Oracle® Fusion Middleware
Application Adapter for Siebel User's Guide for Oracle WebLogic Server
11g Release 1 (11.1.1)
E17056-01

April 2010
Preface

Audience................................................................................................................................................vii
Documentation Accessibility ......................................................................................................................vii
Related Documents ...................................................................................................................................vii
Conventions .............................................................................................................................................viii

1 Introduction

Adapter Features.......................................................................................................................................1-1
The Siebel Application Model ..................................................................................................................1-4
Integration with Siebel ..........................................................................................................................1-4

Integrating with Siebel EAI Architecture .........................................................................................1-5
Using Application Explorer with Oracle Application Adapter for Siebel ........................................1-5
BSE Versus Oracle Adapter J2CA Deployment ....................................................................................1-6

2 Configuring Oracle Application Server Adapter for Siebel

Starting Application Explorer ..................................................................................................................2-2
Configuring Repository Settings ............................................................................................................2-3
Configuring the Database Repository for BSE .....................................................................................2-3
Configuring the Database Repository for J2CA ..................................................................................2-7

Creating a Repository Configuration......................................................................................................2-9
Creating a Configuration for BSE ...........................................................................................................2-9
Creating a Configuration for J2CA .........................................................................................................2-10
Connecting to a BSE or J2CA Configuration .......................................................................................2-12

Establishing a Connection (Target) for Siebel .....................................................................................2-12
Defining a Target to Siebel ....................................................................................................................2-13
Connecting to a Defined Target ............................................................................................................2-16
Disconnecting From Siebel ....................................................................................................................2-16
Editing a Target ......................................................................................................................................2-16
Deleting a Target to Siebel ....................................................................................................................2-17

Viewing Application System Objects ....................................................................................................2-17
Viewing Metadata ....................................................................................................................................2-17

Creating XML Schemas .........................................................................................................................2-18

Siebel Prerequisites for Working With Integration Objects ....................................................................2-22
Creating Schemas for Siebel Integration Objects ..................................................................................2-22
Creating Integration Object (IO) Nodes for Siebel ...............................................................................2-25
<table>
<thead>
<tr>
<th>3</th>
<th>Oracle WebLogic Server Deployment and Integration</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Adapter Integration with Oracle WebLogic Server</td>
</tr>
<tr>
<td></td>
<td>Deployment of Adapter</td>
</tr>
<tr>
<td></td>
<td>Updating Adapter Configuration</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4</th>
<th>Integration With BPEL Service Components in the Oracle SOA Suite</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Overview</td>
</tr>
<tr>
<td></td>
<td>Deployment of Adapter</td>
</tr>
<tr>
<td></td>
<td>Configuring a New Application Server Connection</td>
</tr>
<tr>
<td></td>
<td>Designing an Outbound BPEL Process for Service Integration</td>
</tr>
<tr>
<td></td>
<td>Siebel Service Integration</td>
</tr>
<tr>
<td></td>
<td>Creating a New SOA Application for the Outbound BPEL Process</td>
</tr>
<tr>
<td></td>
<td>Defining a BPEL Outbound Process</td>
</tr>
<tr>
<td></td>
<td>Deploying the BPEL Outbound Process</td>
</tr>
<tr>
<td></td>
<td>Invoking the Input XML Document in the Oracle Enterprise Manager Console</td>
</tr>
<tr>
<td></td>
<td>Testing Outbound BPEL and Mediator Processes</td>
</tr>
<tr>
<td></td>
<td>Designing an Inbound BPEL Process for Event Integration</td>
</tr>
<tr>
<td></td>
<td>Generating WSDL for Event Integration</td>
</tr>
<tr>
<td></td>
<td>Creating a New SOA Application for the Inbound BPEL Process</td>
</tr>
<tr>
<td></td>
<td>Defining a BPEL Inbound Process</td>
</tr>
<tr>
<td></td>
<td>Deploying the BPEL Inbound Process</td>
</tr>
<tr>
<td></td>
<td>Triggering an Event in Siebel</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>5</th>
<th>Integration With Mediator Service Components in the Oracle SOA Suite</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Configuring a New Application Server Connection</td>
</tr>
<tr>
<td></td>
<td>Configuring a Mediator Outbound Process</td>
</tr>
<tr>
<td></td>
<td>Creating a New SOA Application for the Outbound Mediator Process</td>
</tr>
<tr>
<td></td>
<td>Defining a Mediator Outbound Process</td>
</tr>
<tr>
<td></td>
<td>Deploying the Mediator Outbound Process</td>
</tr>
<tr>
<td></td>
<td>Invoking the Input XML Document in the Oracle Enterprise Manager Console</td>
</tr>
<tr>
<td></td>
<td>Configuring a Mediator Inbound Process</td>
</tr>
<tr>
<td></td>
<td>Creating a New SOA Application for the Inbound Mediator Process</td>
</tr>
<tr>
<td></td>
<td>Defining a Mediator Inbound Process</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>6</th>
<th>Troubleshooting and Error Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Troubleshooting</td>
</tr>
<tr>
<td></td>
<td>BSE Error Messages</td>
</tr>
<tr>
<td></td>
<td>General Error Handling in BSE</td>
</tr>
</tbody>
</table>
Preface

Oracle Fusion Middleware Application Adapter for Siebel User’s Guide for Oracle WebLogic Server describes how to provide connectivity and integrate with Siebel systems.

Audience


Documentation Accessibility

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

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

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

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

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

- *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter Upgrade Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter Best Practices Guide for Oracle WebLogic Server*
- Oracle's Unified Method (OUM)

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

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

Conventions

The following text conventions are used in this document:

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

1.1 Adapter Features

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

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

Oracle Application Adapter for Siebel uses XML messages to enable non-Siebel applications to communicate and exchange transactions with Siebel using services and events. Services and events are defined as follows:

- Services (also known as outbound processing): Enables applications to initiate a Siebel business event.
- Events (also known as inbound processing): Enables applications to access Siebel data only when a Siebel business event occurs.

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

The channel is the adapter component that receives events in real time from the EIS application. The channel component can be a File reader, an HTTP listener, or an MQ 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.
- Supports synchronous and asynchronous, bidirectional message interactions for Siebel Business Services, Business Components, and Integration Objects.

- Includes Oracle WebLogic Server Adapter Application Explorer (Application Explorer), a GUI tool that uses the Siebel Object Manager to explore Siebel metadata and build XML schemas or Web services.

- Supports Siebel transports—MQSeries, File, and HTTP. It also supports MSMQ messaging.

- XML schemas for Oracle Adapter J2CA.

- Web services for BSE.

Oracle Application Adapter for Siebel supports all 23 Siebel Industry Applications (SIA) through business objects, business components, business services, and integration objects. Siebel Industry Applications include industry verticals such as insurance, high technology, automotive, communications, media, financial services, life sciences, manufacturing, and consumer goods.

Siebel Industry Applications is tailored to the specific business requirements and processes of a particular industry with additional business logic in the form of business objects, business components, business services, and integration objects. Oracle Application Adapter for Siebel exposes and generates metadata and interacts with these industry-specific objects.

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

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

*Figure 1–1* shows the generic architecture for the Oracle Web service adapter for packaged applications. The adapter works with BSE, as deployed to a Web container in a J2EE application server. BSE serves as host to the adapters, enabling Web service requests to the adapters.

Application Explorer, a design-time tool deployed along with BSE, is used to configure adapter connections, browse EIS objects, and configure services. Metadata created while you perform these operations are stored in the repository by BSE.

BSE uses SOAP as a protocol for receiving requests from clients, interacting with the EIS, and sending responses from the EIS back to clients.
Oracle Adapter J2CA Generic Architecture

Figure 1–2 shows the generic architecture for the Oracle Adapter J2CA for packaged applications. The Oracle Adapter J2CA is deployed to a standard J2CA container and serves as host container to the adapters. The connector is configured with a repository.

Application Explorer, a design tool that works with the connector, is used to configure adapter connections, browse EIS objects, and configure services. Metadata created while you perform these operations are stored in the repository by the connector. The repository can be a file system or an Oracle database. It is deployed as a RAR file and has an associated deployment descriptor called ra.xml. You can create multiple connector factories by editing the Oracle WebLogic Server deployment descriptor ra.xml. See Chapter 3, "Oracle WebLogic Server Deployment and Integration" for more information.

Note: Do not use a file repository for BSE in production environments.
1.2 The Siebel Application Model

The Siebel Enterprise application defines a data abstraction layer that removes dependencies on the underlying database. It accomplishes this by using intermediate Business Components and Business Objects that represent database structures. A Business Component usually represents a table in a database. A Business Object is a group of related business components.

From a given business component, you can navigate the relationships defined for that component to another component. The path you use to traverse component relationships is called the navigation path. For example, if you want to obtain all addresses for a particular account, you can traverse the parent/child relationship between Account and Address to obtain those addresses. By using navigation paths, you can traverse nearly all of the business component relationships defined in the Siebel system.

In Siebel, Integration Objects are similar to Siebel Business Components but describe more complex hierarchal data relationships.

1.3 Integration with Siebel

You can use Oracle Application Adapter for Siebel to initiate a Siebel business process, such as add/update account, or you can use the adapter as part of an integration effort to connect Siebel and non-Siebel systems. Oracle Application Adapter for Siebel is bidirectional and can detect an event from Siebel by receiving a Siebel XML document emitted by Siebel.
When integrating with Siebel using Siebel XML documents, the adapter application developer must use existing Siebel Integration Objects or create new Siebel Integration Objects to use within a Siebel Workflow. The Workflow processes inbound or outbound Siebel XML and uses various transports such as MQSeries, File, and HTTP to exchange transactions with external systems. The Siebel Workflow is usually created by the Siebel administrator or developer using Siebel Workflow Administration screens.

When integrating with Siebel directly using the Java Data Bean or COM Data Interface, Oracle Application Adapter for Siebel does not require a Siebel Integration Object or Siebel Workflow. Instead, it executes Siebel Business Services and Siebel Business Components directly.

The following table lists Siebel objects and processes.

<table>
<thead>
<tr>
<th>Siebel Objects</th>
<th>API or Transport</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Services</td>
<td>Java Data Bean (Siebel 6.3-8.0)</td>
<td>Service</td>
</tr>
<tr>
<td></td>
<td>Com Data Interface (Siebel 6.01-6.2)</td>
<td>Service</td>
</tr>
<tr>
<td>Business Components</td>
<td>Java Data Bean (Siebel 6.3-8.0)</td>
<td>Service</td>
</tr>
<tr>
<td></td>
<td>Com Data Interface (Siebel 6.01-6.2)</td>
<td>Service</td>
</tr>
<tr>
<td>Integration Objects</td>
<td>File</td>
<td>Event, Service</td>
</tr>
<tr>
<td></td>
<td>HTTP</td>
<td>Event, Service</td>
</tr>
<tr>
<td></td>
<td>MQSeries</td>
<td>Event, Service</td>
</tr>
<tr>
<td></td>
<td>MQ Read</td>
<td>Service</td>
</tr>
</tbody>
</table>

### 1.3.1 Integrating with Siebel EAI Architecture

Siebel enables integration with other applications and systems using its Siebel EAI (Enterprise Application Integration) framework and its Business Integration Manager facility. Oracle Application Adapter for Siebel uses the Siebel EAI framework and leverages various integration access methods to provide the greatest amount of flexibility and functionality while working within the Siebel framework.

Oracle Application Adapter for Siebel supports the following integration access methods:

- Siebel Java Data Bean for services involving Siebel Business Components or Siebel Business Services.
- Siebel COM Data Interface for services involving Siebel Business Components or Siebel Business Services.
- Siebel XML for events and services involving Siebel Integration Objects.

### 1.4 Using Application Explorer with Oracle Application Adapter for Siebel

Application Explorer uses an explorer metaphor for browsing the Siebel system for Business Services, Business Objects, Business Components, and Integration Objects. The explorer enables you to create XML schemas and Web services for the associated object. External applications that access Siebel through Oracle Application Adapter for Siebel use either XML schemas or Web services to pass data between the external application and the adapter.
Application Explorer uses interfaces provided by Siebel and in-depth knowledge of the Siebel application systems to access and browse business object metadata. After an object is selected, Application Explorer can generate an XML schema or Web service to define the object for use with Oracle Application Adapter for Siebel.

Key features of Application Explorer include:

- The ability to connect to and explore a variety of application systems.
- Access to application system object metadata.
- A point-and-click process for generating XML schemas and Web services.

**See Also:**

- Oracle Application Server Adapter Concepts Guide
- Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server

### 1.5 BSE Versus Oracle Adapter J2CA Deployment

If you are using Oracle Application Adapter for Siebel with Business Process Execution Language (BPEL) Process Manager, please note that:

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

The following three factors explain the differences between deploying BSE and the Oracle 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 the Oracle WebLogic Server.
   - Provides better distribution of load.
   - Provides better isolation from any errors from third party libraries.
   - Provides better capability to isolate issues for debugging purposes.
   - Conforms more closely to the Service Oriented Architecture (SOA) model for building applications.

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

3. **Oracle 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 the Oracle Adapter J2CA, user name and password can be passed using the connection specification of the CCI.
This chapter describes how to configure Oracle Application Adapter for Siebel and
create schemas for Siebel Business Objects.

This chapter discusses the following topics:

- Starting Application Explorer
- Configuring Repository Settings
- Creating a Repository Configuration
- Establishing a Connection (Target) for Siebel
- Viewing Application System Objects
- Creating XML Schemas
- Siebel Prerequisites for Working With Integration Objects
- Creating Schemas for Siebel Integration Objects
- Creating Integration Object (IO) Nodes for Siebel
- Creating a Service Node for a Siebel Business Service
- Creating and Testing a Web Service (BSE Configurations Only)
- Generating WSDL (J2CA Configurations Only)
- Configuring an Event Adapter

**Encoding Support on UNIX Platforms**

**Important (All UNIX Platforms):** Before you attempt to connect to a Siebel target using a BSE or J2CA configuration in a UNIX environment, you must perform the additional steps described in "Adding Required Encoding Option (All UNIX Platforms)" on page 2-1. Failure to add the encoding option as described in this section results in an error and prevents you from connecting to the Siebel target. The error message may indicate that the encoding is not supported, for example:

Error: Problem activating adapter -- UTF-8 is not supported. Check logs for more information.

Error: Error getting target [Siebel] -- UTF-8 is not supported.

**Adding Required Encoding Option (All UNIX Platforms)**

Before attempting to connect to a Siebel target, add the following Java file encoding option to the startWebLogic.sh file:
JAVA_OPTIONS=""{SAVE_JAVA_OPTIONS} -Dfile.encoding=ISO8859_1"

The startWebLogic.sh file is located in the following directory:
C:\oracle\Middleware\user_projects\domains\base_domain\bin

**Siebel Connectivity Prerequisites**

You must perform the following steps to connect to your Siebel system (version 6.2 and lower) using COM connectivity for a J2CA configuration.

1. Install Siebel thick client on the same system where the adapters are installed.
2. Install the database client (Microsoft SQL Server or Oracle) on the same system.
3. The Siebel .DLL files (iwsiebel.local.dll and iwsiebel.core.dll) in the adapter lib folder must be added to the Application server path.
4. Edit the uagent.cfg file and change the data source parameter value from "local" to "server".
   
The uagent.cfg file can be found in the following Siebel thick client folder:
   c:\sea\client\bin

5. Edit the data source for SEA MSQl with appropriate parameters.
   
   You can edit a data source in Windows by accessing the Control Panel, Administrative Tools, and Data Sources (ODBC).

6. Use the following target type when creating the adapter target connection:
   
   Siebel 6.2 - (Local COM Access Implementation)

7. Provide the full path to the uagent.cfg file when creating an adapter target connection, for example:
   
   c:\sea\client\bin\uagent.cfg

### 2.1 Starting Application Explorer

To start Application Explorer:

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

![Application Explorer](image)

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

C:\oracle\Middleware\home_0309\Oracle_SOAI\soa\thirdparty\ApplicationAdapters\tools\iwa\bin\ae.bat

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

Application Explorer starts. You are ready to define new targets to your Siebel system.

2.2 Configuring Repository Settings

You need not configure BSE for a file-based repository because it is configured during the installation.

2.2.1 Configuring the Database Repository for BSE

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

---

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

---

To configure BSE:

1. Execute the iwse.ora SQL script on the machine where the database is installed.

   The iwse.ora SQL script is located in the following directory:
   
   C:\oracle\Middleware\Oracle_SOAl\soa\thirdparty\ApplicationAdapters\etc

   This script creates the required tables that are used to store the adapter configuration information in the database. These tables are used by Application Explorer and by adapters during design time and runtime. It is recommended that you use the same credentials to create the database repository and also in the web.xml file for database user credentials.

   C:\oracle\Middleware\Oracle_SOAl\soa\thirdparty\ApplicationAdapters\etc>sqlplus

   SQL*Plus: Release 10.1.0.2.0 - Production on Tue Dec 27 18:10:44 2005
   Copyright (c) 1982, 2004, Oracle. All rights reserved.

   Enter user-name: scott
   Enter password: scott1

   Connected to:
   Oracle Database 11g Enterprise Edition Release 11.1.1.2.0 - Production
   With the Partitioning, OLAP and Data Mining options

   SQL> @ iwse.ora

2. Open 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 a managed Oracle WebLogic server (for example, soa_server1).

