# Contents

**Preface** ............................................................................................................................................................................ ix  
  Audience ........................................................................................................................................................................ ix  
  Documentation Accessibility ............................................................................................................................................... ix  
  Related Documents .................................................................................................................................................. x  
  Conventions ................................................................................................................................................................. x  
  Help Us to Serve You Better .................................................................................................................................... xiii

1 **Introduction**

  Adapter Features ......................................................................................................................................................... 1-1  
  PeopleSoft Concepts ................................................................................................................................................ 1-2  
    PeopleSoft Component Interface ..................................................................................................................... 1-2  
    PeopleSoft Application Messaging / Integration Broker ............................................................................. 1-2  
  Integration with PeopleSoft ................................................................................................................................. 1-2  
  Adapter Architecture ........................................................................................................................................ 1-3  
  BSE Versus OracleAS Adapter J2CA Deployment ............................................................................................ 1-5

2 **Configuring OracleAS Adapter for PeopleSoft**

  Starting Application Explorer ........................................................................................................................... 2-1  
  Configuring Settings for BSE or J2CA .................................................................................................................. 2-2  
    Configuring BSE ............................................................................................................................................... 2-2  
      Configuring BSE System Settings ................................................................................................................ 2-3  
      Configuring J2CA .......................................................................................................................................... 2-6  
        Password Encryption ................................................................................................................................. 2-7  
  Creating a Repository Configuration .................................................................................................................. 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  
  Establishing a Connection (Target) for PeopleSoft ......................................................................................... 2-11  
    Defining a Target to PeopleSoft .................................................................................................................... 2-11  
    Connecting to a Defined PeopleSoft Target ............................................................................................. 2-12  
    Managing a Target .......................................................................................................................................... 2-13  
  Viewing Application System Objects ................................................................................................................. 2-14  
  Creating XML Schemas .................................................................................................................................. 2-15  
  Generating WSDL (J2CA Configurations Only) ............................................................................................... 2-15  
  Creating and Testing a Web Service (BSE Configurations Only) ............................................................... 2-16
This Preface contains these topics:

- **Audience**
- **Documentation Accessibility**
- **Related Documents**
- **Conventions**
- **Help Us to Serve You Better**

### Audience

*Oracle Application Server 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 SOA Suite (BPEL, ESB).

### 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/](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 Application Server Adapter Concepts
- Oracle Application Server 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.

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

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Italics</strong></td>
<td>Italic typeface indicates book titles or emphasis.</td>
<td><em>Oracle Database 10g Concepts</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ensure that the recovery catalog and target database do <em>not</em> reside on the same disk.</td>
</tr>
<tr>
<td>UPPERCASE</td>
<td>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.</td>
<td>You can specify this clause only for a <code>NUMBER</code> column.</td>
</tr>
<tr>
<td>monospace</td>
<td></td>
<td>You can back up the database by using the <code>BACKUP</code> command.</td>
</tr>
<tr>
<td>(fixed-width) font</td>
<td></td>
<td>Query the <code>TABLE_NAME</code> column in the <code>USER_TABLES</code> data dictionary view.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Use the <code>DBMS_STATS.GENERATE_STATS</code> procedure.</td>
</tr>
<tr>
<td><strong>lowercase</strong></td>
<td>Lowercase monospace typeface indicates executable programs, filenames, directory names, and sample user-supplied elements.</td>
<td>Enter <code>sqlplus</code> to start <code>SQL*Plus</code>.</td>
</tr>
<tr>
<td>monospace</td>
<td></td>
<td>The password is specified in the <code>orapwd</code> file.</td>
</tr>
<tr>
<td>(fixed-width) font</td>
<td><em>Note:</em> Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.</td>
<td>Back up the datafiles and control files in the <code>/disk1/oracle/dbs</code> directory.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The <code>department_id</code>, <code>department_name</code>, and <code>location_id</code> columns are in the <code>hr.departments</code> table.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Connect as oe user.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The <code>JRepUtil</code> class implements these methods.</td>
</tr>
<tr>
<td><strong>lowercase</strong></td>
<td>Lowercase italic monospace font represents placeholders or variables.</td>
<td>You can specify the <code>parallel_clause</code>.</td>
</tr>
<tr>
<td>italic</td>
<td></td>
<td>Run <code>old_release.SQL</code> where <code>old_release</code> refers to the release you installed before upgrading.</td>
</tr>
<tr>
<td>monospace</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(fixed-width) font</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Other symbols</strong></td>
<td>You must use symbols other than brackets ([ ]), braces ({}), vertical bars (</td>
<td>), and ellipsis points (...) exactly as shown.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>acct CONSTANT NUMBER(4) := 3;</code></td>
</tr>
</tbody>
</table>
Conventions for Windows Operating Systems

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

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
</table>
| **Italics** | Italicized text indicates placeholders or variables for which you must provide particular values. | CONNECT SYSTEM/system_password  
DB_NAME = database_name |
| **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. **Note:** 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.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click <strong>Start</strong>, and then choose the menu item</td>
<td>How to start a program.</td>
<td>To start the Database Configuration Assistant, click <strong>Start</strong>, and choose Programs. In the Programs menu, choose Oracle - <strong>HOME_NAME</strong> and then click <strong>Configuration and Migration Tools</strong>. Choose <strong>Database Configuration Assistant</strong>.</td>
</tr>
<tr>
<td>File and directory names</td>
<td>File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (&lt;), right angle bracket (&gt;), colon (:), double quotation marks (&quot;), slash (/), pipe (</td>
<td>), 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.</td>
</tr>
<tr>
<td>C:&gt;</td>
<td>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 command prompt in this manual.</td>
<td>C:\oracle\oradata&gt;</td>
</tr>
</tbody>
</table>
| 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" |
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.

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOME_NAME</td>
<td>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.</td>
<td>C:&gt; net start OracleHOME_NAME\TNSListener</td>
</tr>
<tr>
<td>ORACLE_HOME</td>
<td>In Oracle8i release 8.1.3 and lower, when you installed Oracle components, all subdirectories were located under a top level ORACLE_HOME directory. This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level ORACLE_HOME directory. There is a top level directory called ORACLE_BASE that by default is C: \oracle\product\10.1.0. 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 C: \oracle\product\10.1.0\db_n, where n is the latest Oracle home number. The Oracle home directory is located directly under ORACLE_BASE. All directory path examples in this guide follow OFA conventions. Refer to Oracle Database Installation Guide for Windows for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.</td>
<td>Change to the ORACLE_BASE\ORACLE_HOME\rdbms\admin directory.</td>
</tr>
</tbody>
</table>

---

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

<table>
<thead>
<tr>
<th>Component</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adapter</td>
<td></td>
</tr>
<tr>
<td>EIS (DBMS/APP)</td>
<td></td>
</tr>
<tr>
<td>HOTFIX/Service Pack</td>
<td></td>
</tr>
</tbody>
</table>

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

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

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
Oracle Application Server connects to a PeopleSoft system through Oracle Application Server Adapter for PeopleSoft (OracleAS Adapter for PeopleSoft). OracleAS 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 OracleAS Adapter J2CA Deployment

### Adapter Features

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

OracleAS 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.

OracleAS Adapter for PeopleSoft provides:

- Synchronous and asynchronous, bidirectional message interactions for PeopleSoft component interfaces and PeopleSoft Application Messaging / Integration Broker.
PeopleSoft Concepts

Oracle Application 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. OracleAS 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 OracleAS 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. OracleAS 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. OracleAS Adapter for PeopleSoft provides a handler that must be configured within the PeopleSoft application gateway using TCP/IP transport services.

Integration with PeopleSoft

OracleAS 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 OracleAS 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**

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

- Oracle Application 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 Application 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.
Oracle Application Server Adapter J2CA Architecture

Figure 1–2 shows the generic architecture for OracleAS Adapter J2CA for packaged applications. OracleAS 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 OC4J deployment descriptor oc4j-ra.xml. See Chapter 3, "OC4J Deployment and Integration" for more information on OC4J deployment.

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.

Note: Do not use a file repository for BSE in production environments.
BSE Versus OracleAS Adapter J2CA Deployment

If you are using OracleAS Adapter for PeopleSoft with BPEL Process Manager, please note that:

- Only OracleAS Adapter J2CA deployment supports inbound integration (event notification) with BPEL Process Manager.
- Both OracleAS Adapter J2CA and BSE deployments support outbound integration (request-response service) with BPEL Process Manager.

The following three factors explain the differences between deploying the BSE and OracleAS 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 Application 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. **OracleAS Adapter J2CA provides slightly better performance.**

OracleAS Adapter J2CA does provide slightly better performance than BSE. However, the difference decreases as the transaction rate increases.
3. Oracle AS 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 Oracle AS Adapter J2CA, user name and password can be passed using the connection specification of the CCI.
OracleAS 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 WSDL (J2CA Configurations Only)
- 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, select Start, Programs, OracleAS_home Adapters, Application Explorer.

   On Windows, iaexplorer.bat is found under OracleAS_home\adapters\application\tools, where OracleAS_home is the directory where Oracle Application Server is installed.

   On UNIX, load the iwae script, iwae.sh, found under OracleAS_home/adapters/application/tools, where OracleAS_home is the directory where Oracle Application 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 OracleAS Adapter J2CA because the ra.xml file is configured automatically during installation.

