# Contents

## Preface
- Audience ix
- Documentation Accessibility ix
- Related Documents x
- Conventions x

## 1 Introduction
- Adapter Features 1-1
  - J.D. Edwards OneWorld Platforms, Products, and Releases Supported 1-2
    - J.D. Edwards OneWorld Inbound WSDL Generation on UNIX Platforms 1-2
    - J.D. Edwards OneWorld Unit Of Work (UOW) 1-2
    - J.D. Edwards OneWorld Versions and Library Files 1-3
  - J.D. Edwards OneWorld Concepts 1-3
  - Integration with J.D. Edwards OneWorld 1-4
- Adapter Architecture 1-6
- BSE Versus Oracle Adapter J2CA Deployment 1-9

## 2 Configuring Oracle Application Adapter for J.D. Edwards OneWorld
- Starting Application Explorer 2-1
- Configuring Repository Settings 2-2
- Creating a Repository Configuration 2-2
  - Creating a Configuration for BSE 2-2
  - Creating a Configuration for J2CA 2-3
  - Connecting to a BSE or J2CA Configuration 2-5
- Establishing a Connection (Target) for J.D. Edwards OneWorld 2-5
  - Defining a Target to J.D. Edwards OneWorld 2-6
- Creating an XML Schema 2-10
  - Creating a Request and a Response Schema 2-10
- Generating WSDL (J2CA Configurations Only) 2-12
- Creating and Testing a Web Service (BSE Configurations Only) 2-15
- Configuring an Event Adapter 2-18
  - Creating and Editing a Channel 2-18
  - The J.D. Edwards OneWorld Event Listener 2-31
- Configuring the J.D. Edwards OneWorld Event Listener 2-31
- Runtime Overview 2-34
Modifying the JDE.INI File for Outbound and Inbound Processing................................................. 2-34

3 Oracle WebLogic Server Deployment and Integration
Adapter Integration with Oracle WebLogic Server ........................................................................... 3-1
Deployment of Adapter ......................................................................................................................... 3-1
Updating Adapter Configuration........................................................................................................... 3-2

4 Integration With BPEL Service Components in the Oracle SOA Suite
Overview .................................................................................................................................................. 4-1
Deployment of Adapter ......................................................................................................................... 4-1
Configuring a New Application Server Connection ........................................................................... 4-2
Designing an Outbound BPEL Process for Service Integration .......................................................... 4-7
  Generating WSDL for Request/Response Service ............................................................................ 4-8
  Creating a New SOA Application for the Outbound BPEL Process ............................................... 4-9
  Defining a BPEL Outbound Process .................................................................................................. 4-12
  Deploying the BPEL Outbound Process ............................................................................................ 4-33
  Invoking the Input XML Document in the Oracle Enterprise Manager Console .......................... 4-37
  Testing Outbound BPEL and Mediator Processes ............................................................................ 4-40
Designing an Inbound BPEL Process for Event Integration ................................................................. 4-40
  Generating WSDL for Event Integration ............................................................................................ 4-41
  Creating a New SOA Application for the Inbound BPEL Process ............................................... 4-46
  Defining a BPEL Inbound Process ..................................................................................................... 4-49
  Deploying the BPEL Inbound Process ............................................................................................... 4-59
  Triggering an Event in J.D. Edwards OneWorld .............................................................................. 4-63

5 Integration With Mediator Service Components in the Oracle SOA Suite
Configuring a New Application Server Connection ........................................................................... 5-2
Configuring a Mediator Outbound Process ........................................................................................ 5-2
  Creating a New SOA Application for the Outbound Mediator Process ......................................... 5-2
  Defining a Mediator Outbound Process .............................................................................................. 5-6
  Deploying the Mediator Outbound Process ........................................................................................ 5-20
  Invoking the Input XML Document in the Oracle Enterprise Manager Console ......................... 5-24
Configuring a Mediator Inbound Process ............................................................................................ 5-27
  Creating a New SOA Application for the Inbound Mediator Process ......................................... 5-27
  Defining a Mediator Inbound Process ............................................................................................... 5-31

6 Integration With BPM Service Components in the Oracle SOA Suite
Overview .................................................................................................................................................. 6-1
Deployment of Adapter ......................................................................................................................... 6-1
Configuring a New Application Server Connection ........................................................................... 6-2
Designing an Outbound BPM Process Using Transformations for Service Integration ................. 6-2
  Creating an Empty Composite for BPM ............................................................................................ 6-2
  Defining a BPM Outbound Process .................................................................................................. 6-4
  Deploying the BPM Outbound Process ............................................................................................ 6-47
  Invoking the Input XML Document in the Oracle Enterprise Manager Console ......................... 6-51
Designing an Inbound BPM Process Using Transformations for Event Integration ......................... 6-54
7 Configuring Outbound and Inbound Processing Using Oracle Service Bus

Overview of Application Adapter Integration with Oracle Service Bus ........................................ 7-1
Configuring Outbound Processing Using Oracle Service Bus (J2CA Configuration) .................... 7-1
  Starting Oracle Service Bus and Creating Project Folders ..................................................... 7-2
  Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus .......... 7-4
  Publishing a WSDL From Application Explorer to Oracle Service Bus ............................... 7-5
  Configuring a File Type Business Service .............................................................................. 7-8
  Configuring a Proxy Service..................................................................................................... 7-13
  Configuring a Pipeline ............................................................................................................. 7-18
Configuring Inbound Processing Using Oracle Service Bus (J2CA Configuration) ................. 7-27
  Starting Oracle Service Bus and Creating Project Folders .................................................... 7-27
  Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus .......... 7-30
  Publishing a WSDL From Application Explorer to Oracle Service Bus ............................... 7-30
  Importing the Inbound J.D. Edwards OneWorld WSDL File Into Oracle Service Bus .......... 7-35
  Configuring a File Type Business Service .............................................................................. 7-42
  Configuring a Pipeline ............................................................................................................. 7-46
Configuring Outbound Processing Using Oracle Service Bus (BSE Configuration) ............... 7-51
  Starting Oracle Service Bus and Creating Project Folders .................................................... 7-52
  Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus .......... 7-54
  Publishing a WSDL From Application Explorer to Oracle Service Bus ............................... 7-54
  Configuring a File Type Business Service .............................................................................. 7-57
  Configuring a WSDL Type Business Service ......................................................................... 7-59
  Configuring a Proxy Service..................................................................................................... 7-64
  Configuring a Pipeline ............................................................................................................. 7-67

8 Troubleshooting and Error Messages

Troubleshooting ........................................................................................................................ 8-1
BSE Error Messages .................................................................................................................. 8-4
  General Error Handling in BSE .............................................................................................. 8-4
  Adapter-Specific Error Handling ............................................................................................ 8-5

9 Advanced User Tools

Web Services Policy-Based Security ............................................................................................ 9-1
  Configuring Web Services Policy-Based Security .................................................................... 9-2
Migrating Repositories ................................................................................................................ 9-9

A Configuring J.D. Edwards OneWorld for Outbound and Inbound Processing

Modifying the JDE.INI File for Outbound and Inbound Processing ...................................... A-1
Using the GenJava Development Tool (Outbound Processing) .............................................. A-2
Triggering J.D. Edwards OneWorld Events ............................................................................. A-3
Glossary

Index
Preface


Audience


Documentation Accessibility

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/support/contact.html or visit http://www.oracle.com/accessibility/support.html if you are hearing impaired.
Related Documents

For more information, see the following documents in the Oracle Enterprise Repository 11g Release 1 (11.1.1.3.0) documentation set:

- Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server
- Oracle Fusion Middleware Application Adapter Upgrade Guide for Oracle WebLogic Server
- Oracle’s Unified Method (OUM)

A wealth of additional Governance information can be found within Oracle’s Unified Method (OUM). OUM can be used by Oracle employees, Oracle Partner Network Certified Partners or Certified Advantage Partners, and Clients who either participate in the OUM Customer Program or are engaged on projects where Oracle provides consulting services. OUM is a web-deployed toolkit for planning, executing and controlling software development and implementation projects.

For more information about OUM, see the OUM FAQ at http://my.oracle.com/portal/page/myo/ROOTCORNER/KNOWLEDGEAREASI/BUSINESS_PRACTICE/Methods/Learn_about_OUM.html

Conventions

The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td><em>italic</em></td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td>monospace</td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
Oracle WebLogic Server connects to a J.D. Edwards OneWorld system through Oracle Application Adapter for J.D. Edwards OneWorld. Oracle Application Adapter for J.D. Edwards OneWorld provides connectivity and carries out interactions on a J.D. Edwards OneWorld system. This chapter discusses the following topics:

This chapter contains the following topics:

- Section 1.1, "Adapter Features"
- Section 1.2, "J.D. Edwards OneWorld Platforms, Products, and Releases Supported"
- Section 1.3, "J.D. Edwards OneWorld Concepts"
- Section 1.4, "Integration with J.D. Edwards OneWorld"
- Section 1.5, "Adapter Architecture"
- Section 1.6, "BSE Versus Oracle Adapter J2CA Deployment"

1.1 Adapter Features

Oracle Application Adapter for J.D. Edwards OneWorld provides a means to exchange real-time business data between J.D. Edwards systems and other applications, databases, or external business partner systems. The adapter enables inbound and outbound processing with J.D. Edwards.

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

Oracle Application Adapter for J.D. Edwards OneWorld uses XML messages to enable non-J.D. Edwards OneWorld applications to communicate and exchange transactions with J.D. Edwards OneWorld using services and events. Services and events are described as follows:

- Services: Enables applications to initiate a J.D. Edwards OneWorld business event.
- Events: Enables applications to access J.D. Edwards OneWorld data only when a J.D. Edwards OneWorld business event occurs.

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

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

Oracle Application Adapter for J.D. Edwards OneWorld provides:

- XML schemas and WSDLs for the J2CA 1.0 and 1.5 resource adapter.
- Web services for BSE.

See Also: Oracle Application Server Adapter Concepts Guide

1.2 J.D. Edwards OneWorld Platforms, Products, and Releases Supported

The following table indicates which combinations of adapter platforms and J.D. Edwards OneWorld platforms are supported, and for each combination, which J.D. Edwards OneWorld products and releases are supported.

<table>
<thead>
<tr>
<th>Adapter Platform</th>
<th>J.D. Edwards OneWorld Platform</th>
<th>J.D. Edwards OneWorld Product and Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows, AS400, HP 9000/B, Sun or IBM RS/6000</td>
<td>Windows, AS400, HP 9000/B, Sun or IBM RS/6000</td>
<td>- XE (B7333) from SP19 to SP23</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ERP 8.0 (B7334)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- EnterpriseOne B9 (8.9)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- EnterpriseOne 8.10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(with Tools release 8.93 and 8.94)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- EnterpriseOne 8.11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(SP1 and Tools Release 8.95)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- EnterpriseOne 8.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(up to Tools Release 8.97 1.2)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- EnterpriseOne 9.0 (Tools Release 8.98)</td>
</tr>
</tbody>
</table>

1.2.1 J.D. Edwards OneWorld Inbound WSDL Generation on UNIX Platforms

On UNIX platforms, manual inbound WSDL documents can be generated for J.D. Edwards OneWorld.

1.2.2 J.D. Edwards OneWorld Unit Of Work (UOW)

The following section provides information on the J.D. Edwards OneWorld Unit of Work (UOW).

1. iWay recommends the customer to generate individual business functions of J.D. Edwards OneWorld and then group them together.

2. Generating the individual business functions and grouping them together is completely based on the experience and knowledge related to the business functions of J.D. Edwards OneWorld. There is no documentation on generating the UOW.

3. To create the XML Schema Document (XSD) files that can be used for UOW, perform the following steps:

   a. Trigger the appropriate event from the J.D. Edwards OneWorld GUI and generate an XML output file based on the event.
b. Using the XML file, create an XSD file using an XML editor, such as XMLSPY.
c. When creating the XSD, ensure that the XSD satisfies the SOA 11g namespace requirements. Manually add the namespace, target namespace, and other items that are required.

4. Once generated, copy the XSD files for the UOW into the repository folder. This repository folder is automatically configured on your file system when a J.D. Edwards OneWorld target is created using Application Explorer.

1.2.3 J.D. Edwards OneWorld Versions and Library Files

The following table lists the library files by version that are required for the Oracle Application Adapter for J.D. Edwards OneWorld.

<table>
<thead>
<tr>
<th>J.D. Edwards OneWorld Version</th>
<th>Required Library Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>XE (B7333)</td>
<td>Connector.jar and Kernel.jar</td>
</tr>
<tr>
<td>ERP 8.0 (B7334)</td>
<td>Connector.jar and Kernel.jar</td>
</tr>
<tr>
<td>EnterpriseOne 8.9 (B9)</td>
<td>Connector.jar, Kernel.jar, jdeutil.jar, and log4j.jar</td>
</tr>
<tr>
<td>EnterpriseOne 8.10</td>
<td>Connector.jar, Kernel.jar, jdeutil.jar, and log4j.jar</td>
</tr>
<tr>
<td>EnterpriseOne 8.11 (SP1 and Tools Release 8.95)</td>
<td>Base_JAR.jar, Connector.jar, JdeNet_JAR.jar, log4j.jar, and System_JAR.jar</td>
</tr>
</tbody>
</table>

On the J.D. Edwards OneWorld system, these library files are located in the following folder:
```
\system\classes
```

1.3 J.D. Edwards OneWorld Concepts

You can use Oracle Application Adapter for J.D. Edwards OneWorld to call a J.D. Edwards OneWorld Master Business Function, such as Address Book, Purchase Order, and Sales Order. You can also use the adapter as a part of an integration effort to connect J.D. Edwards OneWorld with non-J.D. Edwards OneWorld systems.

Oracle Application Adapter for J.D. Edwards OneWorld can receive an XML document, or it can run one or more J.D. Edwards OneWorld Master Business
Functions (MBF) by passing an XML document into J.D. Edwards OneWorld through the J.D. Edwards OneWorld ThinNet API.

1.4 Integration with J.D. Edwards OneWorld

J.D. Edwards OneWorld supports multiple methods and technologies to provide interoperability. The three supported entry points are:

- Flat files
- Database tables
- Master Business Function (MBF) interactive calls

You configure Oracle AS Adapter to send requests to J.D. Edwards OneWorld. The adapter processes requests for J.D. Edwards OneWorld Master Business Functions (MBF), embedded in XML documents, and forwards them to a back-end J.D. Edwards OneWorld system. The resulting response information is then returned and processed for further routing.

Oracle Application Adapter for J.D. Edwards OneWorld can receive an XML request document from a client and call a specific function in the target Enterprise Information System (EIS). Oracle Application Adapter for J.D. Edwards OneWorld acts as a consumer of request messages and provides a response. An adapter performs the following functions:

- Receives requests from a legacy system, another EIS, or a non-EIS client.
- Transforms the XML request document into the EIS-specific format.
  The request document conforms to a request XML schema.
  The schema is based on metadata in the EIS.
- Calls the underlying function in the EIS and waits for its response.
- Transforms the response from the EIS-specific data format to an XML document.
  The response document conforms to a response XML schema that is generated by the adapter.
  The schema is based on metadata in the EIS.

You can configure a channel for the adapter to receive messages from J.D. Edwards OneWorld. The information the channel receives is used to build an XML record and is forwarded to any specified disposition for further processing.

Channels are consumers of EIS-specific messages and may or may not provide a response. A channel performs the following functions:

- Receives messages from an EIS client
- Transforms the EIS-specific message format into an XML format.

Propagating Internal Listeners out of J.D. Edwards OneWorld

Integrating a J.D. Edwards OneWorld listener with external systems is similar to the outbound process, except in reverse. The Data Export Control table maintains the determination of whether a transaction must be integrated with an external system. When a transaction must be integrated, the MBF handles logging of all additions, changes, and deletions to the unedited transaction table. After the transaction information writes to the table, a key for that record is sent from the MBF to the subsystem data queue.
The subsystem data queue triggers the processing of the new record by launching an outbound subsystem batch process that is generic and handles all inbound transactions. The J. D. Edwards outbound subsystem then accesses the Data Export Control table to determine the configured external subscriber to run.

**J.D. Edwards OneWorld Interoperability Framework**

J.D. Edwards OneWorld enables integration with systems through its interoperability framework. The adapter uses the framework and leverages various integration access methods to provide the greatest amount of flexibility and functionality.

Oracle Application Adapter for J.D. Edwards OneWorld supports the following integration access methods:

- J.D. Edwards OneWorld ThinNet API
- J.D. Edwards OneWorld XML
- J.D. Edwards OneWorld unedited transaction tables (Z tables)

*Figure 1–1* illustrates the outbound processing framework.

The adapter uses the J.D. Edwards OneWorld ThinNet API to communicate with the J.D. Edwards OneWorld application. Using the ThinNet API, the adapter can run one or more MBF in a single Unit Of Work (UOW). When any of the MBF fail, the entire UOW fails, preventing partial updates. Validation of data, business rules, and communications to the underlying database are handled by the J.D. Edwards OneWorld application because the adapter runs the MBF.

*Figure 1–1  J.D. Edwards OneWorld Outbound Processing*

![Figure 1–1  J.D. Edwards OneWorld Outbound Processing](image)

*Figure 1–2* illustrates the inbound processing framework.
In the outbound process, the event starts when a specific MBF is executed in the J.D. Edwards OneWorld environment. The MBF writes the required information for the event into the appropriate interface table and then notifies the subsystem Batch Function (BF) that an event occurred. The subsystem BF then places an entry about the event on the Subsystem Data Queue.

The J.D. Edwards OneWorld outbound subsystem retrieves the data queue entry and looks in the Data Export Control table for the external processes to notify. The J.D. Edwards OneWorld outbound subsystem then calls the Oracle Application Adapter for J.D. Edwards OneWorld listener with notification. The listener passes the notification to the generator. The generator then uses the J.D. Edwards OneWorld ThinNet API to retrieve the appropriate information from the interface table.

1.5 Adapter Architecture

Oracle Application Adapter for J.D. Edwards OneWorld uses Application Explorer with one of the following components:

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

**Oracle Adapter Application Explorer (Application Explorer)**

Application Explorer is used to configure database connections and create Web services and events. It can be configured to work in a Web services environment with BSE or with the Enterprise Connector for J2EE Connector Architecture (J2CA). When
working in a J2CA environment, the connector uses the Common Client Interface (CCI) to provide fast integration services using Adapters instead of using Web services.

Both BSE and the connector for J2CA are deployed to an application server with Application Explorer and the adapters.

Application Explorer uses an explorer metaphor for browsing the J.D. Edwards system for business functions. Application Explorer enables you to create XML schemas and Web services for the associated business function.

**Resource Adapters**

Oracle Application Adapter for J.D. Edwards OneWorld is a J2CA-based component also known as resource adapter. Resource adapters connect applications that were not originally designed to communicate with each other. Adapters are bidirectional, that is, they can send requests to an Enterprise Information System (EIS), and receive notification of events occurring in an EIS.

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

Figure 1–3 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.

**Figure 1–3 Oracle Adapter Business Services Engine (BSE) Architecture**

![Oracle Adapter Business Services Engine (BSE) Architecture Diagram]

**Note:** Do not use a file repository for BSE in production environments.
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 WebLogic Server Adapter Generic J2CA Architecture**

Figure 1–4 shows the generic architecture for Oracle J2CA adapter for packaged applications. The J2CA connector is deployed to a standard J2CA Container and serves as host container to the adapters. The connector is configured with a repository.

**Figure 1–4  Oracle Adapter Generic J2CA Architecture**

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 during these operations is stored in the repository by the connector. The repository can be a file system or an Oracle database. It is deployed as a RAR file and has an associated deployment descriptor called `ra.xml`. You can create multiple connector factories by editing the Oracle WebLogic Server deployment descriptor `ra.xml`. For more information, see Chapter 3, "Oracle WebLogic Server Deployment and Integration".

**Processing Business Functions**

Oracle Application Adapter for J.D. Edwards OneWorld enables the processing of J.D. Edwards OneWorld business functions through the J.D. Edwards ThinNet API. Using the API eliminates the requirement of creating complex and impractical batch processes. In addition, a transport layer, such as IBM MQSeries, File, or HTTP is not required because a listener is defined through a TCP connection.

External applications that access J.D. Edwards OneWorld through Oracle Application Adapter for J.D. Edwards OneWorld use either XML schemas or Web services to pass data between the external application and the adapter. Chapter 2, "Configuring Oracle
Application Adapter for J.D. Edwards OneWorld describes how to use Application Explorer to create XML schemas and Web services for the J.D. Edwards Master Business Functions (MBF) used with the adapter.

1.6 BSE Versus Oracle Adapter J2CA Deployment

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

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

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

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

2. Oracle Adapter J2CA does provide slightly better performance than BSE.
This chapter describes how to use Oracle Adapter Application Explorer (Application Explorer) to define a target to connect to a J.D. Edwards OneWorld system, view system objects, and create XML schemas and Web services. This chapter also explains how to configure an event adapter.

This chapter contains the following topics:

- Section 2.1, "Starting Application Explorer"
- Section 2.2, "Configuring Repository Settings"
- Section 2.3, "Creating a Repository Configuration"
- Section 2.4, "Establishing a Connection (Target) for J.D. Edwards OneWorld"
- Section 2.5, "Creating an XML Schema"
- Section 2.6, "Generating WSDL (J2CA Configurations Only)"
- Section 2.7, "Creating and Testing a Web Service (BSE Configurations Only)"
- Section 2.8, "Configuring an Event Adapter"
- Section 2.9, "Runtime Overview"
- Section 2.10, "Modifying the JDE.INI File for Outbound and Inbound Processing"

### 2.1 Starting Application Explorer

To start Application Explorer:

1. Ensure that Oracle WebLogic Server is started, which is where Application Explorer is deployed.

2. Click the Windows Start menu, select All Programs, Oracle Application Adapters, and then click Application Explorer, as shown in Figure 2–1.

![Oracle Application Adapters Program Menu](image_url)

Application Explorer is displayed.
Alternatively, you can run the `ae.bat` file, which is located in the following directory to start Application Explorer:

Oracle SOA Suite:

```
<ORACLE_HOME>\soa\thirdparty\ApplicationAdapters\tools\iwaebin\ae.bat
```

Oracle Service Bus (OSB):

```
<OSB_HOME>\Oracle_OSB1\3rdparty\ApplicationAdapters\tools\iwaebin\ae.bat
```

It is a good practice to create a shortcut for the `ae.bat` file on your desktop. If you are using a UNIX or Linux platform, then you can start Application Explorer by using the `iwae.sh` file.

Application Explorer starts. You can now define new targets to a J.D. Edwards OneWorld system.

---

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

```
chmod +x iwae.sh
```

---

### 2.2 Configuring Repository Settings

A repository holds information about configuration details, adapter targets, channels, and other configuration information. For more information on how to configure BSE and J2CA repository settings, see the *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server* (Section 2.8.4, “Configuring the Database Repository”).

### 2.3 Creating a Repository Configuration

Before you use Application Explorer with Oracle Application Adapter for J.D. Edwards OneWorld, you must create a repository configuration. You can create two kinds of repository configurations, Web services and J2CA, depending on the container to which the adapter is deployed. During design time, the repository is used to store metadata created when using Application Explorer to configure adapter connections, browse EIS objects, configure services, and configure listeners to listen for EIS events. The information in the repository is also referenced at run-time.

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

#### 2.3.1 Creating a Configuration for BSE

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

**Defining a New Configuration for BSE**

To define a new configuration for BSE:

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

   The New Configuration dialog is displayed, as shown in Figure 2–2.
2. Enter a name for the new configuration (for example, myConfig) and click OK. The New Configuration dialog is displayed, as shown in Figure 2–3.

3. From the Service Provider list, select iBSE.

4. In the iBSE URL field, accept the default URL or replace it with a different URL using the following format:

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

   Where host name is the system where your Oracle WebLogic Server resides and port is the HTTP port for a managed Oracle WebLogic Server (for example, soa_server1).

5. Click OK.

   A node representing the new configuration appears beneath the root Configurations node, as shown in Figure 2–4.

### 2.3.2 Creating a Configuration for J2CA

To create a configuration for 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 is displayed.

2. Enter a name for the new configuration (for example, myConfig) and click **OK**, as shown in Figure 2–5.

*Figure 2–5  New Configuration Dialog*

![New Configuration Dialog](image)

3. From the **Service Provider** list, select **JCA**.

4. Click **OK**.

A node representing the new configuration appears beneath the root Configurations node, as shown in Figure 2–6.

*Figure 2–6  Configurations Node*

The Oracle Adapter J2CA configuration folder is stored in:

```
<ADAPTER_HOME>/soa/thirdparty/ApplicationAdapters/config/configuration_name
```

Where **configuration_name** is the name of the configuration you created (for example, myConfig).

**HTTP Repository Connection**

You can create an HTTP repository connection using J2CA, 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, ensure that the iwjcavp 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 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://iwserv14:8001/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 Oracle WebLogic Server for run time purposes.

### 2.3.3 Connecting to a BSE or J2CA Configuration

To connect to a BSE or J2CA 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, then the Business Services node is not shown. The following is an example of a BSE configuration named myConfig, as shown in Figure 2–7.

![Figure 2–7 BSE Configuration](image)

- Use the **Adapters** folder to create inbound interaction with J.D. Edwards OneWorld. For example, you use the JDEdwards node in the Adapters folder to configure a service that updates J.D. Edwards OneWorld.
- Use the **Events** folder (available for J2CA configurations only) to configure listeners that listen for events in J.D. Edwards OneWorld.
- 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 can now define new targets to J.D. Edwards OneWorld.

### 2.4 Establishing a Connection (Target) for J.D. Edwards OneWorld

Part of the application definition includes adding a target for the adapter. Setting up the target in Application Explorer requires information which is specific to the target.

To browse the available Master Business Functions (MBF), you must first define a target to the system you use. After you define the target, it is automatically saved. You must connect to the system every time you start Application Explorer or after you disconnect.

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.
2.4.1 Defining a Target to J.D. Edwards OneWorld

To connect to an application system for the first time, you must define a new target.

---

**Note:** Before you create a new target, you must obtain the required library files for your J.D. Edwards OneWorld system and copy them to the appropriate location where the Oracle Application Adapter for J.D. Edwards OneWorld is deployed. For more information, see the Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server.

---

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 **JDEdwards** node and select **Add Target**.
   
   The Add Target dialog is displayed, as shown in Figure 2–8.

**Figure 2–8 Add Target Dialog**

![Add Target Dialog](image)

Perform the following steps:

- **a.** In the **Name** field, enter a descriptive name, for example, JDEConnection.

- **b.** In the **Description** field, enter a description for the target (optional).

- **c.** From the **Type** list, select **JDE One World**.

3. Click **OK**.

   The JDE One World dialog appears, as shown in Figure 2–9.
Figure 2–9  JDE One World Dialog

![JDE One World Dialog](image)

a. In the **Repository** tab, enter the path to the GenJava repository in the Repository directory field.

   This is the location of the Java wrappers for accessing the J.D. Edwards OneWorld business functions, which are created by the GenJava development tool. Please note that this is a prerequisite step, which must be performed before a new target is created using Application Explorer.

---

**Note:** Generating schemas requires the GenJava repository. For more comprehensive information on building the J.D. Edwards OneWorld Master Business Function repository, see the *J.D. Edwards Interoperability Guide for OneWorld Xe*. For information on how to use the GenJava program, see *Using the GenJava Development Tool (Outbound Processing)* in Appendix A, "Configuring J.D. Edwards OneWorld for Outbound and Inbound Processing".

---

b. From the **Schema style** list, select **ELEMENT_STYLE** or **ATTRIBUTE_STYLE**.

c. Click the **Logon** tab and enter the appropriate information for your target type based on the information in the following table. Fields marked with an asterisk are required, as shown in Figure 2–10.
4. Click OK.

The new target, JDEConnection, appears under the JDEdwards node, as shown in Figure 2–11.

Connecting to a Defined J.D. Edwards OneWorld Target

To connect to a target:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User id*</td>
<td>A valid user ID for J.D. Edwards OneWorld.</td>
</tr>
<tr>
<td>User password*</td>
<td>The password associated with the user ID.</td>
</tr>
<tr>
<td>JDE environment*</td>
<td>The J.D. Edwards OneWorld environment, for example, DU7333. For more information about this parameter, see your J.D. Edwards OneWorld documentation or ask your J.D. Edwards OneWorld system administrator.</td>
</tr>
<tr>
<td>Server IP address*</td>
<td>The name of the server on which J.D. Edwards OneWorld is running. This can be the name of the server, for example, JDEOW, or its IP address.</td>
</tr>
<tr>
<td>Server Port*</td>
<td>The port number on which the server is listening, for example, 6009.</td>
</tr>
<tr>
<td>User role</td>
<td>This property setting depends on your J.D. Edwards OneWorld system version.</td>
</tr>
<tr>
<td></td>
<td>■ For B7333: You can leave this field blank.</td>
</tr>
<tr>
<td></td>
<td>■ For EnterpriseOne 8.10: You must specify *ALL.</td>
</tr>
</tbody>
</table>
Establishing a Connection (Target) for J.D. Edwards OneWorld

1. Expand the Service Adapters node.
2. Expand the JDEdwards node.
   The disconnected target is displayed, as shown in Figure 2–12.

