Oracle® Application Server

Adapter for Siebel User's Guide

10*g* Release 3 (10.1.3.1.0)

B28999-01

March 2007

ORACLE®

Oracle Application Server Adapter for Siebel User's Guide, 10*g* Release 3 (10.1.3.1.0)

B28999-01

Primary Author: Stefan Kostial

Contributing Authors: Sheela Vasudevan, Sunil Gopal, Marian Jones, Vikas Anand, Sunil Wadhwa, Vishal Saxena

Contents

8 Advanced User Tools

A Using Siebel Workflows

Glossary

Index

Preface

This Preface contains the following topics:

Audience

Oracle Application Server Adapter for Siebel User's Guide is intended for those who perform the following tasks:

- Install applications
- Maintain applications

To use this document, you need to know how to install and configure Oracle BPEL Process Manager.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. 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.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

Related Documents

For more information, refer to these Oracle resources:

- *Oracle Application Server Adapter Concepts*
- *Oracle Application Server Adapters Installation Guide*

Printed documentation is available for sale in the Oracle Store at

`http://oraclestore.oracle.com/OA_HTML/ibeCCtdMinisites.jsp?language=US`

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

`http://www.oracle.com/technology/membership/index.html`

If you already have a user name and password for OTN, then you can go directly to the documentation section of the OTN Web site at

`http://www.oracle.com/technology/documentation/index.html`

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- Conventions in Text
- Conventions in Code Examples
- Conventions for Windows Operating Systems

Conventions in Text

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

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an **index-organized table**.
Italics	Italic typeface indicates book titles or emphasis.	*Oracle Database Concepts* Ensure that the recovery catalog and target database do *not* reside on the same disk.

Convention	Meaning	Example
UPPERCASE monospace (fixed-width) font	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, Recovery Manager keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, user names, and roles.	You can specify this clause only for a NUMBER column. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. Use the DBMS_STATS.GENERATE_STATS procedure.
lowercase monospace (fixed-width) font	Lowercase monospace typeface indicates executable programs, filenames, directory names, and sample user-supplied elements. *Note:* Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter sqlplus to start SQL*Plus. The password is specified in the orapwd file. Back up the datafiles and control files in the /disk1/oracle/dbs directory. The department_id, department_name, and location_id columns are in the hr.departments table. Connect as oe user. The JRepUtil class implements these methods.
lowercase italic monospace (fixed-width) font	Lowercase italic monospace font represents placeholders or variables.	You can specify the *parallel_clause*. Run *old_release*.SQL where *old_release* refers to the release you installed prior to upgrading.

Conventions in Code Examples

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

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

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

Convention	Meaning	Example
[ ]	Anything enclosed in brackets is optional.	DECIMAL (*digits* [, *precision*])
{ }	Braces are used for grouping items.	{ENABLE | DISABLE}
|	A vertical bar represents a choice of two options.	{ENABLE | DISABLE} [COMPRESS | NOCOMPRESS]
...	Ellipsis points mean repetition in syntax descriptions.	CREATE TABLE ... AS *subquery*;
	In addition, ellipsis points can mean an omission in code examples or text.	SELECT *col1*, *col2*, ... , *coln* FROM employees;
Other symbols	You must use symbols other than brackets ([ ]), braces ({ }), vertical bars (|), and ellipsis points (...) exactly as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
Italics	Italicized text indicates placeholders or variables for which you must provide particular values.	CONNECT SYSTEM/*system_password* DB_NAME = *database_name*

Convention	Meaning	Example

`UPPERCASE`	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. Because these terms are not case sensitive, you can use them in either UPPERCASE or lowercase.	`SELECT last_name, employee_id FROM employees;`
		`SELECT * FROM USER_TABLES;`
		`DROP TABLE hr.employees;`

`lowercase`	Lowercase typeface indicates user-defined programmatic elements, such as names of tables, columns, or files.
	Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.
		`SELECT last_name, employee_id FROM employees;`
		`sqlplus hr/hr`
		`CREATE USER mjones IDENTIFIED BY ty3MU9;`

Conventions for Windows Operating Systems

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

Convention	Meaning	Example
Click **Start**, and then choose the ***menu item***	How to start a program.	To start the Database Configuration Assistant, click **Start**, and choose **Programs**. In the Programs menu, choose **Oracle** - ***HOME_NAME*** and then click **Configuration and Migration Tools**. Choose **Database Configuration Assistant**.
File and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (|), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the filename begins with \\, then Windows assumes it uses the Universal Naming Convention.	c:\winnt"\"system32 is the same as C:\WINNT\SYSTEM32
`C:\>`	Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the *command prompt* in this manual.	`C:\oracle\oradata>`
Special characters	The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.	`C:\>exp HR/HR TABLES=employees QUERY=\"WHERE job_id='SA_REP' and salary<8000\"`

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

Help Us to Serve You Better

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

The following list includes the specifications our consultants require.

- **Platform**:
- **Operating System**:
- **Operating System Version**:
- **Product List**:
- **Adapters**:
- **Adapter Deployment**:

  For example, *J2CA* or *Business Services Engine (BSE)*
- **Container Version**:

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

Component	Version
Adapter	
EIS (DBMS/APP)	
HOTFIX/Service Pack	

In the following table, specify the JVM version and vendor.

JVM Version	Vendor

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

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

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

- XML schema
- XML instances
- Other input documents (transformation)
- Error screen shots
- Error output files
- Trace and log files
- Log transaction

1 Introduction

Oracle Application Server connects to a Siebel system through Oracle Application Server Adapter for Siebel (OracleAS Adapter for Siebel). OracleAS Adapter for Siebel provides connectivity and enables interactions on a Siebel system.

This chapter discusses the following topics:

Adapter Features

Oracle Application Server 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.

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

OracleAS 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: Enables applications to initiate a Siebel business event.
- Events: Enables applications to access Siebel data only when a Siebel business event occurs.

To support event functionality, the following two features are implemented:

- Port: A **port** associates a particular business object exposed by an adapter with a particular disposition. A disposition defines the protocol and location of the event data. The port defines the end point of the event consumption. The port is the adapter component that pushes the event received from the enterprise information system (EIS) to the adapter client.

Note: You are not required to create or configure ports for use with BPEL Process Manager. However, in this release you can associate an event schema to a port under a J2CA configuration.

The port validation feature is currently not available.

- Channel: A **channel** represents configured connections to particular instances of back-end or other types of systems. A channel binds one or more event ports to a particular **listener** managed by an adapter.

  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. This enables the user to choose the optimal channel component based on deployment requirements.

OracleAS Adapter for Siebel:

- Supports synchronous and asynchronous, bidirectional message interactions for Siebel Business Services, Business Components, and Integration Objects.
- Includes Oracle Application 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 OracleAS Adapter J2CA.
- Web services for BSE.

OracleAS 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. OracleAS Adapter for Siebel exposes and generates metadata and interacts with these industry-specific objects.

See Also: *Oracle Application Server Adapter Concepts*

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 1-3. Failure to add the encoding option as described in this section will result in an error and you will not be 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, do the following:

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

   ```
   -Dfile.encoding="ISO8859_1"
   ```

2. Log in to the **Oracle Application Server** console, go to the **OC4JContainer** home page, click **Administration**, then **Server Properties**, and under **Command Line Options**, edit the **Java Options** field to include the following:

   ```
   -Dfile.encoding="ISO8859_1"
   ```

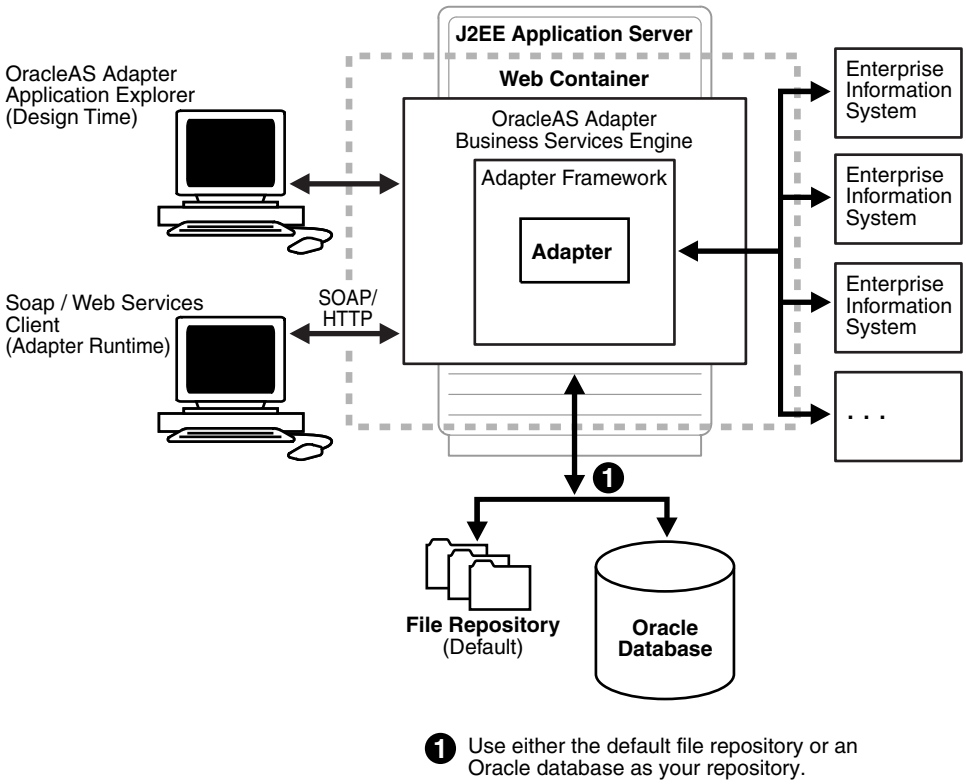
Oracle Application Server Adapter Business Services Engine (BSE) Architecture

Figure 1–1 shows the generic architecture for the Oracle Web service adapter for packaged applications. The adapter works in conjunction 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, configure services, and configure listeners to listen for EIS events. Metadata created while you perform these operations are stored in the repository by BSE.

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

Figure 1–1 Oracle Application Server Adapter Business Services Engine (BSE) Generic Architecture



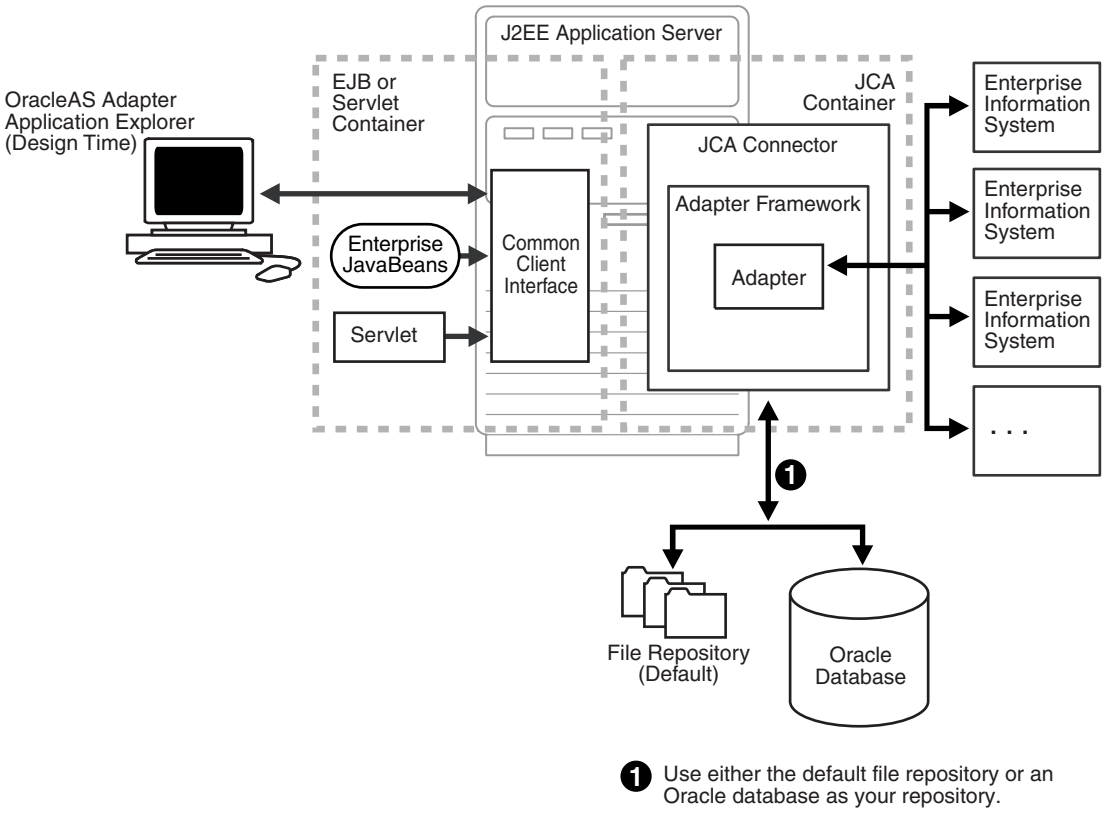
Note: Do not use a file repository for BSE in production environments.

Oracle Application Server Adapter J2CA Generic Architecture

Figure 1–2 shows the generic architecture for the OracleAS Adapter J2CA for packaged applications. The OracleAS 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 in conjunction with the connector, is used to configure adapter connections, browse EIS objects, configure services, and configure listeners to listen for EIS events. Metadata created while you perform these operations are stored in the repository by 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 OC4J deployment descriptor `oc4j-ra.xml`. See Chapter 3, "OC4J Deployment and Integration" for more information on OC4J deployment.

Figure 1–2 Oracle Application Server Adapter J2CA Generic Architecture



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.

Integration with Siebel

You can use OracleAS 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. OracleAS 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, OracleAS 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 1–1 Siebel Objects and Processes

Siebel Objects	API or Transport	Process
Business Services	Java Data Bean (Siebel Version 6.3-7.8)	N/A
	Com Data Interface (Siebel Version 6.01-6.2)	
Business Components	Java Data Bean (Siebel Version 6.3-7.8)	N/A
	Com Data Interface (Siebel Version 6.01-6.2)	
Integration Objects	File	Event, Service
	HTTP	Event, Service
	MQSeries	Event, Service
	MQ Read	Service

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. OracleAS 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.

OracleAS 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.

Using Application Explorer with OracleAS 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 OracleAS 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 in conjunction with OracleAS 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*
- *Oracle Application Server Adapters Installation Guide*

BSE Versus OracleAS Adapter J2CA Deployment

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

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

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

1. BSE is the preferred deployment option because it:
   - Can be deployed in a separate instance of the Oracle Application Server.
   - Provides better distribution of load.
   - Provides better isolation from any errors from third party libraries.
   - Provides better capability to isolate issues for debugging purposes.
   - Conforms more closely to the Service Oriented Architecture (SOA) model for building applications.

2. OracleAS Adapter J2CA provides slightly better performance.

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

3. OracleAS Adapter J2CA and the BSE option both provide identity propagation at runtime.

   The BSE option provides the capability to pass identity using the SOAP header. For the OracleAS Adapter J2CA, user name and password can be passed using the connection specification of the CCI.

2

Configuring Oracle Application Server Adapter for Siebel

This chapter describes how to configure OracleAS Adapter for Siebel and create schemas for Siebel Business Objects.

This chapter discusses the following topics:

Encoding Support on UNIX Platforms

When using OracleAS Adapter for Siebel on UNIX environments, you must manually edit the startup script for your server to add a JVM option specifying the file encoding:

```
java $REMDBG -cp $CLASSPATH -DIWAY55=$IWAY55 -Dfile.encoding=ISO8859_1
edaqm -config $SCRIPT $2 $3 $4 $5 $6
```

Starting Application Explorer

Prerequisites

Before starting OracleAS Adapter Application Explorer (Application Explorer) and using Oracle Application Server Adapter for Siebel (OracleAS Adapter for Siebel), you must create `\endorsed` directories under your *`OracleAS_home`* directory and place a copy of the `xalan.jar` file in those directories. Otherwise, you will receive a transformation error when adding an IO node under an Integration Object in Application Explorer.

1. Navigate to the *OracleAS_home*\j2ee\home\connectors\jca-app-adapter\jca-app-adapter directory and copy the xalan.jar file.
2. Search on your machine for the following directories, then create an \endorsed subdirectory under each of these directories:
   - *OracleAS_home*\jdk\jre\lib\
   - *OracleAS_home*\jdk\lib\
   - *OracleAS_home*\jdk\....\lib\
   - *OracleAS_home*\jre\....\lib\
   - Any \jdk\....\lib or \jre\....\lib that is in your PATH environment variable
3. Paste the xalan.jar file you copied in step 1 into each of the new \endorsed subdirectories.

   Important: Please note that you may not have all the directories listed above. This depends on your individual environment.

Launching Application Explorer

To start Application Explorer:

1. Start the server where Application Explorer is deployed.
2. From the Windows **Start** menu, select **Programs**, ***OracleAS_home*** **Adapters**, and then **Application Explorer**.

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

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

   Application Explorer starts. You can now define new targets to your Siebel system.

Configuring Settings for BSE or J2CA

Before a configuration can be created, you must configure OracleAS Adapter Business Services Engine (BSE). You need not configure OracleAS Adapter J2CA because the ra.xml file is configured automatically during installation.

Configuring BSE

After BSE is deployed to Oracle Application Server, you can configure it through the BSE configuration page.

To configure BSE:

1. Display the following page in your browser:

   http://*hostname*:*port*/ibse

   Where hostname is the host name of Oracle Application Server and port is the HTTP port for Oracle Application Server.

   For example,

   http://localhost:7777/ibse

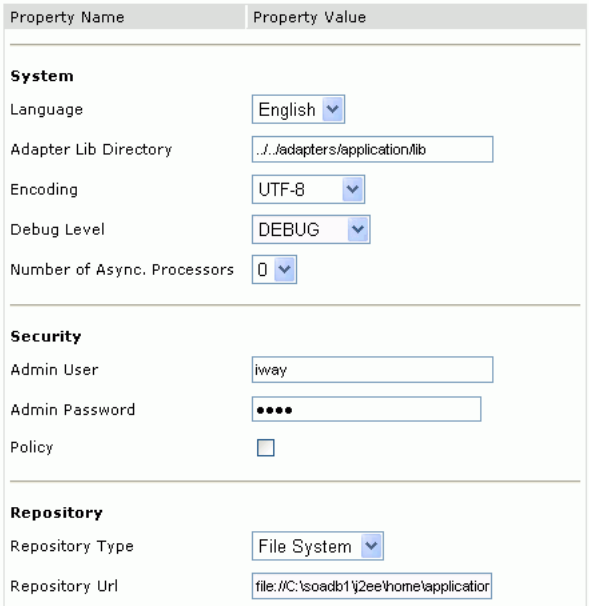
> **Note:** This page might load slowly when accessed for the first time.

2. Log on when prompted.

   When first installed, the user ID and the password are:

   - User name: iway
   - Password: iway

The BSE configuration page is displayed.



3. Ensure that the Adapter Lib Directory parameter specifies the path to the lib directory, for example:

   OracleAS_home\adapters\application\lib

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

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

> **Note:** The Repository URL field specifies where the file system repository is located. To use a database repository, you must enter the repository connection information. For the initial verification, use a file system repository. See "Configuring an Oracle Repository" on page 2-6 for information on switching to a database repository.

5. Click **Save**.

Configuring BSE System Settings

To configure BSE system settings:

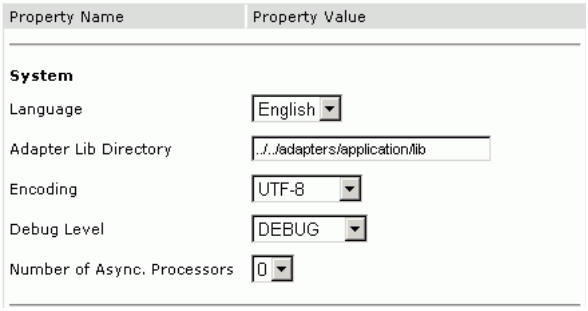
1. Display the BSE configuration page by using the following URL:

   http://*hostname*:*port*/ibse/IBSEConfig

Where `hostname` is the machine where BSE is installed and `port` is the port number on which BSE is listening.

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

The BSE settings window is displayed.



2. Configure the system settings by providing information for the parameters according to the following table.

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

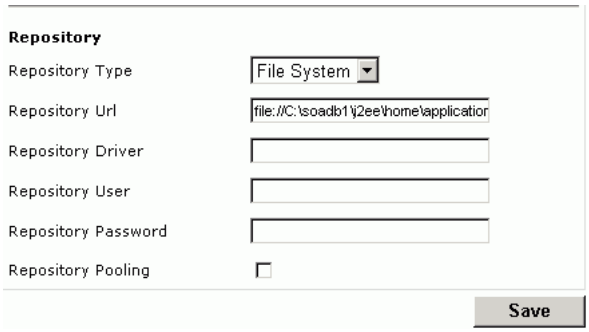
The following image illustrates the Security pane of the window.

3. Configure the security settings by providing information for the parameters according to the following table.

Parameter	Description
Admin User	Provide a BSE administrator ID.

Parameter	Description
Admin Password	Enter the password associated with the BSE administrator ID.
Policy	Select the check box to enable policy security.

The following image shows all of the fields and the check boxes for the Repository pane.



4. Configure the repository settings by providing information for the parameters according to the following table.

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

   See "Configuring a File System Repository" on page 2-5 and "Configuring an Oracle Repository" on page 2-6 for more information.

Parameter	Description
Repository Type	Select one of the following repositories from the list: ■ Oracle ■ File (Do not use a file repository for BSE in production environments.)
Repository URL	Enter the URL to use when opening a connection to the database.
Repository Driver	Provide the driver class to use when opening a connection to the database (optional).
Repository User	Enter the user ID to use when opening a connection to the database.
Repository Password	Enter the password associated with the user ID.
Repository Pooling	Select the check box to enable pooling.

5. Click **Save**.

Configuring a File System Repository

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

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

The default location for the repository on Windows is:

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

On other platforms, use the corresponding location.

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

Configuring an Oracle Repository

To configure an Oracle repository:

1. Contact your database administrator to obtain an Oracle user ID and password to create the BSE repository.

   This user ID should have rights to create and modify tables as well as the ability to create and run stored procedures.

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

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

   For other platforms, see the corresponding location.

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

   ```
   iwse.ora
   ```

   Note: If Oracle is not on the same machine as Oracle Application Server, copy the iwse.ora file to the Oracle machine. Then, from a command prompt on the Oracle machine, navigate to the directory containing the iwse.ora file.

3. Enter the following command:

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

Configuring J2CA

During the J2CA deployment of OracleAS Adapter for Siebel, OC4J generates a deployment descriptor called `oc4j-ra.xml`. This descriptor provides OC4J-specific deployment information for resource adapters. See Chapter 3, "OC4J Deployment and Integration" for more information on J2CA deployment and configuration.

No configuration changes are necessary if you are using the default file based repository with J2CA deployment.

Configuring a Database Repository for J2CA

To configure a database repository for J2CA:

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

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

   ```
   OracleAS_HOME\adapters\application\config\jca_sample
   ```

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

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

4. Open the `oc4j-ra.xml` file in a text editor.
5. Provide the JDBC connection information as a value for the IWAYRepo_URL property.
6. Provide a valid user name for the IWAYRepo_User property.
7. Provide a valid password for the IWAYRepo_Password property.
8. Save your changes to the `oc4j-ra.xml` file.
9. Alter the JDBC driver path in Application Explorer's lcp. For example:

```
lcp=..\lib\orabpel-adapters.jar;C:\jdev\jdbc\lib\ojdbc14.jar;C:\jdev\jdbc\lib\n
ls_charset12.jar;%lcp%
to
lcp=..\lib\orabpel-adapters.jar;..\..\..\jdbc\lib\ojdbc14.jar;..\..\..\jdbc\lib
\nls_charset12.jar;%lcp%
```

Password Encryption

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

Configuring Password Encryption

To encrypt a password:

1. Open Application Explorer.
2. Click **Help** and select **Encryption**.

   The Encryption dialog box opens.
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.

