# Contents

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

## 1 Introduction

**Adapter Features** ........................................................................................................................................................................... 1-1
  Supported Versions and Platforms ............................................................................................................................................... 1-2
**Classical SAP Technologies for ABAP** ......................................................................................................................................... 1-2
**Integration with MySAP ERP** ....................................................................................................................................................... 1-3
**Adapter Architecture** ................................................................................................................................................................... 1-4
**BSE Versus OracleAS Adapter J2CA Deployment** ....................................................................................................................... 1-6

## 2 Getting Started

**Verifying the SAP Java Connector (SAP JCo)** ........................................................................................................................................ 2-1
  Verifying SAP JCo on Windows Platforms .................................................................................................................................. 2-1
  Verifying SAP JCo on UNIX Platforms ......................................................................................................................................... 2-2
**Identifying mySAP Logon Parameters** ........................................................................................................................................ 2-3
  User Parameters ................................................................................................................................................................................. 2-3
  System Settings (Application Server) Parameters ........................................................................................................................ 2-3
  System Settings (Message Server) Parameters .......................................................................................................................... 2-4
  Connection Pool Parameters .......................................................................................................................................................... 2-4
  SAP Gateway Parameters .............................................................................................................................................................. 2-5
  ALE Parameters .............................................................................................................................................................................. 2-5
  Global Processing Parameters ...................................................................................................................................................... 2-6
  SNC Parameters ........................................................................................................................................................................... 2-7

## 3 Configuring OracleAS Adapter for MySAP ERP

**Starting Application Explorer** ....................................................................................................................................................... 3-1
**Configuring Settings for BSE or J2CA** ........................................................................................................................................... 3-2
  Configuring BSE ............................................................................................................................................................................ 3-2
  Configuring J2CA ........................................................................................................................................................................ 3-6
**Creating a Repository Configuration** ........................................................................................................................................... 3-8
Creating a Configuration for BSE ................................................................. 3-8
Creating a Configuration for J2CA .............................................................. 3-9
Connecting to a BSE or J2CA Configuration.............................................. 3-10
Establishing a Connection (Target) for MySAP ERP ............................. 3-11
Defining a Target to MySAP ERP ................................................................. 3-11
Connecting to a Defined MySAP ERP Target ........................................ 3-17
Managing a Connection to MySAP ERP ..................................................... 3-17
Viewing Application System Objects ......................................................... 3-18
Creating XML Schemas ............................................................................. 3-19
Generating WSDL (J2CA Configurations Only) ....................................... 3-19
Creating and Testing a Web Service (BSE Configurations Only) .............. 3-21
Configuring an Event Adapter ................................................................. 3-23
Creating and Editing a Channel ............................................................... 3-23
Schema Validation .................................................................................... 3-26

4  OC4J Deployment and Integration
Adapter Integration with OC4J .................................................................. 4-1
Deployment of Adapter ............................................................................. 4-1
Updating Adapter Configuration ............................................................... 4-2
How to Write a Java Application Client Using the CCI API ........................ 4-4

5  Integration with Oracle BPEL Process Manager
Overview of Adapter Integration with Oracle BPEL Process Manager .......... 5-1
Deployment of Adapter ............................................................................. 5-2
Design Time ............................................................................................ 5-2
  Namespace Requirements .................................................................... 5-2
  Design a BPEL Process for Request-Response Service (Outbound) .... 5-3
MySAP ERP Service Integration ............................................................... 5-9
  Design-Time Configuration ................................................................ 5-9
Invoking Adapter Request-Response Service from Oracle BPEL Process Manager .................. 5-18
MySAP ERP Event Integration ................................................................. 5-20
  Design-Time Configuration ................................................................ 5-20
  Runtime Configuration ..................................................................... 5-30

6  ESB Integration Examples
Configuring an ESB Outbound Process .................................................. 6-2
Configuring an ESB Inbound Process ...................................................... 6-32

7  Troubleshooting and Error Messages
Troubleshooting .................................................................................... 7-1
BSE Error Messages ............................................................................... 7-5
  General Error Handling in BSE ......................................................... 7-5
  Adapter-Specific Error Handling ....................................................... 7-6
8 Advanced User Tools

Web Services Policy-Based Security ................................................................. 8-1
  Configuring Web Services Policy-Based Security ........................................ 8-2
Migrating Repositories ................................................................................... 8-8

A Configuring MySAP ERP for Inbound and Outbound Processing

Configuring MySAP ERP Inbound Processing ................................................. A-1
  Configuring a Logical System ...................................................................... A-1
  Configuring a Distribution Model ............................................................... A-4
  Defining a Partner Profile .......................................................................... A-7
Configuring MySAP ERP Outbound Processing ........................................... A-9
  Related Concepts and Terminology ............................................................ A-9
  Registering Your Program ID in SAPGUI .................................................... A-11
  Testing the MySAP ERP Event Adapter ....................................................... A-12
  Application Link Embedding Configuration for the Event Adapter ............. A-14
  Defining a Port .......................................................................................... A-14
  Creating a Logical System ......................................................................... A-14
  Creating a Partner Profile .......................................................................... A-16
  Collected IDocs ......................................................................................... A-17
  Creating a Distribution Model for the Partner and Message Type ............... A-17
  Testing the MySAP ERP ALE Configuration ............................................... A-19

Glossary

Index
This Preface contains these topics:

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

**Audience**

*Oracle Application Server Adapter for MySAP ERP User’s Guide* is intended for those who perform the following tasks:

- Install applications
- Maintain applications

To use this document, you need to know how to install and configure Oracle SOA Suite (BPEL, ESB).

**Documentation Accessibility**

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

**Accessibility of Code Examples in Documentation**

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.
Accessibility of Links to External Web Sites in Documentation
This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

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

Related Documents
For more information, refer to these Oracle resources:
- Oracle Application Server Adapter Concepts
- Oracle Application Server Adapters Installation Guide

Printed documentation is available for sale in the Oracle Store at http://oraclestore.oracle.com/

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

If you already have a user name and password for OTN, then you can go directly to the documentation section of the OTN Web site at http://www.oracle.com/technology/documentation/

Conventions
This section describes the conventions used in the text and code examples of this documentation set. It describes:
- Conventions in Text
- Conventions in Code Examples
- Conventions for Windows Operating Systems

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

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

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

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

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

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

**Conventions in Code Examples**

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

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

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

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ ]</td>
<td>Anything enclosed in brackets is optional.</td>
<td>DECIMAL (<code>digits [ , precision ]</code>)</td>
</tr>
<tr>
<td>{ }</td>
<td>Braces are used for grouping items.</td>
<td>`{ENABLE</td>
</tr>
<tr>
<td></td>
<td>A vertical bar represents a choice of two options.</td>
<td>`{ENABLE</td>
</tr>
<tr>
<td>...</td>
<td>Ellipsis points mean repetition in syntax descriptions. In addition, ellipsis points can mean an omission in code examples or text.</td>
<td>CREATE TABLE ... AS subquery; SELECT col1, col2, ..., coln FROM employees;</td>
</tr>
<tr>
<td>Other symbols</td>
<td>You must use symbols other than brackets ([ ]), braces ({ }), vertical bars (</td>
<td>), and ellipsis points (...) exactly as shown.</td>
</tr>
</tbody>
</table>
### Conventions for Windows Operating Systems

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

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

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

The following list includes the specifications our consultants require.

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

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

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

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

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

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

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

- XML schema
- XML instances
- Other input documents (transformation)
- Error screen shots
- Error output files
- Trace and log files
- Log transaction
Oracle Application Server connects to a MySAP ERP system through Oracle Application Server Adapter for MySAP ERP (OracleAS Adapter for MySAP ERP). OracleAS Adapter for MySAP ERP provides connectivity and carries out interactions on a MySAP ERP system. This chapter discusses the following topics:

- Adapter Features
- Classical SAP Technologies for ABAP
- Integration with MySAP ERP
- Adapter Architecture
- BSE Versus OracleAS Adapter J2CA Deployment

**Adapter Features**

OracleAS Adapter for MySAP ERP is a remote function call adapter that provides a means to exchange real-time business data between SAP Enterprise Central Component (ECC) 5.0/6.0 systems and other application, database, or external business partner systems. The adapter enables external applications for inbound and outbound processing with MySAP ERP. OracleAS Adapter for MySAP ERP can be deployed as a J2EE Connector Architecture (J2CA) version 1.0 resource adapter. This deployment is referred to as OracleAS Adapter J2CA. It can also be deployed as a Web services servlet and is referred to as OracleAS Adapter Business Services Engine (BSE).

OracleAS Adapter for MySAP ERP uses XML messages to enable non-MySAP ERP applications to communicate and exchange transactions with MySAP ERP using services and events. The role of services and events is outlined. Services and events are described as follows:

- **Services**: Enable applications to call a MySAP ERP business object or business operation.
- **Events**: Enable applications to access MySAP ERP data only when an 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 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. In the case of this adapter, the channel is an RFC server.

OracleAS Adapter for MySAP ERP provides:
Support for bidirectional message interactions.

OracleAS Adapter Application Explorer (Application Explorer), a GUI tool which uses MySAP ERP object repository metadata to build XML schemas and Web services to handle adapter requests or event data.

Support for Remote Function Calls (RFC), Business Application Programming Interfaces (BAPI), and Intermediate Documents (IDoc) interfaces to MySAP ERP.

XML schemas and WSDL files for the J2CA 1.0 and J2CA 1.5 resource adapter.

Web services for BSE.

Data Type Limitation: Data types h and g are not supported. Type h represents a deep structure. Type g represents a variable length string. RFCTYPE_XSTRING and RFCTYPE_XMLDATA, as defined in SAPRFC.H, are not supported due to a limitation in the RFC Protocol.

See Also: Oracle Application Server Adapter Concepts

Supported Versions and Platforms

The following MySAP ERP platforms are supported by OracleAS Adapter for MySAP ERP:

- SAP R/3 Enterprise 47x100
- SAP R/3 Enterprise 47x200
- mySAP ERP Central Component (ECC) 5.0, deployed on SAP NetWeaver 2004
- mySAP ERP Central Component (ECC) 6.0, deployed on SAP NetWeaver 2004s
- SAP Java Connector (SAP JCo) Version 2.18.

For the current release status of the SAP Java Connector, refer to SAP Note #549268 in the SAP Service Marketplace.

Note: Release versions may vary by product component. In addition, SAP functions may vary by SAP product version and support package.

Classical SAP Technologies for ABAP

OracleAS Adapter for MySAP ERP is designed to provide standard access to MySAP ERP interfaces such as Remote Function Call (RFC) modules, BAPIs (Business Application Programming Interfaces), and IDocs (Intermediate Documents), that are used to support existing business processes.

The adapter only supports Enterprise Central Components (ECC) that are accessed by classical SAP technologies. If you require support for additional SAP functionality and components, please contact your iWay Software Sales Representative.

These business components and methods are available to the adapter as requests of MySAP ERP and to the event adapter when SAP invokes its remote requests and work in the following ways:

- Business Application Programming Interfaces (BAPIs) are interfaces within the business framework that are used to link SAP components to one another or to third-party components. BAPIs are called synchronously and return information.
Remote Function Call (RFC) Modules are SAP application interfaces that enable clients to invoke SAP technologies and receive responses.

**Note:** Depending on the release or service pack installed, certain RFCs, for example, `RFC_CUSTOMER_GET`, may not exist in your particular MySAP ERP system. Therefore, the examples included in this document may not be relevant to your system. If this is the case, then you should use the examples as a general reference for adapter functionality and choose an RFC that exists within your MySAP ERP application environment.

As described in SAP Release Note 109533, SAP Function Modules (RFCs) can be delivered with different release statuses. SAP supports only RFCs that are awarded with the Released for Customer status. There is no claim to the release independencies of the interfaces and the continued existence/functionality of the modules. For more information on the status of a specific function module, consult the SAP Service Marketplace.

Intermediate Documents (IDocs) are the “logical messages” that correspond to different business processes. They enable different application systems to be linked by a message-based interface. The IDoc type indicates the SAP format to use to transfer the data for a business transaction. An IDoc is a real business process in the form of an IDoc type that can transfer several message types. An IDoc type is described by the following components:

- Control records. A control record contains data that identifies the sender, the receiver, and the IDoc structure. An IDoc contains one control record.
- Data records. A data record consists of a fixed administration part and a data part (segment). The number and format of the segments can be different for each IDoc type.
- Status records. A status record describes the processing stages through which an IDoc passes. The following scenario is an example of IDoc functionality and its components:

  Purchase order number 4711 was sent to a vendor as IDoc number 0815. IDoc number 0815 is formatted in IDoc type ORDERS01 and has the status records “created” and “sent.” The purchase order corresponds to the “logical” message ORDERS.

### Integration with MySAP ERP

You can use OracleAS Adapter for MySAP ERP to initiate a MySAP ERP business process, such as add/update account, or you can use the adapter as part of an integration effort to connect MySAP ERP and non-MySAP ERP systems.

All functions are processed synchronously, but all content in ALE IDocs is asynchronous.

In service mode, the OracleAS Adapter for MySAP ERP can send requests to SAP using the BAPI, RFC, or ALE interfaces.

The adapter quickly and easily integrates your MySAP ERP IDocs, RFCs, and BAPIs with mission critical MySAP ERP system applications and other enterprise applications. The benefits of the adapter include:
- Elimination of the requirement for custom coding.
- Consistent data representation. Provides a standard XML representation of event data and request/response documents for MySAP ERP.
  The developer is freed from the specific details of the MySAP ERP interface (BAPI, RFC, IDoc) and the specific configuration details of the target MySAP ERP system.
- Adherence to MySAP ERP ABAP serialization rules and MySAP ERP Interface Repository standards published by SAP AG.

During event processing, the adapter receives RFCs and IDocs directly from MySAP ERP. The MySAP ERP system can be configured to send an IDoc or RFC to a logical system when a certain event occurs, in this case to the adapter. The output sent by MySAP ERP can be in any of the following forms:
- An RFC request, for example, RFC_SYSTEM_INFO
- A BAPI request, for example, BAPI_COMPANYCODE_GETLIST
- An IDoc

### Adapter Architecture

OracleAS Adapter for MySAP ERP uses Application Explorer with one of the following components:
- Oracle Application Server Adapter Business Services Engine (BSE)
- Enterprise Connector for J2EE Connector Architecture (J2CA)

Application Explorer (used to configure SAP connections and create Web services and events) can be configured to work in a Web services environment with BSE. When working in a J2CA environment, the connector uses the Common Client Interface (CCI) to provide integration services using adapters instead of Web services.

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

**Figure 1–1** shows the generic architecture for BSE for packaged applications.

Application Explorer works with BSE, as deployed to the Oracle Containers for J2EE (OC4J) container of Oracle 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.

BSE supports both a file-based and an Oracle database repository. The BSE repository stores the EIS connection information and the Web Service Definition Language (WSDL) for adapter services. A single instance of BSE can connect to multiple EIS applications.

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

Figure 1–2 shows the generic architecture for OracleAS Adapter J2CA for packaged applications. This is a pure J2CA 1.0 resource adapter deployed in managed mode in the OC4J container of the Oracle Application Server. It is a universal adapter. One adapter can connect to many EIS applications.

The OracleAS Adapter J2CA repository contains the list of EIS connection names and the associated connection parameters. The repository can be a file system or an Oracle database. It is deployed as a RAR file and has an associated deployment descriptor called ra.xml. You can create multiple connector factories by editing the OC4J deployment descriptor oc4j-ra.xml. See Chapter 4, "OC4J Deployment and Integration" for more information on OC4J deployment.
BSE Versus OracleAS Adapter J2CA Deployment

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

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

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

1. BSE is the preferred deployment option because it:
   - Can be deployed in a separate instance of Oracle Application Server.
   - Provides better distribution of load.
   - Provides better isolation from any errors from third party libraries.
   - Provides better capability to isolate issues for debugging purposes.

See Also:

- Oracle Application Server Adapter Concepts
- Oracle Application Server Adapters Installation Guide
- Conforms more closely to the Service Oriented Architecture (SOA) model for building applications.

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

3. OracleAS Adapter J2CA and the BSE option both provide identity propagation at run-time.
   The BSE option provides the capability to pass identity using the SOAP header. For OracleAS Adapter J2CA, user name and password can be passed using the connection specification of the CCI.
This section provides a quick start guide to use the Oracle Application Server Adapter for MySAP ERP. This chapter discusses the following topics:

- Verifying the SAP Java Connector (SAP JCo)
- Identifying mySAP Logon Parameters

**Verifying the SAP Java Connector (SAP JCo)**

Once you have installed the SAP Java Connector (SAP JCo), as a best practice, you can verify the connector to make sure it is installed correctly and that all the required SAP JCo library files are available.

**Verifying SAP JCo on Windows Platforms**

Perform the following steps to verify SAP JCo on Windows:

1. Navigate to the directory where the sapjco.jar file is located.
2. Right-click the sapjco.jar file, select Open With from the context menu, and click Java 2 Platform Standard Edition binary.

   The SAP Java Connector (JCo) dialog box is displayed.
Verifying SAP JCo on UNIX Platforms

Perform the following steps to verify SAP JCo on UNIX:

1. Navigate to a UNIX command prompt.
2. Execute the following command:

   `$ java -jar sapjco.jar -stdout`

   All the required information that pertains to the SAP Java Connector on your
   UNIX platform is provided, as shown in the following example.

------------------------------------------------------------------------
| SAP Java Connector (JCo) | Copyright (c) 2000-2005 SAP AG. All rights reserved. |
| Version Information     |                                                                |
------------------------------------------------------------------------

Java Runtime:
Operating System: SunOS 5.7 for sparc
Java VM: 1.4.0-beta3 Sun Microsystems Inc.
Java Codepage: ASCII
Versions:
JCo API: 2.1.8 (2006-12-11)
JCo middleware: 2.1.8 (2006-12-11)
JCo library: 2.1.8 (2006-12-11)
RFC library: 640.0.165
Paths:
JCo classes: /u4/fpgjpr/iWay55sm/lib/sapjco.jar
JCo library: /u4/fpgjpr/iWay55sm/lib/libsapjcorfc.so
RFC library: System-defined path
Identifying mySAP Logon Parameters

This section identifies the mySAP logon parameters, which are used to configure a connection to mySAP using the iWay Application Adapter for mySAP ERP. This information can be used as a reference.

