

**Oracle® WebLogic Server**

Application Adapter for PeopleSoft User's Guide

10g Release 3 (10.3.1.0)

**E15302-01**

June 2009

Oracle WebLogic Server Application Adapter for PeopleSoft User's Guide, 10g Release 3 (10.3.1.0)

E15302-01

Copyright © 2008, 2009, Oracle. All rights reserved.

Primary Author: Stefan Kostial

Contributing Authors: Sheela Vasudevan, Sunil Gopal, Marian Jones, Vikas Anand, Sunil Wadhwa, Vishal Saxena

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States 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, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.





---

---

# Contents

<b>Preface</b> .....	ix
Audience .....	ix
Documentation Accessibility .....	ix
Related Documents .....	x
Conventions .....	x
Help Us to Serve You Better .....	xiii
<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 .....	1-2
<b>Integration with PeopleSoft</b> .....	1-3
<b>Adapter Architecture</b> .....	1-3
<b>BSE Versus OracleWLS Adapter J2CA Deployment</b> .....	1-5
<b>2 Configuring OracleWLS Application Adapter for PeopleSoft</b>	
<b>Starting Application Explorer</b> .....	2-1
<b>Configuring Settings for BSE or J2CA</b> .....	2-2
Configuring BSE .....	2-2
Configuring BSE System Settings .....	2-3
Configuring J2CA .....	2-6
Password Encryption .....	2-7
<b>Creating a Repository Configuration</b> .....	2-8
Creating a Configuration for BSE .....	2-8
Creating a Configuration for J2CA .....	2-9
Connecting to a BSE or J2CA Configuration .....	2-10
<b>Establishing a Connection (Target) for PeopleSoft</b> .....	2-11
Defining a Target to PeopleSoft .....	2-11
Connecting to a Defined PeopleSoft Target .....	2-12
Managing a Target .....	2-13
<b>Viewing Application System Objects</b> .....	2-14
<b>Creating XML Schemas</b> .....	2-15
<b>Generating a WSDL (J2CA Configurations Only)</b> .....	2-15
<b>Publishing a WSDL</b> .....	2-16

Creating and Testing a Web Service (BSE Configurations Only) .....	2-18
<b>Configuring an Event Adapter .....</b>	<b>2-20</b>
Creating and Editing a Channel.....	2-20
Creating an HTTP Channel .....	2-21
Creating a TCP Channel.....	2-23
Creating a File Channel.....	2-25
Editing a Channel.....	2-27
Deleting a Channel .....	2-27

### **3 Oracle WebLogic Server Deployment and Integration**

Adapter Integration with OracleWLS .....	3-1
Deployment of Adapter .....	3-1
Updating Adapter Configuration.....	3-2
How to Write a Java Application Client Using the CCI API.....	3-4

### **4 Configuring Outbound Processing Using Oracle Service Bus (BSE Configuration)**

Overview of Application Adapter Integration with Oracle Service Bus .....	4-1
Publishing a WSDL From Application Explorer to Oracle Service Bus.....	4-1
Starting Oracle Service Bus and Creating Project Folders .....	4-3
Configuring a File Type Business Service.....	4-6
Configuring a WSDL Type Business Service .....	4-9
Configuring a Proxy Service .....	4-13
Configuring a Pipeline.....	4-16

### **5 Configuring Inbound and Outbound Processing Using Oracle Service Bus (J2CA Configuration)**

Overview of Application Adapter Integration with Oracle Service Bus .....	5-1
<b>Configuring Inbound Processing Using Oracle Service Bus (J2CA Configuration) .....</b>	<b>5-1</b>
Creating a Channel and Publishing a WSDL From Application Explorer to Oracle Service Bus..	5-2
Starting Oracle Service Bus and Creating Project Folders .....	5-5
Configuring a File Type Business Service .....	5-7
Configuring a Proxy Service.....	5-10
Configuring a Pipeline .....	5-15
<b>Configuring Outbound Processing Using Oracle Service Bus (J2CA Configuration) .....</b>	<b>5-18</b>
Publishing a WSDL From Application Explorer to Oracle Service Bus.....	5-18
Starting Oracle Service Bus and Creating Project Folders .....	5-20
Configuring a File Type Business Service .....	5-22
Configuring a WSDL Type Business Service .....	5-25
Configuring a Proxy Service.....	5-29
Configuring a Pipeline .....	5-31

### **6 Troubleshooting and Error Messages**

Troubleshooting.....	6-1
BSE Error Messages .....	6-4

General Error Handling in BSE.....	6-5
Adapter-Specific Error Handling.....	6-5
<b>7 Advanced User Tools</b>	
<b>Web Services Policy-Based Security</b> .....	7-1
Configuring Web Services Policy-Based Security .....	7-2
<b>Migrating Repositories</b> .....	7-8
<b>A Generating Component Interface APIs</b>	
<b>Building the PeopleSoft API Java Programs</b> .....	A-1
Compiling the PeopleSoft API Java Programs.....	A-4
<b>B Configuring the PeopleSoft Message Router</b>	
<b>Configuring the TCP/IP or HTTP Target Connector for PeopleSoft 8.4</b> .....	B-1
Configuring the TCP/IP Target Connector.....	B-2
Configuring the Node for the TCP/IP84 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
Testing Your PeopleSoft Configuration.....	B-14
<b>C Using Component Interfaces</b>	
<b>Creating a Component Interface</b> .....	C-1
Working With Properties .....	C-1
Creating a New Component Interface .....	C-1
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-12
<b>D Using PeopleSoft Integration Broker</b>	
<b>PeopleSoft Integration Broker</b> .....	D-1
Configuring Integration Broker in PeopleSoft 8.4.....	D-2
<b>Configuring Application Messaging in PeopleSoft Release 8.1</b> .....	D-11
<b>Viewing the PeopleCode for a Message</b> .....	D-20
<b>Testing the Integration Broker</b> .....	D-21
<b>Using Outbound Synchronous Messages</b> .....	D-23
Configuring Outbound Synchronous Messages .....	D-23
Viewing the PeopleCode for a Synchronous Message .....	D-25
<b>Generating Events Using PeopleTools Tutorial</b> .....	D-26
Configuring PeopleSoft Services.....	D-26
Triggering the Event in PeopleSoft.....	D-34
Verifying the Event Results .....	D-35

**Glossary**

**Index**

---

---

# Preface

This Preface contains these topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions](#)
- [Help Us to Serve You Better](#)

## Audience

*Oracle WebLogic Server Application Adapter for PeopleSoft User's Guide* is intended for those who perform the following tasks:

- Install applications
- Maintain applications

To use this document, you need to know how to install and configure Oracle Service Bus (Business Service and Proxy Service).

## Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

### Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

## Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

## TTY Access to Oracle Support Services

To reach AT&T Customer Assistants, dial 711 or 1.800.855.2880. An AT&T Customer Assistant will relay information between the customer and Oracle Support Services at 1.800.223.1711. Complete instructions for using the AT&T relay services are available at <http://www.consumer.att.com/relay/tty/standard2.html>. After the AT&T Customer Assistant contacts Oracle Support Services, an Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process.

## Related Documents

For more information, refer to these Oracle resources:

- *Oracle WebLogic Server Adapter Concepts*
- *Oracle WebLogic Server Application Adapters Installation Guide*

Printed documentation is available for sale in the Oracle Store at

<http://oraclestore.oracle.com/>

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

<http://www.oracle.com/technology/membership/>

If you already have a user name and password for OTN, then you can go directly to the documentation section of the OTN Web site at

<http://www.oracle.com/technology/documentation/>

## Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- [Conventions in Text](#)
- [Conventions in Code Examples](#)
- [Conventions for Windows Operating Systems](#)

### Conventions in Text

We use the following conventions in text to help you more quickly identify special terms. The table also provides examples of their use.

Convention	Meaning	Example
<b>Bold</b>	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an <b>index-organized table</b> .

Convention	Meaning	Example
<i>Italics</i>	Italic typeface indicates book titles or emphasis.	<i>Oracle Database 10g Concepts</i> Ensure that the recovery catalog and target database do <i>not</i> reside on the same disk.
UPPERCASE monospace (fixed-width) font	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, Recovery Manager keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, and system-supplied column names, database objects and structures, user names, and roles.	You can specify this clause only for a NUMBER column. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. Use the DBMS_STATS.GENERATE_STATS procedure.
lowercase monospace (fixed-width) font	Lowercase monospace typeface indicates executable programs, filenames, directory names, and sample user-supplied elements.  <i>Note:</i> Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter sqlplus to start SQL*Plus. The password is specified in the orapwd file. Back up the datafiles and control files in the /disk1/oracle/dbs directory. The department_id, department_name, and location_id columns are in the hr.departments table. Connect as oe user. The JRepUtil class implements these methods.
lowercase italic monospace (fixed-width) font	Lowercase italic monospace font represents placeholders or variables.	You can specify the <i>parallel_clause</i> . Run <i>old_release.SQL</i> where <i>old_release</i> refers to the release you installed before upgrading.

## Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL\*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
[ ]	Anything enclosed in brackets is optional.	DECIMAL ( <i>digits</i> [ , <i>precision</i> ])
{ }	Braces are used for grouping items.	{ENABLE   DISABLE}
	A vertical bar represents a choice of two options.	{ENABLE   DISABLE} [COMPRESS   NOCOMPRESS]
...	Ellipsis points mean repetition in syntax descriptions.  In addition, ellipsis points can mean an omission in code examples or text.	CREATE TABLE ... AS <i>subquery</i> ;  SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM employees;
Other symbols	You must use symbols other than brackets ([ ]), braces ({ }), vertical bars ( ), and ellipsis points (...) exactly as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;

Convention	Meaning	Example
<i>Italics</i>	Italicized text indicates placeholders or variables for which you must provide particular values.	CONNECT SYSTEM/ <i>system_password</i> DB_NAME = <i>database_name</i>
UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. Because these terms are not case sensitive, you can use them in either UPPERCASE or lowercase.	SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;
lowercase	Lowercase typeface indicates user-defined programmatic elements, such as names of tables, columns, or files.  <b>Note:</b> Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	SELECT last_name, employee_id FROM employees; sqlplus hr CREATE USER mjones IDENTIFIED BY ty3MU9;

## Conventions for Windows Operating Systems

The following table describes conventions for Windows operating systems and provides examples of their use.

Convention	Meaning	Example
Click <b>Start</b> , and then choose the <i>menu item</i>	How to start a program.	To start the Database Configuration Assistant, click <b>Start</b> , and choose <b>Programs</b> . In the Programs menu, choose <b>Oracle - HOME_NAME</b> and then click <b>Configuration and Migration Tools</b> . Choose <b>Database Configuration Assistant</b> .
File and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe ( ), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the filename begins with \\, then Windows assumes it uses the Universal Naming Convention.	c:\winnt\\"system32 is the same as C:\WINNT\SYSTEM32
C:\>	Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the <i>command prompt</i> in this manual.	C:\oracle\oradata>
Special characters	The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.	C:\>exp HR/HR TABLES=employees QUERY=\"WHERE job_id='SA_REP' and salary<8000\"

Convention	Meaning	Example
<i>HOME_NAME</i>	Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.	<code>C:\&gt; net start OracleHOME_NAME_TNSListener</code>
<i>ORACLE_HOME</i> and <i>ORACLE_BASE</i>	<p>In Oracle8i release 8.1.3 and lower, when you installed Oracle components, all subdirectories were located under a top level <i>ORACLE_HOME</i> directory.</p> <p>This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level <i>ORACLE_HOME</i> directory. There is a top level directory called <i>ORACLE_BASE</i> that by default is <code>C:\oracle\product\10.1.0</code>. If you install the latest Oracle release on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is <code>C:\oracle\product\10.1.0\db_n</code>, where <i>n</i> is the latest Oracle home number. The Oracle home directory is located directly under <i>ORACLE_BASE</i>.</p> <p>All directory path examples in this guide follow OFA conventions.</p> <p>Refer to <i>Oracle Database Installation Guide for Windows</i> for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.</p>	Change to the <i>ORACLE_BASE\ORACLE_HOME\rdbms\admin</i> directory.

## Help Us to Serve You Better

To help our consultants answer your questions effectively, please be prepared to provide specifications and sample files and to answer questions about errors and problems.

The following list includes the specifications our consultants require.

- **Platform:**
- **Operating System:**
- **Operating System Version:**
- **Product List:**
- **Adapters:**
- **Adapter Deployment:**  
For example, *J2CA* or *Business Services Engine (BSE)*
- **Container Version:**

The following table lists components. Specify the version in the column provided.

Component	Version
Adapter	
EIS (DBMS/APP)	
HOTFIX/Service Pack	

In the following table, specify the JVM version and vendor.

JVM Version	Vendor
-------------	--------

The following table lists additional questions to help us serve you better.

Request/Question	Error/Problem Details or Information
Provide usage scenarios or summarize the application that produces the problem.	
Has this happened previously?	
Can you reproduce this problem consistently?	
Any <b>change in the application environment</b> : software configuration, EIS/database configuration, application, and so on?	
Under what circumstance does the problem <i>not</i> occur?	
Describe the <b>steps</b> to reproduce the problem.	
Describe the <b>problem</b> .	
Specify the <b>error</b> message(s).	

The following is a list of error or problem files that might be applicable.

- XML schema
- XML instances
- Other input documents (transformation)
- Error screen shots
- Error output files
- Trace and log files
- Log transaction

---

---

# Introduction

Oracle WebLogic Server connects to a PeopleSoft system through Oracle WebLogic Server Application Adapter for PeopleSoft (OracleWLS Application Adapter for PeopleSoft). OracleWLS Application Adapter for PeopleSoft provides connectivity and carries out interactions on a PeopleSoft system. This chapter discusses the following topics:

- [Adapter Features](#)
- [PeopleSoft Concepts](#)
- [Integration with PeopleSoft](#)
- [Adapter Architecture](#)
- [BSE Versus OracleWLS Adapter J2CA Deployment](#)

## Adapter Features

OracleWLS 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. OracleWLS Application Adapter for PeopleSoft can be deployed as a J2EE Connector Architecture (J2CA) version 1.0 resource adapter. This deployment is referred to as OracleWLS Adapter J2CA. It can also be deployed as a Web services servlet and is referred to as OracleWLS Adapter Business Services Engine(BSE).

OracleWLS 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.

OracleWLS 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*

## 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. OracleWLS 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 OracleWLS Application Adapter for PeopleSoft include:

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

## 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. OracleWLS Application Adapter for PeopleSoft supports both types of component interfaces.

## PeopleSoft Application Messaging / Integration Broker

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

## Integration with PeopleSoft

OracleWLS 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, create, secure, and test one. If the component interface exists but you modified it, 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.

See [Chapter 2, "Configuring OracleWLS Application Adapter for PeopleSoft"](#) for more information.

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*
- *Oracle Application Server Adapters Installation Guide*

## Adapter Architecture

OracleWLS 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.

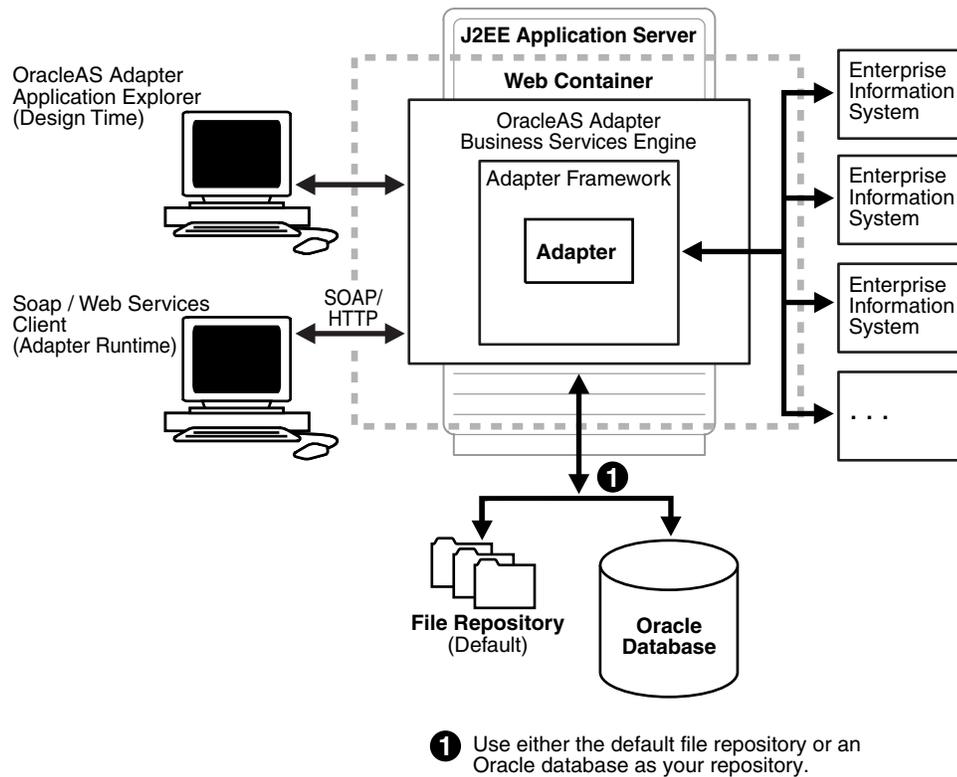
### Oracle WebLogic Server Adapter Business Services Engine (BSE) Architecture

[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**




---

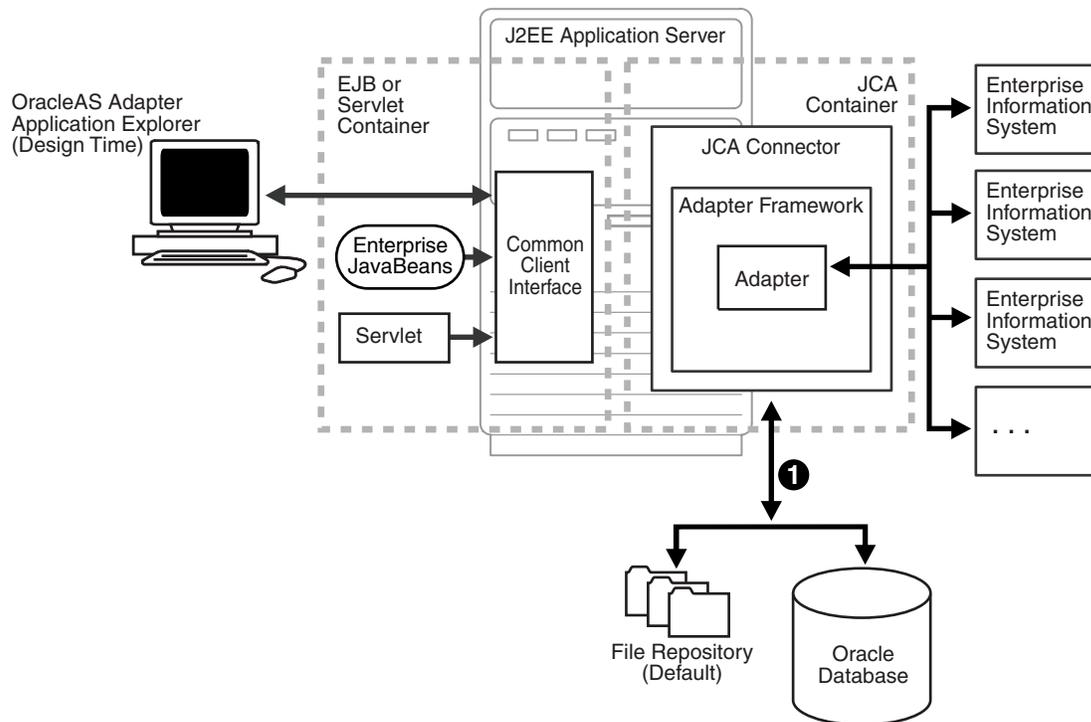
**Note:** Do not use a file repository for BSE in production environments.

---

**Oracle WebLogic Server Adapter J2CA Architecture**

Figure 1–2 shows the generic architecture for OracleWLS Adapter J2CA for packaged applications. OracleWLS 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`. See Chapter 3, "Oracle WebLogic Server Deployment and Integration" for more information.

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** Use either the default file repository or an Oracle database as your repository.

## BSE Versus OracleWLS Adapter J2CA Deployment

If you are using OracleWLS Application Adapter for PeopleSoft with Oracle Service Bus (OSB), please note that:

- Only OracleWLS Adapter J2CA deployment supports inbound integration (event notification) with OSB.
- Both OracleWLS Adapter J2CA and BSE deployments support outbound integration (request-response service) with OSB.

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

1. BSE is the preferred deployment option because it:
  - Can be deployed in a separate instance of Oracle WebLogic Server.
  - Provides better distribution of load.
  - Provides better isolation from any errors from third party libraries.
  - Provides better capability to isolate issues for debugging purposes.
  - Conforms more closely to the Service Oriented Architecture (SOA) model for building applications.
2. OracleWLS Adapter J2CA provides slightly better performance.

OracleWLS Adapter J2CA does provide slightly better performance than BSE. However, the difference decreases as the transaction rate increases.

3. OracleWLS Adapter J2CA and the BSE option both provide identity propagation at run-time.

The BSE option provides the capability to pass identity using the SOAP header. For OracleWLS Adapter J2CA, user name and password can be passed using the connection specification of the CCI.

---

---

# Configuring OracleWLS Application Adapter for PeopleSoft

OracleWLS 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 discusses the following topics:

- [Starting Application Explorer](#)
- [Configuring Settings for BSE or J2CA](#)
- [Creating a Repository Configuration](#)
- [Establishing a Connection \(Target\) for PeopleSoft](#)
- [Viewing Application System Objects](#)
- [Creating XML Schemas](#)
- [Generating a WSDL \(J2CA Configurations Only\)](#)
- [Publishing a WSDL](#)
- [Creating and Testing a Web Service \(BSE Configurations Only\)](#)
- [Configuring an Event Adapter](#)

## Starting Application Explorer

To start Application Explorer:

1. Ensure the server is started where Application Explorer is deployed.
2. On Windows, execute the `ae.bat` file, which is found under `wls_home\erp-adapters\tools\iwaeb\bin`, where `wls_home` is the directory where Oracle WebLogic Server is installed.

On UNIX, load the `iwaeb.sh` script file, which is found under `wls_home/erp-adapters/tools/iwaeb/bin`, where `wls_home` is the directory where Oracle WebLogic Server is installed.

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

## Configuring Settings for BSE or J2CA

Before a repository project can be created, you must configure BSE. You need not configure the OracleWLS Adapter J2CA because the `ra.xml` file is configured automatically during installation.

### Configuring BSE

After BSE is deployed to Oracle WebLogic Server, you can configure it through the BSE configuration page. This configuration is required only when pointing to BSE using a database repository.

---

---

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

---

---

To configure BSE:

1. Display the following page in your browser:

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

Where `host name` is the host name of Oracle WebLogic Server and `port` is the HTTP port for Oracle WebLogic Server.

For example,

`http://localhost:7777/ibse`

---

---

**Note:** If you are accessing this page for the first time, it may take longer to load.

---

---

2. When prompted, log on.

When first installed, the user ID and password are:

- User name: `weblogic`
- Password: `weblogic`

The BSE configuration page is displayed.

Property Name	Property Value
<b>System</b>	
Language	English ▼
Adapter Lib Directory	../erp-adapters/lib
Encoding	UTF-8 ▼
Debug Level	DEBUG ▼
Number of Async. Processors	0 ▼
<b>Repository</b>	
Repository Type	File System ▼
Repository Url	file://C:\wls_home\erp-adapters\libse
Repository Driver	
Repository User	
Repository Password	
Repository Pooling	<input type="checkbox"/>

3. Ensure that the Adapter Lib Directory parameter specifies the path to the lib directory, for example:

```
wls_home\erp-adapters\lib
```

After you specify the path, adapters in the lib directory are available to BSE.

4. For security purposes, enter a new password in the **Admin Password** field.

---

**Note:** The Repository URL field specifies where the file system repository is located. To use a database repository, you must enter the repository connection information. For the initial verification, use a file system repository. See "[Configuring an Oracle Repository](#)" on page 2-5 for information on switching to a database repository.

---

5. Click **Save**.

## Configuring BSE System Settings

To configure BSE system settings:

1. Display the **BSE configuration** page by entering the following URL in a browser:

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

Where `host name` is the system where BSE is installed and `port` is the port number on which BSE is listening.

---

**Note:** The server to which BSE is deployed must be running.

---

The BSE settings pane is displayed, as shown in the following figure.

Property Name	Property Value
<b>System</b>	
Language	English
Adapter Lib Directory	../erp-adapters/lib
Encoding	UTF-8
Debug Level	DEBUG
Number of Async. Processors	0

2. Configure the system settings.

The following table lists the parameters with descriptions of the information to provide.

Parameter	Description
Language	Specify the required language.
Adapter Lib Directory	Enter the full path to the directory where the adapter jar files reside.
Encoding	Only UTF-8 is supported.
Debug Level	Specify the debug level from one of the following options: <ul style="list-style-type: none"> <li>■ None</li> <li>■ Fatal</li> <li>■ Error</li> <li>■ Warning</li> <li>■ Info</li> <li>■ Debug</li> </ul>
Number of Async. Processors	Select the number of asynchronous processors.

3. Configure the repository settings.

BSE requires a repository to store transactions and metadata required for the delivery of Web services.

See ["Configuring a File System Repository"](#) on page 2-5 and ["Configuring an Oracle Repository"](#) on page 2-5 for more information.

The following image shows all fields and check boxes for the Repository pane:

<b>Repository</b>	
Repository Type	File System
Repository Url	file://C:\wls_home\erp-adapters\bse
Repository Driver	
Repository User	
Repository Password	
Repository Pooling	<input type="checkbox"/>
<b>Save</b>	

The following table lists the parameters with descriptions of the information to provide.

Parameter	Description
Repository Type	Select one of the following repositories from the list: <ul style="list-style-type: none"> <li>▪ Oracle</li> <li>▪ File (Do not use a file repository for BSE in production environments.)</li> </ul>
Repository URL	Enter the URL to use when opening a connection to the database. For example, the following repository URL format is used when connecting to Oracle: <code>jdbc:oracle:thin:@host name:port;SID</code>
Repository Driver	Provide the driver class to use when opening a connection to the database (optional). For example, the following repository driver format is used when connecting to Oracle: <code>oracle.jdbc.driver.OracleDriver</code>
Repository User	Enter a valid user ID to use when opening a connection to the database.
Repository Password	Enter a valid password that is associated with the user ID.
Repository Pooling	If selected, repository pooling will be used. This option is disabled by default.

#### 4. Click **Save**.

### Configuring a File System Repository

If you do not have access to a database for the repository, you can store repository information in an XML file on your local system. However, a file system repository is less secure and efficient than a database repository. When BSE is first installed, it is automatically configured to use a file system repository.

---

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

---

The default location for the repository on Windows is:

```
wls_home\erp-adapters\ibse.war\ibserepo.xml
```

On other platforms, use the corresponding location.

If you are using a file system repository, you are not required to configure any additional BSE components.

### Configuring an Oracle Repository

To configure an Oracle repository:

1. Contact your database administrator to obtain an Oracle user ID and password to create the BSE repository.

This user ID should have rights to create and modify tables, and the ability to create and execute stored procedures.

2. Open a command prompt and navigate to the setup directory. Its default location on Windows is:

`wls_home\erp-adapters\etc`

For other platforms, see the corresponding location.

This directory contains SQL to create the repository tables in the following file:

`iwse.ora`

---

---

**Note:** If the Oracle database is not on the same system as the Oracle WebLogic Server, copy the `iwse.ora` file to the system that has the Oracle database installed. Then, from a command prompt on the system, navigate to the directory containing the `iwse.ora` file.

---

---

3. Enter the following command:

```
sqlplus userid/password @database @ iwse.ora
```

## Configuring J2CA

During the J2CA deployment of OracleWLS Application Adapter for PeopleSoft, Oracle WebLogic Server generates a deployment descriptor called `ra.xml`. This descriptor provides Oracle WebLogic Server-specific deployment information for resource adapters. See [Chapter 3, "Oracle WebLogic Server Deployment and Integration"](#) for more information on J2CA deployment and configuration.

No configuration changes are necessary if you are using the default file based repository with J2CA deployment.

### Configuring an Oracle Repository

To configure an Oracle repository:

1. Contact your database administrator to obtain an Oracle user ID and password to create the J2CA repository.

This user ID should have rights to create and modify tables, and the ability to create and execute stored procedures.

2. Open a command prompt and navigate to the setup directory. Its default location on Windows is:

`wls_home\erp-adapters\etc`

For other platforms, see the corresponding location.

This directory contains SQL to create the repository tables in the following file:

`iwse.ora`

---

---

**Note:** If the Oracle database is not on the same system as the Oracle WebLogic Server, copy the `iwse.ora` file to the system that has the Oracle database installed. Then, from a command prompt on the system, navigate to the directory containing the `iwse.ora` file.

---

---

3. Enter the following command:

```
sqlplus userid/password @database @ iwse.ora
```

## Configuring a Database Repository for J2CA

To configure a database repository for J2CA:

1. Execute the `iwse.ora` SQL statement on the system where the database is installed.
2. Create the `jcatransport.properties` file and save it in the following directory if you created a configuration using Application Explorer called "J2CA\_SampleConfig":

```
wls_home\erp-adapters\config\J2CA_SampleConfig
```

The configuration name will vary according to the name you provided using Application Explorer.

3. Enter values for `iwafjca.repo.url`, `iwafjca.repo.user` and `iwafjca.repo.password` fields in the newly created `jcatransport.properties` file. For example:

```
iwafjca.repo.url=jdbc:oracle:thin:@90.0.0.51:1521:orcl
iwafjca.repo.user=scott
iwafjca.repo.password=scott1
```

4. Open the `ra.xml` file in a text editor.
5. Provide the JDBC connection information as a value for the `IWAYRepo_URL` property.
6. Provide a valid user name for the `IWAYRepo_User` property.
7. Provide a valid password for the `IWAYRepo_Password` property.
8. Save your changes to the `ra.xml` file.

## Password Encryption

When creating J2CA configurations, you can also encrypt a password using Application Explorer and use this value in the `jcatransport.properties` and `ra.xml` files for added security.

### Configuring Password Encryption

To encrypt a password:

1. Open Application Explorer.
2. Click **Help** and select **Encryption**.  
The Encryption dialog box is displayed.
3. Type a password in the Password field and click OK.  
An encrypted version of the password displays in the Encryption field.
4. Copy the password.
5. In the `jcatransport.properties` file, which is used during design time, replace the existing password with the encrypted value.

The following is a sample of the `jcatransport.properties` file where the password is replaced:

```
iwafjca.log.level=DEBUG
iwafjca.repo.url=jdbc:oracle:thin:@172.30.166.100:1521:orcl
iwafjca.repo.user=scott
iwafjca.repo.password=ENCR (318931973183297321831293164323332123227)
```

6. In the ra.xml file, which is used during run time, replace the existing password with the encrypted value for the IWayRepoPassword element.
7. Restart the Oracle WebLogic Server.

## Creating a Repository Configuration

Before you use Application Explorer with OracleWLS 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.

A default J2CA repository is created for the default ManagedConnectionFactory. The name of this configuration is `jca_sample`.

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

## Creating a Configuration for BSE

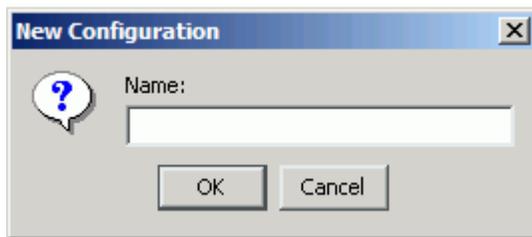
To create a repository configuration for BSE using Application Explorer, you must first define a new configuration.

### Defining a New Configuration for BSE

To define a new configuration for BSE:

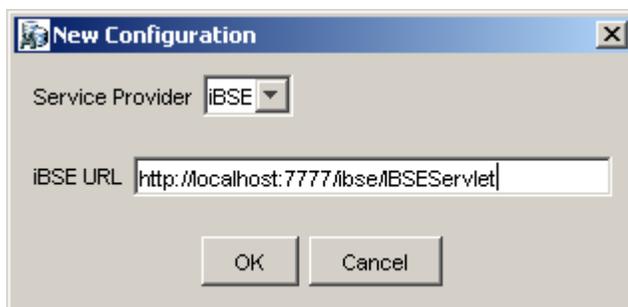
1. Right-click **Configurations** and select **New**.

The New Configuration dialog box is displayed.



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

The New Configuration dialog box is displayed.



3. From the **Service Provider** list, select **iBSE**.
4. 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 number where Oracle WebLogic Server is listening.

5. Click **OK**.

A node representing the new configuration appears beneath the root Configurations node.



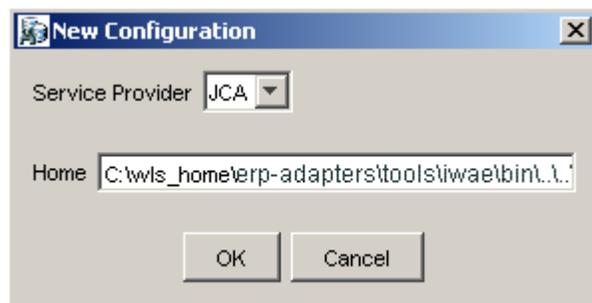
## Creating a Configuration for J2CA

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

### Defining a New Configuration for J2CA

To define a new configuration for J2CA:

1. Right-click **Configurations** and select **New**.  
The New Configuration dialog box is displayed.
2. Enter a name for the new configuration (for example, `myConfig`) and click **OK**.



3. From the **Service Provider** list, select **JCA**.
4. Click **OK**.

A node representing the new configuration appears beneath the root Configurations node.



The OracleWLS Adapter J2CA configuration file is stored in `wls_home\erp-adapters\config\configuration_name`

Where `configuration_name` is the name of the configuration you created; for example, `myConfig`.

### HTTP Repository Connection

J2CA users can create an HTTP repository connection, which enables them to generate and store WSDL documents remotely. Perform the following steps to create an HTTP repository connection in Application Explorer. To use the HTTP repository, make sure that the iwjcaivp test tool(jca-app-adapter-test) is successfully deployed and running.

1. Start the Application Explorer.
2. Right-click the **Configurations** node in the left pane and select **New**.  
The New Configuration dialog box opens.
3. Type a name for the configuration and click **OK**.
4. Select **JCA** from the Service Provider list box and enter an HTTP target value in the Home field.

Use the following format for the HTTP target value:

```
http://host name:port/iwafjca/JCAServlet
```

For example:

```
http://iwserv14:7777/iwafjca/JCAServlet
```

5. Click **OK**.

The new HTTP repository connection is added to the Configurations node.

Once you connect to the remote server, you can create new Adapter targets, generate WSDL documents, and store them in the remote server.

**Note:** When you configure an Adapter target with the J2CA HTTP repository, you are not required to restart the Oracle WebLogic Server for run time purposes.

### 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, myConfig.
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, you will not see the Business Services node. The following is an example of a BSE configuration named myConfig:



- Use the **Adapters** folder to create inbound interaction with PeopleSoft. For example, you use the PeopleSoft node in the Adapters folder to configure a service that updates PeopleSoft.
- Use the **Events** folder to configure listeners that listen for events in PeopleSoft.

- Use the **Business Services** folder (available for BSE configurations only) to test Web services created in the Adapters folder. You can also control security settings for the Web services by using the security features of the Business Services folder.

You are now ready to define new targets to PeopleSoft.

## Establishing a Connection (Target) for PeopleSoft

Part of the application definition includes adding a target for OracleWLS 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.

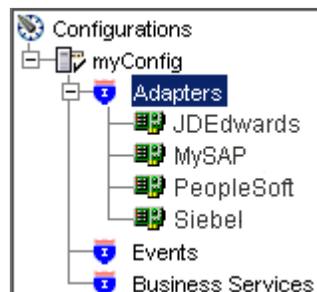
### Defining a Target to PeopleSoft

To connect to PeopleSoft for the first time, you must define a new target. OracleWLS 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.

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.



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

The Add Target dialog box 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**.

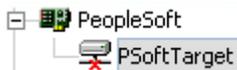
The Application Server dialog box is displayed. You must specify connection information for PeopleSoft and the application server that is hosting PeopleSoft.



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 (**PSoftTarget**) appears the PeopleSoft node.



You are ready to connect to your PeopleSoft target.

## 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, **PSoftTarget**) under the PeopleSoft node.  
The Connection dialog box displays the values you entered for connection parameters.
4. Verify your connection parameters.
5. Provide the correct password.
6. 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.



## 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.

### Disconnecting from a Connection to PeopleSoft

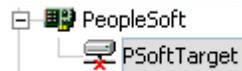
To disconnect from a connection to PeopleSoft:

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**.



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

The x icon appears, indicating that the node is disconnected.



### Modifying Connection Parameters

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

To edit a target:

1. Verify that the target you want to edit is disconnected.
2. Right-click the target and select **Edit**.



The Application Server dialog box displays the target connection information.

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

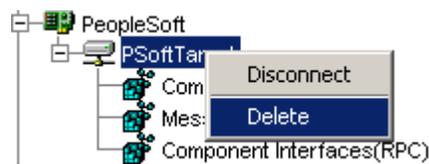
### Deleting a Connection to PeopleSoft

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, PsoffTarget), and select **Delete**.



The node disappears from the list of available connections.

## 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.

For Component Interfaces(RPC), the adapter enables Delete, Insert, Query, Update, and Find.

To view application system objects:

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

The target expands and the available system objects are displayed.



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.

---

**Note:** The Component Interfaces (RPC) node only displays the level 1 Component Interfaces.

---

## 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.

### 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. Click the **Request Schema** or **Response Schema** tab to view the request schema information.

The schema you selected appears.

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.

### Creating XML Request and Response Schemas Against the OracleWLS Application Adapter J2CA

To create XML request and response schemas for a PeopleSoft Component Interface against an OracleWLS Application 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. Click the **Request Schema** or **Response Schema** tab to view the request schema information.

The schema you selected appears.

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.

## 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.

---

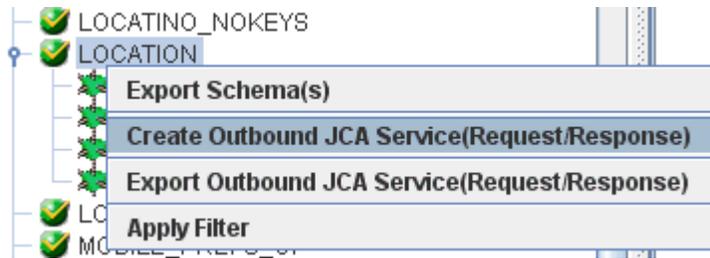


---

To generate a WSDL file for request-response service:

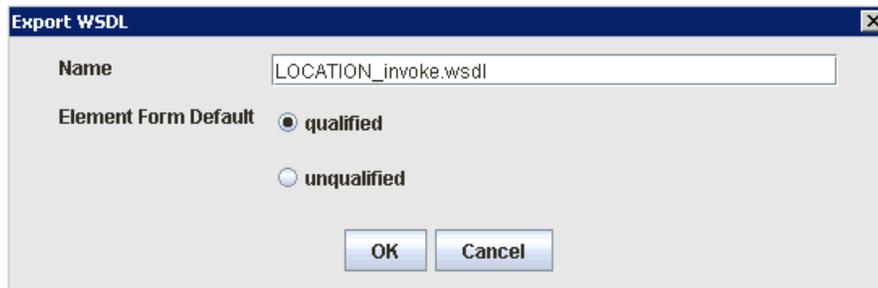
1. After you create a schema, right-click the respective object.

The following menu is displayed.



2. Select **Create Outbound JCA Service (Request/Response)**.

The Export WSDL dialog box is displayed.



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. Ensure that **qualified** is selected as the element form, which is the default.
5. 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.

## Publishing a WSDL

After you browse the PeopleSoft business object repository, you can publish the specific WSDL document for use with Oracle Service Bus. The following section describes how to publish a WSDL using Application Explorer.

---

**Note:** Only users with Group Membership types set to *Administrators* in Oracle Service Bus can publish WSDL files using Application Explorer. For example, a user with the Group Membership type set to *IntegrationDeployers* cannot publish a WSDL file.

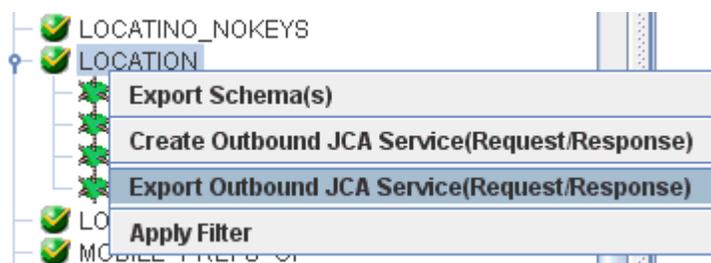
---

### Publishing a WSDL

To publish a WSDL:

1. After you create a schema, right-click the respective object.

The following menu is displayed.



2. Select **Export Outbound JCA Service (Request/Response)**.

The Export WSDL dialog box is displayed.

3. In the Name field, a default file name for the WSDL file is provided. You can accept the default or provide your own.
4. 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 “/”.
5. In the Host field, enter the name of the machine where Oracle WebLogic Server is running.
6. In the Port field, enter the port for the domain you are using. The port for the default domain is 7001.
7. In the User field, enter your username to access Oracle Service Bus.
8. In the Password field, enter your password to access Oracle Service Bus.
9. Ensure that **qualified** is selected as the element form, which is the default.
10. Click **OK**.

The WSDL is published to the location specified in the Export WSDL dialog box and is now available for use with a Business Service or Proxy Service in Oracle Service Bus.

## 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 OracleWLS Adapter J2CA, the Common Client Interface provides integration services using the adapters.

---

### Creating a Web Service

To create a Web service, perform the following steps:

1. Expand the **PeopleSoft** node and then 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**.



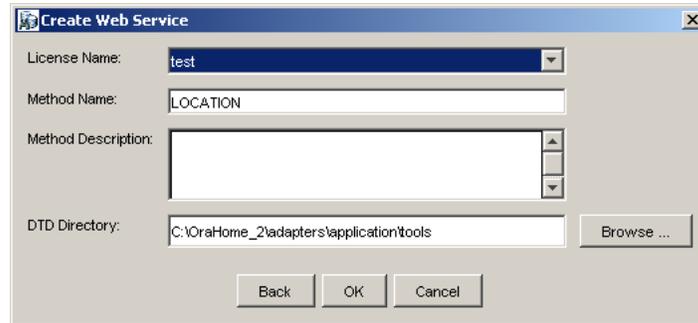
The Create Web Service dialog box is displayed.



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, 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 box 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, enter a descriptive name for the method.
  - c. In the **Method Description** field, enter a brief description of the method.
  - 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.

### 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, an input field appears.

5. Enter the appropriate input.
6. Click **Invoke**.

The BSE Servlet page displays the results.

### Identity Propagation

If you test or execute a Web service using a third party XML editor, for example XMLSPY, the Username 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.

## 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, 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, 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. See "[Creating and Editing a Channel](#)" on page 2-20 for more information.

## Creating and Editing a Channel

The following procedures describe 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, 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.

---

---

### Creating a Channel Using Specific Protocols

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 OracleWLS Application Adapter for PeopleSoft is installed. Configuring and starting a channel for a remote host is not supported.

---

The following procedures explain how to create these channels.

### Creating an HTTP Channel

To create an HTTP Channel:

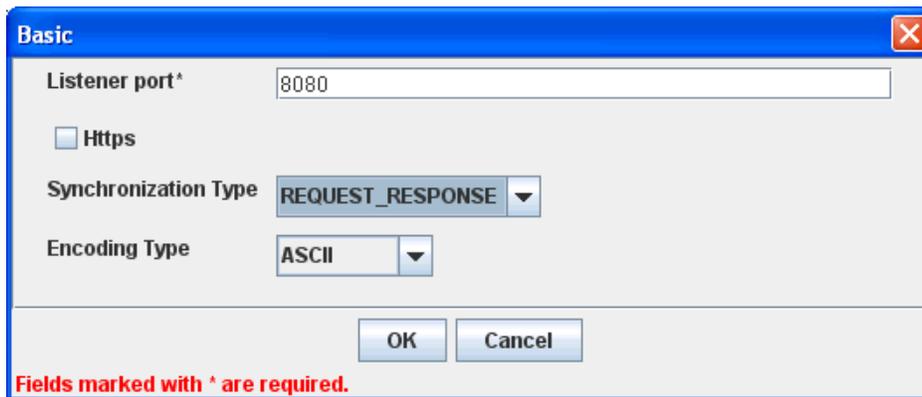
1. Click the **Events** node.
2. Expand the **PeopleSoft** node.  
The ports and channels nodes appear in the left pane.
3. Right-click **Channels** and select **Add Channel**.  
The Add Channel dialog box is displayed.

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 Basic dialog box is displayed.



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> <b>Important:</b> The PeopleSoft channel will not work if the synchronization type is set to REQUEST.
Encoding Type	Choose an encoding type to be used from the drop-down 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 using Oracle WLS Application Adapter for PeopleSoft with Oracle Service Bus (OSB) Proxy Services, do not start the channel as it is managed by OSB. If you start the channel from Application Explorer for testing and debugging purposes, stop it before run-time (when working with OSB 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.

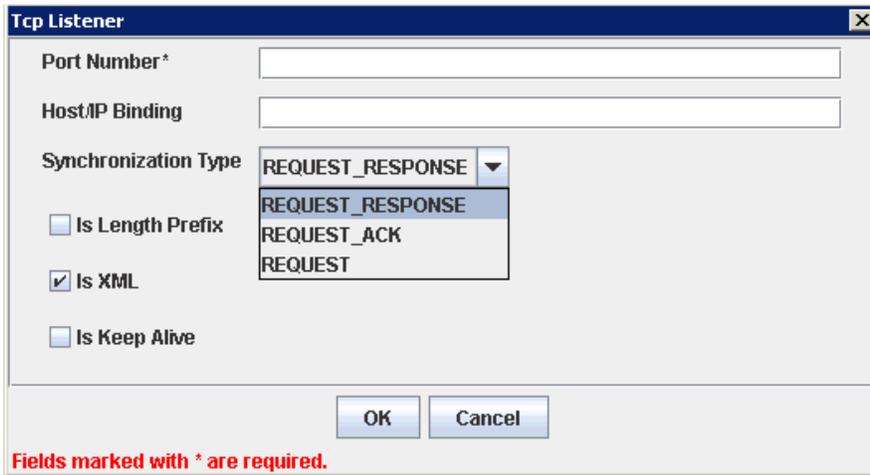
8. To stop the channel, right-click the channel and select **Stop**.

### 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 box is displayed.

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 box is displayed.



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 OracleWLS Application Adapter for PeopleSoft is installed. Configuring and starting a channel for a remote host is not supported.

---

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

**REQUEST\_RESPONSE**

**REQUEST\_ACK**

**REQUEST**

**Important:** The PeopleSoft channel will only work 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 parser 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.



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 using Oracle WLS Application Adapter for PeopleSoft with Oracle Service Bus (OSB) Proxy Services, do not start the channel as it is managed by OSB. If you start the channel from Application Explorer for testing and debugging purposes, stop it before run-time (when working with OSB components).

---

6. Right-click the channel node and select **Start**.

The channel becomes active.



The X over the icon disappears.

7. To stop the channel, right-click the connected channel node and select **Stop**.

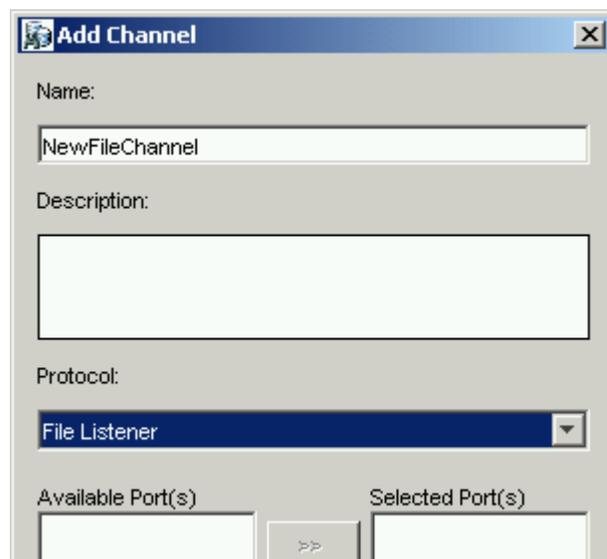
The channel becomes inactive and an X appears over the icon.

### 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**.

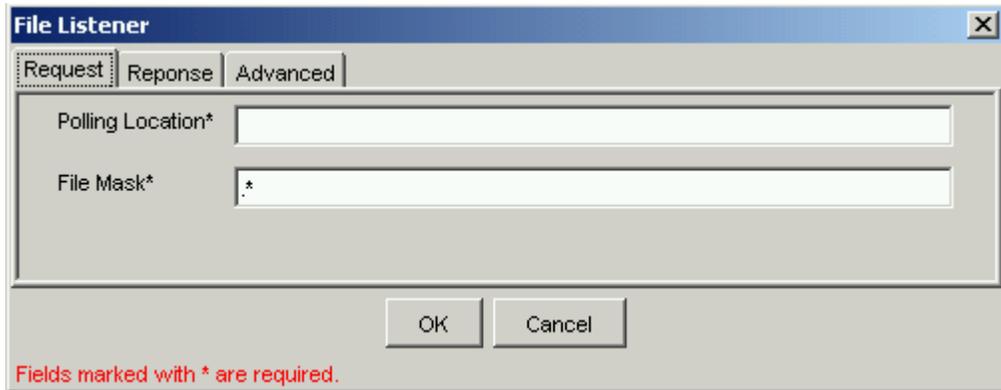
The Add Channel dialog box is displayed.



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 box is displayed.



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 will 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, 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 using OracleWLS Application Adapter for PeopleSoft with Oracle Service Bus (OSB), do not start the channel, as it is managed by OSB. If you start the channel from Application Explorer for testing and debugging purposes, stop it before run-time (when working with OSB 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**.

### Editing a Channel

You must stop the channel before you can edit it. 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**.

### Deleting a Channel

You must stop the channel before you can delete it. 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.



---

---

# Oracle WebLogic Server Deployment and Integration

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

This chapter discusses the following topics:

- [Adapter Integration with OracleWLS](#)
- [Deployment of Adapter](#)
- [Updating Adapter Configuration](#)
- [How to Write a Java Application Client Using the CCI API](#)

**See Also:**

- *Oracle WebLogic Server Adapter Concepts*

## Adapter Integration with OracleWLS

OracleWLS 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.

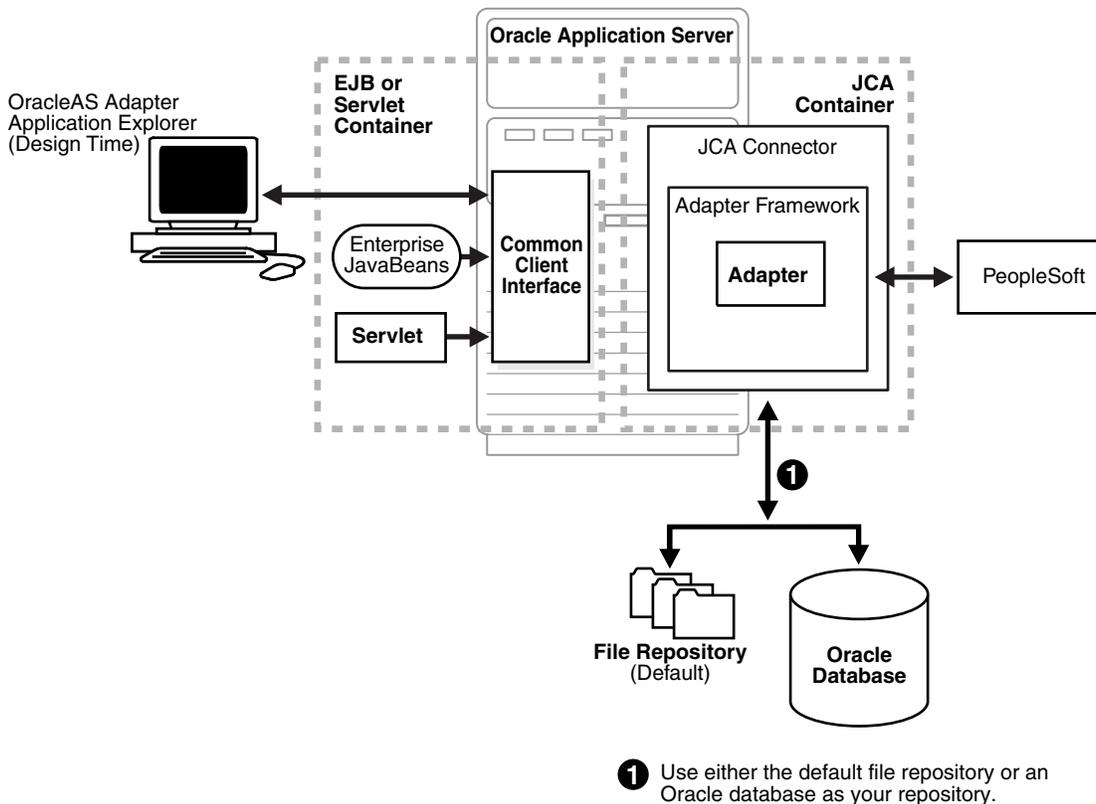
**See Also:**

- "Oracle WebLogic Server Adapters Integration with OracleWLS" in *Oracle WebLogic Server Adapter Concepts*

## Deployment of Adapter

[Figure 3-1](#) shows deployment of the Connector to the Oracle WebLogic 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 WebLogic Server Adapter J2CA Architecture



**See Also:** Oracle WebLogic Server Adapter Concepts

## Updating Adapter Configuration

During the J2CA deployment of OracleWLS Application Adapter for PeopleSoft, OracleWLS generates a deployment descriptor called `ra.xml`, located in `wls_home\erp-adapters\iwafjca.rar\META-INF`.