   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 (318931973183297321831293164323332123227)
```

6. In the oc4j-ra.xml file, which is used during run time, replace the existing password with the encrypted value for the IWayRepoPassword element.
7. Restart the Oracle Application Server.

Creating a Repository Configuration

Before you use Application Explorer with OracleAS 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 runtime.

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

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

Creating a Configuration for BSE

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

Defining a New Configuration for BSE

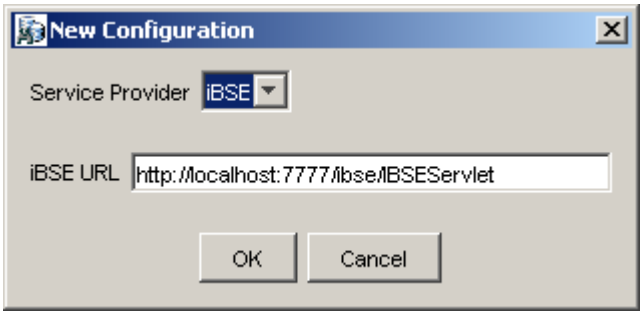
To create a new configuration for BSE:

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

   The New Configuration dialog box is displayed.

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

   The following dialog box is displayed.



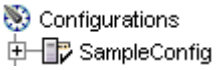
3. From the **Service Provider** list, select **iBSE**.
4. In the **iBSE URL** field, accept the default URL or replace it with a different URL using the following format:

   `http://`*`hostname`*`:`*`port`*`/ibse/IBSEServlet`

   Where `hostname` is the machine where your application server resides and `port` is the port number on which the application server is listening.

5. Click **OK**.

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



The BSE configuration file is stored in *OracleAS_home*\j2ee\oc4j_container\applications\ws-app-adapter\ibse.

Creating a Configuration for J2CA

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

Defining a New Configuration for J2CA

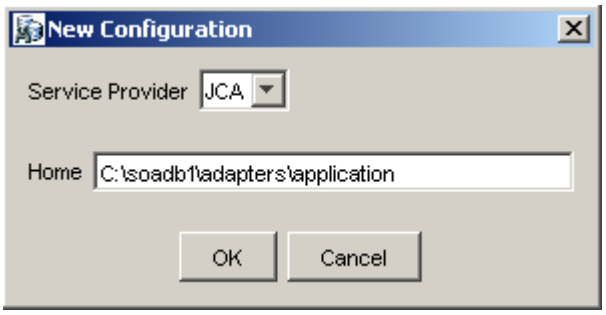
To define a new configuration for J2CA:

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

   The New Configuration dialog box is displayed.

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

   The New Configuration dialog box is displayed.

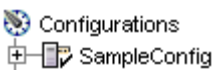


3. From the **Service Provider** list, select **JCA**.
4. In the **Home** field, enter a path to your J2CA configuration directory where the repository, schemas, and other information are stored, for example:

   OracleAS_home\adapters\application

5. Click **OK**.

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



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

Where configuration_name is the name of the configuration you created; for example, SampleConfig.

Connecting to a BSE or J2CA Configuration

To connect to a new configuration:

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

   Nodes appear for Adapters, Events, and Business Services (also known as Web services). The Business Services node is only available for BSE configurations. If you are connected to a J2CA configuration, you will not see the Business Services node. 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 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.

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 2-10. Failure to add the encoding option as described in this section will result in an error and you will not be 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, do the following:

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

   ```
   -Dfile.encoding="ISO8859_1"
   ```

2. Log in to the **Oracle Application Server** console, go to the **OC4JContainer** home page, click **Administration**, then **Server Properties**, and under **Command Line Options**, edit the **Java Options** field to include the following:

```
-Dfile.encoding="ISO8859_1"
```

Defining a Target to Siebel

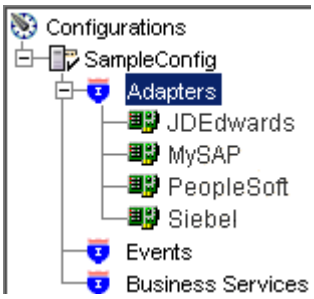
The connection parameters required for defining a Siebel target can be obtained from the `eapps.cfg` file, which is located in the following directory:

`drive:\SiebelRoot\SWEApp\BIN`

Where `Siebelroot` is the Siebel installation directory.

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 box 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 the type of target. For Siebel 6.0, choose **Siebel 6.2 or lower (COM)**. For Siebel 6.3 or higher, choose **Java Bean Data Connection**.

3. Click **OK**.

   When you select **Siebel 6.2 or lower** (COM):

   a. In the **User Agent File** field, enter the name of the configuration file.

   b. In the **Username** field, enter the user name.

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

   d. In the **Repository** field, enter the Siebel Repository where Application Explorer looks for metadata describing Business Services, Business Objects, and Integration Objects.

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

   When you select **6.3 or higher** (JDB):

   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 or 7.8 system.

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

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

f. 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 Manager

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

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.

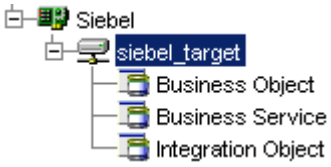
Connecting to a Defined Target

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.

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.

Editing a Target

To edit a target:

1. In the left pane, ensure the target you wish to edit is disconnected.
2. Right-click the disconnected target and select **Edit**.

   The **Edit** pane is displayed on the right.

3. Modify the target information.
4. Click **OK**.

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.

To delete a target:

1. In the left pane, select the target.
2. Right-click the target and select **Delete**.

A confirmation dialog box is displayed.

3. Click **OK** to delete the target you selected.

   The Siebel connection node disappears from the left pane.

Viewing Application System Objects

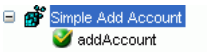
Application Explorer gives you the flexibility to view all Siebel application system objects. One benefit of this flexibility is that you can gain an understanding of the Siebel data structure. You can review parameters, data types, and other attributes of the Siebel data in the right pane.

Viewing Metadata

To view metadata:

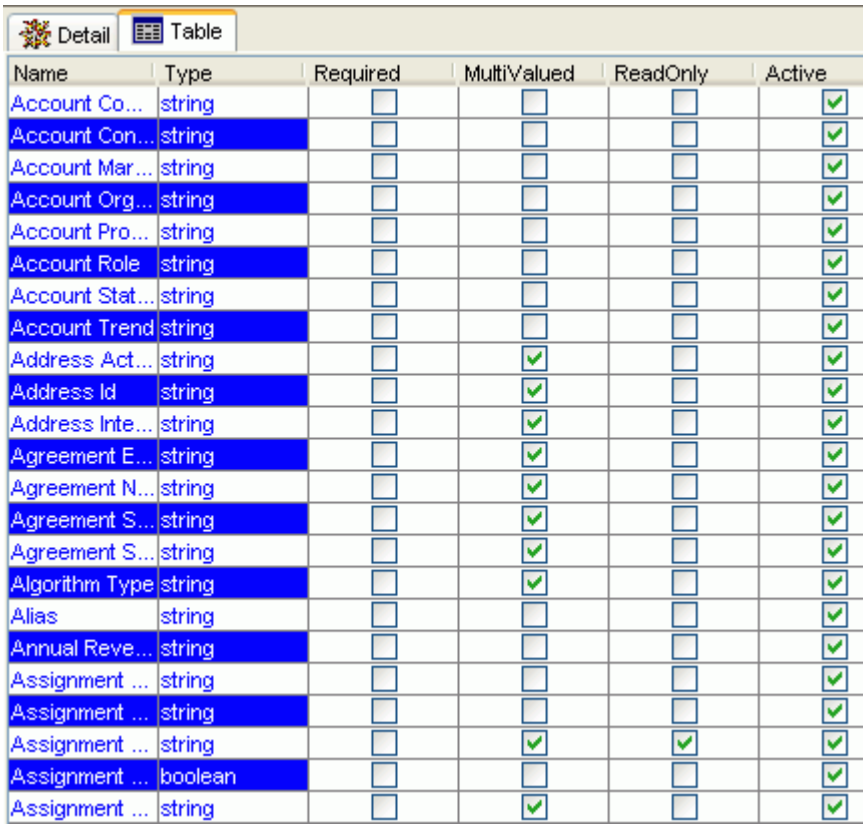
1. If you have not started Application Explorer, start Application Explorer and connect to your Siebel system.
2. In the left pane, expand the **Business Object** or **Business Service** containing the component for which you want to generate schema.
3. Expand the **Business Object** or **Business Service** 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.



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 OracleAS 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 2-24 if you plan to deploy OracleAS Adapter for Siebel in a Web services environment.

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.

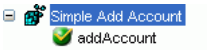
The following image shows the Account Business Object expanded to display all Business Components.

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.

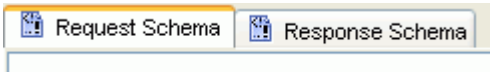
   - For a **Siebel Business Service**, expand the **Business Service** node containing the object for which you want to create schema.



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

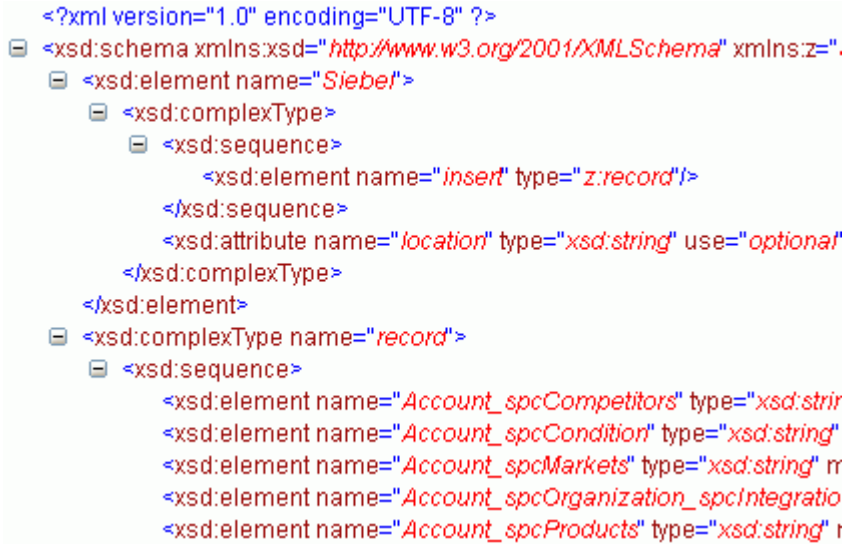
   Application Explorer accesses the Siebel repository and builds schemas.

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.



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 **Integtration 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 will contain 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 will return all fields:

```
<m:Siebel location="S/BO/Account/Account/queryWithView" view="AllView">
```

```
  <m:select>
   <m:Name>Yelena*</m:Name>
  </m:select>
</m:Siebel>
```

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

```
<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
- SubOrganization View

Creating Schemas for Siebel 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 **prior to 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.

OracleAS 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.

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.

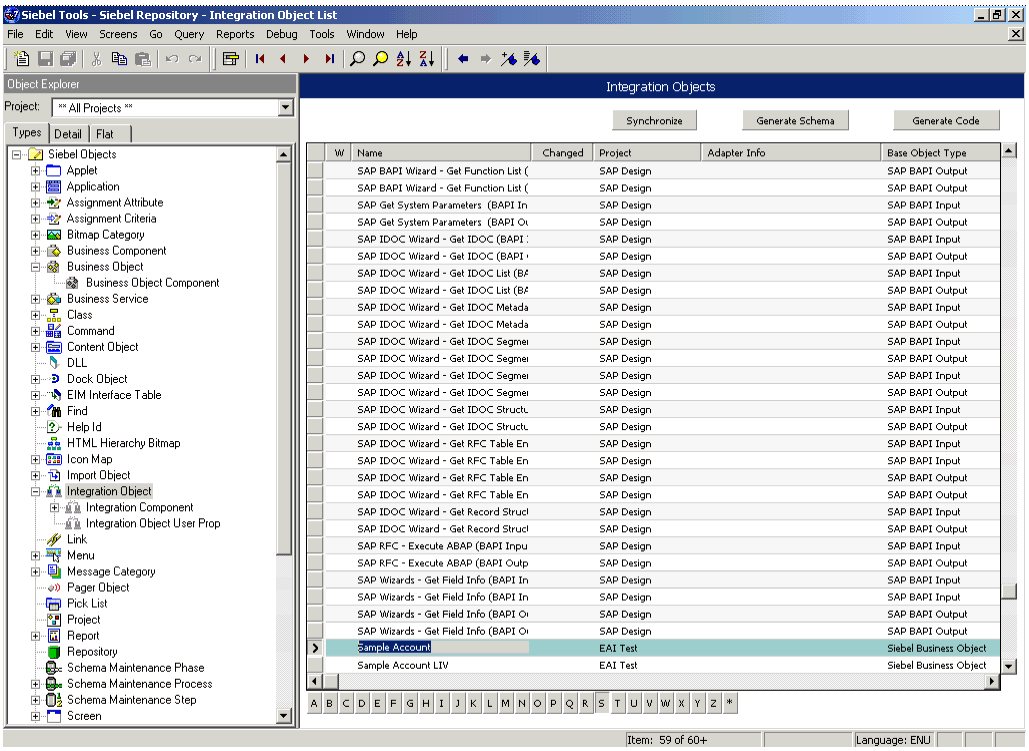
   Perform the following steps:

   a. Enter your user ID and password.

   b. Select a database from the list.

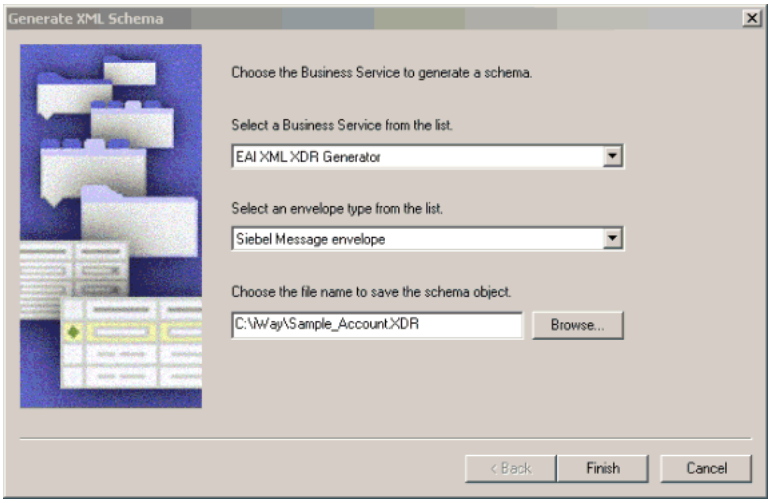
2. Click **OK**.

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.



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 schema file must be saved to a directory on the same computer as Application Explorer.

---

5. Click **Finish**.

   Now you can use Application Explorer to generate XML schemas for the Siebel Integration Object.

Creating a Schema from a Siebel XDR Schema

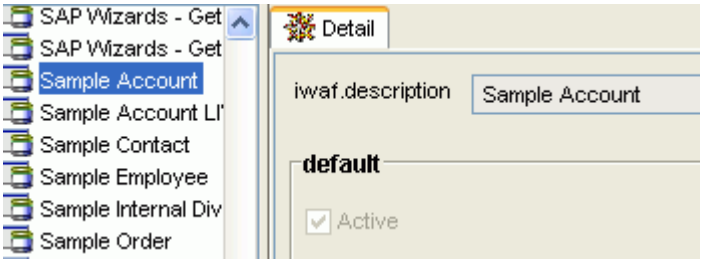
After you create the Siebel XDR schema for a selected Siebel Integration Object, you can create an XML schema using Application Explorer.

You must provide Application Explorer with the location of the previously created Siebel XDR schema for the particular integration object selected.

Note: The XDR file must be on the same computer as Application Explorer or be available through a mapped connection to another drive or machine.

To create a schema from a Siebel XDR 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, Sample Account.
3. To generate event schema, right-click the object and choose **Add event**.

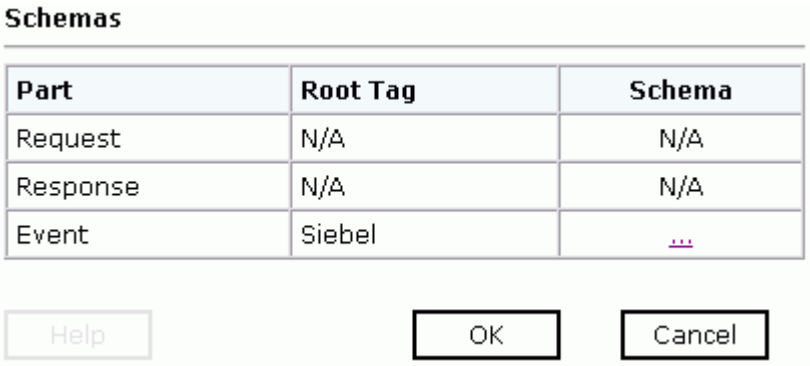
   The Add Event dialog box is displayed. Perform the following steps:

   a. In the **Node** name field, enter a name for the node to be created under Sample Account.

   b. Enter or browse to the location of the XDR schema where Application Explorer generates the service or event schema.

4. Click **Add**.

The Schemas pane is displayed.



5. To view the XML for a schema, click the **Event Schema** tab.

   The results appear in the right pane.

```
<?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>Schema name:
        SiebelMessage</xsd:documentation>
    </xsd:annotation>
    <!--  [XDR-XSD] "SiebelMessage" element    --
    >
  - <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"
```

6. Click the browser **Back** button to return.

   A directory structure is created to store the schemas.

   You can now create an Integration Object node for Siebel. See "Creating Integration Object (IO) Nodes for Siebel" on page 2-23 for more information.

Creating Integration Object (IO) Nodes for Siebel

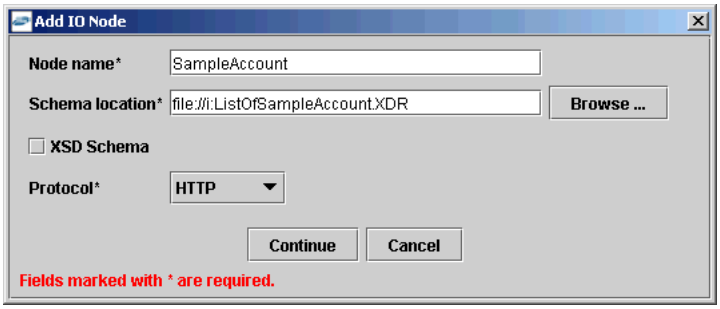
To create an Integration Object node for Siebel, perform the following steps:

1. In Application Explorer, connect to a defined target. See "Connecting to a Defined Target" on page 2-12 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 box is displayed.



   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.

4. Enter a node name, for example SampleAccount in the **Node name** field and a path to the Sample Account XDR 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**.

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 wish 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 OracleAS Adapter J2CA, the Common Client Interface provides integration services using the adapters.

This section contains the following topics:

Creating a Service for a Business Service Having a Method Argument of Type Integration Object Hierarchy Type (Siebel Message)

OracleAS Adapter for Siebel allows you to add a service node for a Business Service that includes methods containing method arguments having hierarchy data types.

Important limitations:

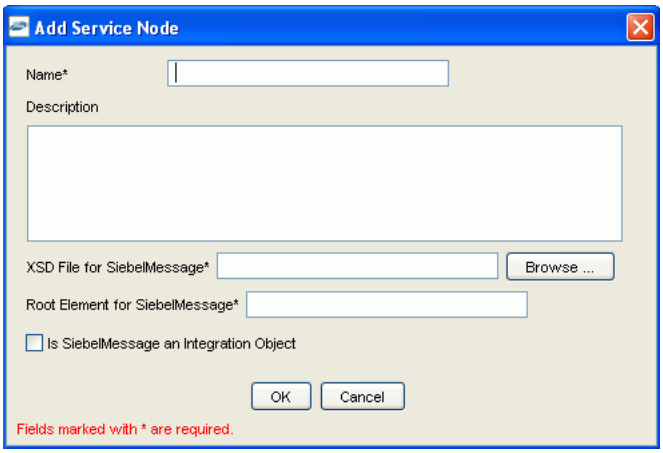
- The adapter supports only Integration Object hierarchy data types.
- Adding a Service node requires that you have previously generated an XSD schema for the Integration Object. For more information on generating XSD schemas for Siebel Integration Objects, see "Creating Schemas for Siebel Integration Objects" on page 2-18.
- 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, you will get the following error:

  ```
  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**.

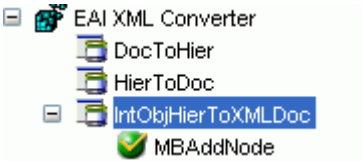
The Add Service Node dialog box is displayed.



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.

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 box 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 box 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**.

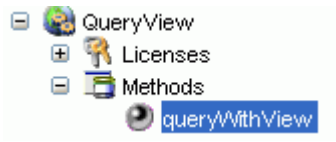
   Application Explorer switches the view to the **Business Services** node, and the new Web service appears in the left pane.

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.

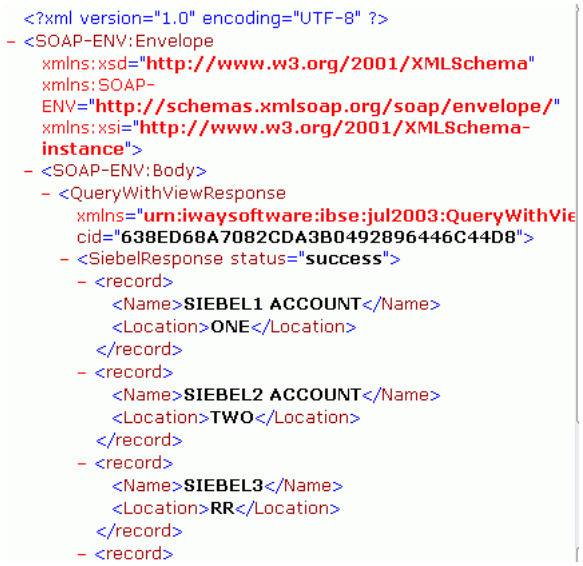


4. Expand the **Methods** node under the service and select the method you want to test.

   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:

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