User Parameters

The following table lists and describes User parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client</td>
<td>Identifies the SAP client.</td>
<td>800</td>
<td>In commercial, organizational, and technical terms, a self-contained unit in an SAP system with separate master records and its own set of tables. A client can, for example, be a corporate group.</td>
</tr>
<tr>
<td>User</td>
<td>SAP login ID.</td>
<td>“abc123”</td>
<td>User type for dialog-free communication between systems.</td>
</tr>
<tr>
<td>Password</td>
<td>Confidential authentication information.</td>
<td>“xyz999”</td>
<td>A protected word or string of characters that identifies or authenticates a user for access to an SAP system.</td>
</tr>
</tbody>
</table>
| Authentication Mode | How the connection is validated.                              | Selection, see next column. | Password - use the value in the supplied field.  
The password parameter that is mentioned here refers to the Password parameter field in Application Explorer. |

System Settings (Application Server) Parameters

The following table lists and describes System Settings (Application Server) parameters.
System Settings (Message Server) Parameters

The following table lists and describes System Settings (Message Server) parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Server</td>
<td>Connects to an ABAP message server.</td>
<td>iwjpsap</td>
<td>For load balancing purposes, application servers from one SAP system are usually configured in logon groups, where each group serves a particular kind of user. The message server is responsible for communication between the application servers. It passes requests from one application server to another within the system. It also contains information about application server groups and the current load balancing within them. It uses this information to choose an appropriate server when a user logs onto the system.</td>
</tr>
<tr>
<td>R/3 Name</td>
<td>Identifies a unique instance on the application server.</td>
<td>P47</td>
<td>Symbolic SAP system name used to identify the system.</td>
</tr>
<tr>
<td>Server Group</td>
<td>Identifies the logon group</td>
<td></td>
<td>Logon group that the user ID belongs with.</td>
</tr>
</tbody>
</table>

Connection Pool Parameters

The following table lists and describes Connection Pool parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection pool</td>
<td>A name for a unique pool of client connections.</td>
<td>&quot;foo&quot;</td>
<td>A pool is a set of client-connections to a certain destination with the same logon data. The pool automatically creates new connections to the specified remote system or returns an already existing one. The reusing of existing connections can increase the performance of your application by avoiding logging on to the remote server.</td>
</tr>
</tbody>
</table>
Identifying mySAP Logon Parameters

Getting Started 2-5

SAP Gateway Parameters

The following table lists and describes SAP Gateway parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP Gateway Host</td>
<td>Enter the name of a SAP Gateway server.</td>
<td>&quot;isdsrv2&quot;</td>
<td>The SAP Gateway carries out CPI-C services within the SAP world, which are based on TCP/IP. These services enable SAP Systems and external programs to communicate with one another.</td>
</tr>
<tr>
<td>SAP Gateway Service</td>
<td>Enter the service name (usually a compound of the service name and system number).</td>
<td>Sapgw00</td>
<td>Service name on the gateway host.</td>
</tr>
<tr>
<td>Program ID</td>
<td>A program identifier that has been specified on the SAP Gateway server (case sensitive).</td>
<td>&quot;S1PROG&quot;</td>
<td>Unique identifier for your communication session specified by your system administrator. The value entered in this field must match the one exposed on the gateway.</td>
</tr>
</tbody>
</table>

ALE Parameters

The following table lists and describes ALE parameters.

Table 2–5 (Cont.)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Pool Size</td>
<td>Maximum number of connections for the pool.</td>
<td>10</td>
<td>Sets the maximum number of connections that can be allocated from the pool.</td>
</tr>
<tr>
<td>Connection Timeout</td>
<td>Maximum time to keep open a free connection (in minutes).</td>
<td>5</td>
<td>Connections that have not been used for at least the connection timeout interval will be closed.</td>
</tr>
<tr>
<td>Connection Wait Time</td>
<td>Maximum wait for a free connection.</td>
<td>30 (seconds)</td>
<td>Sets the maximum time to wait in a connection request for a free connection. If the pool is exhausted, and there is still no connection available after the specified time, a JCO Exception with the key JCO_ERROR_RESOURCE will be thrown. The default value is 30 seconds.</td>
</tr>
</tbody>
</table>
Identifying mySAP Logon Parameters

Table 2–6

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDI Version</td>
<td>Specifies the ALE version of the target system.</td>
<td>3</td>
<td>Version &quot;3&quot; (Release 4.0 onwards) should be selected in the port description for all R/3 partner systems with Release 4.0 or higher.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Version &quot;2&quot; (release 3.0/3.1) must be set in the port description for all R/3 partner systems with releases lower than 4.0.</td>
</tr>
<tr>
<td>IDOC Release</td>
<td>Specifies the version in which the IDOC definition was released.</td>
<td>Blank or a specific SAP release version, for example, &quot;46C&quot;.</td>
<td>You can assign segment definitions from previous releases to an IDoc type in the current release. This may be necessary if, for example, the partner is using an older release which supports your current IDoc type but not your current segment definitions.</td>
</tr>
<tr>
<td>IDOC Release Provider</td>
<td>Specifies where the adapter will get the release information.</td>
<td>Selection, see next column.</td>
<td>IDOCDOREL uses the information in the IDOC header. SAP release gets the information from the user account logon. USERINPUT uses the IDOC release field above to get the information.</td>
</tr>
</tbody>
</table>

Global Processing Parameters

The following table lists and describes Global Processing parameters.

Table 2–7

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Handling</td>
<td>Specifies the error handling method of the adapter.</td>
<td>Selection, see next column.</td>
<td>Creates error document writes an exception document with the full error text to the output destination.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Throws exception creates a java exception, this may or may not display the full error text depending on the underlying component error.</td>
</tr>
<tr>
<td>Commit with Wait</td>
<td>Specifies the commit behavior.</td>
<td>Selection, see next column.</td>
<td>Off - default</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Sends Commit Request to Application Server at the end of the document. If there is a commit error it will not be reflected back (Optimal performance).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>On (checked) - waits for a full database server commit at the end of the document before returning. Commit errors are reflected back to the adapter level (slowest performance).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>See your SAP DB admin for your site’s recommended setting.</td>
</tr>
</tbody>
</table>
### SNC Parameters

The following table lists and describes SNC parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRACE</td>
<td>Turns on the SAP Java connectors trace behavior.</td>
<td>Selection, see next column.</td>
<td>Off default - only hard errors are written to the trace file (dev rfc.trc) in append mode.</td>
</tr>
</tbody>
</table>

#### Table 2–8

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNC mode</td>
<td>Flag for activating SNC.</td>
<td>1 (on)</td>
<td>Required.</td>
</tr>
<tr>
<td>SNC library path</td>
<td>Specifies the path and file name of the external library.</td>
<td>C:\SAP J2EE_Engine\SAPCryp\sapcactivex.dll</td>
<td>The default is the system-defined library as defined in the environment variable SNC LIB.</td>
</tr>
</tbody>
</table>
| SNC level          | Specifies the level of protection to use for the connection. | Selection, see next column.      | 1: Authentication only  
                      2: Integrity protection  
                      3: Privacy protection (default)  
                      8: Use the value from snc/data protection/use on the application server  
                      9: Use the value from snc/data_protection/max on the application server  
                      Default value = 3 |
| SNC Name           | Specifies SNC name.                   | p:CN=SAPJ2EE, O=MyCompany, C=US   | Although this parameter is optional, we do recommend setting it to make sure that the correct SNC name is used for the connection. |
| SNC Partner        | Specifies the application server’s SNC name | p:CN=ABC, O=MyCompany, C=US       | You can find the application server’s SNC name in the profile parameter snc/identity/as. |
This chapter describes how to use OracleAS Adapter Application Explorer (Application Explorer) to define a target to connect to a MySAP ERP system, view system objects, and create XML schemas and Web services. This chapter also explains how to configure an event adapter.

This chapter discusses the following topics:

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

Starting Application Explorer

To start Application Explorer:

1. Start the server where Application Explorer is deployed.

2. From the Windows Start menu, select Programs, OracleAS_home Adapters, and then Application Explorer.

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

   On UNIX, load the script iwae.sh, located under OracleAS_home/adaipers/application/tools, where OracleAS_home is the directory where Oracle Application Server is installed.

   Application Explorer starts. You can now define new targets to your MySAP ERP system.
Configuring Settings for BSE or J2CA

You need not configure BSE for a file-based repository because it is configured during the Oracle installation. You also need not configure the OracleAS Adapter J2CA because the `ra.xml` file is configured automatically during installation.

Configuring BSE

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

To configure BSE:

1. Display the following page in your browser:

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

   Where `host name` is the system where BSE is installed and `port` is the HTTP port for Oracle Application Server.

   For example,

   `http://localhost:7777/ibse`

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

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

2. Log on when prompted.

   When first installed, the user ID and password are:

   - User name: iway
   - Password: iway

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

OracleAS_home\adapters\application\lib

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

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

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

**Note:** The Repository URL field specifies where the file system repository is located. To use a database repository, you must enter the repository connection information. For the initial verification, use a file system repository. See "Configuring an Oracle Repository" on page 3-6 for information on switching to a database repository.

5. Click Save.

**Configuring BSE System Settings**

To configure BSE system settings:

1. Display the BSE configuration page in a browser:

   http://host_name:port/ibse/IBSEConfig

   Where host_name is the system where BSE is installed and port is the port number on which BSE is listening.

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

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

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

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

The following image shows the Security pane.

3. Configure the security settings.

   The following table lists the parameters with descriptions of the information to provide.
4. Configure the repository settings.

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

See "Configuring a File System Repository" on page 3-6 and "Configuring an Oracle Repository" on page 3-6 for more information.

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

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

**Configuring a File System Repository**

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

---

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

---

The default location for the repository on Windows is:

```
OracleAS_home\j2ee\OC4J_CONTAINER\applications\ws-app-adapter\ibs\ibserepo.xml
```

On other platforms, use the corresponding location.

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

**Configuring an Oracle Repository**

To configure an Oracle repository:

1. Contact your database administrator to obtain an Oracle user ID and password to create the BSE repository.
   
   This user ID should have rights to create and modify tables, and the ability to create and run stored procedures.

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

   ```
   OracleAS_home\adapters\application\etc
   ```

   For other platforms, use the corresponding location.

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

   `iwse.ora`

   ---

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

   ---

3. Enter the following command:

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

**Configuring J2CA**

During the J2CA deployment of OracleAS Adapter for MySAP ERP, OC4J generates a deployment descriptor called `oc4j-ra.xml`. This descriptor provides OC4J-specific deployment information for resource adapters. See Chapter 4, "OC4J Deployment and Integration" for more information on J2CA deployment and configuration.
No configuration changes are necessary if you are using the default file based repository with J2CA deployment.

**Configuring a Database Repository for J2CA**

To configure a database repository for J2CA:

1. Execute the `iwse.ora` SQL statement on the system where the database is installed.
2. Create the `jcatransport.properties` file and save it in the following directory:
   
   `OracleAS_HOME\adapters\application\config\jca_sample`

3. Enter values for `iwafjca.repo.url`, `iwafjca.repo.user` and `iwafjca.repo.password` fields in the newly created `jcatransport.properties` file. For example:
   
   ```
   iwafjca.repo.url=jdbc:oracle:thin:@90.0.0.51:1521:orcl
   iwafjca.repo.user=scott
   iwafjca.repo.password=scott1
   ```

4. Open the `oc4j-ra.xml` file in a text editor.
5. Provide the JDBC connection information as a value for the `IWAYRepo_URL` property.
6. Provide a valid user name for the `IWAYRepo_User` property.
7. Provide a valid password for the `IWAYRepo_Password` property.
8. Save your changes to the `oc4j-ra.xml` file.
9. Copy the Oracle database JDBC drivers (Classes12.jar, Classes12.zip, and nls_charset12.jar) to the following directory:
   
   `OracleAS_home\adapters\application\lib`

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

**Password Encryption**

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

**Configuring Password Encryption**

To encrypt a password:

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

Before you use Application Explorer with OracleAS Adapter for MySAP ERP, you must create a repository configuration. You can create two kinds of repository configurations, Web services and J2CA, depending on the container to which the adapter is deployed.

During design time, the repository is used to store metadata created when using Application Explorer to configure adapter connections, browse EIS objects, configure services, and configure listeners to listen for EIS events. The information in the repository is also referenced at run-time.

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

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

Creating a Configuration for BSE

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

Defining a New Configuration for BSE

To define a new configuration for BSE:

1. Right-click Configurations and select New.
   
   The New Configuration dialog box is displayed.

2. Enter a name for the new configuration, for example, SampleConfig, and click OK.

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 with the following format:

   http://host name:port/ibse/IBSEServlet
Creating a Repository Configuration

Configuring OracleAS Adapter for MySAP ERP

Where host name is the system on which your application server resides and port is the HTTP port number where the application server is listening.

5. Click OK.

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

The Web service repository configuration file is stored in OracleAS_home\j2ee\home\applications\ws-app-adapter\ibse.

Creating a Configuration for J2CA

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

Defining a New Configuration for J2CA

To define a new configuration for J2CA:

1. Right-click Configurations and select New.
   
   The New Configuration dialog box is displayed.

2. Enter a name for the new configuration, for example, SampleConfig, and click OK.

3. From the Service Provider list, select JCA.

4. In the Home field, enter a path to your J2CA configuration directory where the repository, schemas, and other information is stored, for example:

   OracleAS_home\adapters\application

5. Click OK.

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

The OracleAS Adapter J2CA configuration file is stored in OracleAS_home\adapters\application\config\configuration_name

Where OracleAS_home is the directory where Oracle Application Server is installed and configuration_name is the name of the configuration you created; for example, SampleConfig.
HTTP Repository Connection

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

1. Start the Application Explorer.
2. Right-click the Configurations node in the left pane and select New.
   The New Configuration dialog box opens.
3. Type a name for the configuration and click OK.
4. Select JCA from the Service Provider list box and enter an HTTP target value in the Home field.
   Use the following format for the HTTP target value:
   $\text{http://host name:port/iwafjca/JCAServlet}$
   For example:
   $\text{http://iwserv14:7777/iwafjca/JCAServlet}$
5. Click OK.
   The new HTTP repository connection is added to the Configurations node.
Once you connect to the remote server, you can create new Adapter targets, generate WSDL documents, and store them in the remote server.

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

Connecting to a BSE or J2CA Configuration

To connect to a new configuration:

1. Right-click the configuration to which you want to connect, for example, 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, you will not see the Business Services node.
   Events are not applicable when using a BSE configuration. You can configure events using a J2CA configuration only. As a result, you can disregard the Events node that appears for a BSE configuration.
   The following is an example of a BSE configuration named SampleConfig:

   - Use the Adapters folder to create inbound interaction with MySAP ERP. For example, you use the MySAP ERP node in the Adapters folder to configure a service that updates MySAP ERP.
Establishing a Connection (Target) for MySAP ERP

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

You can now define new targets to MySAP ERP.

**Establishing a Connection (Target) for MySAP ERP**

Defining the application includes adding a target for OracleAS Adapter for MySAP ERP. Setting up the target in Application Explorer requires information that is specific to the target.

To browse the available business functions, you must first define a target to MySAP ERP. After you define the target, it is automatically saved. You must connect to the MySAP ERP 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.

**Defining a Target to MySAP ERP**

To connect to MySAP ERP for the first time, you must define a new target. OracleAS Adapter for MySAP ERP supports MySAP ERP standard security and the additional protocol of SNC. Once connected to the MySAP ERP application server, application security is managed by user ID, roles and profiles. For more information on SAP application security, see the appropriate SAP documentation.

To define a target:

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

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

   The Add Target dialog box is displayed. Provide the following information:

   a. In the **Name** field, enter a descriptive name, for example, **SAPTarget**.
   b. In the **Description** field, enter a description for the target (optional).
c. From the Type list, select the type of target you are connecting to. The supported target types include Message Server or Application Server (default).

Note: For load balancing purposes, application servers from one MySAP ERP system are usually configured in logon groups, where each group serves a particular kind of user. The application servers in each group are assigned to users by a least-heavily-loaded strategy. This load balancing is done by message servers. Each MySAP ERP system has exactly one message server, which can be reached through TCP on a specific message server port.

3. Click OK.

The Application Server dialog box is displayed.

![Application Server Dialog Box]

The following tabs are available:

- User (Required)
- System (Required)
- Advanced
- Security

4. For the User tab (required), enter the appropriate information for your MySAP ERP target based on the information in the following table.

<table>
<thead>
<tr>
<th>Table 3-1 User Tab Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Target Parameter</strong></td>
</tr>
<tr>
<td>Client</td>
</tr>
<tr>
<td>User</td>
</tr>
<tr>
<td>Password</td>
</tr>
<tr>
<td>Language</td>
</tr>
<tr>
<td>Codepage</td>
</tr>
</tbody>
</table>
For more information, see your MySAP ERP system documentation.

5. For the **System** tab (required), enter the appropriate information for your MySAP ERP target based on the information in this section.

<table>
<thead>
<tr>
<th>Application Server</th>
<th>System number</th>
<th>Connection pool name</th>
<th>Connection pool size</th>
<th>Connection timeout(min)</th>
<th>Connection wait time(sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Server</td>
<td>System number</td>
<td>Connection pool name</td>
<td>Connection pool size</td>
<td>Connection timeout(min)</td>
<td>Connection wait time(sec)</td>
</tr>
<tr>
<td>&lt;null&gt;</td>
<td>&lt;null&gt;</td>
<td>01</td>
<td>2</td>
<td>10</td>
<td>30</td>
</tr>
</tbody>
</table>

The System tab enables you to provide the application server name, system number, and connection pooling information for the MySAP ERP system to which you are connecting.

