Oracle® Fusion Middleware Application Adapters
Application Adapter for J.D. Edwards OneWorld User’s Guide for
Oracle WebLogic Server
12c Release 1 (12.1.3.0.0)
E58248-01

October 2014
Provides information on how to integrate with J.D. Edwards
OneWorld systems and develop applications.
Preface ................................................................................................................................................................... ix
  Audience .............................................................................................................................................................. ix
  Documentation Accessibility .................................................................................................................................. ix
  Related Documents ........................................................................................................................................... ix
  Conventions ........................................................................................................................................................ x

1 Introduction
   Adapter Features ......................................................................................................................................................... 1-1
   J.D. Edwards OneWorld Concepts ........................................................................................................................... 1-2
   Integration with J.D. Edwards OneWorld ................................................................................................................... 1-2
       Propagating Internal Listeners Out of J.D. Edwards OneWorld ..................................................................... 1-3
       J.D. Edwards OneWorld Interoperability Framework ...................................................................................... 1-3
   Adapter Architecture ........................................................................................................................................ 1-5
       Oracle Adapter Application Explorer (Application Explorer) ............................................................................. 1-6
       Resource Adapters ............................................................................................................................................. 1-6
       Oracle WebLogic Server Adapter Business Services Engine (BSE) Architecture ........................................... 1-6
       Oracle WebLogic Server Adapter Generic J2CA Architecture ...................................................................... 1-7
       Processing Business Functions ....................................................................................................................... 1-8
   BSE Versus Oracle Adapter J2CA Deployment ........................................................................................................ 1-8
   Sample Projects ...................................................................................................................................................... 1-9
   Quick Start Guide .................................................................................................................................................... 1-10
       Installation ........................................................................................................................................................ 1-10
       Copying Third-Party Library Files ....................................................................................................................... 1-11
       Configuration .................................................................................................................................................... 1-11
       WebLogic Server Deployment and Integration ................................................................................................. 1-12
       Creating Configurations, Targets, and Channels in Application Explorer ....................................................... 1-12
       Working With Service Components in the SOA Suite ..................................................................................... 1-13
       Working With Oracle Service Bus .................................................................................................................... 1-13
       Other Features ................................................................................................................................................... 1-13

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 ...................................................... 2-4
Establishing a Connection (Target) for J.D. Edwards OneWorld ...................... 2-5
Defining a Target to J.D. Edwards OneWorld .................................................. 2-5
Creating an XML Schema .................................................................................. 2-10
Creating a Request and a Response Schema .................................................... 2-10
Generating WSDL (J2CA Configurations Only) .................................................. 2-11
Generating a WSDL for Outbound Interaction ................................................... 2-11
Creating and Testing a Web Service (BSE Configurations Only) ......................... 2-13
Creating a Web Service ..................................................................................... 2-13
Testing a Web Service ....................................................................................... 2-14
Identity Propagation ......................................................................................... 2-14
Configuring an Event Adapter .......................................................................... 2-15
Creating and Editing a Channel........................................................................... 2-15
The J.D. Edwards OneWorld Event Listener ..................................................... 2-28
Configuring the J.D. Edwards OneWorld Event Listener ..................................... 2-28
Runtime Overview ............................................................................................ 2-30
Modifying the JDE.INI File for Outbound and Inbound Processing ...................... 2-30

3 Oracle WebLogic Server Deployment and Integration

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

4 Integration With BPEL Service Components in the Oracle SOA Suite

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

Designing an Inbound BPEL Process for Event Integration (J2CA Configuration) 4-33
Generating WSDL for Event Integration ........................................................... 4-34
Creating an Empty Composite for SOA ............................................................ 4-39
Defining a BPEL Inbound Process .................................................................... 4-40
Deploying the BPEL Inbound Process ............................................................... 4-46
Triggering an Event in J.D. Edwards OneWorld .............................................. 4-47

Designing an Outbound BPEL Process for Service Integration (BSE Configuration) 4-52
Generating a WSDL File for Request and Response Services Using a Web Service 4-52
Creating an Empty Composite for SOA ............................................................ 4-53
# 5 Integration With Mediator Service Components in the Oracle SOA Suite

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring a New Application Server Connection</td>
<td>5-2</td>
</tr>
<tr>
<td>Configuring a Mediator Outbound Process (J2CA Configuration)</td>
<td>5-2</td>
</tr>
<tr>
<td>Creating an Empty Composite for SOA</td>
<td>5-2</td>
</tr>
<tr>
<td>Defining a Mediator Outbound Process</td>
<td>5-3</td>
</tr>
<tr>
<td>Deploying the Mediator Outbound Process</td>
<td>5-10</td>
</tr>
<tr>
<td>Invoking the Input XML Document in the Oracle Enterprise Manager Console</td>
<td>5-11</td>
</tr>
<tr>
<td>Configuring a Mediator Inbound Process (J2CA Configuration)</td>
<td>5-11</td>
</tr>
<tr>
<td>Creating an Empty Composite for SOA</td>
<td>5-11</td>
</tr>
<tr>
<td>Defining a Mediator Inbound Process</td>
<td>5-11</td>
</tr>
<tr>
<td>Configuring a Mediator Outbound Process (BSE Configuration)</td>
<td>5-17</td>
</tr>
<tr>
<td>Creating an Empty Composite for SOA</td>
<td>5-18</td>
</tr>
<tr>
<td>Defining a Mediator Outbound Process</td>
<td>5-18</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>6-1</td>
</tr>
<tr>
<td>Deployment of Adapter</td>
<td>6-2</td>
</tr>
<tr>
<td>Configuring a New Application Server Connection</td>
<td>6-2</td>
</tr>
<tr>
<td>Designing an Outbound BPM Process Using Transformations for Service Integration (J2CA Configuration)</td>
<td>6-2</td>
</tr>
<tr>
<td>Creating an Empty Composite for BPM</td>
<td>6-2</td>
</tr>
<tr>
<td>Defining a BPM Outbound Process</td>
<td>6-3</td>
</tr>
<tr>
<td>Adjusting for Known Deployment Issues With 12c</td>
<td>6-25</td>
</tr>
<tr>
<td>Deploying the BPM Outbound Process</td>
<td>6-25</td>
</tr>
<tr>
<td>Invoking the Input XML Document in the Oracle Enterprise Manager Console</td>
<td>6-26</td>
</tr>
<tr>
<td>Designing an Inbound BPM Process Using Transformations for Event Integration (J2CA Configuration)</td>
<td>6-27</td>
</tr>
<tr>
<td>Creating an Empty Composite for BPM</td>
<td>6-27</td>
</tr>
<tr>
<td>Defining a BPM Inbound Process</td>
<td>6-28</td>
</tr>
<tr>
<td>Designing an Outbound BPM Process Using Transformations for Service Integration (BSE Configuration)</td>
<td>6-42</td>
</tr>
<tr>
<td>Creating an Empty Composite for BPM</td>
<td>6-42</td>
</tr>
<tr>
<td>Defining a BPM Outbound Process</td>
<td>6-42</td>
</tr>
</tbody>
</table>

# 7 Configuring Outbound and Inbound Processing Using Oracle Service Bus

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of Application Adapter Integration with Oracle Service Bus</td>
<td>7-1</td>
</tr>
<tr>
<td>Configuring an Outbound Process Using sbconsole (J2CA Configuration)</td>
<td>7-1</td>
</tr>
<tr>
<td>Starting Oracle Service Bus and Creating Project Folders</td>
<td>7-2</td>
</tr>
<tr>
<td>Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus</td>
<td>7-6</td>
</tr>
<tr>
<td>Publishing a WSDL From Application Explorer to Oracle Service Bus</td>
<td>7-6</td>
</tr>
<tr>
<td>Configuring a WSDL-based Business Service</td>
<td>7-7</td>
</tr>
<tr>
<td>Configuring a File Type Business Service</td>
<td>7-9</td>
</tr>
<tr>
<td>Configuring a Pipeline With Proxy Service</td>
<td>7-13</td>
</tr>
<tr>
<td>Configuring an Inbound Process Using sbconsole (J2CA Configuration)</td>
<td>7-22</td>
</tr>
</tbody>
</table>
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
    Running GenJava ........................................................................................................... A-2
Triggering J.D. Edwards OneWorld Events........................................................................ A-4
    Starting the Outbound Scheduler Subsystem Process (R00460) .................................. A-6
    Verifying the Subsystem Process.................................................................................. A-7
Configuring P4210 (Sales Order) to Trigger an Event...................................................... A-9
    Verifying the Configuration Steps............................................................................... A-11

Glossary

Index
Welcome to Oracle Fusion Middleware Application Adapter for J.D. Edwards OneWorld User’s Guide for Oracle WebLogic Server. This document provides information on how to integrate with J.D. Edwards OneWorld systems and develop applications.

Audience
This document is intended for system administrators and developers who integrate with J.D. Edwards OneWorld systems and develop applications.

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

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

Related Documents
For more information, see the following documents in the Oracle Enterprise Repository 12c: Release 1 (12.1.3.0.0) 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/KNOWLEDGEAREAS1/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><strong>monospace</strong></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>

**Note:** Throughout this document, `<ORACLE_HOME>` refers to the 12c installed home location.

`<ADAPTER_HOME>` refers to the following:

- For SOA:
  
  `<ORACLE_HOME>`\soa\soa\thirdparty\ApplicationAdapters

- For OSB:
  
  `<ORACLE_HOME>`\osb\3rdparty\ApplicationAdapters

This chapter contains the following sections:

- Section 1.1, "Adapter Features"
- Section 1.2, "J.D. Edwards OneWorld Concepts"
- Section 1.3, "Integration with J.D. Edwards OneWorld"
- Section 1.4, "Adapter Architecture"
- Section 1.5, "BSE Versus Oracle Adapter J2CA Deployment"
- Section 1.6, "Sample Projects"
- Section 1.7, "Quick Start Guide"

### 1.1 Adapter Features

Oracle Application Adapter for 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 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.

### 1.3 Integration with J.D. Edwards OneWorld

This section contains the following topics:

- **Section 1.3.1**, "Propagating Internal Listeners Out of J.D. Edwards OneWorld"
- **Section 1.3.2**, "J.D. Edwards OneWorld Interoperability Framework"

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.

### 1.3.1 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.

### 1.3.2 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*

![Diagram of 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 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.

### 1.4 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)

This section contains the following topics:

- Section 1.4.1, "Oracle Adapter Application Explorer (Application Explorer)"
- Section 1.4.2, "Resource Adapters"
1.4.3 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.
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.

### 1.4.4 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.

---

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

1.4.5 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, is not required because a listener is defined through a HTTP, TCP, or File 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.

1.5 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.

### 1.6 Sample Projects

Sample projects for the Oracle Application Adapter for J.D. Edwards OneWorld that demonstrate outbound and inbound integration scenarios using Oracle BPEL, Mediator, BPM, and OSB tools are packaged with the Application Adapters installation. The following table lists the locations of the sample projects:

<table>
<thead>
<tr>
<th>Sample Project</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outbound BPEL Process (J2CA)</td>
<td><code>&lt;ADAPTER_HOME&gt;\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPEL\J2CA\Outbound_Project</code></td>
</tr>
<tr>
<td>Inbound BPEL Process (J2CA)</td>
<td><code>&lt;ADAPTER_HOME&gt;\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPEL\J2CA\Inbound_Project</code></td>
</tr>
<tr>
<td>Outbound BPEL Process (BSE)</td>
<td><code>&lt;ADAPTER_HOME&gt;\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPEL\BSE\Outbound_Project</code></td>
</tr>
<tr>
<td>Outbound Mediator Process (J2CA)</td>
<td><code>&lt;ADAPTER_HOME&gt;\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\Mediator\J2CA\Outbound_Project</code></td>
</tr>
<tr>
<td>Inbound Mediator Process (J2CA)</td>
<td><code>&lt;ADAPTER_HOME&gt;\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\Mediator\J2CA\Inbound_Project</code></td>
</tr>
<tr>
<td>Outbound Mediator Process (BSE)</td>
<td><code>&lt;ADAPTER_HOME&gt;\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\Mediator\BSE\Outbound_Project</code></td>
</tr>
<tr>
<td>Outbound BPM Process (J2CA)</td>
<td><code>&lt;ADAPTER_HOME&gt;\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPM\J2CA\Outbound_Project</code></td>
</tr>
<tr>
<td>Inbound BPM Process (J2CA)</td>
<td><code>&lt;ADAPTER_HOME&gt;\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPM\J2CA\Inbound_Project</code></td>
</tr>
<tr>
<td>Outbound BPM Process (BSE)</td>
<td><code>&lt;ADAPTER_HOME&gt;\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPM\BSE\Outbound_Project</code></td>
</tr>
</tbody>
</table>
1.7 Quick Start Guide

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

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

1.7.1 Installation

1. Download the Oracle Fusion Middleware Application Adapters installation file for the corresponding platform being used and execute the file.

- **Windows**: iwora12c_application-adapters_win.exe
- **Linux**: iwora12c_application-adapters_linux.bin
- **Solaris**: iwora12c_application-adapters_solaris.bin
- **HPUX**: iwora12c_application-adapters_hpux.bin
1. AIX: `iwora12c_application-adapters_aix.bin`

2. The Oracle Fusion Middleware Application Adapters must be installed in one of the following directories:

   - **For Oracle SOA Suite:**
     
     `<ORACLE_HOME>\soa\soa\thirdparty\ApplicationAdapters`
   
   - **For OSB:**
     
     `<ORACLE_HOME>\osb\3rdparty\ApplicationAdapters`

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

### 1.7.2 Copying Third-Party Library Files

Once the adapter installation is completed, copy the required third-party library files for J.D. Edwards OneWorld to the following directories:

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

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

### 1.7.3 Configuration

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

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

   **Table 1–1**
   
<table>
<thead>
<tr>
<th>Config-Property-Name</th>
<th>Config-Property-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>IWayHome</code></td>
<td><code>&lt;ADAPTER_HOME&gt;</code></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>For example:</td>
</tr>
<tr>
<td></td>
<td>- For SOA:</td>
</tr>
<tr>
<td></td>
<td>C:\12C_soa\soa\thirdparty\ApplicationAdapters</td>
</tr>
<tr>
<td></td>
<td>- For OSB:</td>
</tr>
<tr>
<td></td>
<td>C:\12c_OSB\osb\3rdparty\ApplicationAdapters</td>
</tr>
</tbody>
</table>

   | `IWayConfig`         | The name of the configuration. For example: |
   |                     | `jca_sample`           |

2. Open `ibse.war\WEB-INF\web.xml` and add the following values under the specified param-name parameters, as shown in Table 1–2.
1. Start the WebLogic server and open the WebLogic console.

2. Deploy the adapter components (ibse.war, iwafjca.war, and iwafjca.rar files) and start the deployed adapter components.

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

### Table 1–2

<table>
<thead>
<tr>
<th>Param-Name</th>
<th>Param-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ibseroot</td>
<td>$\text{&lt;ADAPTER_HOME&gt;}/ibse.war</td>
</tr>
<tr>
<td></td>
<td>For example:</td>
</tr>
<tr>
<td></td>
<td>■ For SOA:</td>
</tr>
<tr>
<td></td>
<td>$\text{C:{}\text{l2C_soa\soa\soa\thirdparty\ApplicationAdapters\ibse.war}$</td>
</tr>
<tr>
<td></td>
<td>■ For OSB:</td>
</tr>
<tr>
<td></td>
<td>$\text{C:{}\text{l2c_OSB\osb\3rdparty\ApplicationAdapters\ibse.war}$</td>
</tr>
<tr>
<td>IWay.home</td>
<td>$\text{&lt;ADAPTER_HOME&gt;}$</td>
</tr>
<tr>
<td></td>
<td>For example:</td>
</tr>
<tr>
<td></td>
<td>■ For SOA:</td>
</tr>
<tr>
<td></td>
<td>$\text{C:{}\text{l2C_soa\soa\soa\thirdparty\ApplicationAdapters}$</td>
</tr>
<tr>
<td></td>
<td>■ For OSB:</td>
</tr>
<tr>
<td></td>
<td>$\text{C:{}\text{l2c_OSB\osb\3rdparty\ApplicationAdapters}$</td>
</tr>
<tr>
<td>Iway.config</td>
<td>The name of the configuration. For example:</td>
</tr>
<tr>
<td></td>
<td>IBSE</td>
</tr>
</tbody>
</table>

**Note:** These steps are provided only when configuring a File repository. For more information about configuring a database repository and general configuration information, see Chapter 2, "Configuring Oracle Application Adapter for J.D. Edwards OneWorld" and Chapter 3, "Oracle WebLogic Server Deployment and Integration".

### 1.7.4 WebLogic Server Deployment and Integration

1. Start the WebLogic server and open the WebLogic console.

2. Deploy the adapter components (ibse.war, iwafjca.war, and iwafjca.rar files) and start the deployed adapter components.

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

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

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

- Starting Application Explorer: Section 2.1, "Starting Application Explorer"
- Creating a BSE Configuration: Section 2.3.1, "Creating a Configuration for BSE"
- Creating a J2CA Configuration: Section 2.3.2, "Creating a Configuration for J2CA"
1.7.6 Working With Service Components in the SOA Suite

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

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

- Chapter 4, "Integration With BPEL Service Components in the Oracle SOA Suite"
- Chapter 5, "Integration With Mediator Service Components in the Oracle SOA Suite"
- Chapter 6, "Integration With BPM Service Components in the Oracle SOA Suite"

1.7.7 Working With Oracle Service Bus

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

For more information on working with OSB Console, see Chapter 7, "Configuring Outbound and Inbound Processing Using Oracle Service Bus".

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

1.7.8 Other Features

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

- Configuring the Exception Filter: Section 9.1, "Exception Filter"
- Configuring Credential Mapping:
  - Section 9.2, "Credential Mapping for Oracle SOA Suite (BPEL, Mediator, or BPM)"
  - Section 9.3, "Credential Mapping for Oracle Service Bus (OSB)"
Configuring Oracle Application Adapter for J.D. Edwards OneWorld

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 sections:

- 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. Open the command prompt.

3. Navigate to the following directory:
   
   `<ADAPTER_HOME>\user_projects\domains\base_domain\bin`

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

This command sets the class path and other environment variables for Application Explorer in the Oracle WebLogic Server environment. In addition, it allows Application Explorer to access the Oracle WebLogic Server APIs to publish WSDL files to the Oracle Service Bus (OSB) Console.
5. Do not close the command prompt.
6. Navigate to the following directory:
   
   `<ADAPTER_HOME>\tools\iwae\bin`

7. Execute `ae.bat` (Windows) or `iwae.sh` (UNIX/Linux) to start Application Explorer.

   Application Explorer starts. You are ready to define new targets to your 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 2.7.4 "Configuring the Oracle 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.

This section contains the following topics:

- Section 2.3.1, "Creating a Configuration for BSE"
- Section 2.3.2, "Creating a Configuration for J2CA"
- Section 2.3.3, "Connecting to a BSE or J2CA Configuration"

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.

This section contains the following topic:

- Section 2.3.1.1, "Defining a New Configuration for BSE"

#### 2.3.1.1 Defining a New Configuration for BSE

To define a new configuration for BSE:

1. Right-click **Configurations** and select **New**.
Creating a Repository Configuration

Configuring Oracle Application Adapter for J.D. Edwards OneWorld

2.3.2 Creating a Configuration for J2CA

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

To define a new configuration for J2CA:

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

   The New Configuration dialog is displayed, as shown in **Figure 2–1**.

   **Figure 2–1 New Configuration Dialog**

   ![New Configuration Dialog](image)

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

      The New Configuration dialog is displayed, as shown in **Figure 2–2**.

   **Figure 2–2 New Configuration Dialog**

   ![New Configuration Dialog](image)

   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 2–3**.

   **Figure 2–3 Configurations Node**

   ![Configurations Node](image)
The New Configuration dialog is displayed.

2. Enter a name for the new configuration (for example, myConfig) and click OK, as shown in Figure 2–4.

![New Configuration Dialog](image)

**Figure 2–4 New Configuration Dialog**

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 2–5.

![Configurations Node](image)

**Figure 2–5 Configurations Node**

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

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

The `configuration_name` is the name of the configuration you created (for example, SampleConfig).

### 2.3.3 Connecting to a BSE or J2CA Configuration

To connect to a new configuration:

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

2. Select **Connect**.

Nodes appear for Adapters, Events, and Business Services (also known as Web services). The Business Services node is only available for BSE configurations. If you are connected to a J2CA configuration, then the Business Services node is not shown. As shown in Figure 2–6, the following is an example of a BSE configuration named SampleConfig:
Use the **Adapters** node to create inbound interaction with J.D. Edwards OneWorld. For example, you use the J.D. Edwards OneWorld node in the Adapters node to configure a service that updates J.D. Edwards OneWorld.

Use the **Events** node (available for J2CA configurations only) to configure listeners that listen for events in J.D. Edwards OneWorld.

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

You can now define new targets to 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.

This section contains the following topic:

- Section 2.4.1, "Defining a Target to J.D. Edwards OneWorld"

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

This section contains the following topics:

- Section 2.4.1.1, "Connecting to a Defined J.D. Edwards OneWorld Target"
- Section 2.4.1.2, "Disconnecting from J.D. Edwards OneWorld"
- Section 2.4.1.3, "Editing a Target"
- Section 2.4.1.4, "Deleting a Target"

To connect to an application system for the first time, you must define a new target. When you define a target, you must restart the Oracle WebLogic Server to update the repository for runtime purposes.
To define a target:

1. In the left pane, expand the **Adapters** node.
   
   The applications systems supported by Application Explorer appear as nodes based on the adapters that are installed.

2. Right-click the **JDEdwards** node and select **Add Target**.
   
   The Add Target dialog is displayed, as shown in Figure 2–7.

   **Figure 2–7 Add Target Dialog**

   ![Add Target Dialog]

   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 2–8.

---

**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.*
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".

- From the **Schema style** list, select **ELEMENT_STYLE** or **ATTRIBUTE_STYLE**.
- 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 2-9**.
4. Click OK.

The new target, JDEConnection, appears under the JDEdwards node.

### 2.4.1.1 Connecting to a Defined J.D. Edwards OneWorld Target

To connect to a target:

1. Expand the **Service Adapters** node.
2. Expand the **JDEdwards** 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 2–10.
2.4.1.2 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.

2.4.1.3 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.

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

2.4.1.4 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.

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

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.

This section contains the following topic:

- Section 2.5.1, "Creating a Request and a 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 2–11.

5. Click the parameters tab to view the parameter information Figure 2–12.

6. Click Request Schema to view the request schema information, as shown in Figure 2–13.
7. Click **Response Schema** to view the response schema information, as shown in Figure 2–14.

---

### 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.

This section contains the following topic:

- **Section 2.6.1, "Generating a WSDL for Outbound Interaction"**

---

### 2.6.1 Generating a 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, CALLBSFN, and then Addressbook. Select GetEffectiveAddress.

3. Right-click GetEffectiveAddress.
   The following menu is displayed, as shown in Figure 2–15.

   **Figure 2–15  Create Outbound JCA Service (Request/Response) Option**
   
   ![Export Schema(s) Create Outbound JCA Service(Request/Response) Apply Filter]

4. Select Create Outbound JCA Service (Request/Response).
   The Export WSDL dialog is displayed, as shown in Figure 2–16.

   **Figure 2–16  Export WSDL Dialog**
   
   ![Export WSDL Name: jca\Tools\Adapter\J.D.\OneWorld\J2CA_Outbound_Invoke.wsdl Browse Location: Host: Port: User: Password: OK Cancel]

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.
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.

This section contains the following topics:
- Section 2.7.1, "Creating a Web Service"
- Section 2.7.2, "Testing a Web Service"
- Section 2.7.3, "Identity Propagation"

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.

2.7.1 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 CALLBSFN and then Addressbook.
3. Right-click GetEffectiveAddress and select Create Web Service.

The Create Web Service dialog is displayed, as shown in Figure 2–17.

![Create Web Service Dialog](image)

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, leave the default method 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.

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

The Save dialog is displayed.