```
</SOAP-ENV:Header>
```

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

Generating WSDL (J2CA Configurations Only)

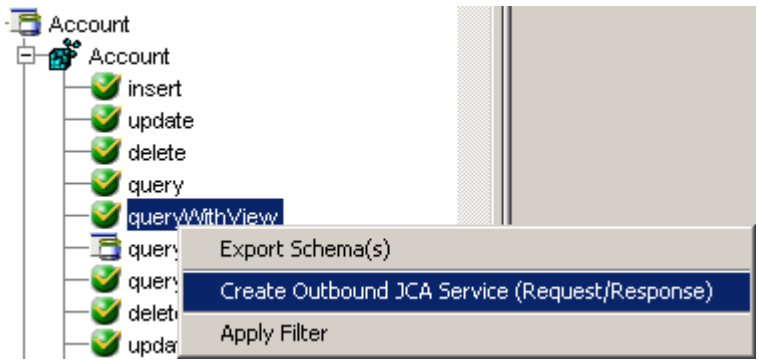
The Web Service Definition Language (WSDL) description of a Web service enables you to make the service available to other services within a host server. You use Application Explorer to create both request-response (outbound) and event notification (inbound) JCA services of the adapter.

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:

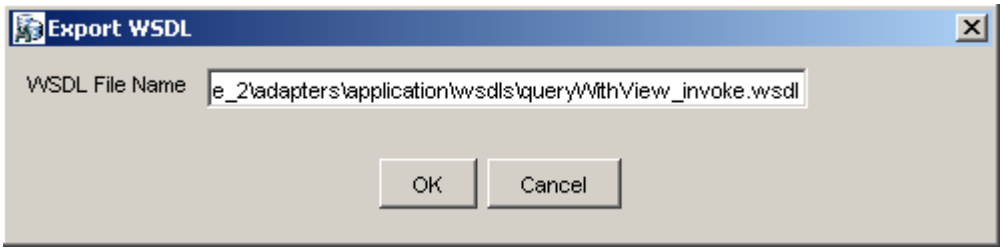
1. Under your connected Siebel target, expand **Business Object**, **Account**, **Account**. Navigate to an object and right-click the object.

   The following menu is displayed.



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

   The Export WSDL dialog box is displayed.



3. Accept the default name and location 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`.

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

4. Click **OK**.

The WSDL file is saved in the specified location.

The procedure for generating WSDL for event notification is similar to request-response. To generate WSDL for event notification, you must first create a channel for every event. See "Generating WSDL for Event Notification" on page 5-14 for a detailed example.

Configuring an Event Adapter

Events are generated as a result of 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 port and a channel.

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

- A **port** associates a particular business object exposed by an adapter with a particular disposition. A disposition defines the protocol and location of the event data. The port defines the end point of the event consumption. See "Creating and Modifying an Event Port" on page 2-29 for more information.
- A **channel** represents configured connections to particular instances of back-end or other types of systems. A channel binds one or more event ports to a particular listener managed by an adapter. See "Creating and Modifying a Channel" on page 2-31 for more information.

> **Note:** OC4J currently conforms to J2CA 1.0, which does not call for event capabilities. When conforming to J2CA 1.0, only service interactions are supported.

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

This section contains the following topics:

Creating and Modifying an Event Port

You can create an event port for a Siebel Integration Object from the Events node in Application Explorer. To create event ports for Siebel Business Objects and Siebel Business Services, you must use the Events node.

> **Note:** You are not required to create event ports for J2CA configurations. You must create event ports for BSE configurations only.

Creating an Event Port from the Adapters Node

For Siebel Integration Objects, you can bypass the Events node and create an event port directly from the Adapters node.

1. Select the IO node you created.
2. Right-click the IO node and select **Add Port**.

   The Add Port dialog box is displayed. Perform the following steps:

   a. Enter a name for the event port and provide a brief description.

   b. From the list, select the required disposition, for example, File.

   c. In the URL field, enter the disposition URL.
3. Click **OK**.
4. Create a DTD for a Siebel event in one of two ways:

   - Right-click the IO node under **Integration Objects**, and select **Create Web Service**. A DTD created this way will have the name of *<Node_Name>*_`request.dtd`.

     Or

   - Right-click the IO node under **Integration Objects**, and select **Create Event Port**. A DTD created this way will have a name of *<Node_Name>*_`event.dtd`.

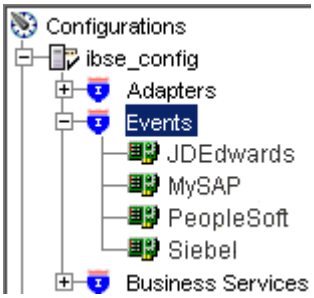
The following procedure describes how to create an event port from the Events node for an File disposition using Application Explorer.

To create an event port for Siebel Integration Objects, you must first indicate the location of the XDR schema for that object. See "Creating a Schema from a Siebel XDR Schema" on page 2-21 for more information.

Creating a File Event Port from the Events Node

To create a specific event port for File:

1. Click the **Events** node.



2. Expand the **Siebel** node.
3. Right-click the **Ports** node and select **Add Port**.

   The Add Port dialog box is displayed. Perform the following steps:

   a. Enter a name for the event port and provide a brief description.

   b. From the **Protocol** list, select **File**.

   c. In the **URL** field, specify a destination file to which the event data is written using the following format:

`file://location`

Where `APPNAME` is the adapter name for the EIS Adapter Plugin you configured. The value for `APPNAME` must be in uppercase.

d. From the **Disposition** protocol list, select **File**.

The following table defines the parameters for the disposition.

Parameter	Description
location	Full directory path and file name to which the data is written..

4. Click **OK**.

The port appears under the ports node in the left pane. In the right pane, a table appears that summarizes the information associated with the event port you created.

You are ready to associate the event port with a channel. See "Creating and Modifying a Channel" on page 2-31 for more information.

Editing an Event Port

To edit an event port:

1. In the left pane, select the event port you want to edit.
2. Right-click the port and select **Edit**.

   The Edit Port pane is displayed.
3. Make the required changes and click **OK**.

Deleting an Event Port

To delete an event port:

1. Select the event port you want to delete.
2. Right-click the port and select **Delete**.

   A confirmation dialog box is displayed.
3. To delete the event port you selected, click **OK**.

   The event port disappears from the list in the left pane.

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.

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

If you are planning to integrate OracleAS Adapter for Siebel with BPEL Process Manager, do not start the channel, as it is managed by the BPEL PM Server. If you start the channel for testing and debugging purposes, stop it before runtime.

Three channel types are available:

- HTTP
- MQ Series
- File

> **Note:** OC4J currently conforms to J2CA 1.0, which does not call for event capabilities. When conforming to J2CA 1.0, only service interactions are supported.

Creating an HTTP Channel

To create a channel:

1. Click the **Events** node.

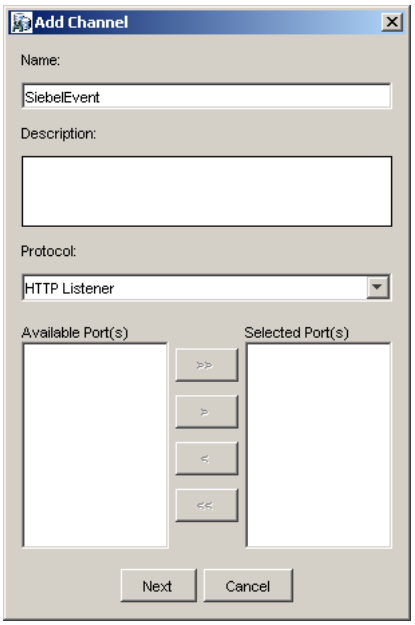
   The Events window is displayed. The adapters that appear in the left pane support events.

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 box is displayed.



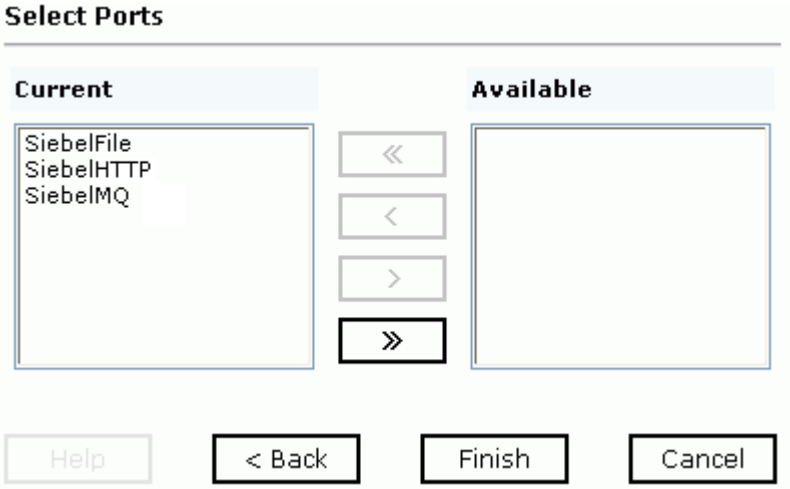
   Perform the following steps:

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

   b. Enter a brief description.

   c. From the list, select **HTTP Listener**.

The following image shows the Select Ports dialog box.



d. Select an event port from the list of current ports.

e. To transfer the port to the list of available ports, click the double right (>>) arrow. To associate all the event ports, control-click to select each port or click one port and press **Control+A**. Then, click the double right (>>) arrow.

4. Click **Next**.

5. When the dialog box is displayed, enter the system information as specified in the following table.

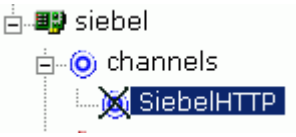
Parameter	Description
Port	Port on which to listen for Siebel event data.
Server port	Port on which the host database is listening.
Synchronization Type	Synchronization types are not applicable to Siebel events.

6. Click **OK**.

   The summary pane is displayed.

   A summary 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**.

Creating an MQ Series Listener

To create an MQ Series listener:

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**.

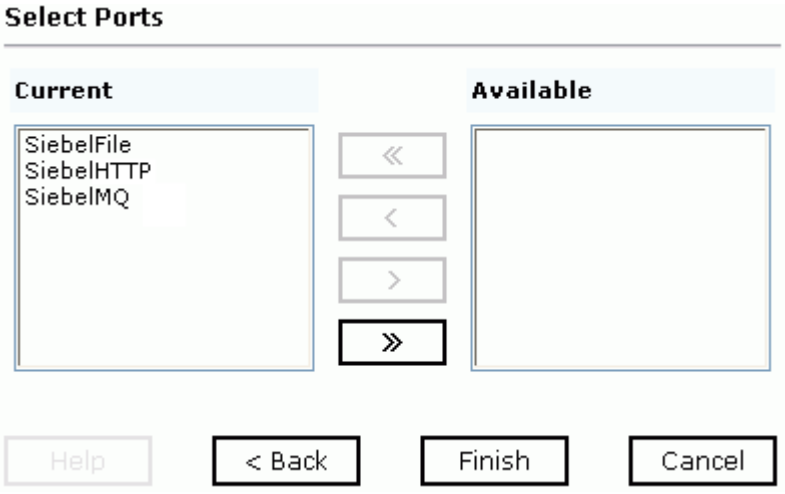
   The Add a new channel pane is displayed. Perform the following steps:

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

   b. Enter a brief description.

   c. From the list, select **MQ Series Listener**.

      The following image shows the Select Ports dialog box.



   d. Select an event port from the list of current ports.

   e. To transfer the port to the list of available ports, click >>. To associate all the event ports, control-click to select each port or click one port and press **Control+A**. Then, click >>.

4. Click **Next**.

5. When the dialog box is displayed, enter the system information as follows.

   a. In the **Request** tab, enter values for the following parameters:

Parameter	Description
Queue manager name	The host on which the MQ Server is located (MQ Client only).
MQ server host for MQClient operation	Port on which the host database is listening.

Parameter	Description

MQ server port for MQClient operation	The number to connect to an MQ Server queue manager (MQ client only).
	`REQUEST`
	`REQUEST_RESPONSE`
	`REQUEST_ACK`

MQ server channel for MQClient operation	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.
Document type XML	Leave the default selection.

Request queue name	Queue where the message is routed and where request documents are received. The name of the queue is case-sensitive and conforms to the following format:
	`Host\queue type$\qName`
	Host
	Is the machine name where the MQ Series queuing system is running.
	queue type
	Private queues are queues that are not published in Active Directory and appear only on the local computer where they reside. Private queues are accessible only by Message Queuing applications that recognize the full path name or format name of the queue.
	qName
	Is the name of the queue where messages are placed, for example,
	`iwaykxc1\Private$\siebel`

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

Parameter	Definition
Synchronization Type	Synchronization types are not applicable to Siebel events.

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

Parameter	Definition
Error Directory	Directory to which documents with errors are written.
Message wait interval (msec)	The interval (in milliseconds) when to check for new input. The default is 3 seconds. Optional.
Mode of operation	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**.

The summary pane is displayed.

A summary 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**.

Creating a File Listener

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**.

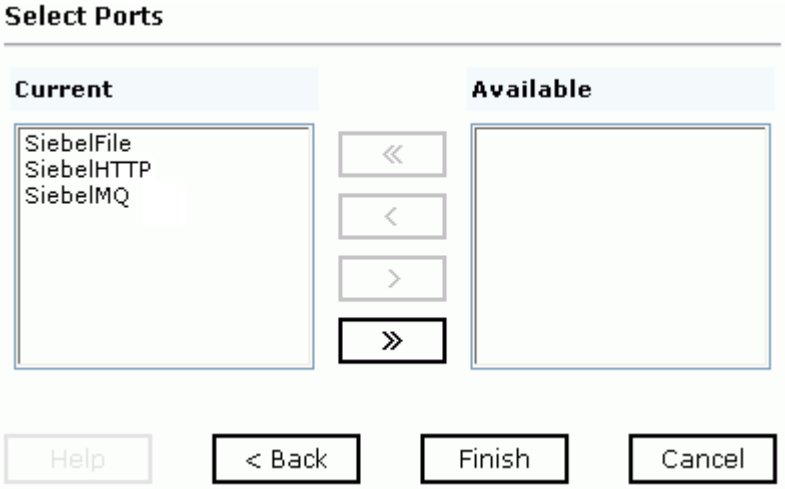
   The Add Channel dialog box is displayed. Perform the following steps:

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

   b. Enter a brief description.

   c. From the list, select **File Listener**.

      The following image shows the Select Ports dialog box.



   d. Select an event port from the list of current ports.

   e. To transfer the port to the list of available ports, click the double right (>>) arrow. To associate all the event ports, control-click to select each port or click one port and press **Control+A**. Then, click the double right (>>) arrow.

4. Click **Next**.

5. When the dialog box is displayed, enter the system information as follows.

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 as a result of this operation.

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

Parameter	Definition
Synchronization Type	Synchronization types are not applicable to Siebel events.
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**.

   The summary pane is displayed. A summary 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**.

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 pane is displayed.

3. Make the required changes to the channel configuration and click **OK**.

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**.

   A confirmation dialog box is displayed.
3. To delete the channel you selected, click **OK**.

   The channel disappears from the list in the left pane.

3

OC4J Deployment and Integration

This chapter describes Oracle Application Server Containers for J2EE (OC4J) deployment and integration with OracleAS Adapter for Siebel.

This chapter discusses the following topics:

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

See Also:

- *Oracle Application Server Adapter Concepts*

Adapter Integration with OC4J

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

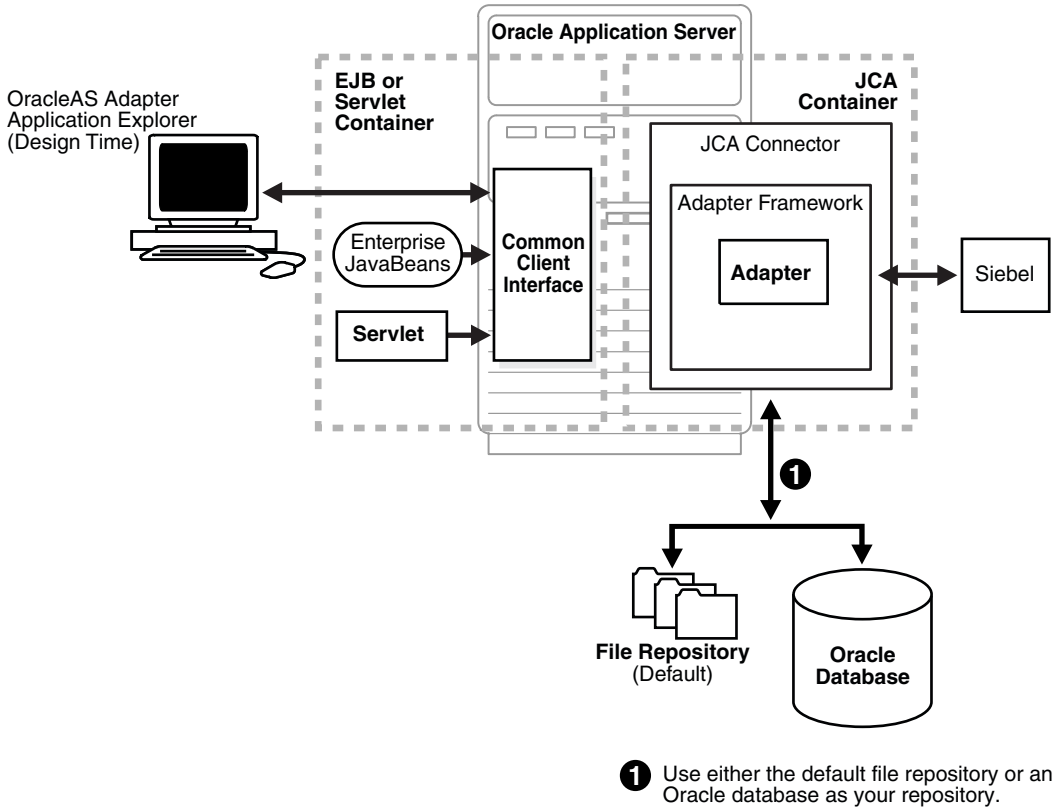
See Also:

- "Oracle Application Server Adapters Integration with OC4J" in *Oracle Application Server Adapter Concepts*

Deployment of Adapter

Figure 3–1 shows deployment of the Connector to the Oracle Application Server. In a runtime service scenario, an EJB, 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.

Figure 3–1 Oracle Application Server Adapter J2CA Architecture



See Also:

- *Oracle Application Server Adapter Concepts*

Updating Adapter Configuration

During the J2CA deployment of OracleAS Adapter for Siebel, OC4J generates a deployment descriptor called `oc4j-ra.xml`, located in *`OC4J_home`*`\integration\orabpel\system\appserver\oc4j\j2ee\home\application-deployments\default\iwafjca`.

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

Creating a Managed Connector Factory Object

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

```
<?xml version="1.0"?>
<!DOCTYPE oc4j-connector-factories PUBLIC "-//Oracle//DTD Oracle Connector
9.04//EN" "http://xmlns.oracle.com/ias/dtds/oc4j-connector-factories-9_04.dtd">
<oc4j-connector-factories>
  <connector-factory location="eis/OracleJCAAdapter/DefaultConnection"
```

```
connector-name="IWAFJCA10">
  <config-property name="IWayHome" value="../../adapters/application"/>
  <config-property name="IWayConfig" value="jca_sample"/>
   <config-property name="IWayRepoURL" value=""/>
   <config-property name="IWayRepoUser" value=""/>
   <config-property name="IWayRepoPassword" value=""/>
   <config-property name="logLevel" value="debug"/>
   </connector-factory>
</oc4j-connector-factories>
```

The parameters defined in the `oc4j-ra.xml` file are described in the following table:

Parameter Name	Description
IWayHome	The base installation directory for the OracleAS packaged application adapter.
IWayConfig	The adapter configuration name as defined in Application Explorer. For example, OracleAS Adapter for Siebel has a preconfigured `jca_sample` configuration in Application Explorer.
IWayRepoURL	The URL to use when opening a connection to the database. This is necessary only when using an Oracle database as the BSE repository. See"Configuring BSE System Settings" in on page 2-3 for more information.
IWayRepoUser	User name to use when connecting to the database. This is necessary only when using an Oracle database as the BSE repository. See"Configuring BSE System Settings" in on page 2-3 for more information.
IWayRepoPassword	Password. If provided, it overwrites configuration. This is necessary only when using an Oracle database as the BSE repository. See"Configuring BSE System Settings" in on page 2-3 for more information.
loglevel	It overwrites the level set by the ManagedConnectionFactory property.

Creating Multiple Managed Connector Factory Objects

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

```
<?xml version="1.0"?>
<!DOCTYPE oc4j-connector-factories PUBLIC "-//Oracle//DTD Oracle Connector
9.04//EN" "http://xmlns.oracle.com/ias/dtds/oc4j-connector-factories-9_04.dtd">
<oc4j-connector-factories>
   <connector-factory location="eis/OracleJCAAdapter/DefaultConnection"
connector-name="IWAFJCA10">
   <config-property name="IWayHome" value="../../adapters/application"/>
   <config-property name="IWayConfig" value="jca_sample"/>
   <config-property name="IWayRepoURL" value=""/>
   <config-property name="IWayRepoUser" value=""/>
   <config-property name="IWayRepoPassword" value=""/>
   <config-property name="logLevel" value="debug"/>
   </connector-factory>
```

```
</oc4j-connector-factories>
```

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

```
<?xml version="1.0"?>
<!DOCTYPE oc4j-connector-factories PUBLIC "-//Oracle//DTD Oracle Connector
9.04//EN" "http://xmlns.oracle.com/ias/dtds/oc4j-connector-factories-9_04.dtd">
<oc4j-connector-factories>
   <connector-factory location="eis/OracleJCAAdapter/DefaultConnection1"
connector-name="IWAFJCA10">
   <config-property name="IWayHome" value="../../adapters/application"/>
   <config-property name="IWayConfig" value="jca_sample"/>
   <config-property name="IWayRepoURL" value=""/>
   <config-property name="IWayRepoUser" value=""/>
   <config-property name="IWayRepoPassword" value=""/>
   <config-property name="logLevel" value="debug"/>
   </connector-factory>
<connector-factory location="eis/OracleJCAAdapter/DefaultConnection2"
connector-name="IWAFJCA10">
   <config-property name="IWayHome" value="../../adapters/application"/>
   <config-property name="IWayConfig" value="jca_sample2"/>
   <config-property name="IWayRepoURL" value=""/>
   <config-property name="IWayRepoUser" value=""/>
   <config-property name="IWayRepoPassword" value=""/>
   <config-property name="logLevel" value="debug"/>
   </connector-factory>
</oc4j-connector-factories>
```

How to Write a Java Application Client Using the CCI API

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

Step 1. Obtain the Connection Factory

The connection factory is obtained by JNDI lookup.

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

Step 2. Obtaining a Connection for the Adapter

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

Parameter Name	Description
adapterName	Name of the packaged application adapter.
config -	Adapter configuration name. NOT REQUIRED FOR IWAEAdapter.
language	Default is `en`.
country	Default is `us`.
userName	User name. If provided, it overwrites configuration.

Parameter Name	Description
password	Password. If provided, it overwrites configuration.
logLevel	It overwrites the level set by the ManagedConnectionFactory property.

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

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

The iWAFConnectionSpec can initiate an interaction with Siebel if the adapter name and configuration parameters are specified in the ConnectionSpec. For example,

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

In this snippet, `ADAPTER` and `TARGET` refer to the adapter being deployed, in this case Siebel, and the name of a target defined in Application Explorer. See"Complete Code Sample" on page 3-6 for more information.

Step 3. Create interaction with interactionSpec for runtime

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

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

Step 4. Create Input Record and Execute Interaction

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

A standard J2CA Indexed Record is used in this example:

```
// Use J2CA IndexRecord, named "input" for runtime processing.
IndexedRecord rIn = cf.getRecordFactory().createIndexedRecord("input");
rIn.add(msg_run);
  IndexedRecord rOut = (IndexedRecord)i.execute(is, rIn);
System.out.println((String)rOut.get(0));
```

A special record is supported in this example:

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

Where *msg_run* is an instance XML document generated from the schema created by Application Explorer.

Complete Code Sample

The following is a sample of the complete code:

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

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

private static String  HOME    = "c:/iway/xfoc/components/iwafcont/dist";
private static String  CONFIG    = "base";
private static String  LOG_LEVEL = "FATAL";

private static String ADAPTER = "Siebel";
private static String TARGET  = "Siebel_connection";

// Input Message
private static String msg_run = "<Siebel/>";

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

// 1. Getting the Connection factory through JNDI lookup
// -----------------------------------------------------
  InitialContext context = new InitialContext();
  ConnectionFactory cf = (ConnectionFactory)context.lookup(iwayJndi)
// 2. Getting a connection for a particular adapter target, in this case Siebel
// -----------------------------------------------------
  IWAFConnectionSpec cs = new IWAFConnectionSpec();
  cs.setAdapterName(ADAPTER);
  cs.setConfig(TARGET);
  cs.setLogLevel(LOG_LEVEL);  // Adapter layer log level
  Connection c = cf.getConnection(cs);// where cf is the connection factory

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

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

  // 4.1 Using JCA standard Indexed Record
// Use JCA IndexRecord, named "input" for runtime processing.
IndexedRecord rIn = cf.getRecordFactory().createIndexedRecord("input");
```

```
rIn.add(msg_run);
  IndexedRecord rOut = (IndexedRecord)i.execute(is, rIn);
System.out.println((String)rOut.get(0));

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

  } // main()

}
```

4

Integration with Oracle BPEL Process Manager

OracleAS 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 of Adapter Integration with Oracle BPEL Process Manager

To integrate with Oracle BPEL Process Manager, OracleAS Adapter for Siebel must be deployed in the same OC4J container as Oracle BPEL Process Manager. The underlying adapter services must be exposed as WSDL files, which are generated during design time in Oracle Application Server Adapter Application Explorer (Application Explorer) for both request-response (outbound) and event notification (inbound) services of the adapter. See "Generating WSDL (J2CA Configurations Only)" on page 2-28 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 a BPEL designer and deployed to a BPEL server. Upon deployment to the BPEL server, every newly built process is automatically deployed to the Oracle BPEL Console, where you run, monitor, and administer BPEL processes, as well as listen to adapter events.

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

```
OracleAS_home\j2ee\OC4J_BPEL
```

See Also:

- *Oracle Application Server Adapter Concepts*
- *Oracle BPEL Process Manager Developer's Guide*

Deployment of Adapter

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

See Also: *Oracle Application Server Adapter Concepts*

Design Time

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

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

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

Before you design a BPEL process, you must create a schema and generate the respective WSDL file using Application Explorer. See "Generating WSDL (J2CA Configurations Only)" on page 2-28 for more information.

Namespace Requirements

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

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

1. Input XML for BPEL based on unqualified namespaces:

```
<?xml version="1.0" encoding="UTF-8"?>
<Siebel location="S/BO/Account/Account/query">
<select>
<Name>A*</Name>
</select>
  <Field>Name</Field>
  <Field>Country</Field>
  <Field>State</Field>
  <Field>Account Status</Field>
  <Field>Employees</Field>
</Siebel>
```

2. Input XML for BPEL based on qualified namespaces:

```
<Siebel xmlns="urn:iwaysoftware:adapter:siebel:oct2004:request"
location="S/BO/Account/Account/query">
            <select>
              <Name>ChennaiQA</Name>
            </select>
            <field>Account Competitors</field>
            <field>Name</field>
            <field>City</field>
            <field>Street Address</field>
            <field>Country</field>
            <field>Currency Code</field>
            <field>State</field>
            <field>Account Status</field>
</Siebel>
```

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

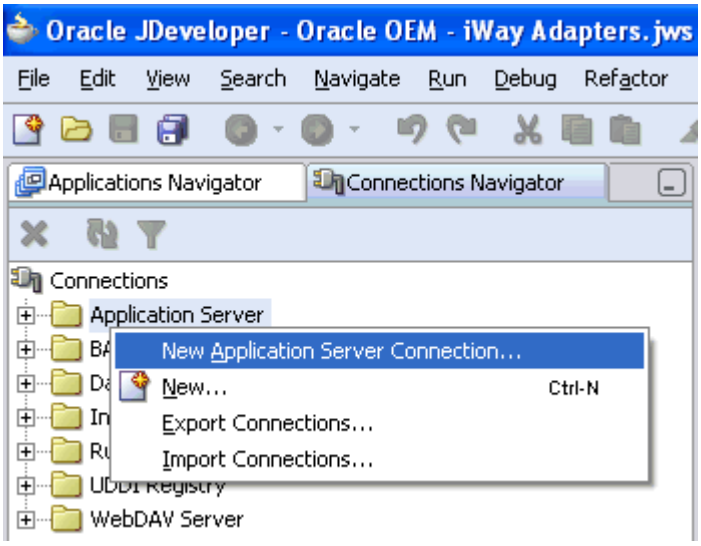
Design a BPEL Process for Request-Response Service (Outbound)

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

Create a New Application Server Connection

To create a new Application Server connection:

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

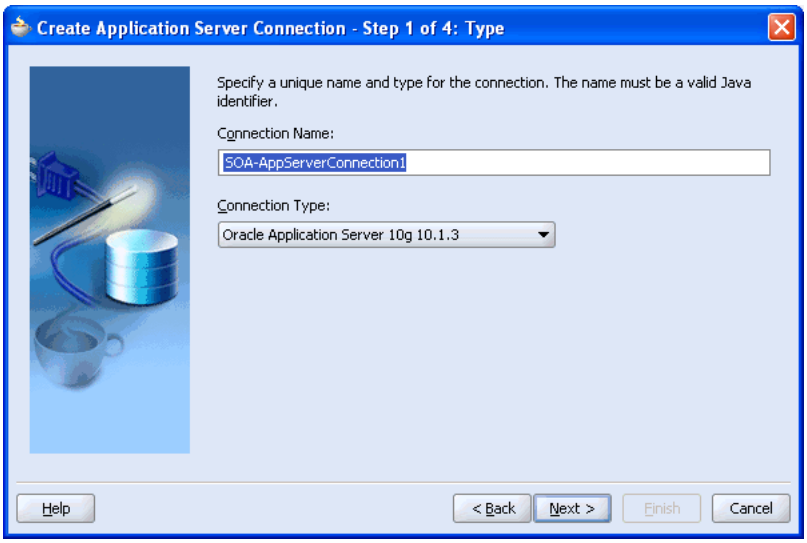


2. Right-click **Application Server** and select **New Application Server Connection**.

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

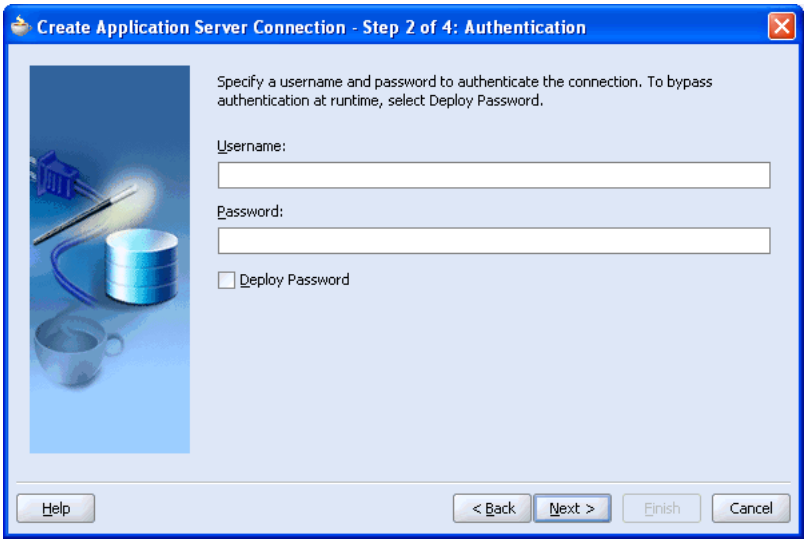
3. Click **Next**.