Table 3–2 **System Tab Parameters**

<table>
<thead>
<tr>
<th>Target Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Server</td>
<td>The host name or IP address for the computer that is hosting the MySAP ERP application.</td>
</tr>
<tr>
<td>System number</td>
<td>The system number defined to MySAP ERP for client communications.</td>
</tr>
<tr>
<td>Connection pool name</td>
<td>The name of your MySAP ERP connection pool. A default value, p1, is already provided.</td>
</tr>
<tr>
<td>Connection pool size</td>
<td>The number of client connections in a pool you want to make available to MySAP ERP for Web service calls. A default connection pool size of 2 is available by default. Important: The default value of 1 does not create a connection pool. Instead, a single MySAP ERP connection with sequential processing is shared. A pooled connection invokes multiple connections to MySAP ERP with parallel processing. If you are using Application Explorer to create Web services, the connection pool size value is used by your Web service during run-time. As a result, ensure that the connection pool size is sufficient for your purposes.</td>
</tr>
<tr>
<td>Connection timeout(min)</td>
<td>The timeout value for your connection pool in minutes. The default value is 10 minutes.</td>
</tr>
</tbody>
</table>
6. For the Advanced tab (optional), enter the appropriate information for your MySAP ERP target based on the information in this section.

The Advanced tab enables you to specify your EDI and IDoc versions, and configure error handling.

Table 3–3 Advanced Tab Parameters

<table>
<thead>
<tr>
<th>Target Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edi version</td>
<td>The Electronic Data Interchange (EDI) document version you are using with the adapter. Version 3 is the default value.</td>
</tr>
<tr>
<td>IDOC release</td>
<td>The IDOC versioning you want to use for your connection.</td>
</tr>
<tr>
<td>IDOC release provider</td>
<td>The IDOC release provider for your connection. Select IDOC DOCREL field (default), SAP release, or user input from the drop-down list.</td>
</tr>
</tbody>
</table>
7. For the Security tab (optional), enter the appropriate information for your MySAP ERP target based on the information in this section.

### Table 3–3 (Cont.) Advanced Tab Parameters

<table>
<thead>
<tr>
<th>Target Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Error Handling   | From the list In the event of an exception, you can select Creates Error Document or Throws Exception. To receive more detailed error messages, select Creates Error Document. As a rule:  
- If your application is Java centric, select Throws Exception so that code components can catch the exception and react accordingly.  
- If your application is document based, select Creates Document to create an XML document that contains the Java exception.  
It is up to your application to read the XML document and obtain the error. |
| Commit with Wait | If a high degree of accuracy is required in your application, select the Commit with Wait check box.  
The adapter waits until all records are physically written to the database before returning from the function call. The “Commit With Wait” has a performance impact on adapter performance, so consider carefully before selecting it. For more information about the commit behavior of BAPIs, see the SAP documentation under “BAPI Programming Guide and Reference (CA-BFA).”  
All SAP Business Objects that change data must commit work to the database. Some BAPIs developed in version 3.1 of the R/3 system use an internal commit behavior, and their commit behavior cannot be changed by the adapter. As soon as they are called, they commit the work they did.  
BAPIs developed since release 3.1 use the external commit method. The adapter issues a commit command, and the commit is put in the database queue. If there is an application error in the first part of the commit, the error message “Posting could not be carried out” is returned, and the adapter rolls back the transaction. If in writing to the database, a database error occurs, a short dump is issued in the database records of SAP, but no message is returned to the adapter about the failure.  
This option is disabled by default. |
| SAP trace        | Select this option to enable traces. |
Establishing a Connection (Target) for MySAP ERP

The Security tab enables you to specify Secure Network Communication (SNC) information for the MySAP ERP system to which you are connecting.

### Table 3–4 Security Tab Parameters

<table>
<thead>
<tr>
<th>Target Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNC mode</td>
<td>By default, SNC is disabled. To enable SNC, select 1 from the list.</td>
</tr>
<tr>
<td>SNC partner</td>
<td>Enter the name of the RFC server or message server (load balancing) that provides the SNC services.</td>
</tr>
<tr>
<td>SNC level</td>
<td>From the list select the version of the SNC library.</td>
</tr>
<tr>
<td>SNC name</td>
<td>Enter the name of the SNC library you are using.</td>
</tr>
<tr>
<td>SNC library path</td>
<td>Enter the path to the SNC library.</td>
</tr>
</tbody>
</table>

SNC provides protection for the communication links between the distributed components of an MySAP ERP system. Using SNC, MySAP ERP can support products that adhere to the GSS-API Version 2 standard. SNC supports application level (end-to-end security), Smartcard authentication, and single sign-on.

If you are using SAP Enterprise Portal, the J2EE engine generates the SAP logon ticket automatically. A possible SNC scenario would be from SAP Enterprise Portal to OracleAS Adapter for MySAP ERP.

If you want to use SAP logon tickets to enable single sign-on to non-SAP components, consult the SAP documentation regarding Pluggable Authentication Services. A possible SNC scenario in this case would be from a non-SAP Enterprise Portal to OracleAS Adapter for MySAP ERP.

8. When you have provided all the required information for your target, click **OK**.

After the extraction finishes, the new target, MySAPTarget, appears under the MySAP adapter node.

You can now connect to your MySAP ERP target.
See "Creating XML Schemas" on page 3-19 for information on how to create schemas for the adapter.

**Connecting to a Defined MySAP ERP Target**

To connect to an existing target:

1. In the left pane, expand the Adapters node.
2. Expand the MySAP node.
3. Click the target name under the MySAP node (for example, MySAPTarget).
   The Connection dialog box displays the values you entered for connection parameters.
4. Verify your connection parameters.
5. Provide the correct password.
6. Right-click the target name and select Connect.
   The x icon disappears, indicating that the node is connected.

**Managing a Connection to MySAP ERP**

To manage MySAP ERP connections, you can:

- Disconnect from a connection that is not currently in use.
  Although you can maintain multiple open connections to different transaction processing systems, it is recommended to disconnect from connections not in use.
- Edit a target.
  You can modify the connection parameters when your system properties change. After you disconnect, you can modify an existing target.
- Delete a connection that is no longer needed.

**Disconnecting from a Connection to MySAP ERP**

To disconnect a target:

1. Expand the Adapters node.
2. Expand the MySAP node.
3. Right-click the target to which you are connected, for example, MySAPTarget, and select Disconnect.
Disconnecting from the MySAP ERP target drops the connection with MySAP ERP, but the node remains.

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

**Modifying Connection Parameters**

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

To edit a target:

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

The Application Server dialog box displays the target connection information.

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

**Deleting a Connection to MySAP ERP**

You can delete a connection, rather than just disconnecting and closing it. When you delete the connection, the node disappears from the list of MySAP ERP connections in the left pane of Application Explorer.

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

To delete a connection to MySAP ERP:

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

The node disappears from the list of available connections.

**Viewing Application System Objects**

As you connect to MySAP ERP, Application Explorer enables you to explore and browse MySAP ERP business objects that are used to support existing business processes.
Creating XML Schemas

After you explore the MySAP ERP business function library and select an object, you can use Application Explorer to create the XML request schema and the XML response schema for that function.

To create request and response schemas for a MySAP ERP business function.

1. Connect to a MySAP ERP target as described in "Connecting to a Defined MySAP ERP Target" on page 3-17.

2. Expand the Business Object Repository node.

3. Click the icon to the left of the Financial Accounting node.

4. Scroll down and click the icon to the left of the CompanyCode business object.

5. Scroll down and select the BAPI named GetDetail.

The following screen appears on the right.

<table>
<thead>
<tr>
<th>Response Schema</th>
<th>Event Schema</th>
<th>Reply Schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail</td>
<td>Request Schema</td>
<td></td>
</tr>
</tbody>
</table>

6. To view the XML for each schema type, click the appropriate tab.

Generating WSDL (J2CA Configurations Only)

The Web Service Definition Language (WSDL) description of a service enables you to make the service available to other services within a host server. You use Application Explorer to create both request-response (outbound) and event notification (inbound) J2CA services of the adapter.

Note: The Create Inbound JCA Service (Event) option is only available when the selected node supports events.

To generate a WSDL file for request-response service:

See the MySAP User’s Guide for more information.
1. After you create a schema, right-click the respective object. The following menu is displayed:

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

3. Accept the default name for the file.

The .wsdl file extension is added automatically. By default, the names of WSDL files generated for request-response services end with _invoke, while those generated for event notification end with _receive.

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

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

5. Click OK. The WSDL file is saved in the specified location.

The procedure for generating WSDL for event notification is similar to request-response. To generate WSDL for event notification, you must first create a channel for every event. See Generating WSDL for Event Notification on page 5-23 for a detailed example.
Creating and Testing a Web Service (BSE Configurations Only)

Using Application Explorer, you can explore the business function repository and generate Web services (also known as a business service) for the MySAP ERP functions you want to use with the adapter. The following procedure uses the MySAP ERP BAPI method called BAPI_MATERIAL_GETLIST as an example and returns a list of materials from MySAP ERP.

Creating a Web Service

To create a Web service for a MySAP ERP business function:

1. Connect to your MySAP ERP target and expand the Business Object Repository node.
2. Select the BAPI_MATERIAL_GETLIST method from the Business Object Repository.
3. Right-click the node from which you want to create a business service and select Create Web Service.

The Create Web Service dialog box is displayed. You can add the business function as a method for a new Web service or as a method for an existing one.

Perform the following steps:

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

b. If you are creating a new service, specify a service name. This name identifies the Web service in the list of services under the Business Services node.

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

4. Click Next.

The License and Method dialog box is displayed.

Provide the following information:

a. In the License Name field, select one or more license codes to assign to the Web service. To select more than one, hold down the Ctrl key and click the licenses.
b. In the **Method Name** field, enter a descriptive name for the method.

c. In the **Method Description** field, enter a brief description of the method.

d. In the **DTD Directory** field, specify the location of the DTD you want to use. The following default location is provided:

   `OracleAShome\adapters\application\tools\`

5. Click **OK**.

   Application Explorer switches the view to the **Business Services** node, and the new Web service appears in the left pane.

**Testing a Web Service**

After a Web service is created, you can test it to ensure it functions properly. A test tool is provided for testing the Web service.

To test a Web service:

1. Click the **Business Services** node to access your Web services.

2. Expand the **Services** node.

3. Select the name of the business service you want to test.

   The business service name appears as a link in the right pane.

4. In the right pane, click the named business services link.

   The test option appears in the right pane. If you are testing a Web service that requires XML input, an input field appears.

5. Enter the appropriate input.

6. Click **Invoke**.

   Application Explorer displays the results.

**Identity Propagation**

If you test or run a Web service using a third party XML editor, the Username and Password values that you specify in the SOAP header must be valid and are used to connect to MySAP ERP. The user name and password values that you provided for MySAP ERP during target creation using Application Explorer are overwritten for this Web service request. The following is a sample SOAP header that is included in the WSDL file for a Web service:

```xml
<SOAP-ENV:Header>
  <m:ibsinfo xmlns:m="urn:schemas-iwaysoftware-com:iwse">
    <m:service>String</m:service>
    <m:method>String</m:method>
    <m:license>String</m:license>
    <m:disposition>String</m:disposition>
    <m:Username>String</m:Username>
    <m:Password>String</m:Password>
    <m:language>String</m:language>
  </m:ibsinfo>
</SOAP-ENV:Header>
```

You can remove the `<m:disposition>` and `<m:language>` tags from the SOAP header, since they are not required.
Configuring an Event Adapter

Events are generated by activity in 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, your application is a consumer of this event.

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

---

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

---

A channel represents configured connections to particular instances of back-end systems. See "Creating and Editing a Channel" on page 3-23 for more information.

Creating and Editing a Channel

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

When you create, modify, or delete a channel, you must restart the Oracle Application Server to recognize the change and update the repository for run time purposes.

---

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

---

Events are not applicable when using a BSE configuration. You can configure events using a J2CA configuration only.

---

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

---

Creating a Channel

To create a channel:

1. Click the Events node.

   ![Tree view of configurations and adapters with expanded Events node]

2. Expand the MySAP node.

   The ports and channels nodes appear in the left pane.
3. Right-click **Channels** and select **Add Channel**.

   The Add Channel dialog box is displayed.

   ![Add Channel dialog box](image)

   Provide the following information:
   
   a. Enter a name for the channel, for example, TEST_CHANNEL.
   
   b. Enter a brief description.
   
   c. From the **Protocol** list, select **Application Server - mySAP** or **Message Server - mySAP**.

4. Click Next.

   The Message Server dialog box is displayed. The following tabs are available:
   
   - User (Required)
   - System (Required)
   - Security
   - Advanced

5. For the **User** tab, enter the appropriate information for your MySAP ERP channel based on the information in the following table.

   **Table 3–5 User Tab Parameters**

<table>
<thead>
<tr>
<th>Target Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Client</td>
<td>The client number defined for the SAP application for client communications.</td>
</tr>
<tr>
<td>User</td>
<td>A valid user ID for the SAP application.</td>
</tr>
<tr>
<td>Password</td>
<td>A valid password for the SAP application.</td>
</tr>
<tr>
<td>Authentication Mode</td>
<td>The authentication mode you want to use when connecting to your MySAP ERP system. By default, Password is selected from the drop-down list.</td>
</tr>
</tbody>
</table>
6. For the **System** tab, enter the appropriate information for your MySAP ERP channel based on the information in the following table.

**Table 3–6  System Tab Parameters**

<table>
<thead>
<tr>
<th>Target Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gateway host</td>
<td>A host name for the MySAP ERP Gateway.</td>
</tr>
<tr>
<td>Gateway service</td>
<td>A service for the MySAP ERP Gateway.</td>
</tr>
<tr>
<td>Program ID of the server</td>
<td>A MySAP ERP program ID you want to use for this channel.</td>
</tr>
<tr>
<td>Message Server</td>
<td>A host name for the message server.</td>
</tr>
<tr>
<td>R/3 name</td>
<td>A MySAP ERP name.</td>
</tr>
<tr>
<td>Server group</td>
<td>A MySAP ERP server group.</td>
</tr>
</tbody>
</table>

7. For the **Security** tab (optional), enter the appropriate information for your MySAP ERP channel based on the information in the following table.

**Table 3–7  Security Tab Parameters**

<table>
<thead>
<tr>
<th>Target Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNC mode</td>
<td>By default, SNC is disabled. To enable SNC, select 1 from the list.</td>
</tr>
<tr>
<td>SNC partner</td>
<td>Enter the name of the RFC server or message server (load balancing) that provides the SNC services.</td>
</tr>
<tr>
<td>SNC level</td>
<td>From the list select the version of the SNC library.</td>
</tr>
<tr>
<td>SNC name</td>
<td>Enter the name of the SNC library you are using.</td>
</tr>
<tr>
<td>SNC library path</td>
<td>Enter the path to the SNC library.</td>
</tr>
</tbody>
</table>

8. For the **Advanced** tab (optional), enter the appropriate information for your MySAP ERP channel based on the information in the following table.

**Table 3–8  Advanced Tab Parameters**

<table>
<thead>
<tr>
<th>Target Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDOC Format</td>
<td>Select an IDOC type from the list.</td>
</tr>
<tr>
<td>IDOC release</td>
<td>The IDOC versioning you want to use for your connection.</td>
</tr>
<tr>
<td>IDOC release provider</td>
<td>The IDOC release provider for your connection. Select IDOC DOCREL field (default), SAP release, or user input from the drop-down list.</td>
</tr>
<tr>
<td>SAP trace</td>
<td>Select this option to enable traces.</td>
</tr>
<tr>
<td>Processing Mode</td>
<td>Select the type of synchronous processing from the list. Possible values include REQUEST and REQUEST_RESPONSE.</td>
</tr>
</tbody>
</table>

9. Click **OK**.

The channel appears under the channels node in the left pane.
An X over the icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.

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

10. Right-click the channels node and select **Start**.
    The channel you created becomes active.

11. To stop the channel, right-click the connected channel node and select **Stop**.
    The channel becomes inactive and an X appears over the icon.

### 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 Channel pane is displayed.
3. Make the required changes to the channel configuration and click **Finish**.
   When you edit a channel, you must restart the Oracle Application Server to recognize the change and update the repository for run time purposes.

### 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 box is displayed.
3. To delete the channel you selected, click **OK**.
    The channel disappears from the list in the left pane.
    When you delete a channel, you must restart the Oracle Application Server to recognize the change and update the repository for run time purposes.

### Schema Validation

Root validation, namespace validation, and schema validation for inbound processing (events) are only supported for the OracleAS Adapter for MySAP ERP with 10.1.3.1.0 BPEL.
To validate inbound processing using the OracleAS Adapter for MySAP ERP, perform the following steps. This procedure uses MATMAS as an example for inbound processing.

1. Start Application Explorer.
2. Connect to the MySAP target.
3. Expand the IDOCs node.
4. Verify that you have already created a channel for the MySAP adapter.
5. Select and expand the MATMAS – Material Master node.
6. Right-click MATMAS01 select Create Inbound JCA Service (event) from the context menu.

   The WSDL and Validation details pane opens and includes three new check boxes for Root, Namespace, and Schema validation.
   - Selection of multiple validation options is allowed.
   - Root validation is used to validate the root element in the inbound XML document.
   - Namespace validation is used to validate the namespace in the inbound XML document.
   - Schema validation is used to validate the inbound XML document with the schema in the WSDL document.
   - During run time, validation is processed based on the validation options that are selected.
   - If more than one validation option is selected, during run time if the first validation option fails, the remaining validation options are not processed.
   - Root and namespace validations are considered modest levels of validation. Schema validation is a stricter validation level.
   - It is recommended to use root and namespace validation options together, unless the root element and namespace are different between the IDOCs in the SAP environment.

7. Generate the WSDL document and create the BPEL process.
8. Trigger the transactions (IDOCs) from the SAP GUI.

Inbound transactions that fail for the validation are shown in the SAP Transaction Monitor (SM58).