   For example,

   http://localhost:8001/ibse
3. Log on when prompted.
   Enter the user ID and password, for example:
   - User name: weblogic
   - Password: weblogic
   The BSE configuration page is displayed.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System</strong></td>
<td></td>
</tr>
<tr>
<td>Language</td>
<td>English</td>
</tr>
<tr>
<td>Adapter Lib Directory</td>
<td>./ApplicationAdapters/lib</td>
</tr>
<tr>
<td>Encoding</td>
<td>UTF-8</td>
</tr>
<tr>
<td>Debug Level</td>
<td>DEBUG</td>
</tr>
<tr>
<td>Number of Async. Processors</td>
<td>0</td>
</tr>
<tr>
<td><strong>Repository</strong></td>
<td></td>
</tr>
<tr>
<td>Repository Type</td>
<td>File System</td>
</tr>
<tr>
<td>Repository Url</td>
<td>file://C:/oracle/Middleware/Oracle_SOAIsoa/thirdparty/ApplicationAdapters/lib</td>
</tr>
<tr>
<td>Repository Driver</td>
<td></td>
</tr>
<tr>
<td>Repository User</td>
<td></td>
</tr>
<tr>
<td>Repository Password</td>
<td></td>
</tr>
<tr>
<td>Repository Pooling</td>
<td></td>
</tr>
</tbody>
</table>

4. The `ojdbc14.jar` file must be copied to the following directory:
   C:/oracle/Middleware/Oracle_SOAIsoa/thirdparty/ApplicationAdapters/lib

5. Ensure that the Adapter Lib Directory parameter specifies the path to the lib directory, for example:
   C:/oracle/Middleware/Oracle_SOAIsoa/thirdparty/ApplicationAdapters/lib

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

### 2.2.1.1 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 page is displayed, as shown in the following image.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Property Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>System</strong></td>
<td></td>
</tr>
<tr>
<td>Language</td>
<td><em>English</em></td>
</tr>
<tr>
<td>Adapter Lib Directory</td>
<td><code>.\ApplicationAdapters\lib</code></td>
</tr>
<tr>
<td>Encoding</td>
<td><em>UTF-8</em></td>
</tr>
<tr>
<td>Debug Level</td>
<td><em>DEBUG</em></td>
</tr>
<tr>
<td>Number of Async. Processors</td>
<td><em>0</em></td>
</tr>
</tbody>
</table>

2. Configure the system settings.

   The following table lists the parameters with descriptions of the information to provide.
The following image shows all fields and check boxes for the Repository page.

<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>
<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 JDBC URL to use when opening a connection to the database.</td>
</tr>
<tr>
<td></td>
<td>For example, the following repository URL format is used when connecting to Oracle:</td>
</tr>
<tr>
<td></td>
<td>jdbc:oracle:thin:@hostname:port;SID</td>
</tr>
</tbody>
</table>

3. Configure the repository settings.

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

   See “Configuring a File System Repository” on page 2-7 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 JDBC URL to use when opening a connection to the database.</td>
</tr>
<tr>
<td></td>
<td>For example, the following repository URL format is used when connecting to Oracle:</td>
</tr>
<tr>
<td></td>
<td>jdbc:oracle:thin:@hostname:port;SID</td>
</tr>
</tbody>
</table>
Configuring Repository Settings

4. Click Save.

Configuring a File System Repository

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

The default location for the repository on Windows is:

C:\oracle\Middleware\user_projects\domains\base_domain\servers\soa_server1\stage\ibse\ibse.war

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.

2.2.2 Configuring the Database Repository for J2CA

This section describes how to configure the database repository for J2CA.

1. Execute the iwse.ora SQL script on the machine where the database is installed.

   The iwse.ora SQL script is located in the following directory:

   C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\etc

   This script creates the required tables that are used to store the adapter configuration information in the database. These tables are used by Application Explorer and by adapters during design time and runtime. It is recommended that you use the same credentials to create the database repository and also in the ra.xml file for database user credentials.

   C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\etc>sqlplus

   SQL*Plus: Release 10.1.0.2.0 - Production on Tue Dec 27 18:10:44 2005
   Copyright (c) 1982, 2004, Oracle. All rights reserved.

   Enter user-name: scott
   Enter password: scott1
Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.1.2.0 - Production
With the Partitioning, OLAP and Data Mining options

SQL> @ iwse.ora

2. Create the jcatransport.properties file and save it in the following directory:

C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\config\J2CA_SampleConfig

---

**Note:** The jcatransport.properties file is required for each J2CA configuration that is created using Application Explorer. The J2CA configuration folder, for example, J2CA_SampleConfig, is named according to the configuration name that is specified in Application Explorer.

3. Enter values for iwafjca.repo.url, iwafjca.repo.user and iwafjca.repo.password fields in the newly created jcatransport.properties file, as shown in the following example:

   iwafjca.repo.url=jdbc:oracle:thin:@90.0.0.51:1521:orcl
   iwafjca.repo.user=scott
   iwafjca.repo.password=scott1

   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>iwafjca.repo.url</td>
<td>Specify the JDBC 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:@host name:port;SID</td>
</tr>
<tr>
<td>iwafjca.repo.user</td>
<td>Specify a valid user ID to use when opening a connection to the database.</td>
</tr>
<tr>
<td>iwafjca.repo.password</td>
<td>Specify a valid password that is associated with the user ID.</td>
</tr>
</tbody>
</table>

4. The ojdbc14.jar file must be copied to the following directory:

   C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\lib

5. Navigate to the following directory:

   C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\iwafjca.rar\META-INF

6. Open the ra.xml file in a text editor.

7. Provide the JDBC connection information as a value for the IWAYRepo_URL property.

8. Provide a valid user name for the IWAYRepo_User property.
9. Provide a valid password for the IWAYRepo_Password property.

10. Save your changes to the ra.xml file.

### 2.2.2.1 Password Encryption

When creating J2CA configurations, you can also encrypt a password using Application Explorer and use this value in the jcatransport.properties and 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 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 only if you are using a database repository.
   
   The following is a sample of the jcatransport.properties file where the password is replaced:
   ```
   iwafjca.log.level=DEBUG
   iwafjca.repo.url=jdbc:oracle:thin:@172.30.166.100:1521:orcl
   iwafjca.repo.user=scott
   iwafjca.repo.password=ENCR (31893197318329732183129316432332123227)
   ```

6. In the ra.xml file, which is used during run time, replace the existing password with the encrypted value for the IWayRepoPassword element. This is applicable for file system and database repositories.

7. Restart the Oracle WebLogic Server.

### 2.3 Creating a Repository Configuration

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

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

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

#### 2.3.1 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 create a new configuration for BSE:
1. Start the Application Explorer.
2. Right-click Configurations and select New.
   The New Configuration dialog is displayed.
3. Enter a name for the new configuration (for example, SampleConfig) and click OK.
   The New Configuration dialog is displayed.
   ![New Configuration dialog]
4. From the Service Provider list, select iBSE.
5. In the iBSE URL field, accept the default URL or replace it with a different URL using the following format:
   
   \[http://host\ name:port/i\text{\lowercase{s}}e/IBSEServlet\]

   Where \text{\lowercase{s}}\text{\lowercase{l\lowercase{e}}} is the system where your Oracle WebLogic Server resides and \text{\lowercase{p}}\text{\lowercase{o}}\text{\lowercase{r}}\text{\lowercase{t}} is the HTTP port number on which the Oracle WebLogic Server is listening.
6. Click OK.
   A node representing the new configuration appears beneath the root Configurations node.

2.3.2 Creating a Configuration for J2CA
To create a configuration for Oracle 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. Start the Application Explorer.
2. Right-click Configurations and select New.
   ![Configuration node]
   The New Configuration dialog is displayed.
3. Enter a name for the new configuration (for example, SampleConfig) and click OK.
Creating a Repository Configuration

Configuring Oracle Application Server Adapter for Siebel

4. From the Service Provider list, select JCA.

5. Click OK.

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

```
Confi gurations
SampleConfig
```

The Oracle Adapter J2CA configuration file is stored in:

```
C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\config\configuration_name
```

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

**HTTP Repository Connection**

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

1. Start the Application Explorer.

2. Right-click the Configurations node in the left pane and select New.

   The New Configuration dialog opens.

3. Type a name for the configuration and click OK.

4. Select JCA from the Service Provider list box and enter an HTTP target value in the Home field.

   Use the following format for the HTTP target value:

   `http://host name:port/iwafjca/JCAServlet`

   For example:

   `http://iwserv14:8001/iwafjca/JCAServlet`

5. Click OK.

   The new HTTP repository connection is added to the Configurations node.

   Once you connect to the remote server, you can create new Adapter targets, generate WSDL documents, and store them in the remote server.

   **Note:** When you configure an Adapter target with the J2CA HTTP repository, you are not required to restart the Oracle WebLogic Server for run time purposes.
### 2.3.3 Connecting to a BSE or J2CA Configuration

To connect to a new configuration:

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

2. Select Connect.

Nodes appear for Adapters, Events, and Business Services (also known as Web services). The Business Services node is only available for BSE configurations. If you are connected to a J2CA configuration, the Business Services node is not shown. The following is an example of a BSE configuration named SampleConfig:

- Use the Adapters folder to create inbound interaction with Siebel. For example, you use the Siebel node in the Adapters folder to configure a service that updates Siebel.
- Use the Events folder (available for J2CA configurations only) to configure listeners that listen for events in Siebel.
- 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 Siebel.

### 2.4 Establishing a Connection (Target) for Siebel

To browse the Siebel Business Services, Business Components, and Integration Objects, you must define a target to Siebel. After you define the target, the parameters are automatically saved. However, you must provide the password to Siebel every time you connect to the target.

**Important (All UNIX Platforms):** Before you attempt to connect to a Siebel target using a BSE or J2CA configuration in a UNIX environment, you must perform the additional steps described in "Adding Required Encoding Option (All UNIX Platforms)" on page 2-12. Failure to add the encoding option as described in this section results in an error and you are not able to connect to the Siebel target. The error message may indicate that the encoding is not supported, for example:

```
Error: Problem activating adapter -- UTF-8 is not supported. Check logs for more information.
```

```
Error: Error getting target [Siebel] -- UTF-8 is not supported.
```

**Adding Required Encoding Option (All UNIX Platforms)**

Before attempting to connect to a Siebel target, perform the following steps:

1. Add the following Java file encoding option to the `startWebLogic.sh` file:

   ```
   JAVA_OPTIONS="($SAVE_JAVA_OPTIONS) -Dfile.encoding=ISO8859_1"
   ```

   The `startWebLogic.sh` file is located in the following directory:
2. Log in to the Oracle WebLogic Server console, click Administration, then Server Properties, and under Command Line Options, edit the Java Options field to include the following:
   -Dfile.encoding="ISO8859_1"

2.4.1 Defining a Target to Siebel

The connection parameters required for defining a Siebel target can be obtained from the eapps.cfg file, which is located in the following directory:

drive:\SiebelRoot\SWEApp\BIN

Where Siebelroot is the Siebel installation directory.

When you are working with a J2CA configuration, creating, updating, and deleting a target requires you to restart the application server. The application server must also be restarted after you create a target, connect to a target, and generate a WSDL for a Siebel object. In addition, make sure to close Application Explorer before you restart the application server.

To define a target to Siebel:

1. In the left pane, expand the Adapters node.

2. Right-click the Siebel node and select Add Target.

   The Add Target dialog is displayed. Provide the following information:
   
   a. In the Name field, enter a name for the new target.
   
   b. In the Description field, enter a description (optional).
   
   c. From the Target Type list, select Java Bean Data Connection (default).

3. Click OK.

   The Java Data Bean Connection dialog is displayed.
Enter the system information as specified in the following steps:

a. In the Gateway Server field, enter the name of the server. To specify a Gateway Server that uses a port other than the default (usually, 2320), add a colon and the port number, for example, `gateway name:port number`.

b. In the Enterprise Name field, enter the appropriate name.

c. In the Siebel Server field, enter the name of your Siebel server. Do not supply a value in this field when connecting to a Siebel 7.7, 7.8, or 8 system.

d. In the User field, enter the user name.

e. In the Password field, enter the password associated with the user name.

f. From the Siebel Version list, select **Siebel 7.7 and above** (default) or **Siebel 7.5 and below**.

g. Click the **Advanced** tab and verify the following:

   Language

   Object Manager
For Siebel 7.0.3, the default Object Manager is EAIObjMgr. For Siebel 7.7, the default is EAIObjMgr_enu. Siebel 7.7 requires that you add a language extension (for example, _enu) to the end of the Object Manager name. Check with your Siebel Administrator for the specific names that apply to your system.

If no repository is specified, a full list of objects from all available repositories is returned. If a specified repository is not found, an empty list of objects is returned.

The configuration parameters supplied are used by Siebel client applications to connect to the Siebel system. For more information about these parameters, see your Siebel documentation or ask your Siebel system administrator.

**Repository Name**

If no repository is specified, a full list of objects from all available repositories is returned. If a specified repository is not found, an empty list of objects is returned.

The configuration parameters supplied are those used by Siebel client applications to connect to the Siebel system. For more information about these parameters, see your Siebel documentation or ask your Siebel system administrator.

**Encryption**

A new parameter named Encryption is now introduced to the Advanced tab when using the Siebel adapter to create a target during design time. This parameter has two values, None and RSA. The default value is None, where no encryption is performed. By choosing RSA, an RSA-encrypted connection to the object manager specified is established.

To use RSA encryption, the Object Manager must be specified as SCCObjMgr_enu.
Establishing a Connection (Target) for Siebel

2.4.2 Connecting to a Defined Target

To connect to a defined target:
1. Expand the Siebel node and click the target name to which you want to connect.

2. In the right pane, enter the password for that target.

3. In the left pane, right-click the target name and select Connect.

The target icon changes, indicating that you are connected to the Siebel system.

You can now browse the available Business Objects, Business Services, and Integration Objects in the Siebel system.

2.4.3 Disconnecting From Siebel

Although you can maintain multiple open connections to different application systems, it is good practice to close connections when not in use.

To disconnect from Siebel:
1. In the left pane, select the target to which you are connected.

2. Right-click the target and select Disconnect.

Disconnecting from the application system drops the target, but the node remains. The SiebelConnection node in the left pane changes to reflect that the target is disconnected.

2.4.4 Editing a Target

To edit a target:

---

**Note:** These parameters are typically found in Siebel configuration files stored under the Siebel server root/bin/<language> directory, where language is the Siebel code for the language you installed (enu for U.S English). For example, for Siebel versions 7 and higher on a Windows platform, for the Siebel Call Center module, these values can be found in the uagent.cfg file. Consult your Siebel administrator and your Siebel bookshelf documentation for more information.

4. Click OK.

In the left pane, the target you create appears under the Siebel node.
1. In the left pane, ensure the target you want to edit is disconnected.
2. Right-click the disconnected target and select **Edit**.

   ![Screenshot showing the Edit pane on the right]

   The Edit pane is displayed on the right.
3. Modify the target information.
4. Click **OK**.

**2.4.5 Deleting a Target to Siebel**

You can delete a target, rather than just disconnecting and closing it. When you delete the target, the node disappears from the list of Siebel targets in the left pane of Application Explorer.

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

To delete a target:
1. In the left pane, select the target.
2. Right-click the target and select **Delete**.
   
   A confirmation message is displayed.
3. Click **OK** to delete the target you selected.
   
   The Siebel connection node disappears from the left pane.

**2.5 Viewing Application System Objects**

Application Explorer gives you the flexibility to view all Siebel application system objects. One benefit of this flexibility is that you can gain an understanding of the Siebel data structure. You can review parameters, data types, and other attributes of the Siebel data in the right pane.

**2.5.1 Viewing Metadata**

To view metadata:

1. If you have not started Application Explorer, start Application Explorer and connect to your Siebel system.
2. In the left pane, expand the Business Object or Business Service containing the component for which you want to generate schema.
3. Expand the Business Object or Business Service node.
4. Expand the Business Component or the Business Service node to view the objects under it.
   - For a Business Component, select the node in which you are interested, for example, Account.
For a Siebel Business Service, select the object in which you are interested, for example, addAccount.

5. In the right pane, click the ellipsis (...) in the Table row of the properties table. The metadata table appears in the right pane.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Required</th>
<th>MultiValued</th>
<th>ReadOnly</th>
<th>Active</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account Co...</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Account Con...</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Account Mar...</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Account Org...</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACCOUNT Pro...</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Account Role</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Account Stat...</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Account Trend...</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address Act...</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address Id...</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Address Inte...</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agreement E...</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agreement N...</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agreement S...</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agreement S...</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Algorithm Type</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alias</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Annual Reve...</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assignment ...</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assignment ...</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assignment ...</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assignment ...</td>
<td>boolean</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assignment ...</td>
<td>string</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.6 Creating XML Schemas

You can create service schemas for Business Services and Business Components using Application Explorer.

The following topic describes how to create schemas for the adapter when you deploy Oracle Application Adapter for Siebel for use either in a J2CA environment or a Web services environment. See "Creating and Testing a Web Service (BSE Configurations Only)" on page 2-29 if you plan to deploy Oracle Application Adapter for Siebel in a Web services environment.

Siebel Schema Considerations

When inserting a record into Siebel, the data can be specified by the user or configured in Siebel to have default values or other system generated values. For example the Account Business Component, Currency Code, by default, has ‘USD’ and the system
fields such as ROW_ID generated by the Siebel system when the record is inserted. The Siebel API does not provide this distinction. Therefore, the Oracle Application Adapter for Siebel can not anticipate what the required fields the user should enter are and what are the required fields that can be filled by Siebel. As a result, the adapter schemas have been modified to have all elements as optional by setting minoccurs=0 for the elements.

Hence, all users must determine which fields are mandatory through Siebel Tools and create a payload (request XML document) for Siebel services (outbound).

Creating an XML Schema for a Siebel Business Object or Business Service

You create schemas for Siebel Business Service methods (for example, the Add method) and Business Components using Application Explorer. After you create a schema, you can use it to generate service request and response schemas for the Business Service or Business Component.

Siebel Business Objects contain one or more Siebel Business Components. You can view Business Components by clicking the associated Business Object.

The following image shows the Account Business Object expanded to display all Business Components.

Creating an XML Schema for a Siebel Business Component or Business Service

To generate service request and response schemas for a Business Component or Business Service:

1. If you have not started the Application, start Application Explorer and connect to your Siebel system.
2. In the left pane, expand the Business Object or the Business Service node.
3. Expand the Business Component or Business Service to view the objects under it.
For a Business Component, expand the Business Object node, then expand the Business Component you want, then expand the node you want, and select the method for which you want to create a schema.

For a Siebel Business Service, expand the Business Service node containing the object for which you want to create schema.

4. Right-click the node and select **Generate Schema**.

Application Explorer accesses the Siebel repository and builds schemas. Schema tabs similar to the following appear in the right pane.

