# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preface</td>
<td>v</td>
</tr>
<tr>
<td>Audience</td>
<td>v</td>
</tr>
<tr>
<td>Documentation Accessibility</td>
<td>v</td>
</tr>
<tr>
<td>Related Documents</td>
<td>vi</td>
</tr>
<tr>
<td>Conventions</td>
<td>vi</td>
</tr>
</tbody>
</table>

## 1 General Best Practices

- Understanding Managed Connections                                     | 1-1  |
- Configuring Oracle Application Adapters in a High Availability Cluster Environment | 1-3  |
  - Prerequisites                                                        | 1-4  |
  - Deploying the J2CA Connector Application to a Clustered Environment | 1-4  |
  - Configuring the HA File Adapter for an Outbound Process             | 1-10 |
  - Configuring the Outbound Process                                     | 1-15 |
  - Configuring Connection Settings for the Inbound Process             | 1-16 |
  - Configuring the Inbound Process                                      | 1-17 |
  - Singleton Testing                                                    | 1-17 |
- Preferred Repository Type                                              | 1-19 |

## 2 Oracle Application Adapter for SAP R/3 Best Practices

- Understanding Connection Pools                                        | 2-1  |
- Security                                                              | 2-2  |
- Load Balancing                                                       | 2-2  |
- Encoding                                                             | 2-3  |

## 3 Oracle Application Adapter for Siebel Best Practices

- Best Practices for Siebel                                            | 3-1  |
- Configuring Connection Pooling for Siebel                             | 3-2  |
4 Oracle Application Adapter for PeopleSoft Best Practices
PeopleSoft Events (Inbound Processing) ................................................................................................. 4-1

5 Oracle Application Adapter for J.D. Edwards OneWorld Best Practices
Connection Pooling and Load Balancing .................................................................................................. 5-1
Security ....................................................................................................................................................... 5-2
Multiple Connections and Multiple Users ................................................................................................. 5-2
Unicode Support ........................................................................................................................................ 5-2
Delivery Confirmation and Error Messages ................................................................................................. 5-2

Index
Preface


Audience

The Oracle Fusion Middleware Application Adapter Best Practices Guide for Oracle WebLogic Server is intended for system administrators and application developers who are using Oracle Application Adapters for Oracle WebLogic Server.

Documentation Accessibility

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

Access to Oracle Support

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

For more information, see the following documents in the Oracle Enterprise Repository 11g Release 1 (11.1.1) documentation set:

- Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server
- Oracle Fusion Middleware Application Adapter Upgrade Guide for Oracle WebLogic Server
- Oracle Fusion Middleware Application Adapter for SAP R/3 User’s Guide for Oracle WebLogic Server
- Oracle Fusion Middleware Application Adapter for PeopleSoft User’s 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>italic</td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td>monospace</td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
This chapter lists and describes general best practices that are common to all Oracle Application Adapters for Oracle WebLogic Server.

This chapter discusses the following topics:

- Understanding Managed Connections
- Configuring Oracle Application Adapters in a High Availability Cluster Environment
- Preferred Repository Type
- Generating Input XML Documents

1.1 Understanding Managed Connections

Oracle WebLogic Server (WLS) creates and maintains a pool of connections to the iWay J2CA resource adapter. These are called "managed connections". Each managed connection maintains a distinct set of connections to a back-end Enterprise Information System (EIS) systems (exposed as an adapter target). The set is initially empty and becomes populated only as specific adapter target connections are requested by an executing SOA process that is using the managed connection.

A managed connection will maintain only a single connection to any given adapter target and will reuse it each time a connection to that target is requested. When WLS destroys a managed connection, the managed connection will close all its internal EIS connections.

By way of illustration, the following steps detail a scenario in which a SOA process containing references to multiple iWay adapter targets is used.

1. SOA process requests a managed connection to the iWay resource adapter ("eis/OracleJCAAdapter/DefaultConnection") and receives a new, empty, managed connection called "Managed1".

2. SOA process invokes a service with adapter target MySAP/SAP1 using Managed1.

3. Managed1 creates a new MySAP adapter instance, activates it for target SAP1, and adds it to its internal EIS connection set.

4. Service is invoked, response returned, and SOA process continues.

5. SOA process invokes a service with adapter target MySAP/SAP2 using Managed1.

6. Managed1 creates a new MySAP adapter instance, activates it for target SAP2, and adds it to its internal EIS connection set.
Understanding Managed Connections

7. Service is invoked, response returned, and SOA process continues.

8. SOA process invokes a service with adapter target JDEdwards/JDE1 using Managed1.

9. Managed1 creates a new JDEdwards adapter instance, activates it for target JDE1, and adds it to its internal EIS connection set.

10. SOA process terminates. WLS returns Managed1 to the connection pool.

After termination of the SOA process, Managed1 is available to be used by some other SOA process. Its internal EIS connections (SAP1, SAP2, JDE1) remain open. They will be reused the next time a process requests connections to those targets from Managed1. They will be closed only when WLS destroys Managed1.

You can monitor the managed connection pool in the WLS admin console, as shown in the following image.

The "current connections" column is the number of managed connections that are currently in use, presumably by executing SOA processes. This number should go up when a SOA process begins execution and should go down after a period of inactivity.

