

**Note:** With Release 12c (12.2.1.0.0) configuring an outbound and inbound process for Oracle Service Bus using sbconsole has changed.

If you want to create a process for Oracle Service Bus using sbconsole, see *Chapter 2, Configuring an Outbound and Inbound Process for Oracle Service Bus Using sbconsole in the Oracle Fusion Middleware Application Adapters Release Notes for 12c (12.2.1.0.0)*.

---

---

Oracle Application Adapter for PeopleSoft integrates seamlessly with Oracle Service Bus (OSB) to facilitate Web service integration. OSB is based on the Service-Oriented Architecture (SOA). It consumes adapter services exposed as Web Service Definition Language (WSDL) documents.

This chapter contains the following topics:

- [Section 7.1, "Overview of Application Adapter Integration with Oracle Service Bus"](#)
- [Section 7.2, "Configuring an Outbound Process Using sbconsole \(J2CA Configuration\)"](#)
- [Section 7.3, "Configuring an Inbound Process Using sbconsole \(J2CA Configuration\)"](#)
- [Section 7.4, "Configuring an Outbound Process Using sbconsole \(BSE Configuration\)"](#)
- [Section 7.5, "Configuring JMS Proxy Services Using Oracle Service Bus \(J2CA Configuration\)"](#)
- [Section 7.6, "Configuring HTTP Proxy Services Using Oracle Service Bus \(J2CA Configuration\)"](#)

## 7.1 Overview of Application Adapter Integration with Oracle Service Bus

To integrate with Oracle Service Bus (OSB), Oracle Application Adapter for PeopleSoft must be deployed in the same Oracle WebLogic Server as OSB. The underlying adapter services must be exposed as WSDL files, which are generated during design time in Oracle Adapter Application Explorer (Application Explorer) for both request-response (outbound) and event notification (inbound) services of the adapter.

## 7.2 Configuring an Outbound Process Using sbconsole (J2CA Configuration)

This section describes how to configure an outbound process using sbconsole for J2CA configurations.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\OSB\J2CA\PeopleSoft_Sample_J2CA_OSB_Outbound_Project
```

This section includes the following topics:

- [Section 7.2.1, "Starting Oracle Service Bus and Creating Project Folders"](#)
- [Section 7.2.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"](#)
- [Section 7.2.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"](#)
- [Section 7.2.4, "Configuring a WSDL-based Business Service"](#)
- [Section 7.2.5, "Configuring a File Type Business Service"](#)
- [Section 7.2.6, "Configuring a Pipeline With Proxy Service"](#)

### 7.2.1 Starting Oracle Service Bus and Creating Project Folders

This section describes how to start Oracle Service Bus (OSB) and create project folders.

Perform the following steps to start Oracle Service Bus and create project folders:

1. Start the Oracle WebLogic Server for the Oracle WebLogic Server domain that you have configured.
2. Open the Oracle Service Bus Console in a Web browser by entering the following URL:

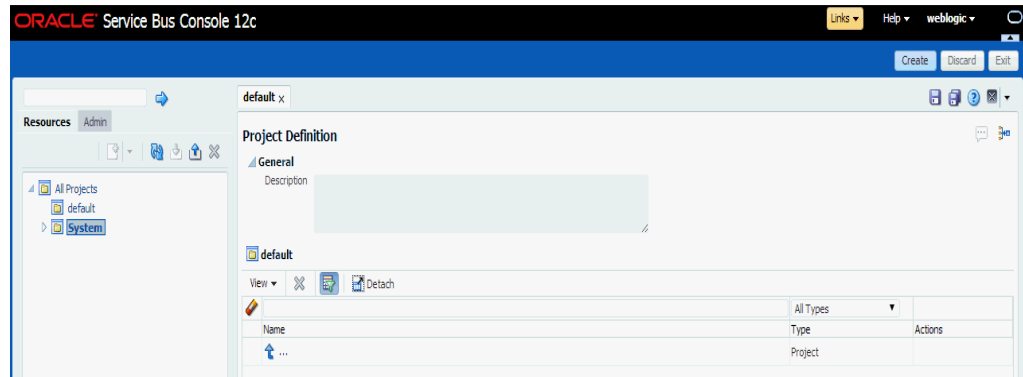
```
http://hostname:port/sbconsole
```

Where *hostname* is the name of the machine where Oracle WebLogic Server is running and *port* is the port for the domain you are using.

The Oracle Service Bus Console logon page is displayed.

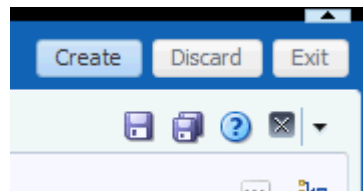
3. Log on to the Oracle Service Bus Console using a valid user name and password. The Oracle Service Bus Console home page is displayed, as shown in [Figure 7-1](#).

**Figure 7-1 Oracle Service Bus Console Home Page**



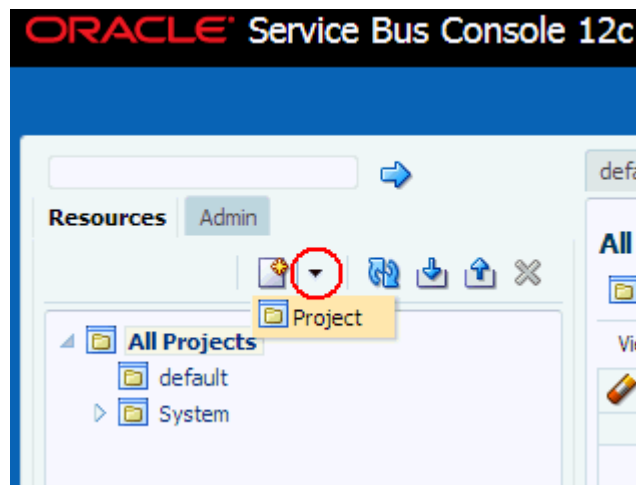
4. Click **Create** in the right pane of the Oracle Service Bus session, as shown in [Figure 7-2](#).

**Figure 7-2 Oracle Service Bus Session**



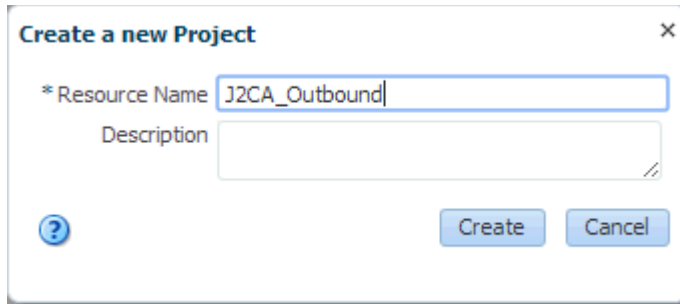
5. Select **All Projects**, click the down arrow in the left pane, and select **Project**, as shown in [Figure 7-3](#).

**Figure 7-3 All Projects Folder**



The Create a new Project window is displayed, as shown in [Figure 7-4](#).

**Figure 7-4 Create New Project Window**

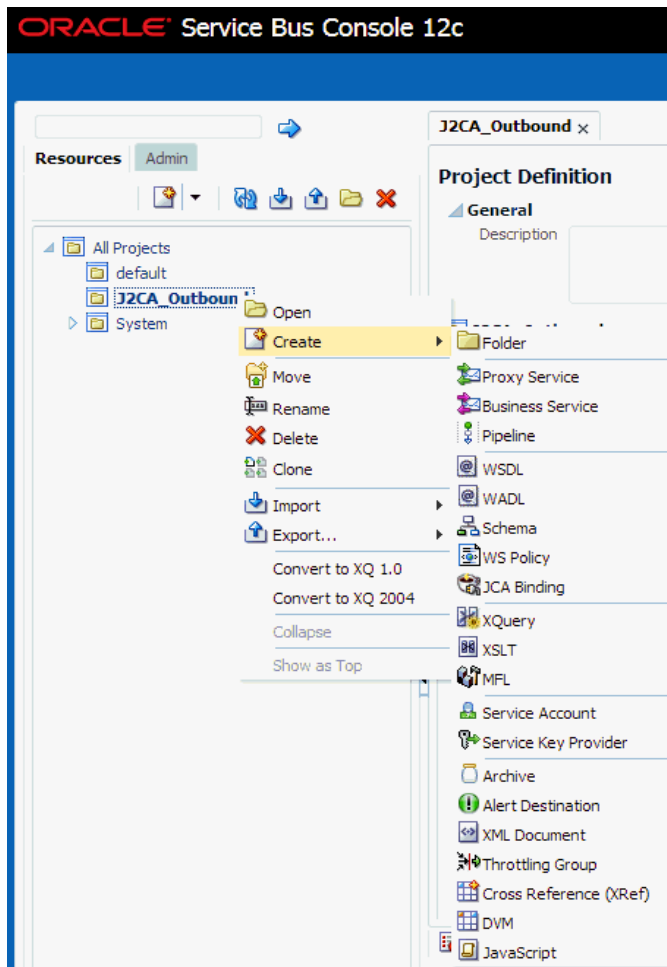


6. Provide a valid name for the new project (for example, J2CA\_Outbound) in the Resource Name field, and click **Create**.

The new project is successfully created and listed.

7. Right-click the newly created project, select **Create**, and click **Folder**, as shown in Figure 7-5.

**Figure 7-5 Create Option**



The Create a new Folder window is displayed.

8. In the Resource Name field, type **Business Service** and click **Create**.

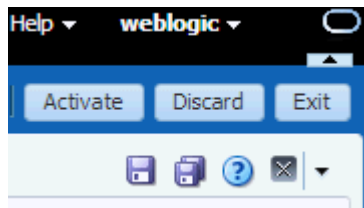
9. Repeat steps 7 and 8 to create folders with the names **Proxy Service** and **Wsdls**.  
The Business Service, Proxy Service, and Wsdls folders are listed in the left pane below the project node, as shown in [Figure 7-6](#).

**Figure 7-6 Project Node**



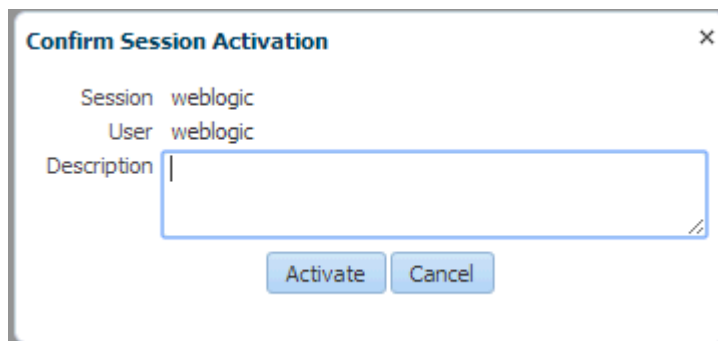
10. Click **Activate** in the right pane of the Oracle Service Bus session, as shown in [Figure 7-7](#).

**Figure 7-7 Activate Button**



11. In the Confirm Session Activation page, click **Activate** to save the changes, as shown in [Figure 7-8](#).

**Figure 7-8 Confirm Session Activation Window**



## 7.2.2 Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus

Before starting and using Application Explorer to publish a WSDL directly to the Oracle Service Bus (OSB) Console (project/folder), OSB users must perform the following steps:

1. Open the command prompt window.
2. Navigate to the following directory:  
`<ORACLE_HOME>\user_projects\domains\base_domain\bin`
3. Execute **setDomainEnv.cmd** (Windows) or **./setDomainEnv.sh** (UNIX/Linux).  
This command sets the class path for Application Explorer to access the Oracle WebLogic Server APIs to publish the WSDLs to the OSB Console.
4. Do not close the command prompt window.
5. Navigate to the following directory:  
`<ADAPTER_HOME>\tools\iwae\bin`
6. Execute **ae.bat** (Windows) or **iwae.sh** (UNIX/Linux) to start Application Explorer.  
You are now ready to publish WSDLs from Application Explorer to the OSB Console.

## 7.2.3 Publishing a WSDL From Application Explorer to Oracle Service Bus

Perform the following steps to publish a WSDL from Application Explorer to Oracle Service Bus:

1. Start Application Explorer, connect to a J2CA configuration, and connect to a PeopleSoft target.  
For more information, see [Chapter 2, "Configuring Oracle Application Adapter for PeopleSoft"](#) on page 2-1.
2. Expand the PeopleSoft target to which you are connected.
3. Expand **Component Interfaces** and then select **LOCATION**.
4. Navigate to and right-click an operation, and then select **Create Outbound JCA Service (Request/Response)**.  
The Export WSDL dialog is displayed, as shown in [Figure 7-9](#).

**Figure 7–9 Export WSDL Dialog**

5. In the Name field, a default file name for the WSDL file is provided. You can accept the default or provide your own.
6. Select the **Export to OSB** option.
7. In the Location field, enter the folder name in Oracle Service Bus where you want to publish the WSDL document.  
The location is composed of an Oracle Service Bus project name and optionally, one or more folder names. The project name and any folder names must be separated by a forward slash character “/”.
8. In the Host field, enter the name of the machine where Oracle Service Bus is installed.
9. In the Port field, enter the port that is being used by Oracle Service Bus.
10. In the User field, enter your username to access Oracle Service Bus.
11. In the Password field, enter your password to access Oracle Service Bus.
12. Click **OK**.

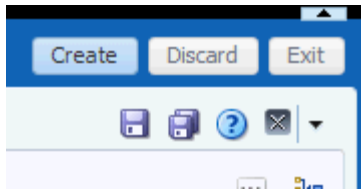
The WSDL is published to the location specified in the Export WSDL dialog and is now available for use with a Business Service or Proxy Service in Oracle Service Bus.

## 7.2.4 Configuring a WSDL-based Business Service

Perform the following steps to configure a WSDL-based Proxy Service:

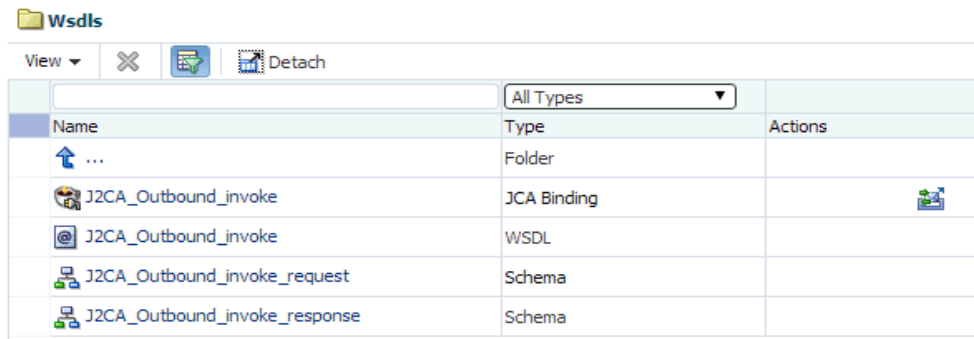
1. Open the Oracle Service Bus Console and click **Create** in the right pane of the Oracle Service Bus session, as shown in [Figure 7–10](#).

**Figure 7–10 Create Button**



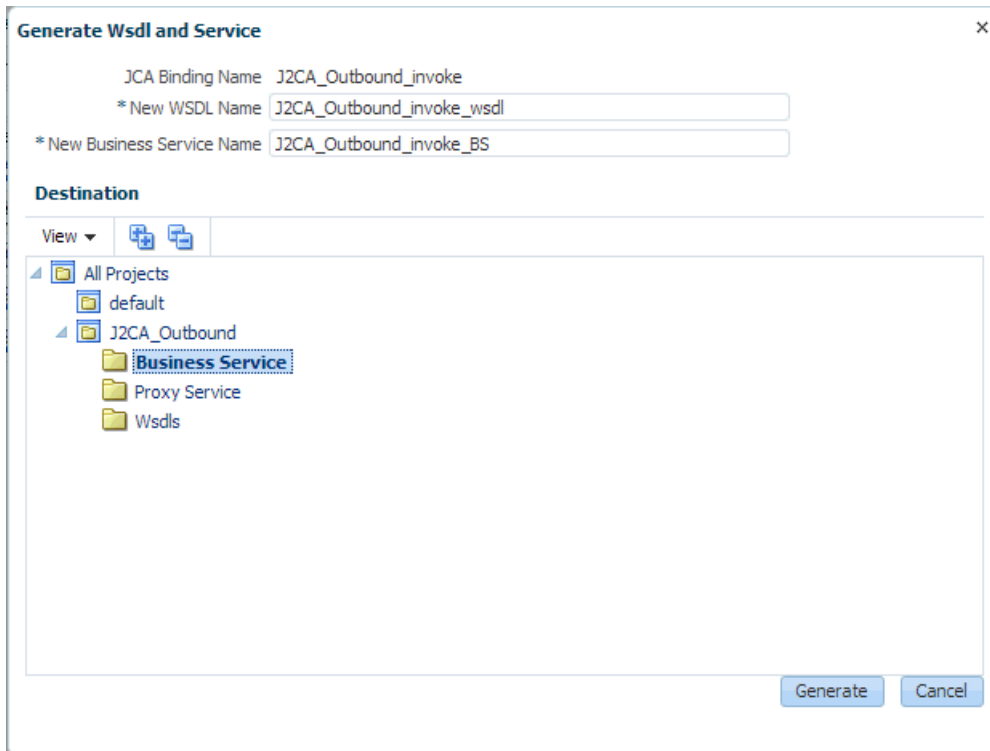
2. Double-click the created WSDL folder in the left pane (for example, Wsdls) and ensure that the exported WSDL is listed in the right pane, as shown in [Figure 7–11](#).

**Figure 7–11 Wsdls Folder**



3. Click the icon that corresponds to the JCA Binding in the Actions column.  
The Generate WSDL and Service window is displayed, as shown in [Figure 7–12](#).

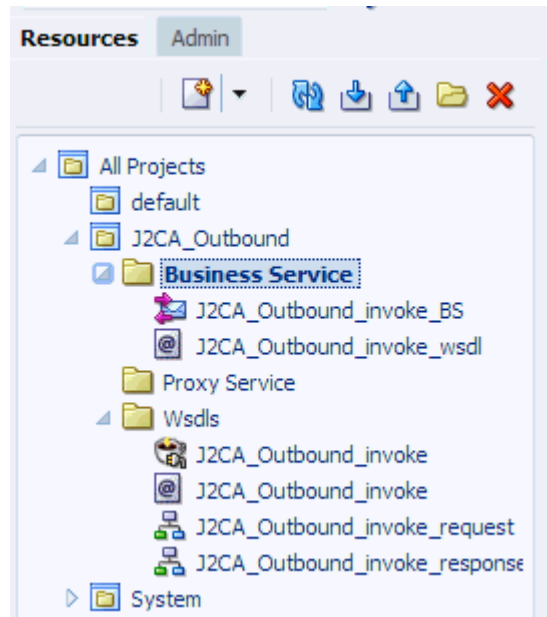
**Figure 7–12 Generate WSDL and Service Window**





4. Provide a new WSDL name and a new Business Service name in the corresponding fields.
5. In the Destination area, select an available project and the sub-folder that is designated for Business Services.
6. Click **Generate**.
7. Expand **Business Service** under the project folder and check if the generated WSDL and Business Service are listed, as shown in [Figure 7-13](#).

**Figure 7-13 Business Service Folder**

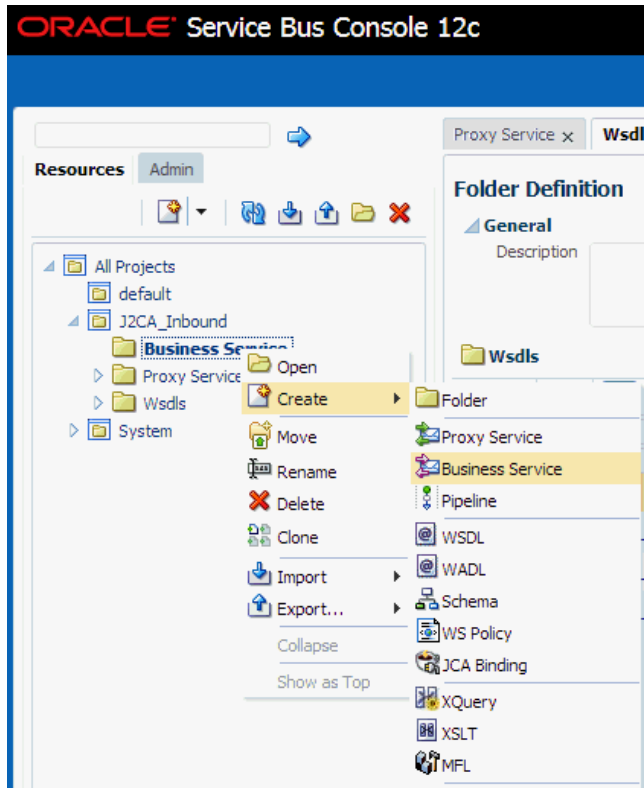


## 7.2.5 Configuring a File Type Business Service

Perform the following steps to configure a File type Business Service:

1. Right-click the **Business Service** folder you created in the left pane, select **Create**, and click **Business Service** as shown in [Figure 7-14](#).

Figure 7–14 Business Service Folder



The Create Business Service window is displayed.

2. In the Resource Name field, provide a name for the Business Service, select the **File** option in the Transport section under Service Definition, and click **Next**, as shown in Figure 7–15.

**Figure 7–15 Service Definition**

The screenshot shows the 'Create Business Service' dialog box with the 'Service Definition' page active. At the top, a progress bar shows 'Create' as the current step. Below it, the 'Create Service' section has a text field for '\*Resource Name' containing 'File\_Out' and an empty 'Description' field. The 'Service Definition' section has two radio buttons: 'WSDL Based Service' (selected) and 'Transport' (unselected). Under 'WSDL Based Service', there are fields for 'Name', 'Path', and 'Port/Binding'. Under 'Transport', there is a dropdown menu set to 'file'. At the bottom, there are four buttons: 'Back', 'Next', 'Create', and 'Cancel'.

3. In the Service Type section, select **Messaging Service**. By default, the Request Type is set to XML, and the Response Type is set to None. Then click **Next**, as shown in [Figure 7–16](#).

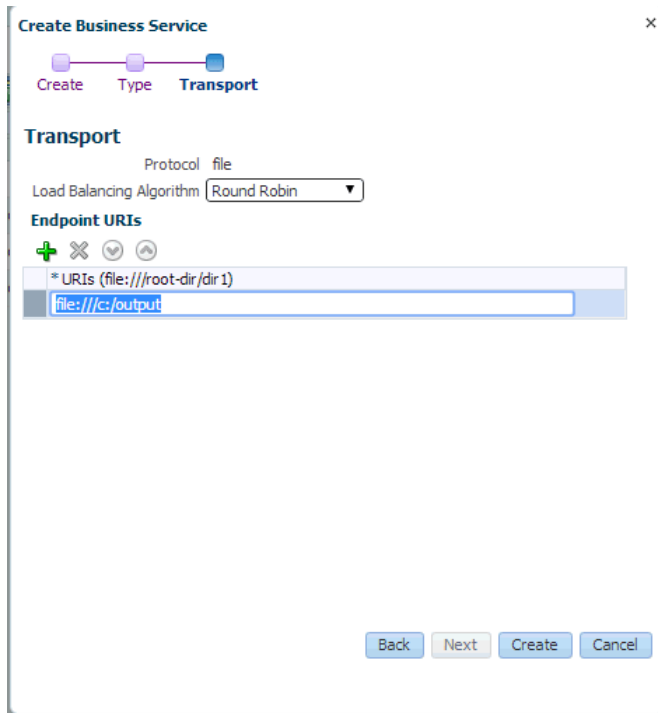
**Figure 7–16 Service Type Configuration Page**

The screenshot shows the 'Create Business Service' dialog box with the 'Service Type' page active. At the top, the progress bar shows 'Type' as the current step. Below it, the 'Service Type' section has four radio buttons: 'WSDL Based Service', 'Any SOAP Service', 'Any XML Service', and 'Messaging Service' (selected). Below the selected radio button, there are dropdown menus for 'Request Type' (set to 'XML') and 'Response Type' (set to 'None'). There are also text fields for 'Schema Name', 'Path', and 'Element/Type'. At the bottom, there are four buttons: 'Back', 'Next', 'Create', and 'Cancel'.

4. Enter the path to a destination folder on your file system in the Endpoint URI field.

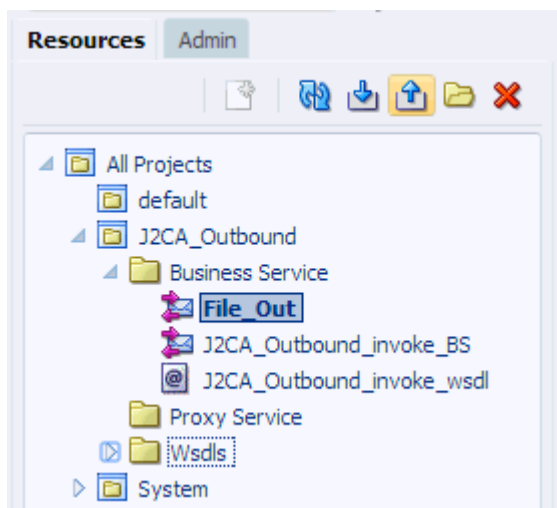
- Click **Create**, as shown in [Figure 7-17](#).

**Figure 7-17 Transport Page**



The Business Service **File\_Out** is created and listed under Business Service, as shown in [Figure 7-18](#).

**Figure 7-18 File\_Out Business Service**



- Double-click **File\_Out**, click **Transport Detail** in the left pane, and enter the prefix and suffix for the output file to be received, as shown in [Figure 7-19](#).

**Figure 7–19 Transport Detail**

The screenshot shows the 'Business Service Definition' configuration page. The 'Configuration' tab is selected, and the 'Transport Detail' sub-tab is active. The left sidebar contains a menu with options: General, Transport, **Transport Detail** (highlighted), Message Handling, and Performance. The main content area is titled 'Transport Detail' and contains three input fields: 'Prefix' with the value 'outbound', 'Suffix' with the value '.xml', and 'Request encoding' with the value 'utf-8'.

7. Click the **Save** or **Save All** icon in the right corner, as shown in [Figure 7–20](#).

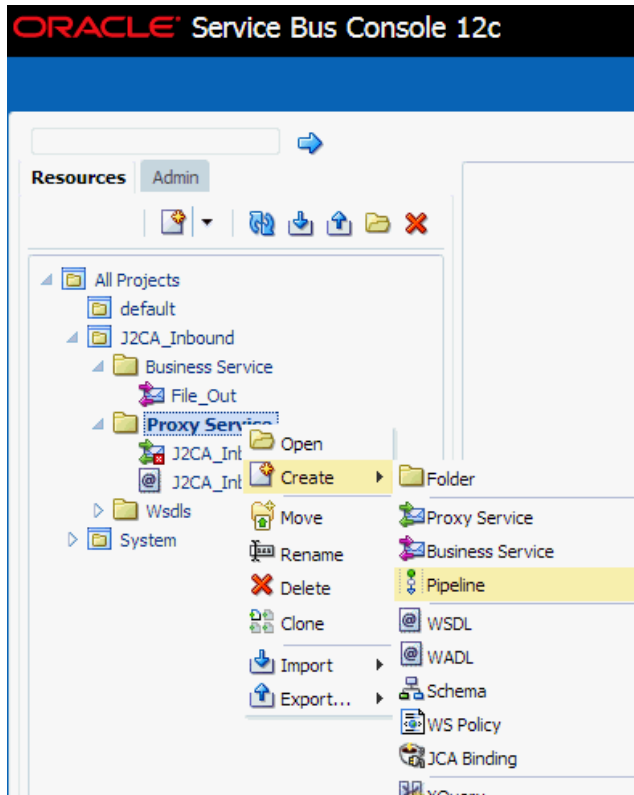
**Figure 7–20 Save/Save All Icons**

## 7.2.6 Configuring a Pipeline With Proxy Service

Perform the following steps to configure a Pipeline:

1. Right-click the Proxy Service folder, select **Create** and click **Pipeline**, as shown in [Figure 7–21](#).

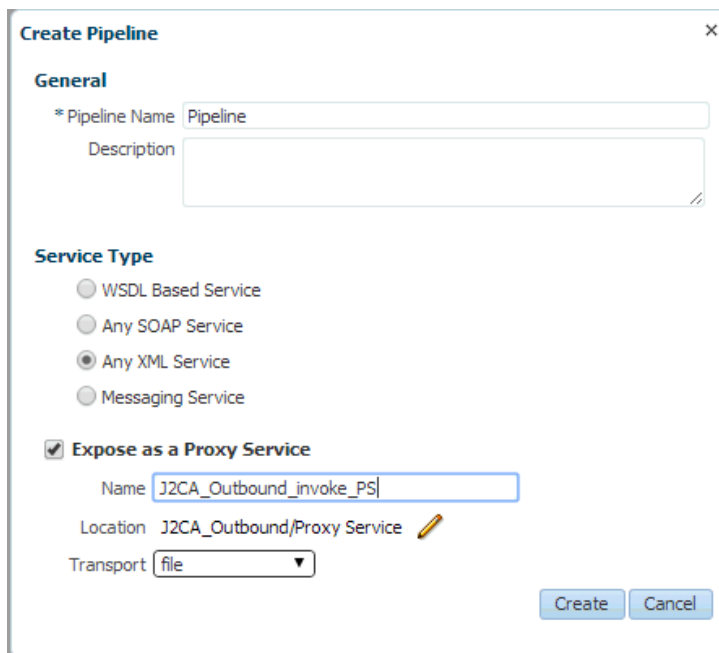
**Figure 7–21 Pipeline Option**



The Create Pipeline window is displayed.

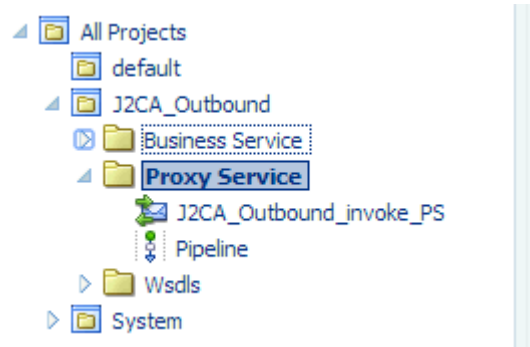
2. Enter a name in the Pipeline Name field. By default, **Expose as a Proxy Service** is selected. If you wish to change the Proxy Service Name, change it and set Transport as **file**, and click **Create** as shown in [Figure 7–22](#).

**Figure 7–22 Create Pipeline Window**



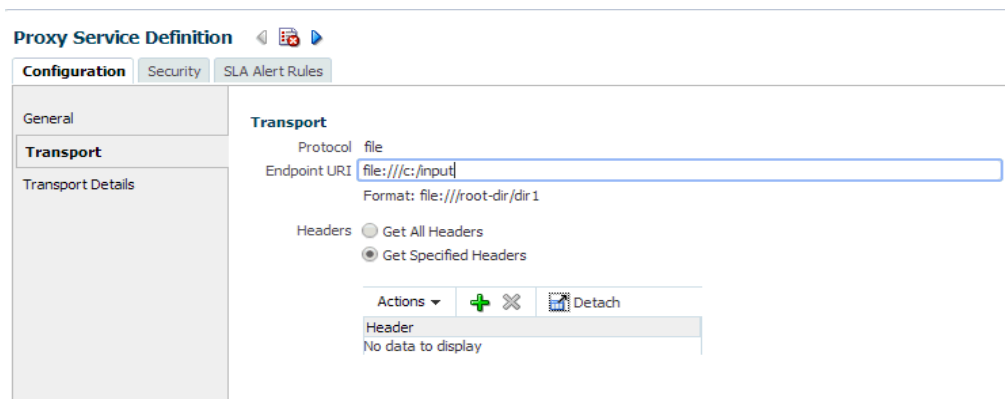
The created Pipeline and the Proxy Service is listed under Proxy Service, as shown in [Figure 7-23](#).

**Figure 7-23 Pipeline Node**



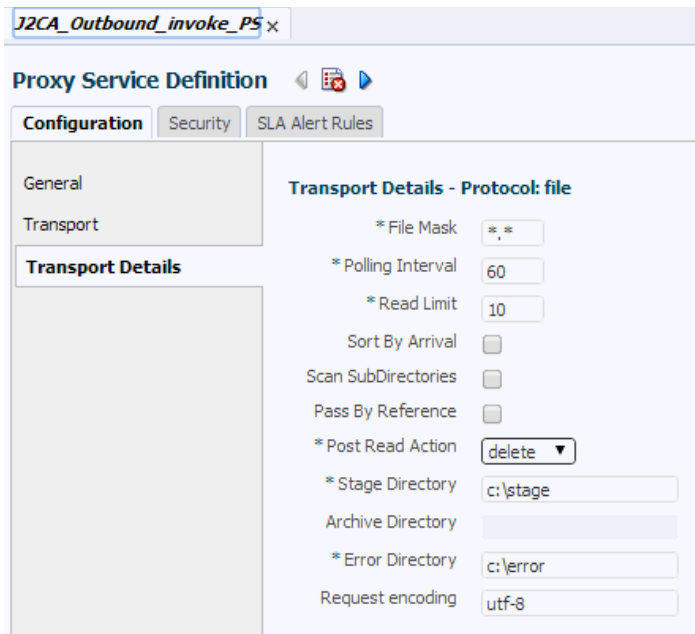
3. Double-click the created proxy service and click **Transport** in the left pane. Provide the input location in the Endpoint URI field, as shown in [Figure 7-24](#).

**Figure 7-24 Transport**



4. Click **Transport Details** in the left pane and provide the location for the Stage Directory and the Error Directory fields, as shown in [Figure 7-25](#).

**Figure 7-25 Transport Details**



5. Click the **Save All** icon in the right corner, as shown in [Figure 7-26](#).

**Figure 7-26 Save All Icon**



6. Double-click the **Pipeline** node and click the **Open Message Flow** icon on the right pane to open the message flow, as shown in [Figure 7-27](#).

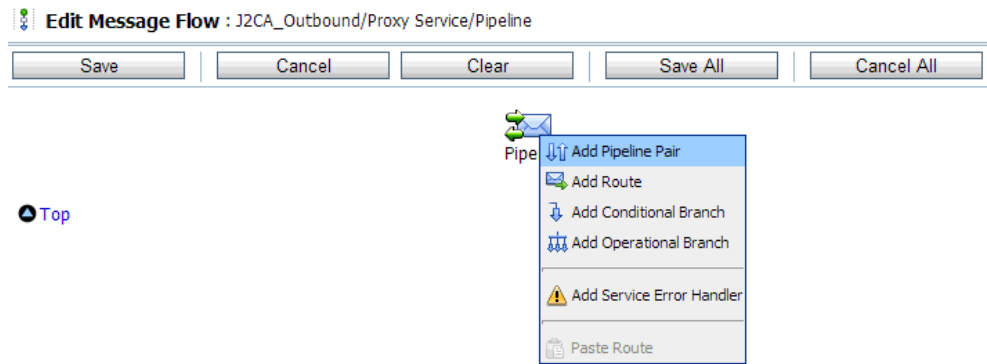
**Figure 7-27 Open Message Flow Icon**



7. Click the Proxy Service icon and select **Add Pipeline Pair** from the menu, as shown in [Figure 7-28](#).

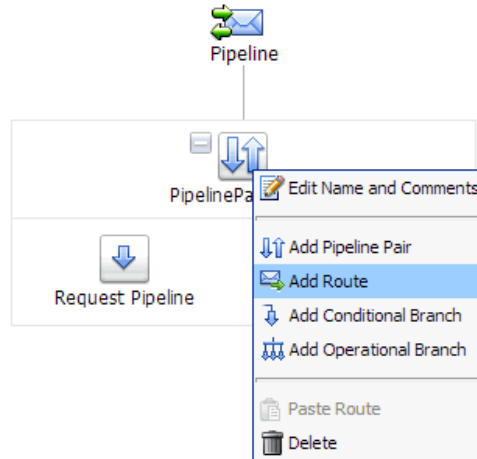


**Figure 7–28 Add Pipeline Pair Option**



8. Click the **PipelinePairNode1** icon and select **Add Route** from the menu, as shown in [Figure 7–29](#).

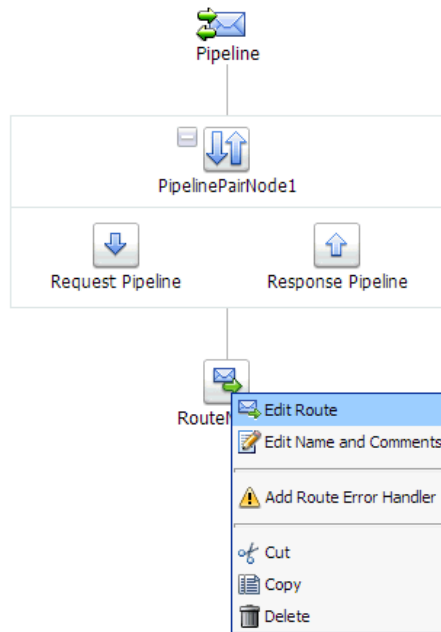
**Figure 7–29 Add Route Option**



The RouteNode1 icon is added below the PipelinePairNode1 icon.

9. Click the RouteNode1 icon and select **Edit Route** from the menu, as shown in [Figure 7–30](#).

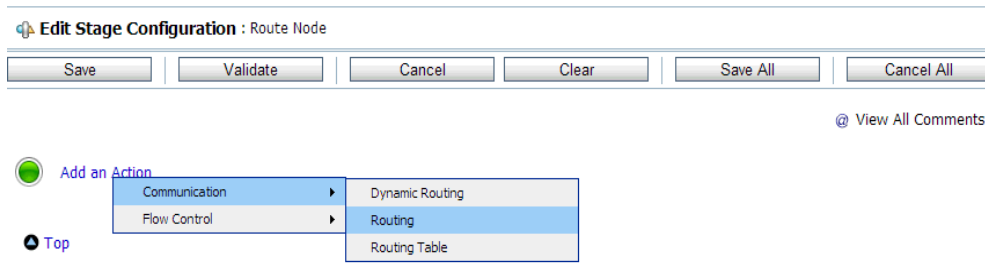
**Figure 7–30 Edit Route Option**



The Edit Stage Configuration workspace area is displayed.

10. Click **Add an Action**, select **Communication** and click **Routing**, as shown in Figure 7–31.

**Figure 7–31 Edit Stage Configuration Workspace Area**



11. Click **<Service>**, as shown in Figure 7–32.

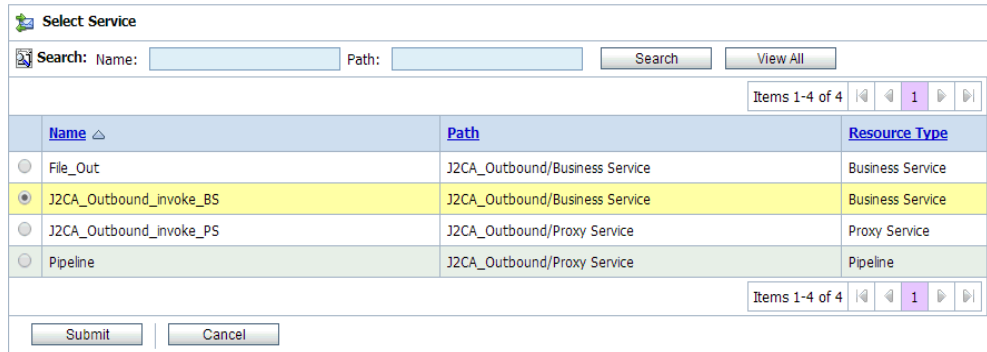
**Figure 7–32 Actions**



The Select Service dialog is displayed.

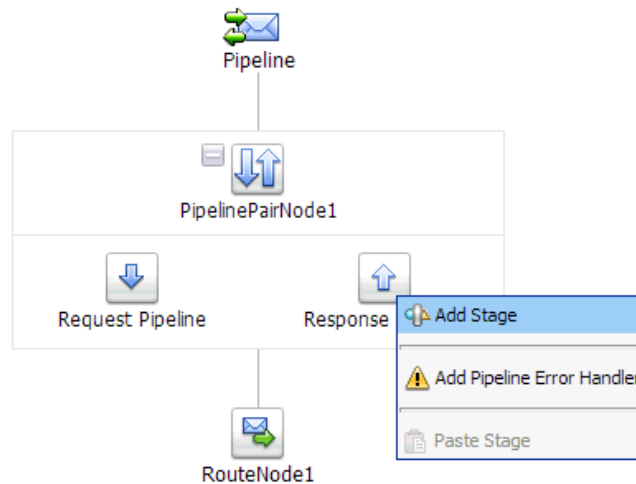
12. Select the WSDL type Business Service configured for PeopleSoft and click on **Submit**, as shown in Figure 7–33.

**Figure 7–33 Select Service Dialog**



13. Select the name of the PeopleSoft business object as the operational attribute from the list, and click **Save**.
14. Click the Response Pipeline icon and select **Add Stage** from the menu, as shown in Figure 7–34.

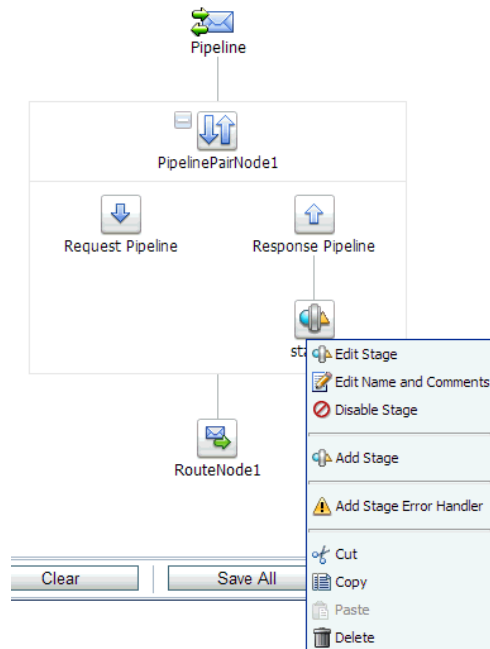
**Figure 7–34 Response Pipeline Icon**



The Stage1 icon is added below the Response Pipeline icon.

15. Click the Stage1 icon and select **Edit Stage** from the menu, as shown in Figure 7–35.

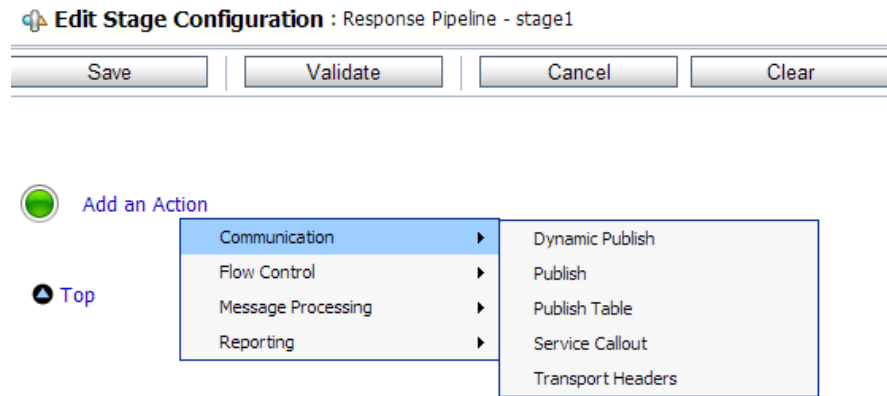
**Figure 7–35 Edit Stage Option**



The Edit Stage Configuration workspace area is displayed.

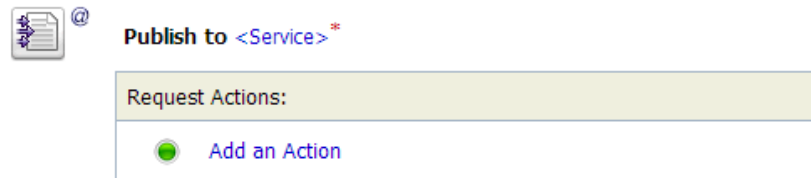
16. Click **Add an Action**, select **Communication**, and then click **Publish**, as shown in [Figure 7–36](#).

**Figure 7–36 Edit Stage Configuration Workspace Area**



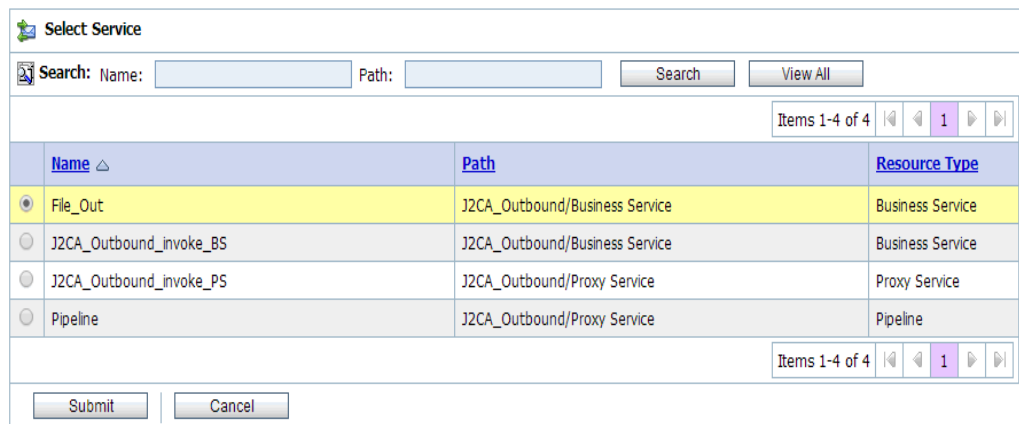
17. Click **<Service>**, as shown in [Figure 7–37](#).

**Figure 7–37 <Service> Action**



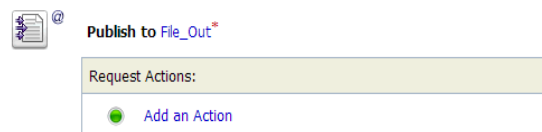
18. In the Select Service dialog, select a File type Business Service and click **Submit**, as shown in [Figure 7–38](#).

**Figure 7–38 Select Service Dialog**



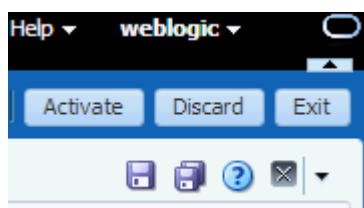
19. Click **Save All**, as shown in [Figure 7–39](#).

**Figure 7–39 Save All Button**



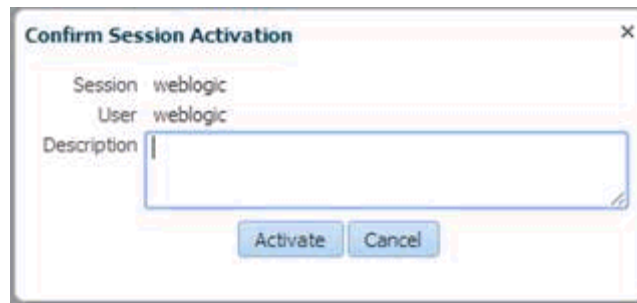
20. Click **Activate** in the right pane of the Oracle Service Bus session, as shown in [Figure 7–40](#).

**Figure 7–40 Activate Button**



21. Click **Activate** to save the changes, as shown in [Figure 7–41](#).

**Figure 7–41 Confirm Session Activation**



22. Copy and paste an input XML file in the input folder you have configured (for example, C:\input). Output is received in the configured output location (for example, C:\output).

## 7.3 Configuring an Inbound Process Using sbconsole (J2CA Configuration)

This section describes how to configure an inbound process using sbconsole for J2CA configurations.

A sample project has been provided for this inbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\OSB\J2CA\PeopleSoft_Sample_J2CA_OSB_Inbound_Project
```

This section includes the following topics:

- [Section 7.3.1, "Starting Oracle Service Bus and Creating Project Folders"](#)
- [Section 7.3.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"](#)
- [Section 7.3.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"](#)
- [Section 7.3.4, "Configuring a WSDL-based Proxy Service"](#)
- [Section 7.3.5, "Configuring a File Type Business Service"](#)
- [Section 7.3.6, "Configuring a Pipeline"](#)

### 7.3.1 Starting Oracle Service Bus and Creating Project Folders

For more information on starting Oracle Service Bus and creating project folders, see [Section 7.2.1, "Starting Oracle Service Bus and Creating Project Folders"](#) on page 7-2.

### 7.3.2 Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus

For more information on setting the class path for Application Explorer to integrate with Oracle Service Bus, see [Section 7.2.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"](#) on page 7-6.

### 7.3.3 Publishing a WSDL From Application Explorer to Oracle Service Bus

Perform the following steps to publish a WSDL from Application Explorer to Oracle Service Bus:

1. Start Application Explorer, connect to a J2CA configuration, and connect to a PeopleSoft target.

For more information, see [Chapter 2, "Configuring Oracle Application Adapter for PeopleSoft"](#) on page 2-1.

2. Create a PeopleSoft channel.

For more information, see [Section 4.5.1.1, "Creating a Channel"](#) on page 4-35.

3. Under the connected target, expand **Messages**, and then select **LOCATION\_SYNC.VERSION\_1**.

4. Right-click **LOCATION\_SYNC.VERSION\_1** and select **Create Inbound JCA Service(Event)** from the context menu.

The Export WSDL dialog is displayed, as shown in [Figure 7-42](#).

**Figure 7-42** Export WSDL Dialog

5. In the Name field, a default file name for the WSDL file is provided. You can accept the default or provide your own.
6. From the Channel list, select the channel you created for this inbound service.
7. Three check boxes for Root, Namespace, and Schema validation are also available. Selection of multiple validation options is allowed.
8. Select the **Export to OSB** option.

9. In the Location field, enter the folder name in Oracle Service Bus where you want to publish the WSDL document.  
The location is composed of an Oracle Service Bus project name and optionally, one or more folder names. The project name and any folder names must be separated by a forward slash character “/”.
10. In the Host field, enter the name of the machine where Oracle Service Bus is installed.
11. In the Port field, enter the port that is being used by Oracle Service Bus.
12. In the User field, enter your username to access Oracle Service Bus.
13. In the Password field, enter your password to access Oracle Service Bus.
14. Click **OK**.

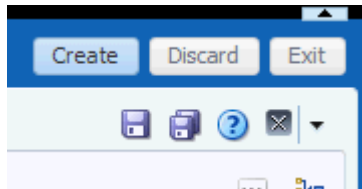
The inbound WSDL is published to the location specified in the Export WSDL dialog and is now available for use with a Proxy Service in Oracle Service Bus.

### 7.3.4 Configuring a WSDL-based Proxy Service

Perform the following steps to configure a WSDL-based Proxy Service:

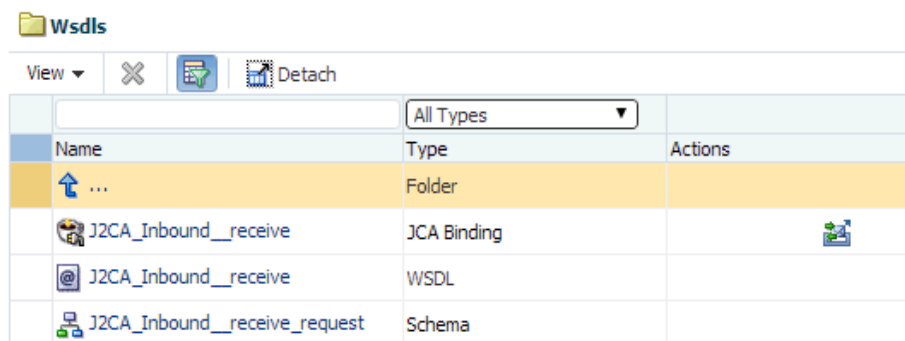
1. Open the Oracle Service Bus Console and click **Create** in the right pane of the Oracle Service Bus session, as shown in [Figure 7-43](#).

**Figure 7-43 Create Button**



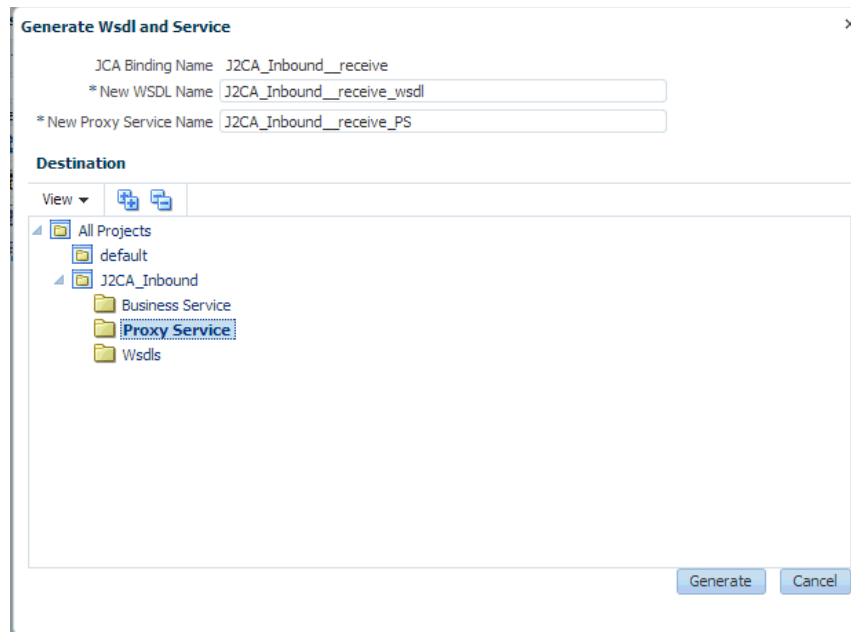
2. Double-click the created WSDL folder in the left pane (for example, Wsdls), and ensure that the exported WSDL is listed in the right pane, as shown in [Figure 7-44](#).

**Figure 7-44 Exported WSDL**

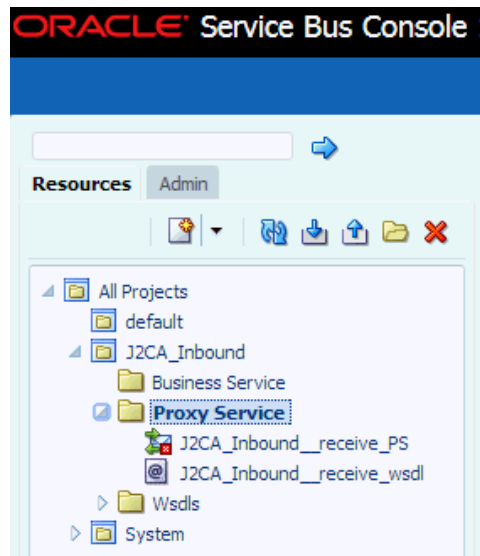


3. Click the icon that corresponds to the JCA Binding in the Actions column.  
The Generate WSDL and Service page is displayed, as shown in [Figure 7-45](#).



**Figure 7–45 Generate WSDL and Service Page**

4. Provide a new WSDL name and a new Proxy Service name in the corresponding fields.
5. In the Destination area, select an available project and the sub-folder that is designated for Proxy Services.
6. Click **Generate**.
7. Expand **Proxy Service** under Project Explorer and check if the generated WSDL and Proxy Service are listed, as shown in [Figure 7–46](#).

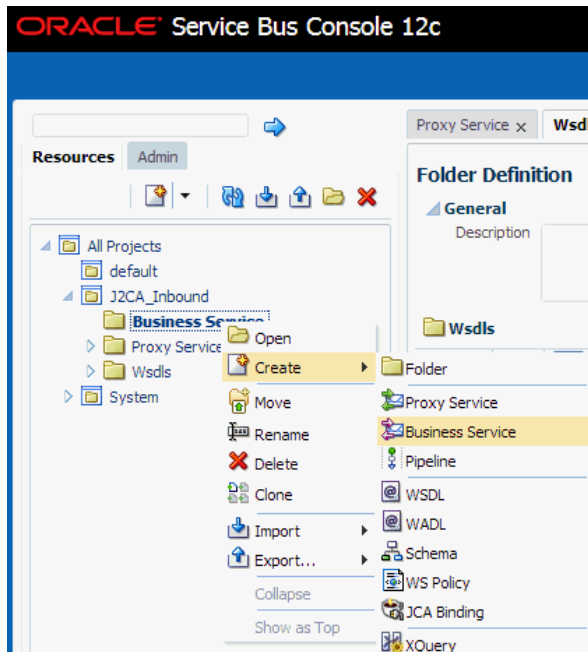
**Figure 7–46 Generated WSDL**

### 7.3.5 Configuring a File Type Business Service

Perform the following steps to configure a File type Business Service:

1. Right-click the Business Service folder you created in the left pane, select **Create**, and click **Business Service**, as shown in [Figure 7-47](#).

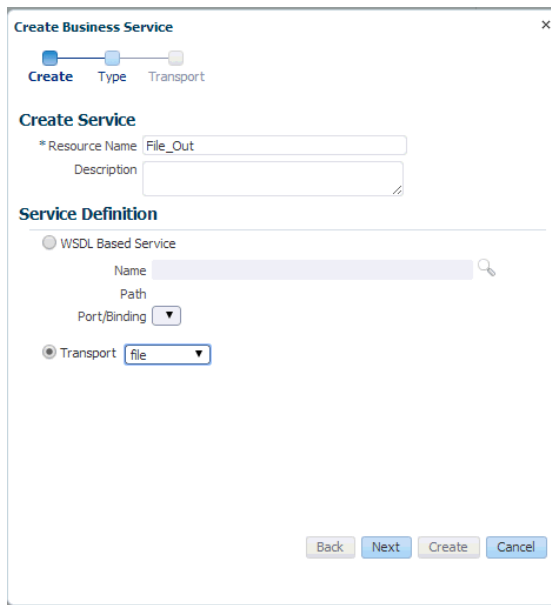
**Figure 7-47 Business Service Folder**



The Create Business Service window is displayed.

2. In the Resource Name field, provide a name for the Business Service and select the **File** option from the Transport drop-down list in the Service Definition area, as shown in [Figure 7-48](#).

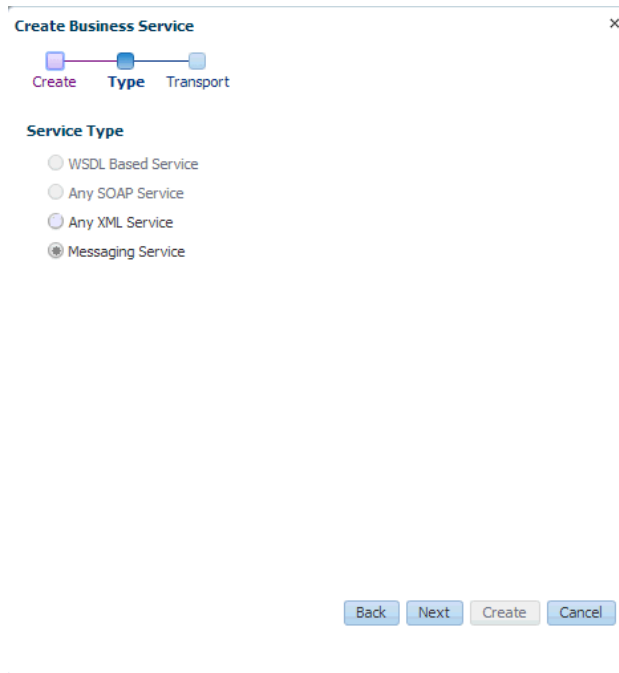
**Figure 7-48 Create Business Service Window**



3. Click **Next**.

- In the Service Type area, select **Messaging Service** as the service type, as shown in [Figure 7-49](#).

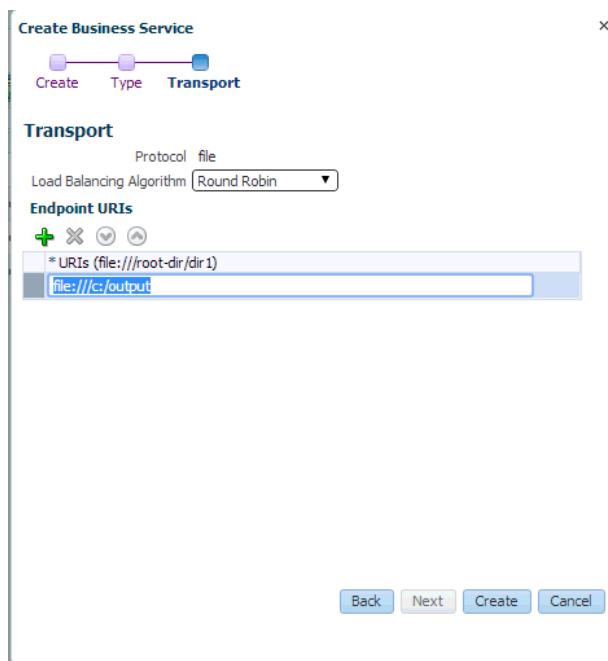
**Figure 7-49 Service Type Area**



- Click **Next**.

The Transport page is displayed, as shown in [Figure 7-50](#).

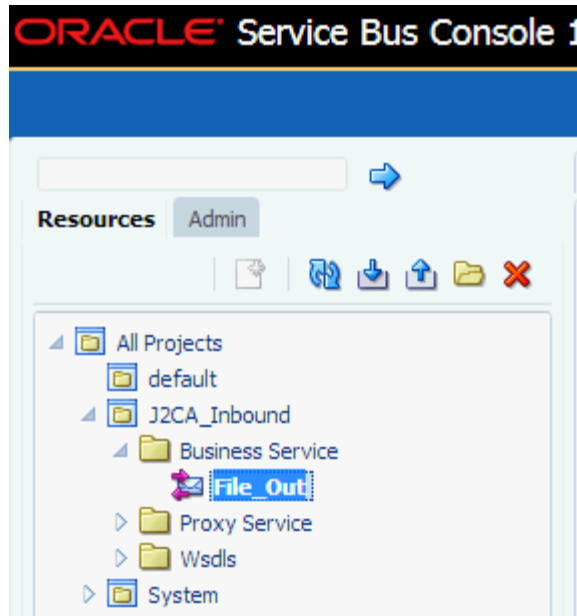
**Figure 7-50 Transport Page**



6. Enter the path to a destination folder on your file system in the Endpoint URI field and click **Create**.

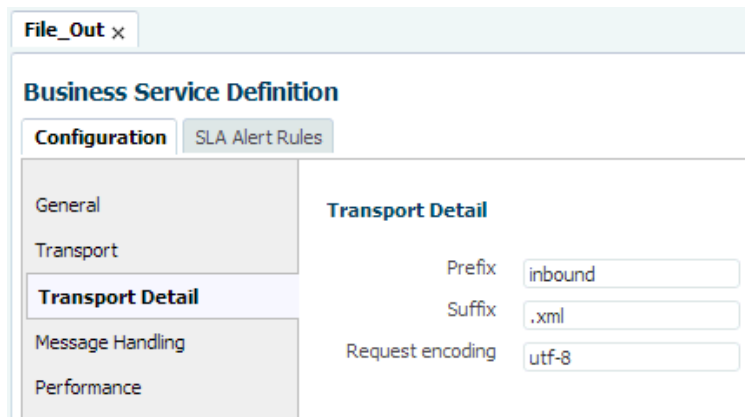
The Business Service File\_Out is created and listed under Business Service, as shown in [Figure 7-51](#).

**Figure 7-51 File\_Out Business Service**



7. Double-click **File\_Out**, click **Transport Detail** in the left pane, and enter the prefix and suffix for the output file to be received, as shown in [Figure 7-52](#).

**Figure 7-52 Transport Detail Page**



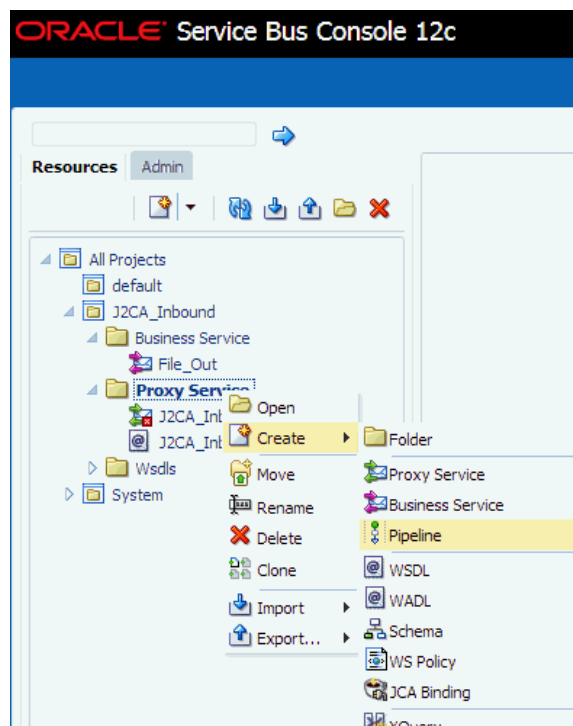
8. Click the Save or Save All icon in the right corner, as shown in [Figure 7-53](#).

**Figure 7–53 Save Icons**

### 7.3.6 Configuring a Pipeline

Perform the following steps to configure a Pipeline:

1. Right-click the proxy service you created and select **Create**, and then click **Pipeline**, as shown in [Figure 7–54](#).

**Figure 7–54 Pipeline Option**

The Create Pipeline window is displayed.

2. In the Pipeline Name field, enter a name and select the Service Type as **WSDL Based Service**, as shown in [Figure 7–55](#).

**Figure 7–55 Create Pipeline Window**

3. Click the Search icon, and in the displayed Search and Select: WSDL Resource window, select **J2CA\_Inbound\_receive\_wsdl**, and click **OK**, as shown in [Figure 7–56](#).

**Figure 7–56 Search and Select: WSDL Resource Window**

Name	Path	Namespace
J2CA_Inbound__receive_wsdl	J2CA_Inbound/Proxy Service	http://xmlns.oracle...
J2CA_Inbound__receive	J2CA_Inbound/Wsdl	http://xmlns.oracle...

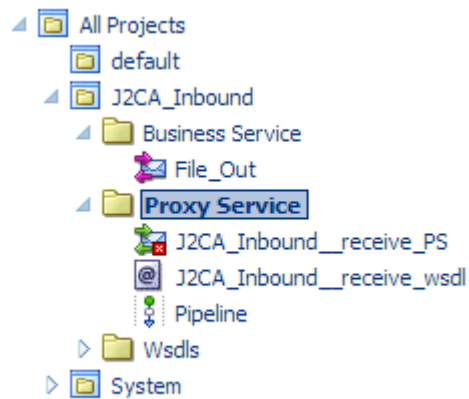
OK Cancel

The Create Pipeline window opens.

4. Clear the check box for **Expose as a Proxy Service**, and click **Create**, as shown in [Figure 7–57](#).

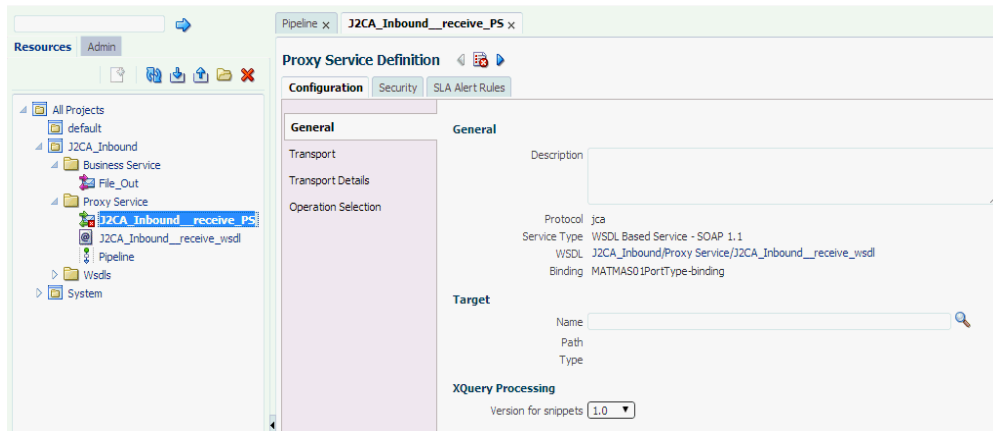
**Figure 7–57 Create Pipeline Window**

The pipeline is created and listed under Proxy Service, as shown in [Figure 7–58](#).

**Figure 7–58 Proxy Service Pipeline**

5. Double-click the **J2CA\_Inbound\_receive\_PS** node under Proxy Service in the left pane and click the **Search** icon in the Target area in right pane, as shown in [Figure 7–59](#).

**Figure 7–59 Proxy Service Definition Window**

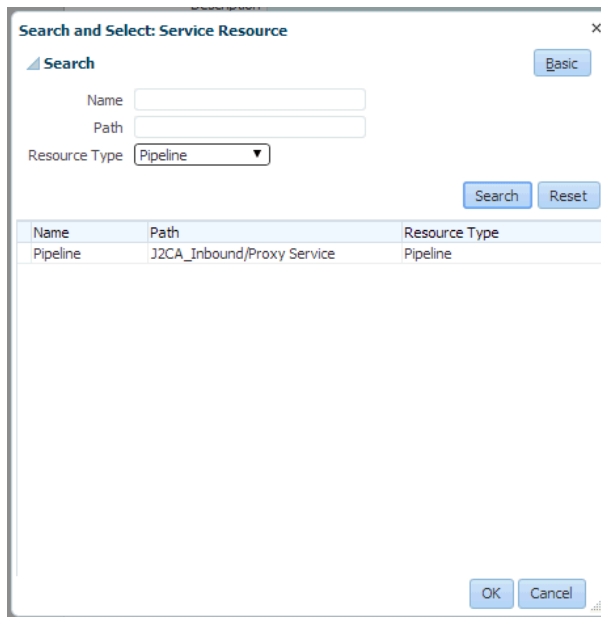


The Search and Select: Service Resource window appears.

6. From the Resource Type drop-down list, select **Pipeline** and then click the **Search** button.

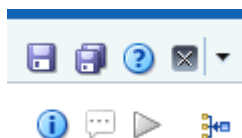
The Pipeline is listed, as shown in [Figure 7–60](#).

**Figure 7–60 Search and Select: Service Resource Window**



7. Select the Pipeline and click **OK**.
8. Click the Save or Save All icon in the right corner, as shown in [Figure 7–61](#).

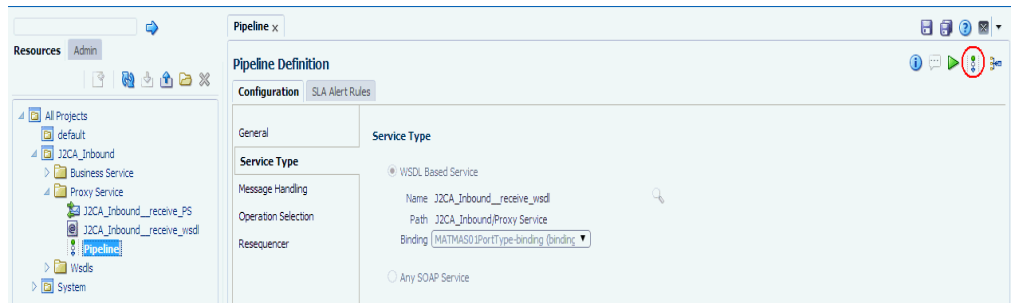
**Figure 7–61 Save and Save All Icons**





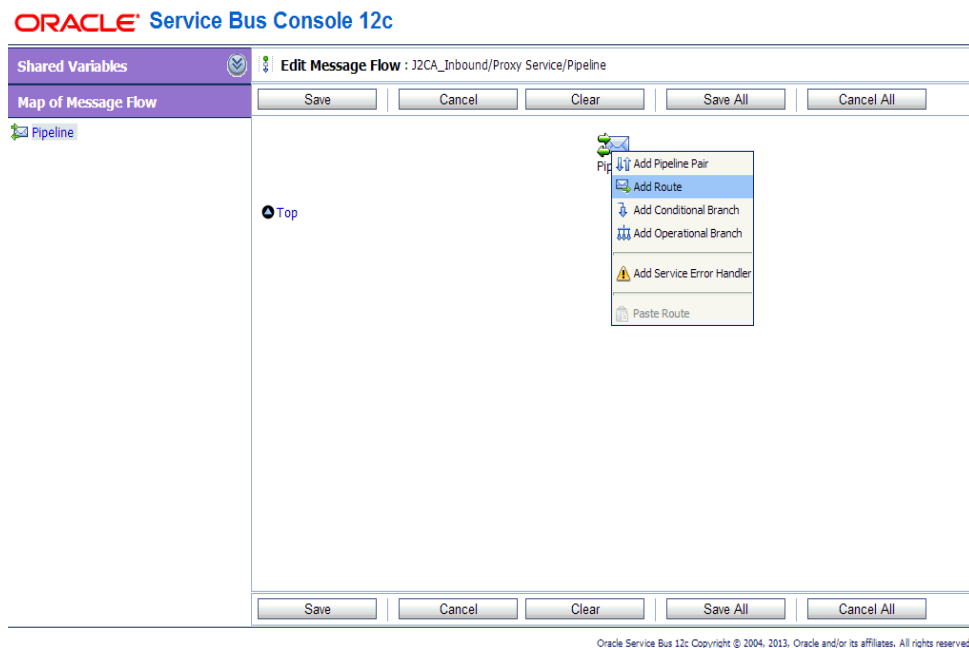
- In the left pane, double-click **Pipeline** under the Proxy Service folder and click the down-pointing icon on the right pane to open the message flow, as shown in Figure 7-62.

**Figure 7-62 Message Flow**



- Click the displayed Proxy service icon and select **Add Route** from the menu, as shown in Figure 7-63.

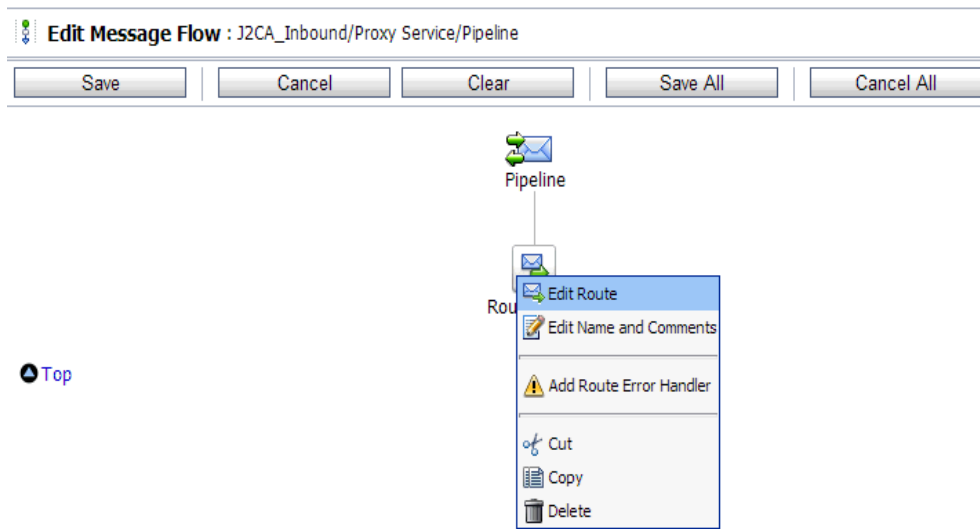
**Figure 7-63 Add Route Option**



The RouteNode1 icon is added.

- Click the RouteNode1 icon and select **Edit Route** from the menu, as shown in Figure 7-64.

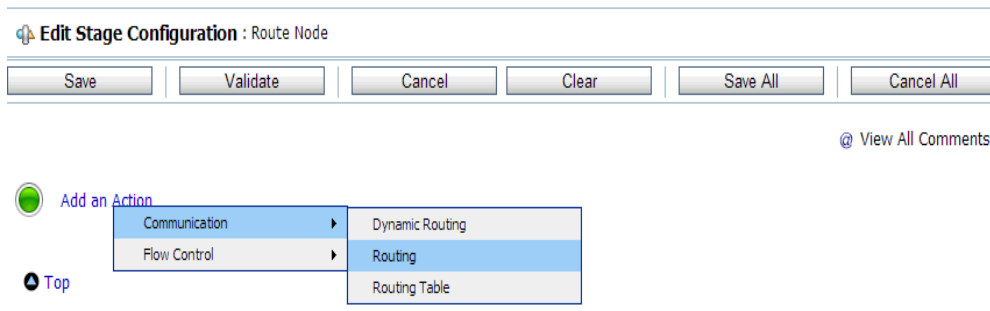
**Figure 7–64 Edit Route Option**



The Edit Stage Configuration workspace area is displayed.

12. Click **Add an Action**, select **Communication** from the menu, and then click **Routing**, as shown in Figure 7–65.

**Figure 7–65 Edit Stage Configuration Workspace**



13. Click **<Service>**, as shown in Figure 7–66.

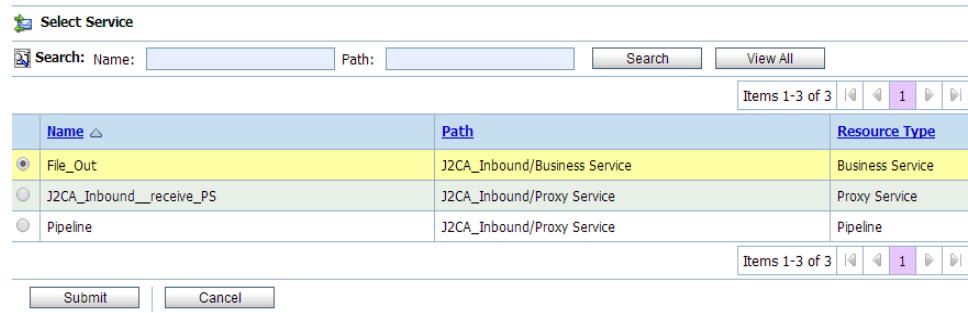
**Figure 7–66 Service Route Actions**



The Select Service dialog is displayed.

14. Select the **File\_Out** Business service and click **Submit** as shown in [Figure 7-67](#).

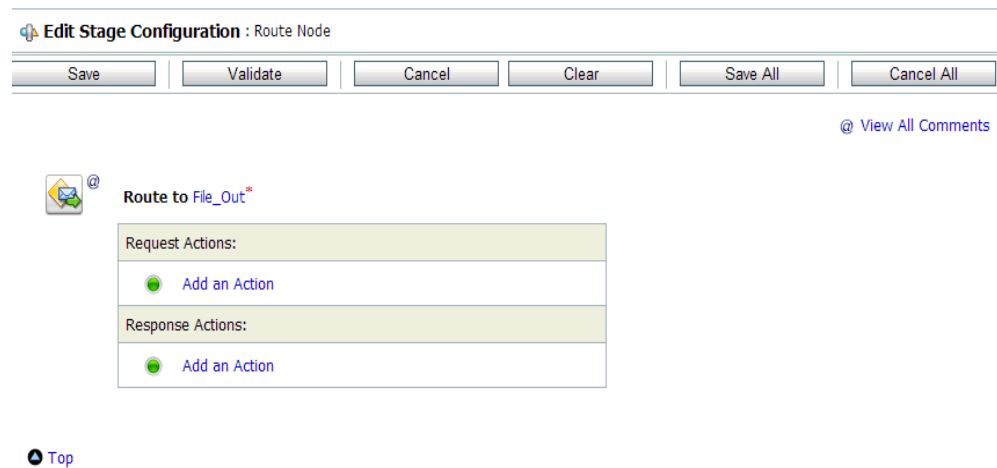
**Figure 7-67 Select Service Dialog**



You are returned to the Edit Stage Configuration workspace area.

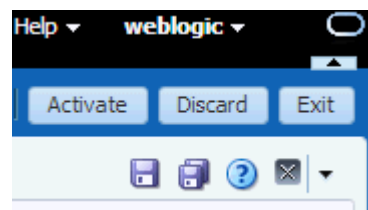
15. Click **Save All**, as shown in [Figure 7-68](#).

**Figure 7-68 Edit Stage Configuration Workspace Area**



16. Click **Activate** in the right pane of the Oracle Service Bus session, as shown in [Figure 7-69](#).

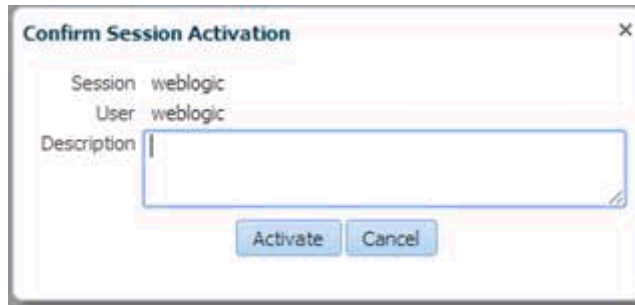
**Figure 7-69 Activate Button**



The Confirm Session Activation window appears.

17. Click **Activate** to save the changes, as shown in [Figure 7-70](#).

**Figure 7–70 Confirm Session Activation Window**



18. Trigger an event from the PeopleSoft system and check if the output is received in the configured output location.

For more information on triggering an event, see [Section 4.5.5, "Triggering an Event in PeopleSoft"](#) on page 4-45.

## 7.4 Configuring an Outbound Process Using sbconsole (BSE Configuration)

This section describes how to configure an outbound process using sbconsole for BSE configurations.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\OSB\BSE\PeopleSoft_Sample_BSE_OSB_Outbound_Project
```

This section includes the following topics:

- [Section 7.4.1, "Starting Oracle Service Bus and Creating Project Folders"](#)
- [Section 7.4.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"](#)
- [Section 7.4.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"](#)
- [Section 7.4.4, "Configuring a File Type Business Service"](#)
- [Section 7.4.5, "Configuring a WSDL-based Business Service"](#)
- [Section 7.4.6, "Configuring a Pipeline With Proxy Service"](#)

### 7.4.1 Starting Oracle Service Bus and Creating Project Folders

For more information on starting Oracle Service Bus and creating project folders, see [Section 7.2.1, "Starting Oracle Service Bus and Creating Project Folders"](#) on page 7-2.

### 7.4.2 Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus

For more information on setting the class path for Application Explorer to integrate with Oracle Service Bus, see [Section 7.2.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"](#) on page 7-6.

### 7.4.3 Publishing a WSDL From Application Explorer to Oracle Service Bus

This section describes how to publish a WSDL from Application Explorer (BSE configuration) to Oracle Service Bus.

1. Start Application Explorer, connect to a BSE configuration, and connect to a PeopleSoft target.
2. Expand **Component Interfaces** and select **Location**.
3. Navigate to and right-click an operation, and then select **Create Web Service** from the menu.

The Create Web Service dialog is displayed, as shown in [Figure 7-71](#).

**Figure 7-71 Create Web Service Dialog**

4. Enter a service name and click **Next**.
5. Click **OK** on the next dialog that is displayed.

Application Explorer switches the view to the Business Services node, and the new Web service appears in the left pane.

6. Right-click the new Web service and select **Export WSDL** from the menu.

The Export WSDL dialog is displayed, as shown in [Figure 7-72](#).

**Figure 7-72 Export WSDL Dialog**

7. In the Name field, a default file name for the WSDL file is provided. You can accept the default or provide your own.

8. In the Location field, enter the location where you want to publish the WSDL document.  
The location is composed of an Oracle Service Bus project name and optionally, one or more folder names. The project name and any folder names must be separated by a forward slash character “/”.
9. In the Host field, enter the name of the machine where Oracle WebLogic Server is running.
10. In the Port field, enter the port for the domain you are using.
11. In the User field, enter your username to access Oracle Service Bus.
12. In the Password field, enter your password to access Oracle Service Bus.
13. Click **OK**.

The WSDL is published to the location specified in the Export WSDL dialog and is now available for use with a Business Service or Proxy Service in Oracle Service Bus.

#### 7.4.4 Configuring a File Type Business Service

For more information on configuring a file type business service, see [Section 7.2.5, "Configuring a File Type Business Service"](#) on page 7-9.

#### 7.4.5 Configuring a WSDL-based Business Service

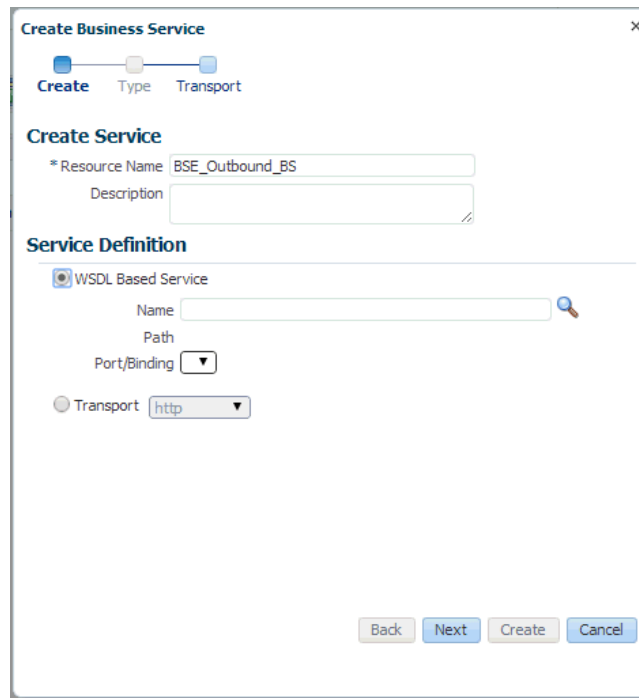
This section describes how to configure a WSDL type Business Service using the Oracle Service Bus Console.

Perform the following steps to configure a WSDL-based Proxy Service:

1. Right-click on the Business Service folder in the left pane and select **Business Service**.

The Create Business Service window is displayed, as shown in [Figure 7-73](#).

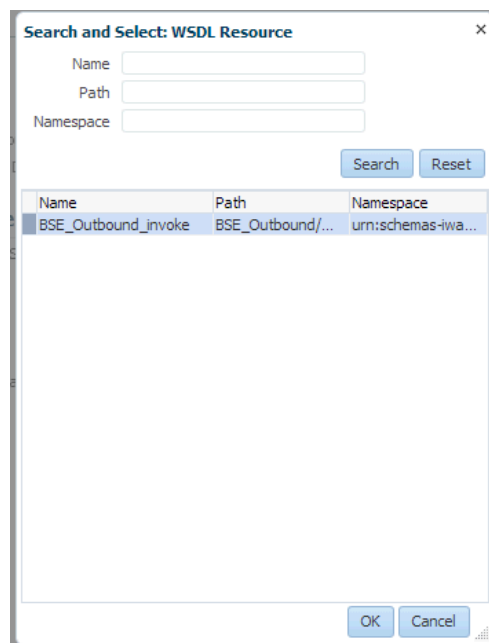
**Figure 7–73 Create Business Service Window**



2. Provide a name for the Business Service, and in Service Definition area, select the WSDL Based Service option and click the search icon.

The Search and Select: WSDL Resource window is displayed, as shown in [Figure 7–74](#).

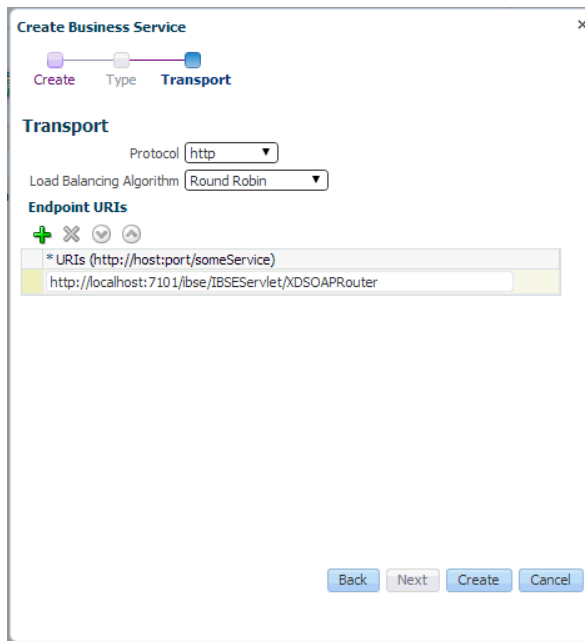
**Figure 7–74 Search and Select: WSDL Resource Window**



3. Click the **Search** button, select the BSE Outbound WSDL, and click **OK**.  
You are returned to the Create Business Service window.

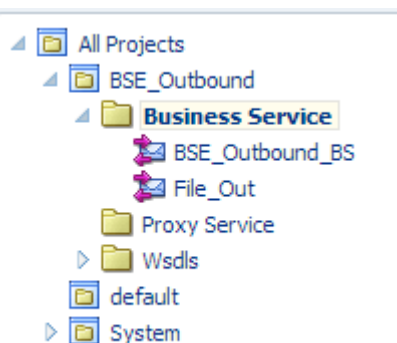
4. Click **Next**.
5. Accept the default values and click the **Create** button, as shown in [Figure 7–75](#).

**Figure 7–75 Create Business Service Window**



The created WSDL-based Business Service is listed under the Business Service folder, as shown in [Figure 7–76](#).

**Figure 7–76 WSDL-based Business Service**

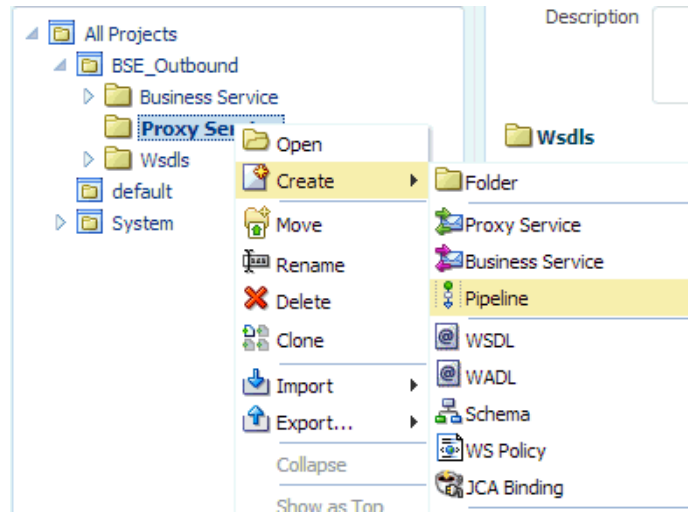


## 7.4.6 Configuring a Pipeline With Proxy Service

This section describes how to configure a Proxy Service using the Oracle Service Bus Console.

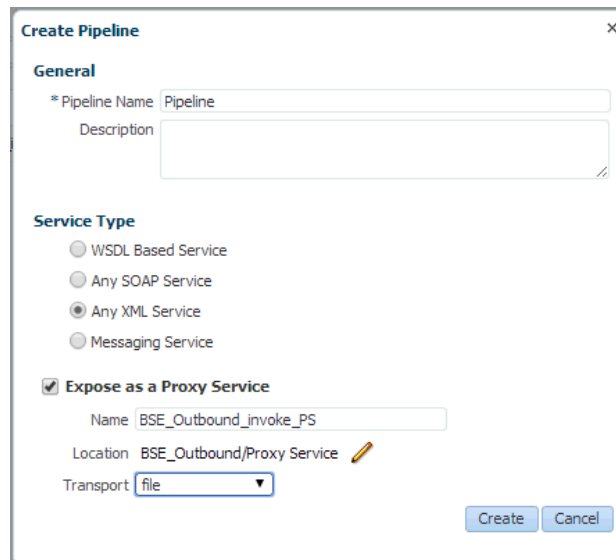
1. Right-click the Proxy Service folder, select **Create** and click **Pipeline**, as shown in [Figure 7–77](#).



**Figure 7–77 Pipeline Option**

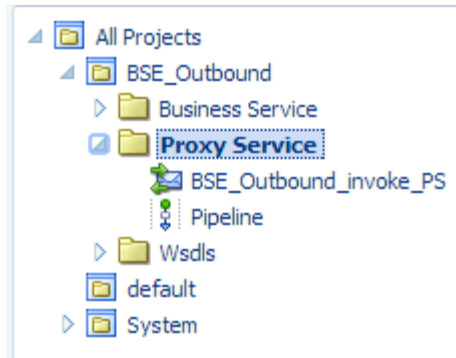
The Create Pipeline window is displayed.

2. Enter a name in the Pipeline Name field. By default, **Expose as a Proxy Service** is selected. If you wish to change the Proxy Service Name, change it and set Transport to **file**, and click **Create** as shown in [Figure 7–78](#).

**Figure 7–78 Create Pipeline Window**

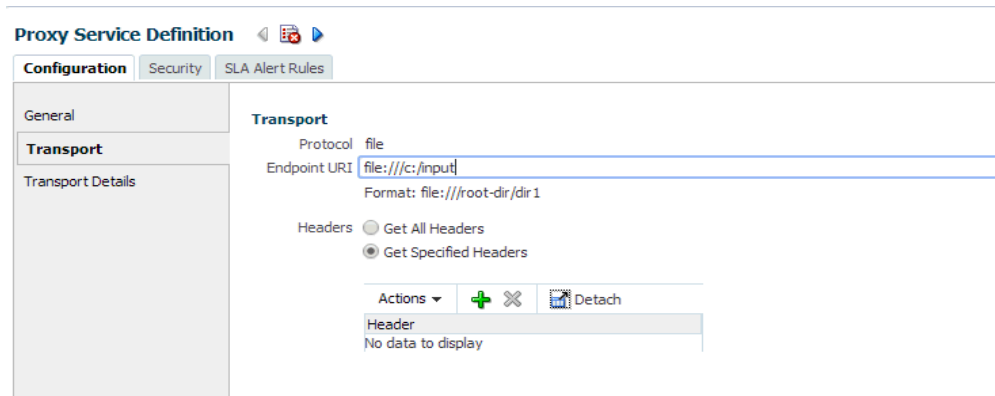
The created Pipeline and the Proxy Service is listed under Proxy Service, as shown in [Figure 7–79](#).

**Figure 7–79 Pipeline Node**



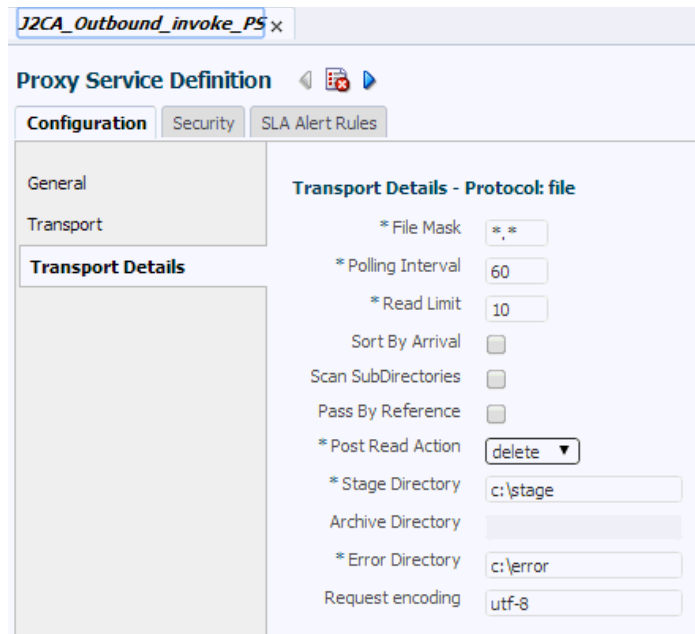
3. Double-click the created proxy service and click **Transport** in the left pane. Provide the input location in the Endpoint URI field, as shown in [Figure 7–80](#).

**Figure 7–80 Transport**



4. Click **Transport Details** in the left pane and provide the location for the Stage Directory and the Error Directory fields, as shown in [Figure 7–81](#).

**Figure 7–81 Transport Details**



5. Click the **Save All** icon in the right corner, as shown in [Figure 7–82](#).

**Figure 7–82 Save All Icon**



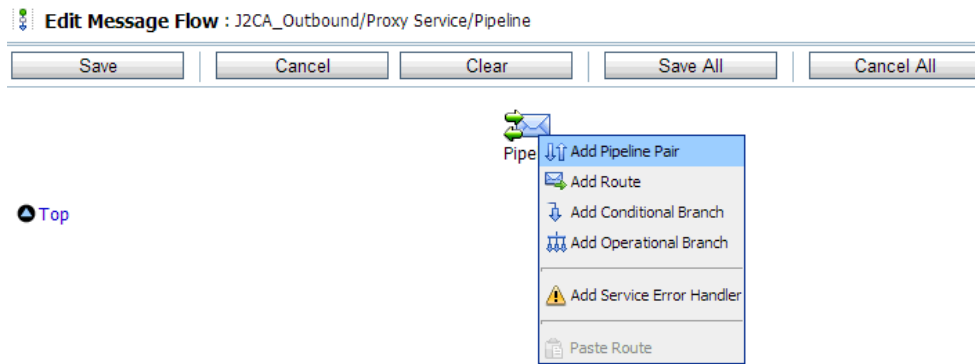
6. Double-click the **Pipeline** node and click the **Open Message Flow** icon on the right pane to open the message flow, as shown in [Figure 7–83](#).

**Figure 7–83 Open Message Flow Icon**



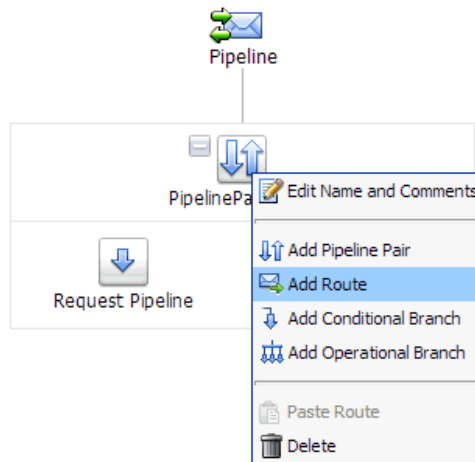
7. Click the Proxy Service icon and select **Add Pipeline Pair** from the menu, as shown in [Figure 7–84](#).

**Figure 7–84 Add Pipeline Pair Option**



8. Click the **PipelinePairNode1** icon and select **Add Route** from the menu, as shown in [Figure 7–85](#).

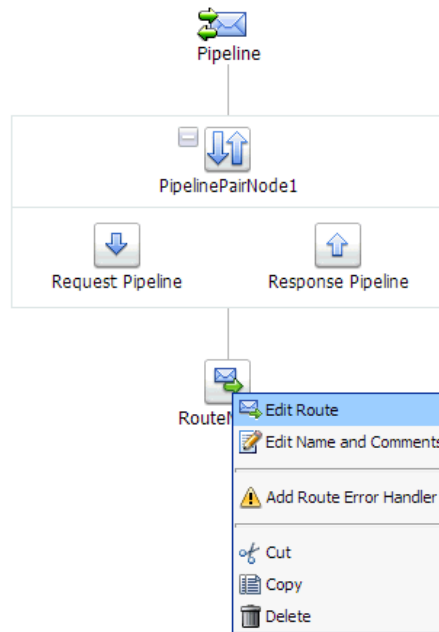
**Figure 7–85 Add Route Option**



The RouteNode1 icon is added below the PipelinePairNode1 icon.

9. Click the RouteNode1 icon and select **Edit Route** from the menu, as shown in [Figure 7–86](#).

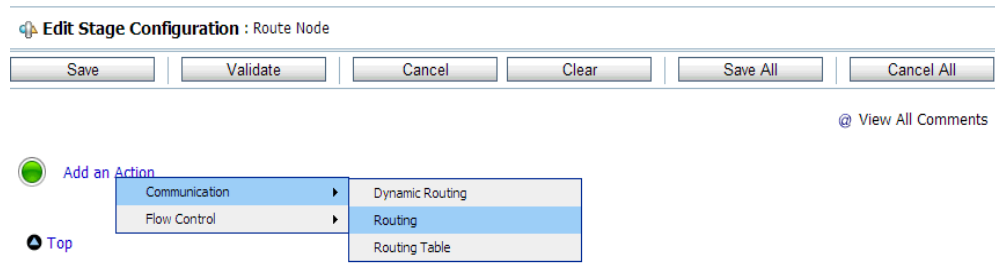
**Figure 7–86 Edit Route Option**



The Edit Stage Configuration workspace area is displayed.

10. Click **Add an Action**, select **Communication** and click **Routing**, as shown in [Figure 7–87](#).

**Figure 7–87 Edit Stage Configuration Workspace Area**



11. Click **<Service>**, as shown in [Figure 7–88](#).

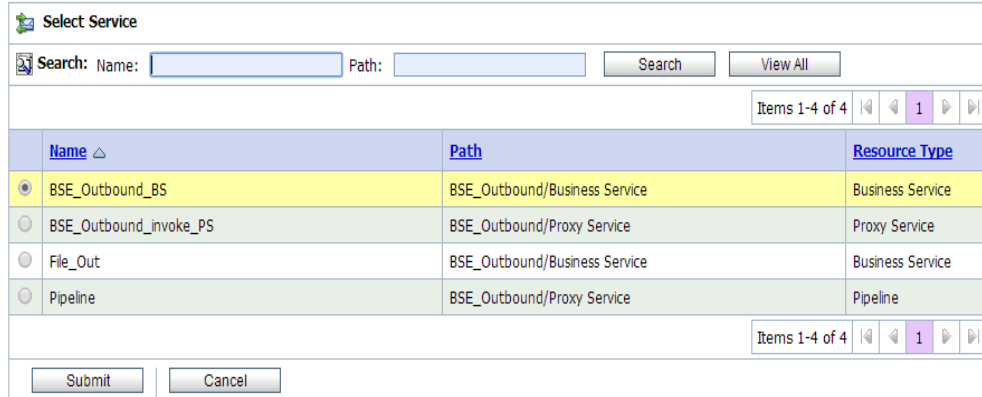
**Figure 7–88 Actions**



The Select Service dialog is displayed.

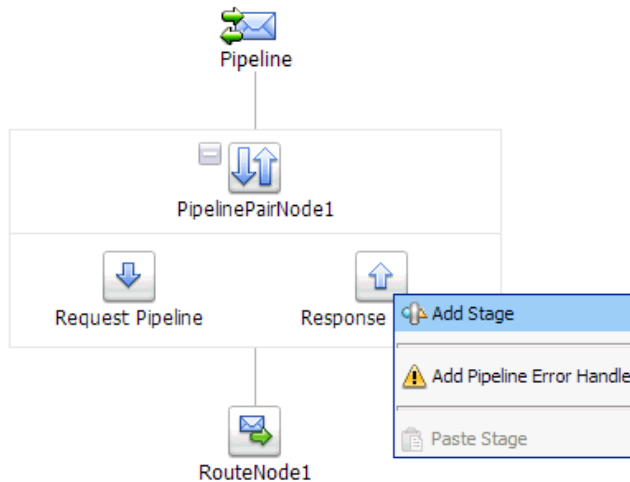
12. Select the WSDL type Business Service configured for PeopleSoft and click on **Submit**, as shown in [Figure 7–89](#).

**Figure 7–89 Select Service Dialog**



13. Select the name of the PeopleSoft business object as the operational attribute from the list, and click **Save**.
14. Click the Response Pipeline icon and select **Add Stage** from the menu, as shown in [Figure 7–90](#).

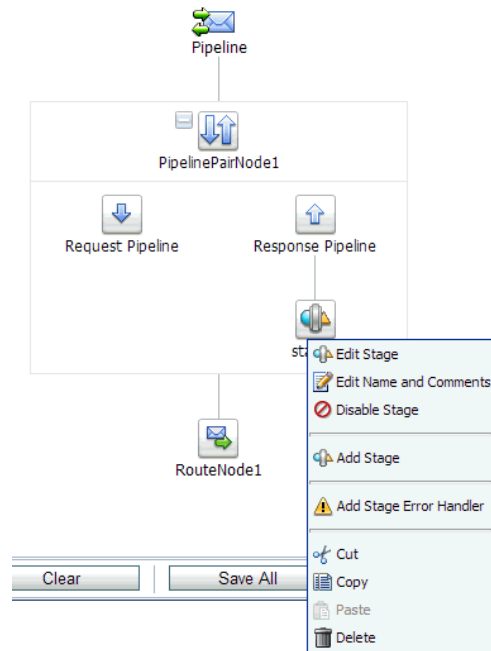
**Figure 7–90 Response Pipeline Icon**



The Stage1 icon is added below the Response Pipeline icon.

15. Click the Stage1 icon and select **Edit Stage** from the menu, as shown in [Figure 7–91](#).

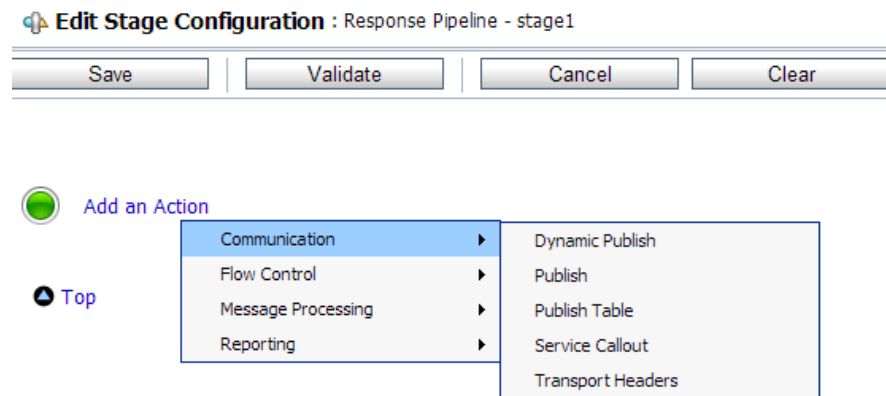
**Figure 7-91 Edit Stage Option**



The Edit Stage Configuration workspace area is displayed.

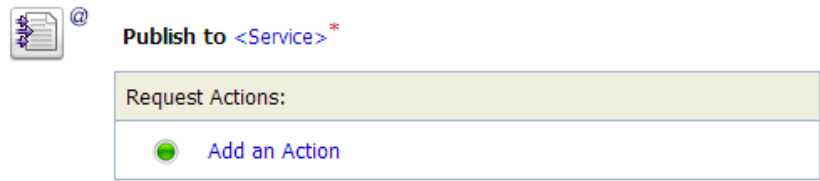
16. Click **Add an Action**, select **Communication**, and then click **Publish**, as shown in [Figure 7-92](#).

**Figure 7-92 Edit Stage Configuration Workspace Area**



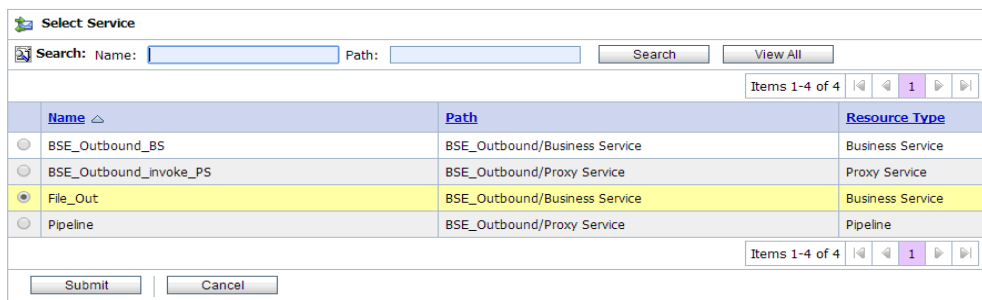
17. Click **<Service>**, as shown in [Figure 7-93](#).

**Figure 7-93** <Service> Action



18. In the Select Service dialog, select a File type Business Service and click **Submit**, as shown in [Figure 7-94](#).

**Figure 7-94** Select Service Dialog



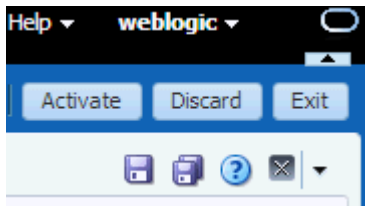
19. Click **Save All**, as shown in [Figure 7-95](#).

**Figure 7-95** Save All Button



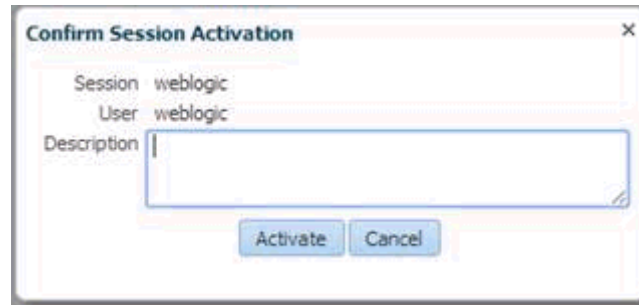
20. Click **Activate** in the right pane of the Oracle Service Bus session, as shown in [Figure 7-96](#).

**Figure 7-96** Activate Button



21. Click **Activate** to save the changes, as shown in [Figure 7-97](#).



**Figure 7–97 Confirm Session Activation**

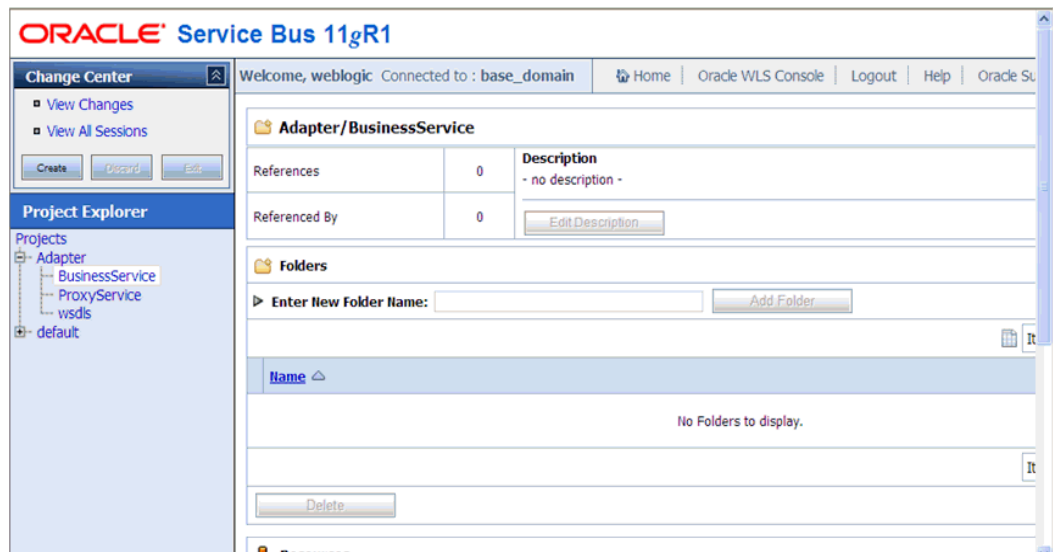
22. Copy and paste an input XML file in the input folder you have configured (for example, C:\input).

Output is received in the configured output location (for example, C:\output).

## 7.5 Configuring JMS Proxy Services Using Oracle Service Bus (J2CA Configuration)

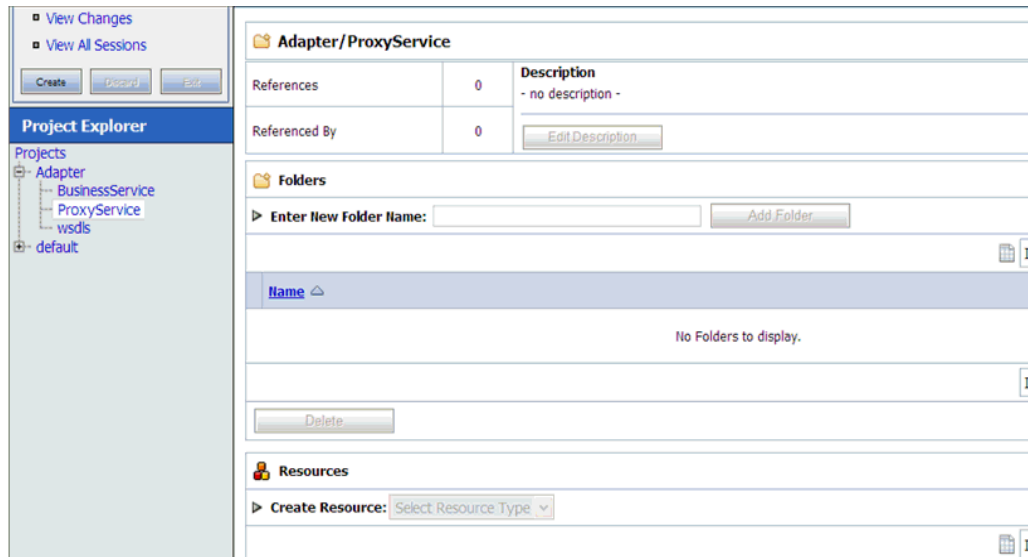
This section describes how to configure JMS Proxy Services using Oracle Service Bus for a J2CA configuration.

1. Start Oracle Service Bus and create the required project folder.  
For more information, see [Section 7.2.1, "Starting Oracle Service Bus and Creating Project Folders"](#) on page 7-2.
2. Generate and publish the WSDL from Application Explorer to the created project folder. Using the published WSDL, create a Business Service.  
For more information, see [Section 7.2.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"](#) on page 7-6.
3. Open the Service Bus Console page, as shown in [Figure 7–98](#).

**Figure 7–98 Service Bus Console**

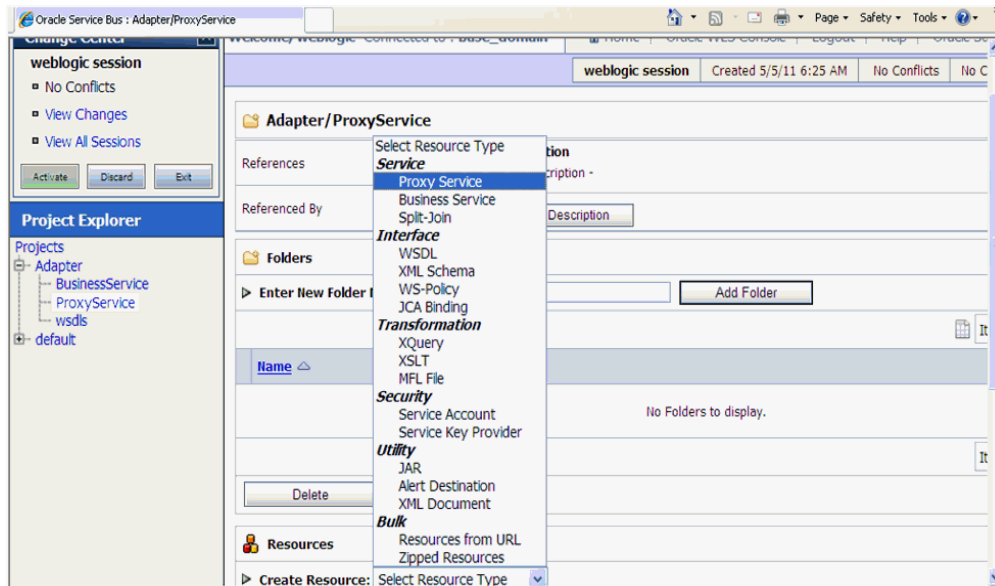
4. Select the ProxyService project folder in the left pane, and click **Create**, as shown in [Figure 7-99](#).

**Figure 7-99 Proxy Service**



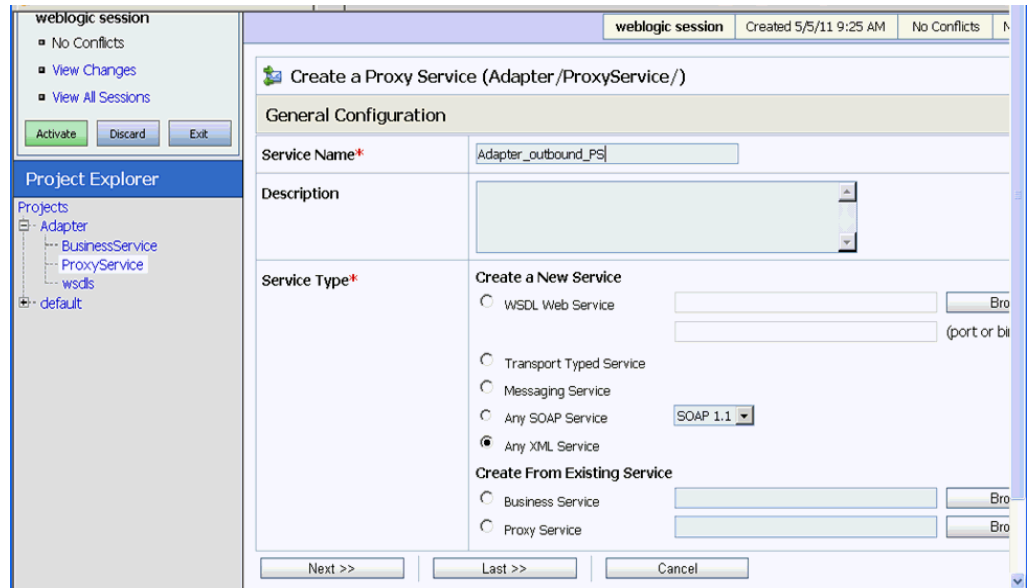
5. In the right pane, select **Proxy Service** from the Create Resource list, as shown in [Figure 7-100](#).

**Figure 7-100 Create Resource Menu**



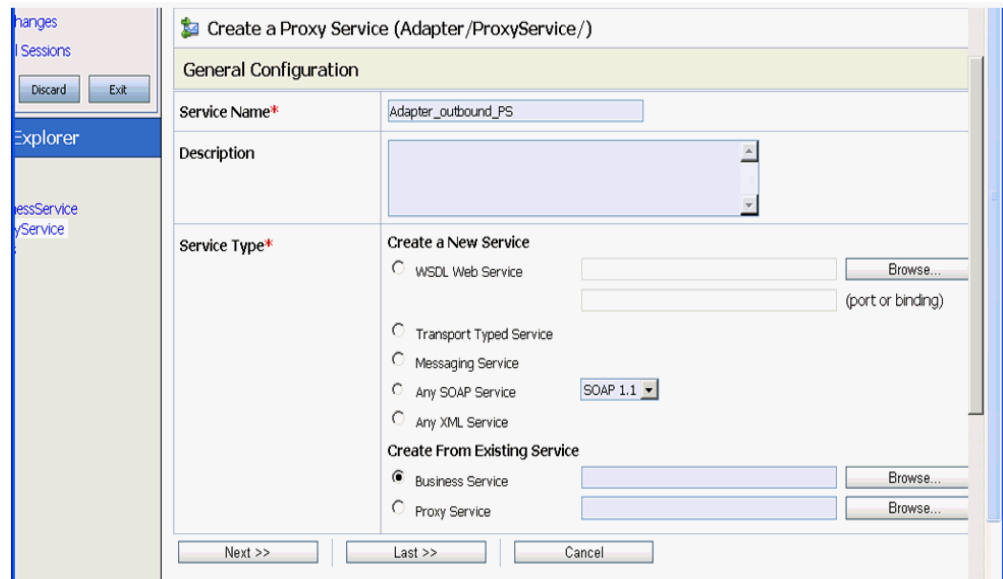
6. Enter an appropriate name in the **Service Name** field, as shown in [Figure 7-101](#).

**Figure 7-101 Service Name**



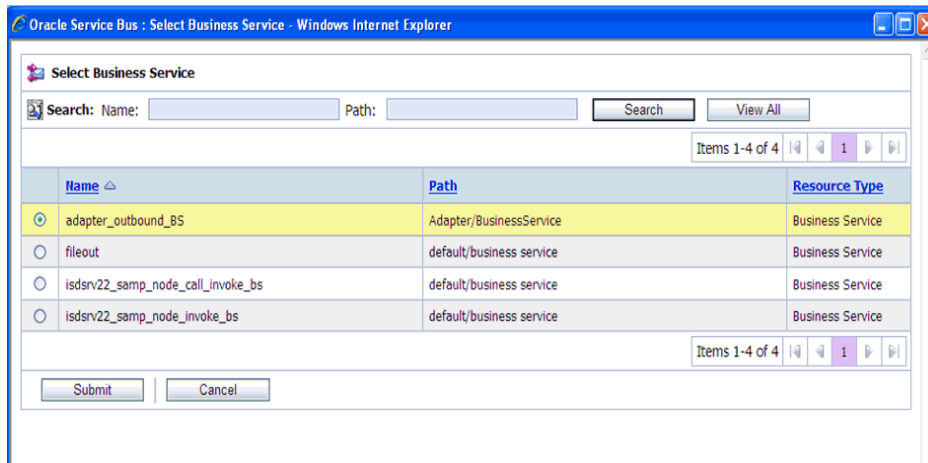
7. In the Service Type section, under Create From Existing Service, select the **Business Service** radio button and click **Browse**, as shown in [Figure 7-102](#).

**Figure 7-102 Business Service**



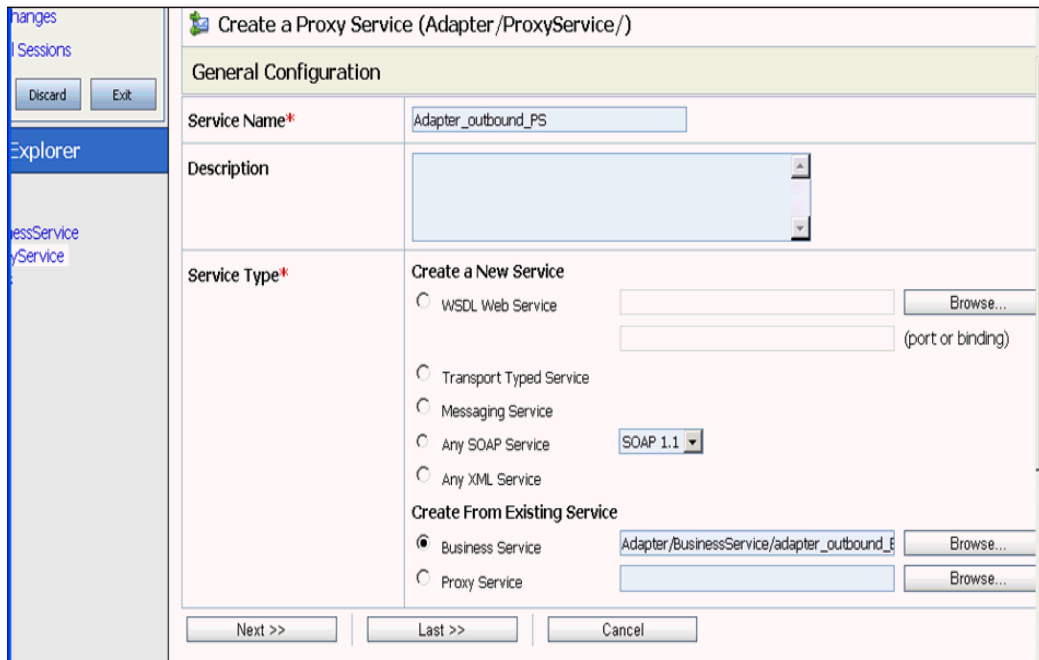
8. Select the existing business service and click **Submit**, as shown in [Figure 7-103](#).

**Figure 7–103 Existing Business Service**



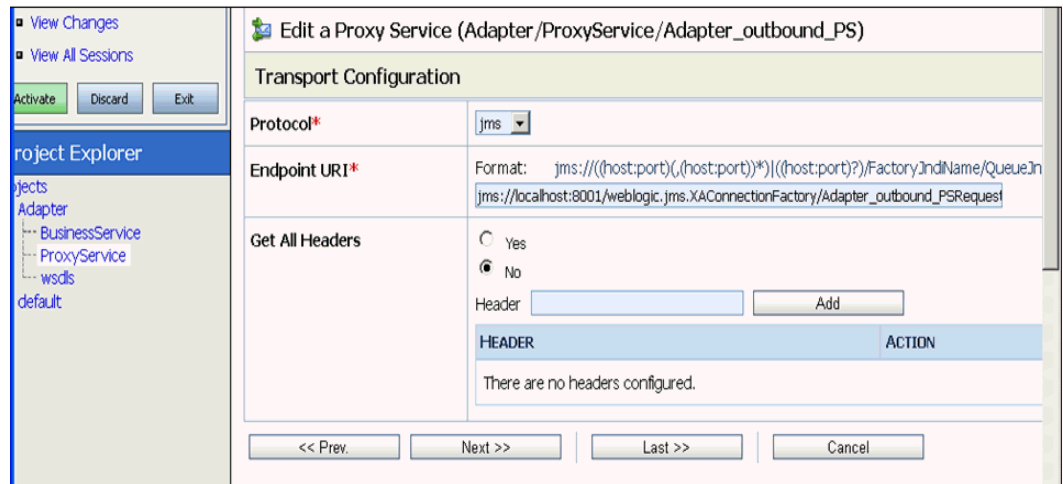
9. Click **Next**, as shown in Figure 7–104.

**Figure 7–104 Next**



10. Select **jms** from the Protocol list and click **Next**, as shown in Figure 7–105.

**Figure 7–105 Protocol List**



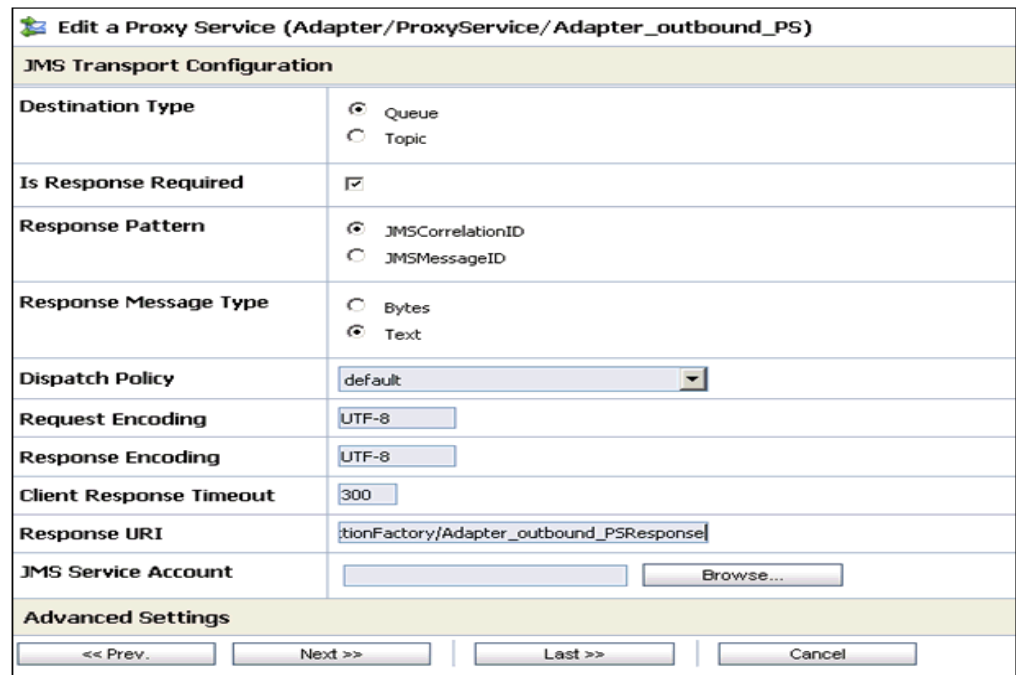
11. Provide the following parameters, as shown in [Figure 7–106](#).

- a. Select **Queue** in the Destination Type section.
- b. Enable the **Is Response Required** check box.
- c. Select **Text** in the Response Message Type section.
- d. In the Response URI field, provide the Endpoint URI used in the Transport Configuration and change Request to Response.

For example:

```
jms://localhost:8001/weblogic.jms.XAConnectionFactory/Adapter_outbound_PSResponse
```

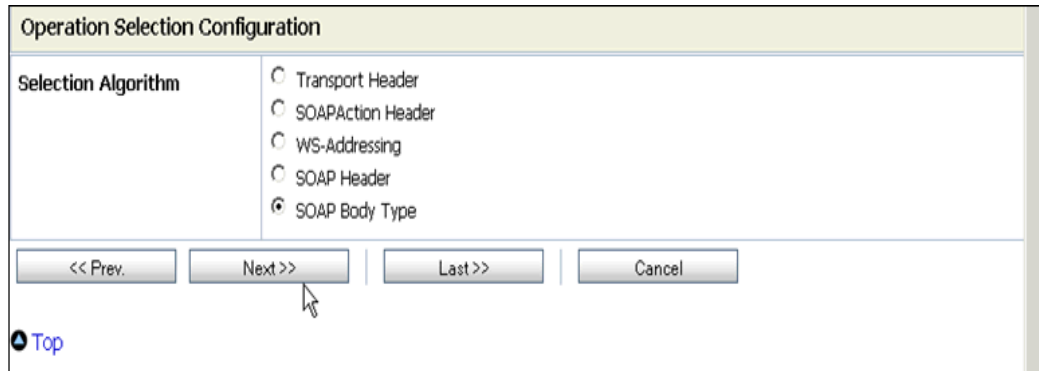
**Figure 7–106 Edit a Proxy Service**



12. Click **Next**.

The Operation Selection Configuration pane appears, as shown in [Figure 7-107](#).

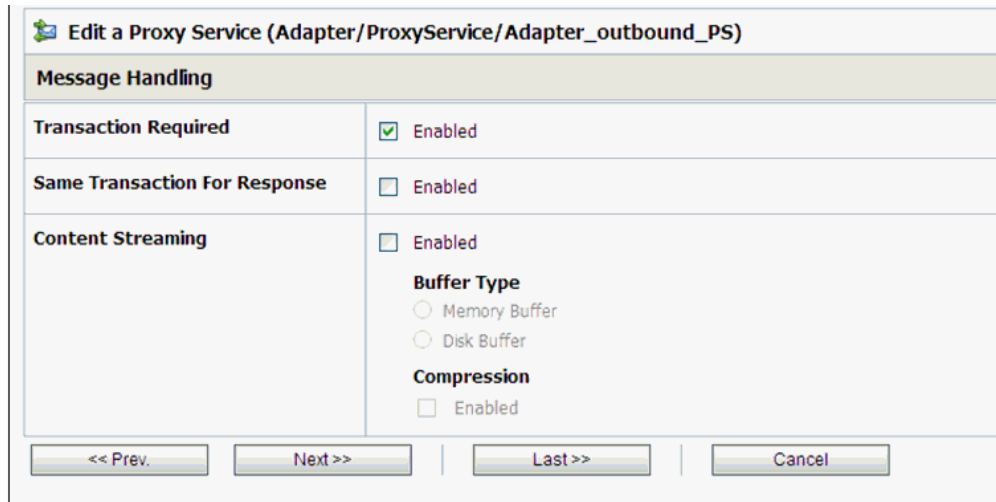
**Figure 7-107 Operation Selection Configuration Pane**



13. Ensure the **SOAP Body Type** is selected and click **Next**.

14. Enable the **Transaction Required** box and click **Next**, as shown in [Figure 7-108](#).

**Figure 7-108 Message Handling**



15. Click **Save**, as shown in [Figure 7-109](#).

**Figure 7–109 Save**

Use SSL	Disabled
Is Response Required	Enabled
Request Encoding	UTF-8
Response Encoding	UTF-8
Response Pattern	JMSCorrelationID
JNDI Timeout	0
Response URI	jms://localhost:8001/weblogic.jms.XAConnectionFactory/Adapter_outbound_BSResponse
Response Message Type	Text
Client Response Timeout	300
Is XA Required	False
<b>Operation Selection Configuration</b>	
Selection Algorithm	SOAP Body Type
<b>Message Handling Configuration</b>	
Transaction Required	Enabled
Same Transaction For Response	Disabled
Content Streaming	Disabled
<input style="margin-right: 10px;" type="button" value=" &lt;&lt; Prev. "/> <input style="margin-right: 10px;" type="button" value=" Save "/> <input style="margin-right: 10px;" type="button" value=" Cancel "/>	

The created Proxy Service is saved, as shown in [Figure 7–110](#).

**Figure 7–110 Proxy Service**

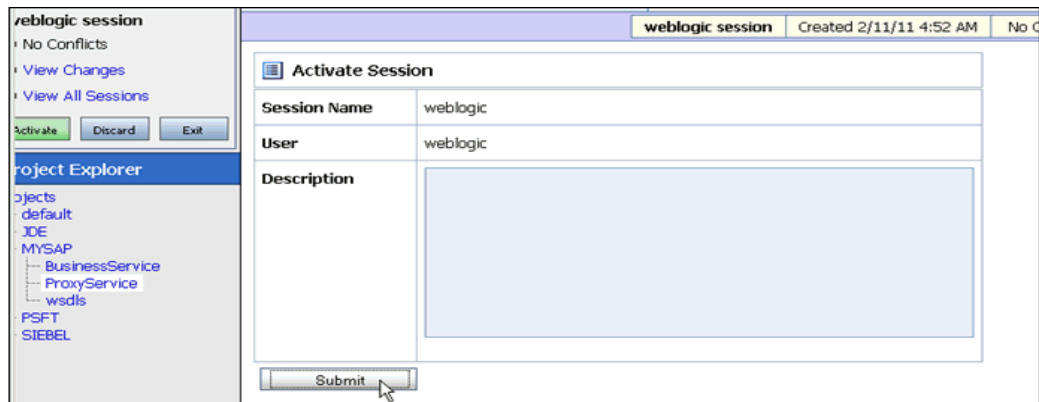
The screenshot shows the Oracle Service Bus console interface. On the left, the 'Project Explorer' pane shows a tree structure: 'Projects' > 'Adapter' > 'BusinessService' > 'ProxyService' > 'wsdls'. Below this are tabs for 'Operations', 'Resource Browser', and 'Project Explorer'. The main workspace is divided into two sections: 'Folders' and 'Resources'. The 'Folders' section has an 'Enter New Folder Name:' field and an 'Add Folder' button, with a message 'No Folders to display.' below. The 'Resources' section has a 'Create Resource:' dropdown menu set to 'Select Resource Type'. Below this is a table with one resource:

Name	Resource Type	Actions
Adapter_outbound_PS	Proxy Service	[Refresh] [Refresh] [Refresh]

Below the table is a 'Delete' button.

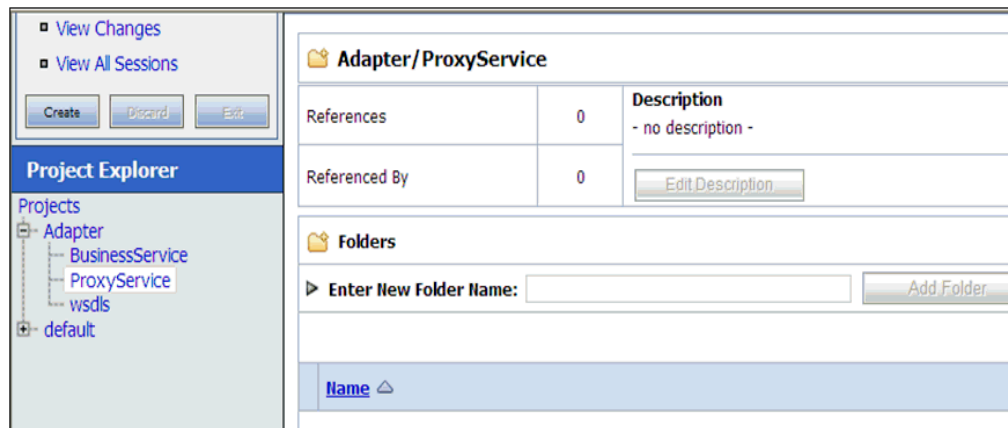
16. In the left pane, click **Activate**, and then **Submit**, as shown in [Figure 7–111](#).

**Figure 7–111 Activate Session**



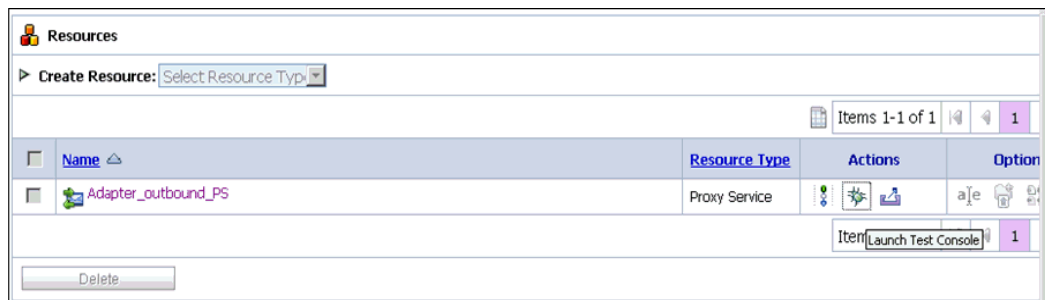
17. In the left pane, click **ProxyService** under the Projects folder, as shown in Figure 7–112.

**Figure 7–112 Adapter/ProxyService**



18. Click the **Launch Test Console** icon for the created Proxy Service, as shown in Figure 7–113.

**Figure 7–113 Launch Test Console Icon**



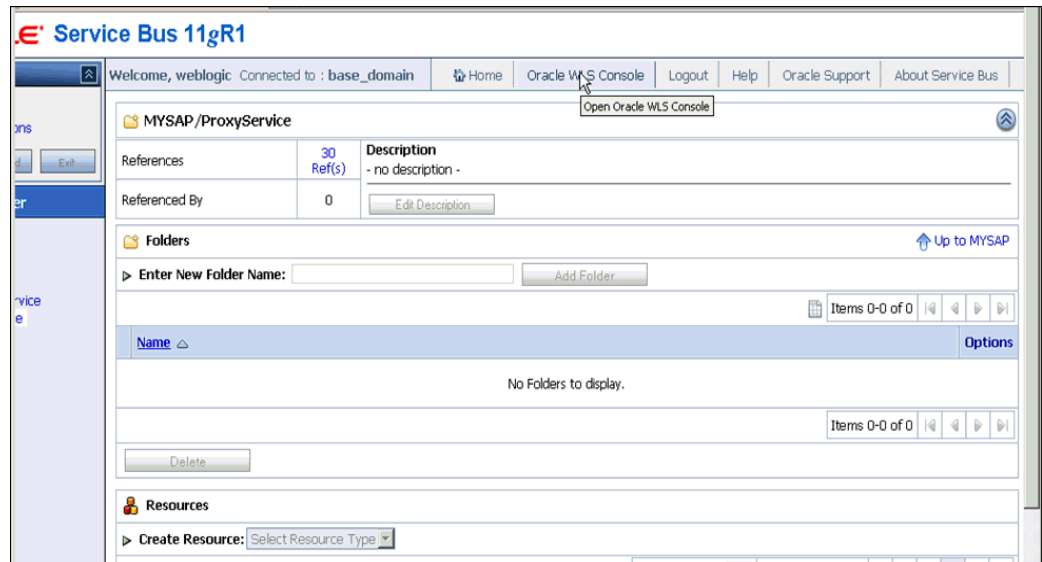
19. Provide the input values for **Payload**, uncheck the **Direct Call** box, and click **Execute**.

20. Review the Response document, and then click **Close**.

21. Click the **Oracle WLS Console** tab, as shown in Figure 7–114.

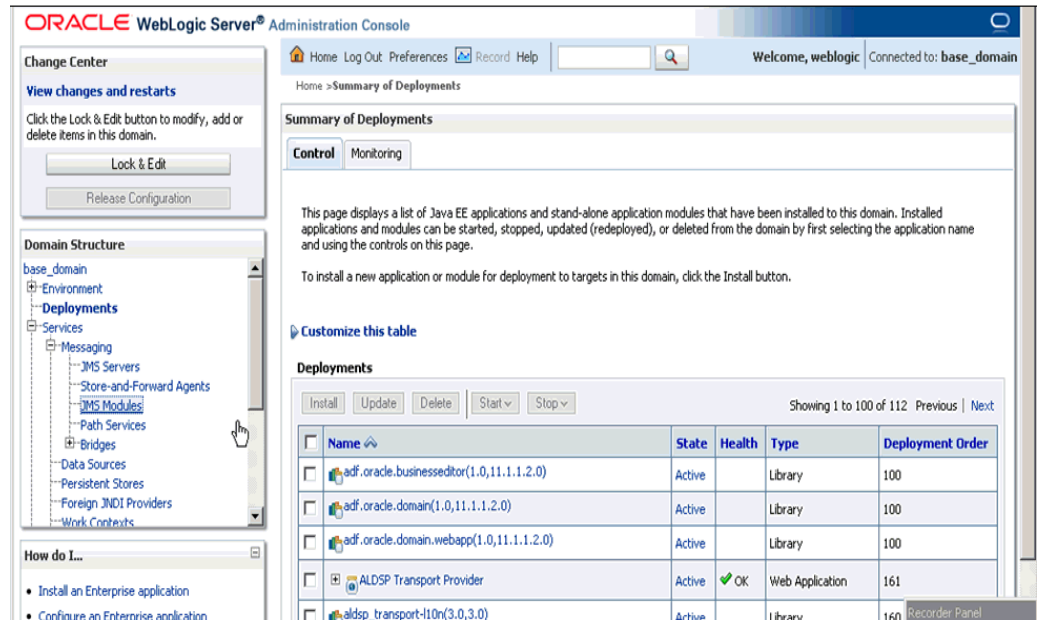


Figure 7–114 ProxyService



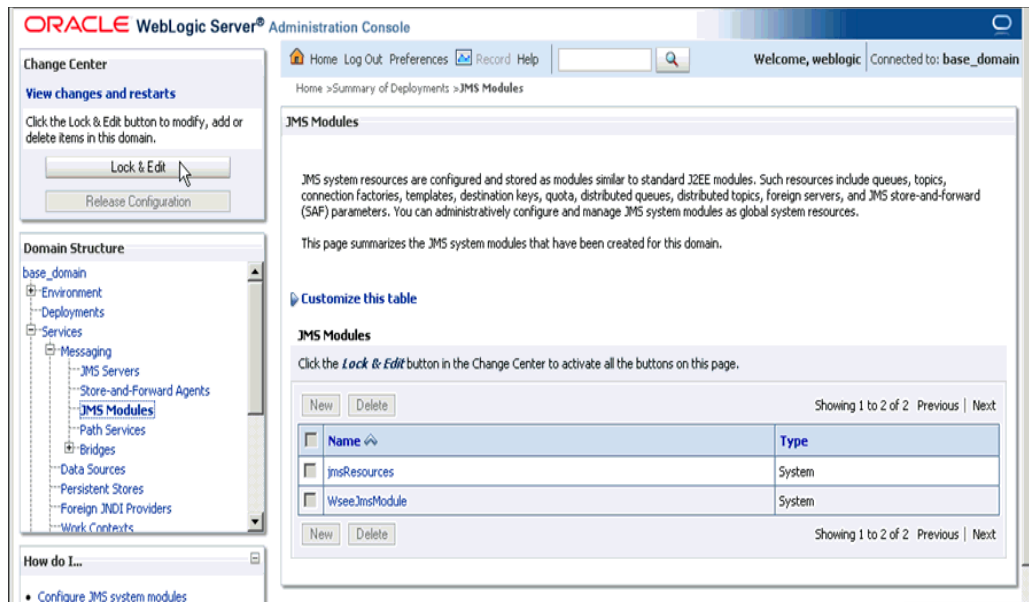
22. In the Oracle WLS Console, expand **Services**, expand **Messaging**, and click **JMS Modules**, as shown in Figure 7–115.

Figure 7–115 Oracle WLS Console



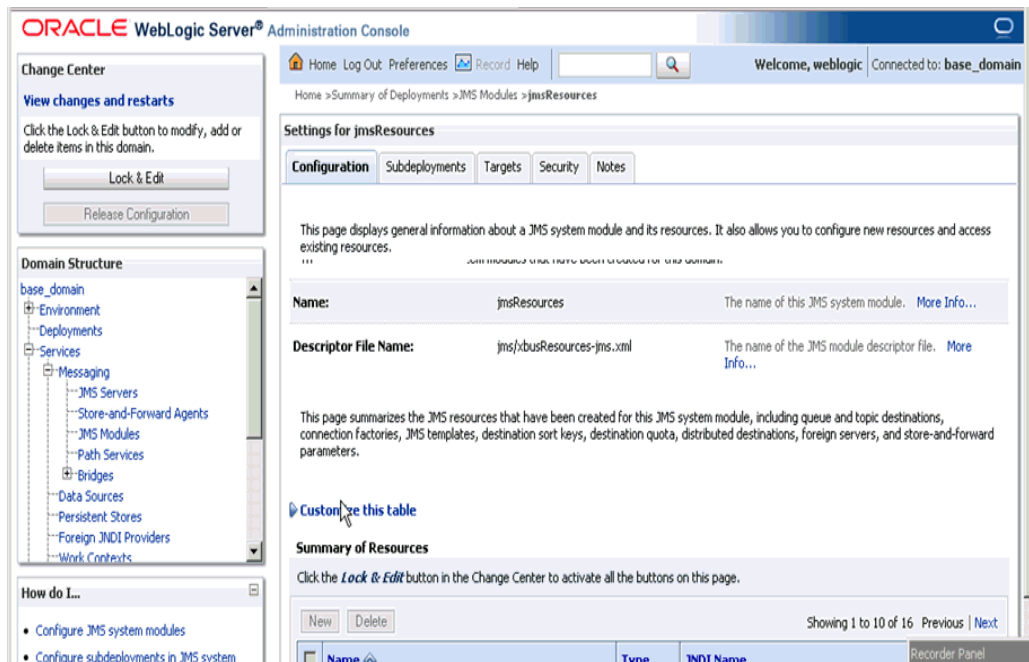
23. Click **jmsResources**, as shown in Figure 7–116.

**Figure 7–116 JMS Modules**



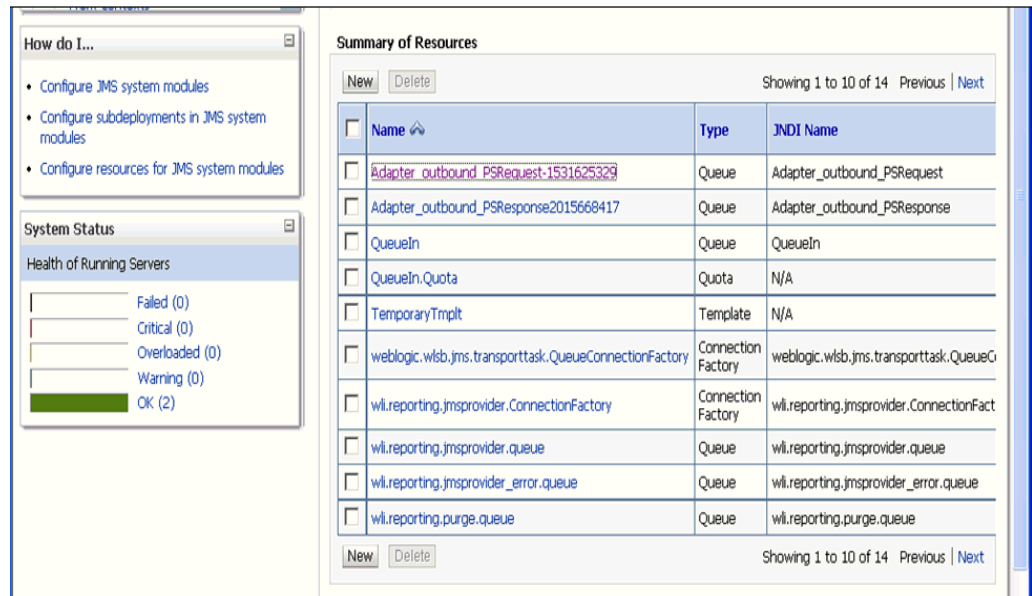
24. Click **Lock & Edit**, as shown in [Figure 7–117](#).

**Figure 7–117 Configuration Settings**



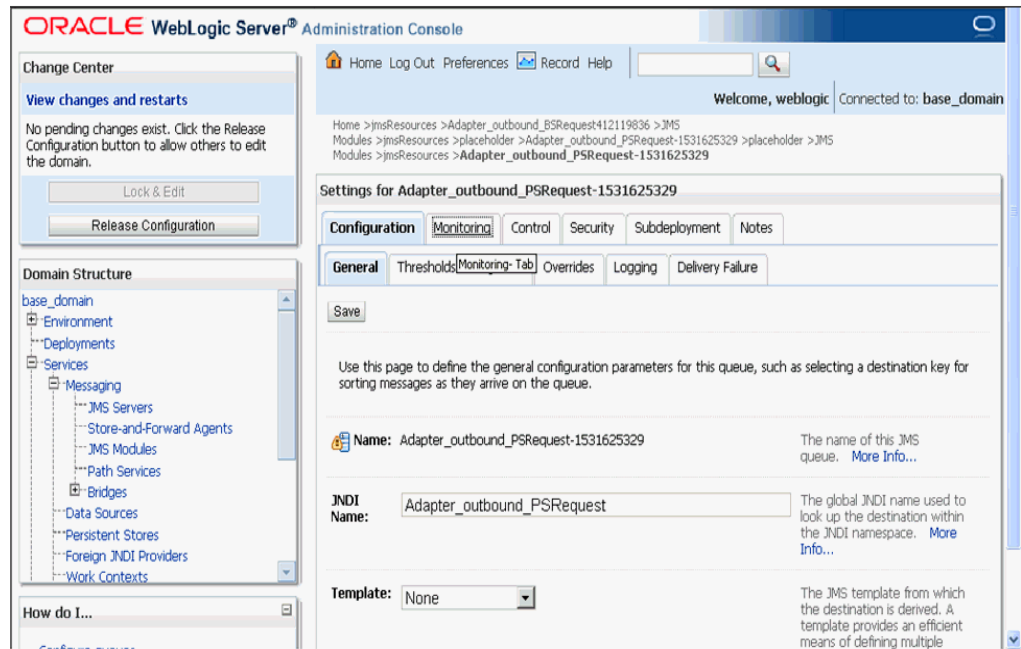
25. Click the appropriate request link, for example, **Adapter\_outbound\_PSRequest**, as shown in [Figure 7–118](#).

**Figure 7–118 Adapter\_outbound\_PSRequest**



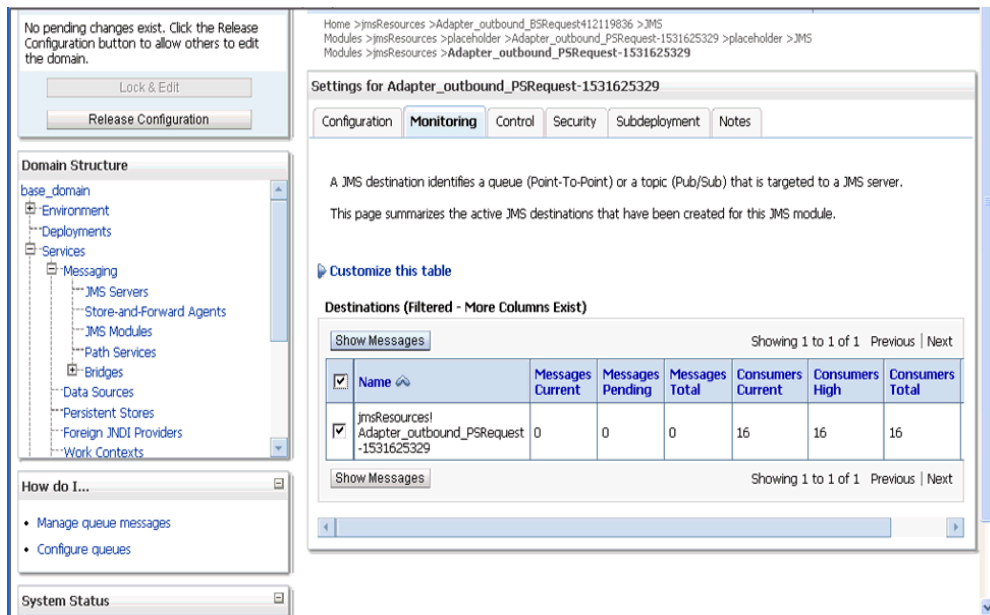
26. Click the **Monitoring** tab, as shown in Figure 7–119.

**Figure 7–119 Monitoring Tab**



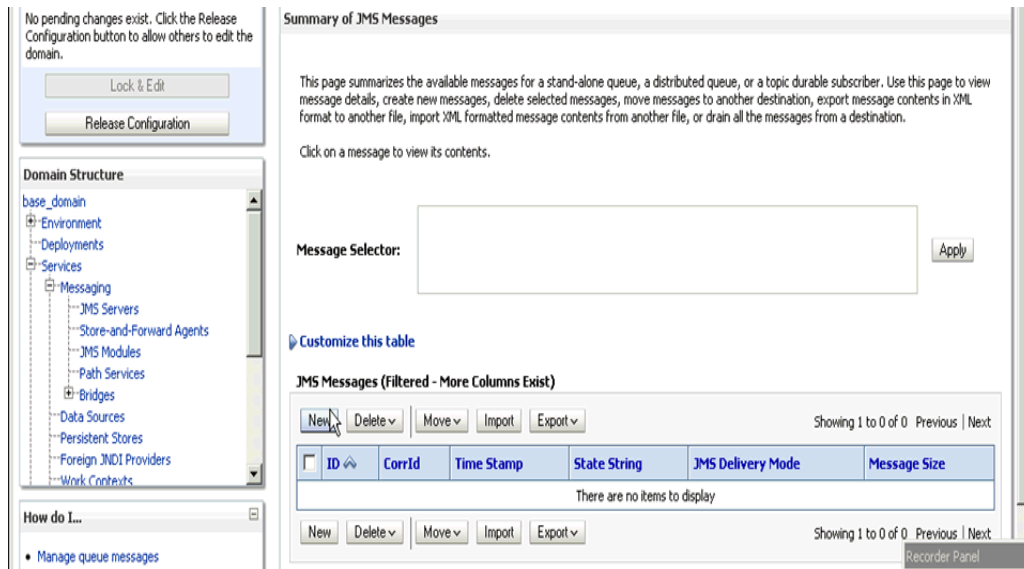
27. Enable the check box and click **Show Messages**, as shown in Figure 7–120.

**Figure 7–120 Adapter Settings**



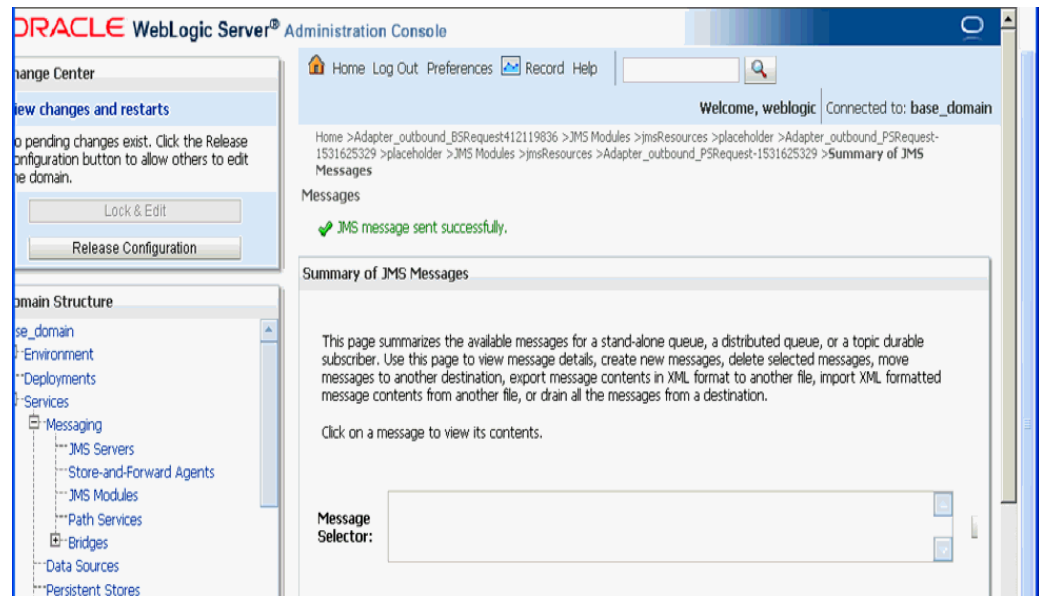
28. Click **New**, as shown in Figure 7–121.

**Figure 7–121 JMS Messages**



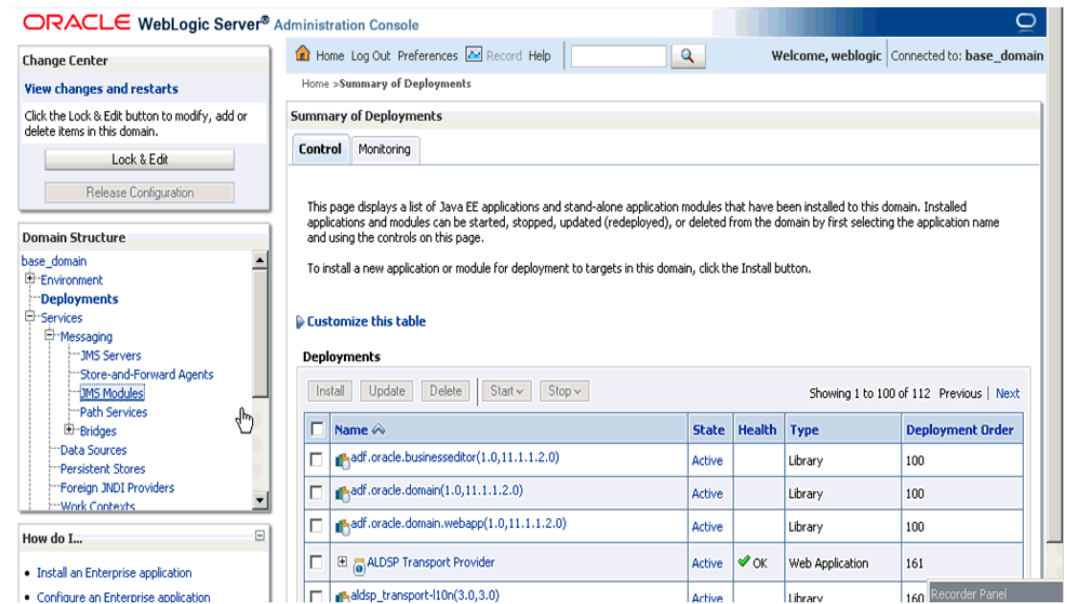
29. Provide the input payload in the **Body** field and click **OK**.  
A Success message appears, as shown in Figure 7–122.

Figure 7–122 JMS Success Message



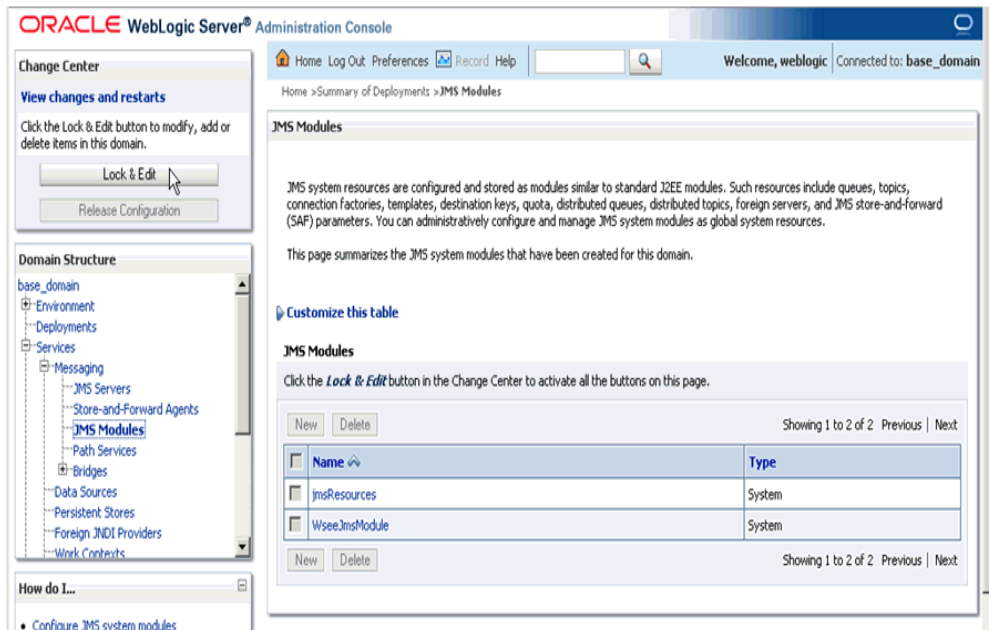
30. In the Oracle WLS Console, expand **Services**, expand **Messaging**, and click **JMS Modules**, as shown in Figure 7–123.

Figure 7–123 JMS Modules



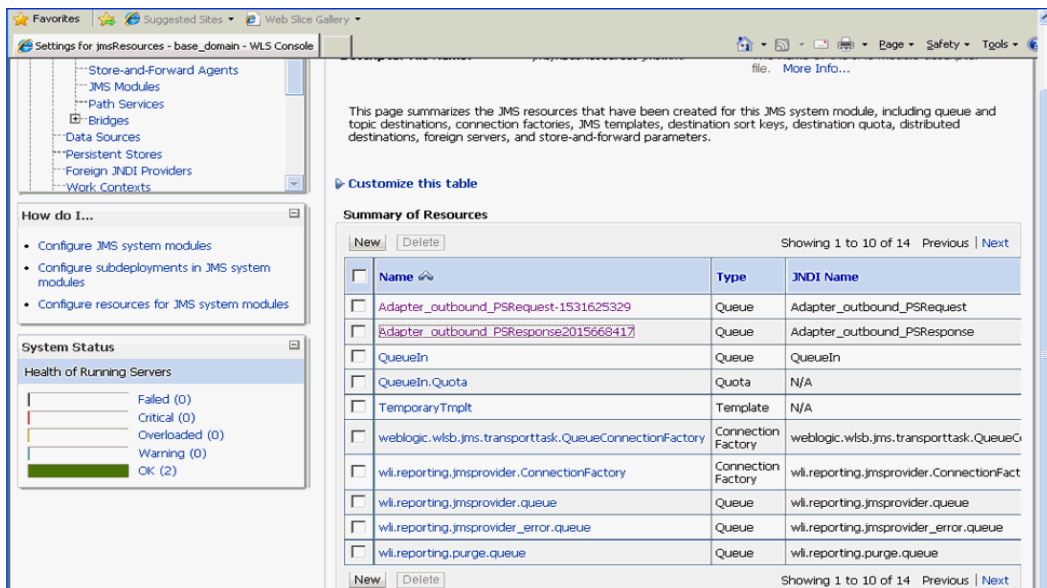
31. Click **.jmsResources**, as shown in Figure 7–124.

**Figure 7–124 jmsResources**



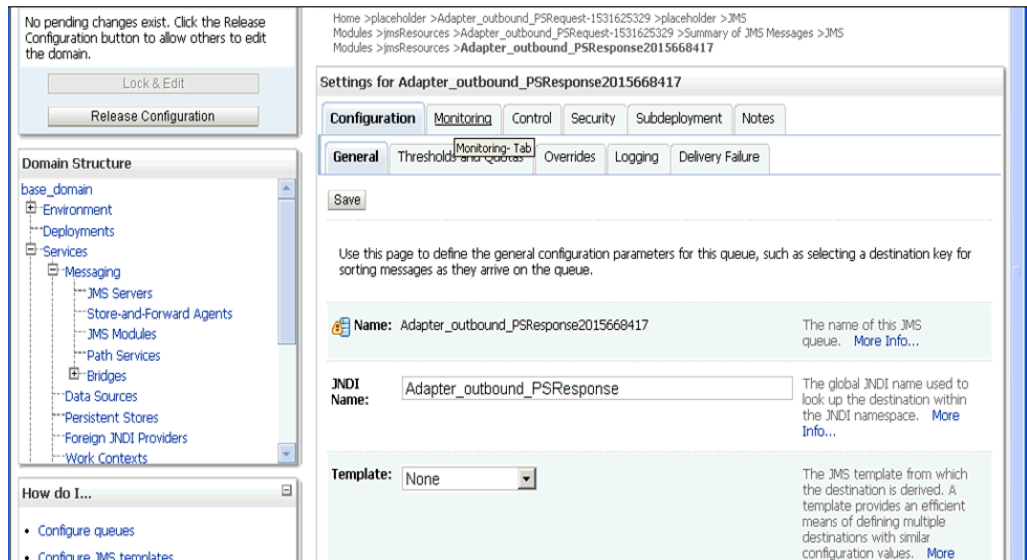
32. Click the appropriate response link, for example, `Adapter_outbound_PSResponse`, as shown in Figure 7–125.

**Figure 7–125 Summary of Resources**



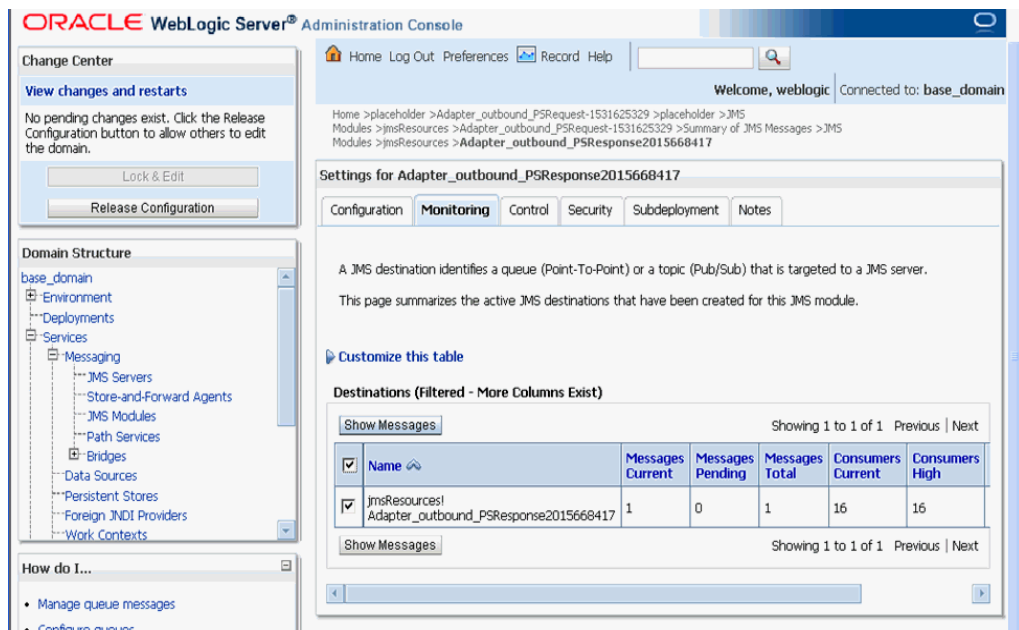
33. Click the **Monitoring** tab, as shown in Figure 7–126.

**Figure 7–126 Monitoring Tab**



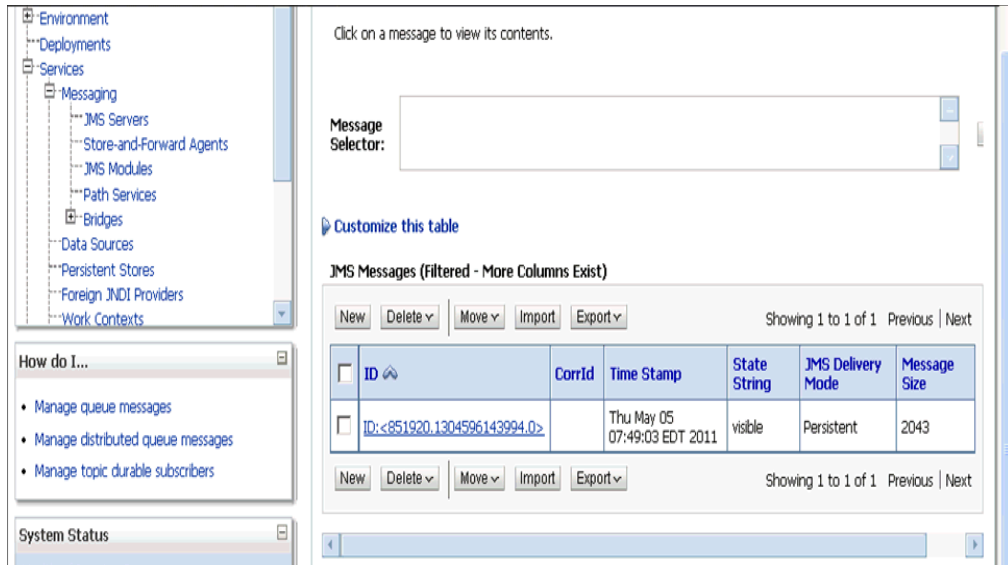
34. Enable the check box and click **Show Messages**, as shown in Figure 7–127.

**Figure 7–127 Destination Messages**



35. Click the **ID** link, as shown in Figure 7–128.

**Figure 7–128 JMS Messages**



The response document is shown under the Text field.

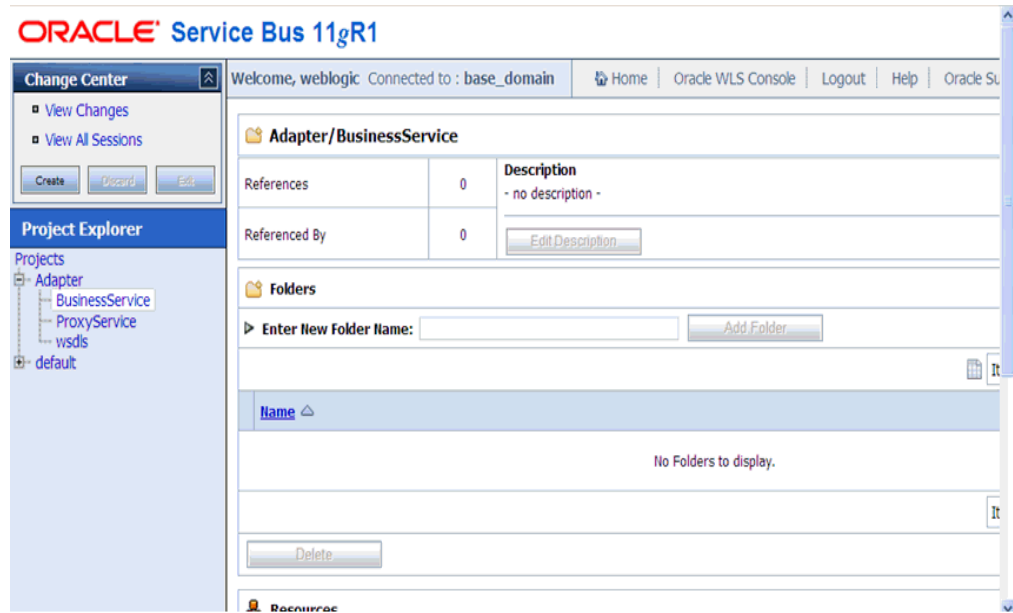
## 7.6 Configuring HTTP Proxy Services Using Oracle Service Bus (J2CA Configuration)

This section describes how to configure HTTP Proxy Services using Oracle Service Bus for a J2CA configuration.

1. Start the Oracle Service Bus and create the required project folders.  
For more information, see [Section 7.2.1, "Starting Oracle Service Bus and Creating Project Folders"](#) on page 7-2.
2. Generate and publish the WSDL from Application Explorer to the created project folder, and create a Business Service using the published WSDL.  
For more information, see [Section 7.2.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"](#) on page 7-6.
3. Open the Service Bus console page, as shown in [Figure 7–129](#).

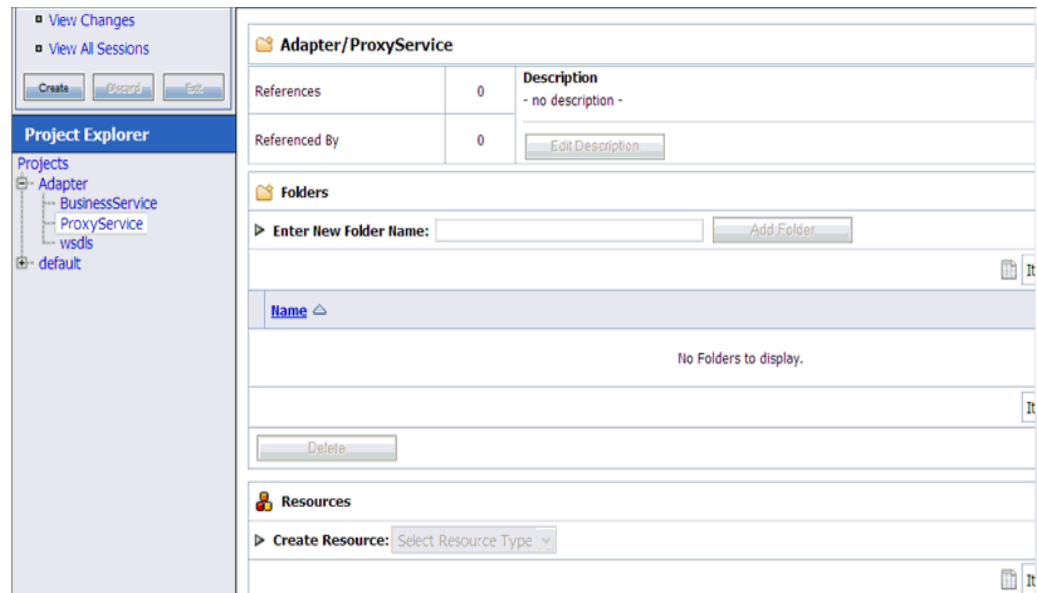


**Figure 7–129 Service Bus Console Page**



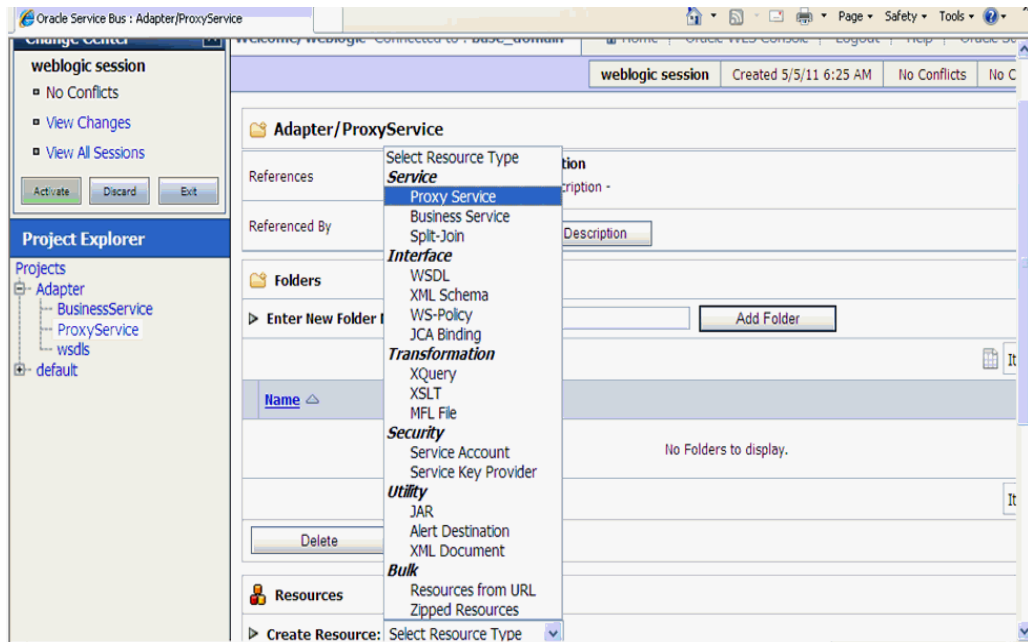
4. In the Project Explorer, select the **ProxyService** project folder, and click **Create**, as shown in [Figure 7–130](#).

**Figure 7–130 Project Explorer**



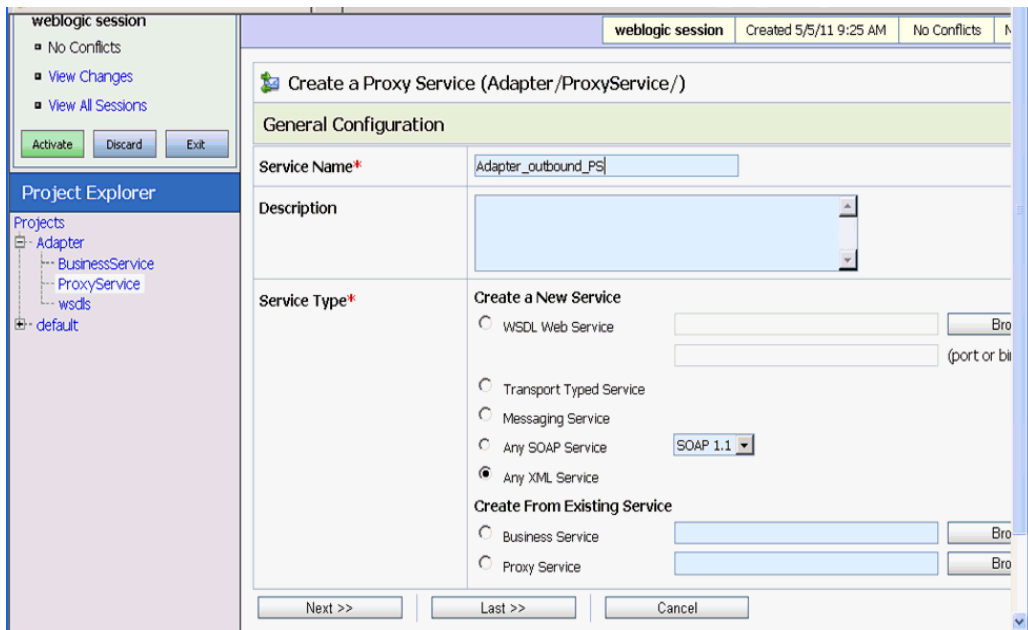
5. In the Create Resource list on the right pane, select **Proxy Service**, as shown in [Figure 7–131](#).

**Figure 7–131 Proxy Service**



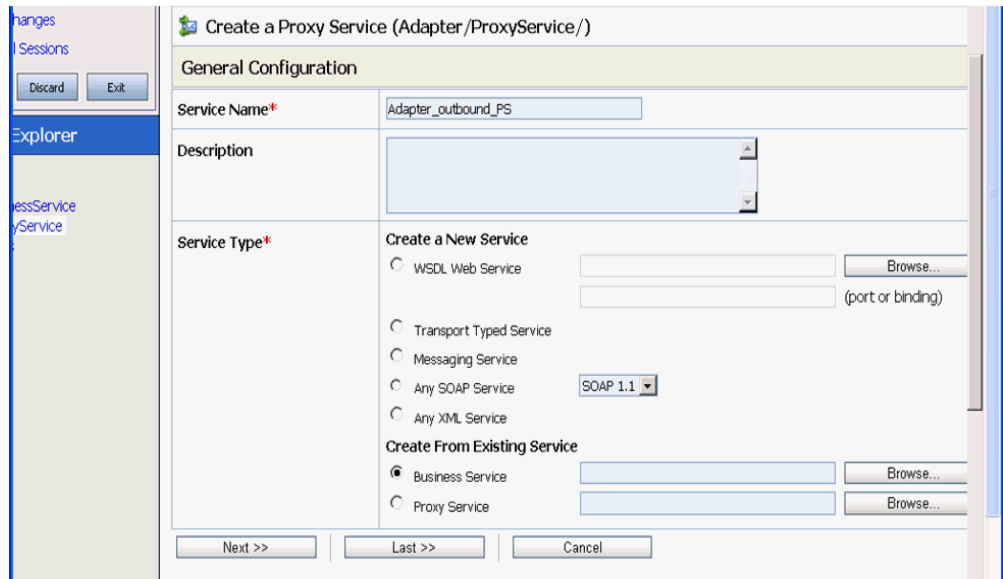
6. In the Service Name field, enter an appropriate name, as shown in [Figure 7–132](#).

**Figure 7–132 Service Name**



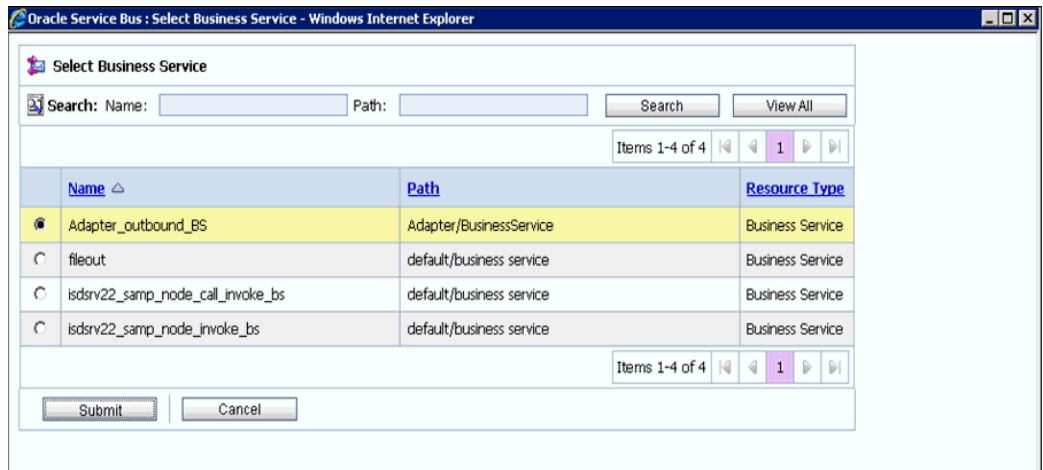
7. In the Service Type section, under Create From Existing Service, select the **Business Service** radio button and click **Browse**, as shown in [Figure 7–133](#).

**Figure 7–133 General Configuration**



8. Select the existing Business Service and click **Submit**, as shown in [Figure 7–134](#).

**Figure 7–134 Business Service**



9. Click **Next**, as shown in [Figure 7–135](#).

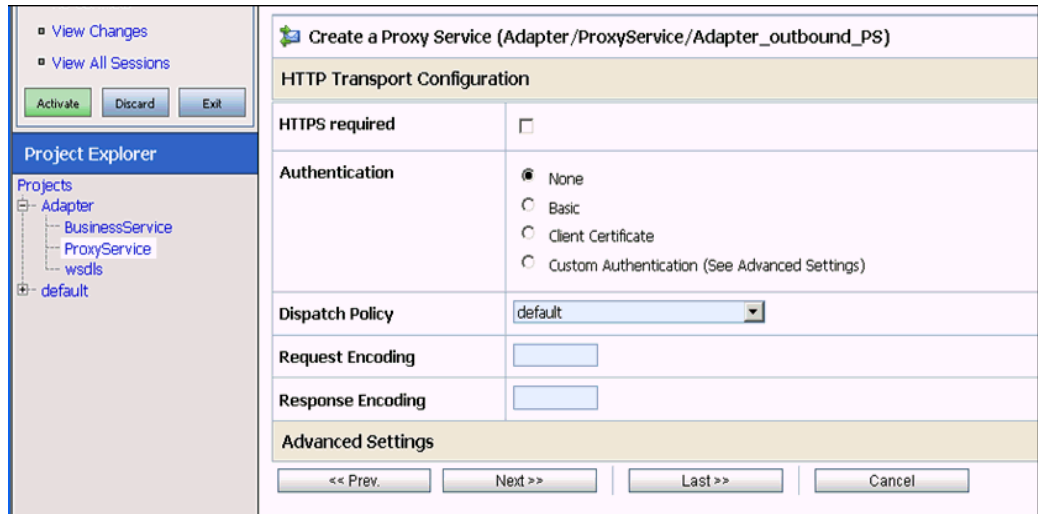
**Figure 7–135 General Configuration**

10. Select **http** in the Protocol list and click **Next**, as shown in [Figure 7–136](#).

**Figure 7–136 Transport Configuration**

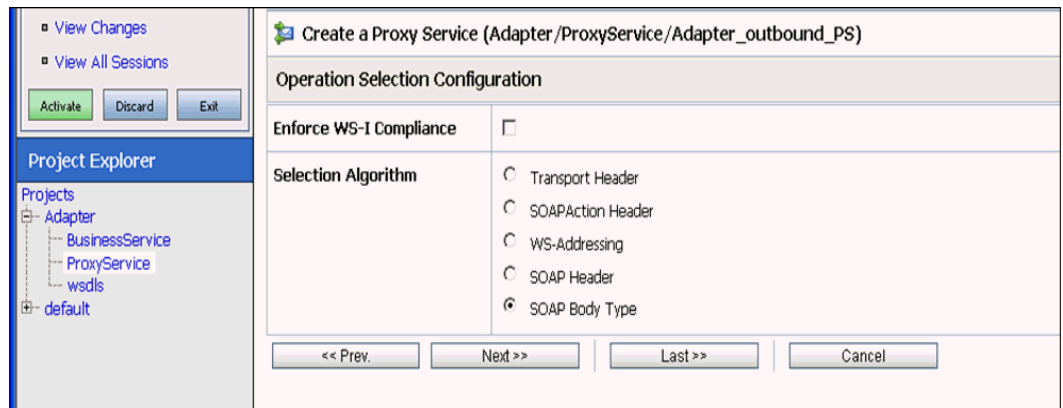
11. Click **Next**, as shown in [Figure 7–137](#).

**Figure 7–137 HTTP Transport Configuration**



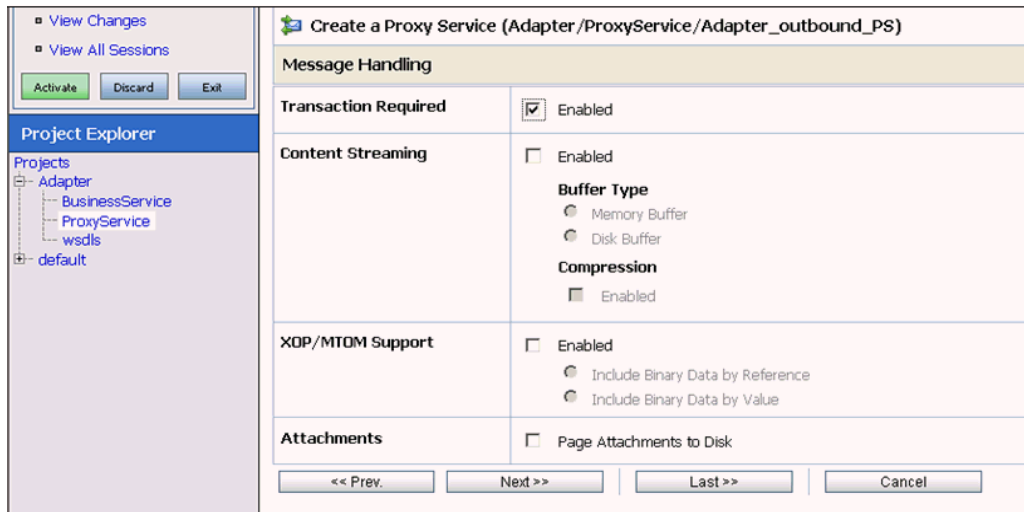
12. Click **Next**, as shown in [Figure 7–138](#).

**Figure 7–138 Operation Selection Configuration**



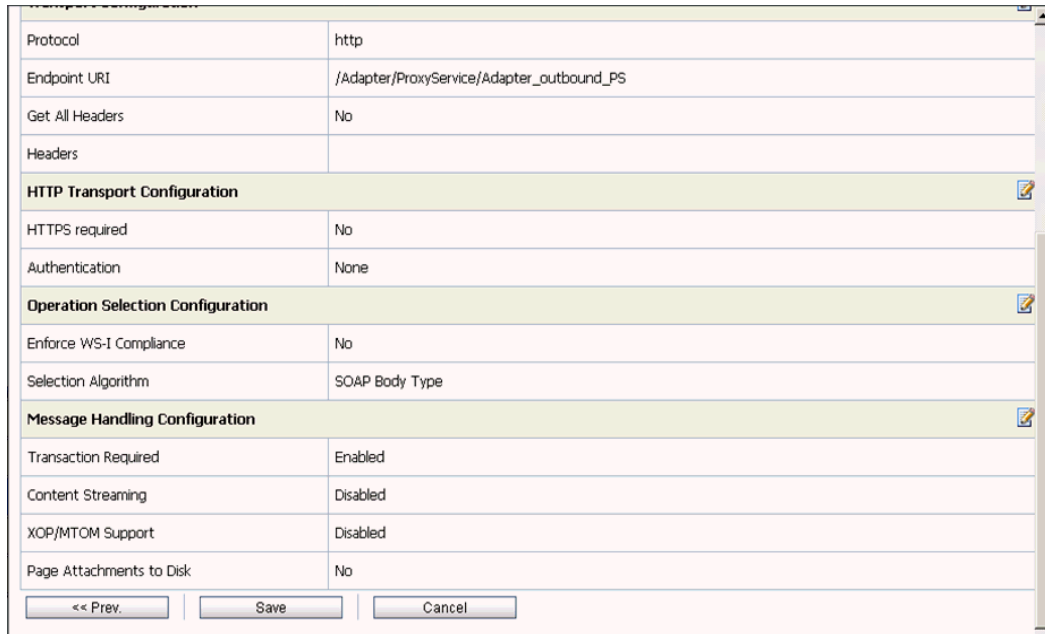
13. Enable the **Transaction Required** check box and click **Next**, as shown in [Figure 7–139](#).

**Figure 7–139 Message Handling**



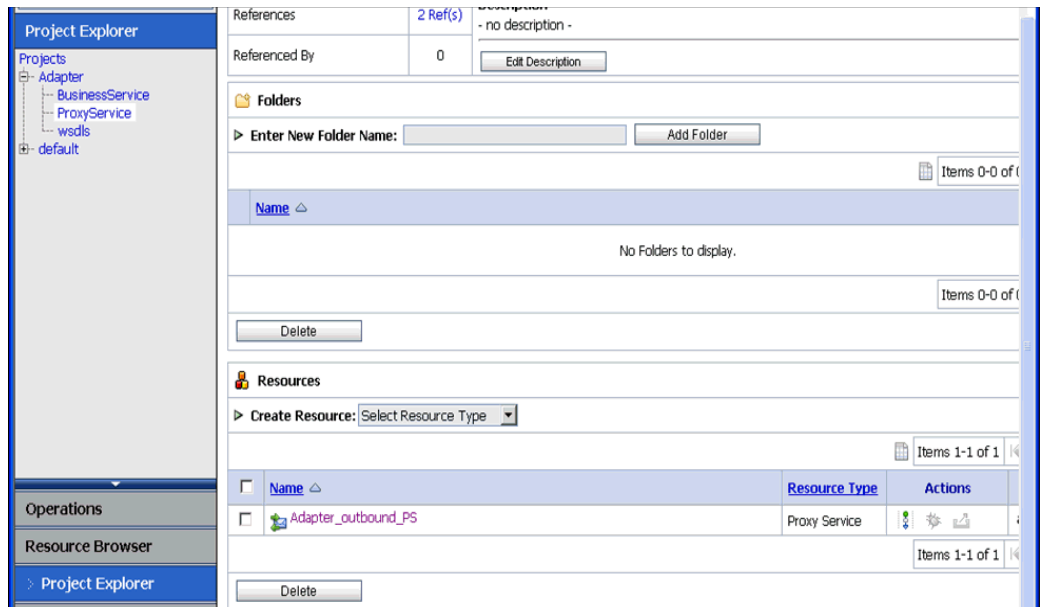
14. Click **Save**, as shown in [Figure 7–140](#).

**Figure 7–140 Save**



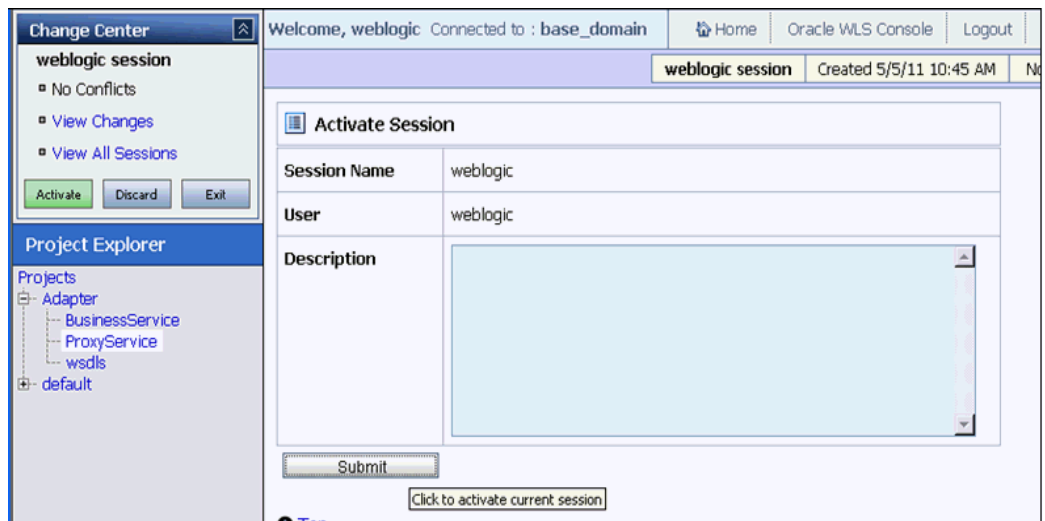
The created Proxy Service is saved, as shown in [Figure 7–141](#).

**Figure 7-141 Proxy Service**



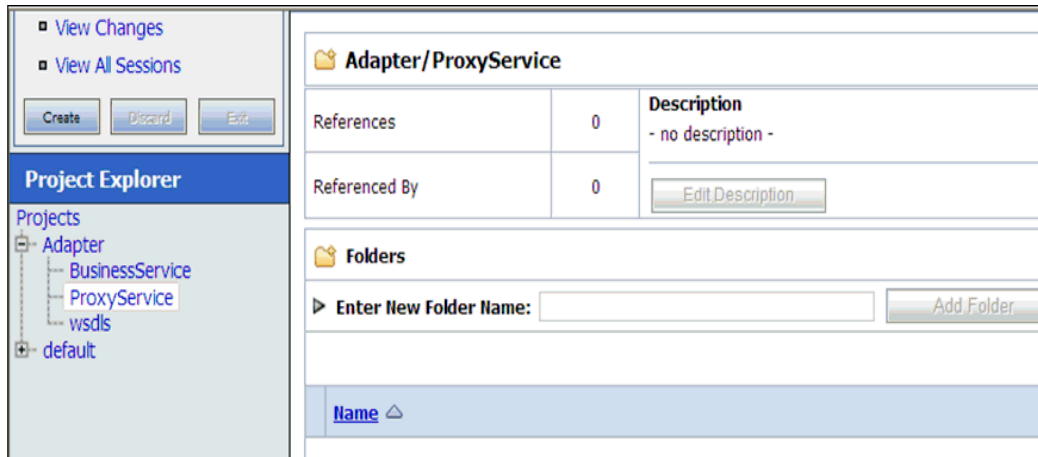
15. Click **Activate** in the left pane, and then **Submit** on the right pane, as shown in Figure 7-142.

**Figure 7-142 Activate Session**



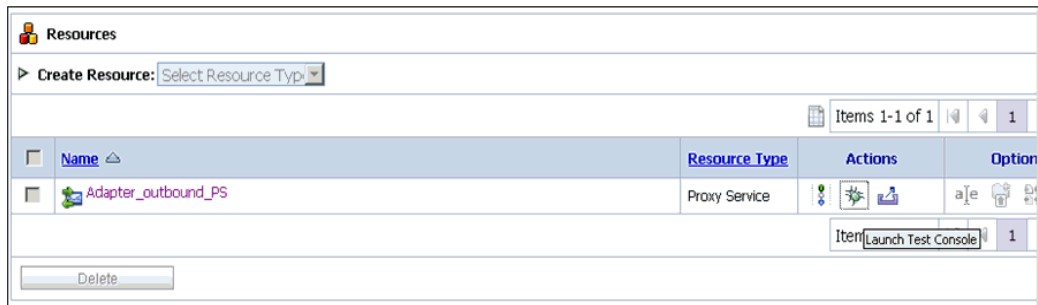
16. Click **ProxyService** in the Projects folder on the left pane, as shown in Figure 7-143.

**Figure 7–143 ProxyService**



17. Click the **Launch Test Console** icon for the created Proxy Service, as shown in Figure 7–144.

**Figure 7–144 Launch Test Console**



18. Uncheck the **Direct Call** check box, provide the input values for **Payload**, and click **Execute**.
19. Review the **Response Document**.



---

---

# Configuring an Outbound and Inbound Process for Oracle Service Bus Using JDeveloper

Oracle Application Adapter for PeopleSoft integrates seamlessly with Oracle JDeveloper to facilitate Web service integration.

This chapter contains the following sections:

- [Section 8.1, "Configuring an OSB Outbound Process Using JDeveloper \(J2CA Configuration\)"](#)
- [Section 8.2, "Configuring an OSB Inbound Process Using JDeveloper \(J2CA Configuration\)"](#)
- [Section 8.3, "Configuring an OSB Outbound Process Using JDeveloper \(BSE Configuration\)"](#)
- [Section 8.4, "Configuring a JMS Inbound Process Using JDeveloper \(J2CA Configuration\)"](#)
- [Section 8.5, "Configuring a JMS Outbound Process Using JDeveloper \(J2CA Configuration\)"](#)
- [Section 8.6, "Configuring an HTTP Outbound Process Using JDeveloper \(J2CA Configuration\)"](#)

## 8.1 Configuring an OSB Outbound Process Using JDeveloper (J2CA Configuration)

This section describes how to configure an OSB outbound process to your PeopleSoft system, using Oracle JDeveloper for J2CA configurations.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\OSB_
JDeveloper\J2CA\PeopleSoft_Sample_J2CA_OSB_Outbound_Project
```

This section includes the following topics:

- [Section 8.1.1, "Creating a Service Bus Application for OSB"](#)
- [Section 8.1.2, "Defining an OSB Outbound Process"](#)
- [Section 8.1.3, "Deploying the OSB Outbound Process"](#)

### Prerequisites

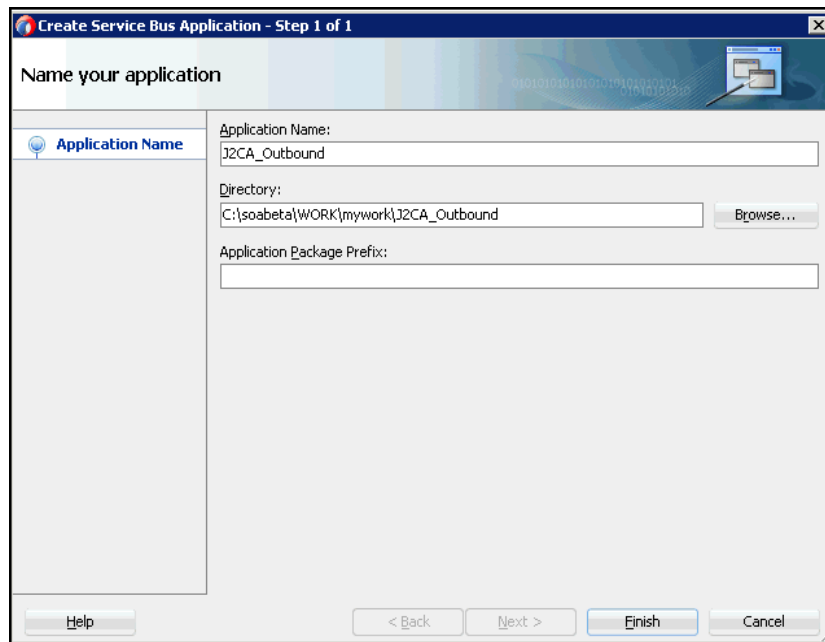
Before you design an OSB outbound process, you must generate the respective WSDL file using Application Explorer. For more information, see [Section 4.4.1, "Generating WSDL for Request/Response Service"](#) on page 4-8.

## 8.1.1 Creating a Service Bus Application for OSB

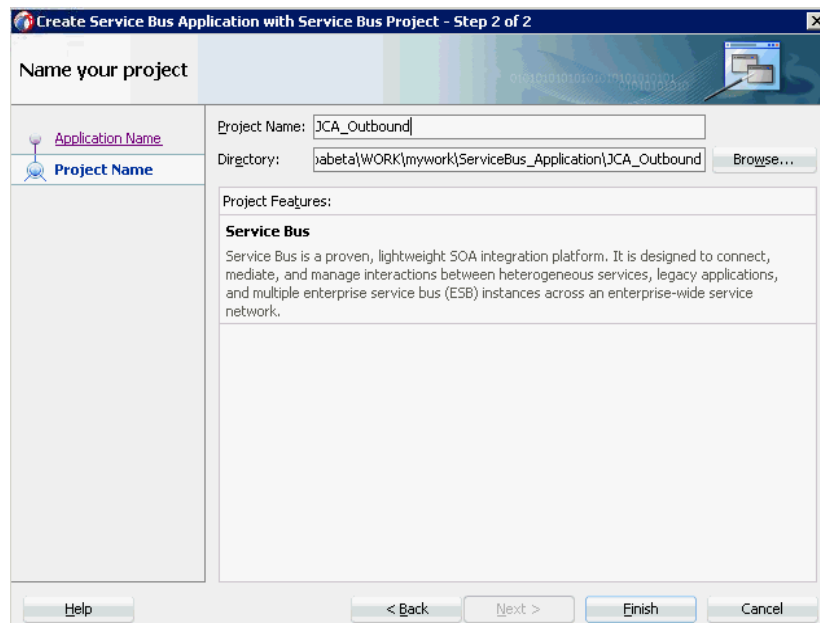
Perform the following steps in JDeveloper to create a service bus application for OSB.

1. Create a new OSB application.
2. Enter a name for the OSB Application (for example, J2CA\_Outbound) and click **Finish**, as shown in [Figure 8-1](#).

**Figure 8-1 Name Your Application Pane**



3. Enter a project name (for example, JCA\_Outbound), and click **Finish**, as shown in [Figure 8-2](#).

**Figure 8–2 Name Your Project Pane**

## 8.1.2 Defining an OSB Outbound Process

This section describes how to define an OSB outbound process. The following topics are included:

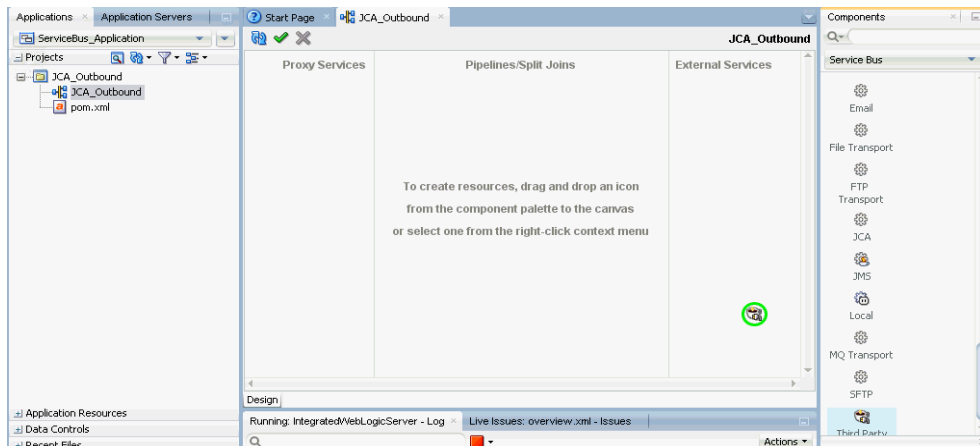
- [Section 8.1.2.1, "Configuring a Third-Party Adapter Service Component"](#)
- [Section 8.1.2.2, "Configuring a File Transport Type Business Service"](#)
- [Section 8.1.2.3, "Creating a Proxy Service With Pipeline"](#)
- [Section 8.1.2.4, "Configuring the Routing Rules"](#)

### 8.1.2.1 Configuring a Third-Party Adapter Service Component

Perform the following steps to create a third party adapter service component along with the Business Service:

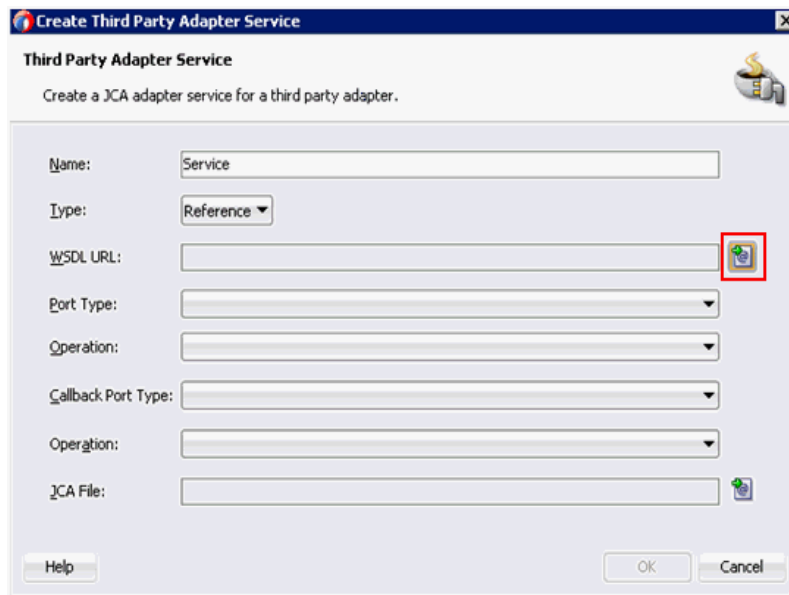
1. Drag and drop the **Third Party Adapter** component from the Service Bus Components pane to the External Services pane, as shown in [Figure 8–3](#).

**Figure 8–3 Third Party Adapter Component**



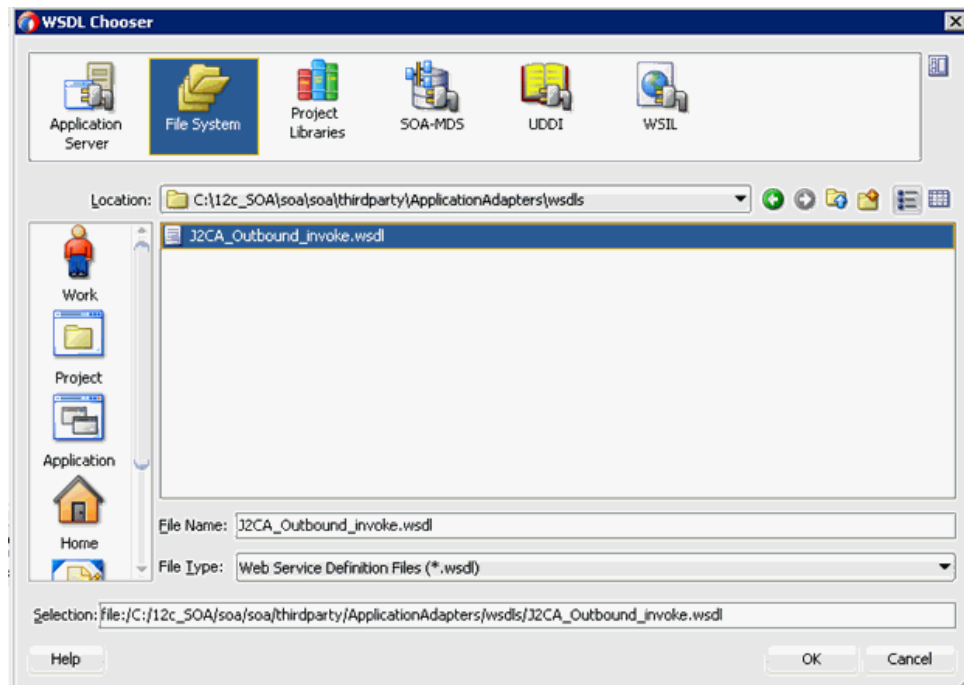
The Create Third Party Adapter Service dialog is displayed, as shown in [Figure 8–4](#).

**Figure 8–4 Create Third Party Adapter Service Pane**

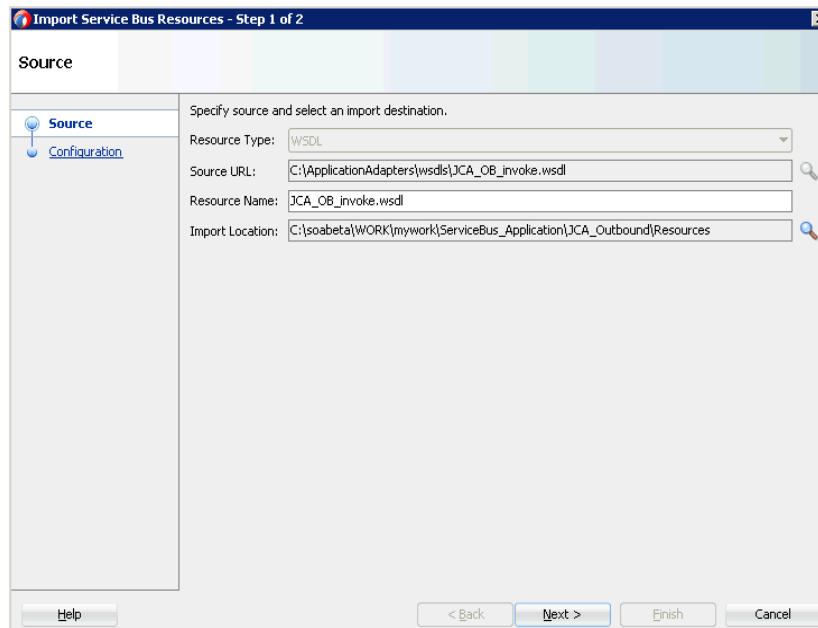


2. Enter an appropriate name for the Third Party Adapter Service which will be used as the Business Service name.
3. Ensure that **Reference** is selected from the Type drop-down list (by default).
4. Click the Find existing WSDLs icon, which is located to the right of the WSDL URL field.

The WSDL Chooser dialog is displayed, as shown in [Figure 8–5](#).

**Figure 8–5 WSDL Chooser Dialog**

5. Select the **File System** tab, then browse, and select an outbound WSDL file from the WSDL directory.
6. Click **OK**.  
The Import Service Bus Resources dialog is displayed.
7. Click **Next**, as shown in [Figure 8–6](#).

**Figure 8–6 Source Pane**

8. In the Configuration pane, click **Finish**.

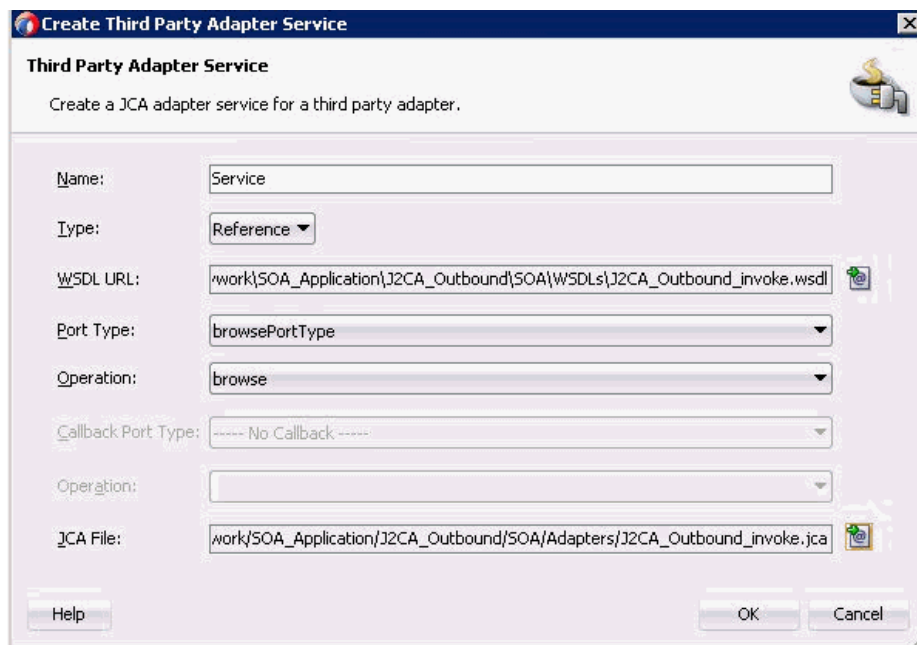
You are returned to the Create Third Party Adapter Service Dialog.

9. Click the Find JCA file icon which is located to the right of the JCA File field.  
The Transformation Chooser dialog is displayed.
10. Select the JCA properties file from the WSDL directory.
11. Click **OK**. The Copy File message is displayed.
12. Click **Yes**.

A copy of the JCA properties file is made in the project folder.

You are returned to the Create Third Party Adapter Service dialog, as shown in [Figure 8-7](#).

**Figure 8-7 Create Third Party Adapter Service Dialog**



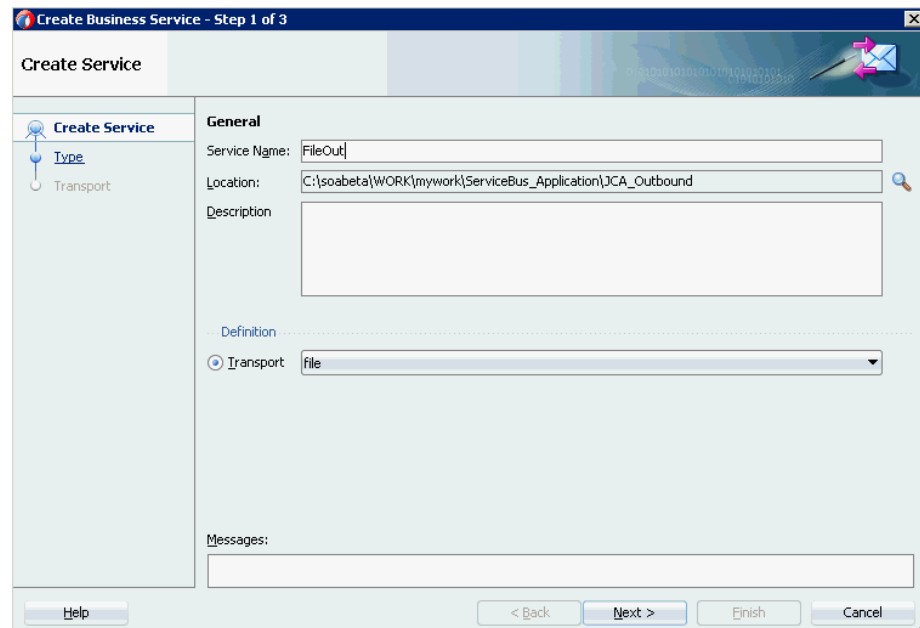
13. Click **OK**.

The Business service component is created in the External Services pane.

### 8.1.2.2 Configuring a File Transport Type Business Service

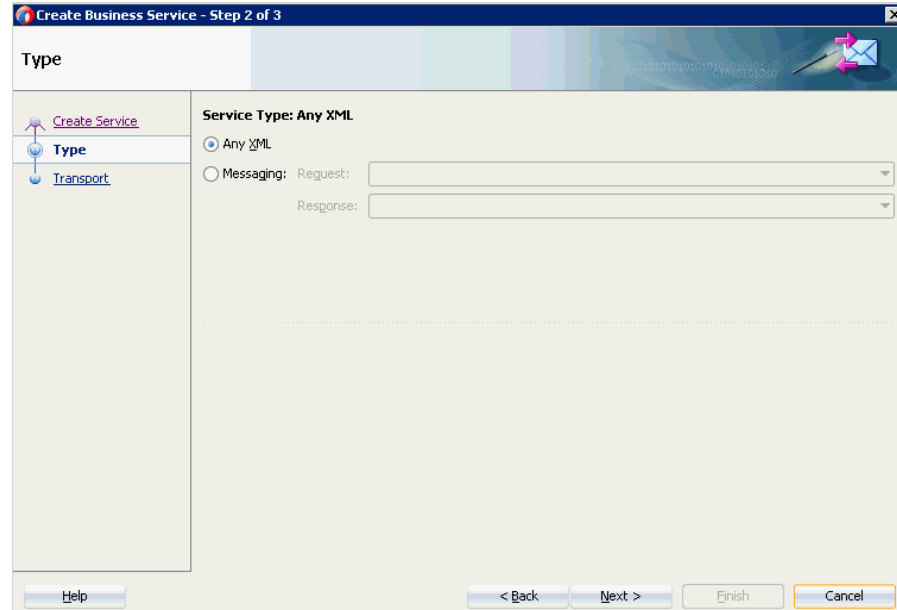
Perform the following steps to create a File Transport Business Service:

1. Drag and drop the **File Transport** component from the Advanced pane to the External Services pane.  
The Create Business Service dialog is displayed.
2. In the Service Name field, enter any name you wish for the Business Service (for example, FileOut), and click **Next**, as shown in [Figure 8-8](#).