---

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

---

### Creating a Managed Connector Factory Object

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

```
<?xml version="1.0"?>
<!DOCTYPE oc4j-connector-factories PUBLIC "-//Oracle//DTD
Oracle Connector 9.04//EN"
"http://xmlns.oracle.com/ias/dtds/oc4j-connector-factories-9_04.dtd">
<oc4j-connector-factories>
  <connector-factory location="eis/OracleJCAAdapter/DefaultConnection"
connector-name="IWAFJCA10">
    <config-property name="IWayHome" value="../../adapters/application"/>
    <config-property name="IWayConfig" value="jca_sample"/>
  </connector-factory>
</oc4j-connector-factories>
```

```

<config-property name="IWayRepoURL" value="" />
<config-property name="IWayRepoUser" value="" />
<config-property name="IWayRepoPassword" value="" />
<config-property name="logLevel" value="debug" />
</connector-factory>
</oc4j-connector-factories>

```

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

Parameter Name	Description
IWayHome	The base installation directory for the OracleAS packaged application adapter.
IWayConfig	The adapter configuration name as defined in Application Explorer. For example, OracleWLS 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 BSE repository. See <a href="#">"Configuring BSE System Settings"</a> on page 2-3 for more information.
IWayRepoUser	User name to use when connecting to the database. This is necessary only when using an Oracle database as the BSE repository. See <a href="#">"Configuring BSE System Settings"</a> on page 2-3 for more information.
IWayRepoPassword	Password. If provided, it overwrites configuration. This is necessary only when using an Oracle database as the BSE repository. See <a href="#">"Configuring BSE System Settings"</a> on page 2-3 for more information.
loglevel	It overwrites the level set by the <code>ManagedConnectionFactory</code> property. .

### Creating Multiple Managed Connector Factory Objects

To establish multiple managed connector factory objects, you must edit the `ra.xml` file and add more `<connector-factory>` nodes. For example, the default `jca_sample` configuration in Application Explorer is represented in the `ra.xml` file as follows:

```

<?xml version="1.0"?>
<!DOCTYPE oc4j-connector-factories PUBLIC "-//Oracle//DTD
Oracle Connector 9.04//EN"
"http://xmlns.oracle.com/ias/dtds/oc4j-connector-factories-9_04.dtd">
<oc4j-connector-factories>
  <connector-factory location="eis/OracleJCAAdapter/DefaultConnection"
connector-name="IWAFJCA10">
    <config-property name="IWayHome" value="../../adapters/application"/>
    <config-property name="IWayConfig" value="jca_sample"/>
    <config-property name="IWayRepoURL" value="" />
    <config-property name="IWayRepoUser" value="" />
    <config-property name="IWayRepoPassword" value="" />
    <config-property name="logLevel" value="debug" />
  </connector-factory>
</oc4j-connector-factories>

```

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

```
<?xml version="1.0"?>
<!DOCTYPE oc4j-connector-factories PUBLIC "-//Oracle//DTD
Oracle Connector 9.04//EN"
"http://xmlns.oracle.com/ias/dtds/oc4j-connector-factories-9_04.dtd">
<oc4j-connector-factories>
  <connector-factory location="eis/OracleJCAAdapter/DefaultConnection1"
connector-name="IWAFJCA10">
    <config-property name="IWayHome" value="../../adapters/application"/>
    <config-property name="IWayConfig" value="jca_sample"/>
    <config-property name="IWayRepoURL" value="" />
    <config-property name="IWayRepoUser" value="" />
    <config-property name="IWayRepoPassword" value="" />
    <config-property name="logLevel" value="debug"/>
  </connector-factory>
<connector-factory location="eis/OracleJCAAdapter/DefaultConnection2"
connector-name="IWAFJCA10">
  <config-property name="IWayHome" value="../../adapters/application"/>
  <config-property name="IWayConfig" value="jca_sample2"/>
  <config-property name="IWayRepoURL" value="" />
  <config-property name="IWayRepoUser" value="" />
  <config-property name="IWayRepoPassword" value="" />
  <config-property name="logLevel" value="debug"/>
</connector-factory>
</oc4j-connector-factories>
```

## How to Write a Java Application Client Using the CCI API

The following example shows the code structure for using CCI with packaged application adapters. The code sample is shown in four steps.

---



---

**Note:** The OracleWLS Application Adapter for PeopleSoft does not support invocation through the CCI API. Only invocation using Oracle Service Bus (OSB) is supported. The following section is only provided for reference purposes.

---



---

### Step 1. Obtain the Connection Factory

The connection factory is obtained by JNDI lookup.

```
InitialContext context = new InitialContext();
ConnectionFactory cf = (ConnectionFactory) context.lookup(iwayJndi)
```

### Step 2. Obtaining a Connection for the Adapter

IWAFConnectionSpec is an implementation of ConnectionSpec used for creating a design time or run-time service adapter connection. The ConnectionSpec has seven parameters. Connection Pooling is fully supported and established based on these parameters, except log level.

Parameter Name	Description
adapterName	Name of the packaged application adapter.
config -	Adapter configuration name. NOT REQUIRED FOR IWAEAdapter.
language	Default is en.

Parameter Name	Description
country	Default is us.
userName	User name. If provided, it overwrites configuration.
password	Password. If provided, it overwrites configuration.
logLevel	It overwrites the level set by the ManagedConnectionFactory property.

A connection pool is a set of client connections to a specific destination. The pool may automatically create new connections to the specified remote system or return an already existing connection. It also provides methods to return a connection back to the pool when it is no longer required.

A connection pool can check which connections are no longer in use and can be closed to save system resources. The time period after which the pool checks the connections and the time after which a connection will time out can be configured by the calling application.

A pool is always bound to one user ID and password, meaning that all connections taken from this pool will also use these credentials. A PeopleSoft connection is always bound to a PeopleSoft user ID and a PeopleSoft Client number.

If you log on with a pool size that is set to one, no connection pool is created (one user ID and one process thread). If you log on with a pool size that is greater than one, a pool is created with a size of n, where n is the number you specified.

---

**Note:** Currently the OracleAS Adapter J2CA supports only basic security mapping. The DEBUG log level provides detailed information on the mapping behavior. It functions as follows:

- If the user name and password are not set and no security is provided by the application server, the OracleAS Adapter J2CA will still let it pass and rely on the adapter configuration security information.
  - If the user name and password are set, these values will overwrite the adapter configuration. The OracleAS Adapter J2CA compares this information with the security information provided by the application server and log in case the values do not match. However, it still allows the information through.
- 

The `iWAFConnectionSpec` can be set to initiate an interaction with PeopleSoft by specifying the adapter name and configuration parameters in the `ConnectionSpec`. For example,

```
iWAFConnectionSpec cs = new IWAFConnectionSpec();
cs.setAdapterName(ADAPTER);
cs.setConfig(TARGET);
cs.setLogLevel(LOG_LEVEL); // Adapter layer log level
Connection c = cf.getConnection(cs); // where cf is the connection factory
```

In this snippet, `ADAPTER` and `TARGET` refer to the adapter being deployed, in this case PeopleSoft, and the name of a target defined in Application Explorer. For more information, see ["Complete Code Sample"](#) on page 3-6.

**Step 3. Create Interaction with InteractionSpec for runtime**

```
Interaction i = c.createInteraction();
IWAFInteractionSpec is = new IWAFInteractionSpec();
is.setFunctionName(IWAFInteractionSpec.PROCESS);
```

Two functions can be set: PROCESS and IWAE. PROCESS is used at run-time. IWAE is used when you are using the IAAdapter at design time.

**Step 4. Create Input Record and Execute Interaction**

In this case, to complete the EIS invocation, a PeopleSoft message is referenced. The schema is provided by Application Explorer.

A standard J2CA indexed record is used in this example:

```
// Use JCA IndexRecord, named "input" for run-time processing.
IndexedRecord rIn = cf.getRecordFactory().createIndexedRecord("input");
rIn.add(msg_run);
IndexedRecord rOut = (IndexedRecord)i.execute(is, rIn);
System.out.println((String)rOut.get(0));
```

A special record is supported in this example:

```
//IWAFRecord rIn = new IWAFRecord("input");
//rIn.setRootXML(msg_run);
//IWAFRecord response = executeRunInteraction(c, rIn);
//IWAFRecord rOut = (IWAFRecord)i.execute(is, rIn);
//System.out.println(rOut.getRootXML());
```

Where msg\_run is an instance XML document generated from the schema created by Application Explorer. For example, the following is a sample PeopleSoft request XML document.

```
<?xml version="1.0" encoding="UTF-8" ?>
<PS8>
  <component perform="browse">LOCATION</component>
  <key name="Setid">SHARE</key>
  <key name="Location">ALBERTA</key>
</PS8>
```

**Complete Code Sample**

The following is a sample of the complete code:

```
import javax.resource.cci.*;
import com.ibi.afjca.cci.*;
import com.ibi.afjca.spi.*;

/**
 * The purpose of this sample is to illustrate how to use the IWAF Universal
 * JCA connector.
 */
public class IWAFJCA Simple {

    private static String HOME = "c:/iway/xfoc/components/iwafcont/dist";
    private static String CONFIG = "base";
    private static String LOG_LEVEL = "FATAL";

    private static String ADAPTER = "PeopleSoft";
    private static String TARGET = "PeopleSoft_connection";

    // Input Message
```

```

private static String msg_run = "<PeopleSoft/>";

    public static void main(String[] args) throws Exception {

// 1. Getting the Connection factory through JNDI lookup
// -----
    InitialContext context = new InitialContext();
    ConnectionFactory cf = (ConnectionFactory)context.lookup(iwayJndi)
// 2. Getting a connection for a particular adapter target, in this case
PeopleSoft
// -----
    IWAFFConnectionSpec cs = new IWAFFConnectionSpec();
    cs.setAdapterName(ADAPTER);
    cs.setConfig(TARGET);
    cs.setLogLevel(LOG_LEVEL); // Adapter layer log level
    Connection c = cf.getConnection(cs); // where cf is the connection factory

// 3. Create interaction with interactionSpec for RUNTIME
// -----
    Interaction i = c.createInteraction();
    IWAFFInteractionSpec is = new IWAFFInteractionSpec();
    is.setFunctionName("PROCESS");

// 4. Create input Record and execute interaction
// -----

// 4.1 Using JCA standard Indexed Record
// Use JCA IndexRecord, named "input" for run-time processing.
IndexedRecord rIn = cf.getRecordFactory().createIndexedRecord("input");
rIn.add(msg_run);
    IndexedRecord rOut = (IndexedRecord)i.execute(is, rIn);
    System.out.println((String)rOut.get(0));

// 4.2 Our own Record is supported here
//IWAFFRecord rIn = new IWAFFRecord("input");
//rIn.setRootXML(msg_run);
//IWAFFRecord response = executeRunInteraction(c, rIn);
//IWAFFRecord rOut = (IWAFFRecord)i.execute(is, rIn);
//System.out.println(rOut.getRootXML());

    } // main()
}

```



---

---

## Configuring Outbound Processing Using Oracle Service Bus (BSE Configuration)

OracleWLS 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 includes the following topics:

- [Overview of Application Adapter Integration with Oracle Service Bus](#)
- [Publishing a WSDL From Application Explorer to Oracle Service Bus](#)
- [Starting Oracle Service Bus and Creating Project Folders](#)
- [Configuring a File Type Business Service](#)
- [Configuring a WSDL Type Business Service](#)
- [Configuring a Proxy Service](#)
- [Configuring a Pipeline](#)

### Overview of Application Adapter Integration with Oracle Service Bus

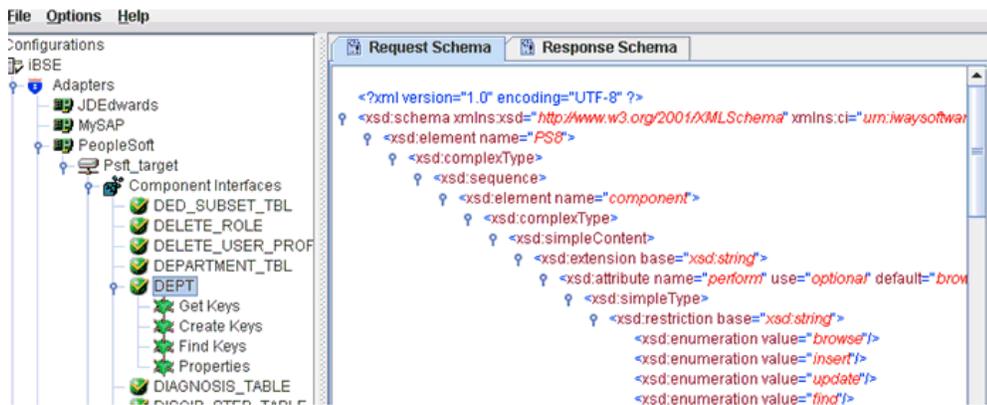
To integrate with Oracle Service Bus (OSB), OracleWLS Application Adapter for PeopleSoft must be deployed on the same Oracle WebLogic Server as OSB. The underlying adapter services must be exposed as WSDL files, which are generated during design time in Oracle WebLogic Server Adapter Application Explorer (Application Explorer) for both request-response (outbound) and event notification (inbound) services of the adapter.

### 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.

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



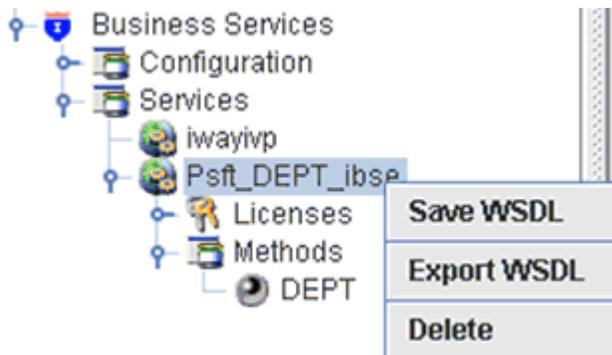
2. Expand the **Component Interfaces** node and select the **DEPT** component interface.
3. Right-click the **DEPT** component interface and select **Create Web Service** from the context menu.

The Create Web Service dialog box is displayed.



4. Enter a service name and click **Next**.
5. Click **OK** on the next dialog box 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 context menu.
- The Export WSDL dialog box is displayed.

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. Ensure that **qualified** is selected as the element form, which is the default.
14. Click **OK**.

The WSDL is published to the location specified in the Export WSDL dialog box and is now available for use with a Business Service or Proxy Service in Oracle Service Bus.

## Starting Oracle Service Bus and Creating Project Folders

This section describes how to access the Oracle Service Bus Console 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 port for the default domain is 7001.

The Oracle Service Bus Console logon page is displayed.

## ORACLE Service Bus 10gR3

The screenshot shows the login interface for Oracle Service Bus 10gR3. At the top, it says 'Welcome'. Below that is a form with two input fields: 'Username:' with the text 'weblogic' and 'Password:' with masked characters. A 'Login' button is located at the bottom right of the form.

Oracle Service Bus 10gR3  
 Copyright © 2004,2008, Oracle and/or its affiliates. All rights reserved.  
 Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

- Log on to the Oracle Service Bus Administrative Console using a valid user name and password.

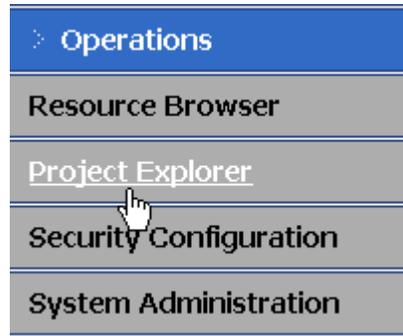
The Oracle Service Bus Console home page is displayed.

The screenshot shows the Oracle Service Bus 10gR3 Administrative Console home page. The page has a navigation menu on the left and a main content area. The navigation menu includes sections like 'Change Center', 'Operations', 'Monitoring', 'Configuration', and 'Reporting'. The main content area displays 'SLA Alerts (30 mins)', 'Services With Most Alerts', and 'Alert History (30 mins)'. The 'Alert History' section shows a table with columns for 'Timestamp', 'Alert Name', 'Alert Severity', 'Service', 'Service Type', and 'Action'. The table is currently empty, displaying 'No Alerts to display.'

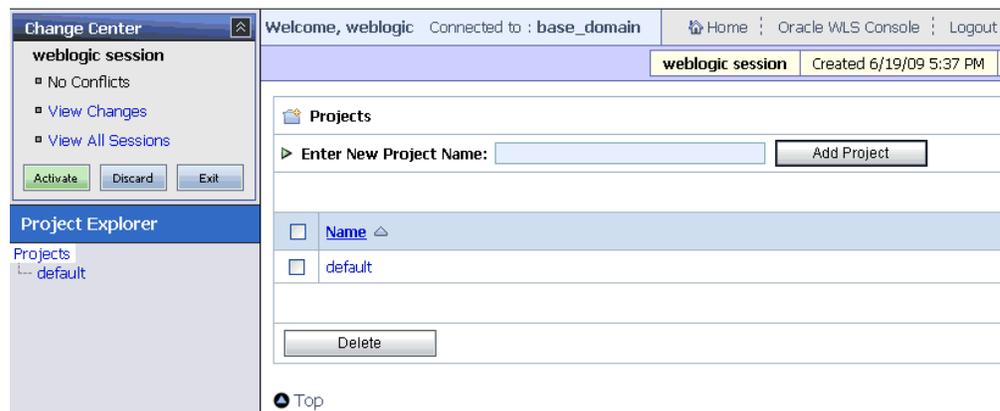
- Click **Create** in the Change Center area to start a new Oracle Service Bus session.



5. Click **Project Explorer** in the left pane.

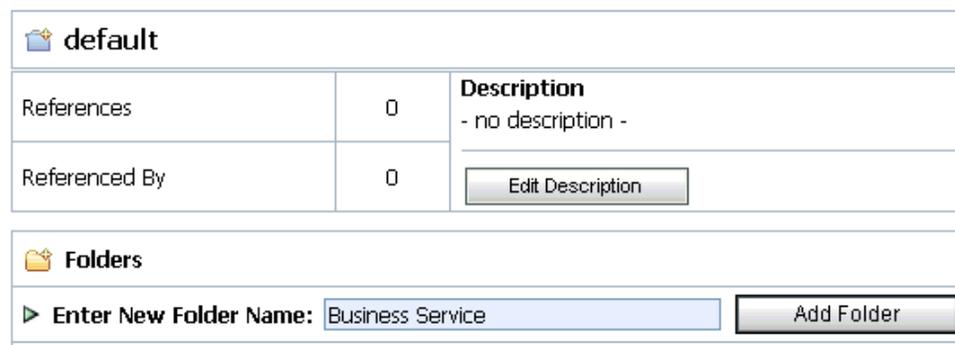


The Project Explorer page is displayed.



6. Click the **default** project node in the left pane.

The default project page is displayed.



7. In the Enter New Folder Name field, type **Business Service** and click **Add Folder**.

The Business Service folder is listed in the left pane below the default project node.



8. In the Enter New Folder Name field, type **Proxy Service** and click **Add Folder**.

The Business Service and Proxy Service folders are listed in the left pane below the default project node.



## Configuring a File Type Business Service

This section describes how to configure a File type Business Service using the Oracle Service Bus Console.

1. Select the **Business Service** folder you created in the left pane.



2. In the right pane, select **Business Service** from the Create Resource menu.  
The Create a Business Service - General Configuration page is displayed.

The screenshot shows the 'Create a Business Service (default/BusinessService/)' wizard in the 'General Configuration' step. The 'Service Name' field contains 'Psft\_DEPT\_IBSE\_out'. The 'Description' field is empty. Under 'Service Type\*', the 'Create a New Service' section has 'Messaging Service' selected. The 'Any SOAP Service' option has a dropdown menu set to 'SOAP 1.1'. The 'Create From Existing Service' section has 'Business Service' and 'Proxy Service' options, each with an empty text field and a 'Browse...' button. At the bottom, there are 'Next >>', 'Last >>', and 'Cancel' buttons.

3. Provide a name for the Business Service and from the Service Type area select **Messaging Service**.
4. Click **Next**.

The screenshot shows the 'Create a Business Service (default/BusinessService/Psft\_DEPT\_IBSE\_out)' wizard in the 'Message Type Configuration' step. Under 'Request Message Type', 'XML' is selected. Under 'Response Message Type', 'None' is selected. Both sections have 'MFL' and 'XML' options with empty text fields and 'Browse...' buttons. The 'XML' option in both sections has a sub-field for '(element or type)'. At the bottom, there are '<< Prev.', 'Next >>', 'Last >>', and 'Cancel' buttons.

5. Select **XML** as the Request Message Type and **None** as the Response Message Type.
6. Click **Next**.

**Create a Business Service (default/BusinessService/Psft\_DEPT\_IBSE\_out)**

**Transport Configuration**

Protocol*	file
Load Balancing Algorithm	round-robin
Endpoint URI*	<p>Format: file:///root-dir/dir1</p> <p>file:/// <input type="text"/> <input type="button" value="Add"/></p> <p><b>EXISTING URIS</b> 0</p> <p>file:///C:/output</p>
Retry Count	0
Retry Iteration Interval	30

<< Prev. | Next >> | Last >> | Cancel

7. Select **file** from the Protocol drop-down list.
8. Enter the path to a destination folder on your file system in the Endpoint URI field and click **Add**.
9. Click **Next**.

**Create a Business Service (default/BusinessService/Psft\_DEPT\_IBSE\_out)**

**FILE Transport Configuration**

Prefix	Psft_Dept_out
Suffix	.xml
Request encoding	utf-8

<< Prev. | Next >> | Last >> | Cancel

10. Enter the prefix and suffix for the output file to be received and click **Next**.

Description	
Service Type	Messaging Service
<b>Message Type Configuration</b>	
Request Message Type	XML
Response Message Type	None
<b>Transport Configuration</b>	
Protocol	file
Load Balancing Algorithm	round-robin
Endpoint URI	file:///C:/output
Retry Count	0
Retry Iteration Interval	30
<b>FILE Transport Configuration</b>	
Prefix	Psft_Dept_out
Suffix	.xml
Request encoding	utf-8
<input style="margin-right: 20px;" type="button" value=" &lt;&lt; Prev. "/> <input style="margin-right: 20px;" type="button" value=" Save "/> <input style="margin-right: 20px;" type="button" value=" Cancel "/>	

11. Review all the information for your Business Service and click **Save**.

## Configuring a WSDL Type Business Service

This section describes how to configure a WSDL type Business Service using the Oracle Service Bus Console.

1. Select the **Business Service** folder you created in the left pane.



2. In the right pane, select **Business Service** from the Create Resource menu. The Create a Business Service - General Configuration page is displayed.

**Create a Business Service (default/BusinessService/)**

**General Configuration**

Service Name\*

Description

Service Type\*

**Create a New Service**

WSDL Web Service     
 (port or binding)

Transport Typed Service

Messaging Service

Any SOAP Service

Any XML Service

**Create From Existing Service**

Business Service

Proxy Service

3. Provide a name for the Business Service and from the Service Type area select **WSDL Web Service**.

4. Click **Browse**.

The Select a WSDL Definition dialog box is displayed.

**Oracle Service Bus : Select a WSDL definition - Microsoft Internet Explorer**

**Select a WSDL definition**

Search: Name:  Path:    [Adv. Search](#)

Name	Path	WSDL Namespace
Psft_dept_ibse_wsdl	default/wsdl	urn:schemas-iwaysoftware-com:iwse

Description:

> Select WSDL definitions

**Bindings**  
 PS\_dept\_ibseSoap

**Ports**  
 PS\_dept\_ibseSoap1

5. Select an available WSDL definition and click **Submit**.

You are returned to the General Configuration page where the WSDL you selected is now available.

Create a Business Service (default/BusinessService/)	
<b>General Configuration</b>	
Service Name*	Psft_DEPT_IBSE_BS
Description	
Service Type*	<p><b>Create a New Service</b></p> <p><input checked="" type="radio"/> WSDL Web Service <input type="text" value="default/wsdl/Psft_dept_ibse_wsdl"/> <input type="button" value="Browse..."/>  <input type="text" value="PS_dept_ibseSoap1"/> (port)</p> <p><input type="radio"/> Transport Typed Service</p> <p><input type="radio"/> Messaging Service</p> <p><input type="radio"/> Any SOAP Service <input type="text" value="SOAP 1.1"/></p> <p><input type="radio"/> Any XML Service</p> <p><b>Create From Existing Service</b></p> <p><input type="radio"/> Business Service <input type="text"/> <input type="button" value="Browse..."/></p> <p><input type="radio"/> Proxy Service <input type="text"/> <input type="button" value="Browse..."/></p>
<input type="button" value="Next &gt;&gt;"/>   <input type="button" value="Last &gt;&gt;"/>   <input type="button" value="Cancel"/>	

6. Click Next.

Create a Business Service (default/BusinessService/Psft_DEPT_IBSE_BS)					
<b>Transport Configuration</b>					
Protocol*	<input type="text" value="http"/>				
Load Balancing Algorithm	<input type="text" value="round-robin"/>				
Endpoint URI*	<p>Format: http://host:port/someService</p> <p><input type="text" value="http://"/> <input type="button" value="Add"/></p> <table border="1"> <thead> <tr> <th>EXISTING URIS</th> <th>OP</th> </tr> </thead> <tbody> <tr> <td>http://AMTEX-CH-QA162.AMTEXPDC:7001/ibse/IBSEServlet/XDSOAPRouter</td> <td></td> </tr> </tbody> </table>	EXISTING URIS	OP	http://AMTEX-CH-QA162.AMTEXPDC:7001/ibse/IBSEServlet/XDSOAPRouter	
EXISTING URIS	OP				
http://AMTEX-CH-QA162.AMTEXPDC:7001/ibse/IBSEServlet/XDSOAPRouter					
Retry Count	<input type="text" value="0"/>				
Retry Iteration Interval	<input type="text" value="30"/>				
Retry Application Errors	<input checked="" type="radio"/> Yes <input type="radio"/> No				
<input type="button" value=" &lt;&lt; Prev."/>   <input type="button" value="Next &gt;&gt;"/>   <input type="button" value="Last &gt;&gt;"/>   <input type="button" value="Cancel"/>					

7. Select **http** from the Protocol drop-down list.
8. Enter the Endpoint URI in HTTP format and click **Add**.
9. Click **Next**.

 Create a Business Service (default/BusinessService/Psft\_DEPT\_iBSE\_BS)

**HTTP Transport Configuration**

Timeout	<input type="text" value="0"/>
HTTP Request Method	<input type="text" value="POST"/>
Authentication	<input checked="" type="radio"/> None <input type="radio"/> Basic <input type="radio"/> Client Certificate
Service Account	<input type="text"/> <input type="button" value="Browse..."/>
Dispatch Policy	<input type="text" value="default"/>
Request Encoding	<input type="text"/>
Response Encoding	<input type="text"/>

**Advanced Settings**

10. Click Next.

 Create a Business Service (default/BusinessService/Psft\_DEPT\_iBSE\_BS)

**SOAP Binding Configuration**

Enforce WS-I Compliance	<input type="checkbox"/>
-------------------------	--------------------------

11. Click Next.

 Create a Business Service (default/BusinessService/Psft\_DEPT\_iBSE\_BS)

**Message Content Handling**

XOP/MTOM Support	<input type="checkbox"/> Enabled <input checked="" type="radio"/> Include Binary Data by Reference <input checked="" type="radio"/> Include Binary Data by Value
Attachments	<input type="checkbox"/> Page Attachments to Disk

12. Click Next.

Endpoint URI	http://AMTEX-CH-QA162.AMTEXPDC:7001/lbse/IBSEServlet/XDSOAPRouter
Retry Count	0
Retry Iteration Interval	30
Retry Application Errors	Yes
<b>HTTP Transport Configuration</b>	
Timeout	0
HTTP Request Method	POST
Authentication	None
Proxy Server	
Follow HTTP redirects	DISABLED
Use Chunked Streaming Mode	ENABLED
<b>SOAP Binding Configuration</b>	
Enforce WS-I Compliance	No
<b>Message Content Handling Configuration</b>	
XOP/MTOM Support	Disabled
Page Attachments to Disk	No
<input data-bbox="456 863 615 886" type="button" value=" &lt;&lt; Prev. "/> <input data-bbox="673 863 833 886" type="button" value=" Save "/> <input data-bbox="891 863 1050 886" type="button" value=" Cancel "/>	

- Review all the information for your Business Service and click **Save**.

## Configuring a Proxy Service

This section describes how to configure a Proxy Service using the Oracle Service Bus Console.

- Select the **Proxy Service** folder you created in the left pane.



- In the right pane, select **Proxy Service** from the Create Resource menu. The Create a Proxy Service - General Configuration page is displayed.

**Create a Proxy Service (default/ProxyService/)**

**General Configuration**

Service Name\*

Description

Service Type\*

**Create a New Service**

WSDL Web Service    
 (port or binding)

Messaging Service

Any SOAP Service

Any XML Service

**Create From Existing Service**

Business Service

Proxy Service

|  |

3. Provide a name for the Proxy Service and from the Service Type area select **Any XML Service**.
4. Click **Next**.

**Create a Proxy Service (default/ProxyService/Psft\_DEPT\_IBSE\_PS)**

**Transport Configuration**

Protocol\*

Endpoint URI\* Format: file://root-dir/dir1

Get All Headers

Yes

No

Header

HEADER	ACTION
There are no headers configured.	

|  |  |

5. Select **file** from the Protocol drop-down list.
6. Enter the path to an input folder on your file system in the Endpoint URI field.
7. Click **Next**.

Create a Proxy Service (default/ProxyService/Psft_DEPT_IBSE_PS)	
FILE Transport Configuration	
File Mask*	*.*
Polling Interval*	60
Read Limit*	10
Sort By Arrival	<input type="checkbox"/>
Scan SubDirectories	<input type="checkbox"/>
Pass By Reference	<input type="checkbox"/>
Post Read Action*	delete
Stage Directory*	C:\input\stage
Archive Directory	
Error Directory*	C:\input\error
Request encoding	utf-8
<input data-bbox="457 873 657 909" type="button" value=" &lt;&lt; Prev. "/> <input data-bbox="685 873 885 909" type="button" value=" Next &gt;&gt; "/> <input data-bbox="954 873 1154 909" type="button" value=" Last &gt;&gt; "/> <input data-bbox="1224 873 1424 909" type="button" value=" Cancel "/>	

8. Provide any folder locations on your file system for the Stage Directory and Error Directory fields.
9. Click Next.

Create a Proxy Service (default/ProxyService/Psft_DEPT_IBSE_PS)	
Message Content Handling	
Content Streaming	<input type="checkbox"/> Enabled <b>Buffer Type</b> <input checked="" type="radio"/> Memory Buffer <input checked="" type="radio"/> Disk Buffer <b>Compression</b> <input type="checkbox"/> Enabled
<input data-bbox="457 1451 657 1486" type="button" value=" &lt;&lt; Prev. "/> <input data-bbox="685 1451 885 1486" type="button" value=" Next &gt;&gt; "/> <input data-bbox="954 1451 1154 1486" type="button" value=" Last &gt;&gt; "/> <input data-bbox="1224 1451 1424 1486" type="button" value=" Cancel "/>	

10. Click Next.

Protocol	file
Endpoint URI	file:///C:/input
Get All Headers	No
Headers	
<b>FILE Transport Configuration</b>	
File Mask	*.*
Polling Interval	60
Read Limit	10
Sort By Arrival	false
Scan SubDirectories	false
Pass By Reference	false
Post Read Action	delete
Stage Directory	C:\input\stage
Error Directory	C:\input\error
Request encoding	utf-8
<b>Message Content Handling Configuration</b>	
Content Streaming	Disabled

11. Review all the information for your Proxy Service and click **Save**.

## Configuring a Pipeline

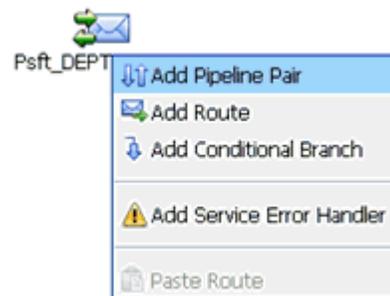
This section describes how to configure a Pipeline using the Oracle Service Bus Console.

1. Click the **Edit Message Flow** icon in the row of the Proxy Service you created.

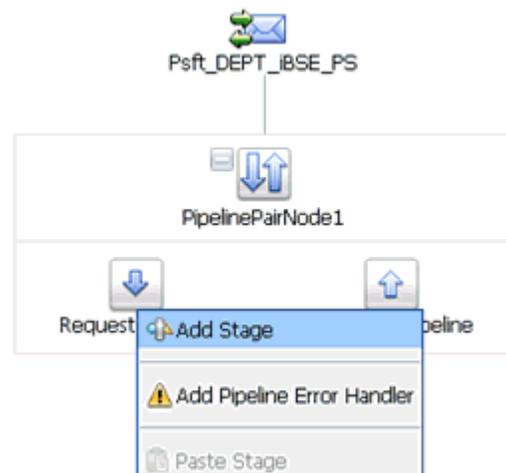


The Edit Message Flow workspace area is displayed.

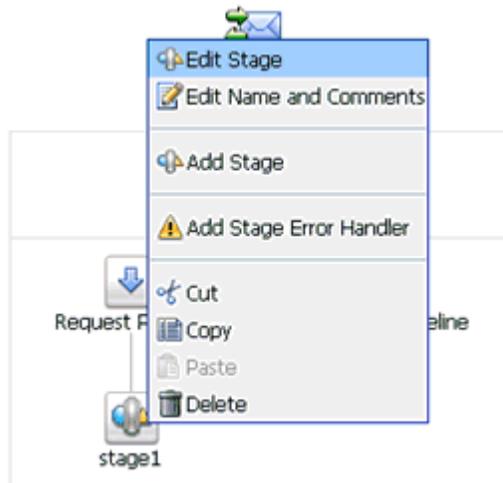
2. Click the **Proxy Service** icon and select **Add Pipeline Pair** from the context menu, as shown in the following image.



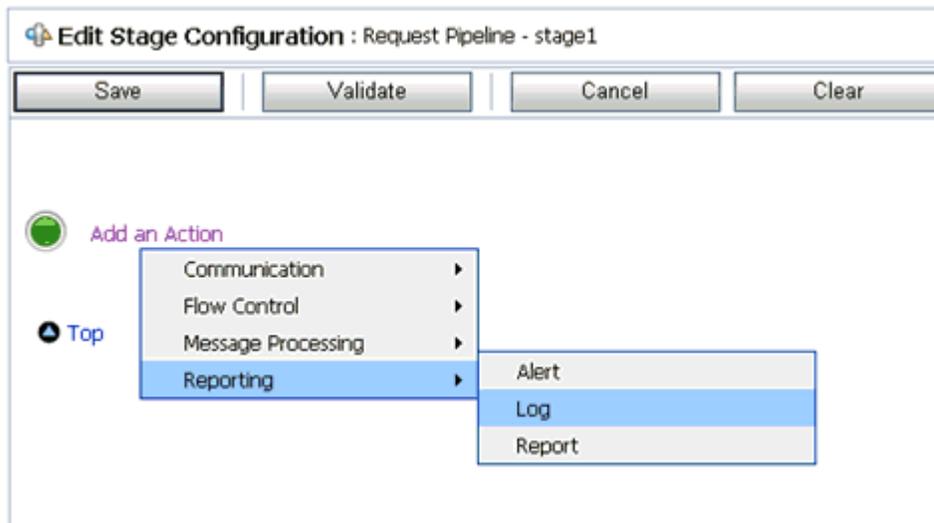
3. Click the **Request Pipeline** icon and select **Add Stage** from the context menu.



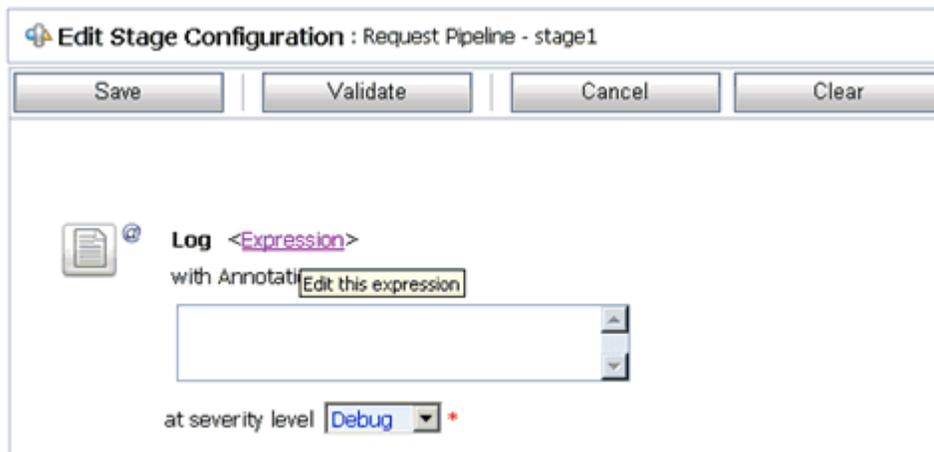
The Stage1 icon is added below the Request Pipeline icon.



- Click the **Stage1** icon and select **Edit Stage** from the context menu. The Edit Stage Configuration workspace area is displayed.

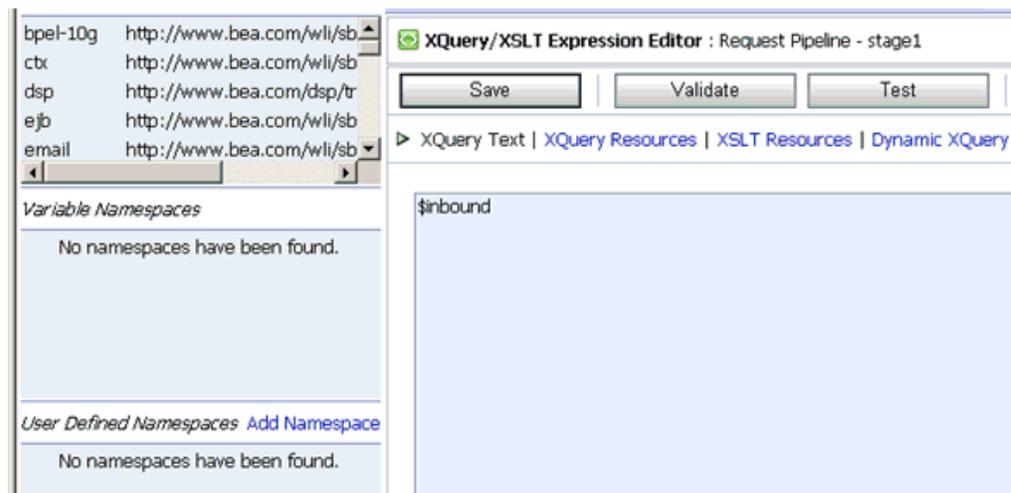


- Click **Add an Action**, select **Reporting** from the context menu, and click **Log**.



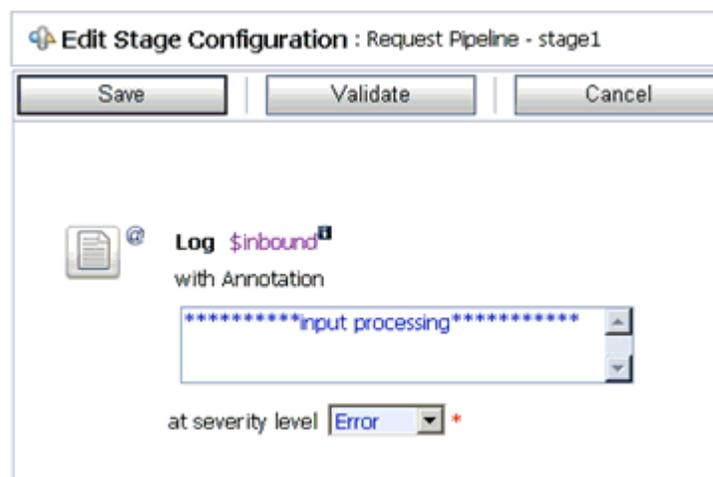
- Click **<Expression>** to edit the expression.

The XQuery/XSLT Expression Editor is displayed.

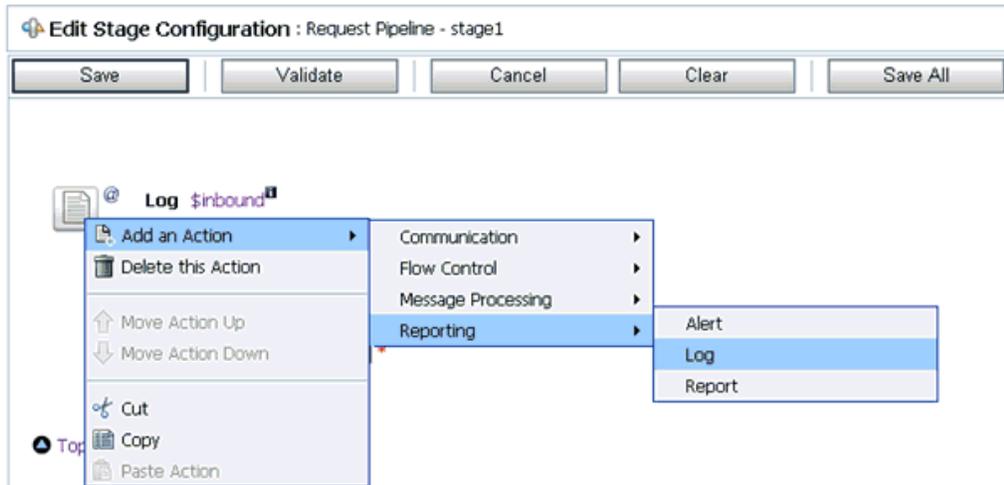


7. In the XQuery Text area, type **\$inbound**.
8. Click **Validate** and then **Save**.

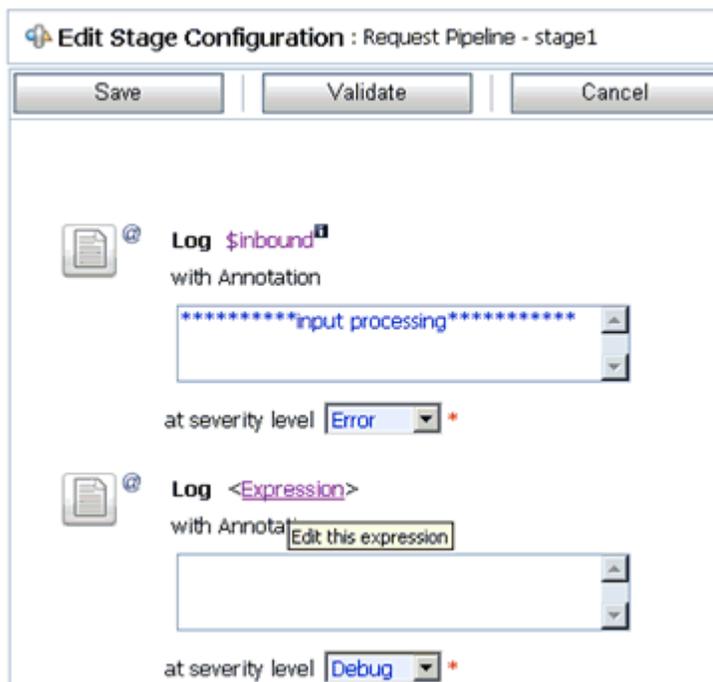
You are returned to the Edit Stage Configuration workspace area.



9. Type any annotation/comments in the text box (for example, \*\*\*\*\*input processing\*\*\*\*\*).
10. Select **Error** from the severity level drop-down list.
11. Add one more Log action as shown in the following image.



A new Log configuration is added, as shown in the following image.



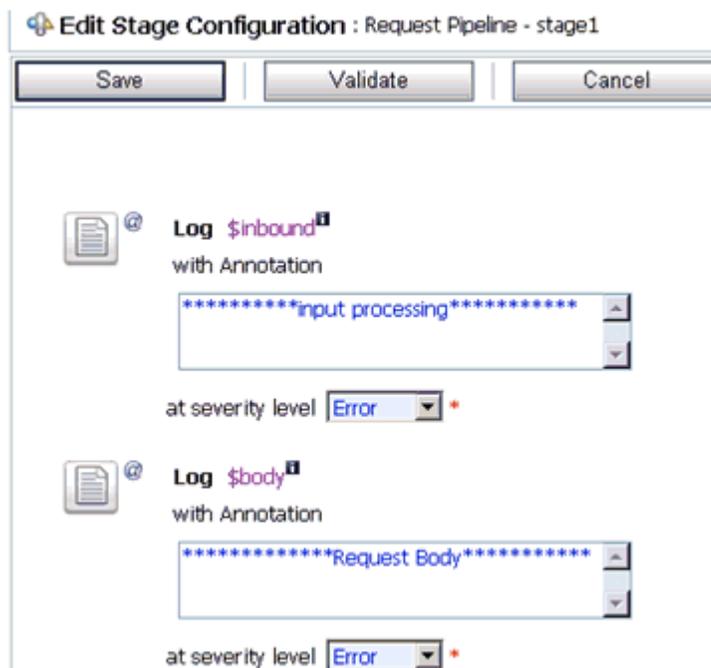
12. Click **<Expression>** to edit the expression.  
The XQuery/XSLT Expression Editor is displayed.



13. In the XQuery Text area, type `$body`.

14. Click **Validate** and then **Save**.

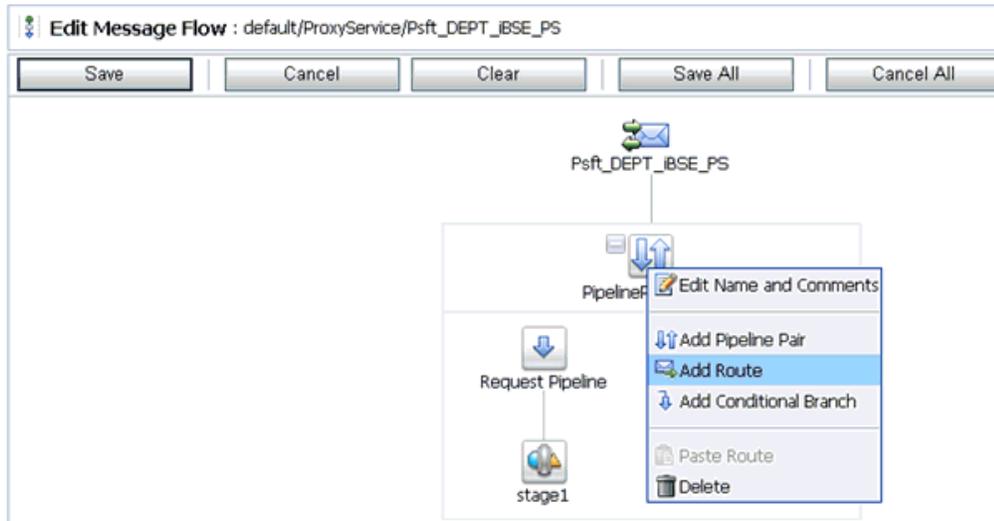
You are returned to the Edit Stage Configuration workspace area.



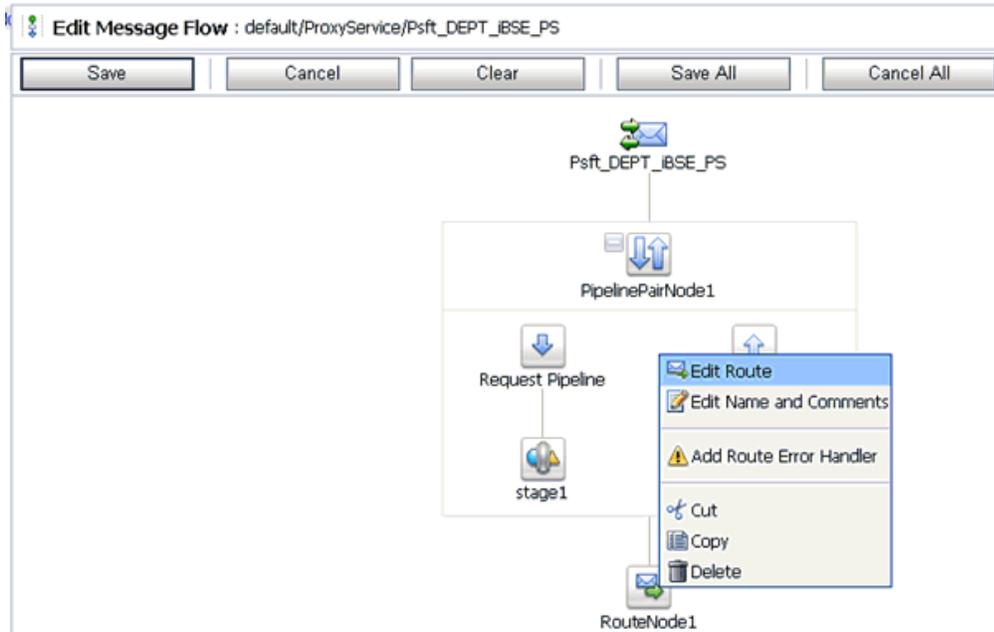
15. Type any annotation/comments in the text box (for example, `*****Request Body*****`).

16. Select **Error** from the severity level drop-down list.

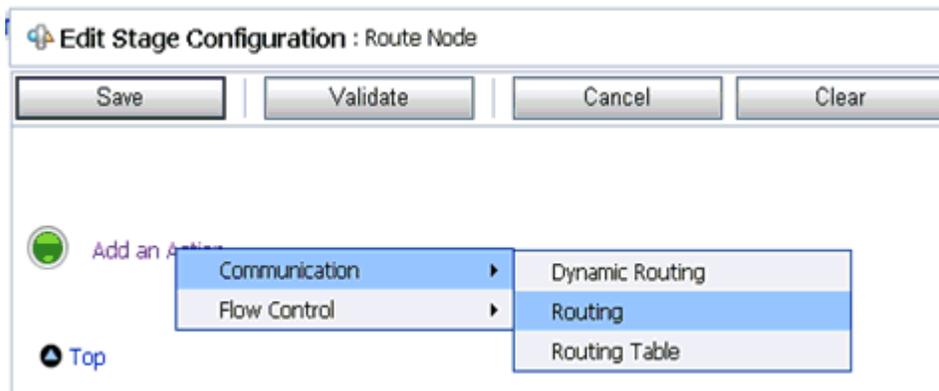
17. Click **Validate** and then **Save**.



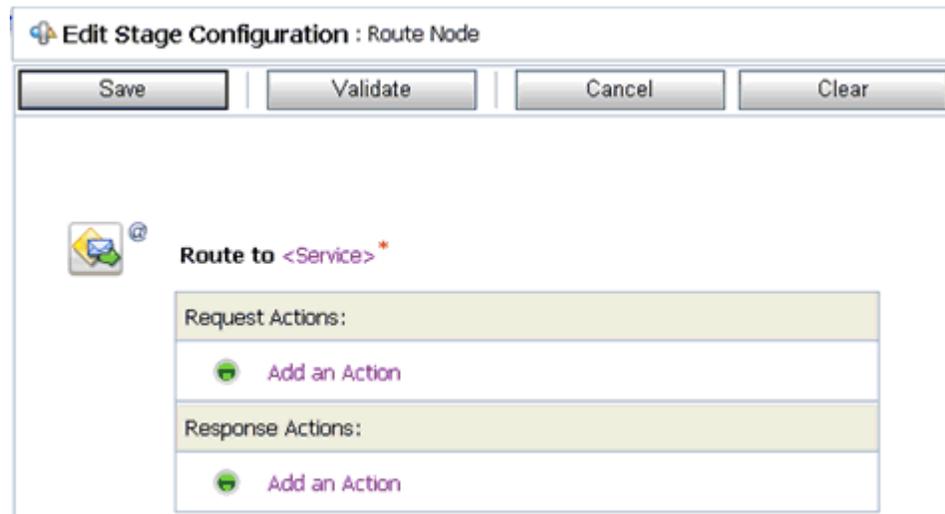
18. Click the **PipelinePairNode1** icon and select **Add Route** from the context menu.



19. Click the **RouteNode1** icon and select **Edit Route** from the context menu.

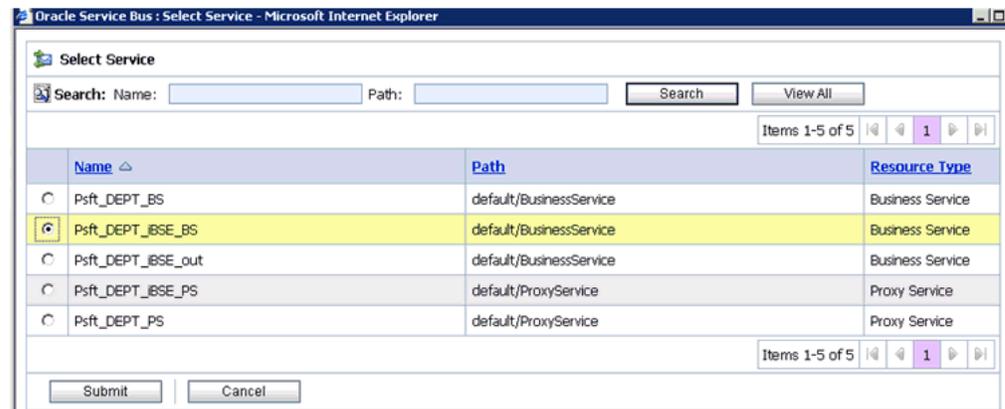


20. Click **Add an Action**, select **Communication** from the context menu, and click **Routing**.



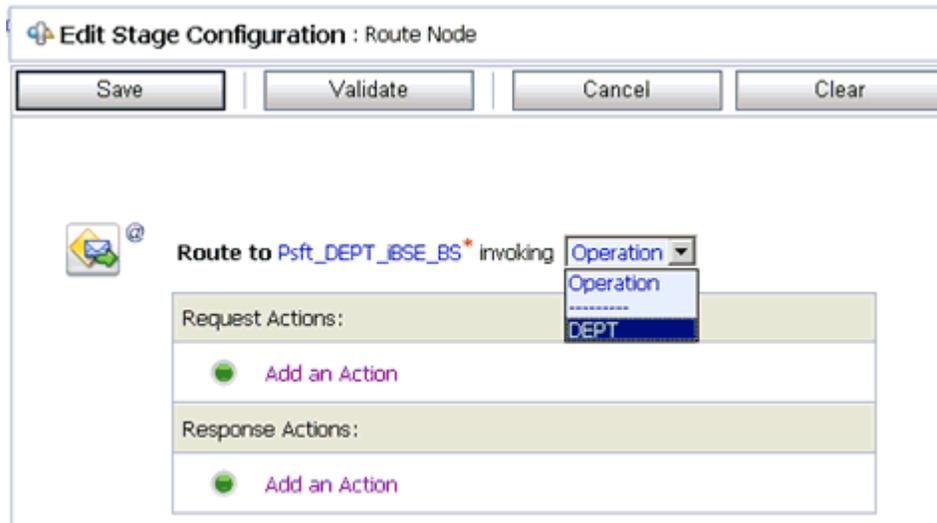
21. Click **<Service>**.