Configuring BSE

After BSE is deployed to Oracle Application 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 Application Server and port is the HTTP port for Oracle Application 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: iway
   - Password: iway

   The BSE configuration page is displayed.
3. Ensure that the Adapter Lib Directory parameter specifies the path to the lib directory, for example:

```
OracleAS_home\adapters\application\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 2-6 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.
2. Configure the system settings.

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language</td>
<td>Specify the required language.</td>
</tr>
<tr>
<td>Adapter Lib Directory</td>
<td>Enter the full path to the directory where the adapter jar files reside.</td>
</tr>
<tr>
<td>Encoding</td>
<td>Only UTF-8 is supported.</td>
</tr>
<tr>
<td>Debug Level</td>
<td>Specify the debug level from one of the following options:</td>
</tr>
<tr>
<td></td>
<td>■ None</td>
</tr>
<tr>
<td></td>
<td>■ Fatal</td>
</tr>
<tr>
<td></td>
<td>■ Error</td>
</tr>
<tr>
<td></td>
<td>■ Warning</td>
</tr>
<tr>
<td></td>
<td>■ Info</td>
</tr>
<tr>
<td></td>
<td>■ Debug</td>
</tr>
<tr>
<td>Number of Async. Processors</td>
<td>Select the number of asynchronous processors.</td>
</tr>
</tbody>
</table>

3. Configure the security settings.

<table>
<thead>
<tr>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admin User</td>
</tr>
<tr>
<td>Admin Password</td>
</tr>
<tr>
<td>Policy</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admin User</td>
<td>Provide a BSE administrator ID.</td>
</tr>
<tr>
<td>Admin Password</td>
<td>Enter the password associated with the BSE administrator ID.</td>
</tr>
<tr>
<td>Policy</td>
<td>Select the check box to enable policy security.</td>
</tr>
</tbody>
</table>
4. 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 2-6 for more information.

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

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repository Type</td>
<td>Select one of the following repositories from the list:</td>
</tr>
<tr>
<td></td>
<td>■ Oracle</td>
</tr>
<tr>
<td></td>
<td>■ File (Do not use a file repository for BSE in production environments.)</td>
</tr>
<tr>
<td>Repository URL</td>
<td>Enter the URL to use when opening a connection to the database. For example,</td>
</tr>
<tr>
<td></td>
<td>the following repository URL format is used when connecting to Oracle:</td>
</tr>
<tr>
<td></td>
<td>jdbc:oracle:thin:@host name:port;SID</td>
</tr>
<tr>
<td>Repository Driver</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>oracle.jdbc.driver.OracleDriver</td>
</tr>
<tr>
<td>Repository User</td>
<td>Enter a valid user ID to use when opening a connection to the database.</td>
</tr>
<tr>
<td>Repository Password</td>
<td>Enter a valid password that is associated with the user ID.</td>
</tr>
<tr>
<td>Repository Pooling</td>
<td>If selected, repository pooling will be used. This option is disabled by default.</td>
</tr>
</tbody>
</table>

5. 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.
The default location for the repository on Windows is:

OracleAS_home\j2ee\OC4J_CONTAINER\applications\ws-app-adapter\ibs\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:

   OracleAS_home\adapters\application\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 Application 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 OracleAS Adapter for PeopleSoft, OC4J generates a deployment descriptor called oc4j-ra.xml. This descriptor provides OC4J-specific deployment information for resource adapters. See Chapter 3, "OC4J 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 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 "jca_sample":

```
OracleAS_HOME\adapters\application\config\jca_sample
```

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 `oc4j-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 `oc4j-ra.xml` file.

9. Copy the Oracle database JDBC drivers (Classes12.jar, Classes12.zip, and nls_charset12.jar) to the following directory:

```
OracleAS_home\adapters\application\lib
```

Where `OracleAS_home` is the directory where Oracle Application Server is installed.

**Password Encryption**

When creating J2CA configurations, you can also encrypt a password using Application Explorer and use this value in the `jcatransport.properties` and `oc4j-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
```

Configuring OracleAS Adapter for PeopleSoft  2-7
Creating a Repository Configuration

Before you use Application Explorer with OracleAS 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 your application server resides and *port* is the HTTP port number where the application 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 OracleAS 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. In the **Home** field, enter a path to your J2CA configuration directory where the repository, schemas, and other information is stored, for example:

   `OracleAS_home\adapters\application`

5. Click **OK**.

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

---

The OracleAS Adapter J2CA configuration file is stored in `OracleAS_home\adapters\application\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 Application 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`:

   ![Configurations Tree](image)

   - 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 OracleAS 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. OracleAS 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 under 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 Application Server to update the repository for run time purposes.

To delete a connection to PeopleSoft:

1. Locate the target you want to delete.
2. Right-click the target (for example, PsoftTarget), and select Delete.

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, PSoftTarget. 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 OraclesAS Adapter J2CA

To create XML request and response schemas for a PeopleSoft Component Interface against an OraclesAS 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 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:
Creating and Testing a Web Service (BSE Configurations Only)

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.

   **Note:** You can organize your WSDL files in subfolders, creating your own WSDL hierarchy structure. Create the folders under OracleAS_home\adapters\application\wsdls\ The WSIL browser in JDeveloper will display the full tree structure of your WSDL hierarchy.

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.

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 OracleAS Adapter for 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.

   ![Create Web Service dialog box]

   The Create Web Service dialog box is displayed.

   ![Create Web Service dialog box]

   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.

   ![Create Web Service dialog box]

   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 the right pane. If you are testing a Web service that requires XML input, an input field appears.
5. Enter the appropriate input.
6. Click Invoke.

   Application Explorer 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:

```xml
<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.
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.

A channel represents configured connections to particular instances of back-end systems. See "Creating and Editing a Channel" on page 2-19 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 Application Server to recognize the change and update the repository for run time purposes.

Creating a Channel Using Specific Protocols

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

- HTTP
- TCP
- File

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

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Listener port</td>
<td>Port on which to listen for PeopleSoft event data.</td>
</tr>
<tr>
<td>Https</td>
<td>For a secure HTTP connection, select the Https check box. This option is currently not supported.</td>
</tr>
</tbody>
</table>
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.

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.

---

**Parameter** | **Description**
--- | ---
Synchronization Type | Choose from the following synchronization options:
  - REQUEST_RESPONSE
  - REQUEST_ACK

**Important:** The PeopleSoft channel will not work if the synchronization type is set to REQUEST.

**Note:** If you are using OracleAS Adapter for PeopleSoft with BPEL Process Manager, do not start the channel, as it is managed by the BPEL PM Server. If you start the channel for testing and debugging purposes, stop it before run-time.
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 OracleAS 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 preparser is required.

f. Is Keep Alive check box: Maintains continuous communication between the event transaction and the channel.

5. Click OK.

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

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 OracleAS Adapter for PeopleSoft with BPEL Process Manager, do not start the channel, as it is managed by the BPEL PM Server. If you start the channel for testing and debugging purposes, stop it before run-time.

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polling Location</td>
<td>Target file system location for the PeopleSoft XML file.</td>
</tr>
<tr>
<td>File Mask</td>
<td>File name to be used for the output file generated by the operation.</td>
</tr>
</tbody>
</table>

**Parameter** | **Description**
--- | ---
Synchronization Type | Choose from the following synchronization options:
- REQUEST_RESPONSE
- REQUEST_ACK
**Important:** The PeopleSoft channel will not work if the synchronization type is set to REQUEST.

Response/Ack Directory | Target file system location for the PeopleSoft XML file.

**Parameter** | **Description**
--- | ---
Poll interval (msec) | Interval (in milliseconds) when to check for new input. The default is three seconds. Optional.

**Processing Mode** | Sequential indicates single processing of requests.
- Threaded 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 OracleAS Adapter for PeopleSoft with BPEL Process Manager, do not start the channel, as it is managed by the BPEL PM Server. If you start the channel for testing and debugging purposes, stop it before run-time.

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.
   A confirmation dialog box is displayed.
3. To delete the channel you selected, click OK.
   The channel disappears from the list in the left pane.
This chapter describes Oracle Containers for J2EE (OC4J) deployment and integration with OracleAS Adapter for PeopleSoft.

This chapter discusses the following topics:

- Adapter Integration with OC4J
- Deployment of Adapter
- Updating Adapter Configuration
- How to Write a Java Application Client Using the CCI API

See Also:

- Oracle Application Server Adapter Concepts

Adapter Integration with OC4J

OracleAS Adapter for PeopleSoft is deployed within an OC4J container during installation. All client applications run within the OC4J environment. In J2CA deployment, the Common Client Interface (CCI) integrates an OC4J client application with a resource adapter.

See Also:

- Oracle Application Server Adapters Integration with OC4J" in Oracle Application Server Adapter Concepts

Deployment of Adapter

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

During the J2CA deployment of OracleAS Adapter for PeopleSoft, OC4J generates a deployment descriptor called `oc4j-ra.xml`, located in `OC4J_home\j2ee\home\application-deployments\default\jca_app_adapter`.

See Also: Oracle Application Server Adapter Concepts

Creating a Managed Connector Factory Object

The `oc4j-ra.xml` descriptor provides OC4J-specific deployment information for resource adapters. For example, the default `jca_sample` configuration in Application Explorer is represented in the `oc4j-ra.xml` file as follows:

```xml
<?xml version="1.0"?>
<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>
```

Note: Your installation contains more than one file named `oc4j-ra.xml`. The OC4J deployment descriptor described in this section is located in the directory specified earlier.
The parameters defined in the `oc4j-ra.xml` file are described in the following table:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IWayHome</td>
<td>The base installation directory for the OracleAS packaged application adapter.</td>
</tr>
<tr>
<td>IWayConfig</td>
<td>The adapter configuration name as defined in Application Explorer. For example, OracleAS Adapter for PeopleSoft has a preconfigured <code>jca_sample</code> configuration in Application Explorer.</td>
</tr>
<tr>
<td>IWayRepoURL</td>
<td>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 &quot;Configuring BSE System Settings&quot; on page 2-3 for more information.</td>
</tr>
<tr>
<td>IWayRepoUser</td>
<td>User name to use when connecting to the database. This is necessary only when using an Oracle database as the BSE repository. See &quot;Configuring BSE System Settings&quot; on page 2-3 for more information.</td>
</tr>
<tr>
<td>IWayRepoPassword</td>
<td>Password. If provided, it overwrites configuration. This is necessary only when using an Oracle database as the BSE repository. See &quot;Configuring BSE System Settings&quot; on page 2-3 for more information.</td>
</tr>
<tr>
<td>loglevel</td>
<td>It overwrites the level set by the ManagedConnectionFactory property.</td>
</tr>
</tbody>
</table>

Creating Multiple Managed Connector Factory Objects

To establish multiple managed connector factory objects, you must edit the `oc4j-ra.xml` file and add more `<connector-factory>` nodes. For example, the default `jca_sample` configuration in Application Explorer is represented in the `oc4j-ra.xml` file as follows:

```xml
<?xml version="1.0"?>
<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
<?xml version="1.0"?>
<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 OracleAS Adapter for PeopleSoft does not support invocation through the CCI API. Only invocation using Oracle BPEL Process Manager and ESB 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.

```java
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.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adapterName</td>
<td>Name of the packaged application adapter.</td>
</tr>
<tr>
<td>config</td>
<td>Adapter configuration name. NOT REQUIRED FOR IWAEAdapter.</td>
</tr>
<tr>
<td>language</td>
<td>Default is en.</td>
</tr>
</tbody>
</table>

---

3-4 Oracle Application Server Adapter for PeopleSoft User's Guide
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,