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

8. Click **Save**.

### 2.7.2 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.

### 2.7.3 Identity Propagation

If you test or execute a Web service using a third party XML editor, for example XMLSpy, then the user name and password values that you specify in the SOAP header must be valid and are used to connect to 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:ibsinfo>
</SOAP-ENV:Header>
```
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.

This section contains the following topics:

- Section 2.8.1, "Creating and Editing a Channel"
- Section 2.8.2, "The J.D. Edwards OneWorld Event Listener"
- Section 2.8.3, "Configuring the J.D. Edwards OneWorld Event Listener"

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

A channel represents configured connections to particular instances of back-end systems. For more information, see “Creating and Editing a Channel” on page 2-15.

2.8.1 Creating and Editing a Channel

The following section describes how to create a channel for your event and contains the following topics:

- Section 2.8.1.1, "Creating an HTTP Channel"
- Section 2.8.1.2, "Creating a TCP Channel"
- Section 2.8.1.3, "Creating a File Channel"
- Section 2.8.1.4, "Editing a Channel"
- Section 2.8.1.5, "Deleting 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.
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.

### 2.8.1.1 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 2–18.
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 2–19.
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 <strong>Https</strong> check box. This option is currently not supported.</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 2–20.
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</td>
</tr>
<tr>
<td></td>
<td>information about this parameter, see your J.D. Edwards OneWorld</td>
</tr>
<tr>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>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>Specify *ALL.</td>
</tr>
<tr>
<td>Schema Location</td>
<td>The location of the XML schema (.xsd file) that was generated from the</td>
</tr>
<tr>
<td></td>
<td>event output. For example:</td>
</tr>
<tr>
<td></td>
<td>$ADAPTER_HOME$/config/configuration_name/schemas/JDEdwards/target_name/JDE-MAIN.xsd</td>
</tr>
</tbody>
</table>

For more information, see Section 4.5.1, "Generating WSDL for Event Integration" on page 4-34.

<table>
<thead>
<tr>
<th>Schema Style</th>
<th>Choose from one of the following options:</th>
</tr>
</thead>
<tbody>
<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**.

### 2.8.1.2 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 2–21.

*Figure 2–21  Add Channel Dialog*

Provide the following information:
a. Enter a name for the channel, for example, JDE_Channel2.
b. Enter a brief description.
c. From the Protocol list, select TCP Listener.

4. Click Next.

The Tcp Listener dialog is displayed, as shown in Figure 2–22.

**Figure 2–22 Tcp Listener Dialog Basic Tab**

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 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><strong>Important:</strong> The J.D. Edwards OneWorld channel does not work if the synchronization type is set to REQUEST.</td>
<td></td>
</tr>
<tr>
<td>Is Length Prefix</td>
<td>For J.D. Edwards OneWorld events that send data back that 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.</td>
</tr>
<tr>
<td>Is XML</td>
<td>For J.D. Edwards OneWorld events that send data back in XML format. No preparser is required.</td>
</tr>
<tr>
<td>Is Keep Alive</td>
<td>Maintains continuous communication between the event transaction and the channel.</td>
</tr>
</tbody>
</table>

6. Click the PreParser tab, as shown in Figure 2–23.
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>Specify <em>ALL</em>.</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: <code>&lt;ADAPTER_HOME&gt;\config\configuration_name\schemas\JDEdwards\target_name\jde-schema.xsd</code></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**.

### 2.8.1.3 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 2–24.

**Figure 2–24  Add Channel Dialog**

![Add Channel Dialog](image)

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 2–25.

Figure 2–25 File Listener Dialog

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 2–26.
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 2–27.
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</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><strong>Sequential</strong> indicates single processing of requests.</td>
</tr>
<tr>
<td></td>
<td><strong>Threaded</strong> 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 2–28.
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>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;\config\configuration_name\schemas\JDEdwards\target_name\jde-schema.xsd</td>
</tr>
</tbody>
</table>

For more information, see Section 4.5.1, "Generating WSDL for Event Integration" on page 4-34.

Schema Style Choose from one of the following options:
- ELEMENT_STYLE (default)
- ATTRIBUTE_STYLE

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**.

### 2.8.1.4 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**.

### 2.8.1.5 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 event stub, (IWOEvent.dll), is located in the \etc\jde directory. For example:
  \<ADAPTER_HOME>\etc\jde\iwoevent.dll

The file extension varies depending on your operating system:

- For Windows, the event stub is iwoevent.dll.
- For Sun Solaris, the event stub is libiwoevent.so.
- For HP-UX, the event stub is libiwoevent.sl.
- For AS/400, the event stub is iwaysav.sav.
- For IBM AIX, the event stub 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 Sever, 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:

   - 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:

```plaintext
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:

`\JDEEdwards\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:

```
[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.sl</td>
<td>libxmltransactions.lo</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

This section contains the following topics:

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

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

<ADAPTER_HOME>\iwarjca.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).
3.3.1 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>
  <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>
  </resourceadapter>
</connector>
```
<config-property>
<config-property-name>IWayRepoUser</config-property-name>
<config-property-type>java.lang.String</config-property-type>
<config-property-value></config-property-value>
</config-property>
<config-property>
<config-property-name>IWayRepoPassword</config-property-name>
<config-property-type>java.lang.String</config-property-type>
<config-property-value></config-property-value>
</config-property>
<config-property>
<config-property-name>LogLevel</config-property-name>
<config-property-type>java.lang.String</config-property-type>
<config-property-value>DEBUG</config-property-value>
</config-property>
<authentication-mechanism>
<authentication-mechanism-type>BasicPassword</authentication-mechanism-type>
<credential-interface>javax.resource.spi.security.PasswordCredential</credential-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>

### 3.3.2 Creating Multiple Managed Connector Factory Objects

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

<ADAPTER_HOME>/iwafjca.rar\META-INF
For example, the first jca_configuration in Application Explorer is represented in the weblogic-ra.xml file as follows:

```xml
<?xml version="1.0"?>
<weblogic-connector xmlns="http://www.bea.com/ns/weblogic/90">
  <enable-access-outside-app>true</enable-access-outside-app>
  <enable-global-access-to-classes>true</enable-global-access-to-classes>
  <outbound-resource-adapter>
    <default-connection-properties>
      <pool-params>
        <initial-capacity>0</initial-capacity>
      </pool-params>
      <transaction-support>LocalTransaction</transaction-support>
    </default-connection-properties>
    <connection-definition-group>
      <connection-factory-interface>javax.resource.cci.ConnectionFactory</connection-factory-interface>
      <connection-instance>
        <jndi-name>eis/OracleJCAAdapter/DefaultConnection</jndi-name>
      </connection-instance>
    </connection-definition-group>
  </outbound-resource-adapter>
</weblogic-connector>
```

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

```xml
<?xml version="1.0"?>
<weblogic-connector xmlns="http://www.bea.com/ns/weblogic/90">
  <enable-access-outside-app>true</enable-access-outside-app>
  <enable-global-access-to-classes>true</enable-global-access-to-classes>
  <outbound-resource-adapter>
    <default-connection-properties>
      <pool-params>
        <initial-capacity>0</initial-capacity>
      </pool-params>
      <transaction-support>LocalTransaction</transaction-support>
    </default-connection-properties>
    <connection-definition-group>
      <connection-factory-interface>javax.resource.cci.ConnectionFactory</connection-factory-interface>
      <connection-instance>
        <jndi-name>eis/OracleJCAAdapter/DefaultConnection</jndi-name>
      </connection-instance>
      <connection-instance>
        <jndi-name>eis/OracleJCAAdapter/DefaultConnection1</jndi-name>
        <connection-properties>
          <property>
            <name>IWayHome</name>
            <value>C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters</value>
          </property>
          <property>
            <name>IWayConfig</name>
            <value>jca_sample2</value>
          </property>
        </connection-properties>
      </connection-instance>
    </connection-definition-group>
  </outbound-resource-adapter>
</weblogic-connector>
```
<property>
  <name>IWayRepoURL</name>
  <value></value>
</property>

<property>
  <name>IWayRepoUser</name>
  <value></value>
</property>

<property>
  <name>IWayRepoPassword</name>
  <value></value>
</property>

<property>
  <name>LogLevel</name>
  <value>Debug</value>
</property>

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.

3.3.3 Modifying WSDL Files for Additional Connection Factory Values

Application Explorer generates the J2CA properties file using the default connection factory name eis/OracleJCAAdapter/DefaultConnection. If you created additional connection factories, 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:

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

The connection factory value must be changed to the following:

```
eis/OracleJCAAdapter/DefaultConnection1
```

For example:

---

**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.
<jca:address location='eis/OracleJCAAdapter/DefaultConnection1'
    ConnectionSpec='com.ibi.afjca.cci.IWAFConnectionSpec'
    csAdapterName='JDEdwards' csConfig='isdsrv2_conn2'
    UIConnectionName='Connection1'/>

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

Oracle Application Adapter for 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 (J2CA Configuration)"
- **Section 4.5, "Designing an Inbound BPEL Process for Event Integration (J2CA Configuration)"
- **Section 4.6, "Designing an Outbound BPEL Process for Service Integration (BSE Configuration)"

### 4.1 Overview

To integrate with Oracle 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 Window and select Application Server Navigator, as shown in Figure 4–1.

Figure 4–1 Application Server Navigator

The Application Server tab is displayed, as shown in Figure 4–2.
3. Right-click Application Servers and 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.
5. Specify a new name for the Application Server connection and click Next. The Authentication page is displayed, as shown in Figure 4–5.

Figure 4–5 Authentication Page
6. Specify a valid user name (for example, weblogic) and a password (for example, welcome1) for your new connection.

7. Click Next.

The Configuration page is displayed, as shown in Figure 4–6.

Figure 4–6 Configuration Page

8. Specify the Oracle WebLogic host name (for example, localhost), which is the system IP where the process must deploy and Oracle WebLogic domain (for example, base_domain).

9. Click Next.

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

11. Make sure that the test status is successful.

12. Click **Next**.

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

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

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

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

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

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

The following tools are required to complete your adapter design-time configuration:
- Oracle Adapter Application Explorer (Application Explorer)
- Oracle JDeveloper BPEL Designer (JDeveloper)

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

This section contains the following topics:
- Section 4.4.1, "Generating WSDL for Request/Response Service"
- Section 4.4.2, "Creating an Empty Composite for SOA"
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 2-5.

2. Expand Services, CALLBSFN, and Addressbook.

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

3. Right-click GetEffectiveAddress, and then select Create Outbound JCA Service (Request/Response), as shown in Figure 4–9.

   The Export WSDL dialog is displayed, as shown in Figure 4–10.
4. 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 an Empty Composite for SOA

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

1. Create a new SOA application.

2. Enter a name for the new SOA Application and click Next, as shown in Figure 4–11.
The Name your project page is displayed, as shown in Figure 4–12.

Figure 4–12 Name Your Project Page
3. Enter a project name and click **Next**.

   The Configure SOA settings page is displayed, as shown in **Figure 4–13**.

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

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

### 4.4.3 Defining a BPEL Outbound Process

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

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

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

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

1. Drag and drop the **Third Party Adapter** component from the Service Adapters pane to the External References pane, as shown in **Figure 4–14**.
The Create Third Party Adapter Service dialog is displayed, as shown in Figure 4–15.

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

   The WSDL Chooser dialog is displayed, as shown in **Figure 4–16**.

   **Figure 4–16  WSDL Chooser Dialog**

4. Browse and select an outbound WSDL file from the following directory:
   
   `<ADAPTER_HOME>/wsdls`

5. Click **OK**.

   The Localize Files dialog is displayed, as shown in **Figure 4–17**.
6. Click **OK**.

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

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

7. Click the **Find JCA file** icon, which is located to the right of the JCA File field.
The Transformation Chooser dialog is displayed, as shown in Figure 4–19.

**Figure 4–19 Transformation Chooser Dialog**

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

<ADAPTER_HOME>\wsdl\s

9. **Click OK.**

The Copy File message is displayed, as shown in Figure 4–20.

**Figure 4–20 Copy File Message**

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

<ADAPTER_HOME>\wsdl\s

9. **Click OK.**

The Copy File message is displayed, as shown in Figure 4–20.

**Figure 4–20 Copy File Message**

10. **Click Yes.**

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

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

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

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

4.4.3.2 Configuring an Outbound BPEL Process Component

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

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

The Create BPEL Process dialog is displayed, as shown in Figure 4–22.
In the Name field, enter a name to identify the new outbound BPEL process component or leave it to the default value.

By default, the BPEL 2.0 Specification option is selected.

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

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

The Type Chooser dialog is displayed, as shown in **Figure 4–23**.
5. Expand Project Schema Files, J2CA_Outbound_invoke_request.xsd, and select jdeRequest.

6. Click OK.

   You are returned to the Create BPEL Process dialog.

7. Click the Browse icon, which is located to the right of the Output field to select the associated XML response schema file.

   The Type Chooser dialog is displayed, as shown in Figure 4–24.
8. Expand **Project Schema Files**, **J2CA_Outbound_invoke_response.xsd**, and select **jdeResponse**.

9. Click **OK**.
   
   You are returned to the Create BPEL Process dialog.

10. Click **OK**.

11. Create a connection between the outbound BPEL process component and the third party adapter service component, as shown in **Figure 4–25**.
12. Double-click the outbound BPEL process component in the Components pane.

13. Drag and drop the **Invoke** activity component under BPEL Constructs - Web Service, to the Components pane and place it between the **receiveInput** activity component and the **replyOutput** activity component, as shown in Figure 4–26.

14. Create a connection between the new Invoke activity component Service and the third party adapter service component (Service), as shown in Figure 4–27.
The Edit Invoke dialog is displayed.

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

   The Create Variable dialog is displayed.

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

   You are returned to the Edit Invoke dialog, as shown in Figure 4–28.
17. Select the **Output** tab and click the **Plus sign** icon, which is located to the right of the **Output** field to configure a new output variable.

The Create Variable dialog is displayed.

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

You are returned to the Edit Invoke dialog, as shown in **Figure 4–29**.
19. Click **Apply** and then **OK**.

20. Drag and drop the **Assign** activity under BPEL Constructs - Basic Activities component, to the Components pane and place it between the Receive activity component (receiveInput) and the Invoke activity component (Invoke1), as shown in Figure 4–30.
21. Double-click the new Assign activity component (Assign1).
   The Edit Assign dialog is displayed.
22. In the left pane, under Variables, expand InputVariable, and then select payload.
23. In the right pane, under Variables, expand Invoke1_GetEffectiveAddress_ InputVariable, and then select input_GetEffectiveAddress.
24. Drag and map the payload variable to the input_GetEffectiveAddress variable.
   The mapped variables are populated in the highlighted area as shown in Figure 4–31.

Figure 4–31  Edit Assign Dialog

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

28. In the left pane, under Variables, expand `Invoke1_GetEffectiveAddress_OutputVariable`, and then select `output_GetEffectiveAddress`.

29. In the right pane, under Variables, expand `outputVariable` and select `payload`.

30. Drag and map the `output_GetEffectiveAddress` variable to the `payload` variable.

The mapped variables are populated in the highlighted area as shown in Figure 4–33.

31. Click Apply and then OK.

You are returned to the Activity component pane, as shown in Figure 4–34.
32. Click the **Save All** icon in the menu bar to save the new outbound BPEL process component that was configured.

You are now ready to deploy the BPEL outbound process.

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

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

1. Double-click **J2CA_Outbound** (created BPEL process) of the created process, as shown in Figure 4–35.

2. Click the **Source** tab below the opened process, as shown in Figure 4–36.
3. Change the `productVersion` property value from `12.1.3.0.0` to `11`, as shown in Figure 4–37.
4. Save the changes and proceed to deploy the project.

4.4.4 Deploying the BPEL Outbound Process

Perform the following steps to deploy the BPEL outbound process.

1. Right-click the project name in the left pane, select **Deploy**, and then click **J2CA_Outbound**, as shown in Figure 4–38.

The Deployment Action page is displayed, as shown in Figure 4–39.
2. Ensure that **Deploy to Application Server** is selected.
3. Click **Next**.

The Deploy Configuration page is displayed, as shown in **Figure 4–40**.

**Figure 4–40  Deploy Configurations Page**

4. Leave the default values selected and click **Next**.
The Select Server page is displayed, as shown in Figure 4–41.

Figure 4–41 Select Server Page

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

The SOA Servers page is displayed, as shown in Figure 4–42.

Figure 4–42 SOA Servers Page
6. Select a target SOA server and click Next.

The Summary page is displayed, as shown in Figure 4–43.

**Figure 4–43 Summary Page**

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

The process is deployed successfully, as shown in Figure 4–44.

**Figure 4–44 Successful Deployment Message**

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

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

1. Logon to the Oracle Enterprise Manager console.
2. Expand SOA, select soa-infra (soa_server1), and then click Default.
3. Select an available project (for example, J2CA_Outbound) and click Test as shown in Figure 4–45.
4. Click the Request tab.

5. Select XML View from the list, as shown in Figure 4–46.

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

The output response is received in the Oracle Enterprise Manager console, as shown in Figure 4–47.
4.4.6 Testing Outbound BPEL and Mediator Processes

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

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

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

This section 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.

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

<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPEL\J2CA\Inbound_Project

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

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

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

This section contains the following topics:

- **Section 4.5.1, "Generating WSDL for Event Integration"**
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.

This section contains the following topics:

- Section 4.5.1.1, "Creating a Channel in Application Explorer"
- Section 4.5.1.2, "Generating WSDL for Event Notification (Command Prompt Only)"

4.5.1.1 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 4–48.
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 4–49.

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 4–50**.

**Figure 4–50 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 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>
</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.

---

### 4.5.1.2 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 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 based on your adapter installation:

   `<ADAPTER_HOME>\config\configuration_name\log\iwaf_jca1500.log`

   Alternatively, you can create a port using the File protocol under the Events node in Application Explorer, which disposés 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:

---

<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;ADAPTER_HOME&gt;\config\configuration_name\schemas\JDEdwards\target_name\jde-schema.xsd</code> For more information, see Section 4.5.1, &quot;Generating WSDL for Event Integration&quot; on page 4-34.</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>

---

Parameter Description
Designing an Inbound BPEL Process for Event Integration (J2CA Configuration)

a. Search for Schemas-jdedwards-com and replace it with iwaysoftware.

```xml
<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:

```xml
<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:

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

c. Paste it before the following line:

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

7. Copy the XML schema (.xsd file) to the following directory based on your adapter installation:

```
<ADAPTER_HOME>\config\configuration_name\schemas\JDEdwards\target_name\n```

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>\config\configuration_name\schemas\JDEdwards\target_name\jde-schema.xsd
```

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

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

9. Execute `setDomainEnv.cmd` (Windows) or `../setDomainEnv.sh` (UNIX/Linux).

10. In the same command prompt, navigate to the following directory:

```
<ADAPTER_HOME>\tools\iwae\bin
```

11. Execute the `obadapter.bat` file to set the environment.

12. Based on your adapter installation, navigate to the following directory where the XML schema (.xsd file) is copied:

```
<ADAPTER_HOME>\config\configuration_name\schemas\JDEdwards\target_name
```

13. Enter the following command to generate a WSDL:

```
java -Diway.oem=oracle12c
com.iwaysoftware.af.container.tools.wsdll.IWayWSILBrowser adapterhome adapter
target channel schemaPrefix wsdlFileName
```

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

For SOA:

<ORACLE_HOME>/soa/soa/thirdparty/ApplicationAdapters

For OSB:

<ORACLE_HOME>/osb/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.

Execute the following command to generate the inbound WSDL.

java -Diway.oem=oracle12c
com.iwaysoftware.af.container.tools.wsdl.IWayWSILBrowser
C:\12c_SOA\soa\soa\thirdparty\ApplicationAdapters\JDEdwards JDEconnection JDEchannel jde-schema J2CA_Inbound_receive.wsdl

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

Running Inbound WSDL generation tool...
-> Generating WSDL...
-> Generating files for OEM oracle12c
-> 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.

14. Stop the channel in Application Explorer.

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 an Empty Composite for SOA

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

1. Create a new SOA application.

2. Enter a name for the new SOA Application and click Next.
   The Name your project page is displayed.

3. Enter a project name and click Next.
   The Configure SOA settings page is displayed.

4. From the Composite Template list, select Empty Composite and click Finish.
   For more information, see Section 4.4.2, "Creating an Empty Composite for SOA" on page 4-9.
4.5.3 Defining a BPEL Inbound Process

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

- Section 4.5.3.1, "Creating a Third Party Adapter Service Component"
- Section 4.5.3.2, "Creating an Inbound BPEL Process Component"
- Section 4.5.3.3, "Adjusting for Known Deployment Issues With 12c"

4.5.3.1 Creating a Third Party Adapter Service Component

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

1. Drag and drop the Third Party Adapter component from the Service Adapters pane to the Exposed Services pane, as shown in Figure 4–51.

**Figure 4–51 Third Party Adapter Component**

The Create Third Party Adapter Service dialog is displayed, as shown in Figure 4–52.
2. Ensure that Service is selected from the Type list (default).

3. Click the Find existing WSDLs icon, which is located to the right of the WSDL URL field.
   
The WSDL Chooser dialog is displayed, as shown in Figure 4–53.

4. Browse and select an inbound WSDL file from the following directory:
5. Click OK.
   The Localize Files dialog is displayed, as shown in Figure 4–54.

**Figure 4–54  Localize Files Dialog**

6. Click OK.
   The inbound WSDL file and associated receive/request XML schema file (.xsd) are imported to the project folder that has been created.
   You are returned to the Create Third Party Adapter Service dialog.

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

8. Browse and select the JCA properties file from the following directory:
   `<ADAPTER_HOME>\wsdls`

9. Click OK.
   A Copy File message is displayed, as shown in Figure 4–55.
10. Click Yes.

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

You are returned to the Create Third Party Adapter Service dialog, as shown in Figure 4–56.

11. Click OK.

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

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

4.5.3.2 Creating an Inbound BPEL Process Component

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

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

The Create BPEL Process dialog is displayed, as shown in Figure 4–57.
2. In the Name field, enter a name to identify the new inbound BPEL process component or leave to default.
   
   By default, the BPEL 2.0 Specification option is selected.

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

4. Uncheck the **Expose as SOAP service** check box.

5. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.
   
   The WSDL Chooser dialog is displayed.

6. Select an inbound WSDL file from the following directory:
   
   `<ADAPTER_HOME>`\wsdls

7. Click **OK**.
   
   The Localize Files dialog is displayed, as shown in **Figure 4–58**.
8. Uncheck the Rename duplicate files option.
9. Click OK.
You are returned to the Create BPEL Process dialog.
10. Click OK.

11. Create a connection between the third party adapter service component and the inbound BPEL process component, as shown in Figure 4-59.
12. Double-click J2CA_Outbound in the left pane.

Figure 4–60  Save All Icon

![Diagram showing BPEL process configuration]

13. Click the Save All icon in the menu bar to save the new inbound BPEL process component that was configured, as shown in Figure 4–60.

You are now ready to deploy the BPEL inbound process.

4.5.3.3 Adjusting for Known Deployment Issues With 12c

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

4.5.4 Deploying the BPEL Inbound Process

Perform the following steps to deploy the BPEL inbound process.

1. Right-click the project name in the left pane, select Deploy, and click J2CA_Inbound.

The Deployment Action page is displayed.

2. Ensure that Deploy to Application Server is selected.

3. Click Next.

The Deploy Configuration page is displayed.

4. Leave the default values selected and click Next.

The Select Server page is displayed.

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

The SOA Servers page is displayed.

6. Select a target SOA server and click Next.

The Summary page is displayed.
7. Review and verify all the available deployment information for your project and click Finish.

The process is deployed successfully.

For more information, see Section 4.4.4, "Deploying the BPEL Outbound Process" on page 4-28.

Once event messages are triggered through J.D. Edwards OneWorld, successful instances are received in the Oracle Enterprise Manager console, as shown in Figure 4–61.

**Figure 4–61 Received Instances**

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 4–62.
3. Right-click Sales Order Detail (P4210).

4. Select Prompt for, and then Values, as shown in Figure 4–63.

The Processing Options dialog is displayed, as shown in Figure 4–64.
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 4–65.

6. Click the Add icon (third icon from left).
7. Enter the values as shown in Figure 4–66.

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

**Figure 4–66 Values**

![Image of values](image)

8. Enter a value for **Quantity Ordered** and **Item Number**, as shown in Figure 4–67.

**Figure 4–67 Sample Values**

![Image of sample values](image)

9. Click the first field in the second row and allow a few seconds for processing, as shown in Figure 4–68.
Figure 4–68 Sample Values

<table>
<thead>
<tr>
<th>Order Number</th>
<th>358</th>
<th>50</th>
<th>10206</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sold To</td>
<td>1442</td>
<td></td>
<td>Capital System</td>
</tr>
<tr>
<td>Ship To</td>
<td>1442</td>
<td></td>
<td>Capital System</td>
</tr>
<tr>
<td>Currency</td>
<td>JPY</td>
<td></td>
<td>Exchange Rate</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Format1</th>
<th>Format2</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Quantity Ordered</code></td>
<td><code>Unit Number</code></td>
</tr>
<tr>
<td><code>1 EA</code></td>
<td><code>210</code></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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. Click SOA, select `soa-infra (soa_server1)`, default, and then click `J2CA_Inbound`.

3. Click Flow Instances.
   Instances will be received, as shown in Figure 4–69.

Figure 4–69 Flow Instances
4.6 Designing an Outbound BPEL Process for Service Integration (BSE Configuration)

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

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

<ADAPTER_HOME>/etc/sample/JDEdwards_Samples.zip/JDEdwards_Samples/BPEL/BSE/Outbound_Project

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

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

This section includes the following topics:

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

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

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

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

1. Start Application Explorer and connect to a defined J.D. Edwards target (BSE configuration).
   
   For more information on defining a target and connecting to J.D. Edwards, see Section 2.4.1, "Defining a Target to J.D. Edwards OneWorld".

2. Expand the J.D. Edwards target to which you are connected.

3. Expand Services, CALLBSFN, and Addressbook.

4. Right-click GetEffectiveAddress, and then select Create Web Service from the menu, as shown in Figure 4–70.

Figure 4–70 Create Web Service Option
The Create Web Service dialog is displayed.

5. Enter a service name, and click Next.

6. Click OK on the next dialog that is displayed.
   
   Application Explorer switches the view to the Business Services node, and the new Web service is displayed in the left pane.

7. Right-click the new Web service and select Save WSDL from the menu.

8. Save the WSDL in the wsdls folder and click Save.
   
   You can now create an empty composite for SOA, which is the first step that is required to define a BPEL outbound process in JDeveloper.

### 4.6.2 Creating an Empty Composite for SOA

To create an empty composite for SOA:

1. Create a new SOA application.

2. Enter a name for the SOA Application and click Next.
   
   The Name your project page is displayed.