The status text field shows "java.lang.exception" for the documents that have failed the validation process.
This chapter describes Oracle Containers for J2EE (OC4J) deployment and integration with OracleAS Adapter for MySAP ERP.

This chapter discusses the following topics:

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

See Also:
- Oracle Application Server Adapter Concepts

Adapter Integration with OC4J

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

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

Deployment of Adapter

Figure 4–1 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.
Updating Adapter Configuration

During the J2CA deployment of OracleAS Adapter for MySAP ERP, OC4J generates a deployment descriptor called oc4j-ra.xml, located in OC4J_home\j2ee\home\application-deployments\default\jca-app-adapter.

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

Creating a Managed Connector Factory Object

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

```xml
<?xml version="1.0"?>
<!DOCTYPE oc4j-connector-factories PUBLIC "-//Oracle//DTD Oracle Connector >
<oc4j-connector-factories>
  <connector-factory location="eis/OracleJCAAdapter/DefaultConnection"
      connector-name="IWAFJCA10">
    <config-property name="IWayHome" value="../../adapters/application"/>
    <config-property name="IWayConfig" value="jca_sample"/>
  </connector-factory>
</oc4j-connector-factories>
```

See Also:
- Oracle Application Server Adapter Concepts
The parameters defined in the oc4j-ra.xml file are described in the following table:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IWayHome</td>
<td>The base installation directory for the OracleAS packaged application adapter.</td>
</tr>
<tr>
<td>IWayConfig</td>
<td>The adapter configuration name as defined in Application Explorer. For example, OracleAS Adapter for MySAP ERP 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 BSE repository. See &quot;Configuring an Oracle Repository&quot; in Chapter 3, &quot;Configuring OracleAS Adapter for MySAP ERP&quot; for more information.</td>
</tr>
<tr>
<td>IWayRepoUser</td>
<td>User name to use when connecting to the database. This is necessary only when using an Oracle database as the BSE repository. See &quot;Configuring an Oracle Repository&quot; in Chapter 3, &quot;Configuring OracleAS Adapter for MySAP ERP&quot; for more information.</td>
</tr>
<tr>
<td>IWayRepoPassword</td>
<td>Password. If provided, it overwrites configuration. This is necessary only when using an Oracle database as the BSE repository. See &quot;Configuring an Oracle Repository&quot; in Chapter 3, &quot;Configuring OracleAS Adapter for MySAP ERP&quot; for more information.</td>
</tr>
<tr>
<td>loglevel</td>
<td>It overwrites the level set by the ManagedConnectorFactory property.</td>
</tr>
</tbody>
</table>

Creating Multiple Managed Connector Factory Objects

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

```xml
<connector-factory location="eis/OracleJCAAdapter/DefaultConnection"
    connector-name="IWAFCAL0">
    <config-property name="IWayHome" value="/../adapters/application"/>
    <config-property name="IWayConfig" value="jca_sample"/>
    <config-property name="IWayRepoURL" value=""/>
    <config-property name="IWayRepoUser" value=""/>
    <config-property name="IWayRepoPassword" value=""/>
    <config-property name="logLevel" value="debug"/>
</connector-factory>
```

---

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

```xml
<?xml version="1.0"?>
<oc4j-connector-factories>
  <connector-factory location='eis/OracleJCAAdapter/DefaultConnection1'
    connector-name='IWAFJCA10'>
    <config-property name='IWayHome' value='../../adapters/application'/>
    <config-property name='IWayConfig' value='jca_sample'/>
    <config-property name='IWayRepoURL' value=''/>
    <config-property name='IWayRepoUser' value=''/>
    <config-property name='IWayRepoPassword' value=''/>
    <config-property name='logLevel' value='debug'/>
  </connector-factory>
  <connector-factory location='eis/OracleJCAAdapter/DefaultConnection2'
    connector-name='IWAFJCA10'>
    <config-property name='IWayHome' value='../../adapters/application'/>
    <config-property name='IWayConfig' value='jca_sample2'/>
    <config-property name='IWayRepoURL' value=''/>
    <config-property name='IWayRepoUser' value=''/>
    <config-property name='IWayRepoPassword' value=''/>
    <config-property name='logLevel' value='debug'/>
  </connector-factory>
</oc4j-connector-factories>
```

How to Write a Java Application Client Using the CCI API

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

---

**Note:** The OracleAS Adapter for MySAP ERP does not support invocation through the CCI API. Only invocation using Oracle BPEL Process Manager and ESB is supported. The following section is only provided for reference purposes.

---

**Step 1. Obtain the Connection Factory**

The connection factory is obtained by JNDI lookup.

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

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

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

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>adapterName</td>
<td>Name of the packaged application adapter.</td>
</tr>
</tbody>
</table>
The iWAFConnectionSpec can be made to initiate an interaction with MySAP ERP by specifying the adapter name and configuration parameters in the ConnectionSpec. For example,

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

In this snippet, ADAPTER and TARGET refer to the adapter being deployed and the name of a target defined in Application Explorer, respectively. See "Complete Code Sample" on page 4-6 for more information.

### Step 3. Create Interaction with InteractionSpec for Runtime

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

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

### Step 4. Create Input Record and Run Interaction

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

---

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config</td>
<td>Adapter configuration name. NOT REQUIRED FOR IWAEAdapter.</td>
</tr>
<tr>
<td>language</td>
<td>Default is en.</td>
</tr>
<tr>
<td>country</td>
<td>Default is us.</td>
</tr>
<tr>
<td>userName</td>
<td>User name. If provided, it overwrites configuration.</td>
</tr>
<tr>
<td>password</td>
<td>Password. If provided, it overwrites configuration.</td>
</tr>
<tr>
<td>logLevel</td>
<td>It overwrites the level set by the ManagedConnectionFactory property.</td>
</tr>
</tbody>
</table>

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

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

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

A special record is supported in this example:

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

Where `msg_run` is an instance XML document generated from the schema created by Application Explorer. For example, the following is a sample MySAP ERP request XML document.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<BAPI_CUSTOMER_GETDETAIL2>
<COMPANYCODE></COMPANYCODE>
<CUSTOMERNO>0000401026</CUSTOMERNO>
</BAPI_CUSTOMER_GETDETAIL2>
```

**Complete Code Sample**

The following is a sample of the complete code:

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

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

private static String  HOME    = "c:/iway/xfoc/components/iwafcont/dist";
private static String  CONFIG    = "base";
private static String  LOG_LEVEL = "FATAL";
private static String  ADAPTER  = "SAP";
private static String  TARGET  = "SAP_connection";
private static String msg_run = "<SAP/>";

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

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

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

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

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

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

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

To integrate with Oracle BPEL Process Manager, OracleAS Adapter for MySAP ERP must be deployed in the same OC4J container as Oracle BPEL Process Manager. The underlying adapter services must be exposed as WSDL files, which are generated during design time in Oracle Application Server Adapter Application Explorer (Application Explorer) for both request-response (outbound) and event notification (inbound) services of the adapter. See "Generating WSDL (J2CA Configurations Only)" on page 3-19 for more information.

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

When using the adapter with Oracle BPEL Process Manager installed on OracleAS Middle Tier, your middle-tier BPEL PM home directory is OC4J_BPEL, located as follows:

\( \text{OracleAS\_home}\text{\_j2ee}\text{\_OC4J\_BPEL} \)
Deployment of Adapter

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

See Also: Oracle Application Server Adapter Concepts
Oracle BPEL Process Manager Developer’s Guide

Design Time

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

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

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

Before you design a BPEL process, you must generate the respective WSDL file using Application Explorer. See “Generating WSDL (J2CA Configurations Only)” on page 3-19 for more information.

Namespace Requirements

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

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

1. Input XML for BPEL based on unqualified namespaces:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<CompanyCode.GetDetail>
  <COMPANYCODEID>1000</COMPANYCODEID>
</CompanyCode.GetDetail>
```

2. Input XML for BPEL based on qualified namespaces:
<?xml version="1.0" encoding="UTF-8"?>
CompanyCodeId="1000">
</bapi:CompanyCode.GetDetail>

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

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

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

Create a New Application Server Connection

To create a new Application Server connection:

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

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

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

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

5. Specify a valid user name and password for the Application Server you wish to connect to.

6. Select **Deploy Password**.

7. Click **Next**.

   The Create Application Server Connection - Step 3 of 4: Connection dialog box is displayed.
8. Select the Single Instance connection option.

9. Enter localhost as the host name and 6003 for the OPMN port.

10. Enter home as the OC4J instance name.

11. Click Next.

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

12. Click Test Connection.

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

13. Click Finish.

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

To create a new Integration Server connection:

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

2. Right-click Integration Server and select New Integration Server Connection.

   The Create Integration Server Connection - Welcome dialog box is displayed.

3. Click Next.

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

5. Select an Application Server connection, which is already created.
6. Enter localhost as the host name and 8888 for the port number.
7. Select Add host name to the list of proxy exceptions and click Next.
   The Create Integration Server Connection - Step 3 of 3: Test Connection dialog box is displayed.
8. Click Test Connection.
   When the test is complete and the connection is successful, a Success! message appears in the status area.

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

Testing Outbound BPEL and ESB Processes

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

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

The ESB data flows can be tested using the Enterprise Manager console. When creating an ESB data flow and interactions, the Web services are created and registered with the Oracle Application Server. For more information on creating an ESB outbound process, see Chapter 6, "ESB Integration Examples".
MySAP ERP Service Integration

This example demonstrates SAP service integration. It describes design-time, followed by run-time configuration.

Design-Time Configuration

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

Generating WSDL for Request/Response Service

Perform the following steps:

1. Start Application Explorer and connect to a defined MySAP ERP target (a J2CA configuration).
   See "Defining a Target to MySAP ERP" on page 3-11 for more information on defining a target and connecting to MySAP ERP.

2. Expand the MySAP ERP target to which you are connected.

3. Expand Remote Function Modules, Financial Accounting, 0002 -- Company Code Business Object, and then select BAPI_COMPANYCODE_GETDETAIL.

   The following image shows a connected and expanded target.

4. Right-click the BAPI_COMPANYCODE_GETDETAIL node.
   The following menu is displayed:
5. Click Create Outbound JCA Service (Request/Response).

The Export WSDL dialog box is displayed.

6. Click OK.

You can now design a BPEL process in JDeveloper.

**Creating a BPEL PM Server Connection in JDeveloper**

Before you design an outbound BPEL process, you must configure a new Application Server and Integration Server connection in Oracle JDeveloper. For more information, see Chapter 5, "Integration with Oracle BPEL Process Manager".

**Creating a BPEL Project for a Synchronous BPEL Process**

To create a BPEL Project for a synchronous BPEL process:

1. At the bottom of the upper left pane, click the Applications Navigator tab and select an application for your project. If an application does not exist, then you must create a new application.

2. Right-click the application and select New Project.

The New Gallery window displays a list of available items.
3. From the Items list, select **BPEL Process Project** and click **OK**. The BPEL Project Creation Wizard - Project Settings dialog box is displayed.

4. Perform the following steps:
   a. Specify a name for the BPEL process, for example, **SAP_GetCCDetail**. The Namespace field is updated automatically.
   b. From the Template list, select **Synchronous BPEL Process**.
5. Click **Next**.
   The following BPEL Project Creation Wizard dialog box is displayed.
6. Review the input and output schema elements that are created by the BPEL Project Creation Wizard and click **Finish**.

**Designing the BPEL Process for BAPI_COMPANYCODE_GETDETAIL**

To design the BPEL Process:

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

2. Click the **Service Explorer** icon (second icon from the left preceding the **WSDL File** field).
   
   The Service Explorer dialog box is displayed.
3. Expand your new connection under Adapter Services, followed by adapters, and then applications.

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

4. Select BAPI_COMPANYCODE_GETDETAIL_invoke.wsdl and click OK.

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

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

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

5. Click **OK**.

The new PartnerLink appears in the visual editor.

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

7. From the **Process Activities** pane on the right, drag an **Invoke** activity to the visual editor and place it between the Receive activity (receiveInput) and the Reply activity (replyOutput).

   The Invoke process activity is shown in the diagram view.

8. Extend a connection between the Invoke activity and your newly-created PartnerLink.
The Edit Invoke dialog box is displayed.

Perform the following steps:

a. In the Name field, enter Get_CCDetail.

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

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

9. Click OK.

10. Drag an Assign process activity and drop it between receiveInput and SAP_GetCCDetail.

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

11. Double-click the Assign activity icon.

The Assign dialog box is displayed.
12. In the Copy Operation tab, click **Create and select Copy Operation**.

   The Create Copy Operation dialog box is displayed.

   a. In the **From** pane, expand **Variables**, then **inputVariable**, and then highlight **payload**.

   b. In the **To** pane, expand **Variables**, then **Invoke_1_GetDetail_InputVariable**, and then highlight **input_GetDetail**.

   Your Create Copy Operation dialog box should look as follows:

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

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

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

16. Map **Invoke_1_GetDetail_OutputVariable**, **output_GetDetail** to **outputVariable**, **payload**.

   Verify that you have mapped all variables as follows:
17. Click OK, then click OK again.
18. Select Save from the File menu.

You have completed the design of this BPEL process.

Deploying the BPEL Process for BAPI_COMPANYCODE_GETDETAIL
JDeveloper deploys the outbound BPEL process for BAPI_COMPANYCODE_GETDETAIL directly to Oracle BPEL Console.

To deploy your BPEL process in JDeveloper:
1. Right-click your process flow in the Applications - Navigator pane.
2. Select Deploy, then Your BPEL PM Server connection, and then Deploy to default domain.
3. Observe the Messages log at the bottom of the window.
   The Messages log displays the deployment status. In this example, it shows a successful deployment message for the process.
Invoking Adapter Request-Response Service from Oracle BPEL Process Manager

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

Manage the Deployed Outbound Process in Oracle BPEL Console

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

To invoke adapter request-response service:

1. Start the Oracle BPEL Console by entering the following URL in a browser:

   http://host:port/BPELConsole

2. Select a domain and provide a valid password.

   The Oracle BPEL Console main page is displayed. All deployed BPEL processes are listed in the Dashboard tab.

3. Click the BPEL Processes tab.

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

If deployment was not successful, click the Compiler tab to view all error and warning messages generated during the deployment process.
4. Click the MySAP ERP process link, `MySAP_outbound_companycode_getdetail (v. 1.0)`. The Manage window provides options for managing this BPEL process. Do not change any of the following default settings.

5. Click the Initiate tab.

   The Initiate tab enables you to test your BPEL process.

   Perform the following steps:
   a. Select XML Source.
b. Select the Java through delivery API link in the right side of the test area.

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

```xml
<?xml version="1.0" encoding="UTF-8"?>
<CompanyCode.GetDetail CompanyCodeId="1000"/>
```

6. Click Post XML Message.

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

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

---

**MySAP ERP Event Integration**

This example demonstrates how OracleAS Adapter for MySAP ERP integrates with SAP to receive event data. In this example, an SAP event occurs when a customer record is added to a SAP system.

The design-time and run-time procedures are outlined in the following sections.

**Design-Time Configuration**

You must create a separate channel for every inbound J2CA service and select that channel when you generate WSDL for inbound interaction using Application Explorer.

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

**Creating a Channel**

To create a channel:

1. Start Application Explorer and connect to a J2CA configuration.
2. In the left pane, expand the Events node.
3. Expand the MySAP node.
   
   The Ports and Channels nodes appear in the left pane.

4. Right-click Channels and select Add Channel.
   
   The Add Channel dialog box is displayed.
Perform the following steps:

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

b. Enter a brief description (optional).

c. From the Protocol list, select Application Server - mySAP.

5. Click Next.

The Application Server dialog box is displayed. The following tabs are available:

- User (Required)
- System (Required)
- Security
- Advanced

6. For the User tab, enter the appropriate information for your MySAP ERP channel based on the information in the following table.
7. Click the **System** tab.

8. For the **System** tab, enter the appropriate information for your MySAP ERP channel based on the information in the following table.

### Table 5–2 System Tab Parameters

<table>
<thead>
<tr>
<th>Target Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gateway host</td>
<td>A host name for the MySAP ERP Gateway.</td>
</tr>
<tr>
<td>Gateway service</td>
<td>A service for the MySAP ERP Gateway.</td>
</tr>
<tr>
<td>Program ID of the server</td>
<td>A MySAP ERP program ID you want to use for this channel.</td>
</tr>
<tr>
<td>Message Server</td>
<td>A host name for the message server.</td>
</tr>
<tr>
<td>Application Server</td>
<td>A name of the MySAP ERP application server you are using.</td>
</tr>
<tr>
<td>System number</td>
<td>A MySAP ERP system number.</td>
</tr>
</tbody>
</table>

9. For the **Security** tab (optional), enter the appropriate information for your MySAP ERP channel based on the information in the following table.

### Table 5–3 Security Tab Parameters

<table>
<thead>
<tr>
<th>Target Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNC mode</td>
<td>By default, SNC is disabled. To enable SNC, select 1 from the list.</td>
</tr>
<tr>
<td>SNC partner</td>
<td>Enter the name of the RFC server or message server (load balancing) that provides the SNC services.</td>
</tr>
<tr>
<td>SNC level</td>
<td>From the list select the version of the SNC library.</td>
</tr>
</tbody>
</table>
For the **Advanced** tab (optional), enter the appropriate information for your MySAP ERP channel based on the information in the following table.

### Table 5-4 Advanced Tab Parameters

<table>
<thead>
<tr>
<th>Target Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDOC Format</td>
<td>Select an IDOC type from the list.</td>
</tr>
<tr>
<td>IDOC release</td>
<td>The IDOC versioning you want to use for your connection.</td>
</tr>
<tr>
<td>IDOC release provider</td>
<td>The IDOC release provider for your connection. Select IDOC DOCREL field (default), SAP release, or user input from the drop-down list.</td>
</tr>
<tr>
<td>SAP trace</td>
<td>Select this option to enable traces.</td>
</tr>
<tr>
<td>Processing Mode</td>
<td>Select the type of synchronous processing from the list. Possible values include REQUEST and REQUEST_RESPONSE.</td>
</tr>
</tbody>
</table>

11. Click **OK**.

The channel appears under the channels node in the left pane. An X over the icon indicates that the channel is currently disconnected.