```java
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 IAEAdapter 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:

```java
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 IWAFJCASimple {

    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
    // ---------------------------------------------------------
    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

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

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

    // 4.1 Using JCA standard Indexed Record
    // Use JCA IndexedRecord, 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
    //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());

} // main()
Overview of Adapter Integration with Oracle BPEL Process Manager

To integrate with Oracle BPEL Process Manager, OracleAS Adapter for PeopleSoft must be deployed in the same OC4J container as Oracle BPEL Process Manager. The underlying adapter services must be exposed as WSDL files, which are generated during design time in Oracle Application Server Adapter Application Explorer (Application Explorer) for both request-response (outbound) and event notification (inbound) services of the adapter. See "Generating WSDL (J2CA Configurations Only)" on page 2-15 for more information.

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

When using the adapter with Oracle BPEL Process Manager, the BPEL PM home directory is OC4J_BPEL, and is located as follows:

```
OracleAS_home\j2ee\OC4J_SOA
```
Deployment of Adapter

During installation, OracleAS Adapter for PeopleSoft is deployed as a J2CA 1.0 resource adapter within the OC4J J2CA container. The adapter must be deployed in the same OC4J container as Oracle BPEL Process Manager.

See Also: Oracle Application Server Adapter Concepts

Design Time

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

- OracleAS Adapter Application Explorer (Application Explorer)
- Oracle JDeveloper BPEL Designer (JDeveloper) or Eclipse

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

Before you design a BPEL process, you must create a schema and generate the respective WSDL file using Application Explorer. See "Generating WSDL (J2CA Configurations Only)" on page 2-15 for more information.

Namespace Requirements

The purpose of an XML namespace is to allow the deployment of XML vocabularies (where element and attribute names are defined) in a global environment and to reduce the risk of name collisions in a given document when vocabularies are combined. Qualified namespaces are used for stricter schema validation. In documents conforming to this specification, element and attribute names appear as qualified names. Syntactically, they are either prefixed names or unprefixed names. An attribute-based declaration syntax is provided to bind prefixes to namespace names and to bind a default namespace that applies to unprefixed element names. These declarations are scoped by the elements on which they appear so that different bindings may apply in different parts of a document. Processors conforming to this specification must recognize and act on these declarations and prefixes.

In the 10.1.3.1.0 SOA release, the recommendations for BPEL integrations is to perform stricter name space validations. As a result, Application Explorer generates Web services for the back-end with the namespace marked as “Qualified”. During testing or usage phases of this service by BPEL, the request XML document that is used should adhere to the schema and WSDL document. Once again, it is important to remember that the namespaces are qualified. To further understand this point, the difference is illustrated with the following example:

1. Input XML for BPEL based on unqualified namespaces:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<PS8>
  <component perform="browse">LOCATION</component>
  <key name="Setid">SHARE</key>
  <key name="Location">ALBERTA</key>
</PS8>
```
2. Input XML for BPEL based on qualified namespaces:

```xml
<PS8 xmlns="urn:iwaysoftware:adapter:peoplesoft:ci">
  <component perform="find">LOCATION</component>
  <key name="Setid">SHARE</key>
  <key name="Location">ALBERTA</key>
</PS8>
```

Note: If you are passing an unqualified input against a WSDL document that is expecting qualified namespaces, BPEL will throw the exception as “Unable to process input xml....

**Design a BPEL Process for Request-Response Service (Outbound)**

An outbound BPEL process consists of PartnerLink, Invoke, and Assign process activities. You must first create a new Application Server connection, Integration Server connection, and a synchronous BPEL process template.

**Create a New Application Server Connection**

To create a new Application Server connection:

1. Display the connections by clicking the **Connections Navigator** tab at the top of the upper left pane in JDeveloper.

2. Right-click **Application Server** and select **New Application Server Connection**.
   The Create Application Server Connection - Welcome dialog box is displayed.

3. Click **Next**.
   The Create Application Server Connection - Step 1 of 4: Type dialog box is displayed.
4. Specify a unique name and select a connection type for your Application Server connection and click **Next**.

   The Create Application Server Connection - Step 2 of 4: Authentication dialog box is displayed.

5. Specify a valid user name and password for the Application Server you want to connect to.
7. Click **Next**.

   The Create Application Server Connection - Step 3 of 4: Connection dialog box is displayed.
8. Select the Single Instance connection option.
9. Enter localhost as the host name and 6003 for the OPMN port.
10. Enter home as the OC4J instance name.
11. Click Next.

The Create Application Server Connection - Step 4 of 4: Test dialog box is displayed.

12. Click Test Connection.

When the test is complete and the connection is successful, a Success! message appears in the status area.

13. Click Finish.

Your newly created Application Server connection is displayed in the Connections Navigator tab under the Application Server node.
Create a New Integration Server Connection

To create a new Integration Server connection:

1. Display the connections by clicking the Connections Navigator tab at the top of the upper left pane in JDeveloper.

2. Right-click Integration Server and select New Integration Server Connection.
   The Create Integration Server Connection - Welcome dialog box is displayed.

3. Click Next.
   The Create Integration Server Connection - Step 1 of 3: Name dialog box is displayed.
4. Specify a unique name and click Next. The Create Integration Server Connection - Step 2 of 3: Connection dialog box is displayed.

5. Select an Application Server connection, which is already created.

6. Enter localhost as the host name and 8888 for the port number.

7. Select Add host name to the list of proxy exceptions and click Next. The Create Integration Server Connection - Step 3 of 3: Test Connection dialog box is displayed.
8. Click Test Connection.

When the test is complete and the connection is successful, a Success! message appears in the status area.

9. Click Finish.

Your newly created Integration Server connection is displayed in the Connections Navigator tab under the Integration Server node.

**Testing Outbound BPEL and ESB Processes**

The BPEL console enables the testing of deployed BPEL processes. Once a process is deployed, you can manage, monitor, and run an end-to-end scenario using the Initiate tab in the console. The OracleAS Adapter for PeopleSoft is certified for testing using the XML Payload option and the option of running using Through Java Delivery API. It is recommended that developers use this method for testing the OracleAS Adapter for PeopleSoft.

When testing an outbound BPEL process from the BPEL console or an outbound ESB process from the Enterprise Manager console, do not use the XML envelopes that are generated by these consoles. Instead, remove them and use the XML payloads that are
generated from the schemas, which conform to the WSDLs for namespace qualifications.

The ESB data flows can be tested using the Enterprise Manager console. When creating an ESB data flow and interactions, the Web services are created and registered with the Oracle Application Server. For more information on creating an ESB outbound process, see Chapter 5, "ESB Integration Examples".

PeopleSoft Service Integration

This topic illustrates PeopleSoft service integration. Design-time and run-time configuration is described.

Design-Time Configuration

Before you design a process for PeopleSoft service integration, you must generate its respective WSDL file using Application Explorer.

Generating WSDL for a Request-Response J2CA Service

To generate WSDL for outbound interaction in Application Explorer:

1. Start Application Explorer and connect to a defined PeopleSoft target or create a new target.
   
   See "Defining a Target to PeopleSoft" on page 2-11 for more information.

2. Expand the PeopleSoft target to which you are connected.

3. Expand Component Interfaces and select LOCATION.

4. Right-click LOCATION.
   
   The following menu is displayed:

5. Click Create Outbound JCA Service (Request/Response).
   
   The Export WSDL dialog box is displayed.

6. Click OK.
   
   You are ready to design a BPEL process in JDeveloper.
Creating a BPEL PM Server Connection
Before you design an outbound BPEL process, you must configure a new Application Server and Integration Server connection in Oracle JDeveloper. For more information, see Chapter 4, "Integration with Oracle BPEL Process Manager".

Creating a BPEL Project for a Synchronous BPEL Process
To create a BPEL Project for a synchronous BPEL process:

1. At the top of the upper left pane, click the Applications Navigator tab and select an application. If an application does not exist, then you must create an application.

2. Right-click the application and select New Project.
   The New Gallery window is displayed.

3. From the Items list, select BPEL Process Project and click OK.
   The BPEL Process Project dialog box is displayed.
4. Perform the following steps:
   a. Specify a name for the BPEL process.
      The Namespace field is updated automatically.
   b. From the Template list, select Synchronous BPEL Process.
5. Click OK.

Designing the BPEL Process for the LOCATION Outbound Service
To design the BPEL Process:
1. From the Services pane on the right, drag and drop a PartnerLink to the visual editor.
   The Create Partner Link dialog box is displayed.
2. Click the **Service Explorer** icon (second icon from the left preceding the **WSDL File** field).

   The Service Explorer dialog box is displayed.

   ![Service Explorer dialog box]

3. Expand your new connection under Adapter Services, followed by **adapters**, and then **applications**.

   The WSDL tree displayed in the Service Explorer dialog box lists any WSDL files you have created using Application Explorer. The WSDL tree is generated by a WSDL servlet, which is automatically deployed as part of the BPEL Server installation.

4. Select **LOCATION_invoke.wsdl** and click OK.

   The **WSDL File** field in the Create Partner Link dialog box displays the name and location of the selected WSDL file. The **Partner Link Type** field specifies the PartnerLink defined in the WSDL file.
Perform the following steps:

**a.** Leave the **My Role** field unspecified. The role of the PartnerLink is null, as it will be synchronously invoked from the BPEL process.

**b.** From the **Partner Role** list, select the default value **LOCATIONRole**. This is the role of the BPEL process.

5. Click **OK**.
   The new PartnerLink appears in the visual editor.

6. Select Save from the File menu.

7. From the **Process Activities** pane on the right, drag an **Invoke** activity to the visual editor and place it between the Receive activity (receiveInput) and the Reply activity (replyOutput).
   The Invoke process activity is shown in the following diagram view.
8. Drag the **blue arrow** from Invoke_1 and connect it to the PeopleSoft PartnerLink. The Edit Invoke dialog box is displayed.

Perform the following steps:

a. Click the first icon to the right of the **Input Variable** field, then click **OK** in the Create Variable window that is displayed.

b. Repeat the previous step to create a default variable for **Output Variable**.

9. Click **OK**.
10. Drag an Assign process activity and drop it between receiveInput and Invoke_1.

The following image shows the new Assign activity in JDeveloper visual editor.