5. To view a schema, click the ellipsis tab corresponding to the schema you want to view.

The schema appears on the right.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:z=""
  <xs:element name="Siebel">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="insert" type="z:record"/>
      </xs:sequence>
      <xs:attribute name="locator" type="xs:string" use="optional"/>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="record">
    <xs:sequence>
      <xs:element name="Account_spCompetition" type="xs:string"/>
      <xs:element name="Account_spCondition" type="xs:string"/>
      <xs:element name="Account_spMarkets" type="xs:string"/>
      <xs:element name="Account_spOrganization_spIntegrative" type="xs:string"/>
      <xs:element name="Account_spProducts" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

**Searching for a Specific Siebel Object**

You can use the search function in Application Explorer to locate a Siebel object or node quickly.

1. If you have not started the explorer, start Application Explorer and connect to your Siebel system through a target.

2. Expand the target and select Business Object, Business Service, or Integration Object.

3. In the right pane, move the cursor over Operations and select Search.
4. Enter the name of the node or object on which you want to search in the text entry box, for example, Account.

5. Click OK.

   A list containing the Siebel items that match your search appears.

6. Select the item in which you are interested.

   Application Explorer locates the item in which you are interested.

**Returning Fields in a Specified Order**

When you create a request document from an XML schema to query the Siebel system, you can limit the expected response to specific fields that are specified in the query. The response contains the fields in the order in which they were specified. If you do not specify a set of fields, the response document contains the entire set.

For example, the following query returns all fields:

```xml
<m:Siebel location="S/BO/Account/Account/queryWithView" view="AllView">
    <m:select>
        <m:Name>Yelena*</m:Name>
    </m:select>
</m:Siebel>
```

The following query returns a response that only contains the fields Name, Location and Account Status fields:

```xml
<m:Siebel location="S/BO/Account/Account/queryWithView" view="AllView">
    <m:select>
        <m:Name>Yelena*</m:Name>
    </m:select>
    <m:field>Name</m:field>
    <m:field>Location</m:field>
    <m:field>Account Status</m:field>
</m:Siebel>
```

**Using QueryWithView**

For Business Components, the iWay Application Adapter for Siebel enables Insert, Update, Delete, and Query. It also enables a method called QueryWithView. The View modes are a visibility feature provided by Siebel.

By using QueryWithView, you can specify a Siebel View mode as a parameter. The API parameters allow different presentations of data depending on the Siebel environment that you configured.

You can use Query except when you want to enable a user to retrieve records based on different view modes. In this case, use QueryWithView. For more information on QueryWithView mode or Siebel "Visibility" concepts, see your Siebel Administrator.

The following levels are available:

- Sales Rep View
- Manager View
- Personal View
- All View
- Organization View
- Group View
- Catalog View
2.7 Siebel Prerequisites for Working With Integration Objects

To create XML schemas for Siebel Integration Objects, you may have to generate XDR schemas first, using the Siebel Tools Schema Wizard.

The XDR schema is used as input to Application Explorer when generating schemas for integration objects. After you generate the XDR schema, Application Explorer uses the XDR file to generate the XML schema.

Please note:
- For **Siebel 7.5 and later**: Generate XSD schemas directly from Siebel tools. These XSD schemas are used to create Web services directly using Application Explorer. After you generate an XSD schema through Siebel tools, use it to create an IO node and Web service.
- For **Siebel 7.0**: You cannot generate XSD schemas directly from Siebel tools; only XDR schemas can be created. Therefore, to create a Web service, Application Explorer must first generate an XSD schema from the XDR schema.
- For releases **before Siebel 6.3**: The Siebel Tools Schema Wizard creates only DTD schemas. You must transform these schemas manually, or by using other tools, into XDR files before Application Explorer can use them as input to create XML schemas. In addition, you must include the SiebelMessage tag reference in your XDR file.

Oracle Application Adapter for Siebel supports access to Siebel Integration Objects by using Siebel XML to handle events. Using Siebel Integration Objects through supported transports requires Siebel workflows.

2.8 Creating Schemas for Siebel Integration Objects

This section describes how to create schemas for Siebel Integration Objects.

**Creating a Siebel XDR or XSD Schema for a Siebel Integration Object**

To generate a Siebel XDR or XSD schema:

1. Log on to Siebel Tools.
Perform the following steps:

a. Enter your user ID and password.

b. Select a database from the list.

2. Click OK.

The Siebel Tools window is displayed. Integration Objects appear in the right pane.

3. To create a schema, select an Integration Object, for example, Sample Account.
4. Click **Generate Schema**.

The Generate XML Schema wizard is displayed.

![Generate XML Schema](image)

Perform the following steps:

a. From the Select a Business Service list, select **EAI XML XDR Generator** for XDR schemas or **EAI XML XSD Generator** for XSD schemas (for Siebel 7.5 and later).

b. From the Select an envelope type list, select **Siebel Message envelope**.

c. In the Choose the file name field, specify a file name for the XDR schema and a directory where it can be accessed by Application Explorer.

**Note:** The XDR or XSD schema file must be saved to a directory on the same computer as Application Explorer.

5. Click **Finish**.

6. Create a workflow to accept incoming XML documents through HTTP and to insert/update Siebel data by using the EAI XML Converter and EAI Siebel Adapter Business Services.

   For more information, see Appendix A, "Using Siebel Workflows".

7. Edit the `eai.cfg` file, which is located in the following directory:

   `<siebel_server>/bin/enu`

8. Add the following line to the [HTTP Services] section:

   ```
   [HTTP Services]
   wt = iWayWorkflow
   ```

9. Confirm that the following line is set in the [EAI_ENU] section of the `Eapps.cfg` file:

   ```
   [EAI_ENU]
   EnableExtServiceOnly = True
   ```

   The `Eapps.cfg` file is located in the following directory:

   `<siebel_server>/bin`
10. Create a named subsystem using Siebel Server Manager by running the following command, where EAITEST is the name of the workflow that was created in step 6:

```
create named subsystem iWayWorkflow for subsystem EAITransportDataHandlingSubsys with DispatchWorkflowProcess="EAITEST"
```

Now you can use Application Explorer to create Integration Object (IO) nodes for Siebel.

### 2.9 Creating Integration Object (IO) Nodes for Siebel

To create an Integration Object node for Siebel, perform the following steps:

1. In Application Explorer, connect to a defined target. See "Connecting to a Defined Target" on page 2-16 for information on how to connect to a target. The X over the icon disappears, indicating that the target is connected.

2. Expand the Integration Object node and select Sample Account.

3. Right-click the Sample Account node and select Add IO Node. The Add IO Node dialog is displayed.

![Add IO Node Dialog](image)

Please note:

- **For Siebel 7.5 or later**: Generate XSD schemas directly from Siebel tools. You use the XSD schemas when you create Web services in Application Explorer. After you generate an XSD schema through Siebel tools, use it to create an IO node and a Web service.

- **For Siebel 7.0**: You cannot generate XSD schemas directly from Siebel tools; only XDR schemas can be created. Before you create a Web service, you must first generate an XSD schema from the XDR schema using Application Explorer.

```
Note: This is the schema file that you generated in Creating Schemas for Siebel Integration Objects on page 2-22.
```

4. Enter a node name, for example SampleAccount in the **Node name** field and a path to the Sample Account XDR or XSD file in the **Schema location** field.
5. If the XSD schema has already been generated, select XSD Schema. If you are using Siebel-generated XDR schemas, do not select the XSD schema option.

6. Select a protocol from the Protocol list.

7. Click Continue.

   The Add IO Node dialog is displayed.

8. Perform the following steps:
   a. In the SWE URL field, type the Base SWE URL. For example:
      
      http://web_server/eai/start.swe
      
      Where web_server is the name of the Web server that is hosting Siebel SWE.
   b. In the SWE External Source field, type the section within the eai.cfg file to execute, which is the [HTTP Services] section.
      
      For more information, see step 8 in Creating Schemas for Siebel Integration Objects on page 2-22.
   c. In the SWE External Command field, type the following command exactly as shown:
      
      Execute
   d. In the User Name and Password fields, type a valid user name and password used to connect to the Siebel SWE.
      
      The user name and password must have privileges to execute the given workflow.

9. Click Finish.

   The new IO node is listed under the Integration Object's Sample Account node.

You can now create an XML schema.
Creating an XML Schema for a Siebel Integration Object

After you create an Integration Object node for Siebel, you can create an XML schema using Application Explorer.

To create an XML schema:

1. In Application Explorer, expand the **Integration Objects** node to browse the Integration Objects in the Siebel system.

2. Scroll down and select an integration object, for example, SampleAccount.
   An XML event schema for the integration object is automatically created.

3. Click the Event Schema tab in the right pane.
   The XML event schema appears in the right pane, as shown in the following image:

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Generated by the iBSE 2004-04-09T18:44:19Z -->
<xsd:schema
 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 elementFormDefault="qualified">
  <xsd:annotation>
    <xsd:documentation>SiebelMessage</xsd:documentation>
  </xsd:annotation>
</xsd:schema>
```

   ```xml
   <xsd:element name="SiebelMessage">
     <xsd:complexType>
       <xsd:choice maxOccurs="unbounded" minOccurs="0">
         <xsd:element ref="ListOfSampleAccount"
           maxOccurs="1" minOccurs="0" />
       </xsd:choice>
       <xsd:attribute name="MessageId" />
       <xsd:attribute name="MessageType" use="required" fixed="Integration Object" />
       <xsd:attribute name="IntObjectName" />
   </xsd:complexType>
```

```xml
</xsd:element>
```

---

**Creating Integration Object (IO) Nodes for Siebel**

Configuring Oracle Application Server Adapter for Siebel  2-27
2.10 Creating a Service Node for a Siebel Business Service

OracleAS Adapter for Siebel enables the addition of a service node for a Business Service that includes methods containing method arguments having hierarchy data types.

**Important limitations:**

- The adapter supports only Integration Object hierarchy data types.
- Adding a Service node requires that you have previously generated an XSD schema for the Integration Object. For more information on generating XSD schemas for Siebel Integration Objects, see "Creating Schemas for Siebel Integration Objects" on page 2-22.
- Only one of the method arguments for the Business Service method for which you want to add a service node can be a hierarchical data type.
- The method argument `XMLCharEncoding` is not supported. Leave this element blank in the XML payload. If you enter a valid `XMLCharEncoding` value such as UTF-8 or UTF-16, the following error is received:

  Invocation of Service failed.

To create the service:

1. Select the Business Service node in which you are interested.
2. Right-click the Business Service method argument for which you want to create a service and select **Add Service Node**.

   The Add Service Node dialog is displayed.

3. Perform the following steps:
   
   a. Provide a service node name.
   b. Enter a description (optional).
   c. Provide the full path (including the file name) to the XSD schema file.
   d. Specify the root element for the XSD schema file. For many XSD schemas for Integration Objects, the root element is `SiebelMessage`.
   e. Specify whether the XSD schema is for an Integration Object.

      **Important:** You must verify that this check box is selected.

4. Click OK.
The Service node is listed under the Business Service object.

You can right-click this node to create a Web service. The request and response schemas are displayed in the right pane.

The following procedure describes how to create a Web service for a Business Object.

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

You can generate a business service (also known as a Web service) for Siebel objects you want to use with your adapter after you have properly configured the servlet BSE.

Note: In a J2EE Connector Architecture (J2CA) implementation of adapters, Web services are not available. When the adapters are deployed to use Oracle Adapter J2CA, the Common Client Interface provides integration services using the adapters.

This section contains the following topics:
- "Creating a Web Service" on page 2-29
- "Testing a Web Service" on page 2-30

2.11.1 Creating a Web Service

To generate a Web service for a Siebel Business Object:

1. Connect to your Siebel system.
2. Expand a Business Object node.
3. Expand the Business Component for which you want to create a Web service.
4. Expand the object and select a method for creating the Web service, for example, QueryWithView under Account.
5. Right-click the node from which you want to create a business service and select Create Business Service.

The Create Web Service dialog is displayed.

You can add the business object 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. Specify a service name if you are creating a new service. This name identifies the Web service in the list of services under the Business Services node.

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

d. Select one of the available licenses.

6. Click Next.
   The License and Method dialog is displayed. Perform the following steps:
   a. In the License 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 Description field, enter a brief description of the method.

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

2.11.2 Testing a Web Service

After you create a Web service for the Siebel Business Object, test it to ensure it functions properly. Application Explorer includes a test tool for testing a Web service.

Testing a Web Service for a Business Object
1. In the left pane of Application Explorer, expand the Business Services node.

2. Expand the Services node.

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

4. Expand the Methods node under the service and select the method you want to test.
   The test option appears in the right pane.
   If you are testing a Web service that requires XML input, an input field appears.

5. Click Invoke.
   Application Explorer displays the results in the results pane.
Testing a Web Service for a Business Service

After you create a Web service for the Siebel Business Service, test it to ensure it functions properly. Application Explorer includes a test tool for testing a Web service.

1. If it is not expanded, expand the Business Services node.
2. Expand the Services node.
3. Select the name of the business service you want to test.
4. Expand the Methods node and select the name of the method you want to test. The test option appears in the right pane.
   - If you are testing a Web service that requires XML input, an input field appears.
5. Provide the appropriate input.
6. Click Invoke.
   - Application Explorer displays the results in the results pane.

Identity Propagation

If you test or execute a Web service using a third party XML editor, for example XMLSpy, the Username and Password values that you specify in the SOAP header must be valid and are used to connect to Siebel. The user name and password values that you provided for Siebel 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">
```

```xml
Testing a Web Service for a Business Service

After you create a Web service for the Siebel Business Service, test it to ensure it functions properly. Application Explorer includes a test tool for testing a Web service.

1. If it is not expanded, expand the Business Services node.
2. Expand the Services node.
3. Select the name of the business service you want to test.
4. Expand the Methods node and select the name of the method you want to test. The test option appears in the right pane.
   - If you are testing a Web service that requires XML input, an input field appears.
5. Provide the appropriate input.
6. Click Invoke.
   - Application Explorer displays the results in the results pane.

Identity Propagation

If you test or execute a Web service using a third party XML editor, for example XMLSpy, the Username and Password values that you specify in the SOAP header must be valid and are used to connect to Siebel. The user name and password values that you provided for Siebel 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">
```
2.12 Generating WSDL (J2CA Configurations Only)

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

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

To generate a WSDL file for request-response service:

1. Under your connected Siebel target, expand Business Object, Account, Account. Navigate to an object and right-click the object.

   The following menu is displayed.

   ![Image](Export Schema(s)

   Create Outbound JCA Service(Request/Response)

   Apply Filter

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

   The Export WSDL dialog is displayed.

   ![Image](Export WSDL

   Name: [wsdl|\]\|wls]\|update_invoke.wsdl]

   OK Cancel

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.

4. 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.

### 2.13 Configuring an Event Adapter

Events are generated by a specific business condition being satisfied or triggered in the Siebel 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 configure an event, you must create a channel.

---

**Note:** If you are using a J2CA configuration, you must create a new channel for every different event object and select this channel when you generate WSDL. Creating a channel is required for J2CA configurations only. For example, if you are working with the Account and Contact Siebel objects, then two separate channels are required for this purpose.

---

A 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 the adapter. See "Creating and Modifying a Channel" on page 2-33 for more information.

Please note that adding IO node functionality is not applicable in event configurations.

### 2.13.1 Creating and Modifying 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 WebLogic Server to recognize the change and update the repository for run time purposes. After successfully creating the channel and inbound WSDL file, close Application Explorer before you restart the application server.

---

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

---

Three channel types are available:

- HTTP
- MQ Series
- File
2.13.1.1 Creating an HTTP Channel

To create an HTTP channel:

1. Click the **Events** node.
   The Events window is displayed. The adapters that appear in the left pane support events.

2. In the left pane, expand the **Siebel** node.
   The ports and channels nodes appear.

3. Right-click **channels** and select **Add channel**.
   The Add Channel dialog is displayed.

   ![Add Channel Dialog](image)

   Perform the following steps:
   a. Enter a name for the channel, for example, NewChannel.
   b. Enter a brief description.
   c. From the Protocol list, select **HTTP Listener**.

4. Click **Next**.
   The Basic dialog is displayed.

---

**Note:** Channels can be configured only on the system where the Oracle Application Adapter for Siebel is installed.
5. Enter the system information as specified in the following table:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Listener port</td>
<td>Port on which to listen for Siebel event data.</td>
</tr>
<tr>
<td>Https</td>
<td>For a secure HTTP connection, select the <strong>Https</strong> check box.</td>
</tr>
<tr>
<td></td>
<td>This option is currently not supported.</td>
</tr>
<tr>
<td>Synchronization</td>
<td>Select <strong>REQUEST_RESPONSE</strong> from the Synchronization Type list, which is</td>
</tr>
<tr>
<td>Type</td>
<td>the recommended option.</td>
</tr>
<tr>
<td>Encoding Type</td>
<td>Choose an encoding type to be used from the Encoding Type list. By default,</td>
</tr>
<tr>
<td></td>
<td>ASCII is selected.</td>
</tr>
</tbody>
</table>

6. Click the **PreParser** tab.

7. Specify the location of the schema file that was generated for the Integration Object node using the **Export Schema(s)** option in Application Explorer.

**Note:** During run time, the Oracle Application Adapter for Siebel adds the namespace to the Siebel published document using the schema that is specified in the PreParser tab. If the Schema location field in the PreParser tab is left blank, BPEL and Mediator processes do not work properly as the Siebel published documents do not contain any namespaces.
8. Click OK.

A summary is displayed, which provides the channel description, channel status, and available ports. All the information is associated with the channel you created. The channel also appears under the channels node in the left pane.

An X over the icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.

9. Right-click the channel and select Start.

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

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

2.13.1.2 Creating an MQ Series Channel

To create an MQ Series channel:

1. Click the Events node.

The Events window is displayed. The adapters that appear in the left pane support events.

2. In the left pane, expand the Siebel node.

The ports and channels nodes appear.