3. Enter a project name and click Next.
   
   The Configure SOA settings page is displayed.

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

### 4.6.3 Defining a BPEL Outbound Process

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

This section includes the following topics:

- Section 4.6.3.1, "Creating a Partner Link"
- Section 4.6.3.2, "Creating BPEL Activities and Mappings With the Created Partner Link"

To define a BPEL outbound process:

1. Drag and drop the BPEL Process component from the Service Components pane to the Components pane, as shown in Figure 4–71.
2. In the Name field, enter a name to identify the new outbound BPEL process component or leave it to the default value.

By default, the BPEL 2.0 Specification option is selected.

3. From the Template drop-down list, select **Base on a WSDL**.

4. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field, as shown in **Figure 4–72**.
The WSDL Chooser dialog is displayed.

5. Navigate to the location where the WSDL is exported from Application Explorer, select the WSDL, and click **OK**, as shown in **Figure 4–73**.
The Localize Files window is displayed.

6. In the displayed Localize Files window, click OK. This imports the WSDL file to the project folder, as shown in Figure 4–74.

The Create BPEL Process window is displayed.

7. In the BPEL Process pane, click OK, as shown in Figure 4–75.
The BPEL Process component is created and displayed, as shown in Figure 4–76.

4.6.3.1 Creating a Partner Link

This section describes how to create a partner link.

To create a partner link:
1. Double-click the outbound BPEL process component in the Components pane.

2. Right-click on the Partner Links pane and select Create Partner Link, as shown in Figure 4–77.

Figure 4–77 Create Partner Link

3. In the displayed Create Partner Link window, provide an appropriate name and click on the SOA Resource Browser tool, as shown in Figure 4–78.

Figure 4–78 SOA Resource Browser Tool

4. In the WSDL Chooser dialog that is displayed, navigate to the location where the WSDL is exported from Application Explorer, select the WSDL, and click OK, as shown in Figure 4–79.
5. In the displayed Localize Files window, uncheck the **Rename duplicate files** check box and click **OK**, as shown in **Figure 4–80**.

**Figure 4–80 Localize Files Window**

6. Click **Yes** in the displayed Partner Link Type window, as shown in **Figure 4–81**.
7. In the displayed Create Partner Link window, expand the **Partner Role** drop-down list and select the available partner role.

8. Click **Apply**, and then **OK**, as shown in *Figure 4–82*.

---

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

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

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

1. Drag and drop the **Invoke** activity component from BPEL Constructs to the Components pane. Place it between the **receiveInput** activity component and the **replyOutput** activity component.

2. Create a connection between the new **Invoke** activity component (Invoke1) and the **Partner Link** component (Partner link1), as shown in *Figure 4–83*. 
3. In the displayed Edit Invoke window, click the Plus (+) icon, located to the right of the Input field, to configure a new input variable.

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

5. Click the Plus (+) icon, which is located to the right of the Output field, to configure a new output variable, as shown in Figure 4–84.

Figure 4–83  Partner Link Component

Figure 4–84  Edit Invoke Window
6. Accept the default values that are provided for the new output variable and click OK.

7. Click Apply and then OK, as shown in Figure 4–85.

Figure 4–85 Edit Invoke Window

![Edit Invoke Window](image)

8. Drag and drop the Assign activity component from BPEL Constructs to the Components pane. Place it between the Receive activity component (receiveInput) and the Invoke activity component (Invoke1), as shown in Figure 4–86.
9. Double-click the new **Assign** activity component (Assign1), as shown in Figure 4–87.

**Figure 4–87 Assign Activity Component**

10. In the left pane, under Variables, expand **InputVariable**, and then select parameters.

11. In the right pane, under Variables, expand **Invoke1_GetEffectiveAddress_InputVariable**, and then select parameters.

12. Drag and map the **InputVariable** parameters to the **Invoke1_GetEffectiveAddress_InputVariable** parameters, as shown in Figure 4–88.
13. Click **Apply** and then **OK**.

14. Drag and drop the **Assign** activity component to the Components pane and place it between the **Invoke** activity (Invoke1) and the **Reply** activity (replyOutput), as shown in Figure 4–89.

**Figure 4–89 Assign Activity Component**

15. Double-click the new **Assign** activity component (Assign2), as shown in Figure 4–90.
16. In the left pane, under Variables, expand `Invoke1_GetEffectiveAddress_OutputVariable`, and then select parameters.

17. In the right pane, under Variables, expand `outputVariable`, and then select parameters.

18. Drag and map the `Invoke1_GetEffectiveAddress_OutputVariable` parameters to the `outputVariable` parameters, as shown in Figure 4-91.

19. Click Apply and then OK.

You are returned to the component pane, as shown in Figure 4-92.
20. Click the **Save All** icon in the menu bar to save the new outbound BPEL process component that was configured.

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

Once deployed you can invoke the input XML, as defined in Section 4.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console" on page 4-31.
This chapter describes integration with Mediator service components in the Oracle SOA Suite. It contains the following sections:

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

The scenarios shown in this chapter require the following prerequisites.

Prerequisites
The following are installation and configuration requirements:

- Oracle Application Adapter for 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" on page 4-2.

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

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

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

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

This section contains the following topics:

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

**Prerequisites**

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

### 5.2.1 Creating an Empty Composite for SOA

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

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

For more information, see Section 4.4.2, "Creating an Empty Composite for SOA" on page 4-9.
5.2.2 Defining a Mediator Outbound Process

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

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

5.2.2.1 Configuring a Third Party Adapter Service Component

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

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

6. Click OK.
7. Click OK.
   
   The outbound WSDL file and associated request and response XML schema files (.xsd) are imported to the project folder that has been created.
8. Click the Find JCA file icon, which is located to the right of the JCA File field.
9. Browse and select the JCA properties file from the following directory:
   
   `<ADAPTER_HOME>\wsdl\`

10. Click OK.
    
    A Copy File confirmation message is displayed.
11. Click Yes.
    
    A copy of the JCA properties file is made in the project folder.
12. Click OK.

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

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

For more information, see Section 4.5.3.1, "Creating a Third Party Adapter Service Component" on page 4-40.

5.2.2.2 Configuring an Outbound Mediator Process Component

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

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

The Create Mediator dialog is displayed, as shown in Figure 5–2.
2. In the Name field, enter a name to identify the new outbound Mediator process component or leave it to the default value.

3. From the Template drop-down list, select **Synchronous Interface**.

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

   The Type Chooser dialog is displayed, as shown in Figure 5–3.
5. Expand Project WSDL Files, J2CA_Outbound_invoke.wsdl, Imported Schemas, J2CA_Outbound_invoke_request.xsd, and select jdeRequest.

6. Click OK.

You are returned to the Create Mediator dialog.

7. Click the Browse icon, which is located to the right of the Output field to select the associated XML response schema file.

The Type Chooser dialog is displayed, as shown in Figure 5–4.

9. Click OK.

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

**Figure 5–5 Create Mediator Dialog**
10. Click OK.

11. Create a connection between the outbound Mediator process component and the third party adapter service component, as shown in Figure 5–6.

**Figure 5–6 Created Connection**

You are now ready to configure the routing rules.

### 5.2.2.3 Configuring the Routing Rules

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

1. Double-click the outbound Mediator process component in the Components pane. The Routing Rules dialog is displayed, as shown in Figure 5–7.

**Figure 5–7 Routing Rules Dialog**
2. In the <<Filter Expression>> area, click the icon to the right of the Transform Using field.

   The Request Transformation Map dialog is displayed, as shown in Figure 5–8.

*Figure 5–8 Request Transformation Map Dialog*

3. Click the Add (+) icon.

   The Create Transformation Map page is displayed.

4. Make sure the Type is selected as XSLT and click OK.

5. Click OK.

6. Map the ns0:jdeRequest source element to the ns0:jdeRequest target element.

   The Auto Map Preferences dialog is displayed.

7. Retain the default values and click OK.

8. Return to the Routing Rules dialog, as shown in Figure 5–9.

*Figure 5–9 Routing Rules Dialog*

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

   The Reply Transformation Map dialog is displayed.

10. Click the Add (+) icon.

    The create Transformation Page is displayed.

11. Make sure the type is selected as XSLT and click OK.
A mapping page is displayed.

12. Click OK.

13. Map the `ns0:jdeResponse` source element to the `ns0:jdeResponse` target element.
   The Auto Map Preferences dialog is displayed.

14. Retain the default values and click OK.
   The mapping is completed, as shown in Figure 5–10.

**Figure 5–10 Completed Mapping**

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

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

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

### 5.2.3 Deploying the Mediator Outbound Process

Perform the following steps to deploy the Mediator outbound process.

1. Right-click the project name in the left pane, select Deploy, and then click J2CA_Outbound.
   The Deployment Action page is displayed.

2. Ensure that Deploy to Application Server is selected.

3. Click Next.
   The Deploy Configuration page is displayed.

4. Leave the default values selected and click Next.
   The Select Server page is displayed.

5. Select an available application server that was configured and click Next.
   The SOA Servers page is displayed.

6. Select a target SOA server and click Next.
   The Summary page is displayed.

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

   For more information, see Section 4.4.4, "Deploying the BPEL Outbound Process" on page 4-28.
5.2.4 Invoking the Input XML Document in the Oracle Enterprise Manager Console

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

5.3 Configuring a Mediator Inbound Process (J2CA Configuration)

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

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

<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\Mediator\J2CA\Inbound_Project

This section contains the following topics:

- Section 5.3.1, "Creating an Empty Composite for SOA"
- Section 5.3.2, "Defining a Mediator Inbound Process"

Prerequisites

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

5.3.1 Creating an Empty Composite for SOA

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

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

5.3.2 Defining a Mediator Inbound Process

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

- Section 5.3.2.1, "Configuring a Third Party Adapter Service Component"
- Section 5.2.2.2, "Configuring an Outbound Mediator Process Component"
- Section 5.2.2.3, "Configuring the Routing Rules"
- Section 5.3.2.4, "Adjusting for Known Deployment Issues With 12c"

5.3.2.1 Configuring a Third Party Adapter Service Component

Perform the following steps to create a third party adapter service component:
1. Drag and drop the **Third Party Adapter** component from the Service Adapters pane to the Exposed Services pane. The Create Third Party Adapter Service dialog is displayed.

2. Enter a name for the third party adapter service.

3. Ensure that **Service** is selected from the Type drop-down list (default).

4. Click the **Find existing WSDLS** icon, which is located to the right of the WSDL URL field. The WSDL Chooser dialog is displayed.

5. Browse and select an inbound WSDL file from the following directory: 
   &lt;ADAPTER_HOME&gt;\wsdls

6. Click **OK**. The Localize Files dialog is displayed.

7. Click **OK**. The inbound WSDL file and associated receive/request schema file (.xsd) are imported to the project folder that has been created.
   You are returned to the Create Third Party Adapter Service dialog.

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

9. Browse and select the JCA properties file from the following directory: 
   &lt;ADAPTER_HOME&gt;\wsdls

10. Click **OK**. The Copy File Confirmation message is displayed.

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

12. Click **OK**. The third party adapter service component is created in the Exposed Services pane.
    You are now ready to configure an inbound Mediator process component.
    For more information, see **Section 6.5.3.1, "Creating a Third Party Adapter Service Component"** on page 6-43.

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

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

1. Drag and drop the **Mediator Process** component from the Service Components pane to the Components pane. The Create Mediator dialog is displayed, as shown in **Figure 5–11**.
In the Name field, enter a name to identify the new inbound Mediator process component.

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

4. Click the **OK**.

   The new Mediator process component is added to the Components pane.

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

   The File Adapter Configuration Wizard is displayed.

6. Type a name for the new File adapter and click **Next**.

   The Adapter Interface page is displayed.

7. Ensure that the **Define from operation and schema (specified later)** option is selected.

8. Click **Next**.

   The Operation page is displayed.

9. Click **Next**.

10. Select **Write File** from the list of Operation Type options and specify an Operation Name (for example, Write).

11. Click **Next**.

   The File Configuration page is displayed.

12. Specify a location on your file system where the output file is written.

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

14. Click **Next**.
The Messages page is displayed, as shown in Figure 5–12.

**Figure 5–12  Messages Page**

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

The Type Chooser dialog is displayed, as shown in Figure 5–13.
16. Expand Project WSDL Files, J2CA_Inbound_receive.wsdl, Imported Schemas, J2CA_Inbound_receive_request.xsd, and select jdeResponse.

17. Click OK.

You are returned to the Messages page.

18. Click Next.

The Finish page is displayed.

19. Click Finish.

20. Create a connection between the inbound Mediator process component and the third party adapter service component.

21. Create a connection between the inbound Mediator process component and the File adapter component, as shown in Figure 5–14.
You are now ready to configure the routing rules.

5.3.2.3 Configuring the Routing Rules
Perform the following steps to configure routing rules for the Mediator inbound process component:

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

2. In the <<Filter Expression>> area, click the icon to the right of the Transform Using field.
   The Request Transformation Map dialog is displayed.

3. Click the Add (+) icon and ensure that the selected Type is XSLT, then click OK.
4. Click OK.

The mapping page is displayed, as shown in Figure 5–16.

**Figure 5–16  Mapping Page**

5. Click OK.

6. Map the `ns0:jdeResponse` source element to the `ns0:jdeResponse` target element.

The Auto Map Preferences dialog is displayed.

7. Retain the default values and click OK.

The mapping is now complete.

8. Click the Save All icon in the menu bar to save the new inbound Mediator process component that was configured.

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

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

You are now ready to deploy the Mediator inbound process. You can follow the same procedure in Section 4.5.4, "Deploying the BPEL Inbound Process" on page 4-46.

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 Section 4.5.5, "Triggering an Event in J.D. Edwards OneWorld" on page 4-47.

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

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

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

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

This section contains the following topics:

- Section 5.4.1, "Creating an Empty Composite for SOA"
- Section 5.4.2, "Defining a Mediator Outbound Process"

**Prerequisites**

Before you design a Mediator outbound process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.6.1,
"Generating a WSDL File for Request and Response Services Using a Web Service" on page 4-52.

5.4.1 Creating an Empty Composite for SOA

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

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

For more information, see Section 4.6.2, "Creating an Empty Composite for SOA" on page 4-53.

5.4.2 Defining a Mediator Outbound Process

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

- Section 5.4.2.1, "Configuring a SOAP Service"
- Section 5.4.2.2, "Creating a Mediator Component"
- Section 5.4.2.3, "Configuring the Routing Rules"

5.4.2.1 Configuring a SOAP Service

Perform the following steps to configure a SOAP Service:

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

The Web Service is created and displayed.

5.4.2.2 Creating a Mediator Component

Perform the following steps to create a Mediator component:

1. Drag and drop the Mediator component from the Components pane in to the Components pane.
2. In the Name field, enter a name to identify the new outbound Mediator process component.
3. From the Template drop-down list, select Synchronous Interface.
4. Click the Browse icon, which is located to the right of the Input field, to select the associated XML request schema file.
5. In the Type Chooser dialog, expand **Project WSDL Files**, select **IBSE_Outbound.wsdl**, and click **GetEffectiveAddress**, as shown in **Figure 5–17**.

**Figure 5–17 Type Chooser Dialog**

6. Click **OK**.

7. Click the **Browse** icon, which is located to the right of the Output field, to select the associated XML response schema file.

8. In The Type Chooser dialog, expand **Project WSDL Files**, select **IBSE_Outbound.wsdl**, and click **GetEffectiveAddressResponse**, as shown in **Figure 5–18**.
9. Click **OK**.

10. Click **OK**.

   The Mediator component is created and displayed.

11. Create a connection between the **Mediator** component and the **SOAP service** component, as shown in Figure 5–19.
5.4.2.3 Configuring the Routing Rules

Perform the following steps to configure the routing rules:

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

6. In the displayed Auto Map Preferences window, retain the default values and click OK.
7. In the Synchronous Reply area, click the icon to the right of the Transform Using field.
8. In the displayed Reply Transformation Map window, click the Add (+) icon and make sure the Type is selected as XSLT in the Create Transformation Map dialog box, and then click OK.
9. Map the ns0:GetEffectiveAddressResponse source element to the ns0:GetEffectiveAddressResponse target element, as shown in Figure 5–21.

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

10. In the displayed Auto Map Preferences window, retain the default values and click OK.

11. Double-click composite.xml in the left pane.

12. Click the Save All icon in the menu bar to save the new outbound Mediator component that was configured, as shown in Figure 5–22.

**Figure 5–22  Save All Icon**

You are now ready to deploy the Mediator IBSE outbound process. You can follow the same procedure found in Section 5.2.3, "Deploying the Mediator Outbound Process" on page 5-10.

Once deployed, you can invoke the input XML, as defined in Section 5.2.4, "Invoking the Input XML Document in the Oracle Enterprise Manager Console" on page 5-11.
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 (J2CA Configuration)"
- **Section 6.5, "Designing an Inbound BPM Process Using Transformations for Event Integration (J2CA Configuration)"
- **Section 6.6, "Designing an Outbound BPM Process Using Transformations for Service Integration (BSE Configuration)"

### 6.1 Overview

To integrate with Oracle BPM, Oracle Application Adapter for 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 2-11.

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" on page 4-2.

6.4 Designing an Outbound BPM Process Using Transformations for Service Integration (J2CA Configuration)
This section describes how to design an outbound BPM process using transformations for service integration.

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

<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPM\J2CA\Outbound_Project

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

■ Oracle Adapter Application Explorer (Application Explorer)
■ Oracle JDeveloper BPM Designer (JDeveloper)

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

This section contains the following topics:

■ Section 6.4.1, "Creating an Empty Composite for BPM"
■ Section 6.4.2, "Defining a BPM Outbound Process"
■ Section 6.4.3, "Adjusting for Known Deployment Issues With 12c"
■ Section 6.4.4, "Deploying the BPM Outbound Process"
■ Section 6.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console"

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

6.4.1 Creating an Empty Composite for BPM
Perform the following steps to create an empty composite for BPM:

1. Create a new BPM application.
2. Enter a name for the new BPM application and click Next.
   The Name your project page is displayed.
3. Enter a project name, in the project features select BPM, and then click Next.
   The Configure SOA settings page is displayed.
4. From the Composite Template list, select Empty Composite and click Finish.
6.4.2 Defining a BPM Outbound Process

This section describes how to define a BPM outbound process, which contains the following topics:

- Section 6.4.2.1, "Configuring a Third Party Adapter Service Component"
- Section 6.4.2.2, "Configuring an Outbound BPM Process Component"
- Section 6.4.2.3, "Creating a File Adapter for the Write Operation"

6.4.2.1 Configuring a Third Party Adapter Service Component

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

1. Double-click the created project to load the components.
2. Drag and drop the **Third Party Adapter** component from the Custom/Thirdparty pane to the External References pane.
   - The Create Third Party Adapter Service dialog is displayed.
3. Enter a name for the third party adapter service.
4. Ensure that **Reference** is selected from the Type list (default).
5. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.
   - The WSDL Chooser dialog is displayed.
6. Browse and select an outbound WSDL file from the following directory: `<ADAPTER_HOME>\wsdls`
7. Click **OK**.
   - The Localize Files dialog is displayed.
8. Click **OK**.
   - The outbound WSDL file and associated request and response XML schema files (.xsd) are imported to the project folder that has been created.
   - You are returned to the Create Third Party Adapter Service dialog.
9. Click the **Find JCA file** icon, which is located to the right of the JCA File field.
   - The Transformation Map dialog is displayed.
10. Browse and select the JCA properties file from the following directory: `<ADAPTER_HOME>\wsdls`
11. Click **OK**.
    - The Copy File message is displayed.
12. Click **Yes**.
    - A copy of the JCA properties file is made in the project folder.
    - You are returned to the Create Third Party Adapter Service dialog.
13. Click **OK**.
    - The third party adapter service component is created and displayed in the External References pane.
    - You are now ready to configure an outbound BPM process component.
For more detailed information, including screen shots, see Section 4.4.3.1, "Configuring a Third Party Adapter Service Component" on page 4-11.

6.4.2.2 Configuring an Outbound BP M Process Component
This section describes how to configure an outbound BPM process component.
Perform the following steps to configure an outbound BPM process component:

1. Drag and drop the BPMN Process component from the Components pane to the Components pane.
   The Create BPMN Process dialog is displayed, as shown in Figure 6–1.

   **Figure 6–1 Create BPMN Process Dialog**

   ![Create BPMN Process Dialog](image)

2. Accept the default option that is selected under the Type area (Asynchronous Service) and click Finish.
   The BPMN process is displayed, as shown in Figure 6–2.
3. Click the **Activity** drop-down menu and select **Service**, as shown in Figure 6–3.

![Figure 6–3 Activity Drop-down Menu](image)

4. Drop the Service icon on the wire between the Start and End event components, as shown in Figure 6–4.

![Figure 6–4 BPMN Process](image)
The Properties - ServiceTask window is displayed.

5. Click the **Implementation** tab.

6. Select **Service Call** from the Message Exchange Type list, as shown in [Figure 6–5](#).

7. Click the Browse icon to the right of the Service field, as shown in [Figure 6–6](#).

The Service dialog is displayed, as shown in [Figure 6–7](#).
8. Select the Third Party Service that has been created and click **OK**.
   You are returned to the Properties - ServiceTask dialog, as shown in Figure 6–8.

9. Click the **Data Associations** hyperlink.
The Data Associations dialog is displayed.

10. Right-click the Data Objects node in the left pane under Process, and select New as shown in Figure 6–9.

Figure 6–9 New Option

The Create Data Object dialog is displayed, as shown in Figure 6–10.

Figure 6–10 Create Data Object Dialog

11. Enter a name in the Name field (for example, Request), click the drop-down button in the Type field, and select Browse from the list, as shown in Figure 6–11.

Figure 6–11 Create Data Object Dialog

The Browse Types dialog is displayed, as shown in Figure 6–12.
12. Select the first component (for example, JdeRequest) and click OK.
   
   You are returned to the Create Data Object dialog.

13. Click OK.
   
   The Data Object (for example, Request) that has been created is displayed under the Data Objects node in the Data Associations dialog.

14. Create another Data Object by right-clicking the Data Objects node in the right pane of the Output tab and selecting New, as shown in Figure 6–13.
The Create Data Object dialog is displayed.

15. Enter a name in the Name field (for example, Response), and then click the drop-down button in the Type field and select Browse from the list. The Browse Types dialog is displayed, as shown in Figure 6–14.

16. Select the second component (for example, jdeResponse) and click OK. You are returned to the Create Data Object dialog.
17. Click OK.

The Data Object (for example, Response) that has been created is displayed under the Process node in the Data Associations dialog.

18. Select the Request Data Object under the Data Objects node in the left pane of the Input tab and drag and connect it to JdeRequest under the Arguments node in the right pane, as shown in Figure 6–15.

**Figure 6–15 Request Data Object**

19. Click on the Output tab and select jdeResponse under the Arguments node in the left pane and drag and connect it to the Response Data Object under the Data Objects node, as shown in Figure 6–16.
20. Click **OK**.

   You are returned to the Properties - ServiceTask dialog.

21. Click **OK**.

   The Service Task is created between the Start and End Event components, as shown in Figure 6–17.

   **Figure 6–17  Service Task**

22. Save the process and double-click the Start event component.

   The Properties - Start dialog is displayed, as shown in Figure 6–18.
Figure 6–18  Properties - Start Dialog

23. Click the Implementation tab, as shown in Figure 6–19.

Figure 6–19  Implementation Tab

24. Click the Plus icon to the right of the Arguments Definition field. The Create Argument dialog is displayed.

25. Enter a name in the Name field (by default, argument1), and then click the drop-down button in the Type field and select Browse from the list, as shown in Figure 6–20.
26. Select the first component (for example, JdeRequest) and click OK. You are returned to the Create Argument dialog.

27. Click OK. You are returned to the Properties - Start dialog.

28. In the Operation Name field, change start (default) to operation as shown in Figure 6–22.

   Note: This change is necessary to work with old BPM payloads.
29. Click the **Data Associations** hyperlink.

   The Data Associations dialog is displayed.

30. Select **arguments1** under the Arguments node in the left pane and drag and connect it to the **Request** Data Object under Data Objects in the right pane.

31. Click **OK** as shown in Figure 6–23.
You are returned to the Properties - Start dialog.

32. Click **OK**.

You are returned to the Process workspace area, as shown in Figure 6–24.

**Figure 6–24  Process Workspace Area**

33. Double-click the created project to load the components.

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

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

**6.4.2.3 Creating a File Adapter for the Write Operation**

This section describes how to create a File adapter for the write operation.
Perform the following steps to create a File adapter for the write operation:

1. Drag and drop the File Adapter component from the Technology Adapters pane to the External References pane, as shown in Figure 6–25.

**Figure 6–25  File Adapter Component**

The Adapter Configuration Wizard is displayed.

2. Provide a Reference Name (for example, FileWrite).

3. Click Next.

The Adapter Interface page is displayed.

4. Ensure that the **Define from operation and schema (specified later)** option is selected.

5. Click Next.

The File Server Connection page is displayed.

6. Click Next.

The Operation page is displayed.

7. Select **Write File** from the list of Operation Type options and specify an Operation Name (for example, Write).

8. Click Next.

The File Configuration page is displayed.

9. Specify a location on your file system where the output file is written.

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

11. Click Next.

The Messages page is displayed.