Figure 2–12  JDEdwards Target Node

3. Click the target name (for example, JDEConnection) under the JDEdwards node.
4. Click the Logon tab on the right.
   The Logon tab displays the values you entered for connection parameters.
5. Verify your connection parameters.
6. Right-click the target name and select Connect.
   The x icon disappears, indicating that the node is connected, as shown in Figure 2–13.

Figure 2–13  JDEdwards Target Node

Disconnecting from J.D. Edwards OneWorld
To disconnect from a target:
1. Expand the Adapters node.
2. Expand the JDEdwards node.
3. Right-click the target to which you are connected (for example, JDEConnection), and select Disconnect.
   The x icon appears, indicating that the node is disconnected, as shown in Figure 2–14.

Figure 2–14  JDEdwards Target Node

Editing a Target
To edit a target:
1. In the left pane, ensure that the target you want to edit is disconnected.
2. Right-click the target and select Edit.
   A window is displayed that enables you to edit the existing connection parameters.
3. Modify the target information.
4. Click OK.

Deleting a Target
You can delete a target, rather than just disconnecting and closing it. When you delete
the target, the node disappears from the list of J.D. Edwards OneWorld targets in the
left pane of the explorer.

To delete a target:
1. Expand the Adapters node.
2. Expand the JDEdwards node.
3. Right-click the target to which you are connected (for example, JDEConnection),
   and select Delete.
   The node disappears from the list of available connections.

For information on how to view application system objects, see J.D. Edwards
Interoperability Guide Release OneWorld XE.

2.5 Creating an XML Schema
To execute an MBF, the adapter must receive a request document through the
J.D. Edwards OneWorld ThinNet API. The agent processes the request and sends an
XML response document indicating the result. Application Explorer creates both the
XML request schema and the XML response schema.

2.5.1 Creating a Request and a Response Schema
The following procedure explains how to create request and response schemas for a
J.D. Edwards OneWorld business function. Application Explorer enables you to create
XML schemas for this function.

1. Connect to a J.D. Edwards OneWorld target as described in "Connecting to a
   Defined J.D. Edwards OneWorld Target" on page 2-8.
2. Expand the Services node.
3. Expand the node of the MBF for which you want to create the schema.
4. Expand and then select the node beneath the MBF, as shown in Figure 2–15.

   Figure 2–15  Detail Tab

<table>
<thead>
<tr>
<th>Detail</th>
<th>parameters</th>
<th>Request Schema</th>
<th>Response Schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>lwaf description</td>
<td>Not Specified</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5. Click the parameters tab to view the parameter information Figure 2–16.
6. Click **Request Schema** to view the request schema information, as shown in Figure 2–17.

**Figure 2–17 Request Schema Tab**

![Request Schema Tab](image1.png)

```xml
<xsd:schema version="1.0" encoding="UTF-8"?>
  <xsd:element name="delRequest">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="params"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

7. Click **Response Schema** to view the response schema information, as shown in Figure 2–18.
2.6 Generating WSDL (J2CA Configurations Only)

The procedure for generating WSDL (Web Service Definition Language) for request-response (outbound) services differs from that of generating WSDL for event notification (inbound) J2CA services of the adapter. The following sections describe each procedure.

**Generating WSDL for Outbound Interaction**

To generate a WSDL file for request-response service:

1. Start Application Explorer and connect to a defined J.D. Edwards OneWorld target.

2. Expand Services, JDEJAVA_CFIN, and then B0100033. Select GetEffectiveAddress.

3. Right-click GetEffectiveAddress.

   The following menu is displayed, as shown in Figure 2–19.

4. Select Create Outbound JCA Service (Request/Response).

   The Export WSDL dialog is displayed, as shown in Figure 2–20.
Generating WSDL (J2CA Configurations Only)

Configuring Oracle Application Adapter for J.D. Edwards OneWorld

2-13

5. 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.

6. Click OK.

   The WSDL file is saved in the specified location.

Generating WSDL for Inbound Interaction

You cannot generate a WSDL for J.D. Edwards OneWorld event notification using Application Explorer. To generate a WSDL from the command prompt, you must perform the steps that are described in the following section.

You can create an inbound J2CA service only if the node that you selected supports events.

1. Create a channel using Application Explorer under the J.D. Edwards Events node.

2. Start the channel.

   Do not restart Oracle WebLogic Server after the channel is started.

3. Send an inbound message from J.D. Edwards OneWorld.

4. Capture the inbound message payload in the log file, which is located in the following directory:

   \<ADAPTER_HOME>\soa\thirdparty\ApplicationAdapters\config\config_name\log\iwaf_jca1500.log
Generating WSDL (J2CA Configurations Only)

Alternatively, you can create a port using the File protocol under the Events node in Application Explorer, which disposes the event message to the file system.

5. Use a third party tool (for example, XMLSpy) to create the XML schema (.xsd file) using the XML payload that was captured in the previous step.

6. In the generated XML schema (.xsd file) perform the following modifications:
   a. Search for Schemas-jdedwards-com and replace it with iwaysoftware.
      
      ```
      <xs:schema
          targetNamespace="urn:Schemas-jdedwards-com:trans.response.JDESOOUT"
          xmlns="urn:Schemas-jdedwards-com:trans.response.JDESOOUT"
          xmlns:xs=http://www.w3.org/2001/XMLSchema elementFormDefault="qualified">
      to:
      ```
      
      ```
      <xs:schema
          targetNamespace="urn:iwaysoftware:trans.response.JDESOOUT"
          xmlns="urn:iwaysoftware:trans.response.JDESOOUT"
          xmlns:xs=http://www.w3.org/2001/XMLSchema elementFormDefault="qualified">
      ```
      
   b. Cut the following syntax:
      
      ```
      <xs:element name="jdeResponse">
        <xs:complexType>
        </xs:complexType>
      </xs:element>
      ```
      
   c. Paste it before the following line:
      
      ```
      <xs:element name="transaction">
      ```
      
7. Copy the XML schema (.xsd file) from the following directory:

   `<ADAPTER_HOME>/soa/thirdparty/ApplicationAdapters/config/config_name/schemas/JDEdwards/target_name`

**Note:** Edit the created channel by providing the location of the schema (.xsd) file (as mentioned in step 7) in the PreParser tab of Application Explorer. For example:

   `<ADAPTER_HOME>/soa/thirdparty/ApplicationAdapters/config/config_name/schemas/JDEdwards/target_name/jde-schema.xsd`

8. Open a command prompt and navigate to the following directory:

   `<ADAPTER_HOME>/soa/thirdparty/ApplicationAdapters/tools/iwae/bin`

9. Execute the obadapter.bat file to set the environment.

10. Navigate to the following directory where the XML schema (.xsd file) is copied:

    `<ADAPTER_HOME>/soa/thirdparty/ApplicationAdapters/config/config_name/schemas/JDEdwards/target_name`

11. Enter the following command to generate a WSDL:

    ```
    java -Diway.oem=oracle11g
    com.iwaysoftware.af.container.tools.wsdl.IWayWSILBrowser adapterhome adapter
target channel schemaPrefix wsdlFileName
    ```
where:

adapterhome is the path to your ApplicationAdapters home. For example:

<ADAPTER_HOME>\soa\thirdparty\ApplicationAdapters\

adapter is the name of the adapter. For example, JDEdwards.

target is the name of the adapter target you created in Application Explorer.

channel is the name of the channel you created in Application Explorer.

schemaPrefix is the prefix for the XSD schema. The schema file must be in the
same directory where the Java command is executed, for example:

java -Diway.oem=oracle11g
com.iwaysoftware.af.container.tools.wsd1.IWayWSILBrowser
C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\wsdls\JDEdwards jde812 tgt jde_ch
Jde812_Schema Jde812_salesorder_receive.wsdl

Once the command is executed, the following is displayed in the command
window:

Running Inbound WSDL generation tool...
-> user.dir = java com.iwaysoftware.af.container.tools.wsd1.IWayWSILBrowser
 C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\wsdls\JDEdwards jde812 tgt jde_ch
 Jde812_Schema Jde812_salesorder_receive.wsdl

-> Generating WSDL...
-> Done.
-> Writing WSDL 'C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\wsdls\Jde812_salesorder_receive.wsdl' to disk...
-> Done.

---

**Note:** It is good practice to append _receive to the names of WSDL
files that are generated for event notification services. This allows you
to easily distinguish between them and those generated for
request-response services.

---

12. Stop the channel in Application Explorer.

2.7 Creating and Testing a Web Service (BSE Configurations Only)

You can generate a Web service (also known as a business service) using Application
Explorer. You can explore the business function repository and generate Web services
for the functions you want to use with the adapter. The following procedure uses an
example called BusinessUnitExistenceCheck.

---

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

To create a Web service for a business function:

1. Expand the JDEdwards node and then the Services node.
2. Expand the MBF, B1000012, also called BusinessUnitExistenceCheck.
3. Right-click the node from which you want to create a business service and select Create Web Service.

The Create Web Service dialog is displayed, as shown in Figure 2–21.

Figure 2–21  Create Web Service Dialog

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

a. From the Existing Service Names list, select either <new service> or an existing service.

b. In the Service Name field, specify a service name if you are creating a new service. This name identifies the Web service in the list of services under the Business Services node.

c. Enter a description for the service (optional).

4. Click Next.

Perform the following steps:

a. In the License Name field, select one or more license codes to assign to the Web service.

b. In the Method Name field, enter a descriptive name for the method or accept the default name.

c. In the Description field, enter a brief description of the method (optional).

d. In the DTD Directory field, specify a location where the Web service are saved. If you want to select a location different than the default, then click Browse and navigate to the desired 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 it functions properly. A test tool is provided for testing the Web service.

To test a Web 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, then an input field is displayed.

5. Enter the appropriate input.

6. Click **Invoke**.
   Application Explorer displays the results, as shown in **Figure 2–22**.

**Figure 2–22  Web Service XML Results**

```
<SOAP-ENV:Envelope
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:m="urn:schemas-iwaysoftware-com:iwse">
  <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:Envelope>
```

**Identity Propagation**

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

```
<SOAP-ENV:Header>
  <m:ibsinfo
      xmlns:m="urn:schemas-iwaysoftware-com:iwse">
    <m:service>String</m:service>
    <m:method>String</m:method>
    <m:license>String</m:license>
    <m:disposition>String</m:disposition>
    <m:Username>String</m:Username>
    <m:Password>String</m:Password>
    <m:language>String</m:language>
  </m:ibsinfo>
</SOAP-ENV:Header>
```
You can remove the `<m:disposition>` and `<m:language>` tags from the SOAP header, since they are not required.

### 2.8 Configuring an Event Adapter

Events are generated by activity in a database or in an application system. You can use events to trigger an action in your application. For example, an update to a database can reflect an update to customer information. If your application must perform when this happens, then 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.

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

A channel represents configured connections to particular instances of back-end systems. For more information, see "Creating and Editing a Channel" on page 2-18.

#### 2.8.1 Creating and Editing a Channel

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

When you create, modify, or delete a channel, you must restart the Oracle WebLogic Server to recognize the change and update the repository for run time purposes. After successfully creating the channel and inbound WSDL file, close Application Explorer before you restart the application server.

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

Three channel types are available:

- HTTP
- TCP
- File

**Note:** Channels can be configured only on the system where the Oracle Application Adapter for J.D. Edwards OneWorld is installed.

**Creating an HTTP Channel**

To create an HTTP Channel:
1. Click the **Events** node.

2. Expand the **JDEdwards** node.
   The ports and channels nodes appear in the left pane.

3. Right-click **Channels** and select **Add Channel**.
   The Add Channel dialog is displayed, as shown in Figure 2–23.

**Figure 2–23   Add Channel Dialog**

![Add Channel Dialog](image)

Provide the following information:

   **a.** Enter a name for the channel, for example, **JDE_Channel1**.
   **b.** Enter a brief description.
   **c.** From the **Protocol** list, select **HTTP Listener**.

4. Click **Next**.
   The Http Listener dialog is displayed, as shown in Figure 2–24.
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 J.D. Edwards OneWorld event data.</td>
</tr>
<tr>
<td>Https</td>
<td>For a secure HTTP connection, select the Https 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>
<tr>
<td>Important:</td>
<td>The J.D. Edwards OneWorld channel does not work if the synchronization type is set to REQUEST.</td>
</tr>
<tr>
<td>Encoding Type</td>
<td>Choose an encoding type to be used from the list. By default, ASCII is selected.</td>
</tr>
</tbody>
</table>

6. Click the PreParser tab, as shown in Figure 2-25.
7. Enter the system information as specified in the following table:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User id</td>
<td>A valid user ID for J.D. Edwards OneWorld.</td>
</tr>
<tr>
<td>User password</td>
<td>The password associated with the J.D. Edwards OneWorld user ID.</td>
</tr>
<tr>
<td>JDE Environment</td>
<td>The J.D. Edwards OneWorld environment, for example, DU7333. For more information about this parameter, see your J.D. Edwards OneWorld documentation or ask your J.D. Edwards OneWorld system administrator.</td>
</tr>
<tr>
<td>Application</td>
<td>The application that is defined in the J.D. Edwards OneWorld environment.</td>
</tr>
<tr>
<td>Server IP address</td>
<td>The name of the server on which J.D. Edwards OneWorld is running. This can be the name of the server, for example, JDEOW, or its IP address.</td>
</tr>
<tr>
<td>Server port</td>
<td>The port number on which the server is listening, for example, 6009.</td>
</tr>
<tr>
<td>User role</td>
<td>This property setting depends on your J.D. Edwards OneWorld system version.</td>
</tr>
<tr>
<td></td>
<td>■ For B7333: You can leave this field blank.</td>
</tr>
<tr>
<td></td>
<td>■ For EnterpriseOne 8.10: You must specify *ALL.</td>
</tr>
<tr>
<td>Schema Location</td>
<td>The location of the XML schema (.xsd file) that was generated from the event output. For example: &lt;\ADAPTER_HOME&gt;\soa\thirdparty\ApplicationAdapters\config\config\name\schemas\JDEdwards\target_name\jde-schema.xsd</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Generating WSDL for Inbound Interaction” on page 2-13.</td>
</tr>
<tr>
<td>Schema Style</td>
<td>Choose from one of the following options:</td>
</tr>
<tr>
<td></td>
<td>■ ELEMENT_STYLE (default)</td>
</tr>
<tr>
<td></td>
<td>■ ATTRIBUTE_STYLE</td>
</tr>
</tbody>
</table>
8. 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.
   
9. Right-click the channel and select **Start**.

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

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

**Creating a TCP Channel**

To create a TCP Channel:

1. Click the **Events** node.

2. Expand the **JDEdwards** node.

   The ports and channels nodes appear in the left pane.

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

   The Add Channel dialog is displayed, as shown in Figure 2–26.
Provide the following information:

- Enter a name for the channel, for example, JDE_Channel2.
- Enter a brief description.
- From the Protocol list, select TCP Listener.

4. Click Next.

The Tcp Listener dialog is displayed, as shown in Figure 2–27.
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>Port Number</td>
<td>Port on which the Host database is listening.</td>
</tr>
<tr>
<td>Host/IP Binding</td>
<td>Name or URL of the system where the database resides.</td>
</tr>
<tr>
<td>Synchronization</td>
<td>Choose from the following synchronization options:</td>
</tr>
<tr>
<td>Type</td>
<td>▪ REQUEST_RESPONSE</td>
</tr>
<tr>
<td></td>
<td>▪ REQUEST_ACK</td>
</tr>
<tr>
<td>Important:</td>
<td>The J.D. Edwards OneWorld channel does not work if the</td>
</tr>
<tr>
<td></td>
<td>synchronization type is set to REQUEST.</td>
</tr>
<tr>
<td>Is Length Prefix</td>
<td>For J.D. Edwards OneWorld events that send data back that is not in XML</td>
</tr>
<tr>
<td></td>
<td>format. The TCP/IP event application must prefix the data with a 4-byte</td>
</tr>
<tr>
<td></td>
<td>binary length field when writing the data to the TCP/IP port.</td>
</tr>
<tr>
<td>Is XML</td>
<td>For J.D. Edwards OneWorld events that send data back in XML format. No</td>
</tr>
<tr>
<td></td>
<td>preparser is required.</td>
</tr>
<tr>
<td>Is Keep Alive</td>
<td>Maintains continuous communication between the event transaction and the</td>
</tr>
<tr>
<td></td>
<td>channel.</td>
</tr>
</tbody>
</table>

6. Click the **PreParser** tab, as shown in Figure 2–28.
Figure 2–28 Tcp Listener Dialog Preparser Tab

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User id</td>
<td>A valid user ID for J.D. Edwards OneWorld.</td>
</tr>
<tr>
<td>User password</td>
<td>The password associated with the J.D. Edwards OneWorld user ID.</td>
</tr>
<tr>
<td>JDE Environment</td>
<td>The J.D. Edwards OneWorld environment, for example, DU7333. For more information about this parameter, see your J.D. Edwards OneWorld documentation or ask your J.D. Edwards OneWorld system administrator.</td>
</tr>
<tr>
<td>Application</td>
<td>The application that is defined in the J.D. Edwards OneWorld environment.</td>
</tr>
<tr>
<td>Server IP address</td>
<td>The name of the server on which J.D. Edwards OneWorld is running. This can be the name of the server, for example, JDEOW, or its IP address.</td>
</tr>
<tr>
<td>Server port</td>
<td>The port number on which the server is listening, for example, 6009.</td>
</tr>
<tr>
<td>User role</td>
<td>This property setting depends on your J.D. Edwards OneWorld system version.</td>
</tr>
<tr>
<td></td>
<td>■ For B7333: You can leave this field blank.</td>
</tr>
<tr>
<td></td>
<td>■ For EnterpriseOne 8.10: You must specify *ALL.</td>
</tr>
<tr>
<td>Schema Location</td>
<td>The location of the XML schema (.xsd file) that was generated from the event output. For example:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;ADAPTER_HOME&gt;\soa\thirdparty\ApplicationAdapters\config\config_name\schemas\JDEdwards\target_name\jde-schema.xsd</code></td>
</tr>
<tr>
<td></td>
<td>For more information, see “Generating WSDL for Inbound Interaction” on page 2-13.</td>
</tr>
<tr>
<td>Schema Style</td>
<td>Choose from one of the following options:</td>
</tr>
<tr>
<td></td>
<td>■ ELEMENT_STYLE (default)</td>
</tr>
<tr>
<td></td>
<td>■ ATTRIBUTE_STYLE</td>
</tr>
</tbody>
</table>

8. 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.

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

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

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

**Creating a File Channel**

To create a File Channel:

1. Click the **Events** node.

2. Expand the **JDEdwards** node.

The ports and channels nodes appear in the left pane.

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

The Add Channel dialog is displayed, as shown in Figure 2–29.

![Add Channel Dialog](image)

**Figure 2–29**  **Add Channel Dialog**
Provide the following information:

a. Enter a name for the channel, for example, JDE_Channel3.

b. Enter a brief description.

c. From the Protocol list, select File Listener.

4. Click Next.

The File Listener dialog is displayed, as shown in Figure 2–30.

**Figure 2–30   File Listener Dialog**

![File Listener Dialog](image)

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polling Location</td>
<td>Target file system location for the J.D. Edwards OneWorld 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>

6. Click the Response tab, as shown in Figure 2–31.
7. Enter the system information in the Response tab as specified in the following table:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>
<tr>
<td>Important:</td>
<td>The J.D. Edwards OneWorld channel does not work if the synchronization type is set to REQUEST.</td>
</tr>
<tr>
<td>Response/Ack Directory</td>
<td>Target file system location for the J.D. Edwards OneWorld XML file.</td>
</tr>
</tbody>
</table>

8. Click the Advanced tab, as shown in Figure 2–32.
9. Enter the system information in the Advanced tab as specified in the following table:

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

10. Click the PreParser tab, as shown in Figure 2–33.
Figure 2–33  File Listener Dialog Preparser Tab

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User id</td>
<td>A valid user ID for J.D. Edwards OneWorld.</td>
</tr>
<tr>
<td>User password</td>
<td>The password associated with the J.D. Edwards OneWorld user ID.</td>
</tr>
<tr>
<td>JDE Environment</td>
<td>The J.D. Edwards OneWorld environment, for example, DU7333. For more information about this parameter, see your J.D. Edwards OneWorld documentation or ask your J.D. Edwards OneWorld system administrator.</td>
</tr>
<tr>
<td>Application</td>
<td>The application that is defined in the J.D. Edwards OneWorld environment.</td>
</tr>
<tr>
<td>Server IP address</td>
<td>The name of the server on which J.D. Edwards OneWorld is running. This can be the name of the server, for example, JDEOW, or its IP address.</td>
</tr>
<tr>
<td>Server port</td>
<td>The port number on which the server is listening, for example, 6009.</td>
</tr>
<tr>
<td>User role</td>
<td>This property setting depends on your J.D. Edwards OneWorld system version.</td>
</tr>
<tr>
<td></td>
<td>■ For B7333: You can leave this field blank.</td>
</tr>
<tr>
<td></td>
<td>■ For EnterpriseOne 8.10: You must specify *ALL.</td>
</tr>
<tr>
<td>Schema Location</td>
<td>The location of the XML schema (.xsd file) that was generated from the event output. For example:</td>
</tr>
<tr>
<td></td>
<td>&lt;ADAPTER_HOME&gt;/soa/thirdparty/ApplicationAdapters/config/config_name/schemas/JDEdwards/target_name/jde-schema.xsd</td>
</tr>
<tr>
<td></td>
<td>For more information, see &quot;Generating WSDL for Inbound Interaction&quot; on page 2-13.</td>
</tr>
<tr>
<td>Schema Style</td>
<td>Choose from one of the following options:</td>
</tr>
<tr>
<td></td>
<td>■ ELEMENT_STYLE (default)</td>
</tr>
<tr>
<td></td>
<td>■ ATTRIBUTE_STYLE</td>
</tr>
</tbody>
</table>

12. 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.

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

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

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

**Editing a Channel**

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**

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 is displayed.

3. To delete the channel you selected, click **OK**.
   
   The channel disappears from the list in the left pane.

### 2.8.2 The J.D. Edwards OneWorld Event Listener

Oracle Application Adapter for J.D. Edwards OneWorld Event Listener is designed specifically to provide J.D. Edwards OneWorld approved access to your business events. The J.D. Edwards OneWorld Event Listener refers to a specialized application that runs with J.D. Edwards OneWorld business functions and is called by the J.D. Edwards OneWorld application system.

The J.D. Edwards OneWorld application system provides the Event Listener with the information required to retrieve the event information for only the desired events. For information about configuring the J.D. Edwards OneWorld environment, see the *J.D. Edwards Interoperability Guide for OneWorld*.

The J.D. Edwards OneWorld Event Listener is called directly from the J.D. Edwards OneWorld application and is passed a Z-file record identifier. This identifier then generates a request document that is passed to the server for processing. The server retrieves the event information from the J.D. Edwards OneWorld system and propagates the information for integration with other application systems.

### 2.8.3 Configuring the J.D. Edwards OneWorld Event Listener

The J.D. Edwards OneWorld Event Listener is installed as part of the basic installation. The J.D. Edwards OneWorld Adapter is automatically installed in the appropriate
directory. If the integration server is not installed on the same computer as the J.D. Edwards OneWorld application server, then you must configure the J.D. Edwards OneWorld Event Listener.

The J.D. Edwards OneWorld Event Listener is invoked by J.D. Edwards OneWorld for specific transactions as configured in the J.D. Edwards OneWorld environment.

The J.D. Edwards OneWorld Event Listener includes the following components:

- The listener exit (IWOEvent), located under adapters_home/etc\jde, where adapters_home is <ADAPTER_HOME>/soa/thirdparty/ApplicationAdapters. For example:
  <ADAPTER_HOME>/soa/thirdparty/ApplicationAdapters/etc/jde/iwoevent.dll

The file extension varies depending on your operating system:
- For Windows, the exit is iwoevent.dll.
- For Sun Solaris, the exit is libiwoevent.so.
- For HP-UX, the exit is libiwoevent.sl.
- For AS/400, the exit is iwaysav.sav.
- For IBM AIX, the exit is libiwoevent.so.

- The listener configuration file (iwoevent.cfg), which must be created by the user.

The J.D. Edwards OneWorld Event listener exit is the function that passes the key fields for a record in the J.D. Edwards OneWorld outbound transaction tables to the integration server for processing by the inbound Oracle Application Adapter for J.D. Edwards OneWorld. The J.D. Edwards OneWorld Event listener is deployed under the J.D. Edwards OneWorld Enterprise Server. The Java class for the J.D. Edwards OneWorld Event listener is called IWOEvent (the file extension depends on the operating system) and is case-sensitive.

1. Create a folder called Outbound under the JDE structure on the JDE Enterprise Server, for example:
   \\JDEdwards\E812\DDP\Outbound

2. Copy the iwoevent.dll file in the new Outbound folder.

3. Create an environment variable, IWOEVENT_HOME, to point to the directory containing the iwoevent.dll file.
   - On Windows: Add IWOEVENT_HOME to the system environment variables.
   - On UNIX: Add the following command to your start-up script:
     
     ```
     export IWOEVENT_HOME =/directory_name
     ```

4. On the J.D. Edwards OneWorld Server, create an iwoevent.cfg file in the defined directory, IWOEVENT_HOME.

   The J.D. Edwards OneWorld Event listener requires connection information for the associated adapter to initiate events properly. This information is contained in the iwoevent.cfg file. You must create this file and add the connection information to it. The J.D. Edwards OneWorld Event Listener requires connection information for the associated integration server to function properly. This information is contained in the iwoevent.cfg file. The iwoevent.cfg file has three distinct sections:
Configuring an Event Adapter

Common

The common section of the configuration file contains basic configuration options. Currently, only the trace option is supported.

To set the trace option, select **on** or **off**.