The Create Application Server Connection - Step 1 of 4: Type dialog box is displayed.



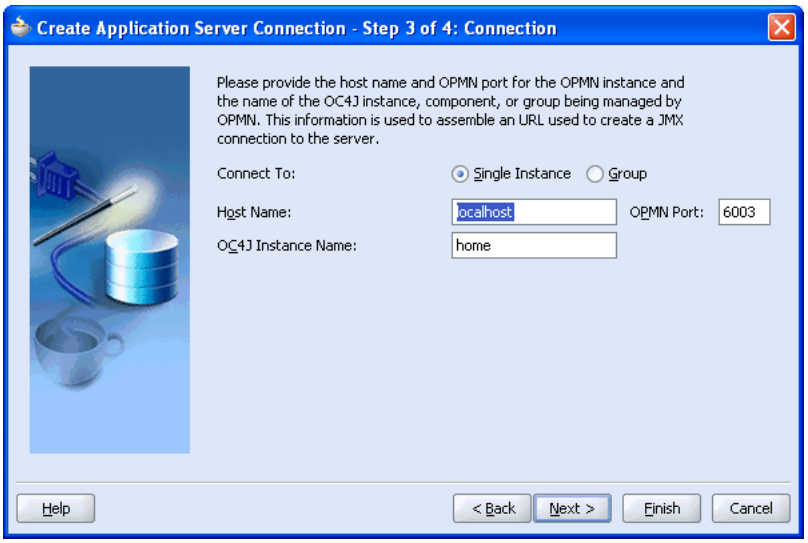
4. Specify a unique name and select a connection type for your Application Server connection and click **Next**.

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



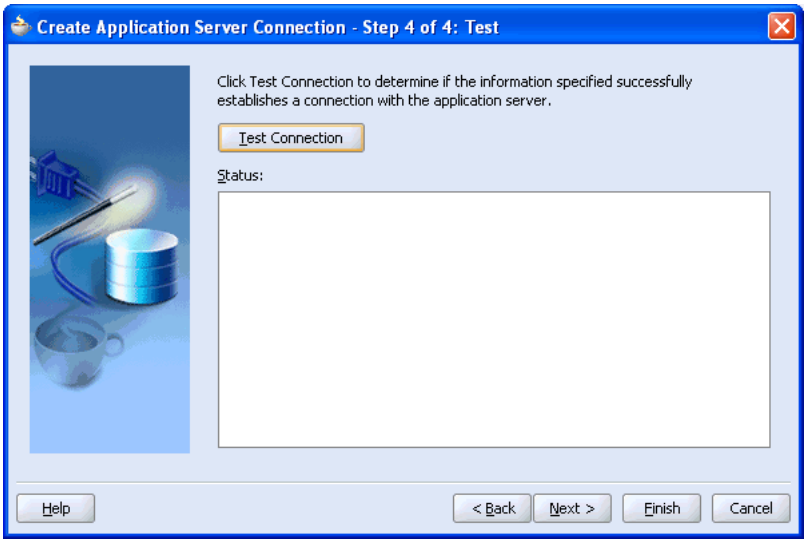
5. Specify a valid user name and password for the Application Server you wish to connect to.
6. Select **Deploy Password**.
7. Click **Next**.

   The Create Application Server Connection - Step 3 of 4: Connection dialog box is displayed.



8. Select the **Single Instance** connection option.
9. Enter **localhost** as the host name and **6003** for the OPMN port.
10. Enter **home** as the OC4J instance name
11. Click **Next**.

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

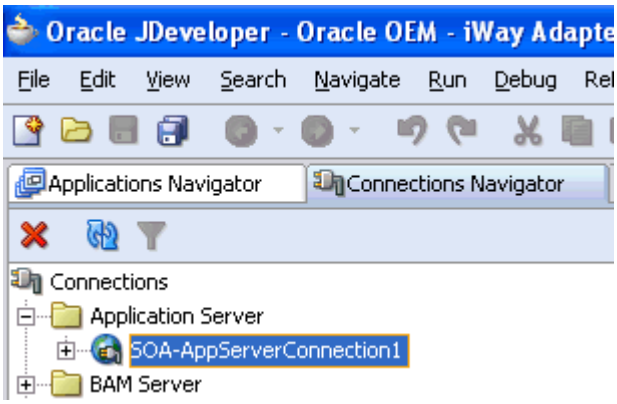


12. Click **Test Connection**.

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

13. Click **Finish**.

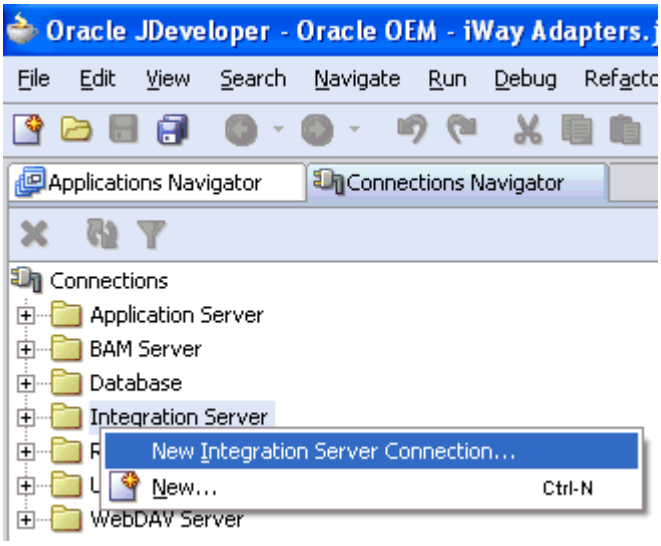
    Your newly created Application Server connection is displayed in the Connections Navigator tab under the Application Server node.



Create a New Integration Server Connection

To create a new Integration Server connection:

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

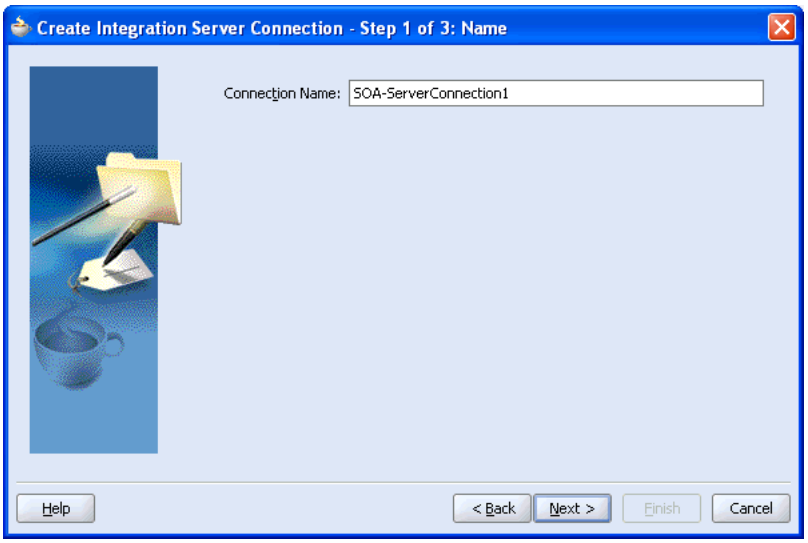


2. Right-click **Integration Server** and select **New Integration Server Connection**.

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

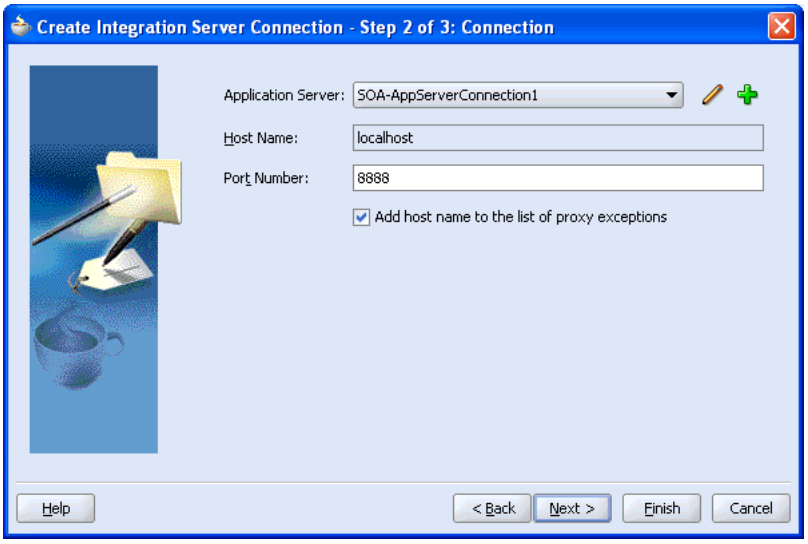
3. Click **Next**.

The Create Integration Server Connection - Step 1 of 3: Name dialog box is displayed.



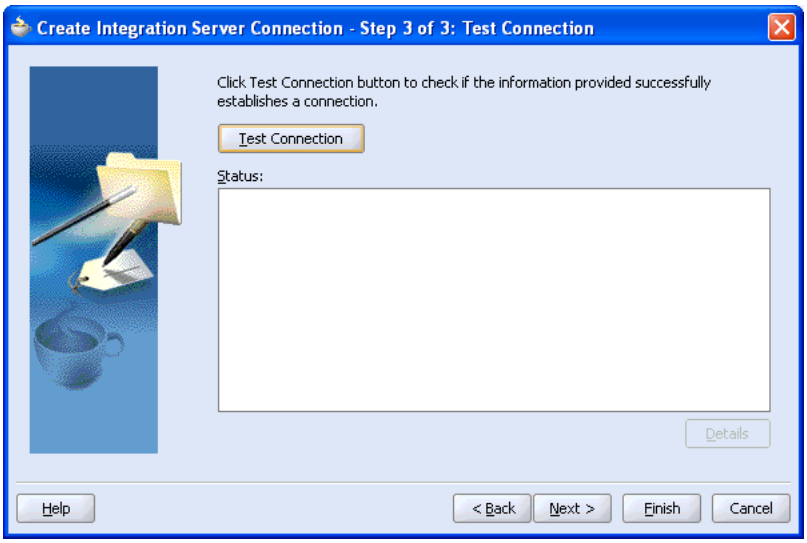
4. Specify a unique name and click **Next**.

   The Create Integration Server Connection - Step 2 of 3: Connection dialog box is displayed.



5. Select an Application Server connection, which is already created.
6. Enter **localhost** as the host name and **8888** for the port number.
7. Select **Add host name to the list of proxy exceptions** and click **Next**.

   The Create Integration Server Connection - Step 3 of 3: Test Connection dialog box is displayed.

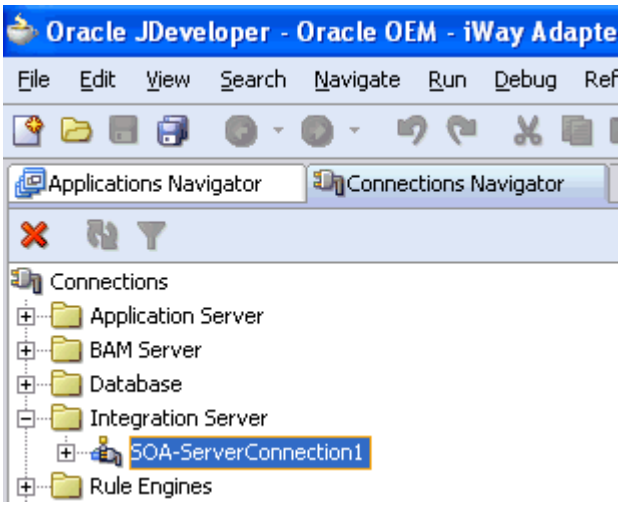


8. Click **Test Connection**.

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

9. Click **Finish**.

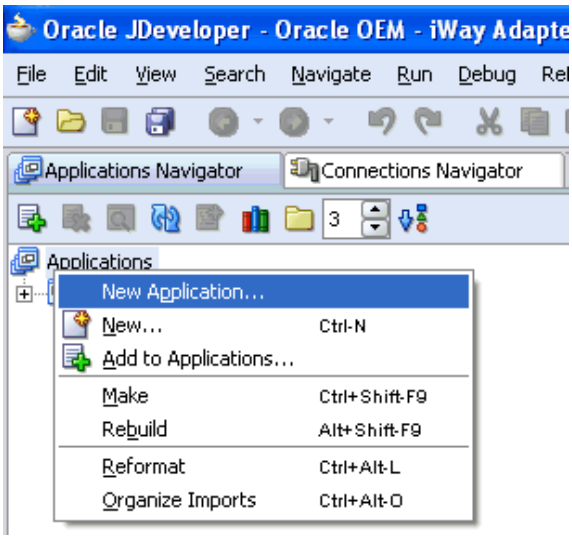
   Your newly created Integration Server connection is displayed in the Connections Navigator tab under the Integration Server node.



Create a New BPEL Project for Outbound Interaction (Synchronous Process)

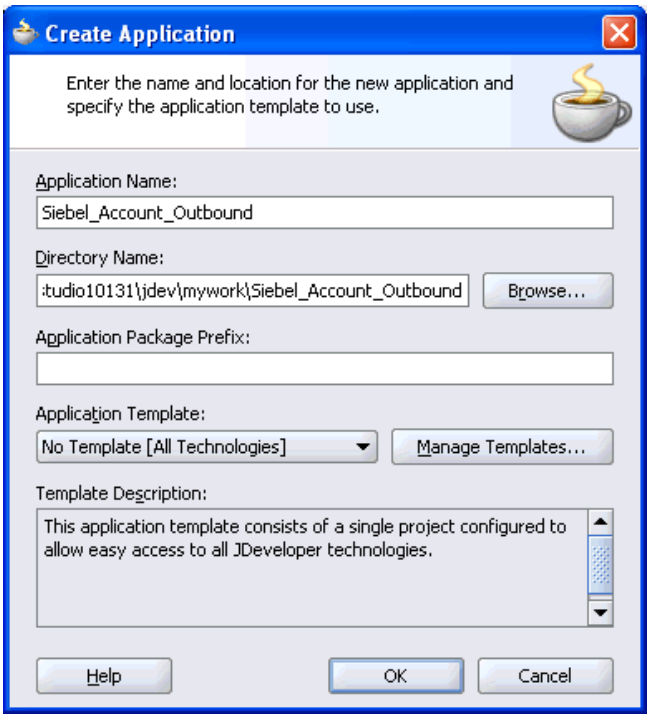
To create a new BPEL project for a synchronous process:

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



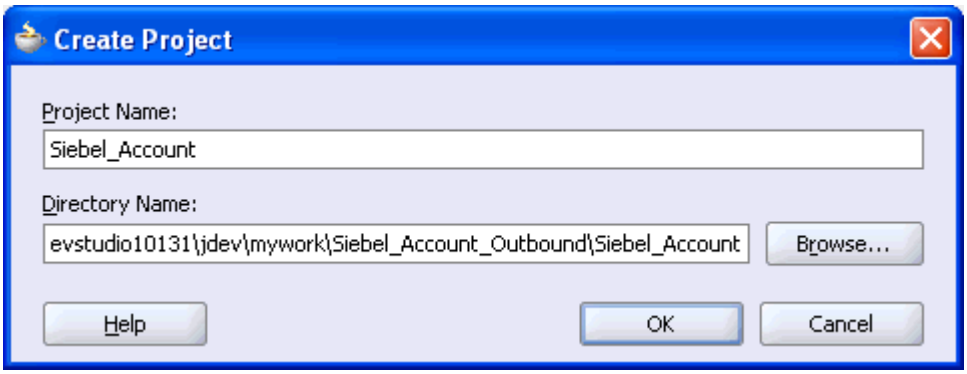
2. Right-click the Applications node and select **New Application**.

   The Create Application dialog box is displayed.



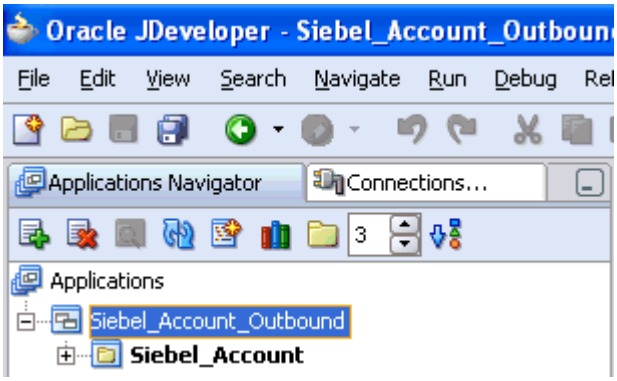
3. Enter a unique name for your application and click **OK**.

   The Create Project dialog box is displayed.



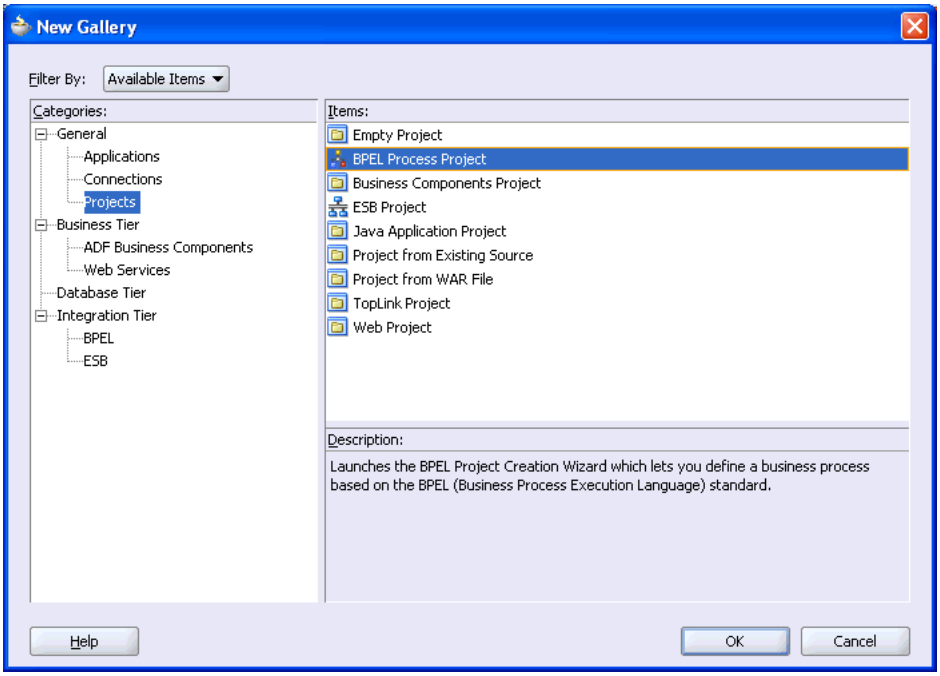
4. Enter a unique name for your project and click **OK**.

   Your new application is displayed in the Applications Navigator tab under the Applications node.



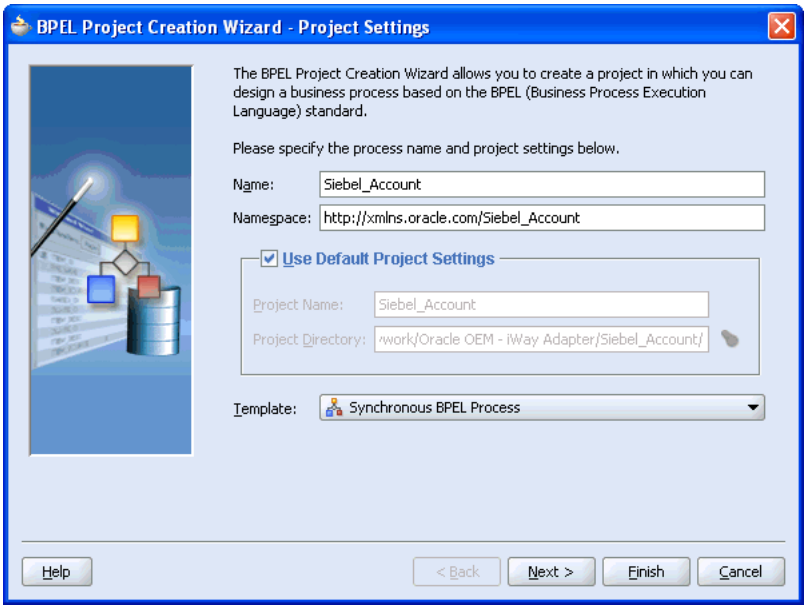
5. Right-click the application node you created and select **New Project**.

   The New Gallery window is displayed.



6. From the Items list, select **BPEL Process Project** and click **OK**.

   The BPEL Project Creation Wizard - Project Settings dialog box is displayed.



7. Perform the following steps:

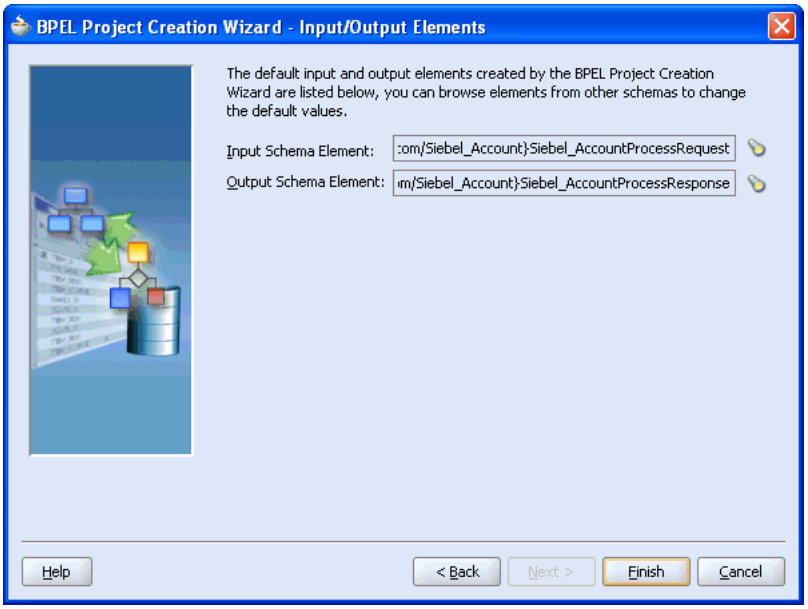
   a. Specify a name for the BPEL process.

      The Namespace field is updated automatically.

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

8. Click **Next**.

   The BPEL Project Creation Wizard - Input/Output Elements dialog box is displayed.



9. Review the input/output schema elements that are created by the BPEL Project Creation Wizard and click **Finish**.

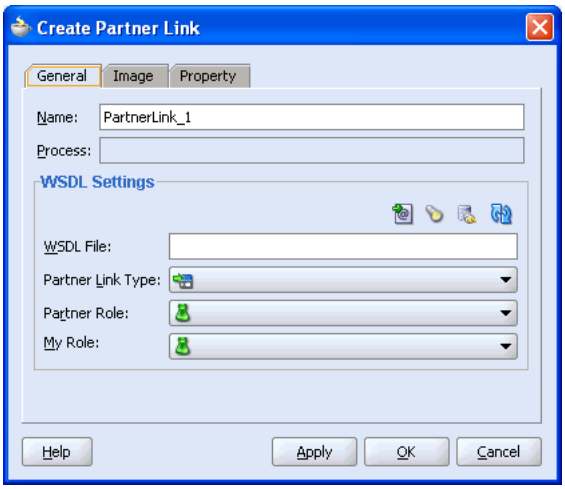
Create an Outbound PartnerLink Activity

When designing a BPEL process, a PartnerLink activity must be created to invoke the Siebel service. A PartnerLink describes a set of operations within a Web service. The WSDL document is the external contract to which the Web service conforms. Given a WSDL, any BPEL process can initiate a Web service through a PartnerLink.

To create an outbound PartnerLink using the WSDL file you generated in Application Explorer:

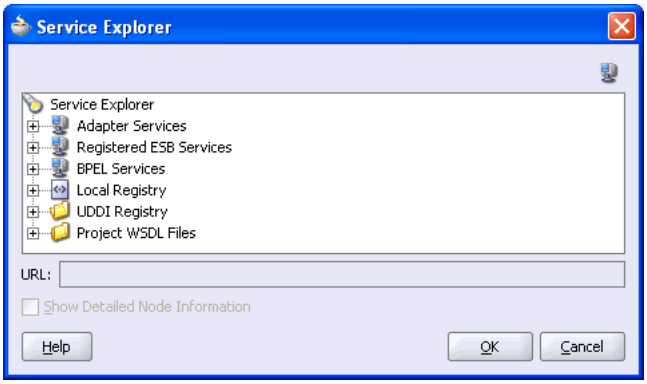
1. From the Services pane on the right, drag and drop a PartnerLink to the visual editor.

   The Create Partner Link dialog box is displayed.



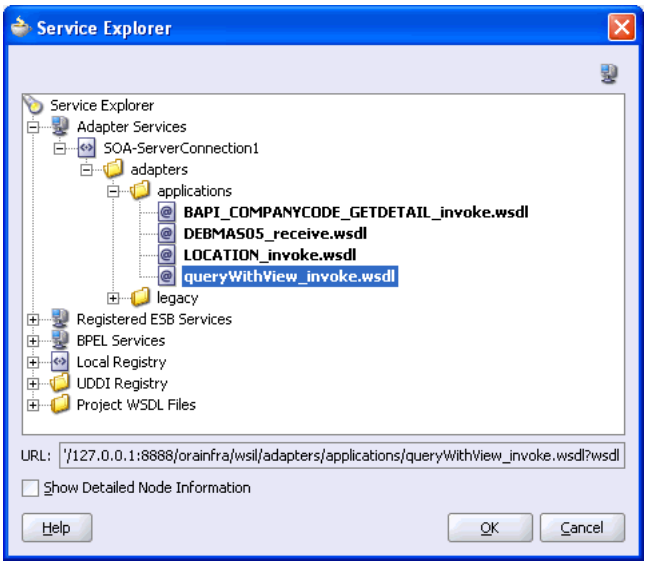
2. Click the **Service Explorer** icon (second icon from the left preceding the **WSDL File** field).