**Figure 8–8 Create Service Pane**

The Type pane is displayed. The **Any XML** option is selected by default.

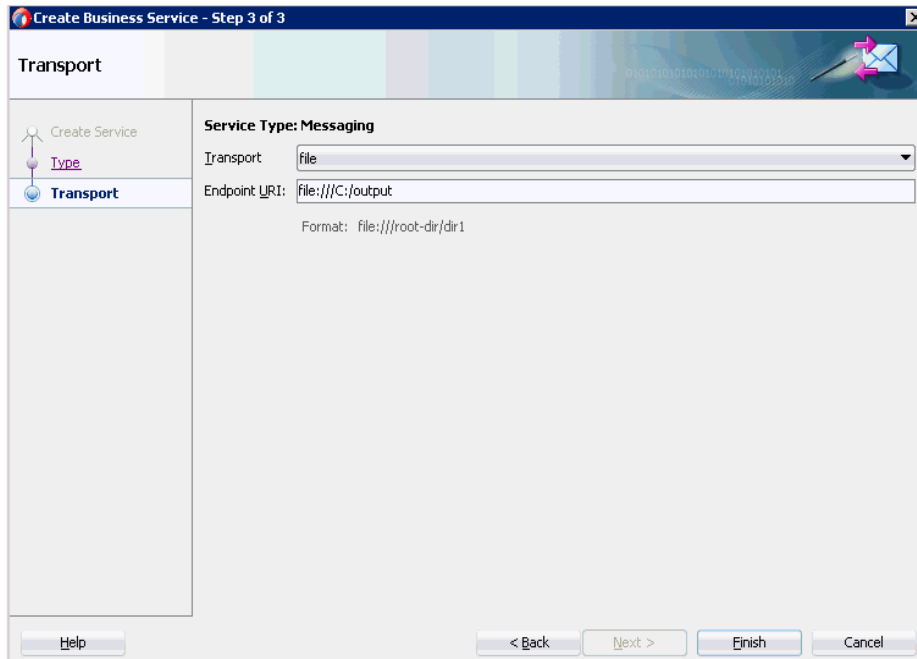
3. Click **Next**, as shown in [Figure 8–9](#).

**Figure 8–9 Type Pane**

The Transport pane appears.

4. Provide the output location in the Endpoint URI field (for example, c:/output) and click **Finish**, as shown in [Figure 8–10](#).

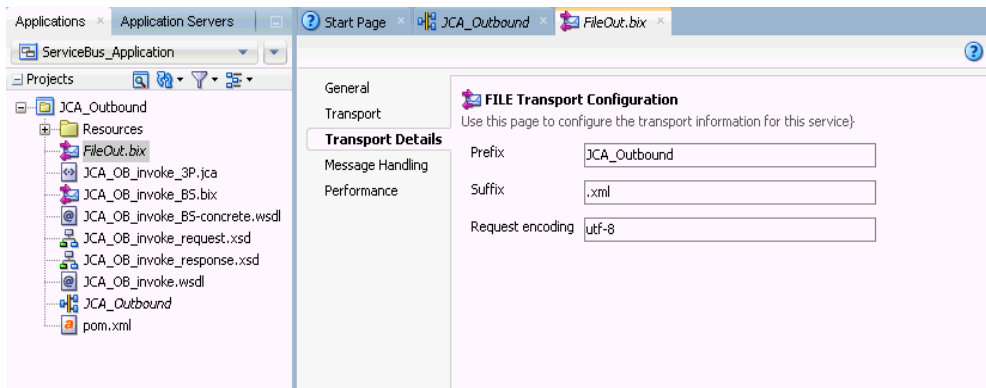
**Figure 8–10 Transport Pane**



The File Transport Business service Fileout is created and displayed.

5. Double-click the created Business service **Fileout** and provide the values for the Prefix and Suffix fields in the Transport Details Tab, as shown in [Figure 8–11](#).

**Figure 8–11 Transport Details**



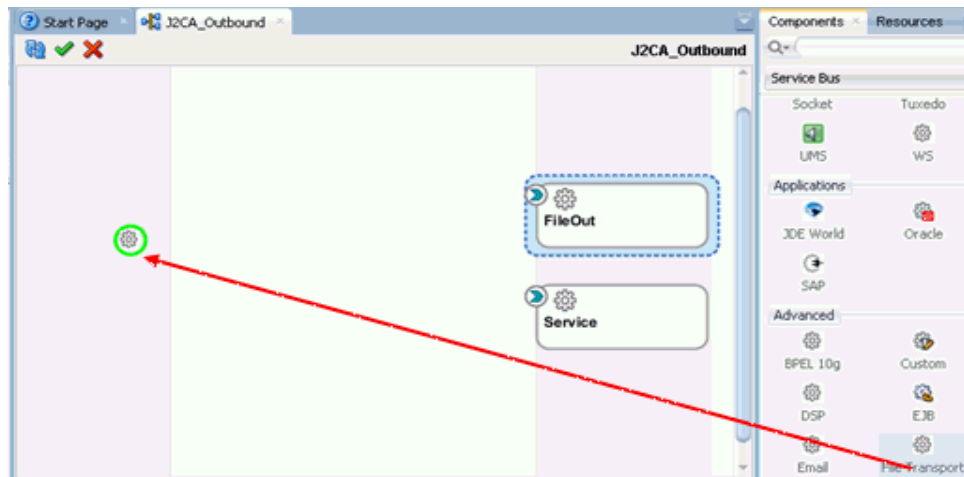
### 8.1.2.3 Creating a Proxy Service With Pipeline

Perform the following steps to create a Proxy Service with Pipeline:

1. Drag and drop the **File Transport** component from the Advanced Components pane to the Proxy Services pane, as shown in [Figure 8–12](#).



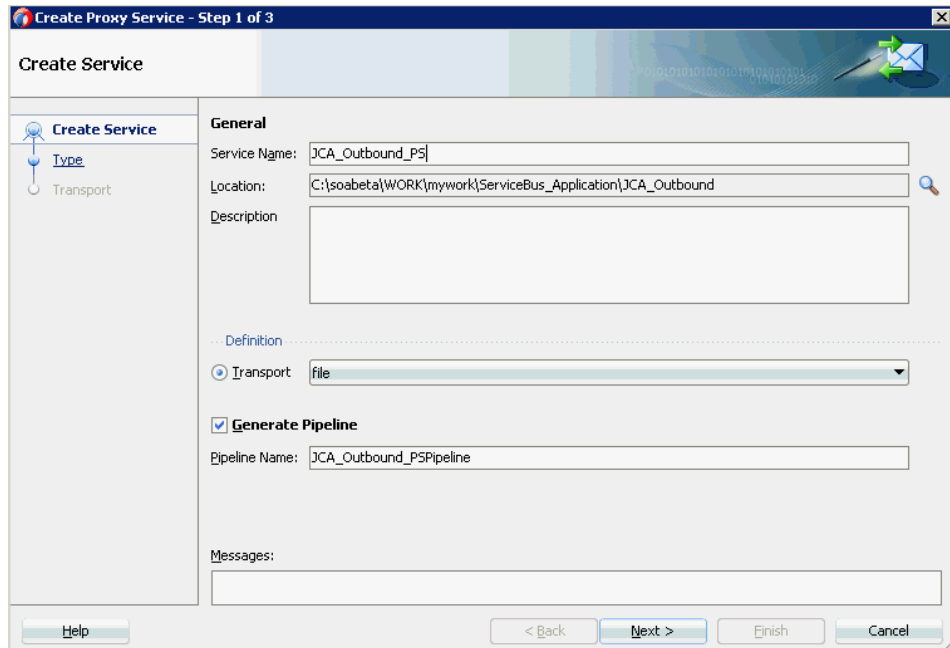
**Figure 8–12 File Transport Component**



The Create Proxy Service pane is displayed.

2. In the Service Name field, enter any name you wish for the Proxy service (for example, JCA\_Outbound\_PS). By default, **Generate Pipeline** is selected.
3. Click **Next**, as shown in [Figure 8–13](#).

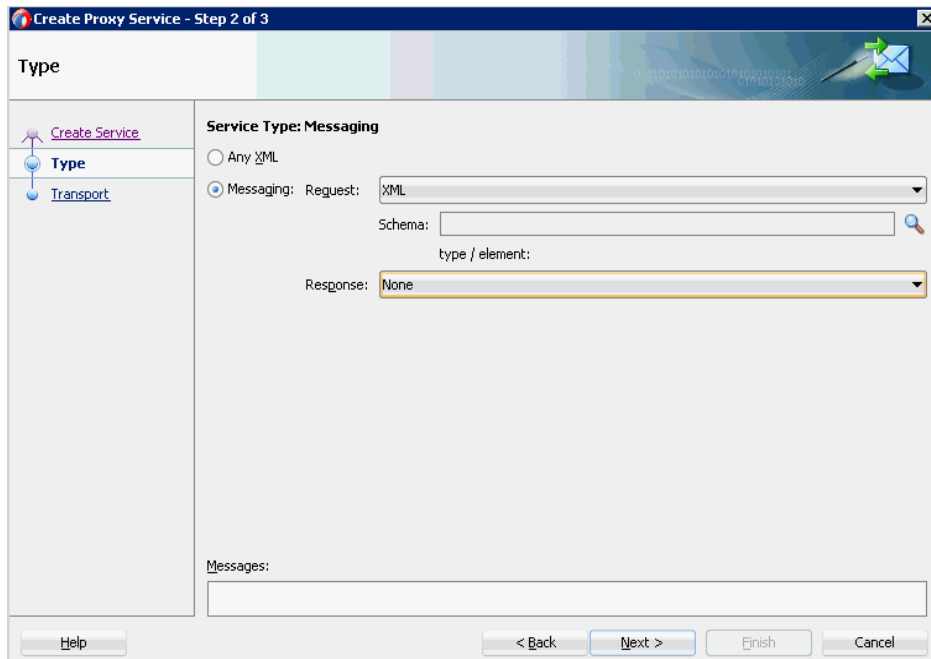
**Figure 8–13 Create Service Pane**



The Type pane is displayed.

4. Select the **Messaging** option, set the Request to **XML** and Response as **None**, and then click **Next**, as shown in [Figure 8–14](#).

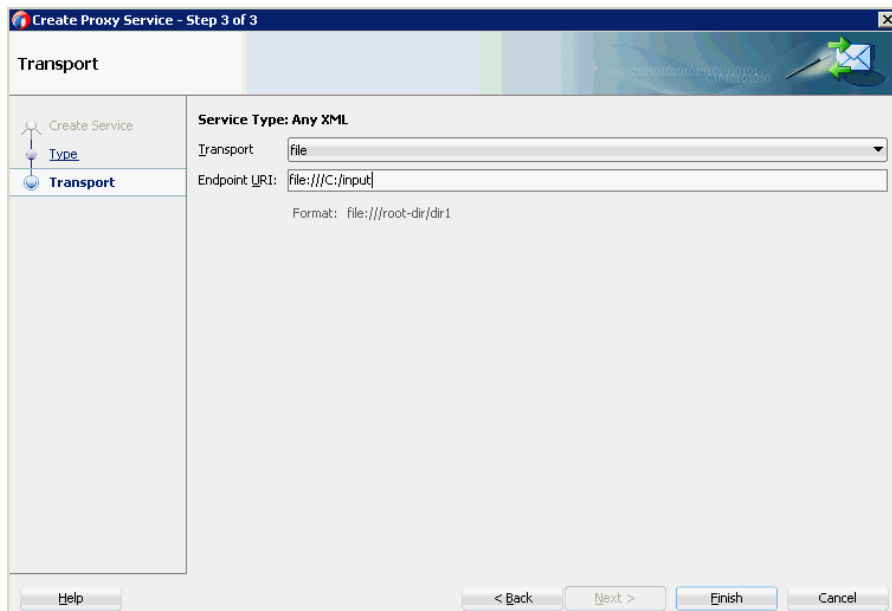
**Figure 8–14 Type Pane**



The Transport window is displayed.

5. Provide the input location in the Endpoint URI field (for example, c:/input) and click **Finish**, as shown in [Figure 8–15](#).

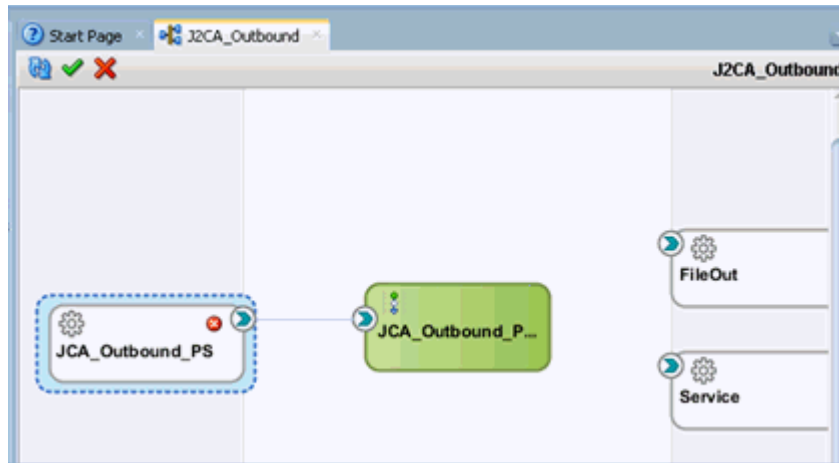
**Figure 8–15 Transport Window**



The Proxy service along with the pipeline is created and displayed.

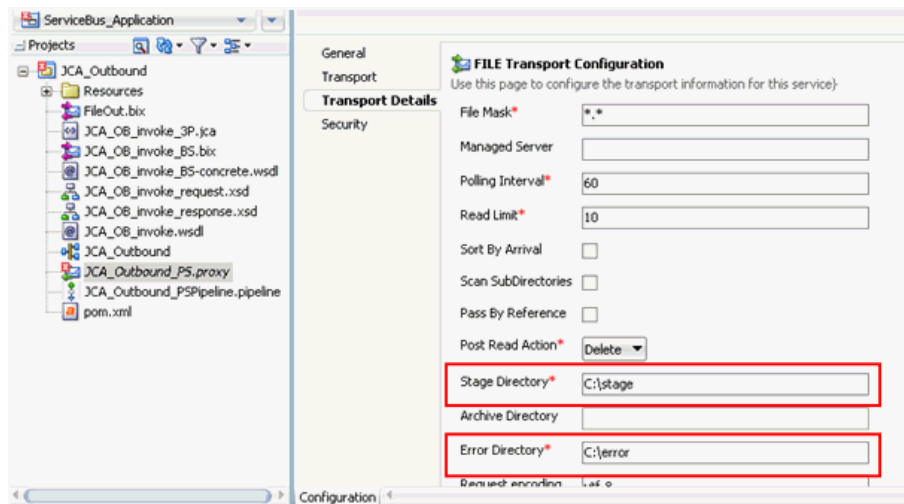
6. Double-click the created Proxy Service (for example: JCA\_Outbound\_PS), as shown in [Figure 8–16](#).

**Figure 8–16 Proxy Service Edit**



7. In the displayed Proxy Service configuration page, select **Transport Details** and provide the values for Stage and Error Directory, as shown in [Figure 8–17](#).

**Figure 8–17 File Transport Configuration**



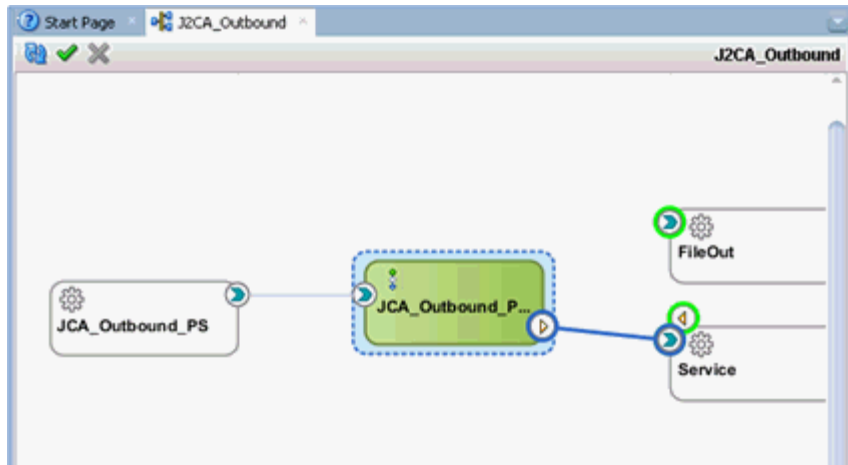
8. Save and close the Proxy Service configuration page.

### 8.1.2.4 Configuring the Routing Rules

Perform the following steps to configure the routing rules:

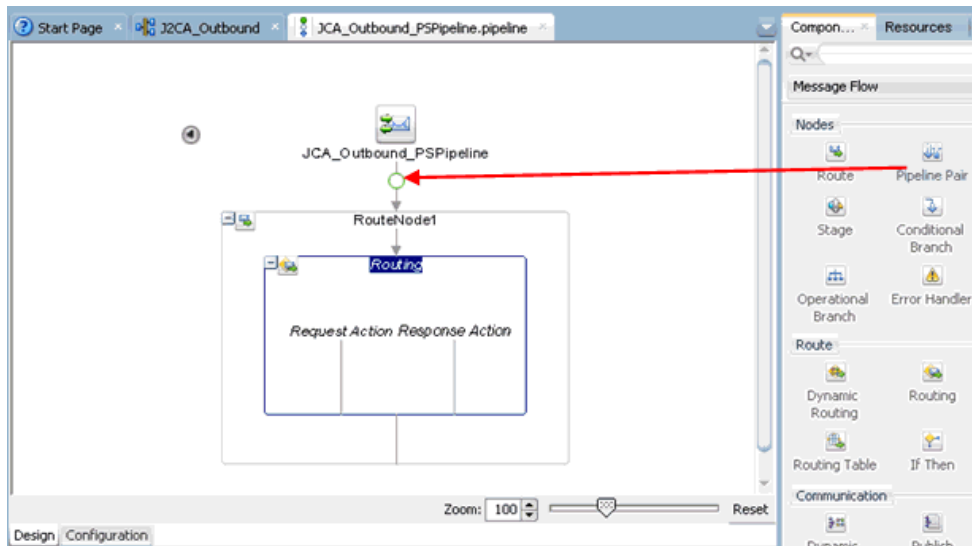
1. Connect the Pipeline to the Business Service (for example, Service) as shown in [Figure 8–18](#).

**Figure 8–18 Business Service Pipeline**



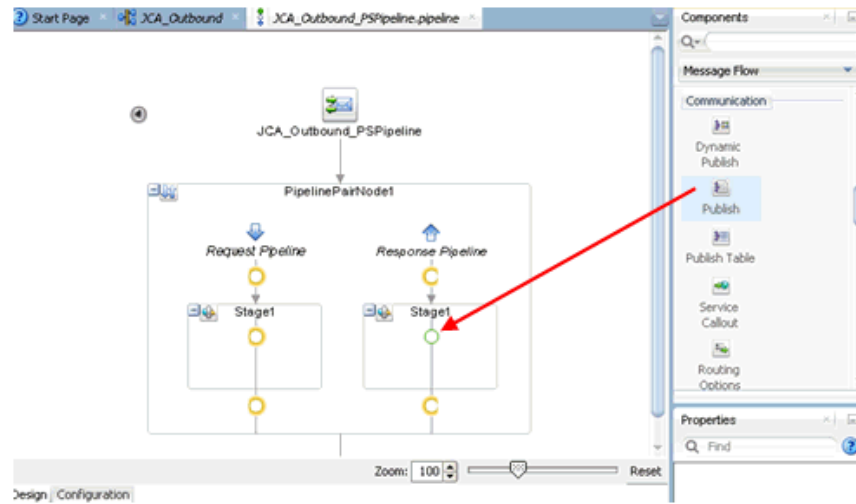
2. Double-click on the pipeline (for example, JCA\_Outbound\_PSPipeline) in the Pipelines/Split Joins pane.  
The Pipeline configuration page is displayed.
3. Drag and drop the **Pipeline Pair** node from Nodes pane to the area below the Pipeline (for example: JCA\_Outbound\_PSPipeline), as shown in [Figure 8–19](#).

**Figure 8–19 Pipeline Pair Node**



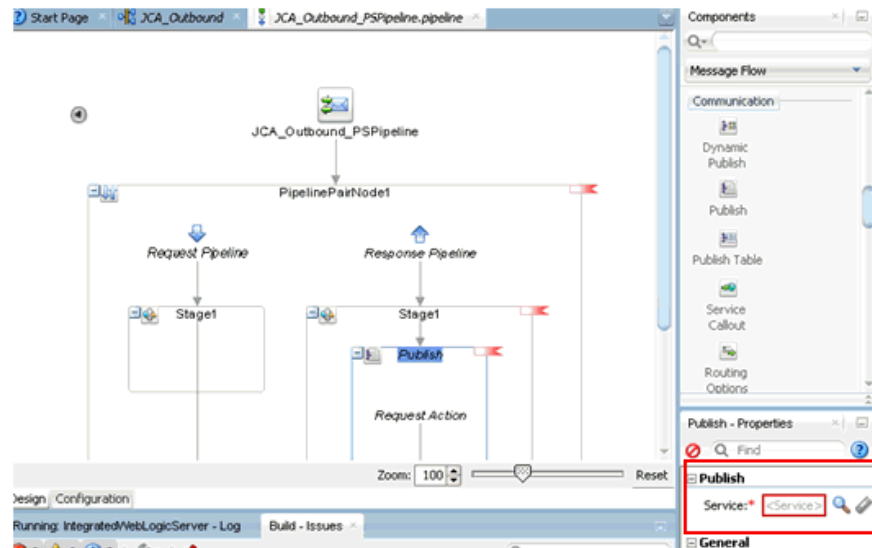
4. Drag and drop the **Publish** node from the Communication pane to the area beneath Stage1 of the Response Pipeline, as shown in [Figure 8–20](#).

**Figure 8–20 Publish Node**



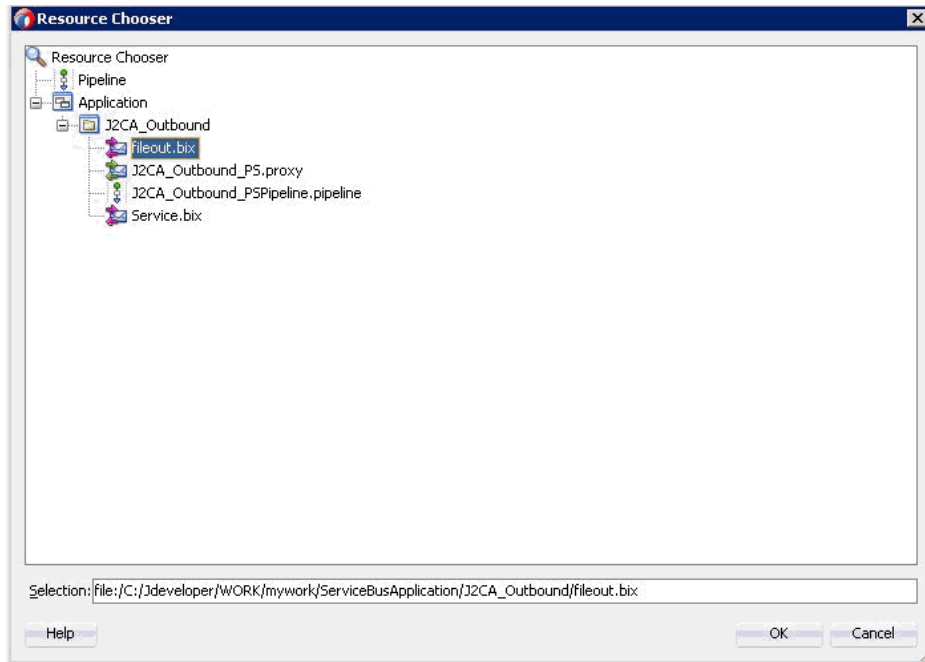
5. Click on the browse icon to the right of the Service field in the right pane of Publish Properties, as shown in Figure 8–21.

**Figure 8–21 Browse Icon**



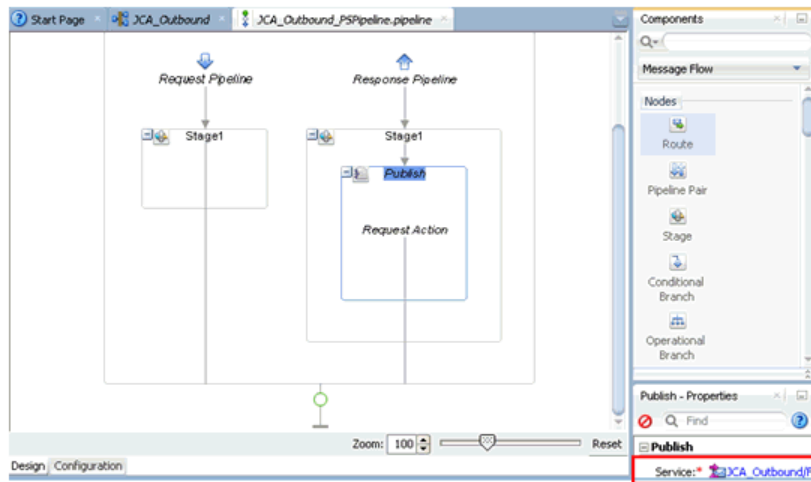
6. In the displayed Resource Chooser window, select the **Fileout.bix** File Transport Business service and click **OK**, as shown in Figure 8–22.

**Figure 8–22 Resource Chooser**



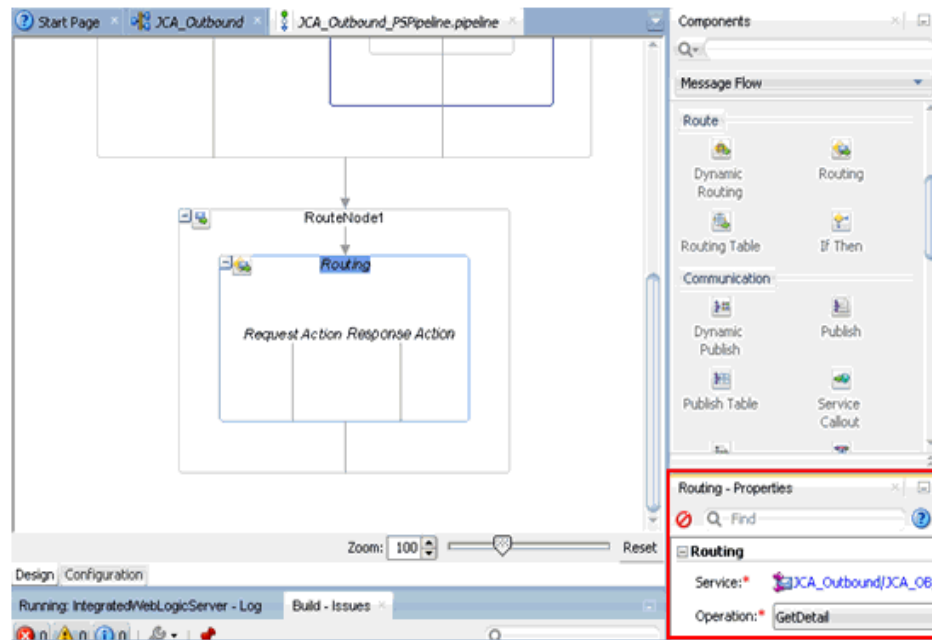
In the right pane, the selected service is configured in the Publish pane, as shown in Figure 8–23.

**Figure 8–23 Publish Pane**



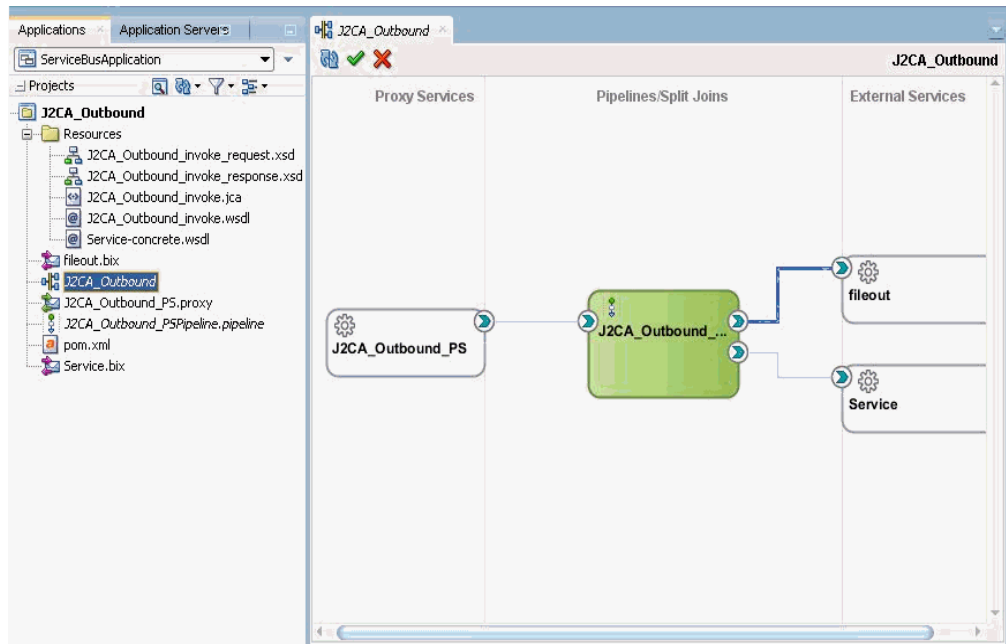
7. Click on the Routing to verify the Service is selected properly, as shown in Figure 8–24.

**Figure 8–24 Pipeline Configuration**



8. Save and Close the Pipeline configuration page.
9. Double-click the overview.xml file (for example: J2CA\_Outbound), and click **Save All** in the menu bar to save the OSB process, as shown in [Figure 8–25](#).

**Figure 8–25 Save All Icon**

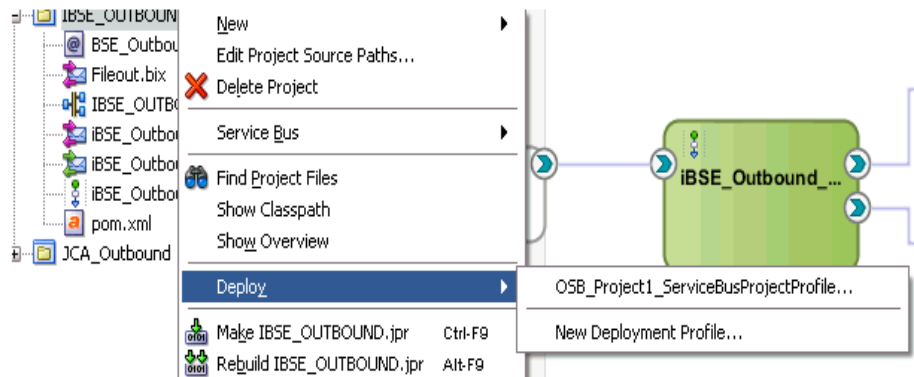


### 8.1.3 Deploying the OSB Outbound Process

Perform the following steps to deploy the OSB outbound process.

1. Right-click the OSB project, select **Deploy**, and then select **OSB\_Project1\_ServiceBusProjectProfile...**, as shown in [Figure 8–26](#).

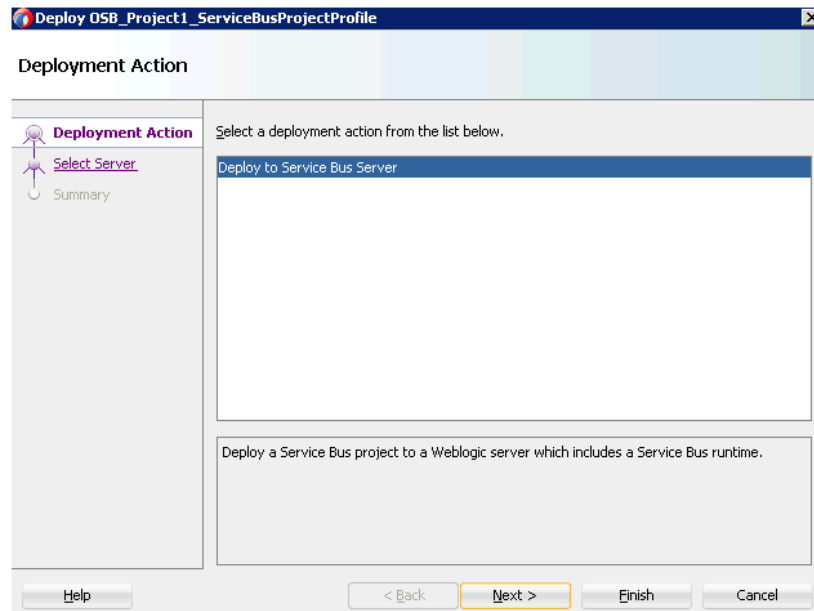
**Figure 8–26 Deploy Option**



The Deployment Action page is displayed.

2. Click **Next**, as shown in [Figure 8–27](#).

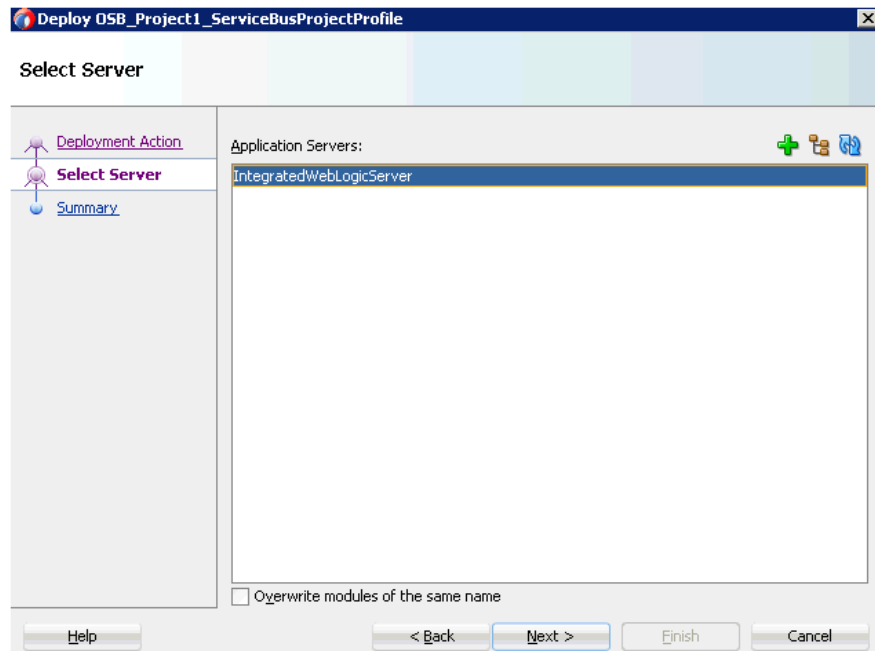
**Figure 8–27 Deployment Action Page**



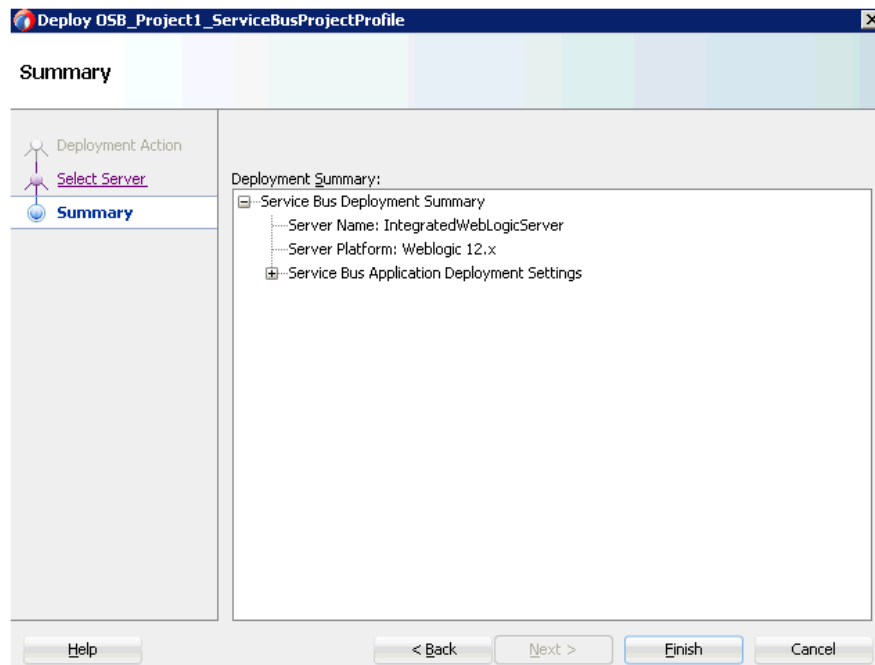
The Select Server page is displayed.

3. Select an available application server that was configured and click **Next**, as shown in [Figure 8–28](#).



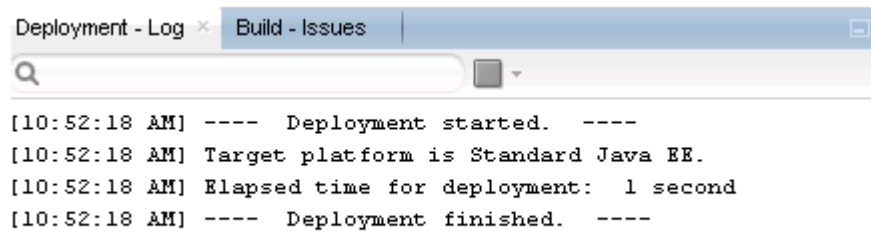
**Figure 8–28 Select Server Page**

The Summary page is displayed, as shown in [Figure 8–29](#).

**Figure 8–29 Summary Page**

4. Review and verify all the available deployment information for your project and click **Finish**.

The process is deployed successfully, as shown in [Figure 8–30](#).

**Figure 8–30 Successful Deployment Message**


```

Deployment - Log x Build - Issues
[10:52:18 AM] ---- Deployment started. ----
[10:52:18 AM] Target platform is Standard Java EE.
[10:52:18 AM] Elapsed time for deployment: 1 second
[10:52:18 AM] ---- Deployment finished. ----

```

5. Copy and paste an input XML file in the input folder you have configured (for example, C:\input).

The output is received in the configured output location (for example, C:\output).

## 8.2 Configuring an OSB Inbound Process Using JDeveloper (J2CA Configuration)

This section describes how to configure an OSB inbound process to your PeopleSoft system, using Oracle JDeveloper for J2CA configurations.

A sample project has been provided for this inbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\OSB_
JDeveloper\J2CA\PeopleSoft_Sample_J2CA_OSB_Inbound_Project
```

This section includes the following topics:

- [Section 8.2.1, "Creating a Service Bus Application for OSB"](#)
- [Section 8.2.2, "Defining an OSB Inbound Process"](#)
- [Section 8.2.3, "Deploying the OSB Inbound Process"](#)

### Prerequisites

Before you design an OSB inbound process, you must generate the respective WSDL file using Application Explorer. For more information, see [Section 4.5.1, "Generating WSDL for Event Integration"](#) on page 4-34.

### 8.2.1 Creating a Service Bus Application for OSB

To configure an OSB inbound process, you must create service bus application for OSB. For more information, see [Section 8.1.1, "Creating a Service Bus Application for OSB"](#) on page 8-2.

### 8.2.2 Defining an OSB Inbound Process

This section describes how to define an OSB inbound process. The following topics are included:

- [Section 8.2.2.1, "Configuring a Third-Party Adapter Service Component"](#)
- [Section 8.2.2.2, "Creating a Pipeline"](#)
- [Section 8.2.2.3, "Configuring a File Transport Type Business Service"](#)

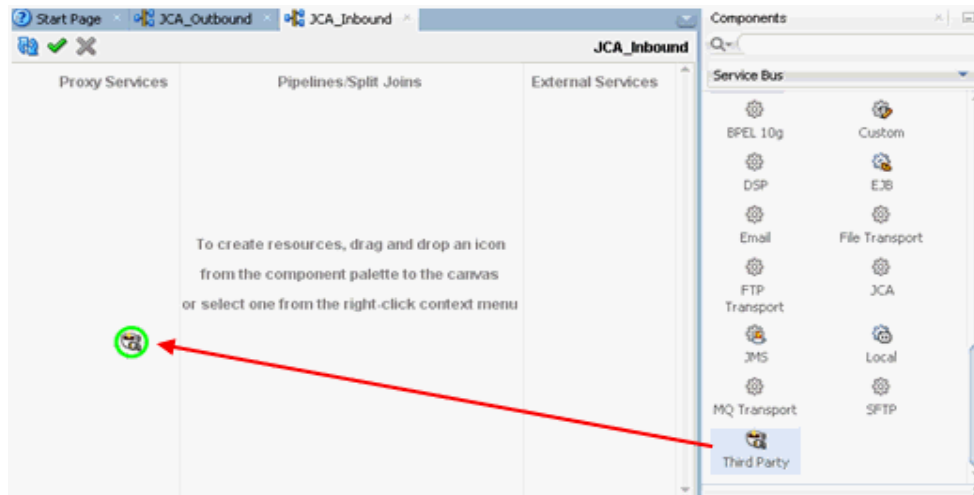
- [Section 8.2.2.4, "Configuring the Routing Rules"](#)

### 8.2.2.1 Configuring a Third-Party Adapter Service Component

Perform the following steps to create a third party adapter service component:

1. Drag and drop the **Third Party** adapter component from the Service Bus Components Pane to the Proxy Services, as shown in [Figure 8–31](#).

**Figure 8–31 Third Party Adapter Service Component**



The Create Third Party Adapter Service dialog is displayed.

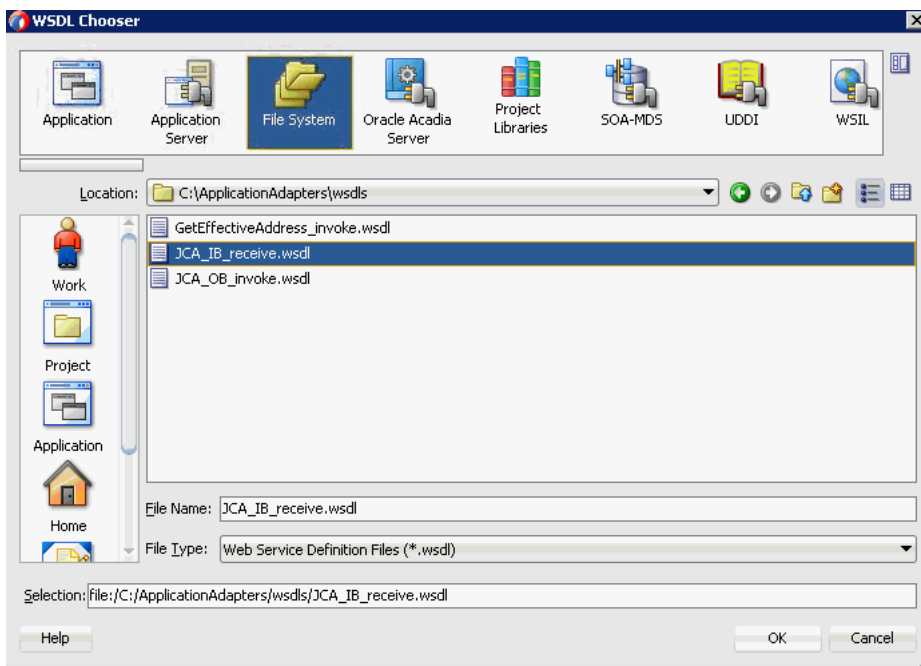
2. Enter any name you wish for the Third Party Adapter Service or leave it to the default value.
3. Ensure that **Service** is selected from the Type drop-down list (by default).
4. Click the Find existing WSDLs icon, which is located to the right of the WSDL URL field, as shown in [Figure 8–32](#).

**Figure 8–32 Third Party Adapter Service Dialog**



The WSDL Chooser dialog is displayed, as shown in [Figure 8–33](#).

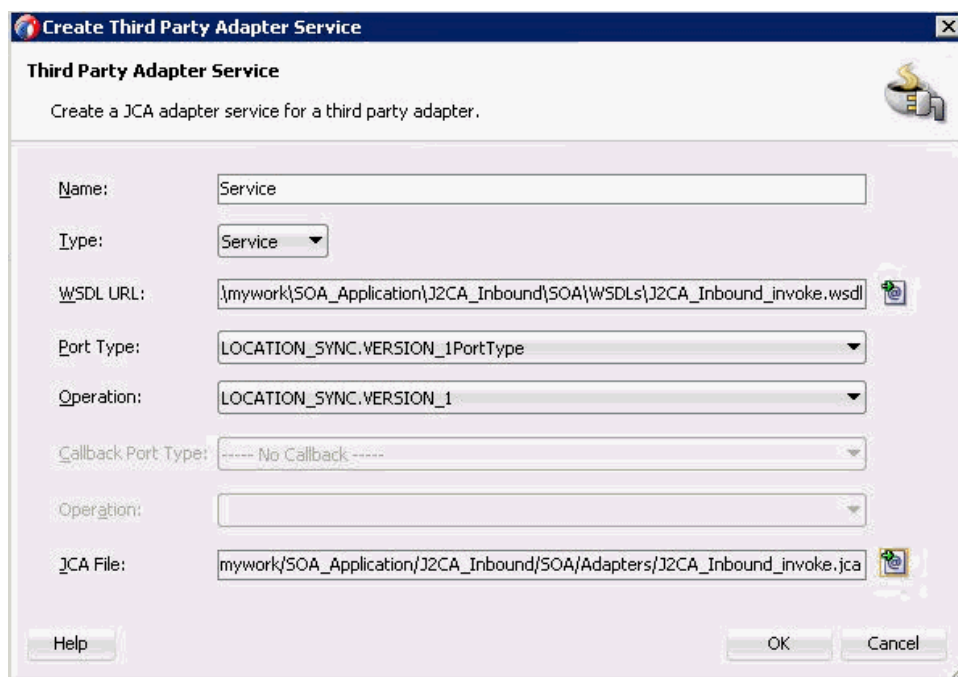
**Figure 8–33 WSDL Chooser Dialog**



5. Select the File system folder, then browse and select an inbound WSDL file from the WSDL directory.
6. Click **OK**.  
The Import Service Bus Resources dialog is displayed.
7. Click **Next**.

8. In the Configuration window, click **Finish**.  
You are returned to the Create Third Party Adapter Service dialog.
9. Click the Find JCA file icon, which is located to the right of the JCA File field.  
The Transformation Chooser dialog is displayed.
10. Select the JCA properties file from the WSDL directory.
11. Click **OK**.  
The Copy File message is displayed.
12. Click **Yes**.  
A copy of the JCA properties file is created in the project folder.  
You are returned to the Create Third Party Adapter Service dialog, as shown in [Figure 8–34](#).

**Figure 8–34 Create Third Party Adapter Service Dialog**



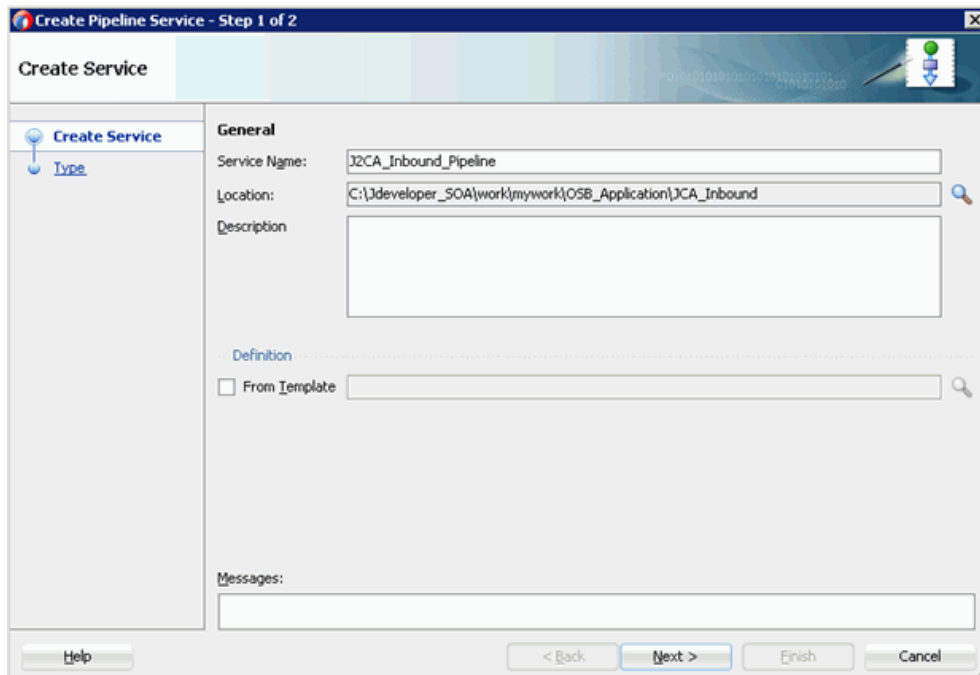
13. Click **OK**.  
The third party adapter service component is created in the Proxy Services pane.

### 8.2.2.2 Creating a Pipeline

Perform the following steps to generate inbound proxy service with Pipeline:

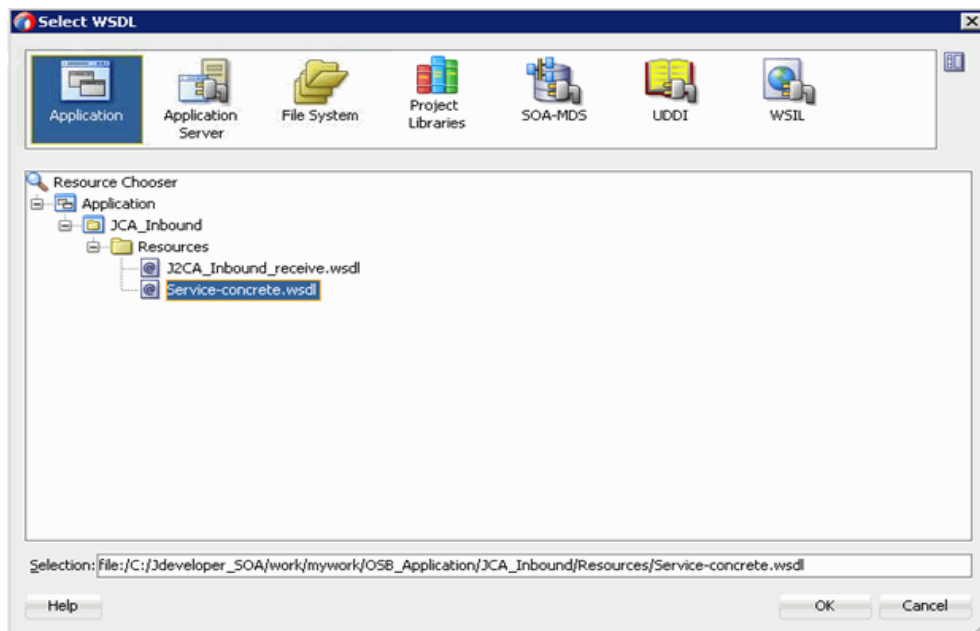
1. Under Service Bus, click **Resources**.
2. Drag and drop the Pipeline to the Pipelines/Split Joins pane.
3. Provide a name for the Pipeline and click next, as shown in [Figure 8–35](#).

**Figure 8–35 Create Service Page**



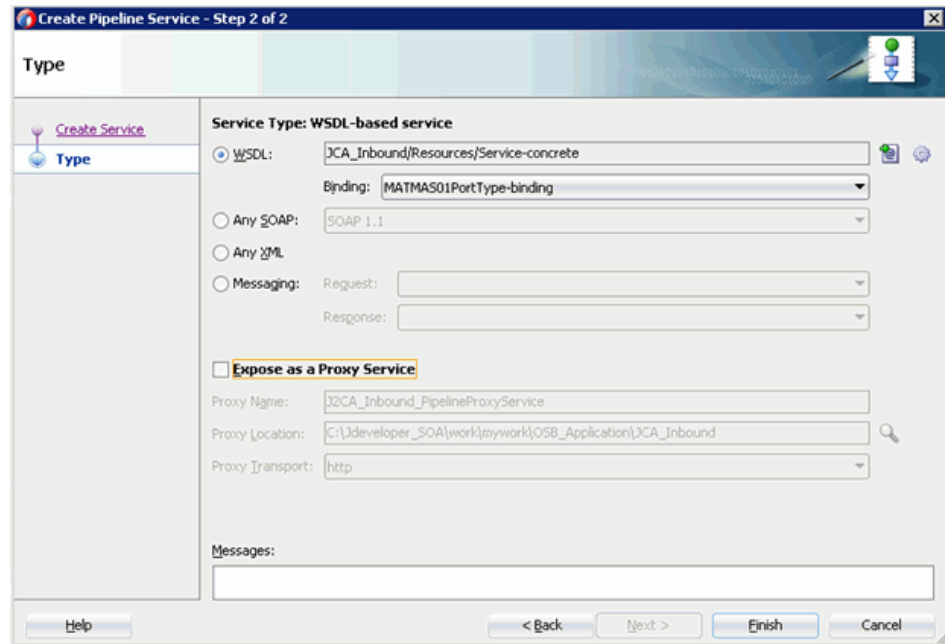
4. In the Create Pipeline Service window, select the **WSDL** option and click on the WSDL URL.
5. Select **Application** in the WSDL chooser window, then select **service-concrete.wsdl** in the appropriate OSB project, and then click **OK**, as shown in [Figure 8–36](#).

**Figure 8–36 Select WSDL Page**



6. Clear the Expose as a Proxy Service check box and click **Finish**, as shown in [Figure 8–37](#).

Figure 8–37 Type Page



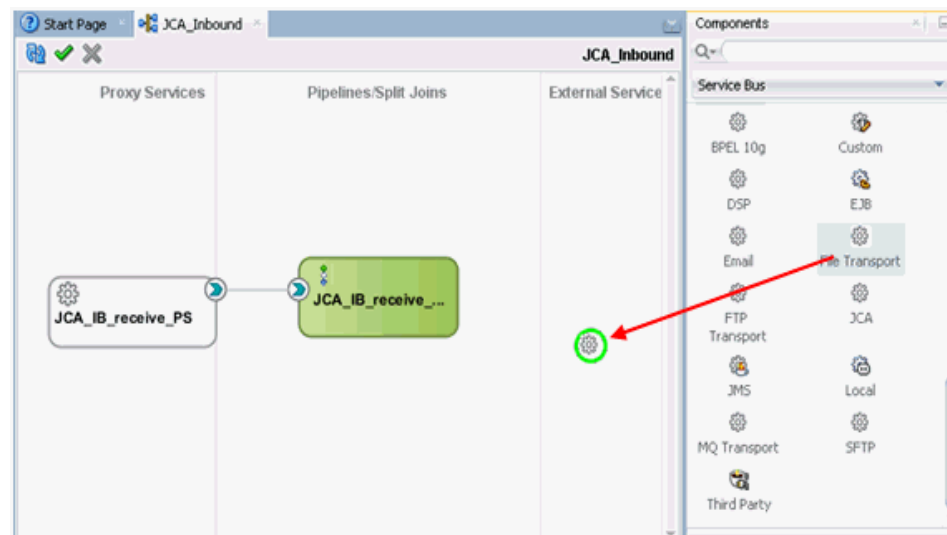
7. Drag and drop the Proxy Service to the Pipelines/Split Joins pane.

### 8.2.2.3 Configuring a File Transport Type Business Service

Perform the following steps to create the File Transport Type Business Service:

1. Drag and drop the **File Transport** component from the Advanced pane to the External Services pane, as shown in Figure 8–38.

Figure 8–38 File Transport Node

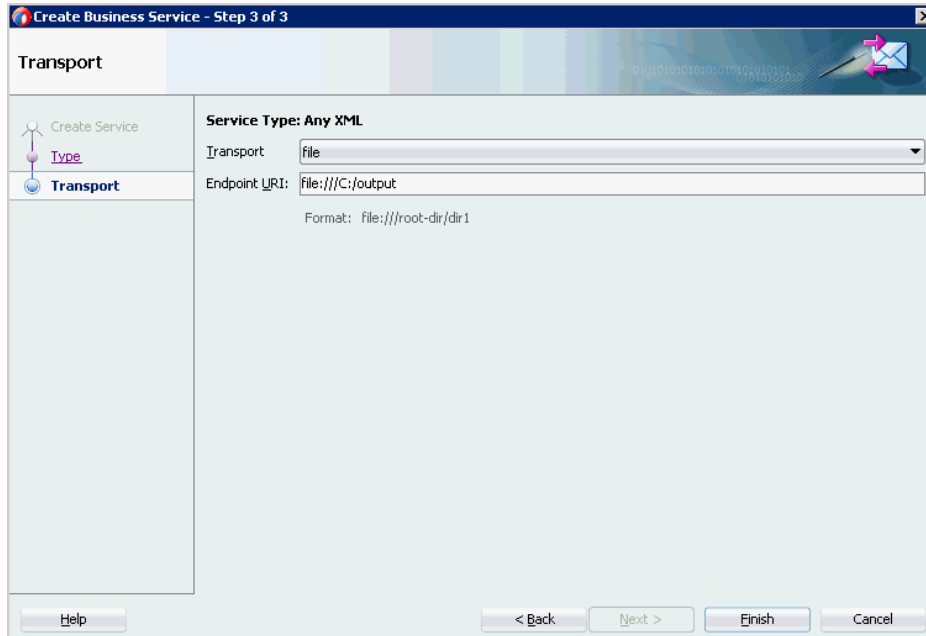


The Create Business Service dialog is displayed.

2. In the Service Name field, enter any name you wish for the Business Service (for example, FileOut), and click **Next**.  
In the displayed Type Window, the Any XML option is selected by default.

3. Click **Next**.
4. In the displayed Transport window, provide the output location in the Endpoint URI field (for example, c:\output), and click **Finish**, as shown in [Figure 8-39](#).

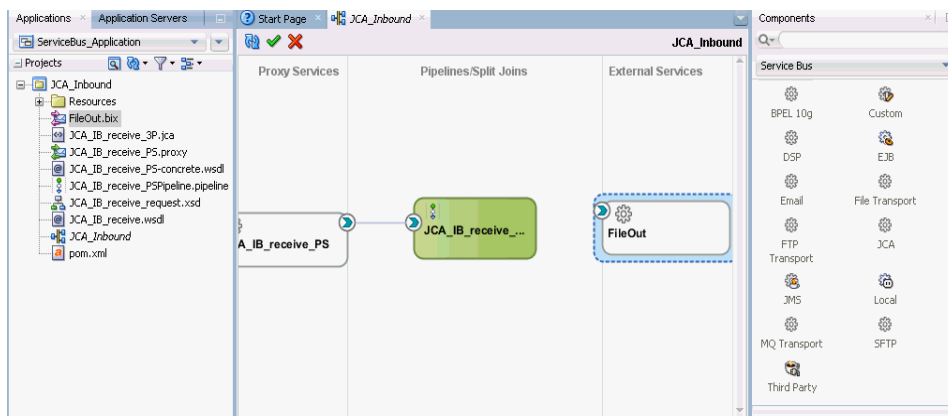
**Figure 8-39 Transport Pane**



The FileOut Business service is created.

5. Double-click the FileOut Business service, as shown in [Figure 8-40](#).

**Figure 8-40 FileOut Business Service**

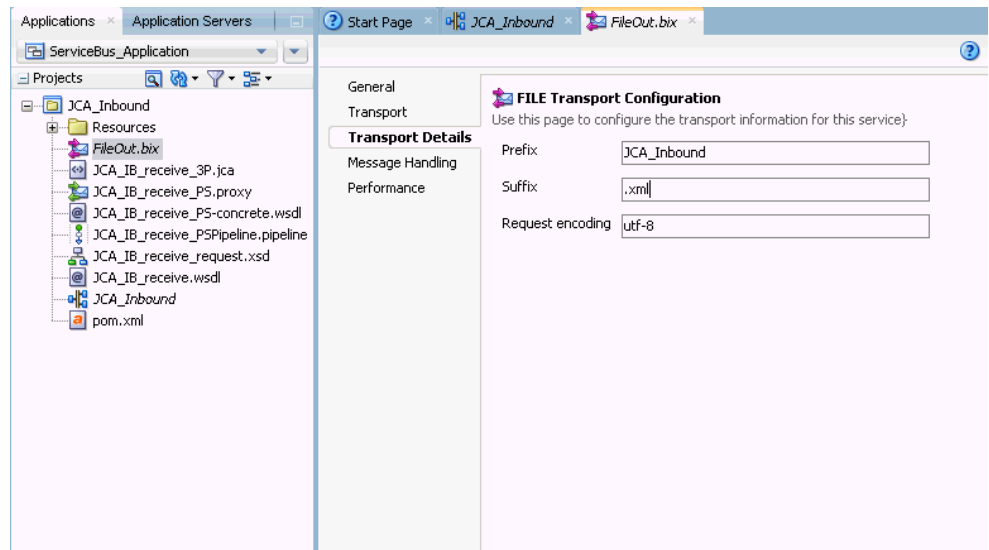


The Configuration page is displayed.

6. Navigate to the Transport Details tab and provide the values for the Prefix and Suffix fields, as shown in [Figure 8-41](#).



**Figure 8–41 File Transport Configuration**



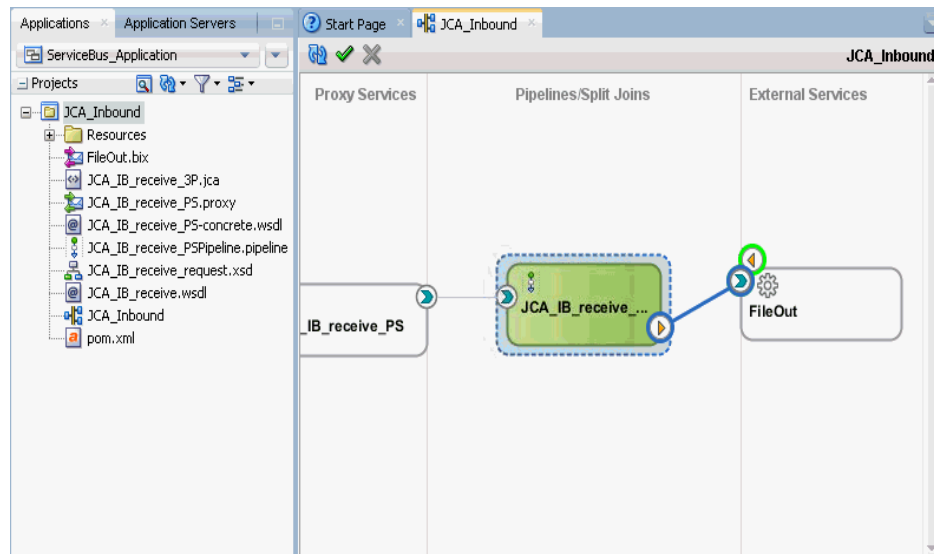
7. Save and close the Configuration page.

### 8.2.2.4 Configuring the Routing Rules

Perform the following steps to configure the routing rules.

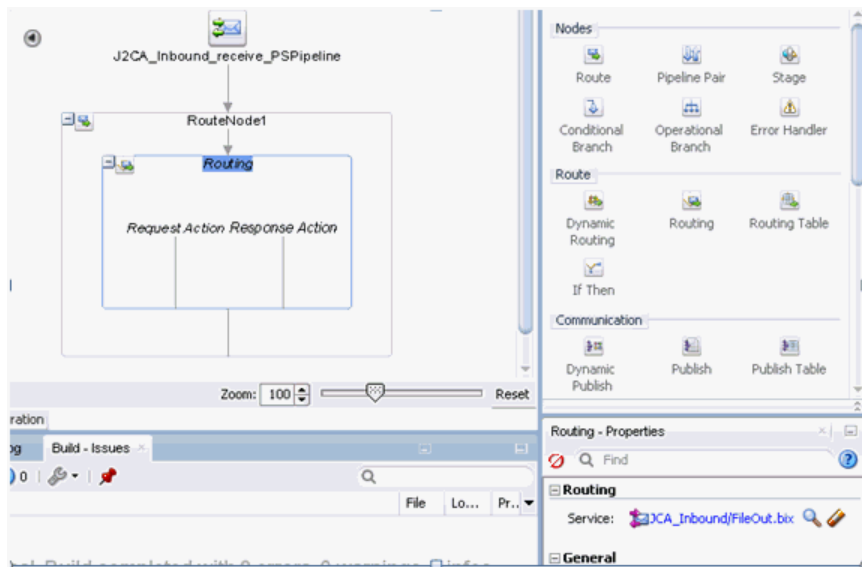
1. Create a connection between the Pipeline (for example, JCA\_IB\_receive\_PSPipeline) and the File Type Business Service (for example, FileOut), as shown in [Figure 8–42](#).

**Figure 8–42 Mapping Proxy and FileOut**



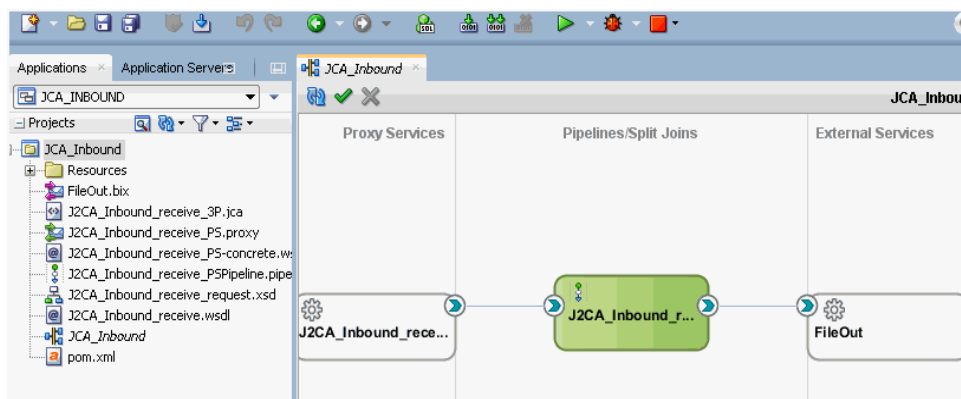
2. Double-click the Pipeline (for example, J2CA\_Inbound\_receive\_PSPipeline).
3. Click the Routing pane and ensure that the File Type Business Service (for example, FileOut) is properly configured in the Service field, as shown in [Figure 8–43](#).

**Figure 8–43 Routing Pane**



4. Save and close the Pipeline configuration page.
5. Double-click on the overview.xml file (for example, JCA\_Inbound) and click **Save All** in the menu bar to save the OSB process, as shown in [Figure 8–44](#).

**Figure 8–44 Save All**



### 8.2.3 Deploying the OSB Inbound Process

To deploy the created OSB inbound process, see steps 1 - 4 in [Section 8.1.3, "Deploying the OSB Outbound Process"](#) on page 8-15.

Once the OSB inbound process is deployed successfully, trigger an event from the PeopleSoft system and check if the output is received in the configured output location (for example, C:\output).

For more information on triggering an event, see [Section 4.5.5, "Triggering an Event in PeopleSoft"](#) on page 4-45.

## 8.3 Configuring an OSB Outbound Process Using JDeveloper (BSE Configuration)

This section describes how to configure an OSB outbound process to your PeopleSoft system, using Oracle JDeveloper for BSE configurations.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:

```
<ADAPTER_HOME>\etc\sample\PeopleSoft_Samples.zip\PeopleSoft_Samples\OSB_
Jdeveloper\BSE\PeopleSoft_Sample_BSE_OSB_Outbound_Project
```

This section includes the following topics:

- [Section 8.3.1, "Creating a Service Bus Application for OSB"](#)
- [Section 8.3.2, "Defining an OSB Outbound Process"](#)
- [Section 8.3.3, "Deploying the OSB Outbound Process"](#)

### Prerequisites

Before you design an OSB outbound process, you must generate the respective WSDL file using Application Explorer. For more information, see [Section 4.6.1, "Generating a WSDL File for Request and Response Services Using a Web Service"](#) on page 4-53.

### 8.3.1 Creating a Service Bus Application for OSB

To configure an OSB outbound process, you must create a service bus application for OSB. For more information, see [Section 8.1.1, "Creating a Service Bus Application for OSB"](#) on page 8-2.

### 8.3.2 Defining an OSB Outbound Process

This section describes how to define an OSB outbound process. The following topics are included:

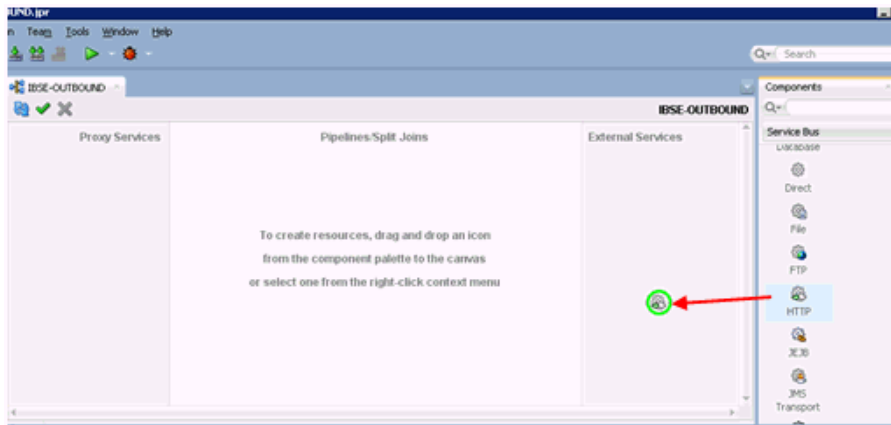
- [Section 8.3.2.1, "Configuring a WSDL-based Business Service"](#)
- [Section 8.3.2.2, "Creating a Proxy Service With Pipeline"](#)
- [Section 8.3.2.3, "Configuring a File Transport Type Business Service"](#)
- [Section 8.3.2.4, "Configuring the Routing Rules"](#)

#### 8.3.2.1 Configuring a WSDL-based Business Service

Perform the following steps to configure a WSDL-based Business Service:

1. Drag and drop the **HTTP** component from the Technology Components pane to the External Services area, as shown in [Figure 8-45](#).

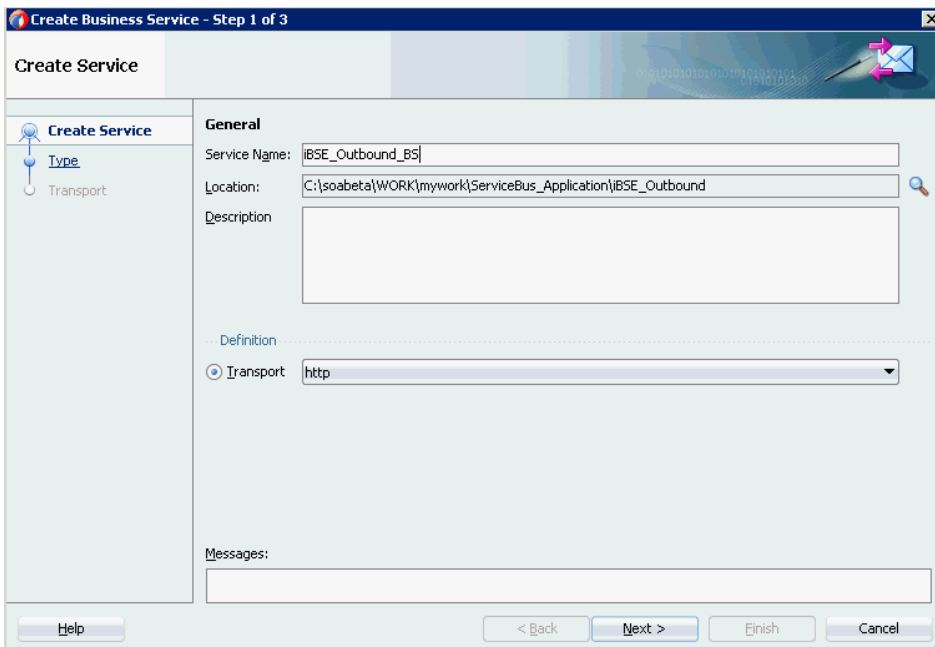
**Figure 8–45 HTTP Component**



The Create Business Service window is displayed.

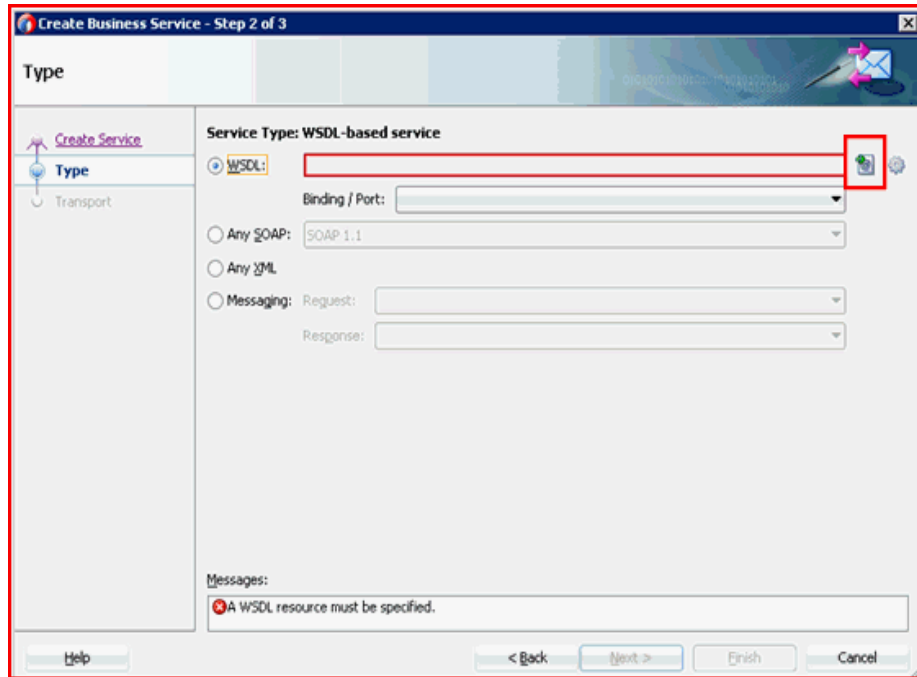
2. In the Service Name field, enter any name you wish for the Business Service and click **Next**, as shown in [Figure 8–46](#).

**Figure 8–46 Create Business Service**



3. In the displayed Service Type window, select the WSDL option and click the **Select WSDL** icon, as shown in [Figure 8–47](#).

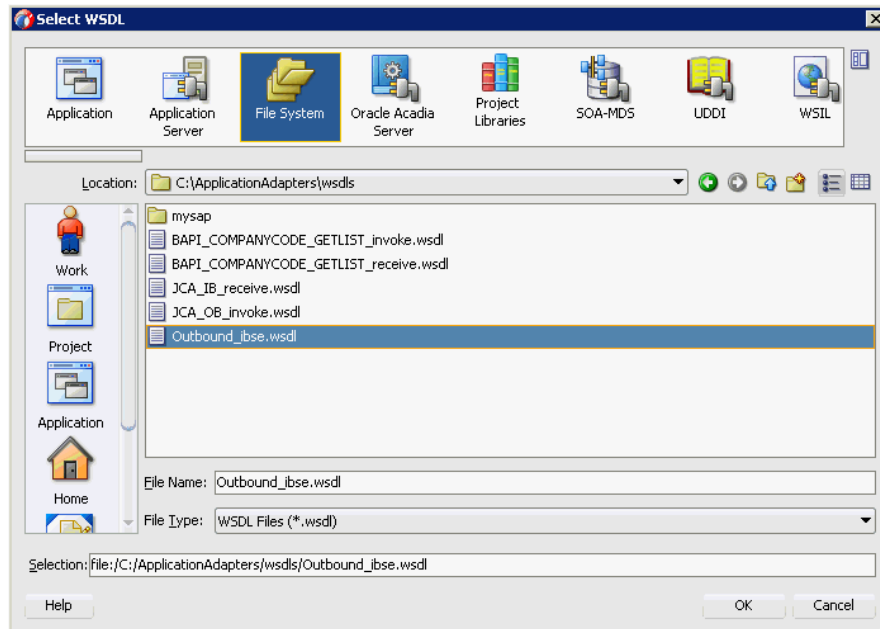
**Figure 8–47 Type Pane**



The Select WSDL window is displayed.

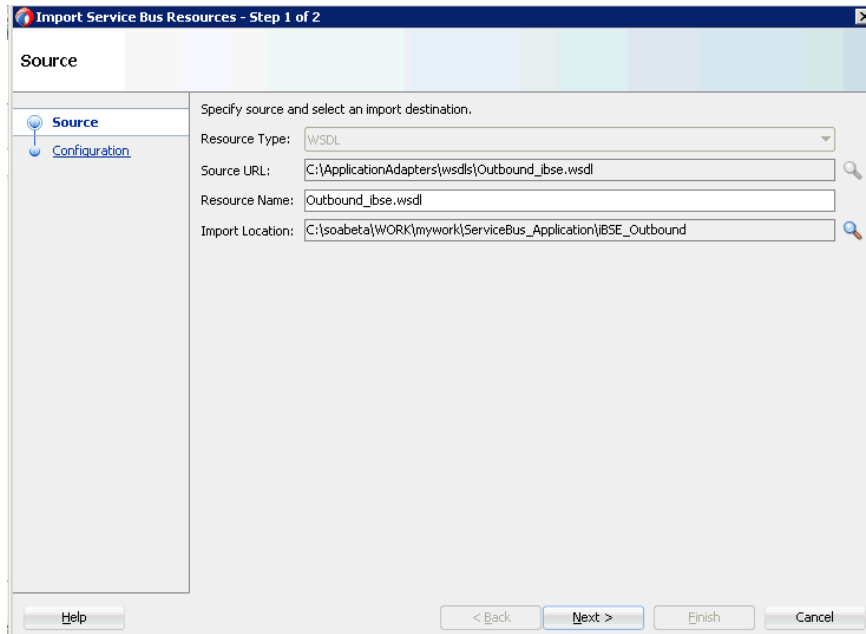
4. Select the File System folder icon, browse to the iBSE WSDL file and select it from the WSDL location, and then click **OK**, as shown in [Figure 8–48](#).

**Figure 8–48 Select WSDL Window**



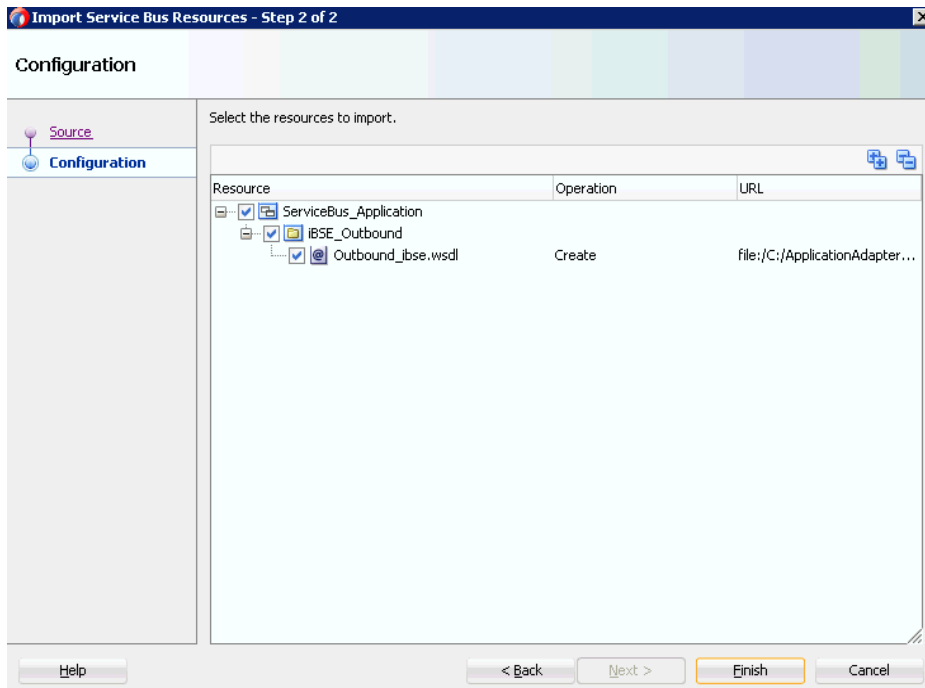
5. In the displayed Source pane, click **Next**, as shown in [Figure 8–49](#).

**Figure 8–49 Source Pane**



- In the displayed Configuration pane, click **Finish**, as shown in [Figure 8–50](#).

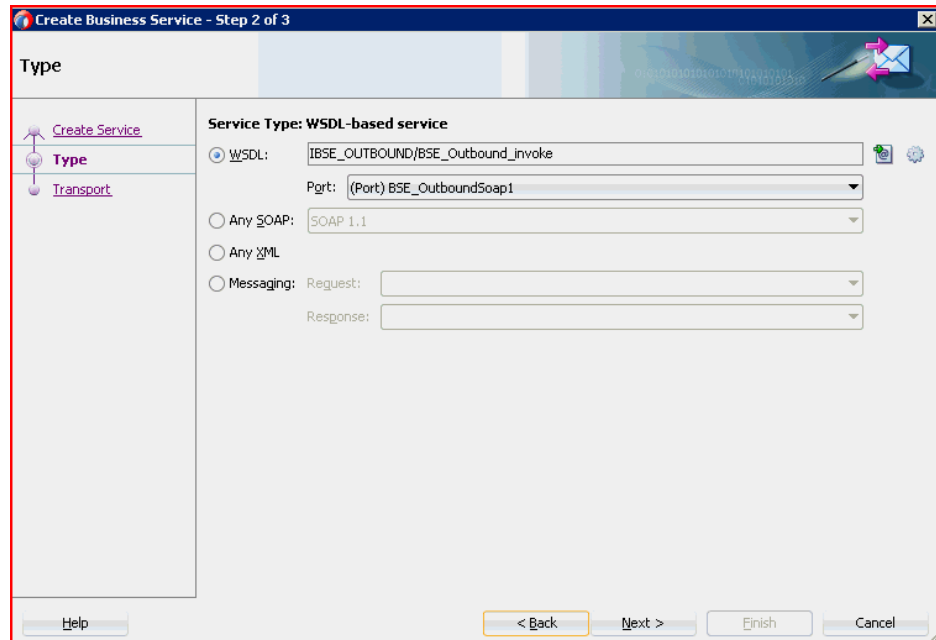
**Figure 8–50 Configuration Pane**



You are returned to the Create Business Service window.

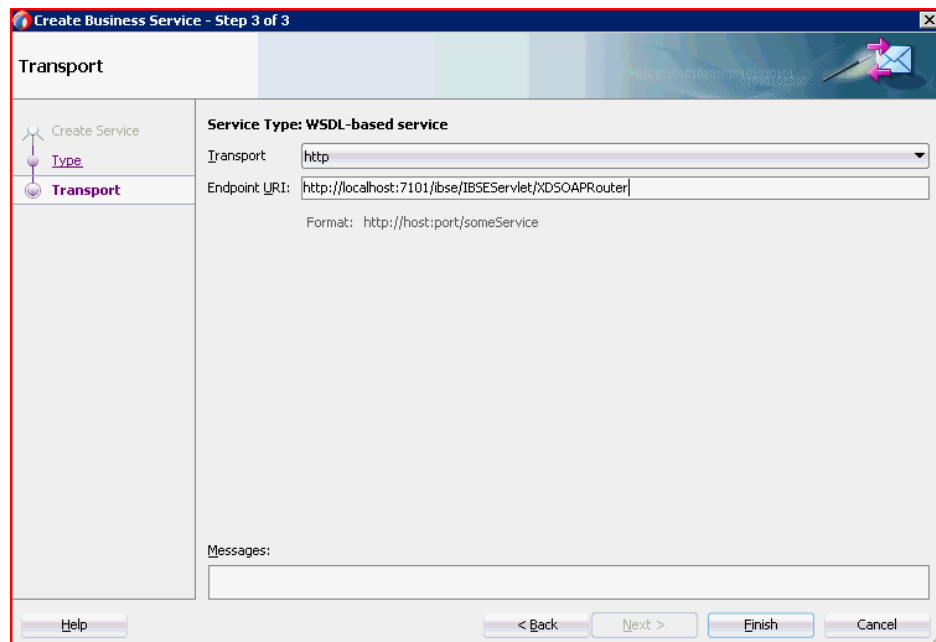
- In the displayed Type pane, click **Next**, as shown in [Figure 8–51](#).

**Figure 8–51 Type Pane**



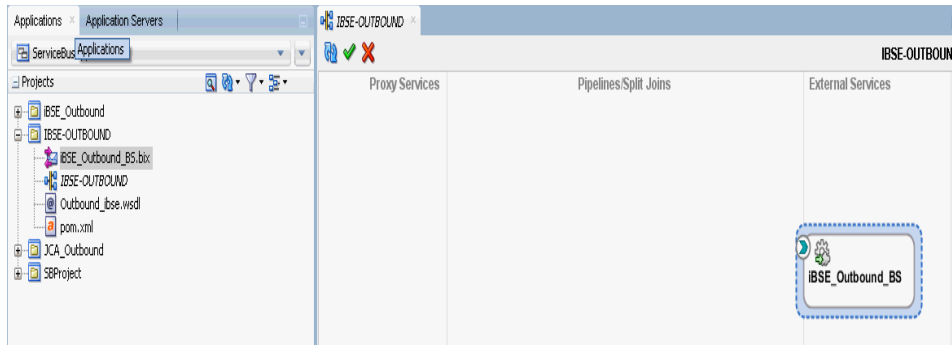
8. In the displayed Transport window, you can modify the Endpoint URI field if the hostname and port number varies, and then click **Finish**, as shown in [Figure 8–52](#).

**Figure 8–52 Transport Pane**



The Business Service is created and displayed in the External Services pane, as shown in [Figure 8–53](#).

**Figure 8–53 External Services Pane**

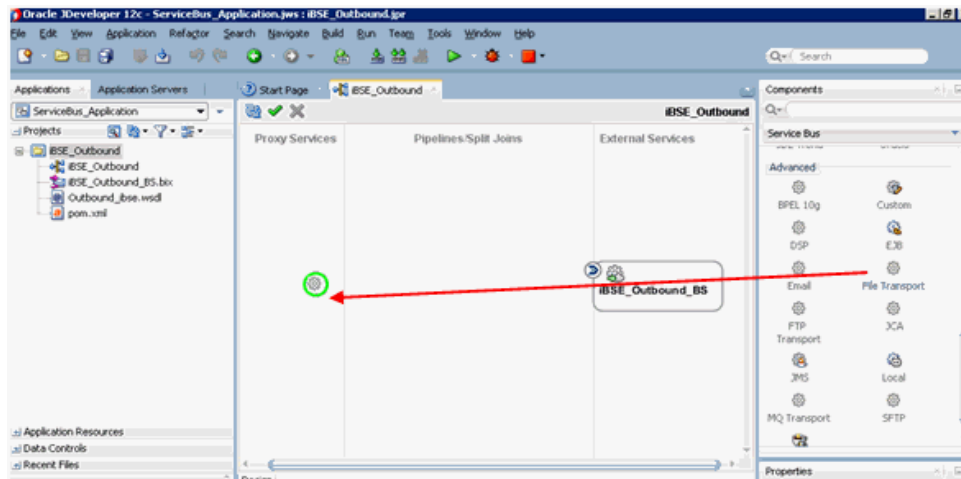


### 8.3.2.2 Creating a Proxy Service With Pipeline

Perform the following steps to create a Proxy Service with Pipeline:

1. Drag and drop the **File Transport** component from the Advanced Components pane to the Proxy Services pane, as shown in [Figure 8–54](#).

**Figure 8–54 File Transport Component**

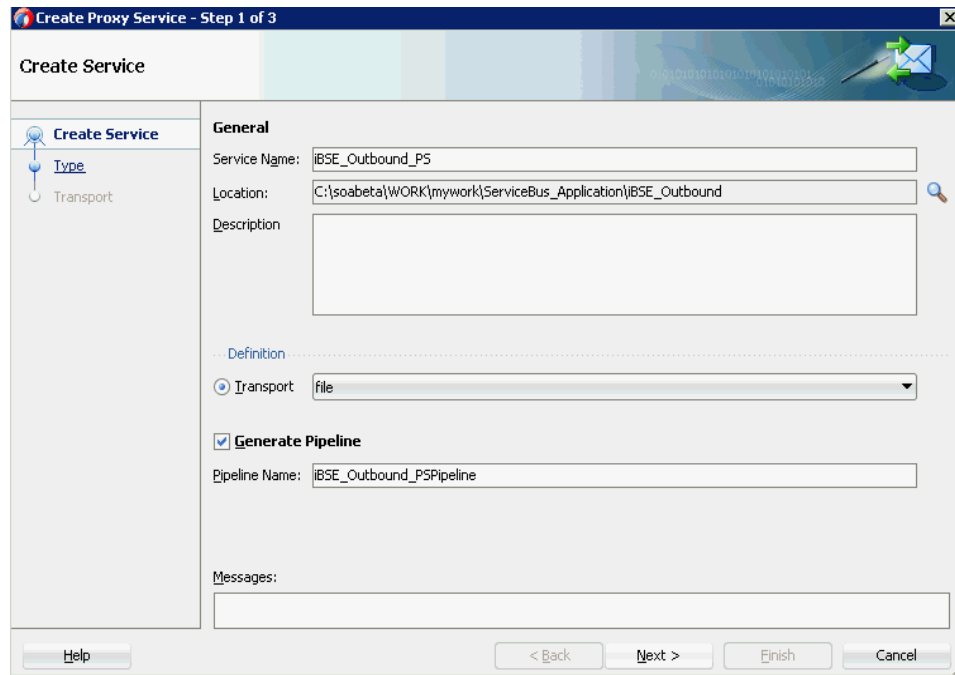


The Create Proxy Service pane is displayed.

2. In the Service Name field, enter any name you wish for the Proxy service (for example, JCA\_Outbound\_PS). By default, **Generate Pipeline** is selected.
3. Click **Next**, as shown in [Figure 8–55](#).



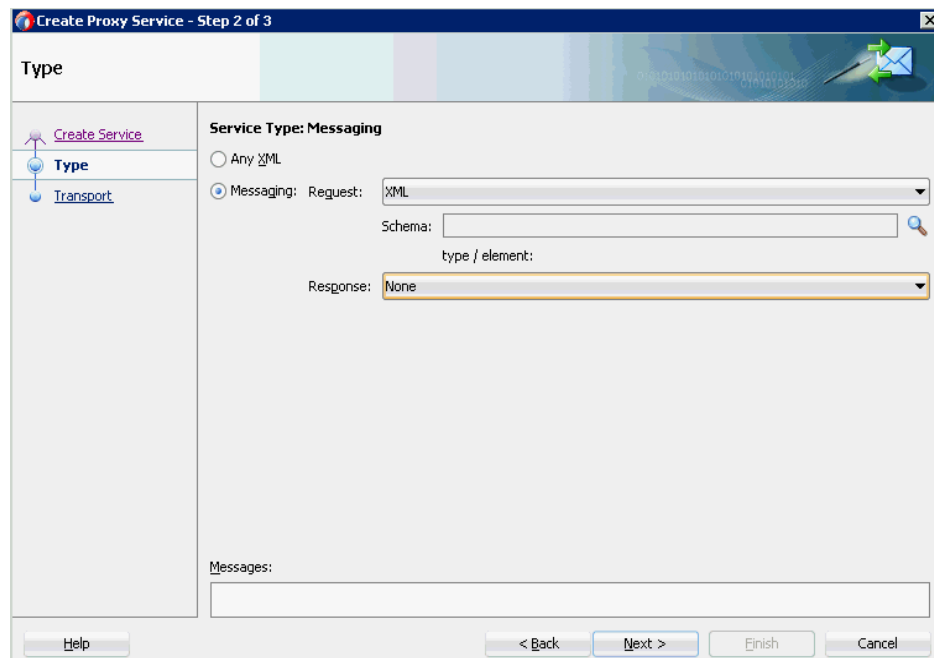
Figure 8–55 Create Service Pane



The Type pane is displayed.

4. Select the **Messaging** option, set the Request to **XML** and Response as **None**, and then click **Next**, as shown in Figure 8–56.

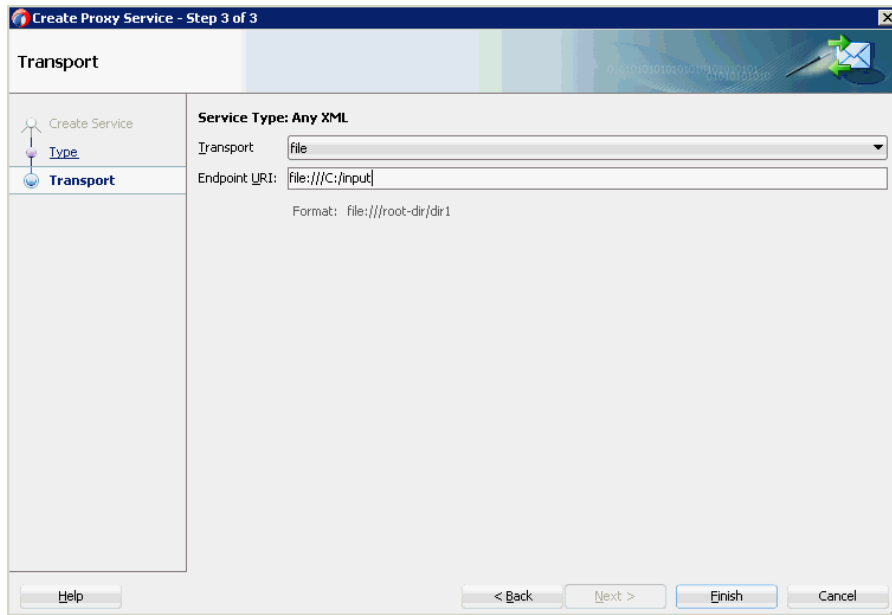
Figure 8–56 Type Pane



The Transport window is displayed.

5. Provide the input location in the Endpoint URI field (for example, c:/input) and click **Finish**, as shown in Figure 8–57.

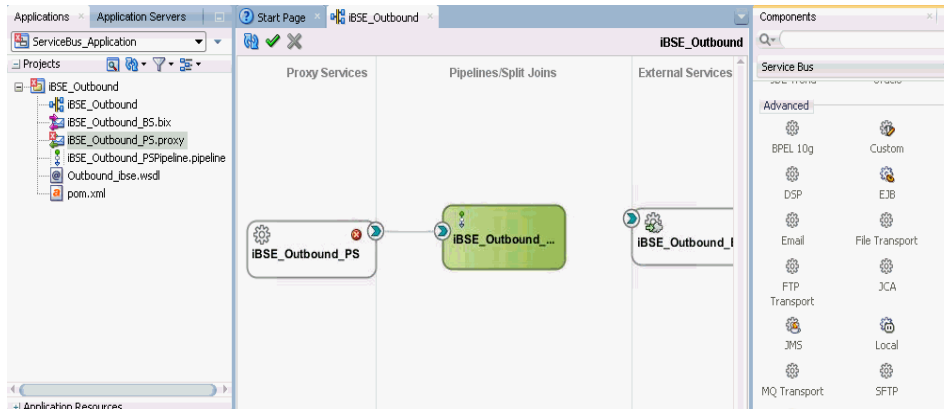
**Figure 8–57 Transport Window**



The Proxy service along with the pipeline is created and displayed.

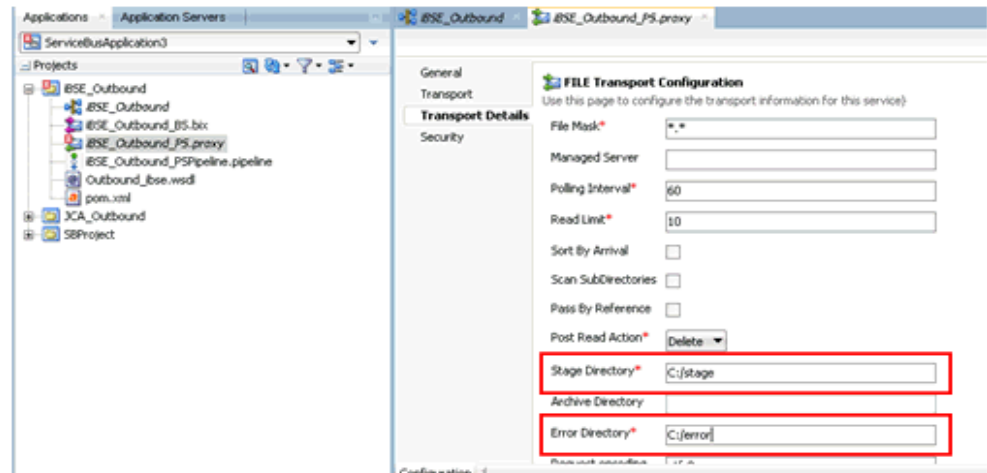
6. Double-click the created Proxy Service (for example: iBSE\_Outbound\_PS), as shown in [Figure 8–58](#).

**Figure 8–58 Proxy Service Edit**



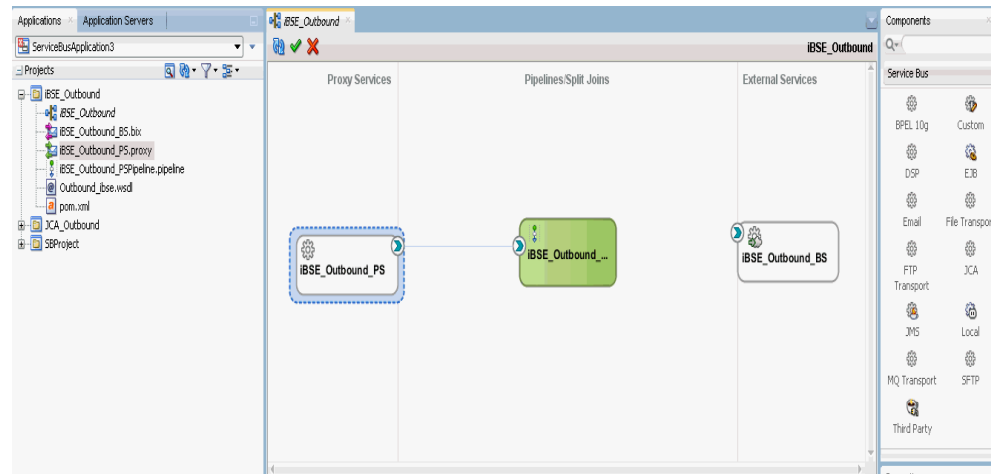
7. In the displayed Proxy Service configuration page, select **Transport Details** and provide the values for Stage and Error Directory, as shown in [Figure 8–59](#).

**Figure 8–59 File Transport Configuration**



8. Save and close the Proxy Service configuration page.
9. Double-click the overview.xml file (for example, iBSE\_Outbound).  
The Proxy service is updated and displayed, as shown in [Figure 8–60](#).

**Figure 8–60 Proxy Service**

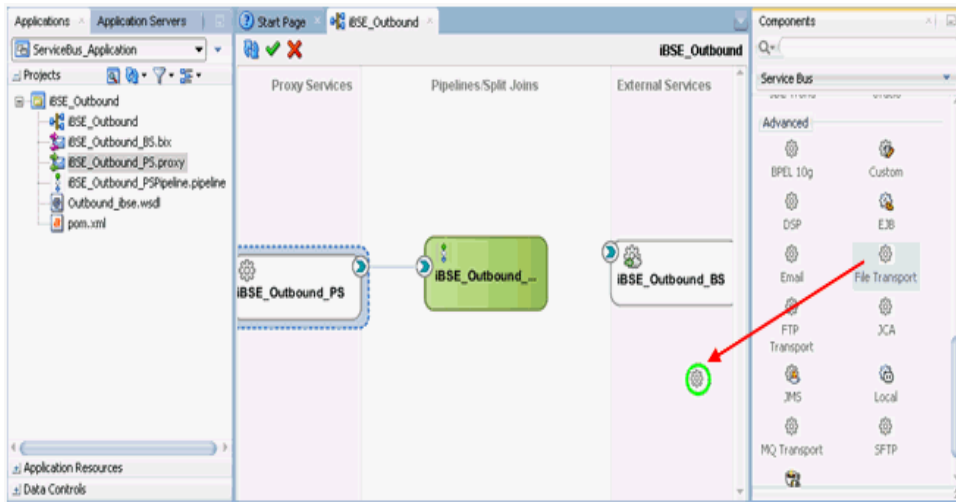


### 8.3.2.3 Configuring a File Transport Type Business Service

Perform the following steps to create a File Transport Type Business Service:

1. Drag and drop the **File Transport** component from the Advanced pane to the External Services pane, as shown in [Figure 8–61](#).

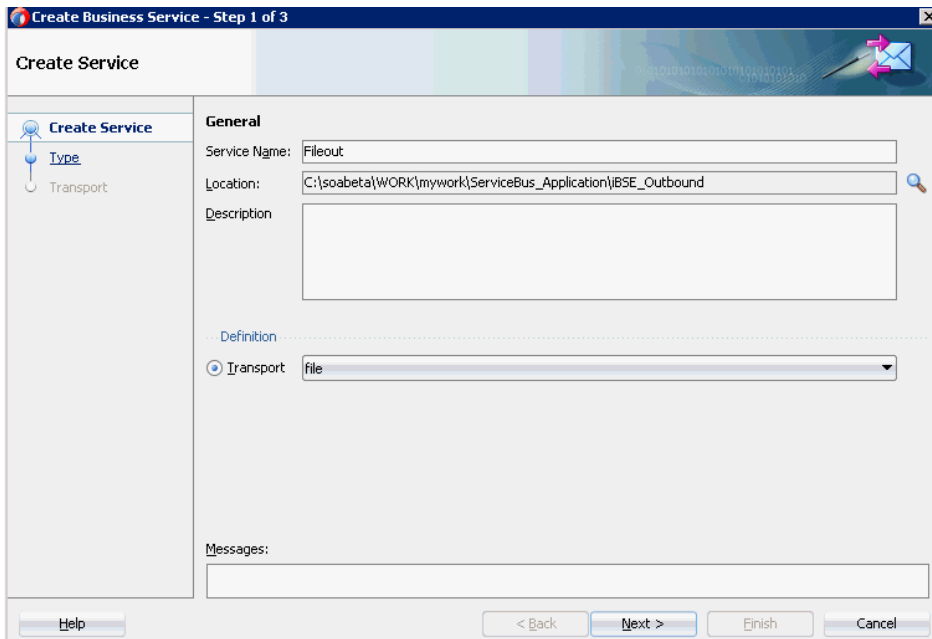
**Figure 8–61 File Transport Component**



The Create Business Service dialog is displayed.

2. In the Service Name field, enter any name you wish for the Business Service (for example, FileOut), and click **Next**, as shown in [Figure 8–62](#).

**Figure 8–62 Create Service Pane**

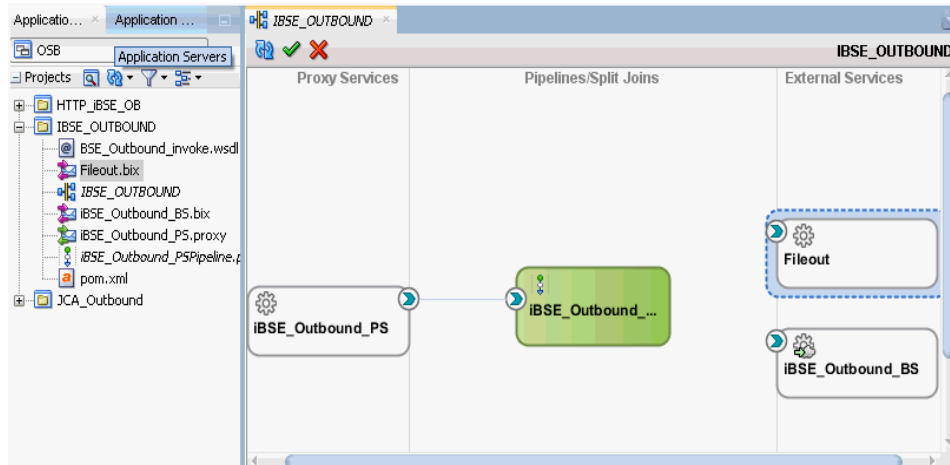


The Type pane is displayed. The **Any XML** option is selected by default.

3. Click **Next**.  
The Transport pane appears.
4. Provide the output location in the Endpoint URI field (for example, c:/output) and click **Finish**, as shown in [Figure 8–63](#).

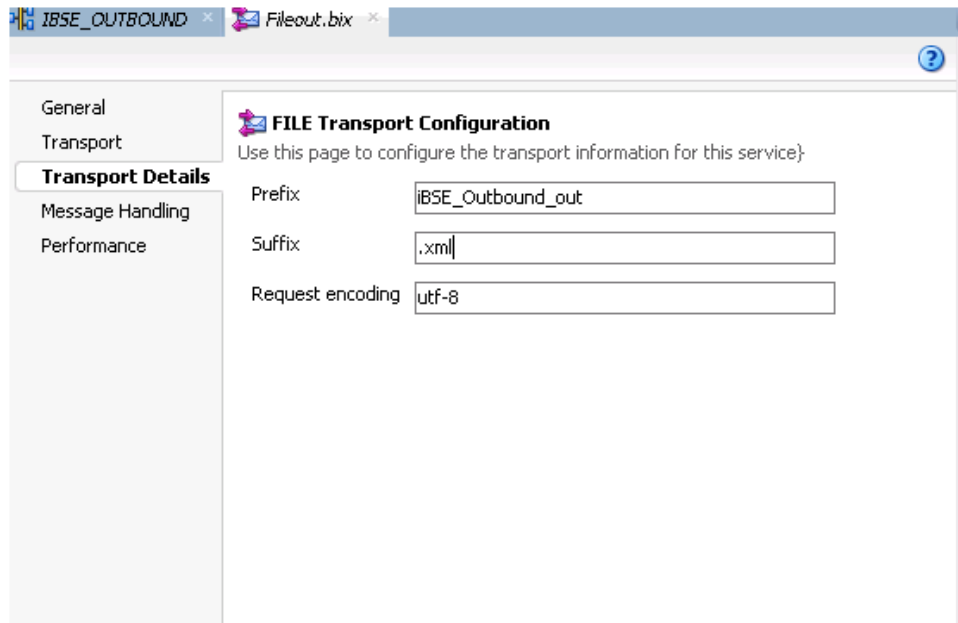
**Figure 8–63 Transport Pane**

The File Transport Business service Fileout is created and displayed, as shown in [Figure 8–64](#).

**Figure 8–64 Fileout Business Service**

5. Double-click the created Business service **Fileout** and provide the values for the Prefix and Suffix fields in the Transport Details Tab, as shown in [Figure 8–65](#).

**Figure 8–65 Transport Details**



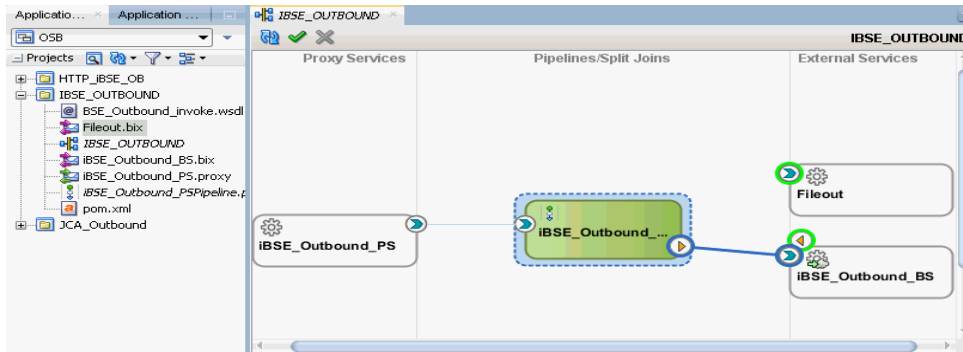
6. Save and close the configuration page, and double-click on overview.xml (for example, iBSE\_Outbound).

### 8.3.2.4 Configuring the Routing Rules

Perform the following steps to configure the routing rules:

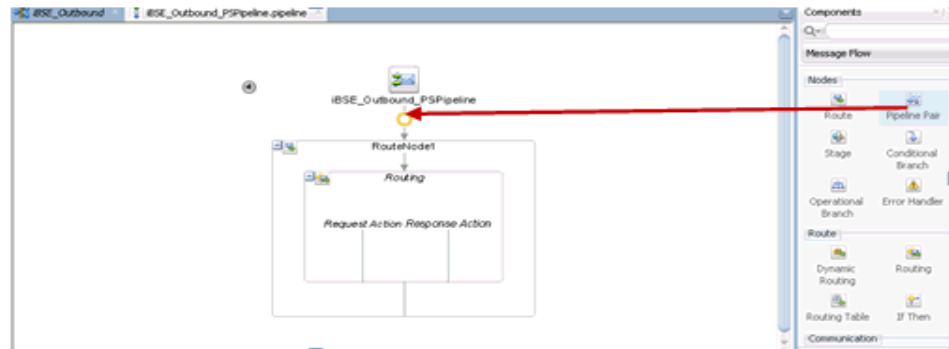
1. Create a connection between the Pipeline Component (for example, iBSE\_Outbound\_PSPipeline) and the WSDL based Business Service (for example, iBSE\_Outbound\_BS), as shown in [Figure 8–66](#).

**Figure 8–66 Pipeline Component**



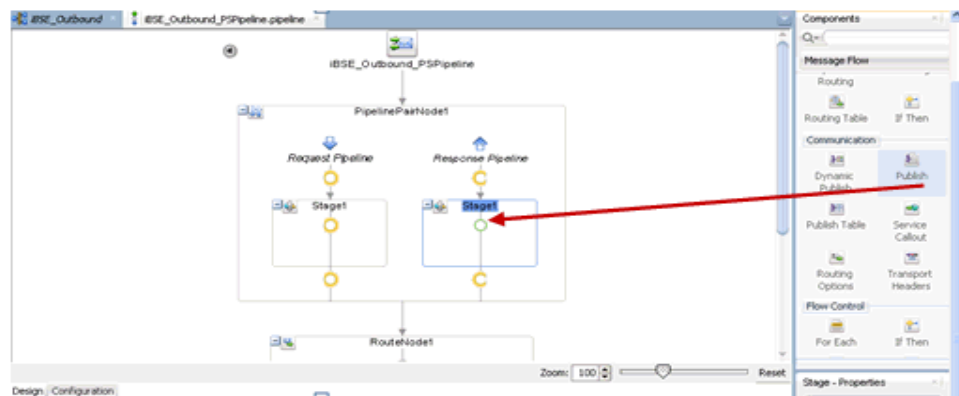
2. Double-click on the **Pipeline** component (for example, iBSE\_Outbound\_PSPipeline) in the Pipelines/Split Joins pane.
3. Drag and drop the **Pipeline Pair** node from Nodes pane to the area between the Pipeline (for example: iBSE\_Outbound\_PSPipeline) and RouteNode1, as shown in [Figure 8–67](#).

**Figure 8-67 Pipeline Pair Node**



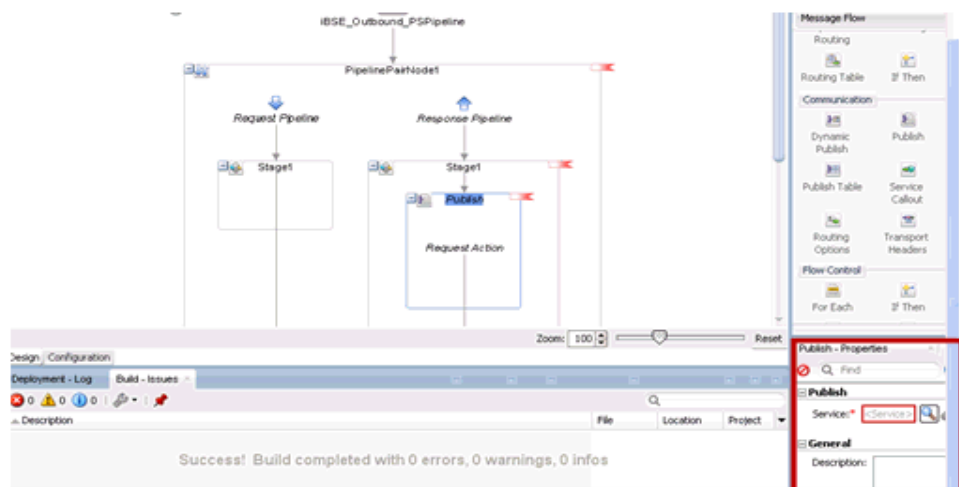
4. Drag and drop the **Publish** node from the Communication pane to the area beneath Stage1 of the Response Pipeline, as shown in Figure 8-68.

**Figure 8-68 Publish Node**



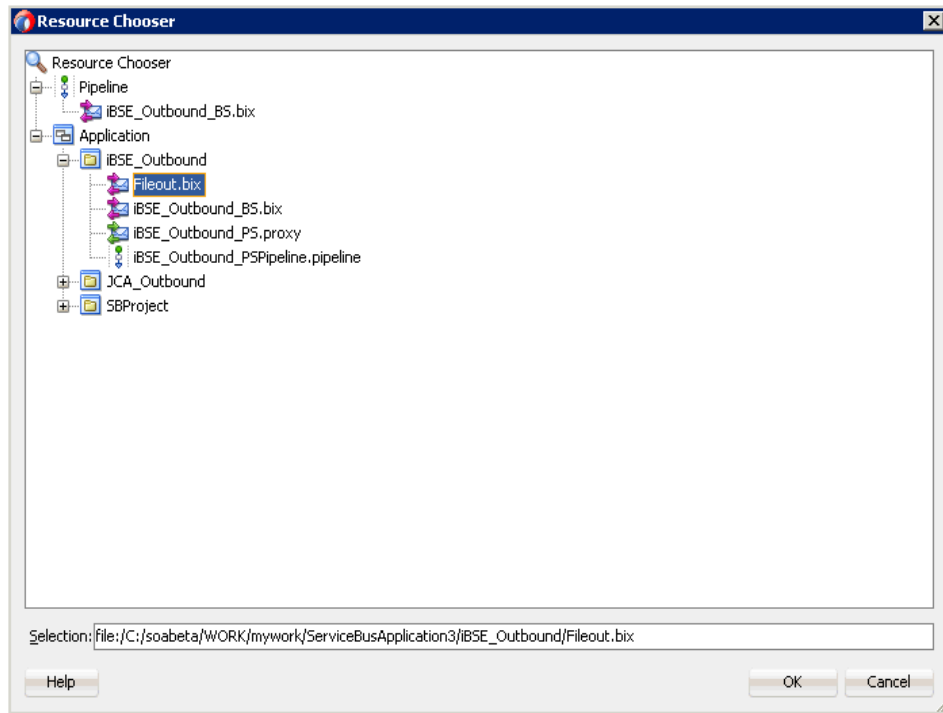
5. Click on the browse icon to the right of the Service field in the right pane of Publish Properties, as shown in Figure 8-69.

**Figure 8-69 Browse Icon**



6. In the displayed Resource Chooser window, select the **Fileout.bix** File Transport Business service and click **OK**, as shown in Figure 8-70.

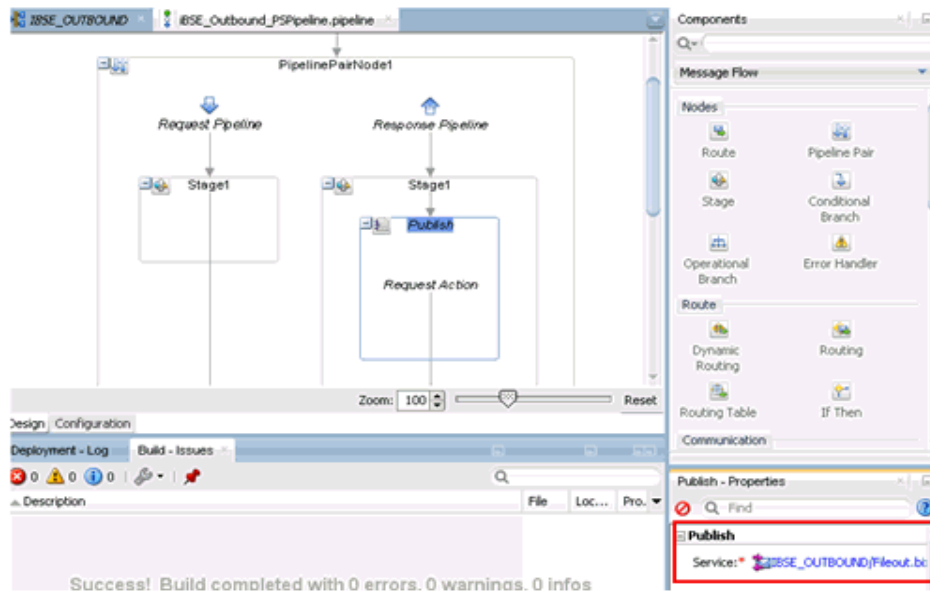
**Figure 8–70 Resource Chooser**



You are returned to the Pipeline configuration page.

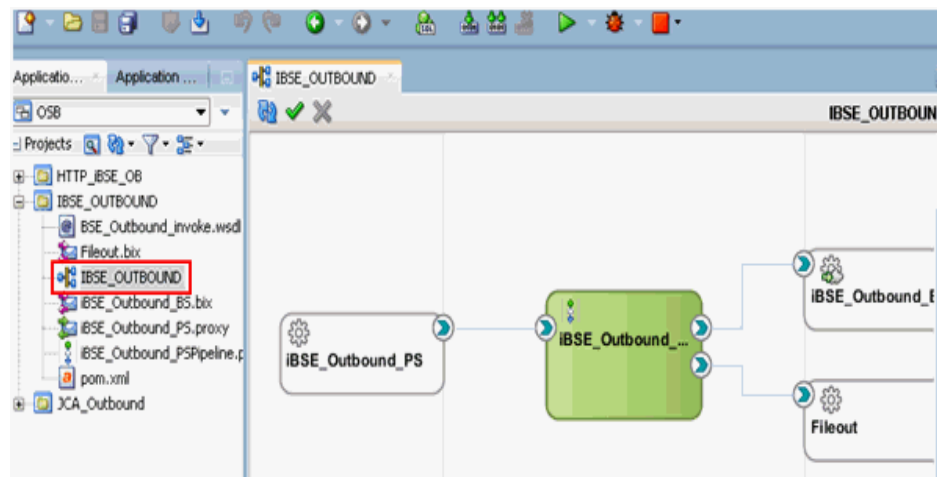
In the right pane, the selected service is configured in the Publish pane, as shown in Figure 8–71.

**Figure 8–71 Publish Pane**



7. Save and close the Pipeline configuration page.
8. Double-click the overview.xml file (for example: iBSE\_Outbound), and click **Save All** in the menu bar to save the OSB process, as shown in Figure 8–72.



**Figure 8–72 Save All Icon**

### 8.3.3 Deploying the OSB Outbound Process

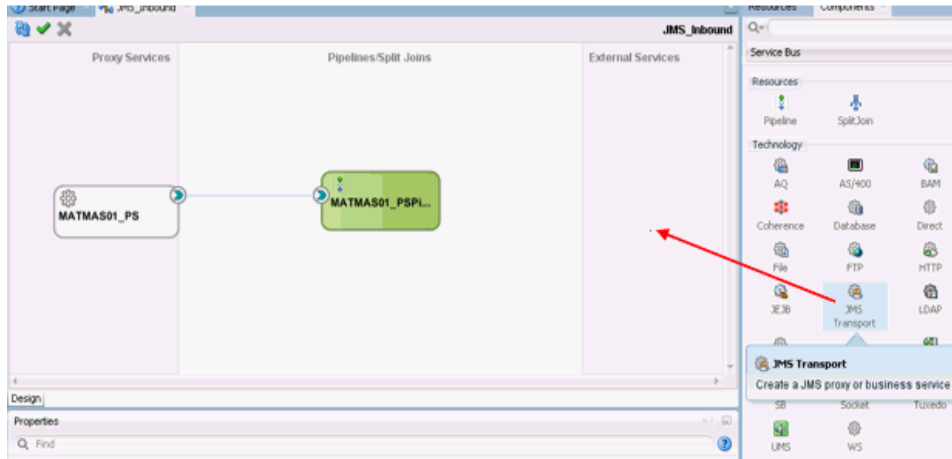
To deploy the created OSB outbound process and invoke the input XML document, see [Section 8.1.3, "Deploying the OSB Outbound Process"](#) on page 8-15.

## 8.4 Configuring a JMS Inbound Process Using JDeveloper (J2CA Configuration)

This section describes how to configure a JMS inbound process to your PeopleSoft system, using Oracle JDeveloper for J2CA configurations.

1. Before you design a JMS process, you must generate the respective WSDL file using Application Explorer. For more information, see [Section 4.5.1, "Generating WSDL for Event Integration"](#) on page 4-34.
2. Start the Oracle JDeveloper and create a Service Bus Application for OSB. For more information, see [Section 8.1.1, "Creating a Service Bus Application for OSB"](#) on page 8-2.
3. Create a Third Party Adapter Service Component. For more information, see [Section 8.2.2.1, "Configuring a Third-Party Adapter Service Component"](#) on page 8-19.
4. Create a Proxy Service along with the pipeline from the JCA Binding File. For more information, see [Section 8.2.2.2, "Creating a Pipeline"](#) on page 8-21.
5. Create a JMS Transport Business Service and perform the following steps:
  - a. Drag and drop the **JMS Transport** component from the Technology Components pane to the External Services pane, as shown in [Figure 8–73](#).

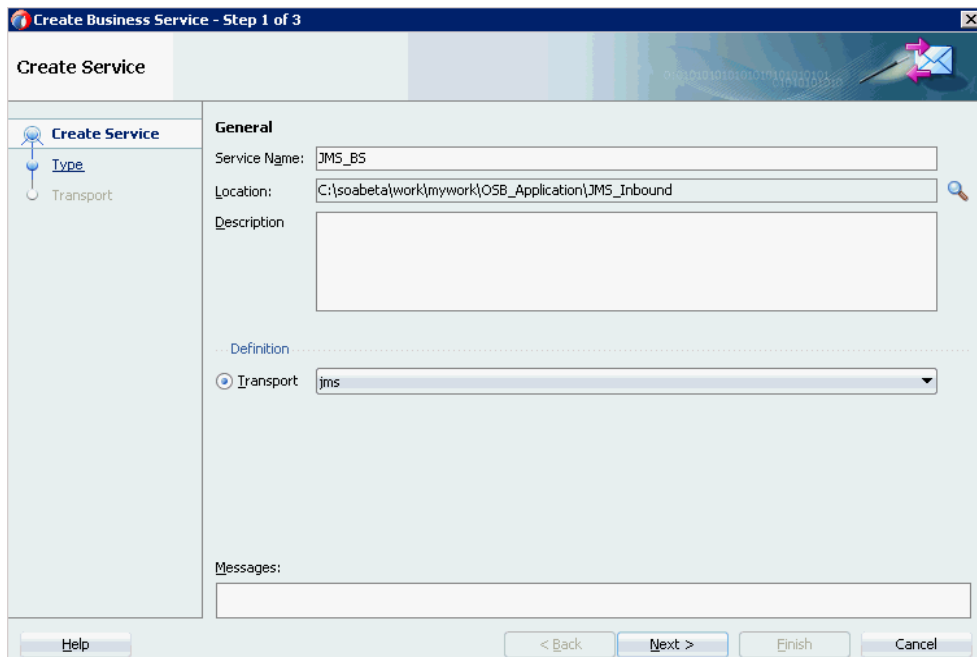
**Figure 8–73 JMS Transport Component**



The Create Business Service dialog is displayed.

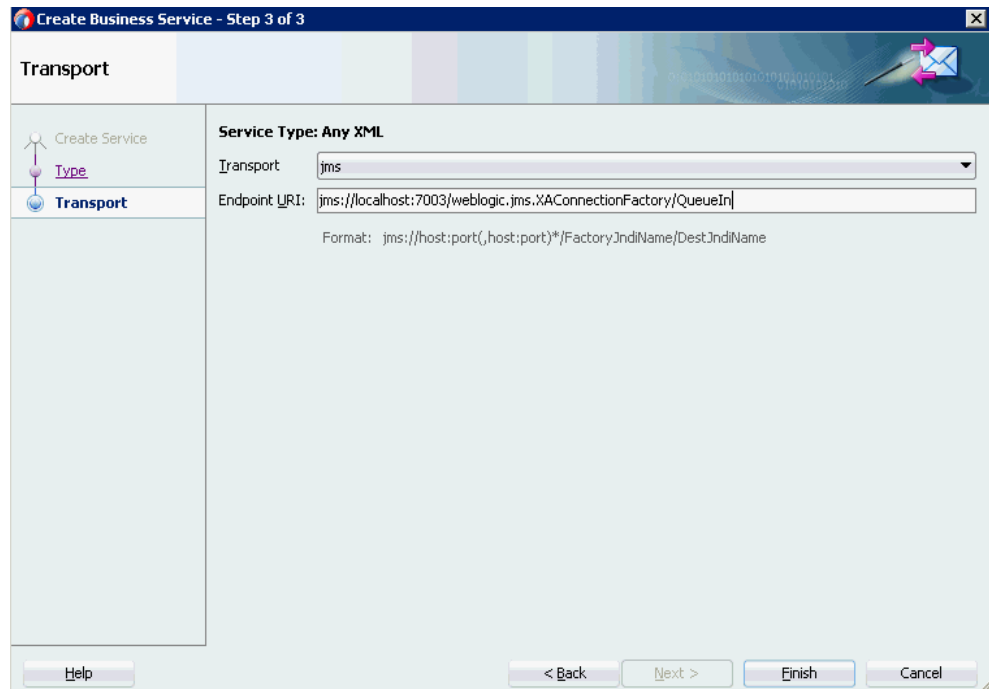
- b. In the Service Name field, enter any name you wish for the Business service (for example, JMS\_BS) and click **Next**, as shown in [Figure 8–74](#).

**Figure 8–74 Create Service Pane**



- c. In the displayed Type window, select **Any XML** and then click **Next**.  
The Transport window is displayed.
- d. Modify the appropriate hostname and port number by replacing DestJndiName with QueueIn in the Endpoint URI field (for example, `jms://localhost:7003/weblogic.jms.XAConnectionFactory/QueueIn`), and then click **Finish**, as shown in [Figure 8–75](#).

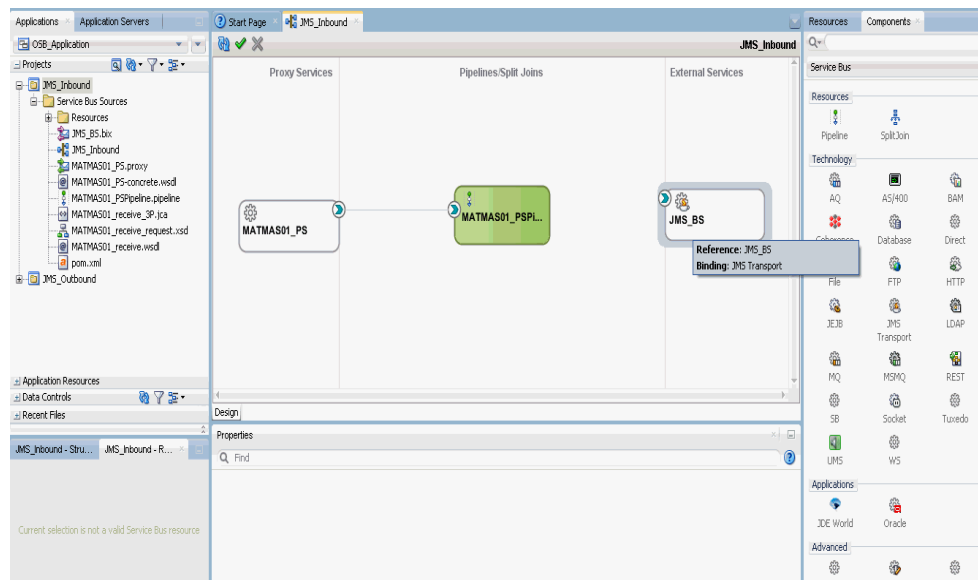
**Figure 8–75 Transport Window**



The JMS Business service is created and displayed.

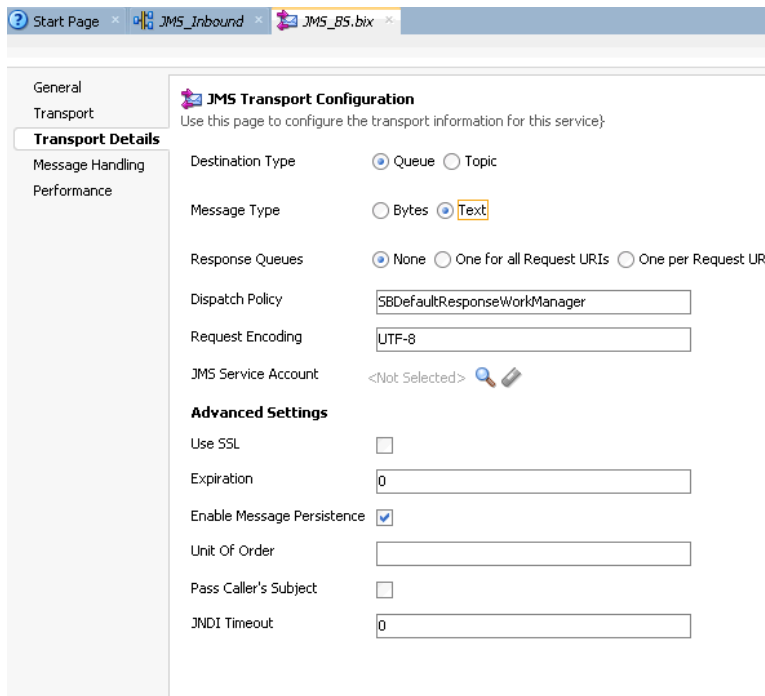
- e. Double-click **JMS\_BS** as shown in [Figure 8–76](#).

**Figure 8–76 JMS Business Service**



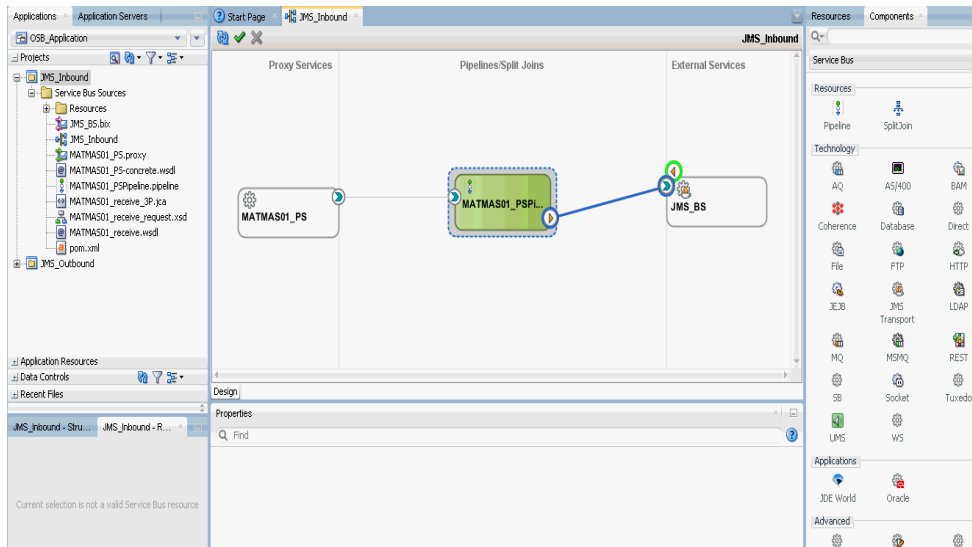
- f. In the displayed Business Service configuration page, provide the following parameters in the Transport Details tab, as shown in [Figure 8–77](#).

**Figure 8–77 JMS Transport Configuration**

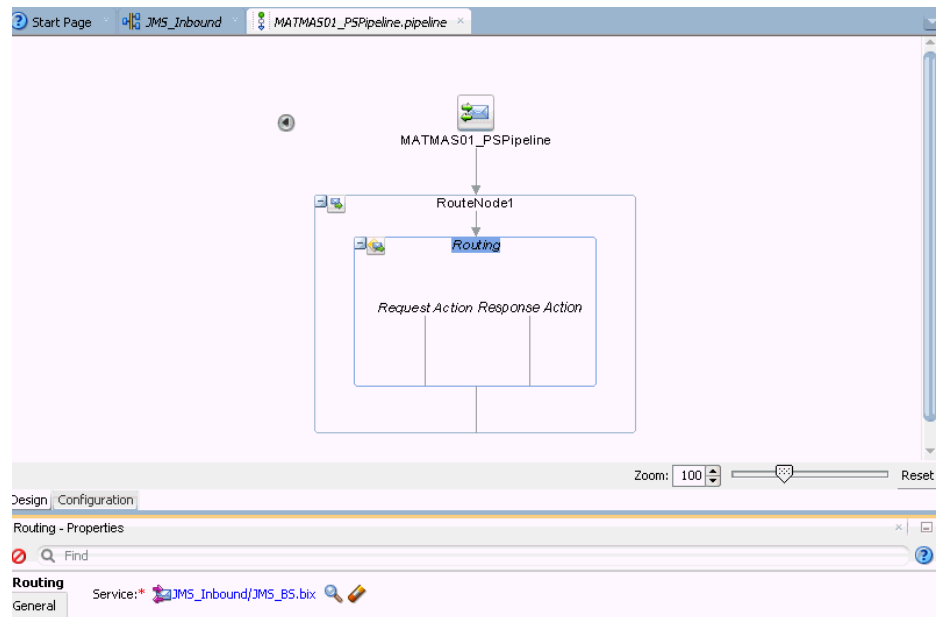


- g. In the Destination Type section, select **Queue**.
- h. In the Message Type section, select **Text**.
6. Save and close the Configuration page of the business service.
7. Create a connection between **Pipeline** (for example, xxxx\_PSPipeline) and **JMS Business Service** (for example, JMS\_BS) as shown in figure [Figure 8–78](#).

**Figure 8–78 Configuration Page**



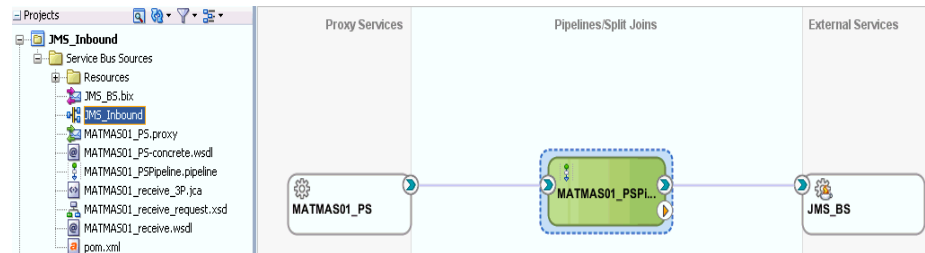
8. Double-click **Pipeline**.  
The Pipeline Configuration page is displayed as shown in [Figure 8–79](#).

**Figure 8–79 Pipeline Configuration**


9. Check that the details are configured properly, and then save and close the Pipeline configuration page.

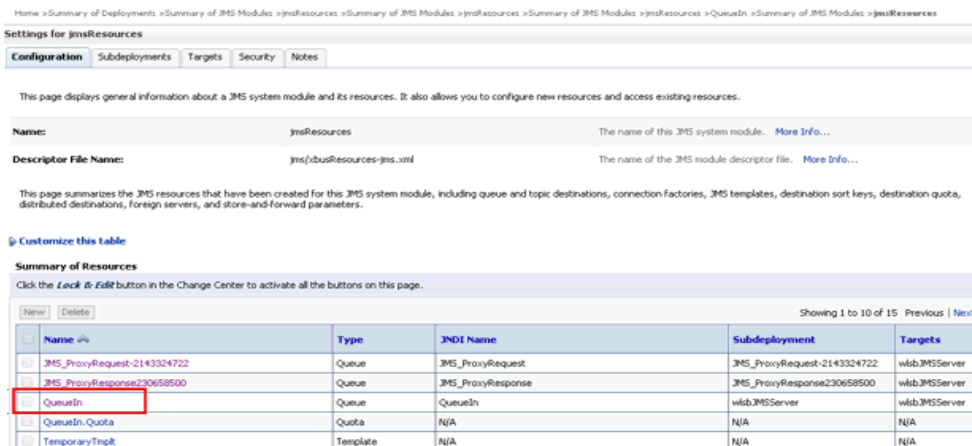
You are returned to the composite editor window.

10. Click **Save All** in the menu bar to save the OSB JMS process, as shown in [Figure 8–80](#).

**Figure 8–80 Save All Icon**


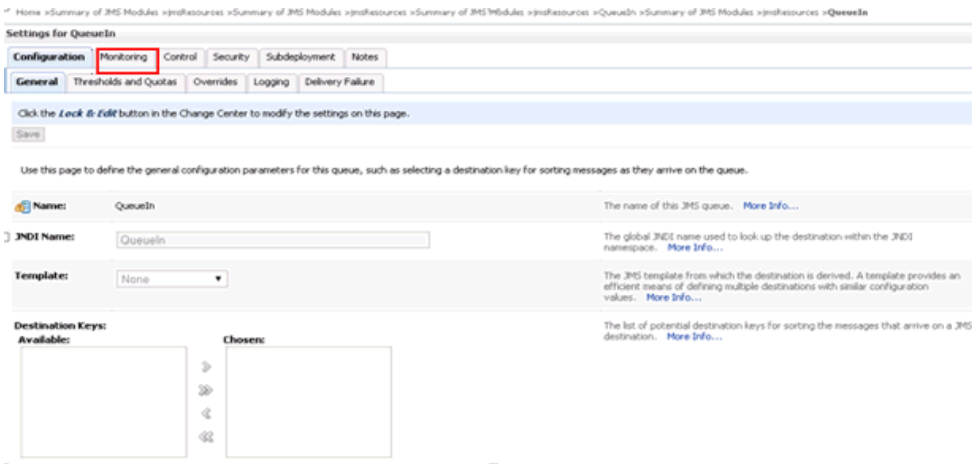
11. Deploy the OSB JMS inbound process. For more information, see [Section 8.2.3, "Deploying the OSB Inbound Process"](#) on page 8-26.
12. Once the process is deployed successfully, trigger the event messages. For more information, see [Section 4.5.5, "Triggering an Event in PeopleSoft"](#) on page 4-45.
13. Log on to the Oracle WLS console.
14. In the Oracle WLS console, expand **Services**, click **Messaging**, select **JMS Modules**, and then click **jmsResources**.
15. Click the appropriate response link (for example, QueueIn) as shown in [Figure 8–81](#).

**Figure 8–81 QueueIn Response Link**



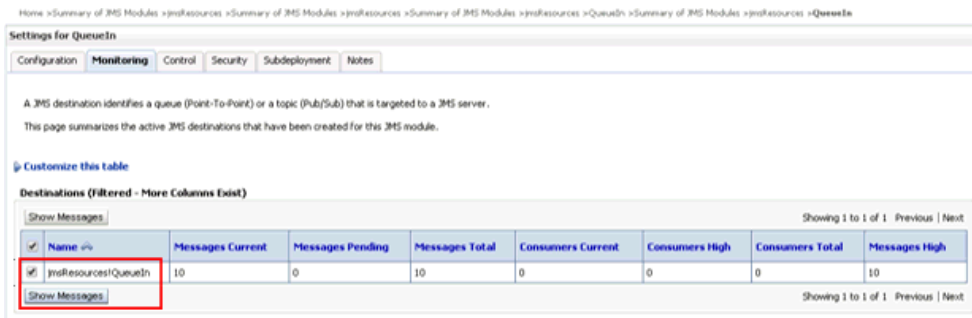
16. Click the Monitoring tab, as shown in Figure 8–82.

**Figure 8–82 Monitoring Tab**



17. Select the check box and click the **Show Messages** button, as shown in Figure 8–83.

**Figure 8–83 Show Messages Button**



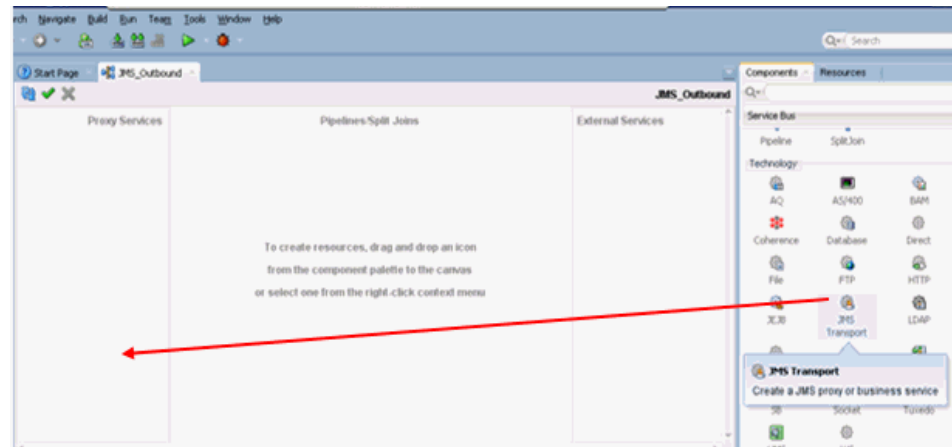
18. Click the ID link with the appropriate time and date.  
The response document is shown under the Text field.

## 8.5 Configuring a JMS Outbound Process Using JDeveloper (J2CA Configuration)

This section describes how to configure a JMS outbound process to your PeopleSoft system, using Oracle JDeveloper for J2CA configurations.

1. Before you design a JMS process, you must generate the respective WSDL file using Application Explorer. For more information, see [Section 4.4.1, "Generating WSDL for Request/Response Service"](#) on page 4-8.
2. Start the Oracle JDeveloper and create a Service Bus Application for OSB. For more information, see [Section 8.1.1, "Creating a Service Bus Application for OSB"](#) on page 8-2.
3. Create a Third Party Adapter Service Component. For more information, see [Section 8.1.2.1, "Configuring a Third-Party Adapter Service Component"](#) on page 8-3.
4. Create a WSDL-based Business Service from the JCA Binding File. For more information, see [Section 8.1.2.2, "Configuring a File Transport Type Business Service"](#) on page 8-6.
5. Create a JMS Proxy Service with a Pipeline and perform the following steps:
  - a. Drag and drop the **JMS Transport** component from the Technology Components pane to the Proxy Services pane, as shown in [Figure 8–84](#).

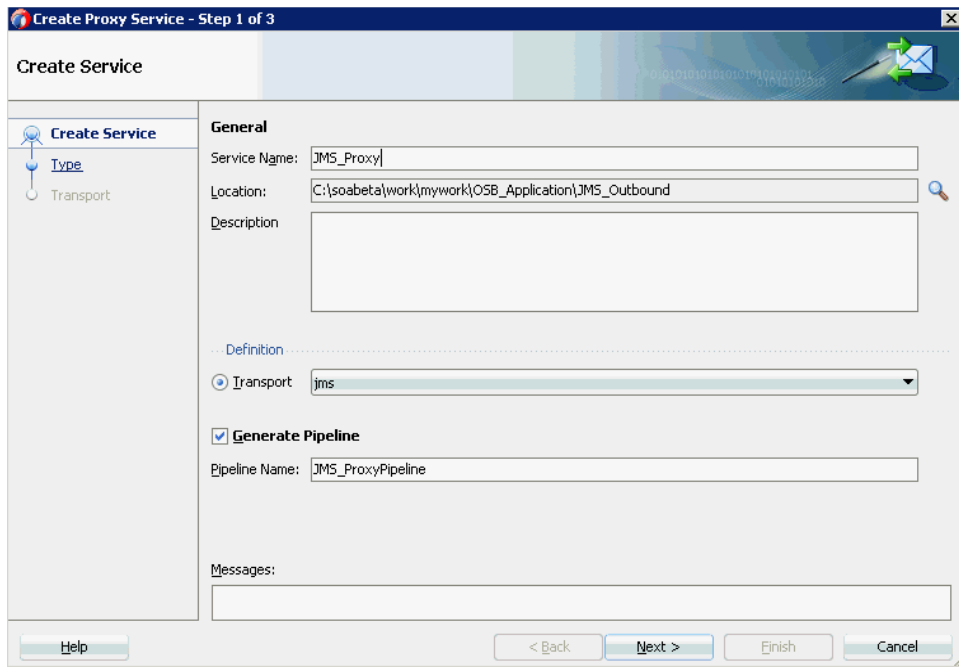
**Figure 8–84 JMS Transport Component**



The Create Business Service dialog is displayed.

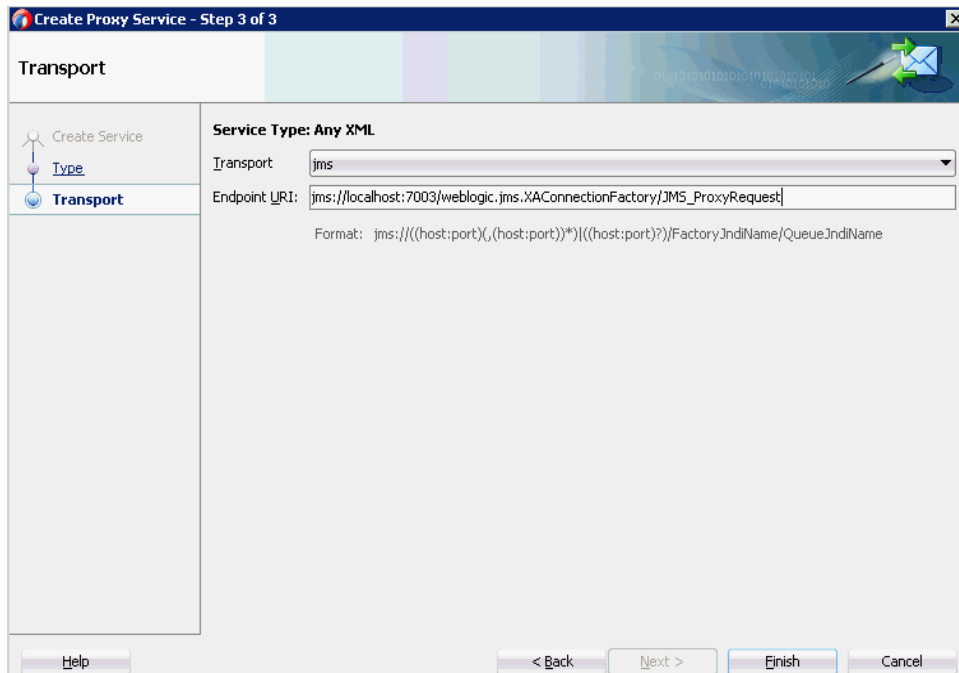
- b. In the Service Name field, enter any name you wish for the Proxy service (for example, JMS\_Proxy). By default, Generate Pipeline is selected.
- c. Click **Next**, as shown in [Figure 8–85](#).

**Figure 8–85 Create Proxy Service Pane**



- d. In the displayed Type window, select **Any XML** and then click **Next**.  
The Transport window is displayed.
- e. Modify the appropriate hostname and port number by replacing the Endpoint URI field (for example, `jms://localhost:7003/weblogic.jms.XAConnectionFactory/JMS_ProxyRequest`), and then click **Finish**, as shown in [Figure 8–86](#).

**Figure 8–86 Transport Window**

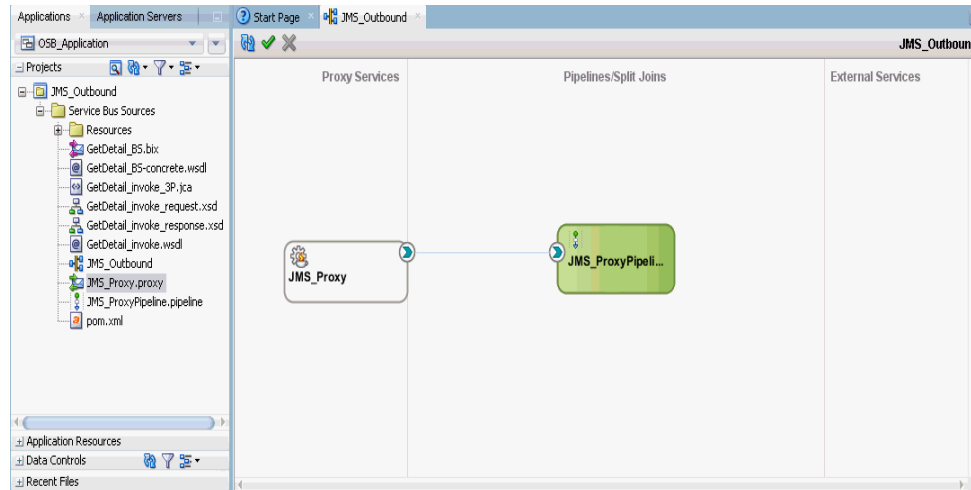




The JMS Proxy service along with the pipeline is created and displayed.

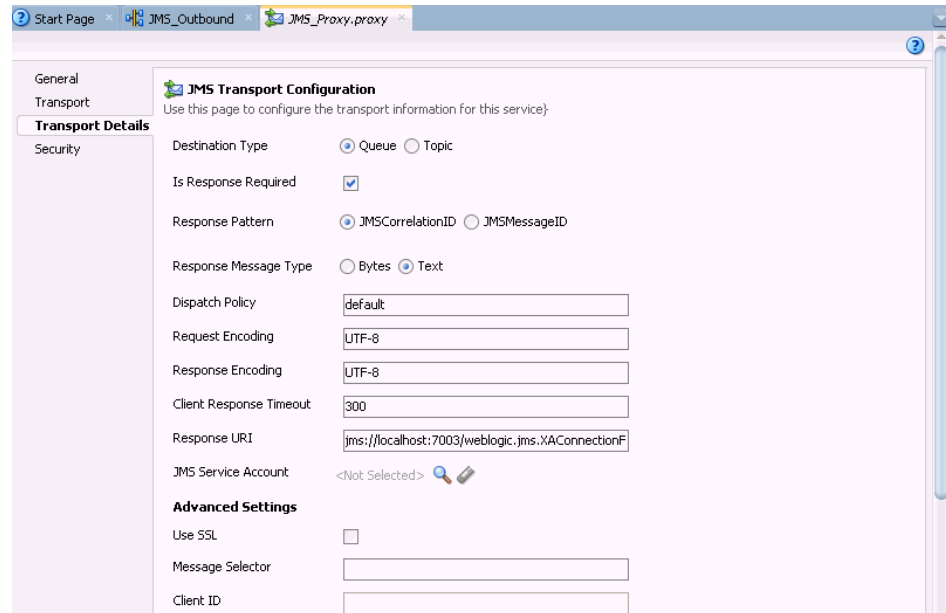
- f. Double-click the created Proxy Service (for example, JMS\_Proxy), as shown in Figure 8-87.

**Figure 8-87 JMS Proxy Service**



- g. In the displayed configuration page of the Proxy Service, provide the following parameters in the Transport Details tab, as shown in Figure 8-88.

**Figure 8-88 JMS Transport Configuration**

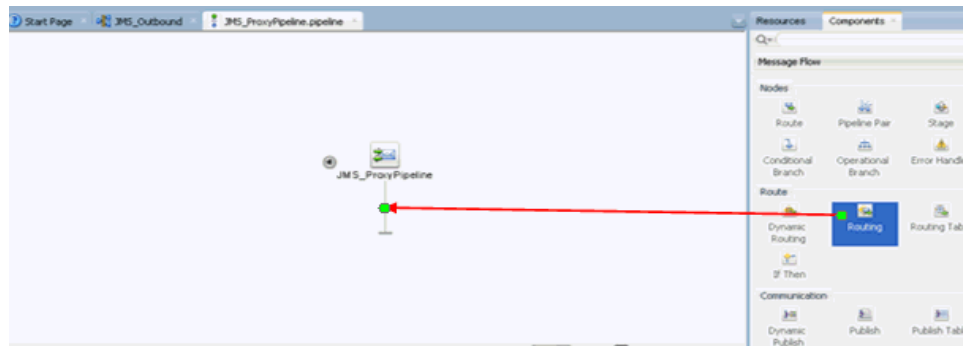


- h. In the Destination Type section, select **Queue**.
- i. Select the **Is Response Required** check box.
- j. In the Response Message Type section, select **Text**.
- k. In the Response URI field, provide the Endpoint URI used in the JMS Transport Configuration and change Request to Response. For example,

```
jms://localhost:7003/weblogic.jms.XAConnectionFactory/JMS_ProxyResponse
```

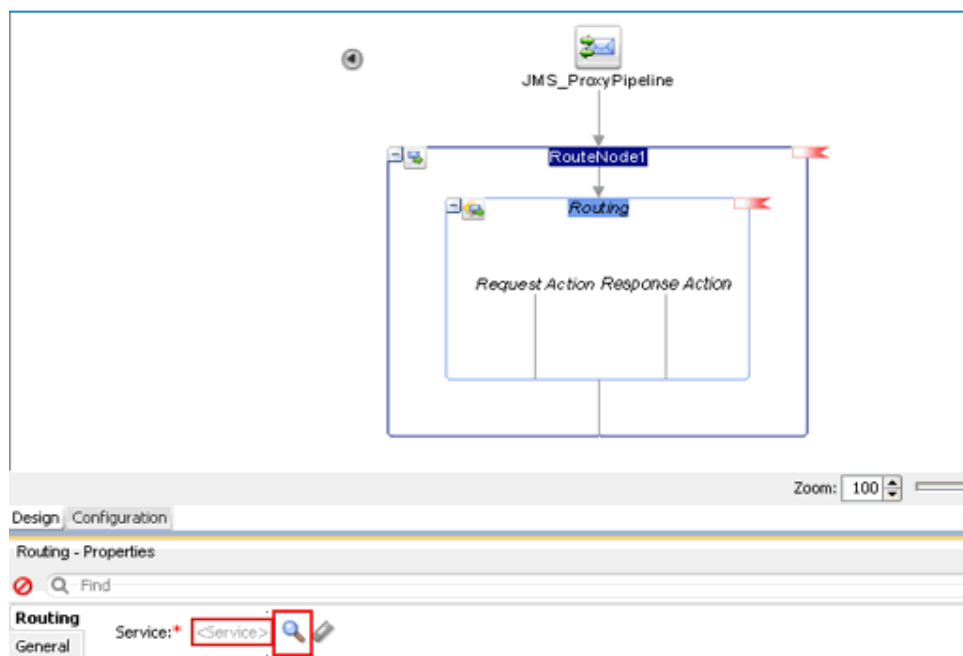
6. Save and close the Configuration page of the Proxy service.
7. Configure the Routing Rules and proceed with the following steps:
  - a. Double-click on the pipeline (for example, JMS\_ProxyPipeline) in the Pipelines/Split Joins pane.  
The Pipeline configuration page is displayed.
  - b. Drag and drop the **Routing** component from the Route section to the area below the Pipeline (for example, JMS\_ProxyPipeline), as shown in [Figure 8–89](#).

**Figure 8–89 Routing Component**



- c. In the Pipeline Configuration page, select **Routing** and click the browse icon to the right of the Service field in the Routing Properties pane, as shown in [Figure 8–90](#).

**Figure 8–90 Browse Service**



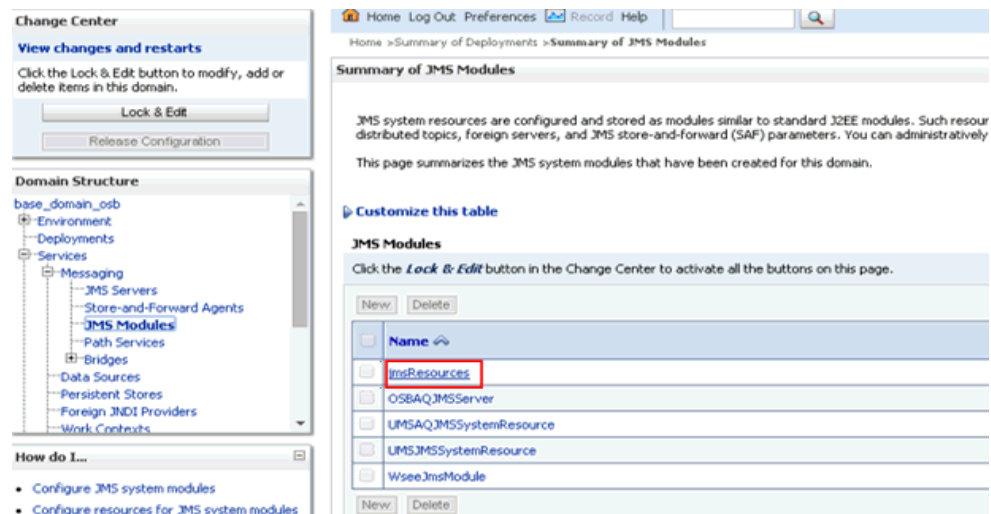
- d. In the displayed Resource Chooser window, select the WSDL-based Business service (for example, xxxxx\_BS.bix) and click **OK**.  
You are returned to the Pipeline configuration page.
- e. Save and Close the Pipeline configuration page.  
You are returned to the composite editor window.
- f. Click **Save All** in the menu bar to save the OSB JMS process, as shown in [Figure 8–91](#).

**Figure 8–91 Transport Window**



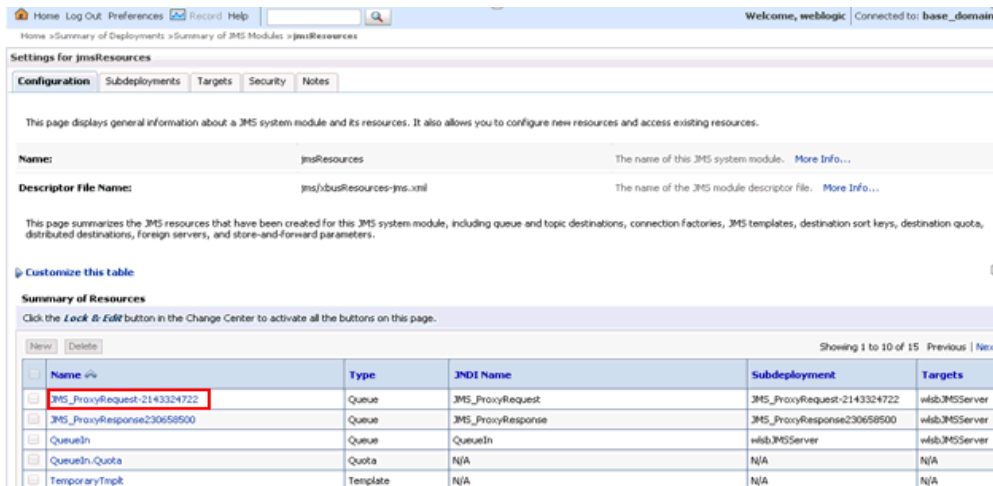
8. Deploy the OSB JMS outbound process. For more information, see [Section 8.1.3, "Deploying the OSB Outbound Process"](#) on page 8-15.
9. Once the process is deployed successfully, log on to the Oracle WLS Console.
10. In the Oracle WLS console, expand **Services**, click **Messaging**, select **JMS Modules**, and then click **jmsResources**, as shown in [Figure 8–92](#).

**Figure 8–92 JMS Resources**



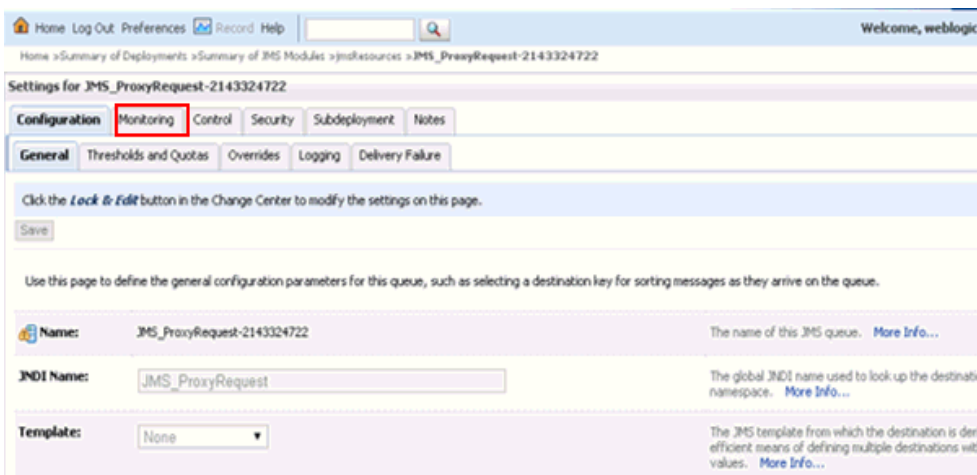
11. Click the appropriate request link (for example, JMS\_ProxyRequest) as shown in [Figure 8–93](#).

**Figure 8–93 JMS\_ProxyRequest Link**



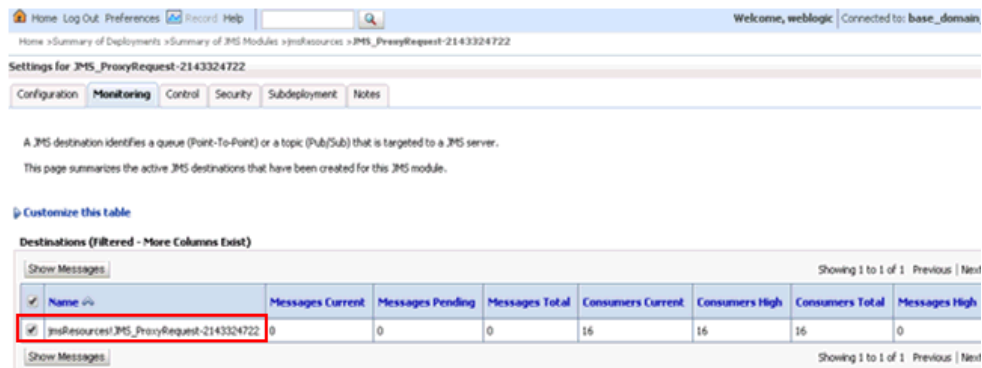
12. Click the Monitoring tab, as shown in Figure 8–94.

**Figure 8–94 Monitoring Tab**



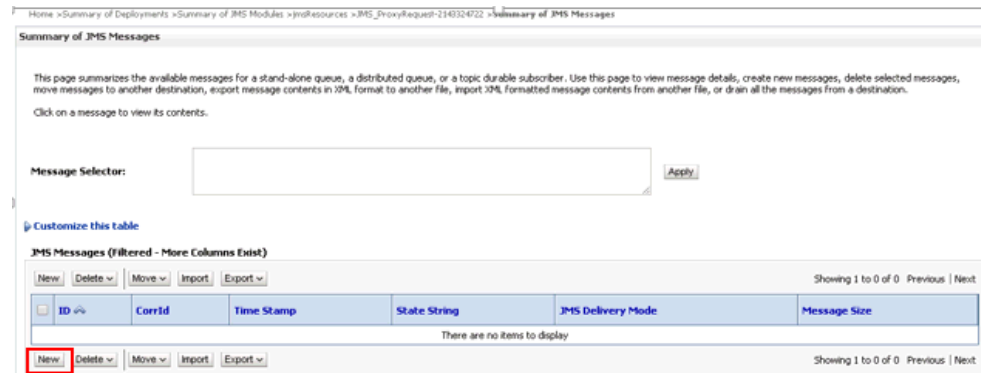
13. Select the check box and click the **Show Messages** button, as shown in Figure 8–95.

**Figure 8–95 Show Messages Button**



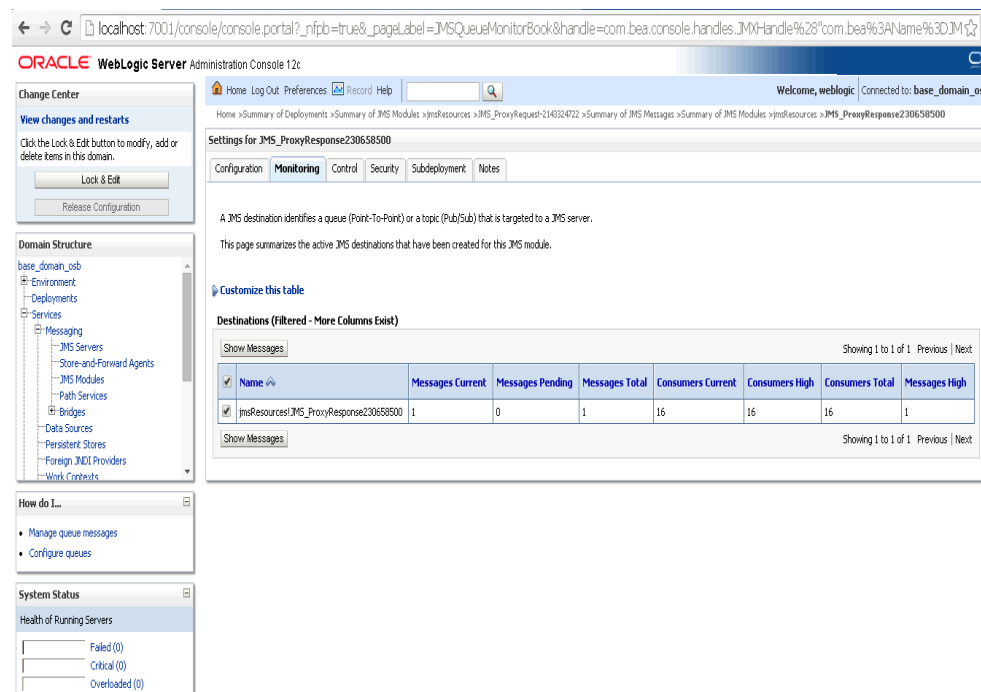
- Click **New**, as shown in [Figure 8–96](#).

**Figure 8–96 JMS Messages**



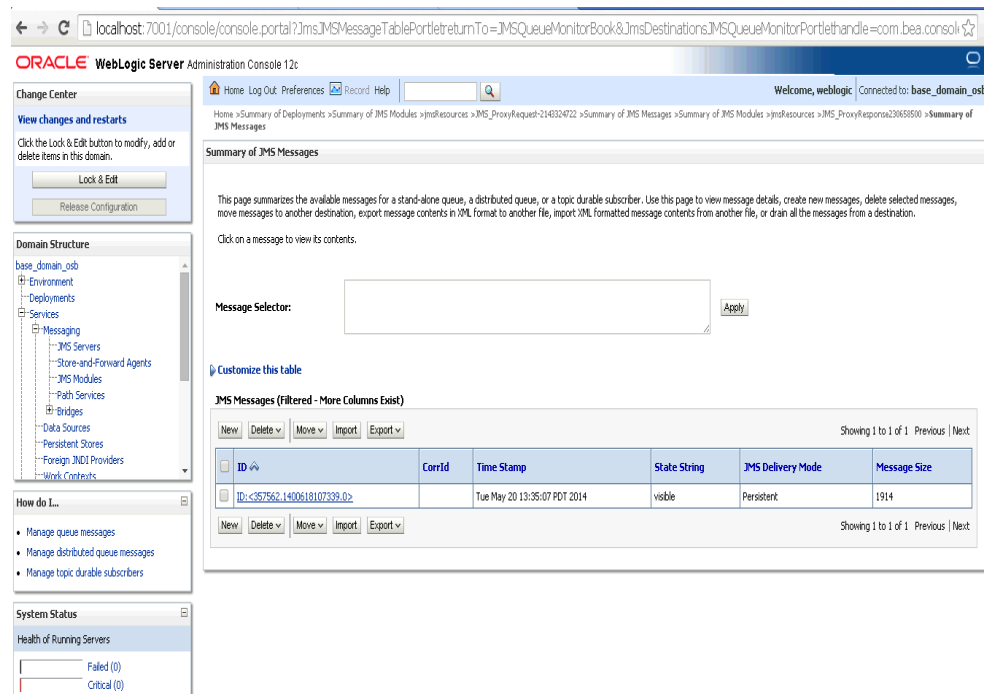
- Provide the input payload in the Body field and click **OK**.
- In the Oracle WLS console, expand **Services**, click **Messaging**, select **JMS Modules**, and then click **jmsResources**.
- Click the appropriate response link (for example, **JMS\_ProxyResponse**).
- Click the **Monitoring** tab.
- Select the check box and click **Show Messages**, as shown in [Figure 8–97](#).

**Figure 8–97 Destination Messages**



- Click the ID link with the appropriate time and date, as shown in [Figure 8–98](#).

**Figure 8–98 Summary of JMS Messages Window**



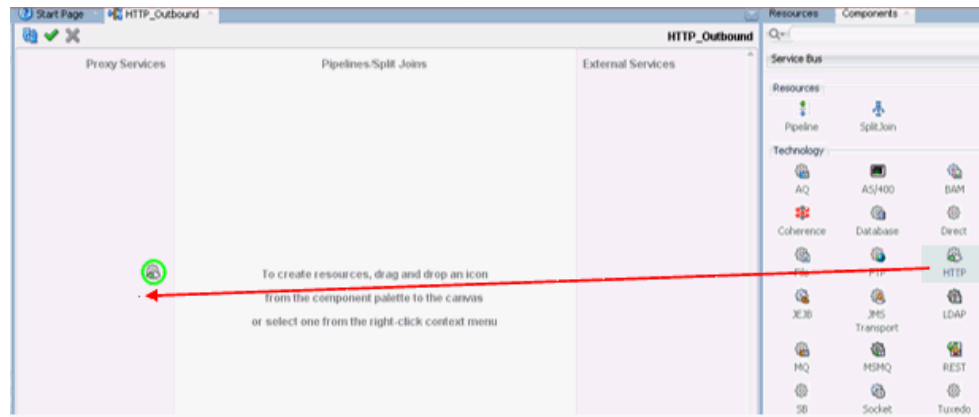
The response document is shown under the Text field.

## 8.6 Configuring an HTTP Outbound Process Using JDeveloper (J2CA Configuration)

This section describes how to configure HTTP Outbound process to your PeopleSoft system, using Oracle JDeveloper for J2CA configurations.

1. Before you design an HTTP Outbound process, you must generate the respective WSDL file using Application Explorer. For more information, see [Section 4.4.1, "Generating WSDL for Request/Response Service"](#) on page 4-8.
2. Start the Oracle JDeveloper and create a Service Bus Application for OSB. For more information, see [Section 8.1.1, "Creating a Service Bus Application for OSB"](#) on page 8-2.
3. Create a Third Party Adapter Service Component. For more information, see [Section 8.1.2.1, "Configuring a Third-Party Adapter Service Component"](#) on page 8-3.
4. Create an HTTP Proxy Service with a Pipeline and perform the following steps:
  - a. Drag and drop the **HTTP** component from the Technology Components pane to the Proxy Services pane, as shown in [Figure 8–99](#).

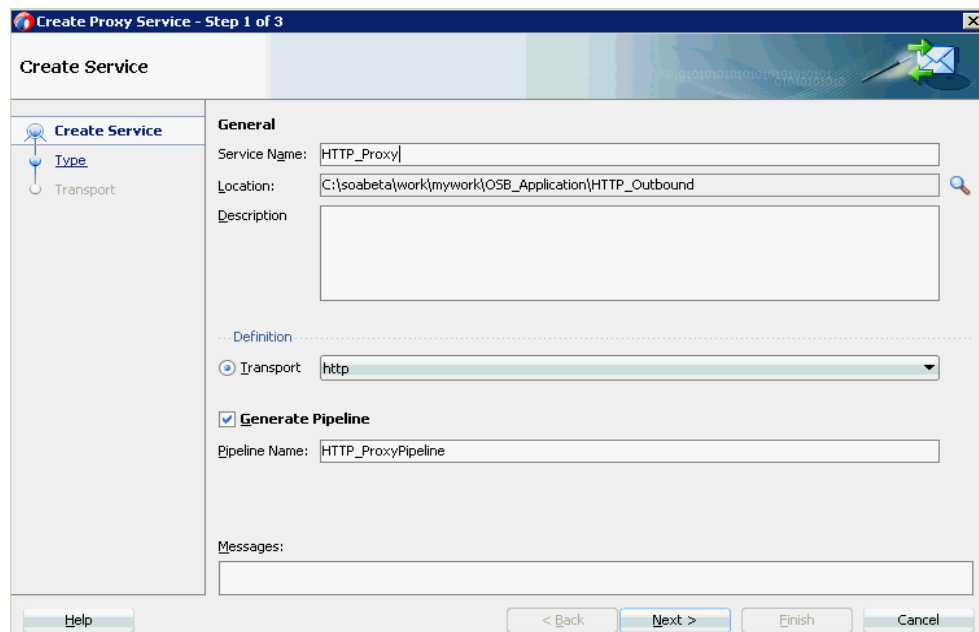
**Figure 8–99 HTTP Component**



The Create Proxy Service dialog is displayed.

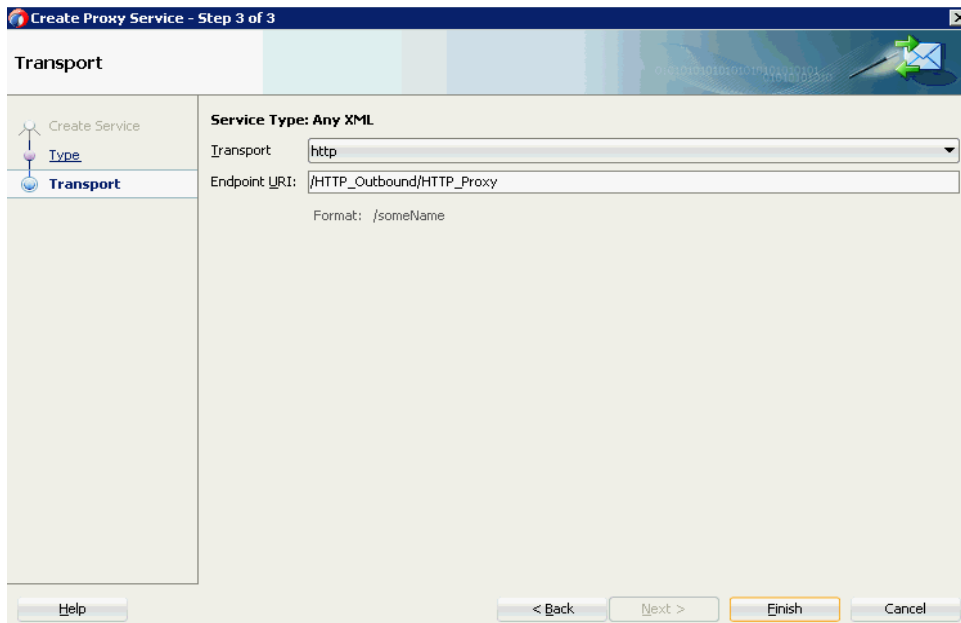
- b. In the Service Name field, enter any name you wish for the Proxy service (for example, HTTP\_Proxy). By default, Generate Pipeline is selected.
- c. Click **Next**, as shown in [Figure 8–100](#).

**Figure 8–100 Create Proxy Service Pane**



- d. In the displayed Type window, select **Any XML** and then click **Next**.  
The Transport window is displayed.
- e. Leave the default values and then click **Finish**, as shown in [Figure 8–101](#).

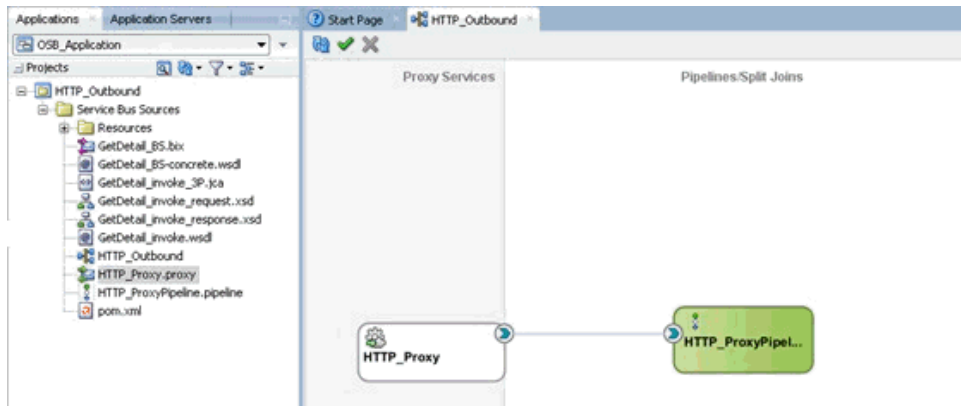
**Figure 8–101 Transport Window**



The HTTP Proxy service along with the pipeline is created and displayed.

- f. Double-click the created pipeline (for example, HTTP\_ProxyPipeline) in the Pipelines/Split Joins pane, as shown in [Figure 8–102](#).

**Figure 8–102 Proxy Service**

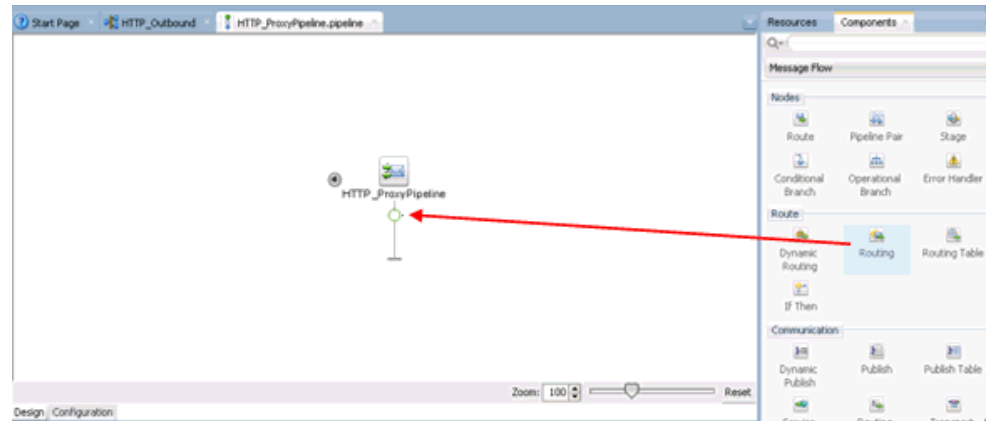


The Pipeline Configuration page is displayed.

5. Configure the Routing Rules and proceed with the following steps:
  - a. Drag and drop the **Routing** component from the Route section to the area below the Pipeline (for example, HTTP\_ProxyPipeline), as shown in [Figure 8–103](#).

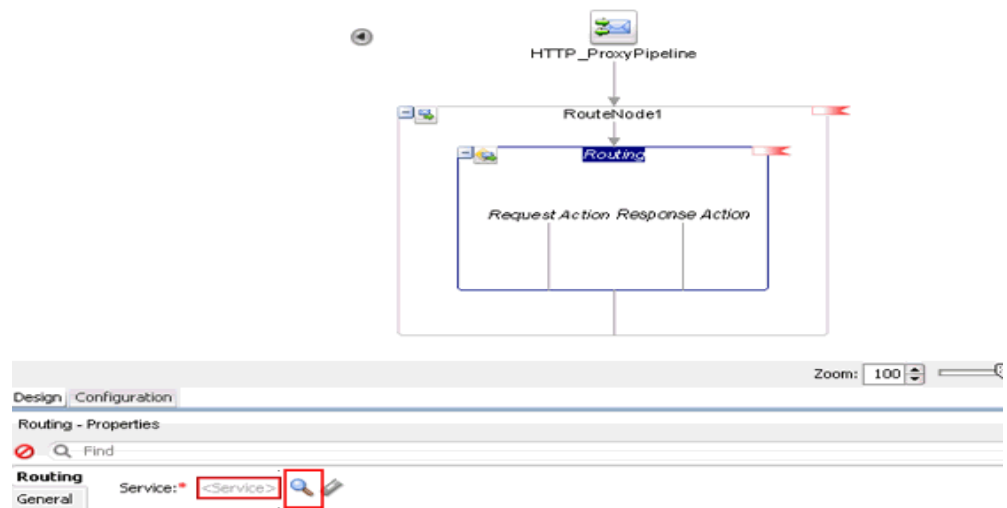


**Figure 8–103 Routing Component**



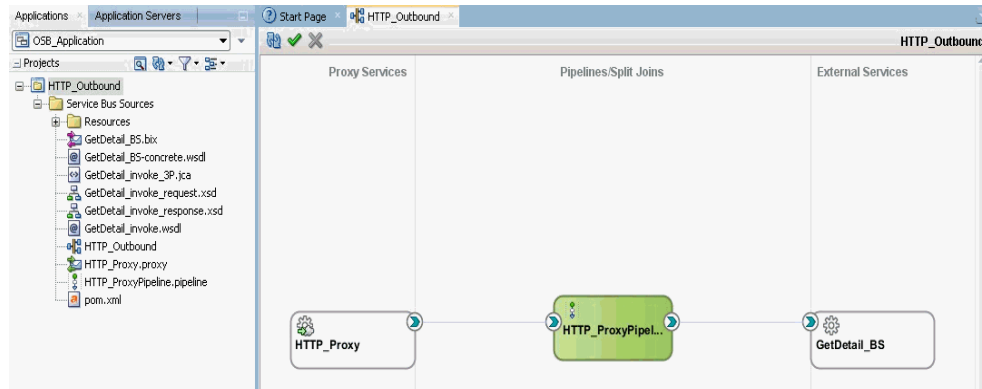
- b. In the Pipeline Configuration page, select **Routing** and click the browse icon to the right of the Service field in the Routing Properties pane, as shown in [Figure 8–104](#).

**Figure 8–104 Browse Service**



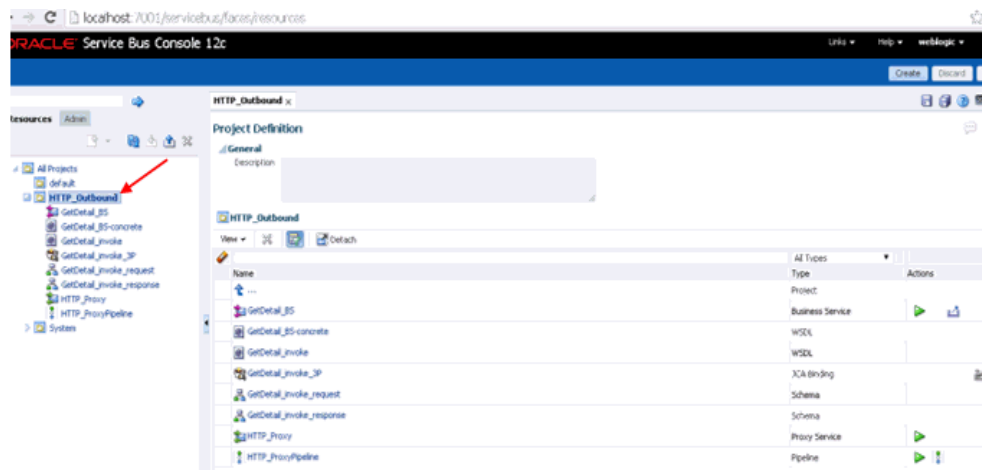
- c. In the displayed Resource Chooser window, select the WSDL-based Business service (for example, xxxxx\_BS.bix) and click **OK**.  
You are returned to the Pipeline configuration page.
- d. Save and Close the Pipeline configuration page.  
You are returned to the composite editor window.
- e. Click **Save All** in the menu bar to save the OSB HTTP process, as shown in [Figure 8–105](#).

**Figure 8–105 Transport Window**



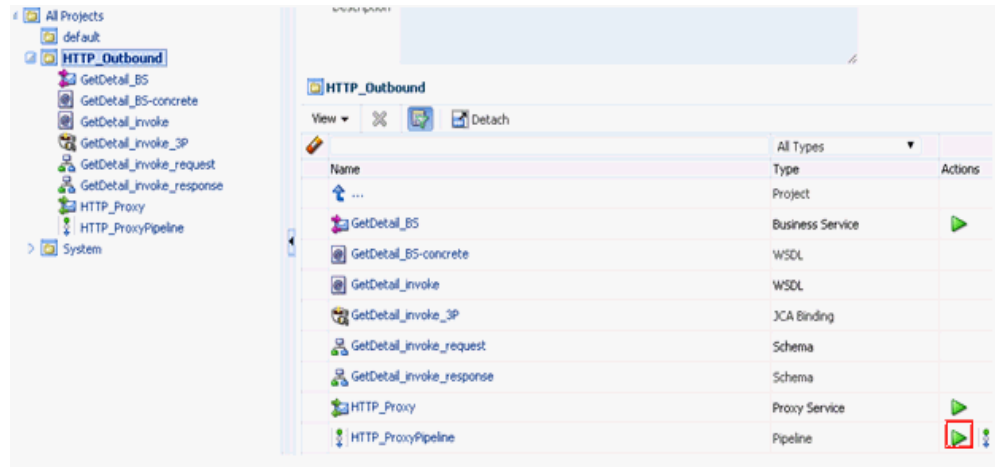
6. Deploy the OSB HTTP outbound process. For more information, see [Section 8.1.3, "Deploying the OSB Outbound Process"](#) on page 8-15.
7. Once the process is deployed successfully, log on to the Service Bus Console.
8. In the Service Bus console, click on the deployed HTTP Outbound project (for example, HTTP\_Outbound), as shown in [Figure 8–106](#).

**Figure 8–106 Service Bus Console**



9. Click on the Test OSB Console icon for the created pipeline, as shown in [Figure 8–107](#).

**Figure 8–107 Test OSB Console Icon**



- In the displayed Test OSB Console page, provide the input XML and click the **Execute** button.

In the displayed Test OSB Console page, the response is received.



---

---

## Key Features

This chapter describes key features for the Oracle Application Adapter for PeopleSoft. This chapter contains the following topics:

- [Section 9.1, "Configuring the Logging Feature"](#)
- [Section 9.2, "Configuring the Diagnosibility Feature"](#)
- [Section 9.3, "Configuring the SOA Debugging Feature"](#)
- [Section 9.4, "Exception Filter"](#)
- [Section 9.5, "Credential Mapping for Oracle SOA Suite \(BPEL, Mediator, or BPM\)"](#)
- [Section 9.6, "Credential Mapping for Oracle Service Bus \(OSB\)"](#)

### 9.1 Configuring the Logging Feature

In Oracle 12c (12.2.1.0.0), J2CA and BSE adapter logs will be updated in Oracle logs in the *{server-name}-diagnostic.log* file available in the following location:

```
<ORACLE_HOME>\user_projects\domains\base_domain\servers\<server_Name>\logs.
```

---

---

**Note:** The Application Explorer log files for J2CA would be created under the `<ADAPTER_HOME>\config\xxxxxxx\log` folder where `xxxxxxx` is the name of the J2CA configuration that was created in Application Explorer. Each J2CA configuration in Application Explorer has a corresponding log folder under the named J2CA configuration folder.

---

---

This section describes how to configure the Logging feature. It contains the following topics:

- [Section 9.1.1, "Configuring Log File Management for the J2CA Connector Application"](#)
- [Section 9.1.2, "Configuring Log File Management for Business Services Engine \(BSE\)"](#)

#### 9.1.1 Configuring Log File Management for the J2CA Connector Application

Log file management for the J2CA Connector Application is governed by the Loggers defined in:

```
<ORACLE_HOME>\user_projects\domains\base_
```

domain\config\fmwconfig\servers\\${server-name}\logging.xml

Any new loggers will have to be added to this file if they are to be managed from the em console.

For example:

```
<logger name='oracle.soa.adapter.iwaf' level='NOTIFICATION:1'
useParentHandlers='true' />
<logger name='oracle.soa.adapter.iwaf.classloader' level='NOTIFICATION:1'
useParentHandlers='true' />
<logger name='oracle.soa.adapter.iwaf.connection' useParentHandlers='true' />
<logger name='oracle.soa.adapter.iwaf.connection.IAEAdapter'
useParentHandlers='true' />
<logger name='oracle.soa.adapter.iwaf.connection.Sample'
useParentHandlers='true' />
<logger name='oracle.soa.adapter.iwaf.connection.PeopleSoft'
useParentHandlers='true' />
<logger name='oracle.soa.adapter.iwaf.inbound' useParentHandlers='true' />
<logger name='oracle.soa.adapter.iwaf.inbound.IAEAdapter'
useParentHandlers='true' />
<logger name='oracle.soa.adapter.iwaf.inbound.Sample' useParentHandlers='true' />
<logger name='oracle.soa.adapter.iwaf.inbound.PeopleSoft'
useParentHandlers='true' />
<logger name='oracle.soa.adapter.iwaf.outbound' useParentHandlers='true' />
<logger name='oracle.soa.adapter.iwaf.outbound.IAEAdapter'
useParentHandlers='true' />
<logger name='oracle.soa.adapter.iwaf.outbound.Sample' useParentHandlers='true' />
<logger name='oracle.soa.adapter.iwaf.outbound.PeopleSoft'
useParentHandlers='true' />
<logger name='oracle.soa.adapter.iwaf.transaction' useParentHandlers='true' />
<logger name='oracle.soa.adapter.iwaf.transaction.IAEAdapter'
useParentHandlers='true' />
<logger name='oracle.soa.adapter.iwaf.transaction.Sample'
useParentHandlers='true' />
<logger name='oracle.soa.adapter.iwaf.transaction.PeopleSoft'
useParentHandlers='true' />
<logger name='oracle.soa.adapter.iwaf.IAEAdapter' useParentHandlers='true' />
<logger name='oracle.soa.adapter.iwaf.Sample' useParentHandlers='true' />
<logger name='oracle.soa.adapter.iwaf.PeopleSoft' useParentHandlers='true' />
```

This sets the logging level of all the loggers under oracle.soa.adapter.iwaf to NOTIFICATION:1 (INFO), which is the default setting level by Oracle.

The logging level of all the loggers can also be configured from the em console with the following steps:

1. Start the Oracle WebLogic Server for the Oracle WebLogic Server domain that you configured.
2. Open the Oracle WebLogic Server Enterprise Manager Console in a web browser by entering the following URL:

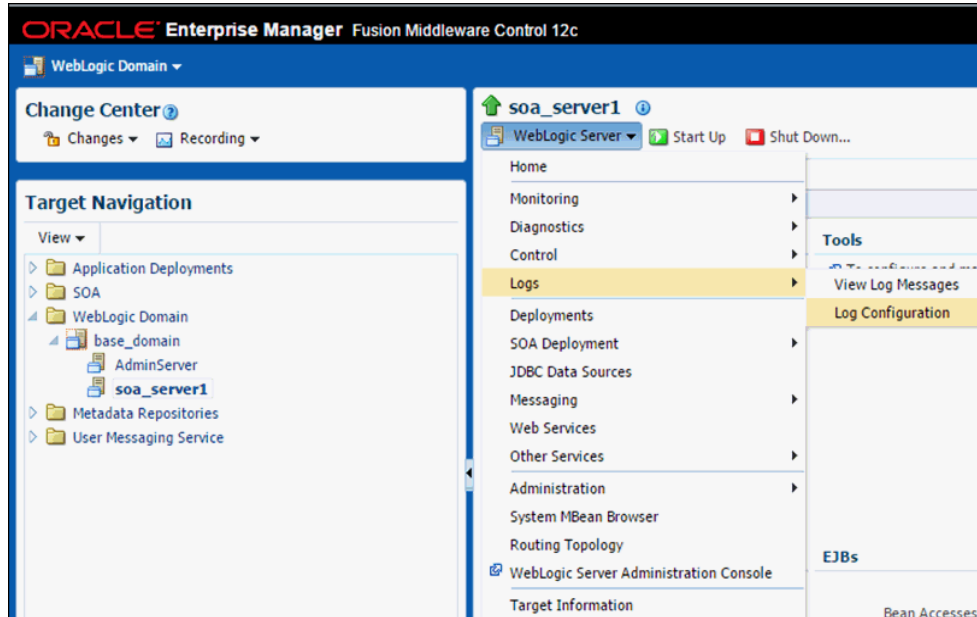
```
http://host name:port/em
```

where `host name` is the name of the system where Oracle WebLogic Server is running and `port` is the port for the Oracle WebLogic Server that is running. The default port for the Oracle WebLogic Server is 7001. However, this value can vary between installations.

3. Log in to the Oracle WebLogic Server Administrative Console using an account that has administrator privileges.

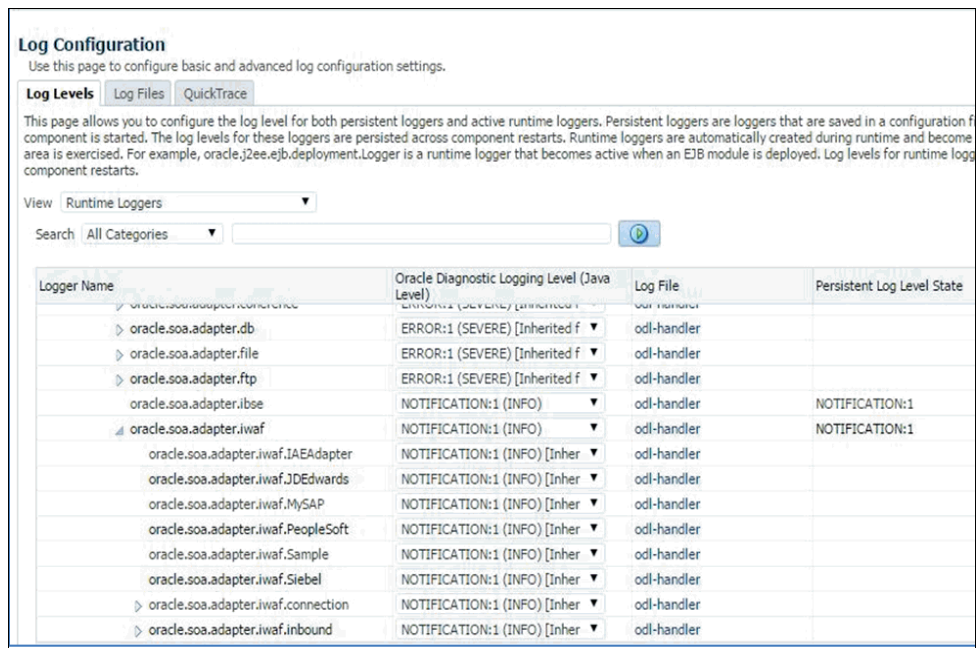
4. Under the Target Navigation pane, click **Weblogic Domain**, select **Domain Created**, and click the appropriate server (Managed Server or Integrated Server).
5. In the Server pane, expand WebLogic Server, select **Logs**, and then click **Log Configuration**, as shown in [Figure 9-1](#).

**Figure 9-1 Log Configuration Option**



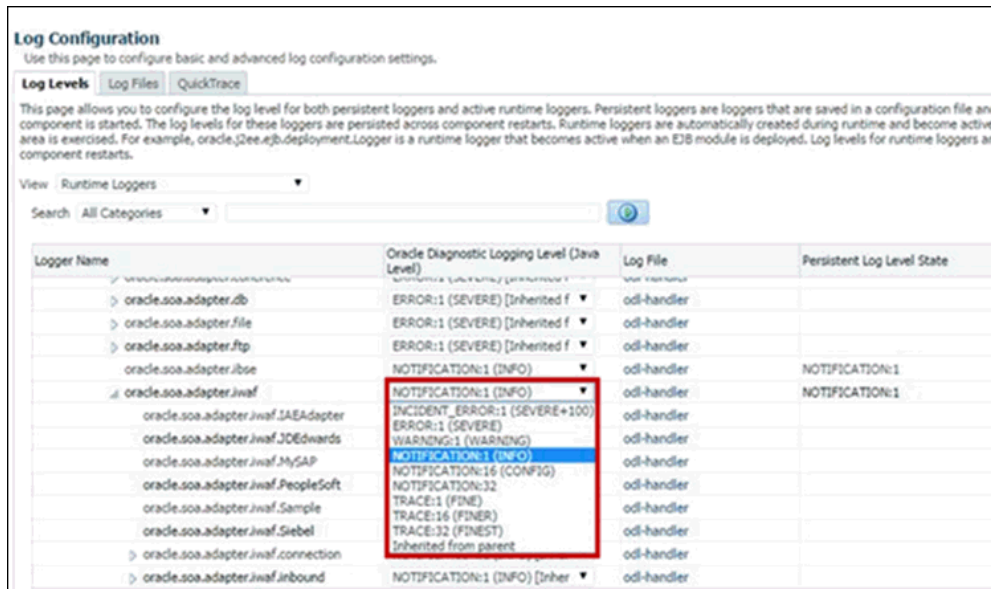
6. Under the **Log Levels** tab, expand the Oracle root logger until `oracle.soa.adapter.iwaf` is visible, as shown in [Figure 9-2](#).

**Figure 9-2 Log Levels Tab**



- In the Oracle Diagnostic Logging Level (Java Level) column, select the required log level from the oracle.soa.adapter.iwaf drop-down list, as shown in [Figure 9-3](#).

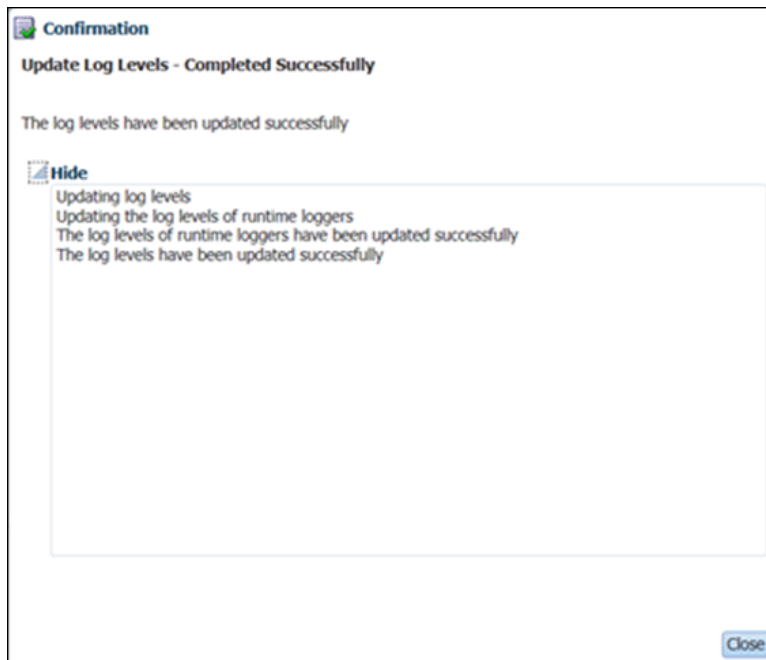
**Figure 9-3 Oracle Diagnostic Logging Level (Java Level) Column**



- Click **Apply**.

A confirmation message appears, indicating that the update was completed successfully, as shown in [Figure 9-4](#).

**Figure 9-4 Confirmation Message**



- Click **Close**.



The following table shows how the Log Level property is updated in the `{server-name}-diagnostic.log` based on the corresponding Log Level property settings in the em console.

**Table 9–1 Log Level Property Settings**

Log Level Set in Em Console	Log Level Updated in <code>{server-name}-diagnostic.log</code>
ERROR:1	ERROR
WARNING:1	WARNING
NOTIFICATION:1	NOTIFICATION
NOTIFICATION:16	NOTIFICATION
NOTIFICATION:32	NOTIFICATION
TRACE:1	NOTIFICATION
TRACE:16	NOTIFICATION
TRACE:32	TRACE:32

**Note:** Setting the LogLevel as TRACE:32 in the em console, displays the FINEST details in the log (displaying the input passed to the adapter, response received from the adapter and other additional details) with the log level displayed as TRACE:32 in the `{server-name}-diagnostic.log`.

For development and test environments, TRACE:32 is the preferred log level, which displays all of the log details. For production environments, ERROR is the preferred log level.

The log messages are written to a disk file, and the file path can be found in the **Log Files** tab. The Handler Name in the Log Files table corresponds to the Log File name in the Log Levels table. All loggers in the hierarchy below `oracle.soa.adapter` are currently handled by the odl-handler, as shown in [Figure 9–5](#).

**Figure 9–5 Log Configuration Pane**

**Log Configuration**  
Use this page to configure basic and advanced log configuration settings.

Log Levels | **Log Files** | QuickTrace

Use this page to create and edit log file configurations. A log file configuration specifies the log file where the log messages will be logged to, the format of the log messages, the rotation policies used, as well as other parameters depending on the log file configuration class.

Create... Create Like... Edit Configuration... View Configuration... Delete Configuration...

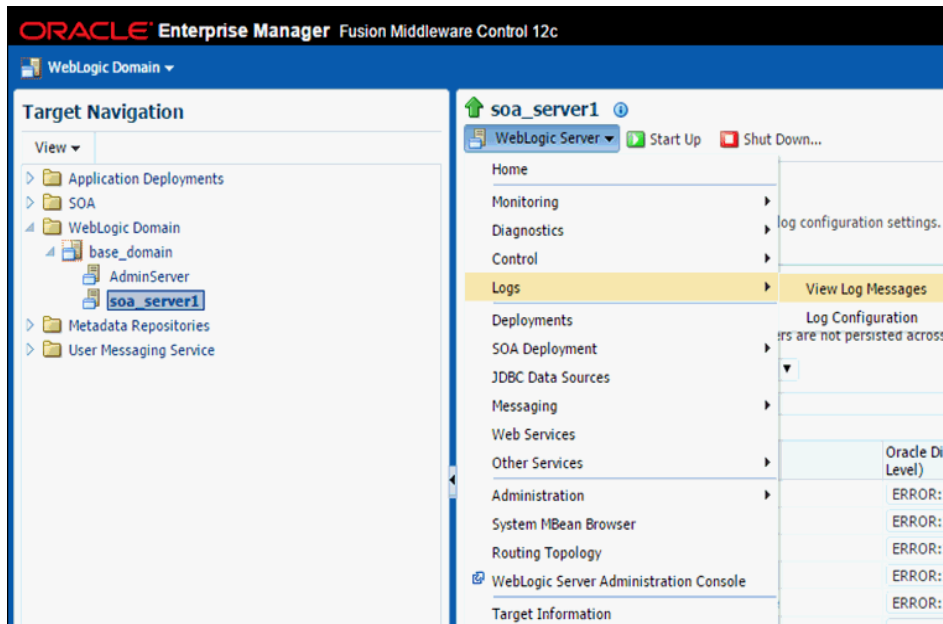
Handler Name	Log Path	Log File Format
em-log-handler	\${domain.home}/servers/\${weblogic.Name}/sysman/log/emoms.log	Oracle Diagnostics Logging
em-trc-handler	\${domain.home}/servers/\${weblogic.Name}/sysman/log/emoms.trc	Oracle Diagnostics Logging
odl-handler	\${domain.home}/servers/\${weblogic.Name}/logs/\${weblogic.Name}-diagnostic.log	Oracle Diagnostics Logging
owsm-message-ha...	\${domain.home}/servers/\${weblogic.Name}/logs/owsm/msglogging/diagnostic.log	Oracle Diagnostics Logging
soa-tracking-trc-h...	\${domain.home}/servers/\${weblogic.Name}/logs/\${weblogic.Name}-soa-tracking.trc	Oracle Diagnostics Logging

The logs are updated in `{server-name}-diagnostic.log` available in the following location:

```
<ORACLE_HOME>\user_projects\domains\base_domain\servers\<server_Name>\logs
```

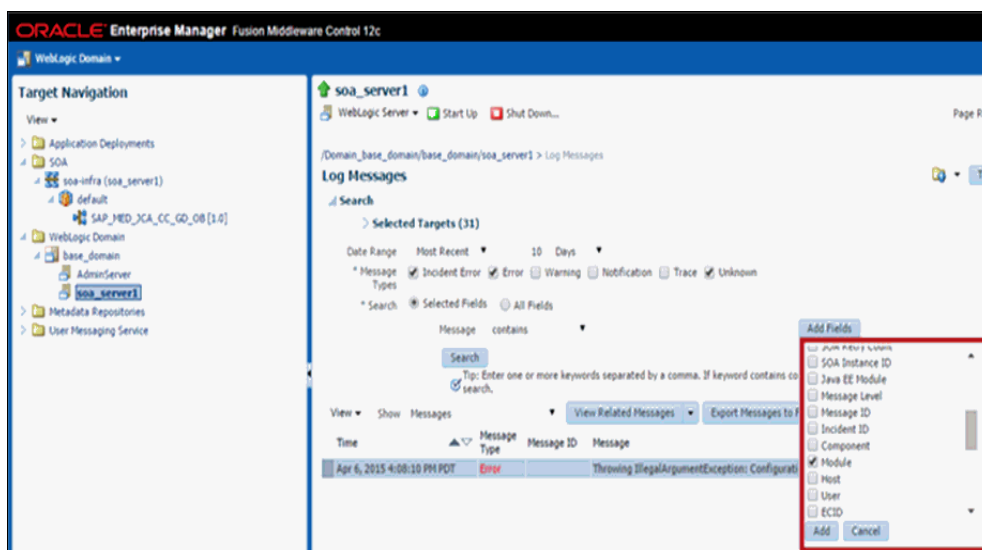
10. In the Server pane, display the log messages in the em console by clicking the **WebLogic Server** drop-down list, selecting **Logs**, and then clicking **View Log Messages**, as shown in [Figure 9-6](#).

**Figure 9-6 View Log Messages Option**



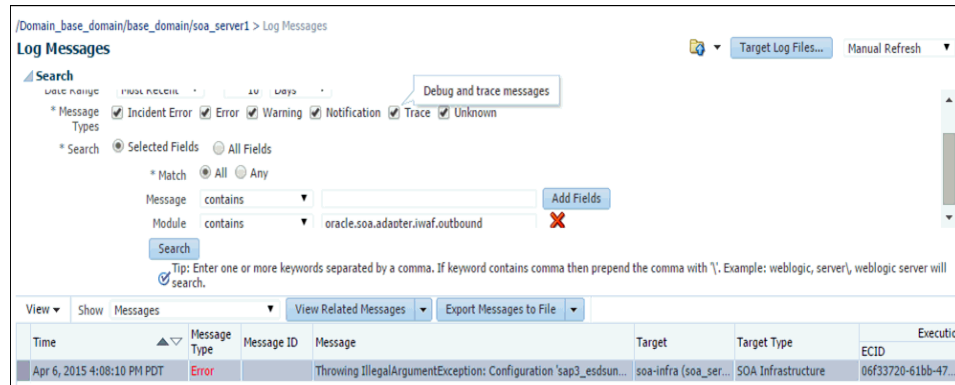
11. On the Log Messages pane, complete the required search criteria. You can also add the **Module** field to the search criteria, which contains the name of the logger of interest, as shown in [Figure 9-7](#).

**Figure 9-7 Module Field**



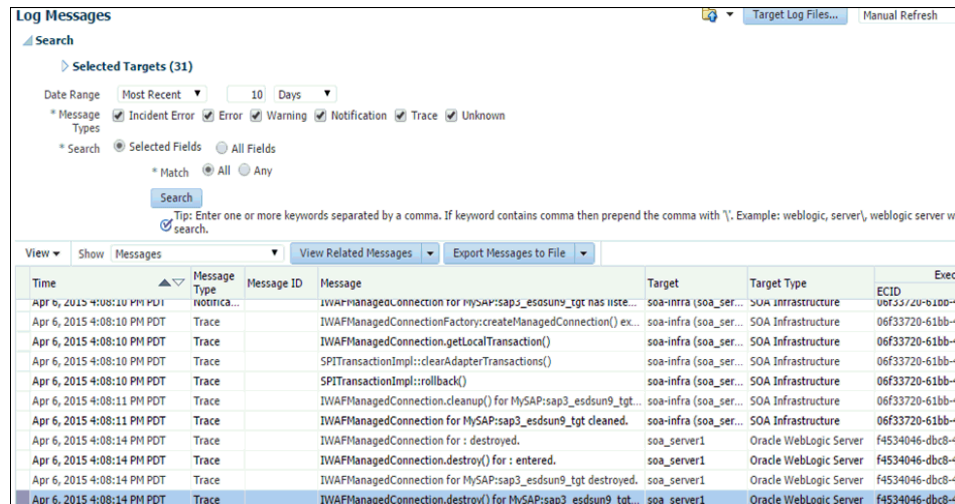
12. Click **Add**.
13. In the **Module** field, enter the name of the logger of interest, and if required, select the additional **Message Types** (**Warning**, **Notification**, **Trace**, and so on) and then click **Search**, as shown in [Figure 9-8](#).

Figure 9–8 Log Messages Pane



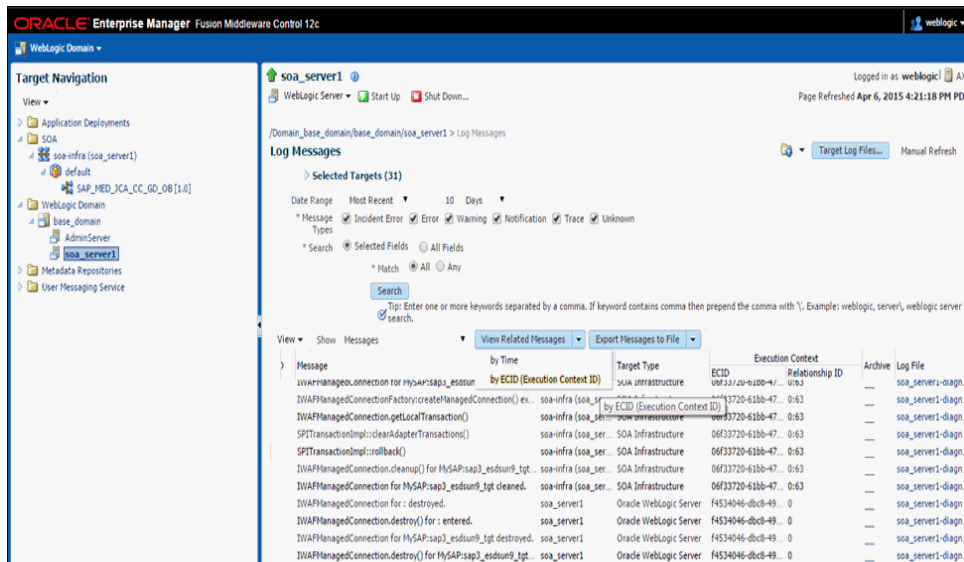
The messages from the specified logger are displayed in the table below the search criteria, as shown in Figure 9–9.

Figure 9–9 Logger Messages



14. Select any row in the table. To get identical details, click the **View Related Messages** drop-down list, and select **ECID** (execution Context ID) as shown in Figure 9–10.

Figure 9–10 ECID Option



Details are displayed, as shown in Figure 9–11.

Figure 9–11 Message Details

Time	Message Type	Message ID	Message	Target	Target Type	Archive	Log File
Apr 6, 2015 4:08:10 PM PDT	Trace		IWAFManagedConnection.getLocalTransaction()	soa-infra (soa_ser...	SOA Infrastructure	---	soa_ser...
Apr 6, 2015 4:08:10 PM PDT	Trace		SPTTransactionImpl::clearAdapterTransactions()	soa-infra (soa_ser...	SOA Infrastructure	---	soa_ser...
Apr 6, 2015 4:08:10 PM PDT	Trace		SPTTransactionImpl::rollback()	soa-infra (soa_ser...	SOA Infrastructure	---	soa_ser...
Apr 6, 2015 4:08:11 PM PDT	Trace		IWAFManagedConnection.cleanup() for MySAP:sap3_esdsun9_tgt...	soa-infra (soa_ser...	SOA Infrastructure	---	soa_ser...
Apr 6, 2015 4:08:11 PM PDT	Trace		IWAFManagedConnection for MySAP:sap3_esdsun9_tgt cleaned.	soa-infra (soa_ser...	SOA Infrastructure	---	soa_ser...
Apr 6, 2015 4:08:11 PM PDT	Error		JCABinding=> SAP_MED_JCA_CC_GD_DB:GetDetail [ GetDetail...	soa-infra (soa_ser...	SOA Infrastructure	---	soa_ser...
Apr 6, 2015 4:08:11 PM PDT	Error		JCABinding=> [default/SAP_MED_JCA_CC_GD_OB1.0]soa_3158...	soa-infra (soa_ser...	SOA Infrastructure	---	soa_ser...
Apr 6, 2015 4:08:11 PM PDT	Error		Rolling back transaction due to ORAMED-03303:[Unexpected exc...	soa-infra (soa_ser...	SOA Infrastructure	---	soa_ser...
Apr 6, 2015 4:08:11 PM PDT	Error		Error in rolling back transaction due to ORAMED-03504:[Transacti...	soa-infra (soa_ser...	SOA Infrastructure	---	soa_ser...
Apr 6, 2015 4:08:11 PM PDT	Error		Updating fault processing DMS metrics	soa-infra (soa_ser...	SOA Infrastructure	---	soa_ser...
Apr 6, 2015 4:08:11 PM PDT	Error		Got an exception: oracle.fabric.common.FabricInvocationExceptio...	soa-infra (soa_ser...	SOA Infrastructure	---	soa_ser...

Message Level	32	Java EE	soa-infra
SRC_CLASS	com.ibm.afjca.spi.SPTTransactionImpl	Application	
WebService Port	execute_pt	Relationship ID	0:63
SCA Entity ID	340004	Component	soa_server1
SOA Retry Count	SAP_MED_JCA_CC_GD_OB1.0	Module	oracle.soa.adapter.iwaf.outbound
SOA Instance ID	320003	Host	AXTST168

### 9.1.2 Configuring Log File Management for Business Services Engine (BSE)

Similar to J2CA for BSE, the Log file management is governed by the Loggers defined in:

```
<ORACLE_HOME>\user_projects\domains\base_
domain\config\fmwconfig\servers\${server-name}\logging.xml
```

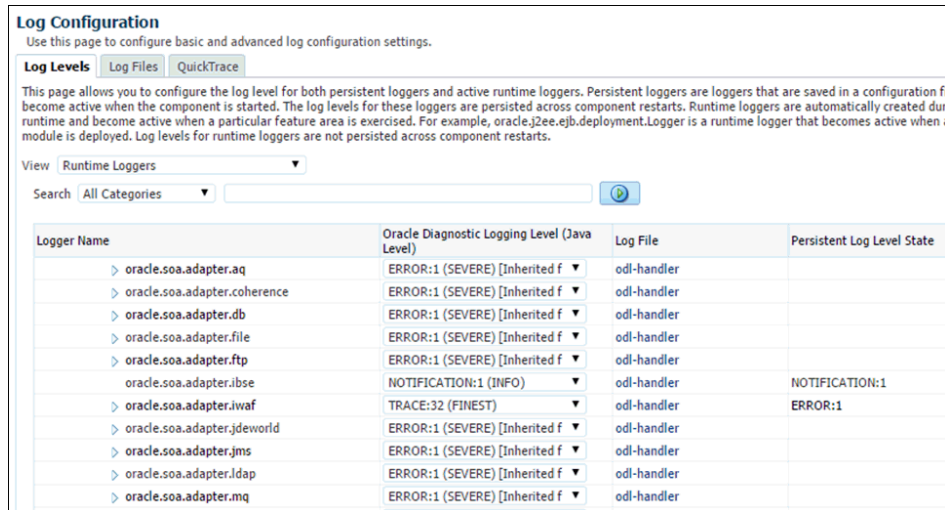
The following syntax sets the logging level of all the loggers under oracle.soa.adapter.ibse to NOTIFICATION:1 (INFO) which is the default setting level by oracle.