When a SOA process needs a managed connection, it will try to retrieve one from the connection pool. If a managed connection is not available, and the number of current connections does not yet exceed the maximum size of the pool, it will create a new one. If the maximum size of the pool has been reached and no free connection is available, then a resource availability error will be thrown. When a process is finished, the connection is returned to the pool.

It is important to remember that these numbers represent connections to the iWay resource adapter and not to any specific EIS. A single iWay managed connection may contain any number of EIS connections depending upon its usage history. Consequently, there is no way to introspect the number of EIS connections in use by the iWay resource adapter.

The maximum size of the managed connection pool, the minimum size, and other pool parameters can be set in the RAR deployment descriptor ("weblogic-ra.xml") and viewed in the admin console, as shown in the following image.
There are many pool parameters, but because they control the pool of connections to the iWay resource adapter and not any specific EIS many of these are immaterial. Setting an initial pool size, for example, accomplishes nothing except creating a set of empty resource adapter connections: iWay managed connections with no internal EIS connections. There is no reason to do this. For similar reasons, the “capacity increment” parameter is also not particularly useful.

You have to set the initial capacity to zero always to work with the iWay adapters.

---

**Note:** You cannot set anything other than the value of zero.

---

Some settings will have implicit effects for all adapter targets. For example, setting a maximum size for the managed connection pool will implicitly establish the maximum number of connections that can be made to any single target since each managed connection can hold at most one connection to a unique adapter target. Similarly, defining the lifetime of a managed connection will implicitly define the lifetime for all EIS connections created by the connection. However, there is no way to control connection pooling for specific EIS targets using the iWay "universal" resource adapter.

### 1.2 Configuring Oracle Application Adapters in a High Availability Cluster Environment

Application adapters integrate Oracle WebLogic Server with various packaged applications, such as SAP R/3 and Siebel. These adapters include Oracle Application Adapter for PeopleSoft, Oracle Application Adapter for SAP R/3, Oracle Application Adapter for Siebel, and Oracle Application Adapter for J.D. Edwards OneWorld.

This section describes the steps required to successfully deploy the 11g Release 1 (11.1.1) Oracle Application Adapters in a cluster environment (in all available SOA nodes, which in this case is two machines) and working with inbound and outbound processes.
1.2.1 Prerequisites

Before continuing, ensure that the following prerequisites are available:

1. Two machines with the High Availability (HA) cluster configuration completed successfully. For more information about configuring the HA cluster, see:
   
   http://download.oracle.com/docs/cd/E12839_01/core.1111/e10106/ha_soa.htm#CHDDAHEC

2. Install Oracle Application Adapters 11g Release 1 (11.1.1) on both machines.

3. Configure a J2CA configuration as a database repository on the first machine.

4. Perform the required changes to the ra.xml and weblogic-ra.xml files before deployment.

1.2.2 Deploying the J2CA Connector Application to a Clustered Environment

To deploy the J2CA Connector Application:

1. Start the Oracle WebLogic Server for the Oracle WebLogic Server domain that has been configured.

2. Start the SOA_Server1 and SOA_Server2 for both machines.

3. Open the Oracle WebLogic Server Administration Console in a Web browser by entering the following URL:

   http://hostname:port/console

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

   The Oracle WebLogic Server Administration Console logon page is displayed.

4. Log in to the Oracle WebLogic Server Administrative Console using an account that has administrator privileges.

   The Oracle WebLogic Server Administration Console home page is displayed.
5. In the Domain Structure section in the left pane, click Deployments.
   The Deployments page is displayed.

6. Click Lock & Edit in the Change Center.

7. Click Install.
   The Install Application Assistant page is displayed.
8. Browse to the following directory:
   C:\oracle\product\fmw\soa\soa\thirdparty\ApplicationAdapters\iwafjca.rar

9. Select the radio button next to iwafjca.rar and click Next.

The Choose Targeting Style page is displayed.
10. Accept the default selection (Install this deployment as an application) and click Next.

The Select Deployment Target page is displayed.

11. In the Clusters section, select SOA_Cluster followed by All servers in the cluster.

12. Click Next.

The Optional Settings page is displayed.
13. Accept the default values and click Next.

The Summary page is displayed.

14. Click Finish.

15. In the displayed Settings page for the J2CA (iwafjca) Connector Application, click Save.
16. In the Domain Structure section in the left pane, click **Deployments** and navigate through the table that lists all the deployed applications and find the J2CA (iwafjca) connector application.

17. Select the check box next to **iwafjca**.

18. Click the **Start** submenu (down arrow) and select **Servicing all requests**.

   The Start Application Assistant page is displayed.

19. Click **Yes**.

20. Verify that the application has successfully started.
21. Similarly, repeat steps 5 through 20 for the iwafjca.war deployment.

1.2.3 Configuring the HA File Adapter for an Outbound Process

The following configuration steps must be performed in the Oracle WebLogic Server Administration Console to work with the outbound process:

1. Open the Oracle WebLogic Server Administration Console in a Web browser by entering the following URL:

   \[http://hostname:port/console\]

   Where \textit{hostname} is the name of the machine where Oracle WebLogic Server is running and \textit{port} is the port number for the domain you are using.