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

## Generating WSDL for Event Notification

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

1. Start Application Explorer.
2. Expand the Adapters node.

A list of all adapters is displayed.

**Perform the following steps:**

a. Expand the MySAP node.

A list of your available targets is displayed.
b. Click a target name under the MySAP node, for example, SAPTarget.

The Connection dialog box displays the saved parameters.

3. Verify your connection parameters.

4. Provide the required password.

5. Right-click the target name and select Connect.

The x icon disappears, indicating that the node is connected.

6. Expand the IDOCs node and select DEBMAS.

The DEBMAS list is displayed.

7. Right-click DEBMAS05 from the DEBMAS list.

8. Select **Create Inbound JCA Service (Event)**.

The Export WSDL dialog box is displayed.
Perform the following steps:

a. In the Name field, specify a name for the WSDL file.
b. Ensure that qualified is selected as the element form, which is the default.
c. From the Channel drop-down list, select the channel you created for this inbound service.

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

9. Click OK.

**Creating a BPEL PM Server Connection in JDeveloper**

Before you design a BPEL process using the WSDL you generated in Application Explorer, you must configure a new Application Server and Integration Server connection in Oracle JDeveloper. For more information, see Chapter 5, "Integration with Oracle BPEL Process Manager".

**Designing the BPEL Process for the SAP_DEBMAS05 Event**

To design a BPEL process for inbound interaction:

1. Click the Applications Navigator tab and select an application for your project.
2. Right-click the application and select New Project.
   The New Gallery window is displayed.
3. From the Items list, select **BPEL Process Project** and click **OK**.

   The BPEL Process Project Creation Wizard dialog box is displayed.

4. Perform the following steps:
   a. Specify a name for the process.
      
      The Namespace field is updated automatically.
   b. From the Template list, select **Empty BPEL Process**.
   c. Click **Finish**.

   An empty BPEL process project template is created.
5. From the Services pane on the right, drag and drop a PartnerLink to the visual editor.

   The Create Partner Link dialog box is displayed.

   ![Create Partner Link dialog box](image)

6. Click the Service Explorer icon (second icon from the left preceding the WSDL File field).

   The Service Explorer dialog box is displayed.

   ![Service Explorer dialog box](image)

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

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

8. Select DEBMAS05_receive.wsdl and click OK.

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

Perform the following steps:

a. From the **My Role** list, select the default value **DEBMAS05Role**.

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

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

The new SAP PartnerLink appears in the visual editor.

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

11. Connect the Receive activity to the SAP PartnerLink.

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

a. Specify a name for the Receive Activity, for example, Receive_DEBMAS05.

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

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

12. Click Apply.

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

13. Click OK.

Your completed process looks as follows.

14. Select Save from the File menu.

**Deploying the BPEL Process for the SAP_DEBMAS05 Inbound Service**

1. Right-click your process flow in the Applications - Navigator pane.

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

Events are generated by activity in an application system. For example, MySAP ERP may generate an event as customer information is updated in the system. For more information on events, see "Configuring an Event Adapter" on page 3-23.

Triggering an Event in MySAP ERP

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

To trigger an event in MySAP ERP:

1. Start the SAP Workbench and log in to the MySAP ERP system.

2. Run the bd12 transaction.

Enter the following information in the Send Customers window:

- In the Customer field, enter a customer number with a range from 1 to 3.
- In the Output type field, enter DEBMAS.
- In the Logical system field, specify the logical system you are using with MySAP ERP.

3. Click the check mark icon in the upper left-hand corner.
4. Ensure DEBMAS appears in the Message type column.

5. Click the Execute button.
   
   Customer master data is sent to the logical system specified. If a channel in Application Explorer defined the Program ID with the same value, the channel receives this customer master data from MySAP ERP.

Verifying the Results
To verify your results:

1. Log in to Oracle BPEL Console at
   
   http://host:port/BPELConsole

2. Enter the password for your BPEL domain.

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

4. Click the SAP_DEBMAS05 instance, then click Audit to see the received SAP_DEBMAS05 event message.
MySAP ERP Event Integration

Title: Instance #002 of SAP_DEBMASOS
Reference Id: 002, Tree Finder
BPEL Process: SAP_DEBMASOS (v.1.0)

Audit trail of this BPEL instance

[2005/05/10 19:30:12] New instance of BPEL process "SAP_DEBMASOS" initiated (# "002").

[2005/05/10 19:50:22] Received "Receive_1_DEBMASOS_inputVariable" call from partner "SAP" instance
<Receive_1_DEBMASOS_inputVariable>
<partnerInfo xsi:schemaLocation="http://www.w3.org/2001/XMLSchema-instance" name="event_DEBMASOS">
<DEBMASOS xmlns="https://xmlns.oracle.com/soap/WSmsg/DOC/SAP/SAP/DEBMASOS/event0" id="AC11">
<DOC BEGIN="1">
<ED1 DOC0 SEGMENT="1">
<TADAM/ED1_DC40</TADAM>
</DOC>
</DEBMASOS>
</partnerInfo>
</Receive_1_DEBMASOS_inputVariable>
This chapter contains the following examples:

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

The scenarios shown in this chapter require the following prerequisites.

**Prerequisites**

The following are installation and configuration requirements:

- OracleAS Adapter for MySAP ERP must be installed on Oracle Application Server.
- MySAP ERP must be configured for inbound and outbound processing. See Appendix A, "Configuring MySAP ERP for Inbound and Outbound Processing" for more information.
- OracleAS Technology adapters must be deployed and properly configured.

**See Also:** Oracle Application Server Adapters Installation Guide

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

- How to configure OracleAS Adapter for MySAP ERP for services and events. For more information, see Chapter 3, "Configuring OracleAS Adapter for MySAP ERP".
- How to configure a new Application Server and Integration Server connection in Oracle JDeveloper. For more information, see Chapter 5, "Integration with Oracle BPEL Process Manager".

**Overview of ESB Integration**

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

Adapter integration with OracleAS ESB is a two-step process:
1. **Design Time**: Oracle AS Adapter for MySAP ERP is configured in Application Explorer for services and events, as described in Chapter 3, "Configuring Oracle AS Adapter for MySAP ERP". Integration logic is modeled in ESB.

2. **Runtime**: The underlying platform treats this metadata as run-time instructions to enable the communication between participating applications.

**Namespace Requirements**

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

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

1. **Input XML for ESB based on unqualified namespaces**:

   ```xml
   <?xml version="1.0" encoding="UTF-8"?>
   <CompanyCode.GetDetail>
   <COMPANYCODEID>1000</COMPANYCODEID>
   </CompanyCode.GetDetail>
   ```

2. **Input XML for ESB based on qualified namespaces**:

   ```xml
   <?xml version="1.0" encoding="UTF-8"?>
   CompanyCodeId="1000">
   </bapi:CompanyCode.GetDetail>
   ```

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

---

**Configuring an ESB Outbound Process**

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

**Prerequisites**

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

1. Create a target using Application Explorer.
2. Connect to the target.
3. Create a WSDL file.
4. Restart the Oracle Application Server.

Creating an Outbound ESB Project and Assigning an Outbound WSDL File

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

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

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

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

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

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

   The Service Explorer dialog box is displayed.

   ![Service Explorer dialog box](image)

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

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

   ![Create Adapter Service dialog box](image)
8. Select an outbound WSDL file that has been created using Application Explorer and click OK.

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

9. Click OK.

The new ESB project appears in the visual editor.
Creating a Read Process Operation Using the File Adapter

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

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

   The Adapter Configuration Wizard - Welcome window is displayed.

3. Click Next.

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

4. Click Next.

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

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

The Adapter Configuration Wizard - Step 5 of 6: File Polling window is displayed.

8. Change the Polling Frequency to seconds and click Next.

The Adapter Configuration Wizard - Step 6 of 6: Messages window is displayed.
9. Click Browse to select the WSDL.

The Type Chooser window is displayed.

10. Click the Import WSDL File icon on the upper right corner of the dialog box.

The Import WSDL File dialog box is displayed.
11. Select the WSDL file and click OK.
   The Imported WSDL Files folder is added.

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

The Adapter Configuration Wizard - Finish window is displayed.

14. Click Finish.

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

The Read operation with a routing service is added to the ESB outbound project view.

Providing a Routing Service for the Read Operation
1. Double-click the routing service.

The Routing Service window is displayed.
2. Expand the Routing Rules.

3. Click the green plus sign icon, which represents the option to Create a new Routing Rule.

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

5. Click OK.

You are returned to the Routing Rules window.

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

The Request Transformation Map dialog box is displayed.
7. Select the Create New Mapper File option, specify the file name, and click OK. The following mapping window is displayed.

8. Select the WSDL file and map it to the Read operation. Once you map the WSDL file, the Auto Map Preferences dialog box is displayed.
9. Click OK.

   The mapping is completed as shown in the following window.

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

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

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

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

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

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

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

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

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

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

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

2. Expand the Routing Rules.

3. Click the icon next to the <<Target Operation>> field (Browse for target service operations).
   The Browse Target Service Operation window is displayed.

4. Expand Services in project, Default System, your adapter service node, for example, MySAP_ESB_Outbound, and select the service name, for example, Write.
5. Click OK.
   You are returned to the Routing Rules window.

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

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

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

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

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

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

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

3. Place the XML file in the folder that you specified during the creation of the Read operation.

4. Check whether you are receiving the response in the output folder, which you have specified during the creation of the write operation and also the corresponding instance in the ESB Control console.
5. If the response is not received in the output folder, check the instance and the logs for the corresponding errors in the ESB Control console.

## Configuring an ESB Inbound Process

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

### Prerequisites

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

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

### Creating an Inbound ESB Project and Assigning an Inbound WSDL File

1. At the top of the upper left pane, click the Applications Navigator tab.
2. Right-click the application node you created and select **New Project**. The New Gallery window is displayed.

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

b. Click **OK**.

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

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

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

   The Service Explorer dialog box is displayed.

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

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

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

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

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

The Create File Adapter Service dialog box is displayed.

2. Enter a name for the File adapter and click the Configure adapter service wsdl icon next to the WSDL File field.
   
The Adapter Configuration Wizard - Welcome window is displayed.
3. Click Next.
   The Adapter Configuration Wizard - Step 1 of 4: Service Name window is displayed.

4. Click Next.
   The Adapter Configuration Wizard - Step 2 of 4: Operation window is displayed.

5. Click Write File as the Operation Type and click Next.
   The Adapter Configuration Wizard - Step 3 of 4: File Configuration window is displayed.
6. Enter the path of the output directory and name of the output file and click Next. The Adapter Configuration Wizard - Step 4 of 4: Messages window is displayed.

7. Click Browse to select the WSDL. The Type Chooser window is displayed.
8. Click the Import WSDL File icon on the upper right corner of the dialog box. The Import WSDL File dialog box is displayed.

9. Select the WSDL file and click OK. The Imported WSDL Files folder is added.
10. Expand the Imported WSDL Files folder, select an Inline Schema, for example, MATMAS01, and click OK. You are returned to the Adapter Configuration Wizard - Step 4 of 4: Messages window.

11. Click Next. The Adapter Configuration Wizard - Finish window is displayed.
12. Click Finish.

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

13. Click OK.

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

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

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

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

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

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

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

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

**Deploying the Project**

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

2. Logon to the ESB Control console to check whether the project has been successfully deployed.
3. Trigger the event.

4. Check whether you are receiving the response in the output folder, which you have specified during the creation of the write operation.
5. If the response is not received in the output folder, check the instance and the logs for the corresponding errors in the ESB Control console.
This chapter explains the limitations and workarounds when connecting to MySAP ERP. The following topics are discussed:

- Troubleshooting
- BSE Error Messages

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

**Troubleshooting**

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

- Application Explorer
- MySAP ERP
- OracleAS Adapter J2CA
- BSE

---

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

- The OracleAS Adapter J2CA trace information can be found under the `OracleAS_home\opmn\logs` directory.
- BSE trace information can be found under the `OracleAS_home\j2ee\home\applications\ws-app-adapter\ibse\ibselogs` directory.
- The log file for Application Explorer can be found under the `OracleAS_home\adapters\application\tools` directory.

---

**Application Explorer**

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

`OracleAS_home\adapters\application\tools`

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

`OracleAS_home/adapters/application/tools`
<table>
<thead>
<tr>
<th>Error</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySAP ERP does not appear in the Application Explorer Adapter node list.</td>
<td>Ensure that the <code>sapjco.jar</code> and <code>sapjcorfc.dll</code> files are added to the <code>lib</code> directory. Ensure the <code>librfc32.dll</code> file is added to the Windows <code>system32</code> folder.</td>
</tr>
</tbody>
</table>
| Cannot connect to OracleAS Adapter for MySAP ERP from Application Explorer. | Ensure that:  
  - MySAP ERP is running.  
  - The Application Server name, System Number, and Client Number are correct.  
  - The MySAP ERP user ID and password are correct. |
<p>| Cannot connect to the MySAP ERP target through Application Explorer. The following error message appears: | Ensure that you enter the correct connection parameters when connecting to the MySAP ERP target. |
| Error getting target [SAP] - java.lang.Exception: Error Logon to SAP System | |
| Cannot connect to your MySAP ERP system through Application Explorer. The following error message appears: | Ensure that MySAP ERP is running and that you are using the correct parameter values to connect to your application server. |
| Problem activating adapter. (com.ibi.sapr3.SapAdapterException : com.sap.mw.jco.JCO$Exception: (102) RFC_ERROR_COMMUNICATION: Connect to SAP gateway failed Connect_PM GHOST=isdsrv8, GWserv=sapgw00, AHOST=isdsrv8, SYSNR=00 LOCATION CPIC (TCP/IP) on local host ERROR partner not reached (host isdsrv8, service 3300) TIME Fri Aug 27 11:49:14 2004 RELEASE 620 COMPONENT NI (network interface) VERSION 36 RC -10 MODULE ninti.c LINE 979 DETAIL NiPConnect2 SYSTEM CALL SO_ERROR ERNNO 10061 ERNNO TEXT WSAECONNREFUSED: Connection refused COUNTER 1). Check logs for more information | |
| Cannot connect to your MySAP ERP system through Application Explorer even though MySAP ERP is running. The following error message appears: | Ensure that the <code>sapjcorfc.dll</code> file is added to the <code>lib</code> directory and the <code>librfc32.dll</code> file is added to the Windows <code>system32</code> folder. |</p>
<table>
<thead>
<tr>
<th>Error</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>The dll is loaded in another classloader (BSE and J2CA are installed on the same server). The following error message appears: com.ibi.sap3.SapAdapterException: java.lang.ExceptionInInitializerError: JCO.classInitialize(): Could not load middleware layer 'com.sap.mw.jco.rfc.MiddlewareRFC' JCO.nativeInit(): Could not initialize dynamic link library sapjcorfc [Native Library F:\iWay55.008.0628\lib\sapjcorfc.dll already loaded in another classloader]. java.library.path</td>
<td>Add sapjco.jar to the server classpath.</td>
</tr>
<tr>
<td>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.iwaysoftware.iwae.common.JCATransport.getConnectionFactory(JCATransport.java:133) at com.iwaysoftware.iwae.common.JCATransport.initJCA(JCATransport.java:69) at com.iwaysoftware.iwae.common.JCATransport.&lt;init&gt;(JCATransport.java:62) at com.iwaysoftware.iwae.common.AdapterClient.&lt;init&gt;(AdapterClient.java:85) at com.ibi.bse.ConfigWorker.run(ConfigWorker.java:41) at java.lang.Thread.run(Thread.java:534)</td>
<td>JAVA_CMD is not set on the user system. Before starting Application Explorer, export JAVA_CMD as follows: JAVA_CMD=/&lt;jdk_home&gt;/bin/java, where &lt;jdk_home&gt; is the directory where JDK is installed on your system.</td>
</tr>
<tr>
<td>Logon failure error at run-time The following exception occurs when you start Application Explorer by activating ae.bat (not iaexplorer.bat): java.lang.ClassNotFoundException: org.bouncycastle.jce.provider.BouncyCastleProvider</td>
<td>If the password for connecting to your MySAP ERP system is not specified when creating a target or with the Edit option in Application Explorer, you will be unable to connect to MySAP ERP. The connection password is not saved in repository.xml. Update the password using the Edit option in Application Explorer, then restart the application server.</td>
</tr>
<tr>
<td>This is a benign exception. It does not affect adapter functionality. Download BouncyCastle files from: ftp://ftp.bouncycastle.org/pub</td>
<td></td>
</tr>
</tbody>
</table>

Troubleshooting
### MySAP ERP

<table>
<thead>
<tr>
<th>Error</th>
<th>Solution</th>
</tr>
</thead>
</table>
| When executing a request, the following error message appears:  
AdapterException:
java.lang.Exception: Function module CUSTOMER_GETDETAIL2 does NOT exist.  
When executing a request, the following error message appears:  
AdapterException:
java.lang.Exception: Object type unknown for business object: CUST  
When executing a request, the following error message appears:  
AdapterException:
java.lang.Exception: Unable to retrieve BAPI name for: CUSTOMER.DETAIL2  
When executing a request, the following error message appears:  
java.lang.RuntimeException: com.sap.mw.jco.JCO$AbapException: (126) OBJECT_UNKNOWN: Basic type or extension does not exist.  
When executing a request, the following error message appears:  
AdapterException:
java.lang.Exception: BapiError/BapiAbort: You are not authorized to display customers.  | Check the syntax of your input XML document and make sure the name of the Remote Function module is correct, available in MySAP ERP, and activated.  
Check the syntax of your input XML document and verify that the Business Object type exists in MySAP ERP and is correct and activated. Also verify that the name of your document is correct.  
Check the syntax of your input XML document and verify that the name of the BAPI is correct, available in MySAP ERP, and activated.  
Check the syntax of your input XML document and verify that the IDoc type or extension type is correct, available in MySAP ERP, and activated.  
Verify that your user ID has the correct permissions configured in MySAP ERP. Consult your MySAP ERP administrator for more information. |

---

**Note:** Activation is an important SAP concept. If an object does not exist in an activated state, it may appear as present on the system, but is not available for use. This is especially important for IDOCs and SAP Business Objects. Consult your SAP documentation for further information.