3. Right-click the channels node and select Add channel.

The Add a new channel pane is displayed. Perform the following steps:

   a. Enter a name for the channel, for example, NewChannel.
   b. Enter a brief description.
   c. From the Protocol list, select MQ Series Listener.

4. Click Next.

The MQ Listener dialog is displayed.
5. Enter the system information as specified in the following steps:
   a. In the Request tab, enter values for the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queue manager name</td>
<td>The host on which the MQ Server is located (MQ Client only).</td>
</tr>
<tr>
<td>MQ server host for MQClient operation</td>
<td>Port on which the host database is listening.</td>
</tr>
<tr>
<td>MQ server port for MQClient operation</td>
<td>The number to connect to an MQ Server queue manager (MQ client only).</td>
</tr>
<tr>
<td>MQ server channel for MQClient operation</td>
<td>The case-sensitive name of the channel that connects with the remote MQ Server queue manager (MQ client only). The default channel name for MQSeries is SYSTEM.DEF.SVRCONN.</td>
</tr>
<tr>
<td>Document type XML</td>
<td>Leave the default selection.</td>
</tr>
<tr>
<td>Request queue name</td>
<td>Queue where the message is routed and where request documents are received.</td>
</tr>
</tbody>
</table>
b. In the Response tab, enter values for the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synchronization</td>
<td>Select REQUEST_RESPONSE from the Synchronization Type list, which is the</td>
</tr>
<tr>
<td>Type</td>
<td>recommended option.</td>
</tr>
</tbody>
</table>

c. In the Advanced tab, enter values for the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message wait interval (msec)</td>
<td>The interval (in milliseconds) when to check for new input. The default is</td>
</tr>
<tr>
<td></td>
<td>3 seconds. Optional.</td>
</tr>
<tr>
<td>Mode of operation</td>
<td>Choose Sequential or Threaded.</td>
</tr>
<tr>
<td></td>
<td>■ Sequential indicates single processing of requests.</td>
</tr>
<tr>
<td></td>
<td>■ Threaded indicates processing of multiple requests simultaneously.</td>
</tr>
<tr>
<td>Thread limit</td>
<td>If you selected threaded processing, indicate the maximum number of</td>
</tr>
<tr>
<td></td>
<td>requests that can be processed simultaneously.</td>
</tr>
</tbody>
</table>

6. Click OK.

A summary is displayed, which provides the channel description, channel status, and available ports. All the information is associated with the channel you created. The channel also appears under the channels node in the left pane.

An X over the icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.

7. Right-click the channel and select Start.

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

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

2.13.1.3 Creating a File Channel

To create a File channel:

1. Click the Events node.

   The Events window is displayed. The adapters that appear in the left pane support events.

2. In the left pane, expand the Siebel node.

   The ports and channels nodes appear.

3. Right-click the channels node and select Add Channel.

   The Add Channel dialog is displayed. Perform the following steps:

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

b. Enter a brief description.

c. From the Protocol list, select File Listener.

4. Click Next.

   The File Listener dialog is displayed.
5. Enter the system information as specified in the following steps:
   
a. In the Request tab, enter values for the following parameters:

   **Parameter** | **Description**
   --- | ---
   Polling Location | The target file system location for the Siebel XML file.
   File Mask | The file name to be used for the output file generated by this operation.

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

   **Parameter** | **Definition**
   --- | ---
   Synchronization Type | Select REQUEST_RESPONSE from the Synchronization Type list, which is the recommended option.
   Response/Ack Directory | Directory where responses or acknowledgments are sent.

   c. In the Advanced tab, enter values for the following parameters:

   **Parameter** | **Definition**
   --- | ---
   Error Directory | Directory to which documents with errors are written.
   Poll interval (msec) | The interval (in milliseconds) when to check for new input. The default is 3 seconds. Optional.
   Processing Mode | Choose Sequential or Threaded.
   ■ Sequential indicates single processing of requests.
   ■ Threaded indicates processing of multiple requests simultaneously.
   Thread limit | If you selected threaded processing, indicate the maximum number of requests that can be processed simultaneously.

6. Click OK.

   A summary is displayed, which provides the channel description and channel status. All the information is associated with the channel you created. The channel also appears under the channels node in the left pane.

   An X over the icon indicates that the channel is currently disconnected. You must start the channel to activate your event configuration.
7. Right-click the channel and select **Start**.
   The channel you created becomes active.
   The X over the icon in the left pane disappears.

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

### 2.13.1.4 Editing a Channel
To edit a channel:

1. In the left pane, select the channel you want to edit.
2. Right-click the channel and select **Edit**.
   The Edit channels dialog is displayed.
3. Make the required changes to the channel configuration and click **OK**.

### 2.13.1.5 Deleting a Channel
To delete a channel:

1. In the left pane, select the channel you want to delete.
2. Right-click the channel and select **Delete**.
   The channel disappears from the list in the left pane.
This chapter describes Oracle WebLogic Server (OracleWLS) deployment and integration with Oracle Application Adapter for Siebel.

This chapter discusses the following topics:

- Adapter Integration with Oracle WebLogic Server
- Deployment of Adapter
- Updating Adapter Configuration

See Also:
- Oracle Application Server Adapter Concepts Guide

3.1 Adapter Integration with Oracle WebLogic Server

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

See Also:
- Oracle Application Server Adapter Concepts Guide

3.2 Deployment of Adapter

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

During the J2CA deployment of OracleAS Adapter for Siebel, OracleWLS generates a deployment descriptor called `ra.xml`, located in:

```
C:\oracle\Middleware\Oracle_SOA\soa\thirdparty\ApplicationAdapters\iwafjca.rar\META-INF
```

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

---

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

---

**Creating a Managed Connector Factory Object**

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

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE connector PUBLIC "-//Sun Microsystems, Inc./DTD Connector 1.0//EN"
```
"http://java.sun.com/dtd/connector_1_0.dtd">
<connector>
  <display-name>IWAFJCA10</display-name>
  <vendor-name>IWAY Software</vendor-name>
  <spec-version>1.0</spec-version>
  <eis-type>IWAF</eis-type>
  <version>1.0</version>
  <license>
    <license-required>false</license-required>
  </license>
</connectoradapter>

<managedconnectionfactory-class>com.ibi.afjca.spi.IWAFManagedConnectionFactory</managedconnectionfactory-class>

<connectionfactory-interface>javax.resource.cci.ConnectionFactory</connectionfactory-interface>

<connectionfactory-impl-class>com.ibi.afjca.cci.IWAFConnectionFactory</connectionfactory-impl-class>

<connection-interface>javax.resource.cci.Connection</connection-interface>

<connection-impl-class>com.ibi.afjca.cci.IWAFConnection</connection-impl-class>

<transaction-support>NoTransaction</transaction-support>

<config-property>
  <config-property-name>AdapterName</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>
</config-property>

<config-property>
  <config-property-name>Config</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>
</config-property>

<config-property>
  <config-property-name>IWayHome</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters</config-property-value>
</config-property>

<config-property>
  <config-property-name>IWayConfig</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>jca_sample</config-property-value>
</config-property>

<config-property>
  <config-property-name>IWayRepoDriver</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>
</config-property>

<config-property>
  <config-property-name>IWayRepoURL</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>
</config-property>

<config-property>
  <config-property-name>IWayRepoUser</config-property-name>
  <config-property-type>java.lang.String</config-property-type>
  <config-property-value>
</config-property>
Updating Adapter Configuration

The parameters defined in the ra.xml file are described in the following table:

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IWayHome</td>
<td>The base installation directory for the OracleWLS packaged application adapter.</td>
</tr>
<tr>
<td>IWayConfig</td>
<td>The adapter configuration name as defined in Application Explorer. For example, Oracle Application Adapter for Siebel has a preconfigured jca_sample configuration in Application Explorer.</td>
</tr>
<tr>
<td>IWayRepoURL</td>
<td>The URL to use when opening a connection to the database. This is necessary only when using an Oracle database as the repository.</td>
</tr>
<tr>
<td>IWayRepoUser</td>
<td>User name to use when connecting to the database. This is necessary only when using an Oracle database as the repository.</td>
</tr>
<tr>
<td>IWayRepoPassword</td>
<td>Password. If provided, it overwrites configuration. This is necessary only when using an Oracle database as the repository.</td>
</tr>
<tr>
<td>LogLevel</td>
<td>It overwrites the level set by the ManagedConnectorFactory property.</td>
</tr>
</tbody>
</table>

Creating Multiple Managed Connector Factory Objects

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

C:\oracle\Middleware\Oracle_SOAI\soa\thirdparty\ApplicationAdapters\iwafjca.rar\META-INF

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

```xml
<?xml version="1.0"?>
<weblogic-connector xmlns="http://www.bea.com/ns/weblogic/90">
    <enable-access-outside-app>true</enable-access-outside-app>
    <enable-global-access-to-classes>true</enable-global-access-to-classes>
</weblogic-connector>
```
<outbound-resource-adapter>
  <default-connection-properties>
    <pool-params>
      <initial-capacity>0</initial-capacity>
    </pool-params>
    <transaction-support>LocalTransaction</transaction-support>
  </default-connection-properties>
</outbound-resource-adapter>

<connection-definition-group>
  <connection-factory-interface>javax.resource.cci.ConnectionFactory</connection-factory-interface>
  <connection-instance>
    <jndi-name>eis/OracleJCAAdapter/DefaultConnection</jndi-name>
  </connection-instance>
</connection-definition-group>
</weblogic-connector>

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

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

  <outbound-resource-adapter>
    <default-connection-properties>
      <pool-params>
        <initial-capacity>0</initial-capacity>
      </pool-params>
      <transaction-support>LocalTransaction</transaction-support>
    </default-connection-properties>
  </outbound-resource-adapter>

  <connection-definition-group>
    <connection-factory-interface>javax.resource.cci.ConnectionFactory</connection-factory-interface>
    <connection-instance>
      <jndi-name>eis/OracleJCAAdapter/DefaultConnection</jndi-name>
    </connection-instance>
    <connection-instance>
      <jndi-name>eis/OracleJCAAdapter/DefaultConnection1</jndi-name>
      <connection-properties>
        <properties>
          <property>
            <name>IWayHome</name>
            <value>C:\oracle\Middleware\Oracle_SOA1\soa\thirddparty\ApplicationAdapters</value>
          </property>
          <property>
            <name>IWayConfig</name>
            <value>jca_sample2</value>
          </property>
          <property>
            <name>IWayRepoURL</name>
            <value></value>
          </property>
          <property>
            <name>IWayRepoUser</name>
            <value></value>
          </property>
        </properties>
      </connection-properties>
    </connection-instance>
  </connection-definition-group>
</weblogic-connector>
```
<property>
  <name>IWayRepoPassword</name>
  <value></value>
</property>
<property>
  <name>LogLevel</name>
  <value>Debug</value>
</property>
</properties>
</connection-properties>
</connection-instance>
</connection-definition-group>
</outbound-resource-adapter>
</weblogic-connector>

If you do not specify a <property> element in the <connection-instance> section, the value is taken from the ra.xml file. You can specify the default properties in the ra.xml file and then override them as required in the weblogic-ra.xml file. In addition, note that the J2CA configuration (for example, jca_sample2) must already be created in Application Explorer.

---

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

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

---

**Modifying WSDL Files for Additional Connection Factory Values**

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

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

eis/OracleJCAAdapter/DefaultConnection

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

The connection factory value must be changed to the following:

eis/OracleJCAAdapter/DefaultConnection1

For example:

```
<jca:address location="eis/OracleJCAAdapter/DefaultConnection1"
  ConnectionSpec="com.ibi.afjca.cci.IWAFConnectionSpec"
  cs.AdapterName="Siebel" cs.Config="isdsrv2_conn2"
  UIConnectionName="Connection1"/>
```

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

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

This chapter includes the following topics:

- Overview
- Deployment of Adapter
- Configuring a New Application Server Connection
- Designing an Outbound BPEL Process for Service Integration
- Designing an Inbound BPEL Process for Event Integration

4.1 Overview

To integrate with Oracle BPEL Process Manager, Oracle Application Adapter for Siebel must be deployed in the same WLS container as Oracle BPEL Process Manager. The underlying adapter services must be exposed as WSDL files, which are generated during design time in Oracle Adapter Application Explorer (Application Explorer) for both request-response (outbound) and event notification (inbound) services of the adapter. See "Generating WSDL (J2CA Configurations Only)" on page 2-32 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 Oracle JDeveloper and deployed to a BPEL server. Upon deployment to the BPEL server, every newly built process is automatically deployed to the Oracle Enterprise Manager console, where you run, monitor, administer BPEL processes, and listen to adapter events.

4.2 Deployment of Adapter

During installation, Oracle Application Adapter for Siebel is deployed as a J2CA 1.0 resource adapter within the WLS container. The adapter must be deployed in the same WLS container as Oracle BPEL Process Manager.
4.3 Configuring a New Application Server Connection

To configure a new Application Server connection in Oracle JDeveloper:

1. Open Oracle JDeveloper on your system.
2. From the menu bar, click View and select Application Server Navigator.

   ![Application Server tab displayed](image)

   The Application Server tab is displayed.

3. Right-click Application Servers, and then select New Application Server.
   The Create Application Server Connection Wizard - Usage page is displayed.

   ![Create Application Server Wizard](image)
4. Accept the default selection (Standalone Server) and click Next.
The Name and Type page is displayed.

5. Specify a new name for the Application Server connection and click Next.
The Authentication page is displayed.
6. Specify a valid user name (for example, weblogic) and a password (for example, welcome1) for your new connection.

7. Click Next.

   The Configuration page is displayed.
8. Specify the Oracle WebLogic host name (for example, localhost), which is the machine IP where the process needs to deploy and Oracle WebLogic domain (for example, base_domain).

9. Click Next.
   The Test page is displayed.

10. Click Test Connection.

11. Ensure that the test status is successful.

12. Click Next.
    The Finish page is displayed.
13. Click Finish.

The new Application Server connection is listed in the left pane (Application Server tab), as shown in the following image.

4.4 Designing an Outbound BPEL Process for Service Integration

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

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

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

Before you design a BPEL process, you must create a schema and generate the respective WSDL file using Application Explorer. See "Generating WSDL (J2CA Configurations Only)" on page 2-32 for more information.
4.4.1 Siebel Service Integration

This section describes how to perform service integration with Siebel. The following topics are included:

- Creating a Request-Response J2CA Service in Application Explorer
- Creating an Integration Object Node

Creating a Request-Response J2CA Service in Application Explorer

To generate WSDL in Application Explorer:

1. Start **Application Explorer** and connect to a defined Siebel target or create a new target.
   
   See **Connecting to a Defined Target** on page 2-16 for more information.

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

3. Expand **Business Object, Account**, and then **Account**.

4. Right-click **queryWithView**, and then select **Create Outbound JCA Service (Request/Response)**.

   The Export WSDL dialog is displayed.

5. Specify an export location on your file system or accept the default path.

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

6. Click **OK**.
You can now create an empty composite for SOA, which is the first step that is required to define a BPEL outbound process in Oracle JDeveloper.

4.4.2 Creating a New SOA Application for the Outbound BPEL Process

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

1. Open Oracle JDeveloper on your system.
2. In the Application Navigator tab, click **New Application**.

The Create SOA Application wizard is displayed.

3. From the Application Template list, click **SOA Application**.
4. Enter name for the new SOA application (for example, Siebel_Outbound_BPEL) and click **Next**.

   The Name your project page is displayed.

![Name your project screenshot](image)

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

   The Configure SOA settings page is displayed.
6. From the Composite Template list, select **Empty Composite** and click **Finish**.

The new SOA application (Siebel_Outbound_BPEL) and associated project (QueryWithView_Invoke) are added to the Application Navigator tab in the left pane.

### 4.4.3 Defining a BPEL Outbound Process

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

1. Configuring a Third Party Adapter Service Component
2. Configuring an Outbound BPEL Process Component

Configuring a Third Party Adapter Service Component
Perform the following steps to create a third party adapter service component:

1. Drag and drop the **Third Party Adapter** component from the Component Palette tab (Service Adapters section) to the External References design pane.

![External References](image)

The Create Third Party Adapter Service dialog is displayed.

![Create Third Party Adapter Service](image)

- **Name:** Query/WEB/New
- **Type:** Reference

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

The SOA Resource Browser dialog is displayed.

![SOA Resource Browser](image)
5. Browse and select an outbound WSDL file from the following directory:
   C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\wsdls

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

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

You are returned to the Create Third Party Adapter Service dialog.

8. Click the Find JCA Files icon, which is located to the right of the JCA File field.

The SOA Resource Browser dialog is displayed.
9. Browse and select the JCA properties file from the following directory:
   C:\oracle\Middleware\Oracle_SOAl\soa\thirdparty\ApplicationAdapters\wsdlsls

10. Click **OK**.
    The following message is displayed.

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

The third party adapter service component (GetDetail) is created in the External References pane, as shown in the following image.

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

**Configuring an Outbound BPEL Process Component**

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

1. Drag and drop the BPEL Process component from the Component Palette tab (Service Components section) to the Components design pane.
The Create BPEL Process dialog is displayed.

2. In the Name field, enter a name to identify the new outbound BPEL process component (for example, querywithview_out).
3. From the Template list, select **Synchronous BPEL Process**.
4. Click the **Browse Input Elements** icon, which is located to the right of the Input field to select the associated XML request schema file.

   The Type Chooser dialog is displayed.

   The Create BPEL Process dialog is displayed.

   ![Create BPEL Process dialog](image)

   The Create BPEL Process dialog is displayed.

   ![Create BPEL Process dialog](image)
5. Expand **Project Schema Files**, `queryWithView_invoke_request.xsd`, and then select **Siebel**.

6. Click **OK**.

   You are returned to the Create BPEL Process dialog.