   The Oracle WebLogic Server Administration Console logon page is displayed.
2. Log in to the Oracle WebLogic Server Administrative Console using an account that has administrator privileges.

   The Oracle WebLogic Server Administration Console home page is displayed.

3. In the Domain Structure section in the left pane, click **Deployments**.

   The Deployments page is displayed.

4. Click **FileAdapter**.

   The Settings for FileAdapter page is displayed.
5. Click the **Configuration** tab followed by the **Outbound Connection Pools** tab. The Outbound Connection Pool Configuration Table page is displayed.

7. Click the **Properties** tab.

8. Provide a valid location for the **controlDir** property and click **Save**.

9. Check if the success message is displayed.

10. In the Domain Structure section in the left pane, click **Deployments**.

11. Select **FileAdapter** from the deployments list and click **Update**.
   The Update Application Assistant page is displayed.
12. Select **Update this application in place with new deployment plan changes** and click **Next**. The Summary page is displayed.

13. Click **Finish**.

14. Check if the success message is displayed.

15. Copy the generated `plan.xml` file from the first machine (first node in the cluster configuration) to the same location (Oracle_Home\Middleware\Oracle_SOA1\soa) in the second machine (second node in the cluster configuration).

16. Restart the servers (Admin_Server, SOA_Server1, SOA_Server2).
17. Create a shared input location that is accessible by both machines. For example, if the inputs are pasted in this location, they can be consumed by machine1 and machine2.

### 1.2.4 Configuring the Outbound Process

To configure the outbound process:

1. Create a target using Application Explorer on the first machine.
2. Connect to the target (on the first machine) and create an outbound WSDL for the Oracle Application Adapter for SAP R/3 (MySAP node).
3. On the second machine, create a target using Application Explorer with the same name as specified on the first machine.
4. Restart the Admin Server and the soa_servers on both machines.
5. Create an outbound Mediator process using Oracle JDeveloper with the configuration shown in the following image.

![Outbound Mediator Process](image)

6. Select the `Read_file_adapter.jca` file and make the following changes:
   
a. Change `<connection-factory location="eis/FileAdapter" UIincludeWildcard="*.xml" adapterRef=""/>` to `<connection-factory location="eis/HAFileAdapter" UIincludeWildcard="*.xml" adapterRef=""/>

b. Add the value `<property name="MaxRaiseSize" value="5"/>` to the end.
7. Save the process and deploy the process to both SOA servers.
8. Provide the input files in the input location configured in the Read file adapter and check if the outputs are shared and placed in the output location (in two machines) configured in the Write file adapter.

### 1.2.5 Configuring Connection Settings for the Inbound Process

The following configuration steps must be completed (for PeopleSoft, Siebel, and J.D. Edwards OneWorld Application Adapters) before executing an inbound process:

1. Navigate to the following location:
   
   \`\`\`\`Oracle_Home\product\11.1.0\ohs_1\Oracle_WT1\instances\instance1\config\OHS\ohs1

2. Open the `mod_wl_ohs.conf` file.

3. Add the IP address and port number (port number configured while creating the channel for the adapter) for both machines in the `mod_wl_ohs.conf` file.

   For example (for the PeopleSoft adapter):

   ```
   <Location /
   name>
   SetHandler weblogic-handler
   WebLogicCluster machine1_ip:port, machine2_ip:port
   WLLogFile c:\tmp\psft.log
   </Location>
   ```

   Where:
   - `name` - Is any appropriate name.
   - `machine1_ip` and `machine2_ip` - Are the machine IP addresses where SOA_Server1 and SOA_Server2 are configured.
   - `port` - The port number that is configured in the channel configuration for machine1 and machine2.

4. Open a command prompt and navigate to:

   \`\`\`\`Oracle_Home\product\11.1.0\ohs_1\Oracle_WT1\instances\instance1\bin\`

5. Restart the Oracle HTTP server using the following command:

   opmnctl restartproc ias-component=ohs1

6. When providing the URL in the backend for the adapters (PeopleSoft, Siebel, JDEdwards) the URL must be in the following format:

   `http://ohsserver_ip:7777/`
Where `ohsserver_ip` is the IP address of the machine where the Oracle HTTP server is installed and `name` is the name of the Location configured in the `mod_wl_ohs.conf` file.

### 1.2.6 Configuring the Inbound Process

To configure the inbound process:

1. Create a target and channel using Application Explorer on the first machine.
2. Connect to the target (on the first machine) and create an inbound WSDL for the Oracle Application Adapter for SAP R/3 (MySAP node).
3. On the second machine, create a target and channel using Application Explorer with the same name as specified on the first machine.
4. Restart the Admin Server and the `soa_servers` on both machines.
5. Create an inbound Mediator process using Oracle JDeveloper with the configuration shown in the following image.

![Composite Diagram](image.png)

6. Save the process and deploy the process (to both SOA servers).
7. Trigger from the backend or use HTTP publisher and check if the output is shared and placed in the output locations (in both machines) that are configured in the Write File adapter.