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

The Type Chooser dialog is displayed, as shown in Figure 6–26.
13. Expand **Project Schema Files** and **J2CA_Outbound_invoke_response.xsd**.
14. Select the available schema (for example, jdeResponse).
15. Click **OK**.
   You are returned to the Messages page.
16. Click **Next**.
   The Finish page is displayed.
17. Click **Finish**.
   The File Adapter service is created in the External References pane, as shown in Figure 6–27.
18. Double-click the BPMN Process component.

The BPMN process is displayed, as shown in Figure 6–28.

19. Click the Activity icon, and select Service.

20. Drop the Service icon on the wire between the Service Task and End event components, as shown in Figure 6–29.
Figure 6–29   Activity Icon

The Properties - ServiceTask1 dialog is displayed.

21. Click the **Implementation** tab.

22. Select **Service Call** from the Type drop-down list in the Message Exchange section, as shown in Figure 6–30.

Figure 6–30   Service Call

23. Click the **Browse** icon to the right of the Service field.

The Service dialog is displayed, as shown in Figure 6–31.
24. Select the service for write operation that has been created (for example, FileWrite) and click OK.

You are returned to the Properties - ServiceTask1 dialog, as shown in Figure 6–32.
25. Click the **Data Associations** hyperlink.

The Data Associations dialog is displayed, as shown in Figure 6–33.

*Figure 6–33  Data Associations Dialog*
26. In the Input tab, click the XSL Transformation icon in the top right pane.

27. Drag and drop the XSL Transformation icon to the \texttt{jdeResponse} node, as shown in Figure 6–34.

\textit{Figure 6–34  CompanyCodeJDEResponse Node}

The Create Transformation dialog is displayed.

28. Select \texttt{Response} in the Sources section and click the right arrow symbol.

The Response object is added to the Selected elements area as shown in Figure 6–35.
Figure 6–35  Response Object

29. Accept the default value selected in the Target drop-down list and the default name in the Create field by clicking OK.

You are returned to the Data Associations dialog window with the XSL transformation created, as shown in Figure 6–36.

Figure 6–36  Data Associations Dialog
30. Click OK.
   You are returned to the Properties - ServiceTask1 dialog.

31. Click OK.
   The Response_body.xsl tab is displayed.

32. Automap the Source and Target elements.
   The Auto Map Preferences dialog is displayed.

33. Accept the default values and click OK.
   The transformation is completed, as shown in Figure 6–37.

**Figure 6–37 Completed Transformation**

![Completed Transformation](image)

34. Save the transformation.

35. Return to the Process workspace area.
   The ServiceTask1 component is created between the ServiceTask component and the End event component.

36. Click the Save All icon in the menu bar to save the new outbound BPM process component that was configured.
   You are now ready to deploy the outbound BPM process.

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

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

### 6.4.4 Deploying the BPM Outbound Process

Perform the following steps to deploy the Mediator outbound process.

1. Right-click the project name in the left pane, select Deploy, and then click J2CA_Outbound.
   The Deployment Action page is displayed.

2. Ensure that Deploy to Application Server is selected.

3. Click Next.
   The Deploy Configuration page is displayed.

4. Leave the default values selected and click Next.
   The Select Server page is displayed.

5. Select an available application server that was configured and click Next.
The SOA Servers page is displayed.

6. Select a target SOA server and click Next.

The Summary page is displayed.

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

For more information, see Section 4.4.4, "Deploying the BPEL Outbound Process" on page 4-28.

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

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

1. Logon to the Oracle Enterprise Manager console.
2. Expand your domain in the left pane followed by the SOA folder.
3. Select an available project (for example, J2CA_Outbound).
4. Click Test.
5. Click the Request tab.

Figure 6–38 Request Tab

6. Provide an appropriate input value in the Value field and click Test Web Service, as shown in Figure 6–38.

A response is received in the Response tab to indicate that invocation was successful in the Oracle Enterprise Manager console, as shown in Figure 6–39.
6.5 Designing an Inbound BPM Process Using Transformations for Event Integration (J2CA Configuration)

This section demonstrates how Oracle Application Adapter for J.D. Edwards OneWorld integrates with J.D. Edwards OneWorld to receive event data.

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

<ADAPTER_HOME>/etc/sample\JDEdwards_Samples.zip\JDEdwards_Samples\BPM\J2CA\Inbound_Project

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

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

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

This section contains the following topics:

- Section 6.5.1, "Creating an Empty Composite for BPM"
- Section 6.5.2, "Defining a BPM Inbound Process"

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

6.5.1 Creating an Empty Composite for BPM

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

7. Navigate to the defined output directory on your file system and open the XML response document that was received.
   The XML response document contains the generated output with values.
6.5.2 Defining a BPM Inbound Process

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

- Section 6.5.2.1, "Configuring a Third Party Adapter Service Component"
- Section 6.5.2.2, "Configuring an Inbound BPM Process Component"
- Section 6.5.2.3, "Creating a File Adapter for the Write Operation"
- Section 6.5.2.4, "Adjusting for Known Deployment Issues With 12c"

6.5.2.1 Configuring a Third Party Adapter Service Component

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

1. Double-click the created project to load the components.
2. Drag and drop the Third Party Adapter component from the Custom/Thirdparty pane to the Exposed References pane, as shown in Figure 6–40.

Figure 6–40 Third Party Adapter Component

The Create Third Party Adapter Service dialog is displayed.

3. Enter a name for the third party adapter service.
4. Ensure that Service is selected from the Type list (default).
5. Click the Find existing WSDLs icon, which is located to the right of the WSDL URL field.
   The WSDL Chooser dialog is displayed.
6. Select File System, and then browse and select an inbound WSDL file from the following directory:
   \<ADAPTER_HOME>\wsdls
7. Click OK.
The Localize Files dialog is displayed.

8. Click **OK**.
   The inbound WSDL file and associated receive_request XML schema file (.xsd) are imported to the project folder that has been created.
   You are returned to the Create Third Party Adapter Service dialog.

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

10. Select **File System**, and then browse and select the JCA properties file from the following directory:
    `<ADAPTER_HOME>/wsdls`

11. Click **OK**.
    The Copy File message is displayed.

12. Click **Yes**.
    A copy of the JCA properties file is made in the project folder.
    You are returned to the Create Third Party Adapter Service dialog.

13. Click **OK**.
    The third party adapter service component (matmas) is created in the Exposed References pane.
    You are now ready to configure an inbound BPM process component.

For more information, see Section 4.4.3.1, “Configuring a Third Party Adapter Service Component” on page 4-11.

### 6.5.2.2 Configuring an Inbound BPM Process Component

This section describes how to configure an inbound BPM process component.

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

1. Drag and drop the **BPMN Process** component from the Components pane to the Components pane.
   The Create BPMN Process dialog is displayed, as shown in Figure 6–41.

3. Click Finish.

The BPMN process is displayed, as shown in Figure 6–42.

4. Right-click UserTask and select Delete from the menu.

5. Double-click the Start event component.

The Properties - Start dialog is displayed.

6. Click the Implementation tab.
7. Select **Message** from the Implementation Type list.

8. Select **Use Interface** from the Message Exchange Type drop-down list.

9. Click the **Browse** icon to the right of the Reference field, as shown in Figure 6–43.

![Figure 6–43   Browse Icon](image)

The Service dialog is displayed, as shown in Figure 6–44.
10. Select the Third Party Service that has been created and click **OK**.

You are returned to the Properties - Start dialog, as shown in Figure 6–45.

**Figure 6–45  Properties - Start Dialog**
11. Click the **Data Associations** icon.

The Data Associations dialog is displayed, as shown in **Figure 6–46**.

**Figure 6–46 Data Associations Dialog**

12. Right-click the **Data Object** node in the right pane and select **New**.

The Create Data Object dialog is displayed.

13. Enter a name in the Name field, and then click the drop-down button in the Type field and select **Browse** from the list, as shown in **Figure 6–47**.

**Figure 6–47 Create Data Object Dialog**

The Browse Types dialog is displayed, as shown in **Figure 6–48**.
14. Select the component and click **OK**.

You are returned to the Create Data Object dialog.

15. Click **OK**.

The Data Object that has been created is displayed under the Data Objects node in the Data Associations dialog, as shown in Figure 6–49.
16. Select and drag the jdeResponse Argument under the Start node in the left pane and drag it to the Data Object in the right pane.

17. Click OK.

   You are returned to the Properties - Start dialog.

18. Click OK.

   You are returned to the Process workspace area.

19. Double-click the created project to load the components.

20. Click the Save All icon in the menu bar to save the new inbound BPM process component that was configured.

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

6.5.2.3 Creating a File Adapter for the Write Operation

This section describes how to create a File adapter for the write operation.

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

1. Drag and drop the File Adapter component from the Technology Adapters pane to the External References pane.

   The Adapter Configuration Wizard is displayed.

2. Type a name for the new File adapter in the Name field and click Next.

   The Adapter Interface page is displayed.

3. Ensure that the Define from operation and schema (specified later) option is selected.

4. Click Next.

   The File Server Connection page is displayed.

5. Click Next.

   The Operation page is displayed, as shown in Figure 6–50.
6. Select **Write File** from the list of Operation Type options and specify an Operation Name (for example, Write).

7. Click **Next**.
   
The File Configuration page is displayed.

8. Specify a location on your file system where the output file is written.

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

10. Click **Next**.
    
The Messages page is displayed.

11. Click **Browse**, which is located to the right of the URL field.
    
The Type Chooser dialog is displayed, as shown in **Figure 6–51**.
12. Expand *Project Schema Files* and *J2CA_Inbound_receive_request.xsd*.
13. Select the available schema.
14. Click **OK**.
   
   You are returned to the Messages page.
15. Click **Next**.
   
   The Finish page is displayed.
16. Click **Finish**.
   
   The File Adapter service is created in the External References pane.
17. Double-click the BPMN Process component.
   
   The BPMN process is displayed.
18. Click the **Activity** icon, and select **Service**.
19. Drop the Service icon on the wire between the Start and End event components, as shown in Figure 6–52.
20. Click the **Implementation** tab.

21. Select **Service Task** from the Implementation Type list.

22. Select **Service Call** from the Message Exchange Type list.

23. Click the **Browse** icon to the right of the Service field.

The Type dialog is displayed, as shown in Figure 6–53.

---

**Figure 6–52 Activity Icon**

![Activity Icon](image)

**Figure 6–53 Type Dialog**

![Type Dialog](image)
24. Select the service for write operation that has been created and click OK.
   You are returned to the Properties - ServiceTask dialog, as shown in Figure 6–54.

   Figure 6–54  Properties - ServiceTask Dialog

25. Click the Data Associations hyperlink.
   The Data Associations dialog is displayed.

26. Right-click the jdeResponse argument on the right pane and select XSL Transformation, as shown in Figure 6–55.
The Create Transformation dialog is displayed.

27. Select the created data object in the Sources area and click the right arrow icon so that the created data object is added to the Selected elements area.

28. Click OK.

You are returned to the Data Associations dialog, as shown in Figure 6–56.

Figure 6–56  Data Associations Dialog
29. Click OK.
   You are returned to the Properties - ServiceTask dialog.

30. Click OK.
   The dataobject1_body.xsl tab is displayed.

31. Automap the Source and Target elements.
   The Auto Map Preferences dialog is displayed.

32. Accept the default values and click OK.
   The transformation is completed, as shown in Figure 6–57.

![Figure 6–57 Completed Transformation](image)

33. Save the transformation.

34. Return to the Process workspace area, as shown in Figure 6–58.

![Figure 6–58 Process Workspace Area](image)

The ServiceTask component is created between the Start event component and the End event component.

35. Click the Save All icon in the menu bar to save the new inbound BPM process component that was configured.

6.5.2.4 Adjusting for Known Deployment Issues With 12c
For more information on how to adjust for known deployment issues with 12c, see Section 4.4.3.3, "Adjusting for Known Deployment Issues With 12c" on page 4-26.
You are now ready to deploy the inbound BPM process. You can follow the same procedure that is described in Section 4.5.4, "Deploying the BPEL Inbound Process" on page 4-46.

For more information on how to trigger events in J.D. Edwards OneWorld, see Section 4.5.5, "Triggering an Event in J.D. Edwards OneWorld" on page 4-47.

6.6 Designing an Outbound BPM Process Using Transformations for Service Integration (BSE Configuration)

This section describes how to configure a BPM outbound process to your J.D. Edwards OneWorld system, using a BPM project in Oracle JDeveloper.

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

<ADAPTER_HOME>/etc/sample/JDEdwards_Samples.zip/JDEdwards_Samples/BPM/BSE/Outbound_Project

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

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

This section contains the following topics:

- Section 6.6.1, "Creating an Empty Composite for BPM"
- Section 6.6.2, "Defining a BPM Outbound Process"

Prerequisites

Before you design a BPM outbound process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.6.1, "Generating a WSDL File for Request and Response Services Using a Web Service" on page 4-52.

6.6.1 Creating an Empty Composite for BPM

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

1. Create a new BPM application.
2. Enter a name for the BPM Application, and click Next.
3. Enter a name in the Project Name field, and click Next.
4. From the Composite Template list, select Empty Composite and click Finish.

For more information, see Section 4.5.2, "Creating an Empty Composite for SOA" on page 4-39.

6.6.2 Defining a BPM Outbound Process

This section describes how to define a BPM outbound process. It contains the following topics:

- Section 6.6.2.1, "Configuring a Web Service Component"
- Section 6.6.2.2, "Configuring a BPM Process Component"
- Section 6.6.2.3, "Creating a File Adapter for the Write Operation"
6.6.2.1 Configuring a Web Service Component

Perform the following steps to configure a Web Service component:

1. Double-click the created project to load the components.
2. Drag and drop the Web Service node from the Technology Adapters pane to the External References pane, as shown in Figure 6–59.

![Figure 6–59 Web Service Node](image)

3. Enter an appropriate name for the Web Service and click on the Find existing WSDLs icon, which is located to the right of the WSDL URL field.
4. In the displayed WSDL Chooser window, navigate to the location where the WSDL is exported from the Application Explorer, and select the WSDL.
5. Click OK.
6. In the Web Service pane, click OK, as shown in Figure 6–60.
7. In the displayed Localize Files window, click **OK**.
   
   This will import the WSDL file to the project folder

### 6.6.2.2 Configuring a BPM Process Component

This section describes how to configure an outbound BPM process component.

Perform the following steps to configure a BPM Component:

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

2. Accept the default option that is selected under the Type area (Asynchronous Service) and click **Finish**, as shown in **Figure 6–61**.
3. Double click on the Start Event component, as shown in Figure 6–62.

4. In the displayed Properties-start window, click the Implementation tab.

5. Click the Plus (+) icon to the right of the Arguments Definition field. The Edit Argument window is displayed.

6. Enter a name in the Name field, and then click the Type drop-down list and select Browse.

7. Select the Request component (for example, GetEffectiveAddress), and click OK, as shown in Figure 6–63.
Figure 6–63  Request Component

8. In the Edit Argument window that is displayed, click OK. The Properties - Start window is displayed.

9. In the Operation Name field, change the default entry from start to operation.

10. Click the Data Associations hyperlink, as shown in Figure 6–64.
11. Right-click the Data Objects node in the right pane, under Process, and select New, as shown in Figure 6–65.

The Create Data Object window is displayed.

12. Enter a name in the Name field, click the Type drop-down list, and select Browse.
13. Select the **Request** component (for example, GetEffectiveAddress) and click **OK**, as shown in Figure 6–66.

**Figure 6–66 Request Component**

14. In the Create Data Object window, click **OK**.

   The Data Associations window is displayed.

15. Select **argument1** under the Arguments node in the left pane and drag and connect it to **dataObject1**, under Data Objects, in the right pane.

16. Click **OK**, as shown in Figure 6–67.
17. In the Properties - Start window that is displayed, click OK.
   You are returned to the Process workspace area.

18. Click the Activity drop-down menu and select Service.

19. Drop the Service icon on the wire between the Start and End event components.

20. In the displayed Properties - ServiceTask window, click the Implementation tab.

21. Select Service Call from the Message Exchange Type list.

22. Click the Browse icon to the right of the Service field, as shown in Figure 6–68.
The Service window is displayed.

23. Select the Web Service that has been created and click **OK**, as shown in **Figure 6–69**.
24. In the Properties - ServiceTask window that is displayed, click the Data Associations hyperlink.

The Data Associations window is displayed.

25. Create response Data Object by right-clicking the Data Objects node in the right pane of the Output tab and selecting New, as shown in Figure 6–70.

The Create Data Object window is displayed.

26. Enter a name in the Name field, click the Type drop-down list, and select Browse.

27. Select the Response component (for example, GetEffectiveAddressResponse) and click OK, as shown in Figure 6–71.
28. In the Create Data Object window, click **OK**.
   The Data Associations window is displayed.

29. Select **dataObject1**, under the Data Objects node in the left pane of the Input tab, and drag and connect it to the **getEffectiveAddress** node, under the Arguments node in the right pane, as shown in Figure 6–72.

![Figure 6–71 Response Component](image1)

![Figure 6–72 Data Associations](image2)
30. Click on the Output tab and select `GetEffectiveAddressResponse` under the Arguments node in the left pane and drag and connect it to `dataObject2` under the Data Objects node.

31. Click OK, as shown in Figure 6–73.

**Figure 6–73 Output Tab**

![Image](image.png)

32. In the Properties - ServiceTask window that is displayed, click OK.

33. Click the Save All icon in the menu bar to save the new outbound BPM process component that was configured.

34. Double-click the composite.xml node in the left pane.

### 6.6.2.3 Creating a File Adapter for the Write Operation

This section describes how to create a File adapter for the write operation.

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

1. Drag and drop the File Adapter component from the Technology Adapters pane to the External References pane, and provide a name for the File Adapter.

2. In the Adapter Interface pane that is displayed, ensure that the Define from operation and schema (specified later) option is selected, and click Next.

3. Click Next.

4. In the Operation pane that is displayed, select Write File from the list of Operation Type options, and click Next, as shown in Figure 6–74.
The File Configuration pane is displayed.

5. In the Directory for Outgoing Files (physical path) field, specify a location on your file system where the output file is written.

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

7. Click Next, as shown in Figure 6–75.

Figure 6–75 File Configuration Pane
The Messages pane is displayed.

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

9. In the displayed Type Chooser window, expand **Project WSDL Files, IBSE_Outbound.wsdl, Inline Schemas** and then select **GetEffectiveAddressResponse**.

10. Click **OK**.

11. In the Messages pane, click **Next**.

12. In the Finish pane that is displayed, click **Finish**.

13. Double-click the **BPMN Process** component, as shown in **Figure 6–76**.

**Figure 6–76  Composite.xml Tab**

14. Click the **Activity** icon.

15. Drop the **Activity** icon on the wire between the **Service Task** and **End event** components, as shown in **Figure 6–77**.
16. In the displayed Properties-ServiceTask1 window, click the **Implementation** tab.

17. Select **Service Call** from the Type drop-down list in the Message Exchange section.

18. Click the **Browse** icon to the right of the Service field.

19. Select the service for write operation that has been created and click **OK**, as shown in **Figure 6–78**.

**Figure 6–78  Service Window**

20. In the Properties - ServiceTask1 window, click the **Data Associations** hyperlink, as shown in **Figure 6–79**.
21. In the Input tab, click the XSL Transformation icon in the top right pane.

22. Drag and drop the XSL Transformation icon to the GetEffectiveAddressResponse node, as shown in Figure 6–80.

Figure 6–80 GetEffectiveAddressResponse Node
23. In the displayed Create Transformation window, select **dataObject2** in the Sources section and click the right arrow symbol.

24. Accept the default value selected in the Target drop-down list and the default name in the Create field by clicking **OK**.

25. In the Data Associations window, click **OK**, as shown in **Figure 6–81**.

**Figure 6–81 Data Associations Window**

26. In the Properties - ServiceTask1 window, click **OK**.

27. In the response_body.xsl tab, map the **ns0:GetEffectiveAddressResponse** source element to the **ns0:GetEffectiveAddressResponse** target element.

28. In the displayed Auto Map Preferences window, retain the default values and click **OK**.

29. Return to the Process workspace area and double-click the **End** event component.

30. In the displayed Properties - End window, click the **Implementation** tab.

31. Select **None** from the Implementation Type drop-down list.

32. Click **OK**, as shown in **Figure 6–82**.
33. Click the **Save All** icon in the menu bar to save the new outbound BPM component that was configured, as shown in Figure 6–83.

You are now ready to deploy the BPM BSE Outbound process. You can follow the same procedure as Section 6.4.4, "Deploying the BPM Outbound Process" on page 6-25.

Once deployed, you can invoke the input XML as defined in Section 6.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console" on page 6-26.
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 an Outbound Process Using Sbconsole (J2CA Configuration)"
- Section 7.3, "Configuring an Inbound Process Using sbconsole (J2CA Configuration)"
- Section 7.4, "Configuring an Outbound Process Using Sbconsole (BSE Configuration)"
- Section 7.5, "Configuring JMS Proxy Services Using Oracle Service Bus (J2CA Configuration)"
- Section 7.6, "Configuring HTTP Proxy Services Using Oracle Service Bus (J2CA Configuration)"

7.1 Overview of Application Adapter Integration with Oracle Service Bus

To integrate with Oracle Service Bus (OSB), Oracle Application Adapter for 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 an Outbound Process Using Sbconsole (J2CA Configuration)

This section describes how to configure an outbound process using sbconsole for J2CA configurations.

A sample project has been provided for this outbound use case scenario in the following folder of the Application Adapters installation:
<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\OSB\J2CA\JDEdwards_Sample_J2CA_OSB_Outbound_Project

This section includes the following topics:

- Section 7.2.1, "Starting Oracle Service Bus and Creating Project Folders"
- Section 7.2.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"
- Section 7.2.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"
- Section 7.2.4, "Configuring a WSDL-based Business Service"
- Section 7.2.5, "Configuring a File Type Business Service"
- Section 7.2.6, "Configuring a Pipeline With Proxy Service"

### 7.2.1 Starting Oracle Service Bus and Creating Project Folders

This section describes how to start Oracle Service Bus (OSB) and create project folders. Perform the following steps to start Oracle Service Bus and create project folders:

1. Start the Oracle WebLogic Server for the Oracle WebLogic Server domain that you have configured.
2. Open the Oracle Service Bus Console in a Web browser by entering the following URL:
   
   `http://hostname:port/sbconsole`

   Where `hostname` is the name of the machine where Oracle WebLogic Server is running and `port` is the port for the domain you are using.

   The Oracle Service Bus Console logon page is displayed.
3. Log on to the Oracle Service Bus Console using a valid user name and password.

   The Oracle Service Bus Console home page is displayed, as shown in Figure 7–1.

   **Figure 7–1 Oracle Service Bus Console Home Page**

4. Click **Create** in the right pane of the Oracle Service Bus session, as shown in Figure 7–2.
5. Select All Projects, click the down arrow in the left pane, and select Project, as shown in Figure 7–3.

The Create a new Project window is displayed, as shown in Figure 7–4.

6. Provide a valid name for the new project (for example, J2CA_Outbound) in the Resource Name field, and click Create.

The new project is successfully created and listed.

7. Right-click the newly created project, select Create, and click Folder, as shown in Figure 7–5.
The Create a new Folder window is displayed.

8. In the Resource Name field, type **Business Service** and click **Create**.

9. Repeat steps 7 and 8 to create folders with the names **Proxy Service** and **Wsdls**.

   The Business Service, Proxy Service, and Wsdls folders are listed in the left pane below the project node, as shown in **Figure 7–6**.
10. Click **Activate** in the right pane of the Oracle Service Bus session, as shown in Figure 7–7.

**Figure 7–7 Activate Button**

11. In the Confirm Session Activation page, click **Activate** to save the changes, as shown in Figure 7–8.

**Figure 7–8 Confirm Session Activation Window**
7.2.2 Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus

Before starting and using Application Explorer to publish a WSDL directly to the Oracle Service Bus (OSB) Console (project/folder), OSB users must perform the following steps:

1. Open the command prompt window.
2. Navigate to the following directory:
   `<ORACLE_HOME>\user_projects\domains\base_domain\bin`
3. Execute `setDomainEnv.cmd` (Windows) or `./setDomainEnv.sh` (UNIX/Linux).
   This command sets the class path for Application Explorer to access the Oracle WebLogic Server APIs to publish the WSDLs to the OSB Console.
4. Do not close the command prompt window.
5. Navigate to the following directory:
   `<ADAPTER_HOME>\tools\iwae\bin`
6. Execute `ae.bat` (Windows) or `iwae.sh` (UNIX/Linux) to start Application Explorer.
   You are now ready to publish WSDLs from Application Explorer to the OSB Console.

7.2.3 Publishing a WSDL From Application Explorer to Oracle Service Bus

Perform the following steps to publish a WSDL from Application Explorer to Oracle Service Bus:

1. Start Application Explorer, connect to a J2CA configuration, and connect to a J.D. Edwards OneWorld target.
   For more information, see Chapter 2, "Configuring Oracle Application Adapter for J.D. Edwards OneWorld".
2. Expand the J.D. Edwards OneWorld target to which you are connected.
3. Right-click a method and then select Create Outbound JCA Service (Request/Response) from the menu.
   The Export WSDL dialog is displayed, as shown in Figure 7–9.
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.