The Select Service dialog box is displayed.



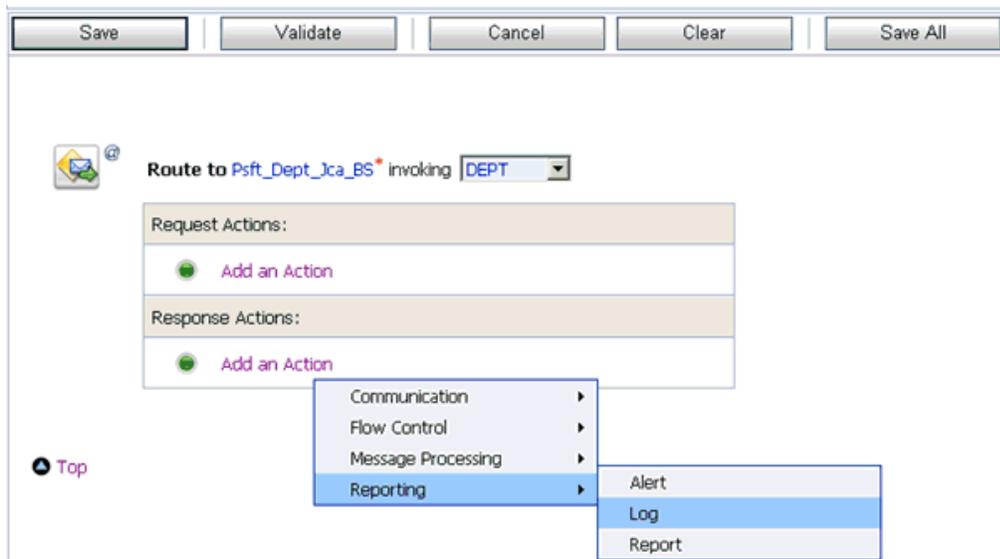
22. Select a WSDL type Business Service and click **Submit**.

You are returned to the Edit Stage Configuration workspace area.

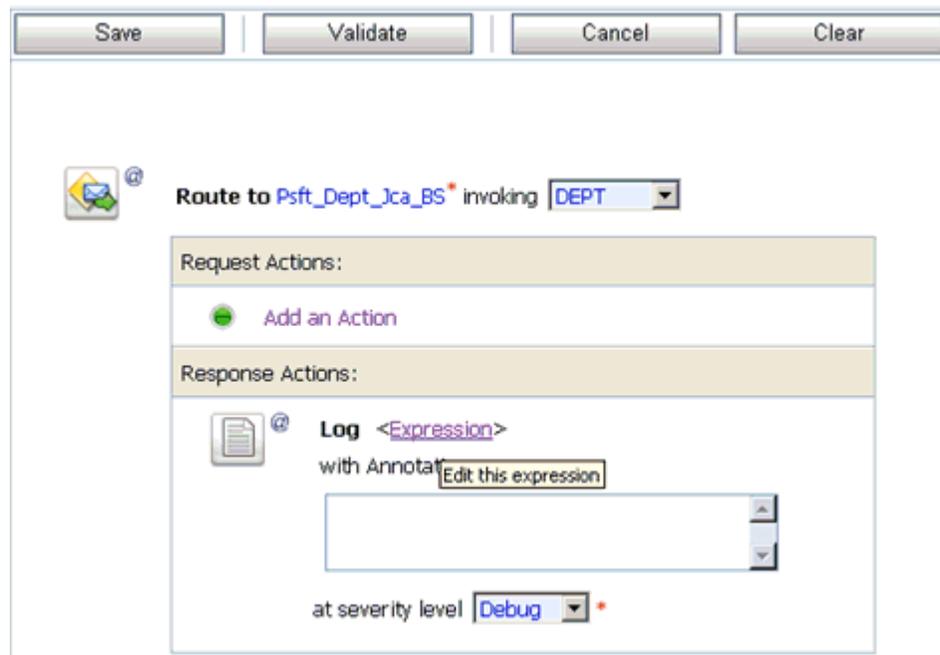


23. Select **DEPT** as the operational attribute from the drop-down list.

24. Click **Validate** and then **Save**.

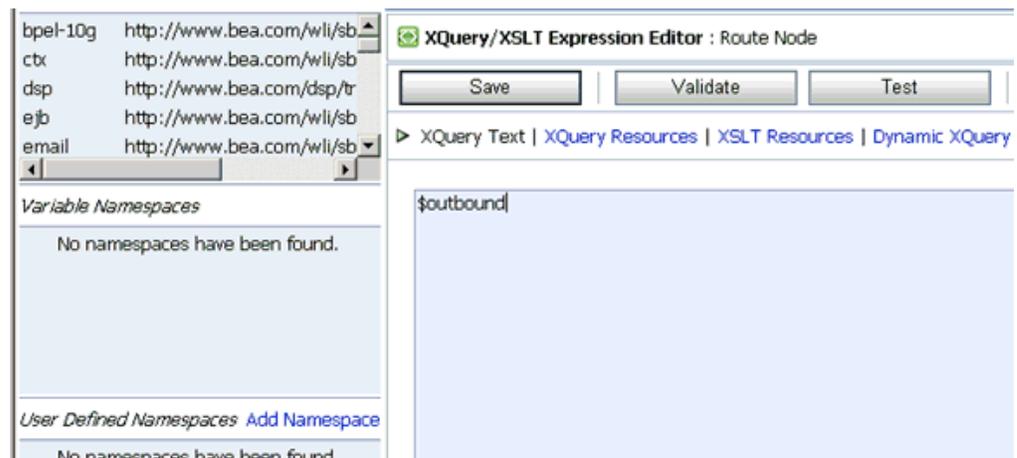


25. In the Response Actions area, click **Add an Action**, select **Reporting** from the context menu, and click **Log**.



26. Click **<Expression>** to edit the expression.

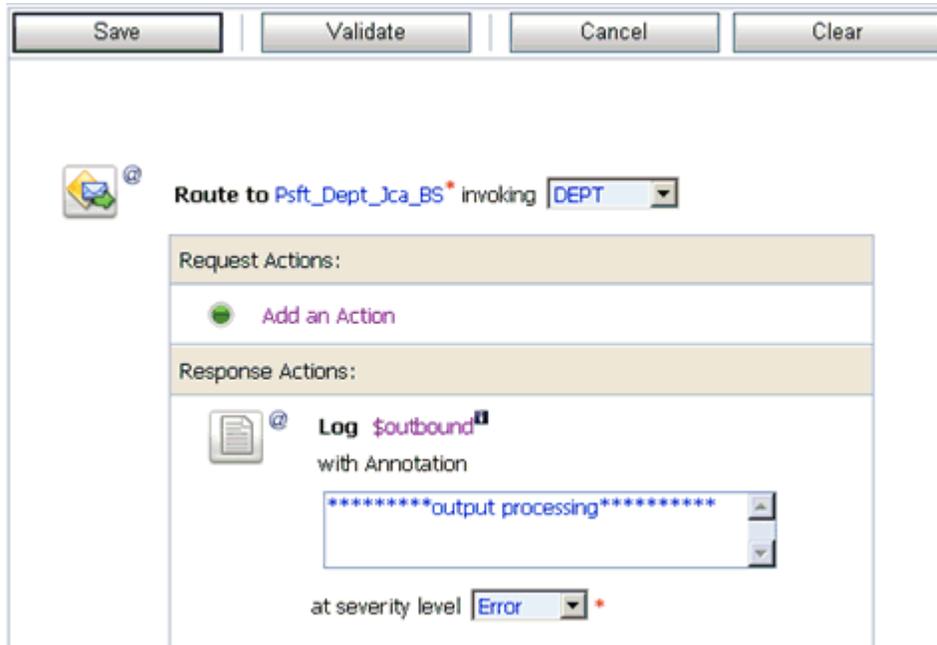
The XQuery/XSLT Expression Editor is displayed.



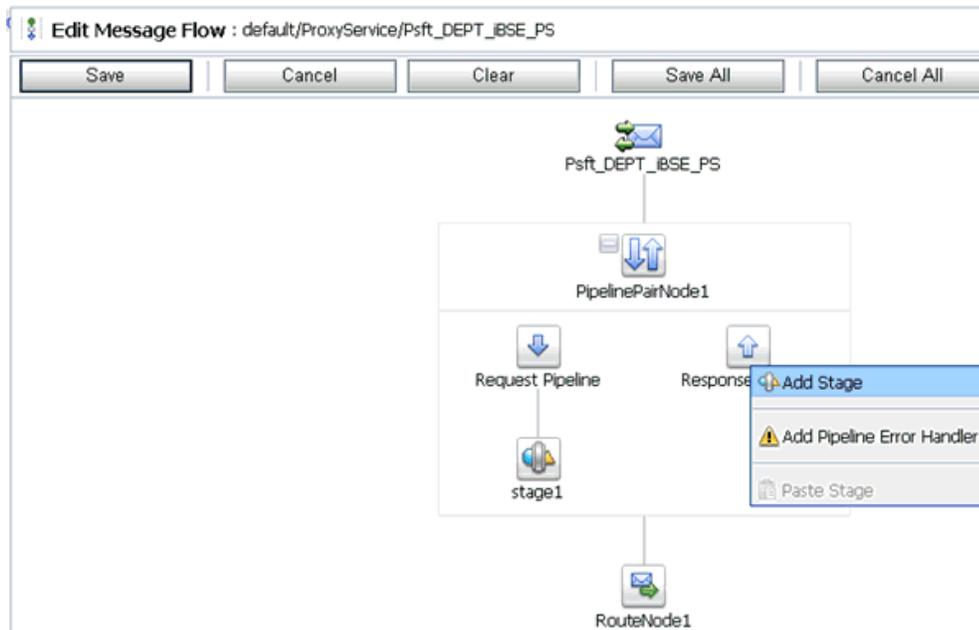
27. In the XQuery Text area, type **\$outbound**.

28. Click **Validate** and then **Save**.

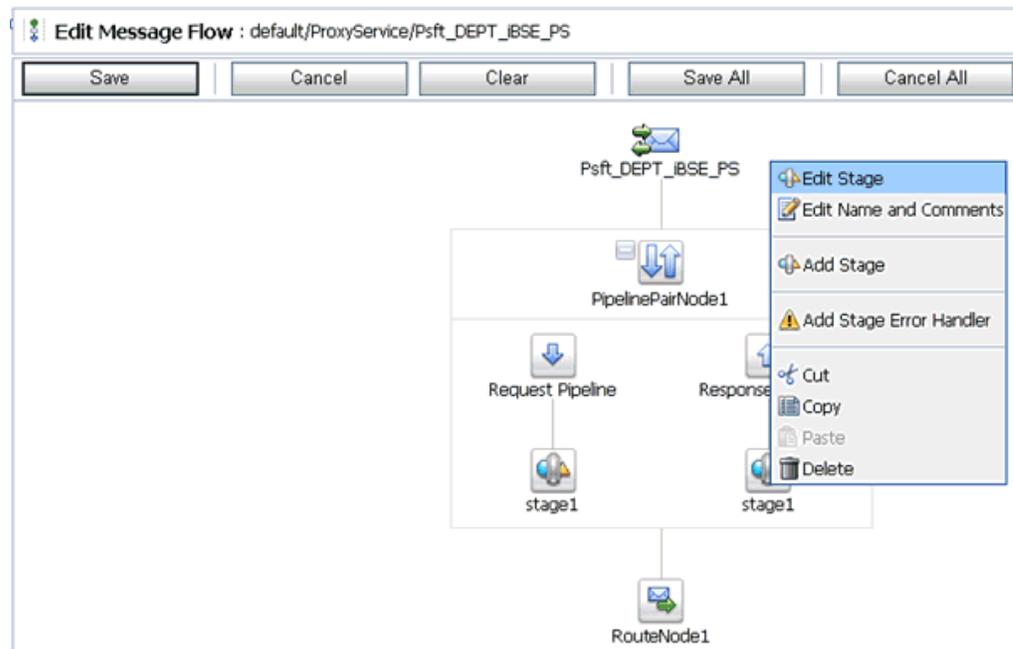
You are returned to the Edit Stage Configuration workspace area.



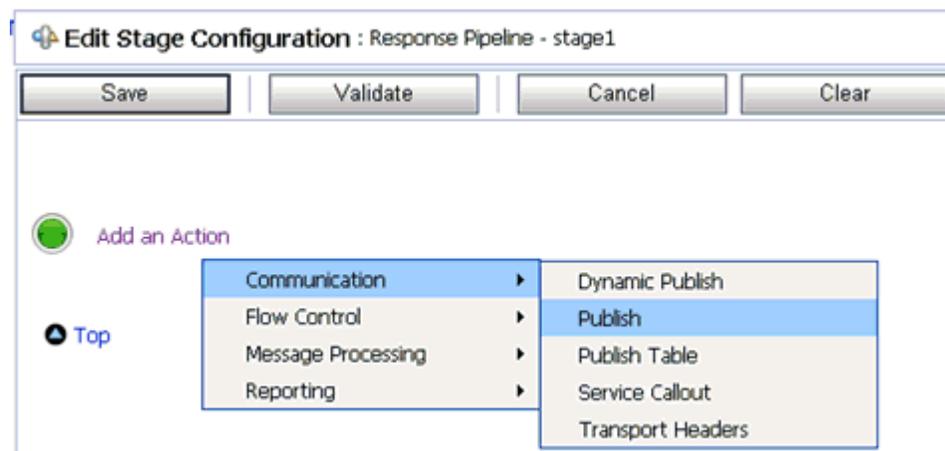
29. Type any annotation/comments in the text box (for example, \*\*\*\*\*output processing\*\*\*\*\*).
30. Select **Error** from the severity level drop-down list.
31. Click **Validate** and then **Save**.



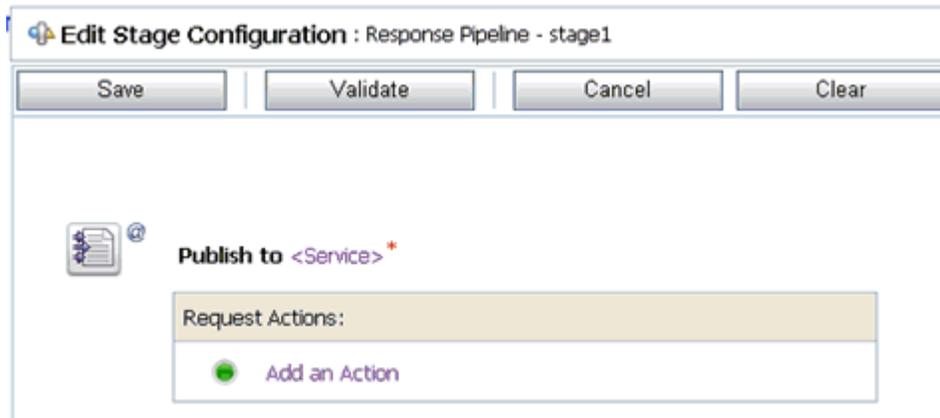
32. Click the **Response Pipeline** icon and select **Add Stage** from the context menu. The Stage1 icon is added below the Response Pipeline icon.



33. Click the **Stage1** icon and select **Edit Stage** from the context menu.  
The Edit Stage Configuration workspace area is displayed.

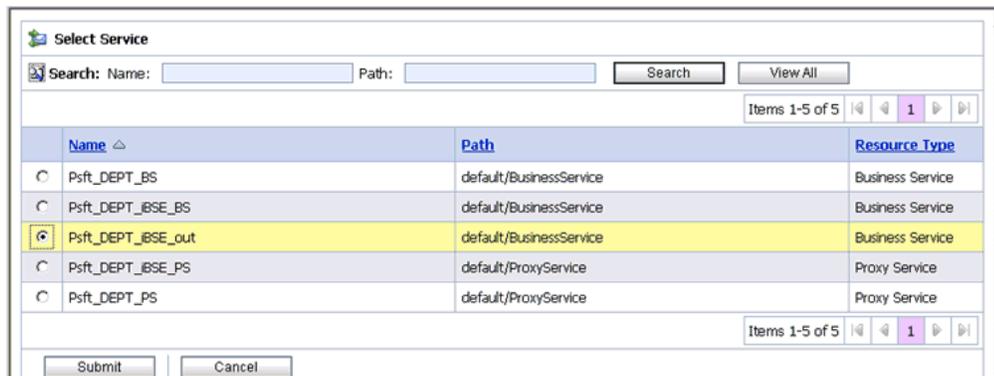


34. Click **Add an Action**, select **Communication** from the context menu, and click **Publish**.



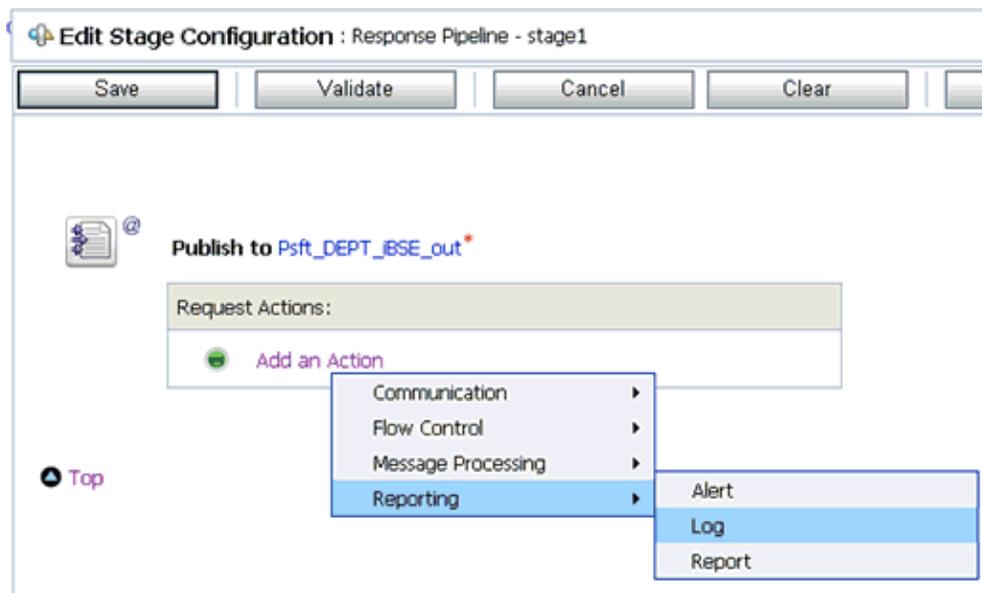
35. Click <Service>.

The Select Service dialog box is displayed.

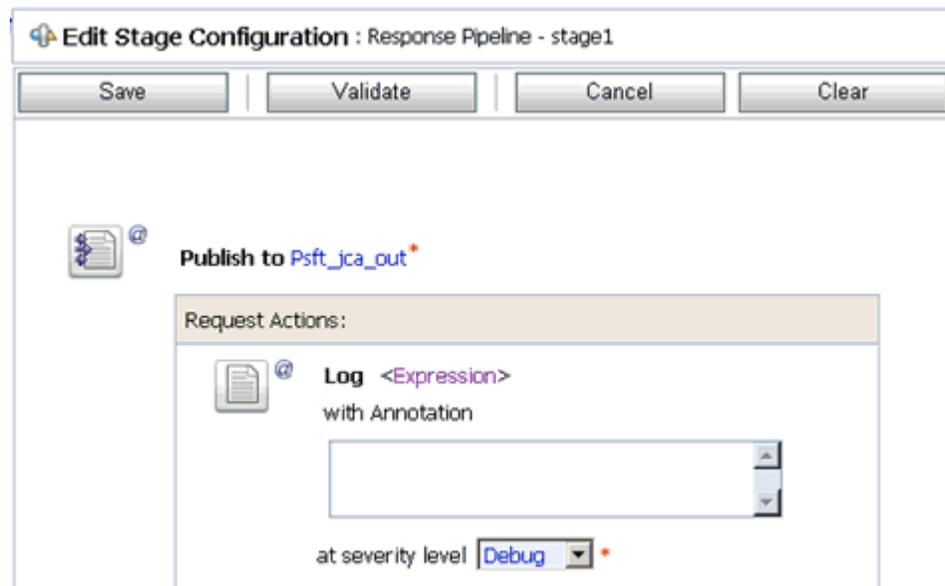


36. Select a File type Business Service and click **Submit**.

You are returned to the Edit Stage Configuration workspace area.

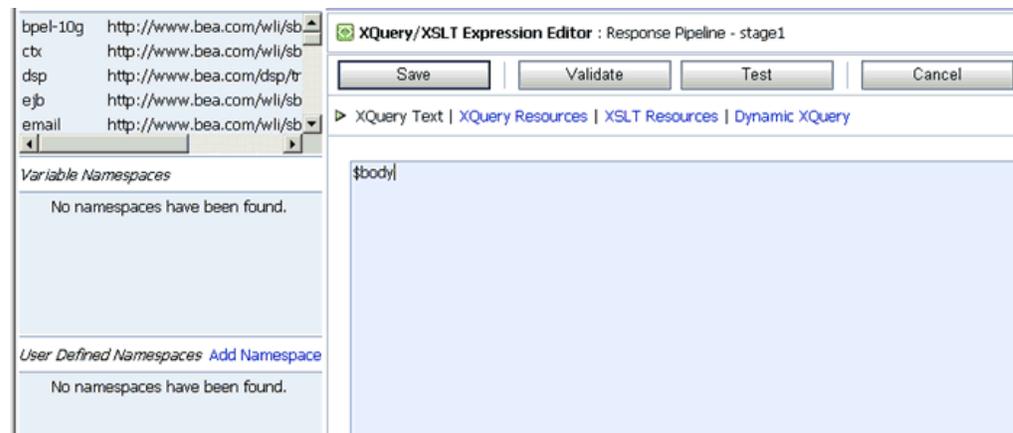


37. In the Request Actions area, click **Add an Action**, select **Reporting** from the context menu, and click **Log**.



38. Click **<Expression>** to edit the expression.

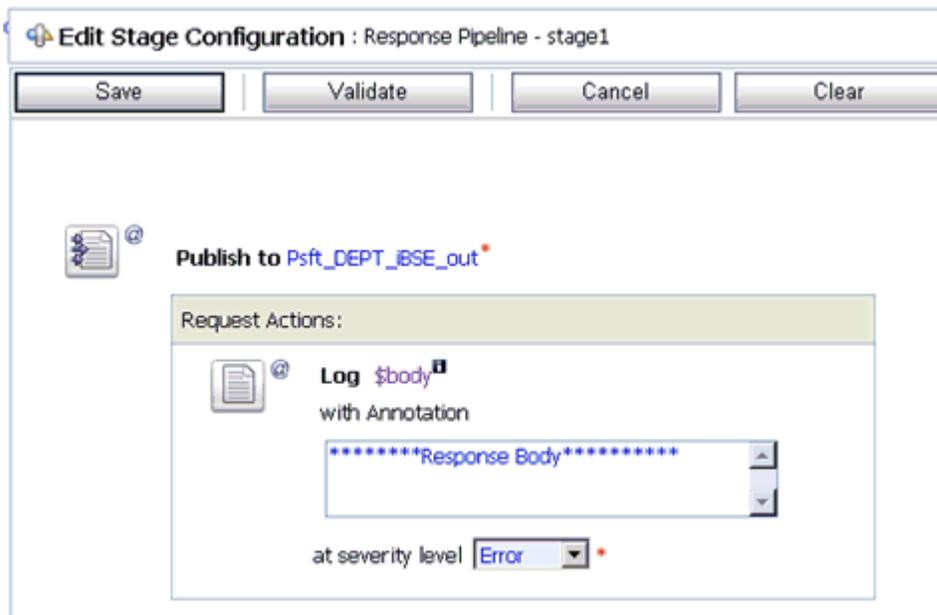
The XQuery/XSLT Expression Editor is displayed.



39. In the XQuery Text area, type **\$body**.

40. Click **Validate** and then **Save**.

You are returned to the Edit Stage Configuration workspace area.



41. Type any annotation/comments in the text box (for example, \*\*\*\*\*Response Body\*\*\*\*\*).
42. Select **Error** from the severity level drop-down list.
43. Click **Validate** and then **Save**.



44. Click **Save**.
45. Click **Activate** in the Change Center area to activate your changes in the Oracle Service Bus session.



46. Copy and paste an input XML file in the input folder you have configured to receive an output XML file in the destination folder.



---

---

# Configuring Inbound and Outbound Processing Using Oracle Service Bus (J2CA Configuration)

OracleWLS 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 includes the following topics:

- [Overview of Application Adapter Integration with Oracle Service Bus](#)
- [Configuring Inbound Processing Using Oracle Service Bus \(J2CA Configuration\)](#)
- [Configuring Outbound Processing Using Oracle Service Bus \(J2CA Configuration\)](#)

## Overview of Application Adapter Integration with Oracle Service Bus

To integrate with Oracle Service Bus (OSB), OracleWLS Application Adapter for PeopleSoft must be deployed on 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 WebLogic Server Adapter Application Explorer (Application Explorer) for both request-response (outbound) and event notification (inbound) services of the adapter.

## Configuring Inbound Processing Using Oracle Service Bus (J2CA Configuration)

This section describes how to configure inbound processing using Oracle Service Bus (J2CA Configuration).

This section includes the following topics:

- [Creating a Channel and Publishing a WSDL From Application Explorer to Oracle Service Bus](#)
- [Starting Oracle Service Bus and Creating Project Folders](#)
- [Configuring a File Type Business Service](#)
- [Configuring a Proxy Service](#)
- [Configuring a Pipeline](#)

## Creating a Channel and Publishing a WSDL From Application Explorer to Oracle Service Bus

This section describes how to create a channel and publish a WSDL from Application Explorer (J2CA configuration) 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 OracleWLS Application Adapter for PeopleSoft"](#).



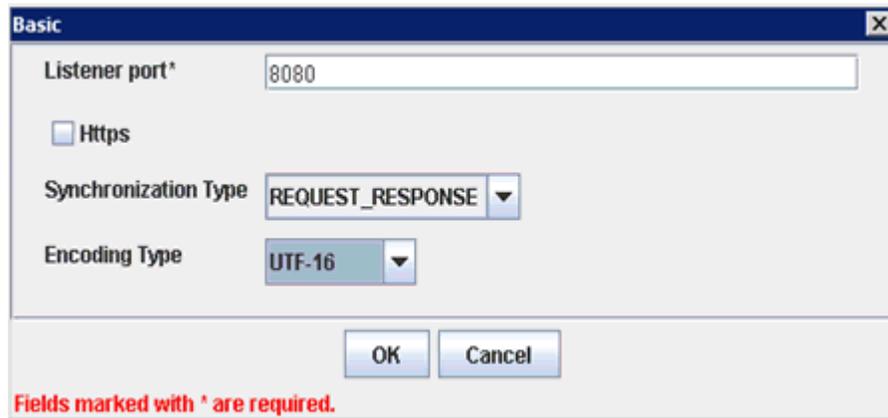
2. Right-click the **Channels** node for PeopleSoft and select **Add Channel** from the context menu.

The Add Channel dialog box is displayed.

The 'Add Channel' dialog box is shown. It has a title bar with 'Add Channel' and a close button. The 'Name' field contains 'psft\_dept\_ch'. The 'Description' field is empty. The 'Protocol' dropdown menu is set to 'HTTP Listener'. Below this are two columns: 'Available Port(s)' and 'Selected Port(s)', both empty. Between these columns are four arrow buttons: '>>', '>', '<', and '<<'. At the bottom are 'Next' and 'Cancel' buttons. The 'Next' button is highlighted.

3. Enter a name for the new PeopleSoft channel, select **HTTP** from the Protocol drop-down list, and click **Next**.

The Basic dialog box is displayed.



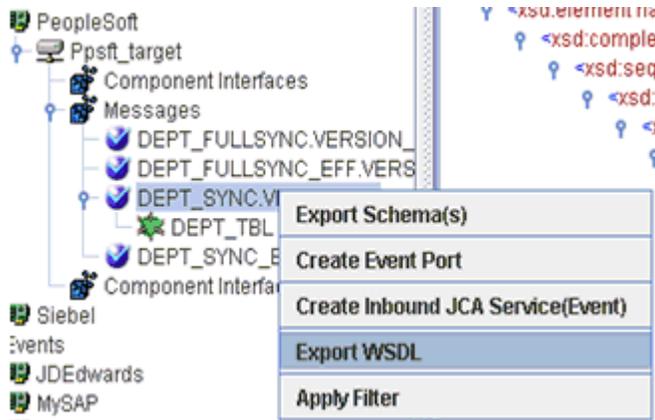
4. Enter a port number in the Listener port field, select **REQUEST\_RESPONSE** from the Synchronization Type drop-down list, and select **UTF-8** from the Encoding Type drop-down list.
5. Click **OK**.



6. Right-click the new channel that you just created and select **Start** from the context menu.

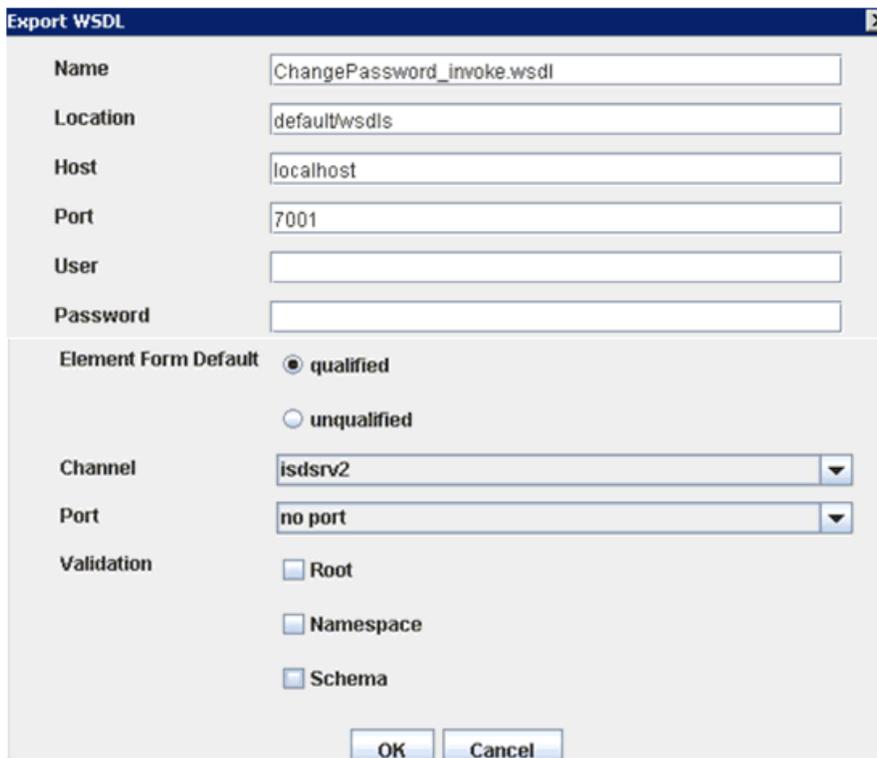


7. Right-click the started channel and select **Stop** from the context menu.



8. Expand the **Messages** node, right-click the **DEPT SYNC.VERSION\_1** message, and select **Export WSDL** from the context menu.

The Export WSDL dialog box is displayed.



9. In the Name field, a default file name for the WSDL file is provided. You can accept the default or provide your own.
10. 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 “/”.
11. In the Host field, enter the name of the machine where Oracle WebLogic Server is running.
12. In the Port field, enter the port for the domain you are using.

13. In the User field, enter your username to access Oracle Service Bus.
14. In the Password field, enter your password to access Oracle Service Bus.
15. Ensure that **qualified** is selected as the element form, which is the default.
16. Select an available channel from the Channel drop-down list.
17. Select **no port** from the Port drop-down list.
18. Select the validation type (if required) by selecting either Root, Namespace, or Schema.
19. Click **OK**.

The WSDL is published to the location specified in the Export WSDL dialog box and is now available for use with a Business Service or Proxy Service in Oracle Service Bus.

## Starting Oracle Service Bus and Creating Project Folders

This section describes how to access the Oracle Service Bus Console 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 port for the default domain is 7001.

The Oracle Service Bus Console logon page is displayed.

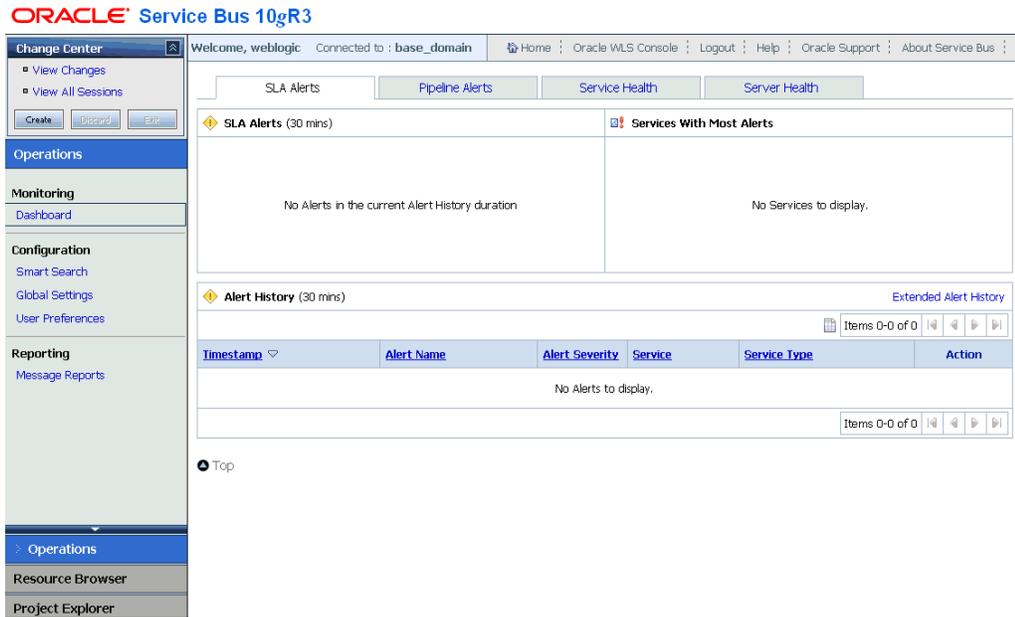
### ORACLE<sup>®</sup> Service Bus 10gR3

Welcome	
Username:	<input type="text" value="weblogic"/>
Password:	<input type="password" value="••••••••"/>
<input type="button" value="Login"/>	

Oracle Service Bus 10gR3  
 Copyright © 2004,2008, Oracle and/or its affiliates. All rights reserved.  
 Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

3. Log on to the Oracle Service Bus Administrative Console using a valid user name and password.

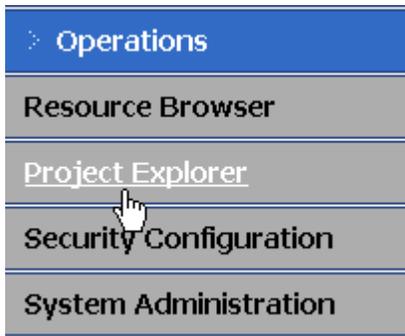
The Oracle Service Bus Console home page is displayed.



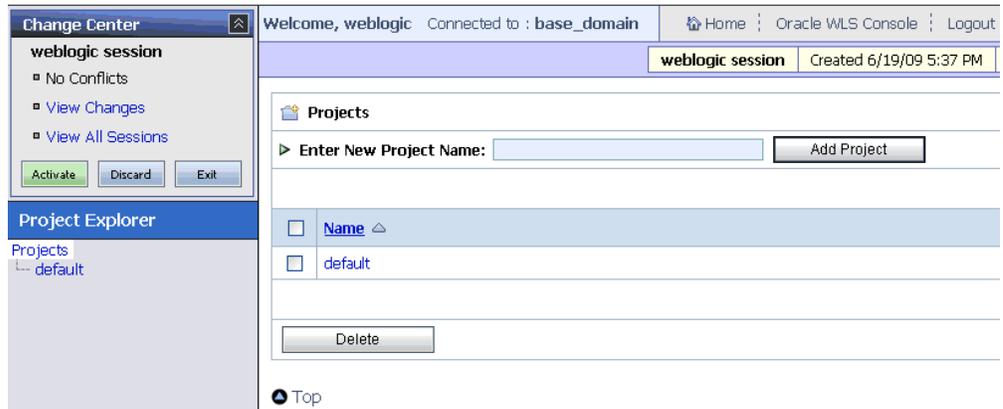
4. Click **Create** in the Change Center area to start a new Oracle Service Bus session.



5. Click **Project Explorer** in the left pane.



The Project Explorer page is displayed.



- Click the **default** project node in the left pane.  
The default project page is displayed.



- In the Enter New Folder Name field, type **Business Service** and click **Add Folder**.  
The Business Service folder is listed in the left pane below the default project node.



- In the Enter New Folder Name field, type **Proxy Service** and click **Add Folder**.  
The Business Service and Proxy Service folders are listed in the left pane below the default project node.



## Configuring a File Type Business Service

This section describes how to configure a File type Business Service using the Oracle Service Bus Console.

1. Select the **Business Service** folder you created in the left pane.



2. In the right pane, select **Business Service** from the Create Resource menu.  
The Create a Business Service - General Configuration page is displayed.

The screenshot shows the 'Create a Business Service (default/Business Service/)' configuration page. The 'General Configuration' section includes the following fields:

- Service Name\***: Psft\_dept\_inbound\_BS
- Description**: (Empty text area)
- Service Type\***:
  - Create a New Service**:
    - WSDL Web Service (with a 'Browse...' button and '(port or binding)' label)
    - Transport Typed Service
    - Messaging Service
    - Any SOAP Service (with a 'SOAP 1.1' dropdown menu)
    - Any XML Service
  - Create From Existing Service**:
    - Business Service (with a 'Browse...' button)
    - Proxy Service (with a 'Browse...' button)

At the bottom of the page are three buttons: 'Next >>', 'Last >>', and 'Cancel'.

3. Provide a name for the Business Service and from the Service Type area select **Messaging Service**.
4. Click **Next**.

5. Select **XML** as the Request Message Type and **None** as the Response Message Type.
6. Click **Next**.

7. Select **file** from the Protocol drop-down list.
8. Enter the path to a destination folder on your file system in the Endpoint URI field and click **Add**.
9. Click **Next**.

Create a Business Service (default/Business Service/Psft_dept_inbound_BS)	
FILE Transport Configuration	
Prefix	psft_dept_inbound_
Suffix	txt
Request encoding	utf-8
<input type="button" value=" &lt;&lt; Prev."/> <input type="button" value=" Next &gt;&gt;"/> <input type="button" value=" Last &gt;&gt;"/> <input type="button" value=" Cancel"/>	

10. Enter the prefix and suffix for the output file to be received and click **Next**.

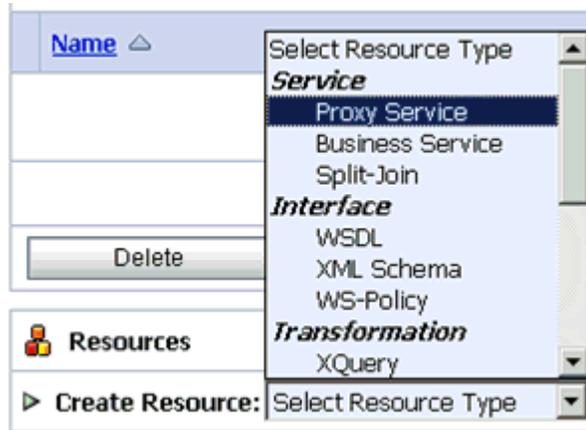
Service Name	Psft_dept_inbound_BS
Description	
Service Type	Messaging Service
Message Type Configuration	
Request Message Type	XML
Response Message Type	None
Transport Configuration	
Protocol	file
Load Balancing Algorithm	round-robin
Endpoint URI	file:///C:/output
Retry Count	0
Retry Iteration Interval	30
FILE Transport Configuration	
Prefix	psft_dept_inbound_
Suffix	txt
Request encoding	utf-8
<input type="button" value=" &lt;&lt; Prev."/> <input type="button" value=" Save"/> <input type="button" value=" Cancel"/>	

11. Review all the information for your Business Service and click **Save**.

## Configuring a Proxy Service

This section describes how to configure a Proxy Service using the Oracle Service Bus Console.

1. Select the **Proxy Service** folder you created in the left pane.



2. In the right pane, select **Proxy Service** from the Create Resource menu.  
The Create a Proxy Service - General Configuration page is displayed.

The screenshot shows the 'Create a Proxy Service (default/Business Service/)' configuration page. The title bar indicates the current page is 'General Configuration'. The form contains the following fields and options:

- Service Name\***: A text input field containing 'Psft\_dept\_inbound\_proxy'.
- Description**: A large text area for entering a description.
- Service Type\***: A section with two sub-sections:
  - Create a New Service**:
    - WSDL Web Service: Includes a text input field and a 'Browse...' button, with '(port or binding)' text below it.
    - Messaging Service
    - Any SOAP Service: Includes a dropdown menu currently set to 'SOAP 1.1'.
    - Any XML Service
  - Create From Existing Service**:
    - Business Service: Includes a text input field and a 'Browse...' button.
    - Proxy Service: Includes a text input field and a 'Browse...' button.

At the bottom of the form are three buttons: 'Next >>', 'Last >>', and 'Cancel'.

3. Provide a name for the Proxy Service and from the Service Type area select **WSDL Web Service**.
4. Click **Browse**.  
The Select a WSDL definition dialog box is displayed.

5. Select an inbound WSDL and click **Submit**, then select any WSDL definition and click **Submit**.

If you select jcabinding, then enter the adapterjndi value in the Endpoint URI manually. If you select the port value, then the adapterjndi value will update automatically in the Endpoint URI field. The default adapterjndi value is eis/OracleJCAAdapter/DefaultConnection.

6. Click **Next**.

Create a Proxy Service (default/Business Service/Psft\_dept\_inbound\_proxy)

**Transport Configuration**

Protocol\*

Endpoint URI\* Format: jca://<resource\_adapter\_jndi>

Get All Headers  
 Yes  
 No  
 Header

HEADER	ACTION
There are no headers configured.	

<< Prev. | Next >> | Last >> | Cancel

7. Click Next.

Create a Proxy Service (default/Business Service/Psft\_dept\_inbound\_proxy)

**JCA Transport Configuration**

Adapter Name

Adapter Type

Dispatch Policy

JNDI Service Account

Always use configuration from JCA WSDL

Connection Mode  
 Managed  
 Non-Managed

Operation Name

Activation Spec Properties

PROPERTY	VALUE
ChannelName	PSFT_CH
AdapterName	PeopleSoft

<< Prev. | Next >> | Last >> | Cancel

8. Click Next.

The screenshot shows a dialog box titled "Create a Proxy Service (default/Business Service/Psft\_dept\_inbound\_proxy)". The main section is "Operation Selection Configuration". On the left, there is a label "Selection Algorithm". On the right, there are five radio button options: "Transport Header", "SOAPAction Header" (which is selected), "WS-Addressing", "SOAP Header", and "SOAP Body Type". At the bottom of the dialog, there are four buttons: "<< Prev.", "Next >>", "Last >>", and "Cancel".

9. Click Next.

The screenshot shows a dialog box titled "Create a Proxy Service (default/Business Service/Psft\_dept\_inbound\_proxy)". The main section is "Message Content Handling". On the left, there is a label "Content Streaming". On the right, there are three settings: "Enabled" (unchecked checkbox), "Buffer Type" (with radio buttons for "Memory Buffer" and "Disk Buffer", both unselected), and "Compression" (with an unchecked checkbox for "Enabled"). At the bottom of the dialog, there are four buttons: "<< Prev.", "Next >>", "Last >>", and "Cancel".

10. Click Next.

Protocol	jca
Endpoint URI	jca://eis/OracleJCAAdapter/DefaultConnection
Get All Headers	No
Headers	
<b>JCA Transport Configuration</b>	
Adapter Name	iWay ERP Adapter
Adapter Type	ERP
Always use configuration from JCA WSDL	true
Connection Mode	Managed
Operation Name	DEPT_SYNC.VERSION_1
Activation Spec Properties	ChannelName = "PSFT_CH" AdapterName = "PeopleSoft"
<b>Operation Selection Configuration</b>	
Selection Algorithm	SOAPAction Header
<b>Message Content Handling Configuration</b>	
Content Streaming	Disabled
<input style="margin-right: 20px;" type="button" value=" &lt;&lt; Prev. "/> <input style="margin-right: 20px;" type="button" value=" Save "/> <input style="margin-right: 20px;" type="button" value=" Cancel "/>	

11. Review all the information for your Proxy Service and click **Save**.

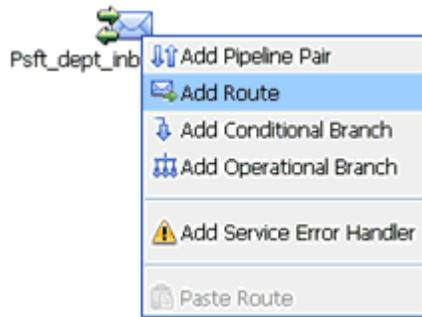
## Configuring a Pipeline

This section describes how to configure a Pipeline using the Oracle Service Bus Console.

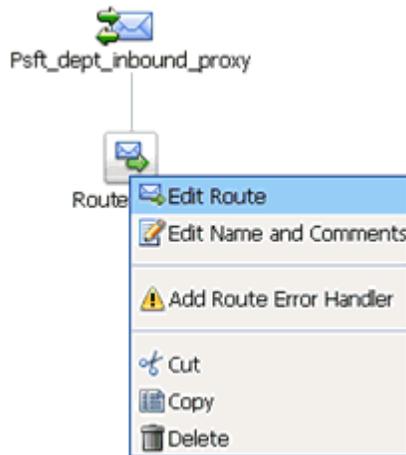
1. Click the **Edit Message Flow** icon in the row of the Proxy Service you created.



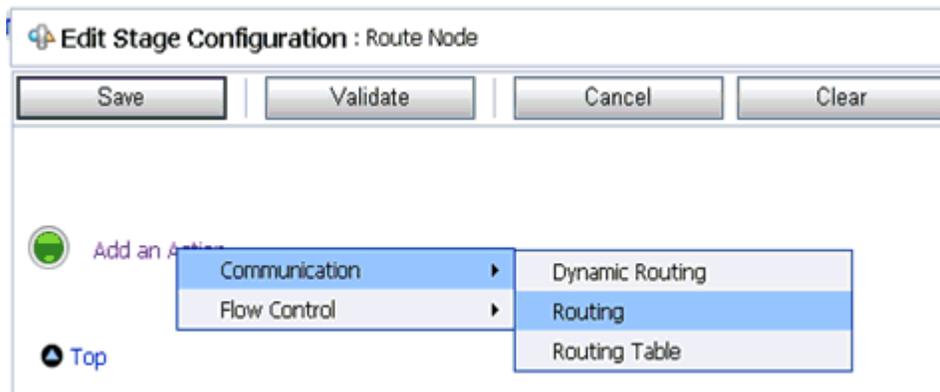
The Edit Message Flow workspace area is displayed.



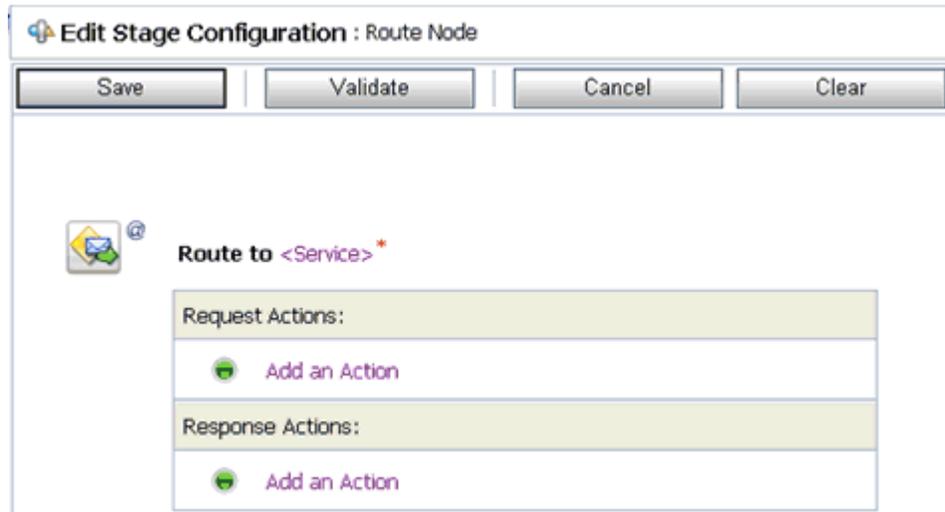
2. Click the **Proxy Service** icon and select **Add Route** from the context menu, as shown in the following image.



3. Click the **RouteNode1** icon and select **Edit Route** from the context menu.

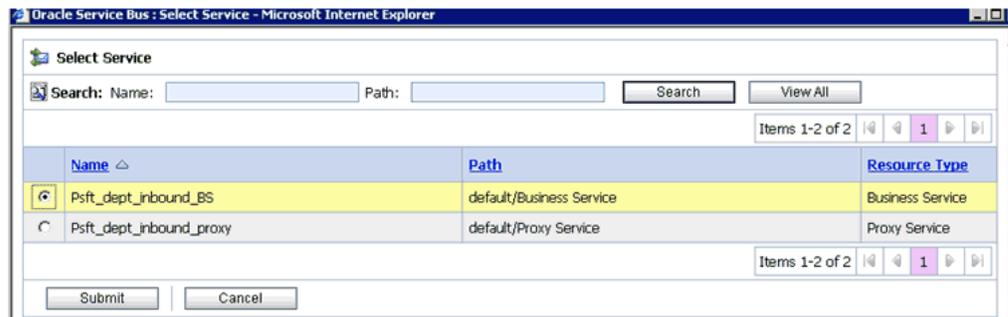


4. Click **Add an Action**, select **Communication** from the context menu, and click **Routing**.



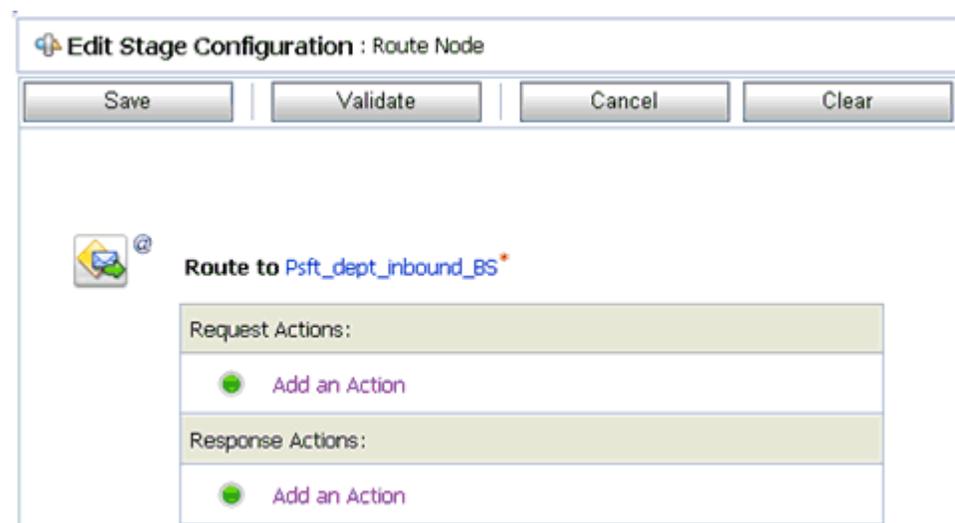
5. Click <Service>.

The Select Service dialog box is displayed.



6. Select a File type Business Service and click **Submit**.

You are returned to the Edit Stage Configuration workspace area.



7. Click **Validate** and then **Save**.



8. Click **Save**.
9. Click **Activate** in the Change Center area to activate your changes in the Oracle Service Bus session.



10. Trigger an event message from the PeopleSoft system and you will receive the output XML in the destination folder.

## Configuring Outbound Processing Using Oracle Service Bus (J2CA Configuration)

This section describes how to configure outbound processing using Oracle Service Bus (J2CA Configuration).

This section includes the following topics:

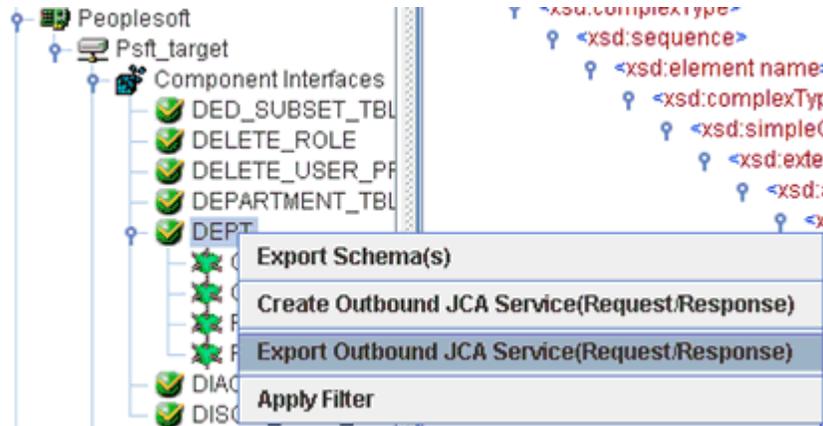
- [Publishing a WSDL From Application Explorer to Oracle Service Bus](#)
- [Starting Oracle Service Bus and Creating Project Folders](#)
- [Configuring a File Type Business Service](#)
- [Configuring a WSDL Type Business Service](#)
- [Configuring a Proxy Service](#)
- [Configuring a Pipeline](#)

### Publishing a WSDL From Application Explorer to Oracle Service Bus

This section describes how to publish a WSDL from Application Explorer (J2CA configuration) to Oracle Service Bus.