### 1.2.7 Singleton Testing

To perform Singleton testing:

1. On the first machine, create a target and channel using Application Explorer.
2. Connect to the target (on the first machine) and create an inbound WSDL for the Oracle Application Adapter for SAP R/3 (MySAP node).
3. On the second machine, create a target and channel using Application Explorer with the same name as specified on the first machine.
4. Restart the Admin Server and the `soa_servers` on both machines.
5. Create an inbound Mediator process using Oracle JDeveloper with the configuration shown in the following image.

![Singleton Testing Diagram](image.png)
6. For singleton testing, open the **composite.xml** file (source view) and add the following property in the Service section:

   `<property name="singleton">true</property>`

   For example:

   ```
   <service name="Service1" ui:wsdlLocation="MATMAS01_receive_cluster.wsdl">
     <interface.wsdl
       interface="http://xmlns.oracle.com/pcbpel/iWay/wsdl/MySAP/isdsrv2_cluster/MATMAS01#wsdl.interface(MATMAS01PortType)"/>
     <binding.jca config="MATMAS01_receive_cluster_3P.jca"/>
     <property name="singleton">true</property>
   </service>
   ```

7. Save the inbound Mediator process and deploy the process to both SOA servers.

8. Trigger from the backend or use HTTP publisher and check if the messages are received by any one of the machines in the output location.

9. Stop the soa_server of the machine that is receiving the messages.

10. Check whether the messages are being received by the second machine.

When you are using the Singleton testing feature with Oracle Application Adapter for SAP R/3, there are some loss of messages (messages are dumped in the SAP GUI). This is caused because the end-point activation of the second system is in progress after the first system is down.

To retrieve the lost messages, perform the following steps:

1. Login to the SAP GUI.

2. Enter the `/sm58` transaction and navigate to the dumped messages.

3. Right-click a dumped message, and then select **Execute LUW**.

4. Repeat step 3 for all the dumped messages.

   All the lost messages are received in the Oracle Enterprise Manager console.
1.3 Preferred Repository Type

As a best practice, it is recommended to use only a database repository (for example, Oracle) for adapters in development, test, and production environments. Do not use the File repository, which is provided by default only for initial startup purposes. The File repository is not supported for troubleshooting any issues.

1.4 Generating Input XML Documents

This section describes how to generate input XML documents that can be used as payloads for outbound BPEL and Mediator processes.

1.4.1 Prerequisites

Before continuing, ensure that the following components and applications are available:

- Outbound WSDL document created using Application Explorer.
- XML editor (for example, Oracle JDeveloper or Altova XML Spy, which is used as an example in this section).
- Oracle WebLogic Server 11g Release 1 (11.1.1) with Oracle JDeveloper Studio.

1.4.2 Creating a WSDL Document Using Application Explorer

To create a WSDL document using Application Explorer:

1. Ensure that Oracle WebLogic Server is started, which is where Application Explorer is deployed.
2. Start Application Explorer by clicking the Windows Start menu, selecting All Programs, Oracle Application Adapters, and clicking Application Explorer.

You can also start Application Explorer by executing the ae.bat file, which is located in the following directory:

C:\oracle\Middleware\home_0309\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\tools\iwae\bin\ae.bat

---

**Note:** It is a good practice to create a shortcut for the ae.bat file on your desktop.

---

If you are using a UNIX or Linux platform you can start Application Explorer by accessing the iwae.sh file.

3. Select an available JCA configuration.
4. Select an appropriate adapter (for example, MySAP).
5. Create a new target or connect to an existing target.
6. Expand the created target adapter and select the appropriate object.

7. Right-click the object and select **Create Outbound JCA Service(Request/Response)**.
   
The Export WSDL dialog is displayed.

8. Accept the default location in the Name field and click **OK** to export the WSDL document to the default location.
   
You can also click **Browse** to provide a different location and then click **OK**, which will export the WSDL document to your defined location.

9. Navigate to the location where the WSDL document was exported and verify that the WSDL, JCA, Request, and Response schema files are exported and available.

### 1.4.3 Generating an Input XML File From a Request Schema

To generate an input XML file from a request schema:

1. Open an XML editor (for example, Altova XML Spy, which is used as an example in this section).

2. Click **File**, and then select **Open**.
   
The Open dialog is displayed.
3. Navigate to the location on your file system where the XML request schema is exported, select the schema file and click Open.

The XML request schema file is opened and displayed in Altova XML Spy.

4. Check if the schema is well formed by clicking Check well-formedness or pressing F7.
5. Validate the schema by clicking Validate or pressing F8.

6. Once you have confirmed that the schema is well-formed and valid, click DTD/Schema, and then select Generate Sample XML File.

The Generate Sample XML File dialog is displayed.
7. Select the appropriate parameters for your sample XML file and click OK when you are ready.

The sample XML file is generated in Altova XML Spy.

8. Verify that the generated input XML file is well-formed and valid.

The generated input XML file can be used to invoke a BPEL or Mediator process after providing the necessary values in the file.