   The Service Explorer dialog box is displayed.



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

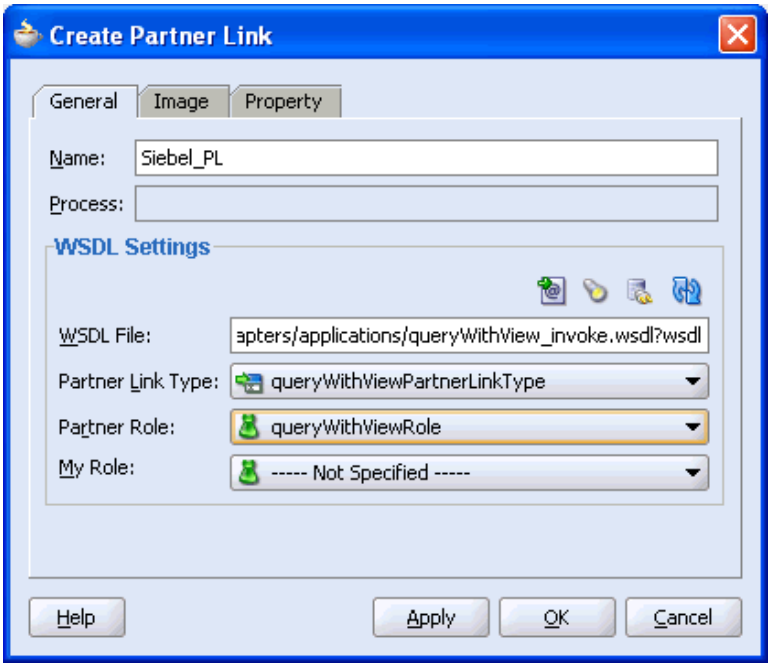
   The WSDL tree displayed in the WSDL Chooser dialog box lists any WSDL files you have created using Application Explorer. The WSDL tree is generated by a WSDL servlet, which is automatically deployed as part of the Oracle BPEL Server installation.



Note: If you have organized your WSDL files in subfolders, the WSIL browser will display the full tree structure of your WSDL hierarchy. By default, the names of all WSDL files generated for outbound adapter services end with `_invoke`.

4. Select **queryWithView_invoke.wsdl** and click **OK**.

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



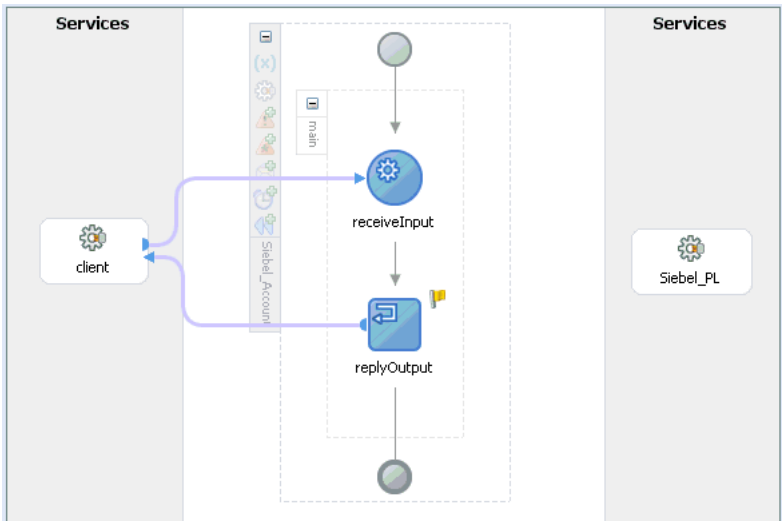
   Perform the following steps:

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

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

5. Click **OK**.

   The new PartnerLink appears in the visual editor.



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

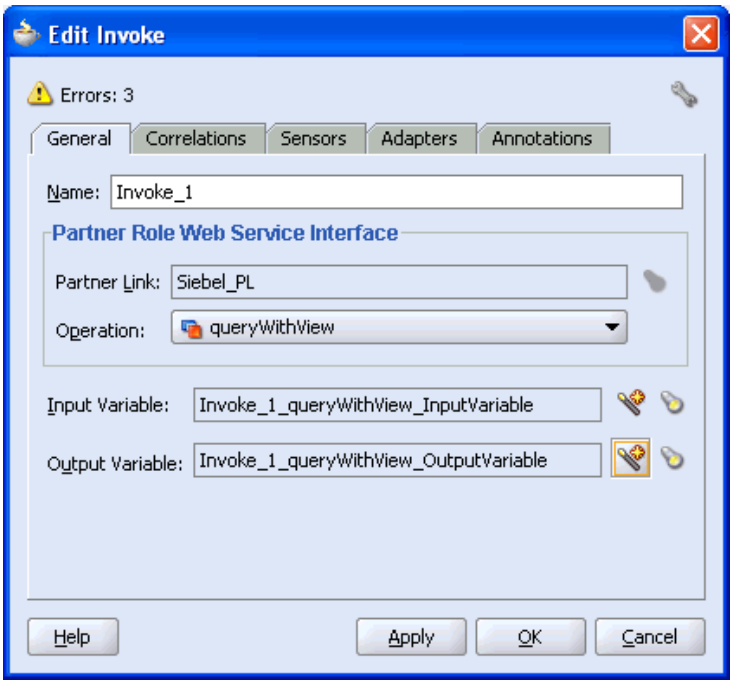
Create an Outbound Invoke Activity

This activity enables you to specify an operation you want to invoke for the service identified by its PartnerLink. The Invoke activity opens a port in the process that is used to send and receive data. It uses this port to submit required data and receive a response. For synchronous callbacks, only one port is needed for both the send and the receive functions.

To create an outbound Invoke activity:

1. From the **Process Activities** pane on the right, drag an **Invoke** activity to the visual editor and place it between the Receive activity (`receiveInput`) and the Reply activity (`replyOutput`).
2. Extend a connection between the Invoke activity and your newly-created PartnerLink.

The Edit Invoke dialog box is displayed. Note that the Partner Link and Operation fields are automatically populated.



Note: Ignore any invalid settings and error warnings.

Perform the following steps:

a. In the **Name** field, provide a meaningful name for the Invoke activity.

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

A global variable is automatically created in the Input Variable field.

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

A global variable is automatically created in the Output Variable field.

d. Click **Apply**.

The Edit Invoke window should no longer display any warnings or errors.

3. Click **OK**.

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

Create an Assign Activity

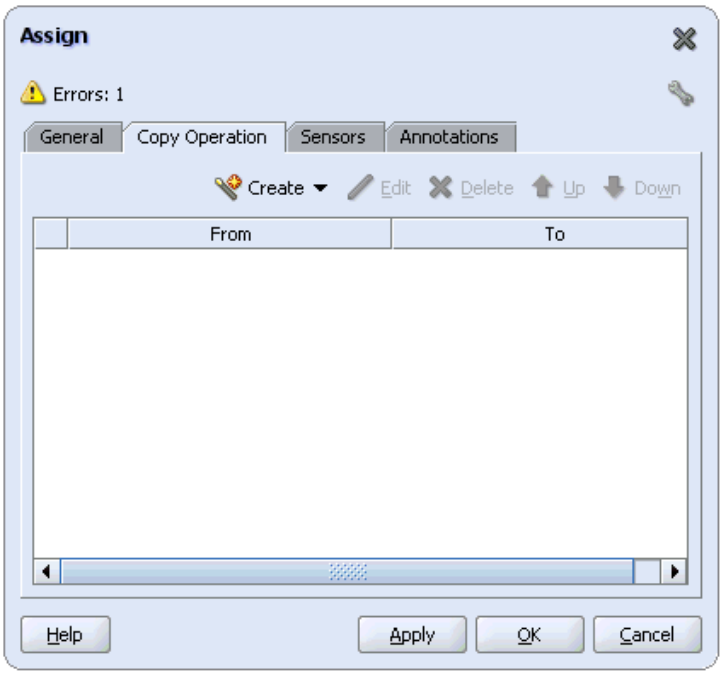
An Assign activity provides a method for simple data manipulation, such as copying the contents of one variable to another. This Assign activity maps the input variable of the Siebel process to the Siebel PartnerLink input.

To create an Assign activity:

1. From the **Process Activities** pane on the right, drag an **Assign** activity to the visual editor and place it between the Receive activity (`receiveInput`) and the new Invoke activity (`Invoke_1`).

2. Double-click the **Assign** activity icon.

   The Assign dialog box is displayed.



> **Note:** Ignore any invalid settings and error warnings.

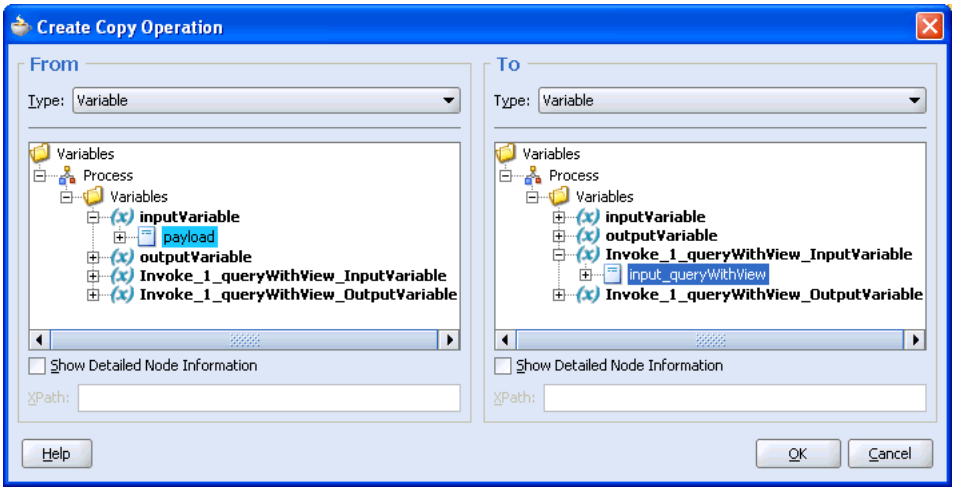
3. In the Copy Operation tab, click **Create** and select **Copy Operation**.

   The Create Copy Operation dialog box is displayed. Perform the following steps:

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

   b. In the **To** pane, expand **Variables**, then **Invoke_1_queryWithView_InputVariable**, and then highlight **input_queryWithView**.

   Your Create Copy Operation dialog box should look as follows:



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

Create a Second Assign Activity

This Assign activity maps the output variable of the Siebel process to the Siebel PartnerLink output.

To create a second Assign activity:

1. From the **Process Activities** pane on the right, drag another **Assign** activity to the visual editor and place it between the Invoke activity (`Invoke_1`) and the Reply activity (`replyOutput`).
2. Double-click the **Assign** activity icon.

   The Assign settings dialog box is displayed.

   Note: Ignore any invalid settings and error warnings.

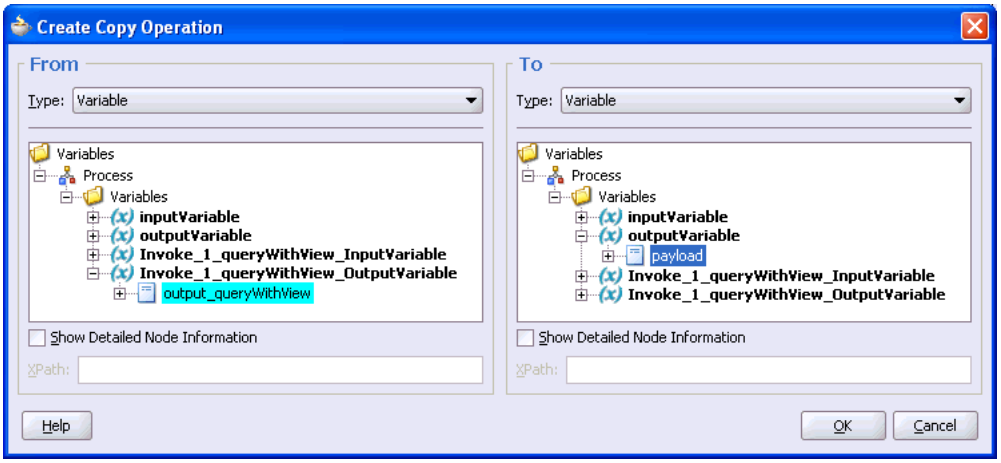
3. In the Copy Operation tab, click **Create** and select **Copy Operation**.

   The Create Copy Operation dialog box is displayed. Perform the following steps:

   a. In the **From** pane, expand **Variables**, then **Invoke_1_queryWithView_OutputVariable**, and then highlight **output_queryWithView**.

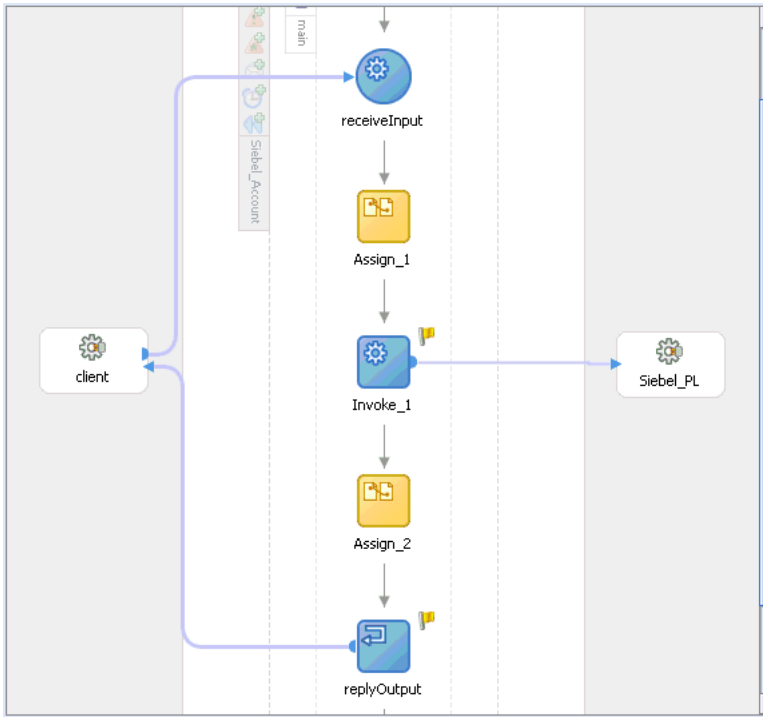
   b. In the **To** pane, expand **Variables**, then **outputVariable**, and then highlight **payload**.

   Your Create Copy Operation dialog box should look as follows:



4. To close the Create Copy Operation dialog box and the Assign dialog box, click **OK**.
5. Select **Save** from the **File** menu.

The following image shows the diagram view of your completed BPEL process.



See "Invoking Adapter Request-Response Service from Oracle BPEL Process Manager" on page 4-24 for information on how to deploy and manage your outbound process.

See Also:

- *Oracle BPEL Process Manager Developer's Guide*
- *Oracle Application Server Adapter Concepts*

Testing Outbound BPEL and ESB Processes

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

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

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

Design a BPEL Process for Event Handling (Inbound)

An inbound BPEL process consists of a PartnerLink and a Receive process activity. You must first create a channel and a new Oracle BPEL Process Manager Server connection. See Chapter 5, "BPEL Process Manager Integration Examples" for instructions on how to perform these procedures.

> **Note:** You must create a separate channel for every event and select that channel when you generate WSDL for inbound interaction using Application Explorer. Do not start the channel in Application Explorer, as Oracle BPEL Process Manager manages endpoint activation independently. See "Siebel Event Integration" on page 5-13 for more information.

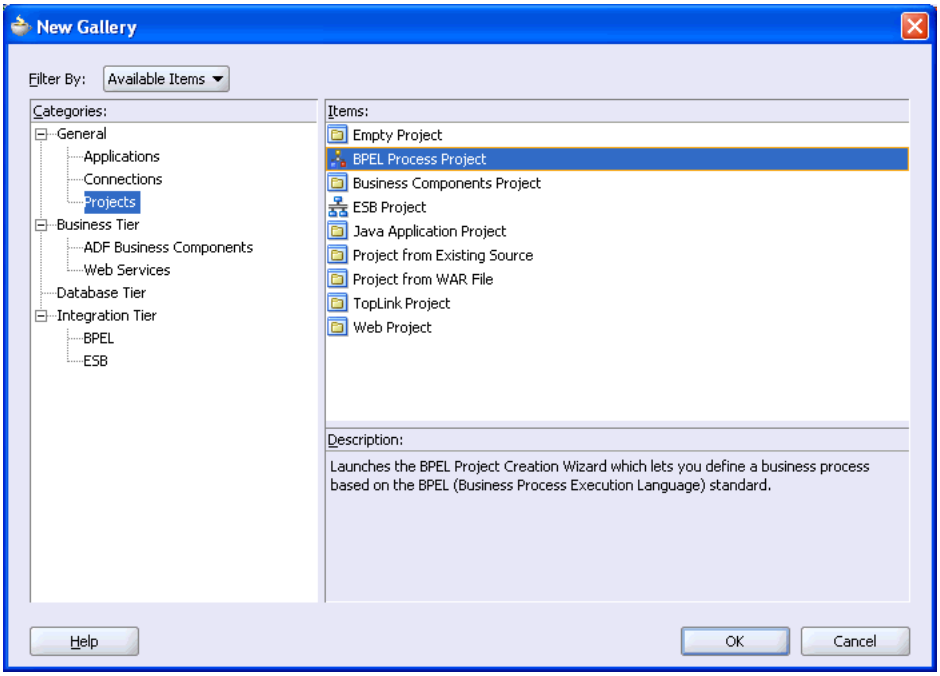
Create a New BPEL Project for Inbound Interaction (Empty Process)

Before you create a BPEL project, verify that your Oracle BPEL Server is running. After you have created a new server connection, you are ready to design an empty process template for your BPEL project.

To create a new BPEL project for inbound interaction:

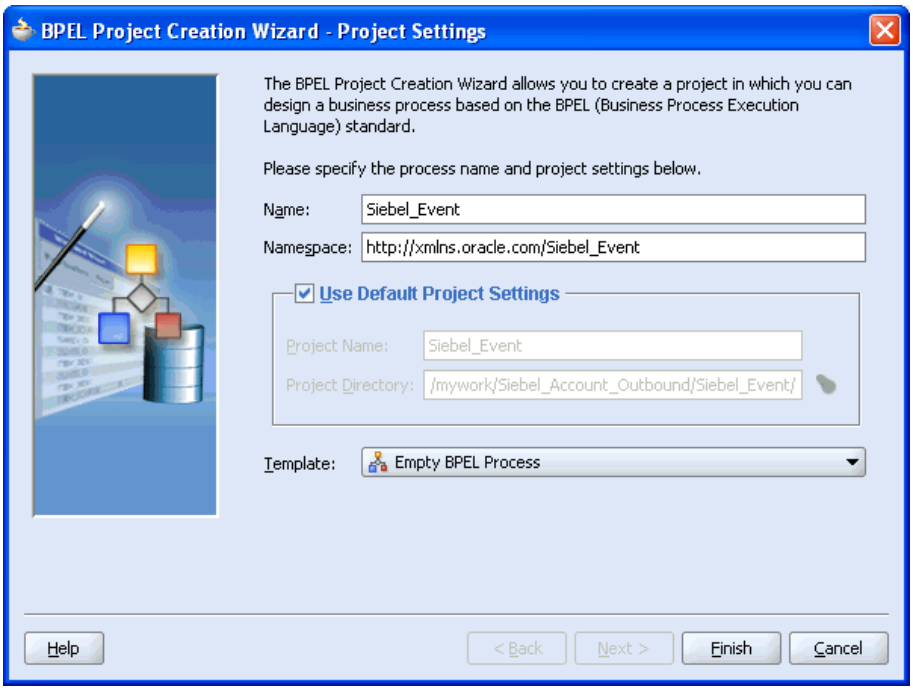
1. Click the **Applications Navigator** tab and select an application for your project.
2. Right-click the application and select **New Project**.

   The New Gallery window is displayed.



3. From the Items list, select **BPEL Process Project** and click **OK**.

The BPEL Project Creation Wizard - Project Settings dialog box is displayed.



Perform the following steps:

a. Specify a name for the process.

The Namespace field is updated automatically.

b. From the Template list, select **Empty BPEL Process**.

c. Click **Finish**.

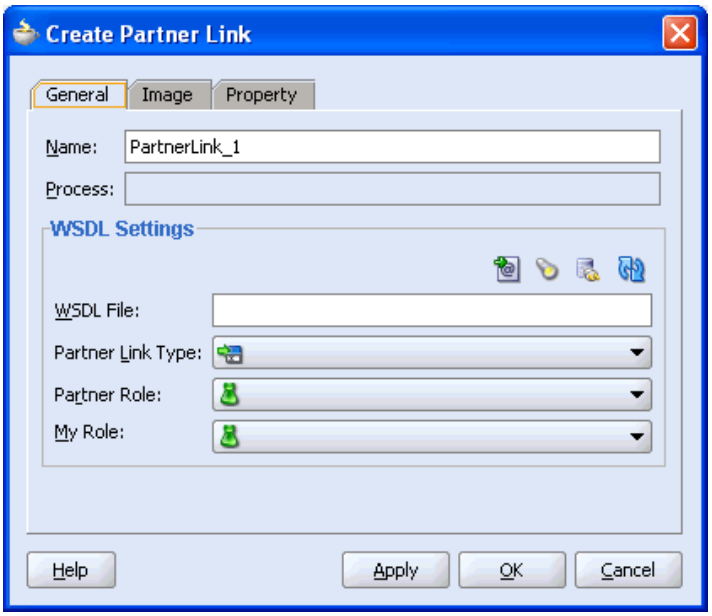
Create an Inbound PartnerLink Activity

When designing a BPEL process, a PartnerLink activity must be created to invoke the Siebel service. A PartnerLink describes a set of operations within a Web service. The WSDL document is the external contract to which the Web service conforms. Given a WSDL, any BPEL process can initiate a Web service through a PartnerLink.

To create an inbound PartnerLink using the WSDL file you generated in Application Explorer:

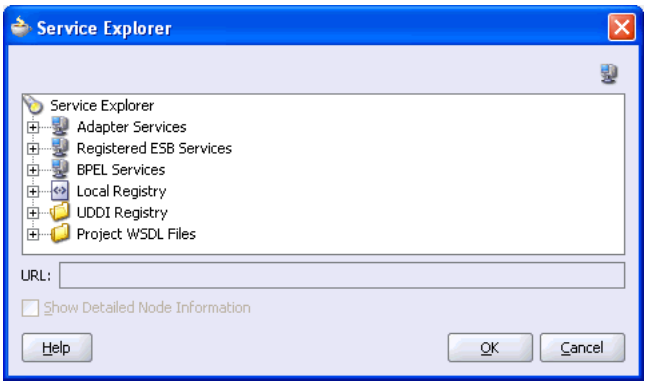
1. From the Services pane on the right, drag and drop a PartnerLink to the visual editor.

The Create Partner Link dialog box is displayed.



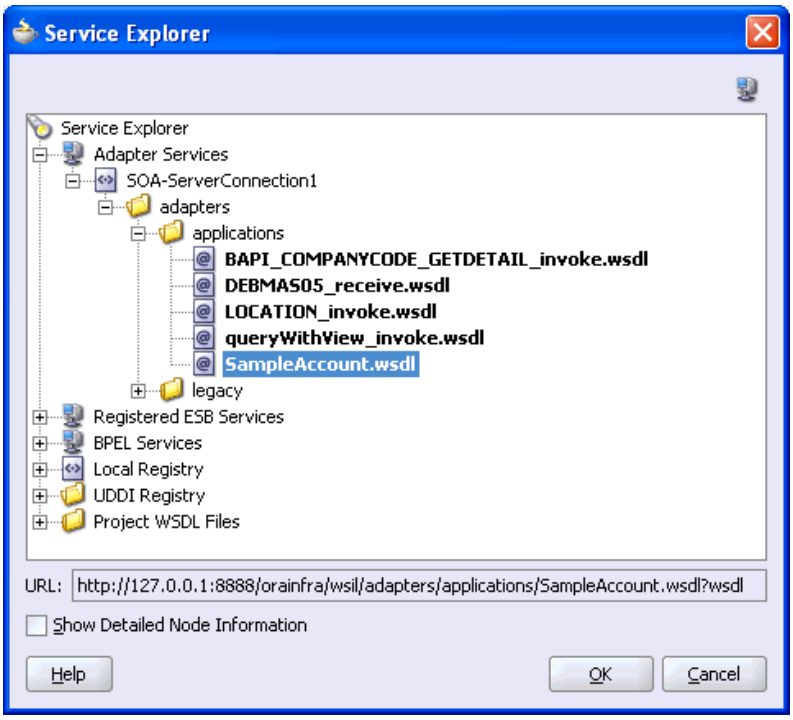
2. Click the **Service Explorer** icon (second icon from the left preceding the **WSDL File** field).

   The Service Explorer dialog box is displayed.