```
<logger name='oracle.soa.adapter.ibse' level='NOTIFICATION:1'
useParentHandlers='true' />
```

The logging level of all the loggers can also be configured from the em console with the following steps:

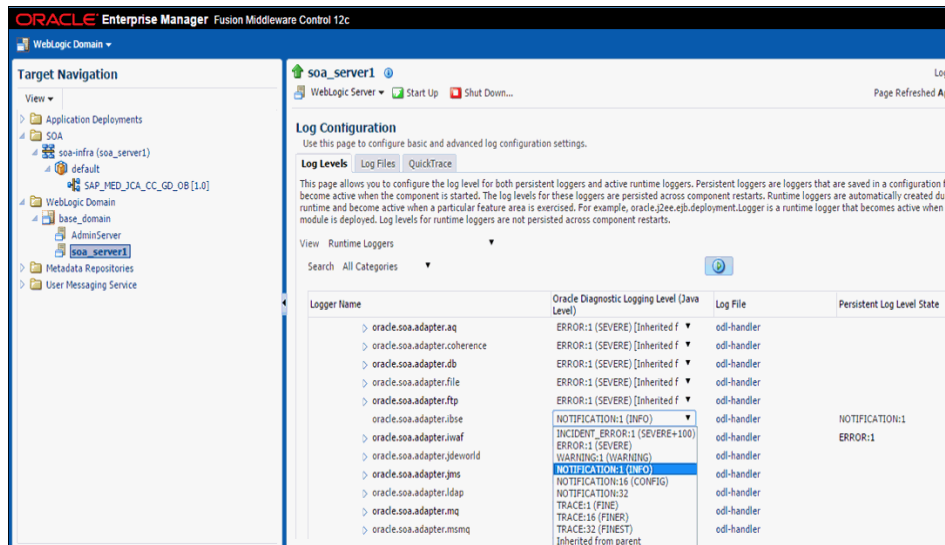
1. Repeat steps 1 through 5, as described in Section 9.1.1, "Configuring Log File Management for the J2CA Connector Application".
2. Under the **Log Levels** tab, expand the Oracle root logger until the oracle.soa.adapter.ibse Logger name is visible, as shown in Figure 9–12.

**Figure 9–12 Log Levels Tab**



3. In the Oracle Diagnostic Logging Level (Java Level) column, select the required log level from the oracle.soa.adapter.ibse drop-down list, as shown in Figure 9–13.

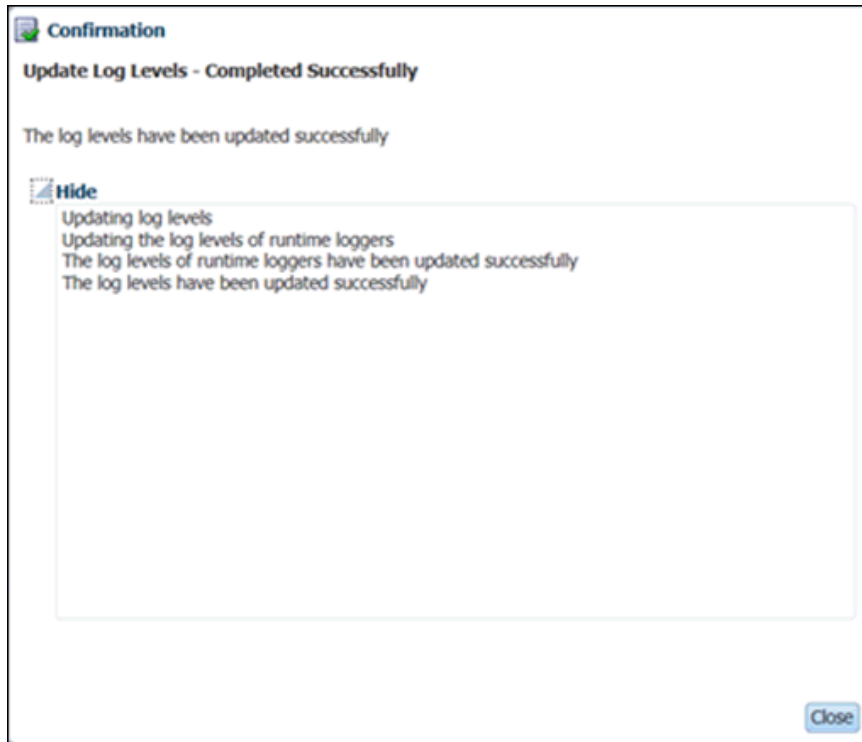
**Figure 9–13 Oracle Diagnostic Logging Level Column**



4. Click **Apply**.

A confirmation message appears, indicating that the update was completed successfully, as shown in Figure 9–14.

**Figure 9–14 Confirmation Message**



5. Click **Close**.

The following table shows how the Log Level property is updated in the *{server-name}-diagnostic.log* based on the corresponding Log Level property settings in the em console.

**Table 9–2 Log Level Property Settings**

Log Level Set in Em Console	Log Level Updated in {server-name}-diagnostic.log
ERROR:1	ERROR
WARNING:1	WARNING
NOTIFICATION:1	NOTIFICATION
NOTIFICATION:16	NOTIFICATION
NOTIFICATION:32	NOTIFICATION
TRACE:1	NOTIFICATION
TRACE:16	NOTIFICATION
TRACE:32	TRACE:32

For verification of logs, see steps 11 to 18 of the J2CA Logging feature, found in [Section 9.1.1, "Configuring Log File Management for the J2CA Connector Application"](#).

## 9.2 Configuring the Diagnosibility Feature

This section describes how to configure and use the Diagnosibility feature for the Oracle Fusion Middleware Application Adapters for Oracle WebLogic Server. It contains the following topic:

- [Section 9.2.1, "Supporting Protocols"](#)

The Diagnosibility feature captures the endpoint health status (where available) of the adapters, and provides a corresponding alert to the Oracle Adapter Framework, so it may be displayed in the EM console.

---

**Note:** The Diagnosibility feature supports only inbound adapter processes

Make sure that there is an inbound process deployed before moving to the next section

---

This information can be viewed for the adapters as mentioned in the following steps:

1. Start the Oracle WebLogic Servers and open the Oracle WebLogic Server Enterprise Manager Console in a web browser by entering the following URL:

`http://host name:port/em`

Where *host name* is the name of the system where Oracle WebLogic Server is running, and *port* is the port for the Oracle WebLogic Server that is running.

2. Log in to the Oracle WebLogic Server Administrative Console using an account that has administrator privileges.
3. On the right pane, expand **SOA, soa-Infra (server\_name), Default**, and then **Deployed inbound process**.
4. Click the process and then select the service listed in the Services and References Section in the right pane, as shown in [Figure 9–15](#).

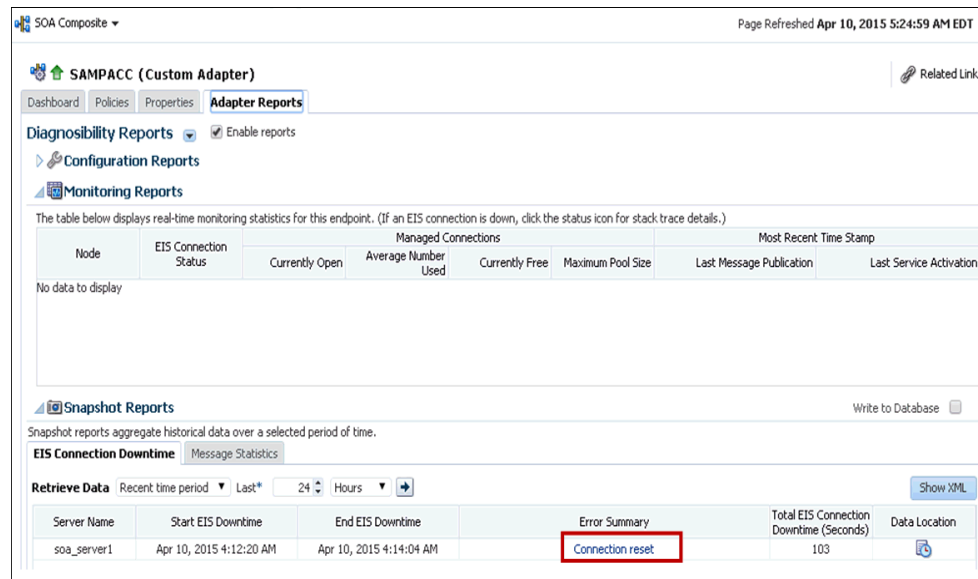
**Figure 9–15 Services and References Section**

The screenshot shows the Oracle WebLogic Server Administrative Console interface. The top navigation bar includes 'SOA Composite' and 'Page Refreshed Apr 23, 2015 12:29:57 PM PDT'. Below the navigation bar, there are tabs for 'Dashboard', 'Composite Definition', 'Flow Instances', 'Unit Tests', and 'Policies'. The 'Services and References' section is expanded, showing a table with the following data:

Name	Type	Usage	Total Messages	Average Processing Time (sec)
Service	JCA Adapter	Service	0	0.000
FileReference	JCA Adapter	Reference	0	0.000

5. In the Adapter Reports tab, select the **Enable reports** check box, and then expand **Snapshot Reports** to view the details of the EIS downtime. Clicking on the Error Summary will show the stack trace, as shown in [Figure 9–16](#).

**Figure 9–16 Stack Trace**



## 9.2.1 Supporting Protocols

This section describes the supporting protocols for the Oracle Fusion Middleware Application Adapter for PeopleSoft, and consists of the following topic:

- [Section 9.2.1.1, "Oracle Fusion Middleware Application Adapter for PeopleSoft Endpoint Status"](#)

### 9.2.1.1 Oracle Fusion Middleware Application Adapter for PeopleSoft Endpoint Status

The Oracle Fusion Middleware Application Adapter for PeopleSoft (inbound) can use the following protocols:

- HTTP
- TCP
- File (Not Supported)

The HTTP and TCP listening protocol adapters listen on a socket. As a result, by their nature, they cannot determine whether there is anything live on the other side until they receive something. Even when receiving a request, it is impossible to determine with certainty where the request originated.

For these protocols, the EIS is regarded and used when a request is being received. EIS determines if the communication error happens while the request is being received. However, this is a very unrefined and rudimentary determination.

## 9.3 Configuring the SOA Debugging Feature

This section describes how to configure and use the SOA Debugging feature for the Oracle Application Adapter for PeopleSoft. It contains the following topics:

- [Section 9.3.1, "Guidelines for Using the SOA Debugger"](#)
- [Section 9.3.2, "Prerequisite"](#)
- [Section 9.3.3, "Debugging a BPEL Process in Oracle JDeveloper"](#)



- [Section 9.3.4, "Debugging an OSB Process in Oracle JDeveloper"](#)

---

**Note:** The SOA Debugging feature is currently supported only for J2CA configurations and it is not applicable for BSE configurations.

For SOA, this feature is explained using a BPEL process. The same is applicable for Mediator and BPM processes.

---

### 9.3.1 Guidelines for Using the SOA Debugger

This section describes guidelines for using the SOA Debugger.

1. Only one client at a time can connect to the SOA Debugger.
2. Adapter endpoint errors are not displayed in the SOA Debugger in Oracle JDeveloper. These errors are logged in the log file.

---

**Note:** The SOA Debugger is currently available for BPEL, Mediator, BPM, and OSB processes with Development mode only.

---

### 9.3.2 Prerequisite

Ensure that the *IntegratedWebLogicServer* domain and a BPEL process are already created in Oracle JDeveloper.

### 9.3.3 Debugging a BPEL Process in Oracle JDeveloper

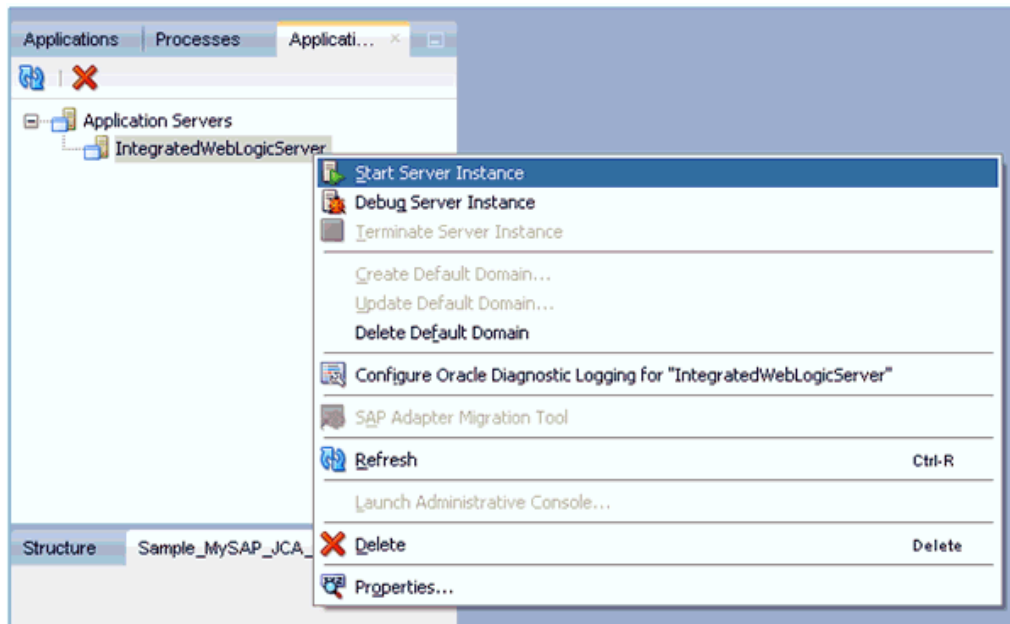
This section describes how to debug a BPEL process in Oracle JDeveloper. It contains the following topics:

- [Section 9.3.3.1, "Debugging an Outbound BPEL Process in Oracle JDeveloper"](#)
- [Section 9.3.3.2, "Debugging an Inbound BPEL Process in Oracle JDeveloper"](#)

#### 9.3.3.1 Debugging an Outbound BPEL Process in Oracle JDeveloper

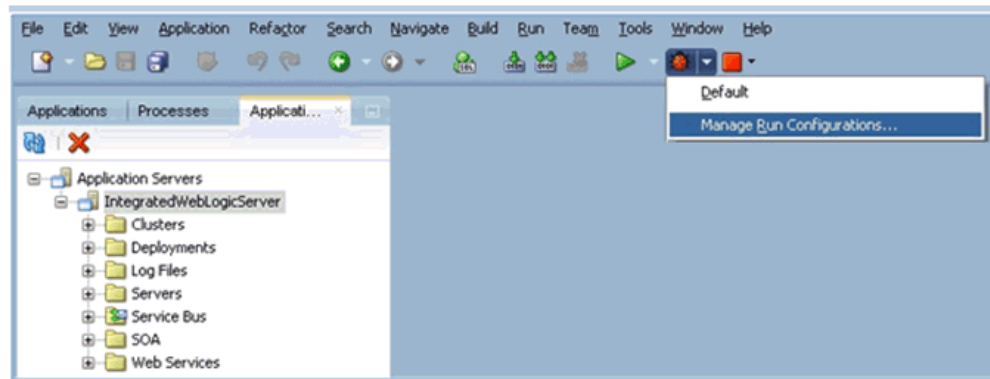
1. Open Oracle JDeveloper.
2. Start the *IntegratedWeblogicServer* domain.
  - a. Click the **Application Servers** tab in the left pane.
  - b. Under the Application Servers node, right-click **IntegratedWeblogicServer** and select **Start Server Instance** from the context menu, as shown in [Figure 9–17](#).

**Figure 9–17 Start Server Instance**



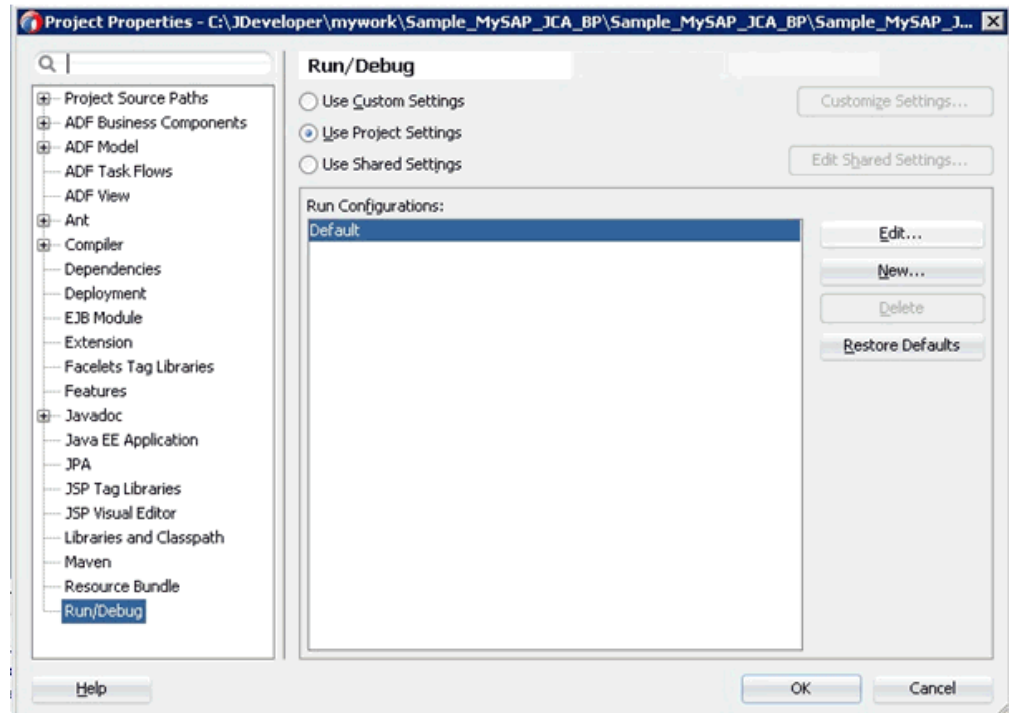
3. Set the Debugging environment.
  - a. Click the down arrow next to the Debug icon and select **Manage Run Configurations** from the context menu, as shown in [Figure 9–18](#).

**Figure 9–18 Manage Run Configurations**



- b. Or, right-click the project and select **Project Properties**.  
The Project Properties dialog is displayed, as shown in [Figure 9–19](#).

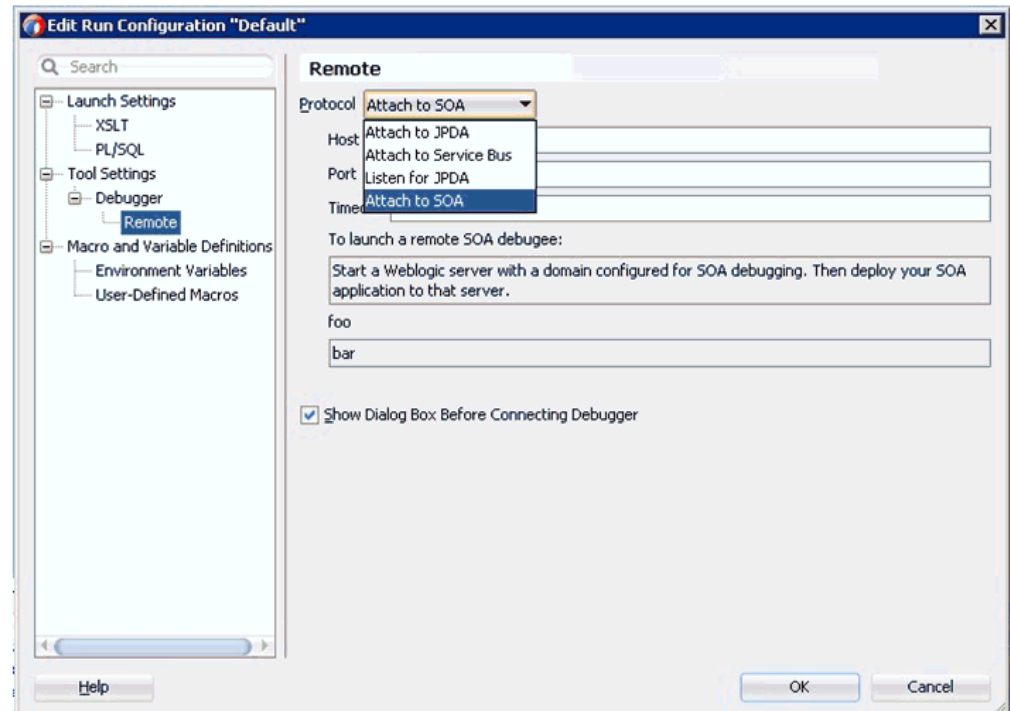
Figure 9–19 Project Properties Dialog



- c. Select **Run/Debug** in the left pane and then click **Edit** in the Run/Debug pane (Run Configurations area).

The Edit Run Configuration dialog is displayed, as shown in [Figure 9–20](#).

Figure 9–20 Edit Run Configuration Dialog



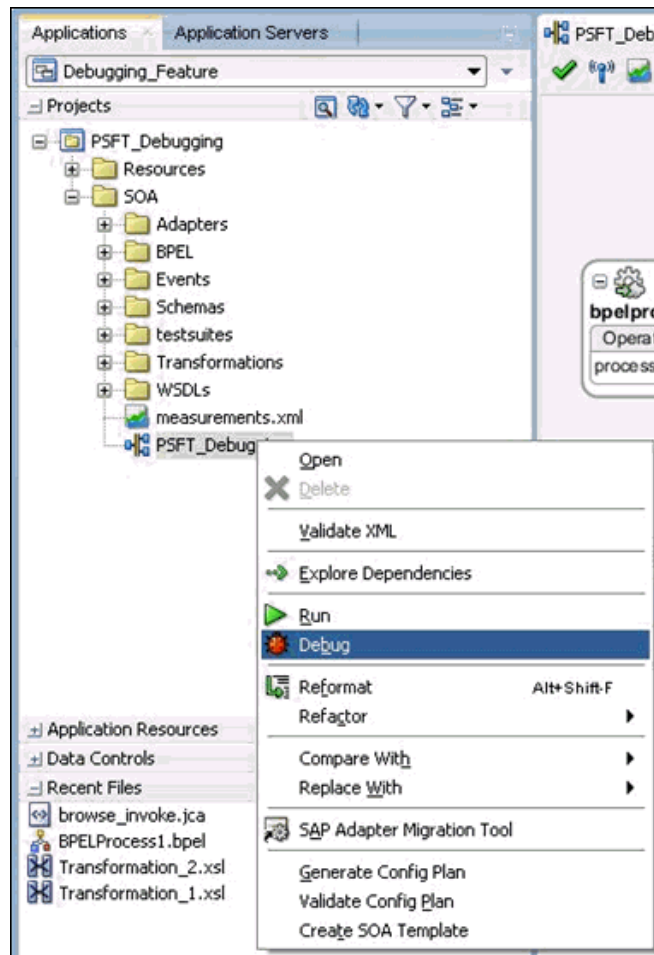
- d. Expand **Tool Settings, Debugger** in the left pane, and then click **Remote**.
                - e. From the Protocol list, select **Attach to SOA**.
                - f. Leave the default values for the Host and Port.
                - g. Click **OK**.
4. Deploy the project.
  - a. Right-click the project and select **Deploy**.

The Deployment Action dialog is displayed.

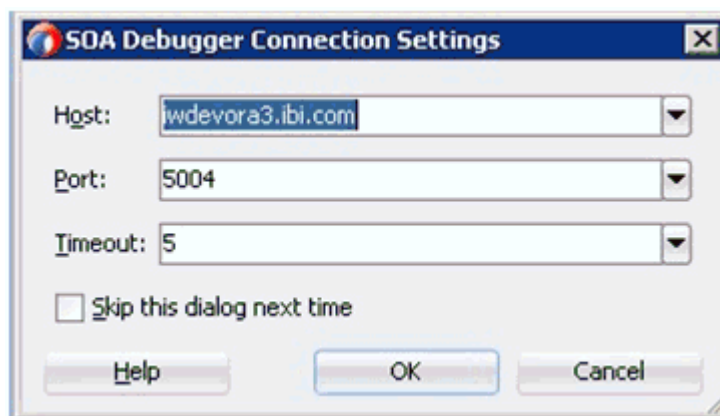
Select the application to deploy and click **Next**.

The Deploy Configuration dialog is displayed.
  - b. Click **Next**.

The Application Servers dialog is displayed.
  - c. Select **IntegratedWebLogicServer** and then click **Finish** to complete the deployment.
  - d. Ensure that the project deployment has completed without any errors or issues before proceeding to the next step.
5. Connect a BPEL process to the SOA Debugger.
  - a. In the Applications tab on the left pane, right-click a composite XML or project for an existing BPEL process and then select **Debug** from the context menu, as shown in [Figure 9-21](#).

**Figure 9–21 Select BPEL Process to Debug**

The SOA Debugger Connection Settings dialog is displayed, as shown in Figure 9–22.

**Figure 9–22 SOA Debugger Connection Settings Dialog**

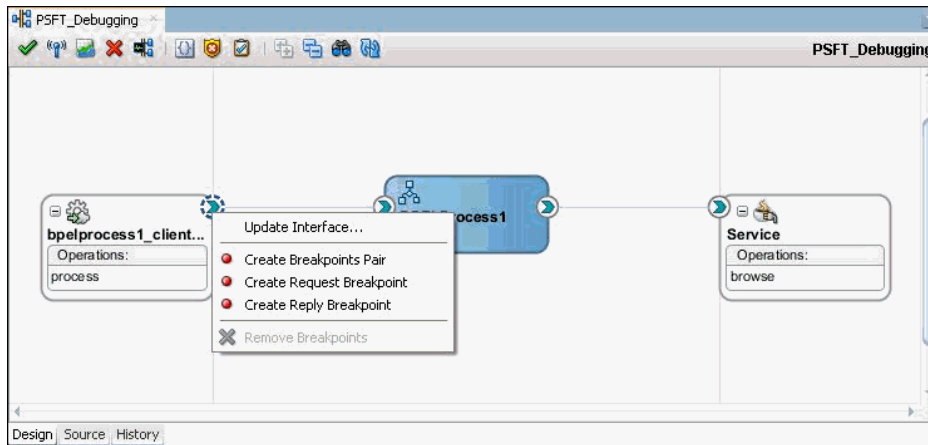
- b. Click OK.

Once the BPEL process is connected to the SOA Debugger, the following messages will be displayed in the Debugging log:

```
Debugger attempting to connect to remote process at iwdevora3.ibi.com 5004.
Debugger connected to remote process at iwdevora3.ibi.com 5004.
Debugger process virtual machine is SOA Debugger
```

6. Set the Breakpoints and initiate debugging.
  - a. Right-click on the components and select the Breakpoint type to set, as shown in [Figure 9–23](#).

**Figure 9–23 Selecting Breakpoints**



**Create Breakpoints Pair** - Set this Breakpoint type for a request-reply (outbound-inbound) interaction. This is useful for scenarios in which both the request and reply are important.

**Create Request Breakpoint** - Set this Breakpoint type for a request (outbound) interaction. This is useful for scenarios in which only the request is important.

**Create Reply Breakpoint** - Set this Breakpoint type for a reply (inbound) interaction. This is useful for scenarios in which only the reply is important.

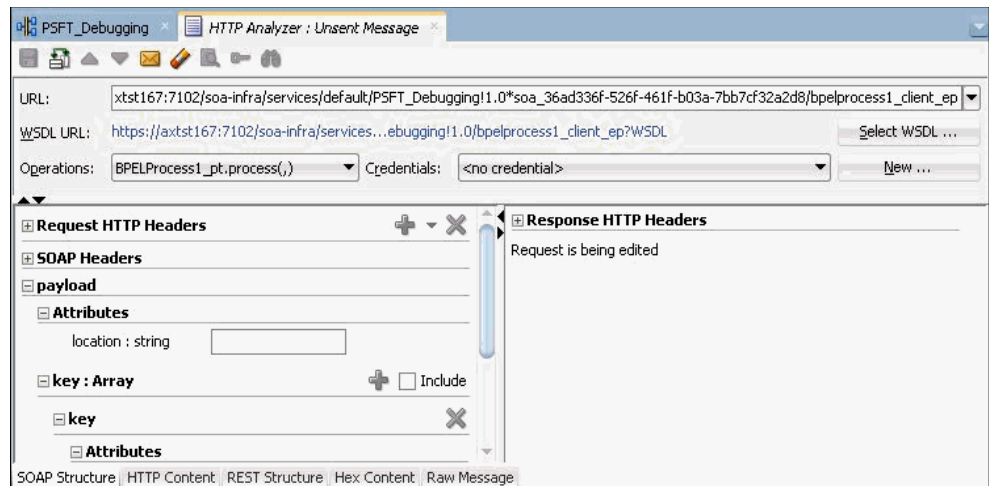
**Initiate WS Debugging** - Set this Breakpoint type to initiate a debugging session. For example, the debugging session encompasses an initiating SOAP request from a web service to a BPEL process to an adapter reference binding component.

- b. Once the Breakpoints are set, right-click the right handle and select **Initiate WS Debugging**, as shown in [Figure 9–24](#).

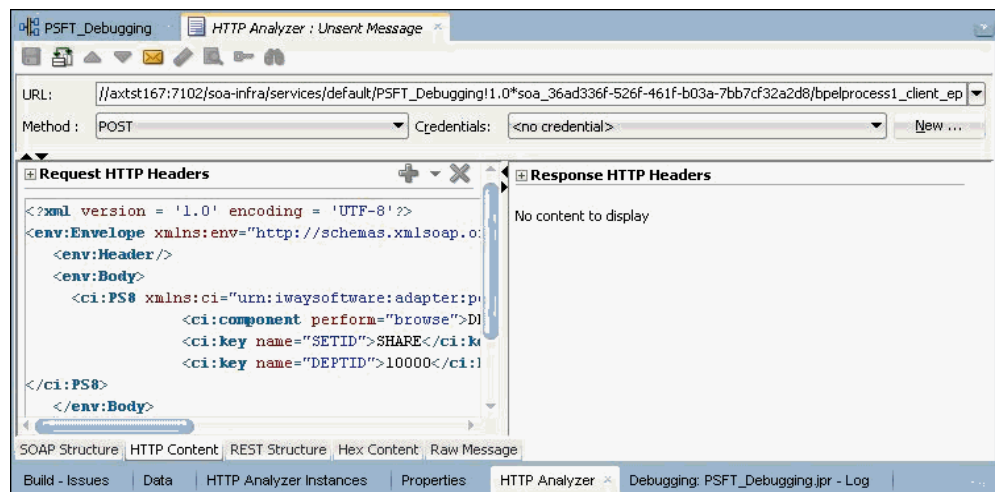
**Figure 9–24 Initiate WS Debugging**



The HTTP Analyzer dialog is displayed, as shown in [Figure 9–25](#).

**Figure 9–25 HTTP Analyzer Dialog**

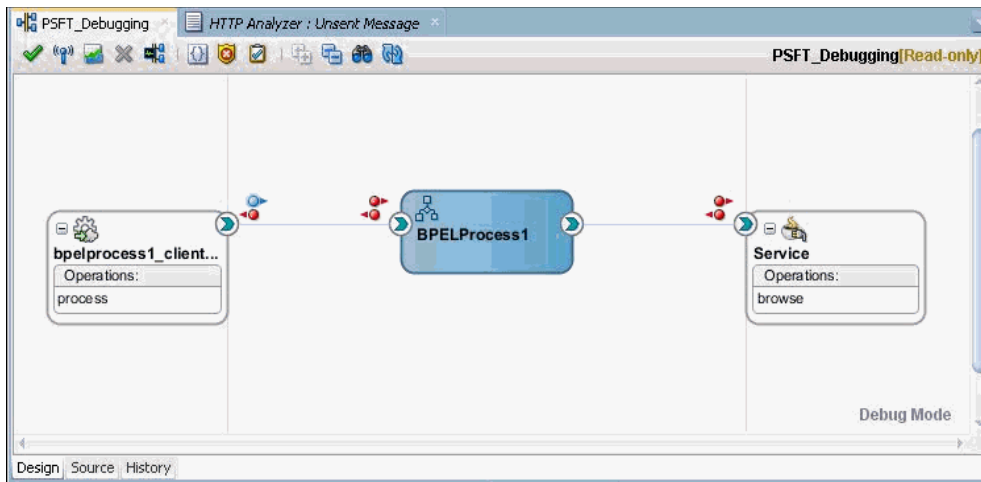
- c. Select HTTP content from the below tab. Now, copy and paste the payload into the body, as shown in [Figure 9–26](#).

**Figure 9–26 Select HTTP Content**

- d. Click **Send Request**.

The BPEL Process stops at the designated Breakpoint and blinks in a blue color, as shown in [Figure 9–27](#).






**Figure 9–27 BPEL Process Stopped at Breakpoint**



- e. Use the available Step options to step through the Debugging process, as shown in Figure 9–28.

**Figure 9–28 Step Options**



Icon	Description
	Ends or detaches from a debugging session.
	Steps over a frame. This places you at the next Breakpoint (for example, the receive activity in the BPEL process on which a Breakpoint was set. If there are no Breakpoints, it steps over all the frames and returns to the first frame. You can also press F8 to step over a frame.
	Steps into the next valid location. This can be a new frame or the same frame, but in a different location. You can also press F7 to step into a frame.
	Steps out of a frame. This option is only used to process a BPEL scope or sequence activity. After completion of scope processing, it pauses at the next scope or activity in the process. You can also press Shift-F7.
	Resumes a step operation. You can also press F9 to resume.

7. View the Request payload and header information.
  - a. Click **Windows**, select **Debugger**, and then **Data**.  
To view sample header information when a Breakpoint stops at the Request Breakpoint for Oracle Application Adapter for PeopleSoft, see Figure 9–29.



**Figure 9–29 Header Information**

Name	Value	Type	Declared Type	Hex Value	Address	ID	Count
(x) completionPe...	null						
(x) ecid	1b072b21-...						
(x) masterConve...	null						
(x) MESH_METRICS	null						
(x) nativePayload							
(x) replyToPort...	null						
(x) transaction	Name=[EJ...						
(x) transport.ht...	192.168.1...						
(x) XID	BEA1-079...						

A sample response payload for Oracle Application Adapter for PeopleSoft is shown in [Figure 9–30](#).

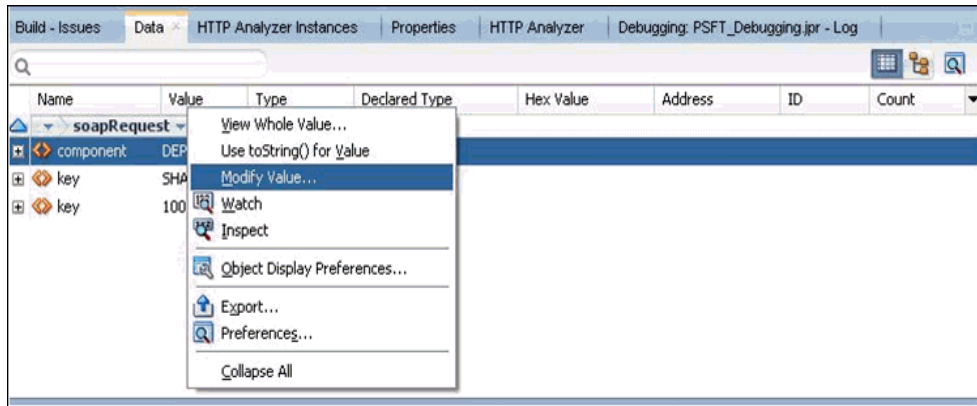
**Figure 9–30 Sample Response Payload**

Name	Value	Type	Declared Type	Hex Value	Address	ID	Count
(x) jca.iwayOut...	PeopleSoft...						
(x) jca.iwayOut...	PeopleSoft...						
(x) jca.iwayOut...	PeopleSoft...						
(x) jca.iwayOut...	PeopleSoft...						
(x) jca.iwayOut...	{}						
(x) jca.iwayOut...	{HP_STATS...						
(x) jca.iwayOut...	{HP_STATS...						
(x) jca.iwayOut...	{HP_STATS...						
(x) jca.iwayOut...	{ADMIN_R...						

**Note:** The payload display is limited to the screen size as shown in [Figure 9–30](#). However, the Breakpoint at the BPEL process displays the complete payload, allows scrolling and viewing all elements of the payload.

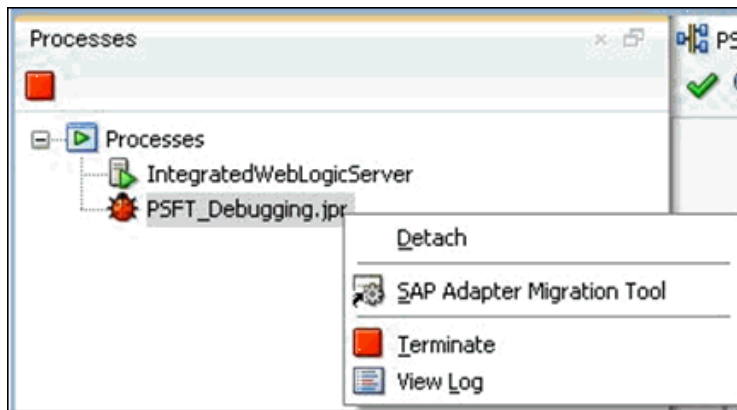
8. Modify the Request payload content.
  - a. Expand the SOAP request, select the field to modify, right-click and select **Modify Value** from the context menu, as shown in [Figure 9–31](#).

**Figure 9–31 Modify Value**



9. End or detach the Debugging session.
  - a. Click **Window** and then **Processes**. Right click on the process in the Processes tab and select **Detach** or **Terminate**, as shown in [Figure 9–32](#).

**Figure 9–32 Detach Debugging Session**



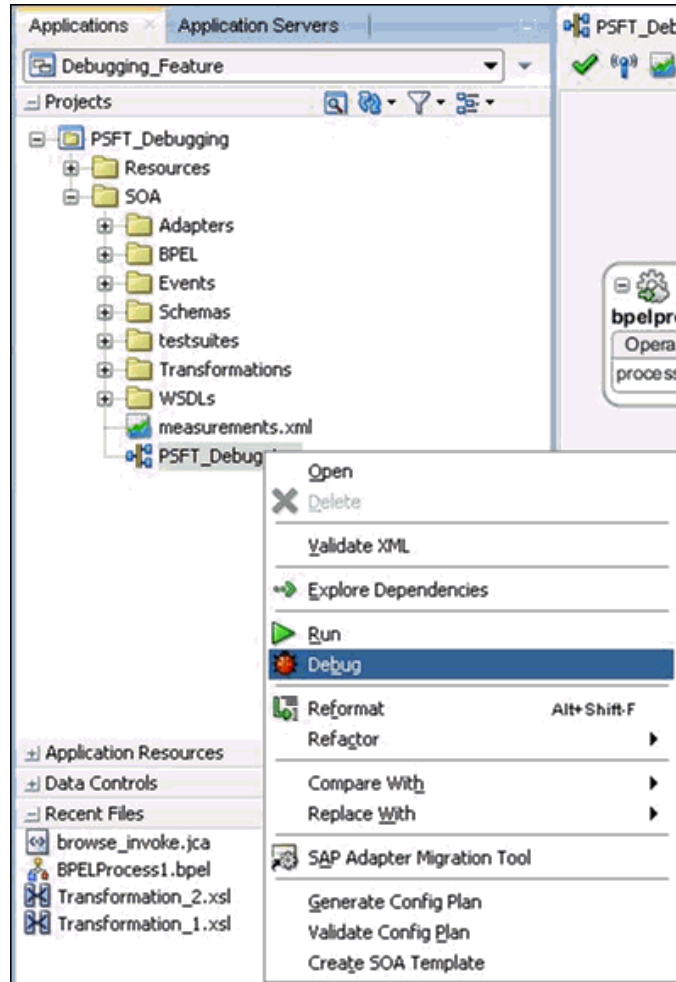
- b. Select one of the following options:
  - Detach** - Removes the SOA Debugger without ending the debugging process.
  - Terminate** - Ends the debugging process.

### 9.3.3.2 Debugging an Inbound BPEL Process in Oracle JDeveloper

1. Deploy the Inbound BPEL process.
  - a. Right-click the project and select **Deploy**.  
The Deployment Action dialog is displayed.
  - b. Select **Deploy to Application Server** and click **Next**.  
The Deploy Configuration dialog is displayed.
  - c. Click **Next**.  
The Select Server dialog is displayed.
  - d. Select the server to deploy and click **Finish**.

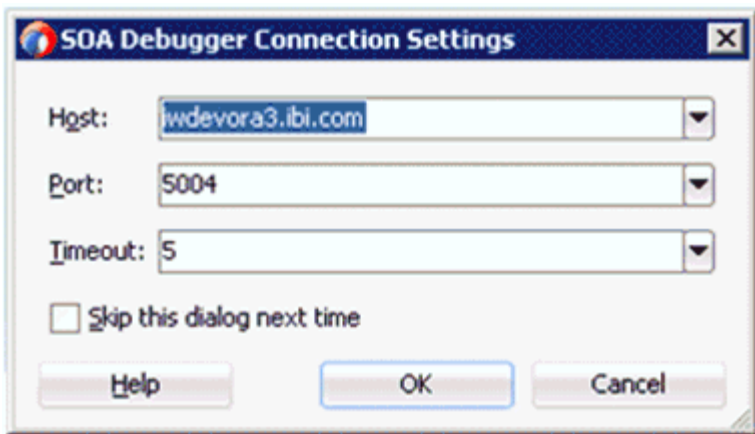
- e. Ensure that the project deployment has completed without any errors or issues before proceeding to the next step.
2. Connect a BPEL process to the SOA Debugger.
    - a. Right-click an inbound composite and select **Debug**, as shown in [Figure 9-33](#).

**Figure 9-33 Select BPEL Process to Debug**



The SOA Debugger Connection Settings dialog is displayed, as shown in [Figure 9-34](#).

**Figure 9–34 SOA Debugger Connection Settings Dialog**



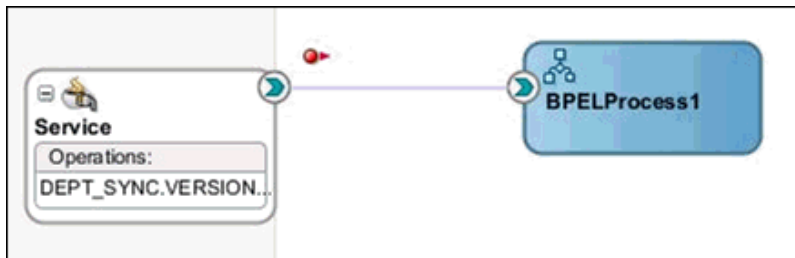
- b. Click **OK**.

Once the BPEL process is connected to the SOA Debugger, the following messages will be displayed in the Debugging log:

```
Debugger attempting to connect to remote process at iwdevora3.ibi.com 5004.
Debugger connected to remote process at iwdevora3.ibi.com 5004.
Debugger process virtual machine is SOA Debugger
```

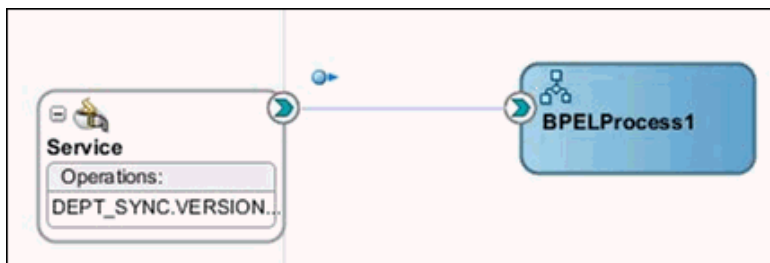
- 3. Set the Breakpoints, as shown in [Figure 9–35](#).

**Figure 9–35 Setting Breakpoints**



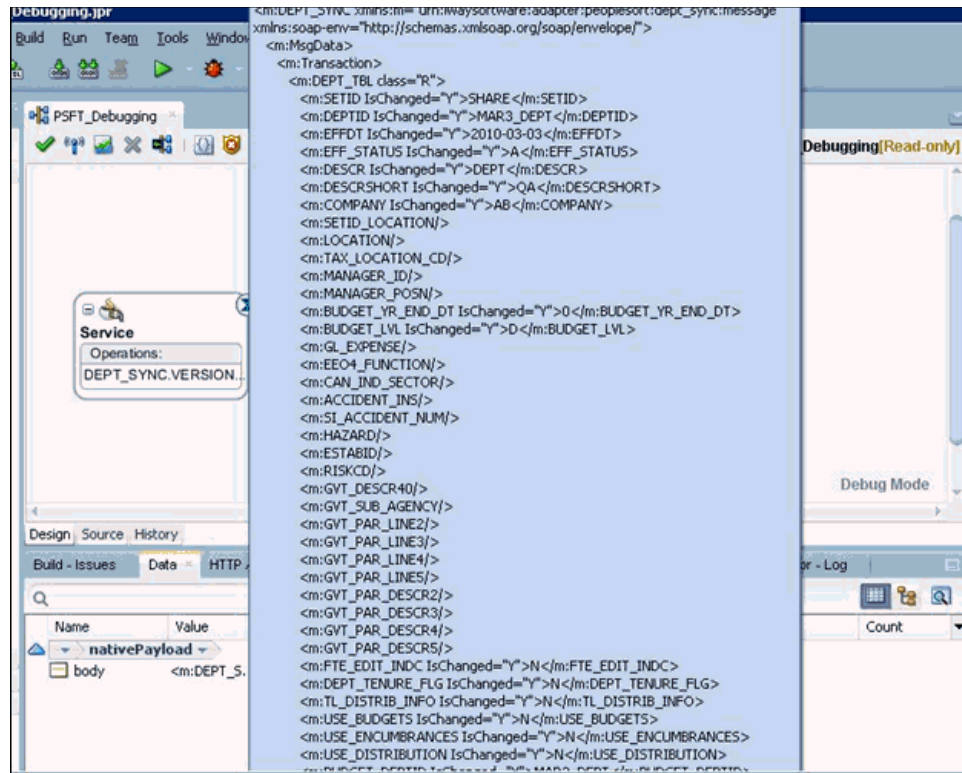
- 4. Once the Breakpoints are set and a message is received through the inbound process (for example, by triggering from PeopleSoft), the process stops at the designated Breakpoints, as shown in [Figure 9–36](#).

**Figure 9–36 Stopping the Process at the Breakpoint**



- 5. View the Request payload and header information.
  - a. Click **Windows**, select **Debugger**, and then **Data**, as shown in [Figure 9–37](#).

Figure 9–37 Debugger Data



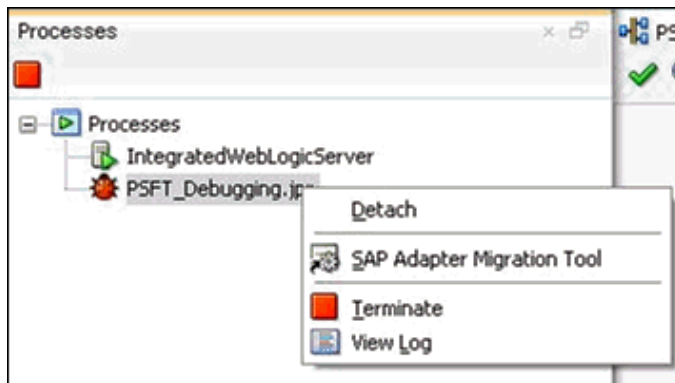
- b. Header information displayed for Inbound Request Breakpoint (DEPT\_Sync) is shown in Figure 9–38.

Figure 9–38 Inbound Request Breakpoint (DEPT\_Sync)

Name	Value	Type	Declared Type	Hex Value	Address	ID	Count
(x) jca.iiwayInb...	POST						
(x) jca.iiwayInb...	False						
(x) jca.iiwayInb...	EXTERNAL						
(x) jca.iiwayInb...	/psft						
(x) jca.iiwayInb...	Java1.5.0_04						
(x) jca.iiwayInb...	0						
(x) nativePayload							
(x) subject	weblogic, A...						
(x) XID	<none>						

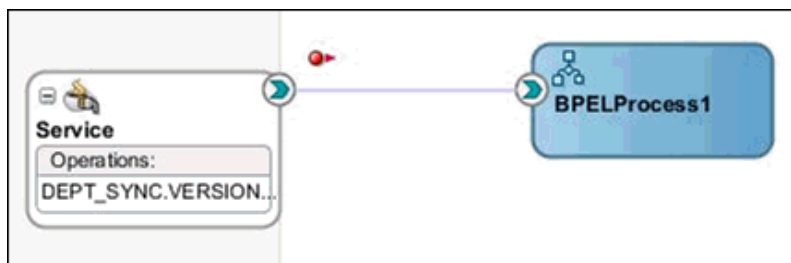
6. End or detach the Debugging session.
  - a. Click **Window** and then **Processes**. Right click on the process in the Processes tab and select **Detach** or **Terminate**, as shown in Figure 9–39.

**Figure 9–39 Detach Debugging Session**



- b. Select one of the following options:
  - Detach** - Removes the SOA Debugger without ending the debugging process.
  - Terminate** - Ends the debugging process.
 The process will be detached and is displayed, as shown in [Figure 9–40](#).

**Figure 9–40 BPEL Process Detached**



### 9.3.4 Debugging an OSB Process in Oracle JDeveloper

This section describes how to debug an OSB process in Oracle JDeveloper. It contains the following topics:

- [Section 9.3.4.1, "Prerequisites"](#)
- [Section 9.3.4.2, "Debugging an Outbound OSB Process in Oracle JDeveloper"](#)
- [Section 9.3.4.3, "Debugging an Inbound OSB Process in Oracle JDeveloper"](#)

#### 9.3.4.1 Prerequisites

Ensure that the *IntegratedWebLogicServer* domain and an OSB process with file output are already created in Oracle JDeveloper.

---

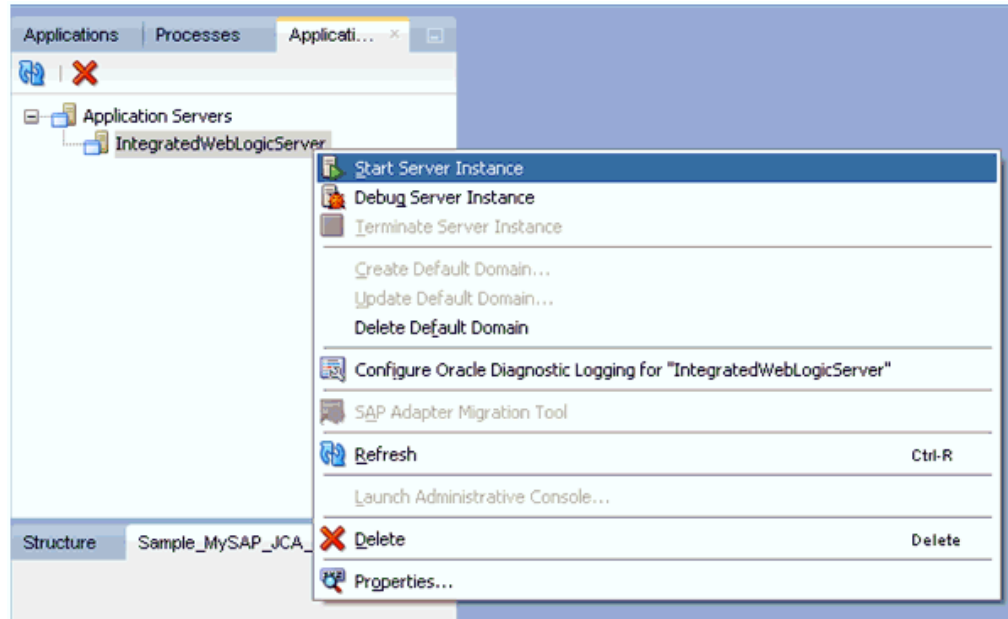
**Note:** Ensure that the application name and the project name are the same.

---

1. Open Oracle JDeveloper and start *IntegratedWeblogicServer* or *startWebLogic.cmd*.
  - a. Click the **Application Servers** tab in the left pane.

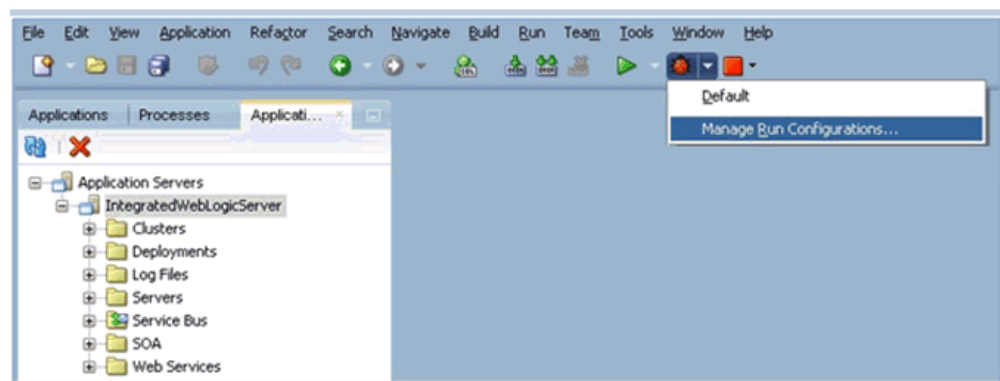
- b. Under the Application Servers node, right-click **IntegratedWebLogicServer** and select **Start Server Instance** from the context menu, as shown in [Figure 9–41](#).

**Figure 9–41 Start Server Instance**



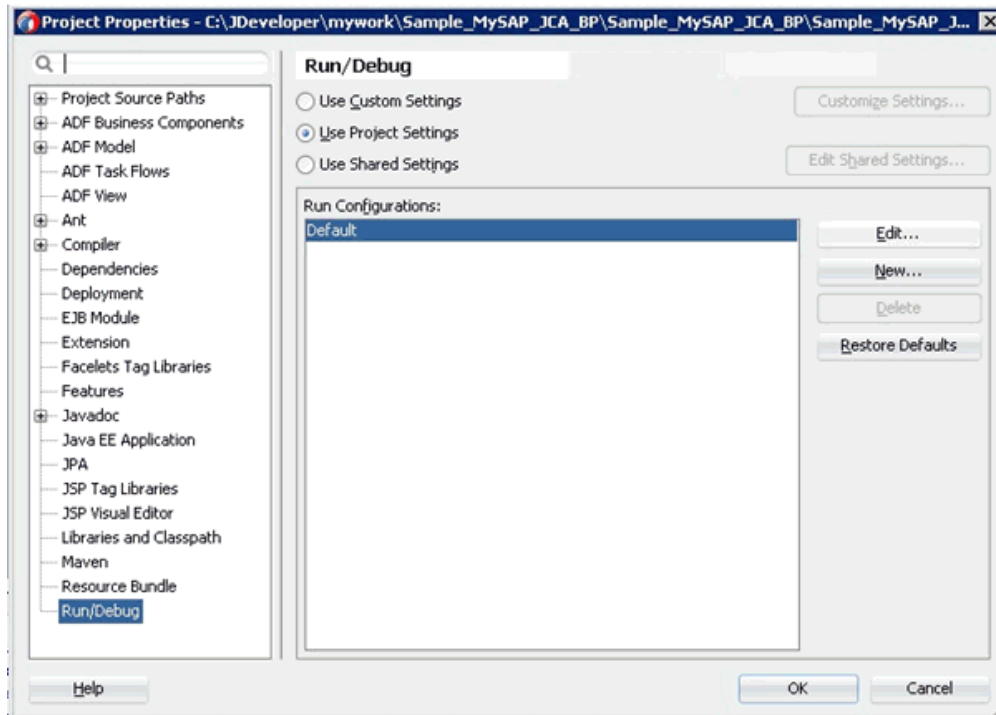
- c. Or, start Oracle WebLogic server from the command prompt using `startWebLogic.cmd`.
2. Set the Debugging environment.
    - a. Click the down arrow next to the Debug icon and select **Manage Run Configurations** from the context menu, as shown in [Figure 9–42](#).

**Figure 9–42 Manage Run Configurations**



- b. Or, right-click the project and select **Project Properties**.  
The Project Properties dialog is displayed, as shown in [Figure 9–43](#).

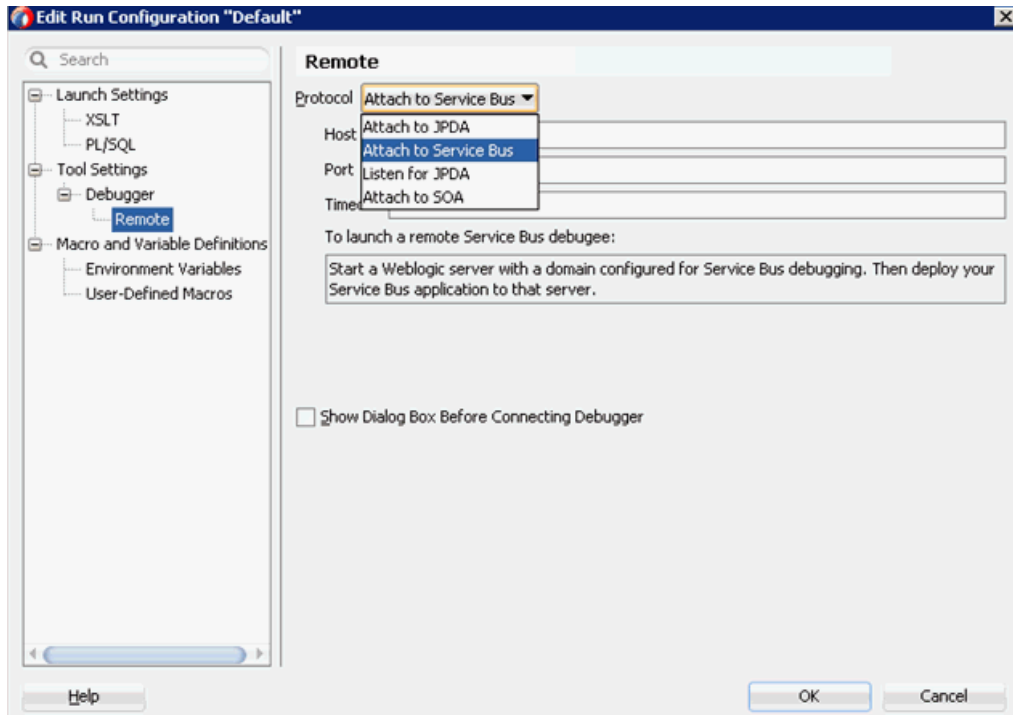
**Figure 9–43 Project Properties Dialog**



- c. Select **Run/Debug** in the left pane and then click **Edit** in the Run/Debug pane (Run Configurations area).

The Edit Run Configuration dialog is displayed, as shown in [Figure 9–44](#).

**Figure 9–44 Edit Run Configuration Dialog**



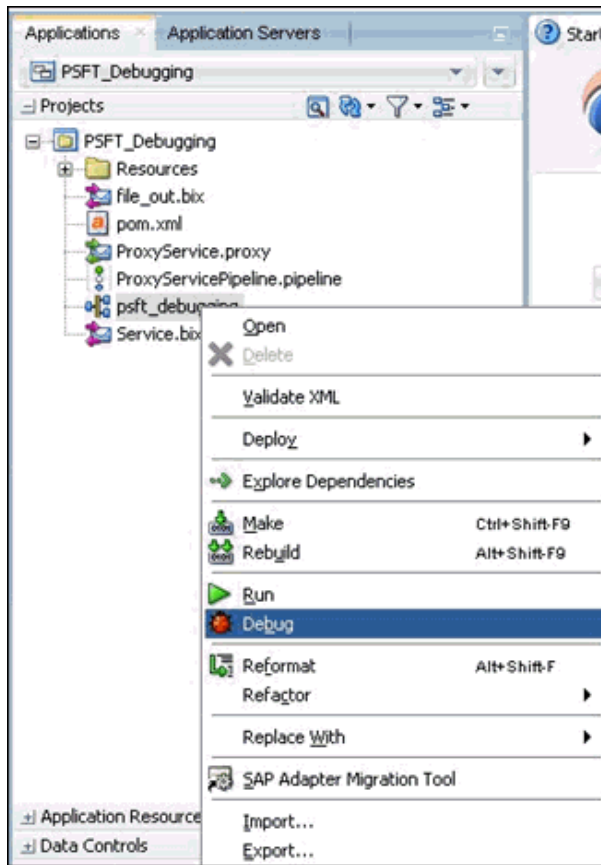


- d. Expand **Tool Settings, Debugger** in the left pane, and then click **Remote**.
- e. From the Protocol list, select **Attach to Service Bus**.
- f. Leave the default values for the Host and Port.
- g. Click **OK**.

### 9.3.4.2 Debugging an Outbound OSB Process in Oracle JDeveloper

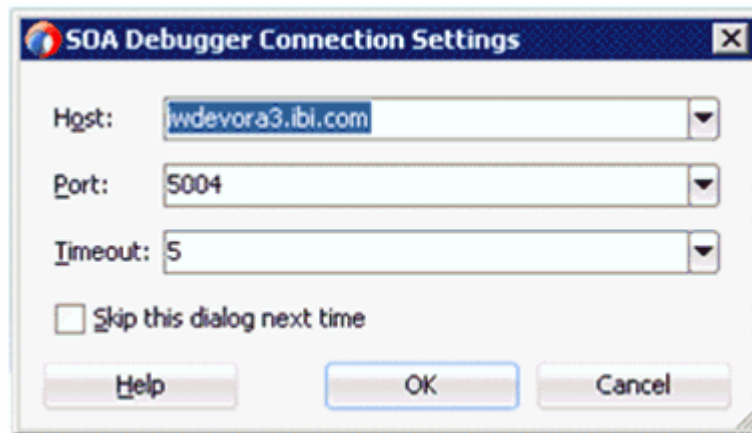
1. Deploy the project.
  - a. Right-click the project and select **Deploy**.  
The Deployment Action dialog is displayed.
  - b. Select the application to deploy and click **Next**.  
The Deploy Configuration dialog is displayed.
  - c. Click **Next**.  
The Application Servers dialog is displayed.
  - d. Select **IntegratedWebLogicServer** and then click **Finish** to complete the deployment.
  - e. Ensure that the project deployment has completed without any errors or issues before proceeding to the next step.
2. Connect an OSB process to the SOA Debugger.
  - a. In the Applications tab on the left pane, right-click a composite XML or project for an existing OSB process and then select **Debug** from the context menu, as shown in [Figure 9-45](#).

**Figure 9–45 Select OSB Process to Debug**



The SOA Debugger Connection Settings dialog is displayed, as shown in Figure 9–46.

**Figure 9–46 SOA Debugger Connection Settings Dialog**



- b. Click **OK**.

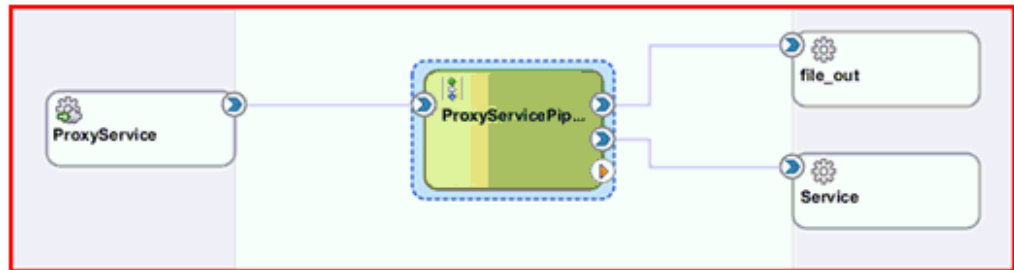
Once the OSB process is connected to the SOA Debugger, the following messages will be displayed in the Debugging log:

```
Debugger attempting to connect to remote process at iwdevora3.ibi.com 5004.
Debugger connected to remote process at iwdevora3.ibi.com 5004.
```

Debugger process virtual machine is SOA Debugger

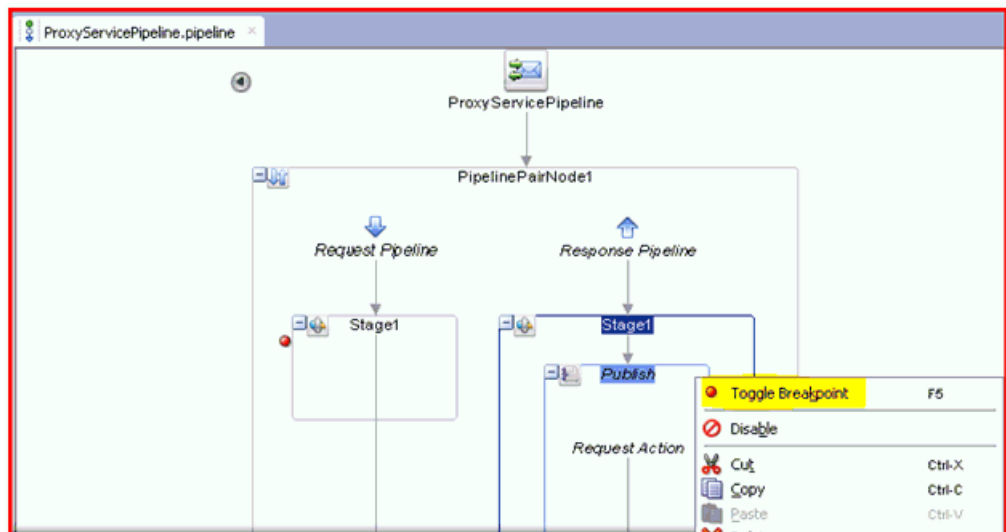
3. Set the Breakpoints.
  - a. Open the pipeline in its editor by double-clicking on the pipeline, as shown in Figure 9-47.

Figure 9-47 ProxyServicePipeline



- b. Expand the actions until you reach the node where the Breakpoint must be added. In this case, expand **PipelinePairNode1**, right-click the **Stage1** node (under **Request Pipeline**), and select **Toggle Breakpoint** from the context menu, as shown in Figure 9-48.

Figure 9-48 Toggle Breakpoint



Repeat this step for the Publish node (under **Response Pipeline**).

A red icon appears next to the node to indicate that a Breakpoint has been set.

---

**Note:** To disable a Breakpoint, right-click the node and select **Disable Breakpoint**.

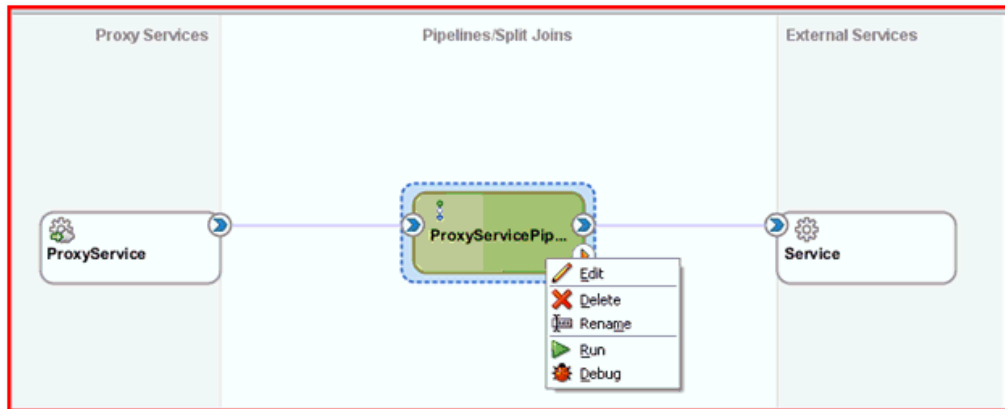
To remove a Breakpoint, right-click the node and select **Toggle Breakpoint** again.

---

4. Initiate Debugging.

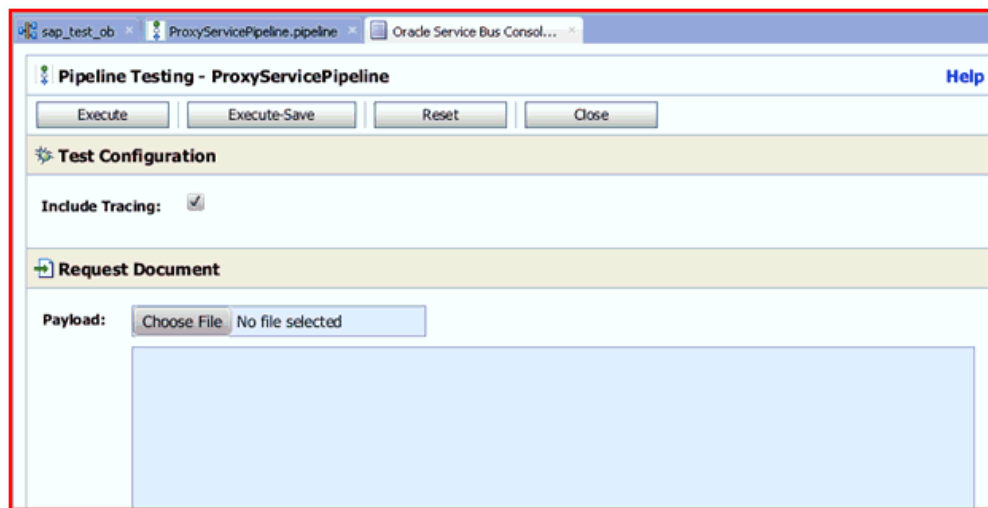
- a. Right-click the pipeline in the Application Navigator, and select **Debug**, as shown in [Figure 9-49](#).

**Figure 9-49 Select Debug**



The process is deployed to the integrated server and the Test Configuration pane will be displayed, as shown in [Figure 9-50](#).

**Figure 9-50 Test Configuration Pane**

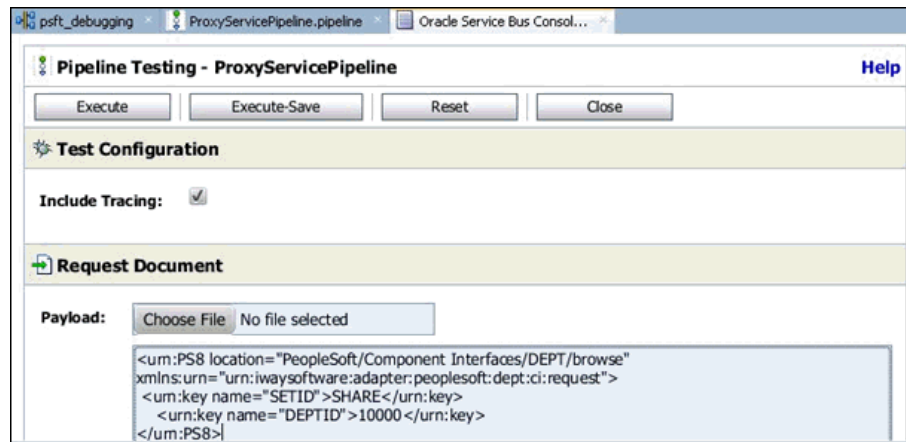



---

**Note:** If there is no domain currently running, then the Create Default Domain dialog will be displayed. Enter the connection information for the integrated server and then click **OK**. This process may take several minutes.

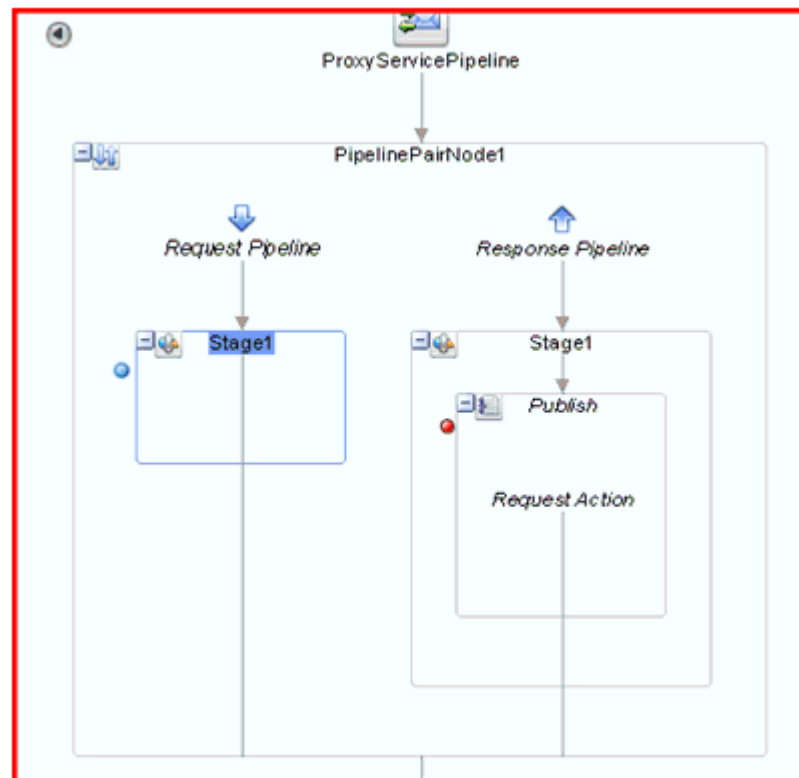
---

- b. In the Test Configuration pane, enter the test data in the Request Document area, and configure any additional input as required, as shown in [Figure 9-51](#).

**Figure 9–51 Test Request Document**

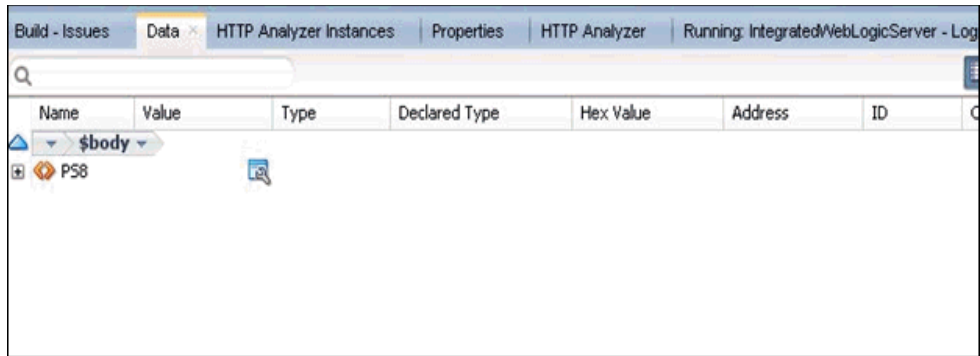
- c. Click **Execute**.

The Test Configuration pane executes the command, but the OSB process stops at the designated Breakpoint and blinks in a blue color, as shown in Figure 9–52.

**Figure 9–52 OSB Process Stopped at Breakpoint**

- d. Check the Data tab at the bottom to verify that the input has passed, as shown in Figure 9–53.

**Figure 9–53 Data Tab**



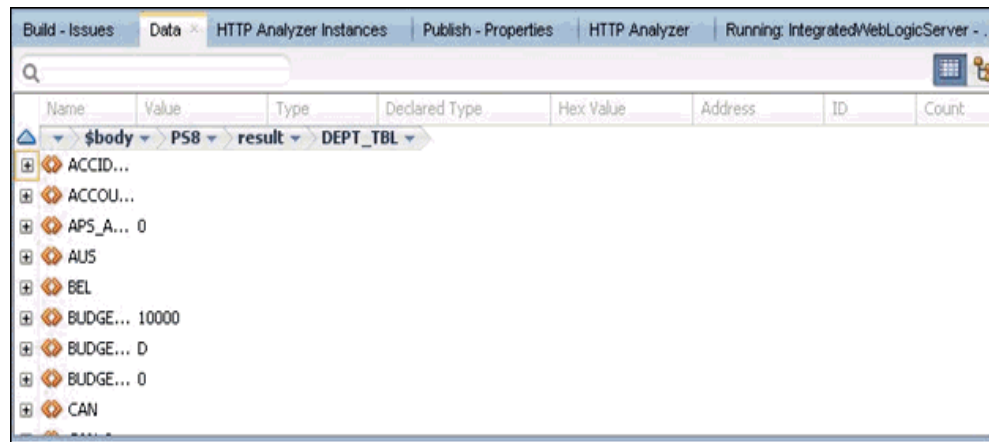
- e. Use the available Step options to step through the Debugging process, as shown in [Figure 9–54](#).

**Figure 9–54 Step Options**

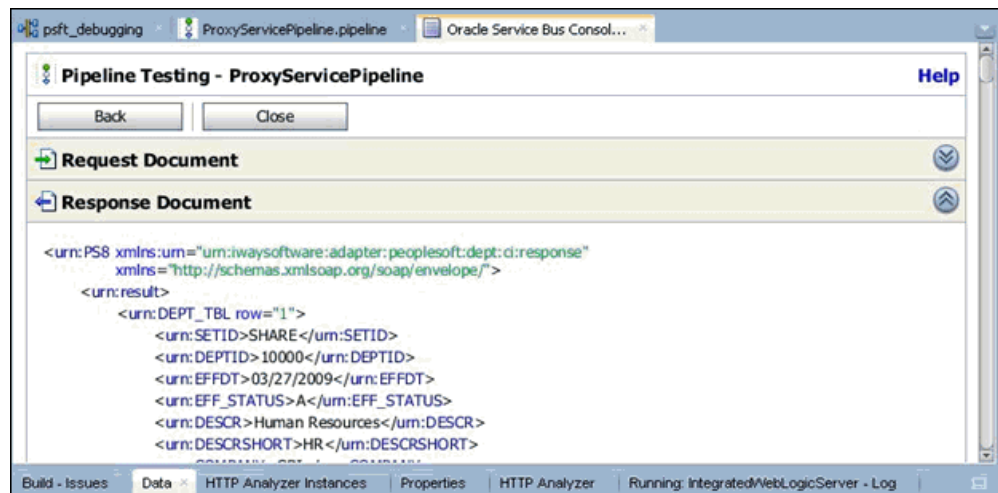


Icon	Description
	Ends or detaches from a debugging session.
	Steps over a frame. This places you at the next Breakpoint (for example, the receive activity in the OSB process on which a Breakpoint was set. If there are no Breakpoints, it steps over all the frames and returns to the first frame. You can also press F8 to step over a frame.
	Steps into the next valid location. This can be a new frame or the same frame, but in a different location. You can also press F7 to step into a frame.
	Steps out of a frame. This option is only used to process an OSB scope or sequence activity. After completion of scope processing, it pauses at the next scope or activity in the process. You can also press <b>Shift-F7</b> .
	Resumes a step operation. You can also press F9 to resume.

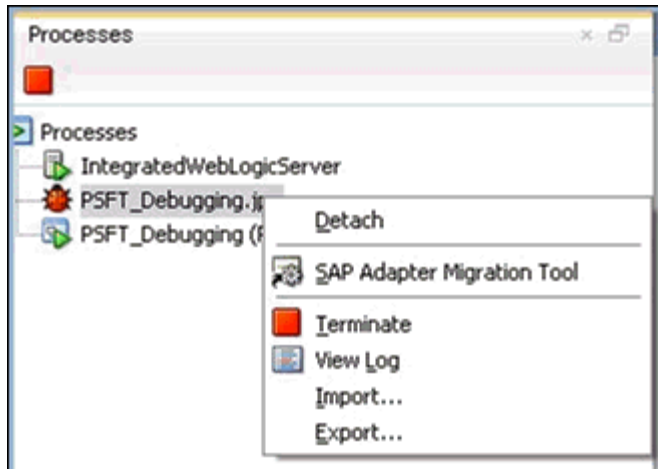
- f. Use **Step Over** to go to the next Breakpoint (Response Pipeline in this example).  
You will be able to see the output in the Data tab, as shown in [Figure 9–55](#).

**Figure 9–55 Data Tab Output**

You will also be able to see the response in the Oracle Service Bus Console, as shown in [Figure 9–56](#).

**Figure 9–56 Oracle Service Bus Console**

5. End or detach the Debugging session.
  - a. Click **Window** and then **Processes**. Right click on the process in the Processes tab and select **Detach** or **Terminate**, as shown in [Figure 9–57](#).

**Figure 9–57 Detach Debugging Session**

- b. Select one of the following options:
  - Detach** - Removes the SOA debugger without ending the debugging process.
  - Terminate** - Ends the debugging process.

### 9.3.4.3 Debugging an Inbound OSB Process in Oracle JDeveloper

1. Create an inbound OSB process.
 

Ensure that the application name and the project name are the same.
2. Deploy the project.
  - a. Right-click the project and select **Deploy**.
 

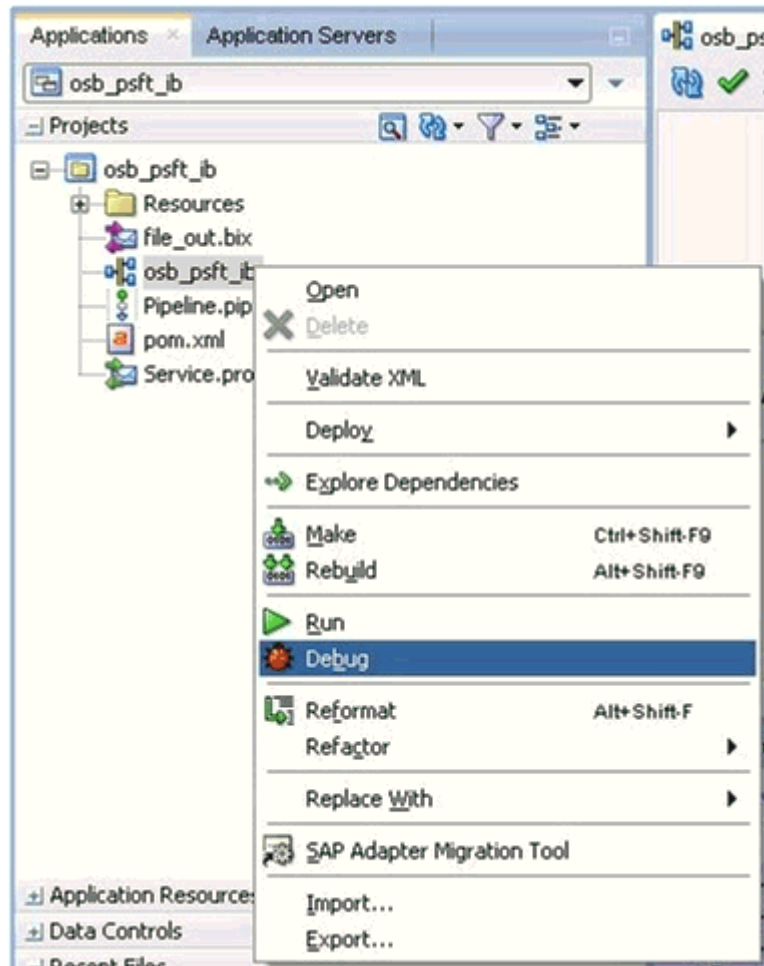
The Deployment Action dialog is displayed.
  - b. Select the application to deploy and click **Next**.
 

The Deploy Configuration dialog is displayed.
  - c. Click **Next**.
 

The Application Servers dialog is displayed.
  - d. Select **IntegratedWebLogicServer** and then click **Finish** to complete the deployment.
  - e. Ensure that the project deployment has completed without any errors or issues before proceeding to the next step.
3. Connect an OSB process to the SOA Debugger.
  - a. In the Applications tab on the left pane, right-click a composite XML or project for an existing OSB process and then select **Debug** from the context menu, as shown in [Figure 9–58](#).

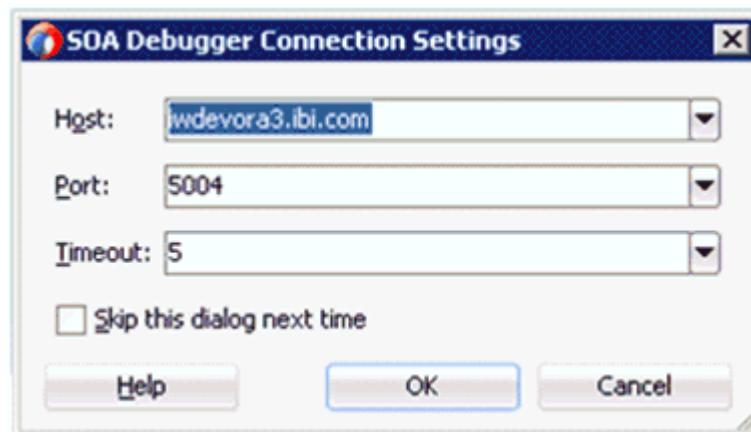


Figure 9–58 Select OSB Process to Debug



The SOA Debugger Connection Settings dialog is displayed, as shown in Figure 9–59.

Figure 9–59 SOA Debugger Connection Settings Dialog



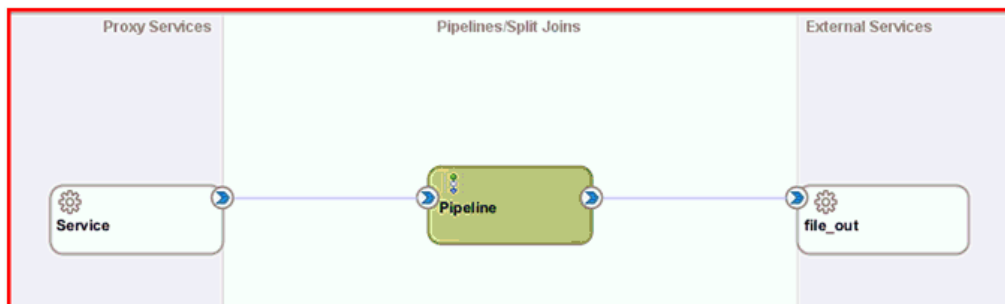
- b. Click OK.

Once the OSB process is connected to the SOA Debugger, the following messages will be displayed in the Debugging log:

```
Debugger attempting to connect to remote process at iwdevora3.ibi.com 5004.  
Debugger connected to remote process at iwdevora3.ibi.com 5004.  
Debugger process virtual machine is SOA Debugger
```

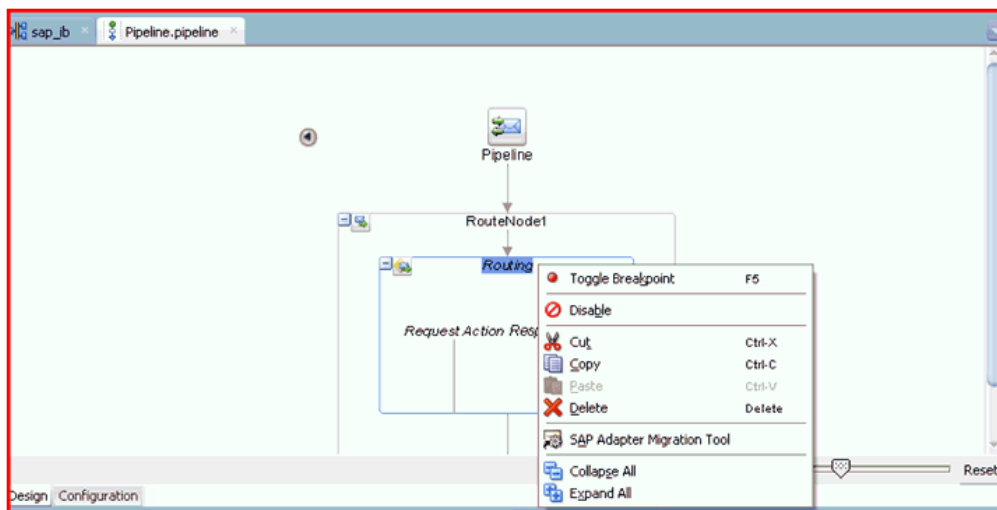
- 4. Set the Breakpoints.
  - a. Open the pipeline in its editor by double-clicking on the pipeline, as shown in Figure 9-60.

Figure 9-60 ProxyServicePipeline



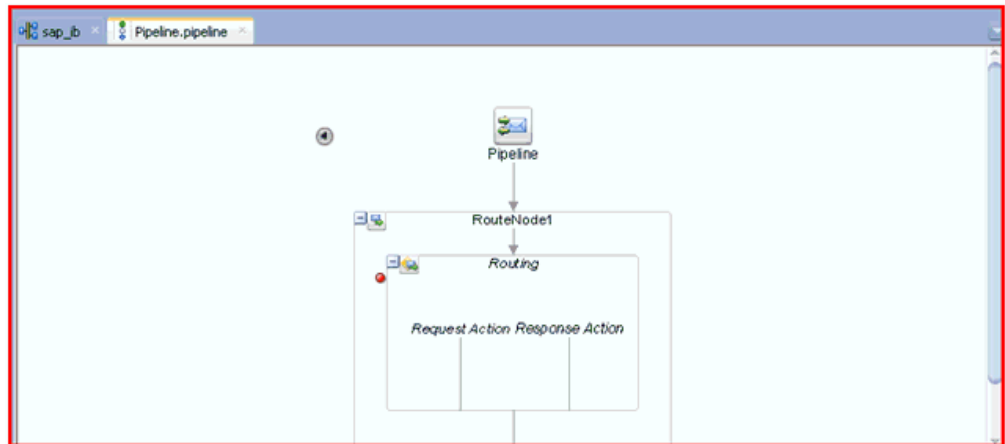
- b. Expand the actions until you reach the node where the Breakpoint must be added. In this case, expand until you reach **Routing**. Right-click the **Routing** node and select **Toggle Breakpoint** from the context menu, as shown in Figure 9-61.

Figure 9-61 Toggle Breakpoint



Repeat the above step for each node to which you want to add a Breakpoint. In this example, a Breakpoint is set only for the **Routing** node.

A red icon appears next to the node to indicate that a Breakpoint has been set, as shown in Figure 9-62.

**Figure 9–62 Red Icon for Set Breakpoint**

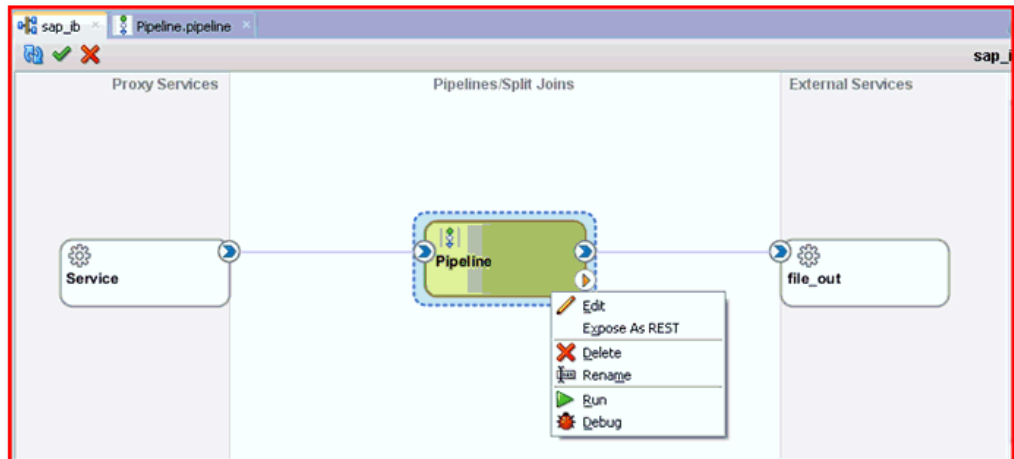

---

**Note:** To disable a Breakpoint, right-click the node and select **Disable Breakpoint**.

To remove a Breakpoint, right-click the node and select **Toggle Breakpoint** again.

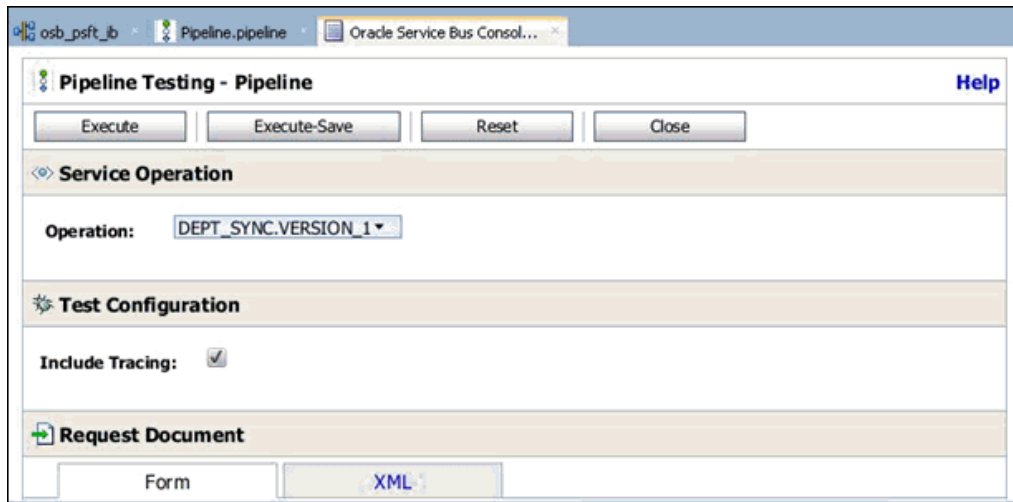
---

5. Initiate Debugging.
  - a. Right-click the pipeline in the Application Navigator, and select **Debug**, as shown in [Figure 9–63](#).

**Figure 9–63 Select Debug**

The process is deployed to the integrated server and the Test Configuration pane will be displayed, as shown in [Figure 9–64](#).

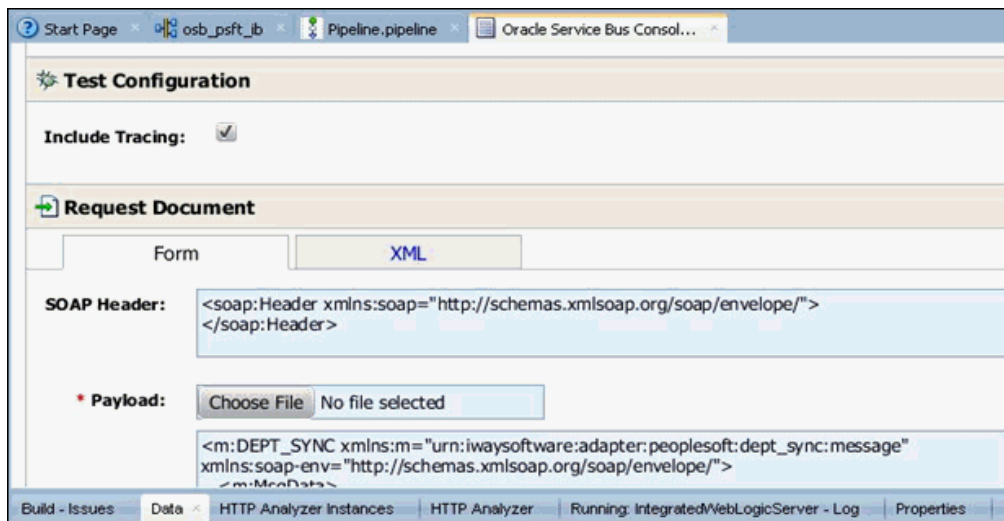
**Figure 9–64 Test Configuration Pane**



**Note:** If there is no domain currently running, then the Create Default Domain dialog will be displayed. Enter the connection information for the integrated server and then click **OK**. This process may take several minutes.

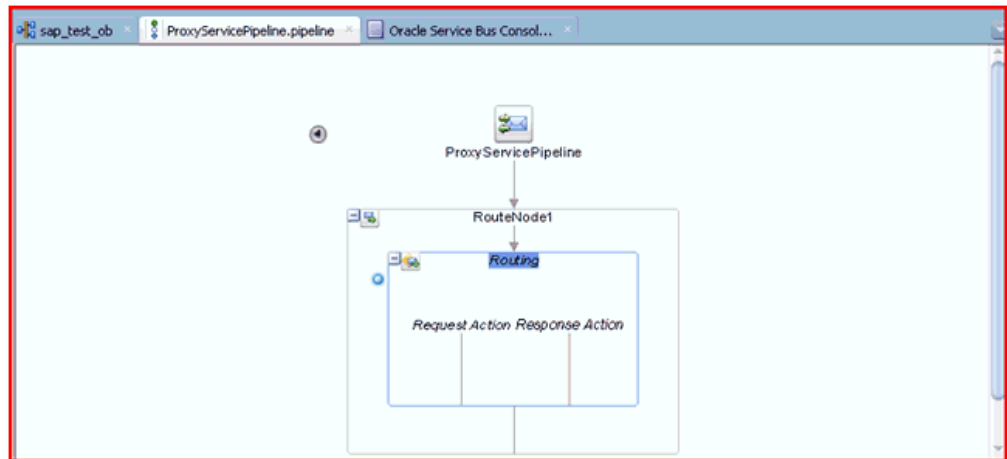
- b. In the Test Configuration pane, enter the test data in the Request Document area, and configure any additional input as required, as shown in [Figure 9–65](#).

**Figure 9–65 Test Request Document**



- c. Click **Execute**.  
The Test Configuration pane executes the command, but the OSB process stops at the designated Breakpoint and blinks in a blue color, as shown in [Figure 9–66](#).






**Figure 9–66 OSB Process Stopped at Breakpoint**



- d. Use the available Step options to step through the Debugging process, as shown in Figure 9–67.

**Figure 9–67 Step Options**



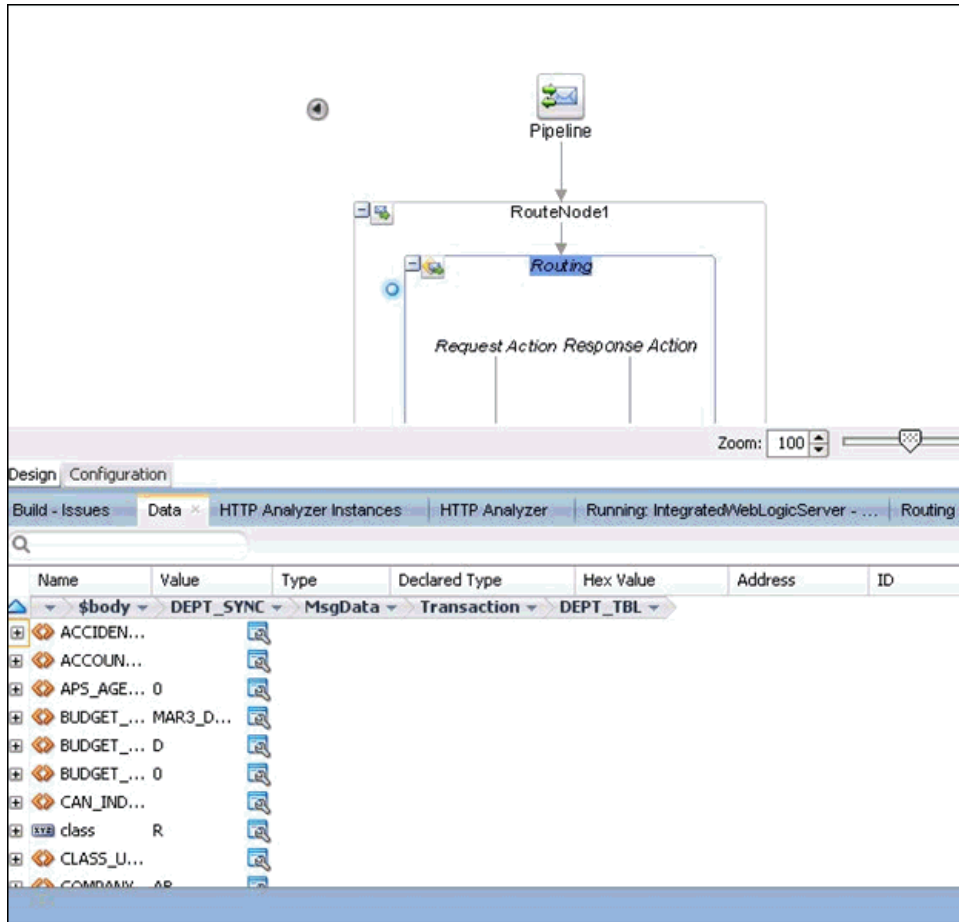
Icon	Description
	Ends or detaches from a debugging session.
	Steps over a frame. This places you at the next Breakpoint (for example, the receive activity in the OSB process on which a Breakpoint was set. If there are no Breakpoints, it steps over all the frames and returns to the first frame. You can also press F8 to step over a frame.
	Steps into the next valid location. This can be a new frame or the same frame, but in a different location. You can also press F7 to step into a frame.
	Steps out of a frame. This option is only used to process an OSB scope or sequence activity. After completion of scope processing, it pauses at the next scope or activity in the process. You can also press Shift-F7.
	Resumes a step operation. You can also press F9 to resume.

- e. Use **Step Over** to complete the process execution

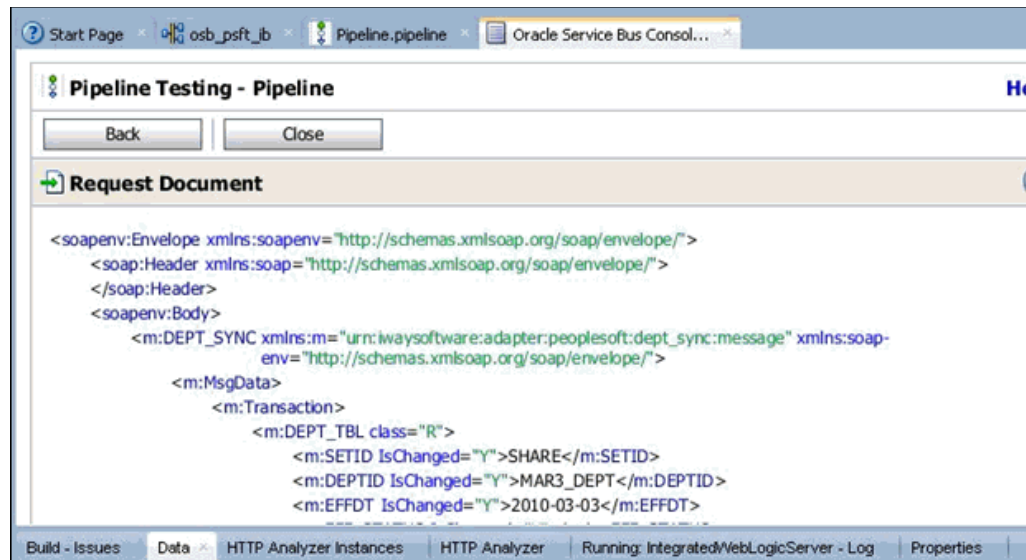
**Note:** Since there is only one Breakpoint in this example, using **Step Over** completes the process execution.

You will be able to see the response document displayed in the Data tab, as shown in [Figure 9-68](#).

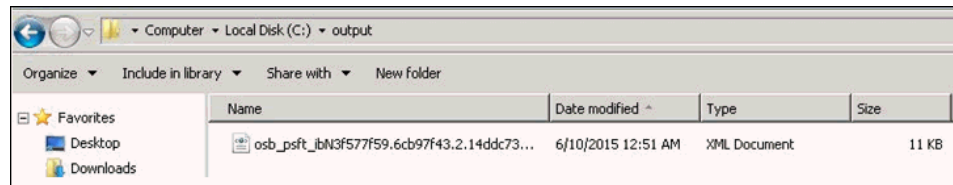
**Figure 9-68 Data Tab Output**



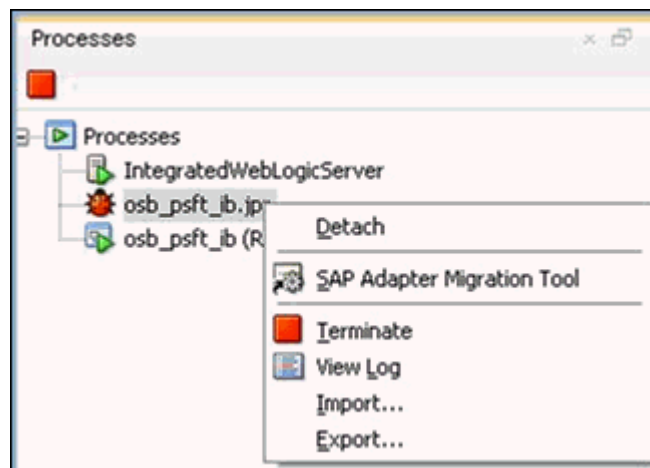
You will also be able to see the response in the Oracle Service Bus Console, as shown in [Figure 9-69](#).

**Figure 9–69 Oracle Service Bus Console**

The output will also be available in the configured output location, as show in [Figure 9–70](#).

**Figure 9–70 Configured Output Location**

6. End or detach the Debugging session.
  - a. Click **Window** and then **Processes**. Right click on the process in the Processes tab and select **Detach** or **Terminate**, as shown in [Figure 9–71](#).

**Figure 9–71 Detach Debugging Session**

- b. Select one of the following options:

**Detach** - Removes the SOA debugger without ending the debugging process.

**Terminate** - Ends the debugging process.

## 9.4 Exception Filter

This section describes how to configure exception filter functionality for the Oracle Application Adapter for PeopleSoft and includes a sample testing scenario. This section contains the following topic:

- [Section 9.4.1, "Configuring the Exception Filter"](#)

The exception filter is supported only for outbound processes that use J2CA configurations. This feature is not supported for BSE configurations and inbound processes that use J2CA configurations.

The exception filter uses the `com.ibi.afjca.oracle.AdapterExceptionFilter` class to filter the generated exceptions. This class filters the exceptions and categorizes them into the following categories:

- `PCRetriableResourceException`
- `PCResourceException`

The following exceptions are represented in the fault policies file:

- `PCRetriableResourceException` - A remote fault.
- `PCResourceException` - A binding fault.

### 9.4.1 Configuring the Exception Filter

Exception filter configuration consists of the following steps:

1. Generate a WSDL file.
2. Create a BPEL process with exception filter functionality.
3. Create fault policies and fault binding files.
4. Deploy and test the BPEL process with exception filter functionality.

#### 9.4.1.1 Generating a WSDL File

To generate the WSDL file:

1. Open Application Explorer and create a J2CA configuration.  
For more information, see ["Creating a Configuration for J2CA"](#) on page 2-3.
2. Create a target for the PeopleSoft adapter and then connect to the target.  
For more information, see ["Establishing a Connection \(Target\) for PeopleSoft"](#) on page 2-5.
3. Generate a WSDL for the appropriate object.  
For more information, see ["Generating a WSDL \(J2CA Configurations Only\)"](#) on page 2-10.

#### 9.4.1.2 Creating a BPEL process With Exception Filter Functionality

To create a BPEL process with exception filter functionality:

1. Open JDeveloper and create a new SOA application.



For more information, see [Section 4.4.2, "Creating an Empty Composite for SOA"](#) on page 4-9.

2. Create a new SOA project (for example, Exception\_Filter).
3. Create a third party adapter service component.

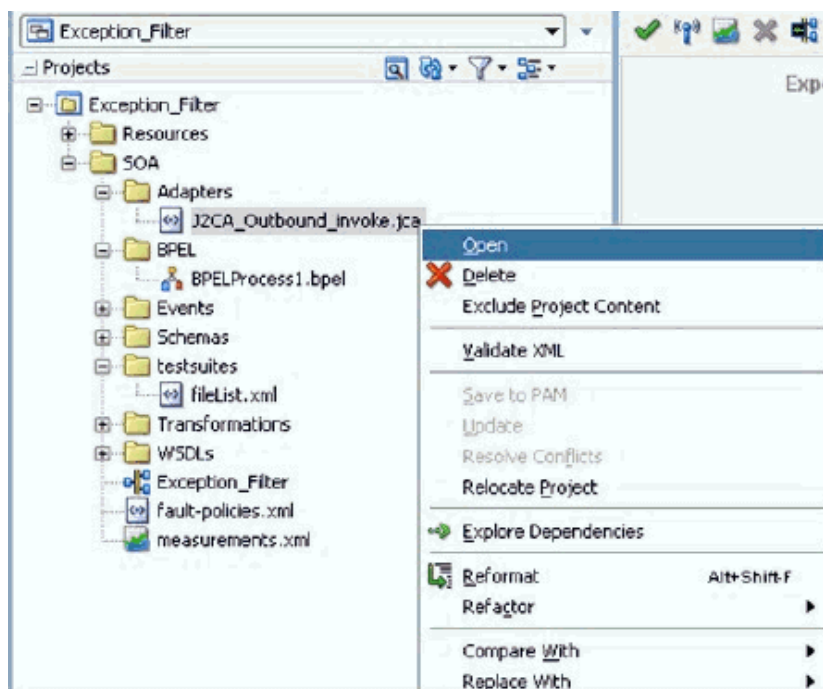
For more information, see [Section 4.4.3.1, "Configuring a Third Party Adapter Service Component"](#) on page 4-11.

Once the third party adapter service component is created, the WSDL file (with corresponding schemas and JCA file) is imported to the JDeveloper project.

For more information, see [Section 4.4.3, "Defining a BPEL Outbound Process"](#) on page 4-11.

4. Modify the imported JCA file.
  - a. Right-click the imported JCA file and select **Open** from the menu, as shown in [Figure 9-72](#).

**Figure 9-72 Application Navigator Tab**



- b. In the <interaction-spec> element, add the ExceptionFilter property. For example:
 