---

### OracleAS Adapter J2CA

<table>
<thead>
<tr>
<th>Error</th>
<th>Solution</th>
</tr>
</thead>
</table>
| In Application Explorer, the following error message appears when you attempt to connect to an OracleAS Adapter J2CA configuration:  
Could not initialize JCA | In the Details tab in the right pane, ensure that the directory specified in the Home field points to the correct directory, for example, OracleAS_home\adapters\application |
## BPEL Process Manager

<table>
<thead>
<tr>
<th>Error</th>
<th>Solution</th>
</tr>
</thead>
</table>
| Endpoint activation error on deployment of MySAP ERP event handling project (inbound) in JDeveloper | Verify that the channel used for this inbound J2CA service is stopped in Application Explorer. If you have started this channel for testing or debugging purposes, you must stop it before starting BPEL PM Server. Endpoint activation is managed by BPEL Process Manager. The following error message appears in BPEL PM Server Console:  
<ERROR><default.collaxa.cube.engine.deployment> <Cube ProcessLoader::create> Failed to read wsdl. Error happened when reading wsdl at  
"http://127.0.0.1:7777/BPELConsole/wsll/adapters/applications/CUSTOM_ER_invoke.wsdl?wsdl", because  
The following exception is thrown in JDeveloper during deployment of the BPEL process:  
java.io.FileNotFoundException: \BPELConsole\wsll\adapters\applications\DEBMAS01_receive.wsdl?wsdl  
(The system cannot find the path specified) | Verify that the specified WSDL file exists at that URL and that the file is valid.  
Workaround: Change the WSDL location to localhost:7777. The default is 127.0.0.1:7777.  
Alternative workaround: Add the IP address to the Dhttp.nonProxyHosts list found in obsetenv.bat (Windows) or obsetenv.sh (UNIX) Verify that you have all the required patches installed. The required patches are listed and updated on the Oracle Technology Network Web site. |

### BSE Error Messages

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

#### General Error Handling in BSE

BSE serves as both a SOAP gateway into the adapter framework and as the engine for some of the adapters. In both design-time and run-time, various conditions can cause errors in BSE when Web services that use adapters run. Some of these conditions and resulting errors are exposed the same way, regardless of the specific adapter; others are exposed differently, based on the adapter being used. This topic explains what you can expect if you encounter some of the more common error conditions on an adapter-specific basis. Usually the SOAP gateway (agent) inside BSE passes a SOAP request message to the adapter required for the Web service. If an error occurs, how it is exposed depends on the adapter and the API or interfaces that the adapter uses. A few scenarios cause the SOAP gateway to generate a SOAP fault. In general, anytime the SOAP agent inside BSE receives an invalid SOAP request, a SOAP fault element is
generated in the SOAP response. The SOAP fault element contains fault string and fault code elements. The fault code contains a description of the SOAP agent error.

The following SOAP response document results when BSE receives an invalid SOAP request:

```xml
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Client</faultcode>
      <faultstring>Parameter node is missing</faultstring>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

In this example, BSE did not receive an element in the SOAP request message that is mandatory for the WSDL for this Web service.

**Adapter-Specific Error Handling**

When an adapter raises an exception during run-time, the SOAP agent in BSE produces a SOAP fault element in the generated SOAP response. The SOAP fault element contains fault code and fault string elements. The fault string contains the native error description from the adapter target system. Since adapters use the target system interfaces and APIs, whether an exception is raised depends on how the target systems interface or API treats the error condition. If a SOAP request message is passed to an adapter by the SOAP agent in BSE and that request is invalid based on the WSDL for that service, the adapter may raise an exception yielding a SOAP fault. While it is almost impossible to anticipate every error condition that an adapter may encounter, the following is a description of how adapters handle common error conditions and how they are then exposed to the Web services consumer application.

**OracleAS Adapter for MySAP ERP Invalid SOAP Request**

If OracleAS Adapter for MySAP ERP receives a SOAP request message that does not conform to the WSDL for the Web services being carried out, then the following SOAP response is generated.

```xml
<?xml version="1.0" encoding="ISO-8859-1" ?>
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

**Empty Result From SOAP Request**

If OracleAS Adapter for MySAP ERP carries out an MySAP ERP object using input parameters passed in the SOAP request message that do not match records in MySAP ERP, then the following SOAP response is generated.

```xml
<?xml version="1.0" encoding="ISO-8859-1" ?>
```
Failure to Connect to MySAP ERP
If OracleAS Adapter for MySAP ERP cannot connect to MySAP ERP when executing a Web service, then the following SOAP response is generated:

Invalid SOAP Request
If OracleAS Adapter for MySAP ERP receives a SOAP request message that does not conform to the WSDL for the Web services being carried out, then the following SOAP response is generated.

Empty Result From OracleAS Adapter for MySAP ERP Request
If OracleAS Adapter for MySAP ERP carries out 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.

```xml
<SOAP-ENV:Envelope xmlns:xsi="http://www.w3.org/1999/XMLSchema-instance"
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/1999/XMLSchema">
  <SOAP-ENV:Body>
    <m:RunDBQueryResponse xmlns:m="urn:schemas-iwaysoftware-com:iwse"
xmlns="urn:schemas-iwaysoftware-com:iwse"
    cid="2A3CB42703EB20203F91951B89F3C5AF">
      <RunDBQueryResult run="1" />
    </m:RunDBQueryResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
This chapter includes the following topics:

- Web Services Policy-Based Security
- Migrating Repositories

**Web Services Policy-Based Security**

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

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

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

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

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

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

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

When a policy is not applied, the default value for a Business Service is to "grant all". For example, anybody can run the Business Service, until the Resource Execution policy is associated to the Business Service. At that time, only those granted execution
permissions, or users not part of the group that has been denied execution permissions, have access to the Business Service.

**Configuring Web Services Policy-Based Security**

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

**Creating and Associating a User with a Policy**

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

1. Open Application Explorer.
2. Right-click the configuration to which you want to connect, for example, SampleConfig. See Chapter 3, "Configuring OracleAS Adapter for MySAP ERP" for information on creating a new configuration.
3. Select Connect.
   
   Nodes appear for Adapters, Events, and Business Services (also known as Web services).

   Perform the following steps:
   a. Expand the Business Services node.
   b. Expand the Configuration node.
   c. Expand the Security node.
   d. Expand the Users and Groups node.
4. Right-click Users and click New User.
   The New User dialog box is displayed.
Provide the following information:

a. In the Name field, enter a user ID.

b. In the Password field, enter the password associated with the user ID.

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

5. Click OK.

The new user is added under the Users node.

**Creating a Group to Use With a Policy**

To create a group to use with a policy:

1. Open Application Explorer.

2. Right-click the configuration to which you want to connect, for example, SampleConfig. See Chapter 3, "Configuring OracleAS Adapter for MySAP ERP" for information on creating a new configuration.

3. Select Connect.

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

Perform the following steps:

a. Expand the Business Services node.

b. Expand the Configuration node.

c. Expand the Security node.

d. Expand the Users and Groups node.

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

Provide the following information:

a. In the Name field, enter a name for the group.

b. In the Description field, enter a description for the group (optional).

c. From the available list of users in the left pane, select one or more users and add them to the Selected list by clicking the double right-facing arrow.

5. When you have selected at least one user, click OK.

The new group is added under the Group node.

Creating an Execution Policy
An execution policy governs who can run the Business Services to which the policy is applied.

To create an execution policy:

1. Open Application Explorer.

2. Right-click the configuration to which you want to connect, for example, SampleConfig. See Chapter 3, "Configuring OracleAS Adapter for MySAP ERP" for information on creating a new configuration.

3. Select Connect.

Nodes appear for Adapters, Events, and Business Services (also known as Web services).
Perform the following steps:

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

4. Right-click Policies and select New Policy.

   ![Policy tree with New Policy highlighted]

   The New policy dialog box is displayed.

   ![New Policy dialog box]

   Provide the following information:

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

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

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

   The New Policy permissions dialog box is displayed.
To grant permission to a user or group to run a Business Service, select the user or group and move them into the Execution Granted list by selecting the double left-facing arrow.

To deny permission to a user or group to run a Business Service, select the user or group and move them into the Execution Denied list by selecting the double right-facing arrow.

7. Click OK.

The following pane summarizes your configuration.

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

**Using the IP and Domain Restrictions Policy Type**

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

1. Open Application Explorer.

2. Right-click the configuration to which you want to connect, for example, SampleConfig. See Chapter 3, "Configuring OracleAS Adapter for MySAP ERP" for information on creating a new configuration.

3. Select Connect.

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

a. Expand the Business Services node.
b. Expand the Configuration node.

c. Expand the Security node.

4. Right-click IP and Domain and select New IP and Domain Restriction.

The New IP and Domain Restriction dialog box is displayed.

Provide the following information:

a. In the IP(Mask)/Domain field, enter the IP or domain name using the following guidelines.
   - If you select Single (Computer) from the Type list, you must provide the IP address for that computer. If you only know the DNS name for the computer, click DNS Lookup to obtain the IP Address based on the DNS name.
   - If you select Group (of Computers), you must provide the IP address and subnet mask for the computer group.
   - If you select Domain, you must provide the domain name.

b. From the Type list, select the type of restriction.

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

d. To grant access, select the Grant Access check box.

5. Click OK.

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

The following pane summarizes your configuration.

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

During design time, the Oracle repository is used to store metadata created when using Application Explorer to configure adapter connections, browse EIS objects, configure services, and configure listeners to listen for EIS events. The information in the repository is also referenced at run-time. For management purposes, you can migrate BSE and J2CA repositories that are configured for Oracle to new destinations without affecting your existing configuration. For example, you may want to migrate a repository from a test environment to a production environment.

Migrating a BSE Repository

To migrate a BSE repository:

1. Copy the BSE control service URL, for example:
   
   http://localhost:7777/ibse/IBSServlet/admin/iwcontrol.ibs

2. Open a third-party XML editor, for example, XMLSpy.

3. From the menu bar, click SOAP.

   A list of options appears.

   ![SOAP Options](image)

4. Select Create new SOAP request.

   The WSDL file location dialog box is displayed.

   ![WSDL Dialog](image)

   Perform the following steps:

   a. In the Choose a file field, paste the BSE control service URL.

   b. Append ?wsdl to the URL, for example:

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

5. Click OK.

   The soap operation name dialog box is displayed and the available control methods are listed.
6. Select the MIGRATEREPO(MIGRATEREPO parameters) control method and click OK.

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

The following window is displayed, showing the structure of the SOAP envelope.

7. Locate the Text view icon in the toolbar.

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

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

9. Locate the following section:

```xml
<m:MIGRATEREPO xmlns:m="urn:schemas-iwaysoftware-com:jul2003:ibse:config" version=""/>
```
Perform the following steps:

a. For the `<m:rconn>` tag, replace the String placeholder with a repository URL where you want to migrate your existing BSE repository. The Oracle repository URL has the following format:
   
jdbc:oracle:thin:@[host]:[port]:[sid]

b. For the `<m:rdriver>` tag, replace the String placeholder with the location of your Oracle driver.

c. For the `<m:ruser>` tag, replace the String placeholder with a valid user name to access the Oracle repository.

d. For the `<m:rpwd>` tag, replace the String placeholder with a valid password to access the Oracle repository.

10. Perform one of the following migration options.

   ■ If you want to migrate a single Web service from the current BSE repository, enter the Web service name in the `<m:servicename>` tag, for example:
       
       `<m:servicename>SAPService1</m:servicename>`

   ■ If you want to migrate multiple Web services from the current BSE repository, duplicate the `<m:servicename>` tag for each Web service, for example:
       
       `<m:servicename>SAPService1</m:servicename>`
       `<m:servicename>SAPService2</m:servicename>`

   ■ If you want to migrate all Web services from the current BSE repository, remove the `<m:servicename>` tag.

11. From the menu bar, click SOAP and select Send request to server.

   
   ![SOAP Menu]

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

**Migrating a J2CA Repository**

To migrate a J2CA repository:

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

   OracleAS_home\adapters\application\config\JCA_CONFIG
Where JCA\_CONFIG is the name of your J2CA configuration.

2. Locate and copy the repository.xml file.

3. Place this file in a new J2CA configuration directory to migrate the existing repository.

Your J2CA repository is migrated to the new J2CA configuration directory.
Configuring MySAP ERP for Inbound and Outbound Processing

During inbound (client) processing, IDocs are transferred to the interface and stored in the MySAP ERP system. The document data is generated in a second step, also in the course of a workflow.

Outbound processing in MySAP ERP involves event handling. An event in MySAP ERP is defined as an occurrence of a status change in an object. Events are created when the relevant status change occurs.

The following topics describe how to enable inbound and outbound MySAP ERP processing.

- Configuring MySAP ERP Inbound Processing
- Configuring MySAP ERP Outbound Processing

Configuring MySAP ERP Inbound Processing

MySAP ERP inbound processing requires the upstream system to transfer an IDoc to the IDoc interface through the ERP System port. For this reason, you do not have to specify a port in the inbound partner profiles; the IDoc interface only must recognize the upstream system as a port. A port definition, which provides a unique ID for the upstream system, must be available for the port. The technical parameters of this port definition can (and usually are) overwritten by the upstream system.

If the upstream system is recognized, then the IDoc is saved in the database. If a partner is defined with the corresponding message in partner profiles, the IDoc is then processed further. This is done independently in the second step. This ensures that the external system can receive the data quickly and reliably (automatically).

You must perform the following steps to configure MySAP ERP for inbound IDoc processing:

1. Configure a logical system.
2. Configure a distribution model.
3. Define an inbound partner profile.

Configuring a Logical System

In any distributed environment, each participating system must have a unique ID to avoid confusion. In MySAP ERP, the name of the logical system is used as the unique ID. This name is assigned explicitly to one client in a MySAP ERP system.
Defining a Logical System

To define a logical system:

1. Run the sale transaction.

SAP Easy Access

The Display IMG window is displayed.

Display IMG

Perform the following steps:

a. Expand Sending and Receiving Systems.

b. Expand Logical Systems.

c. Select Define Logical System.

2. Click the IMG - Activity icon.

A message window is displayed. It indicates that the table is cross-client.

3. Click the check mark icon to continue.

The Change View "Logical Systems": Overview window is displayed.
4. Click New Entries.
   The New Entries: Overview of Added Entries window is displayed.

5. Enter the Logical System, for example, ORACLETDS, in the Log.System column and provide a description in the Name column.

6. Click Save.
   The Prompt for Workbench request dialog box is displayed.

7. Click the Create Request icon.
   The Create Request dialog box is displayed.
8. Enter a name and description for your request and click Save. The logical system you configured, for example, ORACLETDS, is now added to the list.

<table>
<thead>
<tr>
<th>Logical System</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IWAYMKT</td>
<td>IWAY marketing logical system</td>
</tr>
<tr>
<td>IWAY_IN</td>
<td>ale inbound processing</td>
</tr>
<tr>
<td>JR346LS</td>
<td>jr logical</td>
</tr>
</tbody>
</table>

**Configuring a Distribution Model**

A distribution model is used to describe the ALE message flow between logical systems. Business objects are distributed to connected recipients according to a unique distribution model that can contain rules of varying complexity depending on the type of business objects involved.

**Defining a Distribution Model**

To define a distribution model:

1. Run the **bd64** transaction.

**Display IMG**

The Display Distribution Model window is displayed.
2. Click Distribution Model from the menu bar.

3. Select **Switch processing mode**.
   
The Display Distribution Model window is switched to Change Distribution Model.

4. Click Create model view.
   
The Create Model View dialog box is displayed.
5. Enter a model view name in the **Short text** field and a name in the **Technical name** field, which also serves as a description.

6. Click the **check mark** icon to enter the information.

   You are returned to the main Change Distribution Model window. The distribution model you configured is now added to the list.

7. Click Add message type.

   The Add Message Type dialog box is displayed.

   Perform the following steps:

   a. In the **Sender** and **Receiver** fields, enter the logical system you configured, for example, ORACLETDS.

      You can click the icon to the right of each field to browse from a list of logical systems.

   b. In the **Message type** field, enter the message type you want to use, for example, MATMAS.

      You can click the icon to the right of each field to browse from a list of available message types.

8. Click the **check mark** icon to enter the information.

   You are returned to the main Change Distribution Model window.

9. Click Save.
Defining a Partner Profile

Partner profiles are a prerequisite for data exchange. This involves defining who can exchange messages with the MySAP ERP system and using which port.

1. Run the we20 transaction.

2. In the left pane, expand Partner type LS and select the logical system you configured from the list, for example, ORACLETDS.
   In the right pane, the Partn.number field refers to the name of the logical system.
3. Click Save.

4. From the Inbound parameters table, click the Create inbound parameter icon.

   The Partner profiles: Inbound parameters window is displayed.

   Partner profiles: Inbound parameters

   - Partner number: IVY_IN
   - Partner type: LE
   - Partner unit: 
   - Message type: MATMAS
   - Message code: 
   - Message function: 
   - Test: 
   - Inbound options: 
   - Postprocessing: permitted agent
   - Telephone: 

   Process code: MATM
   - Manual check
   - Processing by function module:
   - Trigger by background program
   - Trigger immediately

5. In the Message type field, enter the message type you want to use, for example, MATMAS.

   You can click the icon to the right of each field to browse from a list of available message types.
The Inbound options tab is selected by default.

6. In the **Process code** field, enter the process code you want to use, for example, MATM.
   
   You can click the icon to the right of each field to browse from a list of available process codes.

7. In the **Processing by function module** area, select one of the following options:
   - Trigger by background program.
     
     In this case the adapter writes IDocs to the MySAP ERP database, which is processed immediately.
   - Trigger immediately.
     
     In this case, the adapter waits for the MySAP ERP system to process IDocs. This can take anywhere from 1 to 15 minutes.

8. Click Save.

**Configuring MySAP ERP Outbound Processing**

Event creation must be implemented by you or by MySAP ERP. An event is created from specific application programs (the event creator) and then published systemwide. Any number of receivers can respond to the event with their own response mechanisms. An event is usually defined as a component of an object type.

MySAP ERP pseudo events are not processed by the MySAP ERP Event manager, but are called from an ABAP program or Remote Function Call (using the Destination parameter).

**Related Concepts and Terminology**

The following topic lists and defines specific terminology related to MySAP ERP and MySAP ERP event handling.

**Client and Server Programs**

RFC programs for non-MySAP ERP systems can function as either the caller or the called program in an RFC communication. There are two types of RFC programs:

- RFC Client
- RFC Server

The RFC client is the instance that calls the RFC to run the function that is provided by an RFC server. The functions that can be called remotely are called RFC functions, and the functions provided by the RFC API are called RFC calls.

**MySAP ERP Gateway**

The MySAP ERP Gateway is a secure application server. No connections are accepted unless they have been preregistered previously from the MySAP ERP presentation Client. A server connection presents itself to the Gateway and exposes a Program Identifier. If the Program Identifier is found in the list of registered Program IDs, the Gateway server then offers a connection to the server, which “Accepts” a connection. This ProgramID is then linked with an RFC Destination within MySAP ERP, which enables MySAP ERP Function Modules and ALE documents (IDocs or BAPI IDocs) to be routed to the destination. The RFC Destination functions as a tag to mask the Program ID to MySAP ERP users.
An RFC server program can be registered with the MySAP ERP gateway and wait for incoming RFC call requests. An RFC server program registers itself under a Program ID at a MySAP ERP gateway and not for a specific MySAP ERP system.

In SAPGUI, the destination must be defined with transaction SM59, using connection type T and Register Mode. Moreover, this entry must contain information on the MySAP ERP gateway at which the RFC server program is registered.

**Program IDs and Load Balancing**

If the Gateway Server has a connection to a particular server instance and another server instance presents itself to the gateway, then the gateway offers the connection and then begins functioning in Load Balancing mode. Using a proprietary algorithm, the Gateway sends different messages to each server depending on demand and total processing time. This may cause unpredictable results when messages are validated by schema and application.

When configuring multiple events in the Oracle Application Server using a single MySAP ERP program ID, MySAP ERP load balances the event data. For example, if multiple remote function calls or BAPIs use the same program ID (for example, ORACLETDS) and multiple MySAP ERP listeners are configured with this program ID, then MySAP ERP sends one request to one listener and the next to another listener, and so on.

There is a load-balancing algorithm present in the MySAP ERP Gateway Server. This mechanism is proprietary to MySAP ERP application development and might work by comparing total throughput of the connection, the number of times in wait state, and so on. One connection might receive nine messages and a second connection might receive one message. If five of the nine messages are rejected for schema validation and the one message on the other connection is rejected for schema validation, you might suspect that you are missing MySAP ERP event handling messages.

Load balancing in server (inbound to adapter from MySAP ERP) situations is handled by connecting multiple instances of the adapter to the MySAP ERP system. The MySAP ERP system will then load balance the connections. You cannot tune this performance.

Load balancing in client (outbound from adapter to MySAP ERP) situations is handled only by the MySAP ERP application design. If your system supports a Message Server, then you can load balance in client situations. If you have only one application server, you cannot load balance except by application server tuning, such as maximum number of connections permitted or time of day limits on connections.

The MySAP ERP system default limit is 100 RFC (communication) or adapter users. Each user takes up more than 2 MB of memory on the application server of the MySAP ERP system, and more or less on the adapter depending on the workload.

**Connection Pooling**

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

A connection pool can check which connections are no longer in use and can be closed to save system resources. The time period after which the pool checks the connections and the time after which a connection will time out can be configured by the calling application.
A pool is always bound to one user ID and password, meaning that all connections taken from this pool will also use these credentials. A MySAP ERP connection is always bound to an MySAP ERP user ID and a MySAP ERP Client number.

If you log on with a pool size that is set to 1, no connection pool is created (1 userid – 1 process thread). If you log on with a pool size that is greater than 1, a pool is created with a size of n, which is the number you specified.

For more information about connection pooling, see the SAP JCO API documentation.

Registering Your Program ID in SAPGUI

To enable your MySAP ERP system to issue the following calls or interfaces to the MySAP ERP event adapter, you must register your program ID under an RFC destination.

- Remote Function Calls (RFC)
- Business Application Programming Interfaces (BAPI)
- Intermediate Documents (IDoc)

The RFC destination is a symbolic name (for example, ORACLETDS) that is used to direct events to a target system, masking the program ID. The Program ID is configured in both SAPGUI and the event adapter.

Registering Your Program ID

To register your program ID:

1. Launch the SAP GUI and log in to the MySAP ERP system.
2. Select Tools, Administration, Network, and then RFC destination.
3. Run the SM59 transaction.
   The Display and maintain RFC destinations window is displayed.

4. Select TCP/IP connections and click Create.
   The RFC Destination window is displayed.
Provide the following information:

a. In the RFC destination field, enter a name, for example, ORACLETDS. The value you enter in this field is case sensitive.

b. In the Connection type field, enter T for destination type TCP/IP.

c. In the Description field, enter a brief description.

5. Click Save from the tool bar or select Save from the Destination menu.

The RFC Destination ORACLETDS window is displayed.

Perform the following steps:

a. For the Activation Type, click Registration.

b. In the Program field, enter ORACLETDS.

6. Click Save from the tool bar or select Save from the Destination menu.

7. Ensure your event adapter is running.

8. Verify that the MySAP ERP system and OracleAS Adapter for MySAP ERP are communicating.

9. Click TestConnection.

**Testing the MySAP ERP Event Adapter**

In the SAP Server, the SE37 transaction enables you to send an RFC (Remote Function Call) or a BAPI (Business Application Programming Interface) to any RFC destination.
For more information on RFC destination, see Registering Your Program ID in SAPGUI on page A-11.

**Testing the MySAP ERP Event Adapter by Sending an RFC or a BAPI Manually**

To test the MySAP ERP event adapter:

1. In the Function Builder, select a function module, for example, RFC_CUSTOMER_GET.

   ![Function Builder: Initial Screen](image)

2. To choose single test, press F8 and click the Single Test icon or choose Function module, select Test and then Single Test.

3. Enter an RFC target system, for example, ORACLETDS.

4. Enter input data for the particular RFC modules, for example, AB*.

5. To execute, press F8.

   The Test Function Module: Initial Screen window is displayed.

   ![Test Function Module: Initial Screen](image)

6. Enter data into the SAP GUI and click Execute.

   The function name and input data are transferred through RFC to create an XML document on the Oracle Application Server with the parameters input in SAPGUI.
Application Link Embedding Configuration for the Event Adapter

The MySAP ERP event adapter receives IDocs (Intermediate Documents) from MySAP ERP. To configure an MySAP ERP system to send IDocs to the MySAP ERP event adapter, use the ALE (Application Link Embedding) configuration to:

1. Register your program ID in SAP GUI.
2. Define a port.
3. Create a logical system.
4. Create a partner profile.
5. Create a distribution model for the partner and message type.
6. Test the MySAP ERP event adapter.

Defining a Port

A port identifies where to send messages. This port can be used only if an RFC destination was created previously.

Defining a Port

To define a port:

1. In the ALE configuration, choose Tools, Business Communications, IDocs Basis, IDoc, and then Port Definition.

   You can also run the WE21 transaction.

   The Creating a tRFC port window is displayed.

2. In the left pane under Ports, select Transactional RFC and click Create.
3. Select Generate port name.

   The system generates the port name.
4. Enter the IDoc version you want to send through this port.
5. Click the destination you created, for example, ORACLETDS.
6. Save the session, making note of the system-generated RFC port.

Creating a Logical System

One type of partner is a logical system. A logical system manages one or more RFC destinations.
Creating a Logical System

To create a logical system called ORACLETDS:

1. In the ALE configuration, enter the area menu selection SALE transaction.
2. Select SAP Reference IMG.
3. Expand the following nodes: Basis Components, Application Link Enabling (ALE), Sending and Receiving Systems, Logical Systems, and Define Logical System.
4. Click the check mark beside Define Logical System.

The Change View "Logical Systems": Overview window displays a list of logical systems and their names.

5. Click New entries.

The New Entries: Overview of Added Entries window is displayed with Log.System and Name columns for new log system.

6. Type an entry for Log System, for example, ORACLETDS.
7. In the Name column, enter a name (description) for the partner profile.
8. Click Save to save the session.
Creating a Partner Profile

A partner profile is a definition of parameters for the electronic interchange of data with a trading partner using the IDoc interface.

To communicate with a partner using the IDoc interface, you must create a partner profile.

Creating a Partner Profile

To create a partner profile:

1. In SAP GUI, choose Tools, Business Communication, IDoc Basis, and Partner profile.
   
   You can also run the WE21 transaction.

   The Partner profiles: Outbound parameters window is displayed and shows fields for specifying details for the partner profile.

   Perform the following steps:

   a. Select Partner type LS (Logical system).
   b. Press F5 (Create).

2. For Type, enter USER.

3. For Agent, enter the current user ID, or you may select another agent type.

4. Under the outbound parameter table control, select Create outbound parameter.

   Partner type is LS, and the Message type is DEBMAS, which is the IDoc document type.

5. Leave Partn.funct blank.

6. Click the Outbound options tab.

   Provide the following information:

   a. Depending on your performance requirements, click Transfer IDoc Immed or Collect IDocs.
   b. For the IDoc, enter a message type, for example, DEBMAS.
   c. Enter a receiver port, for example, A000000036.

7. Click Save to save the session.
The Partner profiles summary window is displayed. It contains information for the logical system that you created.

Collected IDocs

When using collected IDocs on any platform during inbound processing (service mode), if the DOCNUM field does not have a unique document number for each IDoc, the system creates an IDoc for each header record in the collected IDoc file and duplicates the data for each IDoc.