3. Expand your new connection, then expand **adapters**, and then **applications**.

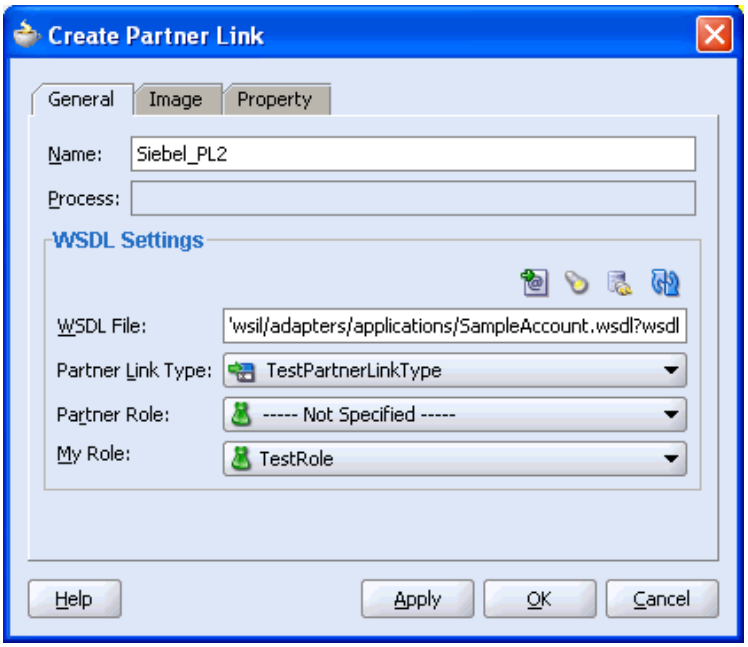
The WSDL tree displays the WSDL files you created using Application Explorer. The WSDL tree is generated by a WSDL servlet, which is automatically deployed as part of the Oracle BPEL Server installation.



Note: If you have organized your WSDL files in subfolders, the WSIL browser will display the full tree structure of your WSDL hierarchy. By default, the names of all WSDL files generated for inbound adapter services end with `_receive`.

4. Select **SampleAccount.wsdl** and click **OK**.

The Create Partner Link dialog box is displayed.



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

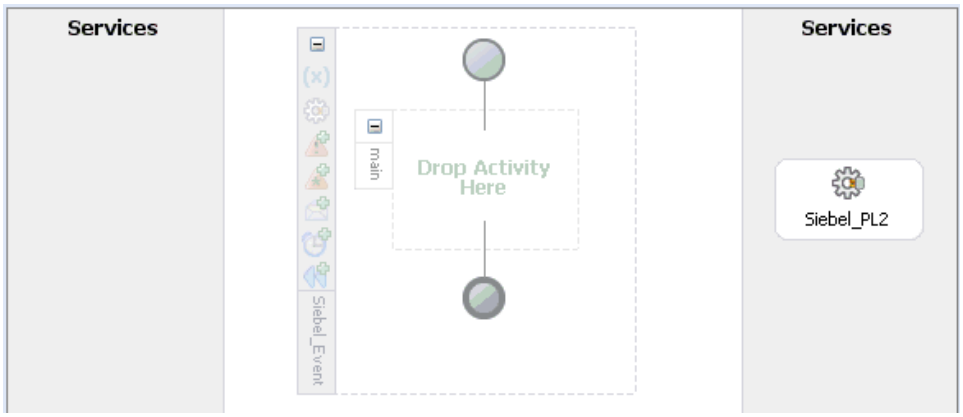
Perform the following steps:

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

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

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

The new Siebel PartnerLink appears in the visual editor.



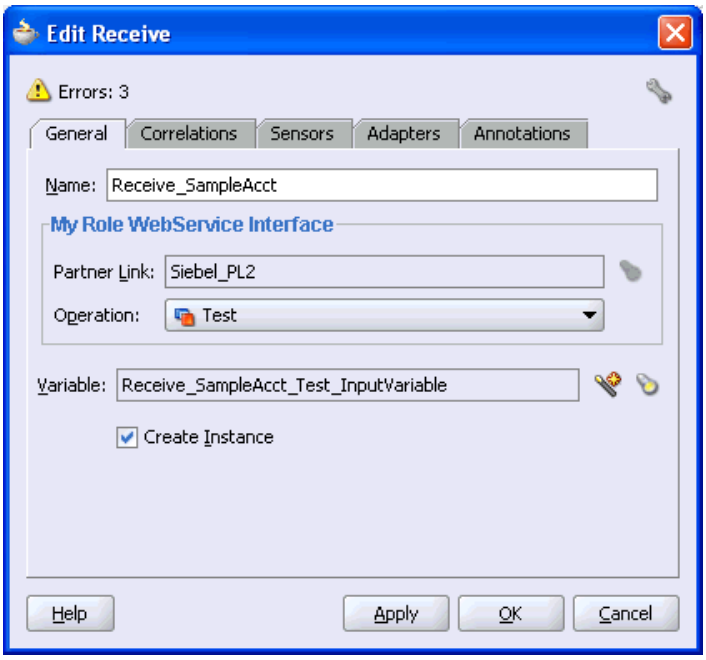
6. Select **Save** from the **File** menu.

Create an Inbound Receive Activity

To create an inbound Receive Activity:

1. From the **Process Activities** pane on the right, drag a **Receive** activity to the visual editor and place it in the designated placeholder labeled **Drop Activity Here**.
2. Connect the **Receive** activity to the **Siebel_PL2** PartnerLink.

The Edit Receive dialog box is displayed.



Perform the following steps:

a. Specify a name for the Receive Activity, for example, **Receive_SampleAcct**.

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

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

3. Click **Apply**.

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

4. Click **OK**.

   A connection is created between the PartnerLink and the Receive activity. You have completed the design of your inbound BPEL process.

See "Listening to Adapter Events Inside Oracle BPEL Process Manager" on page 4-27 for information on how to deploy and manage your inbound process.

See Also:

- *Oracle BPEL Process Manager Developer's Guide*
- *Oracle Application Server Adapter Concepts*

Invoking Adapter Request-Response Service from Oracle BPEL Process Manager

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

Deploy the Outbound BPEL Process

The procedures for deploying an inbound and an outbound BPEL process using the JDeveloper interface are identical.

To deploy your BPEL process in JDeveloper:

1. Right-click your project in the Applications Navigator tab.
2. Select **Deploy**, then ***Your BPEL PM Server connection***, and then **Deploy to default domain**.

   The deployment process starts automatically.
3. Observe the **Messages** log at the bottom of the window.

   The Messages log displays the deployment status. In this example, it shows a successful deployment message for the process.

   If deployment was not successful, click the **Compiler** tab to view all error and warning messages generated during the deployment process.

Manage the Deployed Outbound Process in Oracle BPEL Console

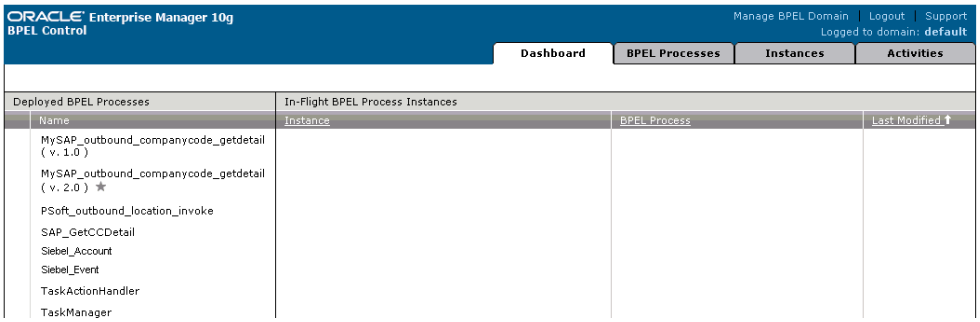
JDeveloper deploys the developed process directly to the Oracle BPEL Console, which enables you to run, monitor, and administer BPEL processes.

To invoke adapter request-response service:

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

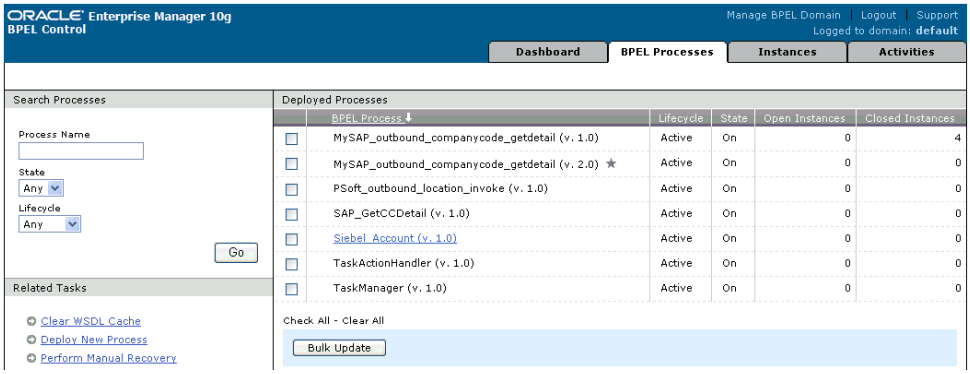
   `http://host:port/BPELConsole`
2. Select a domain and provide a valid password.

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



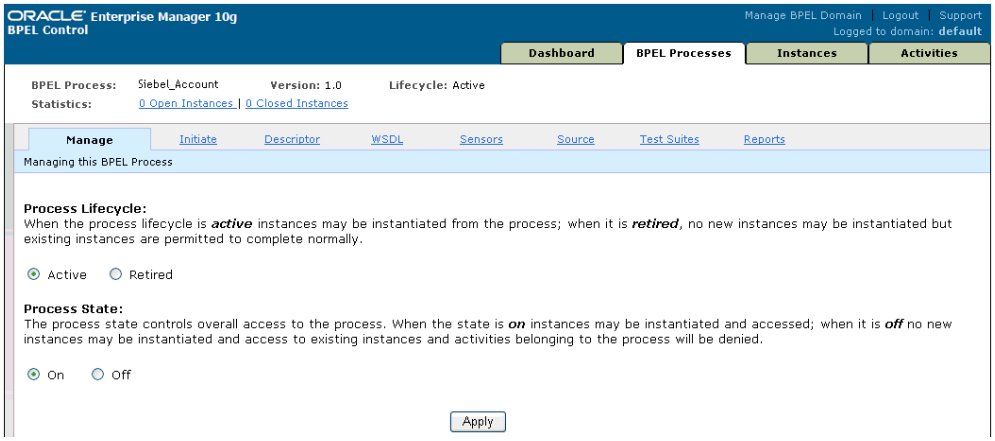
3. Click the **BPEL Processes** tab.

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



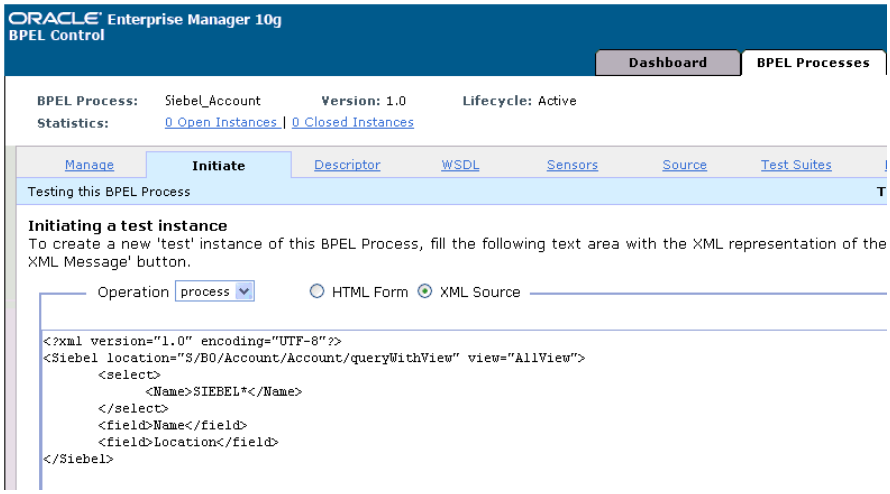
4. Click the **Siebel_Account** process link.

   The Manage window provides options for managing this BPEL process. Do not change any of the following default settings.



5. Click the **Initiate** tab.

   The Initiate tab enables you to test your BPEL process.



   Perform the following steps:

a. From the **Initiating a test instance** list, select **XML Source**.

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

```
<?xml version="1.0" encoding="UTF-8"?>
<Siebel  location="S/BO/Account/Account/queryWithView" view="AllView">
  <select>
    <Name>SIEBEL*</Name>
  </select>
  <field>Name</field>
  <field>Location</field>
</Siebel>
```

6. Click **Post XML Message**.

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

   See Also: *Oracle Application Server Adapter Concepts*

Listening to Adapter Events Inside Oracle BPEL Process Manager

The OracleAS Adapter for Siebel event notification service, also referred to as Inbound Interaction, is used to listen to events that occur in an EIS. The following section describes how to deploy your inbound BPEL process and listen to adapter events at runtime using Oracle BPEL Console.

Deploy the Inbound BPEL Process

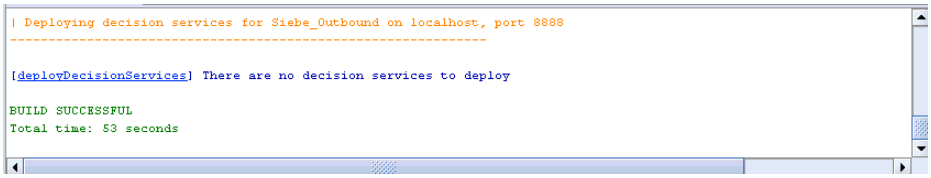
The procedures for deploying an inbound and an outbound BPEL process using the JDeveloper interface are identical.

To deploy your BPEL process in JDeveloper:

1. Right-click your process flow in the Applications pane.
2. Select **Deploy**, then ***Your BPEL PM Server connection***, and then **Deploy to default domain**.

   The deployment process starts automatically.
3. Observe the **Messages** log at the bottom of the window.

   The Messages log displays the deployment status. In this example, it shows a successful deployment message for the process.



   If deployment was not successful, click the **Compiler** tab to view all error and warning messages generated during the deployment process.

Listen to Adapter Events in Oracle BPEL Console

JDeveloper deploys the developed process directly to Oracle BPEL Console, which enables you to run, monitor, and administer BPEL processes, as well as to listen to adapter events at runtime using Oracle BPEL Console.

To listen to adapter events:

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

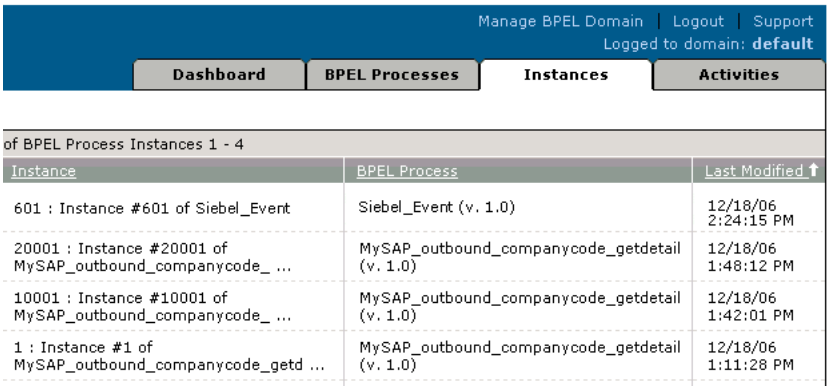
```
http://host:port/BPELConsole
```

2. Provide a valid user name and password.

   The Oracle BPEL Console Dashboard tab is displayed.

3. Click the **Instances** tab.

   Upon receiving a runtime event, an instance of the event is displayed under the Instances tab.



4. To see the event message, click the instance, and then click the **Audit** tab.

   The message received from the Siebel system is displayed.

5. Click **More...** to view the entire message, or **View Raw XML** to view the XML source

   See Chapter 5, "BPEL Process Manager Integration Examples" on page 5-1 for more information.

See Also: *Oracle Application Server Adapter Concepts*

5

BPEL Process Manager Integration Examples

This chapter contains examples of service and event integration with Siebel.

- Creating an Integration Object (IO) Node for Siebel
- Siebel Service Integration
- Siebel Event Integration

The scenarios shown in this chapter require the following prerequisites.

Prerequisites

- OracleAS Adapter for Siebel must be installed on Oracle Application Server.
- Oracle BPEL PM Server must be properly configured and running.
- Oracle JDeveloper must be properly installed.

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

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

- How to create a J2CA configuration, as BPEL PM is only compatible with the J2CA Connector. See "Creating a Configuration for J2CA" on page 2-9 for more information.
- How to configure OracleAS Adapter for Siebel for services and events using Application Explorer. See Chapter 2, "Configuring Oracle Application Server Adapter for Siebel" for more information.
- How to use Siebel workflows. See Appendix A, "Using Siebel Workflows" for information on Siebel design requirements.

> **See Also:** *Oracle BPEL Process Manager Developer's Guide*

Adapter integration with Oracle BPEL Process Manager is a two-step process:

1. **Design Time:** OracleAS 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 using JDeveloper.
2. **Runtime:** After you deploy the BPEL process you designed in JDeveloper, you can test your service configuration or see newly received events in the BPEL Console.

Creating an Integration Object (IO) Node for Siebel

The following example describes how to add an IO node for Siebel.

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.

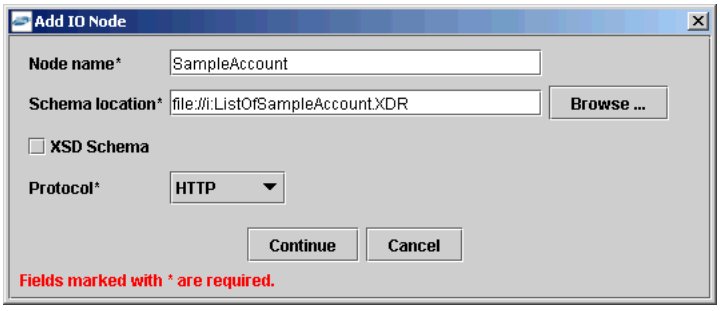
   The Connection dialog box displays the values you entered.

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 following dialog box is displayed.



7. Enter a node name, for example SampleAccount in the **Node name** field and a path to the Sample Account XDR file in the **Schema location** field.

   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.

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 from the **Protocol** list.
10. Click **Continue**.

Siebel Service Integration

This topic illustrates Siebel service integration. It describes design-time and runtime configuration.

Design-Time Configuration

Before you design a process for Siebel service integration, you must create an outbound J2CA service (WSDL) using Application Explorer.

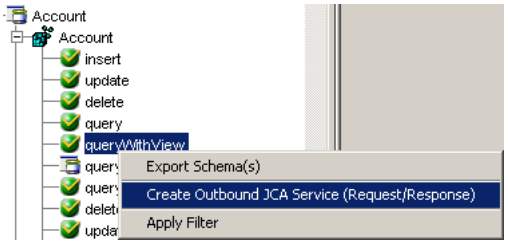
Creating a Request-Response J2CA Service in Application Explorer

To generate WSDL in Application Explorer:

1. Start **Application Explorer** and connect to a defined Siebel target or create a new target.

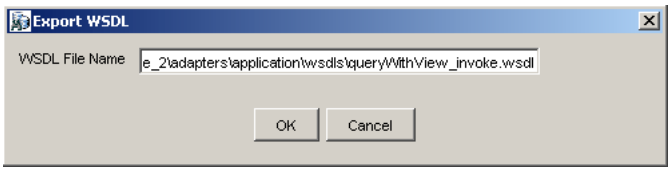
   See Connecting to a Defined Target on page 2-12 for more information.
2. Expand the Siebel target to which you are connected.
3. Expand **Business Object**, **Account**, **Account**. Navigate to **queryWithView** and right-click the object.

   The following menu is displayed.



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

   The Export WSDL dialog box is displayed.



5. Accept the default name and location for the file.

   The **.wsdl** file extension is added automatically.
6. Click **OK**.

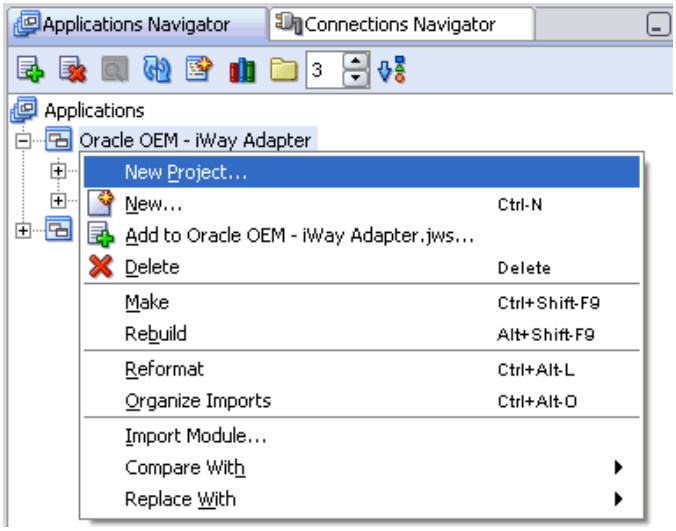
Creating a BPEL PM Server Connection

Before you design an outbound BPEL process, you must configure a new Application Server and Integration Server connection in Oracle JDeveloper. For more information, see: Chapter 4, "Integration with Oracle BPEL Process Manager".

Creating a BPEL Project for a Synchronous BPEL Process

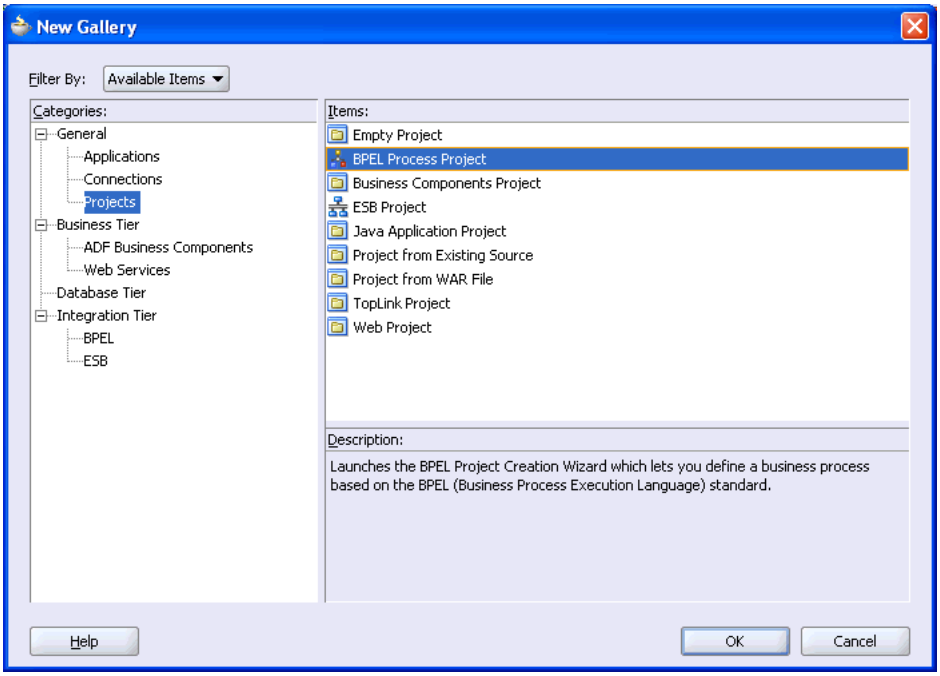
To create a BPEL Project for a synchronous BPEL process:

1. At the top of the upper left pane, click the **Applications Navigator** tab and select an application.



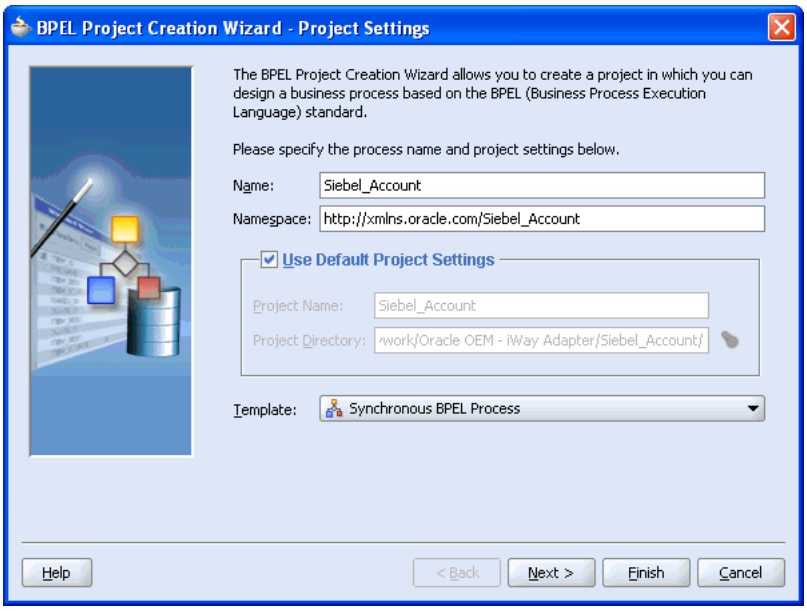
2. Right-click the application and select **New Project**.

   The New Gallery window is displayed.



3. From the Items list, select **BPEL Process Project** and click **OK**.

The BPEL Project Creation Wizard is displayed.



4. Perform the following steps:
   a. Specify a name for the BPEL process.

      The Namespace field is updated automatically.