```
common.trace=on|off
```

Where **on** sets the tracing to on and **off** sets the tracing to off. Off is the default value.

Alias

The alias section of the configuration file contains the connection information required to send transactions to specific servers. Currently, the Oracle Application Adapter for J.D. Edwards OneWorld supports 100 entries (alias names) in the configuration file.

The alias values to these entries are as follows:

```
Alias.aliasname=(ipaddress|dsn):port, trace={on|off}
```

Where:

- **aliasname** is the symbolic name given to the connection.
- **ipaddress|dsn** is the IP address or DSN name for the server containing Oracle Application Adapter for J.D. Edwards OneWorld (required).
- **port** is the port defined for Oracle Application Adapter for J.D. Edwards OneWorld in the TCP channel configuration (required).
- **trace={on|off}** sets the tracing to on for the particular alias.

Trans

The trans section of the configuration file contains transaction information required to route J.D. Edwards OneWorld transactions to specified servers.

If a particular J.D. Edwards OneWorld transaction is not defined to an alias, then it is sent to all aliases. The trans values to these entries are as follows:

```
trans.jdeTransactionName=alias1,alias2,aliasn
```

Where **jdeTransactionName** is the JDE-defined name for the outbound transaction and **alias1,alias2,aliasn** is the list of aliases to which the transactions are sent.

The following is a sample entry for **iwoevent.cfg** that supplies connection information:

```
common.trace=on
alias.edamcs1=172.1.1.1:3694
alias.edamcs1t=172.1.1.1:3694,trace=on
alias.edamcs2=222.2.2.2:1234
trans.JDESOOUT=edamcs1t,edamcs2
trans.JDEPOOUT=edamcs1
```

5. Create a folder using the alias names that are specified in the iwoevent.cfg file under the defined directory, **IWOEVENT_HOME**. For example:

```
\\JDEdwards\E812\DDP\Outbound\edamcs1
```
2.9 Runtime Overview

After J.D. Edwards OneWorld starts the J.D. Edwards OneWorld Event listener, the listener accesses the configuration file, called *iwoevent.cfg* (case-sensitive). Based on the information in the configuration file, the listener sends the event notification to the integration server. All log information is saved in a file called *iwoevent.log*. The *iwoevent.log* file is created in the outbound folder where the *iwoevent.dll* and *iwoevent.cfg* files are located.

2.10 Modifying the JDE.INI File for Outbound and Inbound Processing

This section describes the settings that are required in the JDE.INI file for the XML call object kernel (outbound and inbound processing).

Open the JDE.INI file and modify the *[JDENET_KERNEL_DEF6]* and *[JDENET_KERNEL_DEF15]* sections as follows:

```ini
[JDENET_KERNEL_DEF6]
krnlName=CALL OBJECT KERNEL
dispatchDLLName=XMLCallObj.dll
dispatchDLLFunction=_XMLTransactionDispatch@28
maxNumberOfProcesses=1
numberOfAutoStartProcesses=1

[JDENET_KERNEL_DEF15]
krnlName=XML TRANSACTION KERNEL
dispatchDLLName=XMLTransactions.dll
dispatchDLLFunction=_XMLTransactionDispatch@28
maxNumberOfProcesses=1
numberOfAutoStartProcesses=1
```

The parameters containing an underscore (_) and @28 are for Windows NT operating systems only. For other operating systems, replace the parameters with the values in the following table:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Call Object dispatch DLLName</th>
<th>XML Trans dispatch DLLName</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS400</td>
<td>XMLCALLOBJ</td>
<td>XMLTRANS</td>
</tr>
<tr>
<td>HP9000B</td>
<td>libxmlcallobj.so</td>
<td>libxmltransactions.so</td>
</tr>
<tr>
<td>Sun or RS6000</td>
<td>libxmlcallobj.so</td>
<td>Libxmltransactions.so</td>
</tr>
</tbody>
</table>

**Note:** The J.D. Edwards OneWorld installation for version B7333(XE) does not include *[JDENET_KERNEL_DEF15]*. As a result, if you are using version B7333(XE), you must manually add it to the jde.ini file. For all other J.D. Edwards OneWorld versions, *[JDENET_KERNEL_DEF15]* is included with the installation.
This chapter describes Oracle WebLogic Server (OracleWLS) deployment and integration with Oracle Application Adapter for J.D. Edwards OneWorld. It contains the following topics:

- Section 3.1, "Adapter Integration with Oracle WebLogic Server"
- Section 3.2, "Deployment of Adapter"
- Section 3.3, "Updating Adapter Configuration"

See Also:
- Oracle Application Server Adapter Concepts Guide

### 3.1 Adapter Integration with Oracle WebLogic Server

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

See Also:
- Oracle Application Server Adapter Concepts Guide

### 3.2 Deployment of Adapter

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

During the J2CA deployment of Oracle Application Adapter for J.D. Edwards OneWorld, OracleWLS generates a deployment descriptor called ra.xml, located in:

```plaintext
<ADAPTER_HOME>\soa\thirdparty\ApplicationAdapters\iwafjca.rar\META-INF
```

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

---

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

---

### Creating a Managed Connector Factory Object

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

```xml
<?xml version="1.0" encoding='UTF-8'?>
<!DOCTYPE connector PUBLIC "+//Sun Microsystems, Inc./DTD Connector 1.0//EN" 
'http://java.sun.com/dtd/connector_1_0.dtd'>
<connector>
```

---

See Also:
- Oracle Application Server Adapter Concepts Guide
<display-name>IWAFJCA10</display-name>
<vendor-name>IWAY Software</vendor-name>
<spec-version>1.0</spec-version>
<eis-type>IWAF</eis-type>
<version>1.0</version>
<license>
  <license-required>false</license-required>
</license>
<resourceadapter>
  <managedconnectionfactory-class>com.ibi.afjca.spi.IWAFManagedConnectionFactory</managedconnectionfactory-class>
  <connectionfactory-interface>javax.resource.cci.ConnectionFactory</connectionfactory-interface>
  <connectionfactory-impl-class>com.ibi.afjca.cci.IWAFConnectionFactory</connectionfactory-impl-class>
  <connection-interface>javax.resource.cci.Connection</connection-interface>
  <connection-impl-class>com.ibi.afjca.cci.IWAFConnection</connection-impl-class>
  <transaction-support>NoTransaction</transaction-support>
  <config-property>
    <config-property-name>AdapterName</config-property-name>
    <config-property-type>java.lang.String</config-property-type>
    <config-property-value></config-property-value>
  </config-property>
  <config-property>
    <config-property-name>Config</config-property-name>
    <config-property-type>java.lang.String</config-property-type>
    <config-property-value></config-property-value>
  </config-property>
  <config-property>
    <config-property-name>IWayHome</config-property-name>
    <config-property-type>java.lang.String</config-property-type>
    <config-property-value>C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters</config-property-value>
  </config-property>
  <config-property>
    <config-property-name>IWayConfig</config-property-name>
    <config-property-type>java.lang.String</config-property-type>
    <config-property-value>jca_sample</config-property-value>
  </config-property>
  <config-property>
    <config-property-name>IWayRepoDriver</config-property-name>
    <config-property-type>java.lang.String</config-property-type>
    <config-property-value></config-property-value>
  </config-property>
  <config-property>
    <config-property-name>IWayRepoURL</config-property-name>
    <config-property-type>java.lang.String</config-property-type>
    <config-property-value></config-property-value>
  </config-property>
  <config-property>
    <config-property-name>IWayRepoUser</config-property-name>
    <config-property-type>java.lang.String</config-property-type>
    <config-property-value></config-property-value>
  </config-property>
  <config-property>
    <config-property-name>IWayRepoPassword</config-property-name>
  </config-property>
</resourceadapter>
<config-property-type>java.lang.String</config-property-type>
<config-property-value></config-property-value>
</config-property>
<config-property>
<config-property-name>LogLevel</config-property-name>
<config-property-type>java.lang.String</config-property-type>
<config-property-value>DEBUG</config-property-value>
</config-property>
<authentication-mechanism>
<authentication-mechanism-type>BasicPassword</authentication-mechanism-type>
</authentication-mechanism>
</credential-interface>
</authentication-mechanism>
</reauthentication-support>true</reauthentication-support>
</resourceadapter>
</connector>

The parameters defined in the 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 OracleWLS packaged application adapter.</td>
</tr>
<tr>
<td>IWayConfig</td>
<td>The adapter configuration name as defined in Application Explorer. For example, Oracle Application Adapter for J.D. Edwards OneWorld has a preconfigured jca_sample 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 repository.</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 repository.</td>
</tr>
<tr>
<td>IWayRepoPassword</td>
<td>Password. If provided, then it overwrites configuration. This is necessary only when using an Oracle database as the repository.</td>
</tr>
<tr>
<td>loglevel</td>
<td>It overwrites the level set by the ManagedConnectorFactory property.</td>
</tr>
</tbody>
</table>

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

<ADAPTER_HOME>/soa/thirdparty/ApplicationAdapters/iwafjca.rar/META-INF

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

```xml
<?xml version="1.0"?>
<weblogic-connector xmlns="http://www.bea.com/ns/weblogic/90">
  <enable-access-outside-app>true</enable-access-outside-app>
  <enable-global-access-to-classes>true</enable-global-access-to-classes>
  <outbound-resource-adapter>
  <default-connection-properties>
  <pool-params>
```
To create multiple managed connector factory objects, you must add new `<connection-instance>` nodes in the file. For example:

```xml
<?xml version="1.0"?>
<weblogic-connector xmlns="http://www.bea.com/ns/weblogic/90">
  <enable-access-outside-app>true</enable-access-outside-app>
  <enable-global-access-to-classes>true</enable-global-access-to-classes>

  <outbound-resource-adapter>
    <default-connection-properties>
      <pool-params>
        <initial-capacity>0</initial-capacity>
      </pool-params>
      <transaction-support>LocalTransaction</transaction-support>
    </default-connection-properties>
    <connection-definition-group>
      <connection-factory-interface>javax.resource.cci.ConnectionFactory</connection-factory-interface>
      <connection-instance>
        <jndi-name>eis/OracleJCAAdapter/DefaultConnection</jndi-name>
      </connection-instance>
    </connection-definition-group>
    <connection-definition-group>
      <connection-factory-interface>javax.resource.cci.ConnectionFactory</connection-factory-interface>
      <connection-instance>
        <jndi-name>eis/OracleJCAAdapter/DefaultConnection1</jndi-name>
      </connection-instance>
    </connection-definition-group>
  </outbound-resource-adapter>
</weblogic-connector>
```
If you do not specify a `<property>` element in the `<connection-instance>` section, then the value is taken from the `ra.xml` file. You can specify the default properties in the `ra.xml` file and then override them as required in the `weblogic-ra.xml` file. In addition, note that the J2CA configuration (for example, `jca_sample2`) must already be created in Application Explorer.

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

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

### Modifying WSDL Files for Additional Connection Factory Values

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

Notice that the J2CA properties file has the following default connection factory:

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

The connection factory value must be changed to the following:

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

For example:

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

Note that only the value for the location field in the `jca:address` section should be modified. Do not modify any other field or section.
Integration With BPEL Service Components in the Oracle SOA Suite

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

This chapter contains the following topics:

- Section 4.1, "Overview"
- Section 4.2, "Deployment of Adapter"
- Section 4.3, "Configuring a New Application Server Connection"
- Section 4.4, "Designing an Outbound BPEL Process for Service Integration"
- Section 4.5, "Designing an Inbound BPEL Process for Event Integration"

4.1 Overview

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

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 Oracle JDeveloper and deployed to a BPEL server. Upon deployment to the BPEL server, every newly built process is automatically deployed to the Oracle Enterprise Manager console, where you run, monitor, administer BPEL processes, and listen to adapter events.

4.2 Deployment of Adapter

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

See Also: Oracle Application Server Adapter Concepts Guide
4.3 Configuring a New Application Server Connection

To configure a new Application Server connection in Oracle JDeveloper:

1. Open Oracle JDeveloper on your system.
2. From the menu bar, click View and select Application Server Navigator, as shown in Figure 4–1.

Figure 4–1 Application Server Navigator

![Application Server Navigator](image)

The Application Server tab is displayed, as shown in Figure 4–2.

Figure 4–2 Application Server Tab

![Application Server Tab](image)

3. Right-click Application Servers, and then select New Application Server.
   The Create Application Server Connection Wizard is displayed, as shown in Figure 4–3.
4. Accept the default selection (Standalone Server) and click Next.

The Name and Type page is displayed, as shown in Figure 4–4.

**Figure 4–4 Name and Type Page**
5. Specify a new name for the Application Server connection and click **Next**.
   The Authentication page is displayed, as shown in **Figure 4–5**.

**Figure 4–5  Authentication Page**

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

7. Click **Next**.
   The Configuration page is displayed, as shown in **Figure 4–6**.
8. Specify the Oracle WebLogic host name (for example, localhost), which is the machine IP where the process needs to deploy and Oracle WebLogic domain (for example, base_domain).

9. Click Next.

The Test page is displayed, as shown in Figure 4–7.
10. Click Test Connection.

11. Ensure that the test status is successful.

12. Click Next.

   The Finish page is displayed, as shown in Figure 4–8.
13. Click Finish.

The new Application Server connection is listed in the left pane (Application Server tab), shown in Figure 4–9.

4.4 Designing an Outbound BPEL Process for Service Integration

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

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

Note: The examples in this chapter demonstrate the use of Oracle JDeveloper.
Before you design a BPEL process, you must generate WSDL using Application Explorer. For more information, see “Generating WSDL for Request/Response Service” on page 4-8. The WSDL generated in Application Explorer is used during the BPEL process configuration.

4.4.1 Generating WSDL for Request/Response Service

Perform the following steps to generate a WSDL for request/response service:

1. Start Application Explorer and connect to a defined J.D. Edwards OneWorld target.
   
   For more information, see “Defining a Target to J.D. Edwards OneWorld” on page 2-6.

2. Expand Services, CALLBSFN, and Addressbook.

   *Figure 4–10  Create Outbound JCA Service(Request/Response) Option*

3. Right-click GetEffectiveAddress, and then select Create Outbound JCA Service (Request/Response), as shown in Figure 4–10.

   The Export WSDL dialog is displayed, as shown in Figure 4–11.
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.

5. Click OK.

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

4.4.2 Creating a New SOA Application for the Outbound BPEL Process

Perform the following steps to create a new SOA application for the outbound BPEL process:

1. Open Oracle JDeveloper on your system.

2. Click Application in the menu bar and select New from the menu, as shown in Figure 4–12.
The Create SOA Application wizard is displayed, as shown in Figure 4-13.

**Figure 4-13  Create SOA Application Wizard**
3. From the Application Template list, click **SOA Application**.

4. Enter a name for the new SOA application (for example, JDE_Outbound_BPEL) and click **Next**.

   The Name your project page is displayed, as shown in **Figure 4–14**.

![Figure 4–14 Name Your Project Page](image)

5. Enter a project name (for example, GetEffectiveAddress_Invoke) and click **Next**.

   The Configure SOA settings page is displayed, as shown in **Figure 4–15**.
6. From the Composite Template list, select **Empty Composite** and click **Finish**.

The new SOA application (JDE_Outbound_BPEL) and associated project (GetEffectiveAddress_Invoke) are added to the Application Navigator tab in the left pane, as shown in Figure 4–16.

**4.4.3 Defining a BPEL Outbound Process**

This section describes how to define a BPEL outbound process, which consists of the following stages:
1. Configuring a Third Party Adapter Service Component

2. Configuring an Outbound BPEL Process Component

**Configuring a Third Party Adapter Service Component**

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

1. Drag and drop the Third Party Adapter component from the Component Palette tab (Service Adapters section) to the External References pane, as shown in Figure 4–17.

**Figure 4–17 Third Party Adapter Component**

![Third Party Adapter Component](image)

The Create Third Party Adapter Service dialog is displayed, as shown in Figure 4–18.

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

![Create Third Party Adapter Service Dialog](image)
2. Enter a name for the new third party adapter service.

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

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

   The SOA Resource Browser dialog is displayed, as shown in **Figure 4–19**.

**Figure 4–19  SOA Resource Browser Dialog**

5. Browse and select an outbound WSDL file (for example, GetEffectiveAddress_invoke.wsdl) from the following directory:

   `<ADAPTER_HOME>\soa\thirdparty\ApplicationAdapters\wsdls`

6. Click **OK**.

   The Localize Files dialog is displayed, as shown in **Figure 4–20**.
7. Click OK.

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

You are returned to the Create Third Party Adapter Service dialog, as shown in Figure 4–21.

Figure 4–21  Create Third Party Adapter Service Dialog
8. Click the **Find JCA Files** icon, which is located to the right of the JCA File field. The SOA Resource Browser dialog is displayed, as shown in *Figure 4–22*.

*Figure 4–22  SOA Resource Browser Dialog*

9. Browse and select the JCA properties file (for example, GetEffectiveAddress_invoke.jca) from the following directory:

   `<ADAPTER_HOME>\soa\thirdparty\ApplicationAdapters\wsdls`

10. Click **OK**.

    The Copy File message is displayed, as shown in *Figure 4–23*.

*Figure 4–23  Copy File Confirmation Message*

11. Click **Yes**.

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

    You are returned to the Create Third Party Adapter Service dialog, as shown in *Figure 4–24*. 
12. Click OK.

The third party adapter service component (GetEffectiveAddress) is created and displayed in the External References pane, as shown in **Figure 4–25**.

**Figure 4–25 Third Party Adapter Service Component**

You are now ready to configure an outbound BPEL process component.
Configuring an Outbound BPEL Process Component

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

1. Drag and drop the **BPEL Process** component from the Component Palette tab (Service Components section) to the Components pane, as shown in Figure 4–26.

![Figure 4–26 BPEL Process Component](image)

The Create BPEL Process dialog is displayed, as shown in Figure 4–27.

![Figure 4–27 Create BPEL Process Dialog](image)

2. In the Name field, enter a name to identify the new outbound BPEL process component (for example, GetEffectiveAddress_Out).

3. From the Template list, select **Synchronous BPEL Process**.

4. Click the **Browse Input Elements** icon, which is located to the right of the Input field to select the associated XML request schema file.

   The Type Chooser dialog is displayed, as shown in Figure 4–28.
5. Expand Project Schema Files, GetEffectiveAddress_invoke_request.xsd, and select jdeRequest.

6. Click OK.

You are returned to the Create BPEL Process dialog, as shown in Figure 4–29.

7. Click the Browse Output Elements icon, which is located to the right of the Output field to select the associated XML response schema file.
The Type Chooser dialog is displayed, as shown in Figure 4–30.

**Figure 4–30  Type Chooser Dialog**

8. Expand Project Schema Files, GetEffectiveAddress_invoke_response.xsd, and select jdeResponse.
9. Click OK.

You are returned to the Create BPEL Process dialog, as shown in Figure 4–31.

**Figure 4–31  Create BPEL Process Dialog**

10. Click OK.
11. Create a connection between the outbound BPEL process component (GetEffectiveAddress_Out) and the third party adapter service component (GetEffectiveAddress), as shown in Figure 4–32.

**Figure 4–32  Created Connection**

12. Double-click the outbound BPEL process component (GetEffectiveAddress_Out) in the Components pane, as shown in Figure 4–33.

**Figure 4–33  Outbound BPEL Process Component**

13. Drag and drop the **Invoke** activity component to the Components pane and place it between the **receiveInput** activity component and the **replyOutput** activity component, as shown in Figure 4–34.
14. Create a connection between the new Invoke activity component (Invoke_1) and the third party adapter service component (GetEffectiveAddress), as shown in Figure 4–35.

The Edit Invoke dialog is displayed, as shown in Figure 4–36.
15. Click the **Automatically Create Input Variable** icon, which is located to the right of the Input field to configure a new input variable.

   The Create Variable dialog is displayed, as shown in Figure 4–37.

**Figure 4–37  Create Variable Dialog**

   ![Create Variable Dialog](image)

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

   You are returned to the Edit Invoke dialog, as shown in Figure 4–38.

**Figure 4–38  Edit Invoke Dialog**

   ![Edit Invoke Dialog](image)
17. Click the **Automatically Create Output Variable** icon, which is located to the right of the Output field to configure a new output variable.

The Create Variable dialog is displayed, as shown in Figure 4–39.

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

You are returned to the Edit Invoke dialog, as shown in Figure 4–40.
19. Click **Apply** and then **OK**.

The Invoke activity component (Invoke_1) is updated accordingly, as shown in Figure 4–41.
20. Drag and drop the **Assign** activity component to the Components pane and place it between the Receive activity component (receiveInput) and the Invoke activity component (Invoke_1), as shown in Figure 4-42.

21. Double-click the new Assign activity component (**Assign_1**).
   The Assign dialog is displayed, as shown in Figure 4-43.
22. Click the Copy Operation tab.

23. Click the Plus sign icon and select Copy Operation from the list of available operations.

The Create Copy Operation dialog is displayed, as shown in Figure 4–44.

Figure 4–44 Create Copy Operation Dialog
24. In the From pane, expand Variables, InputVariable, and then select payload.

25. In the To pane, expand Variables, Invoke_1_GetEffectiveAddress_InputVariable, and then select input_GetEffectiveAddress.

26. Click OK.
   
   You are returned to the Assign dialog, as shown in Figure 4–45.

**Figure 4–45  Assign Dialog**

27. Click Apply and then OK.

28. Drag and drop the Assign activity component to the Components pane and place it between the Invoke activity (Invoke_1) and the Reply activity (replyOutput), as shown in Figure 4–46.
Double-click the new Assign activity component (Assign_2).

The Assign dialog is displayed, as shown in Figure 4–47.

Click the Copy Operation tab.

Click the Plus sign icon and select Copy Operation from the list of available operations.
The Create Copy Operation dialog is displayed, as shown in Figure 4–48.

**Figure 4–48 Create Copy Operation Dialog**

32. In the From pane, expand Variables, Invoke_1_GetEffectiveAddress_OutputVariable, and then select output_GetEffectiveAddress.

33. In the To pane, expand Variables, outputVariable, and then select payload.

34. Click OK.

You are returned to the Assign dialog, as shown in Figure 4–49.
35. Click **Apply** and then **OK**.

The completed activity flow is now displayed, as shown in **Figure 4–50**.
36. Double-click **composite.xml** in the left pane, as shown in Figure 4–51.

37. Click the **Save All** icon in the menu bar to save the new outbound BPEL process component that was configured, as shown in Figure 4–52.
You are now ready to deploy the BPEL outbound process.

4.4.4 Deploying the BPEL Outbound Process

Perform the following steps to deploy the BPEL outbound process.

1. Right-click the project name in the left pane (for example, GetEffectiveAddress_Invoke), select Deploy, and then click GetEffectiveAddress_Invoke, as shown in Figure 4–53.

The Deployment Action page is displayed, as shown in Figure 4–54.
2. Ensure that **Deploy to Application Server** is selected.

3. Click **Next**.

   The Deploy Configuration page is displayed, as shown in Figure 4–55.
4. Leave the default values selected and click **Next**.

The Select Server page is displayed, as shown in **Figure 4–56**.

**Figure 4–56 Select Server Page**
5. Select an available application server that was configured and click **Next**.
   
   The SOA Servers page is displayed, as shown in Figure 4–57.

*Figure 4–57  SOA Servers Page*

6. Select a target SOA server and click **Next**.
   
   The Summary page is displayed, as shown in Figure 4–58.
7. Review and verify all the available deployment information for your project and click Finish.

The process is deployed successfully, as shown in Figure 4–59.

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

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

1. Log in to the Oracle Enterprise Manager console by using the following URL:
   http://localhost:7001/em

2. Expand your domain in the left pane followed by the SOA folder.
3. Select an available project (for example, JDE_Outbound_GetEffectiveAddress_BPEL), as shown in Figure 4–60.

4. Click Test in the right pane, as shown in Figure 4–61. The Test Web Service page is displayed, as shown in Figure 4–62.
5. Click the Request tab.

6. Scroll down to the Input Arguments section, as shown in Figure 4–63.

7. Select XML View from the list in the upper-left corner.


For example:

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ns1:GetEffectiveAddress>
      <parameters>
        <param name="msAddressNumber">4742</param>
      </parameters>
    </ns1:GetEffectiveAddress>
  </soap:Body>
</soap:Envelope>
```
Designing an Inbound BPEL Process for Event Integration

4.4.6 Testing Outbound BPEL and Mediator Processes

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

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

4.5 Designing an Inbound BPEL Process for Event Integration

This section illustrates how Oracle Application Adapter for J.D. Edwards OneWorld integrates with J.D. Edwards OneWorld to receive event data. The design-time and run-time configuration procedures are outlined in the following sections.
The following tools are required to complete your adapter design-time configuration:

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

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

Before you design a BPEL process, you must generate the respective WSDL file using Application Explorer. For more information, see "Generating WSDL for Event Integration" on page 4-41.

### 4.5.1 Generating WSDL for Event Integration

Before you design a BPEL process using Oracle JDeveloper, you must create a separate channel for every J2CA 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, then event messages may not be delivered to the right BPEL process.

#### Creating a Channel in Application Explorer

To create a channel:

1. In Application Explorer, expand the **JDEdwards** node.
2. Right-click the **Channels** node, and select **Add Channels**.
   
   The Add Channel dialog is displayed, as shown in Figure 4–65.

   ![Add Channel Dialog](image)

   *Figure 4–65  Add Channel Dialog*

3. In the **Name** field, enter a descriptive name for the channel.
4. In the **Description** field, enter a description (optional).

5. From the **Protocol** list, choose a protocol for your channel.

6. Click **Next**.

   The dialog is displayed for the selected listener, as shown in **Figure 4–66**.

   **Figure 4–66 TCP Listener Dialog**

7. Enter the port number of the channel in the **Port Number** field.

8. Enter the location of the server in the **Host/IP Binding** field.

9. Select the Synchronization type from the **Synchronization Type** list.

10. Select **Is Length Prefix** for events that send data which is not in XML format. The TCP/IP event application must prefix the data with a 4-byte binary length field when writing the data to the TCP/IP port.

11. Select **Is XML** for events that send data back in XML format. No preparser is required.

12. Select **Is Keep Alive** to maintain a continuous communication between the event transaction and the channel.

13. Click the **PreParser** tab, as shown in **Figure 4–67**.
Enter values based on the table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User id*</td>
<td>A valid user ID for J.D. Edwards OneWorld.</td>
</tr>
<tr>
<td>User password*</td>
<td>The password associated with the user ID.</td>
</tr>
<tr>
<td>JDE environment*</td>
<td>Your J.D. Edwards OneWorld environment. For more information about this parameter, see your J.D. Edwards OneWorld documentation or ask your OneWorld system administrator.</td>
</tr>
<tr>
<td>Application</td>
<td>XMLInterop or the application name in J.D. Edwards OneWorld. Optional.</td>
</tr>
<tr>
<td>Server IP address*</td>
<td>The name of the server on which J.D. Edwards OneWorld is running. This can be the name of the server, for example, JDEOW, or its IP address, for example, 123.45.67.89.</td>
</tr>
<tr>
<td>Server Port*</td>
<td>The port number on which the server is listening, for example, 6009.</td>
</tr>
<tr>
<td>User Role</td>
<td>Define a user role according to your requirements.</td>
</tr>
<tr>
<td>Schema Location</td>
<td>The location of the XML schema (.xsd file) that was generated from the event output. For example:</td>
</tr>
<tr>
<td></td>
<td><code>&lt;ADAPTER_HOME&gt;/soa/thirdparty/ApplicationAdapters/config/config_name/target_name/jde-schema.xsd</code></td>
</tr>
<tr>
<td></td>
<td>For more information, see &quot;Generating WSDL for Inbound Interaction&quot; on page 2-13.</td>
</tr>
<tr>
<td>Schema style</td>
<td>Choose from one of the following options:</td>
</tr>
<tr>
<td></td>
<td>■ ELEMENT_STYLE (default)</td>
</tr>
<tr>
<td></td>
<td>■ ATTRIBUTE_STYLE</td>
</tr>
</tbody>
</table>
Click OK.

The channel is created and displayed under the Channels node. An X over the icon indicates that the channel is currently disconnected.

---

**Note:** The channel you created in Application Explorer is managed by BPEL PM Server. If you start the channel for testing and debugging purposes, then stop it before run-time.

---

**Generating WSDL for Event Notification (Command Prompt Only)**

You cannot generate WSDL for J.D. Edwards OneWorld event notification using Application Explorer. To generate WSDL from the command prompt, you must perform the following steps.

You can create inbound J2CA service only if the node you have selected supports events.

**Note:** The schema validation options (Root, Namespace, Schema) are not applicable for the Oracle Application Adapter for J.D. Edwards OneWorld.

To generate a WSDL file for J.D. Edwards OneWorld event notification:

1. Create a channel in Application Explorer under the J.D. Edwards events node.
2. Start the channel.
   
   **Important:** Do not restart the BPEL PM Server or Oracle Application Server after the channel is started.

3. Send an inbound message from J.D. Edwards OneWorld.
4. Capture the inbound message payload in the log file located under:

   `<ADAPTER_HOME>\soa\thirdparty\ApplicationAdapters\config\jca_sample\log\iwaf_jca1500.log`

   Alternatively, you can create a port with the File protocol under Event in Application Explorer to dispose the event message to the file system.

5. Use a third party tool such as XMLSpy to create the XSD schema using the XML payload captured in the previous step.

6. In the generated XML schema (.xsd file) perform the following modifications:
   
   a. Search for `Schemas-jdedwards-com` and replace it with `iwaysoftware`.
   
   b. Cut the following syntax:

   ```
   <xs:element name="jdeResponse">
   <xs:complexType>
   </xs:complexType>
   </xs:element>
   ```

   c. Paste it before the following line:

   ```
   <xs:element name="transaction">
   ```

7. Copy the XML schema (.xsd file) from the following directory:

   `<ADAPTER_HOME>\soa\thirdparty\ApplicationAdapters\config\config_`
8. Open a command prompt and navigate to the following directory:

<ADAPTER_HOME>\soa\thirdparty\ApplicationAdapters\tools\iwae\bin

9. Execute the obadapter.bat file to set the environment.

10. Navigate to the following directory where the XML schema (.xsd) file is copied:

<ADAPTER_HOME>\soa\thirdparty\ApplicationAdapters\config\config_name\schemas\JDEdwards\target_name

11. Enter the following command to generate a WSDL:

```
java -Diway.oem=oracle11g com.iwaysoftware.af.container.tools.wsd1.IWayWSILBrowser adapterhome adapter target channel schemaPrefix wsdlFileName
```

where:

- `adapterhome` is the path to your ApplicationAdapters home. For example:
  
  `<ADAPTER_HOME>\soa\thirdparty\ApplicationAdapters`

- `adapter` is the name of the adapter. For example, JDEdwards.