1.4.4 Using the Generated Input XML File in the Oracle Enterprise Manager Console

Before you can use the generated input XML file in the Oracle Enterprise Manager console, verify that the following prerequisites are available:

- Mediator Outbound process created in JDeveloper and deployed in SOA_Server1
- BPEL Outbound process created in JDeveloper and deployed in SOA_Server1

Mediator Outbound Process

Perform the following steps to use the input XML file in an outbound Mediator process:

1. Log in to the Oracle Enterprise Manager console by using the following URL:
2. Expand your domain in the left pane followed by the SOA folder.

3. Select an outbound Mediator project (for example, MySAP_JCA_Test_BusinessArea_GetDetail_OB_Mediator).

4. Click the Test button on the top right-hand corner of the console.
Generating Input XML Documents

5. In the Input Arguments section, select XML View from the list and verify that the input XML is displayed.

6. The displayed XML in the Oracle Enterprise Manager console can be altered and used as follows:

**XML Displayed in the Oracle Enterprise Manager Console**

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body xmlns:ns1="urn:sap-com:document:sap:business">
    <ns1:BusinessArea.GetDetail />
  </soap:Body>
</soap:Envelope>
```

**Altered XML**

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body xmlns:ns1="urn:sap-com:document:sap:business">
    <ns1:BusinessArea.GetDetail BusinessAreaId="1000" />
  </soap:Body>
</soap:Envelope>
```

**Note:** For Mediator processes, it is mandatory for the input XML to be used with a namespace.
7. The sample input XML that was generated using Altova XML Spy can be used after making the following required modifications:
   a. Remove XML headers.
   b. Add necessary inputs.
   c. Add the soap headers as displayed in the Oracle Enterprise Manager console.
   d. Modify the namespaces to match the namespaces in the input XML.

XML Generated Using Altova XML Spy

```xml
<?xml version="1.0" encoding="UTF-8"?> <!--Sample XML file generated by XMLSpy v2008 rel. 2 sp2 (http://www.altova.com)-->
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body xmlns:ns1="urn:sap-com:document:sap:business">
      <ns1:BusinessArea.GetDetail BusinessAreaId="aaaa">
         <ns1:LANGUAGE>a</ns1:LANGUAGE> <ns1:LANGUAGE_ISO>aa</ns1:LANGUAGE_ISO>
      </ns1:BusinessArea.GetDetail>
   </soap:Body>
</soap:Envelope>
```

Altered XML

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body xmlns:ns1="urn:sap-com:document:sap:business">
      <ns1:BusinessArea.GetDetail BusinessAreaId="1000">
         <ns1:LANGUAGE>D</ns1:LANGUAGE> <ns1:LANGUAGE_ISO>EN</ns1:LANGUAGE_ISO>
      </ns1:BusinessArea.GetDetail>
   </soap:Body>
</soap:Envelope>
```

8. Click **Test Web Service** after providing the input XML.

The output is displayed in the Response tab.

**BPEL Outbound Process**

Perform the following steps to use the input XML file in an outbound BPEL process:

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

   http://localhost:7001/em
2. Expand your domain in the left pane followed by the SOA folder.

3. Select an outbound BPEL project (for example, MySAP_JCA_Test_BusinessArea_GetDetail_OB_BPEL).

4. Click the Test button on the top right-hand corner of the console.
5. In the Input Arguments section, select XML View from the list and verify that the input XML is displayed.

Note: For BPEL processes, it is not mandatory for the input XML to be used with a namespace.

6. The displayed XML in the Oracle Enterprise Manager console can be altered and used as follows:

**XML Displayed in the Oracle Enterprise Manager Console**

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body xmlns:ns1="urn:sap-com:document:sap:business">
    <ns1:BusinessArea.GetDetail/>
  </soap:Body>
</soap:Envelope>
```

**Altered XML With Namespace**

```xml
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body xmlns:ns1="urn:sap-com:document:sap:business">
    <ns1:BusinessArea.GetDetail BusinessAreaId="1000"/>
  </soap:Body>
</soap:Envelope>
```
Generating Input XML Documents

General Best Practices

1-29

Altered XML Without Namespace

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body xmlns="urn:sap-com:document:sap:business">
    <BusinessArea.GetDetail BusinessAreaId="1000"/>
  </soap:Body>
</soap:Envelope>

7. The sample input XML that was generated using Altova XML Spy can be used after making the following required modifications:
   a. Remove XML headers.
   b. Add necessary inputs.
   c. Add the soap headers as displayed in the Oracle Enterprise Manager console.
   d. Modify or remove the namespaces.

XML Displayed in XML View in the Oracle Enterprise Manager Console

  <bapi:LANGUAGE>a</bapi:LANGUAGE> <bapi:LANGUAGE_ISO>aa</bapi:LANGUAGE_ISO>
</bapi:BusinessArea.GetDetail>

Altered XML With Namespace

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body xmlns:ns1="urn:sap-com:document:sap:business">
    <ns1:BusinessArea.GetDetail BusinessAreaId="1000">
      <ns1:LANGUAGE>D</ns1:LANGUAGE> <ns1:LANGUAGE_ISO>EN</ns1:LANGUAGE_ISO>
    </ns1:BusinessArea.GetDetail>
  </soap:Body>
</soap:Envelope>

8. Click Test Web Service after providing the input XML with or without a namespace.

The output is displayed in the Response tab.