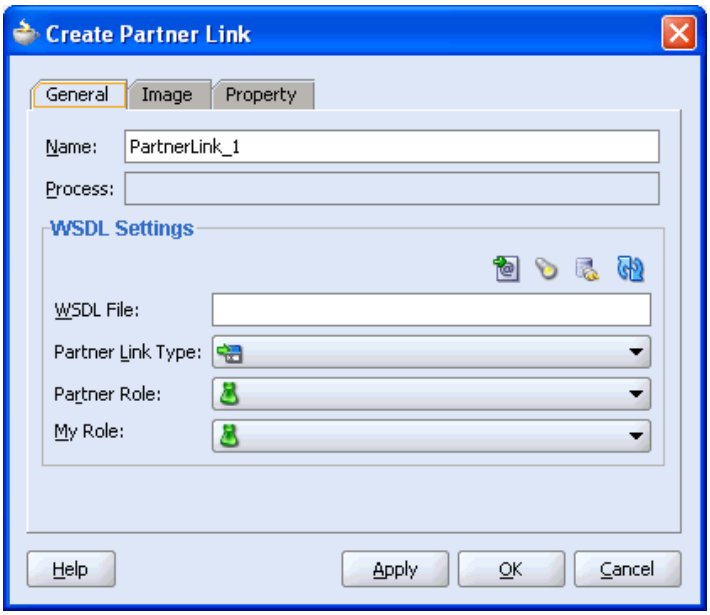
   b. From the Template list, select **Synchronous BPEL Process**.
5. Click **OK**.

Designing the BPEL Process for the queryWithView Outbound Service

To design the BPEL Process:

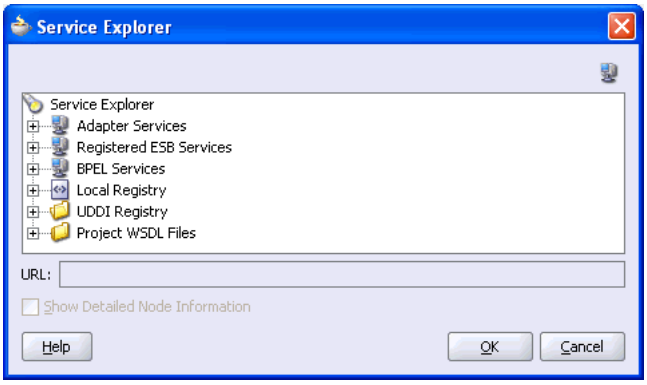
1. From the Services pane on the right, drag and drop a PartnerLink to the visual editor.

The Create Partner Link dialog box is displayed.



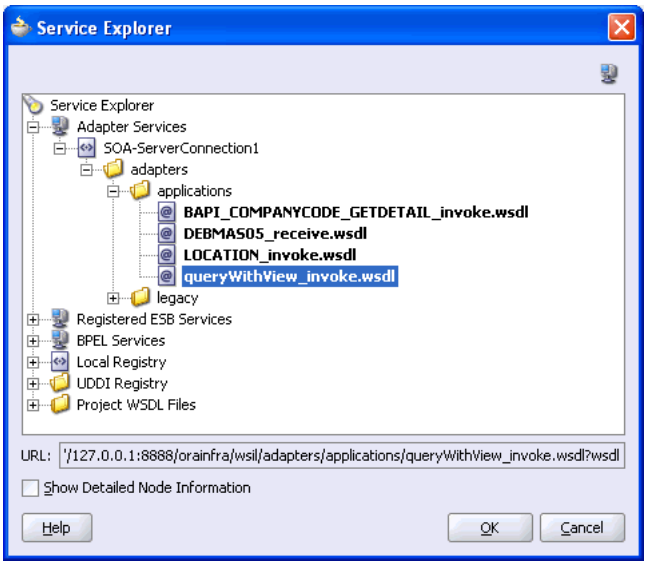
2. Click the **Service Explorer** icon (second icon from the left preceding the **WSDL File** field).

   The Service Explorer dialog box is displayed.



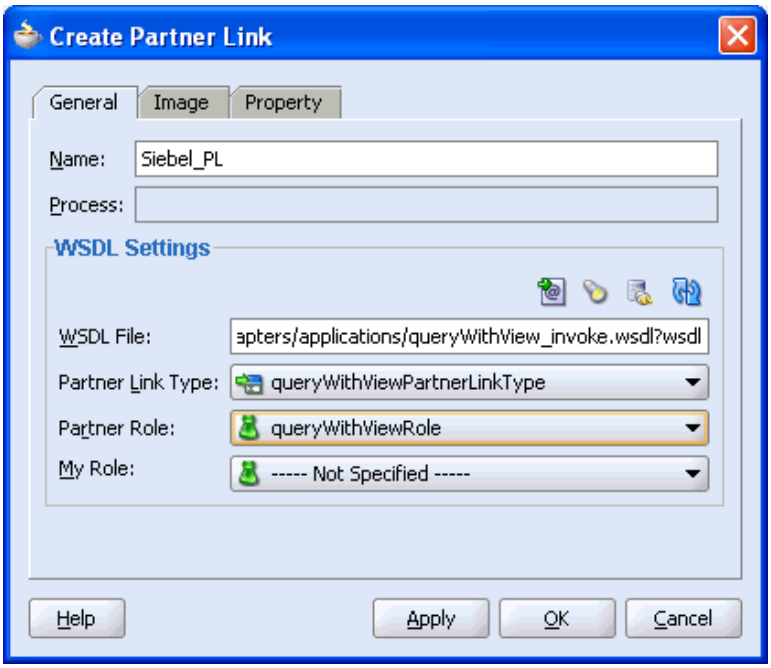
3. Expand your new connection under Adapter Services, followed by **adapters**, and then **applications**.

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



4. Select **queryWithView_invoke.wsdl** and click **OK**.

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



   Perform the following steps:

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

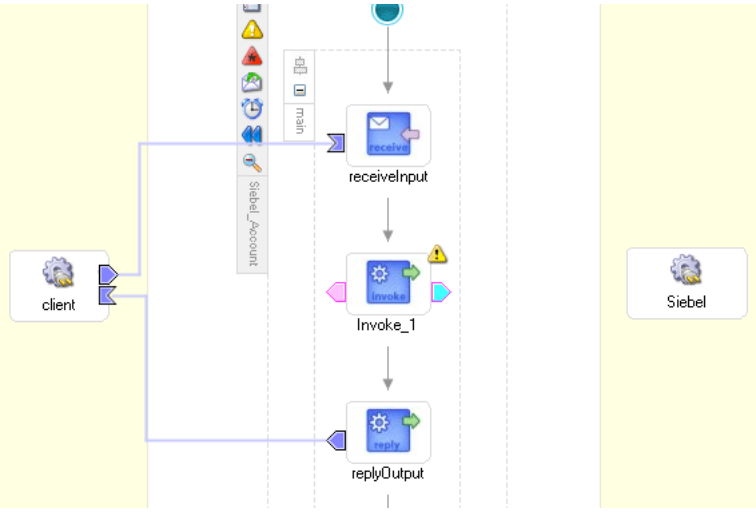
   b. From the **Partner Role** list, select the default value **queryWithViewRole**. This is the role of the BPEL process.

5. Click **OK**.

The new PartnerLink appears in the visual editor.

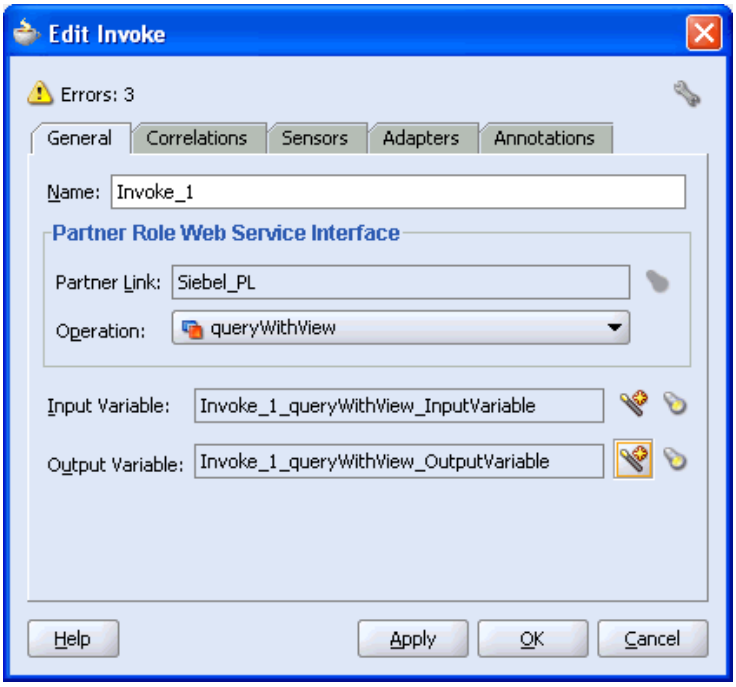
6. Select **Save** from the File menu.
7. From the **Process Activities** pane on the right, drag an **Invoke** activity to the visual editor and place it between the Receive activity (`receiveInput`) and the Reply activity (`replyOutput`).

   The Invoke process activity is shown in the following diagram view.



8. Drag the right arrow from **Invoke_1** and connect it to the Siebel PartnerLink.

   The Edit Invoke dialog box is displayed.



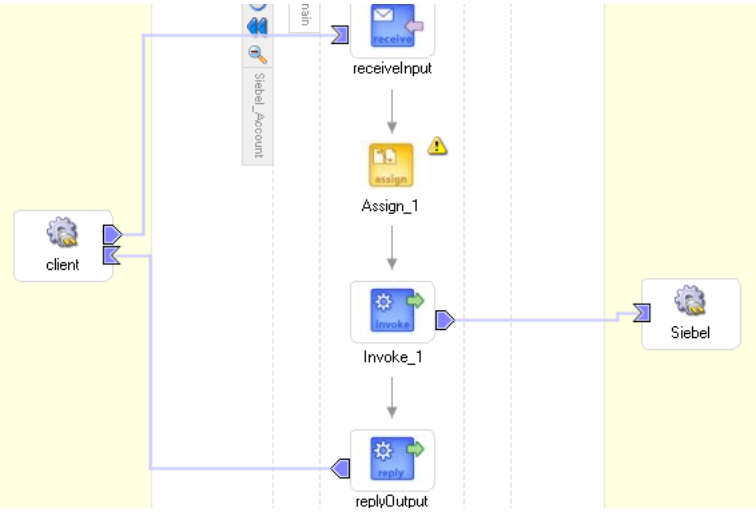
Perform the following steps:

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

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

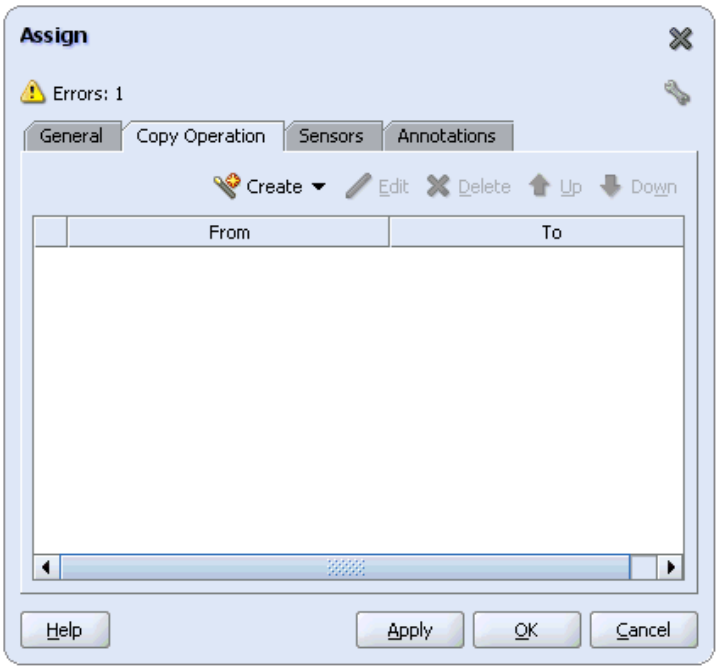
9. Click **OK**.
10. Drag an **Assign** process activity and drop it between the `receiveInput` Receive activity and `Invoke_1` Invoke activity.

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



11. Double-click the **Assign** activity icon.

    The Assign dialog box is displayed.



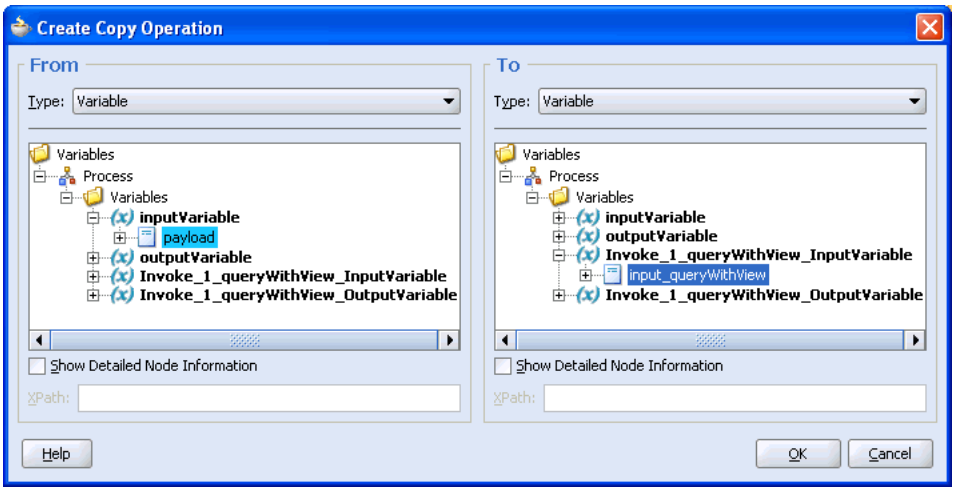
12. In the Copy Operation tab, click **Create**.

    The Create Copy Operation dialog box is displayed. Perform the following steps:

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

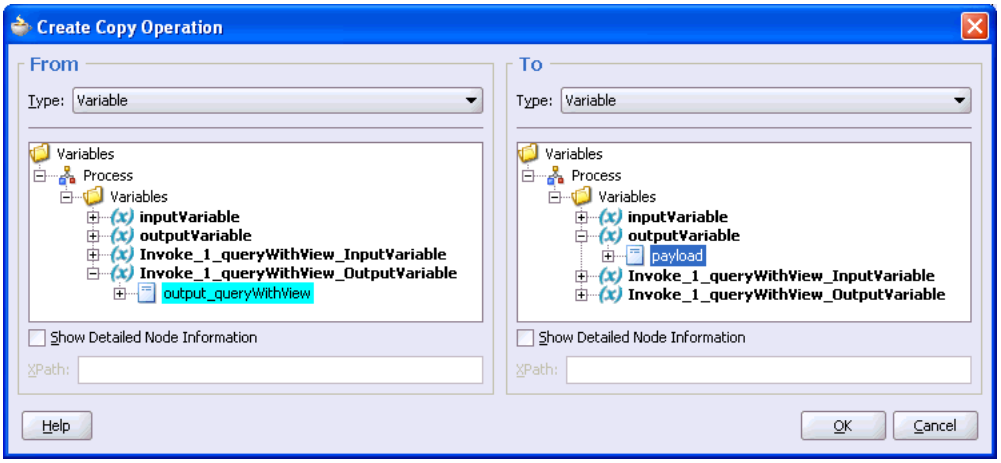
    b. In the **To** pane, expand **Variables**, then **Invoke_1_queryWithView_InputVariable**, and then highlight **input_queryWithView**.

Your Create Copy Operation dialog box should look as follows:



13. To close the Create Copy Operation dialog box and the Assign dialog box, click **OK**.
14. From the **Process Activities** pane on the right, drag another **Assign** activity to the visual editor and place it between the Invoke activity (`Invoke_1`) and the Reply activity (`replyOutput`).
15. Double-click the **Assign** activity icon and click **Create**.
16. In the Create Copy Operation dialog box, map **Invoke_1_queryWithView_OutputVariable**, **output_queryWithView** to **outputVariable**, **payload**.

    Verify that you have mapped all variables as follows:



17. Click **OK**, then click **OK** again.
18. Select **Save** from the File menu.

You have completed the design of your BPEL process.

Deploying the BPEL Process for the queryWithView Outbound Service

JDeveloper deploys BPEL processes directly to Oracle BPEL Console.

To deploy your BPEL process in JDeveloper:

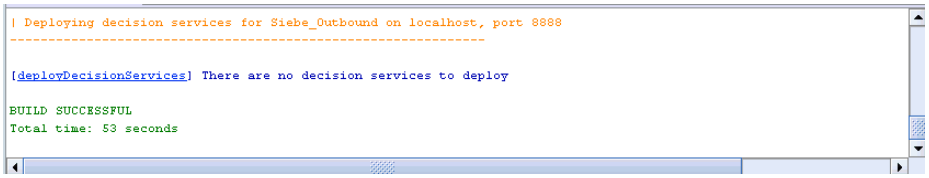
1. Right-click your project in the Applications Navigator tab.

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

   The deployment process starts automatically.

3. Observe the **Messages** log at the bottom of the window.

   The Messages log displays the deployment status. In this example, it shows a successful deployment message for the process.



   If deployment was not successful, click the **Compiler** tab to view all error and warning messages generated during the deployment process.

Runtime Configuration

To invoke the queryWithView process from Oracle BPEL Console:

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

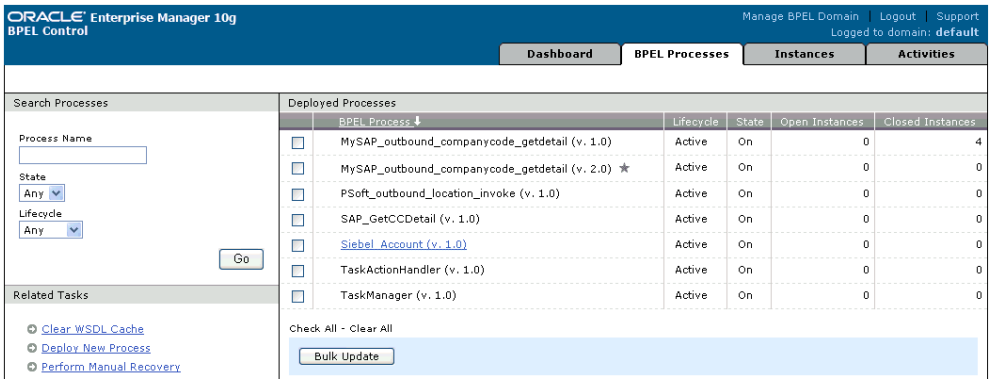
   `http://host:port/BPELConsole`

2. Provide a valid user name and password.

   The Oracle BPEL Console main page is displayed.

3. Click the **BPEL Processes** tab.

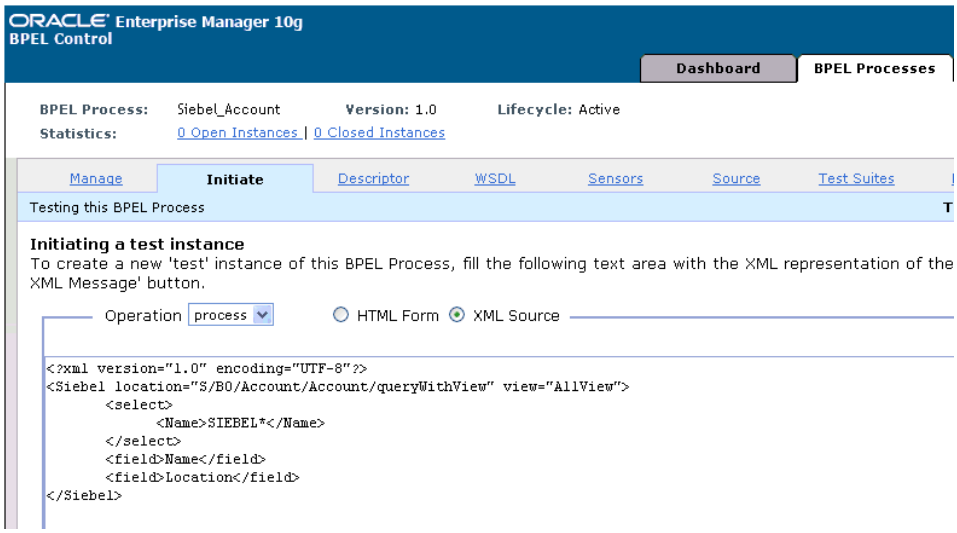
   The BPEL Processes tab is displayed.



4. Click the **Siebel_Account** process link.

5. Click the **Initiate** tab.

The Initiate tab enables you to test your BPEL process.



Perform the following steps:

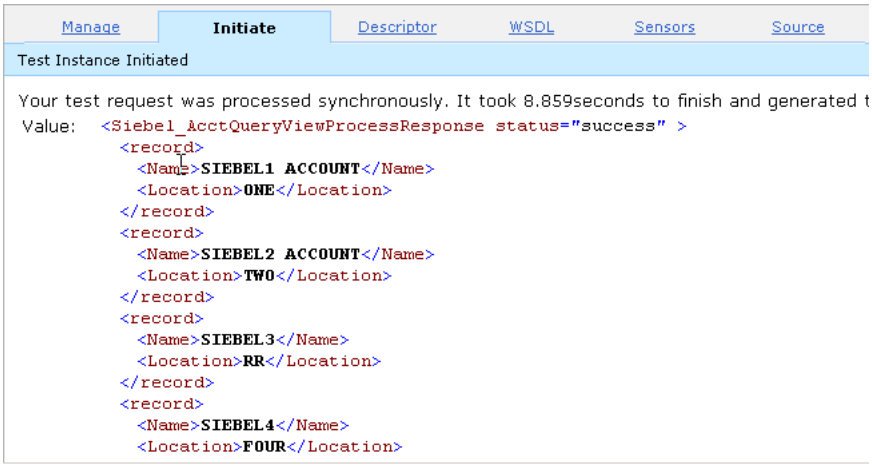
a. From the **Initiating a test instance** list, select **XML Source**.

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

```
<?xml version="1.0" encoding="UTF-8"?>
<Siebel  location="S/BO/Account/Account/queryWithView" view="AllView">
  <select>
    <Name>SIEBEL*</Name>
  </select>
  <field>Name</field>
  <field>Location</field>
</Siebel>
```

6. Click **Post XML Message**.

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



Siebel Event Integration

This topic illustrates Siebel event integration. It describes design-time and runtime configuration.

Design-Time Configuration

Creating a Channel

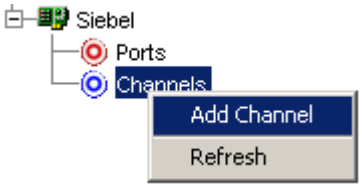
You must create a separate channel for every inbound J2CA service and select that channel when you generate WSDL for inbound interaction using Application Explorer.

> **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:

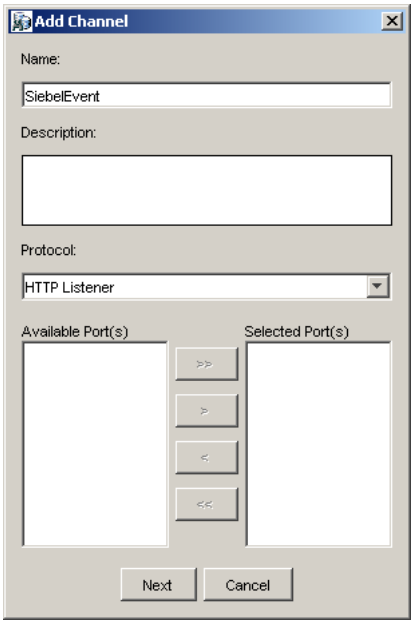
1. In the left pane, click **Events**.
2. Expand the **Siebel** node.

   The ports and channels nodes appear in the left pane.



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

   The Add Channel dialog box is displayed.



Perform the following steps:

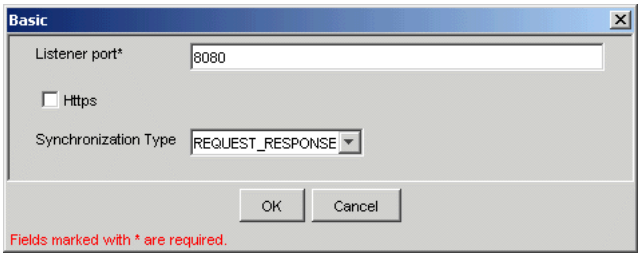
a. Enter a name for the channel, for example, `SiebelEvent`.

b. Enter a brief description.

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

4. Click **Next**.

   The Basic dialog box is displayed.



5. Enter a port number in the **Listener port** field.

6. Leave the default synchronization type.

7. Click **OK**.

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

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

Generating WSDL for Event Notification

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

You must be connected to a Siebel target under the Adapters node in Application Explorer. See "Establishing a Connection (Target) for Siebel" on page 2-10 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. Expand the **Integration Object** node under the target and scroll down to **SampleAccount**.

2. Right-click **SampleAccount**.

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

   The Export WSDL dialog box is displayed.

   Perform the following steps:

   a. In the **WSDL File Name** field, specify a name and location of the WSDL file.

   b. In the **Channel** field, select the channel you created for this inbound service.

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

c. If you are using the optional port feature, you must select a port from the **Port** list.

If you are not using event ports for schema validation, skip this step. In this case, the default value of **no port** is selected automatically.

4. Click **OK**.

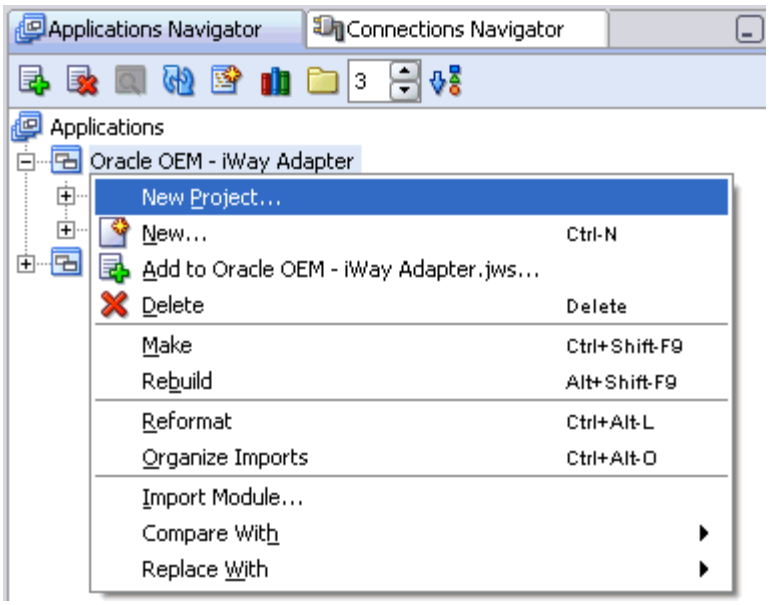
Creating a BPEL PM Server Connection

Before you design a BPEL process using the WSDL you generated in Application Explorer, you must configure a new Application Server and Integration Server connection in Oracle JDeveloper. For more information, see Chapter 4, "Integration with Oracle BPEL Process Manager".