- `target` is the name of the adapter target you created in Application Explorer.

- `channel` is the name of the channel you created in Application Explorer.

- `schemaPrefix` is the prefix for the XSD schema. The schema file must be in the same directory where the Java command is executed, for example:

```
java -Diway.oem=oracle11g com.iwaysoftware.af.container.tools.wsd1.IWayWSILBrowser C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\wsdls\JDEdwards jde812_tgt jde_ch Jde812_Schema Jde812_salesorder_receive.wsdl
```

Once the command is executed, the following is displayed in the command window:

```
Running Inbound WSDL generation tool...
-> user.dir = java com.iwaysoftware.af.container.tools.wsd1.IWayWSILBrowser C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\wsdls\JDEdwards jde812_tgt jde_ch Jde812_Schema Jde812_salesorder_receive.wsdl

-> Generating WSDL...
-> Done.
-> Writing WSDL 'C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\wsdls\Jde812_salesorder_receive.wsdl' to disk...
```
12. Stop the channel in Application Explorer.

**Note:** You can organize your WSDL files in subfolders, creating your own WSDL hierarchy structure. Create the folders under `<ADAPTER_HOME>/soa/thirdparty/ApplicationAdapters/wsdls`. The WSIL browser in Oracle JDeveloper displays the full tree structure of your WSDL hierarchy.

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

### 4.5.2 Creating a New SOA Application for the Inbound BPEL Process

Perform the following steps to create a new SOA application for the inbound BPEL process:

1. Open Oracle JDeveloper on your system.
2. Click **Application** in the menu bar and select **New** from the menu, as shown in Figure 4–68.
The Create SOA Application wizard is displayed, as shown in Figure 4–69.

**Figure 4–69** Create SOA Application Wizard
3. From the Application Template list, click **SOA Application**.

4. Enter a name for the new SOA application (for example, JDE_Inbound_BPEL) and click **Next**.

   The Name your project page is displayed, as shown in Figure 4–70.

**Figure 4–70 Name Your Project Page**

5. Enter a project name (for example, Transactions_Receive) and click **Next**.

   The Configure SOA settings page is displayed, as shown in Figure 4–71.
6. From the Composite Template list, select **Empty Composite** and click **Finish**.

The new SOA application (JDE_Inbound_BPEL) and associated project (Transactions_Receive) are added to the Application Navigator tab in the left pane, as shown in Figure 4–72.

### 4.5.3 Defining a BPEL Inbound Process

This section describes how to define a BPEL inbound process, which consists of the following stages:
1. Configuring a Third Party Adapter Service Component
2. Configuring an Inbound BPEL Process Component

Creating a Third Party Adapter Service Component

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

1. Drag and drop the Third Party Adapter component from the Component Palette tab (Service Adapters section) to the Exposed Services pane, as shown in Figure 4–73.

Figure 4–73  Third Party Adapter Component

The Create Third Party Adapter Service dialog is displayed, as shown in Figure 4–74.

Figure 4–74  Create Third Party Adapter Service Dialog

2. Enter a name for the third party adapter service.
3. Ensure that Service is selected from the Type list (default).
4. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.

The SOA Resource Browser dialog is displayed, as shown in Figure 4–75.

*Figure 4–75  SOA Resource Browser Dialog*

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

   `<ADAPTER_HOME>\soa\thirdparty\ApplicationAdapters\wsdls`

6. Click **OK**.

The Localize Files dialog is displayed, as shown in Figure 4–76.
7. Click OK.

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

You are returned to the Create Third Party Adapter Service dialog, as shown in Figure 4–77.

Figure 4–77  Create Third Party Adapter Service Dialog
8. Click the Find JCA Files icon, which is located to the right of the JCA File field. The SOA Resource Browser dialog is displayed, as shown in Figure 4–78.

*Figure 4–78 SOA Resource Browser Dialog*

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

   `<ADAPTER_HOME>`\soa\thirdparty\ApplicationAdapters\wsdl\*

10. Click OK.

    The Copy File message is displayed, as shown in Figure 4–79.

*Figure 4–79 Copy File Confirmation Message*

11. Click Yes.

    A copy of the JCA properties file is made in the project folder.
You are returned to the Create Third Party Adapter Service dialog, as shown in Figure 4–80.

**Figure 4–80  Create Third Party Adapter Service Dialog**

![Create Third Party Adapter Service Dialog](image)

12. Click OK.

The third party adapter service component (transactions) is created and displayed in the Exposed Services pane, as shown in Figure 4–81.

**Figure 4–81  Third Party Adapter Service Component**

![Exposed Services](image)
You are now ready to configure an inbound BPEL process component.

Creating an Inbound BPEL Process Component

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

1. Drag and drop the BPEL Process component from the Component Palette tab (Service Components section) to the Components pane, as shown in Figure 4–82.

2. In the Name field, enter a name to identify the new inbound BPEL process component (for example, Transactions_Inbound).

3. From the Template list, select Base on a WSDL.

4. Uncheck the Expose as a SOAP Service check box.
5. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.

The SOA Resource Browser dialog is displayed, as shown in **Figure 4-84**.

**Figure 4–84  SOA Resource Browser Dialog**

6. Browse and select an inbound WSDL file (for example, transactions_receive.wsdl) from the project folder.

7. Click **OK**.

The Localize Files dialog is displayed, as shown in **Figure 4–85**.
8. Click OK.

You are returned to the Create BPEL Process dialog, as shown in Figure 4–86.

9. Click OK.
10. Create a connection between the third party adapter service component (transactions) and the inbound BPEL process component (Transactions_Inbound), as shown in Figure 4–87.

11. Double-click composite.xml in the left pane, as shown in Figure 4–88.

12. Click the Save All icon in the menu bar to save the new inbound BPEL process component that was configured, as shown in Figure 4–89.
4.5.4 Deploying the BPEL Inbound Process

Perform the following steps to deploy the BPEL inbound process.

1. Right-click the project name in the left pane (for example, Transactions_Receive), select Deploy, and then click Transactions_Receive, as shown in Figure 4–90.

The Deployment Action page is displayed, as shown in Figure 4–91.

You are now ready to deploy the BPEL inbound process.
2. Ensure that **Deploy to Application Server** is selected.

3. Click Next.

The Deploy Configuration page is displayed, as shown in Figure 4–92.
4. Leave the default values selected and click **Next**.

The Select Server page is displayed, as shown in **Figure 4–93**.

**Figure 4–93  Select Server Page**
5. Select an available application server that was configured and click **Next**.
   The SOA Servers page is displayed, as shown in **Figure 4–94**.

**Figure 4–94  SOA Servers Page**

6. Select a target SOA server and click **Next**.
   The Summary page is displayed, as shown in **Figure 4–95**.
7. Review and verify all the available deployment information for your project and click Finish.

Once event messages are triggered through J.D. Edwards OneWorld, successful instances are received in the Oracle Enterprise Manager console.

4.5.5 Triggering an Event in J.D. Edwards OneWorld

Events are generated by activity in a database or in an application system. You can use events to trigger an action in your application. To trigger an event in J.D. Edwards OneWorld:

1. Log in to your J.D. Edwards OneWorld system.
2. In the Fast Path field of the J.D. Edwards OneWorld Explorer window, type G4211 and press Enter, as shown in Figure 4–96.

Figure 4–96 J.D. Edwards OneWorld Explorer Window
3. Right-click Sales Order Detail (P4210).

![Figure 4-97 Sales Order Detail Menu](image)

Figure 4–97  Sales Order Detail Menu

4. Select Prompt for, and then Values, as shown in Figure 4–97.

   The Processing Options dialog is displayed, as shown in Figure 4–98.

![Figure 4-98 Processing Options Dialog](image)

Figure 4–98  Processing Options Dialog

Perform the following steps:

a. Click the Interop tab.

b. In the Transaction Type field, type JDESOOUT.

c. Verify that the value in the Before/After Image Processing Blank field is 1.

5. Click OK.

   The Sales Order Detail - (Customer Service Inquiry) window is displayed, as shown in Figure 4–99.

![Figure 4-99 Sales Order Detail - (Customer Service Inquiry)](image)
6. Click the **Add** icon (third icon from left).

7. Enter the values as shown in **Figure 4–100**.

   To move to a different field, use the **Tab** key on your keyboard.

**Figure 4–100 Values**

8. Enter a value for **Quantity Ordered** and **Item Number**, as shown in **Figure 4–101**.
9. Click the first field in the second row and allow a few seconds for processing, as shown in Figure 4–102.

10. Click OK.

An event is triggered in the J.D. Edwards OneWorld system.

Verifying the Results
To verify your results:
1. Log in to the Oracle Enterprise Manager console by using the following URL:
   http://localhost:7001/em
2. Expand your domain in the left pane followed by the SOA folder.
3. Select an available project (for example, JDE_Inbound_salesorder_BPEL), as shown in Figure 4–103.

4. Click the **Instances** tab in the right pane, as shown in Figure 4–104.

Recently received run-time events are displayed in the Instances tab, as shown in Figure 4–105.

5. Select a J.D. Edwards OneWorld instance ID.
The Flow Trace page is displayed, as shown in Figure 4–106.

**Figure 4–106  Flow Trace Page**

Flow Trace 
This page shows the flow of the message through various composites and component instances.

<table>
<thead>
<tr>
<th>Faults (0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faults</td>
</tr>
<tr>
<td>Select a fault to locate it in the trace view.</td>
</tr>
<tr>
<td>Error Message</td>
</tr>
<tr>
<td>No faults found</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sensors (0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trace</td>
</tr>
<tr>
<td>Click a component instance to see its detailed audit trail.</td>
</tr>
<tr>
<td>Show Instance IDs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Instance</th>
<th>Type</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service</td>
<td>Service</td>
<td>Completed</td>
</tr>
<tr>
<td>JDE Inbound SaleOrder BPEL</td>
<td>BPEL Component</td>
<td>Completed</td>
</tr>
</tbody>
</table>

6. Select a component instance to view its detailed audit trail.

The Instance page for the selected component is displayed, as shown in Figure 4–107.
7. Click the **Audit Trail** tab to view the event message.

The message received from the J.D. Edwards OneWorld system is displayed in the Audit Trail tab.
This chapter describes integration with Mediator service components in the Oracle SOA Suite. It contains the following topics:

- Section 5.1, "Configuring a New Application Server Connection"
- Section 5.2, "Configuring a Mediator Outbound Process"
- Section 5.3, "Configuring a Mediator Inbound Process"

The scenarios shown in this chapter require the following prerequisites.

**Prerequisites**

The following are installation and configuration requirements:

- Oracle Application Adapter for J.D. Edwards OneWorld must be installed on Oracle WebLogic Server.
- J.D. Edwards OneWorld must be configured for inbound and outbound processing.

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

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

- How to configure Oracle Application Adapter for J.D. Edwards OneWorld for services and events. For more information, see Chapter 2, "Configuring Oracle Application Adapter for J.D. Edwards OneWorld".
- How to configure Oracle JDeveloper. For more information, see Chapter 4, "Integration With BPEL Service Components in the Oracle SOA Suite".

**Overview of Mediator Integration**

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

Adapter integration with Oracle WebLogic Server, Mediator is a two-step process:
1. **Design Time:** Oracle Application Adapter for J.D. Edwards OneWorld is configured in Application Explorer for services and events, as described in Chapter 2, "Configuring Oracle Application Adapter for J.D. Edwards OneWorld". 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.

### 5.1 Configuring a New Application Server Connection

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

### 5.2 Configuring a Mediator Outbound Process

The following example describes how to configure a Mediator outbound process to your J.D. Edwards OneWorld system, using a Mediator project in Oracle JDeveloper.

**Prerequisites**

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

#### 5.2.1 Creating a New SOA Application for the Outbound Mediator Process

Perform the following steps to create a new SOA application for the outbound Mediator process:

1. Open Oracle JDeveloper on your system.
2. Click **Application** in the menu bar and select **New** from the menu, as shown in Figure 5–1.
The Create SOA Application wizard is displayed, as shown in Figure 5–2.
3. From the Application Template list, click **SOA Application**.

4. Enter a name for the new SOA application (for example, JDE_Outbound_Mediator) and click **Next**.

   The Name your project page is displayed, as shown in **Figure 5–3**.

   **Figure 5–3 Name Your Project Page**

5. Enter a project name (for example, GetEffectiveAddress_Invoke) and click **Next**.

   The Configure SOA settings page is displayed, as shown in **Figure 5–4**.
6. From the Composite Template list, select **Empty Composite** and click **Finish** as shown in **Figure 5–5**.

The new SOA application (JDE_Outbound_Mediator) and associated project (GetEffectiveAddress_Invoke) are added to the Application Navigator tab in the left pane.
5.2.2 Defining a Mediator Outbound Process

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

1. Configuring a Third Party Adapter Service Component
2. Configuring an Outbound Mediator Process Component
3. Configuring the Routing Rules

Configuring a Third Party Adapter Service Component

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

1. Drag and drop the Third Party Adapter component from the Component Palette tab (Service Adapters section) to the External References pane, as shown in Figure 5–6.

Figure 5–6 Third Party Adapter

The Create Third Party Adapter Service dialog is displayed, as shown in Figure 5–7.
2. Enter a name for the new third party adapter service.

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

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

   The SOA Resource Browser dialog is displayed, as shown in Figure 5–8.

**Figure 5–8 SOA Resource Browser Dialog**
5. Browse and select an outbound WSDL file (for example, GetEffectiveAddress_invoke.wsdl) from the following directory:

<ADAPTER_HOME>\soa\thirdparty\ApplicationAdapters\wsdls

6. Click **OK**.

   The Localize Files dialog is displayed, as shown in Figure 5–9.

**Figure 5–9  Localize Files Dialog**

7. Click **OK**.

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

   You are returned to the Create Third Party Adapter Service dialog, as shown in Figure 5–10.
8. Click the **Find JCA Files** icon, which is located to the right of the JCA File field. The SOA Resource Browser dialog is displayed, as shown in **Figure 5–11**.

![Figure 5–10  Create Third Party Adapter Service Dialog](image)

**Figure 5–11  SOA Resource Browser Dialog**

![Figure 5–11  SOA Resource Browser Dialog](image)

9. Browse and select the JCA properties file (for example, GetEffectiveAddress_invoke.jca) from the following directory:
10. Click OK.

The following message is displayed, as shown in Figure 5–12.

**Figure 5–12 Copy File Confirmation Message**

The jca file GetEffectiveAddress_invoke.jca is external to the current project. In order to make this file available to your project at runtime, JDeveloper can now make a local copy. Do you want to copy file to GetEffectiveAddress_invoke_3P.jca?

[Yes No]

11. Click Yes.

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

You are returned to the Create Third Party Adapter Service dialog, as shown in Figure 5–13.

**Figure 5–13 Create Third Party Adapter Service Dialog**

12. Click OK.

The third party adapter service component (GetEffectiveAddress) is created and displayed in the External References pane, as shown in Figure 5–14.
You are now ready to configure an outbound Mediator process component.

**Configuring an Outbound Mediator Process Component**

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

1. Drag and drop the **Mediator** component from the Component Palette tab (Service Components section) to the Components pane as shown in Figure 5–15.

The Create Mediator dialog is displayed, as shown in Figure 5–16.
2. In the Name field, enter a name to identify the new outbound Mediator process component (for example, GetEffectiveAddress_Mediator).

3. From the Template list, select **Synchronous Interface**.

4. Click the **Browse Input Elements** icon, which is located to the right of the Input field to select the associated XML request schema file.

   The Type Chooser dialog is displayed, as shown in **Figure 5–17**.
5. Expand Project WSDL Files, GetEffectiveAddress_invoke.wsdl, Inline Schemas, schema, and select jdeRequest.

6. Click OK.

You are returned to the Create Mediator dialog, as shown in Figure 5–18.

Figure 5–18 Create Mediator Dialog

[Diagram of Create Mediator dialog showing fields for Name, Template, Input, and Output with values provided]
7. Click the **Browse Output Elements** icon, which is located to the right of the Output field to select the associated XML response schema file.

The Type Chooser dialog is displayed, as shown in **Figure 5–19**.

**Figure 5–19  Type Chooser Dialog**

8. Expand **Project WSDL Files**, **GetEffectiveAddress_invoke.wsdl**, **Inline Schemas**, **schema**, and select **jdeResponse**.

9. Click **OK**.

You are returned to the Create Mediator dialog, as shown in **Figure 5–20**.
10. Click OK.

11. Create a connection between the outbound Mediator process component (GetEffectiveAddress_Mediator) and the third party adapter service component (GetEffectiveAddress), as shown in Figure 5–21.

You are now ready to configure the routing rules.
Configuring the Routing Rules
Perform the following steps to configure routing rules for the Mediator outbound process component:

1. Double-click the outbound Mediator process component (GetEffectiveAddress_Mediator) in the Components pane, as shown in Figure 5–22.

![Outbound Mediator Process Component](image)

The Routing Rules dialog is displayed, as shown in Figure 5–23.

![Routing Rules Dialog](image)

2. In the <<Filter Expression>> area, click the Select an existing mapper file or create a new one icon to the right of the Transform Using field.

The Request Transformation Map dialog is displayed, as shown in Figure 5–24.
3. Select the Create New Mapper File option and click OK.
   The jdeRequest_To_jdeRequest.xsl tab is displayed, as shown in Figure 5–25.

   The Auto Map Preferences dialog is displayed, as shown in Figure 5–26.
5. Retain the default values and click OK.
6. Click the GetEffectiveAddress_Mediator.mplan tab.

You are returned to the Routing Rules dialog, as shown in Figure 5–27.

Figure 5–27 Routing Rules Dialog

7. In the Synchronous Reply area, click the Select an existing mapper file or create a new one icon to the right of the Transform Using field.

The Reply Transformation Map dialog is displayed, as shown in Figure 5–28.

Figure 5–28 Reply Transformation Map Dialog

8. Select the Create New Mapper File option and click OK.
The jdeResponse_To_jdeResponse.xsl tab is displayed, as shown in Figure 5–29.


The Auto Map Preferences dialog is displayed, as shown in Figure 5–30.

10. Retain the default values and click OK.

The mapping is completed, as shown in Figure 5–31.
11. Click the **Save All** icon in the menu bar to save the new outbound Mediator process component that was configured, as shown in Figure 5–32.

You are now ready to deploy the Mediator outbound process.

### 5.2.3 Deploying the Mediator Outbound Process

Perform the following steps to deploy the Mediator outbound process.

1. Right-click the project name in the left pane (for example, *GetEffectiveAddress_Invoke*), select **Deploy**, and then click *GetEffectiveAddress_Invoke*, as shown in Figure 5–33.
2. Ensure that **Deploy to Application Server** is selected.

3. Click **Next**.

   The Deploy Configuration page is displayed, as shown in Figure 5–35.
4. Leave the default values selected and click **Next**.

The Select Server page is displayed, as shown in **Figure 5–36**.

**Figure 5–36  Select Server Page**
5. Select an available application server that was configured and click Next.
   The SOA Servers page is displayed, as shown in Figure 5–37.

*Figure 5–37  SOA Servers Page*

6. Select a target SOA server and click Next.
   The Summary page is displayed, as shown in Figure 5–38.
7. Review and verify all the available deployment information for your project and click Finish.

The process is deployed successfully, as shown in Figure 5–39.

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

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

1. Log in to the Oracle Enterprise Manager console by using the following URL:
   http://localhost:7001/em

2. Expand your domain in the left pane followed by the SOA folder, as shown in Figure 5–40.
3. Select an available project (for example, JDE_Outbound_GetEffectiveAddress_Mediator), as shown in Figure 5–41.

4. Click **Test** in the right pane.
   
The Test Web Service page is displayed, as shown in Figure 5–42.
5. Click the **Request** tab.