   ![Type Chooser](image)

   - **Type**: urn:waysoftware:adapter:Siebel:request5/BO/Account/Account/queryWithView/Siebel
   - **Show Detailed Node Information**: unchecked
   - **Help**: button
   - **OK**: button
   - **Cancel**: button

   ![Create BPEL Process](image)

   - **Name**: `queryWithview_out`
   - **Namespace**: `http://emrs.oracle.com/Siebel_Outbound_BPEL_jws/queryWithViewInvoke/queryWithview_out`
   - **Invoke**: Synchronous BPEL Process
   - **Service Name**: `queryWithview_out_client`
   - **Expose as a SOAP service**: checked
   - **Input**: `urn:waysoftware:adapter:Siebel:request5/BO/Account/Account/queryWithView/Siebel`
   - **Output**: `Siebel_Outbound_BPEL_jws/QueryWithViewInvoke/queryWithview_out)processResponse`

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

   The Type Chooser dialog is displayed.
8. Expand Project Schema Files, queryWithView_invoke_response.xsd, and then select SiebelResponse.

9. Click OK.

You are returned to the Create BPEL Process dialog.

10. Click OK.

11. Create a connection between the outbound BPEL process component (querywithview_out) and the third party adapter service component (QueryWithView).
12. Double-click the outbound BPEL process component (querywithview_out) in the Components pane. The following is displayed.

13. Drag and drop the **Invoke** activity component to the Components pane and place it between the **receiveInput** activity component and the **replyOutput** activity component.
14. Create a connection between the new Invoke activity component (Invoke_1) and the third party adapter service component (QueryWithView).

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

The Create Variable dialog is displayed.

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

You are returned to the Edit Invoke dialog.
17. Click the **Automatically Create Output Variable** icon, which is located to the right of the Output field to configure a new output variable.

The Create Variable dialog is displayed.

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

You are returned to the Edit Invoke dialog.
19. Click **Apply** and then **OK**.

The Invoke activity component (Invoke_1) is updated accordingly.

20. Drag and drop the **Assign** activity component to the Components pane and place it between the Receive activity component (receiveInput) and the Invoke activity component (Invoke_1).
21. Double-click the new Assign activity component (Assign_1).
   The Assign dialog is displayed.

22. Click the Copy Operation tab.

23. Click the Plus sign icon and select Copy Operation from the list of available operations.
   The Create Copy Operation dialog is displayed.
24. In the From pane, expand Variables, InputVariable, and then select payload.

25. In the To pane, expand Variables, Invoke_1_queryWithView_InputVariable, and then select input_queryWithView.

26. Click OK.

You are returned to the Assign dialog.

27. Click Apply and then OK.
28. Drag and drop the **Assign** activity component to the Components pane and place it between the Invoke activity (Invoke_1) and the Reply activity (replyOutput).

29. Double-click the new Assign activity component (**Assign_2**).

The Assign dialog is displayed.

30. Click the **Copy Operation** tab.

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

The Create Copy Operation dialog is displayed.
32. In the From pane, expand Variables, Invoke_1_queryWithView_OutputVariable, and then select output_queryWithView.

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

34. Click OK.

You are returned to the Assign dialog.

35. Click Apply and then OK.
The completed activity flow is now displayed.

36. Double-click `composite.xml` in the left pane.

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

### 4.4.4 Deploying the BPEL Outbound Process

Perform the following steps to deploy the BPEL outbound process.

1. Right-click the project name in the left pane (for example, **QueryWithView_Invoke**), select **Deploy**, and then click **QueryWithView_Invoke**.

The Deployment Action dialog is displayed.
2. Ensure that **Deploy to Application Server** is selected.

3. Click **Next**.

   The Deploy Configuration dialog is displayed.

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

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

6. Select a target SOA server and click Next. The Summary dialog is displayed.
7. Review and verify all the available deployment information for your project and click Finish.

The process is deployed successfully.

If an Authorization Request dialog is displayed during the deployment process, provide the required user name and password and click OK.

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

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

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

2. Expand your domain in the left pane followed by the SOA folder.
3. Select an available project (for example, Siebel_Outbound_queryWithView_BPEL).

4. Click Test in the right pane.

The Test Web Service page is displayed.
5. Click the **Request** tab.

6. Scroll down to the Input Arguments section.

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


9. Click **Test Web Service**.

   The output response is received in the Response tab of the Oracle Enterprise Manager console.
4.4.6 Testing Outbound BPEL and Mediator Processes

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

4.5 Designing an Inbound BPEL Process for Event Integration

This topic illustrates Siebel event integration. It describes design-time and run-time configuration. The following tools are required to complete your adapter design-time configuration:

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

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

Before you design a BPEL process, you must generate the respective WSDL file using Application Explorer. See “Generating WSDL for Event Integration” on page 4-36 for more information.
4.5.1 Generating WSDL for Event Integration

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

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.

---

To create a channel:

1. In the left pane, click **Events**.
2. Expand the **Siebel** node.

   The ports and channels nodes appear in the left pane.

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

   The Add Channel dialog is displayed.

---
Perform the following steps:

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

b. Enter a brief description.

c. From the Protocol list, select HTTP Listener, MQ Series Listener, or File Listener.

For demonstration purposes, this procedure uses the HTTP Listener as an example.

4. Click Next.

The Basic dialog is displayed.
5. Enter the system information as specified in the following table:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Listener port</td>
<td>Port on which to listen for Siebel event data.</td>
</tr>
<tr>
<td>Https</td>
<td>For a secure HTTP connection, select the Https check box. This option is currently not supported.</td>
</tr>
<tr>
<td>Synchronization Type</td>
<td>Select REQUEST_RESPONSE from the list, which is the recommended option.</td>
</tr>
<tr>
<td>Encoding Type</td>
<td>Choose an encoding type to be used from the list. By default, ASCII is selected.</td>
</tr>
</tbody>
</table>

6. Click the PreParser tab.

7. Specify the location of the schema file that was generated for the Integration Object node using the Export Schema(s) option in Application Explorer.

   **Note:** During run time, the Oracle Application Adapter for Siebel adds the namespace to the Siebel published document using the schema that is specified in the PreParser tab. If the Schema location field in the PreParser tab is left blank, BPEL and Mediator processes do not work properly as the Siebel published documents do not contain any namespaces.

8. 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 now create an Integration Object node.

Creating an Integration Object Node

1. Start Application Explorer.
2. Expand the Adapters node.
   
   Perform the following steps:
   
   a. Expand the Siebel node.
      
      The defined Siebel targets are displayed under the adapter node.
   
   b. Click the target name, for example, siebel, under the Siebel node.
      
      The Connection dialog displays the values you entered.
   
3. Verify your connection parameters. Provide the required password.
4. Right-click the target name and select Connect.
   
   The x icon disappears, indicating that the node is connected.

5. Expand the Integration Object node and select Sample Account.
6. Right-click the Sample Account node and select Add IO Node.

   The Add IO Node dialog is displayed.

7. Enter a node name (for example, Sample_Account) in the Node name field and a path to the Sample Account XSD file in the Schema location field.

   Please 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.
- **For Siebel 7.5 or later:** Generate XSD schemas directly from Siebel tools. You use the XSD schemas when you create Web services in Application Explorer. After you generate an XSD schema through Siebel tools, use it to create an IO node and a Web service.

- **For Siebel 7.0:** You cannot generate XSD schemas directly from Siebel tools; only XDR schemas can be created. Before you create a Web service, you must first generate an XSD schema from the XDR schema using Application Explorer.

8. If the XSD schema has already been generated, select XSD Schema. If you are using Siebel-generated XDR schemas, do not select the XSD schema option.

9. Select a protocol (HTTP, FILE, or MQ Series) from the Protocol list.

10. Click Continue.

The new Integration Object node is added, as shown in the following image.

---

**Note:** You must restart the Oracle WebLogic Server after the Integration Object node and channel are created.

---

**Generating WSDL for Event Notification**

After you create a channel and an associated Integration Object node, you must generate WSDL for the event using Application Explorer.

You must be connected to a Siebel target under the Adapters node in Application Explorer. See "Establishing a Connection (Target) for Siebel" on page 2-12 for detailed information on how to define and connect to a target.

After you connect to a Siebel target, generate WSDL for the event as follows:

1. Right-click the Integration Object node (for example, Sample_Account), and then select **Create Inbound JCA Service (Event)**.

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

a. In the Name field, specify a name for the WSDL file. The .wsdl file extension is added automatically. By default, the names of WSDL files generated for events end with _receive.

b. From the Channel list, select the channel you created for this inbound service (for example, SiebelEvent).

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

2. Click OK.

4.5.2 Creating a New SOA Application for the Inbound BPEL Process

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

1. Open Oracle JDeveloper on your system.

2. In the Application Navigator tab, click New Application.
The Create SOA Application wizard - Name your application page is displayed.

3. From the Application Template list, click **SOA Application**.
4. Enter name for the new SOA application (for example, Siebel_Inbound_BPEL) and click **Next**.

   The Name your project page is displayed.
5. Enter a project name (for example, SampleAccount_Receive) and click Next. The Configure SOA settings page is displayed.

6. From the Composite Template list, select Empty Composite and click Finish.
The new SOA application (Siebel_Inbound_BPEL) and associated project (SampleAccount_Receive) are added to the Application Navigator tab in the left pane.

### 4.5.3 Defining a BPEL Inbound Process

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

1. Configuring a Third Party Adapter Service Component
2. Configuring an Inbound BPEL Process Component

**Creating a Third Party Adapter Service Component**

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

1. Drag and drop the **Third Party Adapter** component from the Component Palette tab (Service Adapters section) to the Exposed Services pane.

The Create Third Party Adapter Service dialog is displayed.
2. Enter a name for the third party adapter service.
3. Ensure that **Service** is selected from the Type list (default).
4. Click the **Find existing WSDLs** icon, which is located to the right of the WSDL URL field.
   The SOA Resource Browser dialog is displayed.

5. Browse and select an inbound WSDL file from the following directory:
   
   ```plaintext
   C:\oracle\Middleware\home_GA\Oracle_ 
   ```
6. Click OK.

The Localize Files dialog is displayed.

7. Click OK.

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

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

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

   `C:\oracle\Middleware\home_GA\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\wsdls`

10. Click **OK**.

    The following message is displayed.

11. Click **Yes**.

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

    You are returned to the Create Third Party Adapter Service dialog.
12. Click OK.

The third party adapter service component (SampleAccount) is created in the Exposed Services pane, as shown in the following image.

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

**Creating an Inbound BPEL Process Component**

Perform the following steps to create an inbound BPEL process component:
1. Drag and drop the **BPEL Process** component from the Component Palette tab (Service Components section) to the Components pane.

The Create BPEL Process dialog is displayed.

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

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

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

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

The SOA Resource Browser dialog is displayed.
6. Browse and select an inbound WSDL file from the project folder (for example, `Sample_Account_receive.wsdl`).

7. Click OK.
   The Localize Files dialog is displayed.

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

10. Create a connection between the third party adapter service component (SampleAccount) and the inbound BPEL process component (SampleAccount_Inbound).

11. Double-click composite.xml in the left pane.
12. Click the **Save All** icon in the menu bar to save the new inbound BPEL process component that was configured.

You are now ready to deploy the BPEL inbound process.

### 4.5.4 Deploying the BPEL Inbound Process

Perform the following steps to deploy the BPEL inbound process.

1. Right-click the project name in the left pane (for example, **SampleAccount_Receive**), select **Deploy** and then click **SampleAccount.Receive**.
The Deployment Action dialog is displayed.

2. Ensure that **Deploy to Application Server** is selected.
3. Click **Next**.

The Deploy Configuration dialog is displayed.
4. Leave the default values selected and click **Next**.
   The Select Server dialog is displayed.

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

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

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

The process is deployed successfully.

If an Authorization Request dialog is displayed during the deployment process, provide the required user name and password and click OK.

Once event messages are triggered through Siebel, successful instances are received in the Oracle Enterprise Manager console.

4.5.5 Triggering an Event in Siebel

The following topic describes how to trigger an event in Siebel and verify event integration using Oracle Application Adapter for Siebel.

**Triggering a Siebel Event to Test Event Runtime Integration**

To trigger an event in Siebel:

1. Start the Siebel Call Center by entering the following URL in a browser:
   
   http://host name/callcenter/start.swe
2. Click **View** and select **Site Map** from the list.
   The Site Map view is displayed.

3. Click **Siebel Workflow Administration**.
   The Siebel Workflow Administration page is displayed.

4. Click **Workflow Processes**.
   The Workflow Processes page is displayed.
5. Click **Query** to search for the Workflow needed to trigger a Siebel event.

6. Enter a Siebel workflow name and click **Search**.
7. Select the workflow.

8. Click the Process Designer tab and double-click the Send Siebel Quote Data HTTP workflow element.

   The Input Arguments tab is displayed.

9. Enter the IP address and port for the HTTPRequestURLTemplate input argument.

10. Click Return To Designer.
11. Click the Process Simulator tab.

The Simulator tab is displayed.

12. Click **Start** then **Continue** to complete the Siebel event triggering process.

**Verifying the Results**

To verify your results:

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

2. Expand your domain in the left pane followed by the **SOA** folder.
3. Select an available project (for example, Siebel_Inbound_sampaccIonode_BPEL).

4. Click the **Instances** tab in the right pane.
   Recently received run-time events are displayed in the Instances tab.

5. Select a Siebel instance ID.
   The Flow Trace page is displayed.
6. Select a component instance to view its detailed audit trail.
   The Instance page for the selected component is displayed.

   7. Click the Audit Trail tab to view the event message.
      The message received from the Siebel system is displayed in the Audit Trail tab.
Integration With Mediator Service Components in the Oracle SOA Suite

This chapter contains the following examples:

- Configuring a New Application Server Connection
- Configuring a Mediator Outbound Process
- Configuring a Mediator Inbound Process

The scenarios shown in this chapter require the following prerequisites.

**Prerequisites**

The following are installation and configuration requirements:

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

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

- How to configure Oracle Application Adapter for Siebel for services and events. See Chapter 2, "Configuring Oracle Application Server Adapter for Siebel".
- How to configure Oracle JDeveloper. For more information, see Chapter 4, "Integration With BPEL Service Components in the Oracle SOA Suite".
- How to use Siebel workflows. See Appendix A, "Using Siebel Workflows" for information on Siebel design requirements.

**Overview of Mediator Integration**

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

Adapter integration with Oracle WebLogic Server, Mediator is a two-step process:

1. **Design Time**: Oracle Application Adapter for Siebel is configured in Application Explorer for services and events, as described in Chapter 2, "Configuring Oracle..."
Application Server Adapter for Siebel”. Integration logic is modeled in iStudio. Metadata are stored in repositories.

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

### 5.1 Configuring a New Application Server Connection

To configure a new Application Server connection in Oracle JDeveloper:

1. Open Oracle JDeveloper on your system.

2. From the menu bar, click **View** and select **Application Server Navigator**.

The Application Server tab is displayed.

3. Right-click **Application Servers**, and then select **New Application Server**.

The Create Application Server Connection Wizard - Usage page is displayed.
4. Accept the default selection (Standalone Server) and click **Next**.

The Name and Type page is displayed.

5. Specify a new name for the Application Server connection and click **Next**.

The Authentication page is displayed.
6. Specify a valid user name (for example, weblogic) and a password (for example, welcome1) for your new connection.

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

9. Click Next.
   The Test page is displayed.

10. Click Test Connection.

11. Ensure that the test status is successful.

12. Click Next.
   The Finish page is displayed.
13. Click Finish.

The new Application Server connection is listed in the left pane (Application Server tab), as shown in the following image.

5.2 Configuring a Mediator Outbound Process

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

Prerequisites

Before you design a Mediator outbound process, you must generate the respective WSDL file using Application Explorer. See “Siebel Service Integration” on page 4-7 for more information.

5.2.1 Creating a New SOA Application for the Outbound Mediator Process

Perform the following steps to create a new SOA application for the outbound Mediator process:
1. Open Oracle JDeveloper on your system.
2. In the Application Navigator tab, click **New Application**.

   ![Oracle JDeveloper](image)

   The Create SOA Application wizard - Name your application page is displayed.

3. From the Application Template list, click **SOA Application**.
4. Enter name for the new SOA application (for example, Siebel_Outbound_Mediator) and click **Next**.

   The Name your project page is displayed.
5. Enter a project name (for example, QueryWithView_Invoke) and click **Next**. The Configure SOA settings page is displayed.

6. From the Composite Template list, select **Empty Composite** and click **Finish**.
The new SOA application (Siebel_Outbound_Mediator) and associated project (QueryWithView_Invoke) are added to the Application Navigator tab in the left pane.

5.2.2 Defining a Mediator Outbound Process

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

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

**Configuring a Third Party Adapter Service Component**

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

1. Drag and drop the Third Party Adapter component from the Component Palette tab (Service Adapters section) to the External References pane.
The Create Third Party Adapter Service dialog is displayed.

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

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

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

   The SOA Resource Browser dialog is displayed.
5. Browse and select an outbound WSDL file from the following directory:
   C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\wsdls

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

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

You are returned to the Create Third Party Adapter Service dialog.

8. Click the **Find JCA Files** icon, which is located to the right of the JCA File field.

The SOA Resource Browser dialog is displayed.
9. Browse and select the JCA properties file from the following directory:

   C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\wsdls

10. Click OK.

    The following message is displayed.

    ![]([image]

    11. Click Yes.

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

    You are returned to the Create Third Party Adapter Service dialog.
12. Click OK.

The third party adapter service component (GetDetail) is created in the External References pane, as shown in the following image.

---

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

**Configuring an Outbound Mediator Process Component**

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

1. Drag and drop the Mediator component from the Component Palette tab (Service Components section) to the Components pane.
2. In the Name field, enter a name to identify the new outbound Mediator process component (for example, QueryWithView_Mediator).

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

4. Click the **Browse Input Elements** icon, which is located to the right of the Input field to select the associated XML request schema file. The Type Chooser dialog is displayed.
5. Expand Project WSDL Files, queryWithView_invoke.wsdl, Inline Schemas, schema, and select Siebel.

6. Click OK.

You are returned to the Create Mediator dialog.

7. Click the Browse Output Elements icon, which is located to the right of the Output field to select the associated XML response schema file.
The Type Chooser dialog is displayed.

8. Expand Project WSDL Files, queryWithView_invoke.wsdl, Inline Schemas, schema, and select SiebelResponse.

9. Click OK.

You are returned to the Create Mediator dialog.
10. Click OK.