### 7.2.4 Configuring a WSDL-based Business Service

Perform the following steps to configure a WSDL-based Proxy Service:

1. Open the Oracle Service Bus Console and click Create in the right pane of the Oracle Service Bus session, as shown in Figure 7–10.
2. Double-click the created WSDL folder in the left pane (for example, Wsdls) and ensure that the exported WSDL is listed in the right pane, as shown in Figure 7–11.

3. Click the icon that corresponds to the JCA Binding in the Actions column. The Generate WSDL and Service window is displayed, as shown in Figure 7–12.
4. Provide a new WSDL name and a new Business Service name in the corresponding fields.

5. In the Destination area, select an available project and the sub-folder that is designated for Business Services.

6. Click Generate.

7. Expand Business Service under the project folder and check if the generated WSDL and Business Service are listed, as shown in Figure 7–13.

**Figure 7–13 Business Service Folder**

![Business Service Folder](image)

### 7.2.5 Configuring a File Type Business Service

Perform the following steps to configure a File type Business Service:

1. Right-click the Business Service folder you created in the left pane, select Create, and click Business Service as shown in Figure 7–14.
The Create Business Service window is displayed.

2. In the Resource Name field, provide a name for the Business Service, select the **File** option in the Transport section under Service Definition, and click **Next**, as shown in Figure 7–15.
3. In the Service Type section, select **Messaging Service**. By default, the Request Type is set to XML, and the Response Type is set to None. Then click **Next**, as shown in Figure 7–16.

4. Enter the path to a destination folder on your file system in the Endpoint URI field.
5. Click Create, as shown in Figure 7–17.

![Figure 7–17 Transport Page](image)

The Business Service File_Out is created and listed under Business Service, as shown in Figure 7–18.

![Figure 7–18 File_Out Business Service](image)

6. Double-click File_Out, click Transport Detail in the left pane, and enter the prefix and suffix for the output file to be received, as shown in Figure 7–19.
7. Click the Save or Save All icon in the right corner, as shown in Figure 7–20.

7.2.6 Configuring a Pipeline With Proxy Service

Perform the following steps to configure a Pipeline:

1. Right-click the Proxy Service folder, select Create and click Pipeline, as shown in Figure 7–21.
2. Enter a name in the Pipeline Name field. By default, **Expose as a Proxy Service** is selected. If you wish to change the Proxy Service Name, change it and set Transport as **file**, and click **Create** as shown in **Figure 7–22**.

![Pipeline Option](image1.png)

The Create Pipeline window is displayed.

![Create Pipeline Window](image2.png)
The created Pipeline and the Proxy Service is listed under Proxy Service, as shown in Figure 7–23.

**Figure 7–23 Pipeline Node**

3. Double-click the created proxy service and click **Transport** in the left pane. Provide the input location in the Endpoint URI field, as shown in Figure 7–24.

**Figure 7–24 Transport**

4. Click **Transport Details** in the left pane and provide the location for the Stage Directory and the Error Directory fields, as shown in Figure 7–25.
5. Click the **Save All** icon in the right corner, as shown in **Figure 7–26**.

![Figure 7–26 Save All Icon](image1)

6. Double-click the **Pipeline** node and click the **Open Message Flow** icon on the right pane to open the message flow, as shown in **Figure 7–27**.

![Figure 7–27 Open Message Flow Icon](image2)

7. Click the Proxy Service icon and select **Add Pipeline Pair** from the menu, as shown in **Figure 7–28**.
8. Click the PipelinePairNode1 icon and select Add Route from the menu, as shown in Figure 7–29.

The RouteNode1 icon is added below the PipelinePairNode1 icon.

9. Click the RouteNode1 icon and select Edit Route from the menu, as shown in Figure 7–30.
The Edit Stage Configuration workspace area is displayed.

10. Click **Add an Action**, select **Communication** and click **Routing**, as shown in Figure 7–31.

11. Click **<Service>**, as shown in Figure 7–32.
The Select Service dialog is displayed.

12. Select the WSDL type Business Service configured for J.D. Edwards OneWorld and click on **Submit**, as shown in **Figure 7–33**.

![Select Service Dialog](image)

**Figure 7–33  Select Service Dialog**

13. Select the name of the J.D. Edwards OneWorld business object as the operational attribute from the list, and click **Save**.

14. Click the Response Pipeline icon and select **Add Stage** from the menu, as shown in **Figure 7–34**.

![Response Pipeline Icon](image)

**Figure 7–34  Response Pipeline Icon**

The Stage1 icon is added below the Response Pipeline icon.

15. Click the Stage1 icon and select **Edit Stage** from the menu, as shown in **Figure 7–35**.
The Edit Stage Configuration workspace area is displayed.

16. Click **Add an Action**, select **Communication**, and then click **Publish**, as shown in Figure 7–36.

17. Click **<Service>**, as shown in Figure 7–37.
18. In the Select Service dialog, select a File type Business Service and click Submit, as shown in Figure 7–38.

Figure 7–38 Select Service Dialog

19. Click Save All, as shown in Figure 7–39.

Figure 7–39 Save All Button

20. Click Activate in the right pane of the Oracle Service Bus session, as shown in Figure 7–40.

Figure 7–40 Activate Button
21. Click **Activate** to save the changes, as shown in Figure 7–41.

**Figure 7–41  Confirm Session Activation**

22. Copy and paste an input XML file in the input folder you have configured (for example, C:\input). Output is received in the configured output location (for example, C:\output).

### 7.3 Configuring an Inbound Process Using sbconsole (J2CA Configuration)

This section describes how to configure an inbound process using sbconsole for J2CA configurations.

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

<ADAPTER_HOME>\etc\sample\JDEdwards_Samples.zip\JDEdwards_Samples\OSB\J2CA\JDEdwards_Sample_J2CA_OSB_Inbound_Project

This section includes the following topics:
- Section 7.3.1, "Starting Oracle Service Bus and Creating Project Folders"
- Section 7.3.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"
- Section 7.3.3, "Generating WSDL for Event Integration"
- Section 7.3.4, "Configuring a WSDL-based Proxy Service"
- Section 7.3.5, "Configuring a File Type Business Service"
- Section 7.3.6, "Configuring a Pipeline"

#### 7.3.1 Starting Oracle Service Bus and Creating Project Folders

For more information on starting Oracle Service Bus and creating project folders, see Section 7.2.1, "Starting Oracle Service Bus and Creating Project Folders" on page 7-2.

#### 7.3.2 Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus

For more information on setting the class path for Application Explorer to integrate with Oracle Service Bus, see Section 7.2.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus" on page 7-6.
7.3.3 Generating WSDL for Event Integration

You cannot publish inbound WSDL for J.D. Edwards OneWorld event notification using Application Explorer. To generate WSDL from the command prompt, see Section 4.5.1, "Generating WSDL for Event Integration" on page 4-34.

7.3.4 Configuring a WSDL-based Proxy Service

Perform the following steps to select the inbound WSDL from the File system and configure a WSDL-based Proxy Service:

1. Open the Oracle Service Bus Console and click **Create** in the right pane of the Oracle Service Bus session, as shown in **Figure 7–42**.

```
Figure 7–42  Create Button

![Create Button](image)
```

2. Right-click the WSDL folder, select **Create**, and click **Schema**, as shown in **Figure 7–43**.

```
Figure 7–43  Schema Option

![Schema Option](image)
```

3. In the displayed window, click **Choose File** and select the available schema file (for example, J2CA_Inbound_receive_request.xsd), as shown in **Figure 7–44**.

```
Figure 7–44  Choose File

![Choose File](image)
```
4. Click **Create**, as shown in **Figure 7–45**.

**Figure 7–45  Create Button**

5. Right-click the WSDL folder, select **Create**, and click **WSDL**. Repeat the steps and select the WSDL file.

6. Right-click the WSDL folder, select **Create**, and click **JCA Binding**. Repeat the steps and select the JCA file.

7. Double-click the created WSDL folder in the left pane (for example, Wsds), and ensure that the WSDL is listed in the right pane, as shown in **Figure 7–46**.

**Figure 7–46  Exported WSDL**

8. Click the icon that corresponds to the JCA Binding in the Actions column. The Generate WSDL and Service page is displayed, as shown in **Figure 7–47**.
9. Provide a new WSDL name and a new Proxy Service name in the corresponding fields.

10. In the Destination area, select an available project and the sub-folder that is designated for Proxy Services.

11. Click Generate.

12. Expand Proxy Service under Project Explorer and check if the generated WSDL and Proxy Service are listed, as shown in Figure 7–48.

Figure 7–48 Generated WSDL

7.3.5 Configuring a File Type Business Service

Perform the following steps to configure a File type Business Service:
1. Right-click the Business Service folder you created in the left pane, select Create, and click Business Service, as shown in Figure 7–49.

![Business Service Folder](image)

Figure 7–49 Business Service Folder

The Create Business Service window is displayed.

2. In the Resource Name field, provide a name for the Business Service and select the File option from the Transport drop-down list in the Service Definition area, as shown in Figure 7–50.

![Create Business Service Window](image)

Figure 7–50 Create Business Service Window

3. Click Next.
4. In the Service Type area, select **Messaging Service** as the service type, as shown in Figure 7–51.

*Figure 7–51  Service Type Area*

5. Click **Next**.

   The Transport page is displayed, as shown in Figure 7–52.

*Figure 7–52  Transport Page*
6. Enter the path to a destination folder on your file system in the Endpoint URI field and click **Create**.

The Business Service File_Out is created and listed under Business Service, as shown in Figure 7–53.

---

**Figure 7–53  File_Out Business Service**

![Image of Oracle Service Bus Console showing File_Out Business Service]

---

7. Double-click **File_Out**, click **Transport Detail** in the left pane, and enter the prefix and suffix for the output file to be received, as shown in Figure 7–54.

---

**Figure 7–54  Transport Detail Page**

![Image of Business Service Definition with Transport Detail configuration]

---

8. Click the Save or Save All icon in the right corner, as shown in Figure 7–55.
7.3.6 Configuring a Pipeline

Perform the following steps to configure a Pipeline:

1. Right-click the proxy service you created and select Create, and then click Pipeline, as shown in Figure 7–56.

The Create Pipeline window is displayed.

2. In the Pipeline Name field, enter a name and select the Service Type as WSDL Based Service, as shown in Figure 7–57.
3. Click the Search icon, and in the displayed Search and Select: WSDL Resource window, select J2CA_Inbound_receive_wsd1, and click OK, as shown in Figure 7–58.

4. Clear the check box for Expose as a Proxy Service, and click Create, as shown in Figure 7–59.
The pipeline is created and listed under Proxy Service, as shown in Figure 7–60.

5. Double-click the J2CA_Inbound_receive_PS node under Proxy Service in the left pane and click the Search icon in the Target area in right pane, as shown in Figure 7–61.
The Search and Select: Service Resource window appears.

6. From the Resource Type drop-down list, select **Pipeline** and then click the **Search** button.
   
   The Pipeline is listed, as shown in **Figure 7–62**.

**Figure 7–62  Search and Select: Service Resource Window**

7. Select the Pipeline and click **OK**.

8. Click the Save or Save All icon in the right corner, as shown in **Figure 7–63**.

**Figure 7–63  Save and Save All Icons**
9. In the left pane, double-click **Pipeline** under the Proxy Service folder and click the down-pointing icon on the right pane to open the message flow, as shown in Figure 7–64.

![Figure 7–64 Message Flow](image)

10. Click the displayed Proxy service icon and select **Add Route** from the menu, as shown in Figure 7–65.

![Figure 7–65 Add Route Option](image)

The RouteNode1 icon is added.

11. Click the RouteNode1 icon and select **Edit Route** from the menu, as shown in Figure 7–66.
The Edit Stage Configuration workspace area is displayed.

12. Click Add an Action, select Communication from the menu, and then click Routing, as shown in Figure 7–67.

13. Click <Service>, as shown in Figure 7–68.
14. Select the **File_Out** Business service and click **Submit** as shown in Figure 7–69.

*Figure 7–69  Select Service Dialog*

You are returned to the Edit Stage Configuration workspace area.

15. Click **Save All**, as shown in Figure 7–70.

*Figure 7–70  Edit Stage Configuration Workspace Area*

16. Click **Activate** in the right pane of the Oracle Service Bus session, as shown in Figure 7–71.

*Figure 7–71  Activate Button*

The Confirm Session Activation window appears.

17. Click **Activate** to save the changes, as shown in Figure 7–72.
7.4 Configuring an Outbound Process Using Sbconsole (BSE Configuration)

This section describes how to configure an outbound process using sbconsole for BSE configurations.

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

<ADAPTER_HOME>/etc/sample\JDEdwards_Samples.zip\JDEdwards_Samples\OSB\BSE\JDEdwards_Sample_BSE_OSB_Outbound_Project

This section includes the following topics:

- Section 7.4.1, "Starting Oracle Service Bus and Creating Project Folders"
- Section 7.4.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus"
- Section 7.4.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus"
- Section 7.4.4, "Configuring a File Type Business Service"
- Section 7.4.5, "Configuring a WSDL-based Business Service"
- Section 7.4.6, "Configuring a Pipeline With Proxy Service"

7.4.1 Starting Oracle Service Bus and Creating Project Folders

For more information on starting Oracle Service Bus and creating project folders, see Section 7.2.1, "Starting Oracle Service Bus and Creating Project Folders" on page 7-2.

7.4.2 Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus

For more information on setting the class path for Application Explorer to integrate with Oracle Service Bus, see Section 7.2.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus" on page 7-6.

18. Trigger an event from the J.D. Edwards OneWorld system and check if the output is received in the configured output location.

For more information on triggering an event, see Section 4.5.5, "Triggering an Event in J.D. Edwards OneWorld" on page 4-47.
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 OneWorld target.
2. Expand Services, CALLBSFN, and then AddressBook business object.
3. Right-click GetEffectiveAddress and select Create Web Service from the menu.
   The Create Web Service dialog is displayed, as shown in Figure 7–73.

   Figure 7–73 Create Web Service Dialog

   ![Create Web Service Dialog]

4. Enter a service name and click Next.
5. Click OK on the next dialog that is displayed.
   Application Explorer switches the view to the Business Services node, and the new Web service appears in the left pane.
6. Right-click the new Web service and select Export WSDL from the menu.
   The Export WSDL dialog is displayed, as shown in Figure 7–74.

   Figure 7–74 Export WSDL Dialog

   ![Export WSDL Dialog]

7. In the Name field, a default file name for the WSDL file is provided. You can accept the default or provide your own.
8. In the Location field, enter the location where you want to publish the WSDL document.
   The location is composed of an Oracle Service Bus project name and optionally, one or more folder names. The project name and any folder names must be separated by a forward slash character “/”.

9. In the Host field, enter the name of the machine where Oracle WebLogic Server is running.

10. In the Port field, enter the port for the domain you are using.

11. In the User field, enter your username to access Oracle Service Bus.

12. In the Password field, enter your password to access Oracle Service Bus.

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

### 7.4.4 Configuring a File Type Business Service

For more information on configuring a file type business service, see Section 7.2.5, "Configuring a File Type Business Service" on page 7-9.

### 7.4.5 Configuring a WSDL-based Business Service

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

Perform the following steps to configure a WSDL-based Proxy Service:

1. Right-click on the Business Service folder in the left pane and select **Business Service**.

   The Create Business Service window is displayed, as shown in Figure 7–75.
2. Provide a name for the Business Service, and in Service Definition area, select the WSDL Based Service option and click the search icon. The Search and Select: WSDL Resource window is displayed, as shown in Figure 7–76.

Figure 7–76 Search and Select: WSDL Resource Window

3. Click the Search button, select the BSE Outbound WSDL, and click OK. You are returned to the Create Business Service window.
4. Click Next.

5. Accept the default values and click the Create button, as shown in Figure 7–77.

**Figure 7–77  Create Business Service Window**

![Create Business Service Window](image)

The created WSDL-based Business Service is listed under the Business Service folder, as shown in Figure 7–78.

**Figure 7–78  WSDL-based Business Service**

![WSDL-based Business Service](image)

7.4.6 Configuring a Pipeline With Proxy Service

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

1. Right-click the Proxy Service folder, select Create and click Pipeline, as shown in Figure 7–79.
The Create Pipeline window is displayed.

2. Enter a name in the Pipeline Name field. By default, **Expose as a Proxy Service** is selected. If you wish to change the Proxy Service Name, change it and set Transport to **file**, and click **Create** as shown in Figure 7–80.

The created Pipeline and the Proxy Service is listed under Proxy Service, as shown in Figure 7–81.
3. Double-click the created proxy service and click **Transport** in the left pane. Provide the input location in the Endpoint URI field, as shown in Figure 7–82.

4. Click **Transport Details** in the left pane and provide the location for the Stage Directory and the Error Directory fields, as shown in Figure 7–83.
5. Click the **Save All** icon in the right corner, as shown in Figure 7–84.

![Figure 7–84 Save All Icon](image)

6. Double-click the **Pipeline** node and click the **Open Message Flow** icon on the right pane to open the message flow, as shown in Figure 7–85.

![Figure 7–85 Open Message Flow Icon](image)

7. Click the Proxy Service icon and select **Add Pipeline Pair** from the menu, as shown in Figure 7–86.
8. Click the **PipelinePairNode1** icon and select **Add Route** from the menu, as shown in **Figure 7–87**.

   **Figure 7–87  Add Route Option**

   The RouteNode1 icon is added below the PipelinePairNode1 icon.

9. Click the RouteNode1 icon and select **Edit Route** from the menu, as shown in **Figure 7–88**.
The Edit Stage Configuration workspace area is displayed.

10. Click Add an Action, select Communication and click Routing, as shown in Figure 7–89.

11. Click <Service>, as shown in Figure 7–90.
The Select Service dialog is displayed.

12. Select the WSDL type Business Service configured for J.D. Edwards OneWorld and click on **Submit**, as shown in **Figure 7–91**.

![Figure 7–91 Select Service Dialog](image)

13. Select the name of the J.D. Edwards OneWorld business object as the operational attribute from the list, and click **Save**.

14. Click the Response Pipeline icon and select **Add Stage** from the menu, as shown in **Figure 7–92**.

![Figure 7–92 Response Pipeline Icon](image)

The Stage1 icon is added below the Response Pipeline icon.

15. Click the Stage1 icon and select **Edit Stage** from the menu, as shown in **Figure 7–93**.
The Edit Stage Configuration workspace area is displayed.

16. Click Add an Action, select Communication, and then click Publish, as shown in Figure 7–94.

17. Click <Service>, as shown in Figure 7–95.
18. In the Select Service dialog, select a File type Business Service and click **Submit**, as shown in Figure 7–96.

![Figure 7–96 Select Service Dialog](image)

19. Click **Save All**, as shown in Figure 7–97.

![Figure 7–97 Save All Button](image)

20. Click **Activate** in the right pane of the Oracle Service Bus session, as shown in Figure 7–98.

![Figure 7–98 Activate Button](image)

21. Click **Activate** to save the changes, as shown in Figure 7–99.
22. Copy and paste an input XML file in the input folder you have configured (for example, C:\input). Output is received in the configured output location (for example, C:\output).

### 7.5 Configuring JMS Proxy Services Using Oracle Service Bus (J2CA Configuration)

This section describes how to configure JMS Proxy Services using Oracle Service Bus for a J2CA configuration.

1. Start Oracle Service Bus and create the required project folder.
   For more information, see Section 7.2.1, "Starting Oracle Service Bus and Creating Project Folders" on page 7-2.

2. Generate and publish the WSDL from Application Explorer to the created project folder. Using the published WSDL, create a Business Service.
   For more information, see Section 7.2.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus" on page 7-6.

3. Open the Service Bus Console page, as shown in Figure 7–100.

---

**Figure 7–99  Confirm Session Activation**

![Confirm Session Activation](image)

**Figure 7–100  Service Bus Console**

![Service Bus Console](image)
4. Select the ProxyService project folder in the left pane, and click Create, as shown in Figure 7–101.

Figure 7–101 Proxy Service

5. In the right pane, select Proxy Service from the Create Resource list, as shown in Figure 7–102.

Figure 7–102 Create Resource Menu

6. Enter an appropriate name in the Service Name field, as shown in Figure 7–103.
7. In the Service Type section, under Create From Existing Service, select the Business Service radio button and click Browse, as shown in Figure 7–104.

8. Select the existing business service and click Submit, as shown in Figure 7–105.
9. Click Next, as shown in Figure 7–106.

**Figure 7–106  Next**

10. Select *jms* from the Protocol list and click Next, as shown in Figure 7–107.
11. Provide the following parameters, as shown in Figure 7–108.
   a. Select **Queue** in the Destination Type section.
   b. Enable the **Is Response Required** check box.
   c. Select **Text** in the Response Message Type section.
   d. In the Response URI field, provide the Endpoint URI used in the Transport Configuration and change **Request** to **Response**.

   For example:

   jms://localhost:8001/weblogic.jms.XAConnectionFactory/Adapter_outbound_PSResponse

**Figure 7–108  Edit a Proxy Service**
12. Click Next.

The Operation Selection Configuration pane appears, as shown in Figure 7–109.

**Figure 7–109  Operation Selection Configuration Pane**

13. Ensure the SOAP Body Type is selected and click Next.

14. Enable the Transaction Required box and click Next, as shown in Figure 7–110.

**Figure 7–110  Message Handling**

15. Click Save, as shown in Figure 7–111.
Figure 7–111  Save

The created Proxy Service is saved, as shown in Figure 7–112.

Figure 7–112  Proxy Service

16. In the left pane, click Activate, and then Submit, as shown in Figure 7–113.
17. In the left pane, click **ProxyService** under the Projects folder, as shown in Figure 7–114.

**Figure 7–114 Adapter/ProxyService**

18. Click the **Launch Test Console** icon for the created Proxy Service, as shown in Figure 7–115.

**Figure 7–115 Launch Test Console Icon**

19. Provide the input values for **Payload**, uncheck the **Direct Call** box, and click **Execute**.

20. Review the Response document, and then click **Close**.

21. Click the **Oracle WLS Console** tab, as shown in Figure 7–116.
22. In the Oracle WLS Console, expand **Services**, expand **Messaging**, and click **JMS Modules**, as shown in **Figure 7–117**.

**Figure 7–117  Oracle WLS Console**

23. Click **jmsResources**, as shown in **Figure 7–118**.
24. Click Lock & Edit, as shown in Figure 7–119.

25. Click the appropriate request link, for example, Adapter_outbound_PSRequest, as shown in Figure 7–120.
26. Click the Monitoring tab, as shown in Figure 7–121.

Figure 7–121 Monitoring Tab

27. Enable the check box and click Show Messages, as shown in Figure 7–122.
28. Click **New**, as shown in **Figure 7–123**.

**Figure 7–123  JMS Messages**

29. Provide the input payload in the **Body** field and click **OK**.

A Success message appears, as shown in **Figure 7–124**.
30. In the Oracle WLS Console, expand **Services**, expand **Messaging**, and click **JMS Modules**, as shown in Figure 7–125.

31. Click **jmsResources**, as shown in Figure 7–126.
Figure 7–126  jmsResources

![Image](image_url)

32. Click the appropriate response link, for example, Adapter_outbound_PSResponse, as shown in Figure 7–127.

Figure 7–127  Summary of Resources

![Image](image_url)

33. Click the Monitoring tab, as shown in Figure 7–128.
34. Enable the check box and click **Show Messages**, as shown in **Figure 7–129**.

**Figure 7–129  Destination Messages**

35. Click the **ID** link, as shown in **Figure 7–130**.
7.6 Configuring HTTP Proxy Services Using Oracle Service Bus (J2CA Configuration)

This section describes how to configure HTTP Proxy Services using Oracle Service Bus for a J2CA configuration.

1. Start the Oracle Service Bus and create the required project folders.
   
   For more information, see Section 7.2.1, "Starting Oracle Service Bus and Creating Project Folders" on page 7-2.

2. Generate and publish the WSDL from Application Explorer to the created project folder, and create a Business Service using the published WSDL.
   
   For more information, see Section 7.2.3, "Publishing a WSDL From Application Explorer to Oracle Service Bus" on page 7-6.

3. Open the Service Bus console page, as shown in Figure 7-131.
4. In the Project Explorer, select the ProxyService project folder, and click Create, as shown in Figure 7–132.

5. In the Create Resource list on the right pane, select Proxy Service, as shown in Figure 7–133.
6. In the Service Name field, enter an appropriate name, as shown in Figure 7–134.

7. In the Service Type section, under Create From Existing Service, select the Business Service radio button and click Browse, as shown in Figure 7–135.
8. Select the existing Business Service and click **Submit**, as shown in **Figure 7–136**.

**Figure 7–136  Business Service**

9. Click **Next**, as shown in **Figure 7–137**.
10. Select **http** in the Protocol list and click **Next**, as shown in Figure 7–138.

**Figure 7–138  Transport Configuration**

11. Click **Next**, as shown in Figure 7–139.
Figure 7–139  HTTP Transport Configuration

12. Click Next, as shown in Figure 7–140.

Figure 7–140  Operation Selection Configuration

13. Enable the Transaction Required check box and click Next, as shown in Figure 7–141.
14. Click Save, as shown in Figure 7–142.

The created Proxy Service is saved, as shown in Figure 7–143.
15. Click **Activate** in the left pane, and then **Submit** on the right pane, as shown in Figure 7–144.

**Figure 7–144  Activate Session**

16. Click **ProxyService** in the Projects folder on the left pane, as shown in Figure 7–145.
17. Click the **Launch Test Console** icon for the created Proxy Service, as shown in Figure 7–146.