![Diagram showing new Assign activity]

11. Double-click the Assign activity icon.

The Assign dialog box is displayed.

![Assign dialog box]

12. In the Copy Operation tab, click Create and select Copy Operation.

The Create Copy Operation dialog box is displayed. Perform the following steps:
a. In the From pane, expand Variables, then inputVariable, and then highlight payload.

b. In the To pane, expand Variables, then Invoke_1LOCATION_InputVariable, and then highlight input_LOCATION.

Your Create Copy Operation dialog box should look as follows:

13. To close the Create Copy Operation dialog box and the Assign dialog box, click OK.

14. From the Process Activities pane on the right, drag another Assign activity to the visual editor and place it between the Invoke activity (Invoke_1) and the Reply activity (replyOutput).

15. Double-click the Assign activity icon and click Create.

16. In the Create Copy Operation dialog box, map Invoke_1LOCATION_OutputVariable, output_LOCATION to outputVariable, payload.

Verify that you have mapped all variables as follows:

17. Click OK, then click OK again.

18. Select Save from the File menu.

You have completed the design of your BPEL process.
Deploying the BPEL Process for the LOCATION Outbound Service

JDeveloper deploys the outbound BPEL process directly to Oracle BPEL Console.

To deploy your BPEL process in JDeveloper:

1. Right-click your project in the Applications Navigator tab.
2. Select Deploy, then Your BPEL PM Server connection, and then Deploy to default domain.
   
   The deployment process starts automatically.
3. Observe the Messages log at the bottom of the window.
   
   The Messages log displays the deployment status. In this example, it shows a successful deployment message for the process.

If deployment was not successful, click the Compiler tab to view all error and warning messages generated during the deployment process.

Invoking Adapter Request-Response Service from Oracle BPEL Process Manager

The OracleAS Adapter for PeopleSoft request-response service is used to create, delete, update, query back-end data, and to call back-end workflows and transactions. The following section describes how to invoke the adapter synchronous request-response service, also referred to as Outbound Interaction, and how to manage the process in Oracle BPEL Console.

Manage the Deployed Outbound Process in Oracle BPEL Console

JDeveloper deploys the developed process directly to the Oracle BPEL Console, which enables you to run, monitor, and administer BPEL processes.

To invoke adapter request-response service:

1. Start the Oracle BPEL Console by entering the following URL in a browser:
   
   http://host:port/BPELConsole
2. Select a domain and provide a valid password.
   
   The Oracle BPEL Console main page is displayed. All deployed BPEL processes are listed in the Dashboard tab.
3. Click the BPEL Processes tab.

   This tab provides a more detailed view of each deployed process.

4. Click the PeopleSoft process link, PSFT_outbound_location_invoke (v. 1.0).

   The Manage window provides options for managing this BPEL process. Do not change any of the following default settings.

5. Click the Initiate tab.

   The Initiate tab enables you to test your BPEL process.
Perform the following steps:

a. From the **Initiating a test instance** list, select **XML Source**.

b. Select the Java through delivery API link in the right side of the test area.

c. Enter the following code in the text area provided for XML input:

   ```xml
   <?xml version="1.0" encoding="UTF-8"?>
   <PS8 SERVICENAME="LOCATION" METHODNAME="LOCATION" LICENSE="test">
   <component perform="browse">LOCATION</component>
   <key name="SETID">SHARE</key>
   <key name="LOCATION">ALBERTA</key>
   </PS8>
   ```

6. Click **Post XML Message**.

   The response received from the PeopleSoft system is displayed in the Initiate window.

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

---

**PeopleSoft Event Integration**

This topic illustrates PeopleSoft event integration. Design-time and run-time configuration is described.

**Design-Time Configuration**

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

---

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

**Creating a Channel**

To create a channel:

1. Click the **Events** node.
2. Expand the **PeopleSoft** node.
   The ports and channels nodes appear in the left pane.

   ![Diagram showing expanded PeopleSoft node]

3. Right-click Channels and select Add Channel.
   The Add Channel dialog box is displayed.

   ![Add Channel dialog box]

   Provide the following information:
   
   a. Enter a name for the channel, for example, **PSFT_Channel**.
   
   b. Enter a brief description (optional).
   
   c. From the Protocol list, select **HTTP Listener**.

4. Click Next.

5. When the Basic dialog box is displayed, enter the system information as specified in the following table:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Listener port</td>
<td>Port on which to listen for PeopleSoft event data.</td>
</tr>
<tr>
<td>Httsp</td>
<td>For a secure HTTP connection, select the <strong>Httsp</strong> check box.</td>
</tr>
<tr>
<td>Synchronization Type</td>
<td>Choose from the following synchronization options:</td>
</tr>
<tr>
<td></td>
<td>■ REQUEST_RESPONSE</td>
</tr>
<tr>
<td></td>
<td>■ REQUEST_ACK</td>
</tr>
</tbody>
</table>

   **Important:** The PeopleSoft channel will not work if the synchronization type is set to REQUEST.

6. 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.

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

---

**Generating WSDL for Event Notification**

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

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

   ![Diagram](image.png)

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

7. Expand Messages and select LOCATION_SYNC.VERSION_1.
8. Right-click LOCATION_SYNC.VERSION_1.

   ![Diagram](image.png)

9. Select Create Inbound JCA Service (Event).
   
   The Export WSDL dialog box is displayed.
Perform the following steps:

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

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

c. From the Channel drop-down list, select the channel you created for this inbound service.

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

10. Click OK.

**Creating a BPEL PM Server Connection**

Before you design a BPEL process using the WSDL you generated in Application Explorer, you must configure a new Application Server and Integration Server connection in Oracle JDeveloper. For more information, see Chapter 4, "Integration with Oracle BPEL Process Manager".

**Designing the BPEL Process for the LOCATION_SYNC.VERSION_1 Inbound Service**

To design a BPEL process for inbound interaction:

1. At the top of the upper left pane, click the Applications Navigator tab and select an application.
2. Right-click the application and select **New Project**.
   The New Gallery dialog box is displayed.

3. From the Items list, select **BPEL Process Project** and click **OK**.
   The BPEL Process Project dialog box is displayed.
4. Perform the following steps:
   a. Specify a name for the project, for example, PSoftTLOCATION_inbound. The Namespace field is updated automatically.
   b. From the Template list, select Empty BPEL Process.
   c. Click OK.

5. From the Services pane on the right, drag and drop a PartnerLink to the visual editor.
   The Create Partner Link dialog box is displayed.

6. Click the Service Explorer icon (second icon from the left preceding the WSDL File field).
   The Service Explorer dialog box is displayed.
7. Expand your new connection, then expand **adapters**, and then **applications**.

The WSDL tree displays the WSDL files you created using Application Explorer. The WSDL tree is generated by a WSDL servlet, which is automatically deployed as part of the BPEL Server installation.

8. Select **LOCATION_SYNC.VERSION_1_receive.wsdl** and click OK. The Create Partner Link dialog box is displayed.
The **WSDL File** field displays the name and location of the selected WSDL file. The **Partner Link Type** field specifies the PartnerLink defined in the WSDL file.

Perform the following steps:

**a.** From the **My Role** list, select the default value `LOCATION_SYNC.VERSION_1Role`.

**b.** Leave the **Partner Role** field unspecified.

9. Click **Apply**, and then **OK**.

The new PSFT_PL1 PartnerLink appears in the visual editor.

10. From the **Process Activities** pane on the right, drag a **Receive** activity to the visual editor and place it in the designated placeholder labeled **Drop Activity Here**.

11. Connect the Receive activity to the PSFT_PL1 PartnerLink.

The Edit Receive dialog box is displayed.
Perform the following steps:

a. Specify a name for the Receive Activity, for example, **Receive_LOCATION**.

b. Click the first icon to the right of the **Variable** field, then click **OK** in the Create Variable dialog box that is displayed.

c. Verify that the **Create Instance** check box is selected.

12. Click **Apply**.

   The Receive dialog box should no longer display any warnings or errors.

13. Click **OK**.

14. Select **Save** from the **File** menu.

### Deploying the BPEL Process for the LOCATION_SYNC.VERSION_1 Inbound Service

1. Right-click your project in the Applications Navigator tab.

2. Select **Deploy**, then **Your BPEL PM Server connection**, and then **Deploy to default domain**.

   The deployment process starts automatically.

### Runtime Configuration

The following topics describe how to trigger an event in PeopleSoft and verify event integration using OracleAS Adapter for PeopleSoft.

**Triggering an Event in PeopleSoft to Test Event Runtime Integration**

To trigger an event in PeopleSoft:

1. Log in to PeopleSoft, select PeopleTools, Integration Broker, and then Node Definitions.
2. Enter the node name you will be using, for example, EXTERNAL; then click Search.

3. Click the Connectors tab.

4. Enter HTTPTARGET in the Connector ID field; then enter the listener URL and its port in the PRIMARYURL field.

5. To save the configuration, click Save.
6. From Menu in the left pane, select Set Up Financials/Supply Chain, Common Definitions, Location, and then Location.

7. To find the location record you want to update, click Search; then make the changes and click Save.

When the change is saved, it triggers an event and sends the event to the listener.

**Verifying the Results**

To verify your results:
1. Log in to Oracle BPEL Console at
   
   http://host:port/BPELConsole

2. Provide a valid user name and password.

3. Click the **Instances** tab.

   Recently received run-time events are displayed in the Instances tab.

4. Click an instance, then click **Audit** to see the received event message.
This chapter contains the following examples:

- Configuring an ESB Outbound Process
- Configuring an ESB Inbound Process

The scenarios shown in this chapter require the following prerequisites.

Prerequisites

The following are installation and configuration requirements:

- OracleAS Adapter for PeopleSoft must be installed on Oracle Application Server.
- PeopleSoft must be configured for inbound and outbound processing.
- OracleAS Technology adapters must be deployed and properly configured.

See Also: Oracle Application Server Adapter Installation Guide

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

- How to configure OracleAS Adapter for PeopleSoft for services and events. For more information, see Chapter 2, "Configuring OracleAS Adapter for PeopleSoft".
- How to configure a new Application Server and Integration Server connection in Oracle JDeveloper. For more information, see Chapter 4, "Integration with Oracle BPEL Process Manager".