Input XML With Namespace

<table>
<thead>
<tr>
<th>Input Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML View</td>
</tr>
</tbody>
</table>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body xmlns:ns1="urn:sap-com:document:sap:business">
    <ns1:BusinessArea.GetDetail BusinessAreaId="1000">
      <ns1:LANGUAGE>D</ns1:LANGUAGE> <ns1:LANGUAGE_ISO>EN</ns1:LANGUAGE_ISO>
    </ns1:BusinessArea.GetDetail>
  </soap:Body>
</soap:Envelope>

Response
Input XML Without Namespace

```
  <BUSINESSAREA_DETAIL xmlns="">
    <BUS_AREA>1000</BUS_AREA>
    <BUS_AR_DES>Mechanical engineering</BUS_AR_DES>
    <CONS_BA>1000</CONS_BA>
  </BUSINESSAREA_DETAIL>
  <RETURN xmlns="">
    <TYPE />
    <CODE />
    <MESSAGE />
    <LOG_NO />
    <LOG_MSG_NO>000000</LOG_MSG_NO>
    <MESSAGE_V1 />
    <MESSAGE_V2 />
    <MESSAGE_V3 />
</BusinessArea.GetDetail>
```

Input XML Without Namespace
This chapter lists and describes best practices that are specific to the Oracle Application Adapter for SAP R/3.

This chapter discusses the following topics:

- Understanding Connection Pools
- Security
- Load Balancing
- Encoding

### 2.1 Understanding Connection Pools

A connection pool is a set of client-connections to a certain destination. The pool may automatically create new connections to the specified remote system or return an already existing connection. It also provides methods to return a connection back to the pool when it is no longer needed. The following list describes a number of key points:

- The maximum number of connections specified can never be increased while the application is running. As a best practice, you must choose a number that is large enough for your application.

- A connection pool can have any name. Since a connection pool is global within the Java Virtual Machine (JVM), it is recommended that different applications running in the same JVM follow a naming convention standard to avoid any issues.

- If multiple users share a connection pool, then they share the SAP authorization permissions of the underlying user ID that created the connection pool.

- When a target is created using Application Explorer during design time, the specified connection parameters are used during run time for all objects created using the target. Ensure that the connection properties are capable of supporting your environment.

- The general parameters for calculating a connection pool size are: TIME=(SAP Application Server execution time for called function) + (size of document) + (network lag). It is recommended to check with a SAP Gateway administrator about the maximum size of allowed connection pools for a particular landscape.

- Larger documents or long running transactions need a larger value for the Connection Timeout parameter. Extremely long running documents should not be
run in the foreground. It is recommended to check with a system administrator about establishing a batch job that can be called by the Oracle Application Adapter for SAP R/3.

- By default, SAP Java Connector (SAP JCo) opens one initial connection for a connection pool. If that initial connection is busy when another request arrives, SAP JCo opens an additional connection, leading up to the maximum pool size. When the maximum pool size is reached, the Connection Wait Time parameter is called, which determines how long SAP JCo must wait for a free connection without aborting the task.

To monitor the number of connections that are being made to SAP from the Oracle Application Adapter for SAP R/3, you can use the SMGW transaction from your SAP GUI.

Optimizing a connection pool effectively requires knowledge about the application and its users. The following questions should be discussed:

- How much time does it normally take to execute this function?
- How much data is returned by this function?
- How many users are going to use this function?

It is not unreasonable to configure multiple connection targets with different connection pools for different functions. For example, a sales order inquiry that checks if an item exists may execute quickly, it does not need maximum resources. However, a sales order creation that creates a header and line items that is executed by many people may require a greater execution time, so configuring a larger connection pool is recommended.

2.2 Security

The following list describes a number of key points regarding security best practices:

- The SAP Java Connector (SAP JCo) default is plain text. If the SAP JCo communication has an insecure network path, it is recommended that the RFC communication is encrypted.

- Restrict the ability of users to monitor or access the SAP Gateway and its functions. Ensure that the run time user ID for SAP JCo communications is not an SAP dialog user but a communications type user.

- As a best practice, consider using the SAP router mechanism for communications between systems that are behind firewalls.

2.3 Load Balancing

There are two types of load balancing mechanisms that are available (logon load balancing and load balancing for registered programs). Logon load balancing logs users to SAP using an SAP Message Server. The Message Server distributes logons to Application Servers and can even distribute logons by specific application groups. Oracle Application Adapter for SAP R/3 supports connections to SAP using Message Servers. Load balancing for registered programs is a technique employed when sending a large amount of data out from SAP to remote destinations. As a best practice, only an SAP Gateway administrator should attempt to change the named parameters in the SAP Gateway that affect load balancing.

Since iWay registered servers (channels) connect to a SAP Gateway and not to specific application servers, they are enabled for load balancing by default through the SAP
Java Connector (SAP JCo). The mechanism for message delivery depends on how the SAP Gateway in enabled by the administrator, but it is generally one of the following forms:

- 0: No load balancing, the first free registered program is used.
- 1: The program with the lowest counter is used. Every time a registered program is assigned a request, the counter is increased by one.
- 2: The program with the least load is used, which is determined by SAP.