6. Scroll down to the Input Arguments section, as shown in **Figure 5–43**.

**Figure 5–43  Input Arguments Section**

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body xmlns:ns="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetEffectiveAddresses">
    <ns:isRequest sessionid="" type="callMethod" session=""/>
    <ns:request target="returnAddressList=yes" name="GetEffectiveAddresses" runOn=notor=yes">
      <ns:parameters>
        <ns:parameter name="returnAddressNumber">4242</ns:parameter>
      </ns:parameters>
      <ns:Error abort="yes"/>
      <ns:callMethod/>
    </ns:request>
  </soap:Body>
</soap:Envelope>
```

7. Select **XML View** from the list in the upper-left corner.


   For example:

   ```xml
   <soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
     <soap:Body xmlns:ns="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetEffectiveAddresses"/>
   </soap:Envelope>
   ```
9. Click Test Web Service.

The output response is received in the Response tab of the Oracle Enterprise Manager console, as shown in Figure 5–44.

Figure 5–44 Oracle Enterprise Manager Console Response Tab

5.3 Configuring a Mediator Inbound Process

The following example describes how to configure a Mediator inbound process to your J.D. Edwards OneWorld system, using a Mediator project in Oracle JDeveloper.

Prerequisites

Before you design a Mediator inbound process, you must generate the respective WSDL file using Application Explorer. For more information, see “Generating WSDL for Event Integration” on page 4-41.

5.3.1 Creating a New SOA Application for the Inbound Mediator Process

Perform the following steps to create a new SOA application for the inbound Mediator process:

1. Open Oracle JDeveloper on your system.

2. Click Application in the menu bar and select New from the menu, as shown in Figure 5–45.
The Create SOA Application wizard is displayed, as shown in Figure 5–46.
3. From the Application Template list, click **SOA Application**.

4. Enter a name for the new SOA application (for example, JDE_Inbound_Mediator) and click **Next**.

   The Name your project page is displayed, as shown in Figure 5–47.

**Figure 5–47  Name Your Project Page**

5. Enter a project name (for example, Transactions_Receive) and click **Next**.

   The Configure SOA settings page is displayed, as shown in Figure 5–48.
6. From the Composite Template list, select **Empty Composite** and click **Finish**, as shown in **Figure 5–49**.

The new SOA application (JDE_Inbound_Mediator) and associated project (Transactions_Receive) are added to the Application Navigator tab in the left pane.
5.3.2 Defining a Mediator Inbound Process

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

1. Configuring a Third Party Adapter Service Component
2. Configuring an Inbound Mediator Process Component With a File Adapter
3. Configuring the Routing Rules

Configuring a Third Party Adapter Service Component

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

1. Drag and drop the Third Party Adapter component from the Component Palette tab (Service Adapters section) to the Exposed Services pane, as shown in Figure 5–50.

Figure 5–50 Third Party Adapter Component

The Create Third Party Adapter Service dialog is displayed, as shown in Figure 5–51.
2. Enter a name for the third party adapter service.
3. Ensure that Service is selected from the Type list (default).
4. Click the Find existing WSDLs icon, which is located to the right of the WSDL URL field.
   
The SOA Resource Browser dialog is displayed, as shown in Figure 5–52.
5. Browse and select an inbound WSDL file from the following directory:

   \<ADAPTER_HOME>\soa\thirdparty\ApplicationAdapters\wsdls

6. Click **OK**.

   The Localize Files dialog is displayed, as shown in **Figure 5–53**.
7. Click **OK**.

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

You are returned to the Create Third Party Adapter Service dialog, as shown in Figure 5–54.

**Figure 5–53  Localize Files Dialog**

![Localize Files Dialog](image)

Configuring a Mediator Inbound Process

**Figure 5–54  Create Third Party Adapter Service Dialog**

![Create Third Party Adapter Service Dialog](image)
8. Click the **Find JCA Files** icon, which is located to the right of the JCA File field. The SOA Resource Browser dialog is displayed, as shown in **Figure 5–55**.

**Figure 5–55  SOA Resource Browser Dialog**

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

   `<ADAPTER_HOME>\soa\thirdparty\ApplicationAdapters\wsdls`

10. Click **OK**.
    
    The following message is displayed, as shown in **Figure 5–56**.

**Figure 5–56  Copy File Confirmation Message**

11. Click **Yes**.
    
    A copy of the JCA properties file is made in the project folder.
You are returned to the Create Third Party Adapter Service dialog, as shown in Figure 5–57.

**Figure 5–57  Create Third Party Adapter Service Dialog**

![Create Third Party Adapter Service Dialog](image)

12. Click **OK**.

The third party adapter service component (transactions) is created and displayed in the Exposed Services pane, as shown in Figure 5–58.

**Figure 5–58  Exposed Services Pane**

![Exposed Services Pane](image)
You are now ready to configure an inbound Mediator process component.

**Configuring an Inbound Mediator Process Component With a File Adapter**

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

1. Drag and drop the Mediator component from the Component Palette tab (Service Components section) to the Components pane, as shown in Figure 5–59.

   ![Figure 5–59 Mediator Component](image)

   The Create Mediator dialog is displayed, as shown in Figure 5–60.

   ![Figure 5–60 Create Mediator Dialog](image)

2. In the Name field, enter a name to identify the new inbound Mediator process component (for example, Transactions_Mediator).

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

   The new Mediator process component is added to the Components pane, as shown in Figure 5–61.

![Mediator Process Component](image)

**Figure 5–61 Mediator Process Component**

5. Drag and drop the **File Adapter** component from the Service Adapters pane to the External References pane.

   The Adapter Configuration Wizard is displayed, showing the Welcome page, as shown in Figure 5–62.

![Adapter Configuration Wizard Welcome page](image)

**Figure 5–62 Adapter Configuration Wizard Welcome page**

6. Click **Next**.
The Service Name page is displayed, as shown in Figure 5–63.

**Figure 5–63  Service Name Change**

![Service Name Change](image)

7. Type a name for the new File adapter in the Service Name field and click **Next**. The Adapter Interface page is displayed, as shown in Figure 5–64.
8. Ensure that the **Define from operation and schema (specified later)** option is selected.

9. Click **Next**.

   The Operation page is displayed, as shown in **Figure 5–65**.
10. Select **Write File** from the list of Operation Type options and specify an Operation Name (for example, Write).

11. Click **Next**.

   The File Configuration page is displayed, as shown in **Figure 5–66**.
12. Specify a location on your file system where the output file is written.

13. In the File Naming Convention field, specify a name for the output file.

14. Click Next.

The Messages page is displayed, as shown in Figure 5–67.
Configuring a Mediator Inbound Process

Integration With Mediator Service Components in the Oracle SOA Suite

15. Click **browse for schema file**, which is located to the right of the URL field. The Type Chooser dialog is displayed, as shown in **Figure 5–68**.

**Figure 5–68 Type Chooser Dialog**
16. Select the available schema.

17. Click OK.

You are returned to the Messages page, as shown in Figure 5–69.

**Figure 5–69 Messages Page**

18. Click Next.

The Finish page is displayed, as shown in Figure 5–70.
19. Click Finish.

20. Create a connection between the inbound Mediator process component and the third party adapter service component, as shown in Figure 5–71.

Figure 5–71 Connection Creation

21. Create a connection between the inbound Mediator process component and the File adapter component, as shown in Figure 5–72.
Configuring the Routing Rules
Perform the following steps to configure routing rules for the Mediator inbound process component:

1. Double-click the inbound Mediator process component in the Components pane, as shown in Figure 5–73.

The Routing Rules dialog is displayed, as shown in Figure 5–74.

2. In the <<Filter Expression>> area, click the Select an existing mapper file or create a new one icon to the right of the Transform Using field.
The Request Transformation Map dialog is displayed, as shown in Figure 5–75.

### Figure 5–75 Request Transformation Map Dialog

![Request Transformation Map Dialog](image)

3. Select the **Create New Mapper File** option and click **OK**.

The jdeResponse_To_jdeResponse.xsl tab is displayed, as shown in Figure 5–76.

### Figure 5–76 jdeResponse_To_jdeResponse.xsl Tab

![jdeResponse_To_jdeResponse.xsl Tab](image)


The Auto Map Preferences dialog is displayed, as shown in Figure 5–77.
5. Retain the default values and click **OK**.

The mapping is completed, as shown in Figure 5–78.

5–78 Completed Mapping

6. Click the **Save All** icon in the menu bar to save the new inbound Mediator process component that was configured, as shown in Figure 5–79.
You are now ready to deploy the Mediator inbound process. You can follow the same procedure that is described in "Deploying the BPEL Inbound Process" on page 4-59.

Once event messages are triggered through J.D. Edwards OneWorld, output XML is received in the location that was specified for the File adapter component. For more information on triggering events in J.D. Edwards OneWorld, see "Triggering an Event in J.D. Edwards OneWorld" on page 4-63.
Integration With BPM Service Components in the Oracle SOA Suite

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

This chapter contains the following topics:

- Section 6.1, "Overview"
- Section 6.2, "Deployment of Adapter"
- Section 6.3, "Configuring a New Application Server Connection"
- Section 6.4, "Designing an Outbound BPM Process Using Transformations for Service Integration"
- Section 6.5, "Designing an Inbound BPM Process Using Transformations for Event Integration"

6.1 Overview

To integrate with Oracle BPM, Oracle Application Adapter for J.D. Edwards OneWorld must be deployed in the same WLS container as Oracle BPM. The underlying adapter services must be exposed as WSDL files, which are generated during design time in Oracle Adapter Application Explorer (Application Explorer) for both request-response (outbound) and event notification (inbound) services of the adapter. For more information, see "Generating WSDL (J2CA Configurations Only)" on page 2-12.

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

6.2 Deployment of Adapter

During installation, Oracle Application Adapter for J.D. Edwards OneWorld is deployed as a J2CA 1.0 resource adapter within the WLS container. The adapter must be deployed in the same WLS container as Oracle BPM.
6.3 Configuring a New Application Server Connection

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

6.4 Designing an Outbound BPM Process Using Transformations for Service Integration

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

- Oracle Adapter Application Explorer (Application Explorer)
- Oracle JDeveloper BPM Designer (JDeveloper) or Eclipse

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

Before you design a BPM process, you must generate the respective WSDL file using Application Explorer. For more information, see "Generating WSDL for Request/Response Service" on page 4-8.

6.4.1 Creating an Empty Composite for BPM

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

1. Create a new BPM application.
2. Enter a name for the new BPM application and click Next, as shown in Figure 6-1.
3. Enter a project name (for example, JDE_JCA_Outbound) and click Next.
The Configure SOA settings page is displayed, as shown in Figure 6–3.

**Figure 6–3 Configure SOA Settings Page**

![Configure SOA Settings Page](image)

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

### 6.4.2 Defining a BPM Outbound Process

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

1. Configuring a Third Party Adapter Service Component
2. Configuring an Outbound BPM Process Component
3. Creating a File Adapter for the Write Operation

**Configuring a Third Party Adapter Service Component**

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

1. Drag and drop the Third Party Adapter component from the Service Adapters pane to the External References pane, as shown in Figure 6–4.
The Create Third Party Adapter Service dialog is displayed, as shown in Figure 6–5.

Figure 6–5  Create Third Party Adapter Service Dialog

- Enter a name for the third party adapter service.
- Ensure that Reference is selected from the Type list (default).
- Click the Find existing WSDLs icon, which is located to the right of the WSDL URL field.

The SOA Resource Browser dialog is displayed, as shown in Figure 6–6.
5. Browse and select an outbound WSDL file from the following directory:
<ADAPTER_HOME>/soa/thirdparty/ApplicationAdapters/wsdls

6. Click OK.
   The Localize Files dialog is displayed, as shown in Figure 6–7.
7. Click OK.

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

You are returned to the Create Third Party Adapter Service dialog, as shown in Figure 6–8.

Figure 6–8  Create Third Party Adapter Service Dialog
8. Click the **Find JCA file** icon, which is located to the right of the JCA File field. The SOA Resource Browser dialog is displayed, as shown in **Figure 6–9**.

**Figure 6–9  SOA Resource Browser Dialog**

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

   `<ADAPTER_HOME>/soa/thirdparty/ApplicationAdapters/wsdl`

10. Click **OK**.

    The following message is displayed, as shown in **Figure 6–10**.

**Figure 6–10  Copy File Confirmation Message**

11. Click **Yes**.
A copy of the JCA properties file is made in the project folder.
You are returned to the Create Third Party Adapter Service dialog, as shown in Figure 6–11.

**Figure 6–11 Create Third Party Adapter Service Dialog**

![Create Third Party Adapter Service Dialog](image)

12. Click OK.

The third party adapter service component (CC_GD) is created and displayed in the External References pane, as shown in Figure 6–12.

**Figure 6–12 Third Party Adapter Service Component**

![Third Party Adapter Service Component](image)

You are now ready to configure an outbound BPM process component.
Configuring an Outbound BPM Process Component

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

1. Drag and drop the BPMN Process component from the Service Components pane to the Components pane, as shown in Figure 6–13.

![Figure 6–13 BPMN Process Component](image)

The Create BPMN Process dialog is displayed, as shown in Figure 6–14.

![Figure 6–14 Create BPMN Process Dialog](image)

2. Accept the default option that is selected in the left pane (Default Process) and click Finish.

The BPMN Process component is created, as shown in Figure 6–15.
3. Double-click the BPMN Process component in the Components pane. The BPMN process is displayed, as shown in Figure 6–16.

4. Click the Activity icon, as shown in Figure 6–17.
5. Drop the Activity icon on the wire between the Start and End event components, as shown in Figure 6–18.

*Figure 6–18  Activity Icon*

The Properties - Activity dialog is displayed, as shown in Figure 6–19.

*Figure 6–19  Properties - Activity Dialog*

6. Click the Implementation tab.
7. Select **Service Task** from the Implementation Type list, as shown in **Figure 6–20**.

8. Select **Service Call** from the Implementation list, as shown in **Figure 6–21**.

9. Click the Browse icon to the right of the Name field, as shown in **Figure 6–22**.
10. Select the Third Party Service that has been created and click OK.

You are returned to the Properties - Activity dialog, as shown in Figure 6–24.
11. Select the Use Associations option that is located under Data Associations.

12. Click the Edit Data Associations icon.

   The Data Associations dialog is displayed, as shown in Figure 6–25.

13. Right-click the Process node in the right pane and select Add.

   The Create Data Object dialog is displayed, as shown in Figure 6–26.
14. Enter a name in the Name field (for example, Request) and click **Browse more types** to the right of the Type field.

The Browse Types dialog is displayed, as shown in Figure 6–27.

15. Select `<Component>` from the Type list.
16. Select the first component (for example, JdeRequest) and click **OK**, as shown in Figure 6–28.

You are returned to the Create Data Object dialog, as shown in Figure 6–29.

17. Click **OK**.

The Data Object (for example, Request) that has been created is displayed under the Process node in the Data Associations dialog, as shown in Figure 6–30.
18. Create another Data Object by right-clicking the **Process** node in the right pane and selecting **Add**, as shown in **Figure 6–31**.

**Figure 6–31  Add Option**

![Add Option](image)

The Create Data Object dialog is displayed, as shown in **Figure 6–32**.

**Figure 6–32  Create Data Object Dialog**

![Create Data Object Dialog](image)

19. Enter a name in the Name field (for example, Response) and click **Browse more types** to the right of the Type field.

The Browse Types dialog is displayed, as shown in **Figure 6–33**.
20. Select `<Component>` from the Type list.

21. Select the second component (for example, CompanyCodeGetDetailResponse) and click OK, as shown in Figure 6–34.

You are returned to the Create Data Object dialog, as shown in Figure 6–35.
22. Click OK.

The Data Object (for example, Response) that has been created is displayed under the Process node in the Data Associations dialog, as shown in Figure 6–36.

23. Select the Request Data Object under the Process node in the right pane and drag it to the field below the Inputs area, as shown in Figure 6–37.

24. Select the Response Data Object under the Process node in the right pane and drag it to the field below the Outputs area, as shown in Figure 6–38.
25. Click OK.
   You are returned to the Properties - Activity dialog, as shown in Figure 6–39.

26. Click OK.
   The Service Task is created between the Start and End Event components, as shown in Figure 6–40.
27. Save the process and double-click the Start event component.

The Properties - Start dialog is displayed, as shown in Figure 6–41.

28. Click the Implementation tab, as shown in Figure 6–42.
29. Click the **Plus** icon to the right of the Arguments Definition field.

The Create Argument dialog is displayed, as shown in **Figure 6–43**.

**Figure 6–43  Create Argument Dialog**

30. Enter a name in the Name field (for example, argument1) and click **Browse more types** to the right of the Type field.

The Browse Types dialog is displayed, as shown in **Figure 6–44**.
31. Select `<Component>` from the Type list.

32. Select the first component (for example, JdeRequest) and click **OK**, as shown in Figure 6–45.

You are returned to the Create Argument dialog, as shown in Figure 6–46.
33. Click OK.

You are returned to the Properties - Start dialog, as shown in Figure 6–47.

34. Select the Use Associations option that is located under Data Associations.

35. Click the Edit Data Associations icon.

The Data Associations dialog is displayed.

36. Select the Request Data Object under the Process node in the right pane and drag it to the field below the Outputs area, as shown in Figure 6–48.
37. Click OK.

You are returned to the Properties - Activity dialog, as shown in Figure 6–49.

38. Click OK.

You are returned to the Process workspace area, as shown in Figure 6–50.

39. Double-click composite.xml in the left pane.
40. Click the Save All icon in the menu bar to save the new outbound BPM process component that was configured.

You are now ready to create a File adapter for the write operation.

Creating a File Adapter for the Write Operation

Perform the following steps to create a File adapter for the write operation:

1. Drag and drop the File Adapter component from the Service Adapters pane to the External References pane, as shown in Figure 6–51.

The Adapter Configuration Wizard is displayed, as shown in Figure 6–52.
2. Click Next.
   The Service Name page is displayed, as shown in Figure 6–53.
3. Type a name for the new File adapter in the Service Name field and click Next. The Adapter Interface page is displayed, as shown in Figure 6–54.
4. Ensure that the **Define from operation and schema (specified later)** option is selected.

5. Click **Next**.

   The Operation page is displayed, as shown in **Figure 6–55**.
6. Select **Write File** from the list of Operation Type options and specify an Operation Name (for example, Write).

7. Click **Next**.

The File Configuration page is displayed, as shown in **Figure 6–56**.
Specify a location on your file system where the output file is written.

9. In the File Naming Convention field, specify a name for the output file.

10. Click Next.

The Messages page is displayed, as shown in Figure 6–57.
11. Click Browse, which is located to the right of the URL field.

The Type Chooser dialog is displayed, as shown in Figure 6–58.

Figure 6–58 Type Chooser Dialog

12. Expand Project Schema Files and JDE90_GetEffectiveAddress_invoke_response.xsd.
13. Select the available schema (for example, jdeResponse).

14. Click **OK**.

   You are returned to the Messages page, as shown in **Figure 6–59**.

![Figure 6–59 Messages Page](image)

15. Click **Next**.

   The Finish page is displayed, as shown in **Figure 6–60**.
16. Click Finish.

The File Adapter service is created in the External References pane, as shown in Figure 6–61.

**Figure 6–61  File Adapter Service**
17. Double-click the BPMN Process component.

The BPMN process is displayed, as shown in Figure 6–62.

**Figure 6–62 BPMN Process**

![BPMN Process Diagram](image)

18. Click the Activity icon.

19. Drop the Activity icon on the wire between the Service Task and End event components, as shown in Figure 6–63.

**Figure 6–63 Activity Icon**

![Activity Icon Diagram](image)

The Properties - Activity dialog is displayed, as shown in Figure 6–64.
20. Click the **Implementation** tab.

21. Select **Service Task** from the Implementation Type list, as shown in Figure 6–65.
22. Select **Service Call** from the Implementation list, as shown in Figure 6–66.
23. Click the **Browse** icon to the right of the Name field, as shown in Figure 6–67.

The Type dialog is displayed, as shown in Figure 6–68.
24. Select the service for write operation that has been created (for example, JDE_File_Adapter) and click OK.

You are returned to the Properties - Activity dialog, as shown in Figure 6–69.
25. Select the **Use Transformations** option that is located under Data Associations.

26. Click the **Edit XSL Transformations** icon.

The Transformations dialog is displayed, as shown in Figure 6–70.
27. Click the **Plus** icon.

The Transformation dialog is displayed, as shown in Figure 6–71.

![Transformations Dialog](image)

28. Select the **Response** Data Object that was created from the Sources list.
29. Click the **Plus** icon so that the Data Object ‘Response’ is added to the Selected Elements area, as shown in **Figure 6–72**.

30. Accept the default value that is selected in the Target list.
31. Enter a name (for example, transformation) in the Create field and click OK, as shown in Figure 6–73.

You are returned to the Transformations dialog, as shown in Figure 6–74.

32. Click OK.

You are returned to the Properties - Activity dialog, as shown in Figure 6–75.
33. Click OK.

The transformation.xsl tab is displayed, as shown in Figure 6–76.

34. Automap the Source and Target elements.

The Auto Map Preferences dialog is displayed, as shown in Figure 6–77.
35. Accept the default values and click **OK**.

   The transformation is completed, as shown in **Figure 6–78**.

**Figure 6–78  Completed Transformation**

36. Save the transformation.

37. Return to the Process workspace area, as shown in **Figure 6–79**.
The ServiceTask1 component is created between the ServiceTask component and the End event component.

38. Save the process.
39. Double-click the End event component.
   The Properties - End dialog is displayed, as shown in Figure 6–80.
40. Click the Implementation tab.

Figure 6–80 Properties - End Dialog

41. Select None from the Implementation Type list, as shown in Figure 6–81.
42. Click OK.

You are returned to the Process workspace area.

43. Save the Process and double-click the composite.xml file.

44. Click the Save All icon in the menu bar to save the new outbound BPM process component that was configured.

You are now ready to deploy the outbound BPM process.

6.4.3 Deploying the BPM Outbound Process

Perform the following steps to deploy the BPM outbound process.

1. Right-click the project name in the left pane (for example, JDE_JCA_Outbound), select Deploy, and then click JDE_JCA_Outbound, as shown in Figure 6–82.
2. Ensure that **Deploy to Application Server** is selected.

3. Click **Next**.

   The Deploy Configuration page is displayed, as shown in Figure 6–84.
4. Leave the default values selected and click **Next**.

   The Select Server page is displayed, as shown in Figure 6–85.

**Figure 6–85 Select Server Page**
5. Select an available application server that was configured and click **Next**.
   The SOA Servers page is displayed, as shown in **Figure 6–86**.

**Figure 6–86  SOA Servers Page**

6. Select a target SOA server and click **Next**.
   The Summary page is displayed, as shown in **Figure 6–87**.
7. Review and verify all the available deployment information for your project and click Finish.

The process is deployed successfully, as shown in Figure 6–88.

**Figure 6–88  Successful Deployment Message**

6.4.4 Invoking the Input XML Document in the Oracle Enterprise Manager Console

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

1. Logon to the Oracle Enterprise Manager console, as shown in Figure 6–89.
2. Expand your domain in the left pane followed by the SOA folder.

3. Select an available project.

4. Click Test, as shown in Figure 6–90.
5. Click the Request tab, as shown in Figure 6–91.

6. Provide an appropriate input value in the Value field and click Test Web Service. A response is received in the Response tab to indicate that invocation was successful in the Oracle Enterprise Manager console, as shown in Figure 6–92.

7. Navigate to the defined output directory on your file system and open the XML response document that was received, as shown in Figure 6–93.
The XML response document contains the generated output with values.

6.5 Designing an Inbound BPM Process Using Transformations for Event Integration

This section demonstrates how Oracle Application Adapter for J.D. Edwards OneWorld integrates with J.D. Edwards OneWorld to receive event data.

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

- Oracle Adapter Application Explorer (Application Explorer)
- Oracle JDeveloper BPM Designer (JDeveloper) or Eclipse

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

Before you design a BPM process, you must generate the respective WSDL file using Application Explorer. For more information, see "Generating WSDL for Event Integration" on page 4-41.

6.5.1 Creating an Empty Composite for BPM

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

1. Create a new BPM application.

2. Enter a name for the new BPM application and click Next, as shown in Figure 6–94.

![Figure 6–94 Name Your Application Page](image)
The Name your project page is displayed, as shown in Figure 6–95.

**Figure 6–95  Name Your Project Page**

3. Enter a project name (for example, JDE_JCA_Inbound) and click **Next**.
   The Configure SOA settings page is displayed, as shown in Figure 6–96.
4. From the Composite Template list, select Empty Composite and click Finish.

6.5.2 Defining a BPM Inbound Process

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

1. Configuring a Third Party Adapter Service Component
2. Configuring an Inbound BPM Process Component
3. Creating a File Adapter for the Write Operation

Configuring a Third Party Adapter Service Component

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

1. Drag and drop the Third Party Adapter component from the Service Adapters pane to the Exposed References pane, as shown in Figure 6–97.
The Create Third Party Adapter Service dialog is displayed, as shown in Figure 6–98.

**Figure 6–98 Create Third Party Adapter Service Dialog**

1. Enter a name for the third party adapter service.
2. Ensure that Service is selected from the Type list (default).
3. Click the Find existing WSDLs icon, which is located to the right of the WSDL URL field.

The SOA Resource Browser dialog is displayed, as shown in Figure 6–99.
5. Browse and select an inbound WSDL file from the following directory:
   `<ADAPTER_HOME>\soa\thirdparty\ApplicationAdapters\wsdls`

6. Click **OK**.
   The Localize Files dialog is displayed, as shown in Figure 6–100.
7. Click OK.

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

You are returned to the Create Third Party Adapter Service dialog, as shown in Figure 6–101.

**Figure 6–101 Create Third Party Adapter Service Dialog**
8. Click the **Find JCA file** icon, which is located to the right of the JCA File field.
   The SOA Resource Browser dialog is displayed, as shown in **Figure 6–102**.

   **Figure 6–102  SOA Resource Browser Dialog**

   ![SOA Resource Browser Dialog](image)

9. Browse and select the JCA properties file from the following directory:
   
   `<ADAPTER_HOME>\soa\thirdparty\ApplicationAdapters\wsdl\`

10. Click **OK**.
    The following message is displayed, as shown in **Figure 6–103**.

   **Figure 6–103  Copy File Confirmation Message**

   ![Copy File Confirmation Message](image)

11. Click **Yes**.
    A copy of the JCA properties file is made in the project folder.
    You are returned to the Create Third Party Adapter Service dialog, as shown in **Figure 6–104**.
Click OK. The third party adapter service component (SalesOrder) is created and displayed in the Exposed References pane, as shown in Figure 6–105.

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

Configuring an Inbound BPM Process Component
Perform the following steps to configure an inbound BPM process component:
1. Drag and drop the BPMN Process component from the Service Components pane to the Components pane, as shown in Figure 6–106.
2. Select the **From Pattern** option in the left pane and ensure that **Manual Process** is selected in the Patterns area.

3. Click **Finish**.
   
The BPMN Process component is created, as shown in **Figure 6–108**.
4. Double-click the BPMN Process component in the Components pane. The BPMN process is displayed, as shown in Figure 6–109.

5. Right-click UserTask and select Delete from the menu.

6. Double-click the Start event component, as shown in Figure 6–110.
The Properties - Start dialog is displayed, as shown in Figure 6–111.

7. Click the Implementation tab, as shown in Figure 6–112.

8. Select Message from the Implementation Type list, as shown in Figure 6–113.
Figure 6–113 Implementation Type List

9. Select Interface from Catalog from the Implementation list.

10. Click the Browse icon to the right of the Name field, as shown in Figure 6–114.

Figure 6–114 Browse Icon

The Type dialog is displayed, as shown in Figure 6–115.
11. Select the Third Party Service that has been created and click **OK**.

You are returned to the Properties - Start dialog, as shown in **Figure 6–116**.
12. Select the Use Associations option that is located under Data Associations.

13. Click the Edit Data Associations icon.

The Data Associations dialog is displayed, as shown in Figure 6–117.

**Figure 6–117  Data Associations Dialog**

![Data Associations Dialog](image)

14. Right-click the Process node in the right pane and select Add.

The Create Data Object dialog is displayed, as shown in Figure 6–118.

**Figure 6–118  Create Data Object Dialog**

![Create Data Object Dialog](image)

15. Enter a name in the Name field (for example, SalesOrder_inbound) and click Browse more types to the right of the Type field.