Overview of InterConnect Integration

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

Adapter integration with OracleAS ESB is a two-step process:

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

### Namespace Requirements

The purpose of an XML namespace is to allow the deployment of XML vocabularies (where element and attribute names are defined) in a global environment and to reduce the risk of name collisions in a given document when vocabularies are combined. Qualified namespaces are used for stricter schema validation. In documents conforming to this specification, element and attribute names appear as qualified names. Syntactically, they are either prefixed names or unprefixed names. An attribute-based declaration syntax is provided to bind prefixes to namespace names and to bind a default namespace that applies to unprefixed element names. These declarations are scoped by the elements on which they appear so that different bindings may apply in different parts of a document. Processors conforming to this specification must recognize and act on these declarations and prefixes.

In the 10.1.3.1.0 SOA release, the recommendations for ESB integrations is to perform stricter namespace validations. As a result, Application Explorer generates Web services for the back-end with the namespace marked as “Qualified”. During testing or usage phases of this service by ESB, the request XML document that is used should adhere to the schema and WSDL document. Once again, it is important to remember that the namespaces are qualified. To further understand this point, the difference is illustrated with the following example:

1. Input XML for ESB based on unqualified namespaces:
   ```xml
   <PS8>
   <component perform="browse">LOCATION</component>
   <key name="Setid">SHARE</key>
   <key name="Location">ALBERTA</key>
   </PS8>
   ```

2. Input XML for ESB based on qualified namespaces:
   ```xml
   <PS8 xmlns="urn:iwaysoftware:adapter:peoplesoft:ci">
   <component perform="find">LOCATION</component>
   <key name="Setid">SHARE</key>
   <key name="Location">ALBERTA</key>
   </PS8>
   ```

   Note: If you are passing an unqualified input against a WSDL document that is expecting qualified namespaces, ESB will throw the exception as “Unable to process input xml....”

### Configuring an ESB Outbound Process

The following example describes how to configure an ESB outbound process to your PeopleSoft system, using an ESB project in Oracle JDeveloper.

**Prerequisites**

Before you proceed, you must create an outbound WSDL file for the adapter by using the following steps:

1. Create a target using Application Explorer.
2. Connect to the target.
3. Create a WSDL file.
4. Restart the Oracle Application Server.
Creating an Outbound ESB Project and Assigning an Outbound WSDL File

1. At the top of the upper left pane, click the Applications Navigator tab.

2. Right-click an application node that you created and select New Project.
   The New Gallery window is displayed.

3. From the Items list, select ESB Project and click OK.
   The Create ESB Project dialog box is displayed.
4. Perform the following steps:
   
a. Specify a name for the ESB project.
   The Directory Name field and Diagram Name fields are updated automatically.

b. Click OK.
   The ESB project is added at the top of the upper left pane.

5. Right-click the ESB project in the middle pane, select Create ESB Service followed by Custom Adapter.

   **Note:** Users who want to create an ESB process for a Web service, must create the ESB process using the SOAP Service. Right-click the work area, select Create ESB Service from the context menu, and click SOAP Service.

   The Create Adapter Service dialog box is displayed.
6. Enter a name for the adapter service and click the Service Explorer icon (second icon from the left preceding the WSDL File field).

The Service Explorer dialog box is displayed.

7. Expand your new connection under Adapter Services, followed by adapters, and then applications.

The WSDL tree displayed in the Service Explorer dialog box lists any WSDL files you have created using Application Explorer. The WSDL tree is generated by a WSDL servlet, which is automatically deployed as part of the BPEL Server installation.
8. Select an outbound WSDL file that has been created using Application Explorer and click OK.

The **WSDL File** field in the Create Adapter Service dialog box displays the name and location of the selected WSDL file.

9. Click OK.
The new ESB project appears in the visual editor.

Creating a Read Process Operation Using the File Adapter

1. Right-click the ESB project in the middle pane, select Create Adapter Service followed by File Adapter.

The Create File Adapter Service dialog box is displayed.
2. Enter a name for the File adapter and click the Configure adapter service wsdl icon next to the WSDL File field.

   The Adapter Configuration Wizard - Welcome window is displayed.

3. Click Next.

   The Adapter Configuration Wizard - Step 1 of 6: Service Name window is displayed.

4. Click Next.

   The Adapter Configuration Wizard - Step 2 of 6: Operation window is displayed.
5. Click Read File as the Operation Type and click Next.
   The Adapter Configuration Wizard - Step 3 of 6: File Directories window is displayed.

6. Enter the path of the input directory where you are placing the incoming XML file and click Next.
   The Adapter Configuration Wizard - Step 4 of 6: File Filtering window is displayed.
7. Enter the input file extension, for example *.xml, and click Next.
   The Adapter Configuration Wizard - Step 5 of 6: File Polling window is displayed.

8. Change the Polling Frequency to seconds and click Next.
   The Adapter Configuration Wizard - Step 6 of 6: Messages window is displayed.
9. Click Browse to select the WSDL.
   The Type Chooser window is displayed.

10. Click the Import WSDL File icon on the upper right corner of the dialog box.
    The Import WSDL File dialog box is displayed.
11. Select the WSDL file and click OK. The Imported WSDL Files folder is added.

12. Expand the Imported WSDL Files folder, select an Inline Schema, for example, PS8, and click OK. You are returned to the Adapter Configuration Wizard - Step 6 of 6: Messages window.
13. Click Next.

The Adapter Configuration Wizard - Finish window is displayed.

14. Click Finish.

You are returned to the Create File Adapter Service dialog box.
15. Click OK.

The Read operation with a routing service is added to the ESB outbound project view.
Providing a Routing Service for the Read Operation

1. Double-click the routing service.
   
   The Routing Service window is displayed.

2. Expand the Routing Rules.
3. Click the green plus sign icon, which represents the option to Create a new Routing Rule.
   The Browse Target Service Operation window is displayed.

4. Expand Services in project, Default System, your adapter service node, for example, PeopleSoft_ESB_Outbound, and select the service name, for example, GetDetail.

5. Click OK.
   You are returned to the Routing Rules window.
6. Click the icon next to the <<Transformation Map>> field (Select an existing mapper file or create a new one).

   The Request Transformation Map dialog box is displayed.

7. Select the Create New Mapper File option, specify the file name, and click OK.

   The following mapping window is displayed.
8. Select the WSDL file and map it to the Write operation.
   Once you map the WSDL file, the Auto Map Preferences dialog box is displayed.

9. Click OK.
   The mapping is completed as shown in the following window.
10. Double-click the ESB outbound project file in the left pane, for example, ESB_Outbound.esb.

Notice that the Routing service is now created for the Read operation.
Creating a Write Process Operation Using the File Adapter

1. Right-click the ESB project in the middle pane, select Create Adapter Service followed by File Adapter.

The Create File Adapter Service dialog box is displayed.
2. Enter a name for the File adapter and click the Configure adapter service wsdl icon next to the **WSDL File** field.

The Adapter Configuration Wizard - Welcome window is displayed.

3. Click Next.

   The Adapter Configuration Wizard - Step 1 of 4: Service Name window is displayed.

4. Click Next.

   The Adapter Configuration Wizard - Step 2 of 4: Operation window is displayed.
5. Click Write File as the Operation Type and click Next.
   The Adapter Configuration Wizard - Step 3 of 4: File Configuration window is displayed.

6. Enter the path of the output directory and name of the output file and click Next.
   The Adapter Configuration Wizard - Step 4 of 4: Messages window is displayed.
7. Click Browse to select the WSDL.
   The Type Chooser window is displayed.

8. Expand the Project WSDL Files folder, select an Inline Schema and click OK.
   You are returned to the Adapter Configuration Wizard - Step 4 of 4: Messages window.
9. Click Next.
   The Adapter Configuration Wizard - Finish window is displayed.

10. Click Finish.
    You are returned to the Create File Adapter Service dialog box.
11. Click OK.

The Write operation is added to the ESB outbound project view.
Providing a Routing Service for the Write Operation

1. Double-click the routing service.
   
   The Routing Service window is displayed.

2. Expand the Routing Rules.

3. Click the icon next to the <<Target Operation>> field (Browse for target service operations).

   The Browse Target Service Operation window is displayed.
4. Expand Services in project, Default System, your adapter service node, for example, PeopleSoft_ESB_Outbound, and select the service name, for example, Write.

5. Click OK.

You are returned to the Routing Rules window.

6. Click the icon next to the <<Transformation Map>> field (Select an existing mapper file or create a new one).

The Reply Transformation Map dialog box is displayed.

7. Select the Create New Mapper File option, specify the file name, and click OK.

The following mapping window is displayed.
8. Select the WSDL file and map it to the Write operation.

   Once you map the WSDL file, the Auto Map Preferences dialog box is displayed.

   ![Auto Map Preferences dialog box]

9. Click OK.

   The mapping is completed as shown in the following window.

   ![Completed mapping window]
10. Double-click the ESB outbound project file in the left pane, for example, ESB_Outbound.esb.

Notice that the Routing service is now created for the Write operation.
Deploying the Project

1. Right-click the created project, for example, ESB_Outbound, select Register with ESB, and the server connection, for example, ServerConnection1. After successful deployment, the Registration of services Successful message is displayed.

2. Logon to the ESB Control console to check whether the project has been successfully deployed.
The deployed process is listed under the Default System node.

3. Place the XML file in the folder that you specified during the creation of the Read operation.
4. Check whether you are receiving the response in the output folder, which you have specified during the creation of the write operation and also the corresponding instance in the ESB Control console.

5. If the response is not received in the output folder, check the instance and the logs for the corresponding errors in the ESB Control console.
Configuring an ESB Inbound Process

The following example describes how to configure an ESB inbound process to your PeopleSoft system, using an ESB project in Oracle JDeveloper.

Prerequisites
Before you proceed, you must create an inbound WSDL file for the adapter by using the following steps:

1. Create a target using Application Explorer.
2. Create a channel.
3. Create a WSDL file with the noport option.
4. Restart the Oracle Application Server.

Creating an Inbound ESB Project and Assigning an Inbound WSDL File

1. At the top of the upper left pane, click the Applications Navigator tab.

2. Right-click the application node you created and select New Project. The New Gallery window is displayed.

The New Gallery window is displayed.
3. From the Items list, select **ESB Project** and click **OK**.

The Create ESB Project dialog box is displayed.

4. Perform the following steps:
   a. Specify a name for the ESB project.
      The Directory Name field and Diagram Name fields are updated automatically.
   b. Click **OK**.

The ESB project is added at the top of the upper left pane.
5. Right-click the ESB project in the middle pane, select Create ESB Service followed by Custom Adapter.