11. Create a connection between the outbound Mediator process component (QueryWithView_Mediator) and the third party adapter service component (QueryWithView).

You are now ready to configure the routing rules.

**Configuring the Routing Rules**

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

1. Double-click the outbound Mediator process component (QueryWithView_Mediator) in the Components pane.
2. In the <<Filter Expression>> area, click the **Select an existing mapper file or create a new one** icon to the right of the Transform Using field.

   The Request Transformation Map dialog is displayed.

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

   The Siebel_To_Siebel.xsl tab is displayed.

   The Auto Map Preferences dialog is displayed.

5. Retain the default values and click **OK**.

6. Click the **QueryWithView_Mediator.mplan** tab.

You are returned to the Routing Rules dialog.
7. In the Synchronous Reply area, click the Select an existing mapper file or create a new one icon to the right of the Transform Using field. The Reply Transformation Map dialog is displayed.

8. Select the Create New Mapper File option and click OK. The SiebelResponse_To_SiebelResponse.xsl tab is displayed.

10. Retain the default values and click **OK**.

The mapping is complete, as shown in the following image.

![Auto Map Preferences dialog box](image)

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

5.2.3 Deploying the Mediator Outbound Process

Perform the following steps to deploy the Mediator outbound process.

1. Right-click the project name in the left pane (for example, QueryWithView_Invoke), select Deploy, and then click QueryWithView_Invoke.

The Deployment Action dialog is displayed.
2. Ensure that **Deploy to Application Server** is selected.

3. Click **Next**.

   The Deploy Configuration dialog is displayed.

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

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

6. Select a target SOA server and click Next. The Summary dialog is displayed.
7. Review and verify all the available deployment information for your project and click Finish.

The process is deployed successfully.

If an Authorization Request dialog is displayed during the deployment process, provide the required user name and password and click OK.

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

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

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

2. Expand your domain in the left pane followed by the SOA folder.
3. Select an available project (for example, Siebel_Outbound_queryWithView_Mediator).

4. Click Test in the right pane.

   The Test Web Service page is displayed.
5. Click the Request tab.

6. Scroll down to the Input Arguments section.

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


9. Click Test Web Service.

   The output response is received in the Response tab of the Oracle Enterprise Manager console.
5.3 Configuring a Mediator Inbound Process

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

Prerequisites
Before you design a Mediator inbound process, you must generate the respective WSDL file using Application Explorer. See "Generating WSDL for Event Integration" on page 4-36 for more information.

5.3.1 Creating a New SOA Application for the Inbound Mediator Process

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

1. Open Oracle JDeveloper on your system.
2. In the Application Navigator tab, click New Application.

The Create SOA Application wizard is displayed.
3. From the Application Template list, click **SOA Application**.
4. Enter name for the new SOA application (for example, Siebel_Inbound_Mediator) and click **Next**.

   The Name your project page is displayed.
5. Enter a project name (for example, SampleAccount_Receive) and click Next. The Configure SOA settings page is displayed.

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

The new SOA application (Siebel_Inbound_Mediator) and associated project (SampleAccount_Receive) are added to the Application Navigator tab in the left pane.

5.3.2 Defining a Mediator Inbound Process

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

2. Configuring an Inbound Mediator Process Component With a File Adapter

3. Configuring the Routing Rules

Configuring a Third Party Adapter Service Component

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

1. Drag and drop the Third Party Adapter component from the Component Palette tab (Service Adapters section) to the Exposed Services pane.

The Create Third Party Adapter Service dialog is displayed.

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

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

4. Click the Find existing WSDLs icon, which is located to the right of the WSDL URL field.
The SOA Resource Browser dialog is displayed.

5. Browse and select an inbound WSDL file from the following directory:
   
   C:\oracle\Middleware\home_GA\Oracle_SOA\soa\thirdparty\ApplicationAdapters\wsdls

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

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

You are returned to the Create Third Party Adapter Service dialog.

8. Click the **Find JCA Files** icon, which is located to the right of the JCA File field. The SOA Resource Browser dialog is displayed.

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

   C:\oracle\Middleware\home_GA\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\wsdls
10. **Click OK.**

    The following message is displayed.

    ![Image of Copy File dialog]

    The jca file Sample_Account_receive.jca is external to the current project. In order to make this file available to your project at runtime, JDeveloper can now make a local copy. Do you want to copy file to Sample_Account_receive_3P.jca?

    ![Yes and No buttons]

11. **Click Yes.**

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

    You are returned to the Create Third Party Adapter Service dialog.

    ![Image of Create Third Party Adapter Service dialog]

12. **Click OK.**

    The third party adapter service component (SampleAccount) is created in the Exposed Services pane, as shown in the following image.
You are now ready to configure an inbound BPEL process component.

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

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

1. Drag and drop the **Mediator** component from the Component Palette tab (Service Components section) to the Components pane.

The Create Mediator dialog is displayed.
2. In the Name field, enter a name to identify the new inbound Mediator process component (for example, SampleAccount_Mediator).

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

4. Click the **OK**.

   The new Mediator process component is added to the Components pane, as shown in the following image.

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

   The Adapter Configuration Wizard - Welcome page is displayed, showing the Welcome page.
6. Click Next.
   The Service Name page is displayed.
7. Type a name for the new File adapter in the Service Name field and click Next. The Adapter Interface page is displayed.

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

9. Click Next. The Operation page is displayed.
10. Select **Write File** from the list of Operation Type options and specify an Operation Name (for example, Write).

11. Click **Next**.

   The File Configuration page is displayed.
12. Specify a location on your file system for the output file.

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

14. Click Next.

The Messages page is displayed.
15. Click **browse for schema file**, which is located to the right of the URL field.

   The Type Chooser dialog is displayed.
16. Expand Project WSDL Files, Sample_Account_receive.wsdl, Inline Schemas, and schema.

17. Select the available schema (for example, SiebelMessage)

18. Click OK.

   You are returned to the Messages page.

19. Click Next.

   The Finish page is displayed.
20. Click Finish.

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

22. Create a connection between the inbound Mediator process component and the File adapter component.
You are now ready to configure the routing rules.

**Configuring the Routing Rules**

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

1. Double-click the inbound Mediator process component in the Components pane.

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

   The Request Transformation Map dialog is displayed.

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

   The SiebelMessage_To_SiebelMessage.xsl tab is displayed.


   The Auto Map Preferences dialog is displayed.
5. Retain the default values and click **OK**.

The mapping is now complete.

6. Click the **Save All** icon in the menu bar to save the new inbound Mediator process component that was configured.
You are now ready to deploy the Mediator inbound process. You can follow the same procedure that is described in "Defining a BPEL Inbound Process" on page 4-44.

Once event messages are triggered through Siebel, output XML is received in the location that was specified for the File adapter component. For more information on triggering events in Siebel, see "Triggering an Event in Siebel" on page 4-55.
This chapter explains the limitations and workarounds when connecting to Siebel. The following topics are discussed:

- **Troubleshooting**
- **BSE Error Messages**

## 6.1 Troubleshooting

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

- General Usage Notes for the Oracle Application Adapter for Siebel
- Application Explorer
- Siebel
- Oracle Adapter J2CA
- Oracle Adapter Business Services Engine (BSE)

---

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

- The Oracle Adapter J2CA trace information can be found under the following directory:
  
  C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\config\config_name\log

- BSE trace information can be found under the following directory:
  
  C:\oracle\Middleware\home_GA\user_projects\domains\base_domain\servers\soa_server1\stage\ibse\ibse.war\ibselogs

- The log file for Application Explorer can be found under the following directory:
  
  C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\tools\iwae\bin

---

**General Usage Notes for the Oracle Application Adapter for Siebel**

The Oracle Application Adapter for Siebel is subject to the following limitations:

- The HTTPS protocol is not supported for services and events.
Updates for multi-value (MVG) fields with join specifications are not supported.

When a connection is lost, the adapter does not automatically reconnect to Siebel.

**Application Explorer**

To use Application Explorer on **Windows** for debugging or testing purposes:

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

2. Start Application Explorer by clicking the Windows **Start** menu, selecting **All Programs, Oracle Application Adapters**, and clicking **Application Explorer**.

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

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

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

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

<table>
<thead>
<tr>
<th>Error</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Siebel does not appear in the Application Explorer Adapter node list.</td>
<td>Ensure that the Siebel jar files supplied with your Siebel distribution media have been placed in the C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\lib directory. For example, for Siebel 7.03 environments, the SiebelJI_Common.jar and SiebelJI_enu.jar should be placed in this directory.</td>
</tr>
<tr>
<td>Target Type drop down contains only Java Data Bean Connection and COM connection type is desired.</td>
<td>Ensure that the Siebel thin client is installed correctly on the system hosting Application Explorer so that appropriate COM environment is available.</td>
</tr>
<tr>
<td>An error message that includes the name of the Siebel Gateway server appears when you try to connect to a Siebel target. For example, Problem activating adapter (&lt;server_name&gt;). Check logs for more information.</td>
<td>Ensure that the name of the Siebel Gateway server is correctly defined for the target you are using.</td>
</tr>
<tr>
<td>You receive the following error when trying to connect to a Siebel target: Problem activating adapter. (You have entered an invalid set of logon parameters. Please type in your logon parameters again.). Check logs for more information.</td>
<td>Ensure that the User ID and password parameter values to connect to your Siebel system are correct.</td>
</tr>
</tbody>
</table>
## Error Solution

<table>
<thead>
<tr>
<th>Error</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>You receive the following error when trying to connect to a Siebel target: Problem activating adapter. ( Couldn’t get nameserver connection). Check logs for more information.</td>
<td>Check on network connectivity to Siebel environment. Correct networking problem and retry connection.</td>
</tr>
<tr>
<td>You receive the following error when attempting to connect to a Siebel target: Problem activating adapter. (NSReadKey request failed (no error information)...). Check logs for more information.</td>
<td>Ensure that the values defined for Siebel Server, Enterprise Name, and Object Manager for the target you are using are correct, and retry the connection.</td>
</tr>
<tr>
<td>You receive the following error when attempting to connect to a Siebel target: Problem activating adapter. (Error loading translatable messages: com.siebel.locale.enux.messages.SSMessages_enux). Check logs for more information.</td>
<td>Ensure that the value of the Language parameter on the Advanced tab is defined correctly for the target you are using to connect to your Siebel system (for example, enu for English).</td>
</tr>
<tr>
<td>A successful connection is made to Siebel environment but no values are available in Business Object, Business Service, and Integration Object nodes in Application Explorer tree.</td>
<td>The Repository Name specified on the Advanced tab in the Siebel target configuration is either void or empty of any components in the targeted Siebel environment or that Repository Name is not valid for the targeted Siebel environment. Verify that the Repository Name is valid and contains components for interrogation then re-connect.</td>
</tr>
<tr>
<td>Logon failure error at run-time.</td>
<td>If the password for connecting to your Siebel system is not specified when creating a target or with the Edit option in Application Explorer, you are unable to connect to Siebel. 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>The following exception occurs when you start Application Explorer by activating ae.bat (not iaexplorer.exe): java.lang.ClassNotFoundException: org.bouncycastle.jce.provider.BouncyCastleProvider</td>
<td>This is a benign exception. It does not affect adapter functionality. Download BouncyCastle files from: ftp://ftp.bouncycastle.org/pub</td>
</tr>
</tbody>
</table>
### Troubleshooting

**Siebel**

The error messages listed can occur when using the adapter with either a BSE or Oracle Adapter J2CA repository project.

<table>
<thead>
<tr>
<th>Error</th>
<th>Solution</th>
</tr>
</thead>
</table>
| 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.<init>(JCATransport.java:62) at com.iwaysoftware.iwae.common.AdapterClient.<init>(AdapterClient.java:85) at com.ibi.bse.ConfigWorker.run(ConfigWorker.java:41) at java.lang.Thread.run(Thread.java:534) | **JAVACMD** is not set on the user system. Before starting Application Explorer, export **JAVACMD** as follows:  
**JAVACMD=*/<jdk_home>/bin/java, where  
<jdk_home> is the directory where JDK is installed on your system. |

**Error Solution**

A successful connection is made to Siebel environment but no values are available in Business Object, Business Service, and Integration Object nodes in Application Explorer tree.

When executing a request, the following error message appears:

```java
AdapterException: Unsupported Action: {0} Tquery
```

Verify that method is available for specific request by verifying schema.

When executing a request, the following error message appears:

```java
AdapterException: Field 'NFame' does not exist in definition for business component 'Account'. Please ask your systems administrator to check your application configuration.
```

Ensure that field names are valid within request document by referring to schema for that specific object, and then re-submit the request.
This topic discusses the different types of errors that can occur when processing Web services through BSE.

### 6.2.1 General Error Handling in BSE

BSE serves as both a SOAP gateway into the adapter framework and as the engine for some of the adapters. In both design time and run-time, various conditions can cause errors in BSE when Web services that use adapters run. Some of these conditions and resulting errors are exposed the same way, regardless of the specific adapter; others are exposed differently, based on the adapter being used. This topic explains what you can expect when you encounter some of the more common error conditions on an adapter-specific basis.

Usually the SOAP gateway (agent) inside BSE passes a SOAP request message to the adapter required for the Web service. If an error occurs, 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 fault, 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:Envelope>
```

### Oracle Adapter J2CA

<table>
<thead>
<tr>
<th>Error</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>In Application Explorer, the following error message appears when you attempt to connect to an Oracle Adapter J2CA configuration:</td>
<td>In the Details tab in the right pane, ensure that the directory specified in the Home field points to the correct directory, for example:</td>
</tr>
<tr>
<td>Could not initialize JCA</td>
<td>C:\oracle\Middleware\Oracle_SOA\soa\thirdparty\ApplicationAdapters\tools\iwaebin.....</td>
</tr>
</tbody>
</table>

**Error Solution**

In Application Explorer, the following error message appears when you attempt to connect to an Oracle Adapter J2CA configuration:

```
Could not initialize JCA
```

The method argument `XMLCharEncoding` is not supported. Leave this element blank in the XML payload.
In this example, BSE did not receive an element in the SOAP request message that is mandatory for the WSDL for this Web service.

6.2.2 Adapter-Specific Error Handling

When an adapter raises an exception during run-time, the SOAP agent in BSE produces a SOAP fault element in the generated SOAP response. The SOAP fault element contains fault code and fault string elements. The fault string contains the native error description from the adapter target system. Since adapters use the target system interfaces and APIs, whether an exception is raised depends on how the target systems interface or API treats the error condition. If a SOAP request message is passed to an adapter by the SOAP agent in BSE, and that request is invalid based on the WSDL for that service, 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.

**Oracle Application Adapter for Siebel Invalid SOAP Request**

If Oracle Application Adapter for Siebel receives a SOAP request message that does not conform to the WSDL for the Web services being executed, then the following SOAP response is generated:

```xml
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Body>
<SOAP-ENV:Fault>
    <faultcode>SOAP-ENV:Server</faultcode>
    <faultstring>XD[FAIL] Parse failure (IS) 3: org.xml.sax.SAXParseException: Premature end of file.</faultstring>
</SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