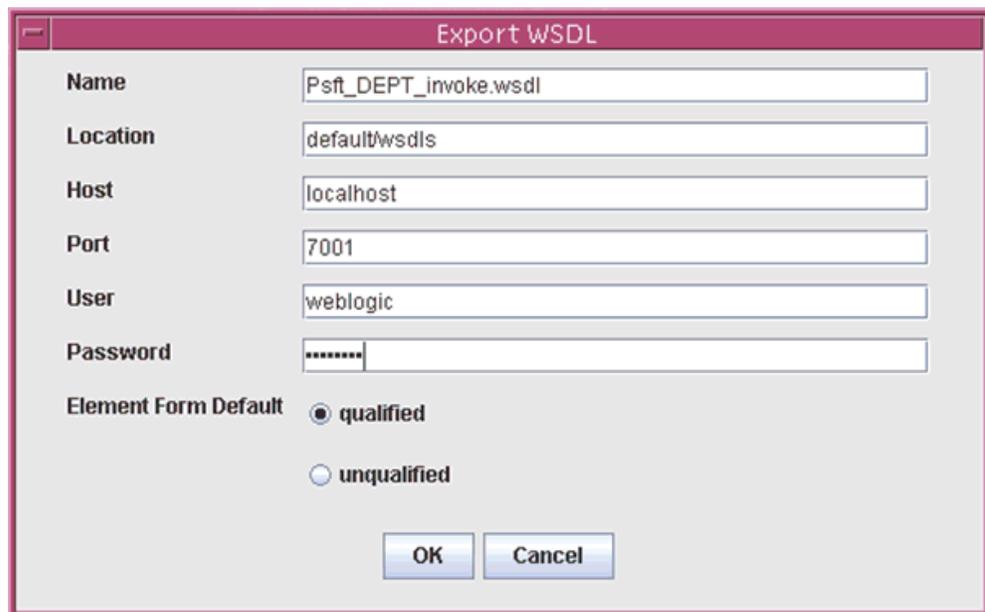
1. Start Application Explorer, connect to a J2CA configuration, and connect to a PeopleSoft target.

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



2. Expand the **Component Interfaces** node and select the **DEPT** component interface.
3. Right-click the **DEPT** component interface and select **Export Outbound JCA Service(Request/Response)** from the context menu.

The Export WSDL dialog box is displayed.



4. In the Name field, a default file name for the WSDL file is provided. You can accept the default or provide your own.
5. 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 “/”.

6. In the Host field, enter the name of the machine where Oracle WebLogic Server is running.
7. In the Port field, enter the port for the domain you are using.
8. In the User field, enter your username to access Oracle Service Bus.

9. In the Password field, enter your password to access Oracle Service Bus.
10. Ensure that **qualified** is selected as the element form, which is the default.
11. Click **OK**.

The WSDL is published to the location specified in the Export WSDL dialog box and is now available for use with a Business Service or Proxy Service in Oracle Service Bus.

## Starting Oracle Service Bus and Creating Project Folders

This section describes how to access the Oracle Service Bus Console 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 port for the default domain is 7001.

The Oracle Service Bus Console logon page is displayed.

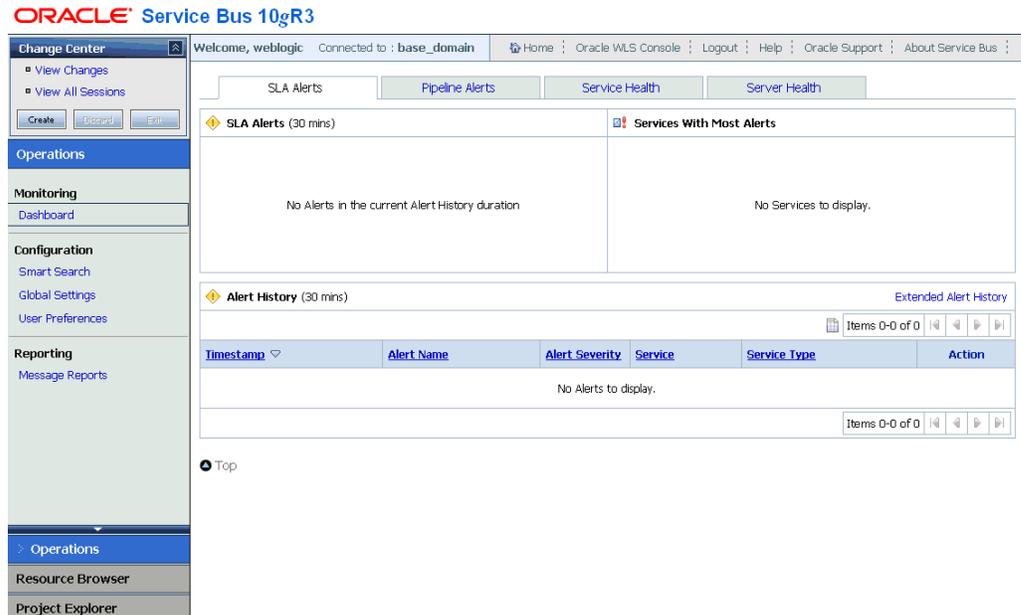
### ORACLE<sup>®</sup> Service Bus 10gR3

<b>Welcome</b>	
<hr/>	
<b>Username:</b>	<input type="text" value="weblogic"/>
<b>Password:</b>	<input type="password" value="●●●●●●●●"/>
<input type="button" value="Login"/>	

Oracle Service Bus 10gR3  
 Copyright © 2004,2008, Oracle and/or its affiliates. All rights reserved.  
 Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

3. Log on to the Oracle Service Bus Administrative Console using a valid user name and password.

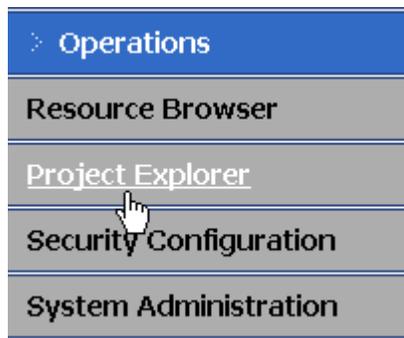
The Oracle Service Bus Console home page is displayed.



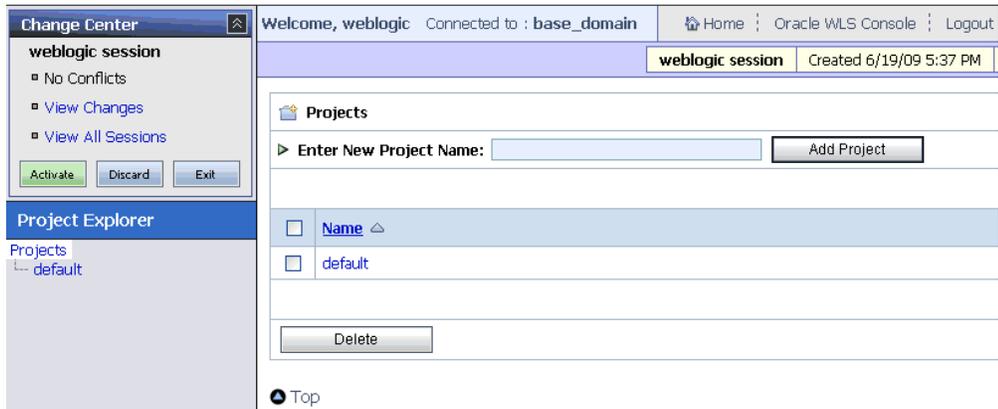
4. Click **Create** in the Change Center area to start a new Oracle Service Bus session.



5. Click **Project Explorer** in the left pane.



The Project Explorer page is displayed.



- Click the **default** project node in the left pane.  
The default project page is displayed.



- In the Enter New Folder Name field, type **Business Service** and click **Add Folder**.  
The Business Service folder is listed in the left pane below the default project node.



- In the Enter New Folder Name field, type **Proxy Service** and click **Add Folder**.  
The Business Service and Proxy Service folders are listed in the left pane below the default project node.



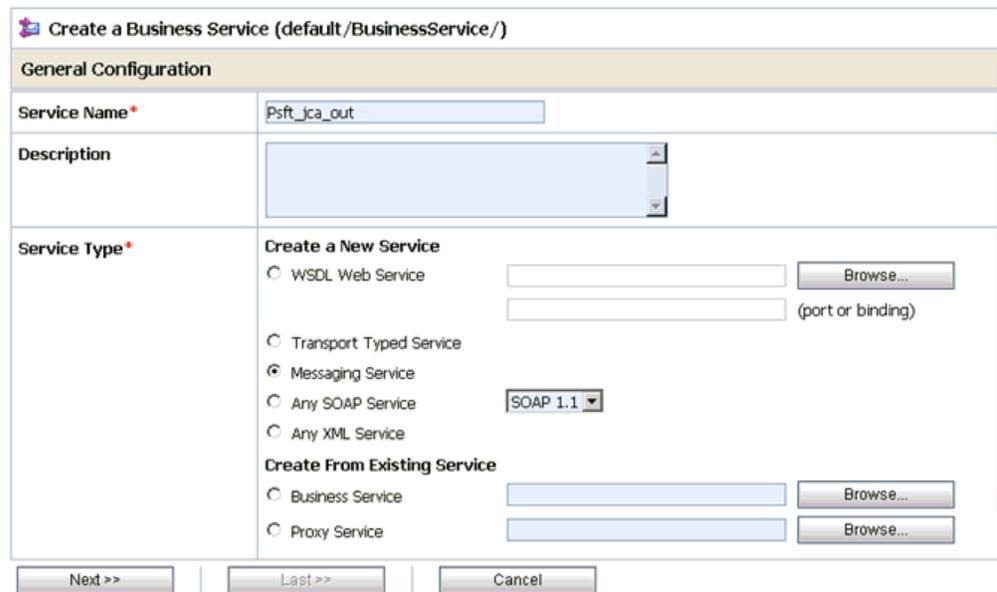
## Configuring a File Type Business Service

This section describes how to configure a File type Business Service using the Oracle Service Bus Console.

1. Select the **Business Service** folder you created in the left pane.



2. In the right pane, select **Business Service** from the Create Resource menu.  
The Create a Business Service - General Configuration page is displayed.



3. Provide a name for the Business Service and from the Service Type area select **Messaging Service**.
4. Click **Next**.

**Create a Business Service (default/BusinessService/Psft\_jca\_out)**

**Message Type Configuration**

**Request Message Type**

- None
- Binary
- Text
- MFL
- XML
- (element or type)

**Response Message Type**

- None
- Binary
- Text
- MFL
- XML
- (element or type)

<< Prev.    Next >>    Last >>    Cancel

5. Select **XML** as the Request Message Type and **None** as the Response Message Type.
6. Click **Next**.

**Create a Business Service (default/BusinessService/Psft\_jca\_out)**

**Transport Configuration**

**Protocol\*** file

**Load Balancing Algorithm** round-robin

**Endpoint URI\*** Format: file:///root-dir/dir1  
file:///

**EXISTING URIS**

- file:///rdbms/cra102/Chennai\_QA/osb/output

**Retry Count** 0

**Retry Iteration Interval** 30

<< Prev.    Next >>    Last >>    Cancel

7. Select **file** from the Protocol drop-down list.
8. Enter the path to a destination folder on your file system in the Endpoint URI field and click **Add**.
9. Click **Next**.

**Create a Business Service (default/BusinessService/Psft\_jca\_out)**

**FILE Transport Configuration**

**Prefix** psft\_dept\_jca\_outbol

**Suffix** out.xml

**Request encoding** utf-8

<< Prev.    Next >>    Last >>    Cancel

10. Enter the prefix and suffix for the output file to be received and click **Next**.

Message Type Configuration	
Request Message Type	XML
Response Message Type	None
Transport Configuration	
Protocol	file
Load Balancing Algorithm	round-robin
Endpoint URI	file:///rdbms/ora102/Chennai_QA/osb/output
Retry Count	0
Retry Iteration Interval	30
FILE Transport Configuration	
Prefix	psft_dept_jca_outbound
Suffix	out.xml
Request encoding	utf-8

11. Review all the information for your Business Service and click **Save**.

## Configuring a WSDL Type Business Service

This section describes how to configure a WSDL type Business Service using the Oracle Service Bus Console.

1. Select the **Business Service** folder you created in the left pane.



2. In the right pane, select **Business Service** from the Create Resource menu. The Create a Business Service - General Configuration page is displayed.

**Create a Business Service (default/BusinessService/)**

**General Configuration**

Service Name\*

Description

Service Type\*

**Create a New Service**

WSDL Web Service    
 (port or binding)

Transport Typed Service

Messaging Service

Any SOAP Service

Any XML Service

**Create From Existing Service**

Business Service

3. Provide a name for the Business Service and from the Service Type area select **WSDL Web Service**.
4. Click **Browse**.

The Select a WSDL Definition dialog box is displayed.

**Select a WSDL definition**

Search: Name:  Path:    [Adv. Search](#)

Name	Path	WSDL Namespace
Psft_DEPT_invoke	default/wsdl	http://xmlns.oracle.com/pcbpel/iWay/wsdl...

Description:

> Select WSDL definitions

**Bindings**  
jcabinding

**Parts**  
DEPT

5. Select an outbound WSDL and click **Submit**, then select any WSDL definition and click **Submit**.

If you select jcabinding, then enter the adapterjndi value in the Endpoint URI manually. If you select the port value, then the adapterjndi value will update automatically in the Endpoint URI field. The default adapterjndi value is eis/OracleJCAAdapter/DefaultConnection.

Create a Business Service (default/BusinessService/)

**General Configuration**

Service Name\* Psft\_Dept\_Jca\_BS

Description

Service Type\*

**Create a New Service**

WSDL Web Service    
 (port)

Transport Typed Service

Messaging Service

Any SOAP Service

Any XML Service

**Create From Existing Service**

Business Service

Proxy Service

6. Click Next.

Create a Business Service (default/BusinessService/Psft\_Dept\_Jca\_BS)

**Transport Configuration**

Protocol\*

Load Balancing Algorithm

Endpoint URI\*

Format: `jca://<resource_adapter_jndi>`  
  
**EXISTING URIS**

Retry Count

Retry Iteration Interval

Retry Application Errors  Yes  No

7. Select `jca` from the Protocol drop-down list.

8. Click Next.

Create a Business Service (default/BusinessService/Psft\_Dept\_Jca\_BS)

**JCA Transport Configuration**

Adapter Name	iWay ERP Adapter				
Adapter Type	ERP				
Dispatch Policy	default				
JNDI Service Account					
Always use configuration from JCA WSDL	<input checked="" type="checkbox"/>				
Connection Mode	<input checked="" type="radio"/> Managed <input type="radio"/> Non-Managed				
Operation Name	DEPT				
Interaction Spec Properties	<table border="1"> <thead> <tr> <th>PROPERTY</th> <th>VALUE</th> </tr> </thead> <tbody> <tr> <td>FunctionName</td> <td>PROCESS</td> </tr> </tbody> </table>	PROPERTY	VALUE	FunctionName	PROCESS
PROPERTY	VALUE				
FunctionName	PROCESS				

9. Click Next.

**Transport Configuration**

Protocol	jca
Load Balancing Algorithm	round-robin
Endpoint URI	jca://eis/OracleJCAAdapter/DefaultConnection
Retry Count	0
Retry Iteration Interval	30
Retry Application Errors	Yes

**JCA Transport Configuration**

Adapter Name	iWay ERP Adapter
Adapter Type	ERP
Always use configuration from JCA WSDL	true
Connection Mode	Managed
Operation Name	DEPT
Interaction Spec Properties	FunctionName = "PROCESS"

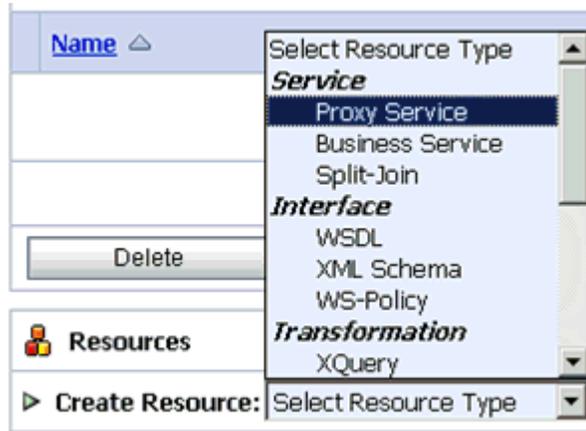
 
 

10. Review all the information for your Business Service and click **Save**.

## Configuring a Proxy Service

This section describes how to configure a Proxy Service using the Oracle Service Bus Console.

1. Select the **Proxy Service** folder you created in the left pane.



2. In the right pane, select **Proxy Service** from the Create Resource menu.  
The Create a Proxy Service - General Configuration page is displayed.

The screenshot shows the 'Create a Proxy Service (default/ProxyService/)' configuration page. The page has a title bar and a 'General Configuration' section. The 'Service Name' field contains 'Psft\_Dept\_jca\_PS'. The 'Description' field is empty. The 'Service Type' section has two main areas: 'Create a New Service' and 'Create From Existing Service'. Under 'Create a New Service', there are radio buttons for 'WSDL Web Service', 'Messaging Service', 'Any SOAP Service', and 'Any XML Service'. The 'Any XML Service' option is selected. There is a dropdown menu for 'SOAP 1.1'. Under 'Create From Existing Service', there are radio buttons for 'Business Service' and 'Proxy Service'. At the bottom, there are three buttons: 'Next >>', 'Last >>', and 'Cancel'.

3. Provide a name for the Proxy Service and from the Service Type area select **Any XML Service**.
4. Click **Next**.

**Create a Proxy Service (default/ProxyService/Psft\_Dept\_jca\_PS)**

**Transport Configuration**

**Protocol\*** file

**Endpoint URI\*** Format: file:///root-dir/dir1  
file:///rdbms/ora102/Chennai\_QA/osb/input

**Get All Headers**  
 Yes  
 No  
 Header    

HEADER	ACTION
There are no headers configured.	

<< Prev. | Next >> | Last >> | Cancel

5. Select **file** from the Protocol drop-down list.
6. Enter the path to an input folder on your file system in the Endpoint URI field.
7. Click Next.

**Create a Proxy Service (default/ProxyService/Psft\_Dept\_jca\_PS)**

**FILE Transport Configuration**

**File Mask\*** \*.\*

**Polling Interval\*** 6

**Read Limit\*** 10

**Sort By Arrival**

**Scan SubDirectories**

**Pass By Reference**

**Post Read Action\*** delete

**Stage Directory\*** i\_QA/osb/input/stage

**Archive Directory**

**Error Directory\*** ai\_QA/osb/input/error

**Request encoding** utf-8

<< Prev. | Next >> | Last >> | Cancel

8. Provide any folder locations on your file system for the Stage Directory and Error Directory fields.
9. Click Next.

**Create a Proxy Service (default/ProxyService/Psft\_Dept\_jca\_PS)**

**Message Content Handling**

<b>Content Streaming</b>	<input type="checkbox"/> Enabled <b>Buffer Type</b> <input checked="" type="radio"/> Memory Buffer <input type="radio"/> Disk Buffer <b>Compression</b> <input type="checkbox"/> Enabled
--------------------------	---

<< Prev. | Next >> | Last >> | Cancel

10. Click Next.

**FILE Transport Configuration**

File Mask	*.*
Polling Interval	6
Read Limit	10
Sort By Arrival	false
Scan SubDirectories	false
Pass By Reference	false
Post Read Action	delete
Stage Directory	/rdbms/ora102/Chennai_QA/osb/input/stage
Error Directory	/rdbms/ora102/Chennai_QA/osb/input/error
Request encoding	utf-8

**Message Content Handling Configuration**

Content Streaming	Disabled
-------------------	----------

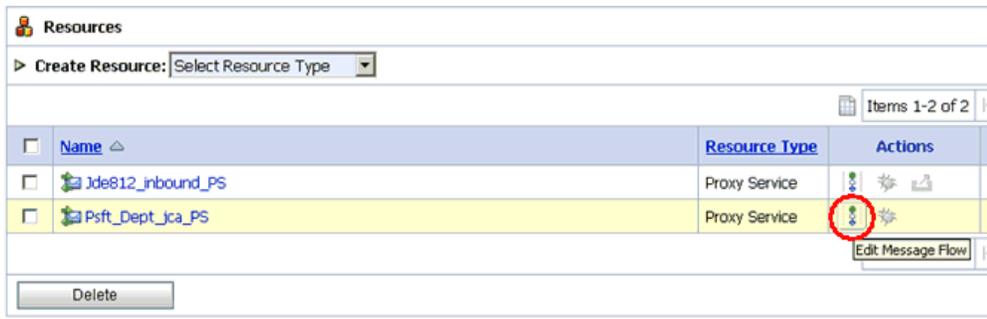
<< Prev. | Save | Cancel

11. Review all the information for your Proxy Service and click **Save**.

## Configuring a Pipeline

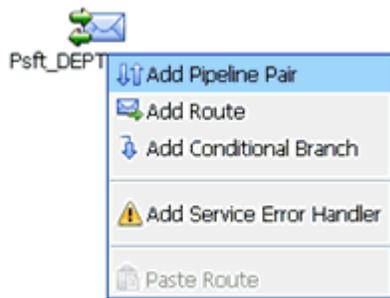
This section describes how to configure a Pipeline using the Oracle Service Bus Console.

1. Click the **Edit Message Flow** icon in the row of the Proxy Service you created.



The Edit Message Flow workspace area is displayed.

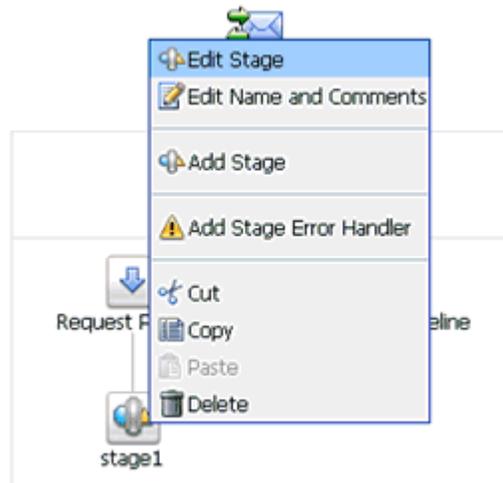
- Click the **Proxy Service** icon and select **Add Pipeline Pair** from the context menu, as shown in the following image.



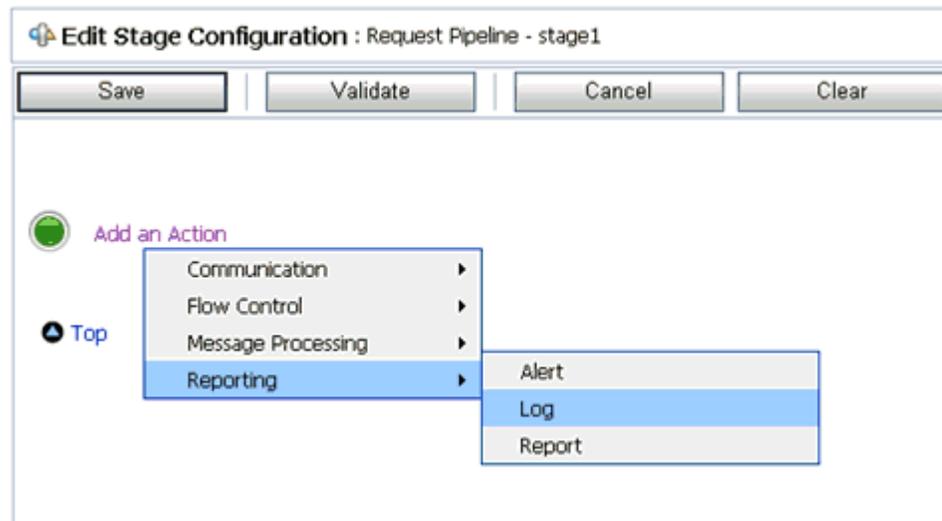
- Click the **Request Pipeline** icon and select **Add Stage** from the context menu.



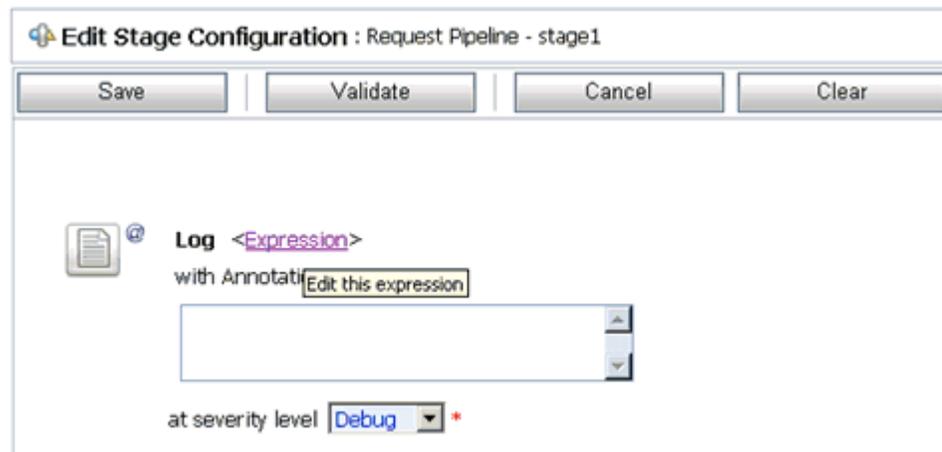
The Stage1 icon is added below the Request Pipeline icon.



4. Click the **Stage1** icon and select **Edit Stage** from the context menu. The Edit Stage Configuration workspace area is displayed.

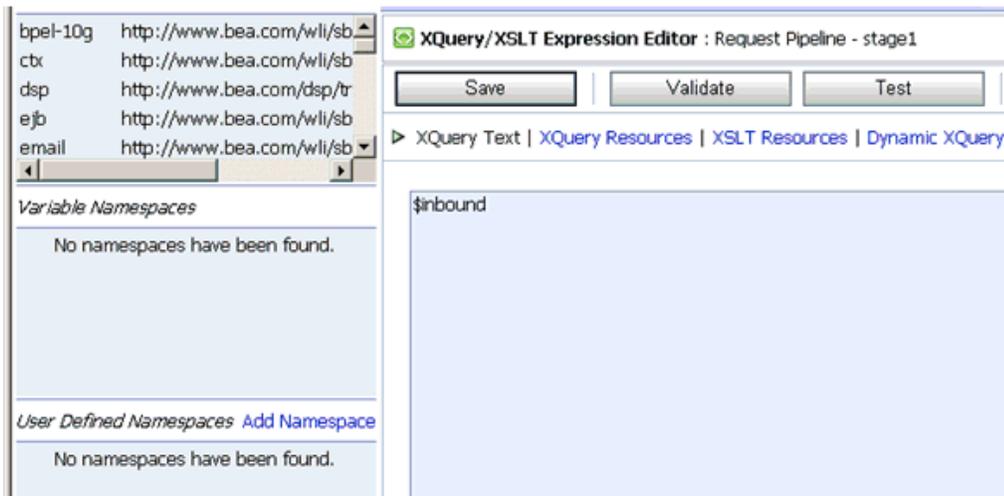


5. Click **Add an Action**, select **Reporting** from the context menu, and click **Log**.



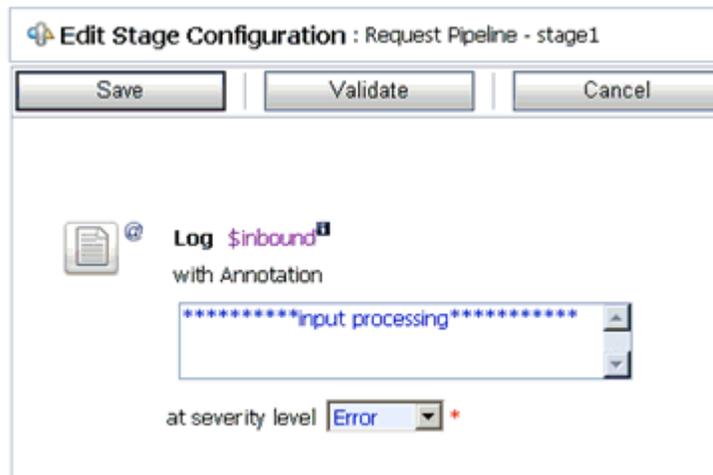
6. Click **<Expression>** to edit the expression.

The XQuery/XSLT Expression Editor is displayed.

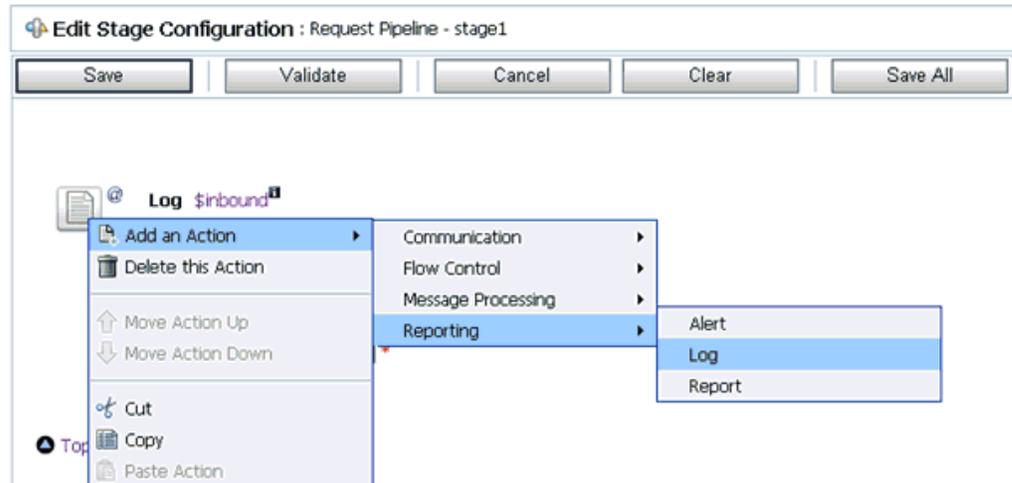


7. In the XQuery Text area, type **\$inbound**.
8. Click **Validate** and then **Save**.

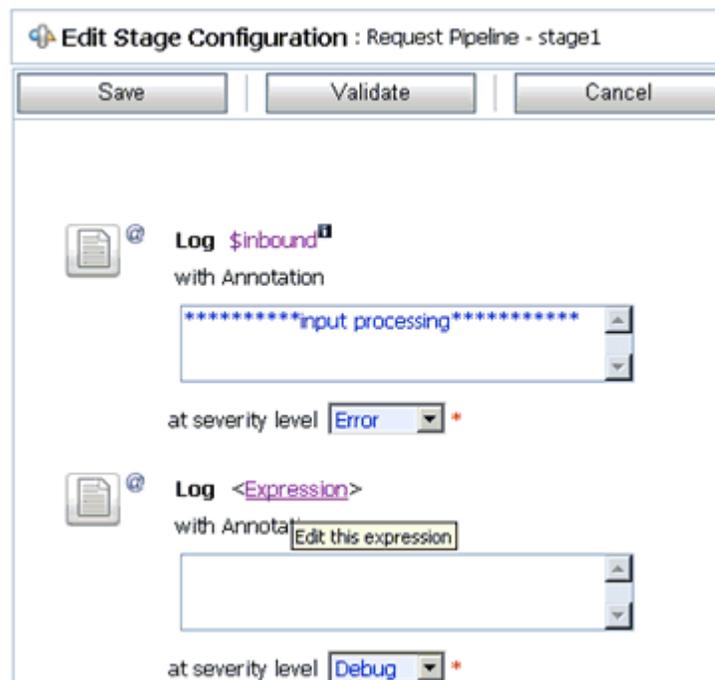
You are returned to the Edit Stage Configuration workspace area.



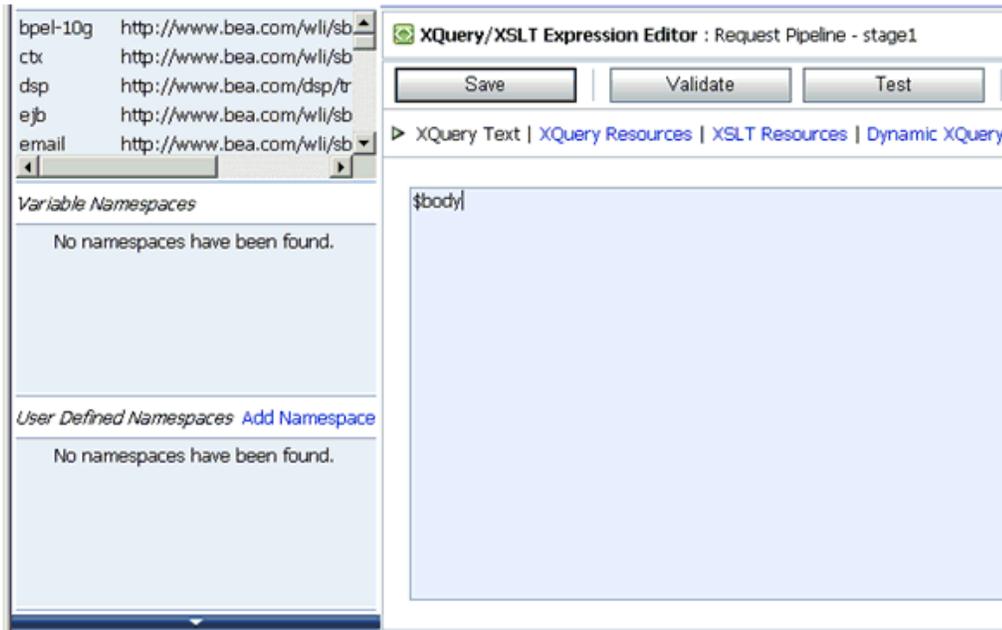
9. Type any annotation/comments in the text box (for example, \*\*\*\*\*input processing\*\*\*\*\*).
10. Select **Error** from the severity level drop-down list.
11. Add one more Log action as shown in the following image.



A new Log configuration is added, as shown in the following image.



12. Click **<Expression>** to edit the expression.  
The XQuery/XSLT Expression Editor is displayed.



13. In the XQuery Text area, type **\$body**.

14. Click **Validate** and then **Save**.

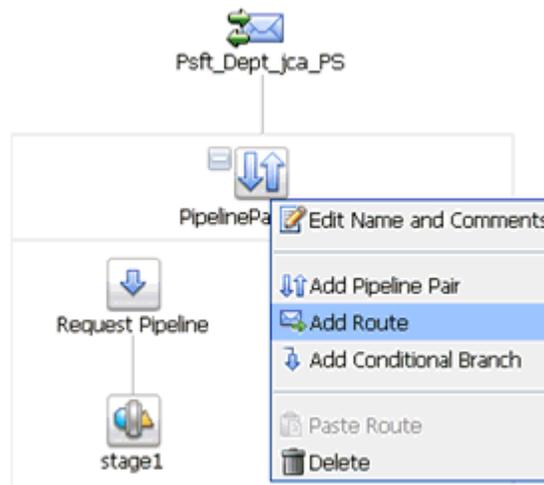
You are returned to the Edit Stage Configuration workspace area.



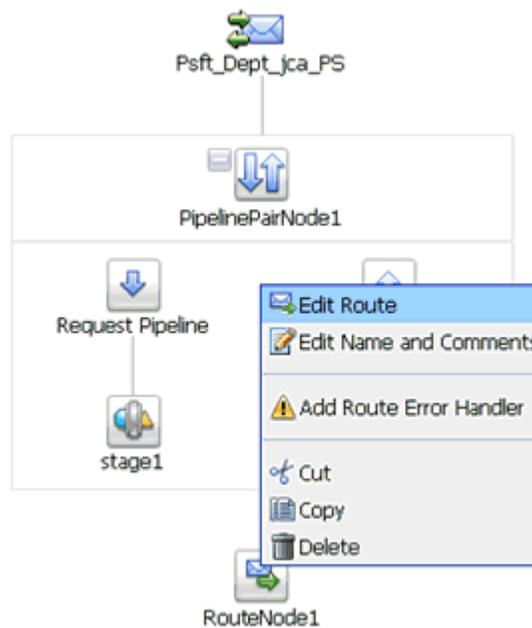
15. Type any annotation/comments in the text box (for example, **\*\*\*\*\*Request Body\*\*\*\*\***).

16. Select **Error** from the severity level drop-down list.

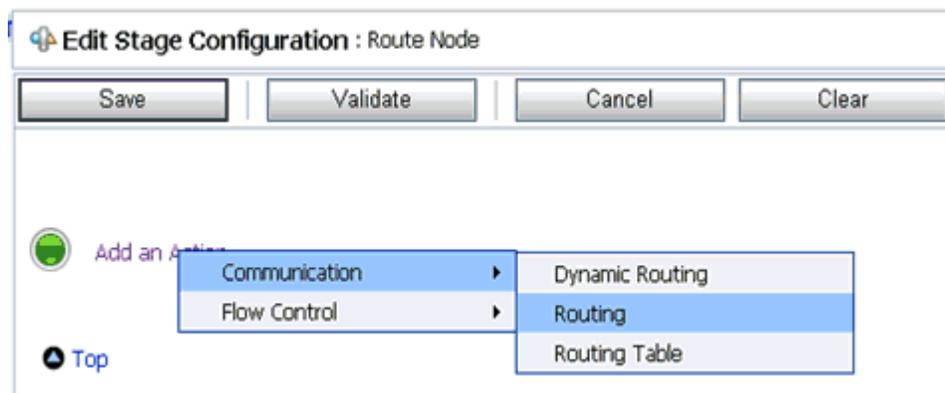
17. Click **Validate** and then **Save**.



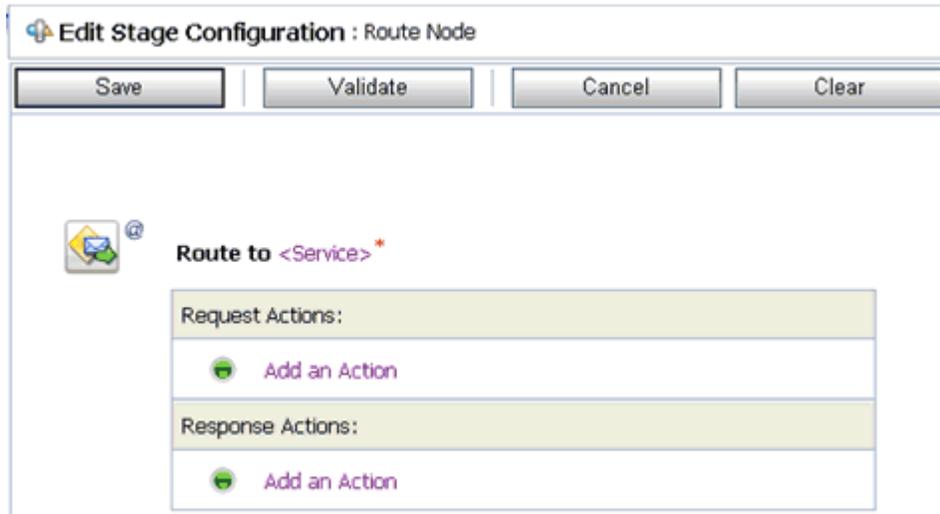
18. Click the **PipelinePairNode1** icon and select **Add Route** from the context menu.



19. Click the **RouteNode1** icon and select **Edit Route** from the context menu.

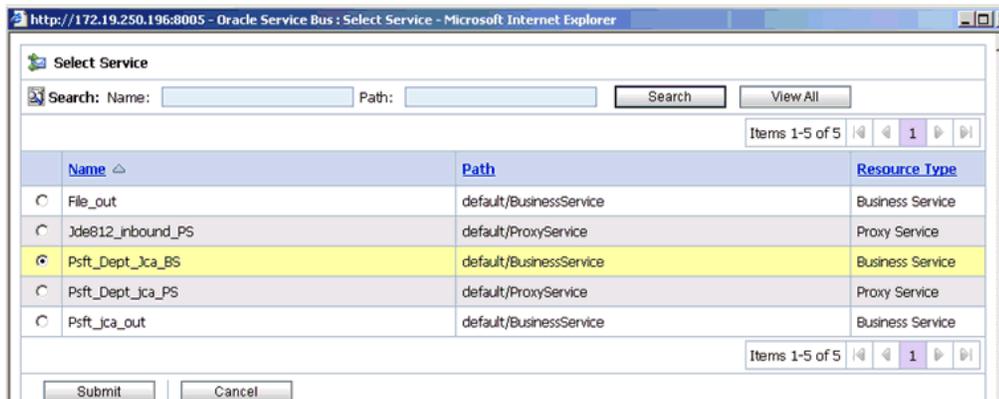


- Click **Add an Action**, select **Communication** from the context menu, and click **Routing**.



- Click **<Service>**.

The Select Service dialog box is displayed.

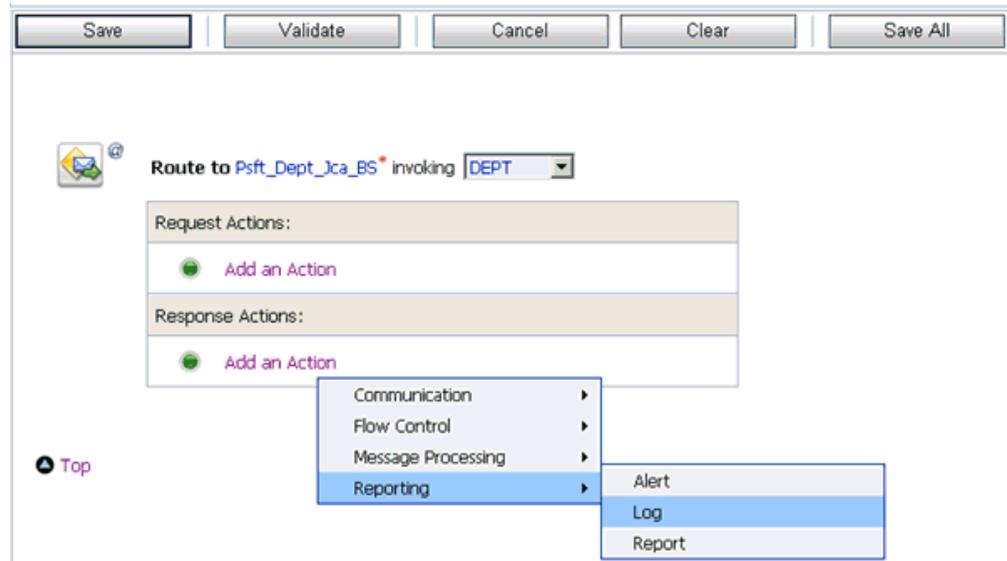


- Select a WSDL type Business Service and click **Submit**.

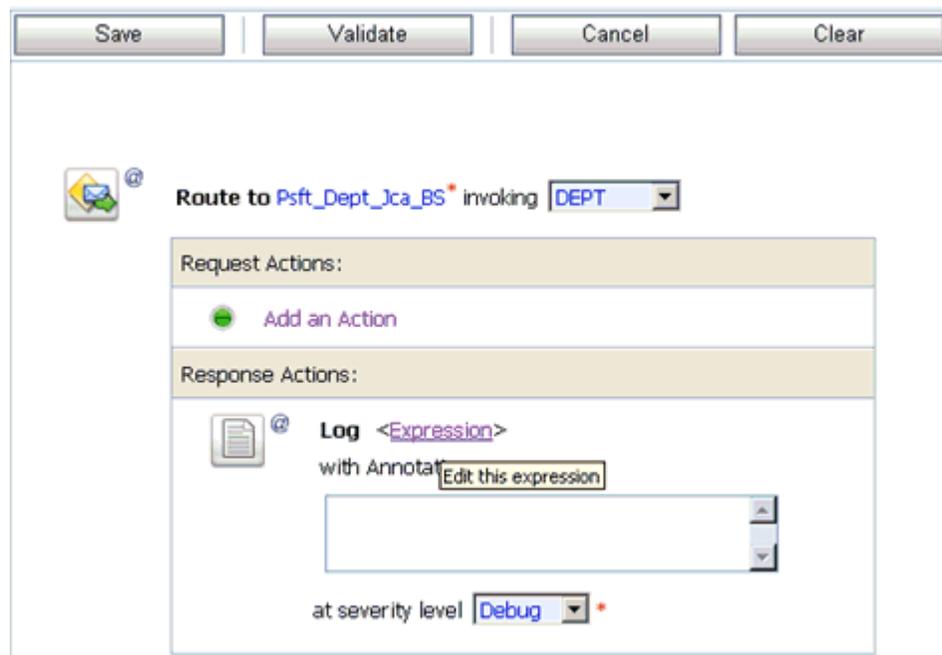
You are returned to the Edit Stage Configuration workspace area.



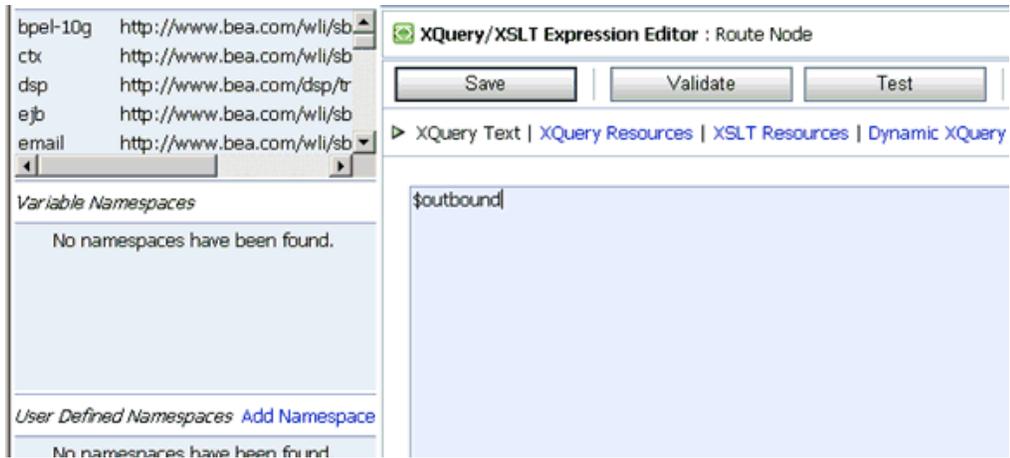
- Select **DEPT** as the operational attribute from the drop-down list.
- Click **Validate** and then **Save**.



25. In the Response Actions area, click **Add an Action**, select **Reporting** from the context menu, and click **Log**.



26. Click **<Expression>** to edit the expression.  
The XQuery/XSLT Expression Editor is displayed.



27. In the XQuery Text area, type **\$outbound**.

28. Click **Validate** and then **Save**.

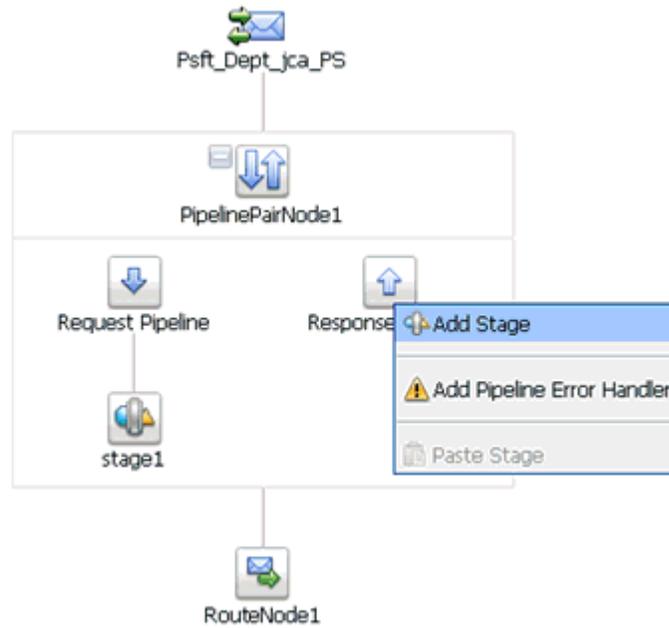
You are returned to the Edit Stage Configuration workspace area.



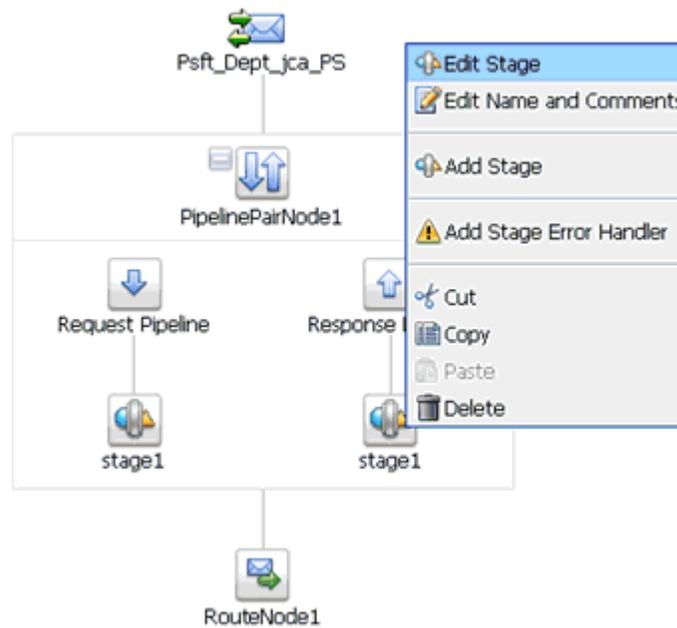
29. Type any annotation/comments in the text box (for example, **\*\*\*\*\*output processing\*\*\*\*\***).

30. Select **Error** from the severity level drop-down list.

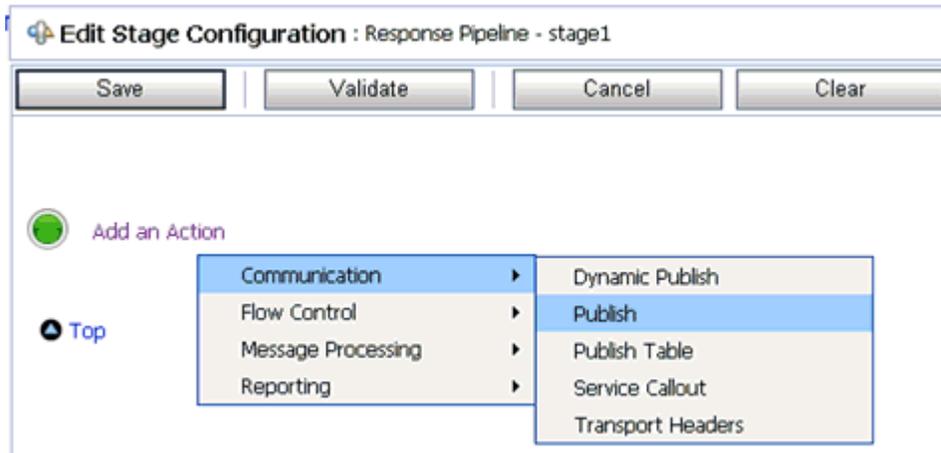
31. Click **Validate** and then **Save**.



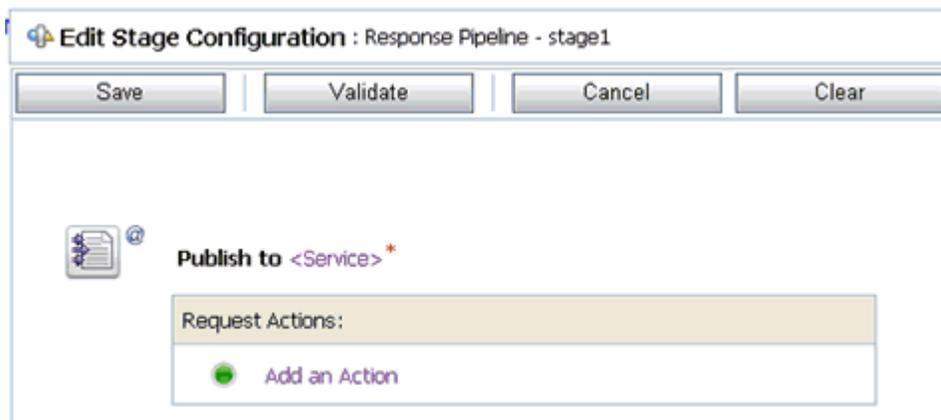
32. Click the **Response Pipeline** icon and select **Add Stage** from the context menu. The Stage1 icon is added below the Response Pipeline icon.



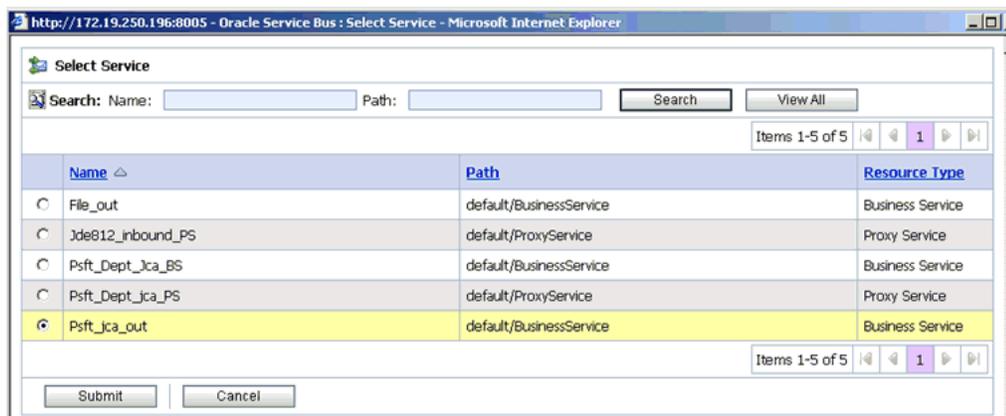
33. Click the **Stage1** icon and select **Edit Stage** from the context menu. The Edit Stage Configuration workspace area is displayed.



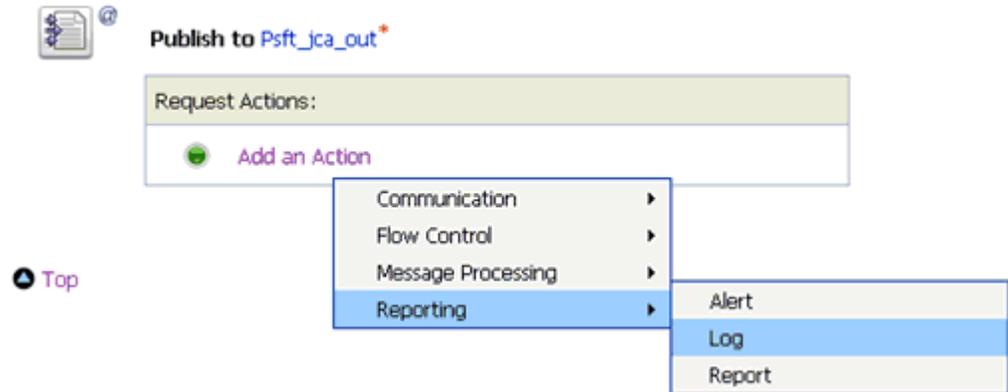
34. Click **Add an Action**, select **Communication** from the context menu, and click **Publish**.