The Create Adapter Service dialog box is displayed.

6. Enter a name for the adapter service and click the Service Explorer icon (second icon from the left preceding the WSDL File field).

The Service Explorer dialog box is displayed.
7. Expand your new connection under Adapter Services, followed by **adapters**, and then **applications**.

The WSDL tree displayed in the Service Explorer dialog box lists any WSDL files you have created using Application Explorer. The WSDL tree is generated by a WSDL servlet, which is automatically deployed as part of the BPEL Server installation.

8. Select an inbound WSDL file that has been created using Application Explorer and click OK.

The **WSDL File** field in the Create Adapter Service dialog box displays the name and location of the selected WSDL file.
9. Click OK.

The new ESB project appears in the visual editor.

Creating a Write Process Operation Using the File Adapter
1. Right-click the ESB project in the middle pane, select Create Adapter Service followed by File Adapter.

The Create File Adapter Service dialog box is displayed.
2. Enter a name for the File adapter and click the Configure adapter service wsdl icon next to the **WSDL File** field.

The Adapter Configuration Wizard - Welcome window is displayed.

3. Click Next.

The Adapter Configuration Wizard - Step 1 of 4: Service Name window is displayed.

4. Click Next.

The Adapter Configuration Wizard - Step 2 of 4: Operation window is displayed.
5. Click Write File as the Operation Type and click Next.  
The Adapter Configuration Wizard - Step 3 of 4: File Configuration window is displayed.

6. Enter the path of the output directory and name of the output file and click Next.  
The Adapter Configuration Wizard - Step 4 of 4: Messages window is displayed.
7. Click Browse to select the WSDL.
   The Type Chooser window is displayed.

8. Click the Import WSDL File icon on the upper right corner of the dialog box.
   The Import WSDL File dialog box is displayed.
9. Select the WSDL file and click OK.
   The Imported WSDL Files folder is added.

10. Expand the Imported WSDL Files folder, select an Inline Schema, for example, LOCATION_SYNC, and click OK.
    You are returned to the Adapter Configuration Wizard - Step 4 of 4: Messages window.
11. Click Next.

The Adapter Configuration Wizard - Finish window is displayed.

12. Click Finish.

You are returned to the Create File Adapter Service dialog box.
13. Click OK.

The Write operation with a routing service is added to the ESB inbound project view.

Providing a Routing Service for the Write Operation
1. Double-click the routing service.
The Routing Service window is displayed.
2. Expand the Routing Rules.
3. Click the green plus sign icon, which represents the option to Create a new Routing Rule.
   The Browse Target Service Operation window is displayed.

4. Expand Services in project, Default System, your adapter service node, for example, PeopleSoft_ESB_Inbound_RS, and select the service name, for example, Write.

5. Click OK.
   You are returned to the Routing Rules window.
6. Click the icon next to the <<Transformation Map>> field (Select an existing mapper file or create a new one).
   The Request Transformation Map dialog box is displayed.

7. Select the Create New Mapper File option, specify the file name, and click OK.
   The following mapping window is displayed.
8. Select the WSDL file and map it to the Write operation.

Once you map the WSDL file, the Auto Map Preferences dialog box is displayed.

9. Click OK.
The mapping is completed as shown in the following window.

10. Double-click the ESB inbound project file in the left pane, for example, ESB_Inbound.esb.

Notice that the Routing service is now created for the Write operation in the middle pane.

Deploying the Project

1. Right-click the created project, for example, ESB_Outbound, select Register with ESB, and the server connection, for example, ServerConnection1.
After successful deployment, the Registration of services Successful message is displayed.

2. Logon to the ESB Control console to check whether the project has been successfully deployed.
The deployed process is listed under the Default System node.

3. Trigger the event.
4. Check whether you are receiving the response in the output folder, which you have specified during the creation of the write operation.
5. If the response is not received in the output folder, check the instance and the logs for the corresponding errors in the ESB Control console.
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 OracleAS Adapter J2CA or with an OracleAS Adapter Business Services Engine (BSE) configuration.

### Troubleshooting

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

- Application Explorer
- PeopleSoft
- OracleAS Adapter J2CA
- BSE

---

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

- The OracleAS Adapter J2CA trace information can be found under the `OracleAS_home\opmn\logs` directory.

- BSE trace information can be found under the `OracleAS_home\j2ee\home\applications\ws-app-adapter\ibse\ibselogs` directory.

- The log file for Application Explorer can be found under the `OracleAS_home\adapters\application\tools` directory.

---

**Application Explorer**

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

`OracleAS_home\adapters\application\tools`

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

`OracleAS_home/adapters/application/tools`
<table>
<thead>
<tr>
<th>Error</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cannot connect to OracleAS Adapter for PeopleSoft from Application</td>
<td>Ensure that:</td>
</tr>
<tr>
<td>Explorer: The following error message appears: Problem activating</td>
<td>■ PeopleSoft is running.</td>
</tr>
<tr>
<td>adapter</td>
<td>■ The PeopleSoft user ID and password are correct.</td>
</tr>
<tr>
<td></td>
<td>■ The port number is correct.</td>
</tr>
<tr>
<td></td>
<td>■ The custom component interface is properly installed.</td>
</tr>
<tr>
<td>The following error message appears:</td>
<td>You have provided invalid connection information for PeopleSoft or the</td>
</tr>
<tr>
<td>java.lang.IllegalStateException:</td>
<td>wrong psjoa.jar is in the lib directory.</td>
</tr>
<tr>
<td>java.lang.Exception: Error Logon to PeopleSoft System</td>
<td>The psjoa.jar file version is specific to the PeopleTools release.</td>
</tr>
<tr>
<td>PeopleSoft does not appear in the Application Explorer Adapter node</td>
<td>Ensure that the PeopleSoft JAR files, iwpsci84.jar (or iwpsci81.jar) and</td>
</tr>
<tr>
<td>list.</td>
<td>psjoa.jar, are added to the lib directory.</td>
</tr>
<tr>
<td>Logon failure error at run-time.</td>
<td>If the password for connecting to your PeopleSoft system is not specified</td>
</tr>
<tr>
<td></td>
<td>when creating a target or with the Edit option in Application Explorer,</td>
</tr>
<tr>
<td></td>
<td>you will be unable to connect to PeopleSoft. The connection</td>
</tr>
<tr>
<td></td>
<td>password is not saved in repository.xml. Update the password using the</td>
</tr>
<tr>
<td></td>
<td>Edit option in Application Explorer, then restart the application</td>
</tr>
<tr>
<td></td>
<td>server.</td>
</tr>
<tr>
<td>The following error message appears:</td>
<td>The host name or port number for PeopleSoft is incorrect.</td>
</tr>
<tr>
<td>Jolt Session Pool cannot provide a connection to the appserver. This</td>
<td>You are using the wrong iwpsci8x.jar file.</td>
</tr>
<tr>
<td>appears to be because there is no available application server domain.</td>
<td>If the error message “Index: -1, Size:0” appears, or if you can log on</td>
</tr>
<tr>
<td>[Fri Aug 27 13:06:27 EDT 2004] bea.jolt.ServiceException: Invalid</td>
<td>to Application Explorer but you cannot see any Component Interfaces or</td>
</tr>
<tr>
<td>Session</td>
<td>Messages, then you may have both the iwpsci81.jar and iwpsci84.jar</td>
</tr>
<tr>
<td></td>
<td>files in your lib directory. Stop your server, remove the unrequired</td>
</tr>
<tr>
<td></td>
<td>jar file, and restart the server.</td>
</tr>
<tr>
<td>Properties are not displayed for a component interface.</td>
<td>This is a benign exception. It does not affect adapter functionality.</td>
</tr>
<tr>
<td></td>
<td>Download BouncyCastle files from:</td>
</tr>
</tbody>
</table>
### Troubleshooting and Error Messages

<table>
<thead>
<tr>
<th>Error</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unable to start Application Explorer in a Solaris environment. The following exception is thrown in the console:</td>
<td>JAVACMD is not set on the user system. Before starting Application Explorer, export JAVACMD as follows: JAVACMD=/&lt;jdk_home&gt;/bin/java, where &lt;jdk_home&gt; is the directory where JDK is installed on your system.</td>
</tr>
</tbody>
</table>

- javax.resource.ResourceException: IWAFManagedConnectionFactory: License violation.
  - at com.ibi.afjca.spi.IWAFManagedConnectionFactory.createConnectionFactory(IWAFManagedConnectionFactory.java:98)
  - at com.iwaysoftware.iwae.common.JCATransport.getConnectionFactory(JCATransport.java:133)
  - at com.iwaysoftware.iwae.common.JCATransport.initJCA(JCATransport.java:69)
  - at com.iwaysoftware.iwae.common.JCATransport.<init>(JCATransport.java:62)
  - at com.iwaysoftware.iwae.common.AdapterClient.<init>(AdapterClient.java:85)
  - at com.ibi.bse.ConfigWorker.run(ConfigWorker.java:41)
  - at java.lang.Thread.run(Thread.java:534)

  Could not create the connection factory.

<table>
<thead>
<tr>
<th>PeopleSoft</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
</tr>
<tr>
<td>Services are not working properly when using the PeopleSoft Component Interface testing tool in three-tier mode.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>The following error message appears: 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</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Error</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Return error code -1 is received from PeopleSoft at run-time, for example:</td>
</tr>
<tr>
<td>&lt;LOCATIONProcessResponse xmlns=&quot;<a href="http://xmlns.oracle.com/LOCATION">http://xmlns.oracle.com/LOCATION</a>&quot;&gt;</td>
</tr>
<tr>
<td>&lt;/LOCATIONProcessResponse&gt;</td>
</tr>
<tr>
<td>Pstools.properties file has not been initialized.</td>
</tr>
<tr>
<td>The following error message appears:</td>
</tr>
<tr>
<td>Cannot find Component Interface {CI name}</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>The following error message appears:</td>
</tr>
<tr>
<td>Not Authorized (90,6) Failed to execute PSSession request</td>
</tr>
<tr>
<td>The following error message appears:</td>
</tr>
<tr>
<td>Must also provide values for keys {keyname}</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

**OracleAS Adapter J2CA**

<table>
<thead>
<tr>
<th>Error</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>In Application Explorer, the following error message appears when you attempt to connect to an OracleAS Adapter J2CA configuration:</td>
<td>In the Details tab in the right pane, ensure that the directory specified in the Home field points to the correct directory, for example, OracleAS_home\adapters\application</td>
</tr>
</tbody>
</table>
BSE Error Messages

This topic discusses the different types of errors that can occur when processing Web services through Oracle Application 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

---

**BPEL Error Messages**

<table>
<thead>
<tr>
<th>Error</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endpoint activation error on deployment of PeopleSoft event handling project (inbound) in JDeveloper</td>
<td>Verify that the channel used for this inbound J2CA service is stopped in Application Explorer. If you have started this channel for testing or debugging purposes, you must stop it before starting BPEL PM Server. Endpoint activation is managed by BPEL Process Manager. Verify that the specified WSDL file exists at that URL and that the file is valid. Workaround: Change the WSDL location to localhost:7777. The default is 127.0.0.1:7777. Alternative workaround: Add the IP address to the Dhttp.nonProxyHosts list found in obsetenv.bat (Windows) or obsetenv.sh (UNIX).</td>
</tr>
</tbody>
</table>

The following error message appears in BPEL PM Server Console:

Process "TestPSFT" (revision "1.0") compilation failed.