**Empty Result From Siebel Request**

If Oracle Application Adapter for Siebel cannot connect to Siebel when executing a Web service, then the following SOAP response is generated:

```xml
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Body>
<SOAP-ENV:Fault>
    <faultcode>SOAP-ENV:Server</faultcode>
    <faultstring><Exception> - major:4096 minor: -1 message:NSReadKey request 11 was abandoned after 37846ms connection:12a due to Connection shutdown request Connection reset by peer:JVM_recv in socket input stream stream read DetailedMessage:Unknown</Exception>
</SOAP-ENV:Fault>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```
Oracle WebLogic Server Integration Adapters
Oracle Adapters connect BSE to adapters whose engines are other Oracle servers. Therefore, since this type of adapter is used to connect BSE to many different target systems, the error handling behavior is consistent. Check the user guide for your adapter to see if you require the Oracle WebLogic Server Integration Adapter when running Web services.

Invalid SOAP Request
If Oracle Application Adapter for Siebel receives a SOAP request message that does not conform to the WSDL for the Web services being executed, then the following SOAP response is generated.

<?xml version="1.0" encoding="ISO-8859-1" ?>
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <faultcode>SOAP-ENV:Server</faultcode>
      <faultstring>RPC server connection failed: Connection refused: connect</faultstring>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

Empty Result From Oracle WebLogic Server Adapter Request
If Oracle Application Adapter for Siebel executes a SOAP request using input parameters passed that do not match records in the target system, then the following SOAP response is generated.

Note: The condition for this adapter does not yield a SOAP fault.
This chapter includes the following topics:

- Web Services Policy-Based Security
- Migrating Repositories

### 7.1 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. Easy application integration is enabled. However, the issue of controlling the use and implementation of critical and sensitive business logic that is run as a Web service is raised.

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

A policy is a set of privileges dealing with the execution of a Business Service (iBS) that can be applied to an existing or new iBS. When you set specific rights or privileges inside a policy, you do not have to re-create privileges for every iBS 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 effect the run-time behavior of the Web services to which they have been applied.

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

When a policy is not applied, the default value for an iBS is to "grant all". For example, anybody can execute the iBS, until the Resource Execution policy is associated to the iBS. 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 iBS.
7.1.1 Configuring Web Services Policy-Based Security

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

Creating and Associating a User with a Policy

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

1. Start Application Explorer.

2. Right-click the configuration to which you want to connect, for example, SampleConfig. See Chapter 2, "Configuring Oracle Application Server Adapter for Siebel" for information on creating a new configuration.

3. Select Connect.

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

   Perform the following steps:

   a. Expand the Business Services node.

   b. Expand the Configuration node.

   c. Expand the Security node.

   d. Expand the Users and Groups node.

4. Right-click Users and then New User.

   The New User dialog box is displayed.
Perform the following steps:

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

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

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

5. Click OK.

The new user is added under the Users node.

Creating a Group to Use With a Policy

To create a group to use with a policy:

1. Start Application Explorer.

2. Right-click the configuration to which you want to connect, for example, SampleConfig. See Chapter 2, "Configuring Oracle Application Server Adapter for Siebel" for information on creating a new configuration.

3. Select Connect.

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

Perform the following steps:

a. Expand the Business Services node.

b. Expand the Configurations 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.
Perform the following steps:

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

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

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

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

The new group is added under the Group node.

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

To create an execution policy:

1. Start Application Explorer.

2. Right-click the configuration to which you want to connect, for example, SampleConfig. See Chapter 2, "Configuring Oracle Application Server Adapter for Siebel" for information on creating a new configuration.

3. Select Connect.

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

Perform the following steps:
a. Expand the Business Services node.
b. Expand the Configurations node.
c. Expand the Security node.
d. Expand the Policies node.

4. Right-click Policies and select New Policy.

The New policy dialog box is displayed.

Perform the following steps:

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

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

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

The New Policy permissions dialog box is displayed.

7. When you have selected at least one user, click OK.
8. Click Next.
The New Policy permissions dialog box is displayed.

![New Policy dialog box with examples of user and group permissions]

- To grant permission to a user or group to execute an iBS, select the user or group and move them into the Execution Granted list by selecting the double left facing arrow.
- To deny permission to a user or group to execute an iBS, select the user or group and move them into the Execution Denied list by selecting the double right facing arrow.

9. Click OK.

The following pane summarizes your configuration.

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

### Using the IP and Domain Restrictions Policy Type

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

1. Start Application Explorer.

2. Right-click the configuration to which you want to connect, for example, SampleConfig. See Chapter 2, "Configuring Oracle Application Server Adapter for Siebel" for information on creating a new configuration.

3. Select Connect.

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

a. Expand the Business Services node.

b. Expand the Configurations 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.

- 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.
7.2 Migrating Repositories

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

**Migrating a BSE Repository**

To migrate a BSE repository:

1. Copy the BSE control service URL, for example:
   
   ```
   http://localhost:8001/ibse/IBSEServlet/admin/iwcontrol.ibs
   ```

2. Open a third party XML editor, for example, XMLSPY.

3. From the menu bar, click SOAP.

   A list of options appears.

   ![SOAP Options](image)

4. Select Create new SOAP request.

   The WSDL file location dialog box is displayed.

   ![WSDL File Location Dialog](image)

   Perform the following steps:

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

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

   ```
   http://localhost:8001/ibse/IBSEServlet/admin/iwcontrol.ibs?wsdl
   ```

5. Click OK.

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

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

The following window is displayed, 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 the Text view icon.

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

9. Locate the following section:

   ```xml
   <m:MIGRATEREPO xmlns:m="urn:schemas-iwaysoftware-com:jul2003:ibse:config" version=""">
   <m:repositorysetting>
     <m:rname>oracle</m:rname>
     <m:rconn>String</m:rconn>
     <m:rdriver>String</m:rdriver>
   </m:repositorysetting>
   ```
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>SiebelService1</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>SiebelService1</m:servicename>
     <m:servicename>SiebelService2</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.

   Your BSE repository and any Web services you selected 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:

   `C:\oracle\Middleware\Oracle_SOA1\soa\thirdparty\ApplicationAdapters\config\JCA_CONFIG`

   Where `JCA_CONFIG` is the name of your J2CA configuration.
2. Locate and copy the repository.xml file.

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

Your J2CA repository is migrated to the new J2CA configuration directory.
When using Siebel XML to integrate with Siebel Integration Objects, the interface uses a Siebel Workflow.

**Note:** This section is intended as a supplement to the documentation designed for Oracle Application Adapter for Siebel user and is not intended as a substitute for Siebel documentation. For complete and up-to-date information on Siebel Workflow and policy topics, see the Siebel Bookshelf for your Siebel system.

## A.1 Overview

A Siebel Workflow is defined within Siebel to emit or to receive Siebel XML. In either case, emitting or receiving is handled by Siebel transport services for MQSeries, File, or HTTP. The following topics describe the use and creation of workflows that employ the supported transport services.

### A.1.1 Siebel Workflows

A Siebel Workflow is a series of Siebel Business Services linked together to accomplish a business task. You create workflows using the Siebel Client Workflow Administration screens. Workflows are invoked through one of the following methods:

- Using a workflow policy
- Using a run-time event (Siebel Event)
- Using a script (eScript or Siebel VB)

The following topic briefly describes how to invoke the workflow through a policy condition.

**See Also:**

Siebel Bookshelf documentation for more information on policy and other methods.

### A.1.2 Using a Policy to Invoke a Siebel EAI Workflow

A workflow policy is defined by a set of conditions that performs a set of defined actions. A Siebel workflow policy consists of:

- Conditions that define circumstances, based on changes in the state of a Siebel database.
Actions that define steps taken when conditions are fulfilled.

Creating a policy to invoke a workflow as an action involves the following steps:

1. Define an action to be executed after a policy is triggered. Use the Run Integration Process program.
2. Create a policy by setting conditions and selecting appropriate policy groups and actions.
3. Activate the policy by choosing an activation date.
4. Run the Generate Triggers server task from Server Administration windows to set the conditions to be monitored.
5. Start the Workflow Monitor agent after editing with the appropriate policy group (to which your policy belongs) to evaluate whether to perform an action.
6. Start the Workflow Action Agent server task from Server Administration windows to perform the action.

A.1.3 Siebel Workflow - Outbound

When a Siebel Workflow is triggered based on a Siebel policy, run-time, or script (eScript or Siebel VB) event, the result is the generation of a Siebel XML document that is placed on one of the Siebel transports. For example, when you add a new account in the Siebel Call Center application, you can design and configure a workflow to be triggered on the account transaction. You can design the workflow to extract the data for the new record, convert it to Siebel XML, and then, place it on an MQSeries message queue.

In this example, the Siebel Workflow process executes the following series of Siebel Business Services:

1. Calls the Siebel EAI Siebel Adapter that queries for the newly updated account record and places the data in its original internal structure into memory.
2. Calls the Siebel EAI XML Converter that converts the data into an XML message.
3. Calls the Siebel EAI MQSeries Transport that places the newly created XML message into the appropriate MQSeries message queue.

After the message is placed in the message queue, it is retrieved by Oracle Application Adapter for Siebel 6.3 and higher. The following Workflow sequence illustrates the previous steps.

A.1.4 Siebel Workflow - Inbound

A Siebel Workflow that is triggered by an external event begins by receiving a Siebel XML document placed on one of its transports. The result might be the update of a Siebel record using the XML as input, for example, when a new account is added in another CRM system but also must be updated in the Siebel Call Center application.
You can design and configure a Workflow to receive or listen on an MQSeries message queue. Upon receipt of the XML message, the Workflow processes the transaction into the Siebel system to update the record.

In this example, upon receipt of the Siebel XML message in the message queue, the Siebel MQSeries Receiver server task initiates a Siebel Workflow process, which in turn executes a series of Siebel Business Services as follows:

1. Calls the Siebel EAI XML Converter, which converts the XML message into Siebel internal format.

2. Calls the Siebel EAI Siebel Adapter, which applies the newly updated account record based on the methods defined in its service.

The following is a sample of the Workflow process.

A.2 Creating a Siebel Workflow

The following topics include procedures for creating Siebel Workflows in the Siebel Workflow Administration window.

A.2.1 Creating a Siebel Workflow for an Event Using MQSeries Transport

The following procedure is an example of a Siebel Workflow illustrated in the Siebel Workflow Administration window. The Workflow was designed for exporting Siebel Account record information using the MQSeries transport.

The following is a Siebel Workflow Administration window.
The following procedure describes how to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center application. The Workflow is then placed on an MQSeries message queue.

To create a Siebel Workflow:

1. In the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties.
   
The Account message contains Siebel Account data in hierarchical format.
   
Account XML specifies the Siebel Account data that the workflow has converted to XML.
   
The following window is displayed, showing the Process Properties tab active.
2. Use the Siebel Workflow Administration windows to create a Workflow.


   The Business Service obtains the Account information from Siebel using the Query method.

   Output from this Business Service is generated in hierarchical format.
4. Define an EAI XML Converter Business Service step and call it **Convert to XML**.

   It is defined to receive the Account data from the EAI Siebel Adapter Business Service in hierarchical format and convert it to XML format.
5. Define an EAI MQSeries server transport Business Service step and call it Send to Q.

It is defined to receive the Account data from the EAI XML Converter Business Service in Siebel XML format and send the Account XML to MQSeries using the Send method.

A.2.2 Creating a Siebel Workflow for an Event Using File Transport

The following procedure is an example of a Siebel Workflow illustrated in the Siebel Workflow Administration window. The Workflow was designed for exporting Siebel Account record information using the File transport.

The following window is displayed with the Process Designer tab active.
This procedure describes how to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center application and then places Siebel XML on the file system.

To create a Siebel Workflow:
   Account message contains the Siebel Account data in hierarchical format.
   Account XML specifies which Siebel Account data the Workflow converted to XML.

2. Use the Siebel Workflow Administration windows to create a Workflow.
   The following is an example of a Siebel Workflow Administration window.

   The Business Service obtains the Account information from Siebel using the Query method.
   Output from this Business Service is generated in hierarchical format.
4. Define an EAI XML Converter Business Service step and call it `Convert Account Data to XML`.

   This Business Service is defined to receive the Account data from the EAI Siebel Adapter Business Service in hierarchical format and convert it to XML format.
5. Define an EAI File Transport Business Service step and call it **Send Account Data**.

This Business Service is defined to receive the Account data from the EAI XML Converter Business Service in Siebel XML format and send the Account XML to the file system in a specified directory using the Send method.

### A.2.3 Creating a Siebel Workflow for an Event Using HTTP Transport

The following procedure is an example of a Siebel Workflow illustrated in the Siebel Workflow Administration window. The Workflow was designed for exporting Siebel Account record information using the HTTP transport.

This procedure describes how to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center application.

To create a Siebel Workflow:

1. In the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties.

   Account message contains the Siebel Account data in hierarchical format.

   Account XML specifies the Siebel Account data that the Workflow has converted to XML.

2. Use the Siebel Workflow Administration windows to create a Workflow.

The Business Service obtains the Account information from Siebel using the Query method.

Output from this Business Service is generated in hierarchical format.

4. Define an EAI XML Converter Business Service step and call it Convert to XML.

This Business Service is defined to receive the Account data from the EAI Siebel Adapter Business Service in hierarchical format and convert it to XML format.
5. Define an EAI HTTP Transport Business Service step and call it \texttt{Send - HTTP}.

This Business Service is defined to receive the Account data from the EAI XML Converter Business Service in Siebel XML format and send the Account XML to HTTP using the Send method.

\textbf{A.2.4 Creating a Siebel Workflow for a Service Using MQSeries Transport}

The following procedure is an example of a Siebel Workflow illustrated in the Siebel Workflow Administration window. The Workflow was designed for importing Siebel Account record information through the MQSeries Transport.

The following is a sample Siebel Workflow Administration window.
This procedure describes how to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center application.

To create a Siebel Workflow:
1. In the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties.

Account message contains the Siebel Account data in hierarchical format.

Account XML specifies the Siebel Account data that the Workflow converted to XML.

2. Define an EAI MQSeries Server Transport Business Service step and call it Receive.

The Business Service is defined to receive the Account data from the MQSeries message queue.

The EAI MQSeries Server Transport Business Service receives the Account data in Siebel XML format and sends it to the EAI XML Converter Business Service.
3. Define an EAI XML Converter Business Service step and call it *Get XML from MQ & Convert to XML*.  

This Business Service is defined to receive the Account data from the EAI MQSeries Server Transport Business Service in XML format and convert it to hierarchical format.
4. Define an EAI Siebel Adapter Business Service step and call it Update Account. This Business Service is defined to receive from the EAI XML Converter Business Service the instance of Account data in hierarchical format.

The Business Service applies the Account information into Siebel using the Insert or Update method.

A.2.5 Creating a Siebel Workflow for a Service Using File Transport