```
<interaction-spec className="com.ibi.afjca.cci.IWAFInteractionSpec">
  <property name="FunctionName" value="PROCESS"/><property
  name="ExceptionFilter"
  value="com.ibi.afjca.oracle.AdapterExceptionFilter"/></interaction-spec>
```
    - c. Save the modified JCA file.
5. Once the third party adapter service component is created and the JCA file is modified, continue with the remainder of the BPEL process creation.

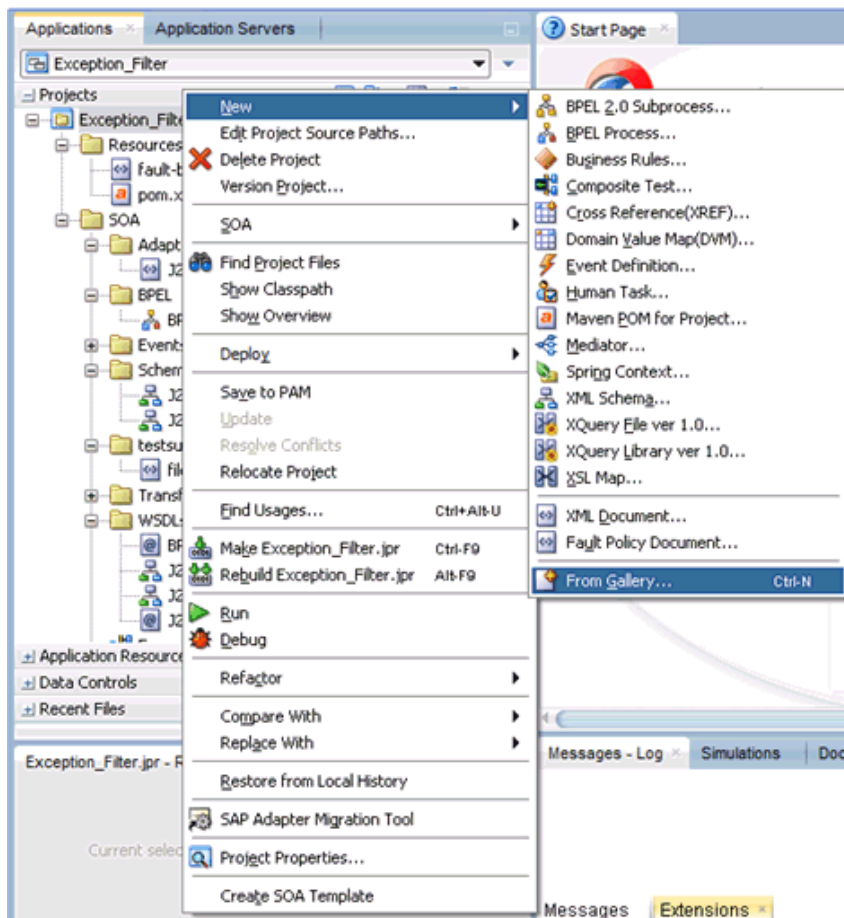
For more information, see [Section 4.4.3, "Defining a BPEL Outbound Process,"](#) on page 4-11.

### 9.4.1.3 Creating Fault Policies and Fault Binding Files

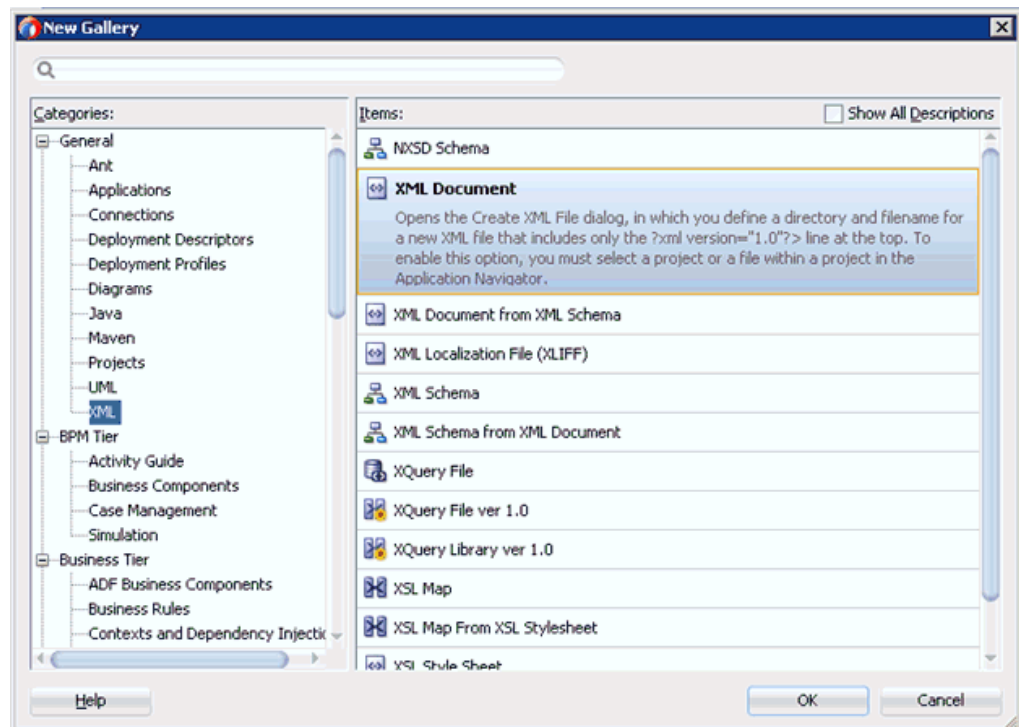
To create fault binding files:

1. Right-click the created SOA project (for example, Exception\_Filter), select **New**, and then click **From Gallery**, as shown in [Figure 9-73](#).

**Figure 9-73 Applications Tab**

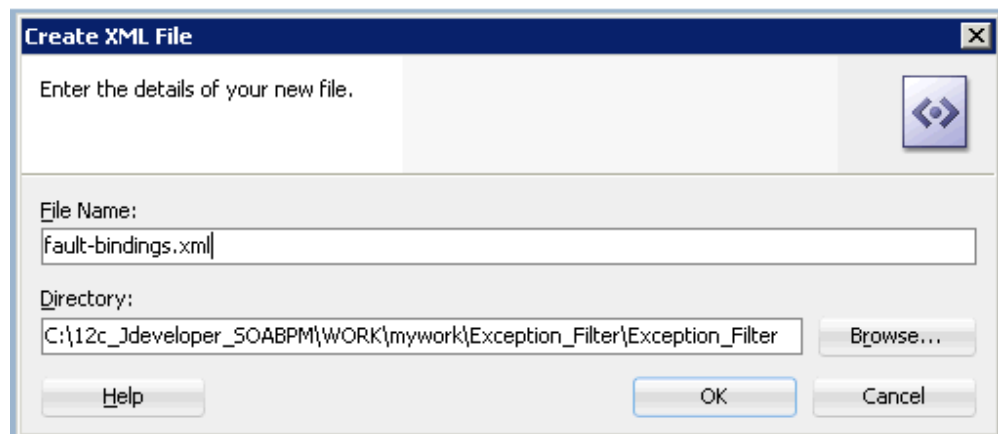


The New Gallery dialog is displayed. Under the General category, click **XML**, as shown in [Figure 9-74](#).

**Figure 9–74** *New Gallery Dialog*

2. Select **XML Document** under Items and then click **OK**.

The Create XML File dialog is displayed, as shown in [Figure 9–75](#).

**Figure 9–75** *Create XML File Dialog*

3. In the File Name field, type **fault-bindings.xml** and click **OK**.
4. Add the appropriate fault binding functions in the **fault-bindings.xml** file.

To view a sample **fault-bindings.xml** file, see "[Sample Fault-Bindings.xml File](#)" on page 9-48.

---

**Note:** The parameter in the <name> element is the name of the created BPEL process.

---

## 5. Save the `fault-bindings.xml` file.

### Sample Fault-Bindings.xml File

```
<?xml version="1.0" encoding="UTF-8" ?>
<faultPolicyBindings version="2.0.1"
xmlns="http://schemas.oracle.com/bpel/faultpolicy"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

<component faultPolicy="bpelFaultHandling">
<name>BPELProcess1</name>
</component>

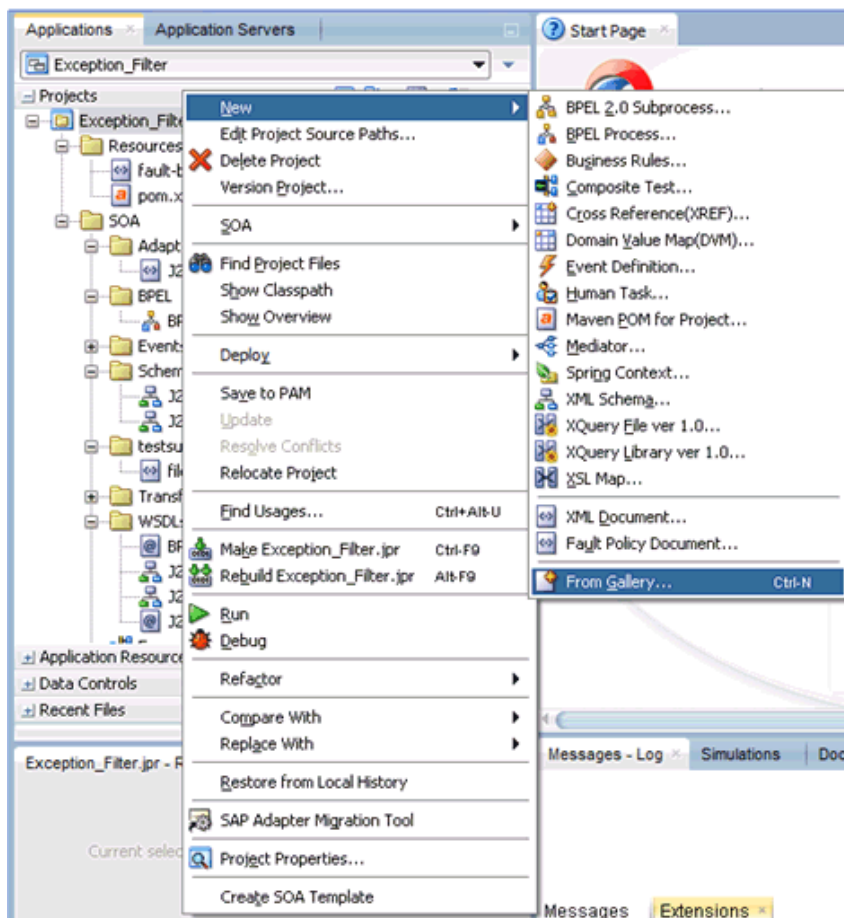
</faultPolicyBindings>
```

### Creating Fault Policies Files

To create fault policies files:

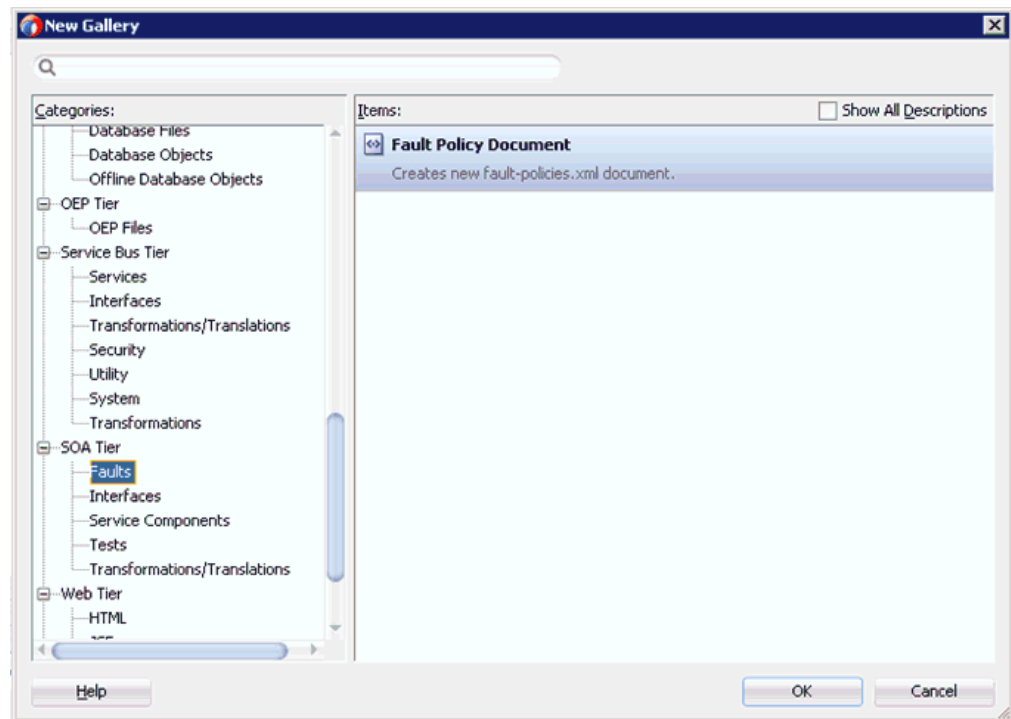
1. Right-click the created SOA project (for example, `Exception_Filter`), select **New**, and then click **From Gallery**, as shown in [Figure 9-76](#).

**Figure 9-76 Applications Tab**



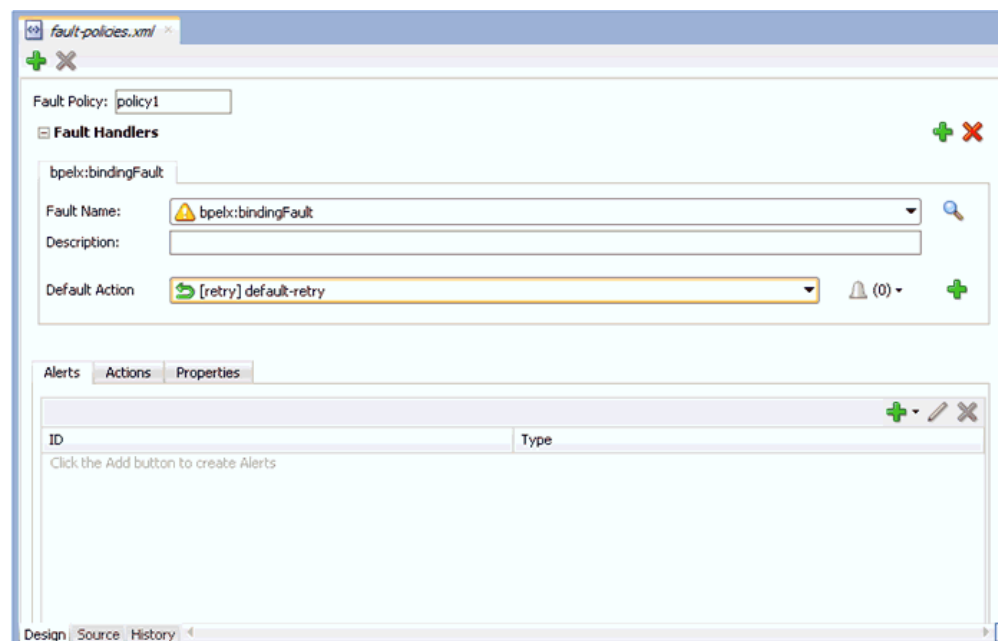
The New Gallery dialog is displayed. Under the SOA Tier category, select **Faults**, as shown in [Figure 9-77](#).

Figure 9-77 New Gallery Dialog



2. Select **Fault Policy Document** under Items and then click **OK**.
3. In the fault-policies.xml tab, select **bpel:bindingFault** from the Fault Name drop-down list and **[retry] default-retry** from the Default Action drop-down list, as shown in Figure 9-78.

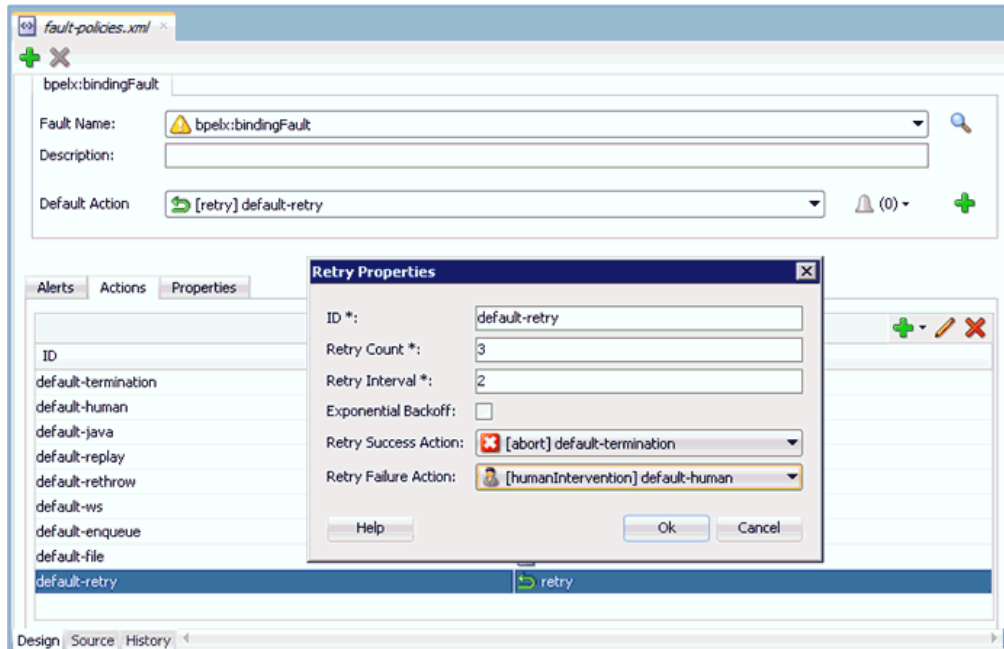
Figure 9-78 Fault-policies.xml Tab



4. Click the **Actions** tab and then double-click **default-retry**.

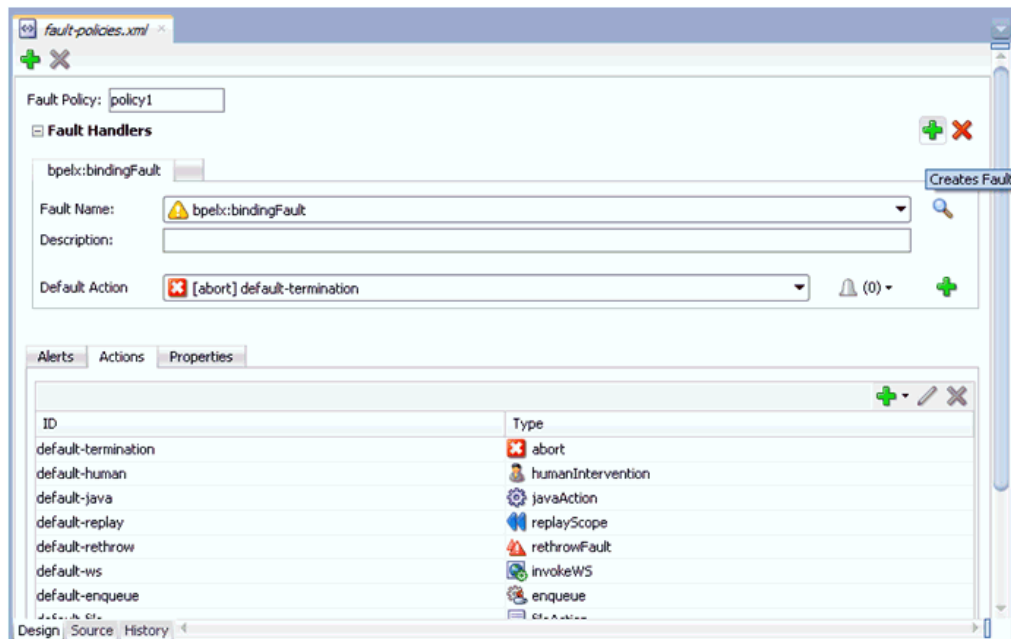
The Retry Properties dialog box is displayed, as shown in [Figure 9–79](#).

**Figure 9–79** *Retry Properties Dialog Box*



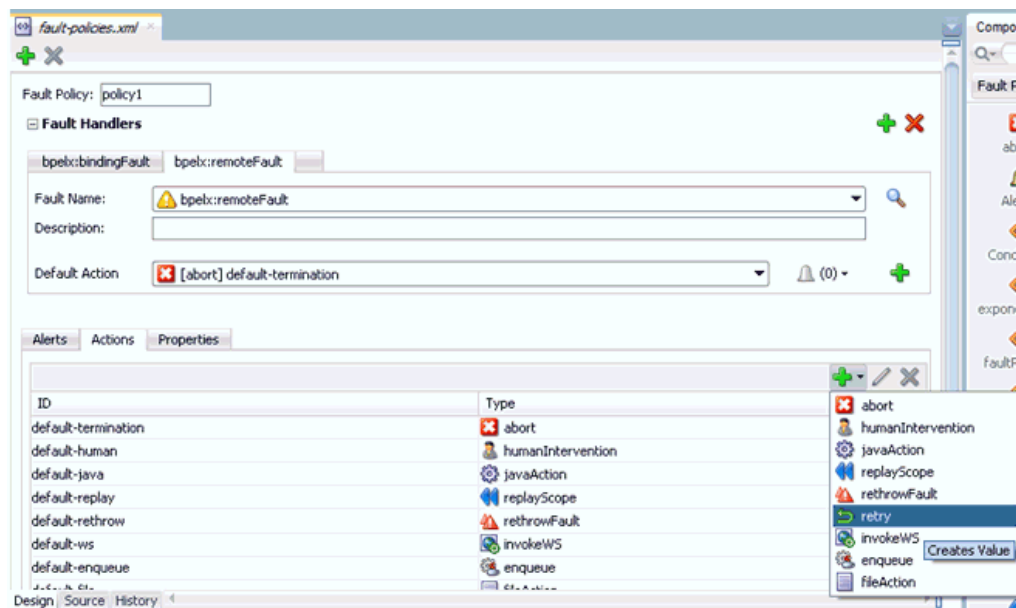
5. Select **[abort] default-termination** from the Retry Success Action drop-down list and **[humanIntervention] default-human** from the Retry Failure Action drop-down list.
6. Click **OK**.
7. Click **Add** to create another fault handler, as shown in [Figure 9–80](#).

**Figure 9–80** *Fault-policies.xml Tab*



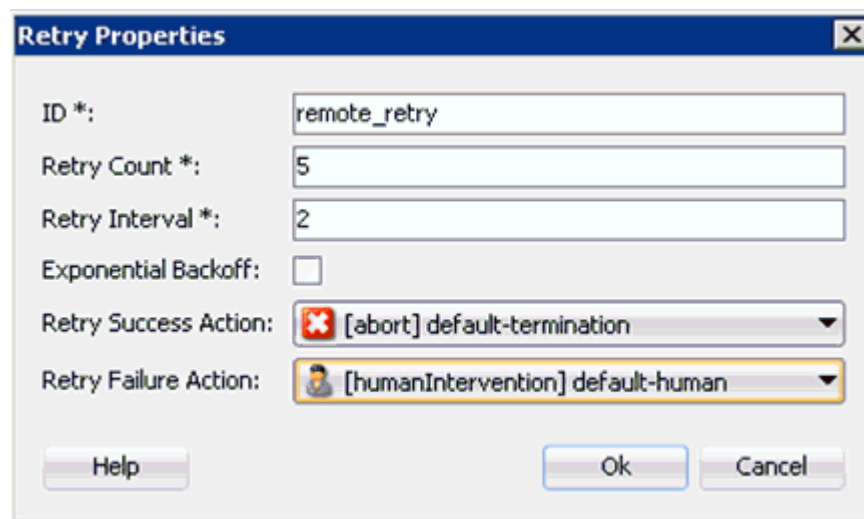
8. In the `fault-policies.xml` tab, select `bpel:remoteFault` from the Fault Name drop-down list and `[abort] default-termination` from the Default Action drop-down list.
9. In the Actions tab, click **Add** and then select `retry`, as shown in [Figure 9–81](#).

**Figure 9–81** *Actions Tab*



The Retry Properties dialog is displayed, as shown in [Figure 9–82](#).

**Figure 9–82** *Retry Properties Dialog Box*

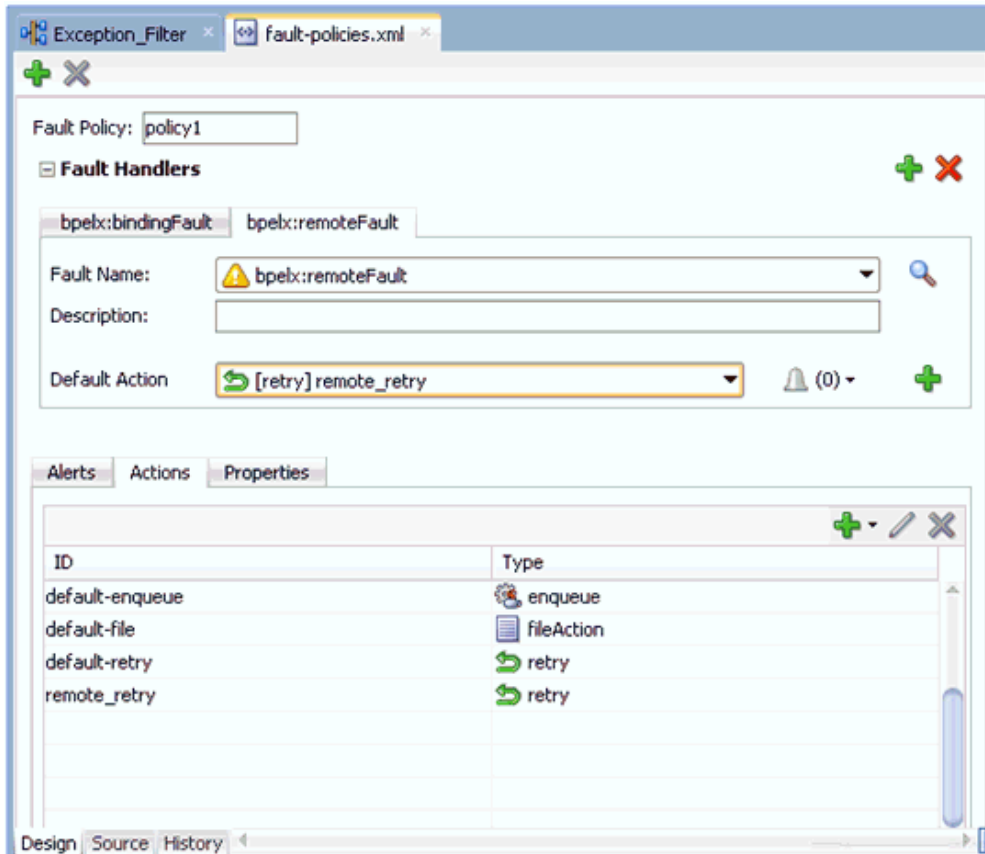


10. Provide values for the ID, Retry Count, and Retry Interval fields.
11. Select `[abort] default-termination` from the Retry Success Action drop-down list and `[humanIntervention] default-human` from the Retry Failure Action drop-down list.
12. Click **OK**.

The created Retry ID will be listed under the Actions tab.

From the Default Action drop-down list, select the newly created Retry ID (for example, remote\_retry) as shown in [Figure 9–83](#).

**Figure 9–83** *Fault-policies.xml* Tab

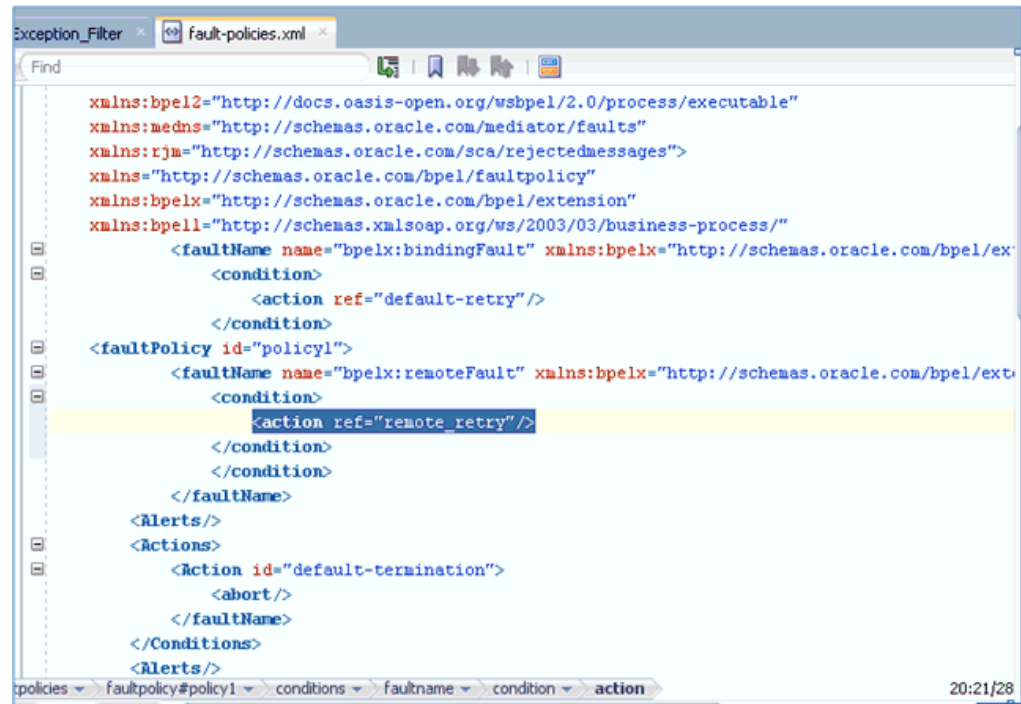


13. Click **Save All**.

14. Click the **Source** tab to verify that the fault policies are added properly, as shown in [Figure 9–84](#).



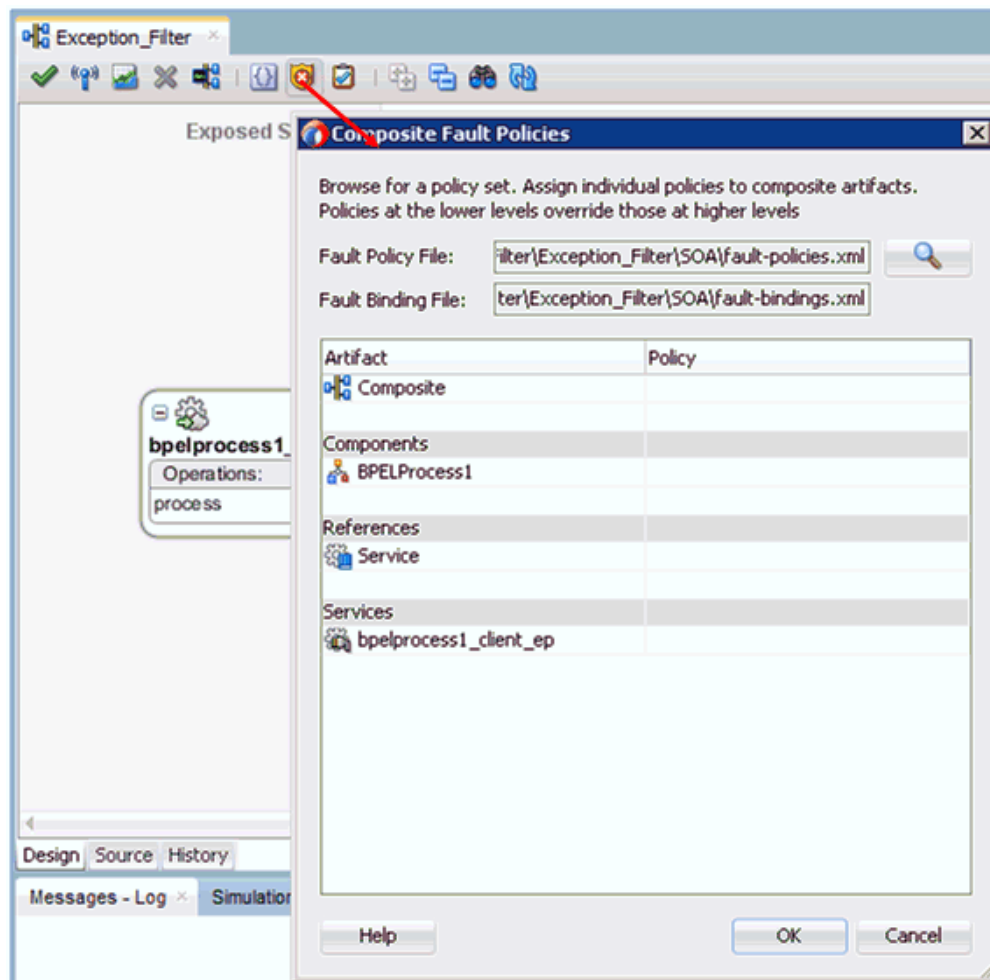
Figure 9–84 Source Tab



15. Double-click the **Exception\_Filter** project and then click **Edit Composite Fault Policies**.

The Composite Fault Policies window is displayed. Ensure that the Fault Policy and the fault-bindings are selected properly, as shown in [Figure 9–85](#).

Figure 9–85 Composite Fault Policies Window



16. Click **Save All**.
17. Click the **Source** tab to verify that the *fault-bindings.xml* and *fault-policies.xml* files are added properly, as shown in Figure 9–86.

Figure 9–86 Source Tab

```

<interface.wsdl interface="http://xmlns.oracle.com/Exception_Filter/Exception_Filter/BPELProcess1.wsdl.interface(BPELProcess1)"
<binding.ws port="http://xmlns.oracle.com/Exception_Filter/Exception_Filter/BPELProcess1.wsdl.endpoint(bpelprocess1_client_ep/B
</service>
<property name="productVersion" type="xs:string" many="false">12.1.3.0.0</property>
<property name="compositeID" type="xs:string" many="false">5e2c51bf-e67a-49b9-8cb2-dee616e506c7</property>
<property name="oracle.composite.faultPolicyFile" type="xs:string" many="false">fault-policies.xml</property>
<property name="oracle.composite.faultBindingFile" type="xs:string" many="false">fault-bindings.xml</property>
<component name="BPELProcess1" version="2.0">
<implementation.bpel src="BPEL/BPELProcess1.bpel"/>
<componentType>
<service name="bpelprocess1_client" ui:wsdlLocation="USDls/BPELProcess1.wsdl">
<interface.wsdl interface="http://xmlns.oracle.com/Exception_Filter/Exception_Filter/BPELProcess1.wsdl.interface(BPELProces
</service>
<reference name="Service" ui:wsdlLocation="USDls/J2CA_Outbound_invoke.wsdl">
<interface.wsdl interface="http://xmlns.oracle.com/pcbpel/iWay/wsdl/PeopleSoft/psft_h852_tgt/browse#wsdl.interface(browsePo
</reference>
</componentType>
<property name="bpel.config.transaction" type="xs:string" many="false">required</property>
</component>
<reference name="Service" ui:wsdlLocation="USDls/J2CA_Outbound_invoke.wsdl">
<interface.wsdl interface="http://xmlns.oracle.com/pcbpel/iWay/wsdl/PeopleSoft/psft_h852_tgt/browse#wsdl.interface(browsePortTy
<binding.jca config="Adapters/J2CA_Outbound_invoke.jca" operation="browse"/>
</reference>
</wire>

```

#### 9.4.1.4 Adjusting for Known Deployment Issues With 12c

For more information on how to adjust for known deployment issues with 12c, see [Section 4.4.3.3, "Adjusting for Known Deployment Issues With 12c"](#) on page 4-26.

#### 9.4.1.5 Deploying and Testing the BPEL Process With Exception Filter Functionality

To deploy and test the BPEL process with exception filter functionality:

1. Deploy the created BPEL process.

For more information, see [Section 4.4.4, "Deploying the BPEL Outbound Process"](#) on page 4-28.

2. Simulate a communication error by disconnecting the system (where the servers are running) from the network.

3. Invoke the deployed BPEL process with a valid input.

For more information, see [Section 4.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console"](#) on page 4-31.

4. Select the process ID.

You can observe the BPEL process being retried or aborted based on the configuration of the **fault-policies.xml** file.

## 9.5 Credential Mapping for Oracle SOA Suite (BPEL, Mediator, or BPM)

This section describes how to configure credential mapping functionality for the Oracle Application Adapter for PeopleSoft in a configuration that uses Oracle SOA Suite (BPEL, Mediator, or BPM). A sample testing scenario is also included. This section contains the following topic:

- [Section 9.5.1, "Configuring Credential Mapping"](#)

Credential mapping is supported only for outbound processes that use J2CA configurations. This feature is not supported for BSE configurations and inbound processes that use J2CA configurations.

---

---

**Note:** The J2CA connector is common to all four application adapters (SAP R/3, PeopleSoft, Siebel, and J.D. Edwards OneWorld). If credential mapping is required, then ensure that only one application adapter is used in a particular instance. For example, in one adapter instance only the PeopleSoft application adapter can be used. Credential mapping cannot be configured at the individual adapter level. If you require the use of credential mapping for two adapters, then both adapters must be running in two independent adapter instances.

---

---

To pass user credentials to the J2CA resource adapter, create a credential map from the Oracle WebLogic Server user credentials to the EIS user credentials (PeopleSoft adapter). These credentials are mapped to the EIS user credentials and then passed to the J2CA container, which uses them to connect with the EIS adapter (PeopleSoft).

## 9.5.1 Configuring Credential Mapping

This section discusses configuring credential mapping, and consists of the following steps and topics:

1. Deploy the adapter.  
For more information, see [Chapter 3, "Oracle WebLogic Server Deployment and Integration"](#).
2. Associate Oracle WebLogic Server credentials with EIS credentials.  
For more information, see [Section 9.5.1.1, "Associating Oracle WebLogic Server Credentials With EIS Credentials"](#) on page 9-56.
3. Generate a WSDL file.  
For more information, see [Section 9.5.1.2, "Generating a WSDL File"](#) on page 9-59.
4. Create and deploy an outbound process.  
For more information, see [Section 9.5.1.3, "Creating and Deploying an Outbound Process"](#) on page 9-60.
5. Invoke and verify that the EIS credentials have passed.  
For more information, see [Section 9.5.1.4, "Verifying the EIS Credentials"](#) on page 9-60.

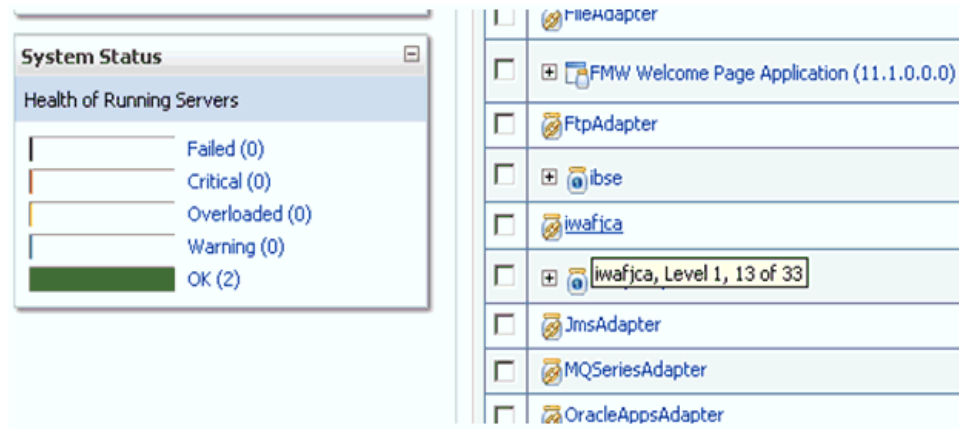
### 9.5.1.1 Associating Oracle WebLogic Server Credentials With EIS Credentials

To associate Oracle WebLogic Server credentials with EIS credentials:

1. Log in to the Oracle WebLogic Server Administration Console.
2. In the Domain Structure section in the left pane, click **Deployments**.

The Deployments page is displayed, as shown in [Figure 9-87](#).

**Figure 9–87 Deployments Page**



3. Click the **iwafjca** resource adapter.

The Settings for iwafjca page is displayed, as shown in [Figure 9–88](#).

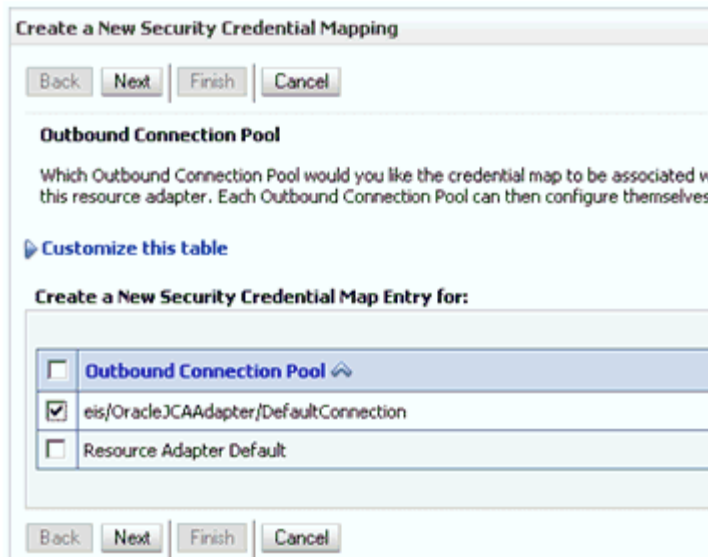
**Figure 9–88 Settings for iwafjca Page**



4. Click the **Outbound Credential Mappings** tab under the Security tab, and then click **New**.

The Create a New Security Credential Mapping page is displayed, as shown in [Figure 9–89](#).

**Figure 9–89 Create a New Security Credential Mapping Page**



5. Select the outbound connection pool.

For example:

`eis/OracleJCAAdapter/DefaultConnection`

6. Click **Next**.

The WebLogic Server User page is displayed, as shown in [Figure 9–90](#).

**Figure 9–90 WebLogic Server User Page**



7. Select **Default User**, enter a valid Oracle WebLogic Server user name, and then click **Next**.

The EIS User Name and Password page is displayed, as shown in [Figure 9–91](#).

**Figure 9–91 EIS User Name and Password Page**

**Create a New Security Credential Mapping**

Back Next Finish Cancel

**EIS User Name and Password**

Configure the EIS User Name and Password that you would like to map the WebLogic Server User to:

\* Indicates required fields

Enter the EIS User Name:

\* EIS User Name::

Enter the EIS Password:

\* EIS Password::

\* Confirm Password::

Back Next Finish Cancel

8. Enter the user name and password for the EIS and click **Finish**.

The credentials for an Oracle WebLogic Server user are now mapped with an EIS user (PeopleSoft). The mapping is invoked automatically before invoking the J2CA service.

### 9.5.1.2 Generating a WSDL File

To generate a WSDL file:

1. Open Application Explorer and create a J2CA configuration.  
For more information, see [Section 2.3.2, "Creating a Configuration for J2CA"](#) on page 2-3.
2. Create a target for the PeopleSoft adapter and then connect to the target.  
For more information, see [Section 2.4, "Establishing a Connection \(Target\) for PeopleSoft"](#) on page 2-5.
3. Generate an outbound WSDL for the appropriate object.  
For more information, see [Section 2.7, "Generating a WSDL \(J2CA Configurations Only\)"](#) on page 2-10.

### 9.5.1.3 Creating and Deploying an Outbound Process

This section describes how to configure an outbound process. For demonstration purposes, specific references to the BPEL outbound process are made. However, the same steps apply to Mediator and BPM outbound processes.

For more information about creating a Mediator outbound process, see [Chapter 5, "Integration With Mediator Service Components in the Oracle SOA Suite"](#).

For more information about creating a BPM outbound process, see [Chapter 6, "Integration With BPM Service Components in the Oracle SOA Suite"](#).

To create and deploy a BPEL outbound process, see the following sections:

- [Section 4.4.2, "Creating an Empty Composite for SOA"](#)

- [Section 4.4.3, "Defining a BPEL Outbound Process"](#)
- [Section 4.4.4, "Deploying the BPEL Outbound Process"](#)

#### 9.5.1.4 Verifying the EIS Credentials

Invoke the input XML and ensure that the EIS target credentials are overridden with the credentials configured in the WebLogic Administration Console for the Default User as described in this section.

1. Invoke the deployed BPEL outbound process with a valid input.  
For more information, see [Section 4.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console"](#) on page 4-31.
2. Check the J2CA log files and locate the encrypted password, which shows that the user credentials have been passed to the EIS through Oracle WebLogic Server.

For example:

```
FINEST IWAManagedConnectionFactory com.ibi.afjca.Util
getPasswordCredential(78) InLoop:
User-iwayqa:Password-ENCR(3109311731831131382333215315332323192322731773172)
FINEST IWAManagedConnectionFactory com.ibi.afjca.Util
getPasswordCredential(90) Use the system PasswordCredential:
User-iwayqa:Password-ENCR(3109311731831131382333215315332323192322731773172)
```

## 9.6 Credential Mapping for Oracle Service Bus (OSB)

This section describes how to configure credential mapping functionality for the Oracle Application Adapter for PeopleSoft in a configuration that uses Oracle Service Bus (OSB). A sample testing scenario is also included. This section contains the following topic:

- [Section 9.6.1, "Configuring Credential Mapping"](#)

Credential mapping is supported only for outbound processes that use J2CA configurations. This feature is not supported for BSE configurations and inbound processes that use J2CA configurations.

---



---

**Note:** The J2CA connector is common to all four application adapters (SAP R/3, PeopleSoft, Siebel, and J.D. Edwards OneWorld). If credential mapping is required, then ensure that only one application adapter is used in a particular instance. For example, in one adapter instance only the PeopleSoft application adapter can be used. Credential mapping cannot be configured at the individual adapter level. If you require the use of credential mapping for two adapters, then both adapters must be running in two independent adapter instances.

---



---

To pass user credentials to the J2CA resource adapter, create a credential map from the Oracle WebLogic Server user credentials to the EIS user credentials (PeopleSoft adapter). Then associate a credential policy with a Web service and invoke the Web service using Oracle WebLogic Server user credentials. These credentials are mapped to the EIS user credentials and then passed to the J2CA container, which uses them to connect with the EIS adapter (PeopleSoft).



## 9.6.1 Configuring Credential Mapping

Configuring credential mapping consists of the following steps:

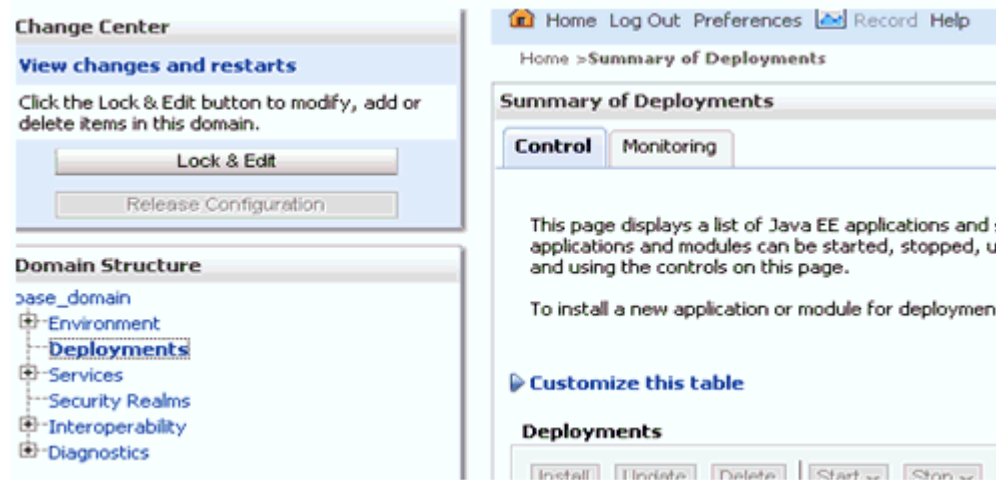
1. Deploy the adapter.  
For more information, see [Chapter 3, "Oracle WebLogic Server Deployment and Integration"](#).
2. Associate Oracle WebLogic Server credentials with EIS credentials.  
For more information, see ["Associating Oracle WebLogic Server Credentials With EIS Credentials"](#) on page 9-61.
3. Generate a WSDL file.  
For more information, see ["Generating a WSDL File"](#) on page 9-64.
4. Create an Oracle Service Bus (OSB) outbound process.  
For more information, see [Section 9.6.1.3, "Creating an Oracle Service Bus \(OSB\) Outbound Process"](#) on page 9-64.

### 9.6.1.1 Associating Oracle WebLogic Server Credentials With EIS Credentials

To associate Oracle WebLogic Server credentials with EIS credentials:

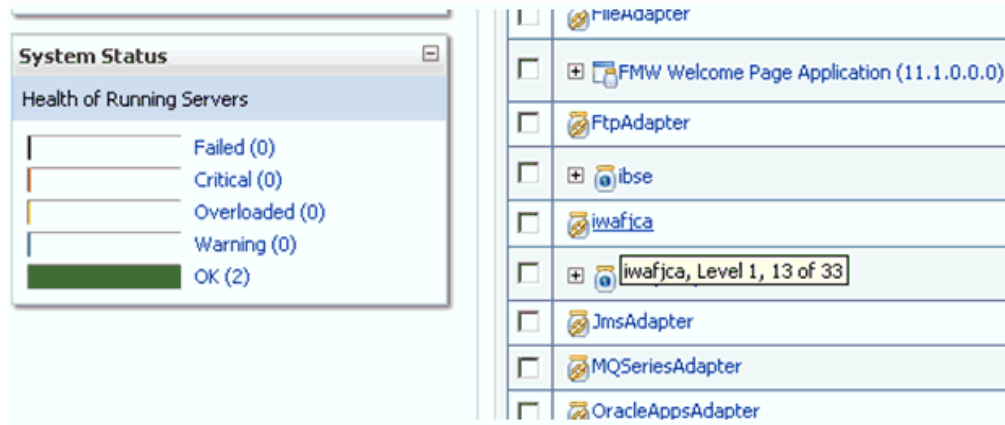
1. Log in to the Oracle WebLogic Server Administration Console.
2. In the Domain Structure section in the left pane, click **Deployments**, as shown in [Figure 9-92](#).

**Figure 9-92 Domain Structure Section**



The Deployments page is displayed, as shown in [Figure 9-93](#).

**Figure 9–93 Deployments Page**



3. Click the `iwafjca` resource adapter.

The Settings for `iwafjca` page is displayed, as shown in [Figure 9–94](#).

**Figure 9–94 Settings for iwafjca Page**



4. Click the **Credential Mappings** tab under the Security tab, and then click **New**.

The Create a New Security Credential Mapping page is displayed, as shown in [Figure 9–95](#).

**Figure 9–95 Create a New Security Credential Mapping Page**

5. Select the outbound connection pool.

For example:

```
eis/OracleJCAAdapter/DefaultConnection
```

6. Click **Next**.

The WebLogic Server User page is displayed, as shown in [Figure 9–96](#).

**Figure 9–96 WebLogic Server User Page**

7. Select **Configured User Name**, enter a valid Oracle WebLogic Server user name, and then click **Next**.

The EIS User Name and Password page is displayed, as shown in [Figure 9–97](#).

**Figure 9–97 EIS User Name and Password Page**

8. Enter the user name and password for the EIS and click **Finish**.

The credentials for an Oracle WebLogic Server user are now mapped with an EIS user (PeopleSoft). The mapping is invoked automatically before invoking the J2CA service.

### 9.6.1.2 Generating a WSDL File

To generate a WSDL file:

1. Set the class path for Application Explorer to integrate with Oracle Service Bus (OSB).  
For more information, see [Section 7.2.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"](#) on page 7-6.
2. Open Application Explorer and create a J2CA configuration.  
For more information, see ["Creating a Configuration for J2CA"](#) on page 2-3.
3. Create a target for the PeopleSoft adapter and then connect to the target.  
For more information, see ["Establishing a Connection \(Target\) for PeopleSoft"](#) on page 2-5.
4. Generate a WSDL for the appropriate object.  
For more information, see [Section 4.4.1, "Generating WSDL for Request/Response Service"](#) on page 4-8.

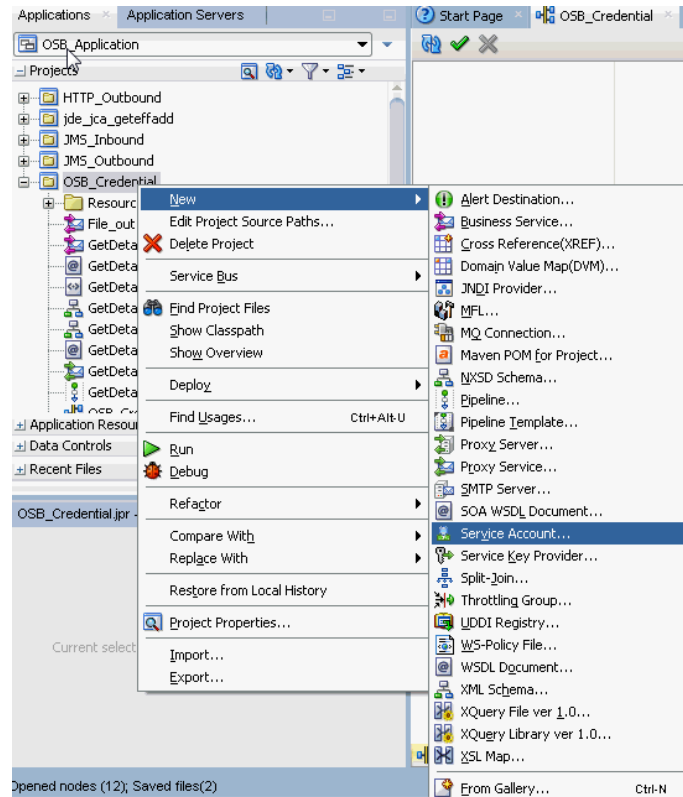
### 9.6.1.3 Creating an Oracle Service Bus (OSB) Outbound Process

This section describes how to create an Oracle Service Bus (OSB) outbound process with credential mapping functionality, using Oracle JDeveloper.

For more information on creating an Oracle Service Bus (OSB) Outbound Process, see [Section 8.1.2, "Defining an OSB Outbound Process"](#) on page 8-3.

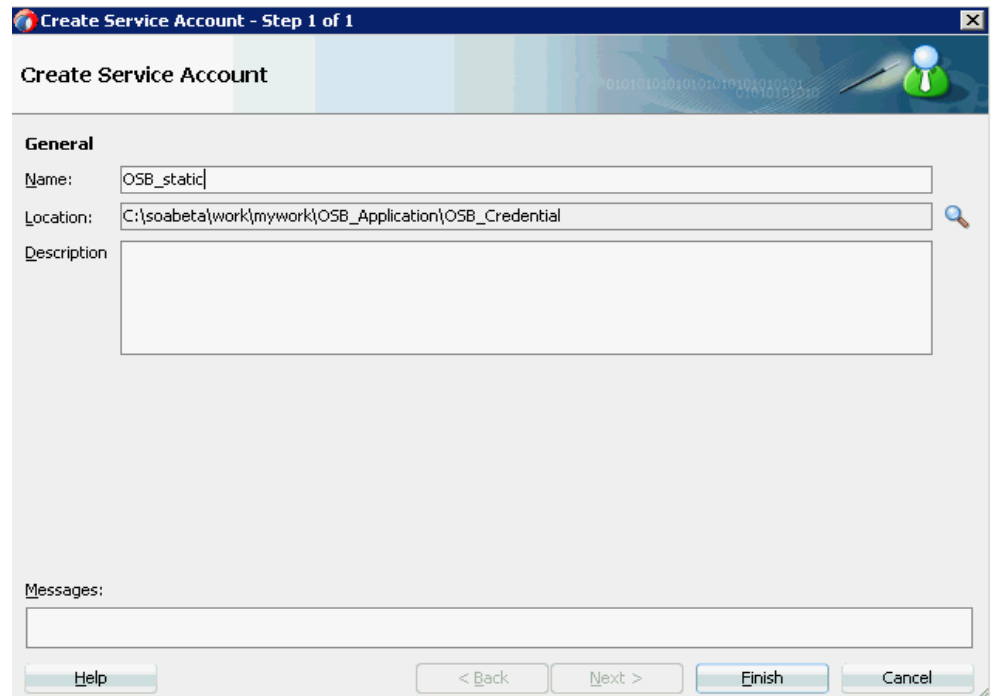
1. Configure a Service account by right-clicking the OSB Project, selecting **New**, and then clicking **Service Account**, as shown in [Figure 9–98](#).

**Figure 9–98 Select Service Account Option**



The Create Service Account pane is displayed, as shown in [Figure 9–99](#).

**Figure 9–99 Create Service Account Pane**

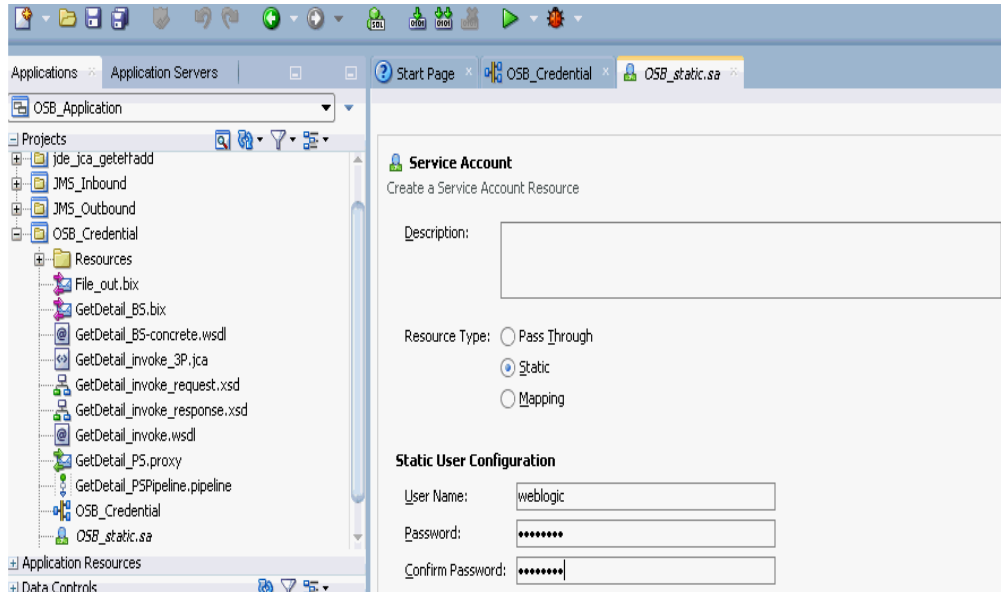


2. Provide a name for the Service Account and click **Finish**.

The configuration page of Service Account is displayed.

3. In the Resource Type section, select **Static**.
4. Provide a valid user name and password for the Oracle WebLogic Server, as shown in [Figure 9-100](#).

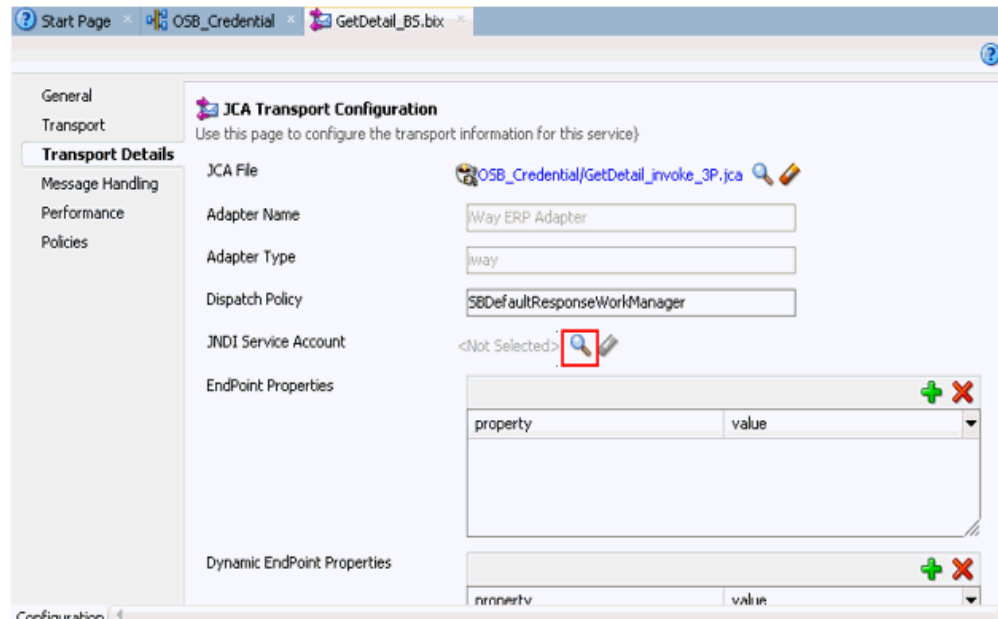
**Figure 9-100 Service Account Configuration Page**



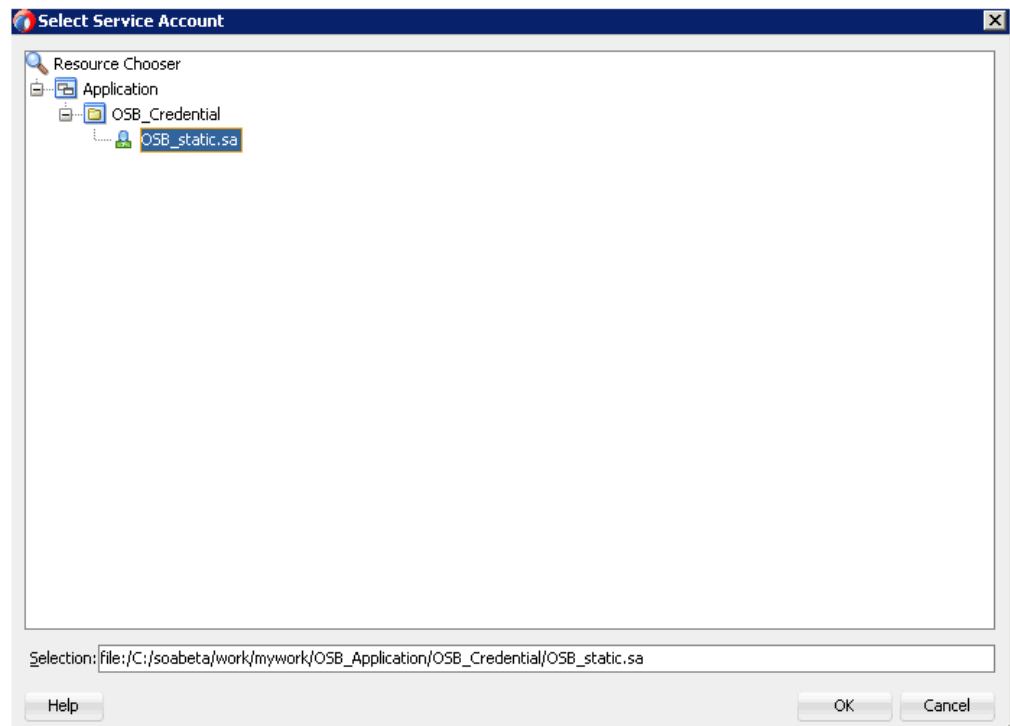
5. Save and close the configuration page.
6. In the composite Editor window, double-click the created WSDL-based Business Service from step 3.

The configuration page of the WSDL-based Business Service is displayed.

7. Select the Transport Details tab, as shown in [Figure 9-101](#).

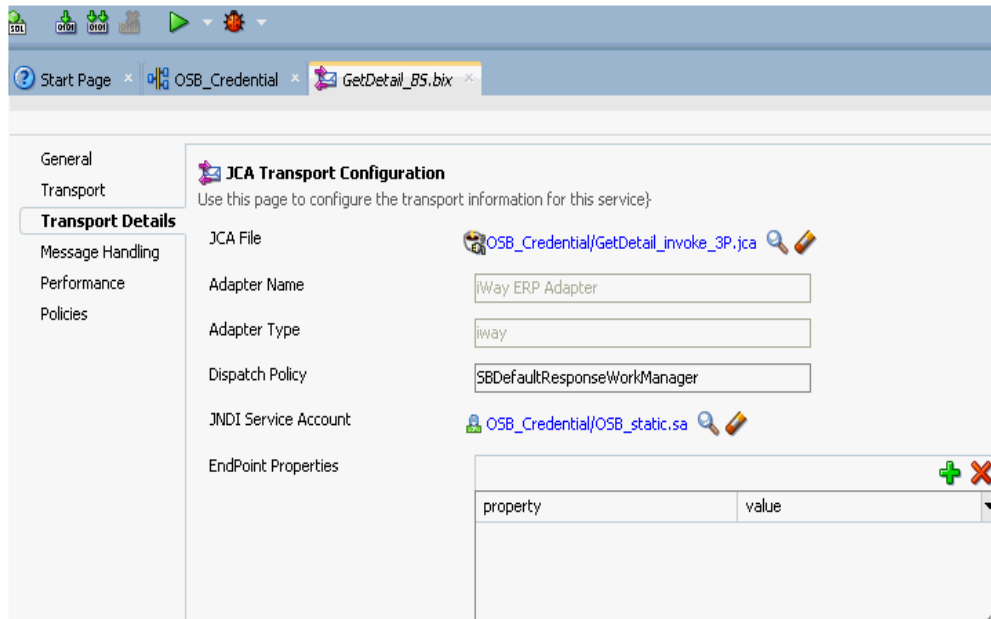
**Figure 9–101 Transport Details Tab**

8. In the JNDI Service Account section, click the Browse icon.  
The Select Service Account window is displayed.
9. Select the created service account and click **OK**, as shown in [Figure 9–102](#).

**Figure 9–102 Select Service Account**

10. Save and close the configuration page, as shown in [Figure 9–103](#)

**Figure 9–103 Business Service Configuration Page**



11. Deploy the OSB process.  
For more information, see [Section 8.1.3, "Deploying the OSB Outbound Process"](#) on page 8-16.
12. Once the process is deployed successfully, copy and paste a valid input XML file in the input folder you configured, and check to see that the output is received in the configured output location.
13. Check the J2CA log files and locate the encrypted password, which shows that the user credentials have been passed to the EIS through Oracle WebLogic Server.

For example:

```
FINEST IWAFManagedConnectionFactory com.ibi.afjca.Util
getPasswordCredential(78) InLoop:
User-iwayqa:Password-ENCR(3189319731831132182333215323332323192322731773252)
FINEST IWAFManagedConnectionFactory com.ibi.afjca.Util
getPasswordCredential(90) Use the system PasswordCredential:
User-iwayqa:Password-ENCR(3109313331831131702333215320132323192322731773236)
```



---

## Troubleshooting and Error Messages

This chapter explains the limitations and workarounds when connecting to PeopleSoft. It contains the following sections:

- [Section 10.1, "Troubleshooting"](#)
- [Section 10.2, "BSE Error Messages"](#)

The adapter-specific errors listed in this chapter can arise whether using the adapter with an Oracle Adapter J2CA or with an Oracle Adapter Business Services Engine (BSE) configuration.

### 10.1 Troubleshooting

This section provides troubleshooting information for PeopleSoft. It contains the following topics:

- [Section 10.1.1, "Application Explorer"](#)
- [Section 10.1.2, "PeopleSoft"](#)
- [Section 10.1.3, "Oracle Adapter J2CA"](#)

Log file information that can be relevant in troubleshooting can be found in the following locations based on your adapter installation:

- The Oracle Adapter J2CA trace information can be found under the following directory:

```
<ADAPTER_HOME>\config\configuration_name\log
```

- BSE trace information can be found under the following directory:

```
<ORACLE_HOME>\user_projects\domains\base_domain\servers\soa_  
server1\stage\ibse\ibse.war\ibselogs
```

- The log file for Application Explorer can be found under the following directory:

```
<ADAPTER_HOME>\tools\iwae\bin
```

#### 10.1.1 Application Explorer

This topic discusses the different types of errors that can occur when using Application Explorer.

Error	Solution
<p>Cannot connect to Oracle Application Adapter for PeopleSoft from Application Explorer. The following error message appears:</p> <pre>Problem activating adapter</pre>	<p>Ensure that:</p> <ul style="list-style-type: none"> <li>■ PeopleSoft is running.</li> <li>■ The PeopleSoft user ID and password are correct.</li> <li>■ The port number is correct.</li> <li>■ The custom component interface is properly installed.</li> </ul>
<p>The following error message appears:</p> <pre>java.lang.IllegalStateException: java.lang.Exception: Error Logon to PeopleSoft System</pre>	<p>You have provided invalid connection information for PeopleSoft or the wrong <code>psj oa . jar</code> is in the <code>lib</code> directory.</p> <p>The <code>psj oa . jar</code> file version is specific to the PeopleTools release.</p>
<p>PeopleSoft does not appear in the Application Explorer Adapter node list.</p>	<p>Ensure that the PeopleSoft JAR files, <code>iwp sci 84 . jar</code> (or <code>iwp sci 81 . jar</code>) and <code>psj oa . jar</code>, are added to the <code>lib</code> directory.</p>
<p>Logon failure error at run-time.</p>	<p>A password is required for connecting to your PeopleSoft system through the Edit option in Application Explorer or when creating a target. The connection password is not saved in <code>repository.xml</code>. Update the password using the Edit option in Application Explorer, then restart the application server.</p>
<p>The following error message appears:</p> <pre>Jolt Session Pool cannot provide a connection to the appserver. This appears to be because there is no available application server domain. [Fri Aug 27 13:06:27 EDT 2004] bea.jolt.ServiceException: Invalid Session</pre>	<p>The host name or port number for PeopleSoft is incorrect.</p>
<p>Properties are not displayed for a component interface.</p>	<p>You are using the wrong <code>iwp sci 8x . jar</code> file.</p>
<p>Cannot generate schemas.</p>	<p>If the error message “Index: -1, Size:0” appears, or if you can log on to Application Explorer but you cannot see any Component Interfaces or Messages, then you may have both the <code>iwp sci 81 . jar</code> and <code>iwp sci 84 . jar</code> files in your <code>lib</code> directory. Stop your server, remove the unrequired jar file, and restart the server.</p>
<p>The following exception occurs when you start Application Explorer by activating <code>ae.bat</code> (not <code>iaexplorer.exe</code>):</p> <pre>java.lang.ClassNotFoundException: org.bouncycastle.jce.provider.Boun cyCastleProvider</pre>	<p>This is a benign exception. It does not affect adapter functionality. Download BouncyCastle files from:</p> <p><a href="ftp://ftp.bouncycastle.org/pub">ftp://ftp.bouncycastle.org/pub</a></p>

Error	Solution
<p>Unable to start Application Explorer in a Solaris environment. The following exception is thrown in the console:</p> <pre> javax.resource.ResourceException: IWAFFManagedConnectionFactory: License violation.at com.ibi.afjca.spi.IWAFFManagedConne ctionFactory.createConnectionFacto ry(IWAFFManagedConnectionFactory.ja va:98)at com.iwaysoftware.iwae.common.JCATr ansport.getConnectionFactory(JCATr ansport.java:133) at com.iwaysoftware.iwae.common.JCATr ansport.initJCA(JCATransport.java: 69)at com.iwaysoftware.iwae.common.JCATr ansport.&lt;init&gt;(JCATransport.java:6 2)at com.iwaysoftware.iwae.common.Adapt erClient.&lt;init&gt;(AdapterClient.java :85)at com.ibi.bse.ConfigWorker.run(Confi gWorker.java:41) at java.lang.Thread.run(Thread.java:5 34)  Could not create the connection factory.</pre>	<p>JAVACMD is not set on the user system. Before starting Application Explorer, export JAVACMD as follows:</p> <p>JAVACMD=/<code>&lt;jdk_home&gt;</code>/bin/java, where <code>&lt;jdk_home&gt;</code> is the directory where JDK is installed on your system.</p>

## 10.1.2 PeopleSoft

Error	Solution
<p>Services are not working properly when using the PeopleSoft Component Interface testing tool in three-tier mode.</p> <p>The following error message appears:</p> <pre> Jolt Session Pool cannot provide a connection to the appserver. This appears to be because there is no available application server domain. [Fri Aug 27 13:06:27 EDT 2004] bea.jolt.ServiceException: Invalid Session</pre> <p>Component Interfaces and Messages do not appear in the adapter tree.</p>	<p>To test properly using the Component Interface testing tool:</p> <ol style="list-style-type: none"> <li>1. Open Application Designer.</li> <li>2. Select the Component Interface.</li> <li>3. Use the test tool.</li> </ol> <p>If service works in test tool, then review the XML and check for redundant fields in XML.</p> <p>The host name or port number for PeopleSoft is incorrect.</p> <p>The project is not installed properly on the PeopleSoft system.</p>

Error	Solution
<p>Return error code -1 is received from PeopleSoft at run-time, for example:</p> <pre>&lt;LOCATIONProcessResponse xmlns="http://xmlns.oracle.com/LOCATION"&gt; &lt;error xmlns=""&gt;-1&lt;/error&gt; &lt;/LOCATIONProcessResponse&gt;</pre> <p>Pstools.properties file has not been initialized.</p>	<p>You are either using the incorrect version of <code>psjoe.jar</code>, or have both the <code>iwpsci81.jar</code> and <code>iwpsci84.jar</code> files in your <code>lib</code> directory. In the second case, you must delete the unused JAR file, and then restart the server.</p> <p>The <code>psjoe.jar</code> file version is specific to the PeopleTools release.</p> <p>This file is required for PeopleSoft 8.1. If you are using PeopleSoft 8.1, you should add this file. If you are not using PeopleSoft 8.1 and this error message still appears, ignore the message.</p>
<p>The following error message appears:</p> <pre>Cannot find Component Interface {CI name}</pre>	<p>The reason may be either of the following:</p> <ul style="list-style-type: none"> <li>■ The Java API for the selected component interface is not found in the API JAR file. Please check the Java API for the class file for the CI. If not found, please add the class file for the CI.</li> <li>■ The component interface name is mentioned incorrectly in the request document.</li> </ul>
<p>The following error message appears:</p> <pre>Not Authorized (90,6) Failed to execute PSSession request</pre>	<p>The component interface does not have the necessary access to perform the operation. Change the permission settings in the PeopleSoft &gt; Security &gt; Permission list for the component interface.</p>
<p>The following error message appears:</p> <pre>Must also provide values for keys {keyname}</pre>	<p>The reason may be any of the following:</p> <ul style="list-style-type: none"> <li>■ The request XML document does not have the element for the mandatory key. Please include the keyname and the value in the request document.</li> <li>■ The Key field name is mentioned incorrectly in the request document.</li> <li>■ The Perform operation is mentioned incorrectly in the request XML document.</li> </ul>

### 10.1.3 Oracle Adapter J2CA

Error	Solution
<p>In Application Explorer, the following error message appears when you attempt to connect to an Oracle Adapter J2CA configuration:</p> <pre>Could not initialize JCA</pre>	<p>In the Details tab in the right pane, ensure that the directory specified in the Home field points to the correct directory, for example:</p> <pre>&lt;ADAPTER_HOME&gt;\tools\iwae\bin\...\...</pre>

## 10.2 BSE Error Messages

This section discusses the different types of errors that can occur when processing Web services through Oracle WebLogic Server Adapter Business Services Engine (BSE).

This section consists of the following topics:

- [Section 10.2.1, "General Error Handling in BSE"](#)
- [Section 10.2.2, "Adapter-Specific Error Handling"](#)

## 10.2.1 General Error Handling in BSE

BSE serves as both a SOAP gateway into the adapter framework and as the engine for some of the adapters. In both design time and run-time, various conditions can cause errors in BSE when Web services that use adapters run. Some of these conditions and resulting errors are exposed the same way, regardless of the specific adapter; others are exposed differently, based on the adapter being used. This topic explains what you can expect when you encounter some of the more common error conditions on an adapter-specific basis.

Usually, the SOAP gateway (**agent**) inside BSE passes a SOAP request message to the adapter required for the Web service. If an error occurs, how it is exposed depends on the adapter and the API or interfaces that the adapter uses. The APIs are generated from PeopleTools and are specific to the PeopleTools release. A few scenarios cause the SOAP gateway to generate a SOAP fault. In general, anytime the SOAP agent inside BSE receives an invalid SOAP request, a SOAP fault element is generated in the SOAP response. The SOAP fault element contains fault string and fault code elements. The fault code contains a description of the SOAP agent error.

The following SOAP response document results when BSE receives an invalid SOAP request:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Client</faultcode>
      <faultstring>Parameter node is missing</faultstring>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

In this example, BSE did not receive an element in the SOAP request message that is mandatory for the WSDL for this Web service.

## 10.2.2 Adapter-Specific Error Handling

When an adapter raises an exception during run-time, the SOAP agent in BSE produces a SOAP fault element in the generated SOAP response. The SOAP fault element contains fault code and fault string elements. The fault string contains the native error description from the adapter target system. Since adapters use the target system interfaces and APIs, whether an exception is raised depends on how the target systems interface or API treats the error condition. If a SOAP request message is passed to an adapter by the SOAP agent in BSE, and that request is invalid based on the WSDL for that service, the adapter may raise an exception yielding a SOAP fault.

While it is almost impossible to anticipate every error condition that an adapter may encounter, the following is a description of how adapters handle common error conditions and how they are then exposed to the Web services consumer application.

This section consists of the following topics:

- [Section 10.2.2.1, "Oracle Application Adapter for PeopleSoft Invalid SOAP Request"](#)
- [Section 10.2.2.2, "Empty Result From PeopleSoft Request"](#)

- [Section 10.2.2.3, "Failure to Connect to PeopleSoft"](#)
- [Section 10.2.2.4, "Invalid SOAP Request"](#)
- [Section 10.2.2.5, "Empty Result From Oracle WebLogic Server Application Adapter Request"](#)

### 10.2.2.1 Oracle Application Adapter for PeopleSoft Invalid SOAP Request

If the PeopleSoft agent receives a SOAP request message that does not conform to the WSDL for the Web service being executed, then the following SOAP response is generated.

```
<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <m:CARRIERResponse xmlns:m="urn:schemas-iwaysoftware-com:iwse"
xmlns="urn:schemas-iwaysoftware-com:iwse"
cid="2A3CB42703EB20203F91951B89F3C5AF">
      <PS8>
        <error>Cannot find Component Interface {VARRIER} (91,2)Initialization
failed (90,7)Not Authorized (90,6)Failed to execute PSSession request Cannot
find Component Interface {VARRIER} (91,2)</error>
      </PS8>
    </m:CARRIERResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### 10.2.2.2 Empty Result From PeopleSoft Request

If Oracle Application Adapter for PeopleSoft executes a component interface as a Web service using input parameters passed in the SOAP request that do not match records in PeopleSoft, then the following SOAP response is generated.

```
<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <m:CARRIERResponse xmlns:m="urn:schemas-iwaysoftware-com:iwse"
xmlns="urn:schemas-iwaysoftware-com:iwse" cid="2A3CB42703EB20203F91951B89F3C5AF">
      <PS8>
        <error>No rows exist for the specified keys. {CARRIER} (91,50)Failed to
execute PSBusComp request</error>
      </PS8>
    </m:CARRIERResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

### 10.2.2.3 Failure to Connect to PeopleSoft

If Oracle Application Adapter for PeopleSoft cannot connect to PeopleSoft, then the following SOAP response is generated.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
  <SOAP-ENV:Fault>
    <faultcode>SOAP-ENV:Server</faultcode>
    <faultstring>java.lang.Exception: Error Logon to PeopleSoft
```

```

    System<faultstring>
  </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

#### 10.2.2.4 Invalid SOAP Request

If Oracle WebLogic Server Application Adapter receives a SOAP request message that does not conform to the WSDL for the Web services being executed, then the following SOAP response is generated.

```

<?xml version="1.0" encoding="ISO-8859-1"
  ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<SOAP-ENV:Body>
  <SOAP-ENV:Fault>
    <faultcode>SOAP-ENV:Server</faultcode>
    <faultstring>RPC server connection failed: Connection refused:
connect</faultstring>
  </SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

#### 10.2.2.5 Empty Result From Oracle WebLogic Server Application Adapter Request

If the adapter executes a SOAP request using input parameters passed that do not match records in the target system, then the following SOAP response is generated.

---

**Note:** The condition for this adapter does not yield a SOAP fault.

---

```

<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <m:RunDBQueryResponse xmlns:m="urn:schemas-iwaysoftware-com:iwse"
      xmlns="urn:schemas-iwaysoftware-com:iwse"
      cid="2A3CB42703EB20203F91951B89F3C5AF">
      <RunDBQueryResult run="1" />
    </m:RunDBQueryResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```





---

---

## Generating Component Interface APIs

This appendix describes how to generate component interface APIs to use with Oracle Application Adapter for PeopleSoft. It contains the following topics:

- [Section A.1, "Building the PeopleSoft API Java Programs"](#)
- [Section A.2, "Compiling the PeopleSoft API Java Programs"](#)

### A.1 Building the PeopleSoft API Java Programs

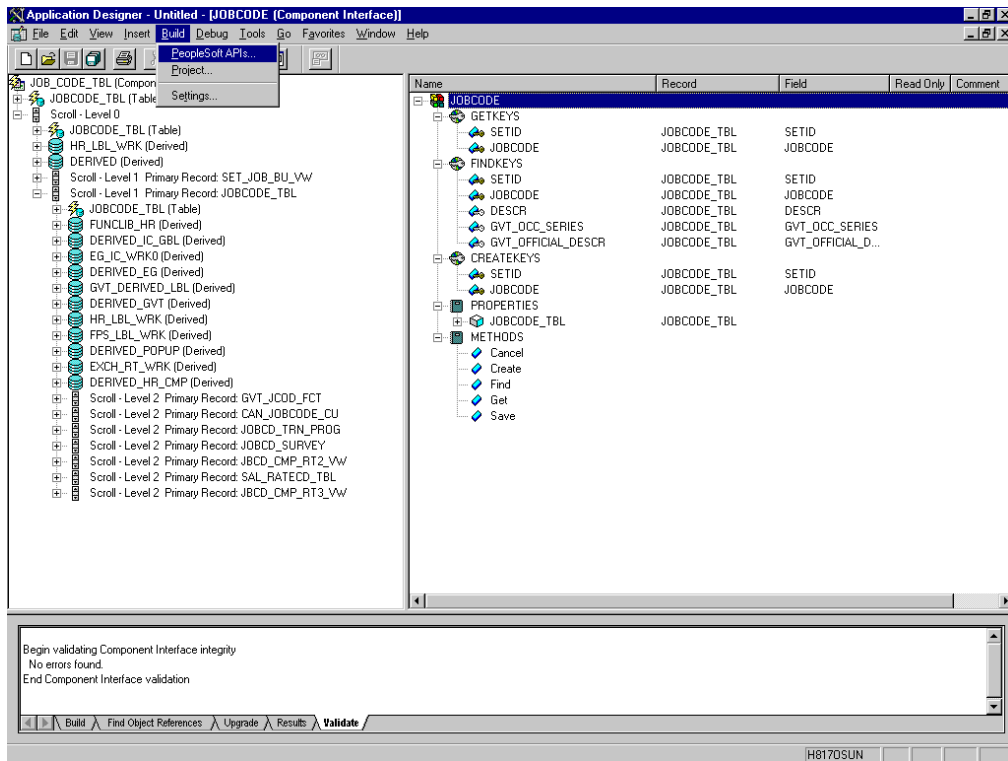
Whether you are using an Enterprise Integration Point (EIP) supplied by PeopleSoft or a customized component interface, you must create a PeopleSoft API to enable communications with the PeopleSoft application. The API is a collection of Java class files that reside on the client system and mediate between the client application layer and PeopleSoft.

Before using your component interface, you must apply security to it and test it.

To build a PeopleSoft API Java program:

1. Open the PeopleSoft Application Designer, as shown in [Figure A-1](#).

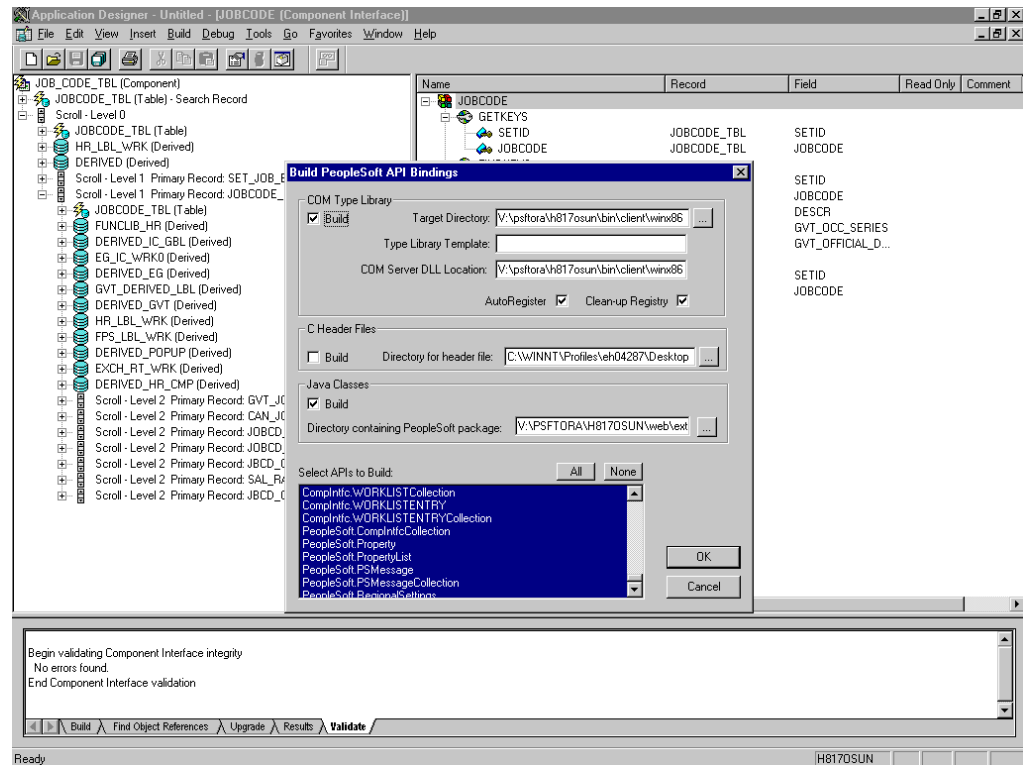
**Figure A-1 PeopleSoft Application Designer Component Interface**



2. From the PeopleSoft Application Designer, open a component interface.
3. Click the right pane and select **PeopleSoft APIs** from the **Build** menu.

The Build PeopleSoft API Bindings dialog prompts you for the types of bindings to create, as shown in [Figure A-2](#).

Figure A-2 Build PeopleSoft API Bindings Dialog



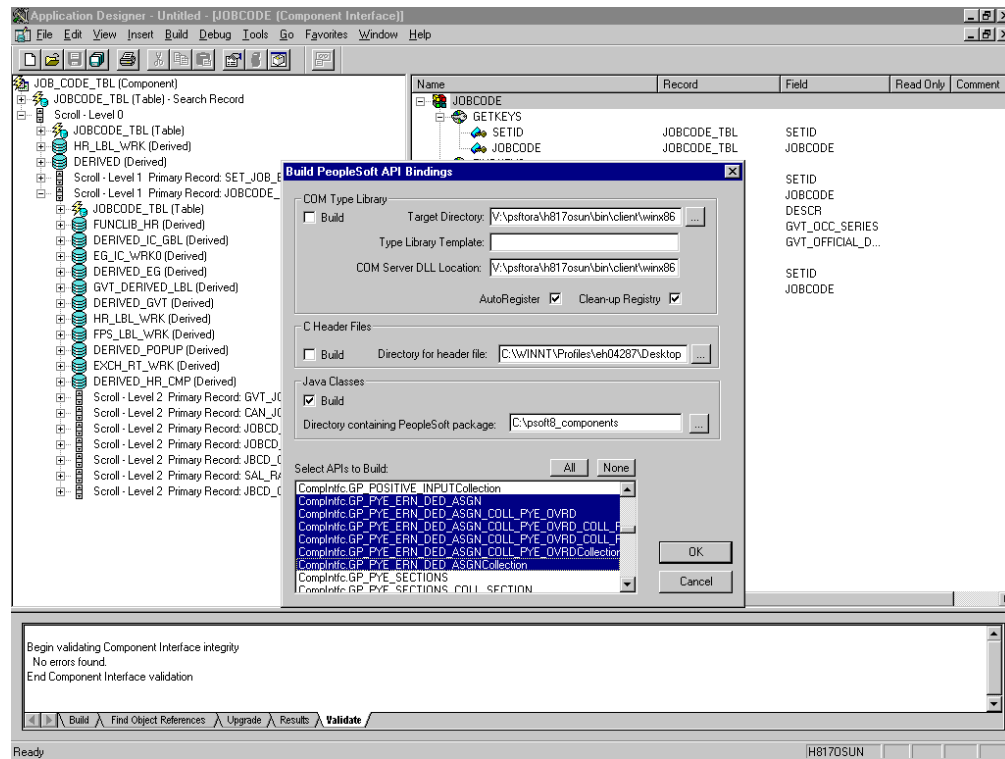
4. Because you are creating Java files, ensure you deselect **COM Type Library Build**.
5. Ensure that **Java Classes Build** is selected, and then select a directory on your local system where the Java files are to be placed, for example, `c:\psft8_components`.
  - To build all API files, select the default, **All** (potentially a large number), and click **OK**.

PeopleSoft generates the files. This takes a few minutes. After the process is complete, a message appears in the output window.

You are now ready to compile the Java files. For more information, see ["Compiling the PeopleSoft API Java Programs"](#) on page A-5.

The following pane illustrates the `GP_PYE_ERN_DED_ASGN` component interface from the HR 8.1 application, as shown in [Figure A-3](#).

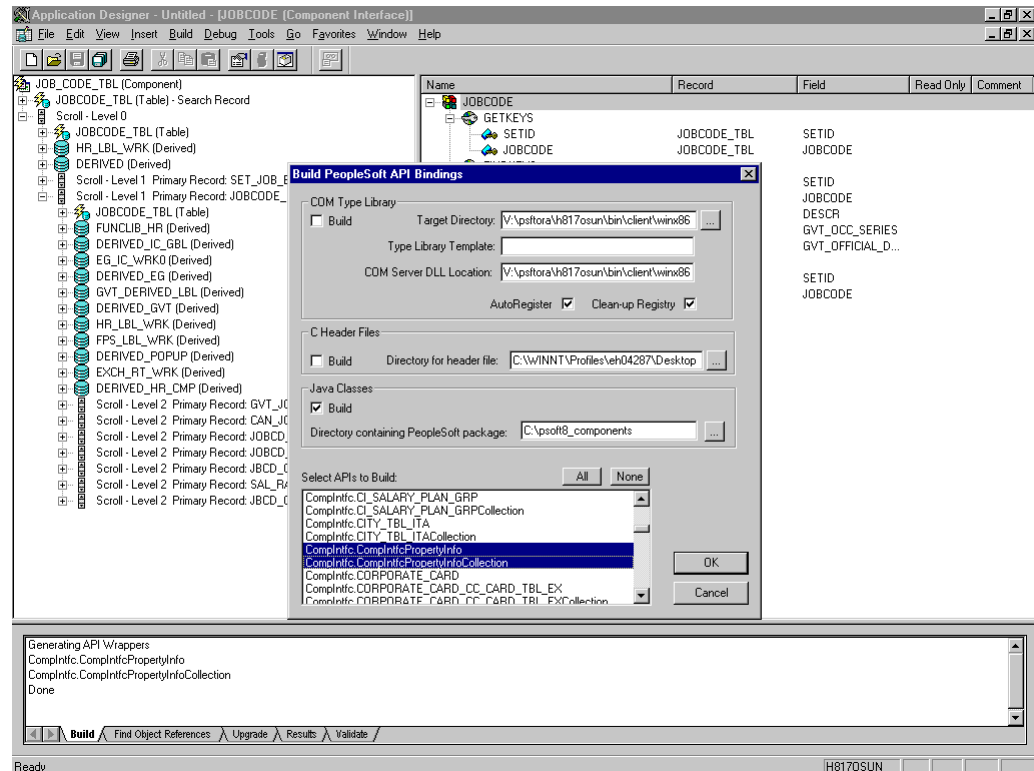
**Figure A-3 GP\_PYE\_ERN\_DED\_ASGN Component Interface**



- To create APIs for a specific component interface or interfaces, click **None**. This clears the selected APIs, so you can select the appropriate APIs for your component interface. These APIs begin with the name of your component interface. There may be fewer than five, or more than 50 APIs, for a particular component interface.

The following dialog displays the following APIs, including generic component interface properties, as shown in [Figure A-4](#).

Figure A-4 Generic Component Interface Properties



- a. In addition to the APIs for the selected component interface, you also must generate the API files for the following generic component interface properties:

`CompIntfcPropertyInfo`

`CompIntfcPropertyInfoCollection`

You may select these items in the same step as the component interface build, or you may select them separately.

- b. Click OK.

PeopleSoft generates the files. This takes a few minutes. After the process is complete, a message appears in the output window. You are now ready to compile the Java files. For more information, see "[Compiling the PeopleSoft API Java Programs](#)" on page A-5.

## A.2 Compiling the PeopleSoft API Java Programs

PeopleSoft places the Java programs to compile in the directory called `psoft8_components\PeopleSoft\Generated\CompIntfc`.

Where `psoft8_components` is the directory specified during the build process.

If you chose to generate all APIs, the systems creates a second directory, `psoft8_components\PeopleSoft\Generated\PeopleSoft`. You are not required to access it.

The process for compiling the PeopleSoft API Java programs depends on whether you are compiling on the system where you installed Application Explorer or on another system.

---



---

**Note:** There are two Java programs for every API file that you selected when you built the Java programs. For more information, see ["Building the PeopleSoft API Java Programs"](#) on page A-1.

---



---

Before you compile the Java programs, you require the PeopleSoft Java Object Adapter, the `psjoa.jar` file that resides on your PeopleSoft Application Server under the `PS_HOME\Web\psjoa` directory. This is the file that you placed in the adapter lib directory during installation.

**If you are compiling on the same system where you installed Application Explorer**

Point to the `psjoa.jar` file or copy it to the directory where you placed the Java API files, for example, `c:\psoft8_components`.

**If you are compiling on a system other than the one where you installed Application Explorer**

Perform the following steps:

1. Obtain a copy of the `psjoa.jar` file from the PeopleSoft Application Server. Ensure that the `psjoa.jar` file is in the Java class path before you compile the programs.
2. Compile the Java programs and ensure that you include the `\PeopleSoft\Generated\CompIntfc` path.

The path is case-sensitive.

The following Windows batch file, run from the `psoft8_components` directory, properly compiles the Java APIs. The code assumes that `psjoa.jar` was placed in `psoft8_components`.

```
@echo off
set JAVA_HOME=<my-java-home>
set PATH=%JAVA_HOME%\bin;%PATH%
set CLASSPATH=%JAVA_HOME%\lib\tools.jar;psjoa.jar;%CLASSPATH%
javac -classpath %CLASSPATH% .\PeopleSoft\Generated\CompIntfc\*.java
```

Where `<my-java-home>` is the fully qualified path name of your Java home directory.

This code places the class files in the same directory with the Java files, but you can choose a different location depending on your site requirements.

3. Compress the class files into a JAR file.

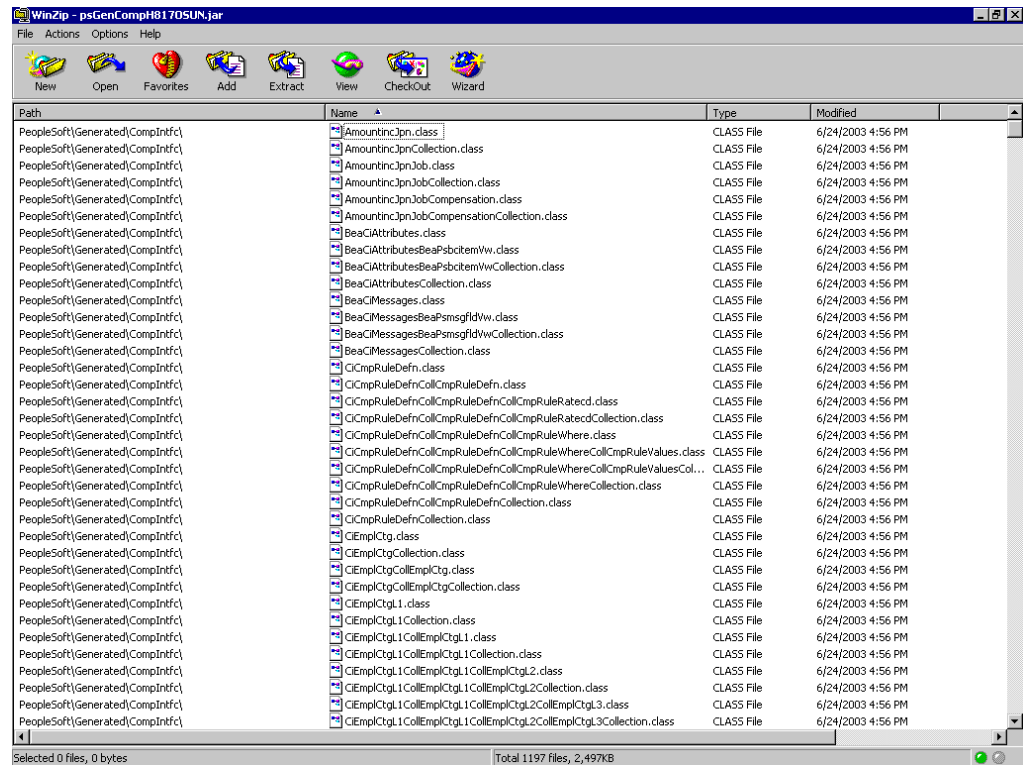
The following Windows batch file, if run from the `psoft8_components` directory, creates a correct JAR file:

```
@echo off
set JAVA_HOME= my-java-home
set PATH=%JAVA_HOME%\bin;%PATH%
set CLASSPATH=%JAVA_HOME%\lib\tools.jar;%CLASSPATH%
jar cvf my-jar-file.jar .\PeopleSoft\Generated\CompIntfc\*.class
```

Where appropriate, substitutions are made for `my-java-home` and `my-jar-file`.

4. To verify that your JAR file is correct, open it with the WinZip application, as shown in [Figure A-5](#).

Figure A-5 PeopleSoft Jar Files



If the JAR file does not use the case-sensitive  
 PeopleSoft\Generated\CompIntfc\ path, you must go back and correct it.

- Place the JAR file in the adapters common lib directory, which enables the Oracle Application Adapter for PeopleSoft to communicate with the PeopleSoft component interface.