When there is a one-to-one relationship between a SAP Gateway server and an instance of the iWay channel, the singleton is in a non load balanced scenario. The types of messages that are sent to the server are determined by the Interface document style and the RFC destination. The RFC destination is used to hold the program ID inside SAP, and is used to route all messages to the iWay channel. For this reason, it is recommended that the SM59 transaction that controls the RFC destination is locked to hide the IP address of the remote server inside SAP. Inside SAP, RFC function modules are routed to the iWay channel by specifying the DESTINATION parameter in the CALL FUNCTION invocation, passing the RFC destination that holds the iWay server. For example:

```
CALL FUNCTION 'RFC_GET_SYSTEM_INFO DESTINATION ' 'MYDEST'
```

Where **MYDEST** is defined as a remote TCP (T) destination in the SM59 transaction and holds a Registered Server Program as one its parameters.

The iWay channel connects to the SAP Gateway and exposes the same Program ID to the SAP Gateway. At this point, one or more servers accept the connection.

SAP IDocs need additional configuration to define the sending and receiving systems. They are contained in a SAP Logical System. All IDocs routed through Message Control use the Logical System linked to a RFC destination to process through the channel. BAPI objects have no outbound form, to use a BAPI outbound object from SAP, use the RFC function form of the BAPI. For example, you can replace `Company.GetDetail` with `BAPI COMPANY_GETDETAIL`.

All messages sent to a Program ID arrive at a channel configured to listen to a SAP Gateway and Program ID. Any final destinations configured to receive messages from the channel receive all messages from the channel. This may have important implications in configuring business processes. Consider using different Program IDs for different messages, message filters or message splitters as coding techniques to route messages based on type or content.

When multiple channels or servers are configured with the same Program ID, messages can be duplicated or never arrive, depending if load balancing is enabled on the SAP system. Use caution when deploying and using Program IDs and assign them in a logical and coordinated manner (for example, by department or by message type).

### 2.4 Encoding

iWay channels on Unicode systems work only in Unicode mode. In the SM59 transaction, ensure that the RFC Destination Parameter, Unicode, is selected when creating the destination in SAP GUI.

The RFC components of SAP Java Connector (SAP JCo) automatically determine the target code page of the sending (client) and adjust the code page translation between the client and server accordingly. The only way to modify this is to change the target machine code page, in Windows using Regional and Language Options in the Control Panel. For other systems, it is recommended to consult the administrator. In general,
machine must have a code page and a language pack. Encoding is used to map one
code page to another. This is straightforward in Unicode systems, but can be difficult
or not possible on non-Unicode systems. As a general rule, in non-Unicode systems,
you are limited to the code page and language pack you have on the machine for the
display of data. However, because of the Java language Unicode support, transmission
can occur correctly if the remote system has the correct configuration.

Receiving document from SAP, especially IDocs, can have segments containing
multiple languages. It is usually not possible to get all languages correct. For example,
setting the Java encoding variable to ISO-8859-2 correctly sends German umlauts, but
distorts Japanese Kanji. The only solution for this situation is multiple sends of the
text segments and combining them into one result.
This chapter lists and describes best practices that are specific to the Oracle Application Adapter for Siebel.

This chapter discusses the following topics:

- **Best Practices for Siebel**
- **Configuring Connection Pooling for Siebel**

### 3.1 Best Practices for Siebel

This section lists and describes best practices for the Oracle Application Adapter for Siebel.

- If you are using the native Siebel Java Data Bean (JDB) API to connect with a Siebel system, integration should be performed using Siebel Business Objects and Business Services. If the HTTP protocol is required, you must use HTTP. As a best practice, it is recommended to use the Siebel JDB API for connectivity when possible.

- If large amounts of transaction volumes need processing, as a best practice, it is recommended to use connection pooling by configuring the siebel.properties file. For more information, see “Configuring Connection Pooling for Siebel” on page 3-2.

- For secure connectivity during outbound (services) processing only, it is recommended to enable RSA encryption using Application Explorer during the adapter target configuration stage.

- As a best practice, it is recommended to use XML schemas (.xsd) files that are generated through Siebel for Integration Objects and Business Services nodes.

- If any changes to the Siebel repository (.srf file) are made, it is a best practice to recompile the .srf file and restart Siebel services to reflect the changes in the adapter.

- Even though Siebel Versions 6.2 and lower (using COM connectivity) are still supported, as a best practice, it is recommended that your integration project use Siebel 6.3 or higher with the Siebel JDB API.

- When sending an Insert, Update, or Query request, it is a best practice to include only the elements in the payload that has data to be inserted, updated, or retrieved. Any blank elements should be removed.
As a best practice, do not include any special characters in the name when creating an Integration Object or Business Services node.

### 3.2 Configuring Connection Pooling for Siebel

To configure connection pooling for Siebel, you must create a `siebel.properties` file for use with the Oracle Application Adapter for Siebel. This file can be used to provide default parameters for applications connecting to Siebel using the Java Data Bean (JDB) API. The `siebel.properties` file must be added to your system classpath.

The following table lists and describes the property values that can be added in the `siebel.properties` file.