**Figure 7–146  Launch Test Console**

18. Uncheck the **Direct Call** check box, provide the input values for **Payload**, and click **Execute**.

19. Review the **Response Document**.
Oracle Application Adapter for J.D. Edwards OneWorld integrates seamlessly with Oracle JDeveloper to facilitate Web service integration.

This chapter contains the following sections:

- Section 8.1, "Configuring an OSB Outbound Process Using JDeveloper (J2CA Configuration)"
- Section 8.2, "Configuring an OSB Inbound Process Using JDeveloper (J2CA Configuration)"
- Section 8.3, "Configuring an OSB Outbound Process Using JDeveloper (BSE Configuration)"
- Section 8.4, "Configuring a JMS Inbound Process Using JDeveloper (J2CA Configuration)"
- Section 8.5, "Configuring a JMS Outbound Process Using JDeveloper (J2CA Configuration)"
- Section 8.6, "Configuring an HTTP Outbound Process Using JDeveloper (J2CA Configuration)"

8.1 Configuring an OSB Outbound Process Using JDeveloper (J2CA Configuration)

This section describes how to configure an OSB outbound process to your J.D. Edwards OneWorld system, using Oracle JDeveloper for J2CA configurations.

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

```bash
<ADAPTER_HOME>/etc/sample\JDEdwards_Samples.zip\JDEdwards_Samples\OSB_Jdeveloper\J2CA\JDEdwards_Sample_J2CA_OSB_Outbound_Project
```

This section includes the following topics:

- Section 8.1.1, "Creating a Service Bus Application for OSB"
- Section 8.1.2, "Defining an OSB Outbound Process"
- Section 8.1.3, "Deploying the OSB Outbound Process"
Prerequisites
Before you design an OSB outbound process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.4.1, "Generating WSDL for Request/Response Service" on page 4-8.

8.1.1 Creating a Service Bus Application for OSB
Perform the following steps in JDeveloper to create a service bus application for OSB.

1. Create a new OSB application.

2. Enter a name for the OSB Application (for example, J2CA_Outbound) and click Finish, as shown in Figure 8–1.

Figure 8–1 Name Your Application Pane

3. Enter a project name (for example, JCA_Outbound), and click Finish, as shown in Figure 8–2.
8.1.2 Defining an OSB Outbound Process

This section describes how to define an OSB outbound process. The following topics are included:

- Section 8.1.2.1, "Configuring a Third-Party Adapter Service Component"
- Section 8.1.2.2, "Configuring a File Transport Type Business Service"
- Section 8.1.2.3, "Creating a Proxy Service With Pipeline"
- Section 8.1.2.4, "Configuring the Routing Rules"

8.1.2.1 Configuring a Third-Party Adapter Service Component

Perform the following steps to create a third party adapter service component along with the Business Service:

1. Drag and drop the Third Party Adapter component from the Service Bus Components pane to the External Services pane, as shown in Figure 8–3.
The Create Third Party Adapter Service dialog is displayed, as shown in Figure 8–4.

2. Enter an appropriate name for the Third Party Adapter Service which will be used as the Business Service name.
3. Ensure that **Reference** is selected from the Type drop-down list (by default).
4. Click the Find existing WSDLs icon, which is located to the right of the WSDL URL field.

The WSDL Chooser dialog is displayed, as shown in Figure 8–5.
5. Select the **File System** tab, then browse, and select an outbound WSDL file from the WSDL directory.

6. Click **OK**.

   The Import Service Bus Resources dialog is displayed.

7. Click **Next**, as shown in **Figure 8–6**.

**Figure 8–6 Source Pane**

8. In the Configuration pane, click **Finish**.
You are returned to the Create Third Party Adapter Service Dialog.

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

10. Select the JCA properties file from the WSDL directory.

11. Click OK. The Copy File message is displayed.

12. Click Yes.

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

You are returned to the Create Third Party Adapter Service dialog, as shown in Figure 8–7.

**Figure 8–7 Create Third Party Adapter Service Dialog**

13. Click OK.

The Business service component is created in the External Services pane.

**8.1.2.2 Configuring a File Transport Type Business Service**

Perform the following steps to create a File Transport Business Service:

1. Drag and drop the **File Transport** component from the Advanced pane to the External Services pane.

The Create Business Service dialog is displayed.

2. In the Service Name field, enter any name you wish for the Business Service (for example, FileOut), and click **Next**, as shown in Figure 8–8.
The Type pane is displayed. The **Any XML** option is selected by default.

3. Click **Next**, as shown in **Figure 8–9**.

4. Provide the output location in the Endpoint URI field (for example, `c:/output`) and click **Finish**, as shown in **Figure 8–10**.
The File Transport Business service Fileout is created and displayed.

5. Double-click the created Business service **Fileout** and provide the values for the Prefix and Suffix fields in the Transport Details Tab, as shown in **Figure 8–11**.

### Figure 8–11  Transport Details

#### 8.1.2.3 Creating a Proxy Service With Pipeline

Perform the following steps to create a Proxy Service with Pipeline:

1. Drag and drop the **File Transport** component from the Advanced Components pane to the Proxy Services pane, as shown in **Figure 8–12**.
2. In the Service Name field, enter any name you wish for the Proxy service (for example, JCA_Outbound_PS). By default, **Generate Pipeline** is selected.

3. Click **Next**, as shown in Figure 8–13.

4. Select the **Messaging** option, set the Request to **XML** and Response as **None**, and then click **Next**, as shown in Figure 8–14.
The Transport window is displayed.

5. Provide the input location in the Endpoint URI field (for example, c:/input) and click Finish, as shown in Figure 8–15.

The Proxy service along with the pipeline is created and displayed.

6. Double-click the created Proxy Service (for example: JCA_Outbound_PS), as shown in Figure 8–16.
7. In the displayed Proxy Service configuration page, select **Transport Details** and provide the values for Stage and Error Directory, as shown in **Figure 8–17**.

8. Save and close the Proxy Service configuration page.

### 8.1.2.4 Configuring the Routing Rules

Perform the following steps to configure the routing rules:

1. Connect the Pipeline to the Business Service (for example, Service) as shown in **Figure 8–18**.
Figure 8–18 Business Service Pipeline

2. Double-click on the pipeline (for example, JCA_Outbound_PSPipeline) in the Pipelines/Split Joins pane. The Pipeline configuration page is displayed.

3. Drag and drop the Pipeline Pair node from Nodes pane to the area below the Pipeline (for example: JCA_Outbound_PSPipeline), as shown in Figure 8–19.

Figure 8–19 Pipeline Pair Node

4. Drag and drop the Publish node from the Communication pane to the area beneath Stage1 of the Response Pipeline, as shown in Figure 8–20.
5. Click on the browse icon to the right of the Service field in the right pane of Publish Properties, as shown in Figure 8–21.

6. In the displayed Resource Chooser window, select the `Fileout.bix` File Transport Business service and click OK, as shown in Figure 8–22.
In the right pane, the selected service is configured in the Publish pane, as shown in Figure 8–23.

7. Click on the Routing to verify the Service is selected properly, as shown in Figure 8–24.
8. Save and Close the Pipeline configuration page.

9. Double-click the overview.xml file (for example: JCA_Outbound), and click **Save All** in the menu bar to save the OSB process, as shown in **Figure 8–25**.

**Figure 8–25  Save All Icon**

---

**8.1.3 Deploying the OSB Outbound Process**

Perform the following steps to deploy the OSB outbound process.
1. Right-click the OSB project, select **Deploy**, and then select **OSB_Project1_ServiceBusProjectProfile...**, as shown in **Figure 8–26**.

\[Image\]

**Figure 8–26**  **Deploy Option**

The Deployment Action page is displayed.

2. Click **Next**, as shown in **Figure 8–27**.

\[Image\]

**Figure 8–27**  **Deployment Action Page**

The Select Server page is displayed.

3. Select an available application server that was configured and click **Next**, as shown in **Figure 8–28**.
4. Review and verify all the available deployment information for your project and click Finish.

The process is deployed successfully, as shown in Figure 8–30.
This section describes how to configure an OSB inbound process to your J.D. Edwards OneWorld system, using Oracle JDeveloper for J2CA configurations.

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

<ADAPTER_HOME>/etc/sample/JEdwards_Samples.zip/JEdwards_Samples/OSB_Jdeveloper/J2CA/JEdwards_Sample_J2CA_OSB_Inbound_Project

This section includes the following topics:

- Section 8.2.1, "Creating a Service Bus Application for OSB"
- Section 8.2.2, "Defining an OSB Inbound Process"
- Section 8.2.3, "Deploying the OSB Inbound Process"

**Prerequisites**

Before you design an OSB inbound process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.5.1, "Generating WSDL for Event Integration" on page 4-34.

### 8.2.1 Creating a Service Bus Application for OSB

To configure an OSB inbound process, you must create service bus application for OSB. For more information, see Section 8.1.1, "Creating a Service Bus Application for OSB" on page 8-2.

### 8.2.2 Defining an OSB Inbound Process

This section describes how to define an OSB inbound process. The following topics are included:

- Section 8.2.2.1, "Configuring a Third-Party Adapter Service Component"
- Section 8.2.2.2, "Creating a Pipeline"
- Section 8.2.2.3, "Configuring a File Transport Type Business Service"
8.2.2.1 Configuring a Third-Party Adapter Service Component

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

1. Drag and drop the Third Party adapter component from the Service Bus Components Pane to the Proxy Services, as shown in Figure 8–31.

![Figure 8–31 Third Party Adapter Service Component](image)

The Create Third Party Adapter Service dialog is displayed.

2. Enter any name you wish for the Third Party Adapter Service or leave it to the default value.

3. Ensure that Service is selected from the Type drop-down list (by default).

4. Click the Find existing WSDLs icon, which is located to the right of the WSDL URL field, as shown in Figure 8–32.
The WSDL Chooser dialog is displayed.

5. Select the File system folder, then browse and select an inbound WSDL file from the WSDL directory.

6. Click OK.

The Import Service Bus Resources dialog is displayed.

7. Click Next.

8. In the Configuration window, click Finish.

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

9. Click the Find JCA file icon, which is located to the right of the JCA File field.

The Transformation Chooser dialog is displayed.

10. Select the JCA properties file from the WSDL directory.

11. Click OK.

The Copy File message is displayed.

12. Click Yes.

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

You are returned to the Create Third Party Adapter Service dialog, as shown in Figure 8–33.
13. Click **OK**.

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

### 8.2.2.2 Creating a Pipeline

Perform the following steps to generate inbound proxy service with Pipeline:

1. Under Service Bus, click **Resources**.
2. Drag and drop the Pipeline to the Pipelines/Split Joins pane.
3. Provide a name for the Pipeline and click next, as shown in Figure 8–34.
4. In the Create Pipeline Service window, select the **WSDL** option and click on the WSDL URL.

5. Select **Application** in the WSDL chooser window, then select `service-concrete.wsdl` in the appropriate OSB project, and then click **OK**, as shown in Figure 8–35.

6. Clear the Expose as a Proxy Service check box and click **Finish**, as shown in Figure 8–36.
Drag and drop the Proxy Service to the Pipelines/Split Joins pane.

### 8.2.2.3 Configuring a File Transport Type Business Service

Perform the following steps to create the File Transport Type Business Service:

1. Drag and drop the File Transport component from the Advanced pane to the External Services pane, as shown in Figure 8–37.

The Create Business Service dialog is displayed.

2. In the Service Name field, enter any name you wish for the Business Service (for example, FileOut), and click Next.

   In the displayed Type Window, the Any XML option is selected by default.
3. Click Next.

4. In the displayed Transport window, provide the output location in the Endpoint URI field (for example, c:\output), and click Finish, as shown in Figure 8–38.

![Figure 8–38 Transport Pane](image)

The FileOut Business service is created.

5. Double-click the FileOut Business service, as shown in Figure 8–39.

![Figure 8–39 FileOut Business Service](image)

The Configuration page is displayed.

6. Navigate to the Transport Details tab and provide the values for the Prefix and Suffix fields, as shown in Figure 8–40.
7. Save and close the Configuration page.

8.2.2.4 Configuring the Routing Rules
Perform the following steps to configure the routing rules.

1. Create a connection between the Pipeline (for example, JCA_IB_receive_PSPipeline) and the File Type Business Service (for example, FileOut), as shown in Figure 8–41.

**Figure 8–41 Mapping Proxy and FileOut**

2. Double-click the Pipeline (for example, J2CA_Inbound_receive_PSPipeline).

3. Click the Routing pane and ensure that the File Type Business Service (for example, FileOut) is properly configured in the Service field, as shown in Figure 8–42.
4. Save and close the Pipeline configuration page.

5. Double-click on the overview.xml file (for example, JCA_Inbound) and click Save All in the menu bar to save the OSB process, as shown in Figure 8–43.

**Figure 8–42 Routing Pane**

![Routing Pane](image1.png)

**Figure 8–43 Save All**

![Save All](image2.png)

### 8.2.3 Deploying the OSB Inbound Process

To deploy the created OSB inbound process, see steps 1 - 4 in Section 8.1.3, "Deploying the OSB Outbound Process" on page 8-15.

Once the OSB inbound process is deployed successfully, trigger an event from the J.D. Edwards OneWorld system and check if the output is received in the configured output location (for example, C:\output).

For more information on triggering an event, see Section 4.5.5, "Triggering an Event in J.D. Edwards OneWorld" on page 4-47.
8.3 Configuring an OSB Outbound Process Using JDeveloper (BSE Configuration)

This section describes how to configure an OSB outbound process to your J.D. Edwards OneWorld system, using Oracle JDeveloper for BSE configurations.

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

<ADAPTER_HOME>/etc/sample/JDEdwards_Samples.zip/JDEdwards_Samples/OSB/BSE/JDEdwards_Sample_BSE_OSB_Outbound_Project

This section includes the following topics:

- Section 8.3.1, "Creating a Service Bus Application for OSB"
- Section 8.3.2, "Defining an OSB Outbound Process"
- Section 8.3.3, "Deploying the OSB Outbound Process"

### Prerequisites

Before you design an OSB outbound process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.6.1, "Generating a WSDL File for Request and Response Services Using a Web Service" on page 4-52.

#### 8.3.1 Creating a Service Bus Application for OSB

To configure an OSB outbound process, you must create a service bus application for OSB. For more information, see Section 8.1.1, "Creating a Service Bus Application for OSB" on page 8-2.

#### 8.3.2 Defining an OSB Outbound Process

This section describes how to define an OSB outbound process. The following topics are included:

- Section 8.3.2.1, "Configuring a WSDL-based Business Service"
- Section 8.3.2.2, "Creating a Proxy Service With Pipeline"
- Section 8.3.2.3, "Configuring a File Transport Type Business Service"
- Section 8.3.2.4, "Configuring the Routing Rules"

##### 8.3.2.1 Configuring a WSDL-based Business Service

Perform the following steps to configure a WSDL-based Business Service:

1. Drag and drop the **HTTP** component from the Technology Components pane to the External Services area, as shown in Figure 8-44.
The Create Business Service window is displayed.

2. In the Service Name field, enter any name you wish for the Business Service and click Next, as shown in Figure 8–45.

Figure 8–45  Create Business Service

3. In the displayed Service Type window, select the WSDL option and click the Select WSDL icon, as shown in Figure 8–46.
4. Select the File System folder icon, browse to the iBSE WSDL file and select it from the WSDL location, and then click OK, as shown in Figure 8–47.

5. In the displayed Source pane, click Next, as shown in Figure 8–48.
6. In the displayed Configuration pane, click Finish, as shown in Figure 8–49.

You are returned to the Create Business Service window.

7. In the displayed Type pane, click Next, as shown in Figure 8–50.
8. In the displayed Transport window, you can modify the Endpoint URI field if the hostname and port number varies, and then click Finish, as shown in Figure 8–51.

The Business Service is created and displayed in the External Services pane, as shown in Figure 8–52.
8.3.2.2 Creating a Proxy Service With Pipeline

Perform the following steps to create a Proxy Service with Pipeline:

1. Drag and drop the File Transport component from the Advanced Components pane to the Proxy Services pane, as shown in Figure 8–53.

2. In the Service Name field, enter any name you wish for the Proxy service (for example, JCA_Outbound_PS). By default, Generate Pipeline is selected.

3. Click Next, as shown in Figure 8–54.
4. Select the **Messaging** option, set the Request to XML and Response as **None**, and then click **Next**, as shown in Figure 8–55.

5. Provide the input location in the Endpoint URI field (for example, c:/input) and click **Finish**, as shown in Figure 8–56.
The Proxy service along with the pipeline is created and displayed.

6. Double-click the created Proxy Service (for example: iBSE_Outbound_PS), as shown in Figure 8–57.

7. In the displayed Proxy Service configuration page, select Transport Details and provide the values for Stage and Error Directory, as shown in Figure 8–58.
8. Save and close the Proxy Service configuration page.

9. Double-click the overview.xml file (for example, iBSE_Outbound).

The Proxy service is updated and displayed, as shown in Figure 8–59.

Figure 8–59  Proxy Service

8.3.2.3 Configuring a File Transport Type Business Service

Perform the following steps to create a File Transport Type Business Service:

1. Drag and drop the File Transport component from the Advanced pane to the External Services pane, as shown in Figure 8–60.
2. In the Service Name field, enter any name you wish for the Business Service (for example, FileOut), and click Next, as shown in Figure 8–61.

The Type pane is displayed. The Any XML option is selected by default.

3. Click Next.

The Transport pane appears.

4. Provide the output location in the Endpoint URI field (for example, c:/output) and click Finish, as shown in Figure 8–62.
The File Transport Business service Fileout is created and displayed, as shown in Figure 8–63.

5. Double-click the created Business service Fileout and provide the values for the Prefix and Suffix fields in the Transport Details Tab, as shown in Figure 8–64.
6. Save and close the configuration page, and double-click on overview.xml (for example, iBSE_Outbound).

### 8.3.2.4 Configuring the Routing Rules

Perform the following steps to configure the routing rules:

1. Create a connection between the Pipeline Component (for example, iBSE_Outbound_PSPipeline) and the WSDL based Business Service (for example, iBSE_Outbound_BS), as shown in Figure 8–65.

![Figure 8–65 Pipeline Component](image)

2. Double-click on the **Pipeline** component (for example, iBSE_Outbound_PSPipeline) in the Pipelines/Split Joins pane.

3. Drag and drop the **Pipeline Pair** node from Nodes pane to the area between the Pipeline (for example: iBSE_Outbound_PSPipeline) and RouteNode1, as shown in Figure 8–66.
4. Drag and drop the **Publish** node from the Communication pane to the area beneath Stage1 of the Response Pipeline, as shown in Figure 8–67.

![Figure 8–66 Pipeline Pair Node](image1)

**Figure 8–66 Pipeline Pair Node**

5. Click on the browse icon to the right of the Service field in the right pane of Publish Properties, as shown in Figure 8–68.

![Figure 8–67 Publish Node](image2)

**Figure 8–67 Publish Node**

6. In the displayed Resource Chooser window, select the **Fileout.bix** File Transport Business service and click **OK**, as shown in Figure 8–69.

![Figure 8–68 Browse Icon](image3)

**Figure 8–68 Browse Icon**
You are returned to the Pipeline configuration page.

In the right pane, the selected service is configured in the Publish pane, as shown in Figure 8–70.

7. Save and close the Pipeline configuration page.

8. Double-click the overview.xml file (for example: iBSE_Outbound), and click Save All in the menu bar to save the OSB process, as shown in Figure 8–71.
8.3.3 Deploying the OSB Outbound Process

To deploy the created OSB outbound process and invoke the input XML document, see Section 8.1.3, "Deploying the OSB Outbound Process".

8.4 Configuring a JMS Inbound Process Using JDeveloper (J2CA Configuration)

This section describes how to configure a JMS inbound process to your J.D. Edwards OneWorld system, using Oracle JDeveloper for J2CA configurations.

1. Before you design a JMS process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.5.1, "Generating WSDL for Event Integration" on page 4-34.

2. Start the Oracle JDeveloper and create a Service Bus Application for OSB. For more information, see Section 8.1.1, "Creating a Service Bus Application for OSB" on page 8-2.

3. Create a Third Party Adapter Service Component. For more information, see Section 8.2.2.1, "Configuring a Third-Party Adapter Service Component" on page 8-19.

4. Create a Proxy Service along with the pipeline from the JCA Binding File. For more information, see Section 8.2.2.2, "Creating a Pipeline" on page 8-21.

5. Create a JMS Transport Business Service and perform the following steps:
   a. Drag and drop the JMS Transport component from the Technology Components pane to the External Services pane, as shown in Figure 8–72.
b. In the Service Name field, enter any name you wish for the Business service (for example, JMS_BS) and click **Next**, as shown in **Figure 8–73**.

c. In the displayed Type window, select **Any XML** and then click **Next**.

The Create Business Service dialog is displayed.  

The Transport window is displayed.

d. Modify the appropriate hostname and port number by replacing `DestJndiName` with `QueueIn` in the Endpoint URI field (for example, `jms://localhost:7003/weblogic.jms.XAConnectionFactory/QueueIn`), and then click **Finish**, as shown in **Figure 8–74**.
Figure 8–74  Transport Window

The JMS Business service is created and displayed.

e. Double-click JMS_BS as shown in Figure 8–75.

Figure 8–75  JMS Business Service

f. In the displayed Business Service configuration page, provide the following parameters in the Transport Details tab, as shown in Figure 8–76.
g. In the Destination Type section, select **Queue**.

h. In the Message Type section, select **Text**.

6. Save and close the Configuration page of the business service.

7. Create a connection between **Pipeline** (for example, xxxx_PSPipeline) and **JMS Business Service** (for example, JMS_BS) as shown in figure **Figure 8–77**.

8. Double-click **Pipeline**.

   The Pipeline Configuration page is displayed as shown in **Figure 8–78**.
9. Check that the details are configured properly, and then save and close the Pipeline configuration page. You are returned to the composite editor window.

10. Click **Save All** in the menu bar to save the OSB JMS process, as shown in Figure 8–79.

**Figure 8–79  Save All Icon**

11. Deploy the OSB JMS inbound process. For more information, see Section 8.2.3, "Deploying the OSB Inbound Process" on page 8-26.

12. Once the process is deployed successfully, trigger the event messages. For more information, see Section 4.5.5, "Triggering an Event in J.D. Edwards OneWorld" on page 4-47.

13. Log on to the Oracle WLS console.

14. In the Oracle WLS console, expand **Services**, click **Messaging**, select **JMS Modules**, and then click **jmsResources**.

15. Click the appropriate response link (for example, QueueIn) as shown in Figure 8–80.
16. Click the Monitoring tab, as shown in Figure 8–81.

**Figure 8–81 Monitoring Tab**

17. Select the check box and click the Show Messages button, as shown in Figure 8–82.

**Figure 8–82 Show Messages Button**

18. Click the ID link with the appropriate time and date.

The response document is shown under the Text field.
8.5 Configuring a JMS Outbound Process Using JDeveloper (J2CA Configuration)

This section describes how to configure a JMS outbound process to your J.D. Edwards OneWorld system, using Oracle JDeveloper for J2CA configurations.

1. Before you design a JMS process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.4.1, "Generating WSDL for Request/Response Service" on page 4-8.

2. Start the Oracle JDeveloper and create a Service Bus Application for OSB. For more information, see Section 8.1.1, "Creating a Service Bus Application for OSB" on page 8-2.

3. Create a Third Party Adapter Service Component. For more information, see Section 7.3.2.1, "Configuring a Third Party Adapter Service Component" on page 7-13.

4. Create a WSDL-based Business Service from the JCA Binding File. For more information, see Section 8.1.2.1, "Configuring a Third-Party Adapter Service Component" on page 8-3.

5. Create a JMS Proxy Service with a Pipeline and perform the following steps:
   a. Drag and drop the JMS Transport component from the Technology Components pane to the Proxy Services pane, as shown in Figure 8–83.

   Figure 8–83  JMS Transport Component

      The Create Business Service dialog is displayed.

   b. In the Service Name field, enter any name you wish for the Proxy service (for example, JMS_Proxy). By default, Generate Pipeline is selected.

   c. Click Next, as shown in Figure 8–84.
d. In the displayed Type window, select **Any XML** and then click **Next**.
   The Transport window is displayed.

e. Modify the appropriate hostname and port number by replacing the Endpoint URI field (for example, `jms://localhost:7003/weblogic.jms.XAConnectionFactory/JMS_ProxyRequest`), and then click **Finish**, as shown in **Figure 8–85**.

---

**Figure 8–84 Create Proxy Service Pane**

**Figure 8–85 Transport Window**
The JMS Proxy service along with the pipeline is created and displayed.

f. Double-click the created Proxy Service (for example, JMS_Proxy), as shown in Figure 8–86.

![JMS Proxy Service](image1)

Figure 8–86  JMS Proxy Service

- In the displayed configuration page of the Proxy Service, provide the following parameters in the Transport Details tab, as shown in Figure 8–87.

![JMS Transport Configuration](image2)

Figure 8–87  JMS Transport Configuration

- In the Destination Type section, select **Queue**.
- Select the **Is Response Required** check box.
- In the Response Message Type section, select **Text**.
- In the Response URI field, provide the Endpoint URI used in the JMS Transport Configuration and change Request to Response. For example,
6. Save and close the Configuration page of the Proxy service.