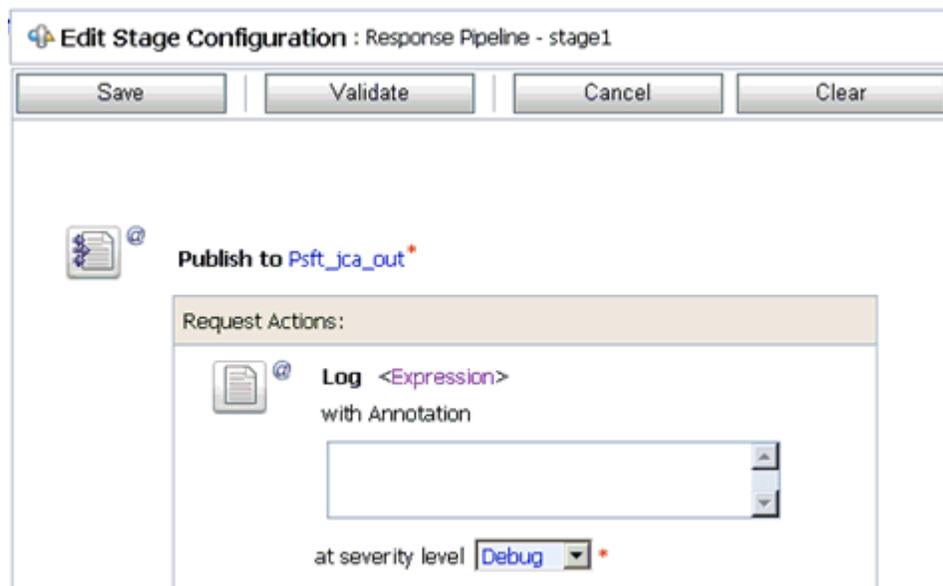
35. Click **<Service>**.  
The Select Service dialog box is displayed.



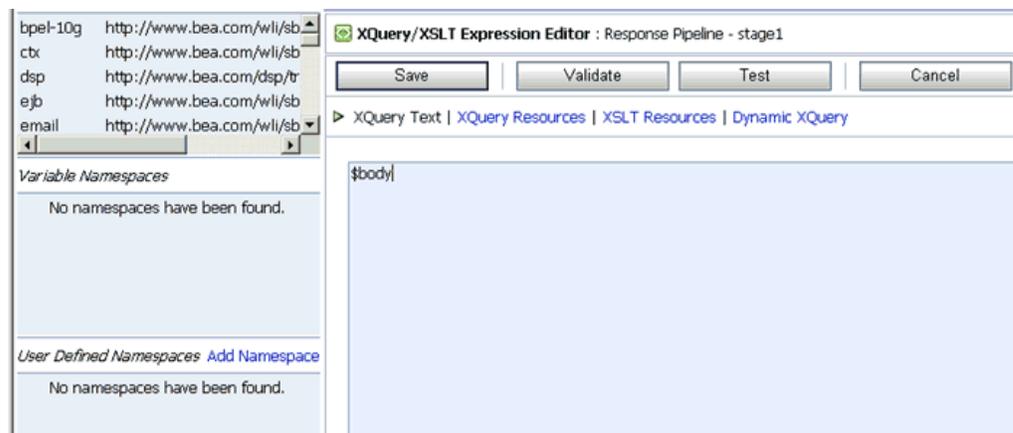
36. Select a File type Business Service and click **Submit**.  
You are returned to the Edit Stage Configuration workspace area.



37. In the Request Actions area, click **Add an Action**, select **Reporting** from the context menu, and click **Log**.



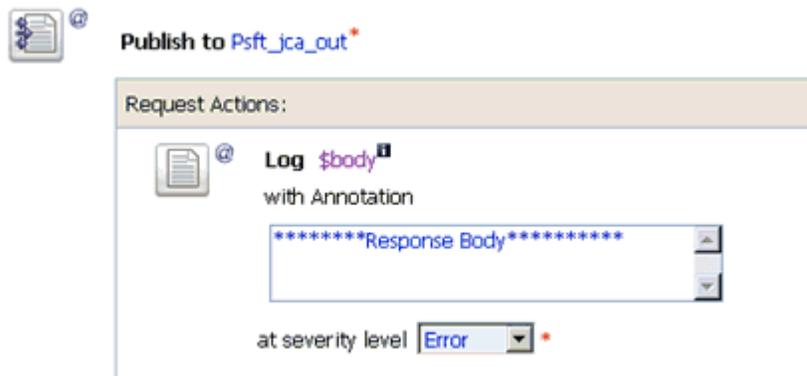
38. Click **<Expression>** to edit the expression.  
The XQuery/XSLT Expression Editor is displayed.



39. In the XQuery Text area, type **\$body**.

40. Click **Validate** and then **Save**.

You are returned to the Edit Stage Configuration workspace area.



41. Type any annotation/comments in the text box (for example, \*\*\*\*\*Response Body\*\*\*\*\*).

42. Select **Error** from the severity level drop-down list.

43. Click **Validate** and then **Save**.



44. Click **Save**.

45. Click **Activate** in the Change Center area to activate your changes in the Oracle Service Bus session.



46. Copy and paste an input XML file in the input folder you have configured to receive an output XML file in the destination folder.



---

---

## Troubleshooting and Error Messages

This chapter explains the limitations and workarounds when connecting to PeopleSoft. The following topics are discussed:

- [Troubleshooting](#)
- [BSE Error Messages](#)

The adapter-specific errors listed in this chapter can arise whether using the adapter with an OracleWLS Adapter J2CA or with an OracleWLS Adapter Business Services Engine (BSE) configuration.

### Troubleshooting

This topic provides troubleshooting information for PeopleSoft, separated into four categories:

- Application Explorer
- PeopleSoft
- OracleWLS Adapter J2CA
- BSE

---

---

**Note:** Log file information that can be relevant in troubleshooting can be found in the following locations:

- The OracleWLS Adapter J2CA trace information can be found under the `wls_home\erp-adapters\config\config_name\log` directory.
  - BSE trace information can be found under the `wls_home\erp-adapters\ibse.war\ibselogs` directory.
  - The log file for Application Explorer can be found under the `wls_home\erp-adapters\tools\iwaeb\bin` directory.
- 
- 

#### Application Explorer

To use Application Explorer on **Windows** for debugging or testing purposes, load the batch script `ae.bat`, found under:

```
wls_home\erp-adapters\tools\iwaeb\bin
```

On **UNIX**, load the shell script `iwaeb.sh`, found under:

```
wls_home/erp-adapters/tools/iwaeb/bin
```

Error	Solution
<p>Cannot connect to OracleWLS Application Adapter for PeopleSoft from Application Explorer. The following error message appears:</p> <p>Problem activating adapter</p>	<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> <p>java.lang.IllegalStateException: java.lang.Exception: Error Logon to PeopleSoft System</p>	<p>You have provided invalid connection information for PeopleSoft or the wrong <code>psjoe.jar</code> is in the <code>lib</code> directory.</p> <p>The <code>psjoe.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>Logon failure error at run-time.</p>	<p>Ensure that the PeopleSoft JAR files, <code>iwpsci84.jar</code> (or <code>iwpsci81.jar</code>) and <code>psjoe.jar</code>, are added to the <code>lib</code> directory.</p> <p>If the password for connecting to your PeopleSoft system is not specified when creating a target or with the Edit option in Application Explorer, you will be unable to connect to PeopleSoft. 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> <p>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</p>	<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>iwpsci8x.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>iwpsci81.jar</code> and <code>iwpsci84.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> <p>java.lang.ClassNotFoundException: org.bouncycastle.jce.provider.BouncyCastleProvider</p>	<p>This is a benign exception. It does not affect adapter functionality. Download BouncyCastle files from:</p> <p><code>ftp://ftp.bouncycastle.org/pub</code></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=/&lt;jdk_home&gt;/bin/java, where &lt;jdk_home&gt; is the directory where JDK is installed on your system.</p>

### 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>

### OracleWLS Adapter J2CA

Error	Solution
<p>In Application Explorer, the following error message appears when you attempt to connect to an OracleWLS 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>wls_ home\erp-adapters\tools\iwae\bin\ .\..\..\</pre>

## BSE Error Messages

This topic discusses the different types of errors that can occur when processing Web services through Oracle WebLogic Server Adapter Business Services Engine (BSE).

## 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.

## 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.

### OracleWLS 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>
```

### Empty Result From PeopleSoft Request

If OracleWLS 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>
```

### Failure to Connect to PeopleSoft

If OracleWLS 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>
```

### 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>
```

### 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>
```



---

---

## Advanced User Tools

This chapter includes the following topics:

- [Web Services Policy-Based Security](#)
- [Migrating Repositories](#)

### Web Services Policy-Based Security

OracleWLS Adapter Application Explorer (Application Explorer) provides a security model called Web services policy-based security. The following topics describe how the feature works and how to configure it.

Web services provide a layer of abstraction between the back-end business logic and the user or application running the Web service. Easy application integration is enabled. However, the issue of controlling the use and implementation of critical and sensitive business logic that is run as a Web service is raised.

Application Explorer controls the use of Web services that use adapters, using a feature called policy-based security. This feature enables an administrator to apply policies to business services (Web services) to deny or permit their execution.

A policy is a set of privileges dealing with the execution of a business service that can be applied to an existing or new business service. When you set specific rights or privileges inside a policy, you do not have to re-create privileges for every business service that has security concerns in common with other business services. Instead, you reuse a policy on multiple business services.

The goal of the feature is to secure requests at both the transport and the SOAP request level transmitted on the wire. Some of the policies do not deal with security issues directly, but do affect the run-time behavior of the Web services to which they have been applied.

The BSE administrator creates an "instance" of a policy type, names it, associates individual users or groups (a collection of users), and then applies that policy to one or more business services.

You can assign a policy to a business service, or to a method within a business service. If a policy is only applied to a method, other methods in that business service will not be governed by it. However, if a policy is applied to the business service, all methods are governed by it. At run-time, the user ID and password that are sent to BSE in the SOAP request message are checked against the list of users for all policies applied to that specific business service. The policy type that is supported is Resource Execution, which dictates who can or cannot execute the business service.

When a policy is not applied, the default value for a business service is to "grant all". For example, anybody can execute the business service, until the Resource Execution

policy is associated to the business service. At that time, only those granted execution permissions, or users not part of the group that has been denied execution permissions, have access to the business service.

## Configuring Web Services Policy-Based Security

The following procedures describe how to configure Web services policy-based security.

### Creating and Associating a User with a Policy

Before you create instances of policies, you must have a minimum of one user or one group to associate to an instance. You can create users and groups using Application Explorer.

1. Start Application Explorer.
2. Right-click the configuration to which you want to connect, for example, `SampleConfig`. See [Chapter 2, "Configuring OracleWLS Application Adapter for PeopleSoft"](#) for information on creating a new configuration.
3. Select **Connect**.

Nodes appear for Adapters, Events, and Business Services (also known as Web services).

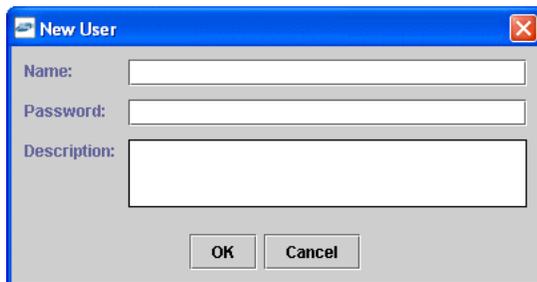


- a. Expand the **Business Services** node.
- b. Expand the **Configuration** node.
- c. Expand the **Security** node.
- d. Expand the **Users and Groups** node.



4. Right-click **Users** and click **New User**.

The New User dialog box is displayed.



- a. In the **Name** field, enter a user ID.
- b. In the **Password** field, enter the password associated with the user ID.

- c. In the **Description** field, enter a description of the user (optional).
5. Click **OK**.



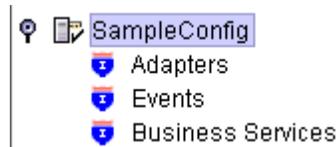
The new user is added under the Users node.

### Creating a Group to Use With a Policy

To create a group to use with a policy:

1. Start Application Explorer.
2. Right-click the configuration to which you want to connect, for example, SampleConfig. See [Chapter 2, "Configuring OracleWLS Application Adapter for PeopleSoft"](#) for information on creating a new configuration.
3. Select **Connect**.

Nodes appear for Adapters, Events, and Business Services (also known as Web services).



- a. Expand the **Business Services** node.
- b. Expand the **Configuration** node.
- c. Expand the **Security** node.
- d. Expand the **Users and Groups** node.



4. Right-click **Groups** and select **New Group**.  
The New Group dialog box is displayed.



- a. In the **Name** field, enter a name for the group.
  - b. In the **Description** field, enter a description for the group (optional).
  - c. From the available list of users in the left pane, select one or more users and add them to the **Selected** list by clicking the double right-facing arrow.
5. When you have selected at least one user, click **OK**.

The following shows the new group added under the **Groups** node.



### Creating an Execution Policy

An execution policy governs who can execute the business services to which the policy is applied.

To create an execution policy:

1. Start Application Explorer.
2. Right-click the configuration to which you want to connect, for example, SampleConfig. See [Chapter 2, "Configuring OracleWLS Application Adapter for PeopleSoft"](#) for information on creating a new configuration.
3. Select **Connect**.

Nodes appear for Adapters, Events, and Business Services (also known as Web services).

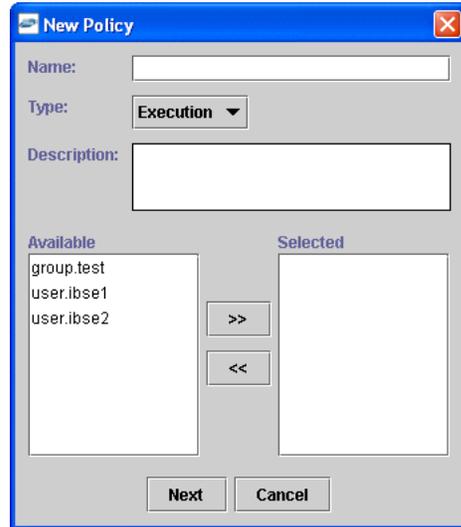


- a. Expand the **Business Services** node.
- b. Expand the **Configuration** node.
- c. Expand the **Security** node.
- d. Expand the **Policies** node.



4. Right-click **Policies** and select **New Policy**.

The New policy dialog box is displayed.



Provide the following information:

- a. In the **Name** field, enter a name for the policy.
- b. From the **Type** list, select **Execution**.
- c. In the **Description** field, enter a description for the policy (optional).
- d. From the available list of users in the left pane, select one or more users and add them to the **Selected** list by clicking the double right-facing arrow.

---

**Note:** This user ID is checked against the value in the user ID element of the SOAP header sent to BSE in a SOAP request.

---

5. When you have selected at least one user, click **OK**.
6. Click **Next**.

The New Policy permissions dialog box is displayed.



7. To grant permission to a user or group to execute a business service, select the user or group and move them into the **Execution Granted** list by selecting the double left-facing arrow.
8. To deny permission to a user or group to execute a business service, select the user or group and move them into the Execution Denied list by selecting the double right-facing arrow.
9. Click **OK**.

The following pane summarizes your configuration.

- **Name** test
- **Type** Execution
- **Description**
- **User and Group Restrictions**
  - group.test Execution Granted

### Using the IP and Domain Restrictions Policy Type

You configure the IP and Domain Restriction policy type slightly differently from other policy types. The IP and Domain Restriction policy type controls connection access to BSE and therefore need not be applied to individual Web services. You need not create a policy; however, you must enable the Security Policy option in Application Explorer.

1. Start Application Explorer.
2. Right-click the configuration to which you want to connect, for example, SampleConfig. See [Chapter 2, "Configuring OracleWLS Application Adapter for PeopleSoft"](#) for information on creating a new configuration.
3. Select **Connect**.

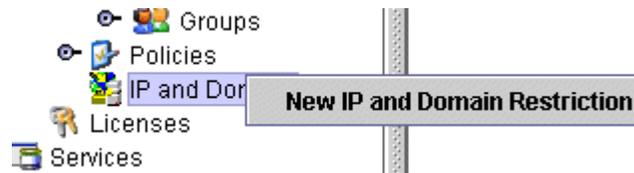
Nodes appear for Adapters, Events, and Business Services (also known as Web services).



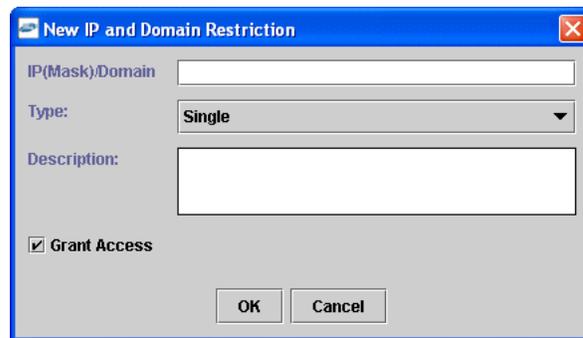
Perform the following steps:

- a. Expand the **Business Services** node.

- b. Expand the **Configuration** node.
  - c. Expand the **Security** node.
4. Right-click **IP and Domain** and select **New IP and Domain Restriction**.



The New IP and Domain Restriction dialog box is displayed.



Perform the following steps:

- a. In the **IP(Mask)/Domain** field, enter the IP or domain name using the following guidelines.
    - If you select **Single** (Computer) from the **Type** list, you must provide the IP address for that computer. If you only know the DNS name for the computer, click **DNS Lookup** to obtain the IP Address based on the DNS name.
    - If you select **Group** (of Computers), you must provide the IP address and subnet mask for the computer group.
    - If you select **Domain**, you must provide the domain name.
  - b. From the **Type** list, select the type of restriction.
  - c. In the **Description** field, enter a description (optional).
  - d. To grant access, select the **Grant Access** check box.
5. Click **OK**.

The new domain is added under the IP and Domain node.

The following pane summarizes your configuration.

- **IP Address (Mask) /Domain** www.yahoo.com
- **Type** Domain
- **Access** Denied
- **Description**

## Migrating Repositories

During design time, the Oracle 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. For management purposes, you can migrate BSE and J2CA repositories that are configured for Oracle to new destinations without affecting your existing configuration. For example, you may want to migrate a repository from a test environment to a production environment.

### Migrating a BSE Repository

To migrate a BSE repository:

1. Copy the BSE control service URL, for example:

```
http://localhost:7777/ibse/IBSEServlet/admin/iwcontrol.ibs
```

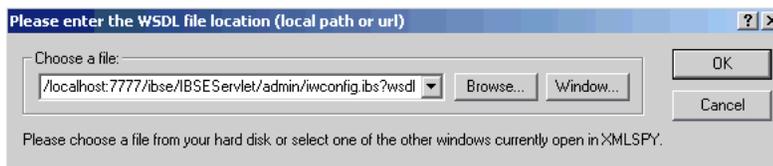
2. Open a third party XML editor, for example, XMLSPY.
3. From the menu bar, click **SOAP**.

A list of options appears.



4. Select **Create new SOAP request**.

The WSDL file location dialog box is displayed.



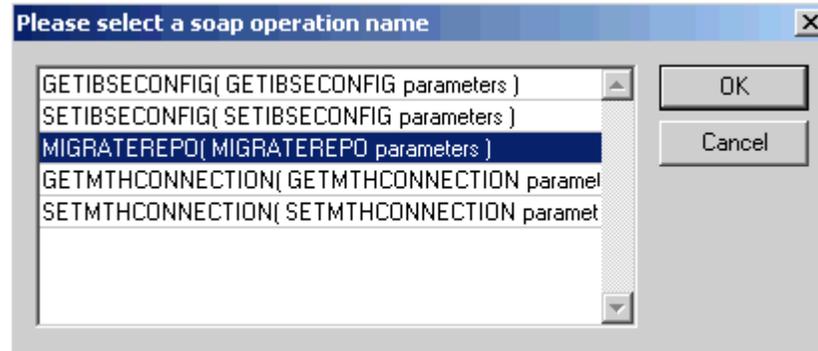
Perform the following steps:

- a. In the **Choose a file** field, paste the BSE control service URL.
- b. Append **?wsdl** to the URL, for example:

```
http://localhost:7777/ibse/IBSEServlet/admin/iwcontrol.ibs?wsdl
```

5. Click **OK**.

The soap operation name dialog box is displayed, listing the available control methods.



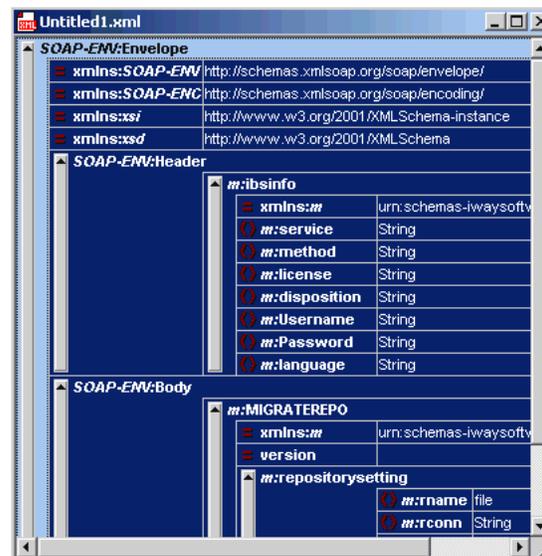
6. Select the **MIGRATEREPO(MIGRATEREPO parameters)** control method and click **OK**.

---

**Note:** The **MIGRATEREPO(MIGRATEREPO parameters)** control method is available from the BSE administration console. This control method migrates all Web services to the new (empty) repository. You can choose to migrate select Web services only.

---

The following window is displayed. It shows the structure of the SOAP envelope.



7. Locate the **Text view** icon in the toolbar.



8. To display the structure of the SOAP envelope as text, click the **Text view** icon.

The `<SOAP-ENV:Header>` tag is not required and can be deleted from the SOAP envelope.

9. Locate the following section:

```
<m:MIGRATEREPO xmlns:m="urn:schemas-iwaysoftware-com:jul2003:ibse:config"
version="">
```

```

<m:repositorysetting>
<m:rname>oracle</m:rname>
<m:rconn>String</m:rconn>
<m:rdriver>String</m:rdriver>
<m:ruser>String</m:ruser>
<m:rpwd>String</m:rpwd>
</m:repositorysetting>
<m:serviceName>String</m:serviceName>
</m:MIGRATEREPO>

```

Perform the following steps:

- a. For the **<m:rconn>** tag, replace the String placeholder with a repository URL where you want to migrate your existing BSE repository.

The Oracle repository URL has the following format:

```
jdbc:oracle:thin:@[host]:[port]:[sid]
```

- b. For the **<m:rdriver>** tag, replace the String placeholder with the location of your Oracle driver.
- c. For the **<m:ruser>** tag, replace the String placeholder with a valid user name to access the Oracle repository.
- d. For the **<m:rpwd>** tag, replace the String placeholder with a valid password to access the Oracle repository.

10. Perform one of the following migration options.

- If you want to migrate a single Web service from the current BSE repository, enter the Web service name in the **<m:serviceName>** tag, for example:

```
<m:serviceName>PeopleSoftService1</m:serviceName>
```

- If you want to migrate multiple Web services from the current BSE repository, duplicate the **<m:serviceName>** tag for each Web service, for example:

```
<m:serviceName>PeopleSoftService1</m:serviceName>
<m:serviceName>PeopleSoftService2</m:serviceName>
```

- If you want to migrate all Web services from the current BSE repository, remove the **<m:serviceName>** tag.

11. From the menu bar, click **SOAP** and select **Send request to server**, as shown in the following image.



Your BSE repository and any Web services you specified are now migrated to the new Oracle repository URL you specified.

### Migrating a J2CA Repository

To migrate a J2CA repository:

1. Navigate to the location of your J2CA configuration directory where the repository schemas and other information is stored, for example:

`wls_home\erp-adapters\config\JCA_CONFIG`

Where *JCA\_CONFIG* is the name of your J2CA configuration.

2. Locate and copy the `repository.xml` file.
3. Place this file in a new J2CA configuration directory to migrate the existing repository.

Your J2CA repository is migrated to the new J2CA configuration directory.



# Generating Component Interface APIs

This section describes how to generate component interface APIs to use with Oracle WebLogic Server Application Adapter for PeopleSoft.

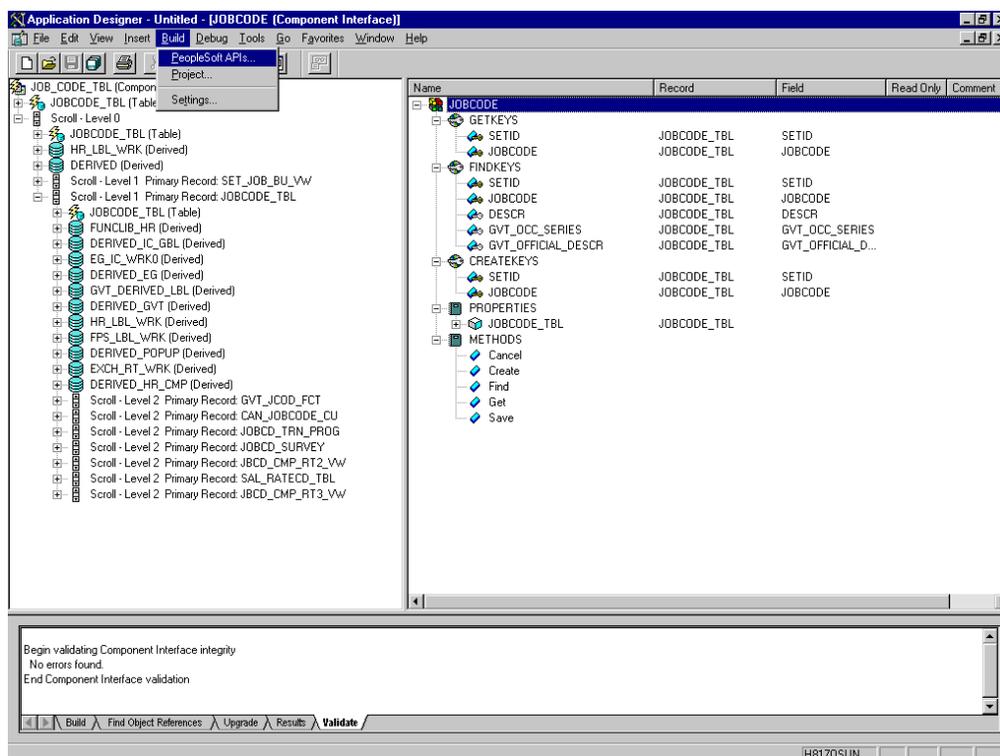
## Building the PeopleSoft API Java Programs

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:

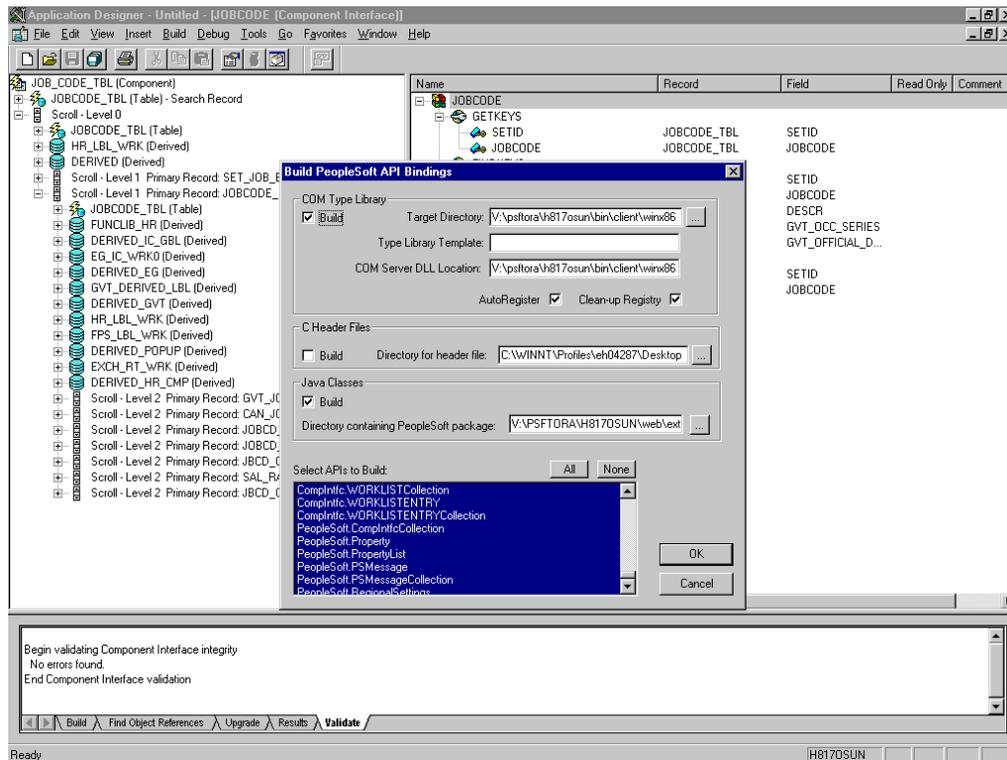
1. Open the PeopleSoft Application Designer.



2. From the PeopleSoft Application Designer, open a component interface.

- Click the right pane and select **PeopleSoft APIs** from the **Build** menu.

The Build PeopleSoft API Bindings dialog box prompts you for the types of bindings to create.



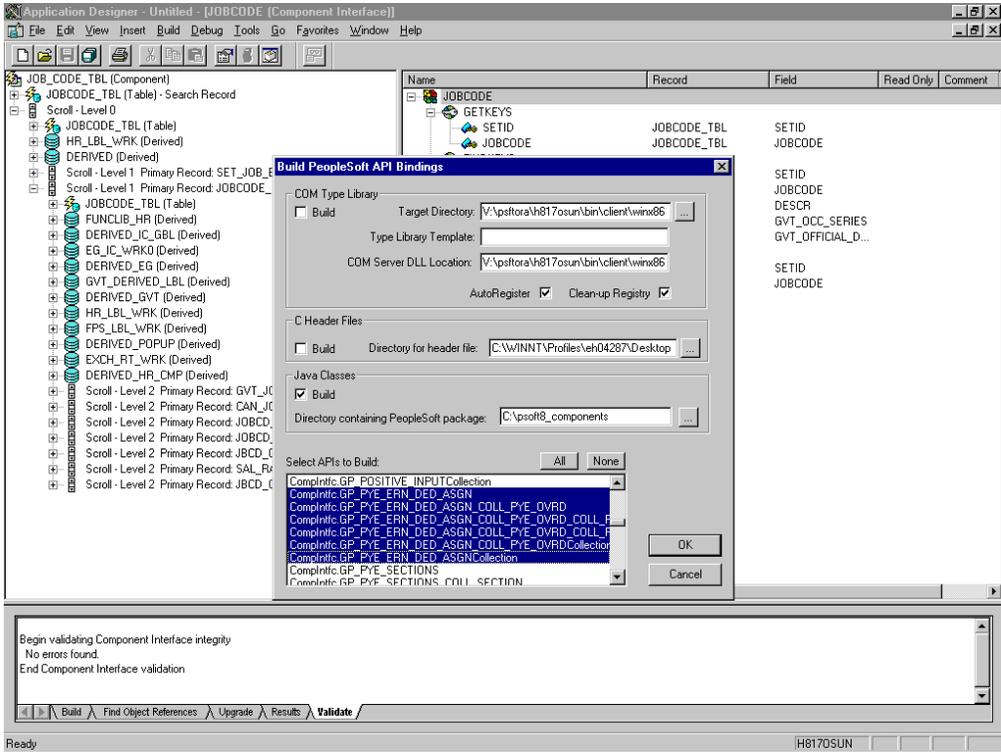
- Because you are creating Java files, ensure you deselect **COM Type Library Build**.
- Ensure that **Java Classes Build** is selected, and then select a directory on your local system where the Java files are to be placed, for example, `c:\psoft8_components`.

- To build all API files, select the default, **All** (potentially a large number), and click **OK**.

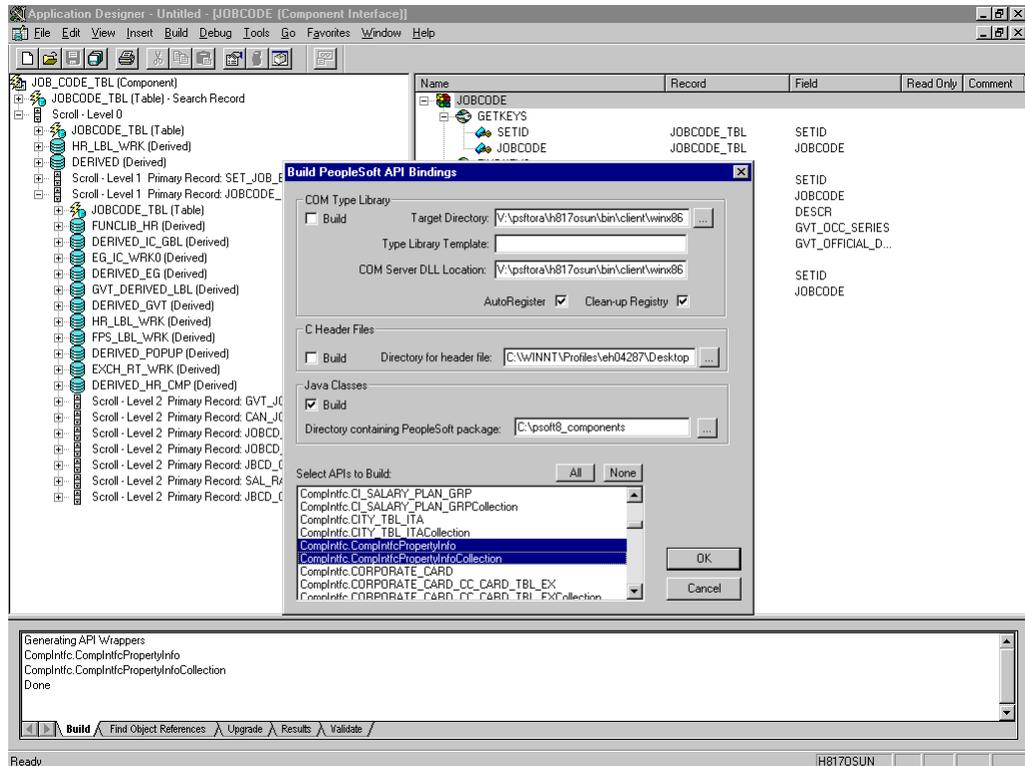
PeopleSoft generates the files. This takes a few minutes. After the process is complete, a message appears in the output window.

You are now ready to compile the Java files. See "[Compiling the PeopleSoft API Java Programs](#)" on page A-4 for more information.

The following pane illustrates the `GP_PYE_ERN_DED_ASGN` component interface from the HR 8.1 application.



- 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 box displays the following APIs, including 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. See "[Compiling the PeopleSoft API Java Programs](#)" on page A-4 for more information.

## 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. See "[Building the PeopleSoft API Java Programs](#)" on page A-1 for more information.

---



---

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.

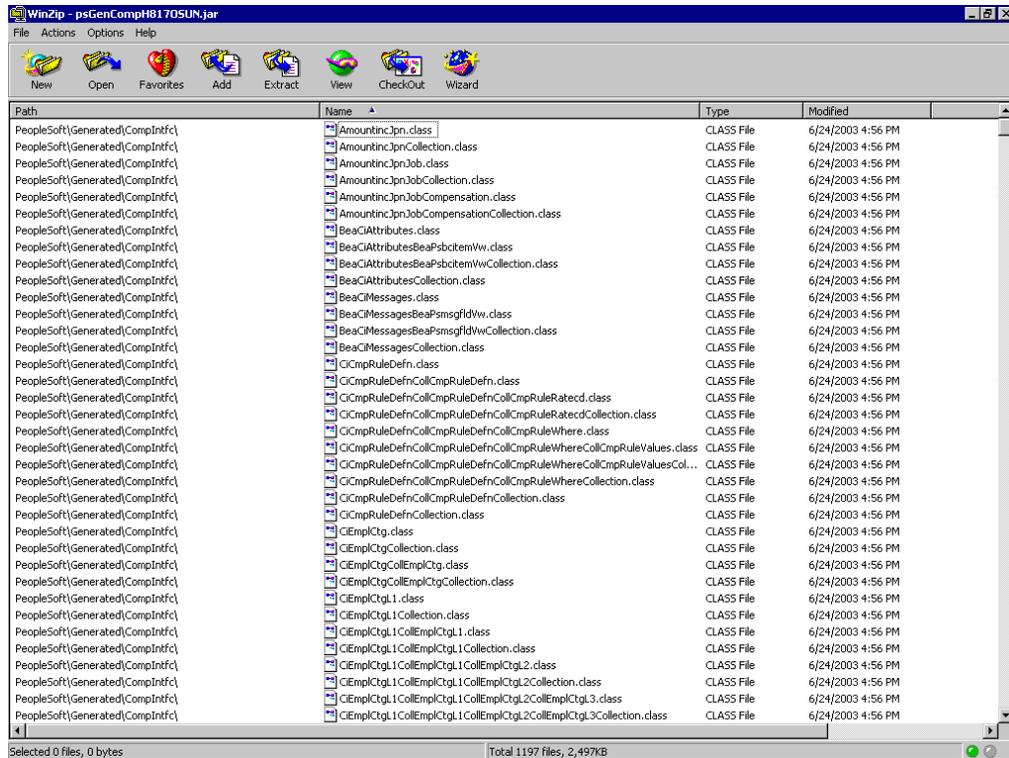
3. Compress the class files into a JAR file.

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.



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 OracleWLS Application Adapter for PeopleSoft to communicate with the PeopleSoft component interface.

`wls_home\erp-adapters\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 section describes how to configure and test a TCP/IP or HTTP target connector and a TCP/IP handler for PeopleSoft.

The following configuration topics assume you are familiar with PeopleSoft Integration Broker (in release 8.4) or Application Messaging (in release 8.1). If not, see [Appendix D, "Using PeopleSoft Integration Broker"](#) for more information. For a complete description *before* you work with OracleWLS 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.

---

---

### 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. See [Appendix D, "Using PeopleSoft Integration Broker"](#) for more information.

1. Configure the gateway for the TCP/IP Target Connector or HTTP Target Connector. See [Configuring the TCP/IP Target Connector](#) on page B-2 for more information.

---

---

**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. See [Configuring the Node for the TCP/IP84 Connector](#) on page B-3 for more information.

---

---

**Note:** Starting with release 8.4, the Integration Broker is delivered with an HTTP Outbound Connector. See [Configuring the HTTP Connector](#) on page B-8 for more information.

---

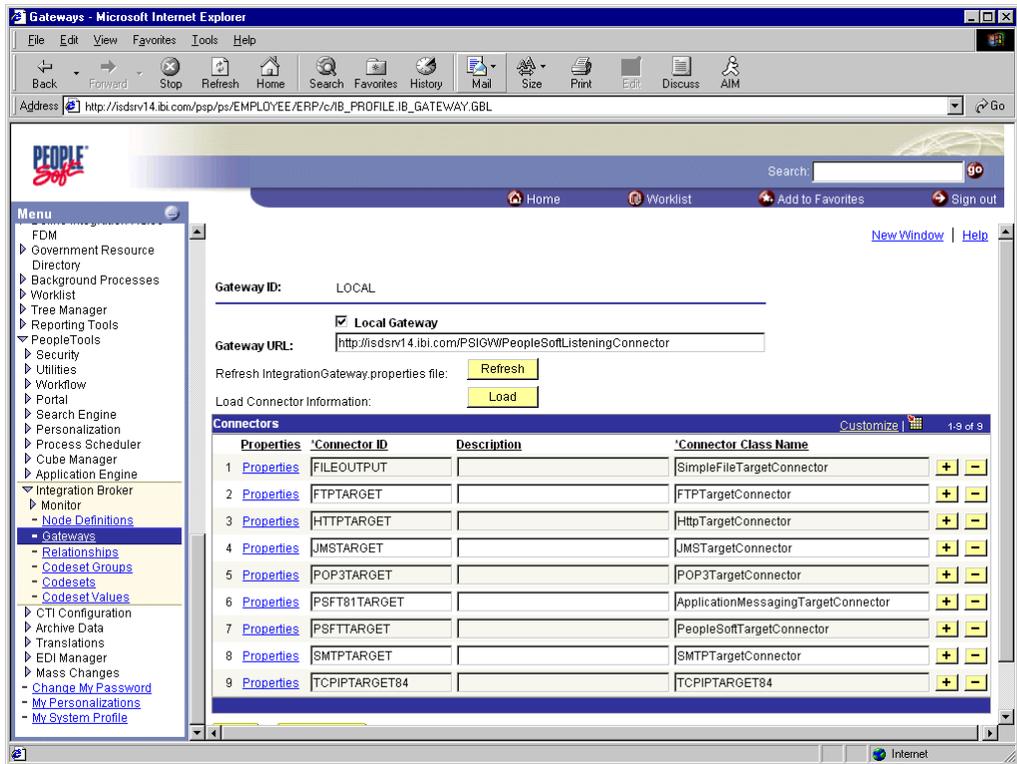
---

## 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.



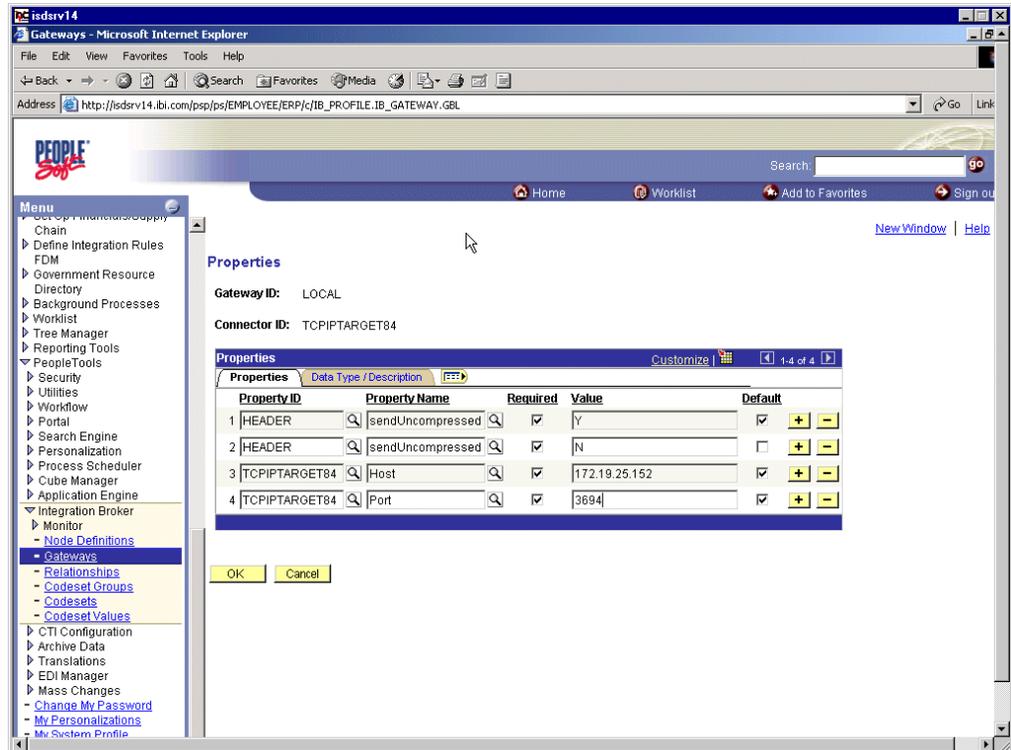
4. If you do not see the TCPIPTARGET84 Connector ID, click **Load** and scroll to locate **TCPIPTARGET84** in the list.

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. Default values appear for the host and the port. For complex business situations, you can override this setting on the individual node.



- 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.

## Configuring the Node for the TCP/IP84 Connector

To configure the node for the TCP/IP84 Connector:

1. In the Menu pane, select **PeopleTools**, **Integration Broker**, and then click **Node Definitions**.
2. Select the node that you want to configure.

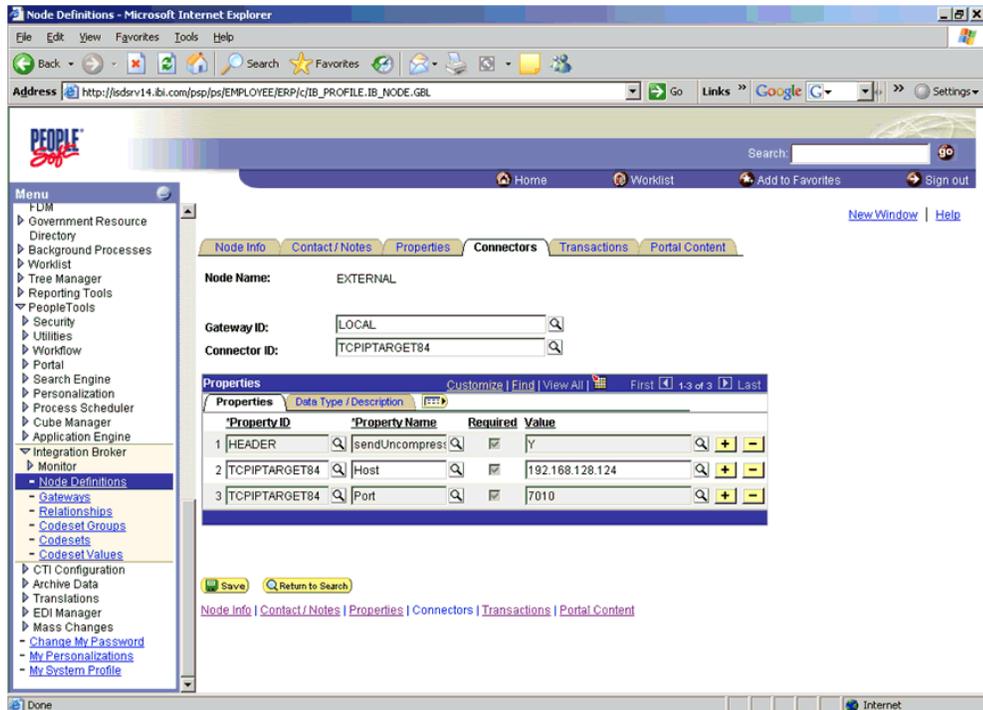
---

**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.

---

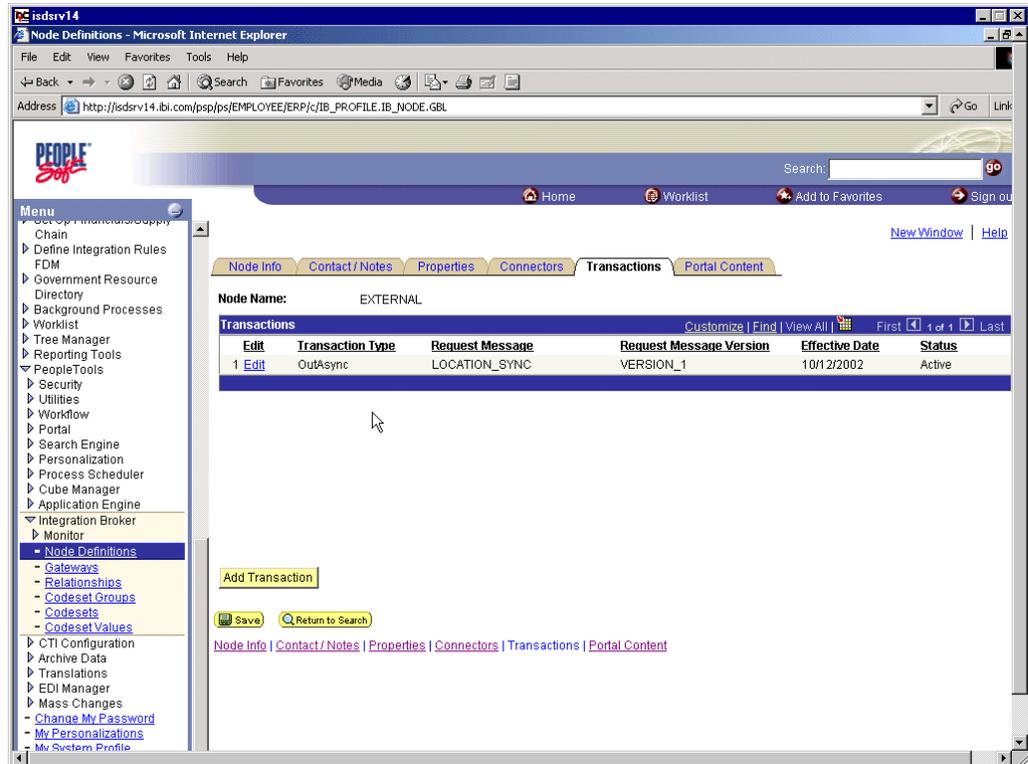
Perform the following steps:

- a. Select **External** from the **Node Type** list.
  - b. Select **Implicit** from the **Routing Type** list.
3. Select the **Connectors** tab.



Perform the following steps:

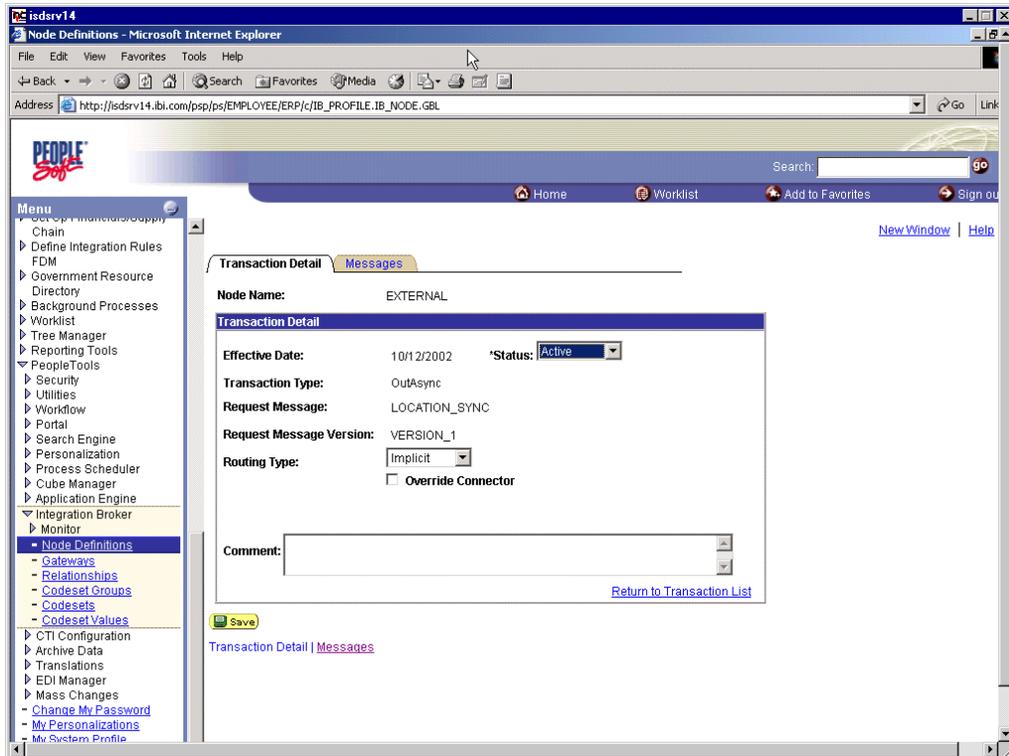
- a. Select **TCPIPTARGET84** as the Connector ID.  
Default values appear for the host and the port.
- b. Enter the values for the host and the port for the system and port that route XML to Oracle WebLogic Server. You can accept or override the default values for individual nodes.
- c. Click **Save**.
4. If you are warned that you are changing the connector, click **OK**.
5. Select the **Transactions** tab.



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



- 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**.

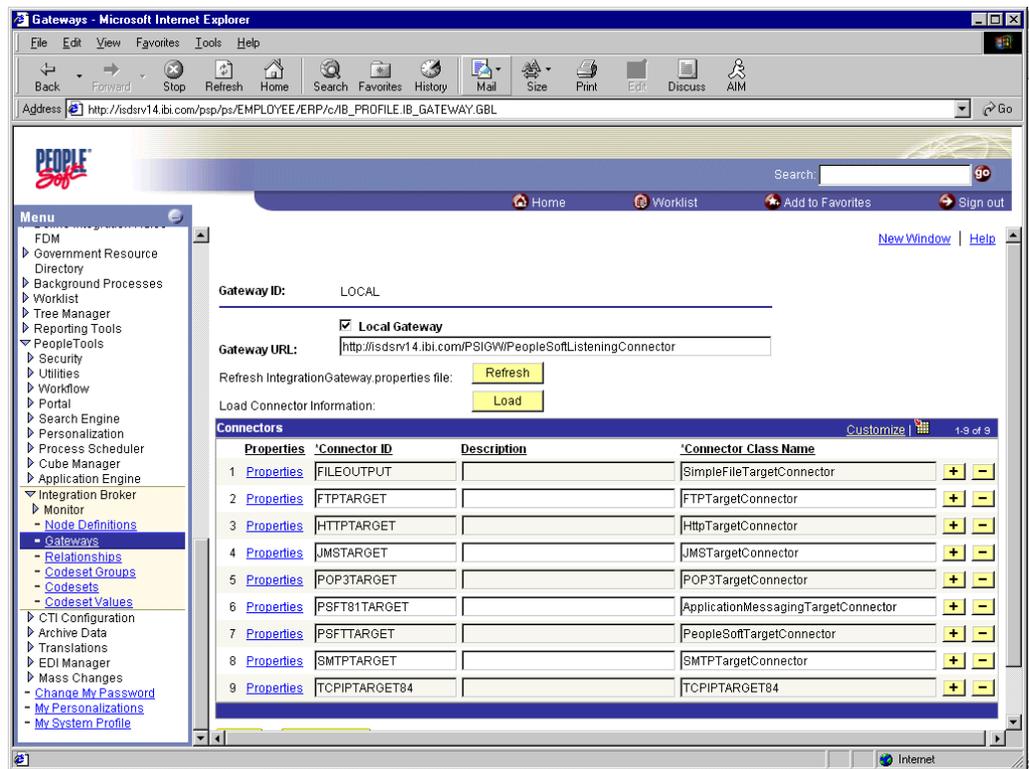
You can now send XML messages to your PeopleSoft XML listener.

### 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.

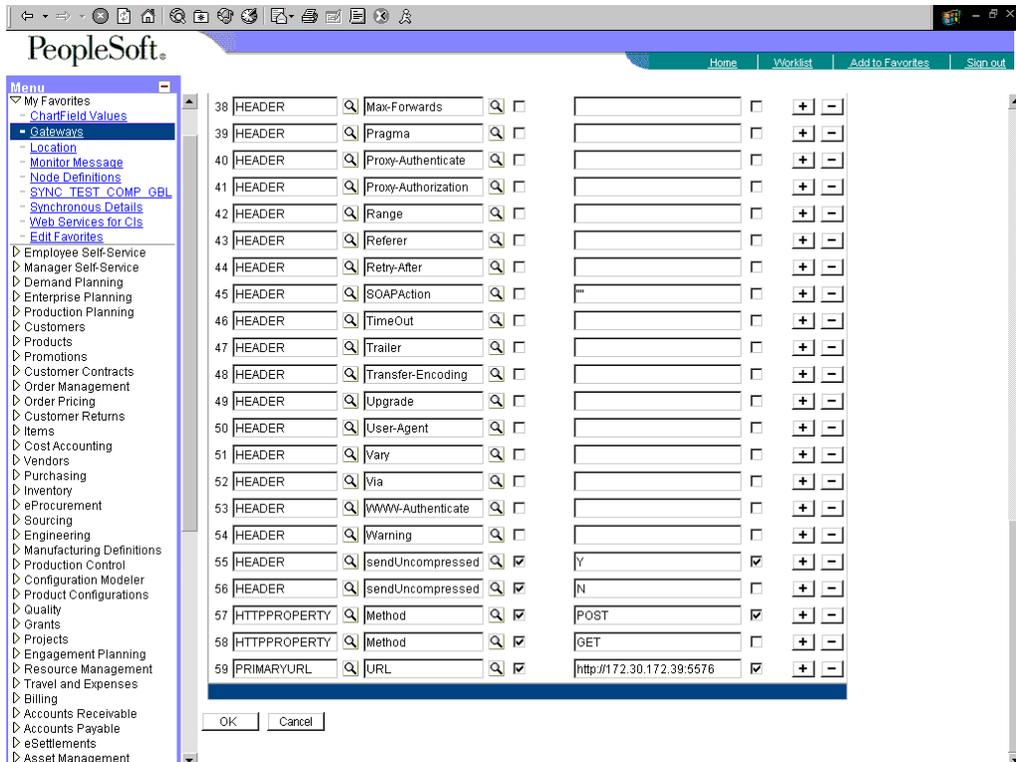


4. If you do not see the **HTTPTARGET** Connector ID, click **Load**.

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:



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.

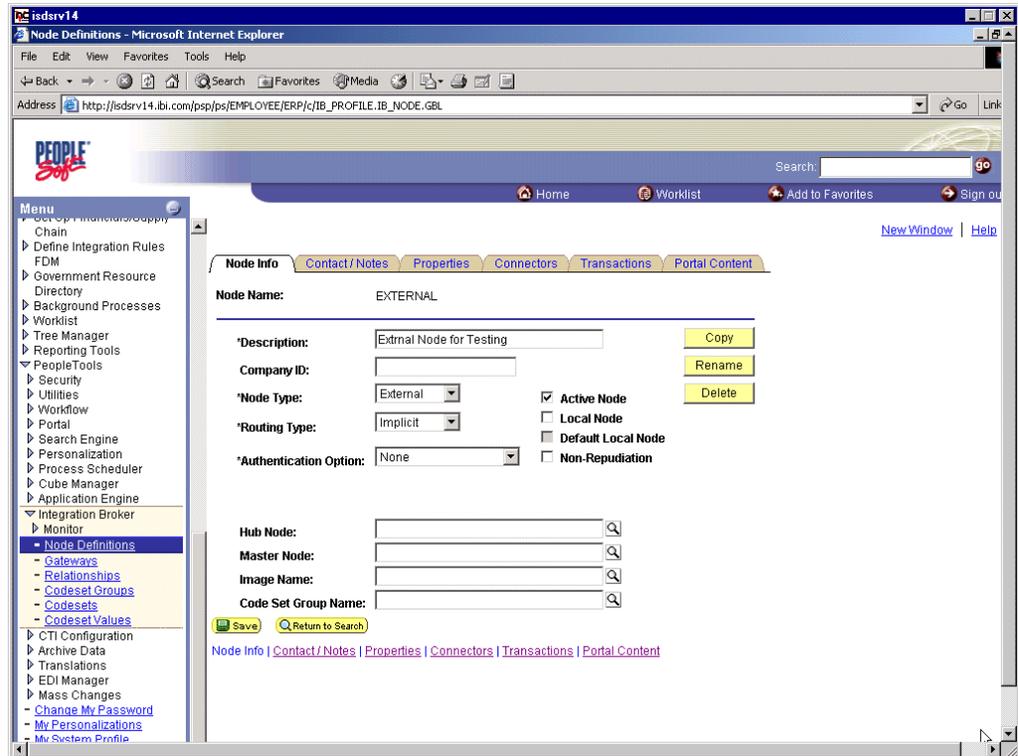
8. Scroll to the bottom of the window and click **Save**.

You have finished configuring the gateway for the HTTP Target Connector.

### 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/IP84 connector for sending messages to Oracle WebLogic Server.

To configure the node to use the HTTP Connector:



1. In the Menu pane, expand **PeopleTools**, **Integration Broker**, and then click **Node Definitions**.
2. Select the node that you want to configure.

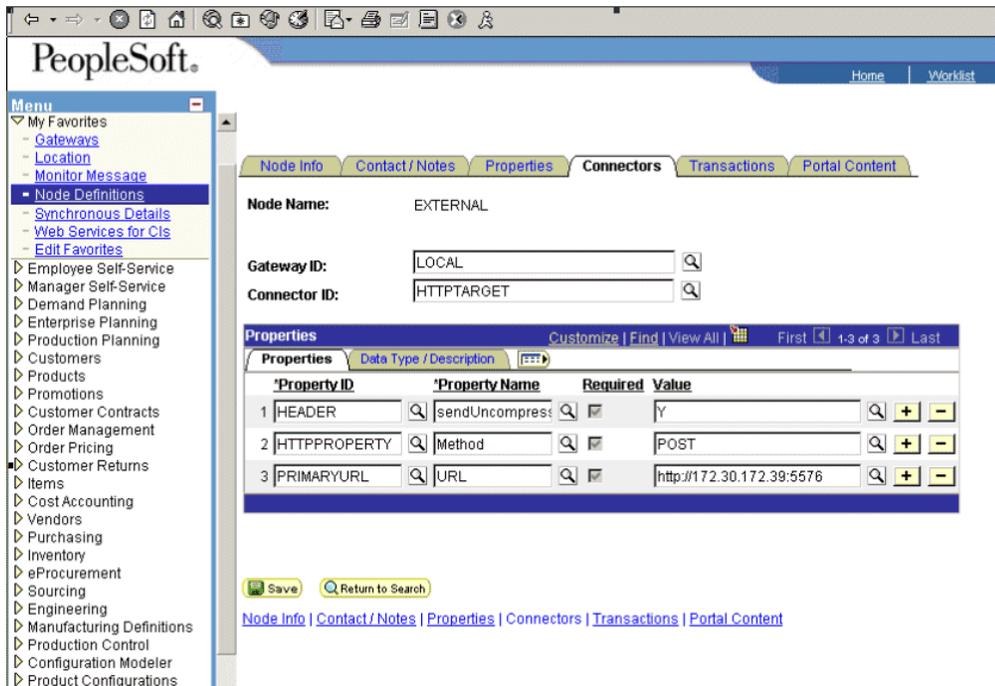
---

**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. From the **Node Type** list, select **External**.
- b. From the **Routing Type** list, select **Implicit**.



3. Select the **Connectors** tab.

Perform the following steps:

- a. Change the Connector ID to **HTTPTARGET**.
- b. Enter a value for each property based on the following:

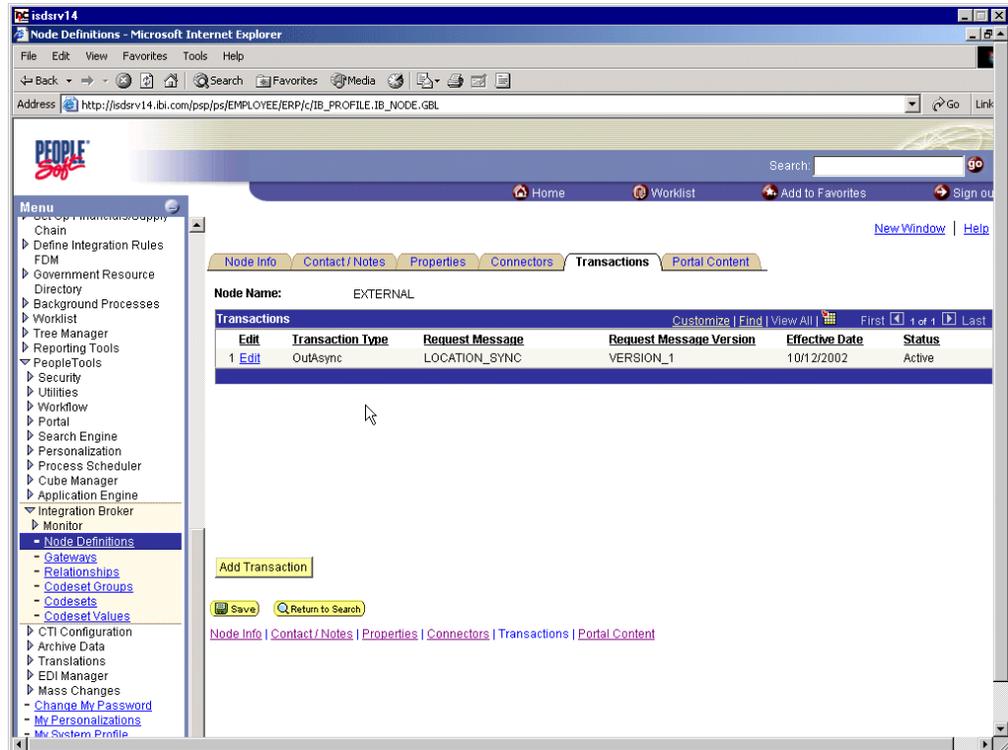
Property ID	Property Name	Value
HEADER	sendUncompressed	Y
HTTTPROPERTY	Method	POST
PRIMARYURL	URL	URL and the port of the HTTP listener

---

**Note:** For complex business situations you can configure multiple nodes and multiple listeners.

---

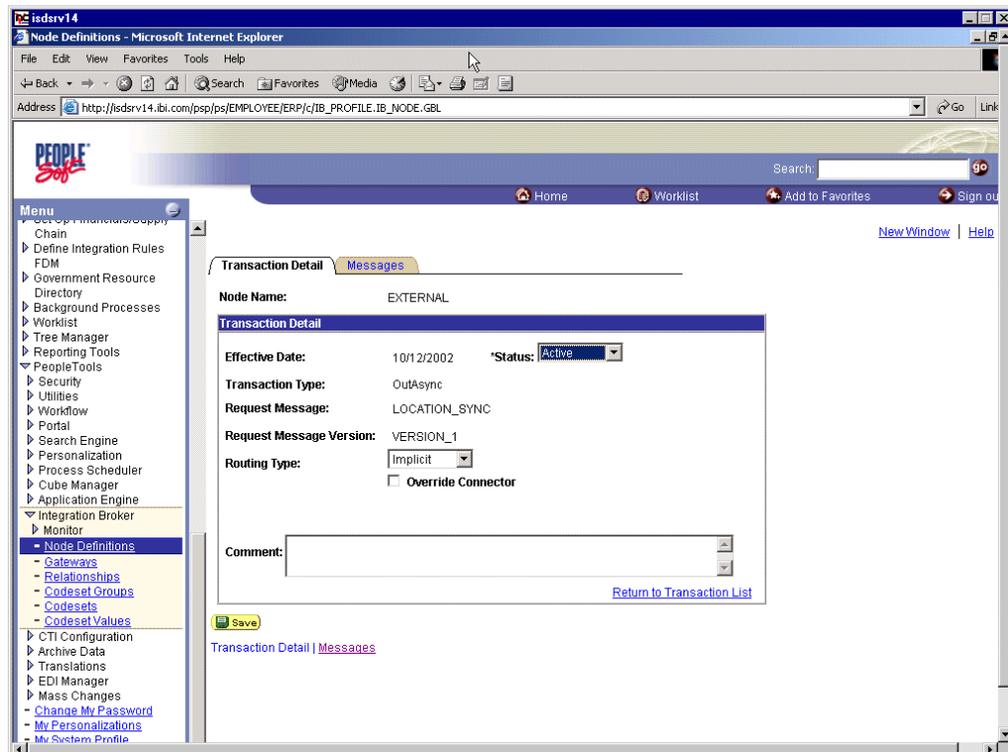
- 4. Click **Save**.
- 5. If you are warned that you are changing the Connector, click **OK**.
- 6. Select the **Transactions** tab.  
The following pane is displayed.



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.



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.

## Configuring the TCP/IP Handler for PeopleSoft 8.1

The following procedure assumes that your Application Messaging environment is properly configured and tested. See [Appendix D, "Using PeopleSoft Integration Broker"](#) for more information.

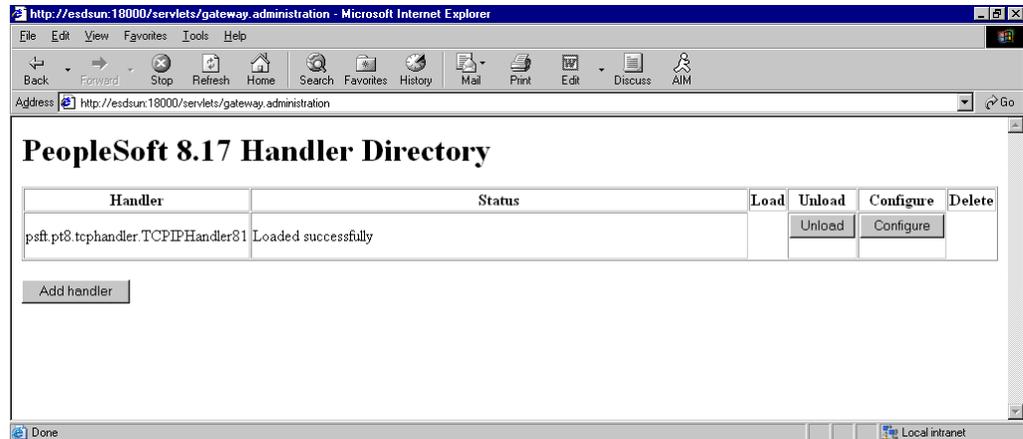
To configure the TCP/IP Handler for PeopleSoft 8.1 to send messages to Oracle WebLogic Server:

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.



3. Click **Add handler**.



Perform the following steps:

- a. Enter the full path of TCPIPHandler81 (case-sensitive):

psft.pt8.tcphandler.TCPIPHandler81

- b. Click **Save**.

4. Click **Load**.

The PeopleSoft Handler Directory window is displayed.

5. Click **Configure**.

The TCPIP81 Handler Directory window is displayed.

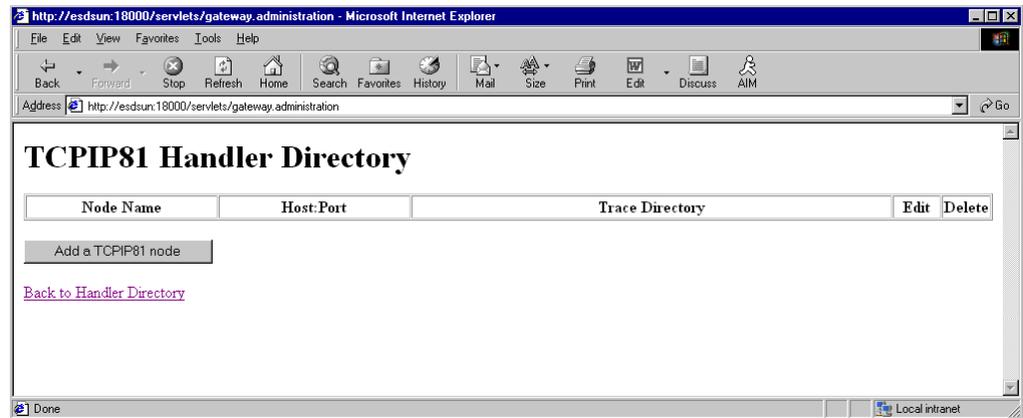
6. Click **Add a TCPIP81 node**.

---

**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.



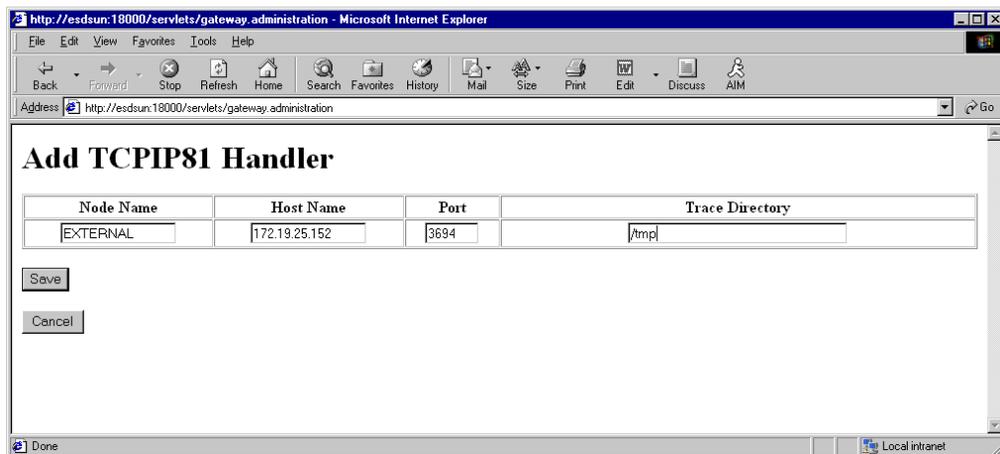
7. Enter the requested values based on the information in the following table.

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.

- Click **Save**.

The TCPIP81 Handler Directory window is displayed.



- For your changes to take effect, click **Back to Handler Directory** to return to the PeopleSoft 8.1 Handler Directory window.

- Click **Unload and re-Load TCPIPHandler81**.

You can now send messages from PeopleSoft to OracleWLS Application Adapter for PeopleSoft.

## 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.

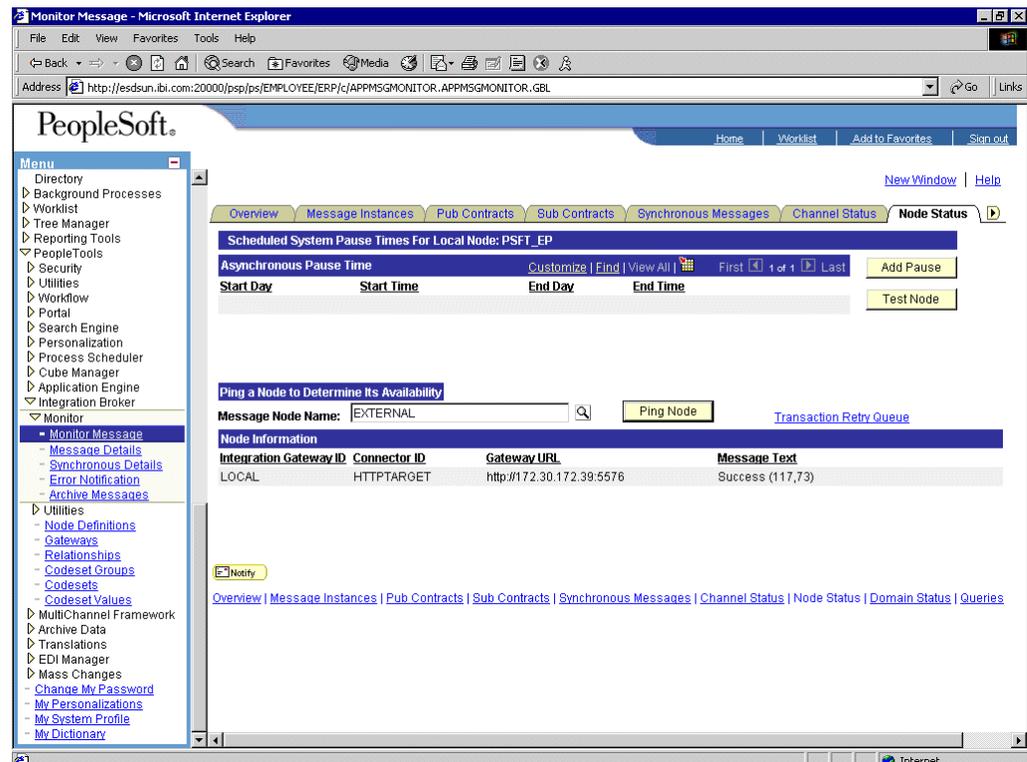
For PeopleSoft 8.4:

- 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.



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 OracleWLS 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, see [Appendix A, "Generating Component Interface APIs"](#) for more information.

---

---

**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.

---

---

### Creating a Component Interface

You create component interfaces using the PeopleSoft Application Designer. For more information about Application Designer, see your PeopleSoft documentation.

### Working With Properties

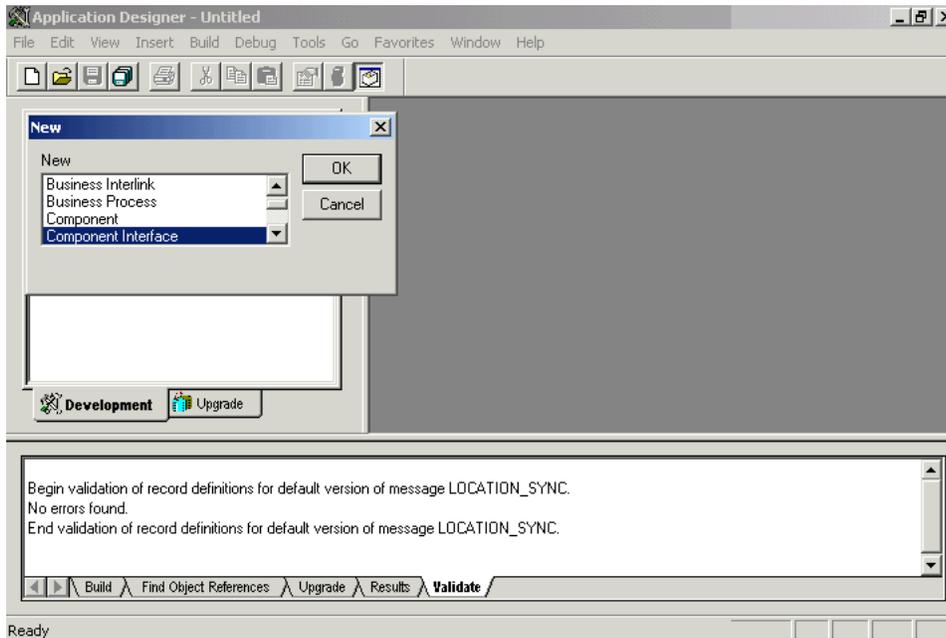
You can add properties from the records in the component view. You can delete a property in the component interface that you do not want to expose. You can rename properties by clicking the property and then clicking again until you can enter a new name. If you rename a property, it can be referenced 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.

### Creating a New Component Interface

To create a component interface:

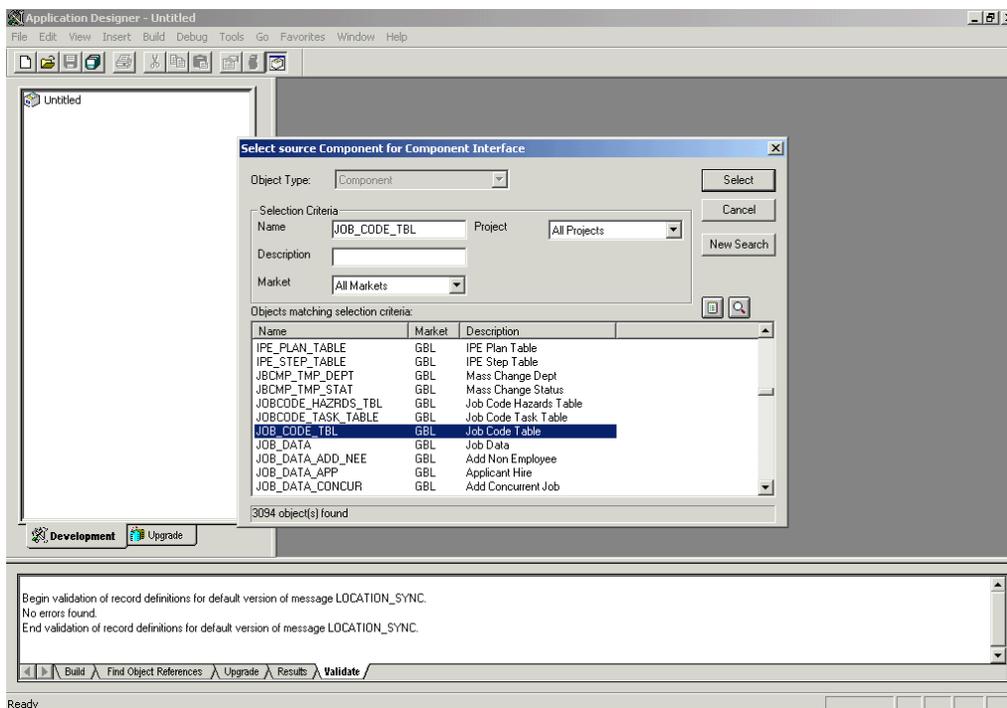
1. Open the PeopleSoft Application Designer.
  2. Select **New** from the **File** menu.
- The New dialog box is displayed.



Perform the following steps:

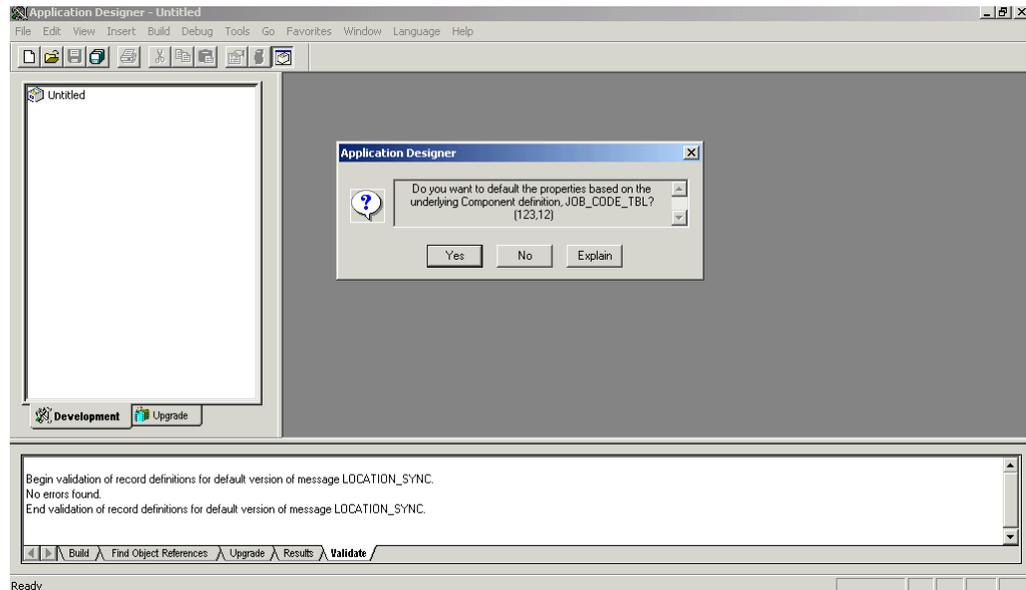
- a. Select **Component Interface**.
- b. Click **OK**.

The Select Source Component for Component Interface dialog box is displayed.



3. Highlight the component to use as a basis for the component interface and click **Select**.

The Application Designer dialog box is displayed.




---

**Note:** If the component interface is large, expose the component properties manually.

---

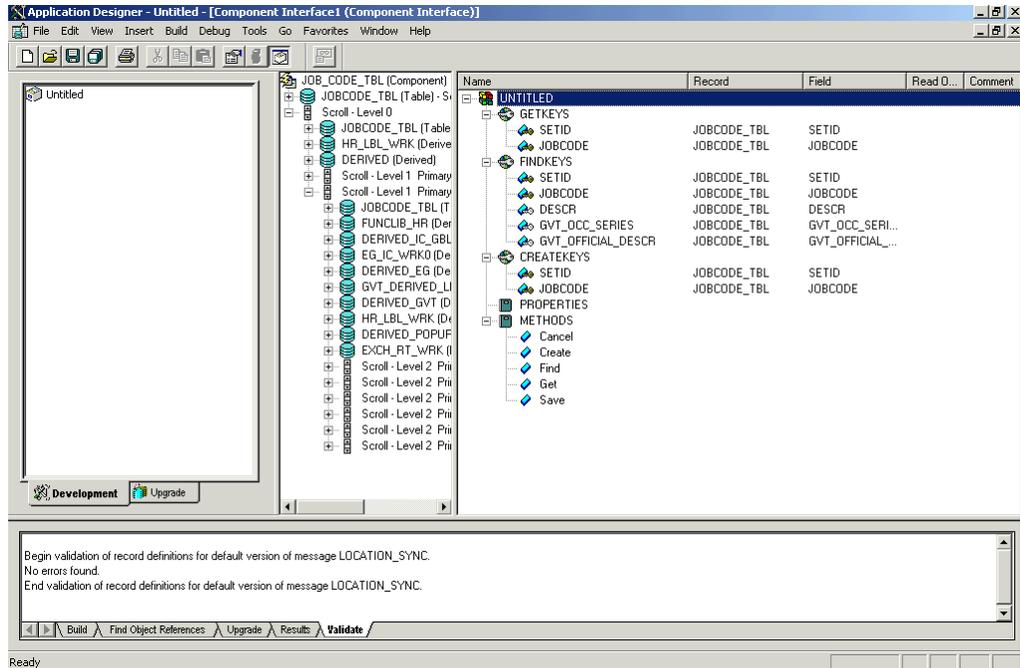
4. To create the component interface without displaying properties and to expose component properties manually, click **No**.

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.

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**.



**Standard Methods**

The standard methods for the component interface are:

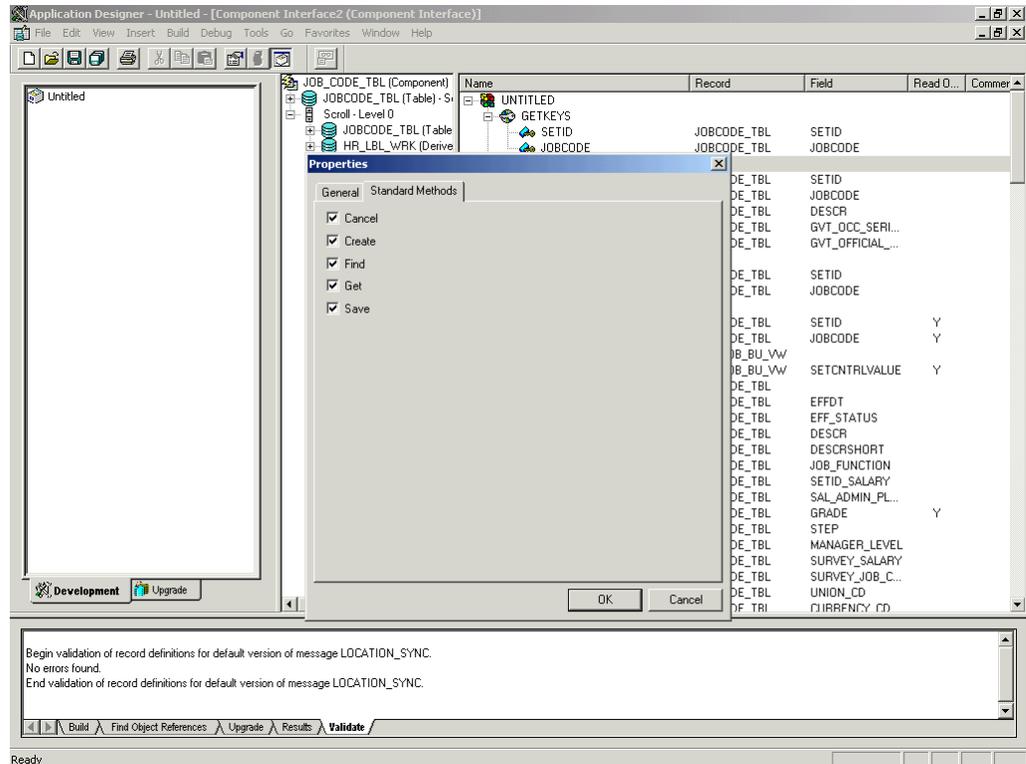
- Create
- Find
- Get
- Save

Only those methods in the underlying component are available. For example, if the underlying component does not contain Add capabilities, Create is not available.

**Viewing or Changing Available Methods**

To view or change available methods:

1. Display the Component Interface Properties dialog box.



2. Click the **Standard Methods** tab.
3. Select the desired methods.

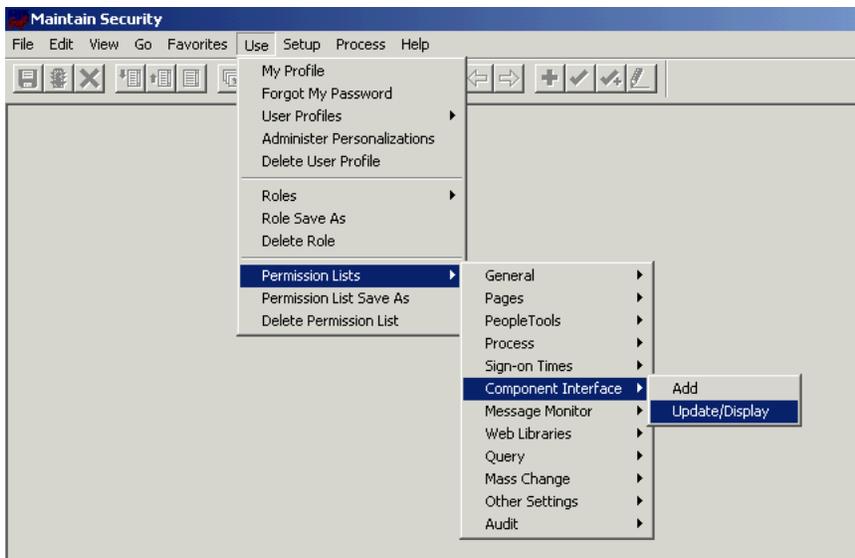
## Securing a Component Interface

You must set up security for the component interface before you can begin testing.

### Configuring Component Interface Security for PeopleSoft Version 8.1x

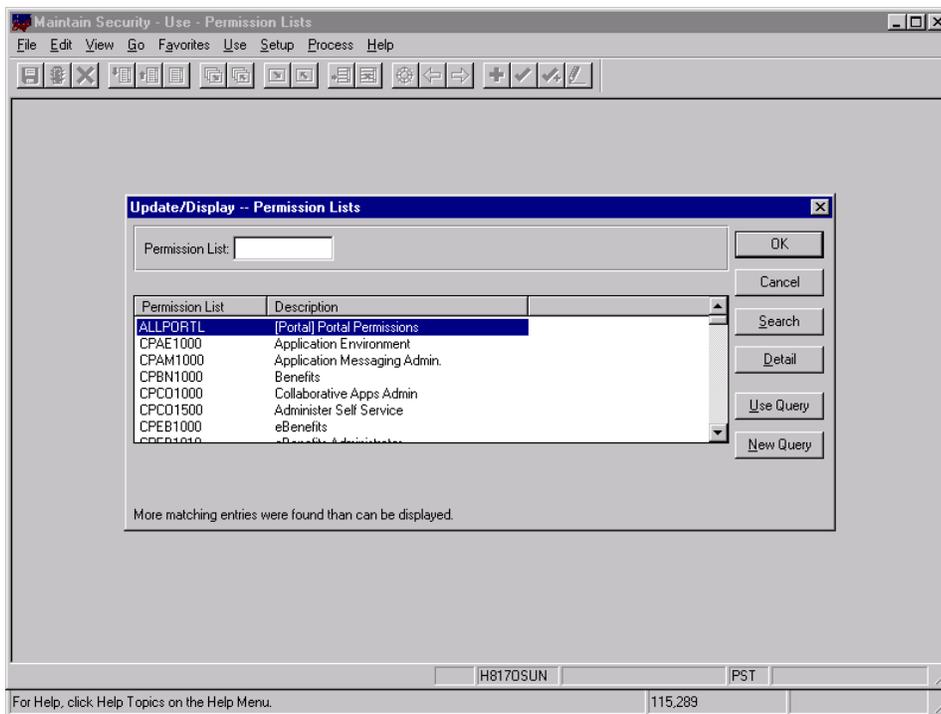
The following procedure describes how to configure component interface security for PeopleSoft Version 8.1 in 2- and 3-tier mode.

To configure component interface security:



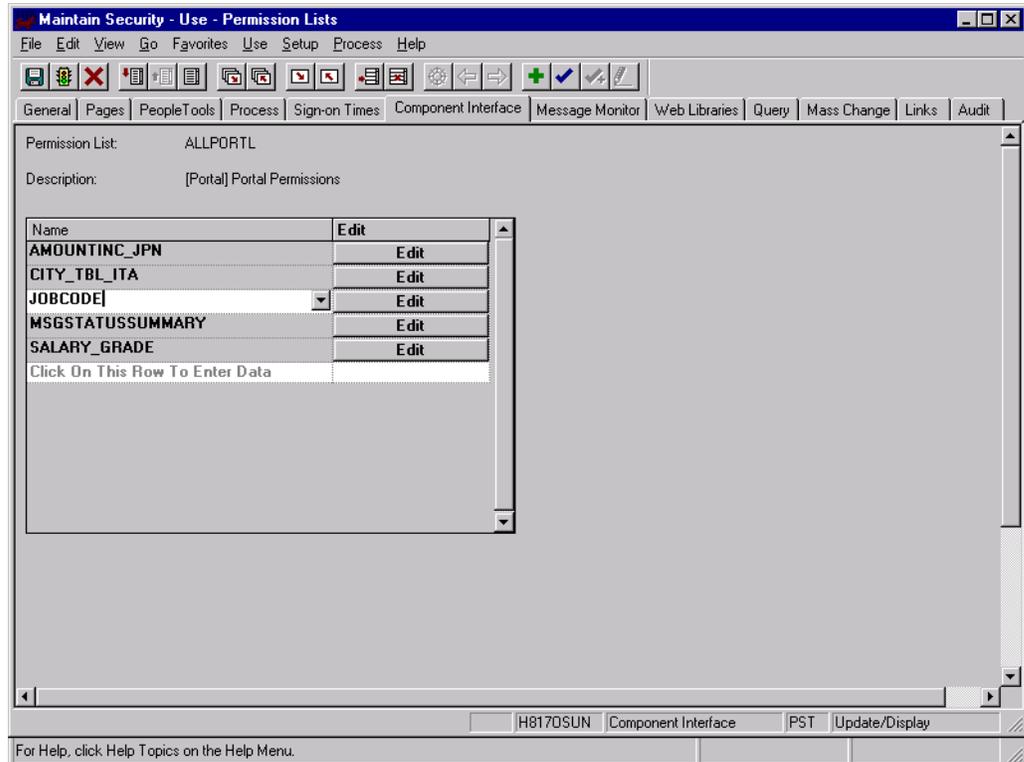
1. From the **Use** menu, select **Permission Lists**, **Component Interface**, and then click **Update/Display**.

The Permission Lists dialog box is displayed.



Before Security can be set, you must identify the permission lists.

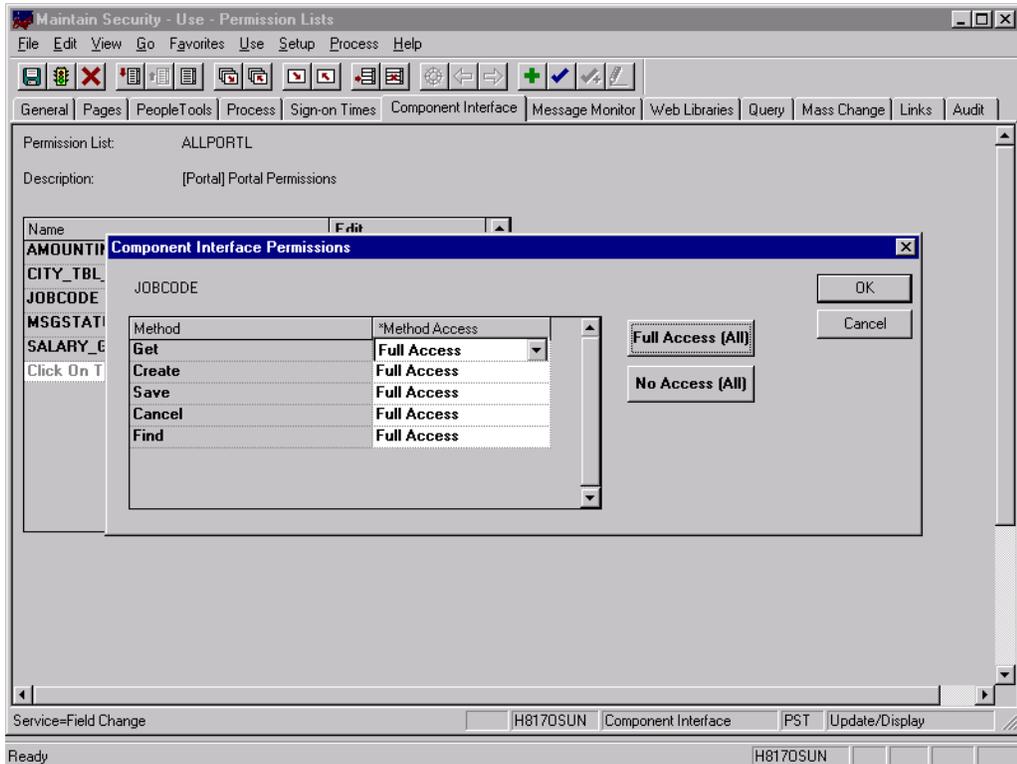
2. Select the relevant permission list and click **OK**.  
For more information on permission lists, see the PeopleBooks documentation.  
The following pane is displayed.



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.

In the following example, the ALLPORTL Permission List has full access to all methods.

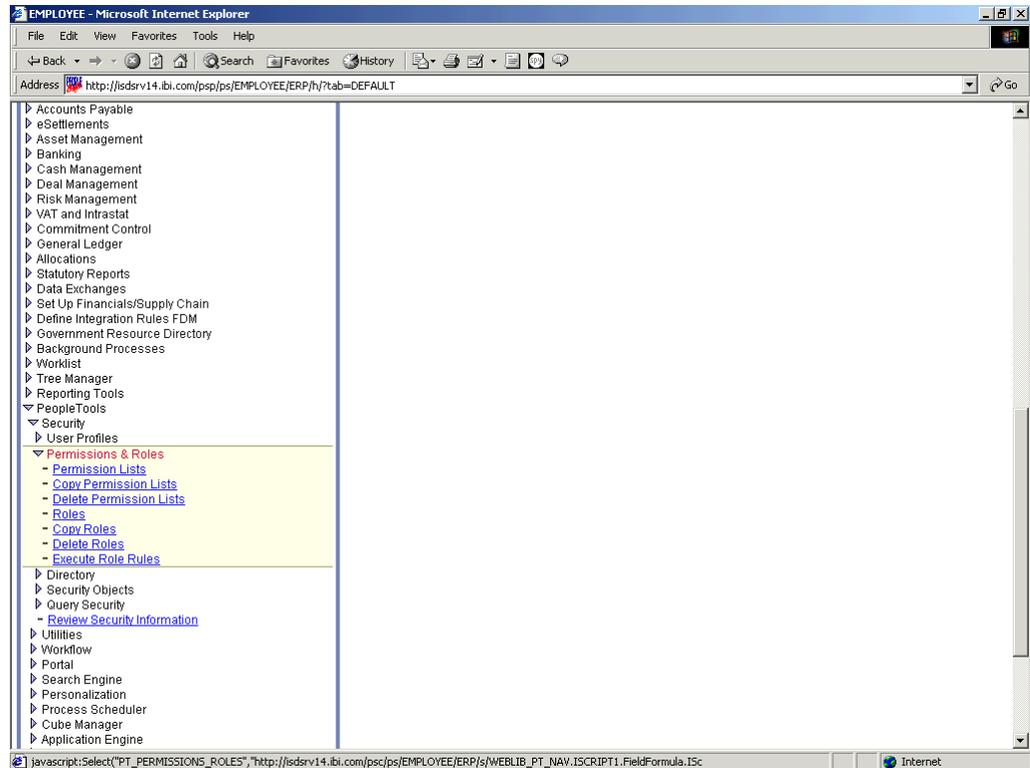


5. Select the desired level of access.
6. Click OK.

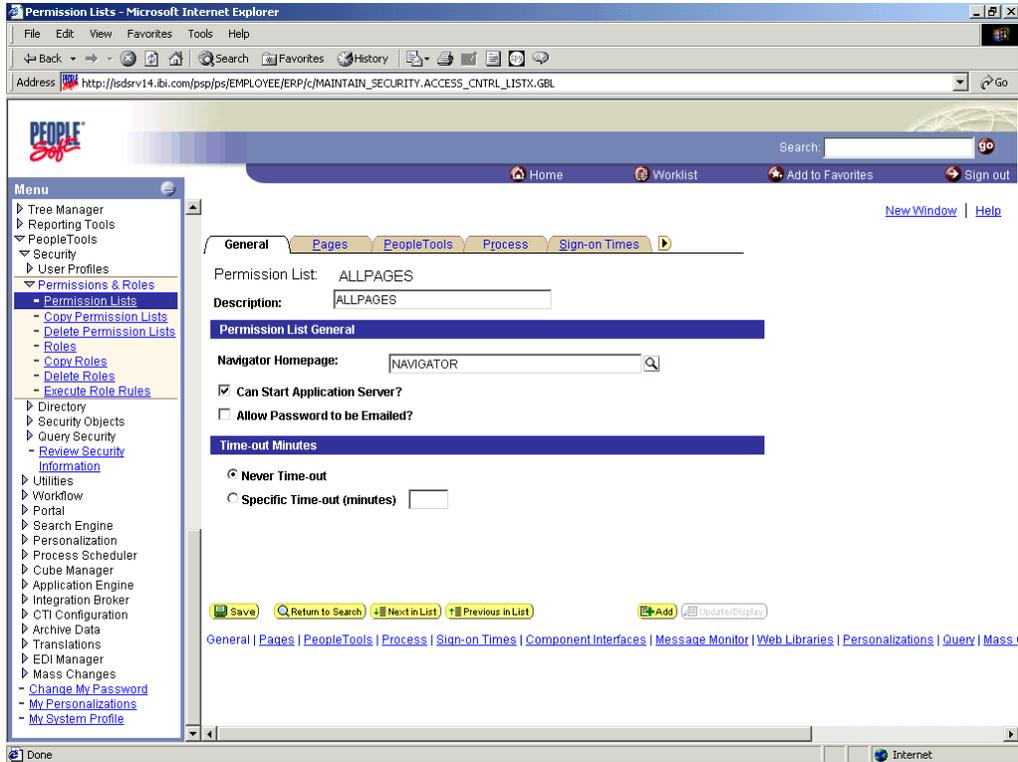
### Configuring Component Interface Security for PeopleSoft Version 8.4 or Higher

The following procedure describes how to configure component interface security for PeopleSoft Version 8.4 or higher.

To configure interface security:

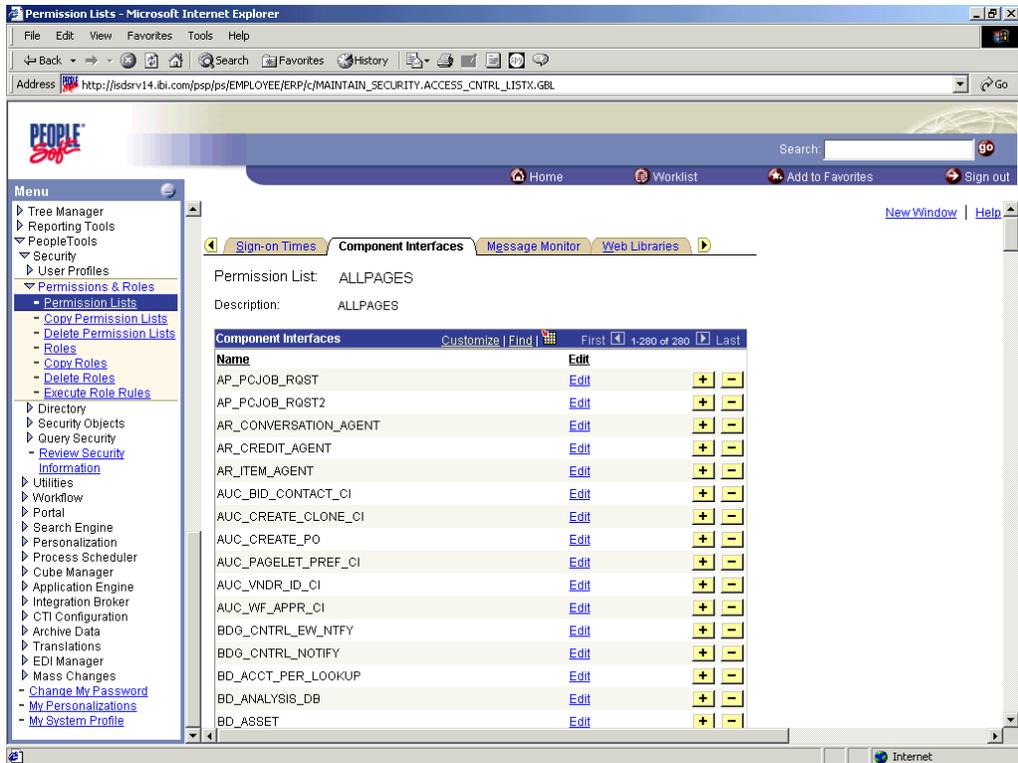


1. Expand **PeopleTools**, **Security**, **User Profiles**, and **Permissions & Roles** and then click **Permission Lists**.
2. Click **Search**.  
The Permission Lists Search pane is displayed.
3. Select the relevant permission list.  
The following pane is displayed.



4. Click the right arrow next to the **Sign-on Times** tab.

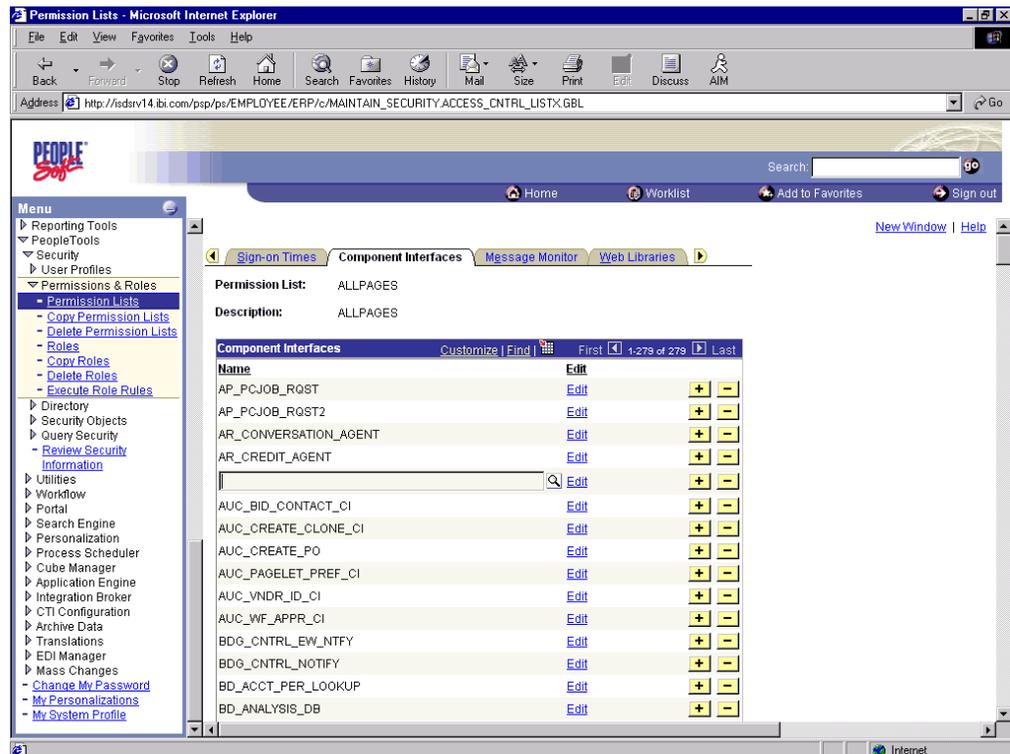
The **Component Interfaces** tab appears.



5. Click the **Component Interfaces** tab.

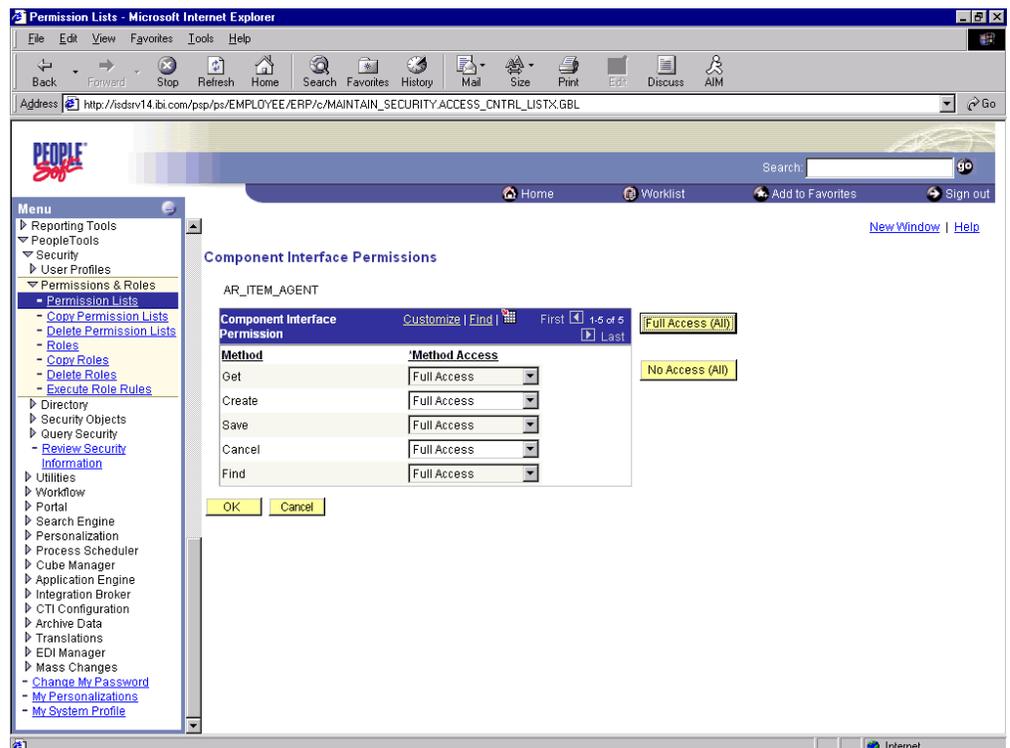
- Click + to add a new row to the Component Interfaces list.

A field appears where you can enter the component interface name.



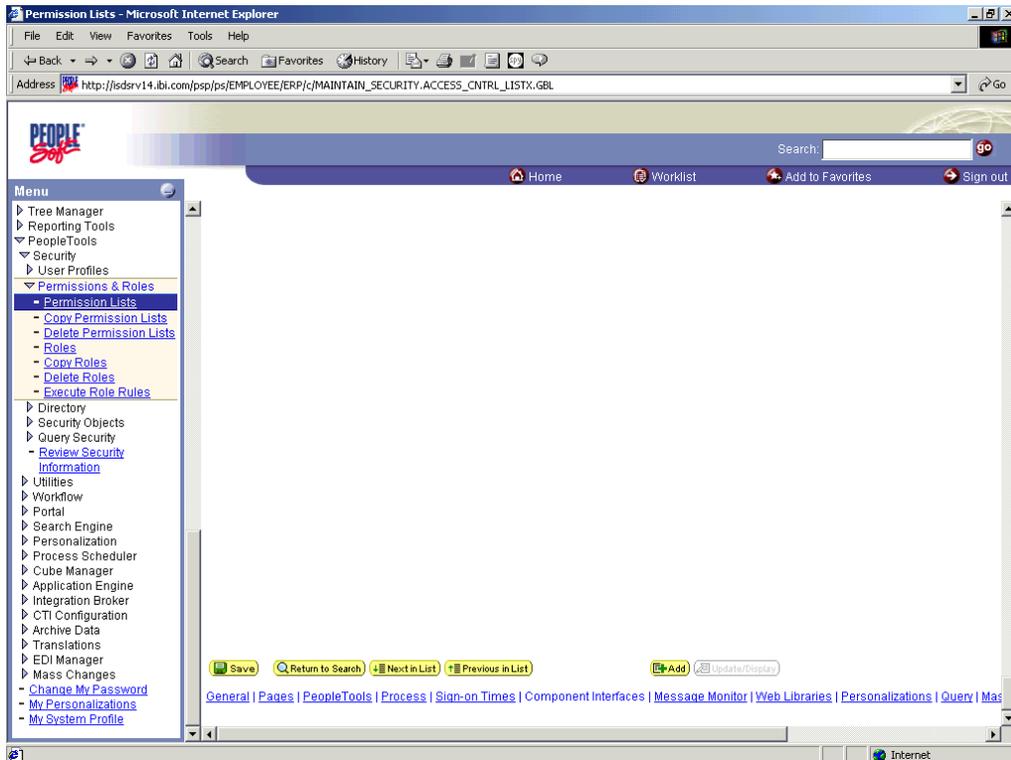
- Enter the component interface name and click **Edit**.

This example uses the component interface AR\_ITEM\_AGENT.



8. From the lists, select the desired access level for each method.
9. Click **OK**.

The following pane is displayed.



10. Scroll down in the right pane and click **Save**.

## Testing a Component Interface

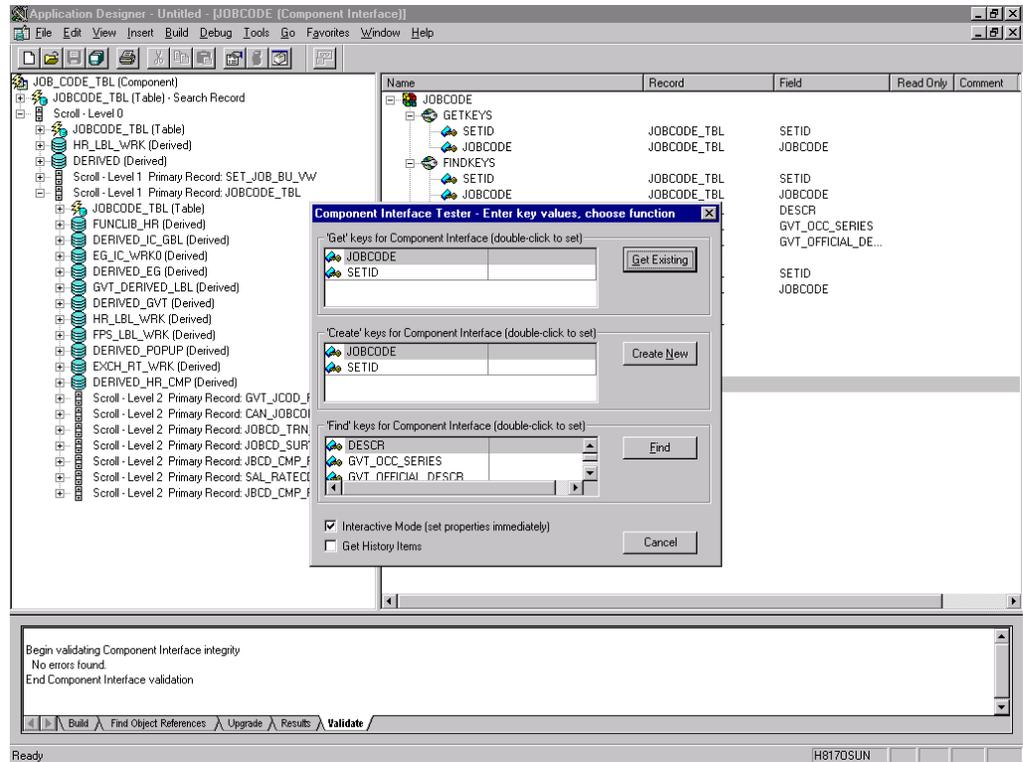
OracleWLS Application Adapter for PeopleSoft uses PeopleSoft metadata and component interfaces; therefore, it can accommodate new or modified component interfaces. The adapter makes no assumptions about component interfaces except that they are logical and valid. Each component interface must be tested 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.

### Testing a Component Interface

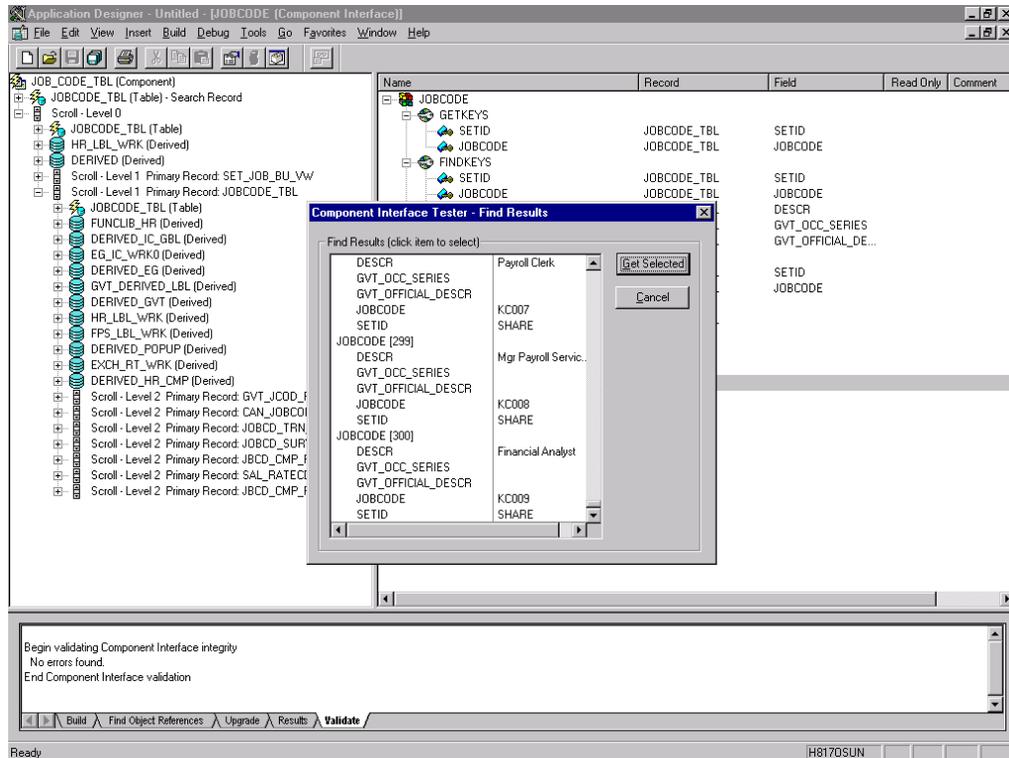
To test a component interface:

1. In Application Designer, select **Test Component Interface** from the **Tools** menu.  
The Component Interface Tester dialog box is displayed.

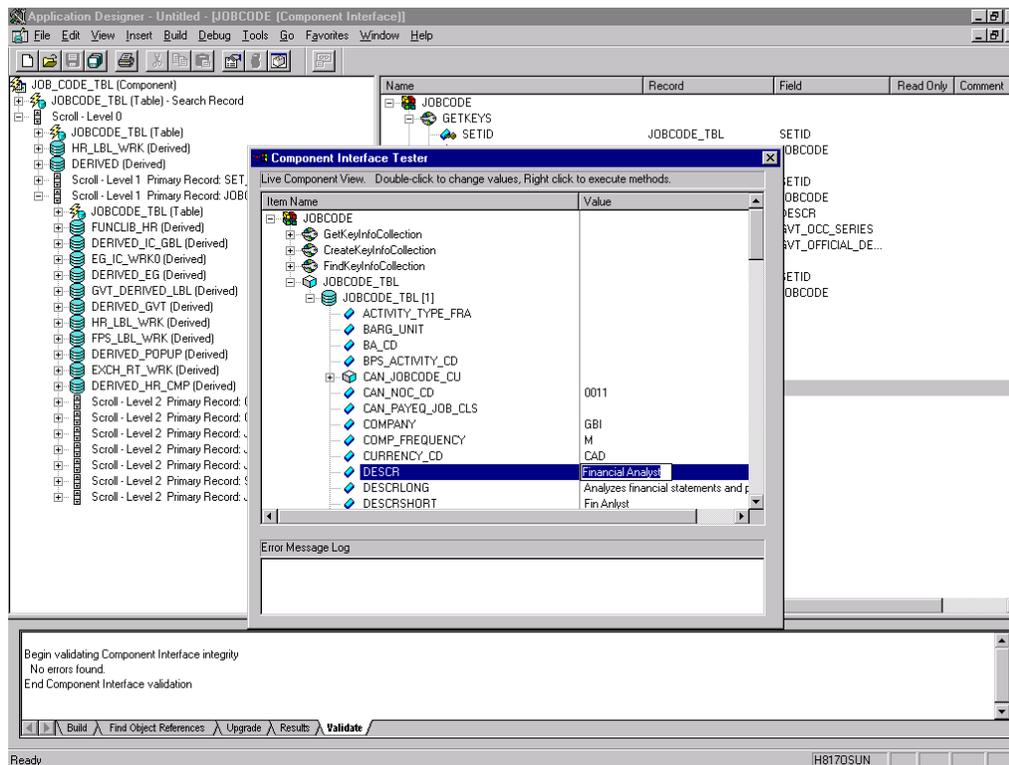


2. If required, click the Component Interface Tester dialog box to bring it to the foreground.
3. To test the component interface, use one the following methods.
  - To test the component interface using the Find method, click **Find**.

The Component Interface Tester - Find Results dialog box displays all of the possible entries for the underlying component. If there are more than 300 entries, a message appears.



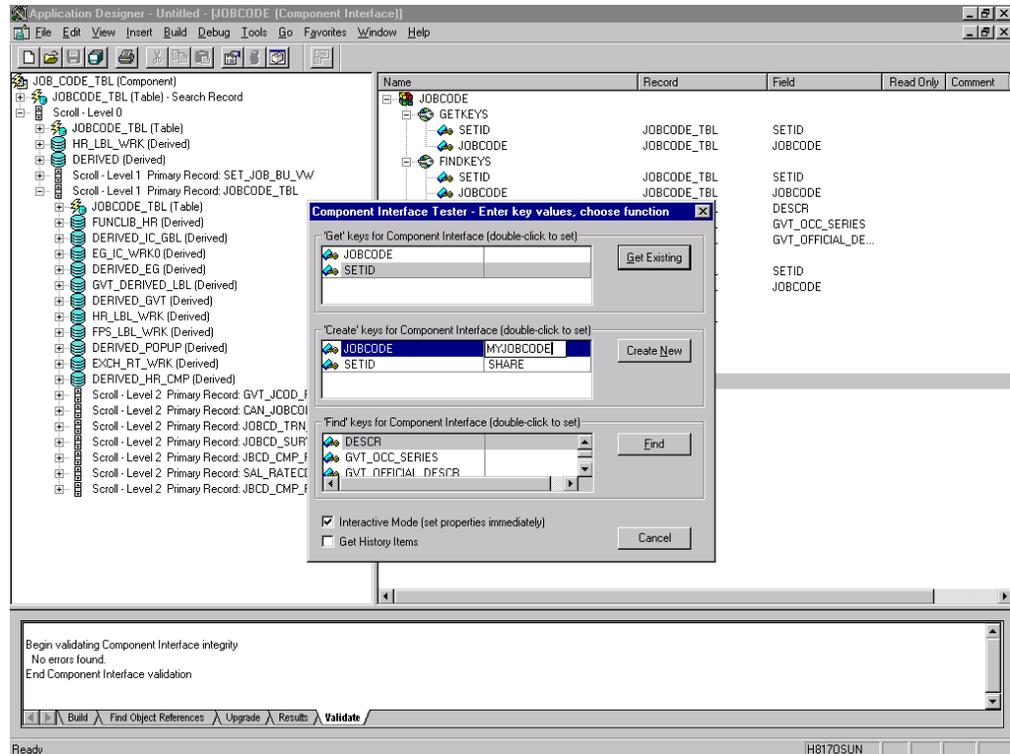
- In the left pane of the Find Results dialog box, select a field.
  - To display the relevant data for that particular field, click **Get Selected**.
- The following dialog box is displayed.



- If the security settings permit, you can change the values in the individual fields.
4. To test the component interface using the Get method, perform the following steps:
    - a. Enter the existing key(s).
    - b. Click **Get Existing**.

This returns the exposed properties for the key that you entered. You can change values if Update access was specified.

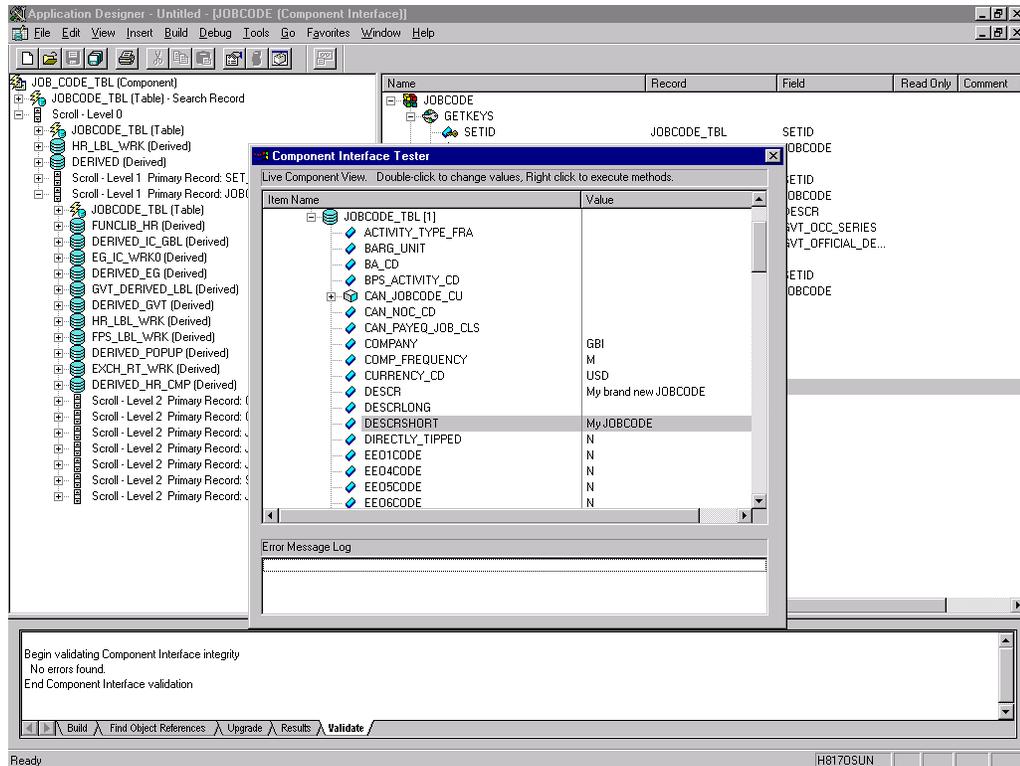
Alternatively, you can test using the Create method.



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.



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. See [Appendix A, "Generating Component Interface APIs"](#) for more information.

---

---

## 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.

---

---

### 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-23.

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.

## Configuring Integration Broker in PeopleSoft 8.4

You can configure PeopleSoft 8.4 to send an asynchronous outbound message to the File Output connector.

To configure application messaging in PeopleSoft 8.4:

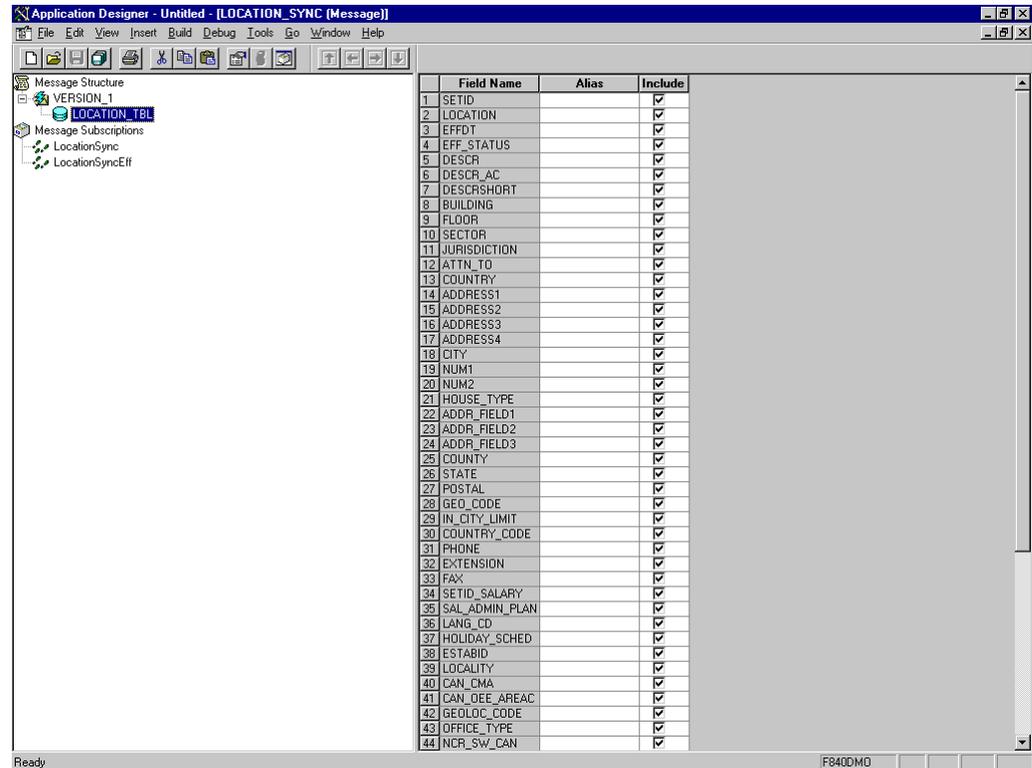
1. Ensure that the message is active and is routed to the proper Message Channel.
2. Configure the Message Channel.
3. Configure the `IntegrationGateway.properties` file to communicate with your PeopleSoft 8.4 application.
4. Configure the Integration Gateway and File Output connector.
5. Create and configure a new Gateway node.

These tasks are described in detail in the following procedures.

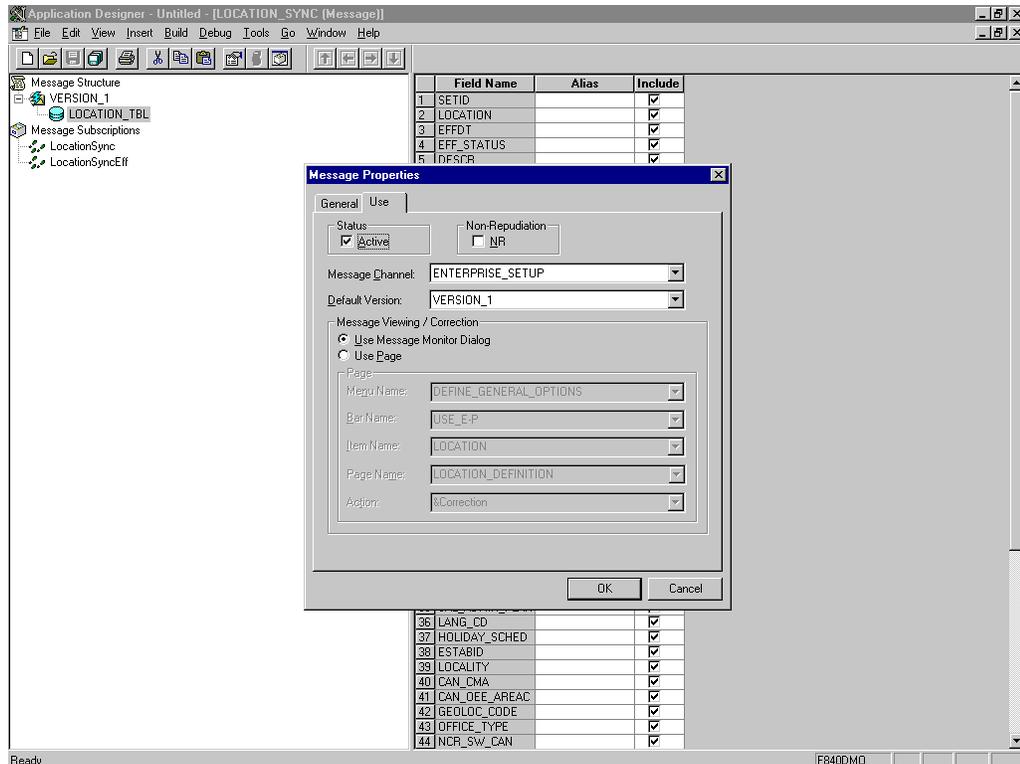
## Ensuring the Message Is Active and Is Routed Correctly

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.



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 box is displayed.



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.

### 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=//isdsv14:9000
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.

### Configuring the Integration Gateway and the File Output Connector

To configure the Integration Gateway and the File Output Connector:

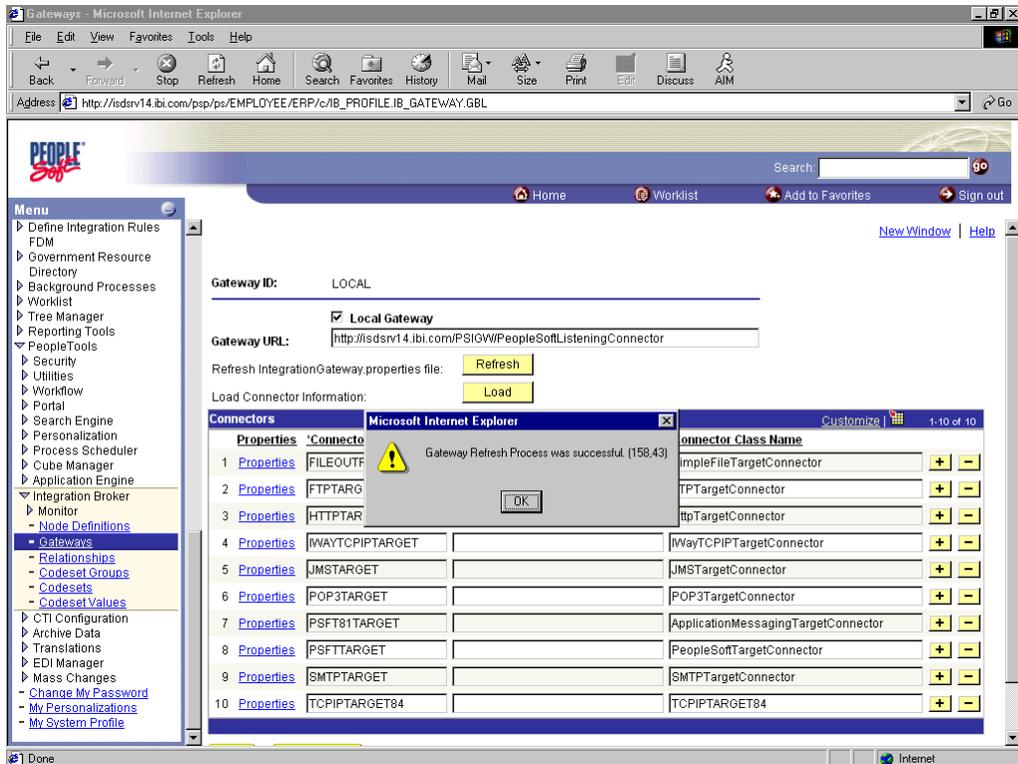
1. In a Web browser, open your PeopleSoft 8.4 application in 4-tier mode.
2. In the Menu pane, expand **PeopleTools, Integration Broker**, and then click **Gateways**.
3. Open the LOCAL Gateway ID and enter the following Gateway URL:

```
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.



5. Click **OK** and scroll down to click **Save**.

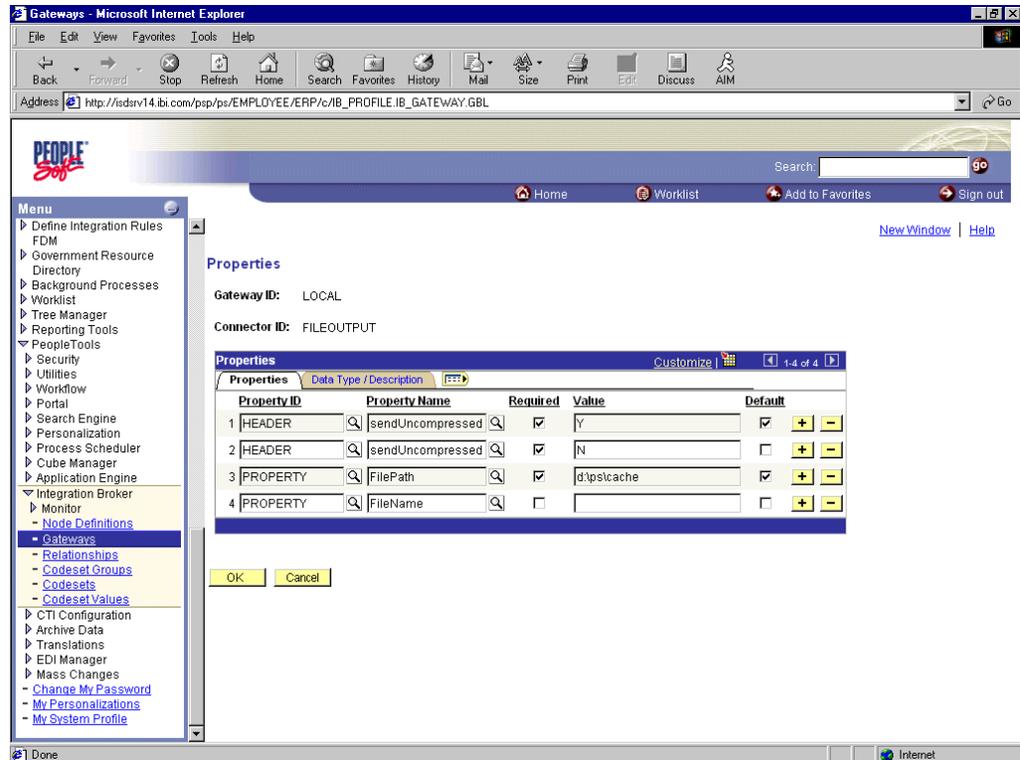
You must click Save before continuing.

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.

In the following figure, the FilePath PROPERTY from the c:\temp default was changed to d:\ps\cache.



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.

### Creating and Configuring a New Gateway Node

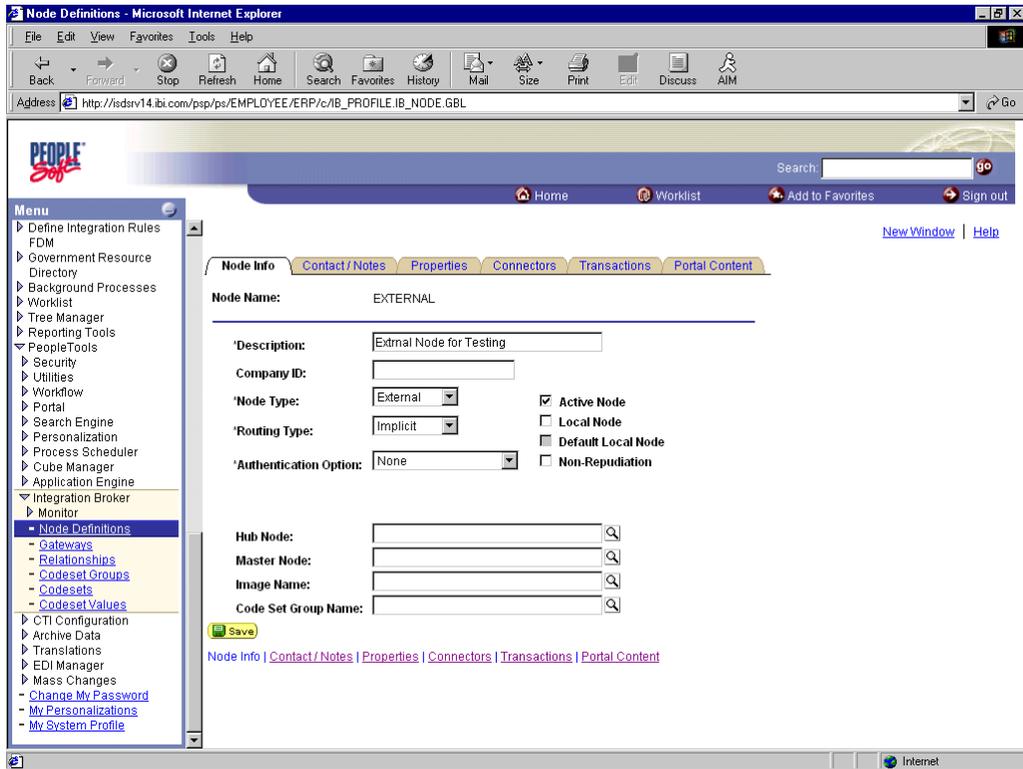
To create and configure a new Gateway Node:

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**.

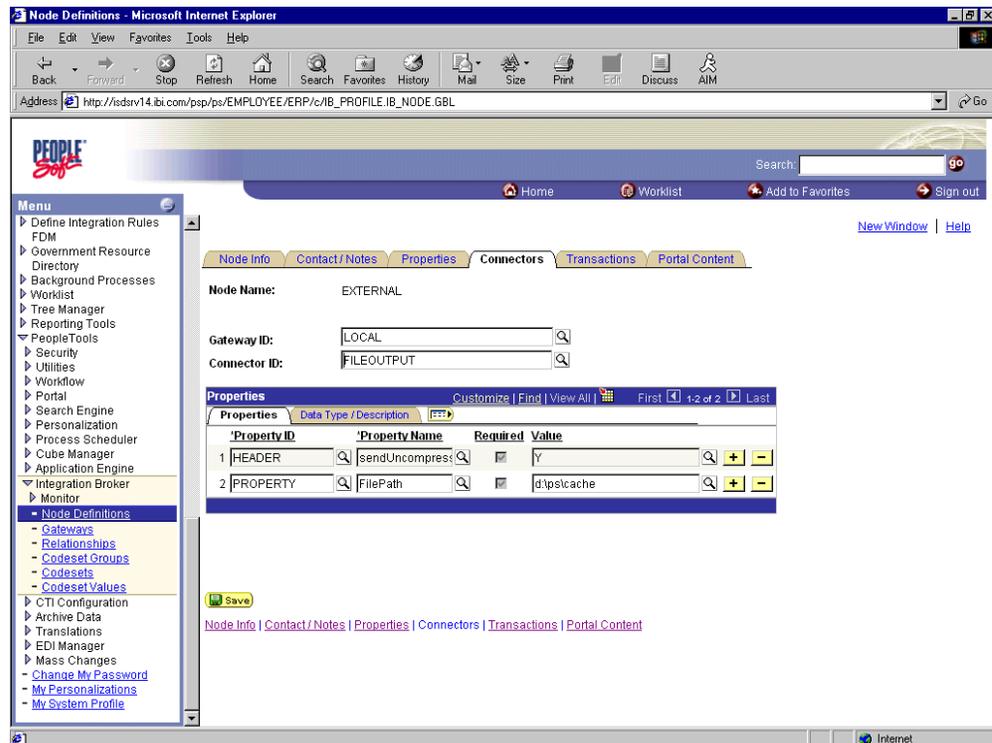
The Node Info tab becomes available.



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.

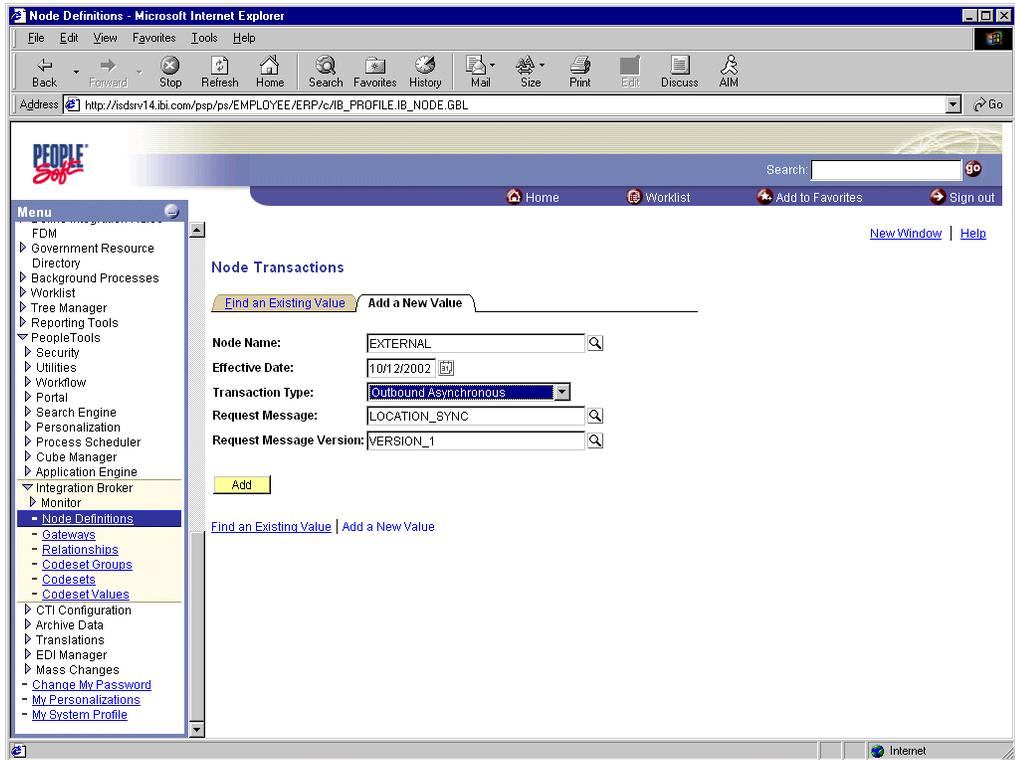
The Connectors tab becomes available.



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**.

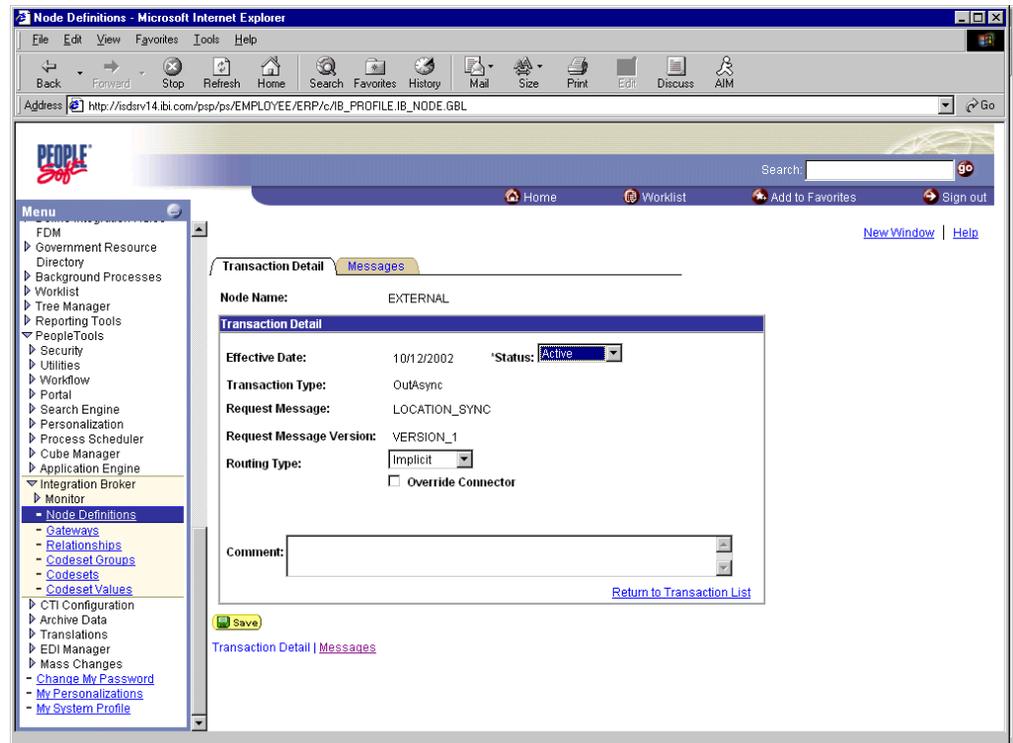
The Node Transactions pane is displayed.



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.



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.

See "[Viewing the PeopleCode for a Message](#)" on page D-25 for more information.

## 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.

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.

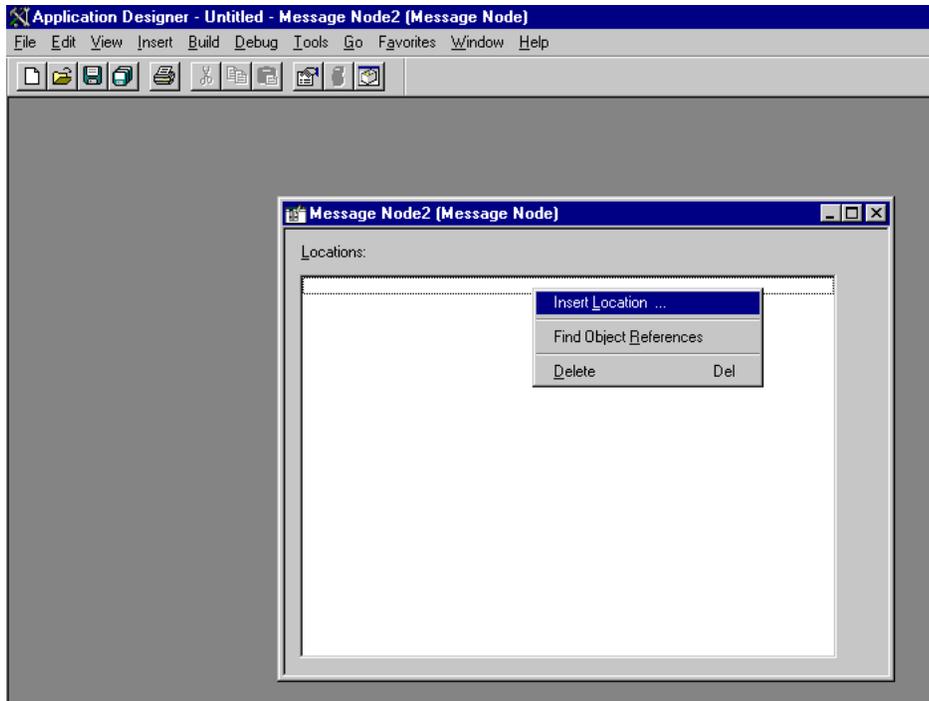
These tasks are described in detail in the following procedures.

### Creating and Configuring a New Message Node

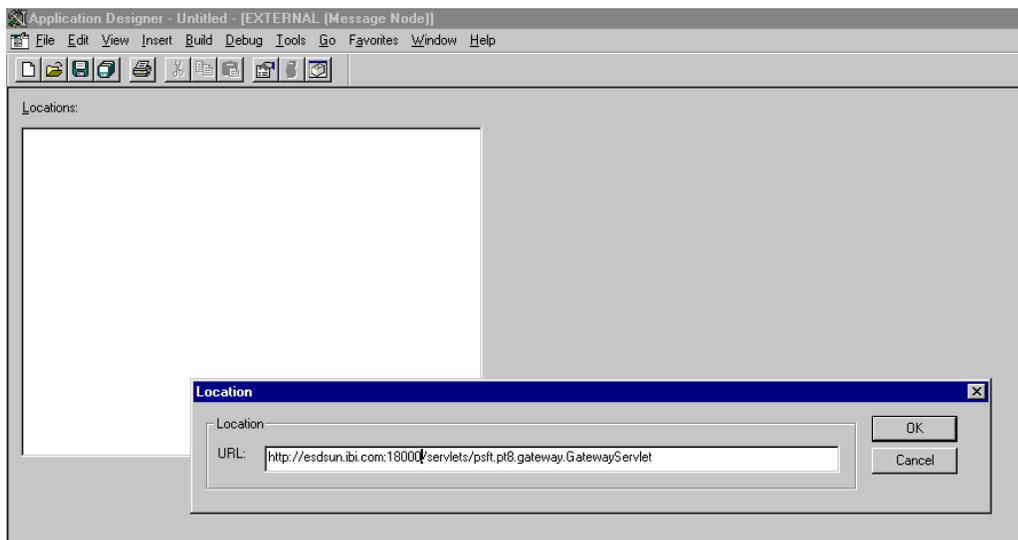
To create and configure a new message node:

1. Select **New** from the File menu and click **Message Node**.

A Message Node window is displayed.



2. Right-click anywhere inside the white space and select **Insert Location**.  
The Location URL box is displayed.

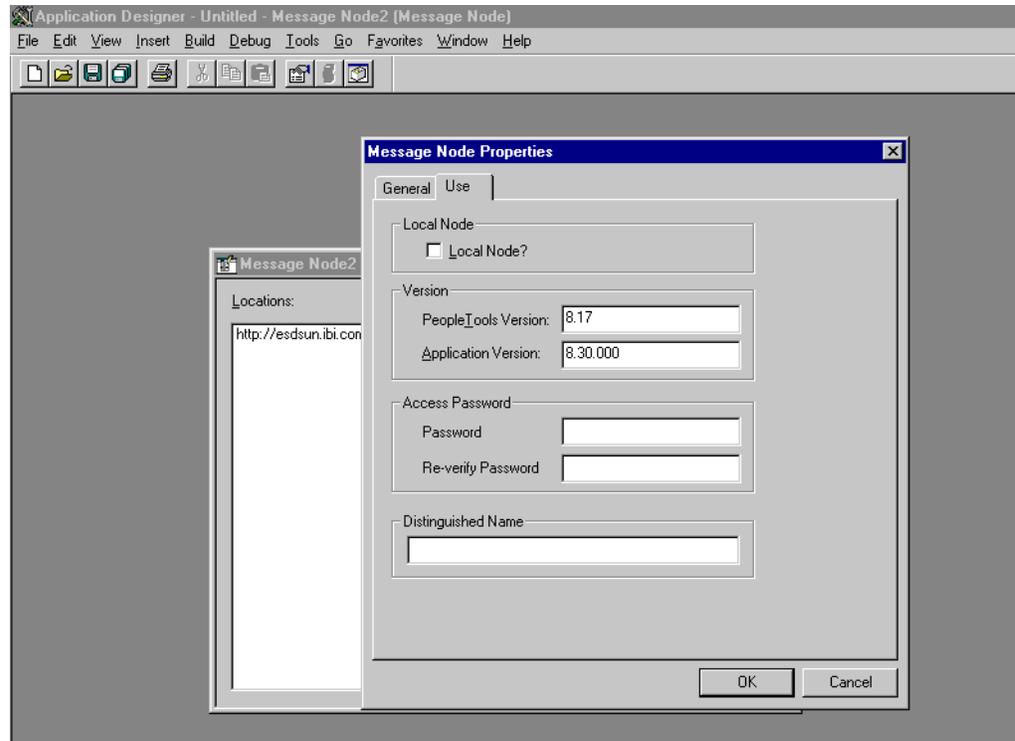


3. Enter the following URL for the PeopleSoft Application Gateway (handler directory):

*host name:port/servlets/psft.pt8.gateway.GatewayServlet*

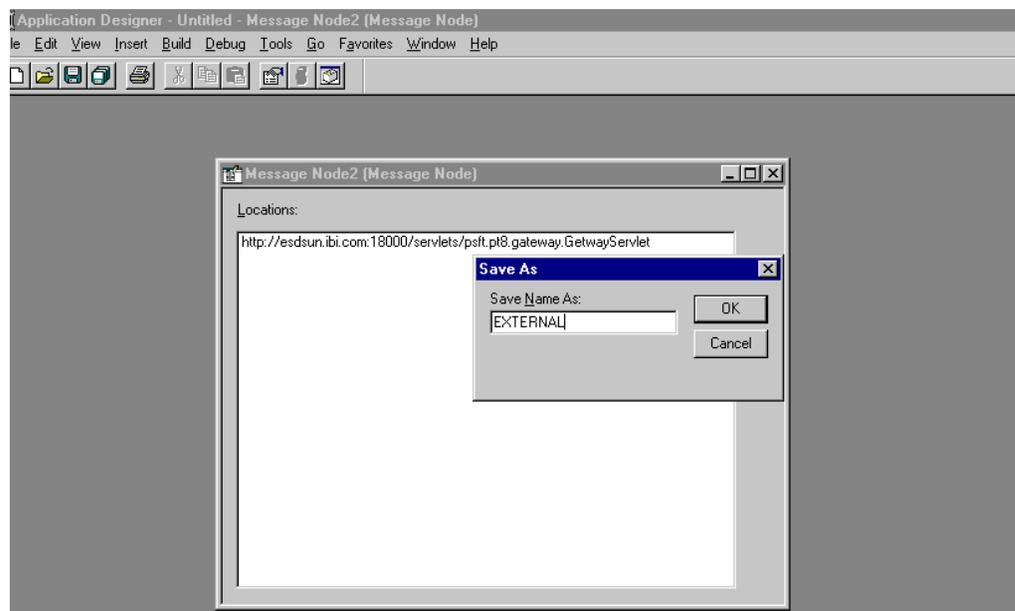
Where *host name* is the host name of your PeopleSoft Web server and *port* is the socket on which the server is listening. The characters you enter after *host name* are case-sensitive.

4. Click **OK**.
5. Display the Message Node Properties dialog box.



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 box.



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.

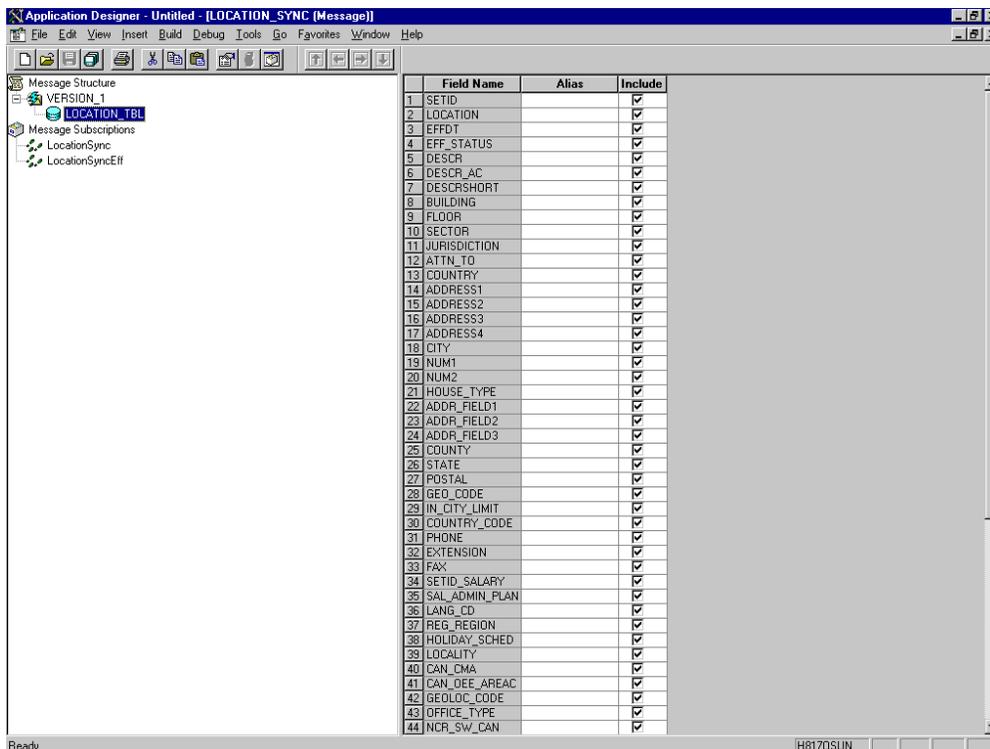
If you intend to migrate this message node to a different PeopleSoft environment (for example, from Test to QA), you can create a PeopleSoft project and insert the Message Node into the project.

You have finished creating and configuring the message node.

### Ensuring the Message Is Active and Is Routed Correctly

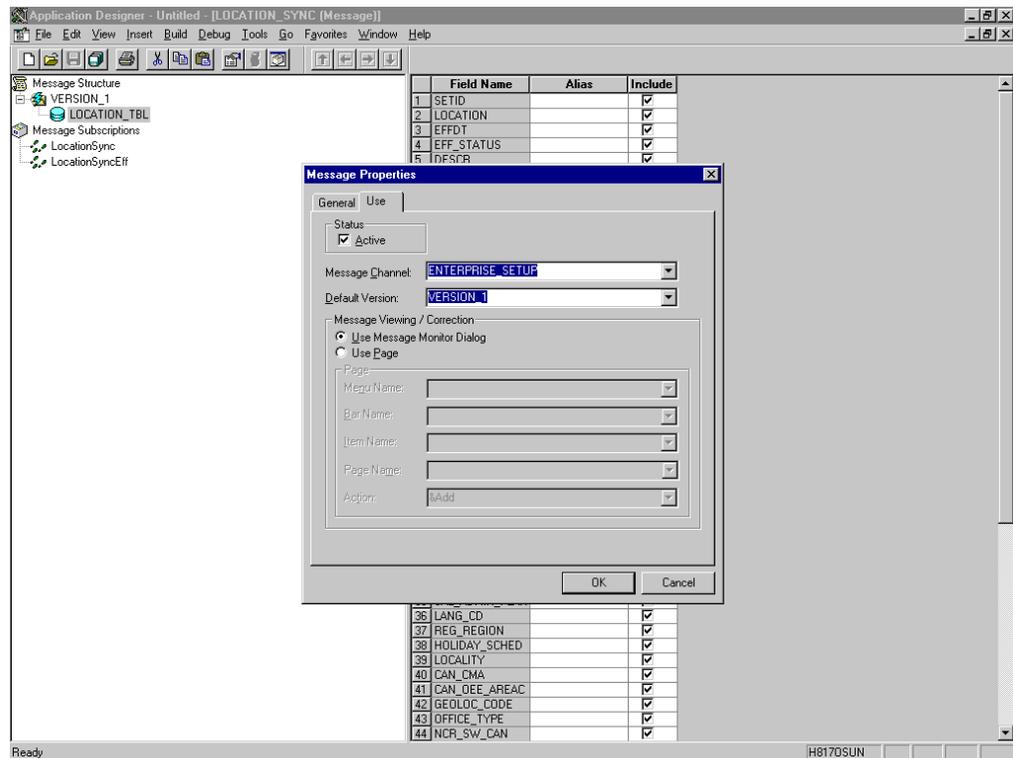
To ensure that the message is active and is routed to the proper message channel:

1. Open Application Designer.
2. On the **File** menu, point to **Open**, click **Message**, and open the **LOCATION\_SYNC** message.



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 box is displayed.