```
<ADAPTER_HOME>\lib
```

---

**Note:** If you run on UNIX, perform the compile and JAR steps on Windows and then move the file to your UNIX system. The JAR file is binary. If you use an FTP-based tool to move your JAR file from Windows to UNIX, the file format must be set to binary.

---





---

---

## Configuring the PeopleSoft Message Router

This appendix describes how to configure and test a TCP/IP or HTTP target connector and a TCP/IP handler for PeopleSoft. It contains the following topics:

- [Section B.1, "Configuring the TCP/IP or HTTP Target Connector for PeopleSoft 8.4"](#)
- [Section B.2, "Configuring the TCP/IP Target Connector"](#)
- [Section B.3, "Configuring the Node for the TCP/IP 8.4 Connector"](#)
- [Section B.4, "Configuring the TCP/IP Handler for PeopleSoft 8.1"](#)
- [Section B.5, "Configuring the HTTP Handler for PeopleSoft 8.1"](#)
- [Section B.6, "Testing Your PeopleSoft Configuration"](#)

The information in this appendix assumes that you are familiar with PeopleSoft Integration Broker (in release 8.4) or Application Messaging (in release 8.1). For more information, see [Appendix D, "Using PeopleSoft Integration Broker"](#). For a complete description *before* you work with Oracle Application Adapter for PeopleSoft, see your PeopleSoft documentation.

---

---

**Note:** In PeopleSoft release 8.1, the messaging architecture is called Application Messaging and includes Application Messaging Gateway. In release 8.4, the messaging architecture is called Integration Broker, which includes Integration Gateway. When discussing release-independent issues, this section uses release 8.4 terminology. When discussing release-specific issues, it uses release-specific terminology.

---

---

### B.1 Configuring the TCP/IP or HTTP Target Connector for PeopleSoft 8.4

The procedures in this topic assume that your Integration Broker environment is configured and tested. For more information, see [Appendix D, "Using PeopleSoft Integration Broker"](#).

1. Configure the gateway for the TCP/IP Target Connector or HTTP Target Connector. For more information, see ["Configuring the TCP/IP Target Connector"](#) on page B-2.

**Note:** This step is optional when configuring the HTTP Connector. The HTTP Target Connector is supplied with your PeopleSoft application, and no special configuration steps are required. If you choose, you may configure default connection values on the Gateway. You can override these values when you configure the node.

2. Configure the node. For more information, see "[Configuring the Node for the TCP/IP 8.4 Connector](#)" on page B-3.

**Note:** Starting with release 8.4, the Integration Broker is delivered with an HTTP Outbound Connector. For more information, see "[Configuring the HTTP Connector](#)" on page B-8.

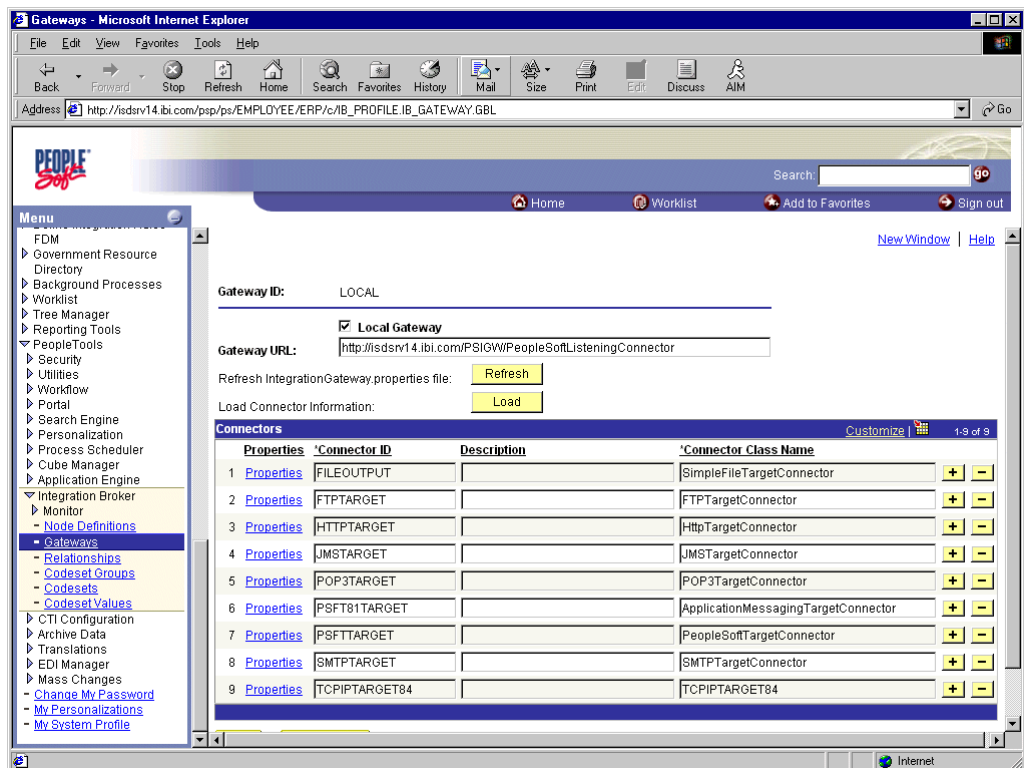
## B.2 Configuring the TCP/IP Target Connector

To configure the gateway for the TCP/IP Target Connector:

1. In a Web browser, open your PeopleSoft release 8.4 application.
2. In the menu pane, expand **PeopleTools**, then expand **Integration Broker**, and click **Gateways**.
3. Open the **LOCAL Gateway ID**.

A pane similar to the following Gateway ID pane is displayed, as shown in [Figure B-1](#).

**Figure B-1 Gateway ID Pane**



4. If you do not see the TCPIPTARGET84 Connector ID, click **Load** and scroll to locate **TCPIPTARGET84** in the list.

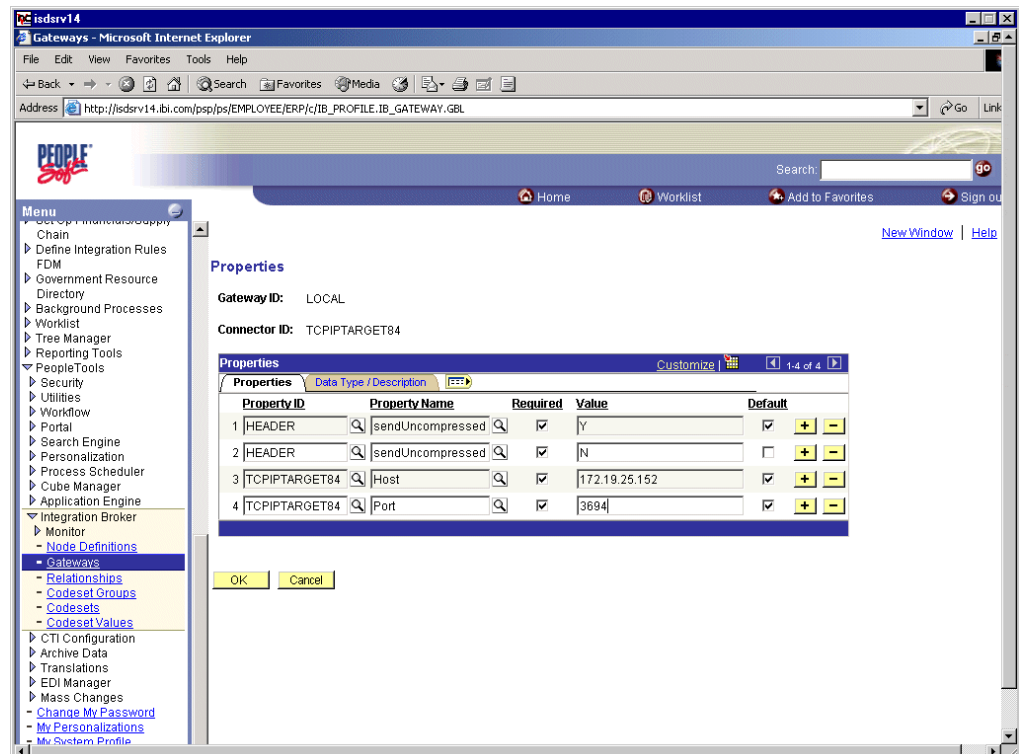
If TCPIPTARGET84 still does not appear, the connector class file was not installed in the Integration Gateway.

Perform the following steps:

- a. Click the **Properties** URL for TCPIPTARGET84.

The Properties pane for TCPIPTARGET84 is displayed, as shown in [Figure B-2](#). Default values appear for the host and the port. For complex business situations, you can override this setting on the individual node.

**Figure B-2 Properties Pane**



- b. Enter the values for the host and the port for the system on which your PeopleSoft XML listener is listening for incoming messages.
  5. Click **OK**.
- The Gateway window is displayed.
6. Scroll to the bottom of the window and click **Save**.

You have finished configuring the gateway for the TCP/IP Target Connector.

## B.3 Configuring the Node for the TCP/IP 8.4 Connector

This section consists of the following topics:

- [Section B.3.1, "Configuring the HTTP Target Connector"](#)
- [Section B.3.2, "Configuring the HTTP Connector"](#)

To configure the node for the TCP/IP 8.4 Connector:

1. In the Menu pane, select **PeopleTools, Integration Broker**, and then click **Node Definitions**.
2. Select the node that you want to configure.

---

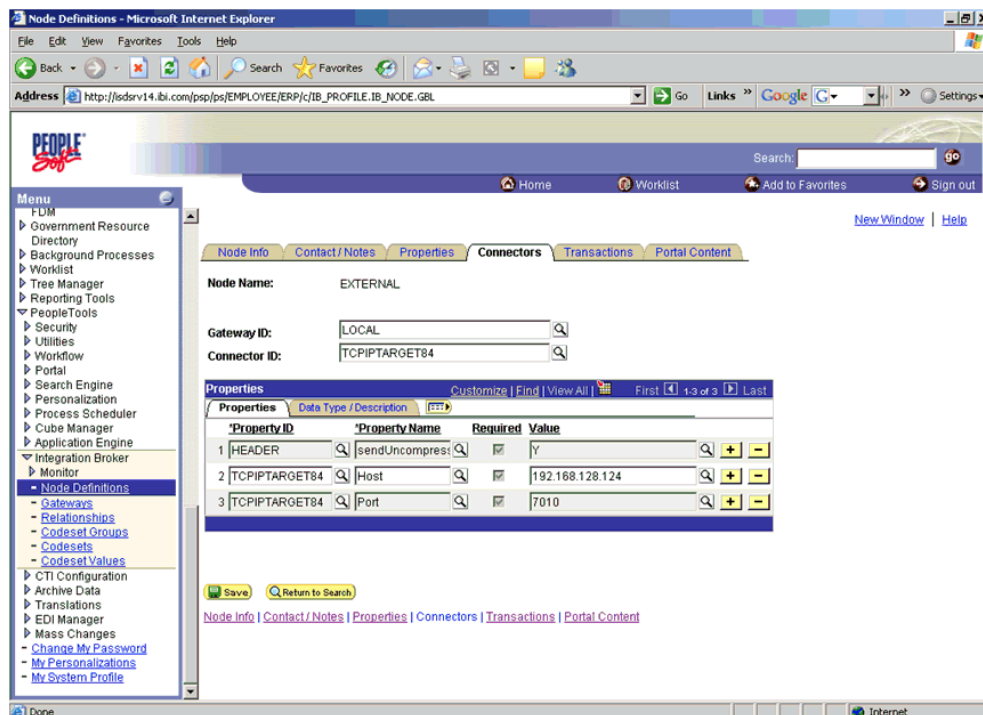
**Note:** This procedure uses a node called EXTERNAL. For more information about creating and using nodes, see [Appendix D, "Using PeopleSoft Integration Broker"](#) or your PeopleSoft documentation.

---

Perform the following steps:

- a. Select **External** from the **Node Type** list.
  - b. Select **Implicit** from the **Routing Type** list.
3. Select the **Connectors** tab, as shown in [Figure B-3](#).

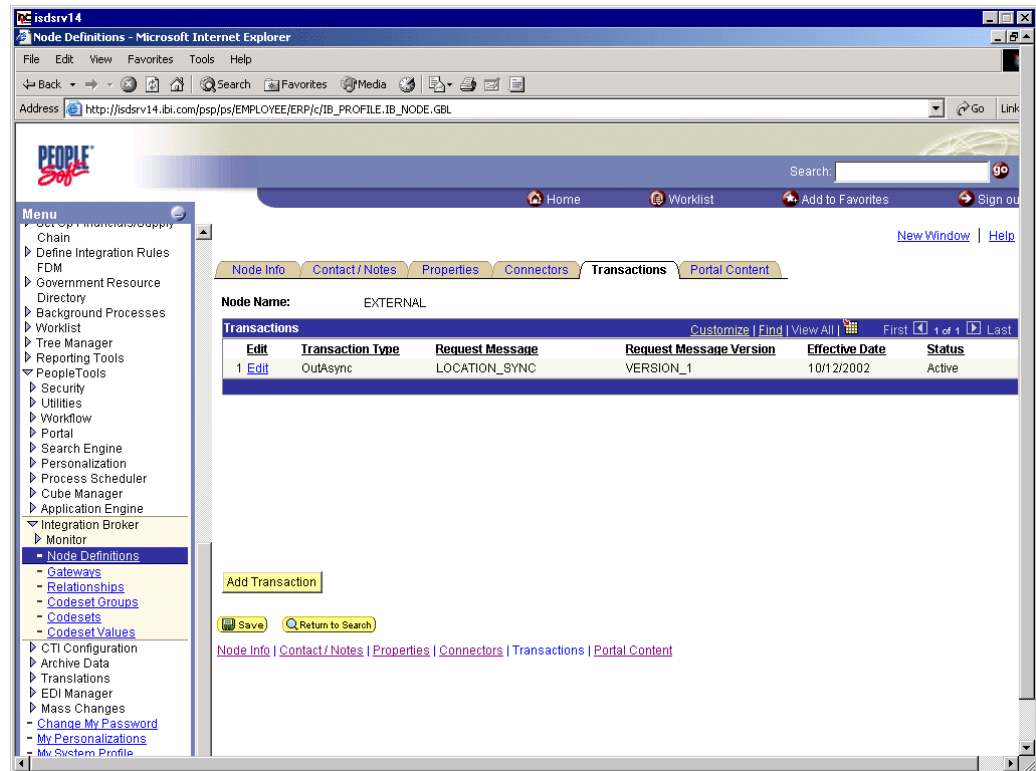
**Figure B-3 Connectors Tab**



Perform the following steps:

- a. Select **TCPIPTARGET84** as the Connector ID.  
Default values appear for the host and the port.
  - b. Enter the values for the host and the port for the system and port that route XML to Oracle WebLogic Server. You can accept or override the default values for individual nodes.
  - c. Click **Save**.
4. If you are warned that you are changing the connector, click **OK**.
  5. Select the **Transactions** tab, as shown in [Figure B-4](#).

Figure B-4 Transactions Tab

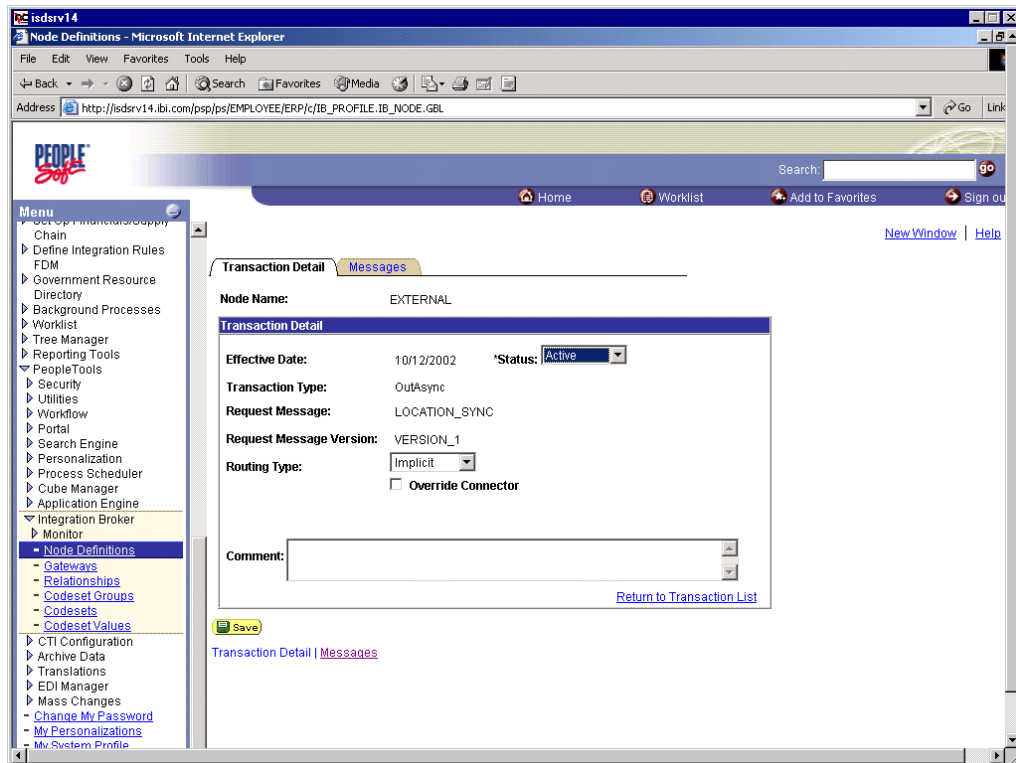


Perform the following steps:

- a. If there are no transactions, click **Add Transaction** to add the message with which you are working. In this procedure, the node is already configured with the LOCATION\_SYNC message.
- b. To view transaction details for the LOCATION\_SYNC message, click **Edit**.

The Transaction Detail tab appears, as shown in [Figure B-5](#).

**Figure B-5 Transaction Detail Tab**



- c. Add the message with which you are working.
  - d. Verify that the **Routing Type** is **Implicit**.
6. Click **Save**.
  7. Return to the **Transactions** tab.
 

Perform the following steps:

    - a. Edit additional transactions by clicking **Edit** and navigating to the **Transaction Detail** tab.
    - b. In the **Transaction Detail** tab, select **Inactive** from the **Status** list.
 

Inactive status is for initial testing only. After you test your configuration, you may change the status to Active and have as many nodes and transactions as required to satisfy your business requirements.
  8. Click **Save**.

You can now send XML messages to your PeopleSoft XML listener.

### B.3.1 Configuring the HTTP Target Connector

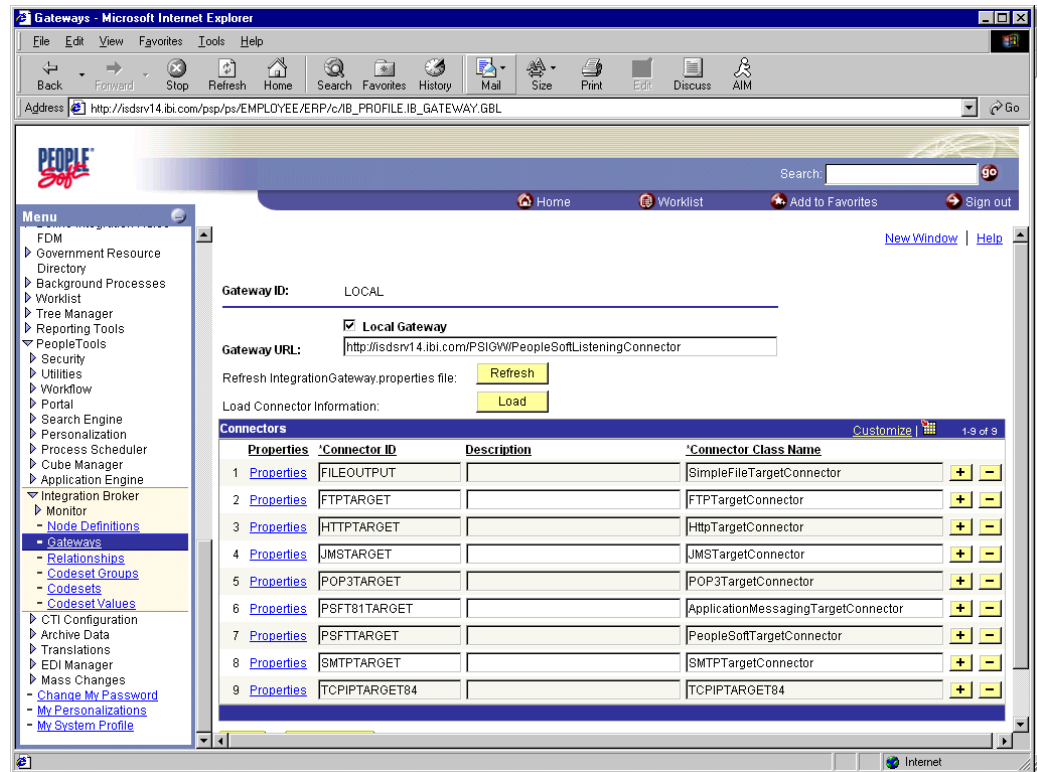
The HTTP Target Connector is supplied with your PeopleSoft application, and no special configuration steps are required. If you choose, you may configure default connection values on the Gateway. You can override these values when you configure the node. To configure the gateway for the HTTP Target Connector:

1. In a Web browser, open your PeopleSoft 8.4 application.
2. In the Menu pane, expand **PeopleTools**, then expand **Integration Broker**, and click **Gateways**.

3. Open the **LOCAL Gateway ID**.

A pane similar to the following Gateway ID pane is displayed, as shown in [Figure B-6](#).

**Figure B-6 HTTP Target Connector Gateway ID Pane**



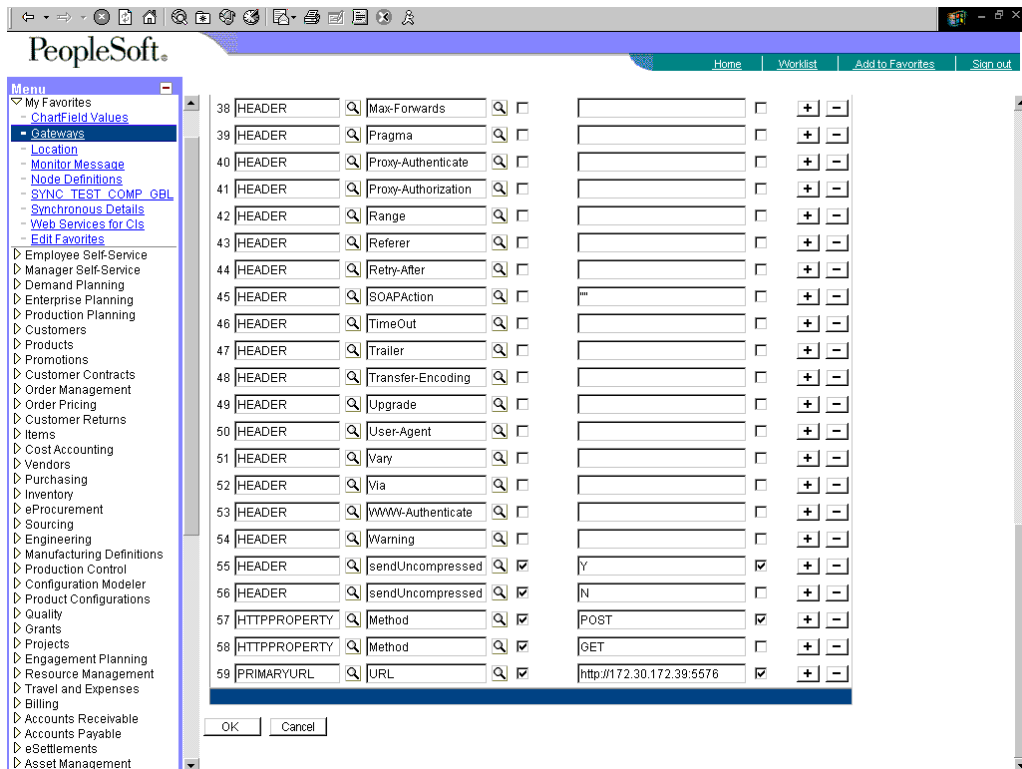
4. If you do not see the **HTTPTARGET Connector ID**, click **Load**.

If it does not appear, your Gateway was not installed properly. Check with your PeopleSoft system administrator.

5. Click the **Properties** URL for **HTTPTARGET**.

The Properties pane for **HTTPTARGET** displays default values, as shown in [Figure B-7](#).

**Figure B-7 HTTPTARGET Properties Pane**



6. Scroll to the bottom and enter a value for the **PRIMARYURL**.

This is the default HTTP address (system and port) on which your PeopleSoft XML listener is listening for incoming messages.

---

**Note:** For complex business situations, you can override this setting on the individual node.

---

7. Click **OK**.

The Gateway window is displayed.

8. Scroll to the bottom of the window and click **Save**.

You have finished configuring the gateway for the HTTP Target Connector.

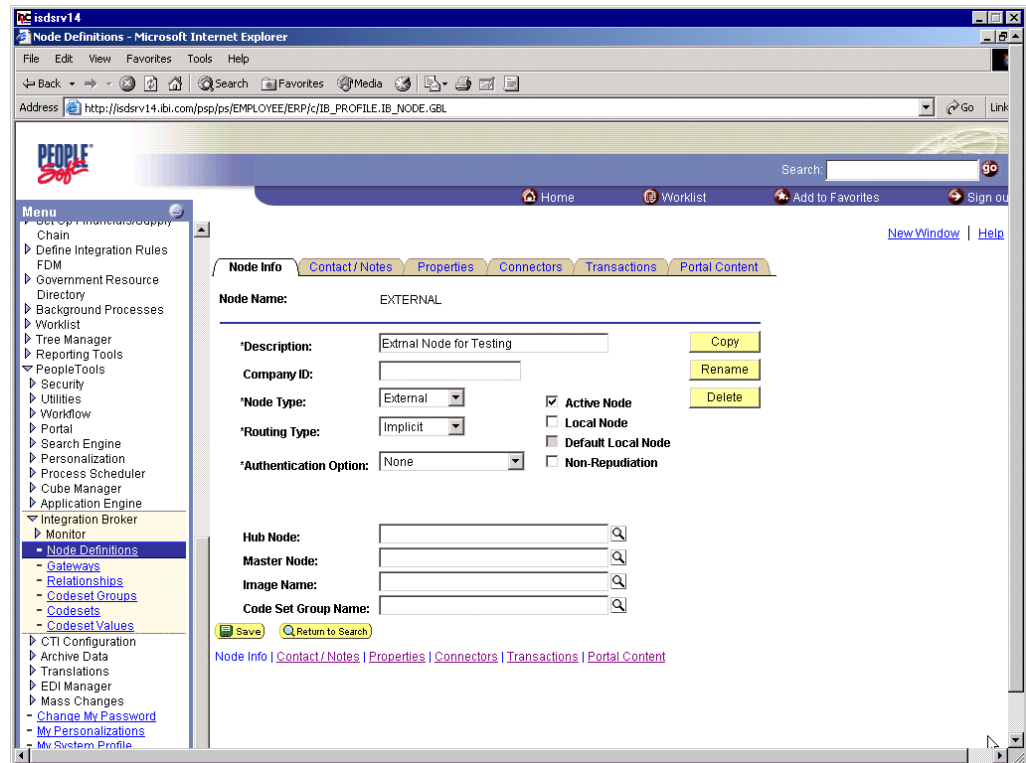
### B.3.2 Configuring the HTTP Connector

Starting with release 8.4, the Integration Broker is delivered with an HTTP Outbound Connector. This connector can be used in place of the TCP/IP 8.4 connector for sending messages to Oracle WebLogic Server.

To configure the node to use the HTTP Connector:



Figure B–8 Connector Configuration Pane



1. In the Menu pane, expand **PeopleTools**, **Integration Broker**, and then click **Node Definitions**, as shown in Figure B–8.
2. Select the node that you want to configure.

---

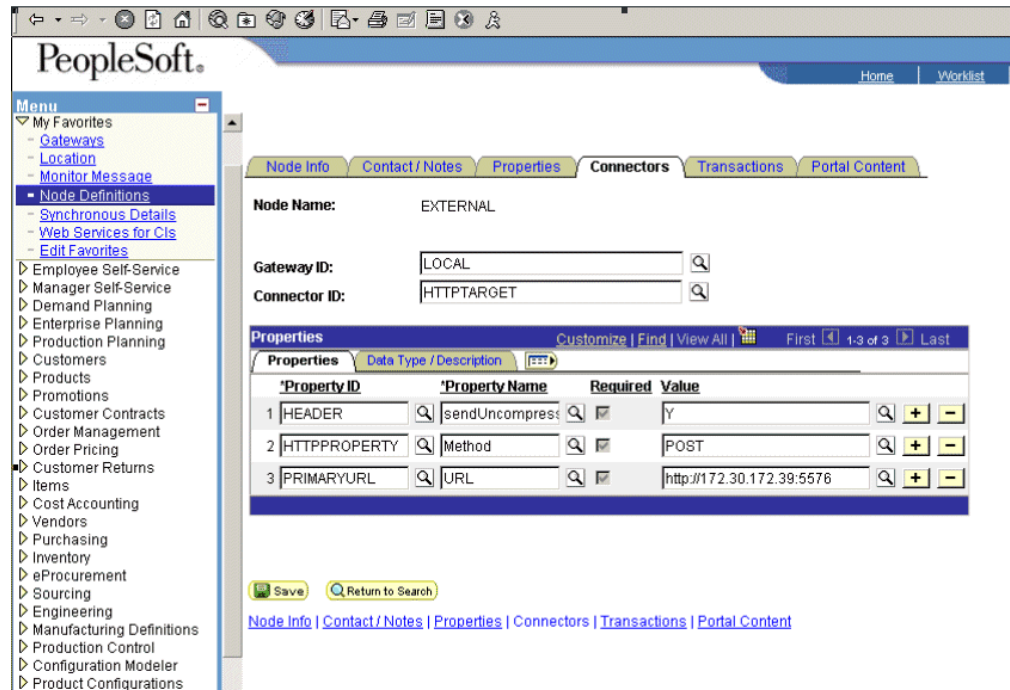
**Note:** This procedure uses a node called EXTERNAL. For more information about creating and using nodes, see [Appendix D, "Using PeopleSoft Integration Broker"](#) or your PeopleSoft documentation.

---

Perform the following steps:

- a. From the **Node Type** list, select **External**.
- b. From the **Routing Type** list, select **Implicit**.

**Figure B–9 PeopleSoft Integration Broker Connectors Tab**



3. Select the **Connectors** tab, as shown in [Figure B–9](#).

Perform the following steps:

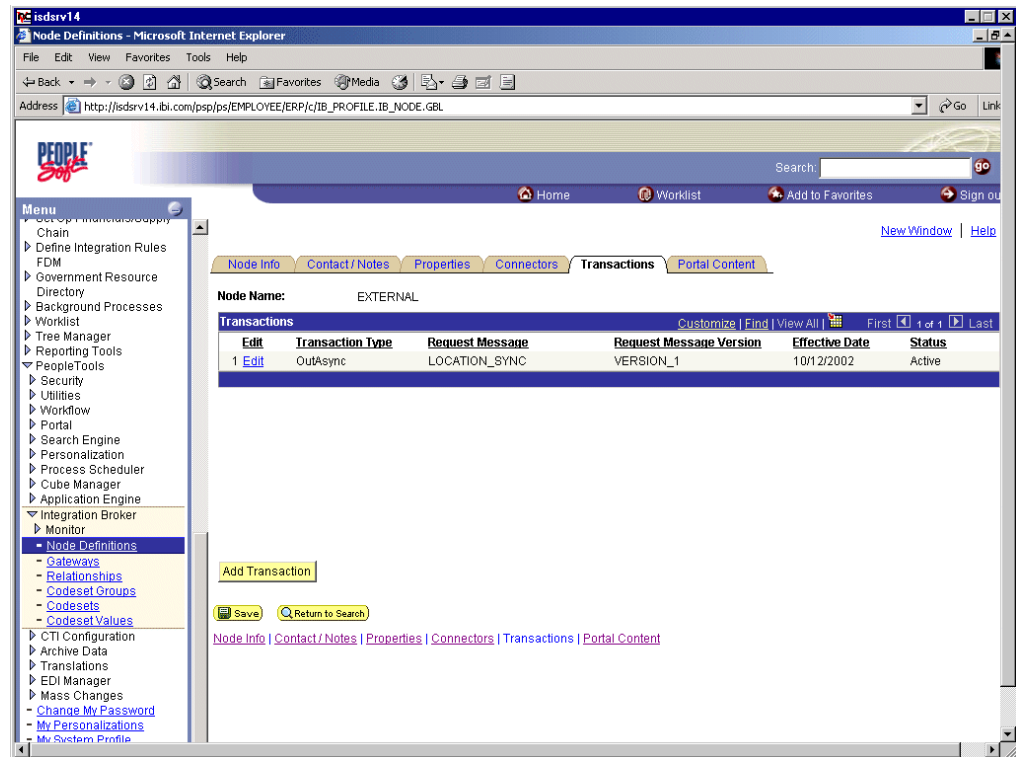
- a. Change the Connector ID to **HTTPTARGET**.
- b. Enter a value for each property based on the following:

Property ID	Property Name	Value
HEADER	sendUncompressed	Y
HTTPPROPERTY	Method	POST
PRIMARYURL	URL	URL and the port of the HTTP listener

**Note:** For complex business situations you can configure multiple nodes and multiple listeners.

4. Click **Save**.
5. If you are warned that you are changing the Connector, click **OK**.
6. Select the **Transactions** tab, as shown in [Figure B–10](#).

Figure B–10 PeopleSoft Integration Broker Transactions Tab

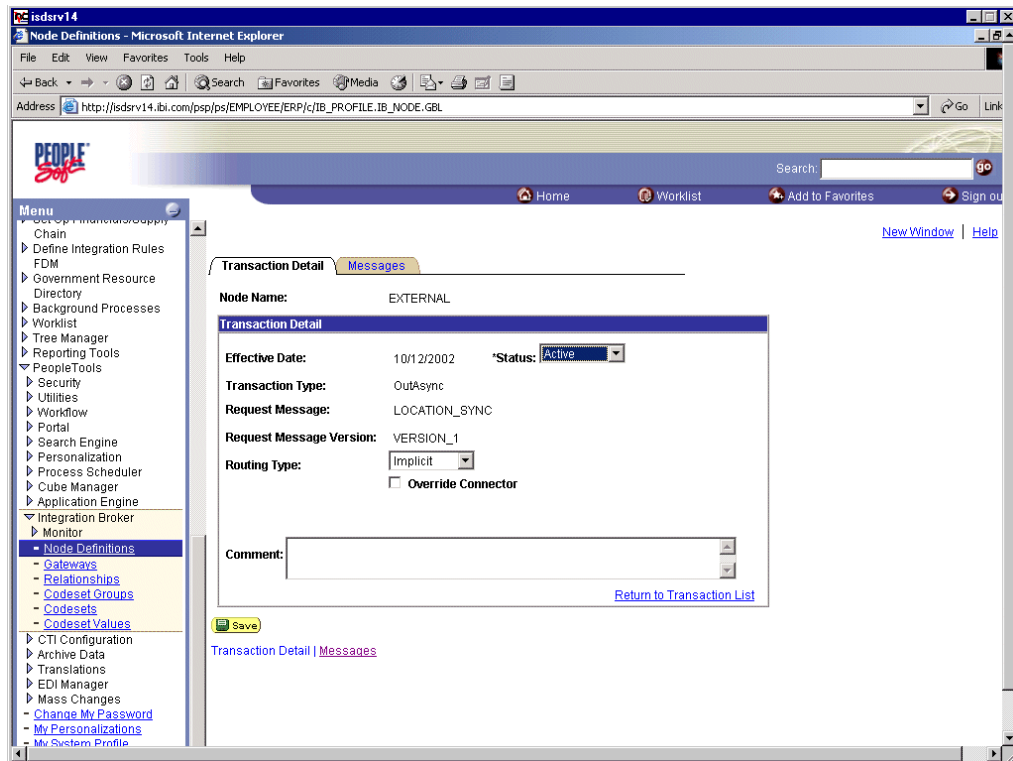


7. If there are no transactions, click **Add Transaction**.

In this procedure, the node is already configured with the LOCATION\_SYNC message.

The Transaction Detail tab appears, as shown in Figure B–11.

**Figure B–11 Transaction Detail Tab**



You can add the message with which you are working.

- a. Verify that the **Routing Type** is **Implicit**.
  - b. Click **Save**.
8. Return to the Transaction List. Perform the following steps:
- a. If there are other transactions, edit them.
  - b. Set the status to **Inactive**.

Inactive status is for initial testing only. After you test your configuration, you may change the status to Active and have as many nodes and transactions as required to satisfy your business requirements.

9. Click **Save** on the Transaction List.

You can now send XML messages to your PeopleSoft XML listener.

## B.4 Configuring the TCP/IP Handler for PeopleSoft 8.1

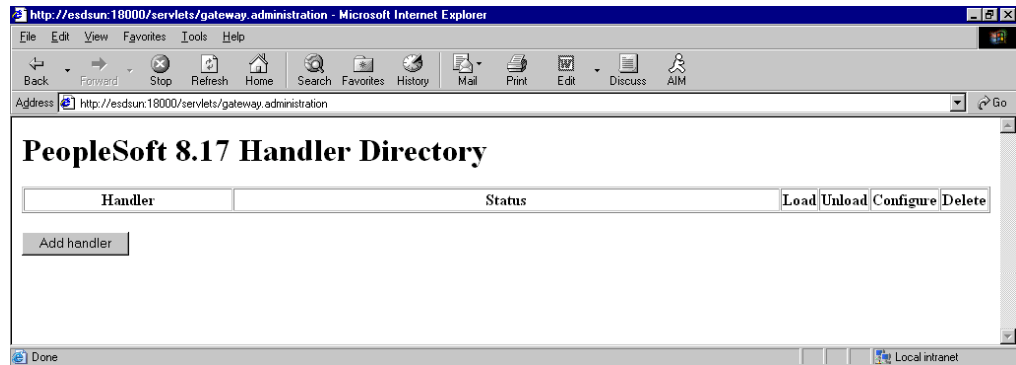
The following procedure assumes that your Application Messaging environment is properly configured and tested. For more information, see [Appendix D, "Using PeopleSoft Integration Broker"](#).

To configure the TCP/IP Handler for PeopleSoft 8.1 to send messages to Oracle WebLogic Server:

1. In a Web browser, launch the **PeopleSoft 8.1 Gateway Configuration** servlet interface.
2. If the Simple File Handler is currently loaded, unload and delete it before proceeding.

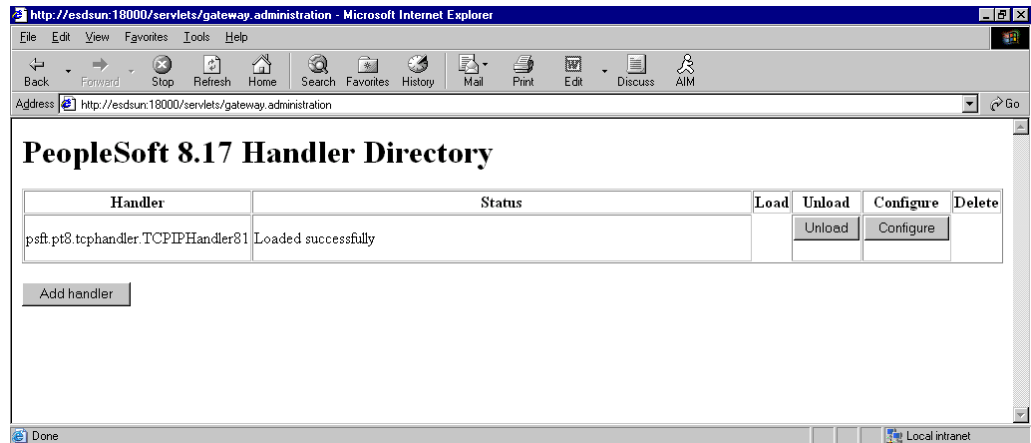
You must see an empty Handler directory, as shown in [Figure B-12](#).

**Figure B-12 Empty Handler Directory**



3. Click **Add handler**, as shown in [Figure B-13](#).

**Figure B-13 Loaded Add Handler Directory**



Perform the following steps:

a. Enter the full path of the TCP/IP Handler for PeopleSoft 8.1:

psft.pt8.tcphandler.TCPIPHandler81

This value is case-sensitive and must be entered exactly as shown.

b. Click **Save**.

4. Click **Load**.

The PeopleSoft Handler Directory window is displayed.

5. Click **Configure**.

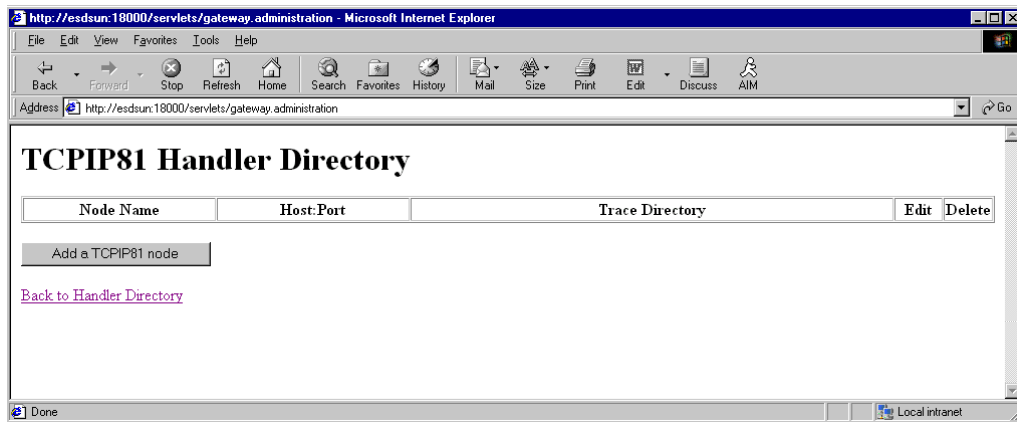
The TCPIP81 Handler Directory window is displayed.

6. Click **Add a TCPIP81 node**.

**Note:** The screens illustrating this procedure show a node named EXTERNAL. For more information about creating and using nodes, see [Appendix D, "Using PeopleSoft Integration Broker"](#) or your PeopleSoft documentation.

The Add TCPIP81 Handler window is displayed, as shown in [Figure B-14](#).

**Figure B-14 Add TCPIP81 Handler Window**



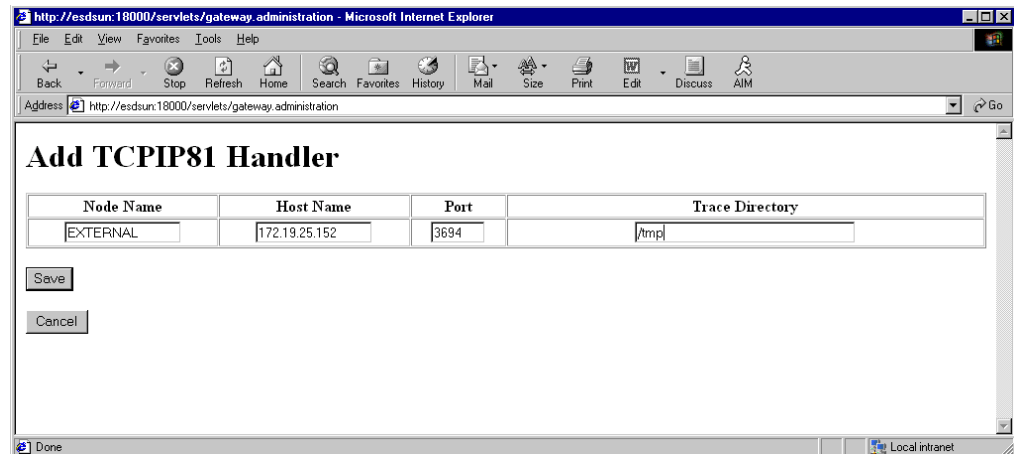
7. Enter the requested values based on the information in the following table.

Field	Value Example	Description
Node Name	EXTERNAL	The name of the TCP/IP node.
Host Name	172.19.25.152	The system on which your PeopleSoft XML listener is listening for incoming messages.
Port	3694	The port on which your PeopleSoft XML listener is listening for incoming messages.
Trace Directory	/tmp	The directory where a trace file is created when errors occur in message delivery.

The system does not validate your entries.

8. Click **Save**.

The TCPIP81 Handler Directory window is displayed, as shown in [Figure B-15](#).

**Figure B–15 TCPIP81 Handler Directory Window**

9. For your changes to take effect, click **Back to Handler Directory** to return to the PeopleSoft 8.1 Handler Directory window.
10. Click **Unload and re-Load TCPIPHandler81**.

You can now send messages from PeopleSoft to Oracle Application Adapter for PeopleSoft.

## B.5 Configuring the HTTP Handler for PeopleSoft 8.1

After the TCP/IP Handler is configured, perform the following steps to configure the HTTP Handler.

1. In a Web browser, launch the **PeopleSoft 8.1 Gateway Configuration** servlet interface.
2. Click **Add handler**.
3. Enter the full path of the HTTP Handler for PeopleSoft 8.1:

```
psft.pt8.tcphandler.HTTPHandler81
```

This value is case-sensitive and must be entered exactly as shown.

4. Click **Save**.
5. Load the TCP/IP Handler to configure the node.
6. Provide the port number configured for the HTTP channel and then save the Handler.

For more information, see step 7 in [Section B.4, "Configuring the TCP/IP Handler for PeopleSoft 8.1"](#) on page B-12.

7. Unload the TCP/IP Handler.
8. Reload the HTTP Handler and trigger the events.

## B.6 Testing Your PeopleSoft Configuration

PeopleSoft 8.1 and 8.4 provide a ping node mechanism for testing your configuration. The mechanism functions identically in both versions.

Test your configuration to ensure that:

- Oracle WebLogic Server is up and running.
- The server name and port number for PeopleSoft and Oracle WebLogic Server match.
- The default page for HTTP exists.

To test a PeopleSoft configuration:

1. In a Web browser, open your PeopleSoft application.
2. Navigate to the message monitoring menu.

For PeopleSoft 8.4:

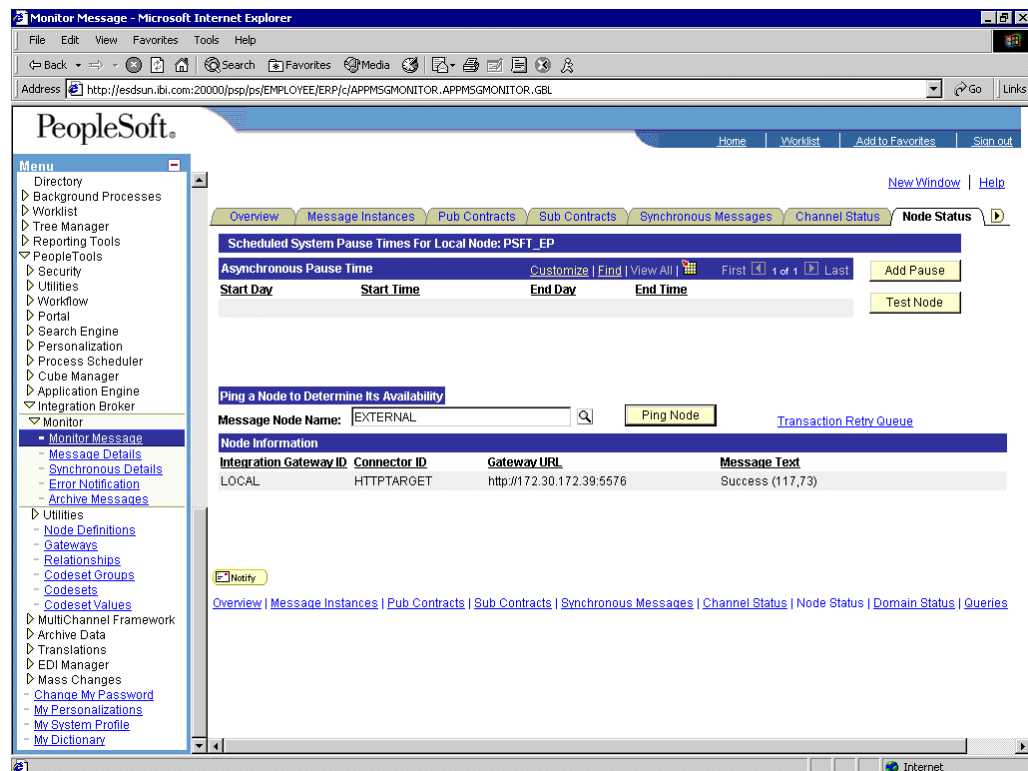
- a. In the menu pane, expand **PeopleTools, Integration Broker, and Monitor**.
- b. Select **Monitor Message**.

For PeopleSoft 8.1:

- a. In the menu pane, expand **Home, PeopleTools, Application Message Monitor, and Use**.
- b. Select **Application Message Monitor**.

3. Click the **Node Status** tab, as shown in [Figure B-16](#).

**Figure B-16 Node Status Tab**



Perform the following steps:

- a. From the Message Node Name list, select your node.
- b. Click **Ping Node**.

If you properly configured both PeopleSoft and Oracle WebLogic Server, you receive a Success message.



An error indicates a configuration problem. For more information, see the Integration Broker error log.



---

---

## Using Component Interfaces

This appendix describes how to create new and modify existing component interfaces for use with Oracle Application Adapter for PeopleSoft. It also describes how to apply security to those component interfaces and how to test them.

You can:

- Use component interfaces supplied by PeopleSoft with your application.
- Component interfaces also are known as Enterprise Integration Points (EIP).
- Modify an existing component interface.
- Create a new component interface.

Before using your component interface you must apply security to it and test it.

After securing and testing a component interface, you must generate its API. For more information, see [Appendix A, "Generating Component Interface APIs"](#).

---

---

**Note:** This section is intended as a helpful supplement; it is not a substitute for PeopleSoft documentation. For complete and up-to-date information about PeopleSoft component interfaces, see the PeopleSoft Online Library for your PeopleSoft system.

---

---

This appendix contains the following topics:

- [Section C.1, "Working With Properties"](#)
- [Section C.2, "Securing a Component Interface"](#)
- [Section C.3, "Testing a Component Interface"](#)

### C.1 Working With Properties

You create component interfaces using the PeopleSoft Application Designer. For more information about Application Designer, see your PeopleSoft documentation.

You can add properties from the records in the component view. You can delete a property in the component interface that you do not want to expose. You can rename properties by clicking the property and then clicking again until you can enter a new name. If you rename a property, it can be referenced in the component interface only by the new name, not by the underlying component name.

Properties may have various icons adjacent to them. For example, EMPLID has an icon indicating that it is a key field from the underlying record. NAME has an icon

indicating that it is an alternate key field from the underlying record. For a complete list of property icons, see the PeopleBooks documentation.

This section consists of the following topics:

- [Section C.1.1, "Creating a New Component Interface"](#)
- [Section C.1.2, "Viewing or Changing Available Methods"](#)

## C.1.1 Creating a New Component Interface

This section contains the following topic:

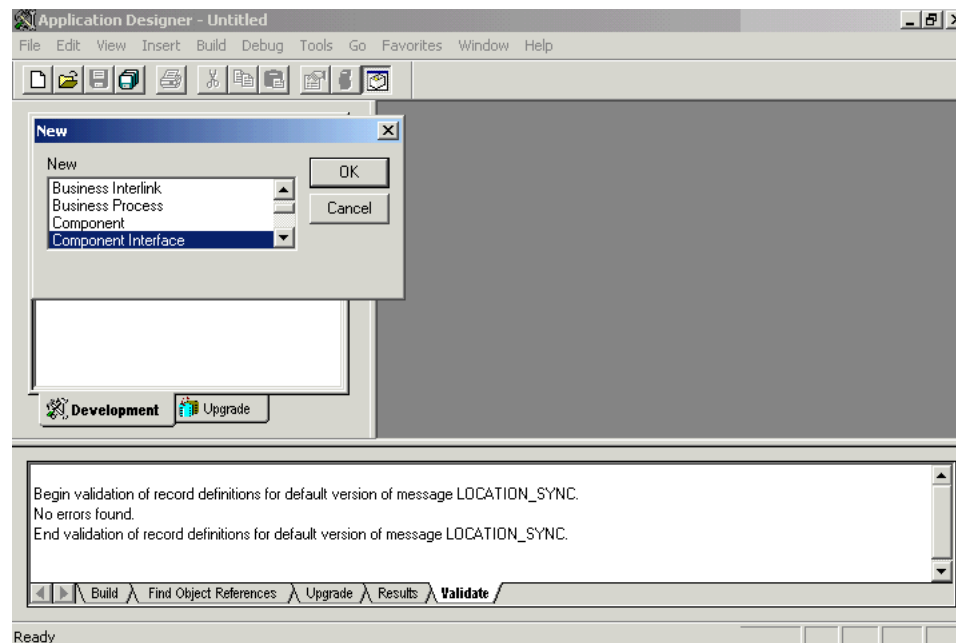
- [Section C.1.1.1, "Standard Methods"](#)

To create a component interface:

1. Open the PeopleSoft Application Designer.
2. Select **New** from the **File** menu.

The New dialog is displayed, as shown in [Figure C-1](#).

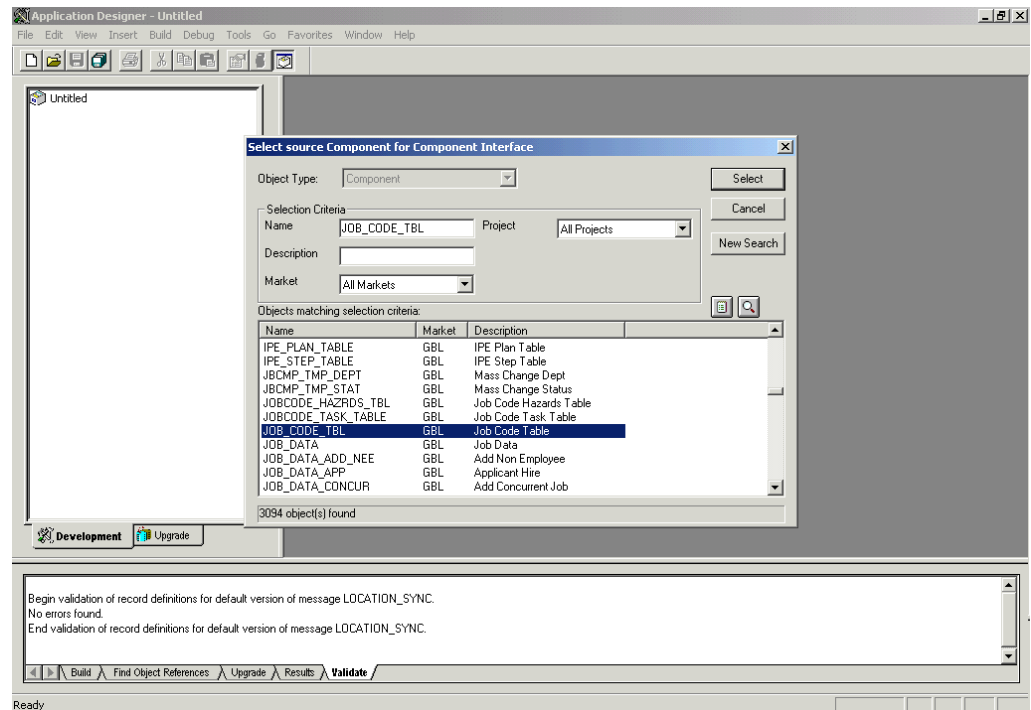
**Figure C-1** New Dialog



Perform the following steps:

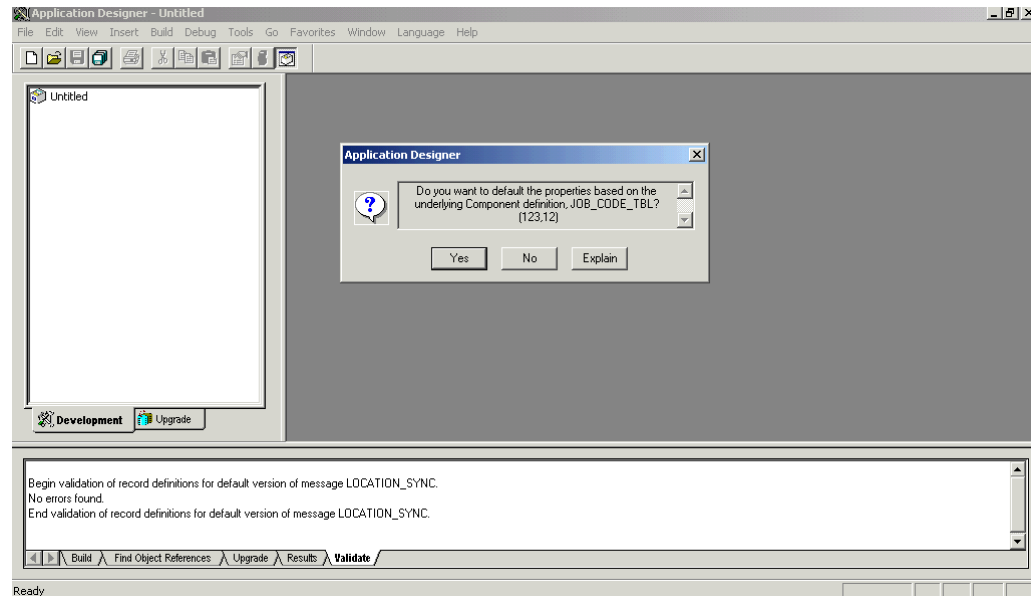
- a. Select **Component Interface**.
- b. Click **OK**.

The Select Source Component for Component Interface dialog is displayed, as shown in [Figure C-2](#).

**Figure C-2** Select Source Component for Component Interface Dialog

3. Highlight the component to use as a basis for the component interface and click **Select**.

The Application Designer dialog is displayed, as shown in [Figure C-3](#).

**Figure C-3** Application Designer Dialog


---

**Note:** If the component interface is large, expose the component properties manually.

---

4. To create the component interface without displaying properties and to expose component properties manually, click **No**.

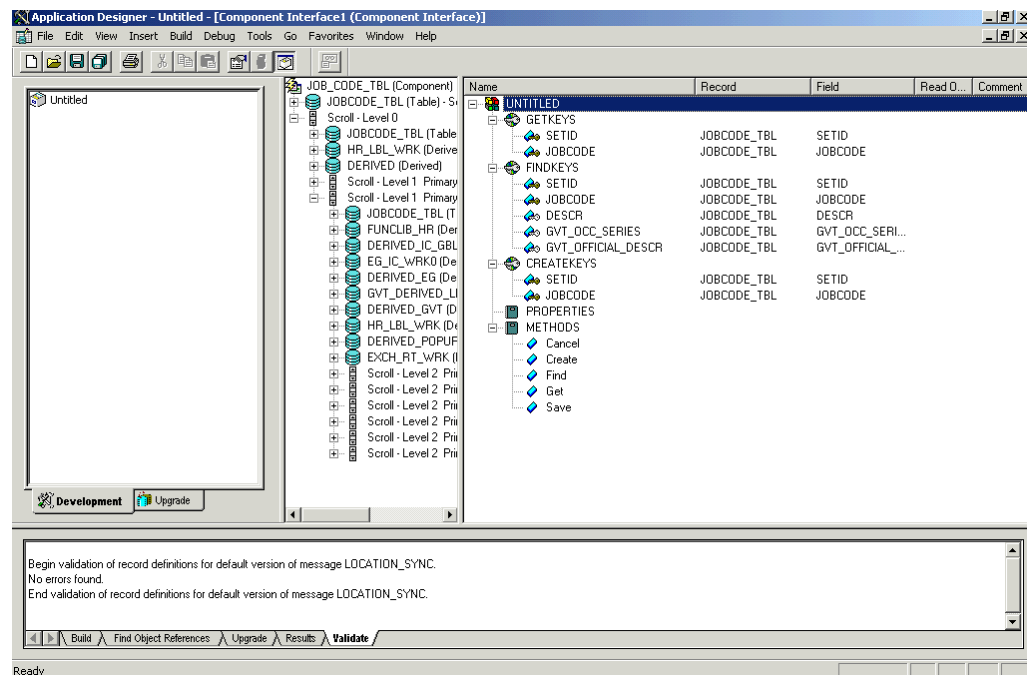
Perform the following steps:

- a. Drag the relevant fields from the left pane to the right pane.
- b. To select various functions to perform, right-click either the right or left pane, depending on which pane is active, as shown in [Figure C-4](#).

For a complete list of functions, see the PeopleBooks documentation.

5. To create the component interface and display the properties of the underlying component interface, click **Yes**.

**Figure C-4 Application Designer Component Interface**



### C.1.1.1 Standard Methods

The standard methods for the component interface are:

- Create
- Find
- Get
- Save

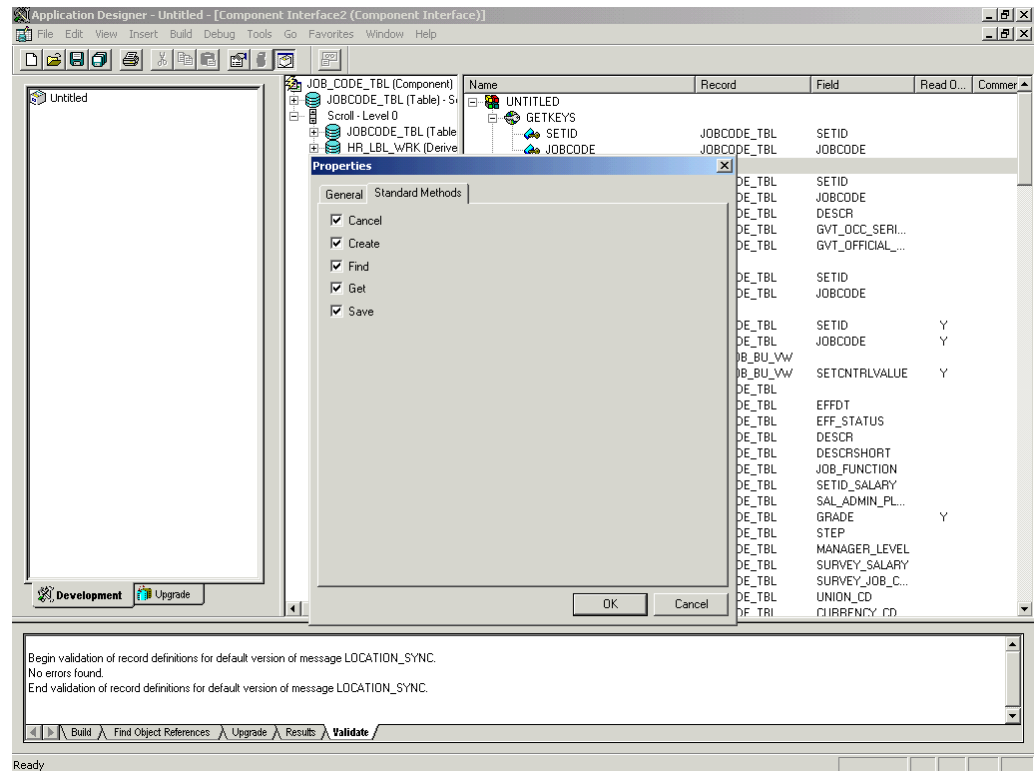
Only those methods in the underlying component are available. For example, if the underlying component does not contain Add capabilities, Create is not available.

## C.1.2 Viewing or Changing Available Methods

To view or change available methods:

1. Display the Component Interface Properties dialog, as shown in [Figure C-5](#).

Figure C-5 Component Interface Properties Dialog



2. Click the **Standard Methods** tab.
3. Select the desired methods.

## C.2 Securing a Component Interface

You must set up security for the component interface before you can begin testing.

This section consists of the following topics:

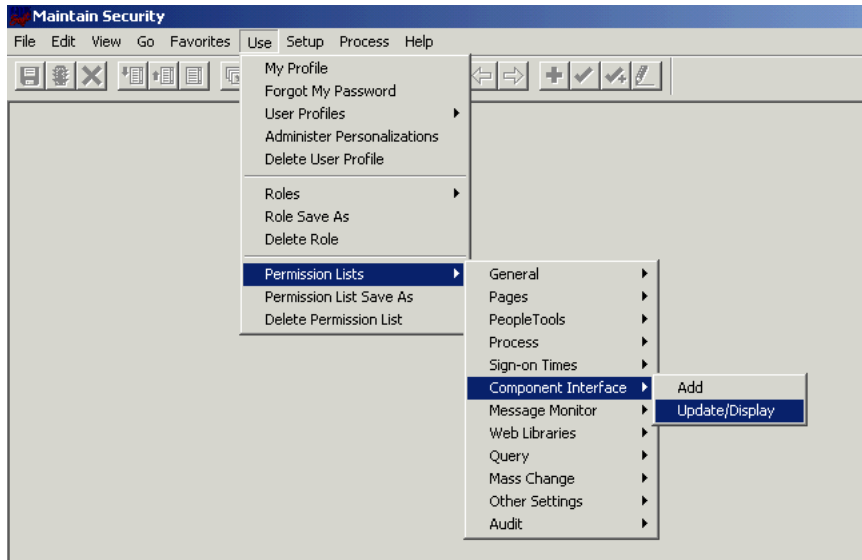
- [Section C.2.1, "Configuring Component Interface Security for PeopleSoft Version 8.1x"](#)
- [Section C.2.2, "Configuring Component Interface Security for PeopleSoft Version 8.4 or Higher"](#)

### C.2.1 Configuring Component Interface Security for PeopleSoft Version 8.1x

The following procedure describes how to configure component interface security for PeopleSoft Version 8.1 in 2- and 3-tier mode.

To configure component interface security:

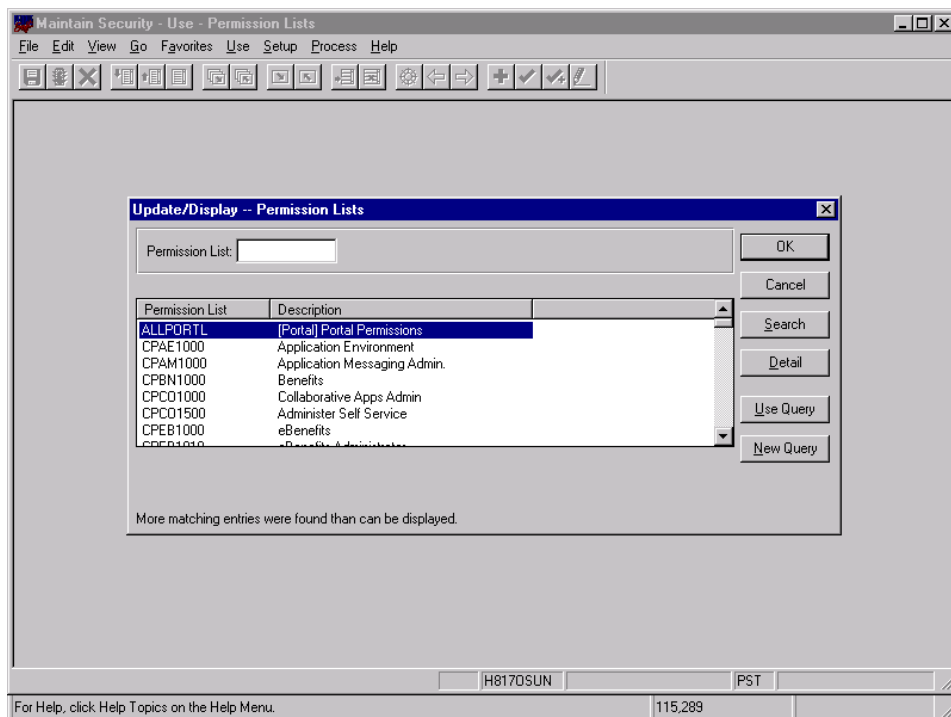
**Figure C-6 Component Interface Security Options**



1. From the **Use** menu, select **Permission Lists**, **Component Interface**, and then click **Update/Display**, as shown in [Figure C-6](#).

The Permission Lists dialog is displayed, as shown in [Figure C-7](#).

**Figure C-7 Permission Lists Dialog**



Before Security can be set, you must identify the permission lists.

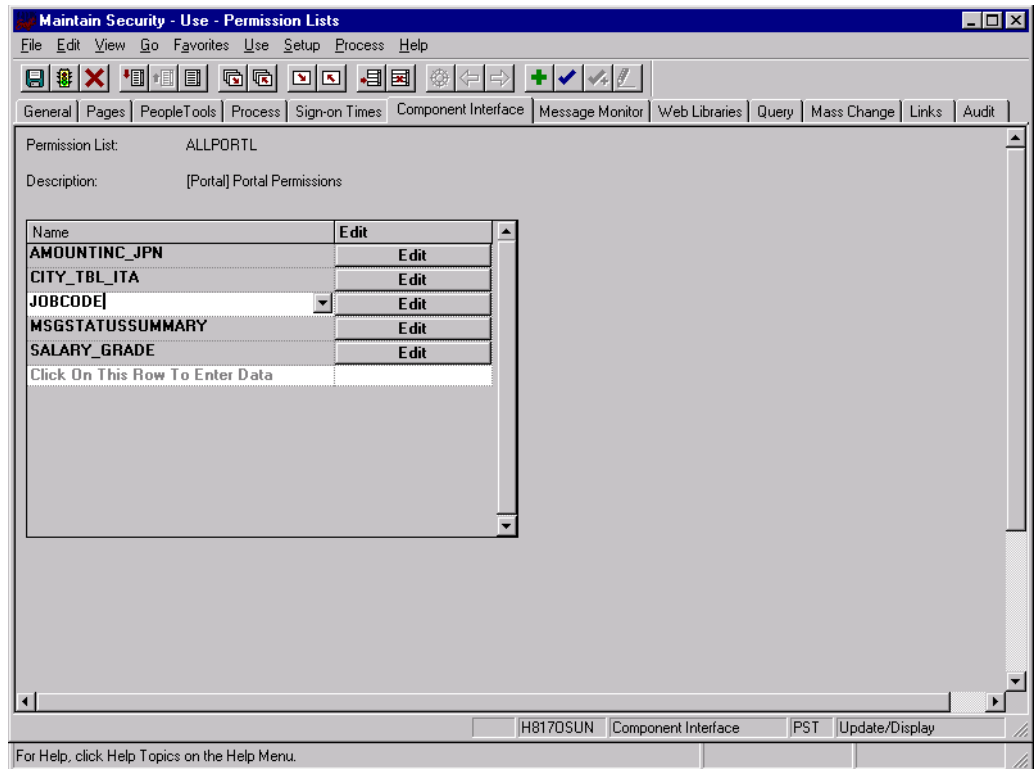
2. Select the relevant permission list and click **OK**.

For more information on permission lists, see the PeopleBooks documentation.

The Permissions Lists pane is displayed, as shown in [Figure C-8](#).



Figure C-8 Permission Lists Pane

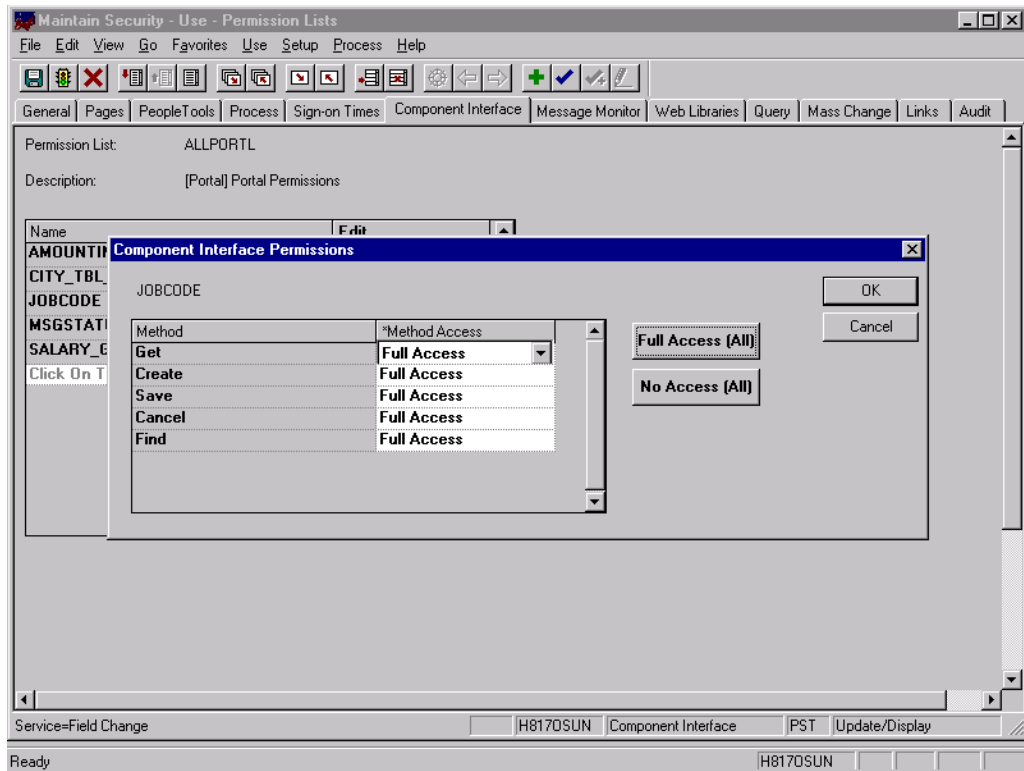


3. Insert the new component interface that you created.
4. Click **Edit**.

When you select the component interface, all available methods appear, including user-defined methods. You can specify whether this particular Permission List must have full or partial access.

The ALLPORTL Permission List has full access to all methods, as shown in [Figure C-9](#).

**Figure C-9 ALLPORTL Permission List**



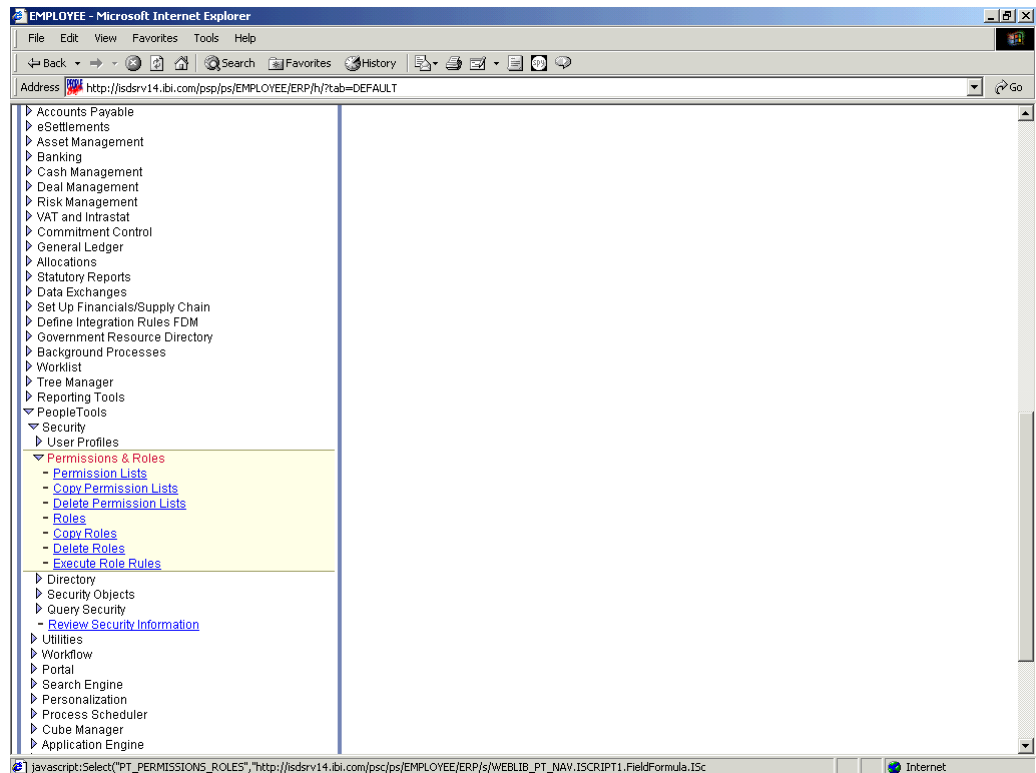
5. Select the desired level of access.
6. Click OK.

### C.2.2 Configuring Component Interface Security for PeopleSoft Version 8.4 or Higher

The following procedure describes how to configure component interface security for PeopleSoft Version 8.4 or higher.

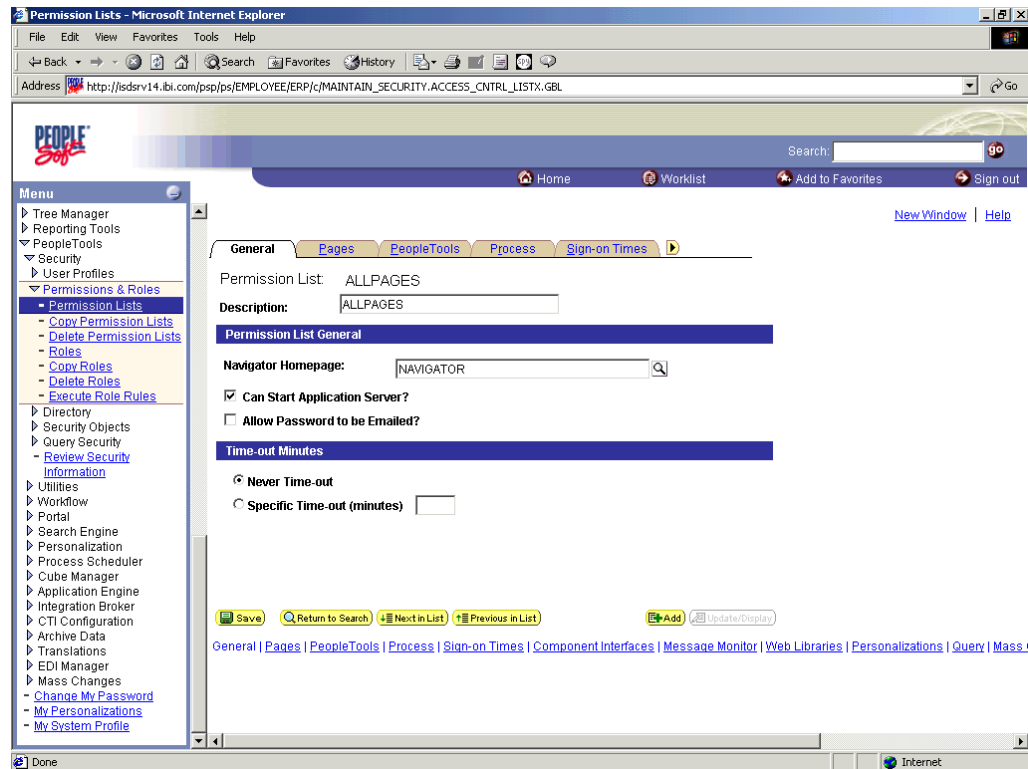
To configure interface security:

Figure C-10 Configure Component Interface Security



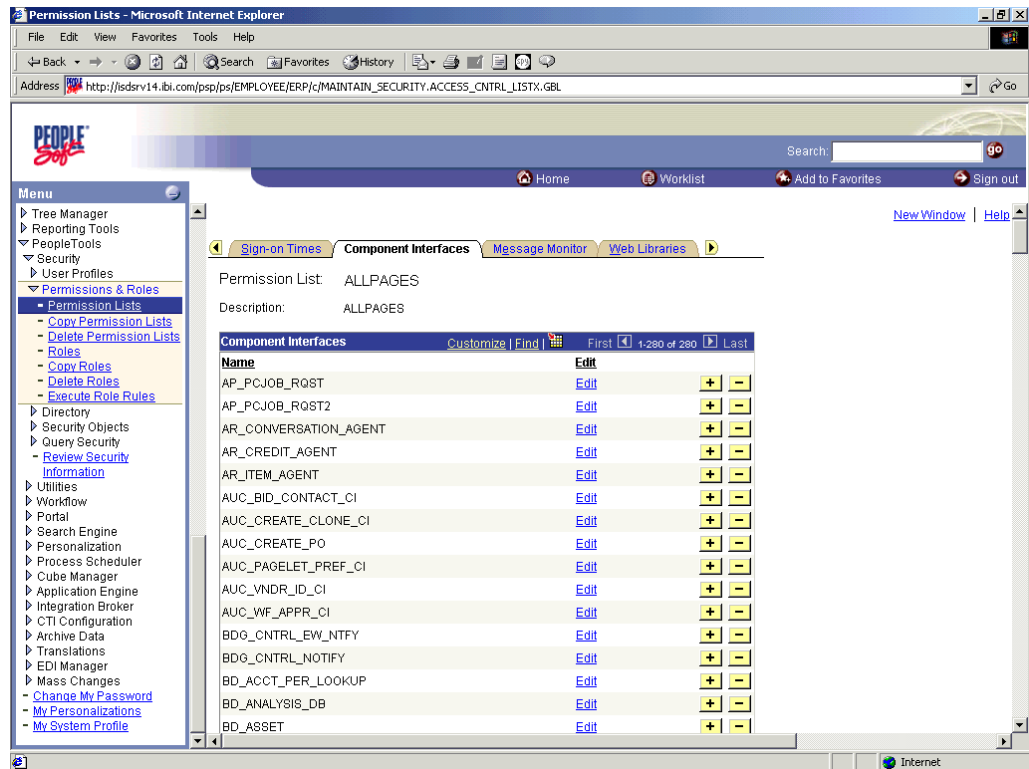
1. Expand **PeopleTools**, **Security**, **User Profiles**, and **Permissions & Roles** and then click **Permission Lists**, as shown in [Figure C-10](#).
2. Click **Search**.  
The Permission Lists Search pane is displayed.
3. Select the relevant permission list, as shown in [Figure C-11](#).

**Figure C–11 Permission List Pane**



4. Click the right arrow next to the **Sign-on Times** tab.  
The **Component Interfaces** tab appears, as shown in [Figure C–12](#).

Figure C-12 Component Interfaces Tab

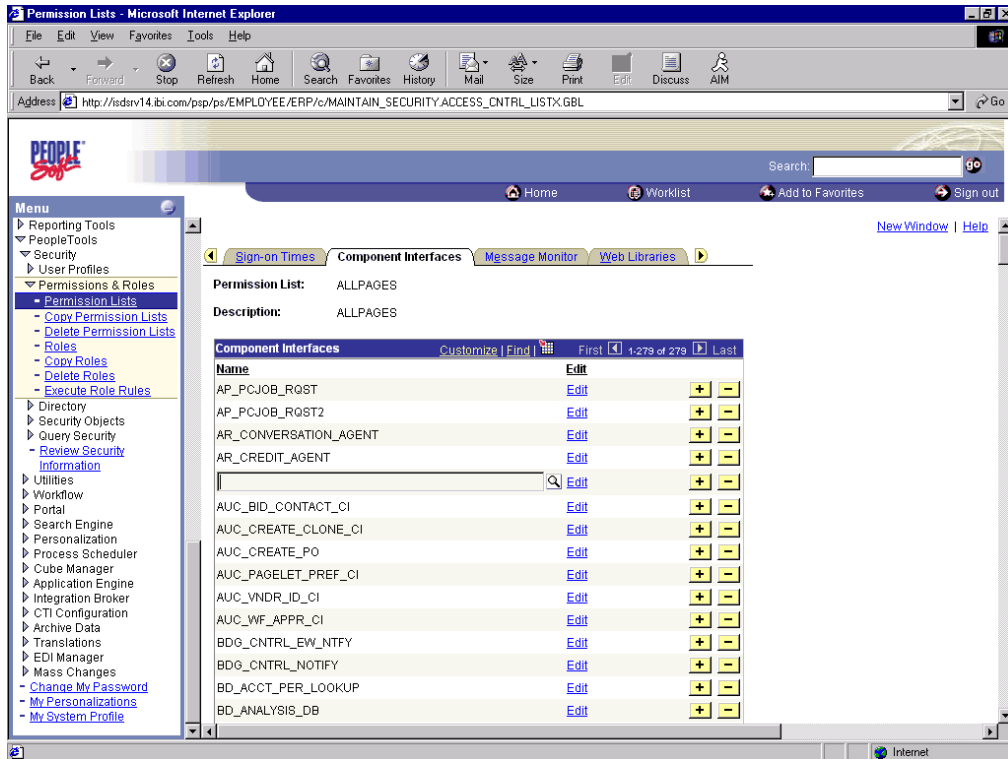


5. Click the **Component Interfaces** tab.

6. Click **+** to add a new row to the Component Interfaces list.

A field appears where you can enter the component interface name, as shown in [Figure C-13](#).

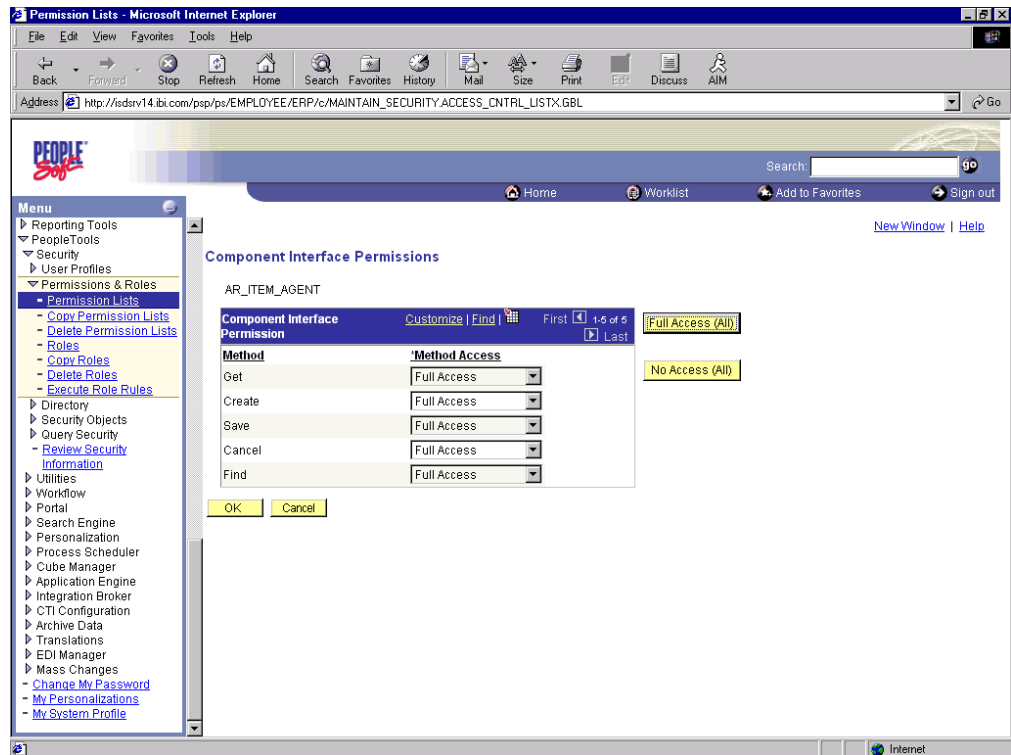
**Figure C-13 Component Interface Name Field**



7. Enter the component interface name and click **Edit**.

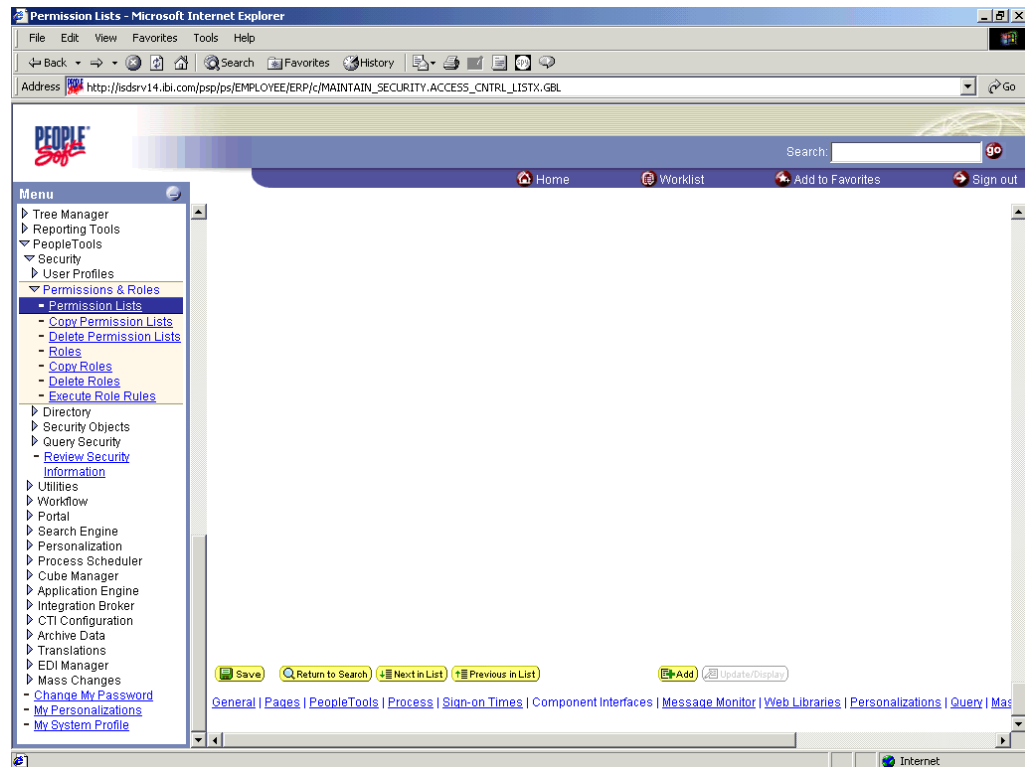
This example uses the component interface `AR_ITEM_AGENT`, as shown in [Figure C-14](#).

Figure C-14 Component Interface Permissions Window



8. From the lists, select the desired access level for each method.
9. Click OK.

The Permissions list window is displayed, as shown in Figure C-15.

**Figure C–15 Permissions List Window**

10. Scroll down in the right pane and click **Save**.

### C.3 Testing a Component Interface

Oracle Application Adapter for PeopleSoft uses PeopleSoft metadata and component interfaces; therefore, it can accommodate new or modified component interfaces. The adapter makes no assumptions about component interfaces except that they are logical and valid. Each component interface must be tested before being used as a source for the adapter.

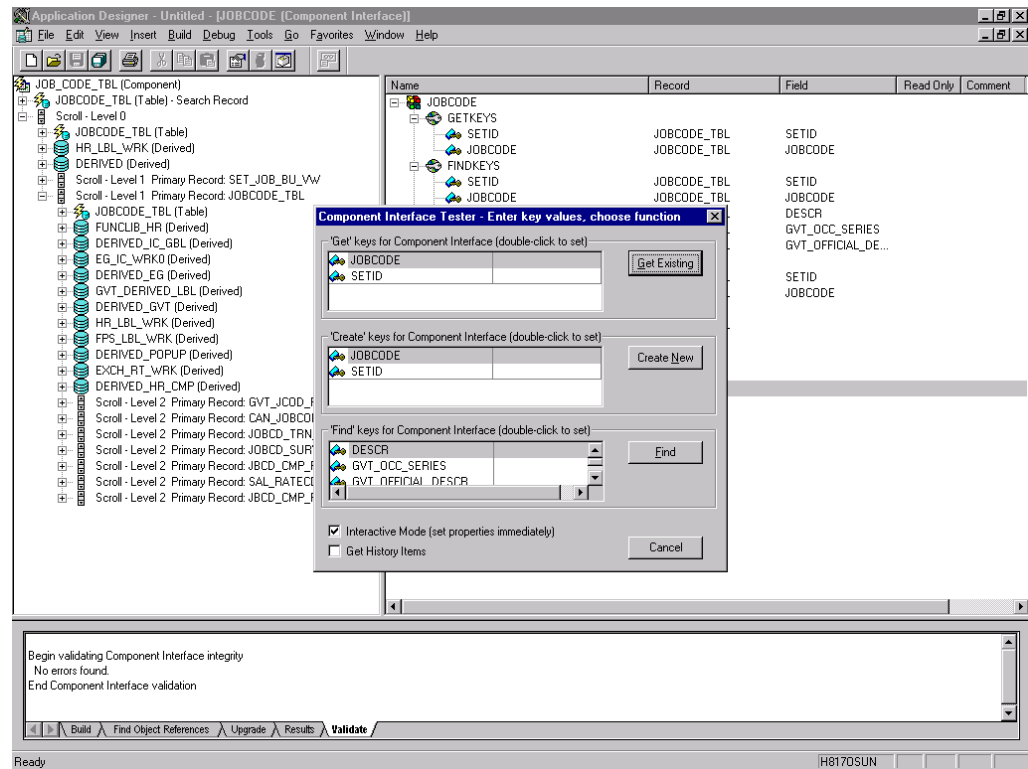
If changes are made to the underlying application by the user or by a PeopleSoft upgrade and the changes invalidate a component interface, the user must repair the invalid component interface before the adapter uses it.

To test a component interface:

1. In Application Designer, select **Test Component Interface** from the **Tools** menu.  
The Component Interface Tester dialog is displayed, as shown in [Figure C–16](#).



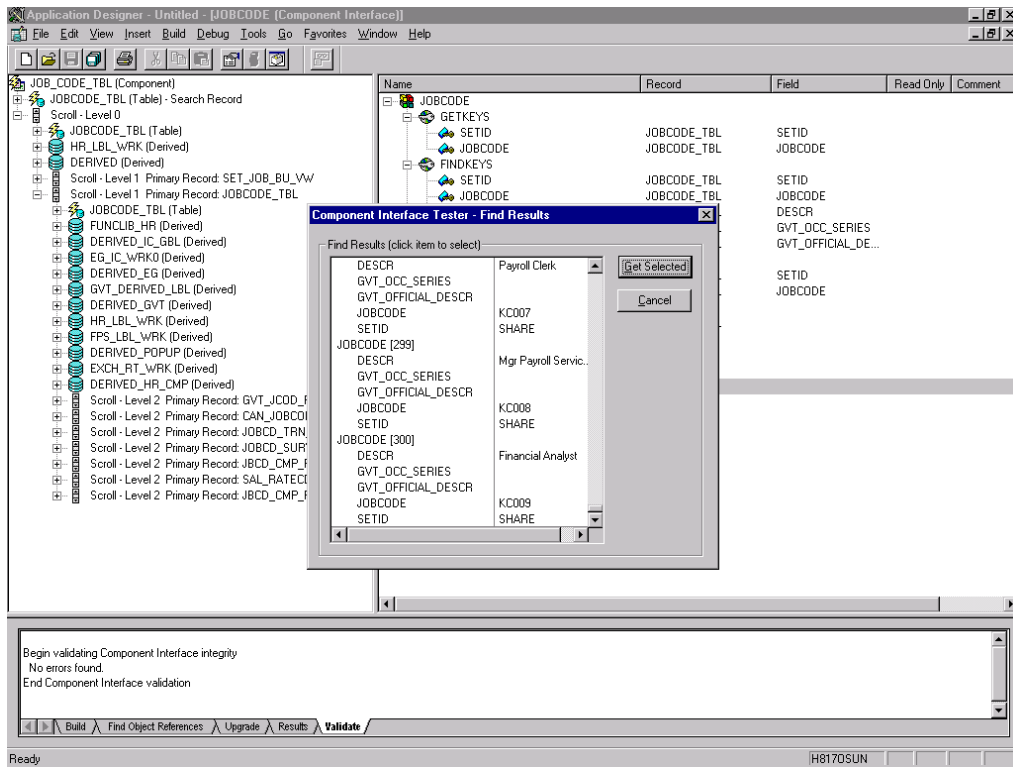
Figure C-16 Component Interface Tester Dialog



2. If required, click the Component Interface Tester dialog to bring it to the foreground.
3. To test the component interface, use one of the following methods.
  - To test the component interface using the Find method, click **Find**.

The Component Interface Tester - Find Results dialog displays all of the possible entries for the underlying component, as shown in Figure C-17. If there are more than 300 entries, a message appears.

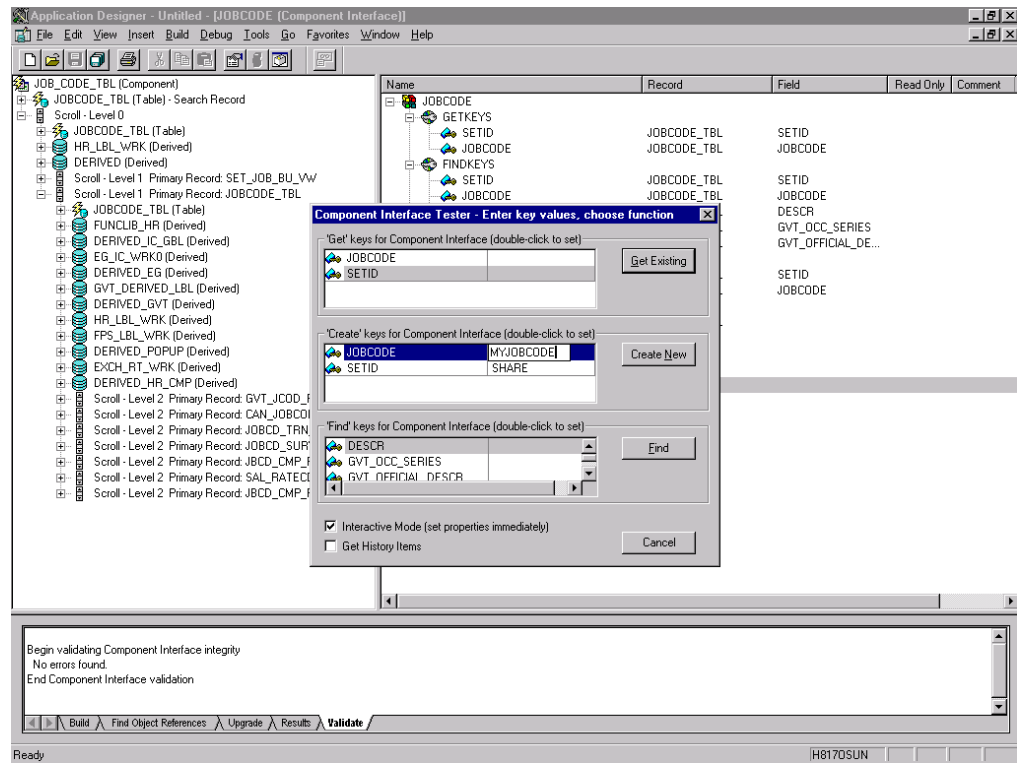
**Figure C-17 Component Interface Tester - Find Results Dialog**



- In the left pane of the Find Results dialog, select a field.
- To display the relevant data for that particular field, click **Get Selected**.  
The Find Results dialog is displayed, as shown in [Figure C-18](#).



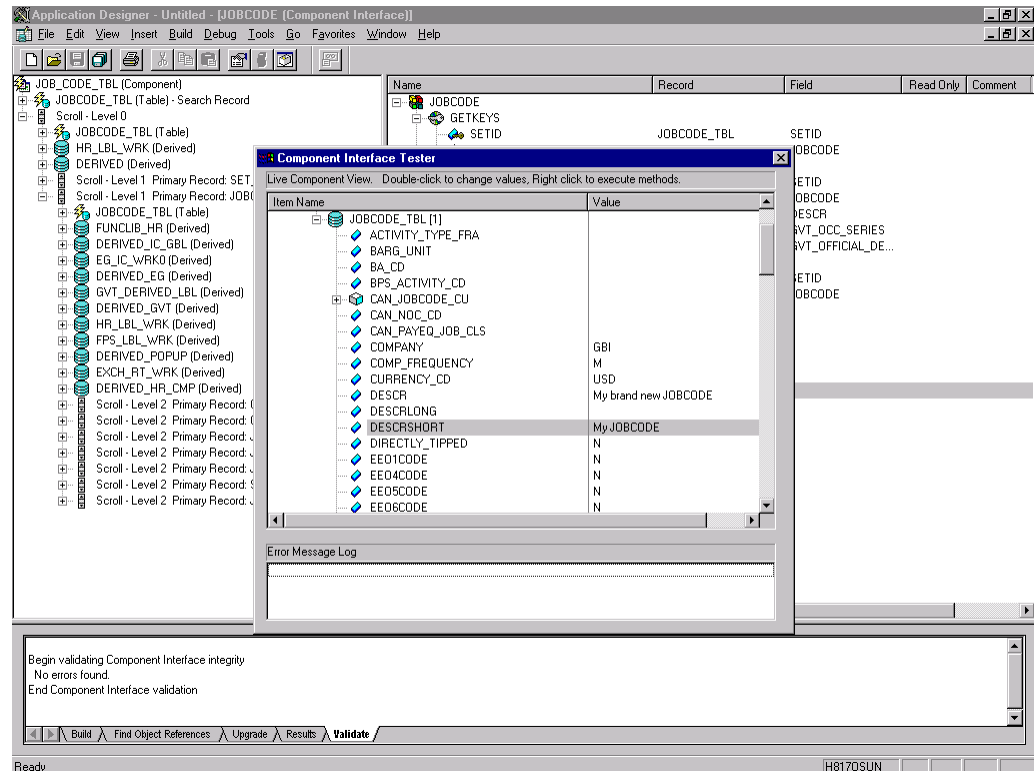
**Figure C-19 Component Interface Tester Dialog**



Perform the following steps:

- a. Enter all required key values.
- b. Click **Create New**.

When you enter valid values in Create keys, a pane showing the JOBCODE data is displayed after the Table name is expanded with default data in place, as shown in [Figure C-20](#).

Figure C-20 **JOBCODE Data**

You can change fields at this point. Changes are validated against the component's underlying business logic.

- c. After you finish making changes, right-click the top item in the pane.
5. To save your changes, click **Save**.

The keys used to create the record can be used with the Get method for viewing data. The data that was added can be viewed in the PeopleSoft Component as shown in the following example. The Effective Date is one of the default values.

You have finished testing the component interface. Before using the component interface, you must generate its API. For more information, see [Appendix A, "Generating Component Interface APIs"](#).



---

---

## Using PeopleSoft Integration Broker

This appendix describes how to configure and test PeopleSoft Integration Broker (release 8.4) and PeopleSoft Application Messaging (release 8.1) using a PeopleSoft-supplied File Output interface. In PeopleSoft release 8.1, the messaging architecture is called Application Messaging and includes Application Messaging Gateway. In release 8.4, the messaging architecture is called Integration Broker, which includes Integration Gateway. When discussing release-generic issues, this section uses release 8.4 terminology. When discussing release-specific issues, it uses release-specific terminology.

---

---

**Note:** This section is not a substitute for PeopleSoft documentation. For more complete and up-to-date information on PeopleSoft Messaging and Integration Broker, see the *PeopleSoft Online Library* for your PeopleSoft system.

---

---

This appendix contains the following sections:

- [Section D.1, "PeopleSoft Integration Broker"](#)
- [Section D.2, "Configuring Application Messaging in PeopleSoft Release 8.1"](#)
- [Section D.3, "Viewing the PeopleCode for a Message"](#)
- [Section D.4, "Testing the Integration Broker"](#)
- [Section D.5, "Using Outbound Synchronous Messages"](#)
- [Section D.6, "Using PeopleTools Version 8.48 and Higher to Generate Events"](#)

### D.1 PeopleSoft Integration Broker

PeopleSoft Integration Broker provides a mechanism for communicating with the outside world using XML files. Communication can take place between different PeopleSoft applications or between PeopleSoft and third-party systems.

To subscribe to data, third-party applications can accept and process XML messages posted by PeopleSoft using the available PeopleSoft connectors or by adding a custom built connector to the Integration Gateway. This topic primarily covers publishing outbound asynchronous messages from a PeopleSoft system to a third-party application using the delivered File Output connector. For information on outbound synchronous messages, see "[Using Outbound Synchronous Messages](#)" on page D-24.

To send a message, you must properly configure various internal structures and processes. The following descriptions are generally release-generic. Detailed differences between releases 8.1 and 8.4 are discussed in other topics.

- **Message**

A Message is a container for the data that goes into the XML. It contains basic structural information, such as records and fields. The Message must be in an Active status to send the XML file
- **Message Channel**

The Message Channel is a mechanism for structuring records into logical groupings. Each Message can belong to only one Message Channel. The Message Channel must be in an Active (Run) status for the Message to be delivered.

In release 8.1, the Message Channel also provides preliminary routing instructions; you can specify which Message Nodes handle the message. Each Message Channel can route messages to multiple Message Nodes
- **Message Node**

Message node functionality changed from 8.1 to 8.4:

In release 8.1, the primary function of the Message Node is to specify which Gateway receives the messages.

In release 8.4, much of the "intelligence" that was built into the Message Channel moved to the Message Node. This provides additional flexibility over release 8.1. You can specify which messages the Message Node can handle. In addition, the Gateway Connector is bound to the Message Node. Each Message Node can route messages to only one Connector.
- **Integration Gateway**

The Integration Gateway is a program that runs on the PeopleSoft Web Server. It is the physical hub between PeopleSoft and the third-party system.
- **Target Connector/Handler**

Connectors are Java programs that run under the control of the Integration Gateway and control the final output destination of the XML file. PeopleSoft release 8.4 comes with several connectors including HTTP, FTP, SMTP, JMS, POP3, and a `Simple File` connector that places the file in a directory on the Web Server. This section discusses the Simple File connector.
- **PeopleCode**

PeopleCode is the programming tool provided with PeopleTools that enables you to create complex application functionality. A message can only be initiated using specific PeopleCode instructions. This code is typically triggered by an application event, such as creating a new database entry through an online panel or through a batch job.

Most of the examples in this section use the `LOCATION_SYNC` message, which is a PeopleSoft Enterprise Integration Point (EIP) and is supplied with most PeopleSoft applications. If `LOCATION_SYNC` is not part of your package, you may use any supplied message.

This section consists of the following topic:

- [Section D.1.1, "Configuring Integration Broker in PeopleSoft 8.4"](#)

## D.1.1 Configuring Integration Broker in PeopleSoft 8.4

You can configure PeopleSoft 8.4 to send an asynchronous outbound message to the File Output connector.



This section consists of the following topics:

- Section D.1.1.1, "Ensuring the Message Is Active and Is Routed Correctly"
- Section D.1.1.2, "Configuring the IntegrationGateway.properties File"
- Section D.1.1.3, "Configuring the Integration Gateway and the File output Connector"
- Section D.1.1.4, "Creating and Configuring a New Gateway Node"

To configure application messaging in PeopleSoft 8.4:

1. Ensure that the message is active and is routed to the proper Message Channel.
2. Configure the Message Channel.
3. Configure the IntegrationGateway.properties file to communicate with your PeopleSoft 8.4 application.
4. Configure the Integration Gateway and File Output connector.
5. Create and configure a new Gateway node.

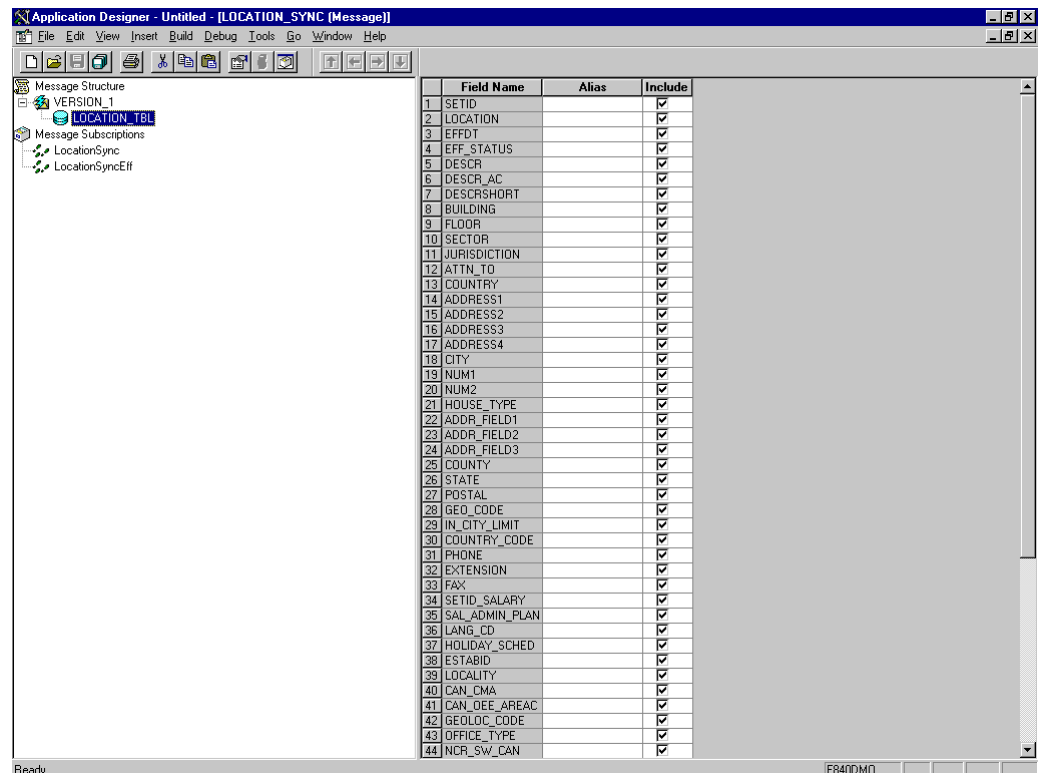
These tasks are described in detail in the following procedures.

### D.1.1.1 Ensuring the Message Is Active and Is Routed Correctly

To ensure that the message is active and is routed to the proper Message Channel:

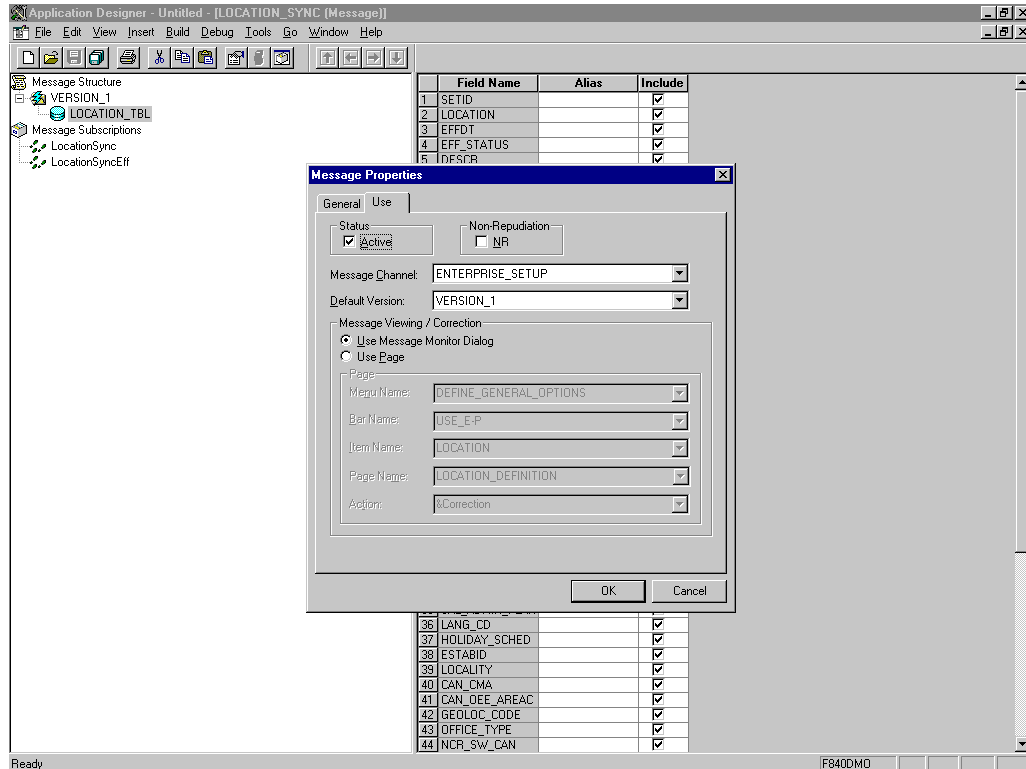
1. Open Application Designer.
2. On the **File** menu, point to **Open**, click **Message**, and then open the **LOCATION\_SYNC** message, as shown in [Figure D-1](#).

**Figure D-1** LOCATION\_SYNC Message



3. To view the fields that are included in the Message, highlight **LOCATION\_TBL**.
4. Right-click **LOCATION\_TBL** and select **Properties**.  
The Message Properties dialog is displayed, as shown in [Figure D-2](#).

**Figure D-2 Message Properties Dialog**



5. Select the **Use** tab.
6. Ensure the **Active** check box is selected.  
The message is routed to the Message Channel, **ENTERPRISE\_SETUP**, and the default message version is **VERSION\_1** (messages can have multiple versions).
7. Click **OK** and then save the message.  
You have finished ensuring that the message is active and is routed correctly.

### D.1.1.2 Configuring the IntegrationGateway.properties File

To configure the `IntegrationGateway.properties` file:

1. Open the `IntegrationGateway.properties` file using the editor of your choice.
2. Find the section of the file that specifies the JOLT connect string setting for the default application server. This is usually near line 75, and looks similar to the following:

```
## JOLT connect string setting for optional Default Application Server. Do NOT
specify a NODENAME.
#
# Example:
#ig.isc.serverURL=//MYSERVER:9000
#ig.isc.userid=MYUSERID
```

```
#ig.isc.password=MYPASSWORD
#ig.isc.toolsRel=8.40
```

3. Uncomment (or copy and uncomment) the four lines that specify the connection.
4. Enter the appropriate information.

In the following example, the tools release is 8.40.09:

```
ig.isc.serverURL>//isdsrv14:9000
ig.isc.userid=VP1
ig.isc.password=VP1
ig.isc.toolsRel=8.40.09
```

The PeopleSoft tools release must be precise to the last decimal.

---



---

**Note:** With release 8.42, the password must be stored in an encrypted format. PeopleSoft provides a script called `PSCipher.bat` (`PSCipher.sh` on UNIX) to accomplish encryption. Typically, this script is located in the path of the `IntegrationGateway.properties` file. Follow the instructions supplied by PeopleSoft to run this script.

---



---

You have finished configuring the `IntegrationGateway.properties` file.

### D.1.1.3 Configuring the Integration Gateway and the File output Connector

To configure the Integration Gateway and the File Output Connector:

1. In a Web browser, open your PeopleSoft 8.4 application in 4-tier mode.
2. In the Menu pane, expand **PeopleTools, Integration Broker**, and then click **Gateways**.
3. Open the LOCAL Gateway ID and enter the following Gateway URL:

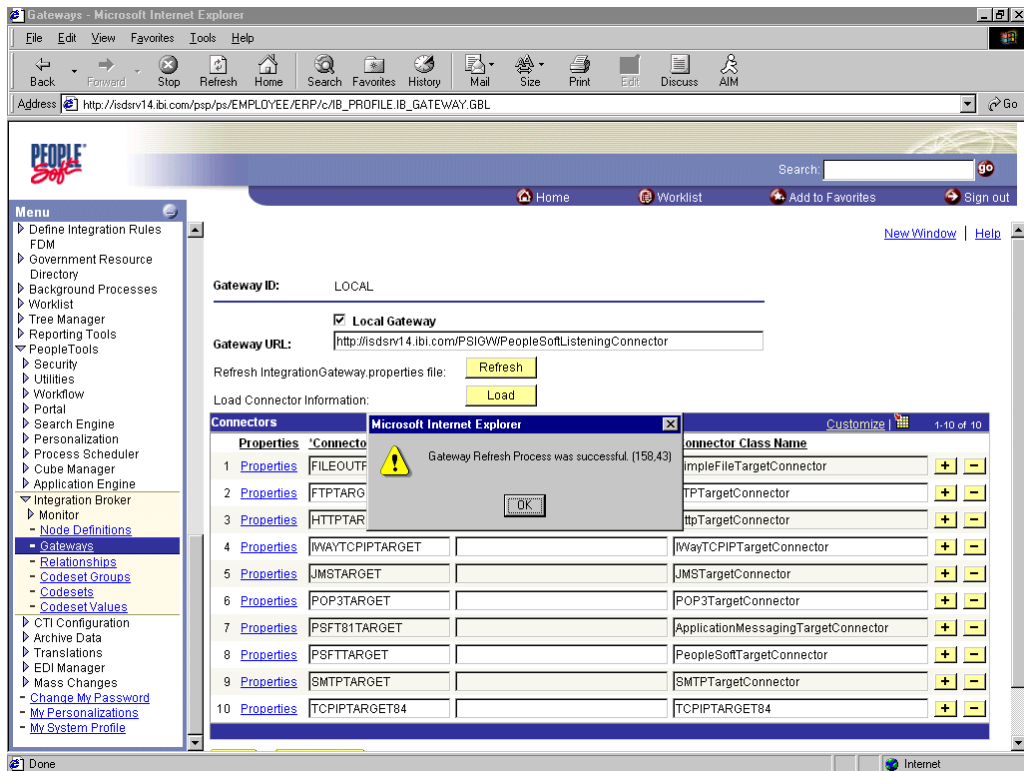
```
host name/PSIGW/PeopleSoftListeningConnector
```

Where `host name` is the name of your PeopleSoft Web Server.

4. Click **Refresh**.

A message appears stating the outcome of the refresh process, as shown in [Figure D-3](#).

**Figure D-3 Refresh Process Outcome Message**



5. Click **OK** and scroll down to click **Save**.

You must click Save before continuing.

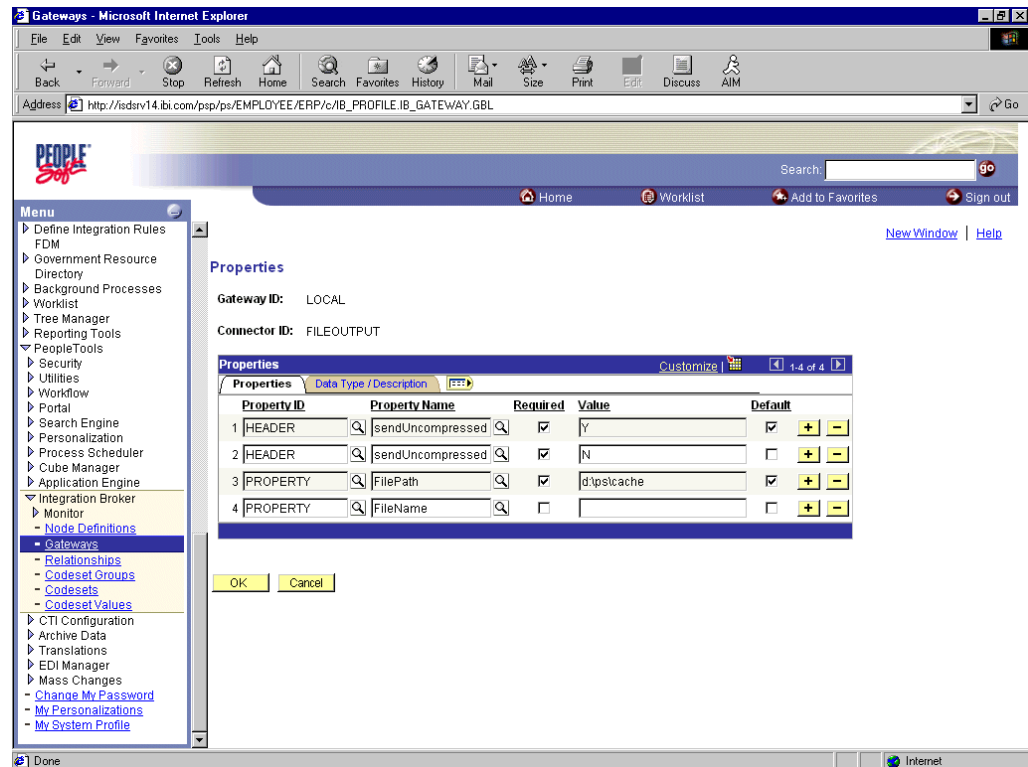
6. Click the **Properties** link for the FILEOUTPUT Connector ID.

The Properties window for the FILEOUTPUT Connector is displayed.

7. Accept or overwrite the default values.

The FilePath PROPERTY from the `c:\temp` default was changed to `d:\ps\cache`, as shown in Figure D-4.

Figure D-4 Gateway ID Property Pane



8. To return to the Gateway window, click **OK**.
9. Scroll down and click **Save**.

You have finished configuring the Integration Gateway and the File Output Connector.

#### D.1.1.4 Creating and Configuring a New Gateway Node

To create and configure a new Gateway Node:

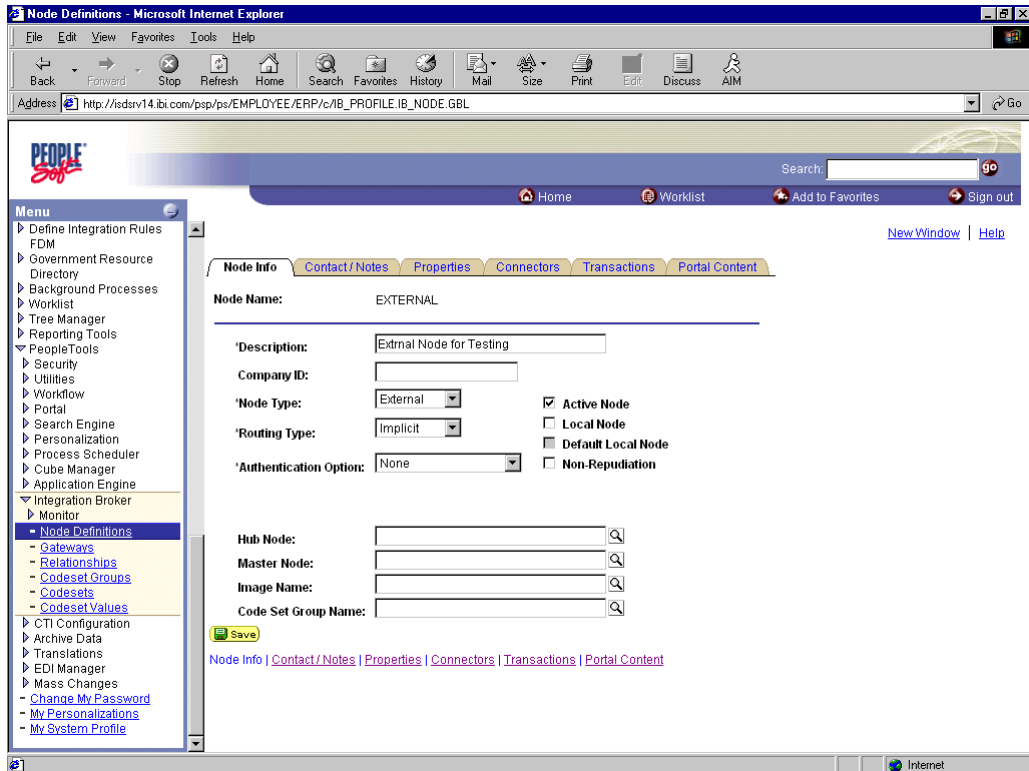
1. In the Menu pane, expand **PeopleTools**, **Integration Broker**, and then, click **Node Definitions**.
2. Select the **Add a New Value** tab.
3. In the **Node Name** field, enter a node name.

It is recommended that you name your first (trial) message node EXTERNAL. After successfully configuring and sending messages using this node, you can create additional message nodes with names appropriate for your application.

4. Click **Add**.

The Node Info tab becomes available, as shown in [Figure D-5](#).

**Figure D-5 Node Info Tab**

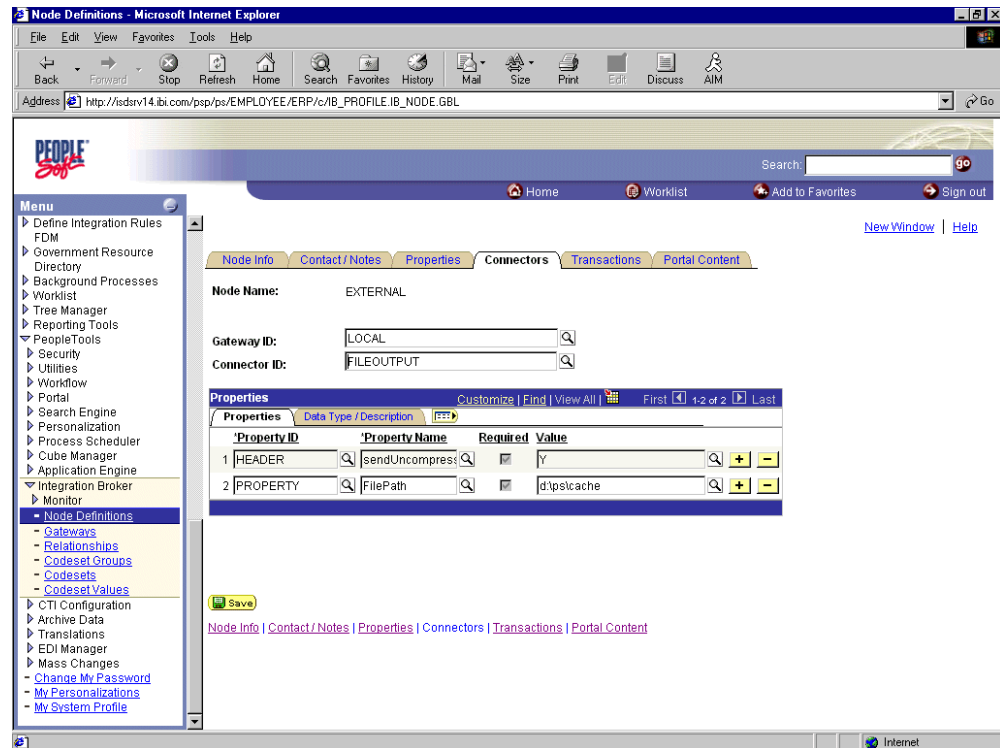


Perform the following steps:

- a. In the **Description** field, enter an appropriate description.
  - b. From the **Node Type** list, select **EXTERNAL**.
  - c. From the **Routing Type** list, select **Implicit**.
5. Select the **Connectors** tab.

The **Connectors** tab becomes available, as shown in [Figure D-6](#).

Figure D-6 Connectors Tab

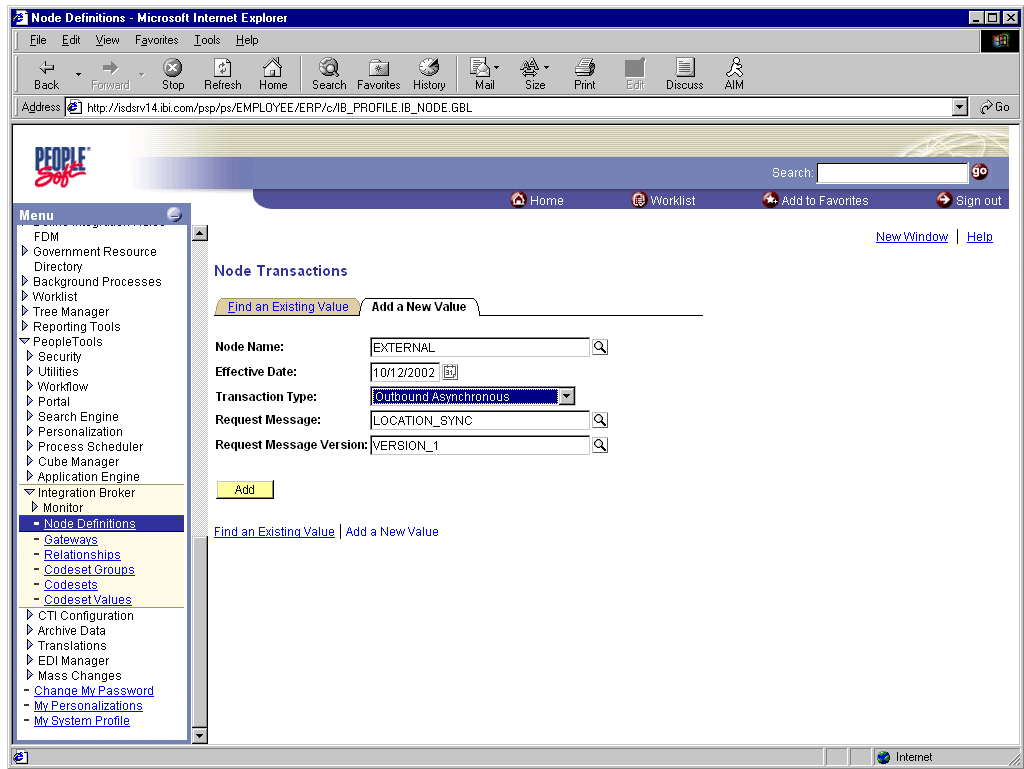


Perform the following steps:

- a. Specify **LOCAL** for the **Gateway ID**.
  - b. Specify **FILEOUTPUT** for the **Connector ID**.
  - c. Accept or overwrite the default Gateway property values.
6. Click **Save**.
  7. To specify the transactions to route messages to your node, select the **Transactions** tab and click **Add Transaction**.

The Node Transactions pane is displayed, as shown in Figure D-7.

**Figure D-7 Node Transactions Pane**



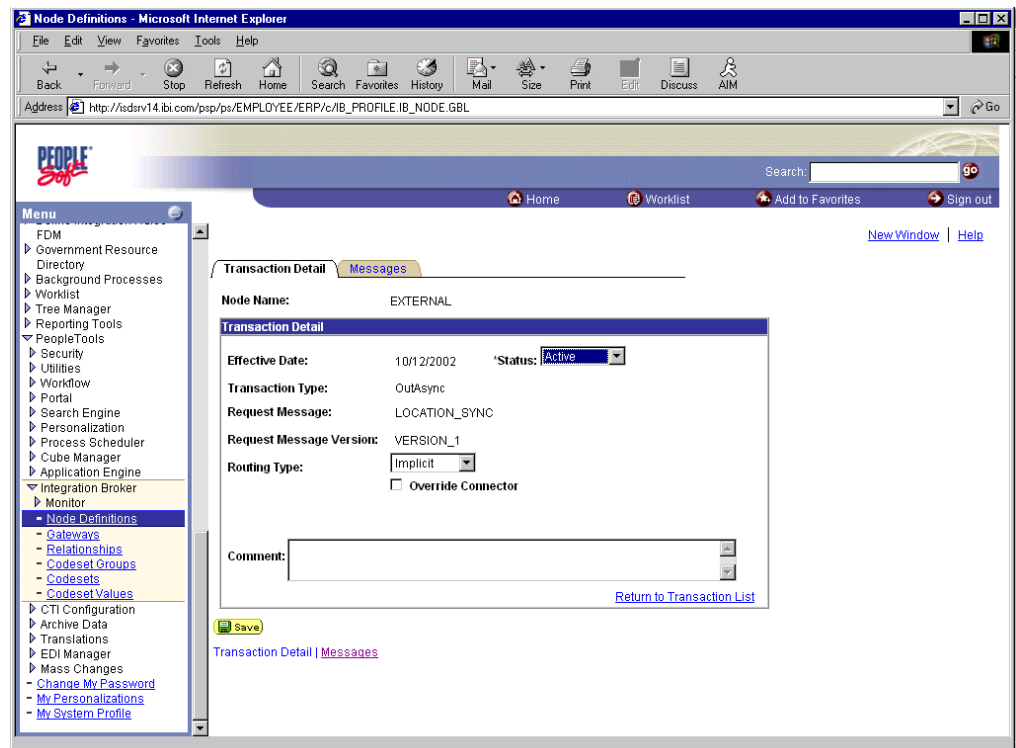
Perform the following steps:

- a. From the **Transaction Type** list, select **Outbound Asynchronous**.
  - b. In the **Request Message** field, specify **LOCATION\_SYNC**.
  - c. In the **Request Message Version** field, specify **VERSION\_1**.
- In the **Request Message Version** field, specify **VERSION\_1**.
8. Click **Add**.

The Transaction Detail pane is displayed, as shown in [Figure D-8](#).



Figure D-8 Transaction Detail Pane



Perform the following steps:

- a. Verify that the **Routing Type** is **Implicit**.
- b. Click **Save**, then click the **Return to Transaction List** link.
- c. To ensure that your data entry is not lost, click **Save** again.

You have finished creating and configuring the new Gateway Node.

For more information, see "Viewing the PeopleCode for a Message" on page D-26.

## D.2 Configuring Application Messaging in PeopleSoft Release 8.1

You can configure PeopleSoft 8.1 to send an asynchronous outbound message to the Simple File Handler.

This section consists of the following topics:

- Section D.2.1, "Creating and Configuring a New Message Node"
- Section D.2.2, "Ensuring the Message is Active and Is Routed Correctly"
- Section D.2.3, "Configuring the Message Channel"
- Section D.2.4, "Configuring the Simple File Handler in the Gateway"

To configure application messaging in PeopleSoft 8.1:

1. Create and configure a new Message Node.
2. Ensure the message is active and is routed to the proper Message Channel.
3. Configure the Message Channel.
4. Configure the Simple File Handler in the Gateway.

These tasks are described in detail in the following procedures.

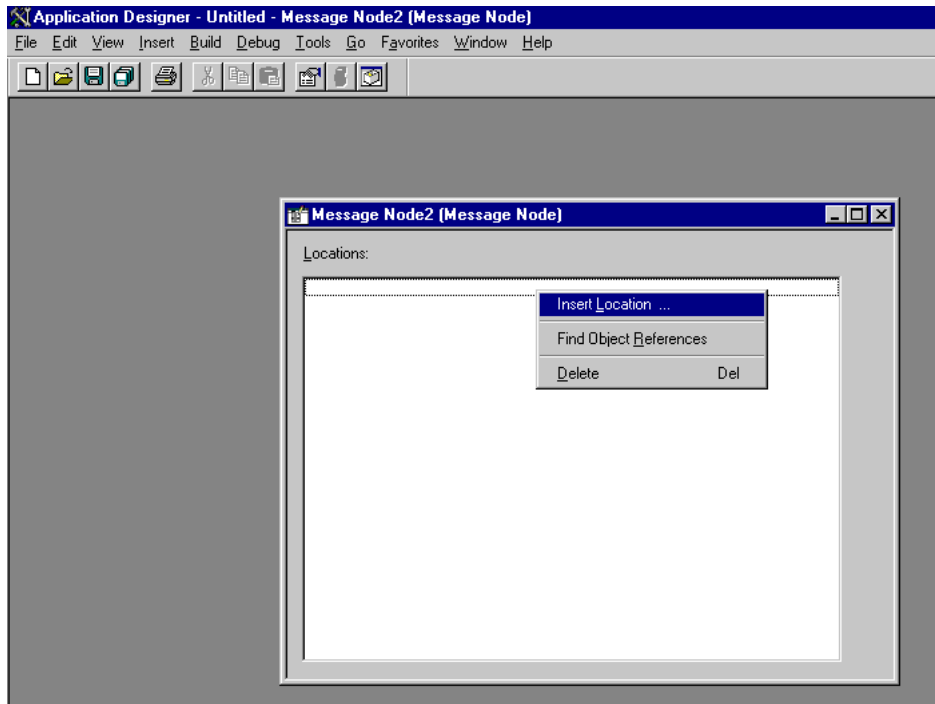
## D.2.1 Creating and Configuring a New Message Node

To create and configure a new message node:

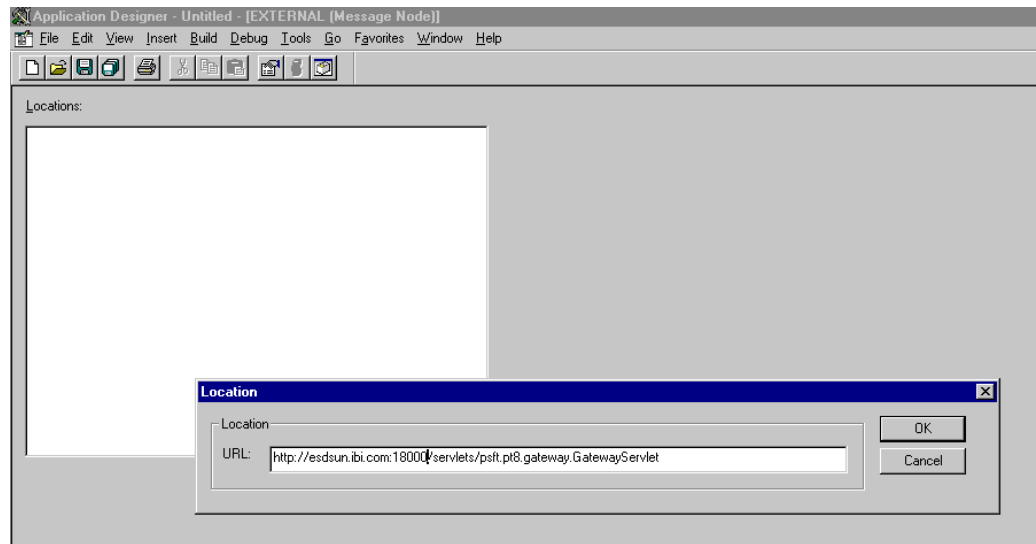
1. Select **New** from the File menu and click **Message Node**.

A Message Node window is displayed, as shown in [Figure D-9](#).

**Figure D-9** Message Node Window



2. Right-click anywhere inside the white space and select **Insert Location**.  
The Location URL box is displayed, as shown in [Figure D-10](#).

**Figure D-10 Location URL Box**

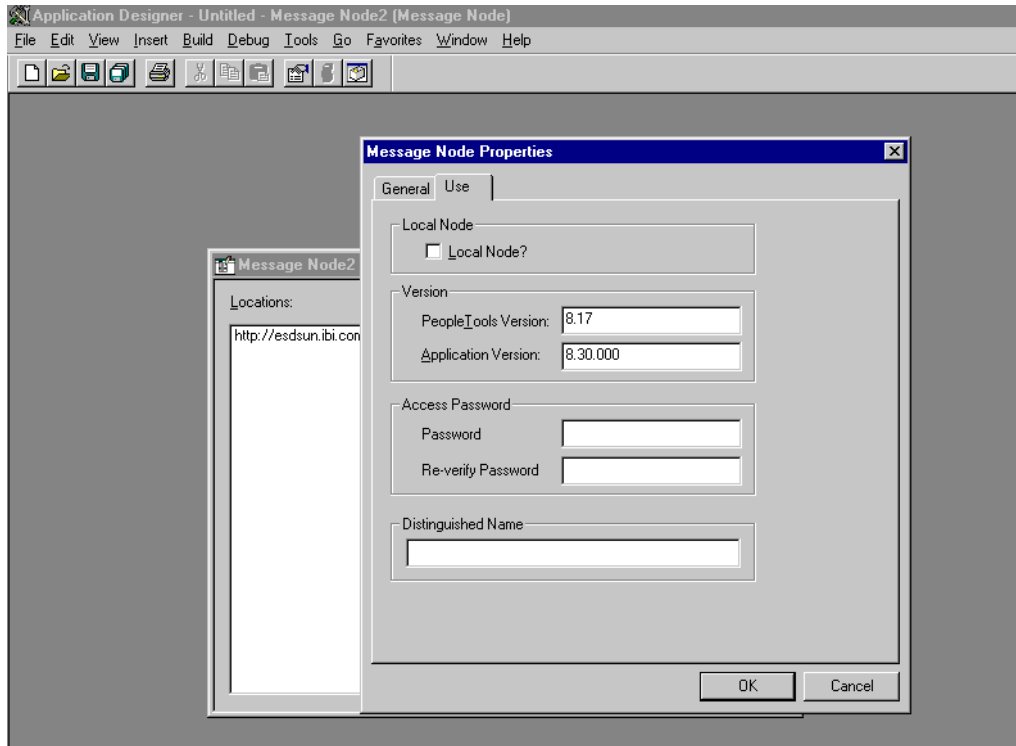
3. Enter the following URL for the PeopleSoft Application Gateway (handler directory):

*host name:port/servlets/psft.pt8.gateway.GatewayServlet*

Where *host name* is the host name of your PeopleSoft Web server and *port* is the socket on which the server is listening. The characters you enter after *host name* are case-sensitive.

4. Click **OK**.
5. Display the Message Node Properties dialog, as shown in [Figure D-11](#).

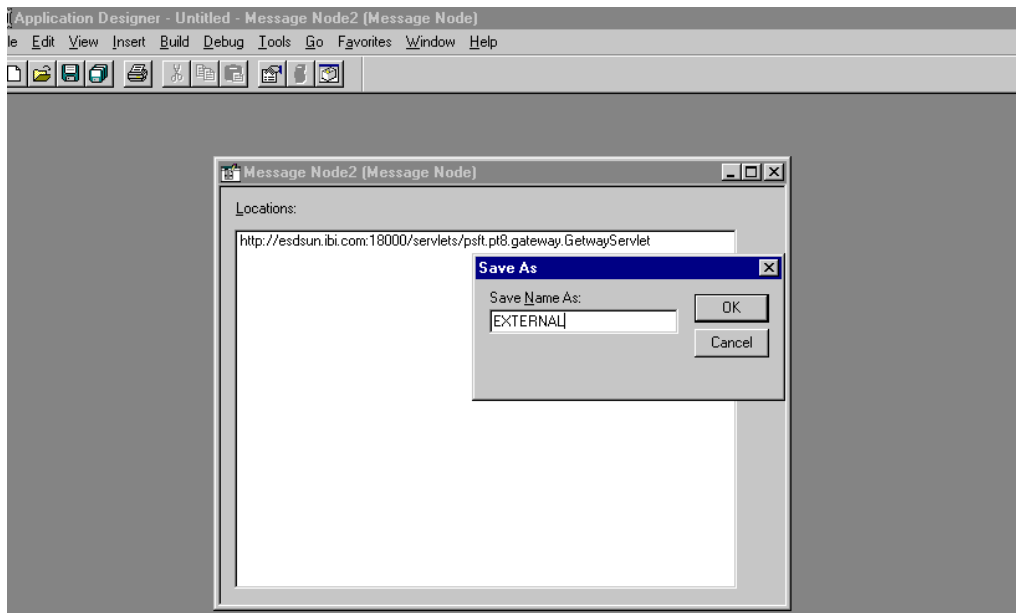
**Figure D-11 Message Node Properties Dialog**



Perform the following steps:

- a. Select the **Use** tab.
  - b. In the text boxes, enter the PeopleTools and Application Version numbers.
  - c. Click **OK**.
6. Display the Save As dialog, as shown in [Figure D-12](#).

**Figure D-12 Save As Dialog**



7. To save the Message Node, click **OK**.

It is recommended that you name your first (trial) message node **EXTERNAL**. After successfully configuring and sending messages using this node, you can create additional message nodes with names appropriate for your application.

If you intend to migrate this message node to a different PeopleSoft environment (for example, from Test to QA), you can create a PeopleSoft project and insert the Message Node into the project.

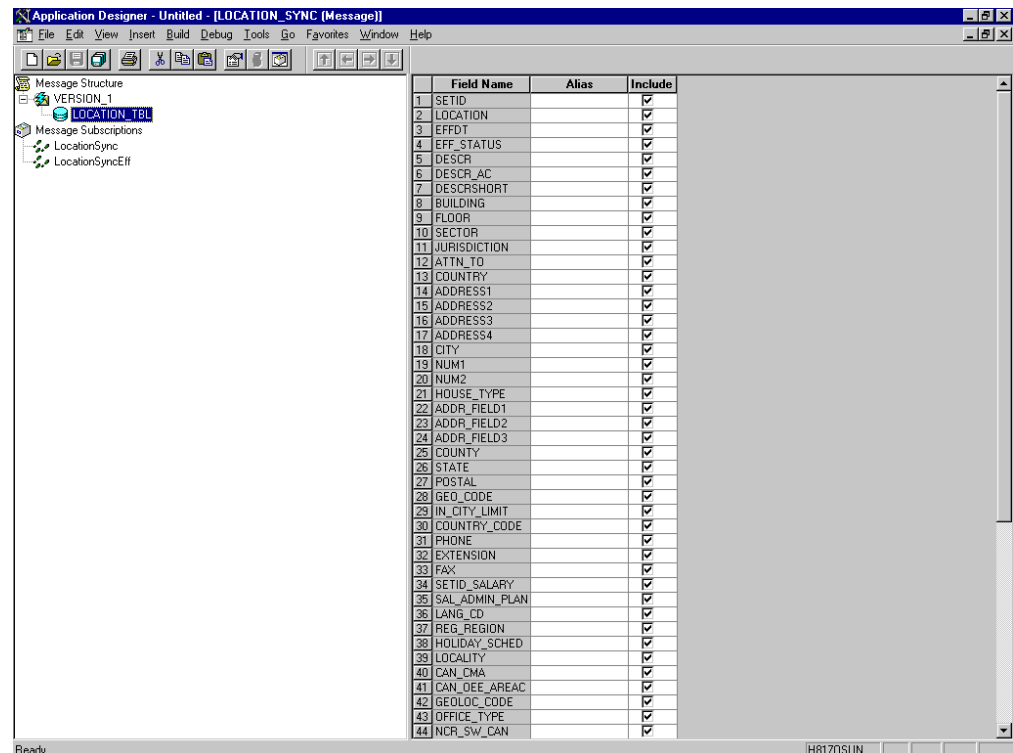
You have finished creating and configuring the message node.

## D.2.2 Ensuring the Message is Active and Is Routed Correctly

To ensure that the message is active and is routed to the proper message channel:

1. Open Application Designer.
2. On the **File** menu, point to **Open**, click **Message**, and open the **LOCATION\_SYNC** message, as shown in [Figure D-13](#).

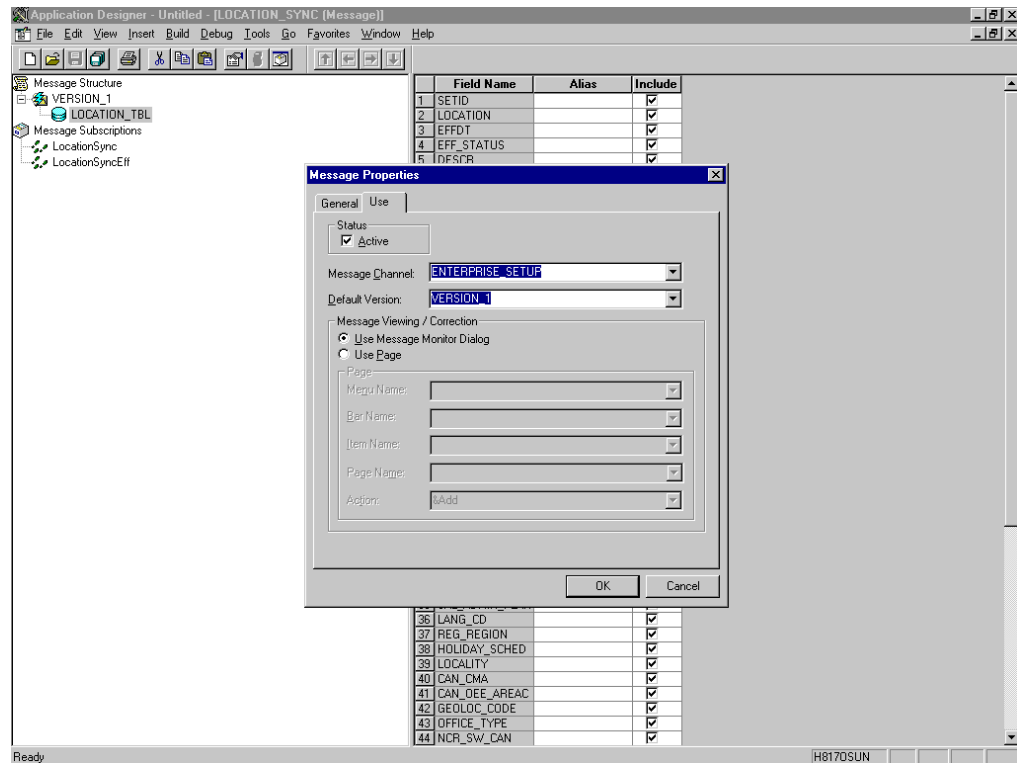
**Figure D-13** LOCATION\_SYNC Message



3. To view the fields that are included in the message, highlight **LOCATION\_TBL**.
4. Right-click **LOCATION\_TBL** and select **Properties**.

The Message Properties dialog is displayed, as shown in [Figure D-14](#).

**Figure D–14 Message Properties Dialog**



Perform the following steps:

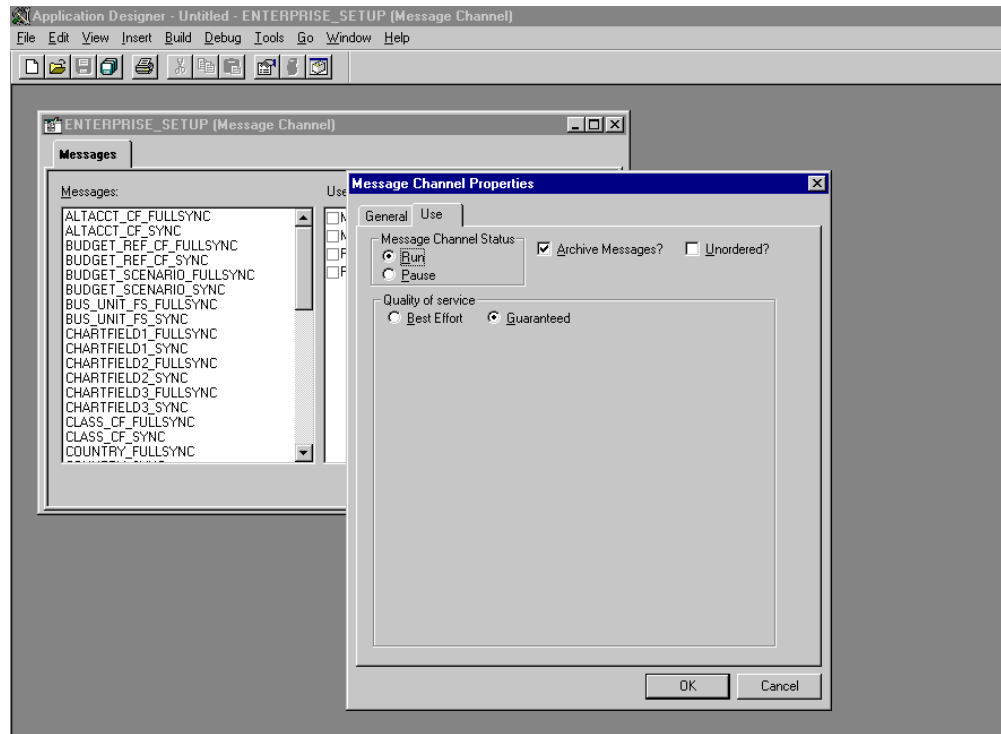
- a. Select the **Use** tab.
- b. Ensure the **Status** check box is selected, which indicates that the message is active.
- c. From the Message Channel list, select **ENTERPRISE\_SETUP**.
- d. From the Default Version list, select **VERSION\_1** (messages can have multiple versions).
5. Click **OK**.
6. Save the message.

### D.2.3 Configuring the Message Channel

To configure the message channel:

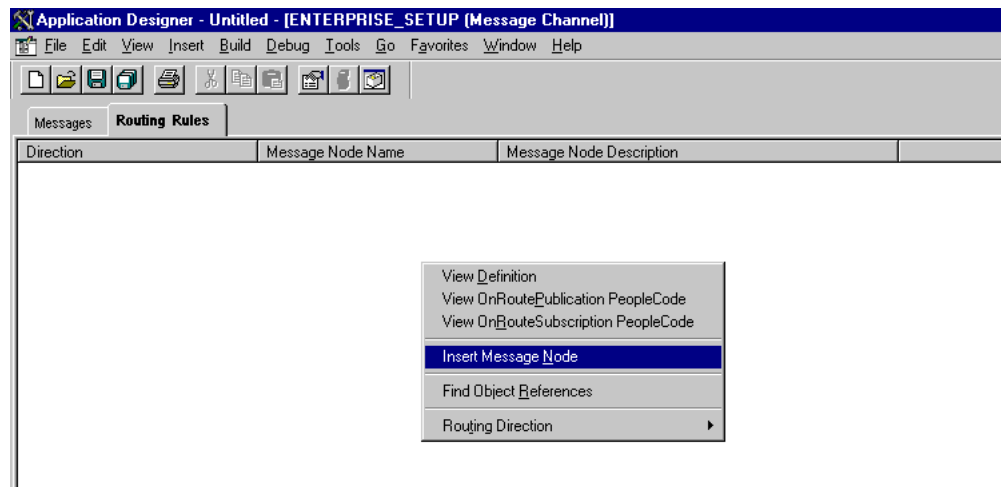
1. From the **File** menu, choose **Open** and click **Message Channel**.
2. To open the **ENTERPRISE\_SETUP** Message Channel, select **ENTERPRISE\_SETUP**.
3. Right-click **ENTERPRISE\_SETUP** and select **Properties**.

The Message Channel Properties dialog is displayed, as shown in [Figure D–15](#).

**Figure D–15 Message Channel Properties Dialog**

Perform the following steps:

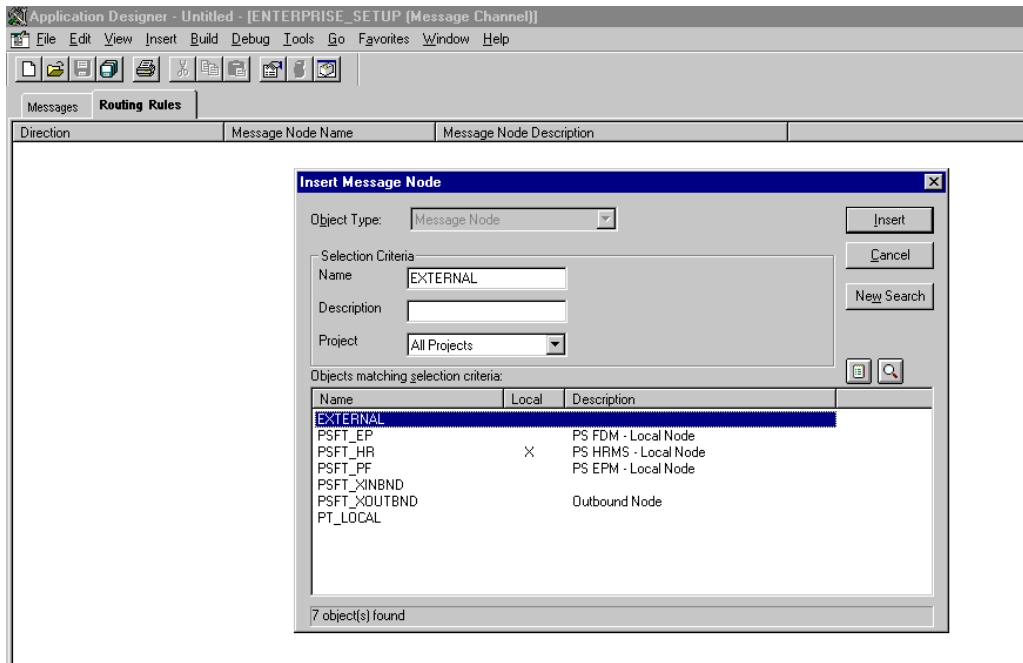
- a. Select the **Use** tab.
  - b. Ensure that the Message Channel status is set to **Run**.
  - c. Click **OK**.
4. From the left pane, select the **Routing Rules** tab.  
The pane is blank, as shown in [Figure D–16](#).

**Figure D–16 Routing Rules Pane**

5. Right-click the pane and select **Insert Message Node**.

The Insert Message Node dialog is displayed, as shown in [Figure D-17](#).

**Figure D-17 Insert Message Node Dialog**

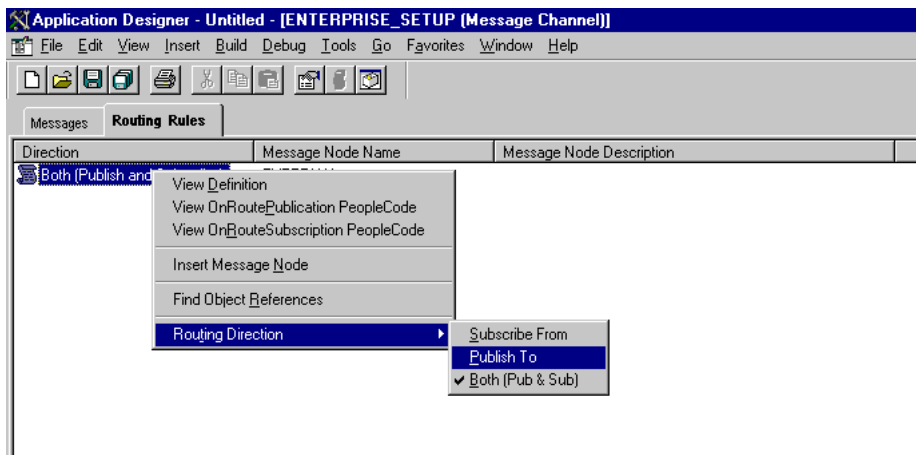


Perform the following steps:

- a. Select the message node that you created in "[Creating and Configuring a New Message Node](#)" on page D-12, for example, EXTERNAL.
  - b. Click **Insert**.
6. Click **Cancel**.

Information appears on the Routing Rules tab, as shown in [Figure D-18](#).

**Figure D-18 Routing Rules Tab**



Perform the following steps:

- a. Right-click the message node and point to **Routing Direction**.



- b. From the **Routing Direction** menus, select **Publish To**.
7. Save the Message Channel, and if you require it, place it in your project.  
You have finished configuring the Message Channel.

## D.2.4 Configuring the Simple File Handler in the Gateway

To configure the Simple File Handler in the Gateway:

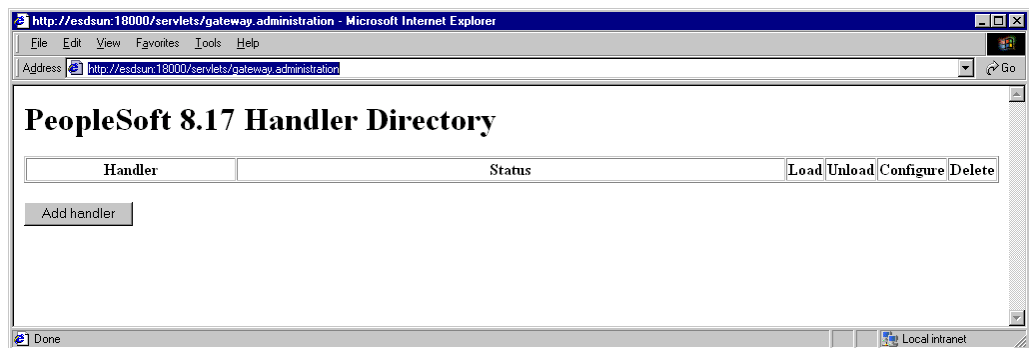
1. In a Web browser, launch the PeopleSoft 8.1 configuration servlet interface (also known as the server gateway) by entering the following URL:

```
host name:port/servlets/gateway.administration
```

Where `host name` is the name of the application server that hosts PeopleSoft and `port` is the port number on which the application server is listening.

The Handler Directory window is displayed, as shown in [Figure D-19](#).

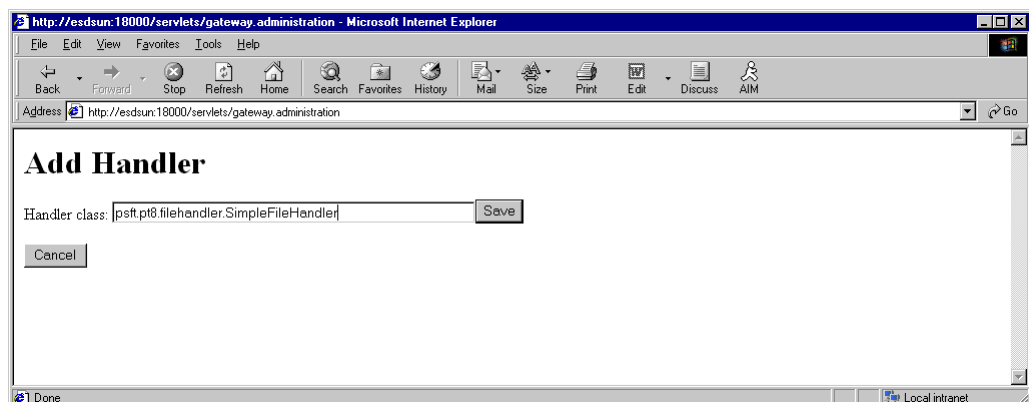
**Figure D-19 Handler Directory Window**



2. Click **Add Handler**.

The Add Handler window is displayed, as shown in [Figure D-20](#).

**Figure D-20 Add Handler Window**



3. Enter the full name of the Simple File Handler class,  
`psft.pt8.filehandler.SimpleFileHandler`.

---

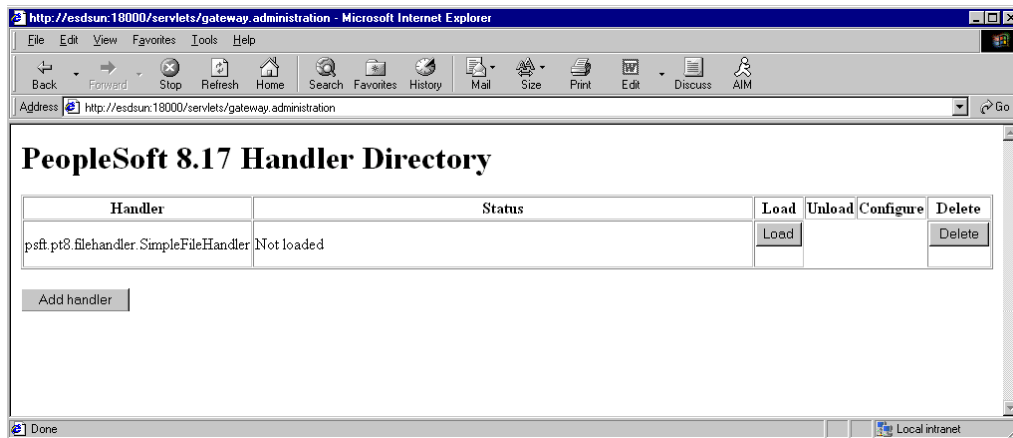
**Note:** The name is case-sensitive.

---

4. Click **Save**.

The Handler Directory window is displayed, as shown in [Figure D-21](#).

**Figure D-21 Handler Directory Window**



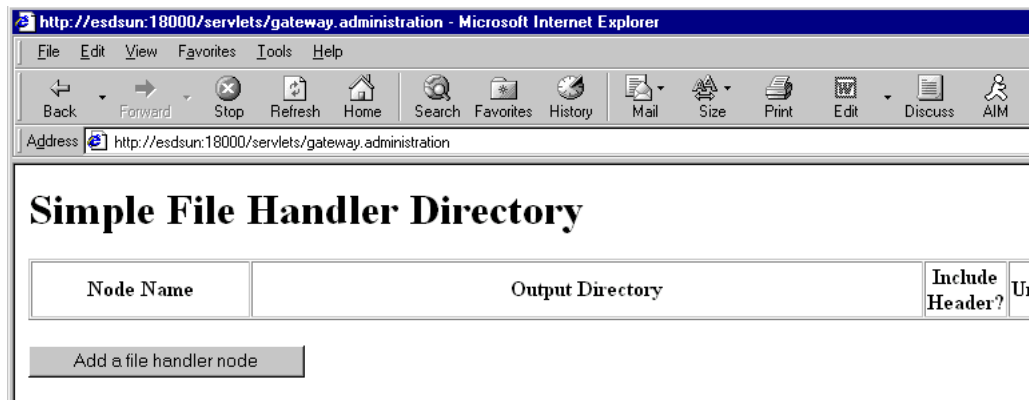
5. To load the handler, click **Load**.

After the handler loads, "Loaded successfully" appears in the Status column.

6. Click **Configure**.

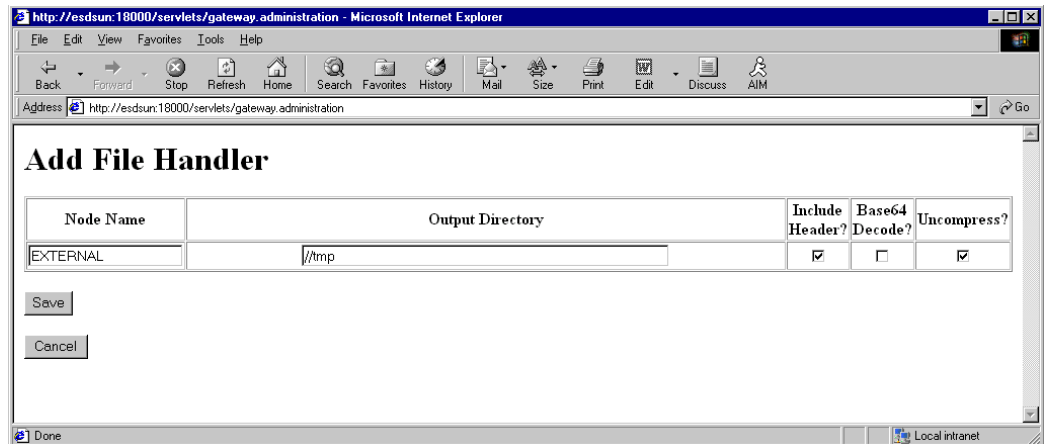
The Simple File Handler Directory window is displayed, as shown in [Figure D-22](#).

**Figure D-22 Simple File Handler Directory Window**



7. Click **Add a file handler node**.

The Add File Handler window is displayed, as shown in [Figure D-23](#).

**Figure D–23 Add File Handler Window**

8. In the Node Name field, enter the name of the message node that you created in "Creating and Configuring a New Message Node" on page D-12, for example, EXTERNAL.

## D.3 Viewing the PeopleCode for a Message

Messages are initiated by the PeopleCode that is attached to a record. Usually, this record is one of the records associated with the message itself.

1. Open Application Designer, as shown in [Figure D–24](#).

**Figure D–24 Application Designer**

Num	Field Name	Type	FDe	FEd	FCh	FFo	RIn	RIs	RDd	RSe	SEd	SPr	SPo	Srl	SrS	Wk	PPr
1	SETID	Char															
2	LOCATION	Char															
3	EFFDT	Date															
4	EFF_STATUS	Char															
5	DESCR	Char															
6	DESCR_AC	Char															
7	DESCRSHORT	Char															
8	BUILDING	Char															
9	FLOOR	Char															
10	SECTOR	Char															
11	JURISDICTION	Char															
12	ATTN_TO	Char															
13	ADDRESS_SBR	SRec															
14	PHONE_SBR	SRec															
15	EXTENSION	Char															
16	FAX	Char															
17	SETID_SALARY	Char															
18	SAL_ADMIN_PLAN	Char															
19	LANG_CD	Char															
20	REG_REGION	Char															
21	HOLIDAY_SCHEDULE	Char															
22	LOC_TBLUSA_SBR	SRec															
23	LOC_TBLCAN_SBR	SRec															
24	WRKS_CNCL_SBR	SRec															
25	LOC_TBLGER_SBR	SRec															
26	LOC_TBLUK_SBR	SRec															
27	LOC_TBL_FED_SBR	SRec															
28	LOC_TBLESP_SBR	SRec															
29	LABEL_FORMAT_ID2	Char															
30	LABEL_FORMAT_ID3	Char															
31	USG_LBL_FORMAT_ID	Char															
32	LOC_TBLMEX_SBR	SRec															
33	ESTABID	Char															
34	COMMENTS_2000	Long															

2. On the **Record Fields** tab, select the **LOCATION\_TBL** record.
3. Select the **PeopleCode** display option.
4. Select the **Save Post Change (SPo)** box for the **LOCATION** field.

The following window displays the PeopleCode that initiates a **LOCATION\_SYNC** message, as shown in [Figure D-25](#).

**Figure D-25 PeopleCode**

```

Application Designer - Untitled - I:LOCATION_TBL.LOCATION.SavePostChange (Record PeopleCode)
File Edit View Insert Build Debug Tools Go Favorites Window Help
LOCATION (field) SavePostChange
Local Message <MSG;
Local Rowset <RS0, <RS01, <RS11;
Local Row <R1;
PanelGroup string <PubNodeName;
PanelGroup boolean <ActionCodeRowAdd;

If ActiveRowCount() = CurrentRowNumber() Then
  <MSG = CreateMessage(Message.LOCATION_SYNC);

  If <MSG.IsActive Then

    /* If the program is called by a Component Interface, then flush the extra row (Created by Component Create()), CopySetupRowset on
    Component Interface "LOCATION".) from panel buffer.*/
    /**=====*/

    If ((%ComponentName = Component.LOCATION) And
        (<ActionCodeRowAdd = True)) Then

      <RS0 = GetLevel0();
      <RS01 = CreateRowset(<RS0);
      <RS0.CopyTo(<RS01);
      <RS11 = <RS01(1).GetRowset(Scroll.LOCATION_TBL);
      <RS11.Flush();

      <R1 = <RS0(1).GetRowset(Scroll.LOCATION_TBL).GetRow(1);
      <R1.CopyTo(<RS11.GetRow(1));

      <RS01.CopyTo(<RS0);

    End-If;
    /**=====*/

    <MSG.CopyRowsetDelta(GetLevel0(1).GetRowset(Scroll.LOCATION_TBL));

    /* prevent circular publishes, do not publish back to originating node */
    If All(<PubNodeName) Then
      <MSG.DoNotPubToNodeName = <PubNodeName;
    End-If;

    <MSG.Publish();

  End-If;
End-If;
Ready H8170SUN

```

For more information about PeopleCode, consult your PeopleSoft Online Library.

You have finished viewing the PeopleCode for a message. You can now test Integration Broker (in PeopleSoft 8.4) or Application Messaging (in PeopleSoft 8.1).

## D.4 Testing the Integration Broker

To test the Integration Broker by generating a message, you can navigate to the Location Transaction window and add, update, or delete a location entry in your application. Depending on your application, the way you navigate varies.

An example that illustrates a Financials 8.4 application where a new location with a SetID of SHARE and a Location Code of TEST001 was added, as shown in [Figure D-26](#).

Figure D-26 Financials 8.4 Application

The screenshot shows a web browser window with the address `http://fsdsv14.ibi.com/ps/ps/EMPLOYEE/ERP/c/DEFINE_GENERAL_OPTIONS.LOCATION_TBL.GBL`. The application interface includes a menu on the left with options like 'Location', 'Country Statistics', and 'Division Area Code'. The main content area is titled 'Location Definition' and shows a form for defining a location. The form fields are as follows:

- SetID: SHARE Location Code: TEST001
- Location Definition: Find | View All | First | 1 of 1 | Last
- 'Eff Date': 10/12/2002 | Active | 'Descr': My first test location
- Country: USA | United States | Prefix: | Phone: |
- Address 1: Some Street | Ext: |
- Address 2: | Fax: |
- Address 3: |
- City: Some City | In City Limit:
- County: | Postal: |
- State: NY | Jurisdiction: |
- Building #: | Floor #: | Sector: |

Buttons at the bottom include Save, Notify, Add, Update/Display, Include History, and Correct History.

A portion of the XML output is shown in Figure D-27.

Figure D-27 XML Output

```

<?xml version="1.0" ?>
- <LOCATION_SYNC>
- <FieldTypes>
+ <LOCATION_TBL class="R">
  <PSCAMA class="R">
    <LANGUAGE_CD type="CHAR" />
    <AUDIT_ACTN type="CHAR" /&;
    <BASE_LANGUAGE_CD type="CHAR" />
    <MSG_SEQ_FLG type="CHAR" />
    <PROCESS_INSTANCE type="NUMBER" />
    <PUBLISH_RULE_ID type="CHAR" />
    <MSGNODENAME type="CHAR" />
  </PSCAMA>
</FieldTypes>
- <MsgData>
- <Transaction>
  - <LOCATION_TBL class="R">
    <SETID>SHARE</SETID>
    <LOCATION>TEST001</LOCATION>
    <EFFDT>2002-10-12</EFFDT>
    <EFF_STATUS>A</EFF_STATUS>
    <DESCR>My first test location</DESCR>
    <DESCR_AC />
    <DESCRSHORT />
    <BUILDING />
    <FLOOR />
    <SECTOR />
    <JURISDICTION />
    <ATTN_TO />
    <COUNTRY>USA</COUNTRY>
    <ADDRESS1>Some Street</ADDRESS1>
    <ADDRESS2 />
    <ADDRESS3 />
    <ADDRESS4 />
    <CITY>Some City</CITY>
    <ATTN1 />
  </LOCATION_TBL>
</Transaction>
</MsgData>
</LOCATION_SYNC>

```

---

---

**Note:** The name of the file is `PSFT_EP.LOCATION_SYNC.69.xml`, which is the concatenation of `PSFT_EP` (the local Publishing Node), the name of the message, and the number of the Publication ID.

---

---

If you cannot send a message successfully, PeopleSoft provides a set of tools for monitoring the progress of your messages. In release 8.1, you use a tool called the Application Messaging Monitor. In release 8.4, you use the Monitor Menu in the Integration Broker.

For a complete description on how to isolate and resolve problems with your messaging environment, consult your PeopleSoft Online Library. If you are still unable to send your XML file, the PeopleSoft Customer Connection can help solve your problem.

## D.5 Using Outbound Synchronous Messages

Starting with PeopleTools 8.4, you can send outbound synchronous messages. From a high-level point of view, the primary difference between outbound synchronous and asynchronous is that with outbound asynchronous, the transaction is completed whether the message is actually sent or received.

This section consists of the following topics:

- [Section D.5.1, "Configuring Outbound Synchronous Messages"](#)
- [Section D.5.2, "Viewing the PeopleCode for a Synchronous Message"](#)

For synchronous outbound messages:

- The transaction must wait for a response from the external system before continuing.
- The transaction must process the response message.
- The external system must ensure that the response message is correctly formatted.

Oracle Application Adapter for PeopleSoft can work with PeopleSoft outbound synchronous messages. Outbound synchronous messages involve additional configuration steps, both within PeopleSoft and in Oracle WebLogic Server. This topic briefly describes the configuration requirements within PeopleSoft.

---

---

**Note:** The instructions in this topic build upon the instructions for outbound asynchronous messages. It is strongly recommended that you familiarize yourself with outbound asynchronous messaging before attempting outbound synchronous. For more information on outbound asynchronous messages, see ["Configuring Integration Broker in PeopleSoft 8.4"](#) on page D-2.

Ensure that both outbound and inbound messages are created and active. PeopleSoft provides template examples called `IB_INST_VER_SYNC_MSG` and `IB_INST_VER_RESP_MSG`. For information on examining these messages, see ["Ensuring the Message is Active and Is Routed Correctly"](#) on page D-15.

---

---

### D.5.1 Configuring Outbound Synchronous Messages

You can use an existing node, or you can create a new node to configure outbound synchronous messages. For more information on creating and configuring a node, see

"Creating and Configuring a New Message Node" on page D-12. In either case, you must set up your outbound synchronous transaction.

This section consists of the following topic:

- [Section D.5.1.1, "Configuring an Outbound Synchronous Message"](#)

The following example uses a node and transaction delivered by PeopleSoft. However, this example is for illustrative purposes only and does not actually work as delivered without additional steps. As of Financials release 8.42, there are no preconfigured outbound synchronous transactions that you can use for testing purposes.

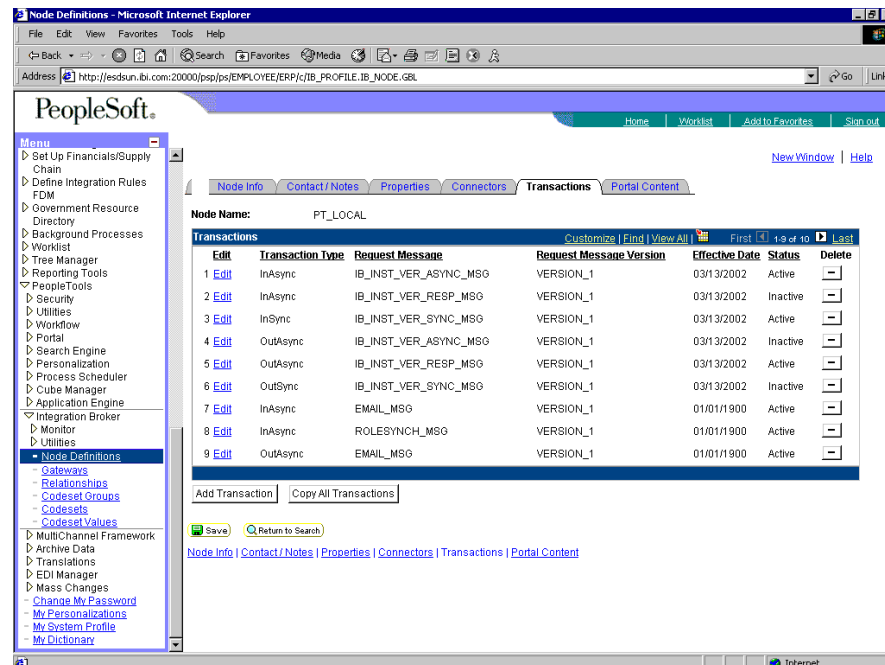
### D.5.1.1 Configuring an Outbound Synchronous Message

To configure an outbound synchronous message:

1. Navigate to the **Node Definitions** page and open the **PT\_LOCAL** node.
2. Click the **Transactions** tab.

The Transactions pane is displayed, as shown in [Figure D-28](#).

**Figure D-28 Transactions Pane**

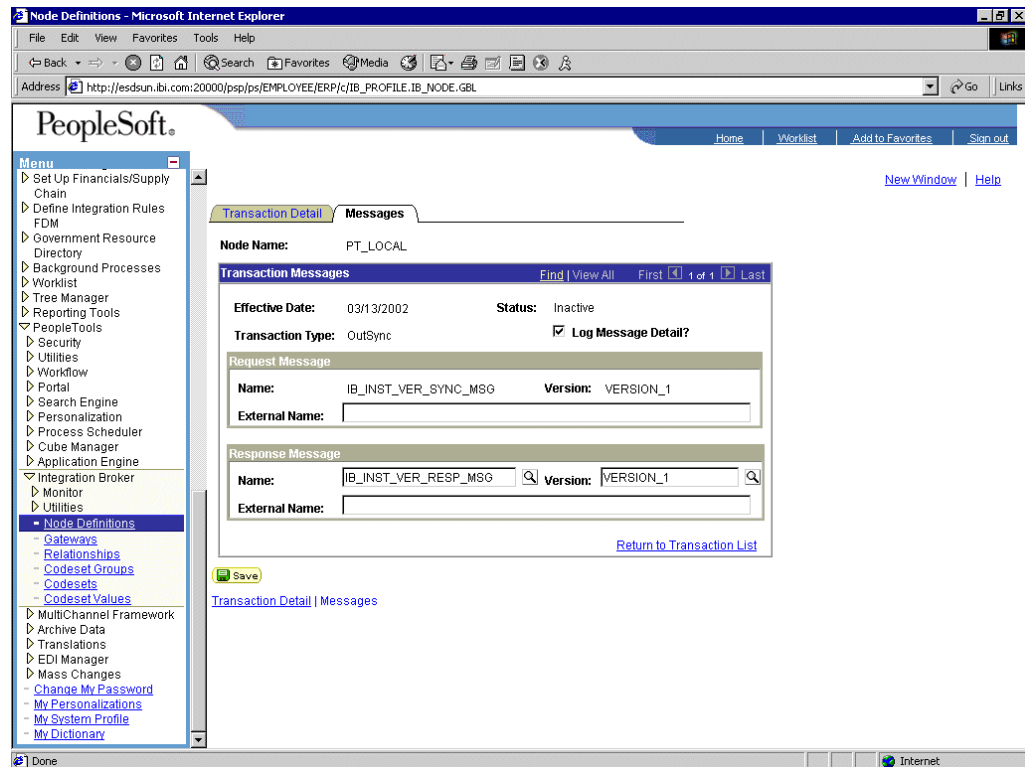


One outbound synchronous message, **IB\_INST\_VER\_SYNC\_MSG**, appears in the Transaction Type list.

3. Click the **Edit** link in the **IB\_INST\_VER\_SYNC\_MSG** row.

The following pane is displayed, as shown in [Figure D-29](#).

Figure D-29 Message Tab



4. Click the **Messages** tab.

Both request and response messages appear. The target system must ensure that the response message follows the format of the request message. As the target system is your Oracle WebLogic Server, you must transform the XML that is sent and returned from your final destination.

---

**Note:** You must use the PeopleSoft-supplied HTTP target connector when you are working with synchronous outbound messages. You cannot use the TCPIP84TARGET connector for outbound synchronous messages.

---

## D.5.2 Viewing the PeopleCode for a Synchronous Message

The sample PeopleCode in the following example is for a synchronous outbound message. It differs from asynchronous outbound in that it must handle a response message.

This section consists of the following topic:

- [Section D.5.2.1, "Viewing the PeopleCode for a Financials Synchronous Outbound Message"](#)

### D.5.2.1 Viewing the PeopleCode for a Financials Synchronous Outbound Message

The following sample code is supplied with the Financials application and is associated with the two messages IB\_INST\_VER\_SYNC\_MSG and IB\_INST\_VER\_RESP\_MSG.

To view the code, perform the following steps:



1. From Application Designer, open the PSINST\_VER record.
2. Select the **PeopleCode** display option.
3. Select the **Field Change (FCh)** box for the IB\_SEND\_SOS\_BTN field.

The following window is displayed, as shown in [Figure D–30](#).

**Figure D–30 Field Change Box**

```

Application Designer - Untitled - [PSINST_VER_IB_SEND_SOS_BTN.FieldChange (Record PeopleCode)]
File Edit View Insert Build Debug Tools Go Window Help
IB_SEND_SOS_BTN (field) FieldChange
/* SyncRequest example */
Local Message <request_MSG, <response_MSG;
Local Rowset <request_RS, <response_RS, <IB_INST_VER_TRX_RS;
Local Record <response_REC, <IB_INST_VER_DB_REC;
Local SQL <delete_SQL;
Local any <I;

<request_RS = GetLevel0();
<request_MSG = CreateMessage(Message.IB_INST_VER_SYNC_MSG);
<request_MSG.CopyRowset(<request_RS);

/* Create the database record object for the response data */
<IB_INST_VER_DB_REC = CreateRecord(Record.PSINST_VER_TRX);

/* publish the request and wait for the response */
<response_MSG = <request_MSG.SyncRequest();

If (<response_MSG.ResponseStatus = 0) Then

    /* Get the response rowset object from the buffer */
    <response_RS = <response_MSG.GetRowset();

    /* Loop through the message rows moving the data into the database table */
    For <I = 1 To (<response_RS.RowCount)
        <response_REC = <response_RS.GetRow(<I).GetRecord(Record.PSINST_VER_TRX);
        <response_REC.CopyFieldsTo(<IB_INST_VER_DB_REC);
        <IB_INST_VER_DB_REC.Insert();
    End-For;
End-If;

/* Manual refresh of scrollable area */
<IB_INST_VER_TRX_RS = GetLevel0()(1).GetRowset(Scroll.PSINST_VER_TRX);
<IB_INST_VER_TRX_RS.Flush();
<IB_INST_VER_TRX_RS.Select(Record.PSINST_VER_TRX);
  
```

## D.6 Using PeopleTools Version 8.48 and Higher to Generate Events

This section provides a tutorial that walks you through the PeopleSoft event generation process. PeopleTools versions 8.48, 8.49, 8.50, and 8.51 are supported. The tutorial uses the PeopleTools version 8.48.02 and 8.90 Financial / SCM application. As a result, if you are using a different platform, then correlate the terms and commands specific to that operating system. For more information, refer to the appropriate user guide for each specific component.

This section consists of the following topics:

- [Section D.6.1, "Configuring PeopleSoft Services"](#)
- [Section D.6.2, "Triggering the Event in PeopleSoft"](#)
- [Section D.6.3, "Verifying the Event Results"](#)

### D.6.1 Configuring PeopleSoft Services

This section describes how to configure PeopleSoft services and consists of the following topics:

- [Section D.6.1.1, "Adding a Remote Node"](#)
- [Section D.6.1.2, "Selecting the Service"](#)

- [Section D.6.1.3, "Activating the Gateway"](#)
- [Section D.6.1.4, "Activating the Domain"](#)
- [Section D.6.1.5, "Activating the Service"](#)
- [Section D.6.1.6, "Activating the Service Operation \(Queue and Message Node\)"](#)

### D.6.1.1 Adding a Remote Node

To add a remote node:

1. Logon to PeopleSoft using the browser-based GUI (Pure Internet Architecture).
2. Select **PeopleTools, Integration Broker, Integration Setup**, followed by **Nodes**.  
The Node Definitions tab is displayed, as shown in [Figure D–31](#).

**Figure D–31 Node Definitions Tab**

The screenshot shows the 'Node Definitions' tab with the 'Connectors' sub-tab active. The form contains the following fields and options:

- Node Name:** EXTERNAL
- Description:** Message publish to third party
- Node Type:** External (dropdown menu)
- Authentication Option:** None (dropdown menu)
- Default User ID:** PS
- Checkboxes:**
  - Default Local Node
  - Local Node
  - Active Node
  - Non-Repudiation
  - Segment Aware
- Action Buttons:** Copy Node, Rename Node, Delete Node

3. Perform the following steps:
  - a. In the Node Name field, type a name for the new node definition, for example, **EXTERNAL**.
  - b. In the Description field, type a brief description for the new node definition.
  - c. From the Node Type drop-down list, select **External**.
  - d. From the Authentication Option drop-down list, ensure that the default option, **none**, is selected.
  - e. In the Default User ID field, type the user ID that is being used by the PeopleSoft system, for example, **PS**.
  - f. Select the **Active Node** check box.
4. Click the **Connectors** tab.

The Connectors tab is displayed, as shown in [Figure D–32](#).

**Figure D–32 Connectors Tab**

Node Name: EXTERNAL Ping Node

**Details**

Gateway ID: LOCAL

Connector ID: HTTPTARGET

**Properties** Customize | Find | First 1-3 of 3 Last

Property ID	Property Name	Required	Value
1	HEADER	<input checked="" type="checkbox"/>	Y
2	HTTPPROPERTY	<input checked="" type="checkbox"/>	Method
3	PRIMARYURL	<input checked="" type="checkbox"/>	URL

► Password Encryption Utility

5. Perform the following steps:
  - a. In the Gateway ID field, type **LOCAL**.
  - b. In the Connector ID field, type **HTTPTARGET**.
  - c. For the PRIMARYURL value, enter the host and port number of the PeopleSoft adapter instance that is used to listen for events.
  - d. Leave the default values for the remaining properties.
6. Save your changes.

### D.6.1.2 Selecting the Service

To select a service:

1. Select **PeopleTools, Integration Broker, Integration Setup**, followed by **Services**.
2. Search for the DEPT\_SYNC service and select it.

The DEPT\_SYNC Services pane is displayed, as shown in [Figure D–33](#).

**Figure D–33 DEPT\_SYNC Services Pane**

**Services**

Service: DEPT\_SYNC  
 \*Description: Dept Sync Incr. Message  
 Comments: Incremental (Component Publish) Dept Sync Message  
 Service Alias:   
 Object Owner ID: Human Resources  
 \*Namespace: http://xmlns.oracle.com/Enterprise/Tools/services

[View WSDL](#)   [Provide Web Service](#)

**Service Operations**

Service Operation:   
 Operation Type:  **ADD**

**Existing Operations**   [Customize](#) | [Find](#) | [View All](#) | First 1-2 of 2 Last

Operation	Message Links	Active	Operation Type
<a href="#">ADD_DEPT.v1</a>		<input checked="" type="checkbox"/>	Asynch
<a href="#">DEPT_SYNC.VERSION_1</a>		<input checked="" type="checkbox"/>	Asynch

- From the Existing Operations section on the lower-left, click the DEPT\_SYNC.VERSION\_1 service operation.

The Service Operation - General pane is displayed, as shown in Figure D–34.

**Figure D–34 Service Operation - General Pane**

**Default Service Operation Version**

\*Version: VERSION\_1  
 Version Description: Dept Sync Incr. Message  
 Version Comments:   
 Non-Repudiation  
 Runtime Schema Validation

[Introspection](#)

Default    Active

**Routing Status**

Any-to-Local: Exists  
 Local-to-Local: Exists

**Routing Actions Upon Save**

Regenerate Any-to-Local  
 Regenerate Local-to-Local

- Click the Active check box.
- Click the **Routings** tab to add a new routing, as shown in Figure D–35.

**Figure D–35** *Routing Tab*

The screenshot shows a tabbed interface with three tabs: 'General', 'Handlers', and 'Routings'. The 'Routings' tab is active. It contains the following fields and controls:

- Service Operation:** DEPT\_SYNC
- Default Version:** VERSION\_1
- Routing Name:** A text input field containing 'ADD\_DEPT'.
- ADD:** A yellow button to the right of the Routing Name field.

6. In the Routing Name field, type a name for the new routing, for example, **ADD\_DEPT**.
7. Click **Add**.
8. Click the **Routing Definitions** tab to add a new routing definition, as shown in [Figure D–36](#).

**Figure D–36** *Routing Definitions Tab*

The screenshot shows a tabbed interface with three tabs: 'Routing Definitions', 'Parameters', and 'Connector Properties'. The 'Routing Definitions' tab is active. It contains the following fields and controls:

- Routing Name:** ADD\_DEPT
- \*Service Operation:** DEPT\_SYNC
- Version:** VERSION\_1
- \*Description:** ADD\_DEPT
- Comments:** A large text area with scrollbars.
- \*Sender Node:** PSFT\_HR
- \*Receiver Node:** EXTERNAL
- Routing Type:** Asynchronous - One Way
- Object Owner ID:** A dropdown menu.
- Active:** A checked checkbox.
- System Generated:** An unchecked checkbox.
- Save:** A yellow button.
- Return:** A yellow button.

9. Perform the following steps:
  - a. In the Sender Node field, type **PSFT\_HR**, which is the default PeopleSoft node that publishes the message.
  - b. In the Receiver Node field, type **EXTERNAL**, which is the new node that has been created to subscribe the message published by PeopleSoft.
10. Click the **Connector Properties** tab, as shown in [Figure D–37](#).

**Figure D–37 Connector Properties Tab**

Property ID	Property Name	Value
HEADER	sendUncompressed	Y
HTTPPROPERTY	Method	POST
PRIMARYURL	URL	http://bpelclient:1971

11. Perform the following steps:
  - a. In the Gateway ID field, type **LOCAL**.
  - b. In the Connector ID field, type **HTTPTARGET**.
  - c. For the PRIMARYURL value, enter the host and port number of the PeopleSoft adapter instance that is used to listen for events.
12. Click **Save** and then click **Return**.

You are returned to the Routing Definitions pane. Notice that the new routing definition (ADD\_DEPT) is now added to the list, as shown in [Figure D–38](#).

**Figure D–38 Routing Definitions Pane**

Selected	Name	Version	Routing Type	Sender Node	Receiver Node	Direction	Status
<input type="checkbox"/>	<a href="#">~GEN-UPG-24704</a>	VERSION_1	Asynch	PSFT_HR	PSFT_EP	Inbound	Inactive
<input type="checkbox"/>	<a href="#">ADD_DEPT</a>	VERSION_1	Asynch	PSFT_EP	EXTERNAL	Outbound	Active

13. Click **Save** to save the details of the Service Operation.
14. Click the **Return to Service** link to return to the Services pane.
15. Click **Save** on the Services pane.

### D.6.1.3 Activating the Gateway

To activate the gateway:

1. Select **PeopleTools, Integration Broker, Configuration**, followed by **Gateways**.
2. Search for the Integration Gateway ID, for example, **Local** and select it.

The Gateways pane is displayed, as shown in [Figure D–39](#).

**Figure D–39 Gateways Pane**

**Gateways**

Gateway ID: LOCAL

Local Gateway  Load Balancer

URL:

[Gateway Setup Properties](#)

3. Click **Ping Gateway**.

A new browser window is displayed, which shows that the gateway is active, as shown in [Figure D–40](#).

**Figure D–40 New Browser Window**

**PeopleSoft Integration Gateway**

PeopleSoft Listening Connector  
Tools Version : 8.48.02  
Status: ACTIVE

4. Close the new browser window to return to the Gateways pane, as shown in [Figure D–41](#).

**Figure D–41 Gateways Pane**

**Gateways**

Gateway ID: LOCAL

Local Gateway  Load Balancer

URL:

[Gateway Setup Properties](#)

Connectors			
*Connector ID	Description	*Connector Class Name	
1	AS2TARGET	AS2TargetConnector	<a href="#">Properties</a> + -
2	FILEOUTPUT	SimpleFileTargetConnector	<a href="#">Properties</a> + -
3	FTPTARGET	FTPTargetConnector	<a href="#">Properties</a> + -
4	GETMAILTARGET	GetMailTargetConnector	<a href="#">Properties</a> + -
5	HTTPTARGET	HttpTargetConnector	<a href="#">Properties</a> + -

5. Ensure that **HTTPTARGET** is listed in the Connectors section. If it is not listed, click **Load Gateway Connectors** to refresh the available connectors.

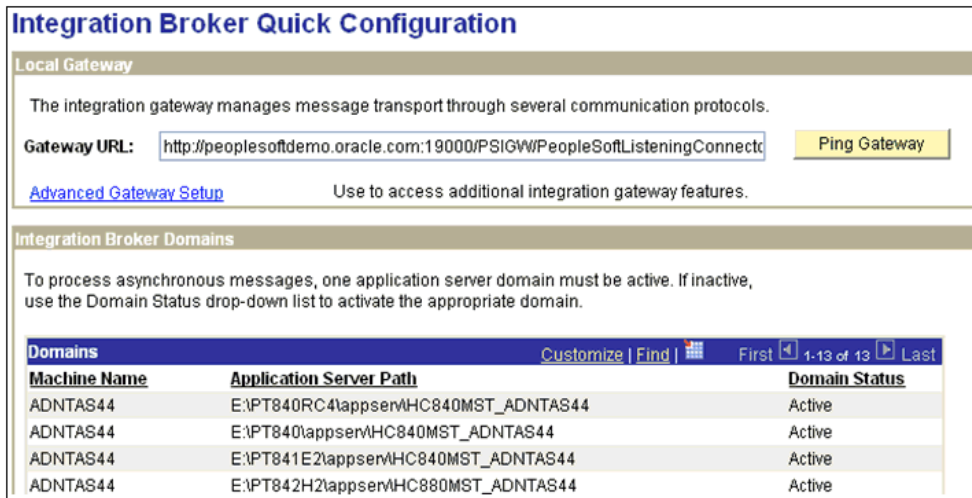
#### D.6.1.4 Activating the Domain

To activate the domain:

1. Select **PeopleTools, Integration Broker, Configuration**, followed by **Quick Configuration**.

The Quick Configuration pane is displayed, as shown in [Figure D–42](#).

**Figure D–42 Quick Configuration Pane**



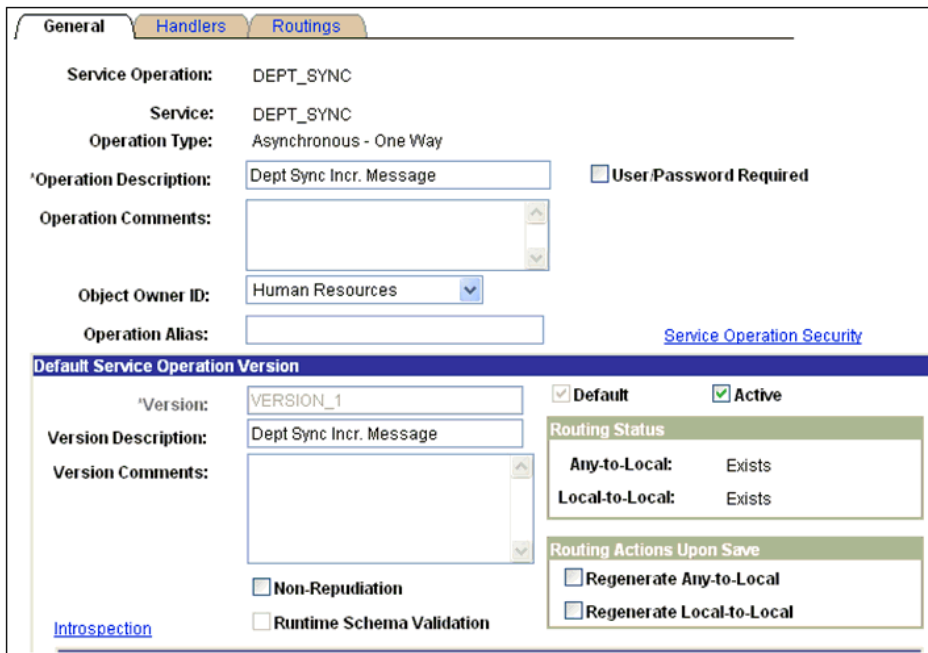
2. Ensure that the domain is active.

### D.6.1.5 Activating the Service

To activate the service:

1. Select **PeopleTools, Integration Broker, Integration Setup**, followed by **Services**.  
The Services pane is displayed, as shown in [Figure D–43](#).

**Figure D–43 Services Pane**



2. Ensure that the service is active.

### D.6.1.6 Activating the Service Operation (Queue and Message Node)

To activate the service operation:



1. Select **PeopleTools, Integration Broker, Configuration, Integration Setup**, followed by **Services**.

The Services pane is displayed, as shown in [Figure D-44](#).

**Figure D-44 Services Pane**

Existing Operations			
Customize   Find   View All   First 1-2 of 2 Last			
Operation	Description	Active	Operation Type
<a href="#">ADD_DEPT.v1</a>	Add Department	<input checked="" type="checkbox"/>	Asynch
<a href="#">DEPT_SYNC.VERSION_1</a>	Dept Sync Incr. Message	<input checked="" type="checkbox"/>	Asynch

2. From the Existing Operations section on the lower-left, click the **DEPT\_SYNC.VERSION\_1** service operation.

The Service Operation - General pane is displayed, as shown in [Figure D-45](#).

**Figure D-45 Service Operation - General Pane**

Default Service Operation Version					
*Version:	<input type="text" value="VERSION_1"/>				
Version Description:	<input type="text" value="Dept Sync Incr. Message"/>				
Version Comments:	<div style="border: 1px solid gray; height: 40px;"></div>				
<input type="checkbox"/> Non-Repudiation <input type="checkbox"/> Runtime Schema Validation					
<input checked="" type="checkbox"/> Default <input checked="" type="checkbox"/> Active					
<b>Routing Status</b> <table border="1"> <tr> <td>Any-to-Local:</td> <td>Exists</td> </tr> <tr> <td>Local-to-Local:</td> <td>Exists</td> </tr> </table>		Any-to-Local:	Exists	Local-to-Local:	Exists
Any-to-Local:	Exists				
Local-to-Local:	Exists				
<b>Routing Actions Upon Save</b> <input type="checkbox"/> Regenerate Any-to-Local <input type="checkbox"/> Regenerate Local-to-Local					
<a href="#">Introspection</a>					

3. In the message information section on the bottom of the page, note the value in the Queue Name field of the service operation, as shown in [Figure D-46](#).

**Figure D-46 Message Information Section**

Message Information	
Type:	Request
Message.Version:	<input type="text" value="DEPT_SYNC.VERSION_1"/> <a href="#">View Message</a>
*Queue Name:	<input type="text" value="ENTERPRISE_SETUP"/> <a href="#">View Queue</a> <a href="#">Add New Queue</a>

4. Click the **View Queue** link.

The Queue Definitions pane is displayed, as shown in [Figure D-47](#).

**Figure D–47 Queue Definitions Pane**

**Queue Definitions**

Queue Name: ENTERPRISE\_SETUP

Description: EC Setup Data

Comments: EC Setup Data

Archive  Unordered

Queue Status: Run

Object Owner ID: Entr Comp

Operations Assigned to Queue

Service	Version
ALTACCT_CF_FULLSYNC	VERSION_1
ALTACCT_CF_SYNC	VERSION_1
BUDGET_REF_CF_FULLSYNC	VERSION_1
BUDGET_REF_CF_SYNC	VERSION_1
BUS_UNIT_FS_FULLSYNC	VERSION_1
BUS_UNIT_FS_SYNC	VERSION_1
CHARTFIELD1_FULLSYNC	VERSION_1
CHARTFIELD1_SYNC	VERSION_1
CHARTFIELD2_FULLSYNC	VERSION_1
CHARTFIELD2_SYNC	VERSION_1

Define Partitioning Fields

Include	Field	Alias Name
<input type="checkbox"/>	OPERATIONNAME	
<input type="checkbox"/>	PUBLISHER	
<input type="checkbox"/>	PUBPROC	

Buttons: Save, Return, Add Field

5. Ensure that the Queue Status drop-down list has **Run** selected.
6. Click **Return** to return to the Service operation pane.
7. Click the **Routings** tab, as shown in [Figure D–48](#).

**Figure D–48 Routings Tab**

General | **Handlers** | **Routings**

Service Operation: DEPT\_SYNC

Default Version: VERSION\_1

Routing Name:  **ADD**

Selected	Name	Version	Routing Type	Sender Node	Receiver Node	Direction	Status
<input type="checkbox"/>	<a href="#">~GENERATED-94652240</a>	VERSION_1	Asynch	~ANY~	PSFT_HR	Inbound	Active
<input type="checkbox"/>	<a href="#">~GEN-UPG-12357</a>	VERSION_1	Asynch	PSFT_HR	PSFT_CR	Outbound	Inactive
<input type="checkbox"/>	<a href="#">ADD_DEPT</a>	VERSION_1	Asynch	PSFT_HR	EXTERNAL	Outbound	Active
<input type="checkbox"/>	<a href="#">~GEN-UPG-20085</a>	VERSION_1	Asynch	PSFT_HR	PSFT_EP	Outbound	Inactive
<input type="checkbox"/>	<a href="#">~GEN-UPG-25397</a>	VERSION_1	Asynch	PSFT_HR	PSFT_HR	Local	Inactive
<input type="checkbox"/>	<a href="#">~GEN-UPG-13180</a>	VERSION_1	Asynch	PSFT_HR	PSFT_LM	Outbound	Inactive

8. Ensure that the routings are active.
9. Click the **Return to Service** link on the bottom of the pane to return to the Services pane.
10. Click **Return to Search** to go to the Services search pane.

In the above steps, if the status has been activated in any of the panes for any of the components, then make sure to save the status at the appropriate pane and the respective components.

11. Select **PeopleTools, Integration Broker, Configuration, Integration Setup, Services**, followed by **Nodes**.

The Node Definitions pane is displayed, as shown in [Figure D–49](#).

**Figure D–49 Node Definitions Pane**

12. Ensure that the status of the External node is active.

This completes the configuration on the PeopleSoft side.

## D.6.2 Triggering the Event in PeopleSoft

This section describes how to trigger the event in PeopleSoft and consists of the following topic:

- [Section D.6.2.1, "Triggering the Event"](#)

### D.6.2.1 Triggering the Event

To trigger the event:

1. Logon to PeopleSoft using the browser-based GUI (Pure Internet Architecture).
2. Navigate to **Main Menu, Set up HRMS, Foundation Tables, followed by Departments**.

The Departments pane is displayed, as shown in [Figure D–50](#).

**Figure D–50 Departments Pane**

3. Click the **Add a New Value** tab, as shown in [Figure D-51](#).

**Figure D-51 Add a New Value Tab**

The screenshot shows a web form titled "Departments". At the top, there are two tabs: "Find an Existing Value" (highlighted in blue) and "Add a New Value" (highlighted in yellow). Below the tabs, there are two input fields: "SetID:" with the value "SHARE" and a magnifying glass icon, and "Department:" with the value "BPELTEST01" and a magnifying glass icon. At the bottom left, there is a yellow "Add" button.

4. Enter the appropriate value in the SetID and Department fields
5. Click **Add**.

The Department Profile tab is displayed, as shown in [Figure D-52](#).

**Figure D-52 Department Profile Tab**

The screenshot shows a web form titled "Department Profile" with a sub-tab "Comm. Acctg. and EG". At the top, it displays "SetID: SHARE" and "Department: BPELTEST01". Below this is a blue header bar with "Department Profile" and navigation links like "Find | View All", "First", "1 of 1", and "Last". The main form area contains several fields: "Effective Date:" (08/25/2008), "Description:" (BPEL Inbound Test), "Status:" (Active), "Location SetID:", "Location:", "Company:", "Manager Type" (with radio buttons for None, EmpIID, and Position), "Budget Year End Date:", "Budget Level:" (None), and "Tax Location:". There are also magnifying glass icons for several fields.

6. Provide the necessary information according to your requirements to create a new Department record.
7. Click **Save** at the bottom of the pane when you are finished.

### D.6.3 Verifying the Event Results

This section describes how to verify the event results and consists of the following topic:

- [Section D.6.3.1, "Verifying the Results"](#)

### D.6.3.1 Verifying the Results

To verify the results:

1. Logon to PeopleSoft using the browser-based GUI (Pure Internet Architecture).
2. Navigate to **Main Menu, PeopleTools, Integration Broker, Service Operation Monitor**, followed by **Asynchronous Services**.

The Monitor Overview tab is displayed, as shown in [Figure D-53](#).

**Figure D-53 Monitor Overview Tab**

Queue Name	Error	New	Started	Working	Done	Retry	Timeout	Edited	Canceled	Hold
ENTERPRISE_SETUP	0	0	0	0	5	0	0	0	0	0
PSRF_REPORTING_FOLDERS	0	0	0	0	1366	0	0	0	0	0
PSXP_MSG_CHNL	0	0	0	0	1	0	0	0	0	0
USER_PROFILE	0	0	0	0	1	0	0	0	0	0

Here you can view the results of the events at the queue level.

3. Click the number link (5) queue (Enterprise\_Setup) in the Result section.

The Operation Instances tab is displayed, as shown in [Figure D-54](#).

**Figure D-54 Operation Instances Tab**

Select	Transaction ID	Queue Name	Publishing Node	Sub Queue	Status	Time Stamp
<input type="checkbox"/>	b727c644-72d0-11dd-9dd0-b53768245214	ENTERPRISE_SETUP	PSFT_HR	<a href="#">Sub Queue Link</a>	Done	08/25/2008 2:07:48PM

Notice that a new record for the event (department) is available.

4. Click the **Details** link on the lower-right.

A new browser window (Asynchronous Details) is displayed, as shown in [Figure D-55](#).

**Figure D–55 Asynchronous Details**

### Asynchronous Details

<b>Transaction ID</b>	b727c644-72d0-11dd-9dd0-b53768245214		
<b>External Service Name</b>	DEPT_SYNC.VERSION_1		
<b>Publishing Node</b>	PSFT_HR	<b>'Segment'</b>	1 <input type="button" value="Refresh"/>
<b>Queue Name</b>	ENTERPRISE_SETUP		ViewXML
<b>Queue Sequence ID</b>	5		
<b>Sub Queue</b>			
<b>Original Pub Node</b>	PSFT_HR	<b>Uncompressed Data Length</b>	7526
<b>Status</b>	DONE	<b>Data Length View Limit</b>	100000

View IB Info

Publication Contracts						
						Customize   Find   First 1 of 1 Last
Actions Information						
Subscriber Node	'Segment'	Status				
EXTERNAL	1	Done	View XML	<input type="button" value="Resubmit"/>	<input type="button" value="Cancel"/>	Error Messages View IB Info

Notice that the status of the transaction is **DONE** for the Publishing and Subscriber nodes.

---

---

# Glossary

**adapter**

Provides universal connectivity by enabling an electronic interface to be accommodated (without loss of function) to another electronic interface.

**agent**

Supports service protocols in listeners and documents.

**business service**

Also known as a Web service. A Web service is a self-contained, modularized function that can be published and accessed across a network using open standards. It is the implementation of an interface by a component and is an executable entity.

**channel**

Represents configured connections to particular instances of back-end systems. A channel binds one or more event ports to a particular listener managed by an adapter.

**listener**

A component that accepts requests from client applications.

**port**

Associates a particular business object exposed by the adapter with a particular disposition. A disposition is a URL that defines the protocol and location of the event data. The port defines the end point of the event consumption.





## A

---

access methods, 1-2  
adapter configuration  
  updating, 3-2  
adapter exceptions, 10-2 to 10-3  
adapter types  
  resource, 1-1, 3-1 to 3-2  
adapters, 1-1 to 1-2  
  configuring, 2-1 to 2-22, 3-2  
  deploying, 1-1, 1-4, 3-1 to 3-2, 4-2  
  integrating with BPEL Process Manager, 4-1, 6-1  
  troubleshooting, 10-1 to 10-7  
Adapters node, 4-36  
Add Channel dialog box, 2-17  
Advanced tab, 2-21  
ae batch script, 10-1  
API Java programs  
  compiling, A-5 to A-7  
  creating, A-1 to A-5  
application adapters, 1-1 to 1-2  
  configuring, 2-1 to 2-22, 3-2  
  deploying, 1-1, 1-4, 3-1 to 3-2, 4-2  
  integrating with BPEL Process Manager, 4-1, 6-1  
  troubleshooting, 10-1 to 10-7  
Application Explorer, 1-2 to 1-4, 2-5, 2-9, 4-7, 4-34  
  application systems and, 2-5  
  debugging and, 10-1 to 10-2  
  J2CA configuration and, 3-2  
  OracleAS Adapter J2CA and, 10-4  
  testing and, 10-1 to 10-2  
  troubleshooting, 10-1 to 10-2  
  WSDL files and, 4-34  
Application Messaging, 1-2  
Application Messaging Manager, 1-3  
Application Server dialog box, 2-6  
Application Server parameter, 2-6  
application system objects, 2-9  
application systems  
  Application Explorer and, 2-5  
  supported, 2-5  
asynchronous support, 1-2

## B

---

BPEL Designer, 4-1, 6-1

BPEL Process Manager  
  integrating with adapters, 4-1, 6-1  
  OracleAS Adapter for PeopleSoft and, 4-1  
browsing metadata, 2-9  
BSE (OracleAS Adapter Business Services Engine), 2-2  
  configuring, 2-2  
  troubleshooting, 10-4  
BSE (OracleWLS Adapter Business Services Engine), 1-1, 1-4  
BSE deployment, 1-4  
BSE URL field, 2-3  
business events, 1-2  
business functions, 2-12  
business objects, 2-5  
  browsing, 2-9  
business processes, 1-3  
business services  
  creating, 2-12 to 2-13  
  testing, 2-13

## C

---

CCI calls, 3-1  
channel configuration parameters, 2-19  
  Error Directory, 2-21  
  File Mask, 2-21  
  Host, 2-19  
  Is Keep Alive, 2-19  
  Is Length Prefix, 2-19  
  Is XML, 2-19  
  Poll interval, 2-21  
  Polling Location, 2-21  
  Port, 2-16, 4-35  
  Port Number, 2-19  
  Processing Mode, 2-21  
  Response/Ack Directory, 2-21  
  Server port, 2-16, 4-35  
  Synchronization Type, 2-19 to 2-21  
  Thread limit, 2-21  
channel types  
  File, 2-20 to 2-22  
channels, 2-14 to 2-22  
  creating, 2-14 to 2-22, 4-34 to 4-36  
  deleting, 2-22  
  editing, 2-22

- ports and, 2-18
- starting, 2-20, 4-36
- stopping, 2-20, 4-36
- testing and debugging, 4-36
- channels. *See also* listeners
- closing connections, 2-8
- component interface API, 1-3
- component interface properties, 10-2
- Component Interface testing tool
  - three-tier mode and, 10-3
- component interfaces, 1-2 to 1-3, 2-9
  - creating, C-1 to C-4
  - security and, C-5 to C-14
  - testing, C-14 to C-19
- component methods, 1-3
- configuration parameters, 3-4
  - IWayConfig, 3-4
  - IWayHome, 3-4
  - IWayRepoPassword, 3-4
  - IWayRepoURL, 3-4
  - IWayRepoUser, 3-4
  - Loglevel, 3-4
- configurations
  - connecting to, 2-4
  - defining, 2-2 to 2-4
  - overwriting, 3-4
  - testing, B-15 to B-17
- Configurations node, 2-3
- configuring adapters, 3-2
- configuring File Output Connector, D-5 to D-7
- configuring Integration Broker, D-2
- configuring Integration Gateway, D-5 to D-7
- configuring IntegrationGateway.properties
  - file, D-4 to D-5
- configuring message channels, D-16 to D-19
- configuring message routers, B-1 to B-15
- configuring nodes
  - Gateway, D-7 to D-11
  - TCP/IP84 connector, B-3 to B-6
- configuring target connectors, B-1 to B-15
- connecting to OracleAS Adapter J2CA, 10-4
- connecting to PeopleSoft, 2-5 to 2-7, 4-36, 10-2, 10-6
- Connection dialog box, 4-36
- connection parameters, 2-6 to 2-7, 4-35 to 4-36, 10-2
  - Application Server, 2-6
  - Hostname, 2-3, 10-2 to 10-3
  - Password, 2-6
  - Port, 2-6, 2-16, 4-35, 10-2 to 10-3
  - User, 2-6
- connections
  - closing, 2-8
  - deleting, 2-9
  - establishing, 2-5 to 2-7, 4-36
- Connector
  - deploying to Oracle Application Server, 3-1
- connector factories, 3-2
- connector factory objects, 3-2
  - multiple, 3-4
- connectors
  - configuring, B-1 to B-15

- Create Business Service dialog box, 2-12
- Create Outbound JCA Service
  - (Request/Response), 4-8
- creating channels, 2-14 to 2-22, 4-34 to 4-36
- creating events, 4-36 to 4-38
- creating repository projects, 2-2 to 2-4
- creating schemas, 2-9 to 2-10, 10-2
- creating Web services, 2-12 to 2-13

## D

---

- database connections
  - opening, 3-4
- databases
  - connecting to, 3-4
  - Oracle, 3-4
- deleting channels, 2-22
- deleting connections, 2-9
- deploying adapters, 3-1 to 3-2, 4-2
- deployments
  - BSE, 1-4
- Description field, 2-12
- design time, 4-34
  - configuring, 4-7, 4-34
- disconnecting from PeopleSoft targets, 2-8

## E

---

- Eclipse. *See* JDeveloper
- Eclipse. *See* JDeveloper
- editing channels, 2-22
- editing targets, 2-8
- EIP (Enterprise Integration Points), 1-3
- EJB (Enterprise Java Beans), 3-1
- Enterprise Connector for J2EE Connector Architecture (J2CA), 1-4
- Enterprise Integration Points (EIP), 1-3
- Enterprise Java Beans (EJB), 3-1
- Error Directory parameter, 2-21
- error messages, 10-2 to 10-7
  - target systems and, 10-5
- event integration
  - verifying, 4-45
- event messages, 4-34
- event ports
  - creating, 4-36 to 4-38
- events, 1-2, 1-3
  - creating, 4-36 to 4-38
  - triggering, 4-45
- Existing Service Names list, 2-12
- Export WSDL dialog box, 4-36

## F

---

- fault code elements, 10-5
- fault string elements, 10-5
- File channel, 2-20 to 2-22
- File Mask parameter, 2-21
- File Output Connector
  - configuring, D-5 to D-7

## G

---

Gateway node, D-7 to D-11  
generating schemas, 2-9 to 2-10, 10-2

## H

---

Host parameter, 2-19  
Hostname parameter, 2-3, 10-2 to 10-3  
HTTP protocol, 2-18

## I

---

inbound J2CA services, 4-34  
installation directories, 3-4  
Integration Broker, 1-2 to 1-3, D-1 to D-27  
    configuring, D-2  
Integration Gateway  
    configuring, D-5 to D-7  
IntegrationGateway.properties file  
    configuring, D-4 to D-5  
Is Keep Alive parameter, 2-19  
Is Length Prefix parameter, 2-19  
Is XML parameter, 2-19  
IWayConfig parameter, 3-4  
IWayHome parameter, 3-4  
IWayRepoPassword parameter, 3-4  
IWayRepoURL parameter, 3-4  
IWayRepoUser parameter, 3-4  
iwpsci84.jar file, 10-2  
iwpsci8x.jar file, 10-2

## J

---

J2CA (Enterprise Connector for J2EE Connector  
    Architecture), 1-4  
    OracleAS Adapter and, 3-1  
J2CA configuration  
    Application Explorer and, 3-2  
J2CA resource adapters, 1-1, 3-1  
J2CA services, 4-34  
JAR files, 10-2  
    errors and, 10-2  
Java program clients, 3-1  
JDeveloper, 4-1, 6-1

## L

---

License and Method dialog box, 2-12  
License field, 2-12  
licenses, 2-12  
list of nodes, 10-2  
listeners, 1-5, B-10  
log files, 10-1  
log levels  
    overwriting, 3-4  
Loglevel parameter, 3-4

## M

---

managed connector factories, 3-2

managed connector factory objects, 3-2  
    multiple, 3-4  
ManagedConnectionFactory parameter, 3-4  
Mediator Inbound Process, 5-11  
Mediator Outbound Process, 5-2  
message channels  
    configuring, D-16 to D-19  
message routers  
    configuring, B-1 to B-15  
message types  
    event, 4-34  
messages, 1-1, 1-3  
metadata, 2-9  
    browsing, 2-9  
Method Name field, 2-12  
methods  
    changing, C-4  
    viewing, C-4  
MySAP Easy Access window, 4-45  
MySAP Workbench, 4-45

## N

---

New Configuration dialog box, 2-3 to 2-4  
Node list, 10-2  
nodes  
    Adapters, 4-36  
    Configurations, 2-3  
    connected, 4-36  
    Gateway, D-7 to D-11  
    PeopleSoft, 2-6, 4-36  
    Service Adapters, 2-6  
    TCP/IP84 connector, B-3 to B-6

## O

---

OC4J-ra.xml file, 3-2 to 3-4  
Oracle Application Server  
    deployment of Connector to, 3-1  
Oracle databases, 3-4  
Oracle JDeveloper, 4-1, 6-1  
Oracle JDeveloper BPEL Designer. *See* BPEL Designer,  
    JDeveloper, or Oracle JDeveloper  
Oracle's Unified Method (OUM), xi  
OracleAS Adapter  
    installation directory and, 3-4  
    J2CA and, 3-1  
OracleAS Adapter Business Services Engine  
    (BSE), 2-2  
    configuring, 2-2  
    troubleshooting, 10-4  
OracleAS Adapter for PeopleSoft  
    BPEL Process Manager and, 4-1  
    deploying, 4-2  
    troubleshooting, 10-1 to 10-7  
OracleAS Adapter J2CA, 2-2 to 2-3  
    Application Explorer and, 10-4  
    connecting to, 10-4  
OracleWLS Adapter Application Explorer. *See also*  
    Application Explorer

OracleWLS Adapter Business Services Engine (BSE), 1-1, 1-4  
OracleWLS Application Adapter for PeopleSoft  
  configuring, 2-1 to 2-22  
  deploying, 1-1

## P

---

parameter types  
  channel configuration, 2-19  
  configuration, 3-4  
  connection, 2-6, 4-35 to 4-36, 10-2  
Password parameter, 2-6  
passwords, 3-4  
PeopleSoft  
  connecting to, 2-5 to 2-7, 10-2, 10-6  
PeopleSoft API Java programs  
  compiling, A-5 to A-7  
  creating, A-1 to A-5  
PeopleSoft business objects, 2-5  
PeopleSoft Java API, 1-2  
PeopleSoft node, 2-6, 4-36  
PeopleSoft XML, 1-2  
Poll interval parameter, 2-21  
Polling Location parameter, 2-21  
Port Number parameter, 2-3, 2-19  
Port parameter, 2-6, 2-16, 4-35, 10-2 to 10-3  
ports  
  channels and, 2-18  
  creating, 4-36 to 4-38  
Process Manager. *See* BPEL Process Manager  
Processing Mode parameter, 2-21  
properties, 3-4  
Protocol list, 2-18  
psjoa.jar file, 10-2

## R

---

repository projects  
  creating, 2-2 to 2-4  
  Web services and, 2-2  
request documents, 2-9  
Request Schema tab, 2-10  
Request tab, 2-21  
resource adapters, 1-1, 3-1 to 3-2  
response documents, 2-9  
Response Schema tab, 2-10  
Response tab, 2-21  
Response/Ack Directory parameter, 2-21

## S

---

schemas, 2-9  
  creating, 1-3, 2-9 to 2-10, 10-2  
security  
  component interfaces and, C-5 to C-14  
Server port parameter, 2-16, 4-35  
Service Adapters node, 2-6  
service names, 2-12  
Service Provider list, 2-3 to 2-4  
Service-Oriented Architecture (SOA), 4-1

services, 1-2, 1-3  
  creating, 2-12 to 2-13  
  testing, 2-13  
servlets, 3-1  
SOA (Service-Oriented Architecture), 4-1  
SOAP agents, 10-5  
SOAP faults, 10-5  
SOAP requests, 10-5 to 10-7  
  errors and, 10-5 to 10-7  
SOAP responses, 10-5 to 10-7  
starting a channel, 2-20  
stopping a channel, 2-20  
Synchronization Type parameter, 2-19 to 2-21  
synchronous support, 1-2  
system objects, 2-9

## T

---

target connectors  
  configuring, B-1 to B-15  
target systems  
  errors and, 10-5  
targets  
  connecting to, 2-5 to 2-7, 4-36, 10-6  
  defining, 2-5 to 2-7  
  deleting, 2-9  
  disconnecting from, 2-8  
  editing, 2-8  
TCP Listener dialog box, 2-18  
TCP protocol, 2-18  
TCP/IP84 connector node, B-3 to B-6  
testing component interfaces, C-14 to C-19  
testing configurations, B-15 to B-17  
testing Web services, 2-13  
Thread limit parameter, 2-21  
trace information, 10-1  
transaction processing, 1-2  
triggering events in MySAP ERP, 4-45  
troubleshooting, 10-1 to 10-7  
  Application Explorer, 10-1 to 10-2  
  BSE, 10-4  
  Web services, 10-4 to 10-7

## U

---

updating adapter configuration, 3-2  
User ID parameter, 2-6

## V

---

viewing methods, C-4

## W

---

Web Service Definition Language (WSDL), 2-10 to 2-11  
Web service names, 2-12  
Web services, 1-1 to 1-3, 2-2  
  creating, 2-12 to 2-13  
  deploying, 10-6  
  integrating, 4-1

- repository projects and, 2-2
- testing, 2-13
- troubleshooting, 10-4 to 10-7
- WSDL (Web Service Definition Language), 2-10 to 2-11
- WSDL documents, 4-1
- WSDL files, 4-1, 6-1
  - Application Explorer and, 4-34
  - creating, 4-34

## **X**

---

- XML messages, 1-1
- XML schemas, 1-2, 2-1, 2-9
  - creating, 2-9 to 2-10