7. Configure the Routing Rules and proceed with the following steps:

   a. Double-click on the pipeline (for example, JMS_ProxyPipeline) in the Pipelines/Split Joins pane. The Pipeline configuration page is displayed.

   b. Drag and drop the **Routing** component from the Route section to the area below the Pipeline (for example, JMS_ProxyPipeline), as shown in Figure 8–88.

   c. In the Pipeline Configuration page, select **Routing** and click the browse icon to the right of the Service field in the Routing Properties pane, as shown in Figure 8–89.

**Figure 8–88  Routing Component**

[Diagram showing routing component]

**Figure 8–89  Browse Service**

[Diagram showing browse service interface]
d. In the displayed Resource Chooser window, select the WSDL-based Business service (for example, xxxxx_B5.bix) and click **OK**.
You are returned to the Pipeline configuration page.

e. Save and Close the Pipeline configuration page.
You are returned to the composite editor window.

f. Click **Save All** in the menu bar to save the OSB JMS process, as shown in Figure 8–90.

**Figure 8–90  Transport Window**

8. Deploy the OSB JMS outbound process. For more information, see Section 8.1.3, “Deploying the OSB Outbound Process” on page 8-15.

9. Once the process is deployed successfully, log on to the Oracle WLS Console.

10. In the Oracle WLS console, expand **Services**, click **Messaging**, select **JMS Modules**, and then click **jmsResources**, as shown in Figure 8–91.

**Figure 8–91  JMS Resources**

11. Click the appropriate request link (for example, JMS_ProxyRequest) as shown in Figure 8–92.
12. Click the Monitoring tab, as shown in Figure 8–93.

Figure 8–93  Monitoring Tab

13. Select the check box and click the Show Messages button, as shown in Figure 8–94.

Figure 8–94  Show Messages Button
14. Click **New**, as shown in Figure 8–95.

**Figure 8–95  JMS Messages**

15. Provide the input payload in the Body field and click **OK**.
16. In the Oracle WLS console, expand **Services**, click **Messaging**, select **JMS Modules**, and then click **jmsResources**.
17. Click the appropriate response link (for example, **JMS_ProxyResponse**).
18. Click the Monitoring tab.
19. Select the check box and click **Show Messages**, as shown in Figure 8–96.

**Figure 8–96  Destination Messages**

20. Click the ID link with the appropriate time and date, as shown in Figure 8–97.
### Configuring an HTTP Outbound Process Using JDeveloper (J2CA Configuration)

This section describes how to configure HTTP Outbound process to your J.D. Edwards OneWorld system, using Oracle JDeveloper for J2CA configurations.

1. Before you design an HTTP Outbound process, you must generate the respective WSDL file using Application Explorer. For more information, see Section 4.4.1, “Generating WSDL for Request/Response Service” on page 4-8.

2. Start the Oracle JDeveloper and create a Service Bus Application for OSB. For more information, see Section 8.1.1, “Creating a Service Bus Application for OSB” on page 8-2.

3. Create a Third Party Adapter Service Component. For more information, see Section 8.1.2.1, “Configuring a Third-Party Adapter Service Component” on page 8-3.

4. Create an HTTP Proxy Service with a Pipeline and perform the following steps:
   a. Drag and drop the HTTP component from the Technology Components pane to the Proxy Services pane, as shown in Figure 8–98.
The Create Proxy Service dialog is displayed.

b. In the Service Name field, enter any name you wish for the Proxy service (for example, HTTP_Proxy). By default, Generate Pipeline is selected.

c. Click **Next**, as shown in Figure 8–99.

**Figure 8–99  Create Proxy Service Pane**

In the displayed Type window, select **Any XML** and then click **Next**.

The Transport window is displayed.

d. In the displayed Type window, select **Any XML** and then click **Next**.

The Transport window is displayed.

e. Leave the default values and then click **Finish**, as shown in Figure 8–100.
The HTTP Proxy service along with the pipeline is created and displayed.

f. Double-click the created pipeline (for example, HTTP_ProxyPipeline) in the Pipelines/Split Joins pane, as shown in Figure 8–101.

The Pipeline Configuration page is displayed.

5. Configure the Routing Rules and proceed with the following steps:

a. Drag and drop the Routing component from the Route section to the area below the Pipeline (for example, HTTP_ProxyPipeline), as shown in Figure 8–102.
b. In the Pipeline Configuration page, select Routing and click the browse icon to the right of the Service field in the Routing Properties pane, as shown in Figure 8–103.

c. In the displayed Resource Chooser window, select the WSDL-based Business service (for example, xxxxx_BS.bix) and click OK.
You are returned to the Pipeline configuration page.

d. Save and Close the Pipeline configuration page.
You are returned to the composite editor window.

e. Click Save All in the menu bar to save the OSB HTTP process, as shown in Figure 8–104.
6. Deploy the OSB HTTP outbound process. For more information, see Section 8.1.3, “Deploying the OSB Outbound Process” on page 8-15.

7. Once the process is deployed successfully, log on to the Service Bus Console.

8. In the Service Bus console, click on the deployed HTTP Outbound project (for example, HTTP_Outbound), as shown in Figure 8–105.

9. Click on the Test OSB Console icon for the created pipeline, as shown in Figure 8–106.
10. In the displayed Test OSB Console page, provide the input XML and click the **Execute** button.

In the displayed Test OSB Console page, the response is received.
This chapter describes key features for the Oracle Application Adapter for J.D. Edwards OneWorld. This chapter contains the following topics:

- Section 9.1, "Exception Filter"
- Section 9.2, "Credential Mapping for Oracle SOA Suite (BPEL, Mediator, or BPM)"
- Section 9.3, "Credential Mapping for Oracle Service Bus (OSB)"

### 9.1 Exception Filter

This section describes how to configure exception filter functionality for the Oracle Application Adapter for J.D. Edwards OneWorld and includes a sample testing scenario.

This section contains the following topic:

- Section 9.1.1, "Configuring the Exception Filter"

The exception filter is supported only for outbound processes that use J2CA configurations. This feature is not supported for BSE configurations and inbound processes that use J2CA configurations.

The exception filter uses the `com.ibi.afjca.oracle.AdapterExceptionFilter` class to filter the generated exceptions. This class filters the exceptions and categorizes them into the following categories:

- `PCRetriableResourceException`
- `PCResourceException`

The following exceptions are represented in the fault policies file:

- `PCRetriableResourceException` - A remote fault.
- `PCResourceException` - A binding fault.

#### 9.1.1 Configuring the Exception Filter

Exception filter configuration consists of the following steps and topics:

1. Section 9.1.1.1, "Generating a WSDL File"
2. Section 9.1.1.2, "Creating a BPEL process With Exception Filter Functionality"
3. Section 9.1.1.3, "Creating Fault Policies and Fault Binding Files"
4. Section 9.1.1.4, "Adjusting for Known Deployment Issues With 12c"
5. Section 9.1.1.5, "Deploying and Testing the BPEL Process With Exception Filter Functionality"

### 9.1.1 Generating a WSDL File

To generate the WSDL file:

1. Open Application Explorer and create a J2CA configuration.
   For more information, see "Creating a Configuration for J2CA" on page 2-3.
2. Create a target for the PeopleSoft adapter and then connect to the target.
   For more information, see "Establishing a Connection (Target) for J.D. Edwards OneWorld" on page 2-5.
3. Generate a WSDL for the appropriate object.
   For more information, see "Generating WSDL (J2CA Configurations Only)" on page 2-11.

### 9.1.1.2 Creating a BPEL process With Exception Filter Functionality

To create a BPEL process with exception filter functionality:

1. Open JDeveloper and create a new SOA application.
   For more information, see Section 4.4.2, "Creating an Empty Composite for SOA" on page 4-9.
2. Create a new SOA project (for example, Exception_Filter).
3. Create a third party adapter service component.
   For more information, see Section 4.4.3.1, "Configuring a Third Party Adapter Service Component" on page 4-11.
   Once the third party adapter service component is created, the WSDL file (with corresponding schemas and JCA file) is imported to the JDeveloper project.
   For more information, see Section 4.4.3, "Defining a BPEL Outbound Process" on page 4-11.
4. Modify the imported JCA file.
   a. Right-click the imported JCA file and select **Open** from the menu, as shown in Figure 9–1.
b. In the `<interaction-spec>` element, add the `ExceptionFilter` property. For example:

```
<interaction-spec className="com.ibi.afjca.cci.IWAFInteractionSpec">
  <property name="FunctionName" value="PROCESS"/>
  <property name="ExceptionFilter" value="com.ibi.afjca.oracle.AdapterExceptionFilter"/>
</interaction-spec>
```

c. Save the modified JCA file.

5. Once the third party adapter service component is created and the JCA file is modified, continue with the remainder of the BPEL process creation.

For more information, see Section 4.4.3, "Defining a BPEL Outbound Process" on page 4-11.

9.1.1.3 Creating Fault Policies and Fault Binding Files

To create fault binding files:

1. Right-click the created SOA project (for example, Exception_Filter), select New, and then click From Gallery, as shown in Figure 9–2.
The New Gallery dialog is displayed. Under the General category, click XML, as shown in Figure 9–3.
2. Select **XML Document** under Items and then click **OK**.

   The Create XML File dialog is displayed, as shown in Figure 9–4.

**Figure 9–4  Create XML File Dialog**

3. In the File Name field, type **fault-bindings.xml** and click **OK**.
4. Add the appropriate fault binding functions in the **fault-bindings.xml** file.

   To view a sample **fault-bindings.xml** file, see "Sample Fault-Bindings.xml File" on page 9-6.

**Note**: The parameter in the `<name>` element is the name of the created BPEL process.
5. Save the fault-bindings.xml file.

Sample Fault-Bindings.xml File

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<faultPolicyBindings version="2.0.1"
xmlns="http://schemas.oracle.com/bpel/faultpolicy"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <component faultPolicy='bpelFaultHandling'>
    <name>BPELProcess1</name>
  </component>
</faultPolicyBindings>
```

Creating Fault Policies Files

To create fault policies files:

1. Right-click the created SOA project (for example, Exception_Filter), select New, and then click From Gallery, as shown in Figure 9–5.

Figure 9–5 Applications Tab

The New Gallery dialog is displayed. Under the SOA Tier category, select Faults, as shown in Figure 9–6.
2. Select Fault Policy Document under Items and then click OK.

3. In the fault-policies.xml tab, select `<bpelx:bindingFault>` from the Fault Name drop-down list and `[retry] default-retry` from the Default Action drop-down list, as shown in Figure 9–7.

4. Click the Actions tab and then double-click `default-retry`. 

---

**Figure 9–6  New Gallery Dialog**

![New Gallery Dialog](image)

**Figure 9–7  Fault-policies.xml Tab**

![Fault-policies.xml Tab](image)
The Retry Properties dialog box is displayed, as shown in Figure 9–8.

**Figure 9–8  Retry Properties Dialog Box**


6. Click OK.

7. Click Add to create another fault handler, as shown in Figure 9–9.

**Figure 9–9  Fault-policies.xml Tab**
8. In the fault-policies.xml tab, select `bpel:remoteFault` from the Fault Name drop-down list and `[abort] default-termination` from the Default Action drop-down list.

9. In the Actions tab, click Add and then select retry, as shown in Figure 9–10.

*Figure 9–10  Actions Tab*

![Actions Tab](image)

The Retry Properties dialog is displayed, as shown in Figure 9–11.

*Figure 9–11  Retry Properties Dialog Box*

![Retry Properties Dialog Box](image)

10. Provide values for the ID, Retry Count, and Retry Interval fields.


12. Click OK.
The created Retry ID will be listed under the Actions tab.

From the Default Action drop-down list, select the newly created Retry ID (for example, remote_retry) as shown in Figure 9–12.

**Figure 9–12  Fault-policies.xml Tab**

13. Click Save All.

14. Click the **Source** tab to verify that the fault policies are added properly, as shown in Figure 9–13.
15. Double-click the Exception Filter project and then click Edit Composite Fault Policies.

The Composite Fault Policies window is displayed. Ensure that the Fault Policy and the fault-bindings are selected properly, as shown in Figure 9–14.
16. Click Save All.

17. Click the Source tab to verify that the `fault-bindings.xml` and `fault-policies.xml` files are added properly, as shown in Figure 9–15.
9.1.1.4 Adjusting for Known Deployment Issues With 12c
For more information on how to adjust for known deployment issues with 12c, see Section 4.4.3.3, "Adjusting for Known Deployment Issues With 12c" on page 4-26.

9.1.1.5 Deploying and Testing the BPEL Process With Exception Filter Functionality
To deploy and test the BPEL process with exception filter functionality:

1. Deploy the created BPEL process.
   For more information, see Section 4.4.4, "Deploying the BPEL Outbound Process" on page 4-28.

2. Simulate a communication error by disconnecting the system (where the servers are running) from the network.

3. Invoke the deployed BPEL process with a valid input.
   For more information, see Section 4.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console" on page 4-31.

4. Select the process ID.
   You can observe the BPEL process being retried or aborted based on the configuration of the fault-policies.xml file.

9.2 Credential Mapping for Oracle SOA Suite (BPEL, Mediator, or BPM)
This section describes how to configure credential mapping functionality for the Oracle Application Adapter for J.D. Edwards OneWorld in a configuration that uses Oracle SOA Suite (BPEL, Mediator, or BPM). A sample testing scenario is also included. This section contains the following topic:

| Section 9.2.1, "Configuring Credential Mapping" |

Credential mapping is supported only for outbound processes that use J2CA configurations. This feature is not supported for BSE configurations and inbound processes that use J2CA configurations.
To pass user credentials to the J2CA resource adapter, create a credential map from the Oracle WebLogic Server user credentials to the EIS user credentials (J.D. Edwards OneWorld adapter). Then associate a credential policy with a BPEL, Mediator, or BPM Web service and invoke the Web service using Oracle WebLogic Server user credentials. These credentials are mapped to the EIS user credentials and then passed to the J2CA container, which uses them to connect with the EIS adapter (J.D. Edwards OneWorld).

### 9.2.1 Configuring Credential Mapping

This section discusses configuring credential mapping, and consists of the following steps and topics:

1. **Deploy the adapter.**
   
   For more information, see Chapter 3, "Oracle WebLogic Server Deployment and Integration".

2. **Associate Oracle WebLogic Server credentials with EIS credentials.**
   
   For more information, see Section 9.2.1.1, "Associating Oracle WebLogic Server Credentials With EIS Credentials" on page 9-14.

3. **Generate a WSDL file.**
   
   For more information, see Section 9.2.1.2, "Generating a WSDL File" on page 9-17.

4. **Create and deploy an outbound process.**
   
   For more information, see Section 9.2.1.3, "Creating and Deploying an Outbound Process" on page 9-17.

5. **Invoke and verify that the EIS credentials have passed.**
   
   For more information, see Section 9.2.1.4, "Verifying the EIS Credentials" on page 9-18.

#### 9.2.1.1 Associating Oracle WebLogic Server Credentials With EIS Credentials

To associate Oracle WebLogic Server credentials with EIS credentials:

1. Log in to the Oracle WebLogic Server Administration Console.

2. In the Domain Structure section in the left pane, click Deployments, as shown in Figure 9–16.

---

**Note:** The J2CA connector is common to all four application adapters (SAP R/3, PeopleSoft, Siebel, and J.D. Edwards OneWorld). If credential mapping is required, then ensure that only one application adapter is used in a particular instance. For example, in one adapter instance only the J.D. Edwards OneWorld application adapter can be used. Credential mapping cannot be configured at the individual adapter level. If you require the use of credential mapping for two adapters, then both adapters must be running in two independent adapter instances.
3. Click the iwafjca resource adapter.
   The Settings for iwafjca page is displayed, as shown in Figure 9–17.

4. Click the Outbound Credential Mappings tab under the Security tab, and then click New.
   The Create a New Security Credential Mapping page is displayed, as shown in Figure 9–18.
Figure 9–18  Create a New Security Credential Mapping Page

5. Select the outbound connection pool.
   For example:
   eis/OracleJCAAdapter/DefaultConnection

6. Click Next.
   The WebLogic Server User page is displayed, as shown in Figure 9–19.

Figure 9–19  WebLogic Server User Page

7. Select Default User, enter a valid Oracle WebLogic Server user name, and then click Next.
   The EIS User Name and Password page is displayed, as shown in Figure 9–20.
8. Enter the user name and password for the EIS and click Finish.

The credentials for an Oracle WebLogic Server user are now mapped with an EIS user (J.D. Edwards OneWorld). The mapping is invoked automatically before invoking the J2CA service.

9.2.1.2 Generating a WSDL File

To generate a WSDL file:

1. Open Application Explorer and create a J2CA configuration.
   For more information, see Section 2.3.2, "Creating a Configuration for J2CA" on page 2-3.

2. Create a target for the J.D. Edwards OneWorld adapter and then connect to the target.
   For more information, see Section 2.4, "Establishing a Connection (Target) for J.D. Edwards OneWorld" on page 2-5.

3. Generate a WSDL for the appropriate object.
   For more information, see Section 2.6, "Generating WSDL (J2CA Configurations Only)" on page 2-11.

9.2.1.3 Creating and Deploying an Outbound Process

This section describes how to configure an outbound process. For demonstration purposes, specific references to the BPEL outbound process are made. However, the same steps apply to Mediator and BPM outbound processes.

For more information about creating a Mediator outbound process, see Chapter 5, "Integration With Mediator Service Components in the Oracle SOA Suite".

For more information about creating a BPM outbound process, see Chapter 6, "Integration With BPM Service Components in the Oracle SOA Suite".

To create a BPEL outbound process, see the following sections:
9.2.1.4 Verifying the EIS Credentials

Invoke the input XML and ensure that the EIS target credentials are overridden with the credentials configured in the WebLogic Administration Console for the Default User as described in this section.

1. Invoke the deployed BPEL outbound process with a valid input.
   
   For more information, see Section 4.4.5, "Invoking the Input XML Document in the Oracle Enterprise Manager Console" on page 4-31.

2. Check the J2CA log files and locate the encrypted password, which shows that the user credentials have been passed to the EIS through Oracle WebLogic Server.
   
   For example:

   ```
   FINEST IWAFManagedConnectionFactory com.ibi.afjca.Util getPasswordCredential(78) InLoop:
   User-iwayqa:Password-ENCR(3109311731831133338233215315332323192322731773172)
   FINEST IWAFManagedConnectionFactory com.ibi.afjca.Util getPasswordCredential(90) Use the system PasswordCredential:
   User-iwayqa:Password-ENCR(3109311731831133338233215315332323192322731773172)
   ```

9.3 Credential Mapping for Oracle Service Bus (OSB)

This section describes how to configure credential mapping functionality for the Oracle Application Adapter for J.D. Edwards OneWorld in a configuration that uses Oracle Service Bus (OSB). A sample testing scenario is also included. This section contains the following topic:

- Section 9.3.1, "Configuring Credential Mapping"

Credential mapping is supported only for outbound processes that use J2CA configurations. This feature is not supported for BSE configurations and inbound processes that use J2CA configurations.

**Note:** The J2CA connector is common to all four application adapters (SAP R/3, PeopleSoft, Siebel, and J.D. Edwards OneWorld). If credential mapping is required, then ensure that only one application adapter is used in a particular instance. For example, in one adapter instance only the J.D. Edwards OneWorld application adapter can be used. Credential mapping cannot be configured at the individual adapter level. If you require the use of credential mapping for two adapters, then both adapters must be running in two independent adapter instances.

To pass user credentials to the iWay J2CA resource adapter, create a credential map from the Oracle WebLogic Server user credentials to the EIS user credentials (J.D. Edwards OneWorld adapter). Then associate a credential policy with a Web service and invoke the Web service using Oracle WebLogic Server user credentials. These credentials are mapped to the EIS user credentials and then passed to the iWay J2CA container, which uses them to connect with the EIS adapter (J.D. Edwards OneWorld).
9.3.1 Configuring Credential Mapping

Configuring credential mapping consists of the following steps and topics:

1. Deploy the adapter.
   For more information, see Chapter 3, "Oracle WebLogic Server Deployment and Integration".

2. Associate Oracle WebLogic Server credentials with EIS credentials.
   For more information, see Section 9.3.1.1, "Associating Oracle WebLogic Server Credentials With EIS Credentials".

   For more information, see Section 9.3.1.2, "Generating a WSDL File".

4. Create an Oracle Service Bus (OSB) outbound process.
   For more information, see Section 9.3.1.3, "Creating an Oracle Service Bus (OSB) Outbound Process" on page 9-22.

9.3.1.1 Associating Oracle WebLogic Server Credentials With EIS Credentials

To associate Oracle WebLogic Server credentials with EIS credentials:

1. Log in to the Oracle WebLogic Server Administration Console.

2. In the Domain Structure section in the left pane, click Deployments, as shown in Figure 9–21.

Figure 9–21  Domain Structure Section

The Deployments page is displayed, as shown in Figure 9–22.
3. Click the **iwafjca** resource adapter.
   
   The Settings for iwafjca page is displayed, as shown in **Figure 9–23**.

**Figure 9–23 Settings for iwafjca Page**

4. Click the **Credential Mappings** tab under the Security tab, and then click **New**.
   
   The Create a New Security Credential Mapping page is displayed, as shown in **Figure 9–24**.
5. Select the outbound connection pool. For example:
   `eis/OracleJCAAdapter/DefaultConnection`

6. Click Next. The WebLogic Server User page is displayed, as shown in Figure 9–25.

7. Select Configured User Name, enter a valid Oracle WebLogic Server user name, and then click Next.
   The EIS User Name and Password page is displayed, as shown in Figure 9–26.
8. Enter the user name and password for the EIS and click **Finish**.

The credentials for an Oracle WebLogic Server user are now mapped with an EIS user (J.D. Edwards OneWorld). The mapping is invoked automatically before invoking the J2CA service.

### 9.3.1.2 Generating a WSDL File

To generate a WSDL file:

1. Set the class path for Application Explorer to integrate with Oracle Service Bus (OSB).
   
   For more information, see Section 7.2.2, "Setting the Class Path for Application Explorer to Integrate With Oracle Service Bus" on page 7-6.

2. Open Application Explorer and create a J2CA configuration.
   
   For more information, see Section 2.3.2, "Creating a Configuration for J2CA" on page 2-3.

3. Create a target for the J.D. Edwards OneWorld adapter and then connect to the target.
   
   For more information, see Section 2.4, "Establishing a Connection (Target) for J.D. Edwards OneWorld" on page 2-5.

4. Generate a WSDL for the appropriate object.
   
   For more information, see Section 4.4.1, "Generating WSDL for Request/Response Service" on page 4-8.

### 9.3.1.3 Creating an Oracle Service Bus (OSB) Outbound Process

For more information on creating an Oracle Service Bus (OSB) outbound process, see Section 8.1.2, "Defining an OSB Outbound Process" on page 8-3.

1. Configure a Service account by right-clicking the OSB Project, selecting **New**, and then clicking **Service Account**, as shown in Figure 9–27.
The Create Service Account pane is displayed, as shown in Figure 9–28.

2. Provide a name for the Service Account and click Finish.
The configuration page of Service Account is displayed.

3. In the Resource Type section, select **Static**.

4. Provide a valid user name and password for the Oracle WebLogic Server, as shown in **Figure 9–29**.

**Figure 9–29  Service Account Configuration Page**

5. Save and close the configuration page.

6. In the composite Editor window, double-click the created WSDL-based Business Service from step 3.

   The configuration page of the WSDL-based Business Service is displayed.

7. Select the Transport Details tab, as shown in **Figure 9–30**.
8. In the JNDI Service Account section, click the Browse icon. The Select Service Account window is displayed.

9. Select the created service account and click OK, as shown in Figure 9–31.

10. Save and close the configuration page, as shown in Figure 9–32.
11. Deploy the OSB process.

For more information, see Section 8.1.3, "Deploying the OSB Outbound Process" on page 8-16.

12. Once the process is deployed successfully, copy and paste a valid input XML file in the input folder you configured, and check to see that the output is received in the configured output location.

13. Check the J2CA log files and locate the encrypted password, which shows that the user credentials have been passed to the EIS through Oracle WebLogic Server.

For example:

```
FINEST IWAFManagedConnectionFactory com.ibi.afjca.Util getPasswordCredential(78) InLoop: User-iwayqa:Password-ENCR(3189319731831132182333215323333232192322731773252)
FINEST IWAFManagedConnectionFactory com.ibi.afjca.Util getPasswordCredential(90) Use the system PasswordCredential: User-iwayqa:Password-ENCR(3109313331831131702333215320132323192322731773236)
```
This chapter explains the limitations and workarounds when connecting to J.D. Edwards OneWorld. It contains the following topics:

- **Section 10.1, "Troubleshooting"**
- **Section 10.2, "BSE Error Messages"**

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

### 10.1 Troubleshooting

This topic provides troubleshooting information for J.D. Edwards OneWorld, and contains the following topics:

- **Section 10.1.1, "Application Explorer"**
- **Section 10.1.2, "J.D. Edwards One World"**
- **Section 10.1.3, "Oracle Adapter J2CA"**

Log file information that can be relevant in troubleshooting can be found in the following locations based on your adapter installation:

- The Oracle Adapter J2CA trace information can be found under the following directory:
  
  ```
  <ADAPTER_HOME>\config\configuration_name\log
  ```

- BSE trace information can be found under the following directory:
  
  ```
  <ORACLE_HOME>\user_projects\domains\base_domain\servers\soa_ server1\stage\ibse\ibse.war\ibselogs
  ```

- The log file for Application Explorer can be found under the following directory:
  
  ```
  <ADAPTER_HOME>\tools\iwae\bin
  ```

### 10.1.1 Application Explorer

This topic discusses the different types of errors that can occur when using Application Explorer.
<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 | You have provided invalid connection information for J.D. Edwards OneWorld or the wrong JAR file is in the lib directory. |
| J.D. Edwards OneWorld does not appear in the Application Explorer Adapter node list.  
  Logon failure error at run-time. | Ensure that the J.D. Edwards OneWorld JAR files, are added to the lib directory. |
| The following exception occurs when you start Application Explorer by activating ae.bat (not iaexplorer.exe):  
  java.lang.ClassNotFoundException: org.bouncycastle.jce.provider.BouncyCastleProvider | 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. |
| This is a benign exception. It does not affect adapter functionality. Download BouncyCastle files from: ftp://ftp.bouncycastle.org/pub |
Unable to start Application Explorer in a Solaris environment. The following exception is thrown in the console:

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)

Could not create the connection factory.

JAVACMD is not set on the user system. Before starting Application Explorer, export JAVACMD as follows:

JAVACMD=/<jdk_home>/bin/java, where <jdk_home> is the directory where JDK is installed on your system.

### 10.1.2 J.D. Edwards One World

<table>
<thead>
<tr>
<th>Error</th>
<th>Cause</th>
<th>Solution</th>
</tr>
</thead>
</table>
| Action code invalid | In the Sales Order request, the Action code appears as "H," an invalid action code. | Use:  
  - "I" for inquiry.  
  - "C" for change.  
  - "D" for delete.  
  - "A" to add a new record. |
| Invalid address number. | The address number does not exist in the Address Book Master file (F0101). | Enter an address number using the Address Book Revisions program (PO1051). Ensure that the number entered is correct. |
| Record invalid | The record being processed either already exists for an ADD function or does not exist for an INQUIRY, CHANGE, or DELETE function. | 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. |
| Item Branch record does not exist. | An Item Branch record (F4102) does not exist for this item in the Branch/Plant specified. | Correct the Branch or enter an Item Branch record for this item in Branch Plant Item Information (P41026). |
| &1 does not match any of the valid values. | The &1 does not match any of the valid values specified in the Data Dictionary for this field. | Enter a valid value. |
10.2 BSE Error Messages

This section discusses the different types of errors that can occur when processing Web services through BSE.

This section contains the following topics:

- Section 10.2.1, "General Error Handling in BSE"
- Section 10.2.2, "Adapter-Specific Error Handling"

10.2.1 General Error Handling in BSE

BSE serves as both a SOAP gateway into the adapter framework and as the engine for some of the adapters. In both design time and run-time, various conditions can cause errors in BSE when Web services that use adapters run. Some of these conditions and resulting errors are exposed the same way, regardless of the specific adapter; others are exposed differently, based on the adapter being used. This topic explains what you can expect when you encounter some of the more common error conditions on an adapter-specific basis. Usually the SOAP gateway (agent) inside BSE passes a SOAP request message to the adapter required for the Web service. If an error occurs, 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:

```
Date out of range. The Last Service Date and the Inspection Date must be within the range of the effective dates of the Service Contract.