The following exception is thrown in JDeveloper during deployment of the BPEL process:

java.io.FileNotFoundException: \BPELConsole\wsll\adapters\applications\LOCATION_SYNC.V_1_receive.wsdl?wsdl (The system cannot find the path specified) Verify that you have all the required patches installed. The required patches are listed and updated on the Oracle Technology Network Web site.
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:

```xml

<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.

**OracleAS 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.

```xml
<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
  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 OracleAS 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.


Failure to Connect to PeopleSoft
If OracleAS Adapter for PeopleSoft cannot connect to PeopleSoft, then the following SOAP response is generated.


Invalid SOAP Request
If Oracle Application Server 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.


Empty Result From Oracle Application Server 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.
This chapter includes the following topics:

- Web Services Policy-Based Security
- Migrating Repositories

Web Services Policy-Based Security

OracleAS 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 OracleAS 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.
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 OracleAS 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).

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 OracleAS 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.
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.

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.

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 OracleAS 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.
   
<table>
<thead>
<tr>
<th>SOAP</th>
<th>Tools</th>
<th>Window</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create new SOAP request</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Send request to server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Change SOAP request parameters</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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:

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

   ![SOAP menu](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:
OracleAS_home\adapters\application\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 Application Server 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.
3. 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.

4. Because you are creating Java files, ensure you deselect COM Type Library Build.

5. Ensure that Java Classes Build is selected, and then select a directory on your local system where the Java files are to be placed, for example, `c:\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.
Building the PeopleSoft API Java Programs

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.

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.
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%;%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%;%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.

---

**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.
If the JAR file does not use the case-sensitive `PeopleSoft\Generated\CompIntfc\` path, you must go back and correct it.

5. Place the JAR file in the adapters common lib directory, which enables the OracleAS Adapter for PeopleSoft to communicate with the PeopleSoft component interface.

   OracleAS_home\adapters\application\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 OracleAS 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 TCPITARGET84 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 Application 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 Application 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:

<table>
<thead>
<tr>
<th>Property ID</th>
<th>Property Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEADER</td>
<td>sendUncompressed</td>
<td>Y</td>
</tr>
<tr>
<td>HTTPPROPERTY</td>
<td>Method</td>
<td>POST</td>
</tr>
<tr>
<td>PRIMARYURL</td>
<td>URL</td>
<td>URL and the port of the HTTP listener</td>
</tr>
</tbody>
</table>

**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 Application 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.
Configuring the TCP/IP or HTTP Target Connector for PeopleSoft 8.4

8. Click Save. The TCPIP81 Handler Directory window is displayed.

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

10. Click Unload and re-Load TCPIPHandler81.

You can now send messages from PeopleSoft to OracleAS 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 Application Server is up and running.
- The server name and port number for PeopleSoft and Oracle Application 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 Application Server, you receive a Success message.

   An error indicates a configuration problem. For more information, see the Integration Broker error log.
This appendix describes how to create new and modify existing component interfaces for use with OracleAS 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.

[Image]

---

**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.
6. Click + to add a new row to the Component Interfaces list. A field appears where you can enter the component interface name.

7. 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

OracleAS 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 of 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.

   ![Component Interface Chart]

   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=//isdsrv14:9000
ig.isc.userid=VP1
ig.isc.password=VP1
ig.isc.toolsRel=8.40.09

The PeopleSoft tools release must be precise to the last decimal.

**Note:** With release 8.42, the password must be stored in an encrypted format. PeopleSoft provides a script called PSCipher.bat (PSCipher.sh on UNIX) to accomplish encryption. Typically, this script is located in the path of the IntegrationGateway.properties file. Follow the instructions supplied by PeopleSoft to run this script.

You have finished configuring the IntegrationGateway.properties file.

**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.

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):

   \[ \text{host name:port/servlets/psft.pt8.gateway.GatewayServlet} \]

   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.
Configuring Application Messaging in PeopleSoft Release 8.1

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.
5. To load the handler, click Load.
   After the handler loads, "Loaded successfully" appears in the Status column.

6. Click Configure.
   The Simple File Handler Directory window is displayed.

7. Click Add a file handler node.
   The Add File Handler window is displayed.
8. In the Node Name field, enter the name of the message node that you created in "Creating and Configuring a New Message Node" on page D-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.

1. Open Application Designer.

   2. On the Record Fields tab, select the LOCATION_TBL record.

   3. Select the PeopleCode display option.

   4. Select the Save Post Change (SPo) box for the LOCATION field.

   The following window displays the PeopleCode that initiates a LOCATION_SYNC message.
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.
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.

OracleAS Adapter for PeopleSoft can work with PeopleSoft outbound synchronous messages. Outbound synchronous messages involve additional configuration steps, both within PeopleSoft and in Oracle Application 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 Application Server, you must transform the XML that is sent and returned from your final destination.

**Note:** You must use the PeopleSoft-supplied HTTP target connector when you are working with synchronous outbound messages. You cannot use the TCPIP84TARGET connector for outbound synchronous messages.

### 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.
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.
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.

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. For example:
      http://bpelclient:1971
   d. Leave the default values for the remaining properties.

6. Save your changes.
Selecting a Service
To select a service:

2. Search for the DEPT_SYNC service and select it.
   The DEPT_SYNC Services pane is displayed.

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.

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.
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. For example: http://bpelclient:1971

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:

   
   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:


   The Services pane is displayed.

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

4. Click the View Queue link.
   The Queue Definitions pane opens.

5. Ensure that the Queue Status drop-down list has Run selected.
6. Click Return to return to the Service operation pane.
7. Click the Routings tab.

8. Ensure that the routings are active.
9. Click the Return to Service link on the bottom of the pane to return to the Services pane.
10. Click Return to Search to go to the Services search pane.
In the above steps, if the status has been activated in any of the panes for any of the components, then make sure to save the status at the appropriate pane and the respective components.

11. Select PeopleTools, Integration Broker, Configuration, Integration Setup, Services, followed by Nodes.

The Node Definitions pane opens.

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.

![Department Profile Page](image)

4. Enter the appropriate value in the SetID and Department fields

5. Click Add.

   The Department Profile tab opens.

![Department Profile](image)

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).
The Monitor Overview tab is displayed.

![Monitor Overview Tab](image1)

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

3. Click the number link (5) queue (Enterprise_Setup) in the Result section.

   The Operation Instances tab is displayed.

![Operation Instances Tab](image2)

Notice that a new record for the event (department) is available.

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

   A new browser window (Asynchronous Details) is displayed.

![Asynchronous Details](image3)
Notice that the status of the transaction is DONE for the Publishing and Subscriber nodes.
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.
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 2-26, 3-2
  deploying, 1-1, 1-3, 3-1 to 3-2, 4-2
    integrating with BPEL Process Manager, 4-1
  troubleshooting, 6-1 to 6-7
Adapters node, 4-21
Add Channel dialog box, 2-21
Admin Password parameter, 2-3 to 2-4
Admin User parameter, 2-4
Advanced tab, 2-25
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 2-26, 3-2
  deploying, 1-1, 1-3, 3-1 to 3-2, 4-2
    integrating with BPEL Process Manager, 4-1
  troubleshooting, 6-1 to 6-7
application clients, 3-4
Application Explorer, 1-2 to 1-3, 2-11, 2-14, 4-2
  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
  WSDL files and, 4-19
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
Applications tab, 4-10, 4-22
Applications-Navigator pane, 4-17, 4-27
Assign activities, 4-15
Assign Activity dialog box, 4-15
Assign activity icon, 4-15
asynchronous support, 1-2
Available list, 7-4 to 7-5
back-end workflows
  calling, 4-17
BPEL Console, 4-17 to 4-19, 4-30
  starting, 4-17
BPEL Designer, 4-1, 4-10, 4-22
  BPEL processes and, 4-17
BPEL domain
  passwords and, 4-30
BPEL PM Server, 4-12, 4-17, 4-25
  connecting to, 4-10, 4-22
BPEL Process Manager
  adapter request-response service and, 4-17
  integrating with adapters, 4-1
  OracleAS Adapter for PeopleSoft and, 4-1
BPEL Process Project dialog box, 4-10, 4-24
BPEL processes, 4-3
  deploying, 4-17
  designing, 4-11
  JDeveloper and, 4-17
  managing, 4-18
  monitoring, 4-17
  testing, 4-17, 4-18
BPEL Processes tab, 4-18
BPEL projects
  creating, 4-3
browsing metadata, 2-14
BSE (OracleAS Adapter Business Services Engine), 1-1, 1-3, 2-8
  configuring, 2-8
  connection access to, 7-6
  troubleshooting, 6-5
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, 2-3
BSE system settings, 2-3 to 2-5
BSE URL field, 2-9
business events, 1-1
business functions, 2-17
business objects, 2-11
  browsing, 2-14
  storing, 2-5
business processes, 1-2
business services
  creating, 2-16 to 2-18
  deploying, 2-1
  testing, 2-18
Business Services node, 7-2 to 7-4, 7-7