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request Timeout</td>
<td><code>siebel.conmgr.txtimeout</code></td>
<td>Indicates the transaction timeout in seconds on the server side. The default is 2700 seconds.</td>
</tr>
<tr>
<td>Poolsize</td>
<td><code>siebel.conmgr.poolsize</code></td>
<td>Indicates the connection pool size. Connection pool maintains a set of connections to a specific server process. The default is 2 with a maximum of 500.</td>
</tr>
<tr>
<td>Session Timeout</td>
<td><code>siebel.conmgr.sesstimeout</code></td>
<td>Indicates the transaction timeout in seconds on the client side. The default is 600 seconds.</td>
</tr>
<tr>
<td>Encryption</td>
<td><code>siebel.conmgr.jce</code></td>
<td>Indicates the usage of Java Cryptography Extension (JCE). Setting a value of 1 enables JCE and setting a value of 0 disables JCE.</td>
</tr>
<tr>
<td>Boolean</td>
<td><code>siebel.user.encrypted</code></td>
<td>Specifies whether the user name and the password is encrypted with com.siebel.extra.MangleString.</td>
</tr>
</tbody>
</table>
This chapter lists and describes best practices that are specific to the Oracle Application Adapter for PeopleSoft.

4.1 PeopleSoft Events (Inbound Processing)

For Release 8.4, iWay Software recommends using the TCP/IP and HTTP target connectors that are delivered by PeopleSoft for the PeopleTools 8.4 series. Do not use the target connectors that are supplied by iWay Software for the PeopleTools 8.1 series. They are only packaged by iWay Software for the PeopleTools 8.4 series to assist existing users who are migrating from Release 8.1 to Release 8.4.
This chapter lists and describes best practices that are specific to the Oracle Application Adapter for J.D. Edwards OneWorld.

This chapter contains the following examples:

- Connection Pooling and Load Balancing
- Security
- Multiple Connections and Multiple Users
- Unicode Support
- Delivery Confirmation and Error Messages

### 5.1 Connection Pooling and Load Balancing

Connection pooling is established for a session by setting the session attribute of the standard jdeRequest element. When the session attribute is an empty string, a new session is started. On the server, the SessionManager singleton class creates a new instance of a session object given the user name, password, and environment name. The session can be reused before it expires to avoid the overhead of session initialization. You can specify the session ID in the session attribute for an already established session in an earlier request.

Session expiration is addressed by the sessionIdle attribute of the standard jdeRequest element. This attribute, when given on a session creation request, specifies the amount of time in seconds that a session is allowed to be idle. If the SessionManager determines that a session has not had any requests processed in this amount of time, it terminates the session and frees all associated resources. For more information, see the *J.D. Edwards OneWorld Interoperability Guide*.

The load balancing configuration is controlled by various interrelated run time settings in the jde.ini file on the server. These settings are in the [JDENET] and applicable [JDENET_KERNEL_DEFx] sections. All relevant settings are listed and described in the preceding section. You can control two types of processes:

- Network jobs (JDENET_n)
- Dedicated kernel process jobs (JDENET_k)

For more information, see the *J.D. Edwards OneWorld System Administration Guide*. 
5.2 Security
The Oracle Application Adapter for J.D. Edwards OneWorld supports standard JDE security. Security is managed by user profiles defined in the system. For more information, see the *J.D. Edwards OneWorld Server and Workstation Administration Guide*.

5.3 Multiple Connections and Multiple Users
Multiple connections and multiple users depends on the user profiles that are defined in J.D. Edwards OneWorld. For more information, see the *J.D. Edwards OneWorld Server and Workstation Administration Guide*.

5.4 Unicode Support
The J.D. Edwards XML solution supports well-formed XML documents. It also supports UTF8 and UTF16 Unicode standards for inbound processing and UTF8 standards for outbound processing.

5.5 Delivery Confirmation and Error Messages
Each transaction is assigned session ID by J.D. Edwards OneWorld. As a best practice, use application P92002 to investigate server-side error messages.
adapter
Provides universal connectivity by enabling an electronic interface to be accommodated (without loss of function) to another electronic interface.

agent
Supports service protocols in listeners and documents.

business service
Also known as a Web service. A Web service is a self-contained, modularized function that can be published and accessed across a network using open standards. It is the implementation of an interface by a component and is an executable entity.

channel
Represents configured connections to particular instances of back-end systems. A channel binds one or more event ports to a particular listener managed by an adapter.

listener
A component that accepts requests from client applications.

port
Associates a particular business object exposed by the adapter with a particular disposition. A disposition is a URL that defines the protocol and location of the event data. The port defines the end point of the event consumption.
Index

B
best practices
  general, 1-1
  Oracle Application Adapter for J.D. Edwards
    OneWorld, 5-1
  Oracle Application Adapter for PeopleSoft, 4-1
  Oracle Application Adapter for SAP R/3, 2-1
  Oracle Application Adapter for Siebel, 3-1

C
cluster environment
  configuring, 1-3

G
general best practices, 1-1

H
high availability cluster environment, 1-3

I
input XML documents
  generating, 1-19

M
managed connections, 1-1

O
Oracle Application Adapter for Siebel
  connection pooling, 3-2
  Oracle's Unified Method (OUM), vi