Perform the following steps:

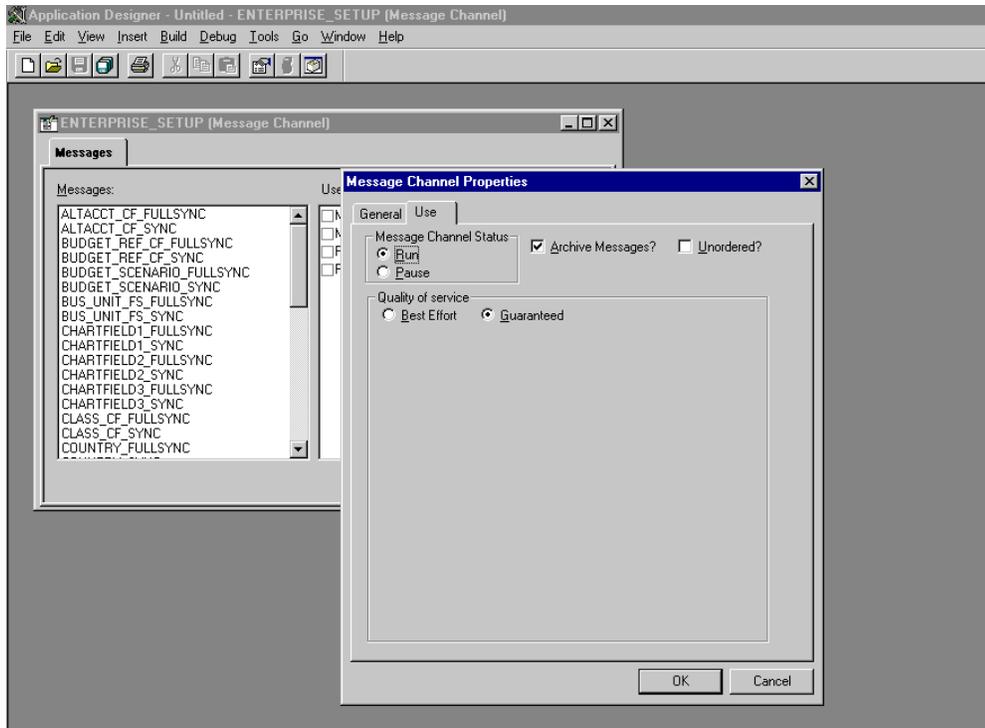
- a. Select the **Use** tab.
  - b. Ensure the **Status** check box is selected, which indicates that the message is active.
  - c. From the Message Channel list, select **ENTERPRISE\_SETUP**.
  - d. From the Default Version list, select **VERSION\_1** (messages can have multiple versions).
5. Click **OK**.
  6. Save the message.

### 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**.

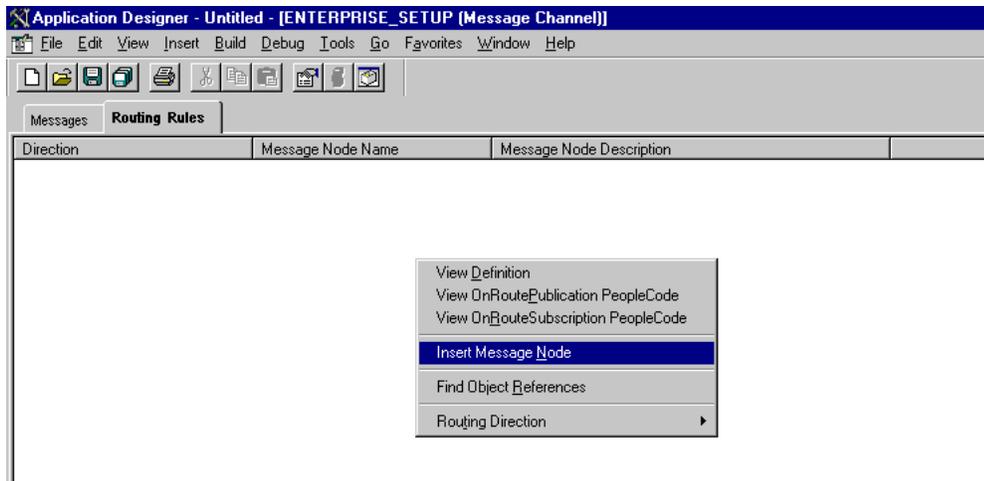
The Message Channel Properties dialog box is displayed.



Perform the following steps:

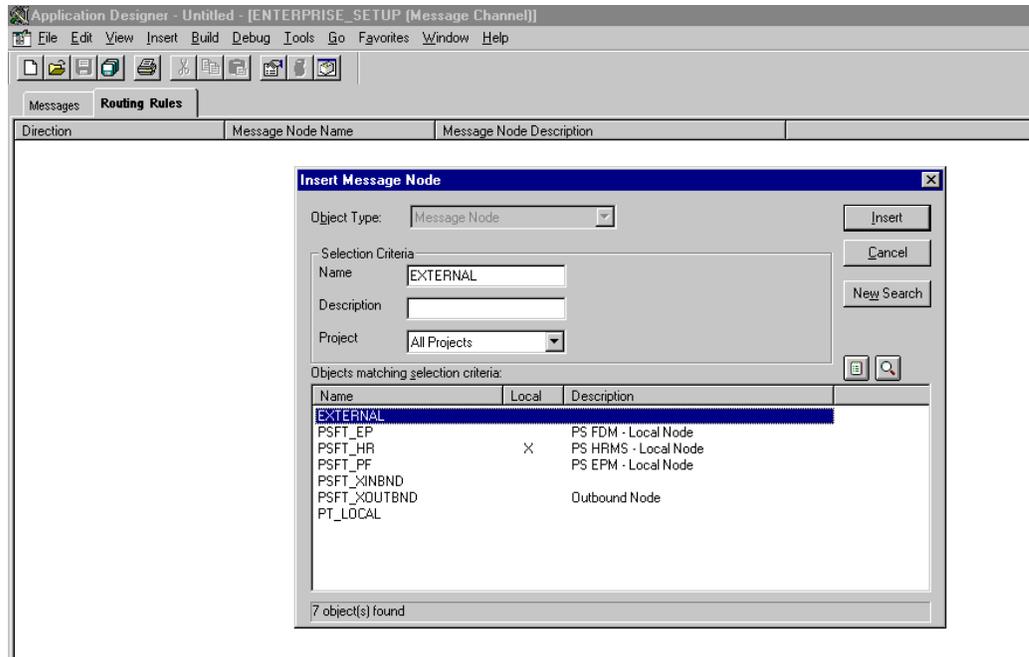
- a. Select the **Use** tab.
  - b. Ensure that the Message Channel status is set to **Run**.
  - c. Click **OK**.
4. From the left pane, select the **Routing Rules** tab.

The pane is blank.



5. Right-click the pane and select **Insert Message Node**.

The Insert Message Node dialog box is displayed.

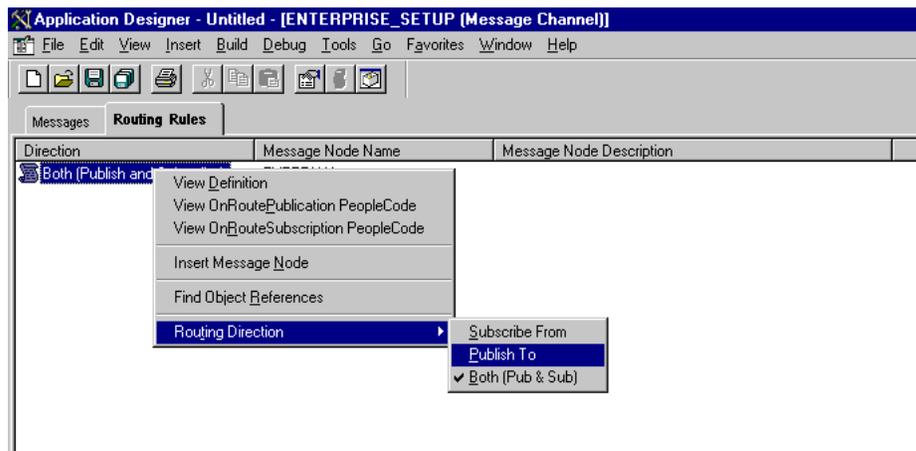


Perform the following steps:

- a. Select the message node that you created in "Creating and Configuring a New Message Node" on page D-11, for example, EXTERNAL.
- b. Click **Insert**.

6. Click **Cancel**.

Information appears on the 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.

### 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*

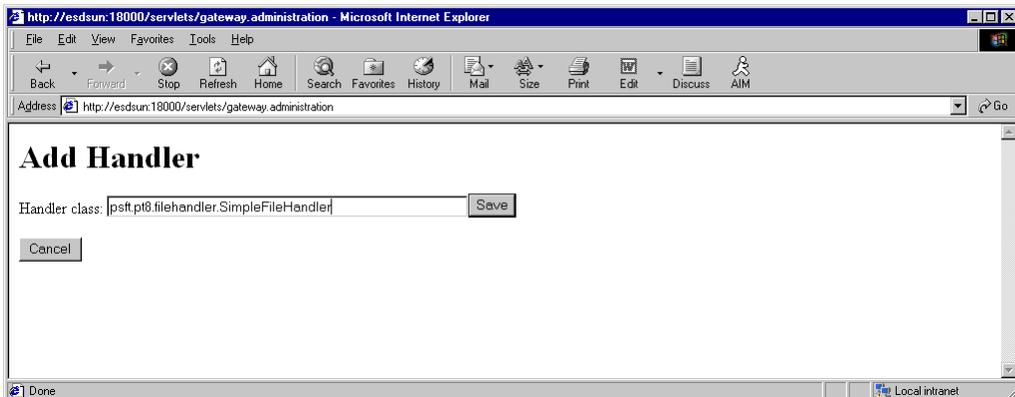
Where *host name* is the name of the application server that hosts PeopleSoft and *port* is the port number on which the application server is listening.

The Handler Directory window is displayed.



2. Click **Add Handler**.

The Add Handler window is displayed.



3. Enter the full name of the Simple File Handler class, `psft.pt8.filehandler.SimpleFileHandler`.

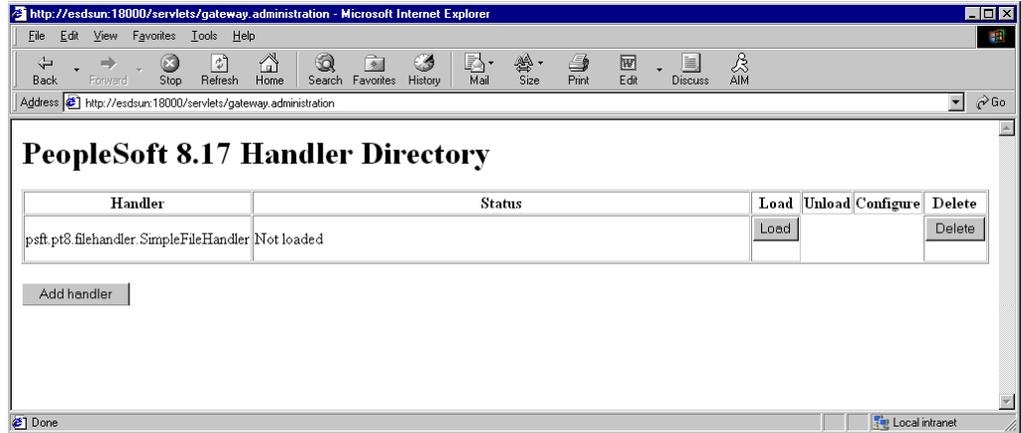
---

**Note:** The name is case-sensitive.

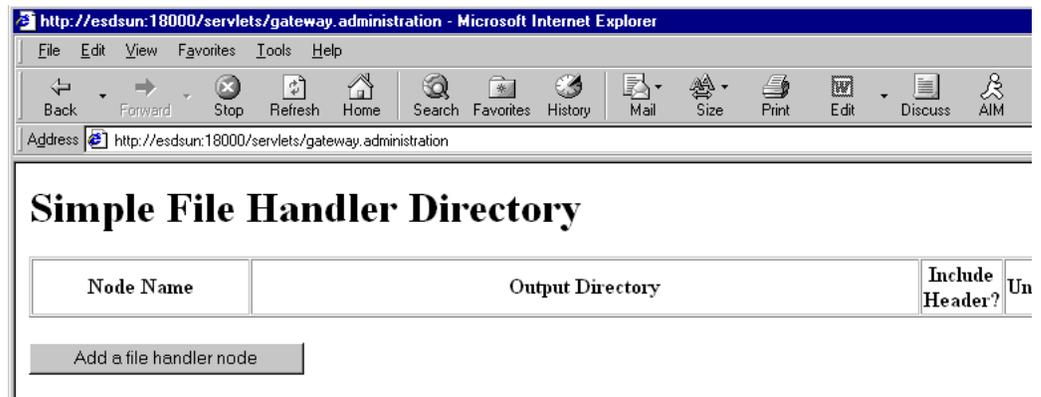
---

4. Click **Save**.

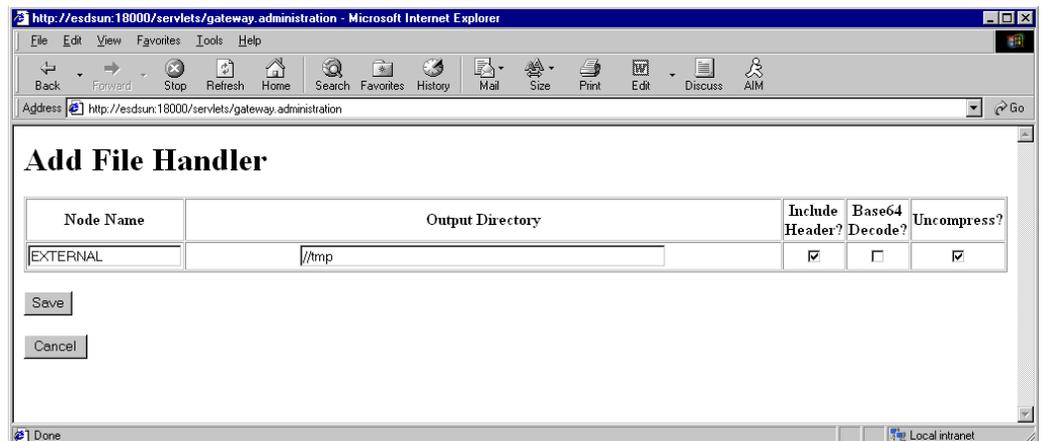
The Handler Directory window is displayed.



- To load the handler, click **Load**.  
After the handler loads, "Loaded successfully" appears in the Status column.
- Click **Configure**.  
The Simple File Handler Directory window is displayed.



- Click **Add a file handler node**.  
The Add File Handler window is displayed.



- 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-11, for example, EXTERNAL.

## 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.

- Open Application Designer.

Num	Field Name	Type	FDe	FEd	FCh	FFo	RIn	RIs	RDc	RSe	SEd	SPr	SPo	Srl	SrS	Wk	PPr
1	SETID	Char															
2	LOCATION	Char											<input checked="" type="checkbox"/>				
3	EFFDT	Date															
4	EFF_STATUS	Char									<input checked="" type="checkbox"/>						
5	DESCR	Char										<input checked="" type="checkbox"/>					
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									<input checked="" type="checkbox"/>						
34	COMMENTS_2000	Long															

- On the **Record Fields** tab, select the **LOCATION\_TBL** record.
- Select the PeopleCode display option.
- Select the **Save Post Change (SPo)** box for the **LOCATION** field.

The following window displays the PeopleCode that initiates a LOCATION\_SYNC message.

```

Application Designer - Untitled - [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

```

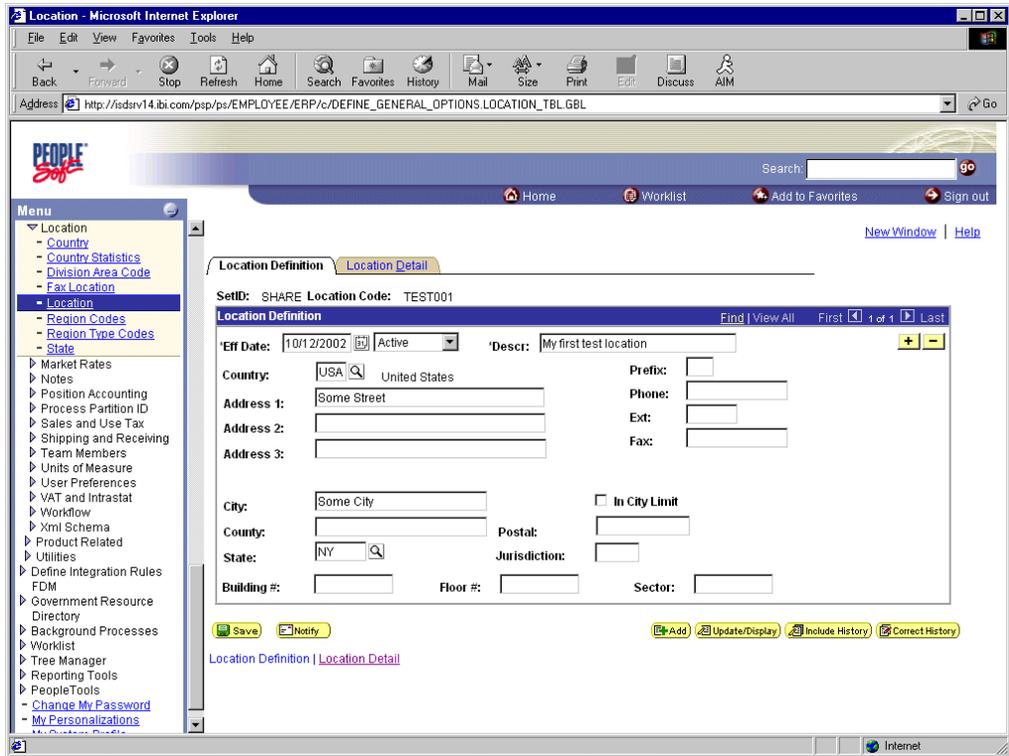
For more information about PeopleCode, consult your PeopleSoft Online Library.

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).

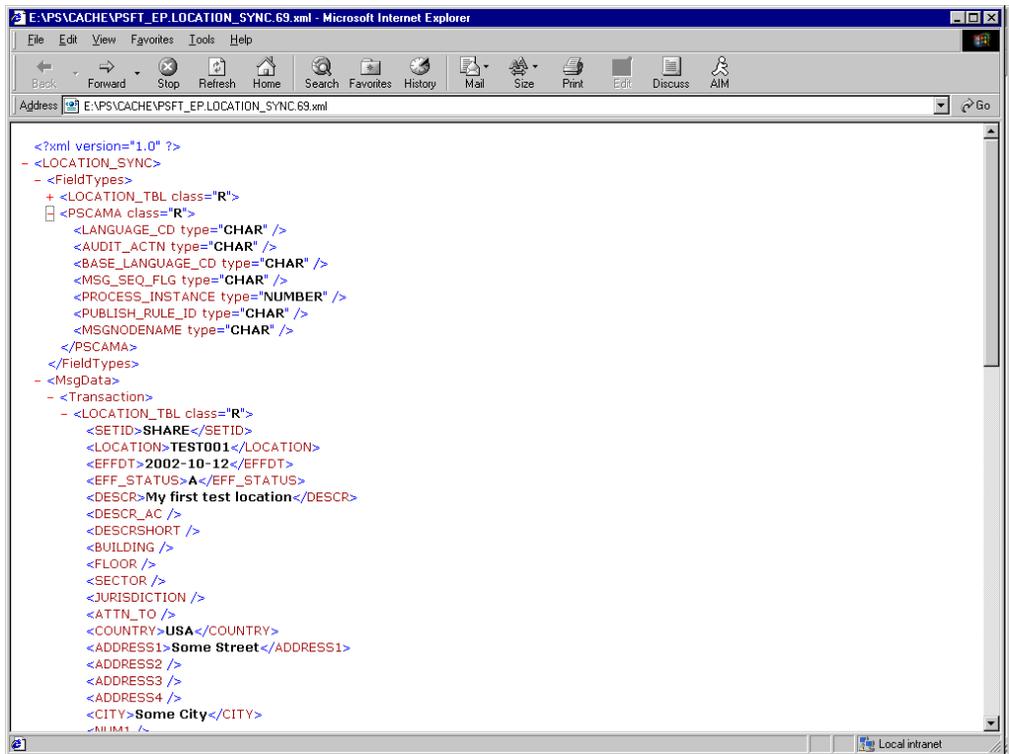
## Testing the Integration Broker

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.

The following example illustrates a Financials 8.4 application where a new location with a SetID of SHARE and a Location Code of TEST001 was added.



The following figure shows a portion of the XML output.



---



---

**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.

## 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.

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.

OracleWLS 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. See ["Configuring Integration Broker in PeopleSoft 8.4"](#) on page D-2 for more information on outbound asynchronous messages.

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`. See ["Ensuring the Message Is Active and Is Routed Correctly"](#) on page D-3 for information on examining these messages.

---



---

## Configuring Outbound Synchronous Messages

You can use an existing node, or you can create a new node to configure outbound synchronous messages. See ["Creating and Configuring a New Gateway Node"](#) on page D-7 for information on creating and configuring a node. In either case, you must set up your outbound synchronous transaction.

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

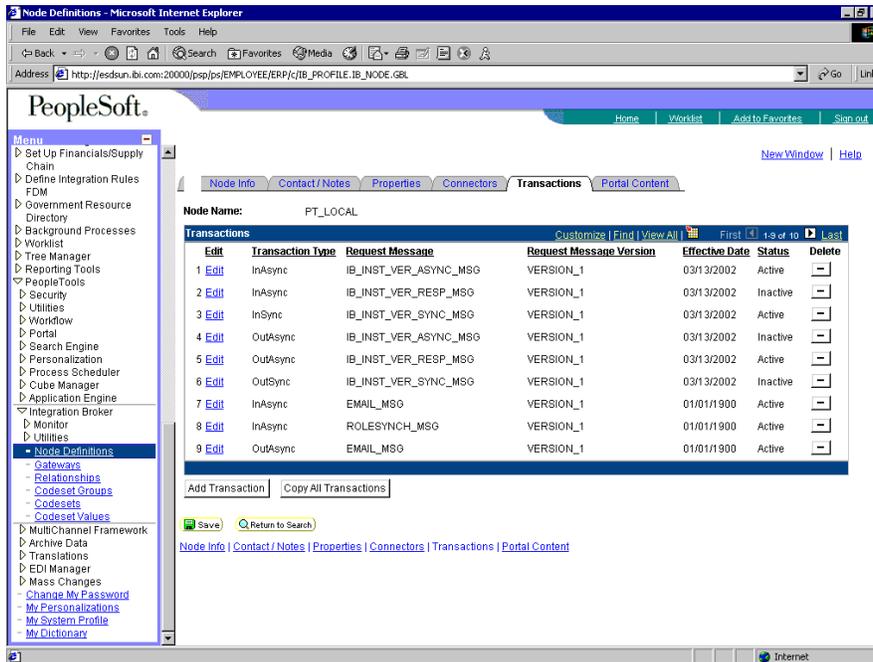
without additional steps. As of Financials release 8.42, there are no preconfigured outbound synchronous transactions that you can use for testing purposes.

### 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.

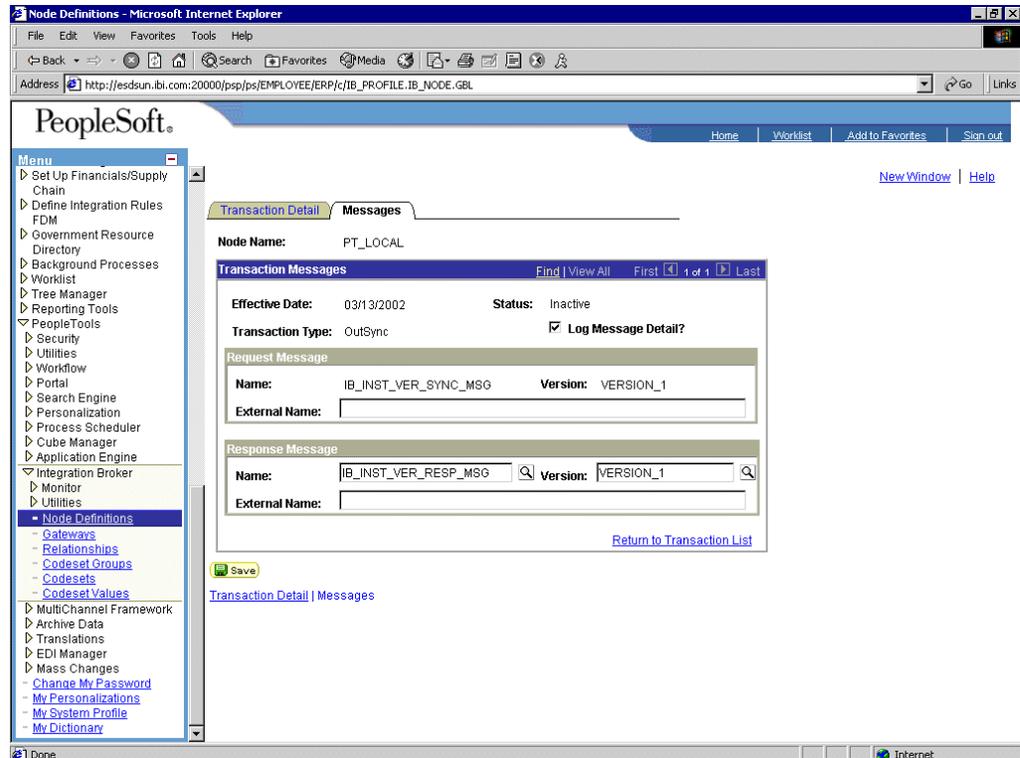
The Transactions pane is displayed.



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.



#### 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 TCP/IP84TARGET connector for outbound synchronous messages.

---

## 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.

### 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.

```

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);

```

## Generating Events Using PeopleTools Tutorial

This section provides a tutorial that walks you through the PeopleSoft event generation process using PeopleTools version 8.48 and 8.49. 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.

## Configuring PeopleSoft Services

This section describes how to configure PeopleSoft services.

### 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.

Node Definitions | Connectors | Portal | W/S Security | Routings

Node Name: EXTERNAL Copy Node

Description: Message publish to third party Rename Node

Node Type: External Default Local Node  
 Default Local Node  
 Local Node  
 Active Node  
 Non-Repudiation  
 Segment Aware Delete Node

Authentication Option: None

Default User ID: PS

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.

Node Definitions | **Connectors** | Portal | W/S Security | Routings Ping Node

Node Name: EXTERNAL

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://bpelclient:1971

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.

### Selecting a 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.

**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

3. From the Existing Operations section on the lower-left, click the **DEPT\_SYNC.VERSION\_1** service operation.

The Service Operation - General pane is displayed.

**Default Service Operation Version**

Version: VERSION\_1    **Default**    **Active**

Version Description: Dept Sync Incr. Message

Version Comments:   
 Non-Repudiation  
 Runtime Schema Validation

[Introspection](#)

**Routing Status**

Any-to-Local: Exists  
 Local-to-Local: Exists

**Routing Actions Upon Save**

Regenerate Any-to-Local  
 Regenerate Local-to-Local

4. Click the **Active** check box.
5. Click the **Routings** tab to add a new routing.

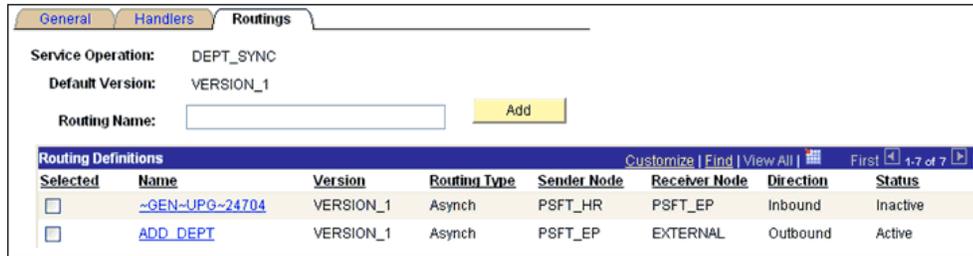
6. In the Routing Name field, type a name for the new routing, for example, **ADD\_DEPT**.
7. Click **Add**.
8. Click the **Routing Definitions** tab to add a new routing definition.

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.

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.



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.

### 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.

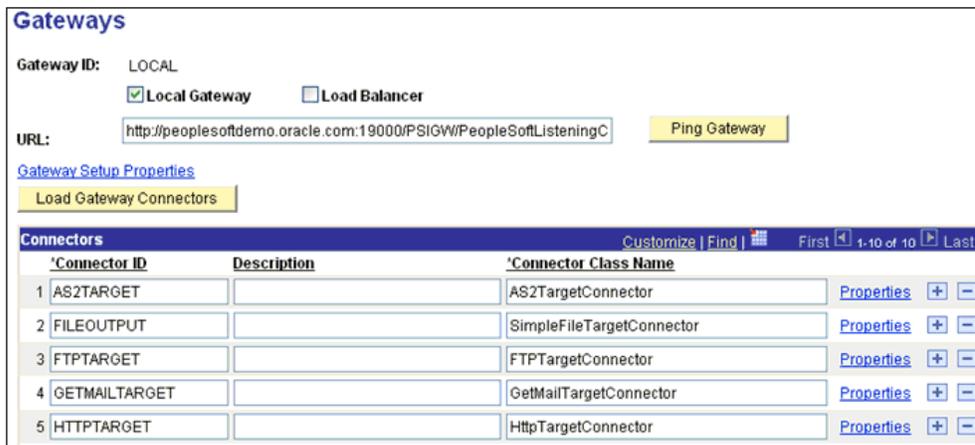


3. Click **Ping Gateway**.

A new browser window is displayed, which shows that the gateway is active.



4. Close the new browser window to return to the Gateways pane.



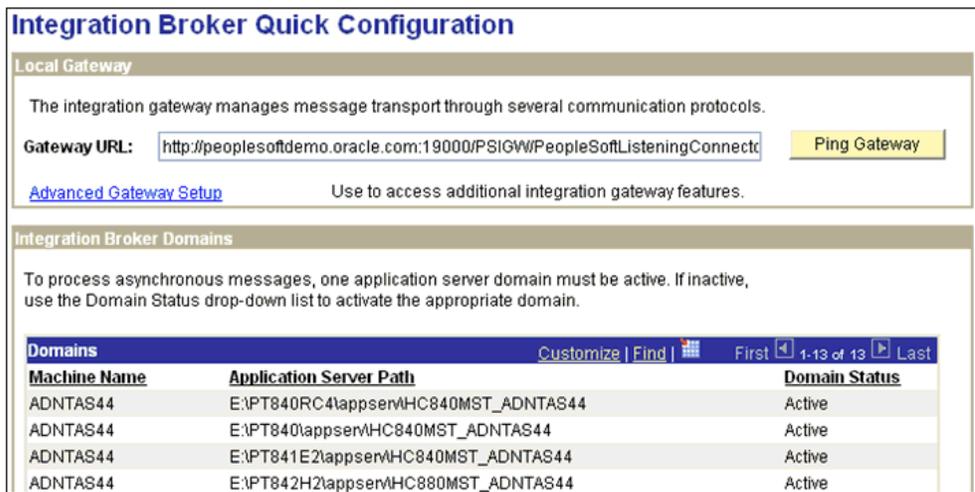
5. Ensure that **HTTPTARGET** is listed in the Connectors section. If it is not listed, click **Load Gateway Connectors** to refresh the available connectors.

### Activating the Domain

To activate the domain:

1. Select **PeopleTools, Integration Broker, Configuration**, followed by **Quick Configuration**.

The Quick Configuration pane is displayed.



2. Ensure that the domain is active.

### Activating the Service

To activate the service:

1. Select **PeopleTools, Integration Broker, Integration Setup**, followed by **Services**.

The Services pane is displayed.

2. Ensure that the service is active.

### 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.

Existing Operations			
Operation	Default Version	Description	Active
<a href="#">ADD_DEPT.v1</a>		Add Department	<input checked="" type="checkbox"/>
<a href="#">DEPT_SYNC.VERSION_1</a>		Dept Sync Incr. Message	<input checked="" type="checkbox"/>

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.

- In the message information section on the bottom of the page, note the value in the Queue Name field of the service operation.

**Message Information**

**Type:** Request

**Message.Version:**  [View Message](#)

**Queue Name:**  [View Queue](#) [Add New Queue](#)

- Click the **View Queue** link.  
The Queue Definitions pane is displayed.

**Queue Definitions**

**Queue Name:** ENTERPRISE\_SETUP  **Archive**  **Unordered**

**Description:**  **Queue Status:**

**Comments:**  **Object Owner ID:**

**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="text"/>
<input type="checkbox"/>	PUBLISHER	<input type="text"/>
<input type="checkbox"/>	PUBPROC	<input type="text"/>

- Ensure that the Queue Status drop-down list has **Run** selected.
- Click **Return** to return to the Service operation pane.
- Click the **Routing** tab.

**General** **Handlers** **Routing**

**Service Operation:** DEPT\_SYNC

**Default Version:** VERSION\_1

**Routing Name:**

**Routing Definitions**

Selected	Name	Version	Routing Type	Sender Node	Receiver Node	Direction	Status
<input type="checkbox"/>	~GENERATED~94652240	VERSION_1	Asynch	~ANY~	PSFT_HR	Inbound	Active
<input type="checkbox"/>	~GEN~UPG~12357	VERSION_1	Asynch	PSFT_HR	PSFT_CR	Outbound	Inactive
<input type="checkbox"/>	ADD_DEPT	VERSION_1	Asynch	PSFT_HR	EXTERNAL	Outbound	Active
<input type="checkbox"/>	~GEN~UPG~20085	VERSION_1	Asynch	PSFT_HR	PSFT_EP	Outbound	Inactive
<input type="checkbox"/>	~GEN~UPG~25397	VERSION_1	Asynch	PSFT_HR	PSFT_HR	Local	Inactive
<input type="checkbox"/>	~GEN~UPG~13180	VERSION_1	Asynch	PSFT_HR	PSFT_LM	Outbound	Inactive

- Ensure that the routings are active.
- Click the **Return to Service** link on the bottom of the pane to return to the Services pane.
- 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.

12. Ensure that the status of the External node is active.

This completes the configuration on the PeopleSoft side.

## Triggering the Event in PeopleSoft

This section describes how to trigger the event in PeopleSoft.

### 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.

3. Click the **Add a New Value** tab.

The screenshot shows a web form titled "Departments". At the top, there are two tabs: "Find an Existing Value" and "Add a New Value". The "Add a New Value" tab is selected. 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.

The screenshot shows the "Department Profile" form. At the top, there are two tabs: "Department Profile" and "Comm. Acctg. and EG". The "Department Profile" tab is selected. Below the tabs, there are several fields: "SetID:" with the value "SHARE", "Department:" with the value "BPELTEST01", and a link "Business Units that use this Setid". Below these are search fields for "Location SetID:", "Location:", and "Company:". The "Effective Date:" is "08/25/2008" and "Status:" is "Active". The "Description:" is "BPEL Inbound Test" and "Short Description:" is "BPEL Inbou". The "Manager Type" section has three radio buttons: "None" (selected), "EmpID" (with "Manager ID:" field), and "Position" (with "Manager Position:" field). The "EmpID:" field is empty. The "Budget Year End Date:" is empty and "Budget Level:" is "None". The "Payroll for North America" section has a "Tax Location:" field.

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.

## Verifying the Event Results

This section describes how to verify the event results.

### 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.

Queue Name	Error	New	Started	Working	Done	Retry	Timeout	Edited	Canceled	Hold
ENTERPRISE_SETUP	0	0	0	0	<a href="#">5</a>	0	0	0	0	0
PSRF_REPORTING_FOLDERS	0	0	0	0	<a href="#">1366</a>	0	0	0	0	0
PSXP_MSG_CHNL	0	0	0	0	<a href="#">1</a>	0	0	0	0	0
USER_PROFILE	0	0	0	0	<a href="#">1</a>	0	0	0	0	0

Here you can view the results of the events at the queue level.

- Click the number link (5) queue (Enterprise\_Setup) in the Result section.

The Operation Instances tab is displayed.

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.

- Click the **Details** link on the lower-right.

A new browser window (Asynchronous Details) is displayed.

Subscriber Node	Segment	Status	View XML	Resubmit	Cancel	Error Messages	View IB Info
EXTERNAL	1	Done	<a href="#">View XML</a>	<a href="#">Resubmit</a>	<a href="#">Cancel</a>	<a href="#">Error Messages</a>	<a href="#">View IB Info</a>

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.



---

---

# Index

## A

---

- access methods, 1-2
- access rights, 7-1
- adapter configuration
  - overwriting, 3-5
  - updating, 3-2
- adapter connections, 3-4
- adapter exceptions, 6-2 to 6-3
- Adapter Lib Directory parameter, 2-3 to 2-4
- adapter types
  - resource, 1-1, 3-1 to 3-2
- AdapterName parameter, 3-4 to 3-5
- adapters, 1-1 to 1-2
  - configuring, 2-1 to 2-27, 3-2
  - deploying, 1-1, 1-3, 3-1 to 3-2
  - troubleshooting, 6-1 to 6-7
- Add Channel dialog box, 2-23
- Admin Password parameter, 2-3 to ??
- Advanced tab, 2-26
- ae batch script, 6-1
- API Java programs
  - compiling, A-4 to A-6
  - creating, A-1 to A-4
- application adapters, 1-1 to 1-2
  - configuring, 2-1 to 2-27, 3-2
  - deploying, 1-1, 1-3, 3-1 to 3-2
  - troubleshooting, 6-1 to 6-7
- application clients, 3-4
- Application Explorer, 1-2 to 1-3, 2-11, 2-14
  - application systems and, 2-11
  - debugging and, 6-1 to 6-2
  - J2CA configuration and, 3-2
  - OracleAS Adapter J2CA and, 6-4
  - security and, 7-1, 7-6
  - testing and, 6-1 to 6-2
  - troubleshooting, 6-1 to 6-2
- Application Messaging, 1-2
- Application Messaging Manager, 1-3
- Application Server dialog box, 2-12
- Application Server parameter, 2-12
- application system objects, 2-14
- application systems
  - Application Explorer and, 2-11
  - supported, 2-11
- asynchronous support, 1-2

Available list, 7-4 to 7-5

## B

---

- browsing metadata, 2-14
- BSE (OracleAS Adapter Business Services Engine), 2-8
  - configuring, 2-8
  - connection access to, 7-6
  - troubleshooting, 6-4
- BSE (OracleWLS Adapter Business Services Engine), 1-1, 1-3
- BSE configuration page, 2-2 to 2-3
- BSE control service URL, 7-8
- BSE deployment, 1-3
- BSE repositories
  - migrating, 7-8
- BSE settings window, 2-4
- BSE system settings, 2-3 to 2-5
- BSE URL field, 2-9
- business events, 1-1
- business functions, 2-18
- business objects, 2-11
  - browsing, 2-14
  - storing, 2-4
- business processes, 1-2
- business services
  - creating, 2-18 to 2-19
  - deploying, 7-1
  - testing, 2-19
- Business Services node, 7-2 to 7-4, 7-7

## C

---

- CCI (Common Client Interface), 3-1
- CCI calls, 3-1
- channel configuration parameters, 2-24
  - Error Directory, 2-26
  - File Mask, 2-26
  - Host, 2-24
  - Is Keep Alive, 2-24
  - Is Length Prefix, 2-24
  - Is XML, 2-24
  - Poll interval, 2-26
  - Polling Location, 2-26
  - Port, 2-22

- Port Number, 2-24
- Processing Mode, 2-26
- Response/Ack Directory, 2-26
- Server port, 2-22
- Synchronization Type, 2-24 to 2-26
- Thread limit, 2-26
- channel types
  - File, 2-25 to 2-27
- channels, 2-20 to 2-27
  - creating, 2-20 to 2-27
  - deleting, 2-27
  - editing, 2-27
  - ports and, 2-23
  - starting, 2-25
  - stopping, 2-25
- channels. *See also* listeners
- closing connections, 2-13
- Common Client Interface (CCI), 3-1
- component interface API, 1-3
- component interface properties, 6-2
- Component Interface testing tool
  - three-tier mode and, 6-3
- component interfaces, 1-2 to 1-3, 2-14
  - creating, C-1 to C-4
  - security and, C-5 to C-12
  - testing, C-12 to C-16
- component methods, 1-3
- configuration parameters, 3-3, 3-5
  - IWayConfig, 3-3
  - IWayHome, 3-3
  - IWayRepoPassword, 3-3
  - IWayRepoURL, 3-3
  - IWayRepoUser, 3-3
  - Loglevel, 3-3
- Configuration under Business Services
  - node, 7-2 to 7-4, 7-7
- configurations
  - defining, 2-8 to 2-9
  - overwriting, 3-3 to 3-5
  - testing, B-14 to B-15
- Configurations node, 2-9
- configuring adapters, 3-2
- configuring BSE system settings, 2-3 to 2-5
- 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-15 to D-17
- configuring message routers, B-1 to B-14
- configuring nodes
  - Gateway, D-7 to D-11
  - TCP/IP84 connector, B-3 to B-6
- configuring repositories, 2-5 to 2-6, ?? to 2-6
- configuring target connectors, B-1 to B-14
- connecting to OracleAS Adapter J2CA, 6-4
- connecting to PeopleSoft, 2-11 to 2-13, 6-2, 6-6
- connection factories, 3-4
- connection parameters, 2-12, 6-2
  - AdapterName, 3-4 to 3-5
  - Application Server, 2-12
  - Config, 3-4
  - Country, 3-4
  - Hostname, 2-9, 6-2 to 6-3
  - Language, 3-4
  - Loglevel, 3-4
  - Password, 2-12, 3-4
  - Port, 2-2, 2-12, 2-22, 6-2 to 6-3
  - User, 2-12
  - UserName, 3-4 to 3-5
- connection pooling, 3-4
- connections
  - closing, 2-13
  - deleting, 2-14
  - establishing, 2-11 to 2-13
- ConnectionSpec, 3-4 to 3-5
- Connector
  - deploying to Oracle Application Server, 3-1
- connector factories, 3-2
- connector factory objects, 3-2
  - multiple, 3-3
- connectors
  - configuring, B-1 to B-14
- control methods, 7-9
- Country parameter, 3-4
- Create Business Service dialog box, 2-18
- creating channels, 2-20 to 2-27
- creating repository projects, 2-8 to 2-9
- creating schemas, 2-15, 6-2
- creating Web services, 2-18 to 2-19

## D

---

- database connections
  - opening, 3-3
- databases
  - connecting to, 3-3
  - Oracle, 3-3
- Debug Level parameter, 2-4
- DEBUG log level, 3-5
- deleting channels, 2-27
- deleting connections, 2-14
- deploying adapters, 3-1 to 3-2
- deployments
  - BSE, 1-3
- Description field, 2-18, 7-2 to 7-5, 7-7
- design time, 7-8
- design time service adapter connections, 3-4
- disconnecting from PeopleSoft targets, 2-13
- DNS Lookup option, 7-7
- DNS name, 7-7
- Domain Name System (DNS), 7-7
- domain names, 7-7
- Domain option, 7-7

## E

---

- editing channels, 2-27
- editing targets, 2-13
- EIP (Enterprise Integration Points), 1-2

EIS (enterprise information systems), 3-6  
EJB (Enterprise Java Beans), 3-1  
Encoding parameter, 2-4  
Enterprise Connector for J2EE Connector Architecture (J2CA), 1-3  
enterprise information systems (EIS), 3-6  
Enterprise Integration Points (EIP), 1-2  
Enterprise Java Beans (EJB), 3-1  
Error Directory parameter, 2-26  
error messages, 6-2 to 6-7  
    target systems and, 6-5  
events, 1-1, 1-3  
Execution Denied list, 7-6  
Execution Granted list, 7-6  
Existing Service Names list, 2-18

## F

---

fault code elements, 6-5  
fault string elements, 6-5  
File channel, 2-25 to 2-27  
File Mask parameter, 2-26  
File Output Connector  
    configuring, D-5 to D-7  
file system repositories  
    configuring, 2-5

## G

---

Gateway node, D-7 to D-11  
generating schemas, 2-15, 6-2  
Grant Access check box, 7-7  
Group (of Computers) option, 7-7  
groups  
    creating, 7-3  
Groups node, 7-3 to 7-4

## H

---

Host parameter, 2-24  
Hostname parameter, 2-9, 6-2 to 6-3  
HTTP protocol, 2-23

## I

---

input records  
    creating, 3-6  
installation directories, 3-3  
instances of policy types, 7-1  
Integration Broker, 1-2 to 1-3, D-1 to D-25  
    configuring, D-2  
Integration Gateway  
    configuring, D-5 to D-7  
IntegrationGateway.properties file  
    configuring, D-4 to D-5  
interactions  
    creating, 3-6  
    executing, 3-6  
IP (Mask)/Domain field, 7-7  
IP addresses, 7-7  
IP and Domain Restriction policy type, 7-6

Is Keep Alive parameter, 2-24  
Is Length Prefix parameter, 2-24  
Is XML parameter, 2-24  
IWAE function, 3-6  
IWAConnectionSpec, 3-4 to 3-5  
IWAInteractionSpec, 3-6  
IWayConfig parameter, 3-3  
IWayHome parameter, 3-3  
IWayRepoPassword parameter, 3-3  
IWayRepoURL parameter, 3-3  
IWayRepoUser parameter, 3-3  
iwpsci84.jar file, 6-2  
iwpsci8x.jar file, 6-2  
iwse.ora file, 2-5, 2-6

## J

---

J2CA (Enterprise Connector for J2EE Connector Architecture), 1-3  
    Oracle Application Server Adapter and, 3-1  
J2CA configuration  
    Application Explorer and, 3-2  
J2CA repositories  
    migrating, 7-10  
J2CA resource adapters, 1-1, 3-1  
JAR files, 6-2  
    errors and, 6-2  
Java application clients, 3-4  
Java program clients, 3-1  
JNDI lookup, 3-4

## L

---

Language parameter, 2-4, 3-4  
License and Method dialog box, 2-18  
License field, 2-18  
licenses, 2-18  
list of nodes, 6-2  
listeners, 1-4, 2-10, B-10  
log files, 6-1  
log levels  
    overwriting, 3-3 to 3-5  
Loglevel parameter, 3-3 to 3-4

## M

---

managed connector factories, 3-2  
managed connector factory objects, 3-2  
    multiple, 3-3  
ManagedConnectionFactory parameter, 3-3 to 3-5  
mapping security, 3-5  
message channels  
    configuring, D-15 to D-17  
message routers  
    configuring, B-1 to B-14  
messages, 1-1, 1-3  
metadata, 2-14  
    browsing, 2-14  
    storing, 2-4, 7-8  
Method Name field, 2-18  
methods, 7-1

- changing, C-4
- viewing, C-4
- migrating repositories, 7-8
- migrating Web services, 7-10

## N

---

- Name field, 7-2 to 7-5
- New Configuration dialog box, 2-8 to 2-9
- New Group dialog box, 7-3
- New Policy permissions dialog box, 7-5
- New User dialog box, 7-2
- Node list, 6-2
- nodes
  - Business Services, 7-2 to 7-4
  - Configuration under Business Services, 7-2 to 7-4, 7-7
  - Configurations, 2-9
  - Gateway, D-7 to D-11
  - Groups, 7-3 to 7-4
  - PeopleSoft, 2-11
  - Policies, 7-4
  - Security, 7-2 to 7-4, 7-7
  - Service Adapters, 2-11
  - TCP/IP84 connector, B-3 to B-6
  - Users, 7-3
  - Users and Groups, 7-2 to 7-3
- Number of Async. Processors parameter, 2-4

## O

---

- Oracle Application Server
  - deployment of Connector to, 3-1
- Oracle Application Server Adapter
  - installation directory and, 3-3
  - J2CA and, 3-1
- Oracle databases, 3-3
- Oracle repositories
  - migrating, 7-8
- OracleAS Adapter Business Services Engine (BSE), 2-8
  - configuring, 2-8
  - connection access to, 7-6
  - troubleshooting, 6-4
- OracleAS Adapter for PeopleSoft
  - troubleshooting, 6-1 to 6-7
- OracleAS Adapter J2CA, 2-8 to 2-9, 3-5
  - Application Explorer and, 6-4
  - connecting to, 6-4
- OracleWLS Adapter Application Explorer. *See also* Application Explorer
- OracleWLS Adapter Business Services Engine (BSE), 1-1, 1-3
- OracleWLS Application Adapter for PeopleSoft
  - configuring, 2-1 to 2-27
  - deploying, 1-1

## P

---

- parameter types
  - channel configuration, 2-24

- configuration, 3-3
- connection, 2-12, 3-4, 6-2
- repository, 2-5
- repository migration, 7-9
- system, 2-4
- Password parameter, 2-2 to 2-3, 2-12, 3-4 to 3-5, 7-2
- passwords, 3-3 to 3-5
- PeopleSoft, 3-6
  - connecting to, 2-11 to 2-13, 6-2, 6-6
- PeopleSoft API Java programs
  - compiling, A-4 to A-6
  - creating, A-1 to A-4
- PeopleSoft business objects, 2-11
- PeopleSoft Java API, 1-2
- PeopleSoft node, 2-11
- PeopleSoft XML, 1-2
- permissions, 7-1
  - denying, 7-6
  - granting, 7-6
- policies, 7-1
  - applying, 7-1
  - creating, 7-4
- Policies node, 7-4
- policy types, 7-1
  - IP and Domain Restriction, 7-6
- policy-based security, 7-1 to 7-7
- Poll interval parameter, 2-26
- Polling Location parameter, 2-26
- Port Number parameter, 2-9, 2-24
- Port parameter, 2-2, 2-12, 2-22, 6-2 to 6-3
- ports
  - channels and, 2-23
- privileges, 7-1
  - setting, 7-1
- PROCESS function, 3-6
- Processing Mode parameter, 2-26
- properties, 3-3
- Protocol list, 2-23
- psjoa.jar file, 6-2

## R

---

- ra.xml file, 2-2
- record types
  - input, 3-6
- records
  - creating, 3-6
- repositories
  - configuring, 2-5 to 2-6, ?? to 2-6
  - migrating, 7-8
- Repository Driver parameter, 2-5
- repository information
  - storing, 2-5
- repository migration parameters, 7-9
- repository parameters
  - Driver, 2-5
  - Password, 2-5
  - Pooling, 2-5
  - Type, 2-5
  - URL, 2-3, 2-5

- User, 2-5
- Repository Password parameter, 2-5
- Repository Pooling parameter, 2-5
- repository projects
  - creating, 2-8 to 2-9
  - Web services and, 2-8
- repository tables
  - creating, 2-5, 2-6
- Repository Type parameter, 2-5
- Repository URL parameter, 2-3, 2-5
- repository URLs, 7-10
- Repository User parameter, 2-5
- repository.xml file, 7-11
- request documents, 2-15
- Request Schema tab, 2-15
- Request tab, 2-26
- resource adapters, 1-1, 3-1 to 3-2
- Resource Execution policy type, 7-1
- response documents, 2-15
- Response Schema tab, 2-15
- Response tab, 2-26
- Response/Ack Directory parameter, 2-26
- runtime, 7-8
- runtime service adapter connections, 3-4

## S

---

- schemas, 2-15
  - creating, 1-3, 2-15, 6-2
- security, 7-1 to 7-7
  - Application Explorer and, 7-1, 7-6
  - component interfaces and, C-5 to C-12
  - configuring, 7-2
  - mapping, 3-5
- Security node, 7-2 to 7-4, 7-7
- security policies
  - applying, 7-1
  - creating, 7-4
- Security Policy option, 7-6
- Selected list, 7-4 to 7-5
- Server port parameter, 2-22
- service adapter connections, 3-4
- Service Adapters node, 2-11
- service names, 2-18
- Service Provider list, 2-9
- services, 1-1, 1-3
  - creating, 2-18 to 2-19
  - testing, 2-19
- servlets, 3-1
- Single (Computer) option, 7-7
- SOAP agents, 6-5
- SOAP envelopes, 7-9
- SOAP faults, 6-5
- soap operation name dialog box, 7-8
- SOAP requests, 6-5 to 6-7, 7-1, 7-5
  - creating, 7-8
  - errors and, 6-5 to 6-7
- SOAP responses, 6-5 to 6-7
- starting a channel, 2-25
- stopping a channel, 2-25

- storing metadata, 7-8
- Synchronization Type parameter, 2-24 to 2-26
- synchronous support, 1-2
- system objects, 2-14
- system parameters
  - Adapter Lib Directory, 2-4
  - Debug Level, 2-4
  - Encoding, 2-4
  - Language, 2-4
  - Number of Async. Processors, 2-4
- system settings
  - configuring, 2-3 to 2-5

## T

---

- target connectors
  - configuring, B-1 to B-14
- target systems
  - errors and, 6-5
- targets
  - connecting to, 2-11 to 2-13, 6-6
  - defining, 2-11 to 2-12
  - deleting, 2-14
  - disconnecting from, 2-13
  - editing, 2-13
- TCP Listener dialog box, 2-23
- TCP protocol, 2-23
- TCP/IP84 connector node, B-3 to B-6
- testing component interfaces, C-12 to C-16
- testing configurations, B-14 to B-15
- testing Web services, 2-19
- Thread limit parameter, 2-26
- trace information, 6-1
- transaction processing, 1-1
- transactions
  - storing, 2-4
- troubleshooting, 6-1 to 6-7
  - Application Explorer, 6-1 to 6-2
  - BSE, 6-4
  - Web services, 6-4 to 6-7
- Type list, 7-5, 7-7

## U

---

- updating adapter configuration, 3-2
- User ID parameter, 2-2, 2-12
- UserName parameter, 3-4 to 3-5
- users
  - associating, 7-2
- Users and Groups node, 7-2 to 7-3
- Users node, 7-3

## V

---

- viewing methods, C-4

## W

---

- Web Service Definition Language (WSDL), 2-15 to 2-16
- Web service names, 2-18

- Web services, 1-1 to 1-3, 2-8
  - creating, 2-18 to 2-19
  - delivering, 2-4
  - deploying, 6-6, 7-1
  - migrating, 7-10
  - repository projects and, 2-8
  - testing, 2-19
  - troubleshooting, 6-4 to 6-7
- Web services policy-based security, 7-1 to 7-7
- WSDL (Web Service Definition Language), 2-15 to 2-16
- WSDL file location dialog box, 7-8

## **X**

---

- XML messages, 1-1
- XML schemas, 1-2, 2-1, 2-15
  - creating, 2-15