C

CCI (Common Client Interface), 3-1
CCI calls, 3-1
channel configuration parameters, 2-23
  Error Directory, 2-25
  File Mask, 2-24
  Host, 2-23
  Is Keep Alive, 2-23
  Is Length Prefix, 2-23
  Is XML, 2-23
  Poll interval, 2-25
  Polling Location, 2-24
  Port, 2-20, 4-20
  Port Number, 2-23
  Processing Mode, 2-25
  Response/Ack Directory, 2-25
  Server port, 2-20, 4-20
  Synchronization Type, 2-23 to 2-25
  Thread limit, 2-25
channel types
  File, 2-24 to 2-25
channels, 2-19 to 2-25
  creating, 2-19 to 2-25, 4-19 to 4-21
  deleting, 2-25
  editing, 2-25
  ports and, 2-22
  starting, 2-23, 4-21
  stopping, 2-23, 4-21
  testing and debugging, 4-21
channels. See also listeners
closing connections, 2-13
Common Client Interface (CCI), 3-1
Compiler tab, 4-17
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
configuring target connectors, B-1 to B-14
connecting to BPEL PM Server, 4-10, 4-22
connecting to OracleAS Adapter J2CA, 6-4
connecting to PeopleSoft, 2-11 to 2-13, 4-21, 6-2, 6-7
Connection dialog box, 4-21
connection factories, 3-4
connection parameters, 2-12, 4-3, 4-20 to 4-21, 4-26, 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, 2-20, 4-20, 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, 4-21
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
Copy Operation tab, 4-15
copy rules
  creating, 4-16
Country parameter, 3-4
Create Business Service dialog box, 2-17
Create Copy Rule dialog box, 4-15
Create Operation dialog box, 4-16
Create Outbound JCA Service (Request/Response), 4-9
Create Partner Link dialog box, 4-11, 4-24 to 4-25
Create Variable dialog box, 4-14, 4-27
creating BPEL processes, 4-3
creating channels, 2-19 to 2-25, 4-19 to 4-21
creating copy rules, 4-16
creating events, 4-21 to 4-22
creating repository projects, 2-8 to 2-9
creating schemas, 2-15, 6-2
creating Web services, 2-16 to 2-18

D
Dashboard tab, 4-17
data
  manipulating, 4-17
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, 2-25
deleting connections, 2-14
deploying adapters, 3-1 to 3-2, 4-2
deploying outbound BPEL processes, 4-17
deployments
    BSE, 1-3
Description field, 2-17, 7-2 to 7-5, 7-7
design time, 4-19, 7-8
    configuring, 4-2
design time service adapter connections, 3-4
designing BPEL processes, 4-3, 4-11
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
Domain Password field, 4-17

E
Eclipse, See JDeveloper
Edit Invoke dialog box, 4-14
Edit Receive dialog box, 4-26
editing channels, 2-25
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, 2-25
error messages, 4-15 to 4-17, 6-2 to 6-7
target systems and, 6-6
event data
    receiving, 4-27 to 4-30
event integration, 4-27 to 4-30, 5-34
event messages, 4-19, 4-30
event ports
    creating, 4-21 to 4-22
events, 1-1, 1-3
    creating, 4-21 to 4-22
Execution Denied list, 7-6
Execution Granted list, 7-6
Existing Service Names list, 2-17
Export WSDL dialog box, 4-22

F
fault code elements, 6-6
fault string elements, 6-6
File channel, 2-24 to 2-25
File Mask parameter, 2-24
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
Home field, 2-9
Host parameter, 2-23
Hostname parameter, 2-1, 2-9, 6-2 to 6-3
HTTP protocol, 2-22

I
inbound BPEL processes, 4-10
inbound interactions, 4-22
inbound J2CA services, 4-19
Initiate tab, 4-18
input records
    creating, 3-6
installation directories, 3-3
instances of policy types, 7-1
Instances tab, 4-30
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
inbound, 4-22
Invalid Settings warning, 4-15
Invoke activities, 4-13
IP (Mask)/Domain field, 7-7
IP addresses, 7-7
IP and Domain Restriction policy type, 7-6
Is Keep Alive parameter, 2-23
Is Length Prefix parameter, 2-23
Is XML parameter, 2-23
IWA function, 3-6
IWAFConnectionSpec, 3-4 to 3-5
IWAFInteractionSpec, 3-6
IWConfig parameter, 3-3
IWHome parameter, 3-3
IRepoPassword parameter, 3-3
IRepoURL parameter, 3-3
IRepoUser parameter, 3-3
iwpsci84.jar file, 6-2
iwpsci8x.jar file, 6-2
iwse.ora file, 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
J2CA services, 4-19
JAR files, 6-2
errors and, 6-2
Java application clients, 3-4
Java program clients, 3-1
JDeveloper, 4-1, 4-10, 4-22
BPEL processes and, 4-17
JNDI lookup, 3-4

L
Language parameter, 2-4, 3-4
License and Method dialog box, 2-17
License field, 2-17
licenses, 2-17
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
mapping variables, 4-16
mappings
verifying, 4-16
message channels
configuring, D-15 to D-17
message routers
configuring, B-1 to B-14
message types
event, 4-19, 4-30
warning, 4-15 to 4-17
messages, 1-1, 1-3
logging, 4-17
Messages log area, 4-17
metadata, 2-14
browsing, 2-14
storing, 2-5, 7-8
Method Name field, 2-17
methods, 7-1
changing, C-4
viewing, C-4
migrating repositories, 7-8
migrating Web services, 7-10
My Role field, 4-12

N
Name field, 7-2 to 7-5
Namespace field, 4-10, 4-24
New Configuration dialog box, 2-8 to 2-9
New Gallery window, 4-10, 4-23
New Group dialog box, 7-3
New Policy permissions dialog box, 7-5
New User dialog box, 7-2
Node list, 6-2
nodes
Adapters, 4-21
Business Services, 7-2 to 7-4
Configuration under Business Services, 7-2 to 7-4, 7-7
Configurations, 2-9
connected, 4-21
Gateway, D-7 to D-11
Groups, 7-3 to 7-4
PeopleSoft, 2-11, 4-21
Policies, 7-4
Process, 4-16
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

OC4J (OracleAS Containers for J2EE)
deploying, 3-1 to 3-6
oc4j-ra.xml file, 3-2 to 3-4
Oracle Application Server
deployment of Connector to, 3-1
Oracle Application Server Adapter
installation directory and, 3-3
J2CA and, 3-1
Oracle BPEL Console, 4-17 to 4-19, 4-30
starting, 4-17
Oracle databases, 3-3
Oracle JDeveloper, 4-1, 4-10, 4-22
BPEL processes and, 4-17
Oracle repositories
migrating, 7-8
OracleAS Adapter Application Explorer. See also
Application Explorer
OracleAS Adapter Business Services Engine
(BSE), 1-1, 1-3, 2-8
configuring, 2-8
connection access to, 7-6
troubleshooting, 6-5
OracleAS Adapter for PeopleSoft
BPEL Process Manager and, 4-1
configuring, 2-1 to 2-26
deploying, 1-1, 4-2
integrating with PeopleSoft, 4-17, 4-27 to 4-30
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
OracleAS Containers for J2EE (OC4J)
deploying, 3-1 to 3-6
outbound BPEL processes, 4-10, 4-22
JDeveloper and, 4-17
outbound integration, 4-17
Outbound Interaction, 4-17
outbound processes
deploying, 4-17

P
parameter types
channel configuration, 2-23
configuration, 3-3
connection, 2-12, 3-4, 4-3, 4-20 to 4-21, 4-26, 6-2
repository, 2-5
repository migration, 7-9
security, 2-4
system, 2-4
Partner Link Type field, 4-12
partner links, 4-11 to 4-12, 4-13, 4-24, 4-26
Partner Role field, 4-12
Password parameter, 2-2 to 2-3, 2-12, 3-4 to 3-5, 7-2
Password Prompt dialog box, 4-17
passwords, 3-3 to 3-5, 4-17, 4-27
BPEL domain and, 4-30
PeopleSoft, 3-6
collecting to, 2-11 to 2-13, 6-2, 6-7
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, 4-21
PeopleSoft process link, 4-18
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 parameter, 2-4
policy types, 7-1
IP and Domain Restriction, 7-6
policy-based security, 7-1 to 7-7
Poll interval parameter, 2-25
Polling Location parameter, 2-24
Port Number parameter, 2-9, 2-23
Port parameter, 2-2, 2-12, 2-20, 4-20, 6-2 to 6-3
ports
channels and, 2-22
creating, 4-21 to 4-22
privileges, 7-1
setting, 7-1
process activities, 4-13
Process Activities pane, 4-11 to 4-13, 4-24, 4-26
PROCESS function, 3-6
Process Manager. See BPEL Process Manager
Process node, 4-16
processes
designing, 4-11
synchronous, 4-10, 4-22
Processing Mode parameter, 2-25
projects
BPEL, 4-3
properties, 3-3
Protocol list, 2-22
psjoa.jar file, 6-2

R
ra.xml file, 2-2
Receive activities, 4-13, 4-26
Receive dialog box, 4-17
record types
input, 3-6
records
creating, 3-6
repositories

trace information, 6-1
transaction processing, 1-1
transactions
  calling, 4-17
  storing, 2-5
troubleshooting, 6-1 to 6-7
  Application Explorer, 6-1 to 6-2
  BSE, 6-5
  Web services, 6-5 to 6-7
Type list, 7-5, 7-7

U
upating adapter configuration, 3-2
User ID parameter, 2-2, 2-12
User Name parameter, 3-4 to 3-5
users
  associating, 7-2
Users and Groups node, 7-2 to 7-3
Users node, 7-3

V
variables
  mapping, 4-16
  verifying mappings, 4-16
viewing methods, C-4
visual editors, 4-11 to 4-15, 4-24, 4-26

W
warning messages, 4-15 to 4-17
Web Service Definition Language
  (WSDL), 2-15 to 2-16
Web service names, 2-17
Web services, 1-1 to 1-3, 2-8
  creating, 2-16 to 2-18
  delivering, 2-5
  deploying, 6-7, 7-1
  integrating, 4-1
  migrating, 7-10
  repository projects and, 2-8
  testing, 2-18
  troubleshooting, 6-5 to 6-7
Web services policy-based security, 7-1 to 7-7
workspaces, 4-10, 4-22
WSDL (Web Service Definition Language), 2-15 to 2-16
WSDL documents, 4-1
WSDL File field, 4-12
WSDL file location dialog box, 7-8
WSDL files, 4-1
  Application Explorer and, 4-19
  creating, 4-19
  WSDL servlet, 4-12, 4-25

X
XML messages, 1-1
XML schemas, 1-2, 2-1, 2-15
  creating, 2-15