The Browse Types dialog is displayed, as shown in Figure 6–119.
16. Select `<Component>` from the Type list.

17. Select the first component (for example, JdeResponse) and click OK, as shown in Figure 6–120.

You are returned to the Create Data Object dialog, as shown in Figure 6–121.
18. Click OK.

The Data Object (for example, SalesOrder_inbound) that has been created is displayed under the Process node in the Data Associations dialog, as shown in Figure 6–122.

19. Select the SalesOrder_inbound Data Object under the Process node in the right pane and drag it to the field below the Outputs area.

20. Click OK.

You are returned to the Properties - Start dialog, as shown in Figure 6–123.
21. Click OK.

You are returned to the Process workspace area, as shown in Figure 6–124.

22. Double-click composite.xml in the left pane.

23. Click the Save All icon in the menu bar to save the new inbound BPM process component that was configured.

You are now ready to create a File adapter for the write operation.

Creating a File Adapter for the Write Operation

Perform the following steps to create a File adapter for the write operation:

1. Drag and drop the File Adapter component from the Service Adapters pane to the External References pane, as shown in Figure 6–125.
The Adapter Configuration Wizard is displayed, as shown in Figure 6–126.

Figure 6–126 Adapter Configuration Wizard

2. Click Next.

The Service Name page is displayed, as shown in Figure 6–127.
3. Type a name for the new File adapter in the Service Name field and click Next. The Adapter Interface page is displayed, as shown in Figure 6–128.
4. Ensure that the **Define from operation and schema (specified later)** option is selected.

5. Click Next.
   
The Operation page is displayed, as shown in Figure 6–129.
6. Select **Write File** from the list of Operation Type options and specify an Operation Name (for example, Write).

7. Click **Next**.

The File Configuration page is displayed, as shown in Figure 6–130.
8. Specify a location on your file system where the output file is written.

9. In the File Naming Convention field, specify a name for the output file.

10. Click Next.

   The Messages page is displayed, as shown in Figure 6–131.
11. Click **Browse**, which is located to the right of the URL field.

    The Type Chooser dialog is displayed, as shown in Figure 6–132.

**Figure 6–132  Type Chooser Dialog**
12. Expand Project Schema Files and JDE90_Salesorder_attrib_http_inbound_request.xsd.

13. Select the available schema (for example, jdeResponse).

14. Click OK.

   You are returned to the Messages page, as shown in Figure 6–133.

**Figure 6–133  Messages Page**

![Image of Messages Page]

15. Click Next.

   The Finish page is displayed, as shown in Figure 6–134.
16. Click Finish.

The File Adapter service is created in the External References pane, as shown in Figure 6–135.

**Figure 6–135 File Adapter Service**

17. Double-click the BPMN Process component.

The BPMN process is displayed, as shown in Figure 6–136.
18. Click the Activity icon.
19. Drop the Activity icon on the wire between the Start and End event components, as shown in Figure 6–137.

The Properties - Activity dialog is displayed, as shown in Figure 6–138.
20. Click the **Implementation** tab, as shown in Figure 6–139.

21. Select **Service Task** from the Implementation Type list, as shown in Figure 6–140.
22. Select **Service Call** from the Implementation list.

23. Click the **Browse** icon to the right of the Name field, as shown in Figure 6–141.

Figure 6–141  **Browse Icon**

The Type dialog is displayed, as shown in Figure 6–142.
24. Select the service for write operation that has been created and click **OK**. You are returned to the Properties - Activity dialog, as shown in **Figure 6–143**.
25. Select the **Use Transformations** option that is located under Data Associations.

26. Click the **Edit XSL Transformations** icon.

   The Transformations dialog is displayed, as shown in Figure 6–144.

**Figure 6–144 Transformations Dialog**

![Transformations Dialog](image)

27. Click the **Plus** icon.

   The Transformation dialog is displayed, as shown in Figure 6–145.

**Figure 6–145 Transformation Dialog**

![Transformation Dialog](image)
28. Select the **SalesOrder_inbound** Data Object that was created from the Sources list.

29. Click the **Plus** icon so that the Data Object 'SalesOrder_inbound' is added to the Selected Elements area.

*Figure 6–146  Target List Default Value*

![Transformation](image)

30. Accept the default value that is selected in the Target list, as shown in *Figure 6–146*. 
31. Enter a name (for example, sales) in the Create field and click OK, as shown in Figure 6–147.

You are returned to the Transformations dialog, as shown in Figure 6–148.

32. Click OK.

You are returned to the Properties - Activity dialog, as shown in Figure 6–149.
33. Click **OK**.

The sales.xsl tab is displayed, as shown in **Figure 6-150**.

**Figure 6–150  Sales.xsl Tab**

34. Automap the Source and Target elements.

The Auto Map Preferences dialog is displayed, as shown in **Figure 6–151**.
35. Accept the default values and click OK.

The transformation is completed, as shown in Figure 6–152.

36. Save the transformation.

37. Return to the Process workspace area.

The ServiceTask component is created between the Start event component and the End event component.

38. Save the Process and double-click the composite.xml file.

39. Click the Save All icon in the menu bar to save the new inbound BPM process component that was configured.

You are now ready to deploy the inbound BPM process. You can follow the same procedure that is described in "Deploying the BPEL Inbound Process" on page 4-59. For more information on how to trigger events in J.D. Edwards OneWorld, see "Triggering an Event in J.D. Edwards OneWorld" on page 4-63.
Configuring Outbound and Inbound Processing Using Oracle Service Bus

Oracle Application Adapter for J.D. Edwards OneWorld integrates seamlessly with Oracle Service Bus (OSB) to facilitate Web service integration. OSB is based on the Service-Oriented Architecture (SOA). It consumes adapter services exposed as Web Service Definition Language (WSDL) documents.

This chapter contains the following topics:

- Section 7.1, "Overview of Application Adapter Integration with Oracle Service Bus"
- Section 7.2, "Configuring Outbound Processing Using Oracle Service Bus (J2CA Configuration)"
- Section 7.3, "Configuring Inbound Processing Using Oracle Service Bus (J2CA Configuration)"
- Section 7.4, "Configuring Outbound Processing Using Oracle Service Bus (BSE Configuration)"

7.1 Overview of Application Adapter Integration with Oracle Service Bus

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

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

This section describes how to configure outbound processing using Oracle Service Bus for J2CA configurations.

This section includes the following topics:

- Section 7.2.1, "Starting Oracle Service Bus and Creating Project Folders"
- Section 7.2.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"
- Section 7.2.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"
7.2.1 Starting Oracle Service Bus and Creating Project Folders

Perform the following steps to start Oracle Service Bus and create project folders:

1. Start the Oracle WebLogic Server for the Oracle WebLogic Server domain that you have configured.

2. Open the Oracle Service Bus Console in a Web browser by entering the following URL:

   `http://hostname:port/sbconsole`

   Where `hostname` is the name of the machine where Oracle WebLogic Server is running and `port` is the port for the domain you are using. The port for the default domain is 7001.

   The Oracle Service Bus Console logon page is displayed.

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

   The Oracle Service Bus Console home page is displayed, as shown in Figure 7–1.

4. Click **Create** in the Change Center area to start a new Oracle Service Bus session, as shown in Figure 7–2.
5. Click Project Explorer in the left pane, as shown in Figure 7–3.

The Project Explorer page is displayed, as shown in Figure 7–4.

6. Provide a valid name for the new project (for example, JDE_Project) and click Add Project.

The project is successfully created and listed.

7. Click on the created project.

The project page is displayed, as shown in Figure 7–5.
8. In the Enter New Folder Name field, type Business Service and click Add Folder.

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

10. In the Enter New Folder Name field, type wsdls and click Add Folder.

   The Business Service, Proxy Service, and wsdls folders are listed in the left pane below the project node, as shown in Figure 7–6.

7.2.2 Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus

Before starting and using Application Explorer to publish a WSDL directly to the Oracle Service Bus (OSB) Console (project/folder), OSB users must perform the following steps:

1. Open the command prompt window.
2. Navigate to the following directory:
   \OsB_Home\user_projects\domains\base_domain\bin
3. Execute setDomainEnv.cmd (Windows) or ./setDomainEnv.sh (UNIX/Linux).
   This command sets the class path for Application Explorer to access the Oracle WebLogic Server APIs to publish the WSDLs to the OSB Console.
4. Do not close the command prompt window.
5. Navigate to the following directory:
   \OsB_Home\3rdparty\ApplicationAdapters\tools\iwae\bin
6. Execute ae.bat (Windows) or iwae.sh (UNIX/Linux) to start Application Explorer.

You are now ready to publish WSDLs from Application Explorer to the OSB Console.
7.2.3 Publishing a WSDL From Application Explorer to Oracle Service Bus

Perform the following steps to publish a WSDL from Application Explorer to Oracle Service Bus:

1. Start Application Explorer, connect to a J2CA configuration, and connect to a J.D. Edwards OneWorld target.
   
   For more information, see Chapter 2, "Configuring Oracle Application Adapter for J.D. Edwards OneWorld".

2. Expand Services, CALLBSFN, and Addressbook, as shown in Figure 7–7.

   **Figure 7–7 Create Outbound JCA Service (Request/Response) Option**

3. Right-click GetEffectiveAddress, and then select Create Outbound JCA Service (Request/Response).

   The Export WSDL dialog is displayed, as shown in Figure 7–8.
Figure 7–8  Export WSDL Dialog

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

5. Select the Export to OSB option.

6. In the Location field, enter the folder name in Oracle Service Bus where you want to publish the WSDL document.
   
   The location is composed of an Oracle Service Bus project name and optionally, one or more folder names. The project name and any folder names must be separated by a forward slash character “/”.

7. In the Host field, enter the name of the machine where Oracle Service Bus is installed.

8. In the Port field, enter the port that is being used by Oracle Service Bus.

9. In the User field, enter your username to access Oracle Service Bus.

10. In the Password field, enter your password to access Oracle Service Bus.

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

12. Open the Oracle Service Bus Console and select the created WSDL folder in the left pane (for example, wsdls), as shown in Figure 7–9.
13. Ensure that the exported WSDL is listed, as shown in Figure 7–10.

Figure 7–10 Resource List

<table>
<thead>
<tr>
<th>Name</th>
<th>Resource Type</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetEffectiveAddress_invoke</td>
<td>JCA Binding</td>
<td>![Image]</td>
</tr>
<tr>
<td>GetEffectiveAddress_invoke_request</td>
<td>WSDL</td>
<td>![Image]</td>
</tr>
<tr>
<td>GetEffectiveAddress_invoke_response</td>
<td>XML Schema</td>
<td>![Image]</td>
</tr>
</tbody>
</table>

14. Click the icon that corresponds to the exported WSDL in the Actions column. The Generate WSDL and Service page is displayed, as shown in Figure 7–11.

Figure 7–11 Generate WSDL and Service Page
15. Provide a new WSDL name and a new service name in the corresponding fields, as shown in Figure 7–12.

Figure 7–12 Generate WSDL and Service Page

![Generate WSDL and Service Page](image)

16. In the Location area, select an available project and the sub-folder that is designated for Business Services.

17. Click Generate.

If the WSDL and Service resources are successfully created, then the following message is displayed, as shown in Figure 7–13.

Figure 7–13 Success Message

![Success Message](image)

7.2.4 Configuring a File Type Business Service

Perform the following steps to configure a File type Business Service:

1. Select the Business Service folder you created in the left pane, as shown in Figure 7–14.
2. In the right pane, select **Business Service** from the Create Resource menu, as shown in Figure 7–15.

The General Configuration page is displayed, as shown in Figure 7–16.
3. Provide a name for the Business Service and from the Service Type area select Messaging Service.

4. Click Next.

The Message Type Configuration page is displayed, as shown in Figure 7–17.

**Figure 7–17 Message Type Configuration Page**

![Message Type Configuration](image)

5. Select XML as the Request Message Type and None as the Response Message Type.

6. Click Next.

The Transport Configuration page is displayed, as shown in Figure 7–18.

**Figure 7–18 Transport Configuration Page**

![Transport Configuration](image)
7. Select file from the Protocol list.
8. Enter the path to a destination folder on your file system in the Endpoint URI field and click Add.
9. Click Next, as shown in Figure 7–19.

Figure 7–19  Transport Configuration Page

The FILE Transport Configuration page is displayed, as shown in Figure 7–20.

Figure 7–20  FILE Transport Configuration Page

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

The Message Handling and Advanced Settings page is displayed, as shown in Figure 7–21.

Figure 7–21  Message Handling and Advanced Settings Page
11. Click Next.

The following Summary page is displayed, as shown in Figure 7–22.

**Figure 7–22 Summary Page**

<table>
<thead>
<tr>
<th>[Create a Business Service - Summary (JDE_Project/Business Service/JDE_File_Out)]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General Configuration</strong></td>
</tr>
<tr>
<td>Service Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Service Type</td>
</tr>
<tr>
<td><strong>Message Type Configuration</strong></td>
</tr>
<tr>
<td>Request Message Type</td>
</tr>
<tr>
<td>Response Message Type</td>
</tr>
<tr>
<td><strong>Transport Configuration</strong></td>
</tr>
<tr>
<td>Protocol</td>
</tr>
<tr>
<td>Load Balancing Algorithm</td>
</tr>
<tr>
<td>Endpoint URI</td>
</tr>
<tr>
<td>Retry Count</td>
</tr>
<tr>
<td>Retry Iteration Interval</td>
</tr>
<tr>
<td><strong>FILE Transport Configuration</strong></td>
</tr>
<tr>
<td>Prefix</td>
</tr>
<tr>
<td>Suffix</td>
</tr>
<tr>
<td>Request encoding</td>
</tr>
<tr>
<td><strong>Message Handling Configuration</strong></td>
</tr>
<tr>
<td>Result Caching</td>
</tr>
</tbody>
</table>

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

13. Check if the success message is displayed and click **Activate** in the Change Center, as shown in Figure 7–23.

**Figure 7–23 Change Center Activate Button**

The Activate Session page is displayed, as shown in Figure 7–24.
Figure 7–24   Activate Session Page

![Activate Session Page](image)

14. Click Submit.

15. Check if the View Configuration Changes page is displayed, as shown in Figure 7–25.

Figure 7–25   View Configuration Changes Page

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
<th>Execution Time</th>
<th>User</th>
<th>Task Status</th>
<th>Undone By</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activate session weblogic</td>
<td>7/20/2013 7:57 PM</td>
<td>weblogic</td>
<td>Activated</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Activate session weblogic</td>
<td>7/20/2013 7:01 PM</td>
<td>weblogic</td>
<td>Activated</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Activate session weblogic</td>
<td>7/20/2013 6:27 PM</td>
<td>weblogic</td>
<td>Activated</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Activate session weblogic</td>
<td>7/20/2013 11:31 AM</td>
<td>weblogic</td>
<td>Activated</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Activate session weblogic</td>
<td>7/20/2013 11:55 AM</td>
<td>weblogic</td>
<td>Activated</td>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

7.2.5 Configuring a Proxy Service

Perform the following steps to configure a Proxy Service:

1. Select the Proxy Service folder you created in the left pane, as shown in Figure 7–26.
2. In the right pane, select **Proxy Service** from the Create Resource menu. The General Configuration page is displayed, as shown in **Figure 7–27**.

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

4. Click **Next**, as shown in **Figure 7–28**.
5. Select file from the Protocol list.
6. Enter the path to an input folder on your file system in the Endpoint URI field.
7. Click Next, as shown in Figure 7–29.

9. Click Next, as shown in Figure 7–30.
10. Accept the default values and click **Next**.

The following Summary page is displayed, as shown in **Figure 7–31**.
11. Review all the information for your Proxy Service and click **Save**.

12. Check if the success message is displayed, as shown in **Figure 7–32**.

**Figure 7–32  Success Message**

- The Service "Get_Address_OB_PS" was created successfully.
- JDE_Project/Proxy Service
  - Folders
    - Enter New Folder Name: [Input Field]  [Add Folder]
7.2.6 Configuring a Pipeline

Perform the following steps to configure a Pipeline:

1. Click the **Edit Message Flow** icon in the row of the Proxy Service you created, as shown in Figure 7–33.

![Figure 7–33  Edit Message Flow Icon](image)

The Edit Message Flow workspace area is displayed, as shown in Figure 7–34.

![Figure 7–34  Edit Message Flow Workspace Area](image)

2. Click the **Proxy Service** icon and select **Add Pipeline Pair** from the menu.

![Add Pipeline Pair](image)

3. Click the **PipelinePairNode1** icon and select **Add Route** from the menu, as shown in Figure 7–35.
4. Click the RouteNode1 icon and select Edit Route from the menu.
The Edit Stage Configuration workspace area is displayed, as shown in Figure 7–37.

Figure 7–37 Edit Stage Configuration Workspace Area

5. Click Add an Action, select Communication from the menu, and click Routing, as shown in Figure 7–38.

Figure 7–38 Edit Stage Configuration Workspace Area

6. Click <Service>.
   The Select Service dialog is displayed, as shown in Figure 7–39.
7. Select a WSDL type Business Service and click **Submit**. You are returned to the Edit Stage Configuration workspace area, as shown in Figure 7–40.

![Select Service Dialog](image)

**Figure 7–39  Select Service Dialog**

8. Select the name of the J.D. Edwards OneWorld business object (for example, `GetEffectiveAddress`) as the operational attribute from the list, as shown in Figure 7–41.

![Edit Stage Configuration Workspace Area](image)

**Figure 7–40  Edit Stage Configuration Workspace Area**
9. Click Save, as shown in Figure 7–42.

Figure 7–42 Add Stage Menu Selection

10. Click the Response Pipeline icon and select Add Stage from the menu. The Stage1 icon is added below the Response Pipeline icon, as shown in Figure 7–43.
Figure 7–43  Edit Stage Menu Selection

11. Click the Stage 1 icon and select Edit Stage from the menu.
   The Edit Stage Configuration workspace area is displayed, as shown in Figure 7–44.

Figure 7–44  Edit Stage Configuration Workspace Area

12. Click Add an Action, select Communication from the menu, and click Publish, as shown in Figure 7–45.
13. Click `<Service>`.

The Select Service dialog is displayed, as shown in Figure 7–46.

**Figure 7–46  Select Service Dialog**

<table>
<thead>
<tr>
<th>Name</th>
<th>Path</th>
<th>Resource Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account_36_evel</td>
<td>Siebel_Project/Proxy Service</td>
<td>Proxy Service</td>
</tr>
<tr>
<td>Get_Address_OEI_PS</td>
<td>JDE_Project/Proxy Service</td>
<td>Proxy Service</td>
</tr>
<tr>
<td>Get_Detail_36_SAP</td>
<td>SAP_Project/Proxy Service</td>
<td>Proxy Service</td>
</tr>
<tr>
<td>GetDetail_inoke_SPS</td>
<td>SAP_Project/Busines Service</td>
<td>Business Service</td>
</tr>
<tr>
<td>GetEffectiveInvoice_inoke_BSP</td>
<td>JDE_Project/Busines Service</td>
<td>Business Service</td>
</tr>
<tr>
<td>JDE_File_Out</td>
<td>JDE_Project/Busines Service</td>
<td>Business Service</td>
</tr>
<tr>
<td>NATHIN01_receive_PS</td>
<td>SAP_Project/Proxy Service</td>
<td>Proxy Service</td>
</tr>
<tr>
<td>NATHIN02_receive_PS</td>
<td>SAP_Project/Proxy Service</td>
<td>Proxy Service</td>
</tr>
<tr>
<td>query_invoice_BSP</td>
<td>Siebel_Project/Busines Service</td>
<td>Business Service</td>
</tr>
<tr>
<td>Query_o3_hubvel</td>
<td>Siebel_Project/Proxy Service</td>
<td>Proxy Service</td>
</tr>
<tr>
<td>Sample_Account_receive_PS</td>
<td>Siebel_Project/Proxy Service</td>
<td>Proxy Service</td>
</tr>
<tr>
<td>SAP_File_Out</td>
<td>SAP_Project/Busines Service</td>
<td>Business Service</td>
</tr>
<tr>
<td>SAP_Out_Nathin</td>
<td>SAP_Project/Busines Service</td>
<td>Business Service</td>
</tr>
<tr>
<td>Siebel_Hop_Out</td>
<td>Siebel_Project/Busines Service</td>
<td>Business Service</td>
</tr>
<tr>
<td>Siebel_Out_SampleAccount</td>
<td>Siebel_Project/Busines Service</td>
<td>Business Service</td>
</tr>
</tbody>
</table>

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

You are returned to the Edit Stage Configuration workspace area, as shown in Figure 7–47.
15. Click **Save**.

You are returned to the Edit Message Flow workspace area, as shown in Figure 7–48.

16. Click **Save**, as shown in Figure 7–49.
17. Check if the success message is displayed and click **Activate** in the Change Center. The Activate Session page is displayed, as shown in Figure 7–50.

18. Click **Submit**.

19. Check if the View Configuration Changes page is displayed, as shown in Figure 7–51.

20. Copy and paste an input XML file in the input folder you have configured, as shown in Figure 7–52.
7.3 Configuring Inbound Processing Using Oracle Service Bus (J2CA Configuration)

This section describes how to configure inbound processing using Oracle Service Bus for J2CA configurations.

This section includes the following topics:

- Section 7.3.1, "Starting Oracle Service Bus and Creating Project Folders"
- Section 7.3.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"
- Section 7.3.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"
- Section 7.3.4, "Importing the Inbound J.D. Edwards OneWorld WSDL File Into Oracle Service Bus"
- Section 7.3.5, "Configuring a File Type Business Service"
- Section 7.3.6, "Configuring a Pipeline"

7.3.1 Starting Oracle Service Bus and Creating Project Folders

Perform the following steps to start Oracle Service Bus and create project folders:

1. Start the Oracle WebLogic Server for the Oracle WebLogic Server domain that you have configured.

2. Open the Oracle Service Bus Console in a Web browser by entering the following URL:

   http://hostname:port/sbconsole

   Where hostname is the name of the machine where Oracle WebLogic Server is running and port is the port for the domain you are using. The port for the default domain is 7001.

21. Output is received in the configured output location.
The Oracle Service Bus Console logon page is displayed.

3. Log on to the Oracle Service Bus Console using a valid user name and password.
   The Oracle Service Bus Console home page is displayed, as shown in Figure 7–53.

Figure 7–53 Oracle Service Bus Console Home Page

4. Click **Create** in the Change Center area to start a new Oracle Service Bus session, as shown in Figure 7–54.

Figure 7–54 Change Center

5. Click **Project Explorer** in the left pane, as shown in Figure 7–55.

Figure 7–55 Project Explorer
The Project Explorer page is displayed, as shown in Figure 7–56.

**Figure 7–56  Project Explorer Page**

6. Provide a valid name for the new project (for example, JDE_Project) and click **Add Project**.

   The project is successfully created and listed.

7. Click on the created project.

   The project page is displayed, as shown in Figure 7–57.

**Figure 7–57  Enter New Folder Name Field**

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

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

10. In the Enter New Folder Name field, type **wsdls** and click **Add Folder**.

   The Business Service, Proxy Service, and wsdls folders are listed in the left pane below the project node, as shown in Figure 7–58.

**Figure 7–58  Project Explorer**
7.3.2 Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus

Before starting and using Application Explorer to publish a WSDL directly to the Oracle Service Bus (OSB) Console (project/folder), OSB users must perform the following steps:

1. Open the command prompt window.
2. Navigate to the following directory:
   `<OSB_Home>\user_projects\domains\base_domain\bin`
3. Execute `setDomainEnv.cmd` (Windows) or `. /setDomainEnv.sh` (UNIX/Linux).
   This command sets the class path for Application Explorer to access the Oracle WebLogic Server APIs to publish the WSDLs to the OSB Console.
4. Do not close the command prompt window.
5. Navigate to the following directory:
   `<OSB_Home>\3rdparty\ApplicationAdapters\tools\iwae\bin`
6. Execute `ae.bat` (Windows) or `iwae.sh` (UNIX/Linux) to start Application Explorer.
   You are now ready to publish WSDLs from Application Explorer to the OSB Console.

7.3.3 Publishing a WSDL From Application Explorer to Oracle Service Bus

Before you publish a WSDL from Application Explorer to Oracle Service Bus, you must create a separate channel for every J2CA event and select that channel when you generate WSDL for inbound interaction using Application Explorer.

Creating a Channel in Application Explorer

To create a channel:

1. In Application Explorer, connect to a J2CA configuration.
2. Expand the Events node and then the JDEdwards node, as shown in Figure 7–59.

Figure 7–59 Application Explorer Add Channel Option
3. Right-click the Channels node, and select Add Channels.
   The Add Channel dialog is displayed, as shown in Figure 7–60.

   **Figure 7–60  Add Channel Dialog**

4. In the Name field, enter a descriptive name for the channel.
5. In the Description field, enter a description (optional).
6. From the Protocol list, choose a protocol for your channel.
7. Click Next.
   The dialog is displayed for the selected listener, as shown in Figure 7–61.

   **Figure 7–61  TcpListener Dialog**

8. Enter the port number of the channel in the Port Number field.
9. Enter the location of the server in the Host/IP Binding field.
10. Select the Synchronization type from the Synchronization Type list.
11. Select **Is Length Prefix** for events that send data which is not in XML format. The TCP/IP event application must prefix the data with a 4-byte binary length field when writing the data to the TCP/IP port.

12. Select **Is XML** for events that send data back in XML format. No preparser is required.

13. Select **Is Keep Alive** to maintain a continuous communication between the event transaction and the channel.

14. Click the **PreParser** tab, as shown in Figure 7–62.

**Figure 7–62  Tcp Listener PreParser Tab**

Enter values based on the table.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User id*</td>
<td>A valid user ID for J.D. Edwards OneWorld.</td>
</tr>
<tr>
<td>User password*</td>
<td>The password associated with the user ID.</td>
</tr>
<tr>
<td>JDE Environment*</td>
<td>Your J.D. Edwards OneWorld environment. For more information about this</td>
</tr>
<tr>
<td></td>
<td>parameter, see your J.D. Edwards OneWorld documentation or ask your</td>
</tr>
<tr>
<td></td>
<td>OneWorld system administrator.</td>
</tr>
<tr>
<td>Application</td>
<td>XMLInterop or the application name in J.D. Edwards OneWorld. Optional.</td>
</tr>
<tr>
<td>Server IP address*</td>
<td>The name of the server on which J.D. Edwards OneWorld is running. This can</td>
</tr>
<tr>
<td></td>
<td>be the name of the server, for example, JDEOW, or its IP address, for example,</td>
</tr>
<tr>
<td></td>
<td>123.45.67.89.</td>
</tr>
<tr>
<td>Server Port*</td>
<td>The port number on which the server is listening, for example, 6009.</td>
</tr>
<tr>
<td>User Role</td>
<td>Define a user role according to your requirements.</td>
</tr>
</tbody>
</table>
Configuring Inbound Processing Using Oracle Service Bus (J2CA Configuration)

Click OK.

The channel is created and displayed under the Channels node. An X over the icon indicates that the channel is currently disconnected.

### Generating WSDL for Event Notification (Command Prompt Only)

You cannot generate WSDL for J.D. Edwards OneWorld event notification using Application Explorer. To generate WSDL from the command prompt, you must perform the following steps.

You can create inbound J2CA service only if the node you have selected supports events.

1. Create a channel in Application Explorer under the J.D. Edwards events node.
2. Start the channel.
   
   **Important:** Do not restart the Oracle Application Server after the channel is started.
3. Send an inbound message from J.D. Edwards OneWorld.
4. Capture the inbound message payload in the log file located under:
   
   `<OSB_Home>\3rdparty\ApplicationAdapters\config\config_name\schemas\JDEdwards\target_name\jde-schema.xsd`

Alternatively, you can create a port with the File protocol under Event in Application Explorer to dispose the event message to the file system.

5. Use a third party tool such as XMLSpy to create the XSD schema using the XML payload captured in the previous step.
6. In the generated XML schema (.xsd file) perform the following modifications:
   
   a. Search for `Schemas-jdedwards-com` and replace it with `iwaysoftware`.
   
   b. Cut the following syntax:

   ```xml
   <xs:element name="jdeResponse">
   <xs:complexType>
   </xs:complexType>
   </xs:element>
   ```

---

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schema Location</td>
<td>The location of the XML schema (.xsd file) that was generated from the event output. For example: <code>&lt;OSB_Home&gt;\3rdparty\ApplicationAdapters\config\config_name\schemas\JDEdwards\target_name\jde-schema.xsd</code></td>
</tr>
</tbody>
</table>

**Note:** The schema validation options (Root, Namespace, Schema) are not applicable for the Oracle Application Adapter for J.D. Edwards OneWorld.

---

To generate a WSDL file for J.D. Edwards OneWorld event notification:

1. Create a channel in Application Explorer under the J.D. Edwards events node.
2. Start the channel.

**Important:** Do not restart the Oracle Application Server after the channel is started.
3. Send an inbound message from J.D. Edwards OneWorld.
4. Capture the inbound message payload in the log file located under:

   `<OSB_Home>\3rdparty\ApplicationAdapters\config\jca_sample\log\iwaf_jca1500.log`

   Alternatively, you can create a port with the File protocol under Event in Application Explorer to dispose the event message to the file system.
5. Use a third party tool such as XMLSpy to create the XSD schema using the XML payload captured in the previous step.
6. In the generated XML schema (.xsd file) perform the following modifications:

   a. Search for `Schemas-jdedwards-com` and replace it with `iwaysoftware`.
   
   b. Cut the following syntax:

   ```xml
   <xs:element name="jdeResponse">
   <xs:complexType>
   </xs:complexType>
   </xs:element>
   ```
7. Copy the XML schema (.xsd file) from the following directory:

<OSB_Home>/3rdparty/ApplicationAdapters/config/config_name/schemas/JDEdwards/target_name/

**Note:** Edit the created channel by providing the location of the schema (.xsd) file (as mentioned in step 7) in the PreParser tab of Application Explorer. For example:

<OSB_Home>/3rdparty/ApplicationAdapters/config/config_name/schemas/JDEdwards/target_name/jde-schema.xsd

8. Open a command prompt and navigate to the following directory:

<OSB_Home>/3rdparty/ApplicationAdapters/tools/iwaebin

9. Execute the `obadapter.bat` file to set the environment.

10. Navigate to the following directory where the XML schema (.xsd file) is copied:

<OSB_Home>/3rdparty/ApplicationAdapters/config/config_name/schemas/JDEdwards/target_name/

11. Enter the following command to generate a WSDL:

```
java -Diway.oem=oracle11g com.iwaysoftware.af.container.tools.wsdli.WayWSILBrowser adapterhome adapter target channel schemaPrefix wsdlFileName
```

where:

- `adapterhome` is the path to your ApplicationAdapters home. For example:
  
  <OSB_Home>/3rdparty/ApplicationAdapters/

- `adapter` is the name of the adapter. For example, JDEdwards.

- `target` is the name of the adapter target you created in Application Explorer.

- `channel` is the name of the channel you created in Application Explorer.

- `schemaPrefix` is the prefix for the XSD schema. The schema file must be in the same directory where the Java command is executed, for example:
  
  java -Diway.oem=oracle11g com.iwaysoftware.af.container.tools.wsdli.WayWSILBrowser C:\OSBP3\Oracle_OSB1\3rdparty\ApplicationAdapters\wsdls\JDEdwards jde812tgt jde_ch Jde812_Schema Jde812_salesorder_receive.wsdl

Once the command is executed, the following is displayed in the command window:

```
Running Inbound WSDL generation tool...
-> user.dir = java com.iwaysoftware.af.container.tools.wsdli.WayWSILBrowser C:\OSBP3\Oracle_OSB1\3rdparty\ApplicationAdapters\wsdls\JDEdwards jde812tgt jde_ch Jde812_Schema Jde812_salesorder_receive.wsdl
```
7.3.4 Importing the Inbound J.D. Edwards OneWorld WSDL File Into Oracle Service Bus

Once you have generated the inbound WSDL for J.D. Edwards OneWorld, you must import the inbound WSDL file into Oracle Service Bus.

1. Open the Oracle Service Bus Console and select the created WSDL folder in the left pane (for example, wsdls), as shown in Figure 7–63.
2. In the right pane, select **XML Schema** from the Create Resource menu. The Create a New XML Schema Resource page is displayed, as shown in Figure 7–64.

*Figure 7–64  Create a New XML Schema Resource Page*

<table>
<thead>
<tr>
<th>Resource Name*</th>
<th>Salesorder_attrb_http_inbound_reque</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Description</td>
<td></td>
</tr>
<tr>
<td>XML Schema*</td>
<td>C:\OSBPSS3\Oracle_OSB1\3rdparty\Appl</td>
</tr>
</tbody>
</table>

Save | Cancel
3. Enter a name for the inbound XML schema in the Resource Name field.
   You can also provide a brief description for the resource (optional).

4. Using the Browse option in the XML Schema field, navigate to the location of the
   inbound XML schema file (.xsd) on your file system.

5. Click Save.
   The inbound XML schema is added to the OSB console, as shown in Figure 7–65.

   **Figure 7–65  OSB Console**

   ![OSB Console](image-url)

<table>
<thead>
<tr>
<th>Name</th>
<th>Resource Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetEffectiveAddress_invoke</td>
<td>JCA Binding</td>
</tr>
<tr>
<td>GetEffectiveAddress_invoke</td>
<td>WSDL</td>
</tr>
<tr>
<td>GetEffectiveAddress_invoke_request</td>
<td>XML Schema</td>
</tr>
<tr>
<td>GetEffectiveAddress_invoke_response</td>
<td>XML Schema</td>
</tr>
<tr>
<td>Salesorder_optimize_http_inbound</td>
<td>WSDL</td>
</tr>
<tr>
<td>Salesorder_optimize_http_inbound_request</td>
<td>XML Schema</td>
</tr>
</tbody>
</table>

6. In the right pane, select WSDL from the Create Resource menu, as shown in
   Figure 7–66.

   **Figure 7–66  Create Resource Menu WSDL Option**

   ![Create Resource Menu WSDL Option](image-url)

<table>
<thead>
<tr>
<th>Name</th>
<th>Resource Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetEffectiveAddress_invoke</td>
<td>JCA Binding</td>
</tr>
<tr>
<td>GetEffectiveAddress_invoke</td>
<td>WSDL</td>
</tr>
<tr>
<td>GetEffectiveAddress_invoke_request</td>
<td>XML Schema</td>
</tr>
<tr>
<td>GetEffectiveAddress_invoke_response</td>
<td>XML Schema</td>
</tr>
<tr>
<td>Salesorder_optimize_http_inbound</td>
<td>WSDL</td>
</tr>
<tr>
<td>Salesorder_optimize_http_inbound_request</td>
<td>XML Schema</td>
</tr>
</tbody>
</table>

   The Create a New WSDL Resource page is displayed, as shown in Figure 7–67.
7. Enter a name for the inbound WSDL in the Resource Name field.
   You can also provide a brief description for the resource (optional).

8. Using the Browse option in the WSDL field, navigate to the location of the
   inbound WSDL file (.wsdl) on your file system.

9. Click Save.
   The inbound WSDL is added to the OSB console, as shown in Figure 7–68.

10. In the right pane, select JCA Binding from the Create Resource menu, as shown in
    Figure 7–69.
Enter a name for the inbound JCA binding in the Resource Name field. You can also provide a brief description for the resource (optional).

12. Using the Browse option in the JCA File field, navigate to the location of the inbound JCA binding file (.jca) on your file system.

13. Click Save.

The inbound JCA binding is added to the OSB console, as shown in Figure 7–71.
14. Click the icon that corresponds to the exported WSDL in the Actions column, as shown in Figure 7–72.

15. Provide a new WSDL name and a new service name in the corresponding fields, as shown in Figure 7–73.
16. In the Location area, select an available project and the sub-folder that is designated for Proxy Services.

17. Click Generate.

If the WSDL and Service resources are successfully created, then the following message is displayed, as shown in Figure 7–74.

**Figure 7–74 Success Message**

![WSDL and Service Resources were generated successfully.]

18. Select the Proxy Service folder you created in the left pane and verify that the Proxy Service is successfully generated and listed, as shown in Figure 7–75.
7.3.5 Configuring a File Type Business Service

Perform the following steps to configure a File type Business Service:

1. Select the Business Service folder you created in the left pane, as shown in Figure 7–76.

   Figure 7–76 Project Explorer

   ![Project Explorer]

2. In the right pane, select Business Service from the Create Resource menu, as shown in Figure 7–77.

   Figure 7–77 Create Resource Menu

   ![Create Resource Menu]

The General Configuration page is displayed, as shown in Figure 7–78.
3. Provide a name for the Business Service and from the Service Type area select **Messaging Service**.

4. Click **Next**.

The Message Type Configuration page is displayed, as shown in **Figure 7–79**.

**Figure 7–79  Message Type Configuration page**
5. Select **XML** as the Request Message Type and **None** as the Response Message Type.

6. Click **Next**.
   The Transport Configuration page is displayed, as shown in Figure 7–80.

**Figure 7–80  Transport Configuration Page**

7. Select **file** from the Protocol list.

8. Enter the path to a destination folder on your file system in the Endpoint URI field and click **Add**.

9. Click **Next**.
   The FILE Transport Configuration page is displayed, as shown in Figure 7–81.

**Figure 7–81  FILE Transport Configuration Page**

10. Enter the prefix and suffix for the output file to be received and click **Next**.
    The Message Handling and Advanced Settings page is displayed, as shown in Figure 7–82.
Configuring Inbound Processing Using Oracle Service Bus (J2CA Configuration)

11. Click Next.
   
   The following Summary page is displayed, as shown in Figure 7–83.

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

13. Check if the success message is displayed and click Activate in the Change Center, as shown in Figure 7–84.
7.3.6 Configuring a Pipeline

Perform the following steps to configure a Pipeline:

1. Click the Edit Message Flow icon in the row of the Proxy Service you created, as shown in Figure 7–87.
Figure 7–87  Edit Message Flow Icon

The Edit Message Flow workspace area is displayed, as shown in Figure 7–88.

Figure 7–88  Edit Message Flow Workspace Area

2. Click the displayed Proxy service icon and select Add Route from the menu. The RouteNode1 icon is added below the Request Pipeline icon, as shown in Figure 7–89.
3. Click the **RouteNode1** icon and select **Edit Route** from the menu. The Edit Stage Configuration workspace area is displayed, as shown in Figure 7–90.

4. Click **Add an Action**, select **Communication** from the menu, and click **Routing**, as shown in Figure 7–91.
5. Click <Service>.

The Select Service dialog is displayed, as shown in Figure 7–92.

6. Select a WSDL type Business Service and click Submit.

You are returned to the Edit Stage Configuration workspace area, as shown in Figure 7–93.
7. Click **Save**.

You are returned to the Edit Message Flow workspace area, as shown in Figure 7–94.

**Figure 7–94  Edit Message Flow Workspace Area**

8. Click **Save**, as shown in Figure 7–95.

**Figure 7–95  Success Message**

9. Check if the success message is displayed and click **Activate** in the Change Center.

The Activate Session page is displayed as shown in Figure 7–96.
10. Click Submit.

11. Check if the View Configuration Changes page is displayed, as shown in Figure 7–97.

12. Trigger an event from the J.D. Edwards OneWorld system and check if the output is received in the configured output location.

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

This section describes how to configure outbound processing using Oracle Service Bus for BSE configurations.

This section includes the following topics:

- Section 7.4.1, "Starting Oracle Service Bus and Creating Project Folders"
- Section 7.4.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"
- Section 7.4.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"
- Section 7.4.4, "Configuring a File Type Business Service"
- Section 7.4.5, "Configuring a WSDL Type Business Service"
- Section 7.4.6, "Configuring a Proxy Service"
7.4.1 Starting Oracle Service Bus and Creating Project Folders

Perform the following steps to start Oracle Service Bus and create project folders:

1. Start the Oracle WebLogic Server for the Oracle WebLogic Server domain that you have configured.

2. Open the Oracle Service Bus Console in a Web browser by entering the following URL:

   \[http://hostname:port/sbconsole\]

   Where \(hostname\) is the name of the machine where Oracle WebLogic Server is running and \(port\) is the port for the domain you are using. The port for the default domain is 7001.

   The Oracle Service Bus Console logon page is displayed.

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

   The Oracle Service Bus Console home page is displayed, as shown in Figure 7–98.

4. Click Create in the Change Center area to start a new Oracle Service Bus session, as shown in Figure 7–99.

---

**Figure 7–98** Oracle Service Bus Console Home Page

**Figure 7–99** Change Center Create Button
5. Click **Project Explorer** in the left pane, as shown in *Figure 7–100*.

*Figure 7–100  Project Explorer*

![Operations](image)

- Resource Browser
- **Project Explorer**
- Security Configuration
- System Administration

The Project Explorer page is displayed, as shown in *Figure 7–101*.

*Figure 7–101  Project Explorer Page*

![Project Explorer](image)

6. Provide a valid name for the new project (for example, JDE_Project) and click **Add Project**.

The project is successfully created and listed.

7. Click on the created project.

The project page is displayed, as shown in *Figure 7–102*.

*Figure 7–102  New Folder Name Field*

![New Folder Name Field](image)

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

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

10. In the Enter New Folder Name field, type **wsdl** and click **Add Folder**.

The Business Service, Proxy Service, and wsdls folders are listed in the left pane below the project node, as shown in *Figure 7–103*. 

*Figure 7–103  Business Service, Proxy Service, and wsdls folders*
7.4.2 Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus

Before starting and using Application Explorer to publish a WSDL directly to the Oracle Service Bus (OSB) Console (project/folder), OSB users must perform the following steps:

1. Open the command prompt window.
2. Navigate to the following directory:
   \<OSB_Home>\user_projects\domains\base_domain\bin

3. Execute `setDomainEnv.cmd` (Windows) or `. ./setDomainEnv.sh` (UNIX/Linux).
   This command sets the class path for Application Explorer to access the Oracle WebLogic Server APIs to publish the WSDLs to the OSB Console.

4. Do not close the command prompt window.
5. Navigate to the following directory:
   \<OSB_Home>\3rdparty\ApplicationAdapters\tools\iwae\bin

6. Execute `ae.bat` (Windows) or `iwae.sh` (UNIX/Linux) to start Application Explorer.
   You are now ready to publish WSDLs from Application Explorer to the OSB Console.

7.4.3 Publishing a WSDL From Application Explorer to Oracle Service Bus

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

1. Start Application Explorer, connect to a BSE configuration, and connect to a J.D. Edwards target, as shown in Figure 7–104.
2. Expand the Services node, followed by CALLBSFN and Addressbook.

3. Right-click the GetEffectiveAddress business object and select Create Web Service from the menu.

   The Create Web Service dialog is displayed, as shown in Figure 7–105.

**Figure 7–105  Create Web Service Dialog**

4. Enter a service name and click Next.

5. Click OK on the next dialog that is displayed.

   Application Explorer switches the view to the Business Services node, and the new Web service appears in the left pane, as shown in Figure 7–106.
6. Right-click the new Web service and select **Export WSDL** from the menu. The Export WSDL dialog is displayed, as shown in **Figure 7–107**.

**Figure 7–107  Export WSDL Dialog**

7. In the Name field, a default file name for the WSDL file is provided. You can accept the default or provide your own.

8. In the Location field, enter the location where you want to publish the WSDL document.

   The location is composed of an Oracle Service Bus project name and optionally, one or more folder names. The project name and any folder names must be separated by a forward slash character “/”.

9. In the Host field, enter the name of the machine where Oracle WebLogic Server is running.

10. In the Port field, enter the port for the domain you are using.

11. In the User field, enter your username to access Oracle Service Bus.

12. In the Password field, enter your password to access Oracle Service Bus.

13. Click **OK**.

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

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

1. Select the Business Service folder you created in the left pane, as shown in Figure 7–108.

2. In the right pane, select Business Service from the Create Resource menu.

   The General Configuration page is displayed, as shown in Figure 7–109.

3. Provide a name for the Business Service and from the Service Type area select Messaging Service.

4. Click Next.

   The Message Type Configuration page is displayed, as shown in Figure 7–110.
5. Select XML as the Request Message Type and None as the Response Message Type.

6. Click Next.

The Transport Configuration page is displayed, as shown in Figure 7–111.

Figure 7–111  Transport Configuration Page

7. Select file from the Protocol list.

8. Enter the path to a destination folder on your file system in the Endpoint URI field and click Add.

9. Click Next.

The FILE Transport Configuration page is displayed, as shown in Figure 7–112.
10. Enter the prefix and suffix for the output file to be received and click **Next**.

The Summary page is displayed, as shown in **Figure 7–113**.

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

### 7.4.5 Configuring a WSDL Type Business Service

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

1. Select the **Business Service** folder you created in the left pane, as shown in **Figure 7–114**.
2. In the right pane, select **Business Service** from the Create Resource menu. The General Configuration page is displayed, as shown in Figure 7–115.

![Create Resource Menu](image1)

**Figure 7–114  Create Resource Menu**

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

4. Click **Browse**.
   
The Select a WSDL dialog is displayed, as shown in Figure 7–116.
5. Select a BSE WSDL that you published for Oracle Application Adapter for J.D. Edwards OneWorld using Application Explorer and click Submit.

The Select a WSDL Definition dialog is displayed, as shown in Figure 7–117.

Figure 7–117 Select a WSDL Definition Dialog
6. Select the WSDL definition under the Ports section and click **Submit**.
   You are returned to the General Configuration page where the WSDL you selected is now available.

7. Click **Next**.
   The Transport Configuration page is displayed, as shown in **Figure 7–118**.

**Figure 7–118  Transport Configuration Page**

![Transport Configuration Page](image1.png)

8. Leave the default values and click **Next**.
   The HTTP Transport Configuration page is displayed, as shown in **Figure 7–119**.

**Figure 7–119  HTTP Transport Configuration Page**

![HTTP Transport Configuration Page](image2.png)

9. Leave the default values and click **Next**.
   The SOAP Binding Configuration page is displayed, as shown in **Figure 7–120**.
10. Click Next.

The Message Content Handling page is displayed, as shown in Figure 7–121.

11. Click Next.

The Summary page is displayed, as shown in Figure 7–122.

12. Review all the information for your Business Service and click Save.
7.4.6 Configuring a Proxy Service

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

1. Select the **Proxy Service** folder you created in the left pane, as shown in Figure 7–123.

![Create Resource Menu](image)

**Figure 7–123 Create Resource Menu**

2. In the right pane, select **Proxy Service** from the Create Resource menu.

The General Configuration page is displayed, as shown in Figure 7–124.

![General Configuration Page](image)

**Figure 7–124 General Configuration Page**

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

4. Click **Next**.

The Transport Configuration page is displayed, as shown in Figure 7–125.
5. Select file from the Protocol list.
6. Enter the path to an input folder on your file system in the Endpoint URI field.
7. Click Next.
   The FILE Transport Configuration page is displayed, as shown in Figure 7–126.

9. Click Next.
The Message Content Handling page is displayed, as shown in Figure 7–127.

**Figure 7–127 Message Content Handling Page**

10. Click Next.

The Summary page is displayed, as shown in Figure 7–128.

**Figure 7–128 Summary Page**
11. Review all the information for your Proxy Service and click **Save**.

### 7.4.7 Configuring a Pipeline

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

1. Click the **Edit Message Flow** icon in the row of the Proxy Service you created, as shown in Figure 7–129.

**Figure 7–129  Edit Message Flow Icon**

The Edit Message Flow workspace area is displayed.

2. Click the **Proxy Service** icon and select **Add Pipeline Pair** from the menu, as shown in the following image, as shown in Figure 7–130.

**Figure 7–130  Add Pipeline Pair Option**

3. Click the **Request Pipeline** icon and select **Add Stage** from the menu, as shown in Figure 7–131.
The Stage1 icon is added below the Request Pipeline icon, as shown in Figure 7–132.

Figure 7–132  Edit Stage Option

4. Click the Stage1 icon and select Edit Stage from the menu.

The Edit Stage Configuration workspace area is displayed, as shown in Figure 7–133.
5. Click Add an Action, select Reporting from the menu, and click Log, as shown in Figure 7–134.

![Figure 7–133 Edit Stage Configuration Workspace Area](image)

6. Click <Expression> to edit the expression.
   The XQuery/XSLT Expression Editor is displayed, as shown in Figure 7–135.

![Figure 7–134 Edit Stage Configuration Workspace Area](image)
7. In the XQuery Text area, type \$inbound.

8. Click **Validate** and then **Save**.

   You are returned to the Edit Stage Configuration workspace area, as shown in Figure 7–136.

**Figure 7–136  Edit Stage Configuration Workspace Area**

9. Type any annotation/comments in the text box (for example, **********input processing**********).

10. Select **Error** from the severity level list.

11. Add one more Log action as shown in the following image, as shown in Figure 7–137.
A new Log configuration is added, as shown in the following image, as shown in Figure 7–138.

12. Click <Expression> to edit the expression.

The XQuery/XSLT Expression Editor is displayed, as shown in Figure 7–139.
13. In the XQuery Text area, type $\text{body}$.

14. Click Validate and then Save.

You are returned to the Edit Stage Configuration workspace area, as shown in Figure 7–140.

15. Type any annotation/comments in the text box (for example, **********Request Body**********).
16. Select **Error** from the severity level list.

17. Click **Validate** and then **Save**, as shown in **Figure 7–141**.

**Figure 7–141  Add Route Option**

18. Click the **PipelinePairNode1** icon and select **Add Route** from the menu, as shown in **Figure 7–142**.

**Figure 7–142  Edit Route Option**

19. Click the **RouteNode1** icon and select **Edit Route** from the menu, as shown in **Figure 7–143**.
20. Click **Add an Action**, select **Communication** from the menu, and click **Routing**, as shown in *Figure 7–144*.

21. Click `<Service>`.

   The Select Service dialog is displayed, as shown in *Figure 7–145*. 
22. Select a WSDL type Business Service and click **Submit**.

You are returned to the Edit Stage Configuration workspace area, as shown in Figure 7–146.

**Figure 7–146  Edit Stage Configuration Workspace Area**

23. Select **GetEffectiveAddress** as the operational attribute from the list.

24. Click **Validate** and then **Save**, as shown in Figure 7–147.
25. In the Response Actions area, click **Add an Action**, select **Reporting** from the menu, and click **Log**.

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

   The XQuery/XSLT Expression Editor is displayed, as shown in **Figure 7–148**.

**Figure 7–148  XQuery/XSLT Expression Editor**

27. In the XQuery Text area, type `$outbound`.

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

   You are returned to the Edit Stage Configuration workspace area, as shown in **Figure 7–149**.
29. Type any annotation/comments in the text box (for example, "**********output processing**********").

30. Select Error from the severity level list.

31. Click Validate and then Save, as shown in Figure 7–150.

**Figure 7–150 Add Stage Option**

32. Click the Response Pipeline icon and select Add Stage from the menu.

The Stage1 icon is added below the Response Pipeline icon, as shown in Figure 7–151.
33. Click the Stage1 icon and select Edit Stage from the menu. The Edit Stage Configuration workspace area is displayed, as shown in Figure 7–152.

34. Click Add an Action, select Communication from the menu, and click Publish, as shown in Figure 7–153.
35. Click `<Service>`.

The Select Service dialog is displayed, as shown in Figure 7–154.

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

You are returned to the Edit Stage Configuration workspace area, as shown in Figure 7–155.
37. In the Request Actions area, click **Add an Action**, select **Reporting** from the menu, and click **Log**, as shown in Figure 7–156.

38. Click **<Expression>** to edit the expression.
   The XQuery/XSLT Expression Editor is displayed, as shown in Figure 7–157.
39. In the XQuery Text area, type $body.

40. Click Validate and then Save.

You are returned to the Edit Stage Configuration workspace area, as shown in Figure 7–158.

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

42. Select Error from the severity level list.

43. Click Validate and then Save.

The completed Pipeline is displayed, as shown in Figure 7–159.
44. Click Save.

45. Click **Activate** in the Change Center area to activate your changes in the Oracle Service Bus session, as shown in *Figure 7–160*.

*Figure 7–160  Change Center Activate Button*

46. Copy and paste an input XML file in the input folder you have configured, as shown in *Figure 7–161*.
An output XML file is received in the destination folder, as shown in Figure 7–162.

**Figure 7–161  Input Folder**

**Figure 7–162  Output XML Destination Folder**
This chapter explains the limitations and workarounds when connecting to J.D. Edwards OneWorld. It contains the following topics:

- Section 8.1, "Troubleshooting"
- Section 8.2, "BSE Error Messages"

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

8.1 Troubleshooting

This topic provides troubleshooting information for J.D. Edwards OneWorld, separated into four categories:

- Oracle Adapter Application Explorer (Application Explorer)
- J.D. Edwards OneWorld
- Oracle Adapter J2CA
- Oracle Adapter Business Services Engine (BSE)

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

- The Oracle Adapter J2CA trace information can be found under the following directory:
  
  `<ADAPTER_HOME>\soa\thirdparty\ApplicationAdapters\config\config_name\log`

- BSE trace information can be found under the following directory:
  
  `<ADAPTER_HOME>\user_projects\domains\base_domain\servers\soa_server1\stage\ibse\ibse.war\ibselogs`

- The log file for Application Explorer can be found under the following directory:
  
  `<ADAPTER_HOME>\soa\thirdparty\ApplicationAdapters\tools\iwae\bin`
Troubleshooting

Application Explorer
To use Application Explorer on Windows for debugging or testing purposes:

1. Ensure that Oracle WebLogic Server is started, which is where Application Explorer is deployed.
2. Start Application Explorer by clicking the Windows Start menu, selecting All Programs, Oracle Application Adapters, and clicking Application Explorer, as shown in Figure 8–1.

Figure 8–1 Oracle Application Adapters Program Menu

You can also start Application Explorer by executing the ae.bat file, which is located in the following directory:

<ADAPTER_HOME>\soa\thirdparty\ApplicationAdapters\tools\iwas\bin\ae.bat

It is a good practice to create a shortcut for the ae.bat file on your desktop.

If you are using a UNIX or Linux platform, then you can start Application Explorer by executing the iwae.sh file.

<table>
<thead>
<tr>
<th>Error</th>
<th>Solution</th>
</tr>
</thead>
</table>
| Cannot connect to Oracle Application Adapter for J.D. Edwards OneWorld from Application Explorer: Problem activating adapter. (Failed to connect to J.D.Edwards OneWorld, check system availability and configuration parameters:...) Check logs for more information. | Ensure that:  
- J.D. Edwards OneWorld is running.  
- The J.D. Edwards OneWorld user ID and password is correct.  
- The port number is correct. |
| The following error message appears: java.lang.IllegalStateException: java.lang.Exception: Error Logon to J.D. Edwards OneWorld System J.D. Edwards OneWorld does not appear in the Application Explorer Adapter node list. Logon failure error at run-time. | You have provided invalid connection information for J.D. Edwards OneWorld or the wrong JAR file is in the lib directory. Ensure that the J.D. Edwards OneWorld JAR files, are added to the lib directory. If the password for connecting to your J.D. Edwards OneWorld system is not specified when creating a target or with the Edit option in Application Explorer, then you are unable to connect to J.D. Edwards OneWorld. The connection password is not saved in repository.xml. Update the password using the Edit option in Application Explorer, then restart the application server. |
### Troubleshooting

**Error**
The following exception occurs when you start Application Explorer by activating `ae.bat` (not `iaexplorer.exe`):

```java
java.lang.ClassNotFoundException: org.bouncycastle.jce.provider.BouncyCastleProvider
```

This is a benign exception. It does not affect adapter functionality. Download BouncyCastle files from:

ftp://ftp.bouncycastle.org/pub

**Error**
Unable to start Application Explorer in a Solaris environment. The following exception is thrown in the console:

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

JAVACMD is not set on the user system. Before starting Application Explorer, export JAVACMD as follows:

```java
JAVACMD=/<jdk_home>/bin/java, where <jdk_home> is the directory where JDK is installed on your system.
```

---

### J.D. Edwards OneWorld

<table>
<thead>
<tr>
<th>Error</th>
<th>Cause</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action code invalid</td>
<td>In the Sales Order request, the Action code appears as &quot;H,&quot; an invalid action code.</td>
<td>Use:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;I&quot; for inquiry.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;C&quot; for change.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;D&quot; for delete.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &quot;A&quot; to add a new record.</td>
</tr>
<tr>
<td>Invalid address number.</td>
<td>The address number does not exist in the Address Book Master file (F0101).</td>
<td>Enter an address number using the Address Book Revisions program (PO1051). Ensure that the number entered is correct.</td>
</tr>
<tr>
<td>Record invalid</td>
<td>The record being processed either already exists for an ADD function or does not exist for an INQUIRY, CHANGE, or DELETE function.</td>
<td>If you are attempting to inquire, change, or delete a record you added previously, then there could be database problems in your production library. Contact your data processing department.</td>
</tr>
<tr>
<td>Item Branch record does not exist.</td>
<td>An Item Branch record (F4102) does not exist for this item in the Branch/Plant specified.</td>
<td>Correct the Branch or enter an Item Branch record for this item in Branch Plant Item Information (P41026).</td>
</tr>
</tbody>
</table>
8.2.1 General Error Handling in BSE

BSE serves as both a SOAP gateway into the adapter framework and as the engine for some of the adapters. In both design time and run-time, various conditions can cause errors in BSE when Web services that use adapters run. Some of these conditions and resulting errors are exposed the same way, regardless of the specific adapter; others are exposed differently, based on the adapter being used. This topic explains what you can expect when you encounter some of the more common error conditions on an adapter-specific basis. Usually the SOAP gateway (agent) inside BSE passes a SOAP request message to the adapter required for the Web service. If an error occurs, then how it is exposed depends on the adapter and the API or interfaces that the adapter uses. A few scenarios cause the SOAP gateway to generate a SOAP fault. In general, anytime the SOAP agent inside BSE receives an invalid SOAP request, a SOAP fault element is generated in the SOAP response. The SOAP fault element contains fault string and fault code elements. The fault code contains a description of the SOAP agent error. The following SOAP response document results when BSE receives an invalid SOAP request:

```xml
    <SOAP-ENV:Fault>
        <SOAP-ENV:Code>
            <soap:Value>Client</soap:Value>
            <soap:SubCode>
                <soap:Value>Error</soap:Value>
            </soap:SubCode>
        </SOAP-ENV:Code>
        <SOAP-ENV:Reason>
            <soap:Text>&1 does not match any of the valid values.</soap:Text>
        </SOAP-ENV:Reason>
        <SOAP-ENV:FaultString>&1 does not match any of the valid values specified in the Data Dictionary for this field.</SOAP-ENV:FaultString>
    </SOAP-ENV:Fault>
</SOAP-ENV:Envelope>
```

### Oracle Adapter J2CA

**Error**

In Application Explorer, the following error message appears when you attempt to connect to an Oracle Adapter J2CA configuration:

```
Could not initialize JCA
```

**Solution**

In the Details tab in the right pane, ensure that the directory specified in the Home field points to the correct directory, for example:

```
<ADAPTER_HOME>/soa/thirdparty/ApplicationAdapters\tools\iwae\bin/..../..
```

### 8.2 BSE Error Messages

This topic discusses the different types of errors that can occur when processing Web services through BSE.
In this example, BSE did not receive an element in the SOAP request message that is mandatory for the WSDL for this Web service.

8.2.2 Adapter-Specific Error Handling

When an adapter raises an exception during run-time, the SOAP agent in BSE produces a SOAP fault element in the generated SOAP response. The SOAP fault element contains fault code and fault string elements. The fault string contains the native error description from the adapter target system. Since adapters use the target system interfaces and APIs, whether an exception is raised depends on how the target systems interface or API treats the error condition. If a SOAP request message is passed to an adapter by the SOAP agent in BSE, and that request is invalid based on the WSDL for that service, then 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.

Invalid SOAP Request

If Oracle WebLogic 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.

```xml
<?xml version="1.0" encoding="ISO-8859-1" ?>
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>RPC server connection failed: Connection refused: connect</faultstring>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Empty Result From Oracle WebLogic Server Adapter Request

If Oracle WebLogic Server 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.

```xml
  <SOAP-ENV:Body>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

**Note:** The condition for this adapter does not yield a SOAP fault.
Error Logging In

If Oracle WebLogic Server Adapter executes an invalid SOAP log in request, then the following SOAP response is generated.

```
<SOAP-ENV:Envelope xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Header>
    <m:ibsinfo xmlns:m="urn:schemas-iwaysoftware-com:iwse">
      <m:service>B0100033</m:service>
      <m:method>GetEffectiveAddress</m:method>
      <m:license>test</m:license>
      <m:Username>user</m:Username>
      <m:Password>password</m:Password>
    </m:ibsinfo>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <m:GetEffectiveAddress
      xmlns:m="urn:iwaysoftware:ibse:jul2003:GetEffectiveAddress">
```

[861] Error (SOAP1) W.SOAP1.2: Attempting string, no encoding recognized in document

```
</m:GetEffectiveAddress>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>```
Empty Result From Oracle WebLogic Server Adapter Request

If Oracle WebLogic Server 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.
Invalid Call Method

If an invalid call is made to Oracle WebLogic Server Adapter, then the following SOAP response is generated.

```xml
  soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAPAction="urn:iwaysoftware:ibse:jul2003:GetEffectiveAddress:response">
<SOAP-ENV:Body>
<GetEffectiveAddressResponse xmlns="urn:iwaysoftware:ibse:jul2003:GetEffectiveAddress:response">
<jdeResponse user="JDE" type="callmethod" environment="DV7333">
<callMethod name="GetEffectiveAddress">
<returnCode code="2"/>
<params>
<param name="mnAddressNumber">12345</param>
</params>
</callMethod>
</jdeResponse>
</GetEffectiveAddressResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
<m:GetEffectiveAddress xmlns:m="urn:iwaysoftware:ibse:jul2003:GetEffectiveAddress">  
  <m:jdeRequest type="callmethod">  
    <m:callMethod name="GetAddress">  
      <m:params>  
        <m:param name="mnAddressNumber">34518</m:param>  
      </m:params>  
      <m:onError/>  
    </m:callMethod>  
  </m:jdeRequest>  
</m:GetEffectiveAddress>
DEBUG (SOAP1) W.SOAP1.2: Agent returned success
INFO  (manager) MGR00X02: Removing active worker: W.SOAP1.2
DEBUG (SOAP1) W.SOAP1.2: doing docTran, docVal, listTran for agent(1)
DEBUG (SOAP1) W.SOAP1.2: sendToAll reply to XDReply:
[protocol=http */null]
DEBUG (SOAP1) W.SOAP1.2: preemitters from doc: null
DEBUG (SOAP1) W.SOAP1.2: no preemitters, emitting contents of doc, usestream=false encoding=UTF-8
DEBUG (SOAP1) W.SOAP1.2: writeEntity, len: 595 data:
<?xml version="1.0" encoding="UTF-8" ?>
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Body><GetEffectiveAddressResponse xmlns="urn:iwaysoftware:ibse:jul2003:GetEffectiveAddress:response">
<jdeResponse user="JDE" type="callmethod" environment="DV7333">
<callMethod name="GetAddress">
<params>
</params></callMethod></jdeResponse></GetEffectiveAddressResponse></SOAP-ENV:Body>
</SOAP-ENV:Envelope>
DEBUG (SOAP1) W.SOAP1.2: writeString: HTTP/1.0
DEBUG (SOAP1) W.SOAP1.2: writeString: 200
DEBUG (SOAP1) W.SOAP1.2: writeString: OK
DEBUG (SOAP1) W.SOAP1.2: writeString: Content-Type:
DEBUG (SOAP1) W.SOAP1.2: writeString: text/xml
DEBUG (SOAP1) W.SOAP1.2: writeString: Content-Length:
DEBUG (SOAP1) W.SOAP1.2: writeString: 595
INFO  (SOAP1) W.SOAP1.2: W0000X13: Ended message processing, rc=0
DEEP  (SOAP1) W.SOAP1.2: storing used socket
DEBUG (SOAP1) W.SOAP1.2: entering waitForDocument
DEEP  (SOAP1) W.SOAP1.2: cleanup: closing sockets(0)
This chapter contains the following topics:

- Section 9.1, "Web Services Policy-Based Security"
- Section 9.2, "Migrating Repositories"

9.1 Web Services Policy-Based Security

Oracle Adapter Application Explorer (Application Explorer) provides a security model called Web services policy-based security. This section describes 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, but 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 (BS) that can be applied to an existing or new BS. When you set specific rights or privileges inside a policy, you do not have to re-create privileges for every BS 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 Business Services 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, then other methods in that Business Service are not 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 Oracle Adapter Business Services Engine (BSE) in the SOAP request message are verified 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.

9.1.1 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, newtest. For information on creating a new configuration, see Chapter 2, "Configuring Oracle Application Adapter for J.D. Edwards OneWorld".
3. Select Connect.

Nodes appear for Adapters and Business Services (also known as Web services), as shown in Figure 9–1.

![Figure 9–1 Service Nodes](image)

Perform the following steps:

a. Expand the Business Services node.
b. Expand the Configuration node.
c. Expand the Security node.
d. Expand the Users and Groups node, as shown in Figure 9–2.

![Figure 9–2 User Group Node](image)
4. Right-click **Users** and click **New User**.

   The New User dialog is displayed, as shown in **Figure 9–3**.

**Figure 9–3  New User Dialog**

![New User Dialog](image)

Perform the following steps:

a. In the **Name** field, enter a user ID.

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

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

5. Click **OK**.

   The new user is added under the Users node.

**Creating a Group to Use with a Policy**

To create a group to use with a policy:

1. Start Application Explorer.

2. Right-click the configuration to which you want to connect, for example, **newtest**. For information on creating a new configuration, see Chapter 2, "Configuring Oracle Application Adapter for J.D. Edwards OneWorld".

3. Select **Connect**.

   Nodes appear for Adapters and Business Services (also known as Web services), as shown in **Figure 9–4**.

**Figure 9–4  Service Node**

![Service Node](image)

Perform the following steps:

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

b. Expand the **Configuration** node.

c. Expand the **Security** node.

d. Expand the **Users and Groups** node, as shown in **Figure 9–5**.
4. Right-click **Groups** and select **New Group**.

   The New Group dialog is displayed, as shown in **Figure 9–6**.

**Figure 9–6** New Group Dialog

Perform the following steps:

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 new group is added under the Groups node.

**Creating an Execution Policy**

An execution policy determines 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. For information on creating a new configuration, see Chapter 2, “Configuring Oracle Application Adapter for J.D. Edwards OneWorld”.

3. Select Connect.

Nodes appear for Adapters and Business Services (also known as Web services), as shown in Figure 9–7.

**Figure 9–7 Service Nodes**

Perform the following steps:

a. Expand the Business Services node.

b. Expand the Configuration node.

c. Expand the Security node.

d. Expand the Policies node, as shown in Figure 9–8.

**Figure 9–8 Policies Node**

4. Right-click Policies and select New Policy.

The New Policy dialog is displayed, as shown in Figure 9–9.

**Figure 9–9 New Policy Dialog**
Perform the following steps:

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 verified 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 selected, click OK.

6. Click Next.

   The New Policy permissions dialog is displayed, as shown in Figure 9–10.

**Figure 9–10 New Policy Permissions Dialog**

7. To grant permission to a user or group to execute a Business Service, select the user or group and move them into the Execution Granted list by selecting the double left facing arrow.

8. To deny permission to a user or group to execute a Business Service, select the user or group and move them into the Execution Denied list by selecting the double right facing arrow.

9. Click OK.

   The following pane summarizes your configuration, as shown in Figure 9–11.
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. For information on creating a new configuration, see Chapter 2, “Configuring Oracle Application Adapter for J.D. Edwards OneWorld”.
3. Select Connect.

   Nodes appear for Adapters 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, as shown in Figure 9–12.

Figure 9–12  IP and Domain Node

The New IP and Domain Restriction dialog is displayed, as shown in Figure 9–13.
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, then you must provide the IP address for that computer. If you only know the DNS name for the computer, then click DNS Lookup to obtain the IP Address based on the DNS name.
   - If you select Group (of Computers), then you must provide the IP address and subnet mask for the computer group.
   - If you select Domain, then 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, as shown in Figure 9–14.

Figure 9–14  Summary

- **IP Address(Mask)/Domain**: www.yahoo.com
- **Type**: Domain
- **Access**: Denied
- **Description**
9.2 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:8001/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, as shown in Figure 9–15.

   **Figure 9–15  Create New SOAP Request Option**

   ![Create New SOAP Request Option](image)

   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:8001/ibse/IBSEServlet/admin/iwcontrol.ibs?wsdl

4. Select Create new SOAP request.

   The WSDL file location dialog is displayed, as shown in Figure 9–16.

   **Figure 9–16  WSDL File Location Dialog**

   ![WSDL File Location Dialog](image)

   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:8001/ibse/IBSEServlet/admin/iwcontrol.ibs?wsdl

5. Click OK.

   The soap operation name dialog is displayed and lists the available control methods, as shown in Figure 9–17.
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 shows the structure of the SOAP envelope, as shown in Figure 9–18.

7. Locate the Text view icon in the toolbar, as shown in Figure 9–19.
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="">
  <m:repositorysetting>
    <m:rname>oracle</m:rname>
    <m:rconn>String</m:rconn>
    <m:rdriver>String</m:rdriver>
    <m:ruser>String</m:ruser>
    <m:rpwd>String</m:rpwd>
  </m:repositorysetting>
  <m:servicename>String</m:servicename>
</m:MIGRATEREPO>
```

Perform the following steps:

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

   The Oracle repository URL has the following format:
   
   `jdbc:oracle:thin:@[host]:[port]:[sid]`

b. For the `<m:rdriver>` tag, replace the String placeholder with the location of your Oracle driver.

c. For the `<m:ruser>` tag, replace the String placeholder with a valid user name to access the Oracle repository.

d. For the `<m:rpwd>` tag, replace the String placeholder with a valid password to access the Oracle repository.