Change the date to be greater than or equal to the beginning effective date and less than or equal to the ending effective date of the Service Contract.
```

```
Jde.net timeout exception
Net timeout is set to a wrong value

Verify that net timeout is set to 180 at jde.ini of [NETWORK QUEUE SETTINGS], for example
JDENETTimeout=180
```

```
Cannot connect to EnterpriseOne Version 8.10
Missing required library files

Kernel.jar and Connector.jar are required for version B7333.
jdeutil.jar and log4j.jar are required for EnterpriseOne Version 8.10, in addition to Kernel.jar and Connector.jar.
```

### Error Cause Solution

<table>
<thead>
<tr>
<th>Error</th>
<th>Cause</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date out of range.</td>
<td>The Last Service Date and the Inspection Date must be within the range of the effective dates of the Service Contract.</td>
<td>Change the date to be greater than or equal to the beginning effective date and less than or equal to the ending effective date of the Service Contract.</td>
</tr>
<tr>
<td>Jde.net timeout exception</td>
<td>Net timeout is set to a wrong value</td>
<td>Verify that net timeout is set to 180 at jde.ini of [NETWORK QUEUE SETTINGS], for example JDENETTimeout=180</td>
</tr>
<tr>
<td>Cannot connect to EnterpriseOne Version 8.10</td>
<td>Missing required library files</td>
<td>Kernel.jar and Connector.jar are required for version B7333. jdeutil.jar and log4j.jar are required for EnterpriseOne Version 8.10, in addition to Kernel.jar and Connector.jar.</td>
</tr>
</tbody>
</table>

10.1.3 Oracle Adapter J2CA

<table>
<thead>
<tr>
<th>Error</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>In Application Explorer, the following error message appears when you attempt to connect to an Oracle Adapter J2CA configuration: Could not initialize JCA</td>
<td>In the Details tab in the right pane, ensure that the directory specified in the Home field points to the correct directory, for example: &lt;ADAPTER_HOME&gt;\tools\iwae\bin........|</td>
</tr>
</tbody>
</table>

10.2 BSE Error Messages

This section discusses the different types of errors that can occur when processing Web services through BSE.

This section contains the following topics:

- Section 10.2.1, "General Error Handling in BSE"
- Section 10.2.2, "Adapter-Specific Error Handling"
In this example, BSE did not receive an element in the SOAP request message that is mandatory for the WSDL for this Web service.

10.2.2 Adapter-Specific Error Handling

This section contains the following topics:

- Section 10.2.2.1, "Invalid SOAP Request"
- Section 10.2.2.2, "Empty Result From Oracle WebLogic Server Adapter Request"
- Section 10.2.2.3, "Error Logging In"
- Section 10.2.2.4, "Empty Result From Oracle WebLogic Server Adapter Request"
- Section 10.2.2.4, "Empty Result From Oracle WebLogic Server Adapter Request"

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.

10.2.2.1 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:Client</faultcode>
<faultstring>Parameter node is missing</faultstring>
</SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
10.2.2.2 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.

```
<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
     xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
     xmlns:xsd="http://www.w3.org/1999/XMLSchema">
     <SOAP-ENV:Body>
         <m:RunDBQueryResponse xmlns:m="urn:schemas-iwaysoftware-com:iwse"
             xmlns="urn:schemas-iwaysoftware-com:iwse"
             cid="2A3CB42703EB20203F91951B89F3CA5F">
             <RunDBQueryResult run="1" />  
         </m:RunDBQueryResponse>
     </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Note: The condition for this adapter does not yield a SOAP fault.

10.2.2.3 Error Logging In

If Oracle WebLogic Server Adapter executes an invalid SOAP log in request, then the following SOAP response is generated.

```
[2004-07-19T16:28:56:718Z] DEBUG (SOAP1) W.SOAP1.2: in XDSOAPHTTPWorker agentName is [XDSOAPRouter]
     xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
     <SOAP-ENV:Header>
         <m:ibsinfo xmlns:m="urn:schemas-iwaysoftware-com:iwse" xmlns="urn:schemas-iwaysoftware-com:iwse"
             cid="2A3CB42703EB20203F91951B89F3CA5F">
             <RunDBQueryResult run="1" />  
         </m:ibsinfo>
     </SOAP-ENV:Header>
     <m:RunDBQueryResponse xmlns:m="urn:schemas-iwaysoftware-com:iwse"
             xmlns="urn:schemas-iwaysoftware-com:iwse"
             cid="2A3CB42703EB20203F91951B89F3CA5F">
             <RunDBQueryResult run="1" />  
         </m:RunDBQueryResponse>
     </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Note: The condition for this adapter does not yield a SOAP fault.
10.2.2.4 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.
10.2.2.5 Invalid Call Method

If an invalid call is made to Oracle WebLogic Server Adapter, then the following SOAP response is generated.

```
[2004-07-19T16:24:34:859Z] DEBUG (SOAP1) W.SOAP1.2: POST received
[2004-07-19T16:24:34:859Z] DEBUG (SOAP1) W.SOAP1.2: in XDSOAPHTTPWorker agentName is [XDSOAPRouter]
[2004-07-19T16:24:34:859Z] DEBUG (SOAP1) W.SOAP1.2: before parse:
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
<SOAP-ENV:Body>
<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>
</SOAP-ENV:Body>
```

```xml
<SOAP-ENV:Envelope>
  <SOAP-ENV:Body>
    <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:callMethod>
      </m:jdeRequest>
    </m:GetEffectiveAddress>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
<SOAPAction agentName="XDSOAPRouter"
cid='4C0AD8398CB7A5B4DED18057D963AA44'>B0100033.GetEffectiveAddressRequest#test##</SOAPAction>
</SOAP-ENV:Envelope>


[2004-07-19T16:24:35:031Z] DEBUG (SOAP1) W.SOAP1.2: input:


[2004-07-19T16:24:36:781Z] INFO (manager) MGR00X02: Removing active worker: W.SOAP1.2

[2004-07-19T16:24:36:781Z] DEBUG (SOAP1) W.SOAP1.2: doing docTran, docVal, listTran for agent (1)

[2004-07-19T16:24:36:781Z] DEBUG (SOAP1) W.SOAP1.2: sendToAll reply to XDReply: [protocol=http */null]


xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
</SOAP-ENV:Envelope>

[2004-07-19T16:24:36:796Z] DEBUG (SOAP1) W.SOAP1.2: writeString: HTTP/1.0


[2004-07-19T16:24:36:796Z] DEBUG (SOAP1) W.SOAP1.2: writeString: Content-Type:


J.D. Edwards OneWorld enables you to specify inbound functionality for Master Business Functions (MBF).

This chapter describe 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>
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.

This section contains the following topic:

- Section A.2.1, "Running GenJava"

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.

A.2.1 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 4 (Minor Business Functions):

```
java -cp <path> com.jdedwards.jdegen.JDEGenJava -appadapter <appadapter_name> -businessfunctions <business_functions> -output <output_directory>
```
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 following wrapper files in Java for all of the business functions that are mentioned in the script file:

- CALLBSFNIInterop.jar
- CALLBSFNIInteropDoc.jar
- CALLBSFN.xml

**Note:** If there is an error while these wrapper files are generated, then ensure that the CLASSPATH is set correctly.

4. Copy the wrapper files to the repository directory.

Ensure that the following files are added to the CLASSPATH of the system where you are running GenJava:

- `base_JAR.jar`
- `jdeNet_JAR.jar`
- `system_JAR.jar`
- `connector.jar`
- `xalan.jar`
- `xerces.jar`

These files are located in the following directory:

```
<JDE_EnterpriseOne_Home>\System\classes\`
```
In addition, ensure that the `bin` directory of your JDK installation is included in the Java Path. For example:

```
PATH
x:\E900\system\jdk\bin
```

For more information on using GenJava, see the *J.D. Edwards EnterpriseOne Tools 8.98 Connectors Guide*.

### A.3 Triggering J.D. Edwards OneWorld Events

This section contains the following topics:

- **Section A.3.1, "Starting the Outbound Scheduler Subsystem Process (R00460)"
- **Section A.3.2, "Verifying the Subsystem Process"
- **Section A.3.3, "Configuring P4210 (Sales Order) to Trigger an Event"
- **Section A.3.4, "Verifying the Configuration Steps"

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**

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.

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.
A.3.1 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.

A.3.2 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.
Triggering J.D. Edwards OneWorld Events

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.

**A.3.3 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.

### A.3.4 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.

![Work With Processing Log Window](image)

**Figure A–20 Data Display Table**

<table>
<thead>
<tr>
<th>User ID</th>
<th>Batch Number</th>
<th>Transaction Number</th>
<th>Line Number</th>
<th>Trans</th>
<th>Or Ty</th>
<th>Seq</th>
<th>UDE Name</th>
<th>Version</th>
<th>S P</th>
</tr>
</thead>
<tbody>
<tr>
<td>JDE</td>
<td>15148</td>
<td>103323</td>
<td>1.000, JDE000UT</td>
<td>50</td>
<td>JDE000UT</td>
<td>1.00</td>
<td>JDE000UT</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>JDE</td>
<td>15148</td>
<td>103324</td>
<td>1.000, JDE000UT</td>
<td>50</td>
<td>JDE000UT</td>
<td>1.00</td>
<td>JDE000UT</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>JDE</td>
<td>15150</td>
<td>103325</td>
<td>1.000, JDE000UT</td>
<td>50</td>
<td>JDE000UT</td>
<td>1.00</td>
<td>JDE000UT</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>JDE</td>
<td>15151</td>
<td>103327</td>
<td>1.000, JDE000UT</td>
<td>50</td>
<td>JDE000UT</td>
<td>1.00</td>
<td>JDE000UT</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>JDE</td>
<td>15152</td>
<td>103328</td>
<td>1.000, JDE000UT</td>
<td>50</td>
<td>JDE000UT</td>
<td>1.00</td>
<td>JDE000UT</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>JDE</td>
<td>15153</td>
<td>103329</td>
<td>1.000, JDE000UT</td>
<td>50</td>
<td>JDE000UT</td>
<td>1.00</td>
<td>JDE000UT</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>JDE</td>
<td>15154</td>
<td>103330</td>
<td>1.000, JDE000UT</td>
<td>50</td>
<td>JDE000UT</td>
<td>1.00</td>
<td>JDE000UT</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>JDE</td>
<td>15155</td>
<td>103331</td>
<td>1.000, JDE000UT</td>
<td>50</td>
<td>JDE000UT</td>
<td>1.00</td>
<td>JDE000UT</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>JDE</td>
<td>15156</td>
<td>103332</td>
<td>1.000, JDE000UT</td>
<td>50</td>
<td>JDE000UT</td>
<td>1.00</td>
<td>JDE000UT</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>JDE</td>
<td>15157</td>
<td>103333</td>
<td>1.000, JDE000UT</td>
<td>50</td>
<td>JDE000UT</td>
<td>1.00</td>
<td>JDE000UT</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>JDE</td>
<td>15158</td>
<td>103334</td>
<td>1.000, JDE000UT</td>
<td>50</td>
<td>JDE000UT</td>
<td>1.00</td>
<td>JDE000UT</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>JDE</td>
<td>15159</td>
<td>103335</td>
<td>1.000, JDE000UT</td>
<td>50</td>
<td>JDE000UT</td>
<td>1.00</td>
<td>JDE000UT</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>JDE</td>
<td>15163</td>
<td>103452</td>
<td>1.000, JDE000UT</td>
<td>50</td>
<td>JDE000UT</td>
<td>1.00</td>
<td>JDE000UT</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>JDE</td>
<td>15164</td>
<td>103453</td>
<td>1.000, JDE000UT</td>
<td>50</td>
<td>JDE000UT</td>
<td>1.00</td>
<td>JDE000UT</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>JDE</td>
<td>15165</td>
<td>103454</td>
<td>1.000, JDE000UT</td>
<td>50</td>
<td>JDE000UT</td>
<td>1.00</td>
<td>JDE000UT</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>JDE</td>
<td>15166</td>
<td>103455</td>
<td>1.000, JDE000UT</td>
<td>50</td>
<td>JDE000UT</td>
<td>1.00</td>
<td>JDE000UT</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

3. Search for the corresponding transaction.

The iwoevent.log file is created in the following directory:

```
\iwJDE812\JDE\edwards\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:
========================
<? xml version="1.0" encoding="UTF-8"?><jde><request><connection><dsn /></user
/><password />><sp><proc>JDESOOUT</proc><data><ediUserId>JDE
</ediUserId><ediBatchNumber>15205
```

---

**Figure A–19 Work With Processing Log Window**

![Work With Processing Log Window](image)
**adapter**
Provides universal connectivity by enabling an electronic interface to be accommodated (without loss of function) to another electronic interface.

**agent**
Supports service protocols in listeners and documents.

**business service**
Also known as a Web service. A Web service is a self-contained, modularized function that can be published and accessed across a network using open standards. It is the implementation of an interface by a component and is an executable entity.

**channel**
Represents configured connections to particular instances of back-end systems. A channel binds one or more event ports to a particular listener managed by an adapter.

**listener**
A component that accepts requests from client applications.

**port**
Associates a particular business object exposed by the adapter with a particular disposition. A disposition is a URL that defines the protocol and location of the event data. The port defines the end point of the event consumption.
access methods, 2-28
action codes, 10-3
adapter configuration requirements, 5-1
upgrading, 3-3
adapter exceptions, 10-2 to 10-5
adapter integration
BPEL Process Manager and, 4-1
InterConnect and, 5-1
adapter types, 1-1, 1-6
resource, 3-1 to 3-3
adapters, 1-1 to 1-8
configuring, 3-3
deploying, 1-5, 3-1 to 3-3, 4-2
integrating, 5-1
integrating with BPEL Process Manager, 6-1
troubleshooting, 10-1 to 10-11
Add Channel dialog box, 4-34
Add Target dialog box, 2-6
Address Book Master file, 10-3
Address Book Revisions program, 10-3
address numbers, 10-3
agents, 1-3
alias section of iwoevent.cfg file, 2-29
aliases, 2-30
application adapter integration
BPEL Process Manager and, 4-1
InterConnect and, 5-1
application adapters, 1-1 to 1-8
configuring, 3-3
deploying, 1-5, 3-1 to 3-3, 4-2
integrating, 5-1
integrating with BPEL Process Manager, 6-1
troubleshooting, 10-1 to 10-11
Application Explorer, 1-5, 2-1, 2-15, 4-7, 4-33
application systems and, 2-5
channels and, 2-15, 2-28
deployment and, 10-2
event ports and, 2-15
J2CA configuration and, 3-3
OracleAS Adapter J2CA and, 10-4
schemas and, 2-10
testing and, 10-2
troubleshooting, 10-2
WSDL files and, 4-34
Application parameter, 4-36
application systems, 2-15
Application Explorer and, 2-5
supported, 2-5
batch processes, 1-3
batch.log file, 2-28
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, 10-3
BSE (OracleAS Adapter Business Services Engine), 1-5
configuring, 2-2
troubleshooting, 10-4 to 10-11
BSE deployment, 1-5
BSE URL field, 2-3
business events, 1-2
business functions, 1-8
creating schemas for, 2-10
business services
creating, 2-13 to 2-14
testing, 2-14
CCI calls, 3-1
channel configuration parameters, 4-36
Application, 4-36
Host, 4-35
Is Keep Alive, 4-35
Is Length Prefix, 4-35
Is XML, 4-35
JDE Environment, 4-36
Port Number, 4-35
Schema style, 4-37
Server IP address, 4-36
Server Port, 4-36
User id, 4-36
User password, 4-36
channels, 2-15
  creating, 2-15, 2-16, 4-34 to 4-37
  debugging and, 4-37
  deleting, 2-28
  editing, 2-28
  ports and, 2-15
  starting, 4-37
  stopping, 4-37
  testing and, 4-37
Channels node, 4-34
Channels. See also listeners
  common section of iwoevent.cfg file, 2-29
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
  connecting to, 2-4
  OracleAS Adapter Business Services Engine (BSE), 2-2 to 2-4
  overwriting, 3-4
  Configurations node, 2-2 to 2-3
  configuring adapters, 3-3
  connecting to J.D. Edwards OneWorld, 2-5 to 2-8, 10-2
  connecting to OracleAS Adapter J2CA, 10-4
  Connection dialog box, 2-8
  connection information, 2-29, 10-2
  connection parameters, 10-2
    Application, 2-8
    Hostname, 2-3
    JDE environment, 2-8
    Server IP address, 2-8
    Server Port, 2-8
    User id, 2-8
    User password, 2-8, 4-36
connections
  creating, 2-5 to 2-8
  troubleshooting, 10-2 to 10-4
Connector
  deploying to Oracle Application Server, 3-1
  connector factories, 3-3
  connector factory objects, 3-3
    multiple, 3-5
  creating channels, 4-34 to 4-37
  creating repository projects, 2-2 to 2-4
  creating schemas, 2-10

D
data dictionaries, 10-3
Data Export Control table, 1-3
  data queues, 1-3
  Data Source Name (DSN), 2-30
database connections
  opening, 3-4
database tables, 1-3
databases
  connecting to, 3-4
  Oracle, 3-4
date ranges, 10-4
deploying adapters, 1-5, 3-1 to 3-3, 4-2
deployments
  BSE, 1-5
  Description field, 2-6, 2-13
  design time, 4-34
    configuring, 4-7, 4-33, 5-2
document types
  request, 1-3
  response, 1-3
  XML, 1-3
DSN (Data Source Name), 2-30

E
  Eclipse. See JDeveloper
eering targets, 2-9
EIS (enterprise information systems), 1-3
EJB (Enterprise Java Beans), 3-1
  Enterprise Connector for J2EE Connector Architecture (J2CA), 1-5, 2-2 to ??
  enterprise information systems (EIS), 1-3
  Enterprise Java Beans (EJB), 3-1
error messages, 10-2 to 10-11
target systems and, 10-5
event adapters, 2-15, 2-28
event data
  receiving, 4-33
event integration, 4-33
event listeners, 2-28 to 2-29
event messages, 4-34
events, 1-2, 2-15
    configuring, 2-15
  Existing Service Names list, 2-13
  Export WSDL dialog box, 4-8
  external listeners, 1-3

F
  fault code elements, 10-5
  fault string elements, 10-5
  flat files, 1-3
  functional modeling, 5-1

G
generating schemas, 2-10
GenJava program, 2-7
GenJava repository, 2-7

H
  Host parameter, 4-35
  Hostname parameter, 2-3
I

inbound J2CA services, 4-34
inbound processing, 1-4, A-1
Inspection date, 10-4
installation directories, 3-4
installation requirements
integration and, 5-1
integration requirements, 5-1
integration scenarios, 5-1
InterConnect
adapter integration and, 5-1
internal listeners, 1-3
interoperability framework, 1-3
inbound processing, 1-4
outbound processing, 1-4
invalid address numbers, 10-3
invalid records, 10-3
IP addresses, 2-8, 2-30
Is Keep Alive parameter, 4-35
Is Length Prefix parameter, 4-35
Is XML parameter, 4-35
Item Branch records, 10-3
IWayConfig parameter, 3-4
IWayHome parameter, 3-4
IWayRepoPassword parameter, 3-4
IWayRepoURL parameter, 3-4
IWayRepoUser parameter, 3-4
IWOEvent listener exit, 2-28
iwoevent.cfg file, 2-29 to 2-30
iwoevent.log file, 2-28

J

J2CA (Enterprise Connector for J2EE Connector Architecture), 1-5, 2-2 to ??
OracleAS Adapter and, 3-1
J2CA configuration
Application Explorer and, 3-3
J2CA resource adapters, 1-1, 3-1
J2CA services, 4-8 to 4-34
JAR files, 4-37, 10-2
Java files, 2-7
Java program clients, 3-1
J.D. Edwards OneWorld
connecting to, 2-5 to 2-8, 10-2
J.D. Edwards OneWorld Event Listener, 2-28 to 2-30
J.D. Edwards OneWorld ThinNet API, 1-3 to 1-4, 1-8, 2-10
JDE Environment parameter, 2-8, 4-36
JDE OneWorld dialog box, 2-6
jdeTransactionName, 2-30
JDEdwards node, 4-34
JDeveloper, 4-7, 6-1

L

Last Service date, 10-4
License and Method dialog box, 2-13
License field, 2-13
licenses, 2-13
list of nodes, 10-2
listener configuration files, 2-29, 2-30
listener exits, 2-28
listener types, 1-3
listeners, 1-3, 2-28 to 2-30
listeners. See also channels
log files, 10-1
log levels
overwriting, 3-4
Loglevel parameter, 3-4
logon parameters, 10-2
Logon tab, 2-7

M

managed connector factories, 3-3
managed connector factory objects, 3-3
multiple, 3-5
ManagedConnectionFactory parameter, 3-4
Master Business Functions (MBF), 1-2 to 1-5, 2-5, A-1
executing, 2-10
MBF (Master Business Functions), 1-2 to 1-5, 2-5, A-1
executing, 2-10
Mediator Inbound Process, 5-11
Mediator Outbound Process, 5-2
messages, 1-2 to 1-3
event, 4-34
Method Name field, 2-13

N

Name field, 2-6
New Configuration dialog box, 2-3 to 2-4
Node list, 10-2
nodes, 2-2 to 2-3, 2-8
Channels, 4-34
Configurations, 2-2 to 2-3
JDEdwards, 4-34

O

OneWorld Event Listener, 2-28 to 2-30
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's Unified Method (OUM), ix
OracleAS Adapter
installation directory and, 3-4
J2CA and, 3-1
OracleAS Adapter Application Explorer. See Application Explorer
OracleAS Adapter Business Services Engine (BSE), 1-5
configuring, 2-2
troubleshooting, 10-4 to 10-11
OracleAS Adapter for J.D. Edwards OneWorld
BPEL Process Manager and, 4-1
configuring, 2-1
generating, 4-8, 4-37
WSDL (Web Services Description Language), 4-34
generating, 4-8, 4-37
WSDL documents, 4-1
WSDL files, 4-1, 6-1
   Application Explorer and, 4-34
   creating, 4-34

X

XDjdeOutboundAgent, 2-28
XML documents, 1-3
XML format, 1-3
XML messages, 1-2 to 1-3
XML schemas, 1-3
   creating, 2-10
XMLInterop parameter, 4-36

Z

Z files, 2-28