The following procedure is an example of a Siebel Workflow illustrated in the Siebel Workflow Administration window. The workflow was designed for importing Siebel Account record information through the File transport.

This procedure describes how to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center application and then places Siebel XML on the file system.

The following is a Siebel Workflow Administration window with the Process Designer tab active.

To create a Siebel Workflow:
1. In the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties.

Account message contains the Siebel Account data in hierarchical format.

Account XML specifies the Siebel Account data that the workflow converted to XML.

The Business Service is defined to receive the Account data from the file system.

The EAI File Transport Business Service receives the Account data in Siebel XML format and sends it to the EAI XML Converter Business Service.

3. Define an EAI XML Converter Business Service step and call it Convert from XML.

This Business Service is defined to receive the Account data from the EAI File Transport Business Service in XML format and convert it to hierarchical format.
4. Define an EAI Siebel Adapter Business Service step and call it **Update** or **Insert New Account**.

   This Business Service is defined to receive from the EAI XML Converter Business Service the instance of Account data in hierarchical format.

   The Business Service applies the Account information into Siebel using the Insert or Update method.

A.2.6 Creating a Siebel Workflow for a Service Using HTTP Transport

The following procedure is an example of a Siebel workflow illustrated in the Siebel Workflow Administration window. The Workflow was designed for importing Siebel Account record information through the HTTP transport.
The following procedure describes how to create a Siebel Workflow that generates Siebel XML when an Account record is updated in the Siebel Call Center application and then places Siebel XML on the file system.

To create a Siebel Workflow:
1. In the Process Properties tab of the Workflow Process window, define the Account message and Account XML process properties.

   Account message contains the Siebel Account data in hierarchical format.

   Account XML specifies the Siebel Account data that the workflow converted to XML.

2. Define an EAI XML Converter Business Service step and call it XML to Property Set.

   The Business Service is defined to receive the Account data from the EAI HTTP Transport Business Service in XML format and convert it to hierarchical format.
3. Define an EAI Siebel Adapter Business Service step and call it Update Siebel. The Business Service is defined to receive from the EAI XML Converter Business Service the instance of Account data in hierarchical format.

The Business Service applies the Account information into Siebel using the Insert or Update method.
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.
A
access rights, 7-1
adapter configuration
   updating, 3-2
adapter exceptions, 6-4
Adapter Lib Directory parameter, 2-4 to 2-5
adapter types
   resource, 3-2
adapters, 1-1
   configuring, 3-2
   deploying, 3-1 to 3-2, 4-1
   integrating with BPEL Process Manager, 4-1
   troubleshooting, 6-1 to 6-7
Adapters node, 4-39, 4-40
Add Channel dialog box, 2-34, 4-36
Add Target dialog box, 2-13
Advanced tab, 2-14, 2-38, 2-39
ae batch script, 6-2
application adapters
   configuring, 3-2
   deploying, 3-1 to 3-2, 4-1
   integrating with BPEL Process Manager, 4-1
   troubleshooting, 6-1 to 6-7
Application Explorer, 1-2, 1-5, 2-17 to 2-18, 4-6
debugging and, 6-2 to 6-3
J2CA configuration and, 3-2
OracleAS Adapter J2CA and, 6-5
security and, 7-1, 7-6
targets and, 4-40
testing and, 6-2 to 6-3
troubleshooting, 6-2 to 6-3
WSDL files and, 4-36
application system objects
   viewing, 2-17
Available list, 7-4, 7-5

B
BPEL Designer, 4-1
BPEL Process Manager
   integrating with adapters, 4-1
   OracleAS Adapter for Siebel and, 4-1
browsing metadata, 2-16
BSE (OracleAS Adapter Business Services Engine)
   configuring, 2-5, 2-9
   connection access to, 7-6
troubleshooting, 6-5
BSE configuration page, 2-4 to 2-5
BSE control service URL, 7-8
BSE repositories
   migrating, 7-8
BSE settings window, 2-5
BSE system settings, 2-5 to 2-7
BSE URL field, 2-10
Business Components, 1-2 to 1-5, 2-19, 2-29
business events, 1-1
Business Integration Manager, 1-5
Business Objects, 1-4, 2-29
   browsing, 2-16
Business Service list, 2-24
business services, 1-2 to 1-5, 2-19, A-1 to A-3
   browsing, 2-16
   creating, 2-29 to 2-30
   deploying, 7-1
testing, 2-30 to 2-31
Business Services node, 7-2 to 7-5, 7-7

C
CCI (Common Client Interface), 3-1
CCI calls, 3-1
channel configuration parameters
   Document type XML, 2-37
   Error Directory, 2-38, 2-39
   File Mask, 2-39
   Message wait interval, 2-38
   Mode of operation, 2-38
   MQ server channel, 2-37
   MQ server host, 2-37
   MQ server port, 2-37
   Poll interval, 2-39
   Polling Location, 2-39
   Port, 2-35, 4-38
   Processing Mode, 2-39
   Queue manager name, 2-37
   Request queue name, 2-38
   Response/Ack Directory, 2-39
   Server port, 2-35, 4-38
   Synchronization Type, 2-35, 2-38, 2-39, 4-38
   Thread limit, 2-38, 2-39
Channel field, 4-41
channel types
  File, 2-38
  HTTP, 2-34, 4-36 to 4-38
  MQSeries, 2-36

channels
  creating, 1-1, 2-33 to 2-40, 4-36 to 4-38
  deleting, 2-40
  editing, 2-40
  starting, 2-36 to 2-40, 4-39
  stopping, 2-36 to 2-40, 4-39
  testing and debugging, 4-39

Channels node, 2-34, 4-36

channels. See also listeners

closing connections, 2-16

COM environment, 6-2

Common Client Interface (CCI), 3-1

Configuration node under Business Services, 7-2 to 7-5, 7-7

configuration parameters, 3-4
  IWayConfig, 3-4
  IWayHome, 3-4
  IWayRepoPassword, 3-4
  IWayRepoURL, 3-4
  IWayRepoUser, 3-4
  Loglevel, 3-4

configurations
  connecting to, 2-12
  defining, 2-10 to 2-11
  overwriting, 3-4

Configurations node, 2-10, 7-2

configuring adapters, 3-2

configuring BSE system settings, 2-5 to 2-7

configuring events, 2-33 to 2-40

configuring repositories, 2-7

connecting to OracleAS Adapter J2CA, 6-5

connecting to Siebel, 2-12 to 2-16, 4-39, 4-40, 6-2 to 6-4, 6-6

Connection dialog box, 4-39

connection parameters, 6-2
  Language, 2-14
  Object Manager, 2-14
  Port, 2-3 to 2-5
  Repository Manager, 2-14

connection types, 6-2

connections
  closing, 2-16
  deleting, 2-16
  establishing, 2-12 to 2-16, 2-25, 4-39

Connector
  deploying to Oracle Application Server, 3-1

connector factories, 3-2

connector factory objects, 3-2

multiple, 3-4

control methods, 7-9

Create Web Service dialog box, 2-29

creating channels, 4-36 to 4-38

creating events, 4-40 to 4-41

creating repository projects, 2-9 to 2-11

creating schemas, 2-18 to 2-20

creating Web services, 2-29 to 2-30

D

database connections
  opening, 3-4

databases
  connecting to, 3-4
  Oracle, 3-4

Debug Level parameter, 2-6

defining targets to Siebel, 2-13
deleting channels, 2-40
deleting targets, 2-16
deploying adapters, 3-1 to 3-2, 4-1

description field, 2-29, 7-3 to 7-4, 7-5, 7-7
design time, 2-9, 4-1, 4-7, 4-35, 7-8
customizing, 4-6
disconnecting from Siebel, 2-16

DNS Lookup option, 7-7

DNS name, 7-7

Document type XML parameter, 2-37
domain names, 7-7
domain option, 7-7

E

EAI (Enterprise Application Integration), 1-5

Eclipse. See JDeveloper

editing channels, 2-40

editing targets, 2-16

EJB (Enterprise Java Beans), 3-1

Encoding parameter, 2-6

Enterprise Application Integration (EAI), 1-5

Enterprise Connector for J2EE Connector Architecture (J2CA), 1-1

Enterprise Java Beans (EJB), 3-1

Ensemble type list, 2-24

Error Directory parameter, 2-38, 2-39
target systems and error messages, 6-2 to 6-7
target systems and error messages, 6-2 to 6-7
target systems and error messages, 6-2 to 6-7
target systems and error messages, 6-2 to 6-7
target systems and error messages, 6-2 to 6-7
target systems and error messages, 6-2 to 6-7
target systems and error messages, 6-2 to 6-7
target systems and error messages, 6-2 to 6-7
target systems and error messages, 6-2 to 6-7
target systems and error messages, 6-2 to 6-7
target systems and error messages, 6-2 to 6-7
target systems and error messages, 6-2 to 6-7
target systems and error messages, 6-2 to 6-7

eScript script, A-1

event adapters, 2-33 to 2-40
event integration, 4-35
event messages, 4-36
event ports
  creating, 4-40 to 4-41
event schemas, 2-22
events, 1-1
  configuring, 2-33 to 2-40
  creating, 4-40 to 4-41
  triggering, 4-55 to 4-59
  workflows and

Events node, 2-34

Execution Denied list, 7-6

Execution Granted list, 7-6

Existing Service Names list, 2-29

Export WSDL dialog box, 4-7

F

fault code elements, 6-6

fault string elements, 6-6
File channel, 2-38
File Mask parameter, 2-39
file system repositories
configuring, 2-7
File transport, A-7 to A-11, A-17

G
Generate XML Schema wizard, 2-24
Grant Access check box, 7-7
Group (of Computers) option, 7-7
  creating, 7-3
Groups node, 7-4

H
Hostname parameter, 2-5, 2-10
HTTP channel, 2-34, 4-36 to 4-38
HTTP transport, A-11, A-20

I
inbound J2CA services, 4-36
inbound processing, 1-1 to 1-5, A-2
input arguments, 4-58
Input Arguments tab, 4-58
installation directories, 3-4
instances of policy types, 7-1
integration access methods
  OracleAS Adapter for Siebel and, 1-5
Integration Object (IO) node, 2-25
  creating, 4-39
Integration Objects, 1-2, 2-22
  browsing, 2-16
IO (Integration Object) node, 2-25
  creating, 4-39
IP (Mask)/Domain field, 7-7
IP addresses, 7-7
IP and Domain Restriction policy type, 7-6
IWayConfig parameter, 3-4
IWayHome parameter, 3-4
IWayRepoPassword parameter, 3-4
IWayRepoURL parameter, 3-4
IWayRepoUser parameter, 3-4

J
J2CA (Enterprise Connector for J2EE Connector
  Architecture), 1-1
J2CA architecture
  Oracle Application Server Adapter and, 3-1
J2CA configuration
  Application Explorer and, 3-2
J2CA repositories
  migrating, 7-10
J2CA resource adapters, 1-1, 3-1
J2CA services, 4-36
JAR files, 6-2
Java Data Bean, 1-5
Java program clients, 3-1
JDeveloper, 4-6

L
Language parameter, 2-6, 2-14, 6-3
License and Method dialog box, 2-29
License field, 2-29
licenses, 2-29
list of nodes, 6-2
listeners. See also channels
  log files, 6-1
  log levels
    overwriting, 3-4
  Loglevel parameter, 3-4
  logon parameters, 6-2

M
managed connector factories, 3-2
managed connector factory objects, 3-2
  multiple, 3-4
ManagedConnectionFactory parameter, 3-4
Mediator Inbound Process, 5-29
Mediator Outbound Process, 5-6
message interactions
  asynchronous, 1-2
  synchronous, 1-2
message types
  event, 4-36
Message wait interval parameter, 2-38
messages, 1-1
metadata
  browsing, 2-16
  storing, 2-6
  viewing, 2-17
  metadata tables, 2-18
Method Name field, 2-29
methods, 7-1
Methods node, 2-30
migrating repositories, 7-8
Mode of operation parameter, 2-38
MQ server channel parameter, 2-37
MQ server host parameter, 2-37
MQ server port parameter, 2-37
MQSeries channel, 2-36
MQSeries transport, A-3, A-13

N
Name field, 7-3 to 7-4, 7-5
navigation paths, 1-4
New Configuration dialog box, 2-10 to 2-11
New Group dialog box, 7-3
New Policy permissions dialog box, 7-5, 7-6
New User dialog box, 7-2
Node list, 6-2
Node name field, 4-39
nodes
  Adapters, 4-39, 4-40
  Business Services, 7-2, 7-7
  Channels, 2-34, 4-36
Configuration under Business Services, 7-2 to 7-5, 7-7
Configurations, 2-10, 7-2
connected, 2-16, 4-39
disconnected, 2-16
Events, 2-34
Groups, 7-4
Integration Object, 2-25, 4-39
Methods, 2-30
Policies, 7-5
Ports, 4-36
Sample Account, 2-25, 4-39
Security, 7-2 to 7-5, 7-7
Services, 2-30
Siebel, 4-39 to 4-40
Users and Groups, 7-2 to 7-3
Number of Async. Processors parameter, 2-6

O
Object Manager, 1-2
Object Manager parameter, 2-14
Oracle Application Server
deployment of Connector to, 3-1
Oracle Application Server Adapter
J2CA architecture and, 3-1
troubleshooting, 6-7
Oracle databases, 3-4
Oracle JDeveloper, 4-6
Oracle repositories
migrating, 7-8
Oracle's Unified Method (OUM), viii
OracleAS Adapter
installation directory and, 3-4
OracleAS Adapter Application Explorer. See Application Explorer
OracleAS Adapter Business Services Engine (BSE), 1-1, 2-9
configuring, 2-5
troubleshooting, 6-5
OracleAS Adapter for Siebel
BPEL Process Manager and, 4-1
deploying, 1-1, 4-1
integrating with Siebel, 4-7
integration access methods and, 1-5
troubleshooting, 6-1 to 6-7
OracleAS Adapter J2CA, 2-9 to 2-10
Application Explorer and, 6-5
connecting to, 6-5
outbound processing, 1-1 to 1-5, A-2
Password parameter, 2-4, 6-2, 7-3
passwords, 2-4, 3-4
permissions, 7-1
denying, 7-6
granting, 7-6
policies, 7-1
applying, 7-1
creating, 7-4
Policies node, 7-5
policy types, 7-1
IP and Domain Restriction, 7-6
policy-based security, 7-1 to 7-7
Poll interval parameter, 2-39
Polling Location parameter, 2-39
Port Number parameter, 2-10
Port parameter, 2-3 to 2-5, 2-35, 4-38
ports
creating, 2-33, 4-40 to 4-41
Ports node, 4-36
privileges, 7-1
setting, 7-1
Process Designer tab, 4-58
Process Manager. See BPEL Process Manager
Process Simulator tab, 4-59
Processing Mode parameter, 2-39
projects
repository, 2-9 to 2-11
properties, 3-4
Protocol list, 2-26, 4-37, 4-39

Q
Query method, A-9
Queue manager name parameter, 2-37

R
repositories
configuring, 2-7
migrating, 7-8
Repository Driver parameter, 2-6
repository information
storing, 2-7
Repository Manager parameter, 2-14
repository migration parameters, 7-9
Repository Name parameter, 2-9 to 2-11
repository parameters
Driver, 2-6
Password, 2-6
Pooling, 2-6
Type, 2-6
URL, 2-6
User, 2-6
Repository Password parameter, 2-6
Repository Pooling parameter, 2-6
repository projects
creating, 2-9 to 2-11
Web services and, 2-9
Repository Type parameter, 2-6
Repository URL parameter, 2-5, 2-6
Repository URLs, 7-10
Repository User parameter, 2-6
repository.xml file, 7-11
REQUEST option, 2-38
Request queue name parameter, 2-38
request schemas, 2-19
Request tab, 2-37, 2-39
REQUEST_ACK option, 2-38
REQUEST_RESPONSE option, 2-38
requests executing, 6-4
Resource Execution policy type, 7-1
response schemas, 2-19
Response/Ack Directory parameter, 2-39
runtime, 2-9, 4-7, 4-39 to 4-41
runtime events (Siebel Events), A-1
runtime integration, 4-55

Sample Account node, 2-25, 4-39
schema considerations, 2-18
Schema location field, 4-39
schemas creating, 2-18 to 2-20, 2-22
scripts, A-1
security, 7-1 to 7-7
Application Explorer and, 7-1, 7-6
configuring, 7-2
Security node, 7-2 to 7-5, 7-7
security policies applying, 7-1
creating, 7-4
Security Policy option, 7-6
Selected list, 7-4, 7-5
Server port parameter, 2-35, 4-38
service integration, 4-7
service names, 2-29
Service Provider list, 2-10 to 2-11
service schemas creating, 2-18 to 2-20
Service-Oriented Architecture (SOA), 4-1
services, 1-1
creating, 2-29 to 2-30
deploying, 7-1
testing, 2-30 to 2-31
workflows and, A-13 to A-23
Services node, 2-30
servlets, 3-1
Siebel connecting to, 1-4, 2-12 to 2-16, 4-39, 6-2 to 6-4, 6-6
disconnecting from, 2-16
Siebel Business Components, 1-2 to 1-5, 2-19, 2-29
Siebel Business Objects, 1-4, 2-29
browsing, 2-16
Siebel business processes invoking, 1-4
Siebel Business Services, 1-2 to 1-5, 2-19, A-1 to A-3
browsing, 2-16
Siebel Call Center starting, 4-55
Siebel Client Workflow Administration screens, A-1
Siebel COM Data Interface, 1-5
Siebel Events (run-time events), A-1
Siebel Gateway server, 6-2
Siebel Integration Objects, 1-2, 2-22
browsing, 2-16
Siebel Java Data Bean, 1-5
Siebel node, 4-39 to 4-40
Siebel Object Manager, 1-2
Siebel Tools Schema Wizard, 2-22, 2-25
Siebel Tools window, 2-23
Siebel transports, 1-5
Siebel VB script, A-1
Siebel Workflow Administration window, 4-55
Siebel workflow policies, A-1 to A-2
Siebel Workflows, 1-5, A-1 to A-23
creating, A-3
SiebelJl_Common.jar file, 6-2
SiebelJl_enu.jar file, 6-2
Single (Computer) option, 7-7
SOA (Service-Oriented Architecture), 4-1
SOAP agents, 6-5 to 6-6
SOAP envelopes, 7-9
SOAP faults, 6-5 to 6-6
soap operation name dialog box, 7-8
SOAP requests, 6-5 to 6-7, 7-1, 7-5
creating, 7-8
errors and, 6-5 to 6-7
SOAP responses, 6-5 to 6-7
starting the Siebel Call Center, 4-55
Synchronization Type parameter, 2-35, 2-38 to 2-39, 4-38
system objects viewing, 2-17
system parameters
Adapter Lib Directory, 2-6
Debug Level, 2-6
Encoding, 2-6
Language, 2-6
Number of Async. Processors, 2-6
system settings configuring, 2-5 to 2-7

target systems errors and, 6-6
Target Type list, 2-13
targets connecting to, 2-12 to 2-16, 4-39, 4-40, 6-2 to 6-4
defining, 2-13
deleting, 2-16
disconnecting from, 2-16
deploying, 2-16
testing Web services, 2-30 to 2-31
thin clients, 6-2

S
Sample Account node, 2-25, 4-39
schema considerations, 2-18
Schema location field, 4-39
schemas creating, 2-18 to 2-20, 2-22
scripts, A-1
security, 7-1 to 7-7
Application Explorer and, 7-1, 7-6
configuring, 7-2
Security node, 7-2 to 7-5, 7-7
security policies applying, 7-1
creating, 7-4
Security Policy option, 7-6
Selected list, 7-4, 7-5
Server port parameter, 2-35, 4-38
service integration, 4-7
service names, 2-29
Service Provider list, 2-10 to 2-11
service schemas creating, 2-18 to 2-20
Service-Oriented Architecture (SOA), 4-1
services, 1-1
creating, 2-29 to 2-30
deploying, 7-1
testing, 2-30 to 2-31
workflows and, A-13 to A-23
Services node, 2-30
servlets, 3-1
Siebel connecting to, 1-4, 2-12 to 2-16, 4-39, 6-2 to 6-4, 6-6
disconnecting from, 2-16
Siebel Business Components, 1-2 to 1-5, 2-19, 2-29
Siebel Business Objects, 1-4, 2-29
browsing, 2-16
Siebel business processes invoking, 1-4
Siebel Business Services, 1-2 to 1-5, 2-19, A-1 to A-3
browsing, 2-16
Siebel Call Center starting, 4-55
Siebel Client Workflow Administration screens, A-1
Siebel COM Data Interface, 1-5
Siebel Events (run-time events), A-1
Siebel Gateway server, 6-2
Siebel Integration Objects, 1-2, 2-22
browsing, 2-16
Siebel Java Data Bean, 1-5
Siebel node, 4-39 to 4-40
Siebel Object Manager, 1-2
Siebel Tools Schema Wizard, 2-22, 2-25
Siebel Tools window, 2-23
Siebel transports, 1-5
Siebel VB script, A-1
Siebel Workflow Administration window, 4-55
Siebel workflow policies, A-1 to A-2
Siebel Workflows, 1-5, A-1 to A-23
creating, A-3
SiebelJl_Common.jar file, 6-2
SiebelJl_enu.jar file, 6-2
Single (Computer) option, 7-7
SOA (Service-Oriented Architecture), 4-1
SOAP agents, 6-5 to 6-6
SOAP envelopes, 7-9
SOAP faults, 6-5 to 6-6
soap operation name dialog box, 7-8
SOAP requests, 6-5 to 6-7, 7-1, 7-5
creating, 7-8
errors and, 6-5 to 6-7
SOAP responses, 6-5 to 6-7
starting the Siebel Call Center, 4-55
Synchronization Type parameter, 2-35, 2-38 to 2-39, 4-38
system objects
viewing, 2-17
system parameters
Adapter Lib Directory, 2-6
Debug Level, 2-6
Encoding, 2-6
Language, 2-6
Number of Async. Processors, 2-6
system settings configuring, 2-5 to 2-7

T
target systems
errors and, 6-6
Target Type list, 2-13
targets
connecting to, 2-12 to 2-16, 4-39, 4-40, 6-2 to 6-4
defining, 2-13
deleting, 2-16
disconnecting from, 2-16
deploying, 2-16
testing Web services, 2-30 to 2-31
thin clients, 6-2
Thread limit parameter, 2-38, 2-39
transaction processing, 6-1
transactions
  storing, 2-6
transports, 1-5
  File, A-7 to A-11, A-17
  HTTP, A-11, A-20
  MQSeries, A-3, A-13
triggering an event, 4-55 to 4-59
troubleshooting, 6-1 to 6-7
  Application Explorer, 6-2 to 6-3
  BSE, 6-5
  Web services, 6-5 to 6-7
Type list, 7-5, 7-7

U
  updating adapter configuration, 3-2
User parameter, 2-4
UserName parameter, 6-2
users
  associating, 7-2
Users and Groups node, 7-2 to 7-3
Users node, 7-3

V
  viewing system objects, 2-17

W
  Web Service Definition Language
    (WSDL), 2-32 to 2-33
  Web service names, 2-29
  Web services, 1-1 to 1-2, 1-5, 2-9
    creating, 2-29 to 2-30
    delivering, 2-6
    deploying, 7-1
    integrating, 4-1
    repository projects and, 2-9
    testing, 2-30 to 2-31
    troubleshooting, 6-5 to 6-7
  Web services policy-based security, 7-1 to 7-7
  workflow elements, 4-58
  workflow policies, A-1 to A-2
  Workflow Process window, A-4 to A-18, A-22
  Workflow Processes window, 4-56
  Workflows, A-1 to A-23
    creating, A-3
  WSDL (Web Service Definition Language), 2-32 to 2-33
  WSDL documents, 4-1
  WSDL file location dialog box, 7-8
  WSDL File Name field, 4-41
  WSDL files, 4-1
  Application Explorer and, 4-36
    creating, 4-36
X
  XDR schemas
    creating, 2-22
  XML documents, 1-4, A-2
  XML messages, 1-1, A-2 to A-3
  XML schemas, 1-2, 1-5
    creating, 2-18 to 2-20, 2-22