10. Perform one of the following migration options.

   - If you want to migrate a single Web service from the current BSE repository, then enter the Web service name in the `<m:servicename>` tag, for example:
     
     `<m:servicename>JDEService1</m:servicename>`

   - If you want to migrate multiple Web services from the current BSE repository, then duplicate the `<m:servicename>` tag for each Web service, for example:
     
     `<m:servicename>JDEService1</m:servicename>
     <m:servicename>JDEService2</m:servicename>`

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

11. From the menu bar, click **SOAP** and select **Send request to server**, as shown in Figure 9–20.
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:
   
   `<ADAPTER_HOME>`/`soa/thirdparty/ApplicationAdapters/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.
J.D. Edwards OneWorld enables you to specify inbound functionality for Master Business Functions (MBF).

This chapter describes how to enable outbound and inbound transaction processing in J.D. Edwards OneWorld and how to modify the *jde.ini* file for XML support. It contains the following topics:

- Section A.1, "Modifying the JDE.INI File for Outbound and Inbound Processing"
- Section A.2, "Using the GenJava Development Tool (Outbound Processing)"
- Section A.3, "Triggering J.D. Edwards OneWorld Events"

### A.1 Modifying the JDE.INI File for Outbound and Inbound Processing

This section describes the settings that are required in the JDE.INI file for the XML call object kernel (outbound and inbound processing).

Open the JDE.INI file and modify the [JDENET_KERNEL_DEF6] and [JDENET_KERNEL_DEF15] sections as follows:

```
[JDENET_KERNEL_DEF6]
krnlName=CALL OBJECT KERNEL
dispatchDllName=XMLCallObj.dll
dispatchDllFunction=_XMLTransactionDispatch@28
maxNumberOfProcesses=1
numberOfAutoStartProcesses=1

[JDENET_KERNEL_DEF15]
krnlName=XML TRANSACTION KERNEL
dispatchDllName=XMLTransactions.dll
dispatchDllFunction=_XMLTransactionDispatch@28
maxNumberOfProcesses=1
numberOfAutoStartProcesses=1
```

The parameters containing an underscore (_) and @28 are for Windows NT operating systems only. For other operating systems, replace the parameters with the values in the following table:

<table>
<thead>
<tr>
<th>Operating System</th>
<th>Call Object dispatch DLLName</th>
<th>XML Trans dispatch DLLName</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS400</td>
<td>XMLCALLOBJ</td>
<td>XMLTRANS</td>
</tr>
</tbody>
</table>

---

Configuring J.D. Edwards OneWorld for Outbound and Inbound Processing  A-1
A.2 Using the GenJava Development Tool (Outbound Processing)

This section describes how to use the GenJava development tool, which is used to create Java wrappers for accessing the J.D. Edwards business functions. The Oracle Application Adapter for J.D. Edwards OneWorld uses these wrappers to call the J.D. Edwards business functions.

J.D. Edwards provides a Java Generation tool called GenJava that you can use to expose J.D. Edwards business functions externally as Java class files. A J.D. Edwards system administrator usually runs the GenJava tool.

During GenJava operation, you must specify a library of business functions, for example CALLBSFN. GenJava creates the associated Java class files for the business functions and related data structures. GenJava also compiles the business functions, generates Java documents, and packages them into two JAR files. One JAR file contains Java classes and the second JAR file contains Java documents.

For example, if the business function library you specified in GenJava is CALLBSFN, the following files are found in the <install>\system\classes directory or any user-specified directory redirected by GenJava:

- JDEJAVA_CALLBSFN.xml
- JDEJAVA_CALLBSFNInterop.jar
- JDEJAVA_CALLBSFNInteropDoc.jar

Once they are generated, these library files must be added to the CLASSPATH.

GenJava also provides access to J.D. Edwards business functions by generating pure Java interfaces for these business functions. GenJava can be generated from a thick client or a deployment server.

Running GenJava

GenJava is located in the <install>\system\bin32 directory. You run GenJava from the command line. There are two GenJava command options that can be used to generate the wrappers.

GenJava Command Option 1

The following command generates Java wrappers for Category 1 (Master Business Functions), Category 2 (Major Business Functions), Category 3 (Minor Business Functions), and Category - (Uncategorized Business Functions) in the CALLBSFN library:

```
GenJava /Cat 1 /Cat 2 /Cat 3 /Cat - CALLBSFN
```
GenJava Command Option 2
The GenJava command can also be run with a JDEScript file and prompts a J.D. Edwards log on window, where you must enter a valid user ID, password, and environment.

1. Using an editor, create a new file called AddressBook.cmd and enter the following commands:

```
define library CALLBSFN
login
library CALLBSFN
interface AddressBook
import B0100031
import B0100019
import B0100032
import B0100002
import B0100033
build
logout
```

2. Run the following GenJava command:

```
GenJava /cmd .\AddressBook.cmd
```

3. GenJava generates the wrappers (CALLBSFNInterop.jar, CALLBSFNInteropDoc.jar, and CALLBSFN.xml) in Java for all business functions that are mentioned in the script file.

A.3 Triggering J.D. Edwards OneWorld Events
The flow of inbound data to third parties is controlled through the Data Export Controls application. For each transaction type and order type, one or more records can be defined with different function names and libraries.

1. Type P0047 in the Fast Path field and press Enter as shown in, Figure A–1.

**Figure A–1  JD Edwards Solution Explorer**

![JD Edwards Solution Explorer](image)

The Work With Data Export Controls window is displayed, as shown in Figure A–2.
2. Click Add.

The Data Export Control Revisions window is displayed. Notice that the sequence (Seq) number automatically increments for each new line, as shown in Figure A–3.

Figure A–3  Data Export Control Revisions Window

3. Perform the following steps:
   a. Type JDESOOUT in the Transaction field.
   b. Type SO in the Order Type field.
   c. Type NotifyOnUpdate in the first row of the Function Name column.
   d. Type the absolute path to the location of the iwoevent.dll file in the first row of the Function Library column, for example: D:\JDEdwards\E812\DDP\Outbound\iwoevent.dll
   e. Type 1 in the first row of the Execute for Add column if you want the notifications for add/insert.
   f. Make the same decision for update, delete, and inquiry and type 1 in the appropriate column.
   g. Type 1 in the Launch Immediately column to launch the Outbound Subsystem batch process (R00460).

4. Click OK, as shown in Figure A–4.
Starting the Outbound Scheduler Subsystem Process (R00460)

Once you have finished defining one or more records for each transaction type and order type, you must manually start the outbound scheduler subsystem process.

1. Type BV in the Fast Path field and press Enter, as shown in Figure A–5.

2. Type R00460 in the Batch Application field and click Find, as shown in Figure A–7.
3. Select Interoperability Generic Outbound Subsystem UBE (XJDE0001) and click Select.
   The Version Prompting window is displayed, as shown in Figure A–8.

4. Click Submit, as shown in Figure A–9.

5. Navigate to the last screen and click OK.

Verifying the Subsystem Process
This section describes how to verify the outbound scheduler subsystem process.
1. Type WSJ in the Fast Path field and press Enter, as shown in Figure A–10.
Figure A–10  JD Edwards Solution Explorer

The Work With Server (Subm Jobs) window is displayed, as shown in Figure A–11.

Figure A–11  Work With Server (Subm Jobs) window

2. Select a corresponding server from the table.
3. Click Row from the menu bar and select Subsystem Jobs, as shown in Figure A–12.

Figure A–12  Find Option Selection
4. Click Find, as shown in Figure A–13.

Figure A–13  Job Status Column

5. Verify that R is listed in the Job Status column.

Configuring P4210 (Sales Order) to Trigger an Event
This section describes how to configure a P4210 (Sales Order) to trigger an event.

1. Type IV in the Fast Path field and press Enter, as shown in Figure A–14.

Figure A–14  JD Edwards Solution Explorer

The Interactive Versions window is displayed, as shown in Figure A–15.
2. Type P4210 in the Interactive Application field and click Find, as shown in Figure A–16.

3. Select a document version from the table, for example, RIS0001 - Sales Order Entry - SO Order Type.

4. Click Row from the menu bar and select Processing Options.

   The Processing Options dialog is displayed, as shown in Figure A–17.
5. Click the Interop tab.
6. Type JDESOOUT in the Transaction Type field.
7. Add Sales Order.

Verifying the Configuration Steps
This section describes how to verify the configuration steps by updating F0046.
1. Type P0046 in the Fast Path field and press Enter, as shown in Figure A–18.

Figure A–18 JD Edwards Solution Explorer

The P0046 - Work With Processing Log window is displayed, as shown in Figure A–19.
2. Click Find.

The following data is displayed, as shown in Figure A–20.

![Figure A–20 Data Display Table](image)

3. Search for the corresponding transaction.

The \iwoevnt.log file is created in the following directory:

```
\iuwJDE812\JDEedwards\E812\DDP\system\bin32
```

The \iwoevent.log file is created in the outbound folder where the \iwoevent.dll and \iwoevent.cfg files are located. The following is an example of the event log file:

```
Event call begin...
Server time : Tue May 27 07:23:55 2008
userId : JDE
batchNumber : 15205
transactionNumber: 103494
lineNumber : 1.000000
transactionType : JDESOOUT
sequenceNumber : 1.000000
Request xml:

==========
```
adapter
Provides universal connectivity by enabling an electronic interface to be accommodated (without loss of function) to another electronic interface.

agent
Supports service protocols in listeners and documents.

business service
Also known as a Web service. A Web service is a self-contained, modularized function that can be published and accessed across a network using open standards. It is the implementation of an interface by a component and is an executable entity.

channel
Represents configured connections to particular instances of back-end systems. A channel binds one or more event ports to a particular listener managed by an adapter.

listener
A component that accepts requests from client applications.

port
Associates a particular business object exposed by the adapter with a particular disposition. A disposition is a URL that defines the protocol and location of the event data. The port defines the end point of the event consumption.
Index

A
access methods, 2-31
access rights, 9-2
action codes, 8-3
adapter configuration
requirements, 5-1
updating, 3-2
adapter exceptions, 8-2 to 8-5
adapter integration
BPEL Process Manager and, 4-1
InterConnect and, 5-1
adapter types, 1-1, 1-7
resource, 3-1 to 3-2
adapters, 1-1 to 1-8
configuring, 3-2
deploying, 1-6, 3-1 to 3-2, 4-1
integrating, 5-1
integrating with BPEL Process Manager, 6-1
troubleshooting, 8-1 to 8-11
Add Channel dialog box, 4-41
Add Target dialog box, 2-6
Address Book Master file, 8-3
Address Book Revisions program, 8-3
address numbers, 8-3
agents, 1-4
alias section of iwoevent.cfg file, 2-32
aliases, 2-33
application adapter integration
BPEL Process Manager and, 4-1
InterConnect and, 5-1
application adapters, 1-1 to 1-8
configuring, 3-2
deploying, 1-6, 3-1 to 3-2, 4-1
integrating, 5-1
integrating with BPEL Process Manager, 6-1
troubleshooting, 8-1 to 8-11
Application Explorer, 1-6, 2-1, 2-18, 4-7, 4-41
application systems and, 2-5
channels and, 2-18, 2-31
debugging and, 8-2
event ports and, 2-18
J2CA configuration and, 3-2
OracleAS Adapter J2CA and, 8-4
schemas and, 2-10
security and, 9-1, 9-7
testing and, 8-2
troubleshooting, 8-2
WSDL files and, 4-41
Application parameter, 4-43
application systems, 2-18
Application Explorer and, 2-5
supported, 2-5
Available list, 9-4 to 9-6

B
batch processes, 1-4
batch.log file, 2-31
BPEL Designer, 4-1, 6-1
BPEL Process Manager
adapter integration and, 4-1
integrating with adapters, 6-1
OracleAS Adapter for J.D. Edwards OneWorld
and, 4-1
Branch Plant Item Information, 8-3
BSE (OracleAS Adapter Business Services Engine), 1-6
configuring, 2-2
collection access to, 9-7
troubleshooting, 8-4 to 8-11
BSE control service URL, 9-9
BSE deployment, 1-6
BSE repositories, 9-9 to 9-11
BSE URL field, 2-3
business events, 1-1
business functions, 1-8
creating schemas for, 2-10
business services
creating, 2-15 to 2-16
deploying, 9-1
testing, 2-16
Business Services node, 9-2 to 9-5, 9-7

C
CCI calls, 3-1
channel configuration parameters, 4-43
Application, 4-43
Host, 4-42
Is Keep Alive, 4-42
Is Length Prefix, 4-42
Is XML, 4-42
JDE Environment, 4-43
Port Number, 4-42
Schema style, 4-43
Server IP address, 4-43
Server Port, 4-43
User id, 4-43
User password, 4-43
channels, 2-18
  creating, 2-18, 4-41 to 4-44
  debugging and, 4-44
  deleting, 2-31
  editing, 2-31
  ports and, 2-18
  starting, 4-44
  stopping, 4-44
  testing and, 4-44
Channels node, 4-41
channels. See also listeners
common section of iwoevent.cfg file, 2-32
Configuration node under Business Services, 9-2 to 9-5, 9-7
configuration parameters, 3-4
  IWayConfig, 3-4
  IWayHome, 3-4
  IWayRepoPassword, 3-4
  IWayRepoURL, 3-4
  IWayRepoUser, 3-4
  Loglevel, 3-4
configuration requirements
  integration and, 5-1
configurations
  OracleAS Adapter Business Services Engine (BSE), 2-2 to 2-4
  overwriting, 3-4
Configurations node, 2-2 to 2-3
configuring adapters, 3-2
connecting to J.D. Edwards OneWorld, 2-5 to 2-9, 8-2
connecting to OracleAS Adapter J2CA, 8-4
Connection dialog box, 2-9
connection information, 2-32, 8-2
connection parameters, 8-2
  Application, 2-9
  Hostname, 2-3
  JDE environment, 2-9
  Server IP address, 2-9
  Server Port, 2-9
  User id, 2-8, 2-9
  User password, 2-8, 4-43
connections
  creating, 2-5 to 2-9
  troubleshooting, 8-2 to 8-4
Connector
  deploying to Oracle Application Server, 3-1
  connector factories, 3-2
  connector factory objects, 3-2
    multiple, 3-4
  control methods, 9-10
  creating channels, 4-41 to 4-44
creating repository projects, 2-2 to 2-4
creating schemas, 2-10

D
data dictionaries, 8-4
Data Export Control table, 1-4
data queues, 1-5
Data Source Name (DSN), 2-33
database connections
  opening, 3-4
database tables, 1-4
databases
  connecting to, 3-4
  Oracle, 3-4
date ranges, 8-4
deploying adapters, 1-6, 3-1 to 3-2, 4-1
deployments
  BSE, 1-6
Description field, 2-6, 2-16, 9-3 to 9-6, 9-8
design time, 4-41, 9-9
  configuring, 4-7, 4-41, 5-1
DNS Lookup option, 9-8
DNS name, 9-8
document types
  request, 1-4
  response, 1-4
  XML, 1-4
Domain Name System (DNS), 9-8
domain names, 9-8
Domain option, 9-8
DSN (Data Source Name), 2-33

E
Eclipse. See JDeveloper
editing targets, 2-9
EIS (enterprise information systems), 1-4
EJB (Enterprise Java Beans), 3-1
Enterprise Connector for J2EE Connector Architecture (J2CA), 1-6, 2-2 to 2-5
target systems and, 8-5
event adapters, 2-18, 2-31
event data
  receiving, 4-40
  event integration, 4-40
event listeners, 2-31 to 2-32
event messages, 4-41
events, 1-1, 2-18
  configuring, 2-18
Execution Denied list, 9-6
Execution Granted list, 9-6
Existing Service Names list, 2-16
Export WSDL dialog box, 4-8
external listeners, 1-4
F
fault code elements, 8-5
fault string elements, 8-5
flat files, 1-4
functional modeling, 5-1

G
generating schemas, 2-10
GenJava program, 2-7
GenJava repository, 2-7
Grant Access check box, 9-8
Group (of Computers) option, 9-8
Group node, 9-4
groups, 9-3

H
Host parameter, 4-42
Hostname parameter, 2-3

I
inbound J2CA services, 4-41
inbound processing, 1-5, A-1
Inspection date, 8-4
installation directories, 3-4
installation requirements
integration and, 5-1
instances of policy types, 9-1
integration requirements, 5-1
integration scenarios, 5-1
InterConnect
adapter integration and, 5-1
internal listeners, 1-4
interoperability framework, 1-5
inbound processing, 1-5
outbound processing, 1-5
invalid address numbers, 8-3
invalid records, 8-3
IP (Mask)/Domain field, 9-8
IP addresses, 2-8, 2-33, 9-8
IP and Domain Restriction policy type, 9-7
Is Keep Alive parameter, 4-42
Is Length Prefix parameter, 4-42
Is XML parameter, 4-42
Item Branch records, 8-3
IWayConfig parameter, 3-4
IWayHome parameter, 3-4
IWayRepoPassword parameter, 3-4
IWayRepoURL parameter, 3-4
IWayRepoUser parameter, 3-4
IWOEvent listener exit, 2-31
iwoevent.cfg file, 2-32 to 2-33
iwoevent.log file, 2-31

J
J2CA (Enterprise Connector for J2EE Connector Architecture), 1-6, 2-2 to 2-5
OracleAS Adapter and, 3-1
J2CA configuration
Application Explorer and, 3-2
J2CA repositories, 9-12
J2CA resource adapters, 1-1, 3-1
J2CA services, 4-8 to 4-41
JAR files, 4-44, 8-2
Java files, 2-7
Java program clients, 3-1
J.D. Edwards OneWorld
connecting to, 2-5 to 2-9, 8-2
J.D. Edwards OneWorld Event Listener, 2-31 to 2-34
J.D. Edwards OneWorld ThinNet API, 1-5, 1-8, 2-10
JDE Environment parameter, 2-8, 4-43
JDE OneWorld dialog box, 2-6
jde TransactionName, 2-33
JDEdwards node, 4-26
JDeveloper, 4-7, 6-1

L
Last Service date, 8-4
License and Method dialog box, 2-16
License field, 2-16
licenses, 2-16
list of nodes, 8-2
listener configuration files, 2-32, 2-34
listener exits, 2-31
listener types, 1-4
listeners, 1-4, 2-31 to 2-33
listeners. See also channels
log files, 8-1
log levels
overwriting, 3-4
Loglevel parameter, 3-4
logon parameters, 8-2
Logon tab, 2-7

M
managed connector factories, 3-2
managed connector factory objects, 3-2
multiple, 3-4
ManagedConnectionFactory parameter, 3-4
Master Business Functions (MBF), 1-3 to 1-6, 2-5, A-1
executing, 2-10
MBF (Master Business Functions), 1-3 to 1-6, 2-5, A-1
executing, 2-10
Mediator Inbound Process, 5-27
Mediator Outbound Process, 5-2
messages, 1-1 to 1-4
event, 4-41
metadata, 9-9
Method Name field, 2-16
methods, 9-1
migrating repositories, 9-9 to 9-11
migrating Web services, 9-11

N
Name field, 2-6, 9-3 to 9-6
New Configuration dialog box, 2-2 to 2-4
New Group dialog box, 9-4
New Policy permissions dialog box, 9-6
New User dialog box, 9-3
Node list, 8-2
nodes, 2-2 to 2-3, 2-9
   Business Services, 9-2 to 9-5
   Channels, 4-41
   Configuration under Business Services, 9-2 to 9-5, 9-7
   Configurations, 2-2 to 2-3
   Group, 9-4
   JDEdwards, 4-41
   Policies, 9-5
   Security, 9-5, 9-7
   Users, 9-3
   Users and Groups, 9-2 to 9-3

O
OneWorld Event Listener, 2-31 to 2-34
Oracle Application Server
   deployment of Connector to, 3-1
Oracle databases, 3-4
Oracle JDeveloper, 4-7, 6-1
Oracle JDeveloper BPEL Designer. See BPEL Designer, JDeveloper, or Oracle JDeveloper
Oracle repositories
   migrating, 9-9 to 9-11
Oracle’s Unified Method (OUM), x
OracleAS Adapter
   installation directory and, 3-4
   J2CA and, 3-1
OracleAS Adapter Application Explorer. See Application Explorer
OracleAS Adapter Business Services Engine (BSE), 1-6
   configuring, 2-2
   troubleshooting, 8-4 to 8-11
OracleAS Adapter for J.D. Edwards OneWorld
   BPEL Process Manager and, 4-1
   configuring, 2-1
   deploying, 1-1, 4-1
   InterConnect and, 5-1
   troubleshooting, 8-1 to 8-11
OracleAS Adapter J2CA
   Application Explorer and, 8-4
   connecting to, 8-4
OracleAS InterConnect
   adapter integration and, 5-1
   outbound agents, 2-31
   outbound processing, 1-5, 2-31
   outbound transactions, 2-31

P
parameter types
   channel configuration, 4-43
   configuration, 3-4
   connection, 2-9, 8-2

Q
queues, 1-5

R
record identifiers, 2-31
records
   processing, 8-3
repositories
   migrating, 9-9 to 9-11
repositoatory migration parameters, 9-10
repositoatory projects
   creating, 2-2 to 2-4
   Web services and, 2-2
Repository tab, 2-7
repository URLs, 9-11
repository.xml file, 9-12
request documents, 1-4
Request Schema tab, 2-11
request schemas, 2-10
resource adapters, 1-1, 1-7, 3-1 to 3-2
Resource Execution policy type, 9-1
response documents, 1-4
Response Schema tab, 2-11
response schemas, 2-10
runtime, 9-9
   configuring, 5-1

S
scenarios
   integration, 5-1
Schema style parameter, 4-43
schemas, 1-4
   creating, 2-10
security, 9-1 to 9-8
Application Explorer and, 9-1, 9-7
configuring, 9-2
Security node, 9-2 to 9-5, 9-7
security policies, 9-1, 9-4
Security Policy option, 9-7
Selected list, 9-4 to 9-6
Server IP address parameter, 2-8, 4-43
server name, 2-8
Server Port parameter, 2-8, 4-43
Service Contract date, 8-4
service names, 2-16
Service Provider list, 2-3 to 2-4
Service-Oriented Architecture (SOA), 4-1
services, 1-1
creating, 2-15 to 2-16
testing, 2-16
servlets, 3-1
Single (Computer) option, 9-8
SOA (Service-Oriented Architecture), 4-1
SOAP agents, 8-4 to 8-5
SOAP envelopes, 9-10
SOAP faults, 8-4 to 8-5
soap operation name dialog box, 9-9
SOAP requests, 8-5 to 8-8, 9-1, 9-6
creating, 9-9
errors and, 8-4 to 8-6, 8-8
SOAP responses, 8-4 to 8-8
software reuse, 5-1
storing metadata, 9-9
Synchronization Type list, 4-42
synchronous access, 1-5

T

target systems, 8-5
Target Type list, 2-6
targets
connecting to, 2-6 to 2-9
defining, 2-6 to 2-8
deleting, 2-10
disconnecting from, 2-9
editing, 2-9
ThinNet API, 1-5, 1-8, 2-10
trace information, 8-1
trace settings, 2-33
trans section of iwoevent.cfg file, 2-32
transaction processing, 1-1, A-1
transaction tables, 1-5
troubleshooting, 8-1 to 8-11
Application Explorer, 8-2
BSE, 8-4 to 8-11
Web services, 8-4 to 8-11
Type list, 9-6, 9-8

U

updating adapter configuration, 3-2
User id parameter, 2-8, 2-9, 4-43
User password parameter, 2-8, 4-43
UserName parameter, 8-2

users
associating, 9-2
Users and Groups node, 9-2 to 9-3
Users node, 9-3

W

Web service names, 2-16
Web services, 1-1
creating, 2-15 to 2-16
deploying, 8-4, 9-1
integrating, 4-1
migrating, 9-11
repository projects and, 2-2
testing, 2-16
troubleshooting, 8-4 to 8-11
Web Services Description Language (WSDL), 4-41
generating, 4-8, 4-44
Web services policy-based security, 9-1 to 9-8
WSDL (Web Services Description Language), 4-41
generating, 4-8, 4-44
WSDL documents, 4-1
WSDL file location dialog box, 9-9
WSDL files, 4-1, 6-1
Application Explorer and, 4-41
creating, 4-41

X

XDJdeOutboundAgent, 2-31
XML documents, 1-4
XML files, 9-12
XML format, 1-5
XML messages, 1-1 to 1-4
XML schemas, 1-4
creating, 2-10
XMLInterop parameter, 4-43

Z

Z files, 2-31