Make sure the DOCNUM field is included in the EDI_DCI40 structure and that each IDoc has a unique sequence number within the collected IDoc file.

Creating a Distribution Model for the Partner and Message Type

You must create a distribution model for the partner and message type you designated.

Creating a Distribution Model

To create a distribution model called ORAMOD:

1. In SAP GUI, choose Tools, AcceleratedSAP, Customizing, and then Project Management.

   You can also run the BD64 transaction.

   The Display Distribution Model window is displayed.

2. Select Create model view.

   If required, switch the processing mode to edit within Distribution Model/Switch Processing Mode.

3. Enter a short text string and a technical name for your new model view.

4. Click Save.

   The Distribution Model Changed window is displayed, showing a tree structure of the distribution model.
Perform the following steps:

a. In the Distribution Model tree, select a new model view.

b. On the right, select Add message type.

The Add Message Type box is displayed. It contains fields for specifying the sender and receiver of the message and the message type.

Provide the following information:

a. In the **Sender** field, provide the sender that points to the MySAP ERP system, which sends the IDoc, for example, **I46_CLI800**.

   In this case, the sender is an SAP 4.6B system.

b. In the **Receiver** field, provide the logical system, for example, **ORACLETDS**.

c. In the **Message type** field, provide the type of IDoc, for example, **DEBMAS**.

5. Click the **check mark** icon.

6. Click Save.

The Change Distribution Model window displays the new model view to use to send message type, **DEBMAS**, from the **I46_CLI800** SAP system to the **ORACLETDS** logical system.
You are now ready to test the connection to the logical system.

**Testing the MySAP ERP ALE Configuration**

In the SAP Server, the BD12 transaction enables you to send IDocs to any logical system, for example, to an event adapter.

**Testing the MySAP ERP ALE Configuration**

To test the MySAP ERP Application Link Embedding (ALE) configuration:

1. In the Send Customers window, enter the IDoc message type, for example, DEBMAS in the **Output type** field.

2. In the **Logical system** field, enter the logical system, for example, ORACLETDS.

3. Click Run.

The MySAP ERP event adapter receives the IDoc in XML format. No response is expected from the event adapter.

A confirmation window is displayed.
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.
ABAP programs, A-9
access rights, 8-1
Activation Type, A-12
adapter configuration
  overwriting, 4-5
  updating, 4-2
adapter connections, 4-4
adapter exceptions, 7-4 to 7-6
Adapter Lib Directory parameter, 3-3 to 3-4
adapter types
  resource, 1-1, 4-1 to 4-2
AdapterName parameter, 4-4 to 4-5
adapters
  configuring, 3-1 to 3-26, 4-2
  deploying, 1-4, 4-1 to 4-2, 5-2
  integrating with BPEL Process Manager, 5-1
  troubleshooting, 7-1 to 7-7
Adapters node, 5-23
Add Channel dialog box, 3-23
Add Message Type dialog box, A-6
Add Target dialog box, 3-11
Admin Password parameter, 3-3, 3-4, 3-5
Admin User parameter, 3-4, 3-5
Advanced tab, 3-14, 3-25, 5-22
ae batch script, 7-1
ALE (Application Link Embedding) documents, A-9, A-14
application adapters
  configuring, 3-1 to 3-26, 4-2
  deploying, 4-1 to 4-2, 5-2
  integrating with BPEL Process Manager, 5-1
  troubleshooting, 7-1 to 7-7
application clients, 4-4
Application Explorer, 1-2, 5-2
  application systems and, 3-11
  channels and, 3-23
  debugging and, 7-1 to 7-3
J2CA configuration and, 4-2
OracleAS Adapter J2CA and, 7-4
schemas and, 3-19
security and, 8-1, 8-6
testing and, 7-1 to 7-3
troubleshooting, 7-2 to 7-3
WSDL files and, 5-9, 5-20
Application Link Embedding (ALE) documents, A-9, A-14
Application Server dialog box, 3-12
Application Server parameter, 3-13
application systems
  Application Explorer and, 3-11
  supported, 3-11
Applications tab, 5-10, 5-25
Applications-Navigator pane, 5-17, 5-29
Assign activities, 5-15
Assign Activity-Navigator pane, 5-15
Assign activity icon, 5-15
Available list, 8-4 to 8-5
back-end workflows
calling, 5-18
BAPI (Business Application Programming Interfaces), 7-4
  registering, A-11
BAPI Exception parameter, 3-14, 3-25, 5-22
BAPI requests, 1-4
BPEL Console, 5-18 to 5-20, 5-31
  starting, 5-18
BPEL Designer, 5-1
  BPEL processes and, 5-17
BPEL domain
  passwords and, 5-31
BPEL PM Server, 5-17
BPEL Process Manager
  adapter request-response service and, 5-18
  integrating with adapters, 5-1
  OracleAS Adapter for MySAP ERP and, 5-1
BPEL Process Project dialog box, 5-11, 5-26
BPEL processes, 5-3 to 5-8
deploying, 5-17
designing, 5-12
managing, 5-19
monitoring, 5-18
testing, 5-19
BPEL Processes tab, 5-18
BPEL projects
creating, 5-3 to 5-8
BSE (OracleAS Adapter Business Services Engine), 1-1, 1-4
configuring, 3-2 to 3-3
connection access to, 8-6
troubleshooting, 7-5
BSE configuration page, 3-2 to 3-3
BSE control service URL, 8-8
BSE repositories
migrating, 8-8
BSE settings window, 3-3
BSE system settings, 3-3 to 3-6
BSE URL field, 3-8
Business Application Programming Interfaces (BAPI), 7-4
registering, A-11
business events, 1-1
business functions, 3-11, 3-19 to 3-22
creating schemas for, 3-19
business processes, 1-3
business services
creating, 3-21 to 3-22
deploying, 8-1
testing, 3-22
Business Services node, 8-2 to 8-5, 8-6

C
CCI (Common Client Interface), 4-1
CCI calls, 4-1
Change Distribution Model window, A-6
channels, 3-23
Application Explorer and, 3-23
creating, 3-23, 5-20 to 5-23
deleting, 3-26
editing, 3-26
starting, 3-26, 5-23
stopping, 3-26, 5-23
testing and debugging and, 5-23
Channels node, 3-23, 5-20
channels. See also listeners
Client parameter, 3-12, 3-25, 5-22
client programs
RFC Client, A-9
Code page parameter, 3-12, 3-25, 5-22
Commit with wait parameter, 3-14, 3-25, 5-22
Common Client Interface (CCI), 4-1
Compiler tab, 5-18
Configuration node under Business Services, 8-2 to 8-5, 8-6
configuration parameters, 4-3, 4-5
IWayConfig, 4-3
IWayHome, 4-3
IWayRepoPassword, 4-3
IWayRepoURL, 4-3
IWayRepoUser, 4-3
Loglevel, 4-3
configurations
connecting to, 3-10
defining, 3-8 to 3-9
overwriting, 4-3 to 4-5
Configurations node, 3-8 to 3-9
configuring adapters, 3-1 to 3-26, 4-2
configuring BSE system settings, 3-3 to 3-6
configuring repositories, 3-6
connecting to MySAP ERP, 3-11 to 3-17, 5-9,
5-23 to 5-24, 7-2, 7-7
connecting to OracleAS Adapter J2CA, 7-4
Connection dialog box, 3-17, 5-24
connection factories, 4-4
connection parameters, 3-12 to 3-17, 3-24 to 3-25, 5-3,
5-28, 7-2
AdapterName, 4-4
Advanced, 3-14, 3-25, 5-22
Application Server, 3-13
BAPI Exception Handling, 3-14
Client, 3-12, 3-25, 5-22
Code page, 3-12
Commit with wait, 3-14
Config, 4-4
Connection pool name, 3-14
Connection pool size, 3-14
Country, 4-4
EDI Version, 3-13
Language, 3-12, 4-4
Loglevel, 4-4
Logon ticket (SSO2), 3-15
Logon ticket (X509), 3-15
MySAP ERP Trace, 3-12
Password, 3-12, 3-25, 4-4, 5-22
Port, 3-2 to 3-3
Security, 3-15
SNC level, 3-15
SNC library path, 3-16, 3-25, 5-23
SNC mode, 3-15
SNC name, 3-15
SNC partner, 3-15
System, 3-13, 3-24, 5-21
System Number, 3-13
User, 3-12, 3-25, 5-22
UserName, 4-4
Connection pool name parameter, 3-14, 3-25, 5-22
Connection pool size parameter, 3-14, 3-25, 5-22
connection pooling, 4-4
Connection type field, A-11
connections
closing, 3-17
deleting, 3-17
establishing, 3-11 to 3-17, 5-24
ConnectionSpec, 4-4 to 4-5
Connector
deploying to Oracle Application Server, 4-1
connector factories, 4-2
connector factory objects, 4-2
multiple, 4-3
control methods, 8-9
copy rules
creating, 5-16
Copy Rules tab, 5-15
Country parameter, 4-4
Create Copy Rule dialog box, 5-16
Create Model View dialog box, A-5
Create Outbound JCA Service
Create Partner Link dialog box, 5-12, 5-27
Create Request dialog box, A-3
Create Variable dialog box, 5-15, 5-29
Create Web Service dialog box, 3-21
creating BPEL processes, 5-3 to 5-8
creating channels, 5-20 to 5-23
creating copy rules, 5-16
creating events, 5-23 to 5-25
creating repository projects, 3-8 to 3-9
customer master data, 5-30
Dashboard tab, 5-18
data
  manipulating, 5-18
data exchange
  partner profiles and, A-7
database connections
  opening, 4-3
databases
  connecting to, 4-3
  Oracle, 4-3
DEBMAS05
  locating, 5-23
Debug Level parameter, 3-4
DEBUG log level, 4-5
deploying adapters, 4-1 to 4-2, 5-2
deploying outbound processes, 5-17
Description field, 3-11, 3-21, 8-3 to 8-5, 8-7, A-11
design time, 5-20, 8-8
  configuring, 5-2
design time service adapter connections, 4-4
designing BPEL processes, 5-3 to 5-8, 5-12
Destination parameter, A-9
disconnecting from MySAP ERP, 3-17
Display and maintain RFC Destination window, A-11
Display Distribution Model window, A-4
Display IMG window, A-2
DNS Lookup option, 8-7
DNS name, 8-7
Domain Name System (DNS), 8-7
domain names, 8-7
Domain option, 8-7
Domain Password field, 5-17
Eclipse. See JDeveloper
EDI Version parameter, 3-13
Edit Invoke dialog box, 5-15
Edit Receive dialog box, 5-28
editing channels, 3-26
editing targets, 3-18
EIS (enterprise information systems), 4-5
EJB (Enterprise Java Beans), 4-1
Encoding parameter, 3-4
Enterprise Connector for J2EE Connector Architecture (J2CA), 1-4
enterprise information systems (EIS), 4-5
Enterprise Java Beans (EJB), 4-1
error messages, 5-15, 5-18, 7-2 to 7-7
target systems and, 7-6
ESB Inbound Process, 6-32
ESB Outbound Process, 6-2
event adapters
  configuring, 3-23
testing, A-12
event data
  receiving, 5-20, 5-31
event handling, A-1, A-9
event integration, 5-20, 5-31
  verifying, 5-30
Event Manager, A-9
event messages, 5-20, 5-31
event ports
  creating, 5-23 to 5-25
events, 1-1, 3-23, A-1
  configuring, 3-23
  creating, 5-23 to 5-25, A-9
  triggering, 5-30
Execution Denied list, 8-6
Execution Granted list, 8-6
Existing Service Names list, 3-21
Export WSDL dialog box, 5-24
fault code elements, 7-6
fault string elements, 7-6
file system repositories
  configuring, 3-6
Function Builder, A-13
functions, A-9
Gateway Server, A-10
Grant Access check box, 8-7
Group (of Computers) option, 8-7
Group node, 8-4
groups
  creating, 8-3
Home field, 3-9
Hostname parameter, 3-1 to 3-3, 3-9
IDoc (Intermediate Documents), 7-4, A-1
defining, A-1
registering, A-11
saving, A-1
distributing, A-1
IDoc interface, A-1
IDoc requests, 1-4
in, 3-25
inbound IDoc
   configuring, A-1
inbound interactions, 5-25
inbound J2CA services, 5-20
inbound MySAP ERP processing, A-1
   configuring, A-1
Inbound options tab, A-8
Initiate tab, 5-19
input records
   creating, 4-5
input XML documents, 7-4
installation directories, 4-3
instances of policy types, 8-1
Instances tab, 5-31
interactions
   creating, 4-5
   executing, 4-5
   inbound, 5-25
Intermediate Documents (IDoc), 7-4, A-1
   defining, A-1
   registering, A-11
   saving, A-1
   transferring, A-1
Invalid Settings warning, 5-15
Invoke activities, 5-14
IP (Mask)/Domain field, 8-7
IP addresses, 8-7
IP and Domain Restriction policy type, 8-6
IWAE function, 4-5
IWAFConnectionSpec, 4-4 to 4-5
IWAFInteractionSpec, 4-5
IWayConfig parameter, 4-3
IWayHome parameter, 4-3
IWayRepoPassword parameter, 4-3
IWayRepoURL parameter, 4-3
IWayRepoUser parameter, 4-3
iwse.ora file, 3-6

J

J2CA (Enterprise Connector for J2EE Connector Architecture), 1-4
   OracleAS Adapter and, 4-1
J2CA configuration
   Application Explorer and, 4-2
J2CA repositories
   migrating, 8-10
J2CA resource adapters, 1-1, 4-1
J2CA services, 5-20
Java application clients, 4-4
Java program clients, 4-1
JDeveloper, 5-1
   BPEL processes and, 5-17
JNDI lookup, 4-4

L

Language parameter, 3-4, 3-12, 3-25, 4-4, 5-22
librfc32.dll file, 7-2
License and Method dialog box, 3-21
License field, 3-21
licenses, 3-21
list of nodes, 7-2
listeners, 1-4, 3-11, A-10
listeners. See also channels
load balancing, A-10
load balancing algorithms, A-10
locating DEBMAS05, 5-23
log files, 7-1
log levels
   overwriting, 4-3 to 4-5
logical systems, 5-30, A-1 to A-6
   configuring, A-1
   defining, A-2
   unique IDs, A-1 to A-2
Loglevel parameter, 4-3 to 4-4
Logon ticket (SS02) parameter, 3-15
Logon ticket (X509) parameter, 3-15
Log.System column, A-3

M

managed connector factories, 4-2
managed connector factory objects, 4-2
   multiple, 4-3
ManagedConnectionFactory parameter, 4-3 to 4-5
manipulating data, 5-18
mapping security, 4-5
mapping variables, 5-16
mappings
   verifying, 5-16
Message Server dialog box, 3-23
Message Type field, A-8
message types
   error, 5-15, 5-18
   event, 5-20, 5-31
   warning, 5-15, 5-18
messages, 1-1
   logging, 5-17
Messages log area, 5-17
metadata
   storing, 3-5, 8-8
Method Name field, 3-21
methods, 8-1
migrating repositories, 8-8
migrating Web services, 8-10
My SAP ERP System port, A-1
MySAP Easy Access window, 5-30
MySAP ERP, 4-5
   connecting to, 1-3, 3-11 to 3-17, 5-9, 5-23 to 5-24,
   7-2, 7-7
   disconnecting from, 3-17
   integrating, 1-3
MySAP ERP business functions, 3-11, 3-19 to 3-22
   creating schemas for, 3-19
MySAP ERP business processes, 1-3
MySAP ERP event adapters
   configuring, A-14
   testing, A-13
MySAP ERP event handling, A-9 to A-10
MySAP ERP Event Manager, A-9
MySAP ERP function modules
  routing, A-9
MySAP ERP Gateway server, A-9
MySAP ERP inbound processing, A-1
MySAP ERP outbound processing, A-9
MySAP ERP platforms, 1-3
MySAP ERP process link, 5-19
MySAP ERP processes
  testing, 5-18
MySAP ERP program ID (identifiers)
  BAPI and, A-10
  multiple remote function calls and, A-10
  registering, A-11 to A-12
  RFC Destination and, A-9
MySAP ERP Trace parameter, 3-12, 3-25, 5-22
MySAP node, 5-23 to 5-24
MySAP program ID (identifiers), 5-30
MySAP Workbench, 5-30

N
Name field, 3-11, 8-3 to 8-5
Namespace field, 5-11, 5-26
New Configuration dialog box, 3-8 to 3-9
New Gallery window, 5-10, 5-25
New Group dialog box, 8-4
New Policy permissions dialog box, 8-5
New User dialog box, 8-2
Node list, 7-2
nodes
  Adapters, 5-23
  Business Services, 8-2 to 8-5
  Channels, 3-23, 5-20
  Configuration under Business Services, 8-2 to 8-5, 8-6
  Configurations, 3-8 to 3-9
  connected, 3-17, 5-24
  disconnected, 3-18
  Group, 8-4
  MySAP, 5-23 to 5-24
  Policies, 8-5
  Ports, 5-20
  Process, 5-16
  Security, 8-5, 8-6
  Users, 8-3
  Users and Groups, 8-2 to 8-3
Number of Async. Processors parameter, 3-4

O
Object type, 7-4
OC4J (Oracle Application Server Containers for J2EE), 4-1
deploying, 4-1 to 4-6
Oracle BPEL Console, 5-18 to 5-20, 5-31
starting, 5-18
Oracle databases, 4-3
Oracle JDeveloper, 5-1
BPEL processes and, 5-17
Oracle JDeveloper BPEL Designer. See BPEL Designer, JDeveloper, or Oracle JDeveloper
Oracle repositories
  migrating, 8-8
OracleAS Adapter
  installation directory and, 4-3
  J2CA and, 4-1
OracleAS Adapter Application Explorer. See Application Explorer
OracleAS Adapter Business Services Engine (BSE), 1-1, 1-4
  configuring, 3-2 to 3-3
  troubleshooting, 7-5
OracleAS Adapter for MySAP ERP
  BPEL Process Manager and, 5-1
  configuring, 3-1 to 3-26
  deploying, 1-1, 5-2
  integrating with MySAP ERP, 5-20
  troubleshooting, 7-1 to 7-7
OracleAS Adapter J2CA, 4-5
  Application Explorer and, 7-4
  connecting to, 7-4
  outbound integration, 5-9 to 5-18
  outbound interaction, 5-18
  outbound MySAP ERP processing, A-1 to A-9
  configuring, A-9
  outbound processes
    deploying, 5-17
    JDeveloper and, 5-17
  parameter types
    channel configuration, 3-26
    configuration, 4-3
    connection, 3-24 to 3-25, 4-4, 5-3, 5-28, 7-2
    repository, 3-5
    repository migration, 8-9
    security, 3-4, 3-5
    system, 3-4
  partner links, 5-12 to 5-13, 5-27 to 5-28
  partner profiles
    data exchange and, A-7
    defining, A-7
    Partner profiles window, A-7
    Partner type LS, A-7
    Partn.number field, A-7
    Password parameter, 3-2 to 3-3, 3-12, 3-25, 4-4 to 4-5, 5-22, 8-3
    Password Prompt dialog box, 5-17
  passwords, 3-2 to 3-3, 3-12, 3-25, 4-3 to 4-5, 5-17, 5-22, 5-29, 8-3
    BPEL domain and, 5-31
  permissions, 8-1
configuring, 7-4
denying, 8-6
granting, 8-6
policies, 8-1
applying, 8-1
creating, 8-4
Policies node, 8-5
Policy parameter, 3-4, 3-5
policy types
instances of, 8-1
IP and Domain Restriction, 8-6
policy-based security, 8-1 to 8-7
port definitions, A-1
Port Number parameter, 3-9
Port parameter, 3-2 to 3-3
ports
creating, 5-23 to 5-25
Ports node, 5-20
privileges, 8-1
setting, 8-1
process activities, 5-14
Process Activities pane, 5-12 to 5-13, 5-27, 5-28
Process code field, A-8
PROCESS function, 4-5
Process Manager. See BPEL Process Manager
Process node, 5-16
processes
designing, 5-12
Processing by function module area, A-9
Program field, A-12
program ID (identifiers), 5-30
 BAPI and, A-10
 multiple remote function calls and, A-10
 registering, A-11 to A-12
 RFC Destination and, A-9
projects
 BPEL, 5-3 to 5-8
Prompt for Workbench request dialog box, A-3
properties, 4-3
pseudo events
 Remote Function Calls (RFC) and, A-9
repository information
 storing, 3-6
repository migration parameters, 8-9
repository parameters
 Driver, 3-5
Password, 3-5
Pooling, 3-5
Type, 3-5
URL, 3-5
User, 3-5
Repository Password parameter, 3-5
Repository Pooling parameter, 3-5
repository projects
creating, 3-8 to 3-9
Web services and, 3-8
repository tables
creating, 3-6
Repository Type parameter, 3-5
Repository URL parameter, 3-3, 3-5
repository URLs, 8-10
Repository User parameter, 3-5
repository.xml file, 8-11
request processing
BAPI, 1-4
IDoc, 1-4
RFC, 1-4
request schemas, 3-19
request-response services, 5-18
 creating, 5-9 to 5-18
requests
 executing, 7-4
resource adapters, 4-1 to 4-2
deploying, 1-1
Resource Execution policy type, 8-1
response schemas, 3-19
Restrictions page, 5-30
RFC (Remote Function Calls)
 pseudo events and, A-9
 registering, A-11
RFC API
 RFC calls and, A-9
RFC Client program, A-9
RFC Destination, A-9 to A-11
RFC Destination field, A-11
RFC Destination window, A-11
RFC functions, A-9
RFC programs, A-9
 Client, A-9
 Server,  A-9
RFC requests, 1-4
RFC target systems, A-13
RMI event ports, 5-20
runtime, 8-8
runtime service adapter connections, 4-4
S
SAP GUI, A-11, A-14
sapjco.jar file, 7-2 to 7-3
sapjorfc.dll file, 7-2
Index-6
schemas, A-10
  Application Explorer and, 3-19
  creating, 3-19
  storing, 3-9
SE37 transaction, A-12
security, 8-1 to 8-7
  Application Explorer and, 8-1, 8-6
  configuring, 8-2
  mapping, 4-5
Security node, 8-2 to 8-5, 8-6
security parameters
  Admin Password, 3-4, 3-5
  Admin User, 3-4, 3-5
  Policy, 3-4, 3-5
security policies
  applying, 8-1
  creating, 8-4
Security Policy option, 8-6
Security tab, 3-15
Selected list, 8-4 to 8-5
Send Customers window, 5-30
server programs
  RFC Client, A-9
  RFC Server, A-9
service adapter connections, 4-4
service integration, 5-9 to 5-18
service names, 3-21
Service Provider list, 3-8 to 3-9
Service-Oriented Architecture (SOA), 5-1
services, 1-1
  creating, 3-21 to 3-22
  testing, 3-22
servlets, 4-1
Single (Computer) option, 8-7
SM59 transaction, A-9 to A-11
SNC level parameter, 3-15
SNC library path parameter, 3-15
SNC mode parameter, 3-15
SNC name parameter, 3-15
SNC partner parameter, 3-15
SOA (Service-Oriented Architecture), 5-1
SOAP agents, 7-6
SOAP envelopes, 8-9
SOAP faults, 7-6
soap operation name dialog box, 8-8
SOAP requests, 7-6 to 7-7, 8-1, 8-5
  creating, 8-8
  errors and, 7-6 to 7-7
  storing metadata, 8-8
System Number parameter, 3-13
system parameters
  Adapter Lib Directory, 3-4
  Debug Level, 3-4
  Encoding, 3-4
  Language, 3-4
  Number of Async. Processors, 3-4
system settings
  configuring, 3-3 to 3-6
  System tab, 3-13, 3-24, 5-21
T
target systems
  errors and, 7-6
Target Type list, 3-12
targets
  connecting to, 3-17, 5-24, 7-7
  defining, 3-11
  deleting, 3-17
  disconnecting from, 3-17
  editing, 3-18
Template list, 5-11, 5-26
Test Function Module Initial Screen window, A-13
testing event adapters, A-13
testing Web services, 3-22
trace information, 7-1
transaction processing, 1-1
transactions
  calling, 5-18
  storing, 3-5
trigger by background program option, A-9
trigger immediately option, A-9
triggering events in MySAP ERP, 5-30
troubleshooting, 7-1 to 7-7
  Application Explorer, 7-2 to 7-3
  BSE, 7-5
  Web services, 7-5 to 7-7
Type list, 8-7
U
unique IDs, A-1
updating adapter configuration, 4-2
User parameter, 3-2, 3-12, 3-25, 5-22, 7-4
User tab, 3-12, 3-25, 5-22
UserName parameter, 4-4 to 4-5
users
  associating, 8-2
Users and Groups node, 8-2 to 8-3
Users node, 8-3
V
variables
  mapping, 5-16
  verifying mappings, 5-16
visual editors, 5-12 to 5-14, 5-15, 5-27
W
warning messages, 5-15
Web Service Definition Language (WSDL), 3-19 to 3-20
Web service names, 3-21
Web service projects
  creating, 3-8
Web services, 1-1
  creating, 3-21 to 3-22
  delivering, 3-5
  deploying, 7-5 to 7-7, 8-1
  integrating, 5-1
migrating, 8-10
repository projects and, 3-8
testing, 3-22
troubleshooting, 7-5 to 7-7
Web services policy-based security, 8-1 to 8-7
workspaces, 5-10, 5-25
WSDL (Web Service Definition Language), 3-19 to 3-20
WSDL Chooser dialog box, 5-12
WSDL documents, 5-1
WSDL file location dialog box, 8-8
WSDL files, 5-1
Application Explorer and, 5-9, 5-20
WSIL browser icon, 5-12

X

XML documents
input, 7-4
XML messages, 1-1
XML schemas
creating, 3-19
storing, 3-9