Designing the BPEL Process for the Inbound Service

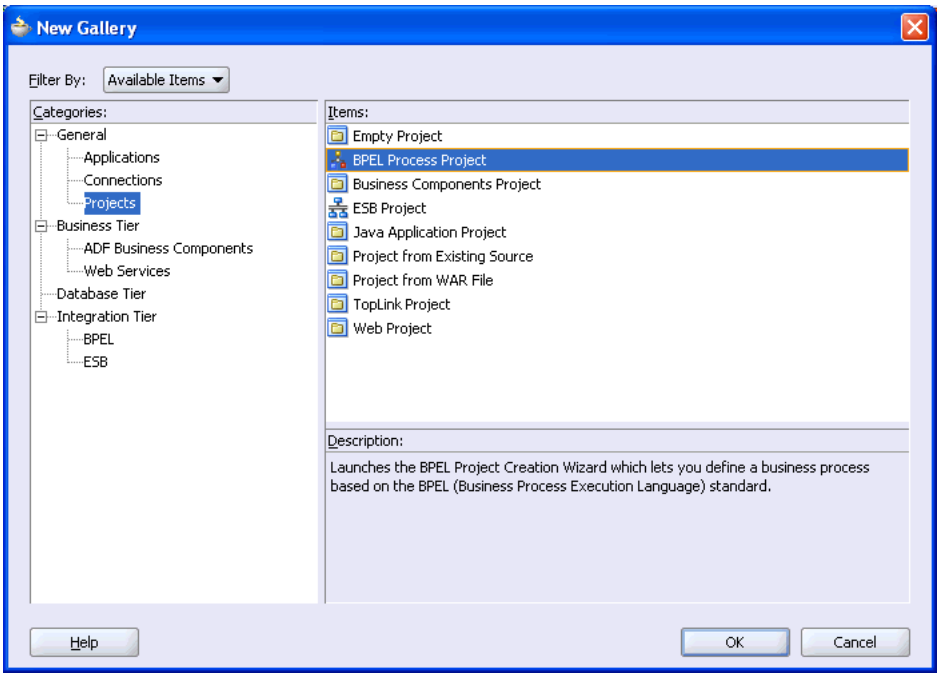
To design a BPEL process for inbound interaction:

1. At the top of the upper left pane, click the **Applications Navigator** tab and select an application.



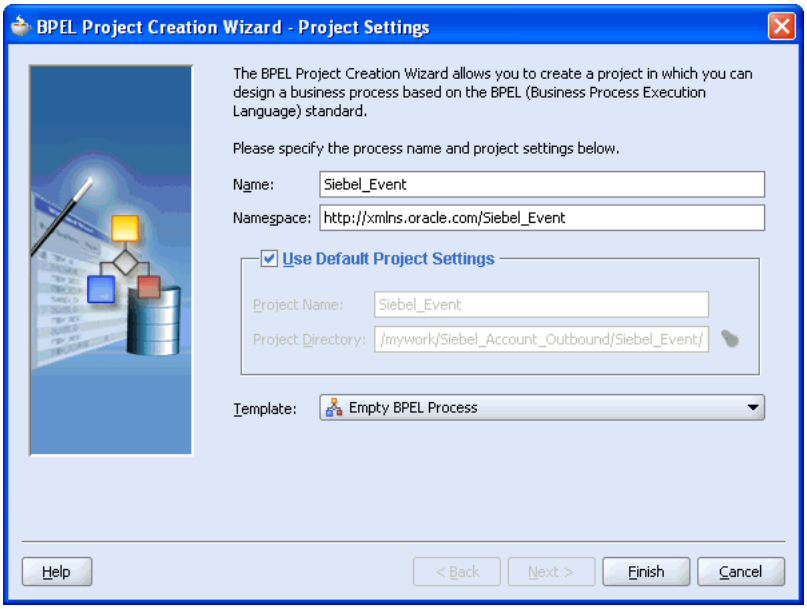
2. Right-click the application and select **New Project**.

The New Gallery dialog box is displayed.



3. From the Items list, select **BPEL Process Project** and click **OK**.

   The BPEL Project Creation Wizard is displayed.



4. Perform the following steps:

   a. Specify a name for the process, for example, **Siebel_Event**.

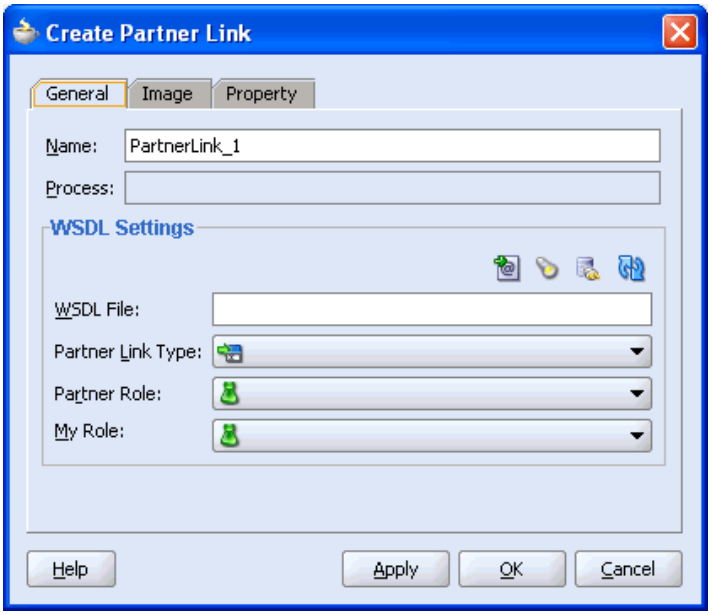
      The Namespace field is updated automatically.

   b. From the Template list, select **Empty BPEL Process**.

   c. Click **OK**.

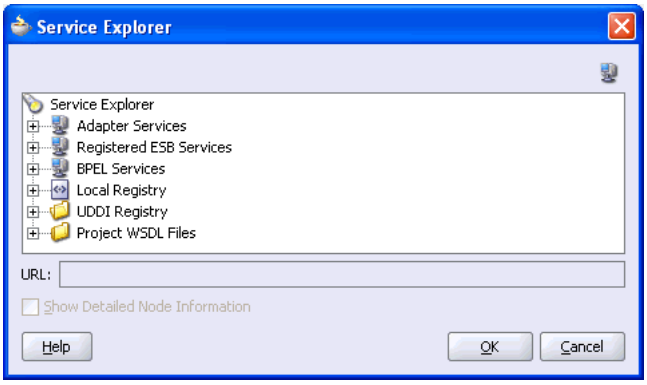
5. From the Services pane on the right, drag and drop a **PartnerLink** to the visual editor.

   The Create Partner Link dialog box is displayed.



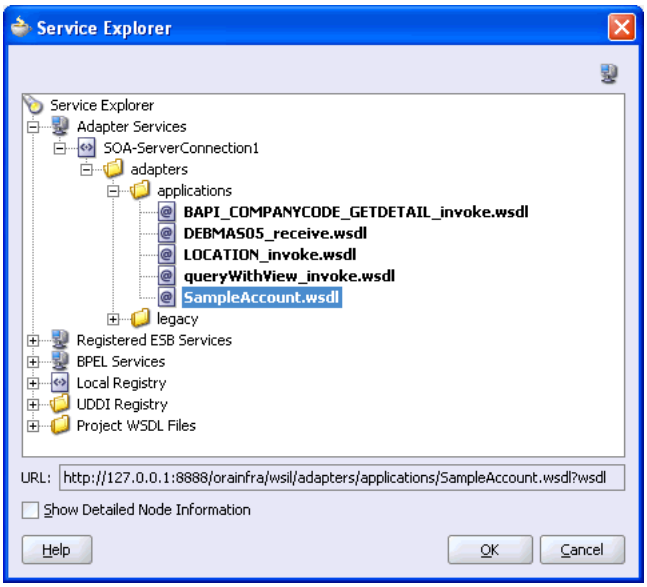
6. Click the **Service Explorer** icon (second icon from the left preceding the **WSDL File** field).

   The Service Explorer dialog box is displayed.



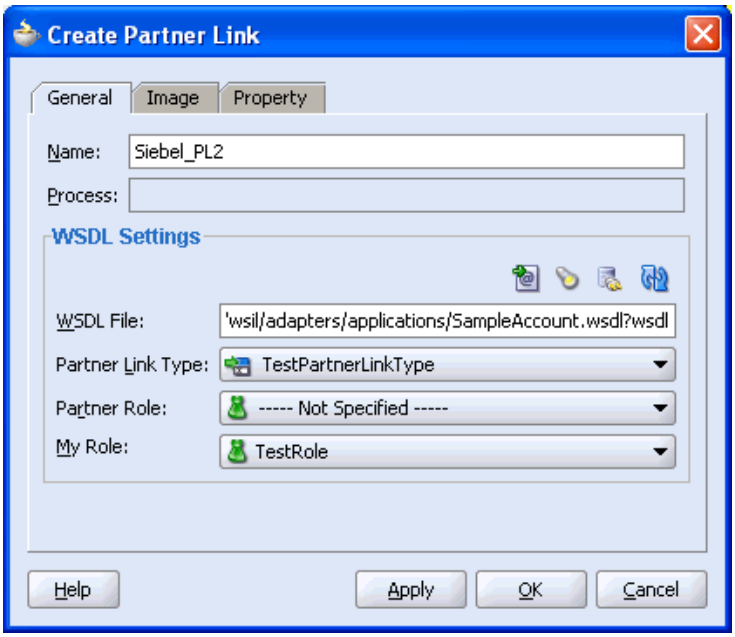
7. Expand your new connection under Adapter Services, followed by **adapters**, and then **applications**.

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



8. Select **SampleAccount.wsdl** and click **OK**.

   The Create Partner Link dialog box is displayed.



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

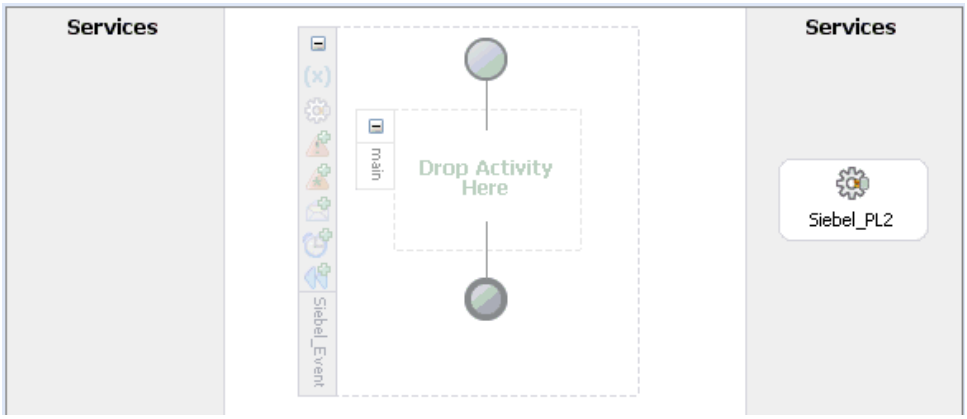
   Perform the following steps:

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

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

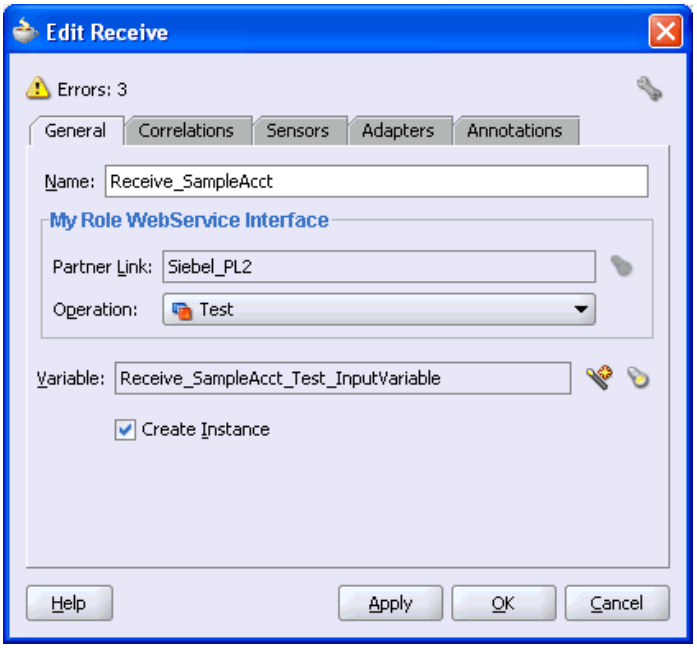
9. Click **Apply**, and then **OK**.

The new Siebel_PL PartnerLink appears in the visual editor.



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

    The Edit Receive dialog box is displayed.

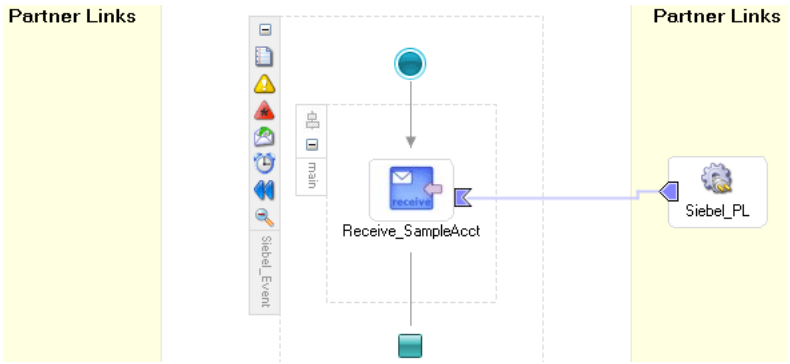


    Perform the following steps:

    a. Specify a name for the Receive Activity, for example, **Receive_SampleAcct**.
    b. Click the first icon to the right of the **Variable** field, then click **OK** in the Create Variable dialog box that is displayed.
    c. Verify that the **Create Instance** check box is selected.
12. Click **Apply**.

    The Edit Receive dialog box should no longer display any warnings or errors.
13. Click **OK**.
14. Select **Save** from the **File** menu.

The completed inbound BPEL process looks as follows:



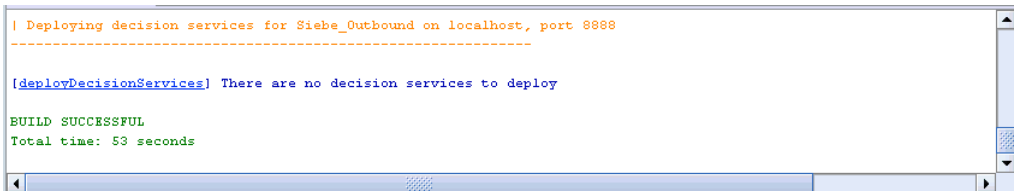
Deploying the BPEL Process for the Inbound Service

Perform the following steps:

1. Right-click your process flow in the Applications - Navigator pane.
2. Select **Deploy**, then ***Your BPEL PM Server connection***, and then **Deploy to default domain**.
3. When prompted, enter your BPEL Process Manager password and click **OK**.

   The deployment process starts automatically after you enter the correct password.
4. Observe the **Messages** tab at the bottom of the JDeveloper screen.

   The following image shows successful deployment.



Runtime Configuration

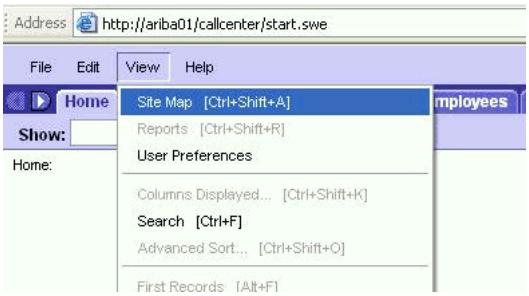
The following topic describes how to trigger an event in Siebel and verify event integration using OracleAS Adapter for Siebel.

Triggering a Siebel Event to Test Event Runtime Integration

To trigger an event in Siebel:

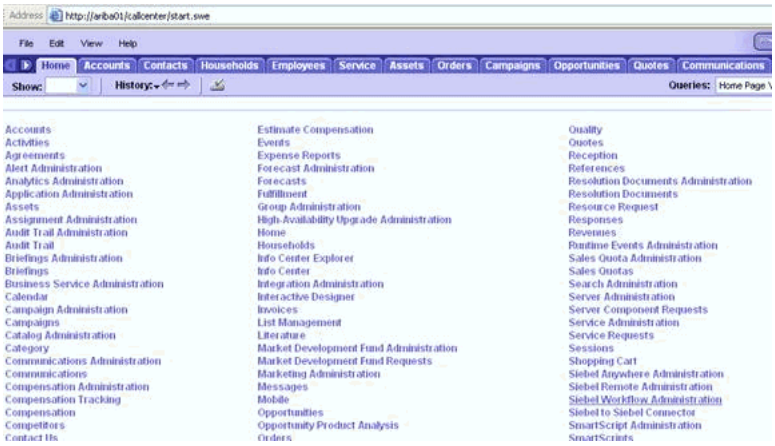
1. Start the Siebel Call Center by entering the following URL in a browser:

   `http://hostname/callcenter/start.swe`



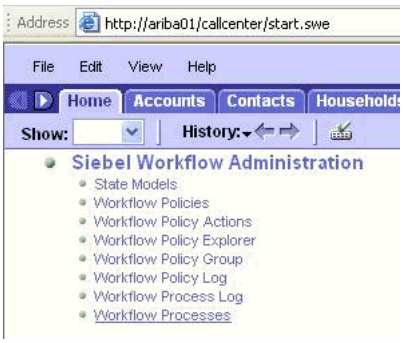
2. Click **View** and select **Site Map** from the list.

   The Site Map view is displayed.



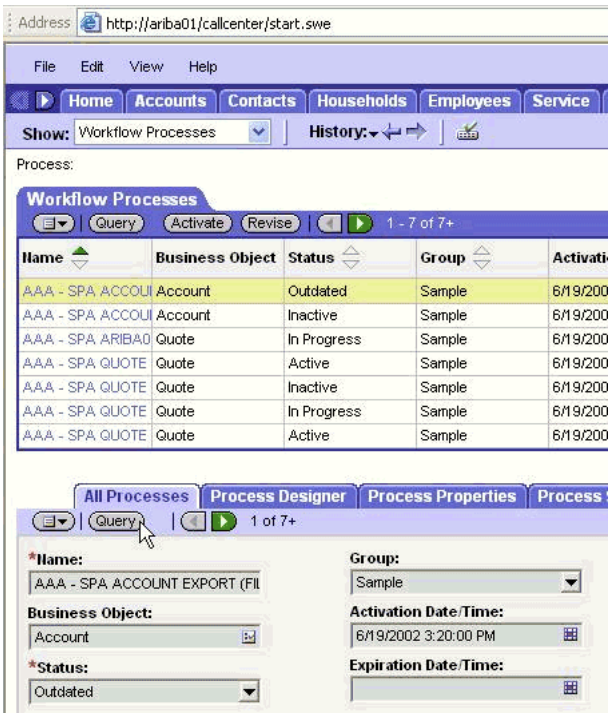
3. Click **Siebel Workflow Administration**.

   The Siebel Workflow Administration page is displayed.

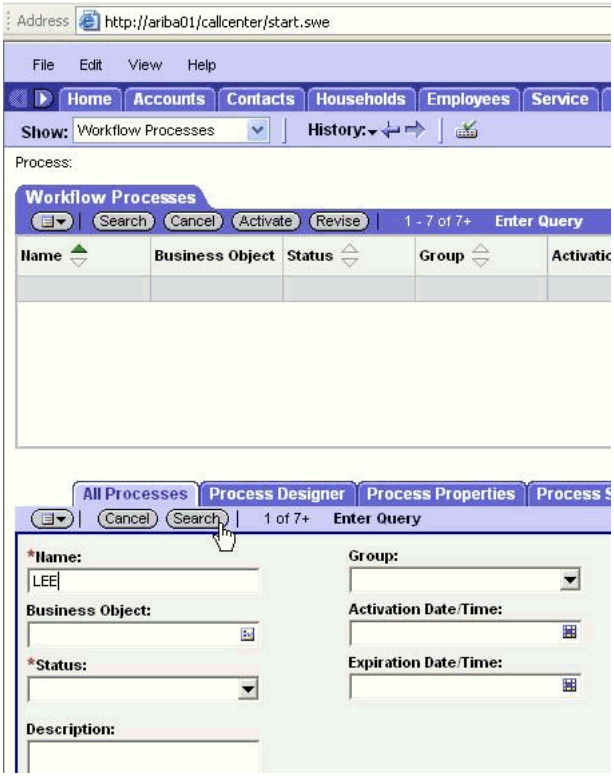


4. Click **Workflow Processes**.

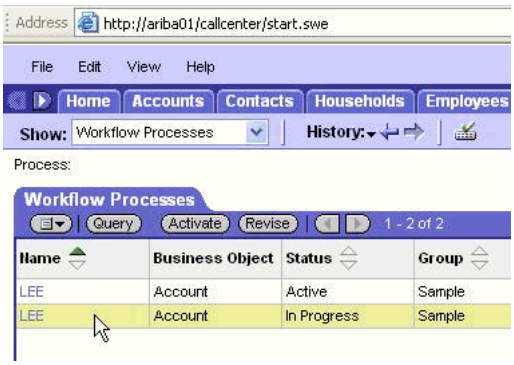
The Workflow Processes page is displayed.



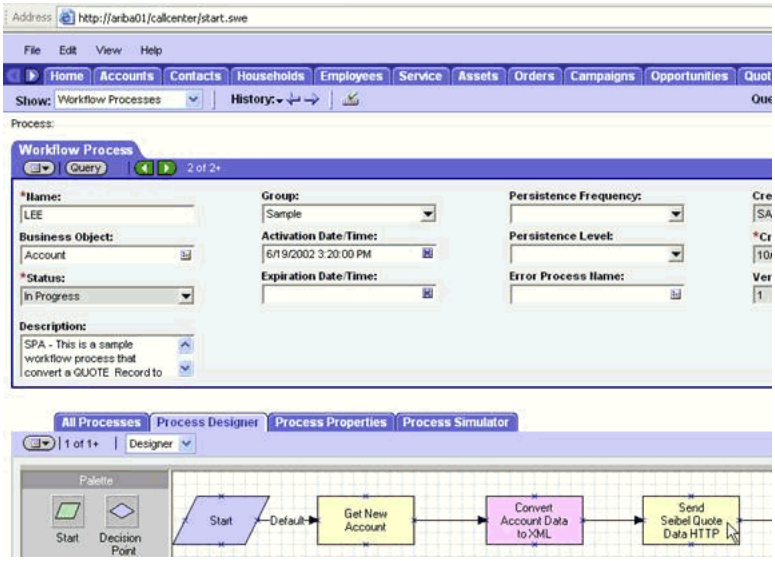
5. Click **Query** to search for the Workflow needed to trigger a Siebel event.



6. Enter a Siebel workflow name and click **Search**.

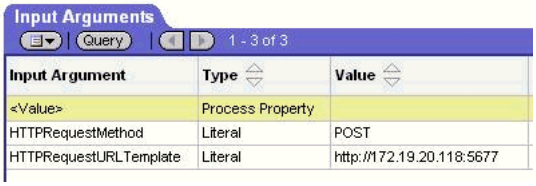


7. Select the workflow.



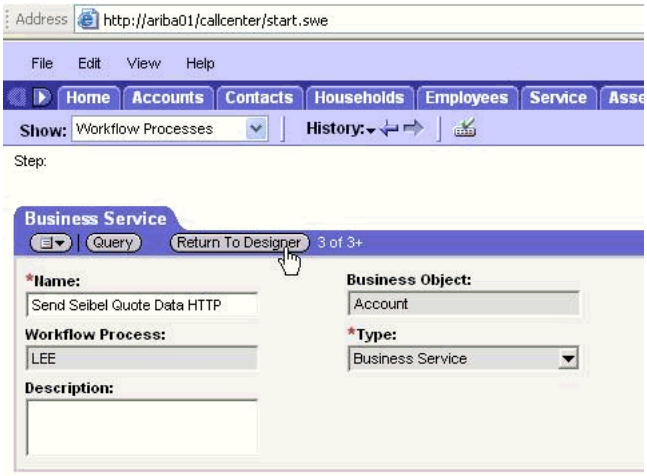
8. Click the **Process Designer** tab and double-click the **Send Siebel Quote Data HTTP** workflow element.

   The Input Arguments tab is displayed.

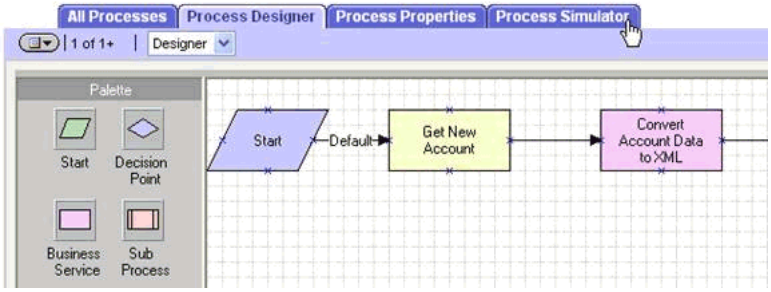


9. Enter the IP address and port for the HTTPRequestURLTemplate input argument.

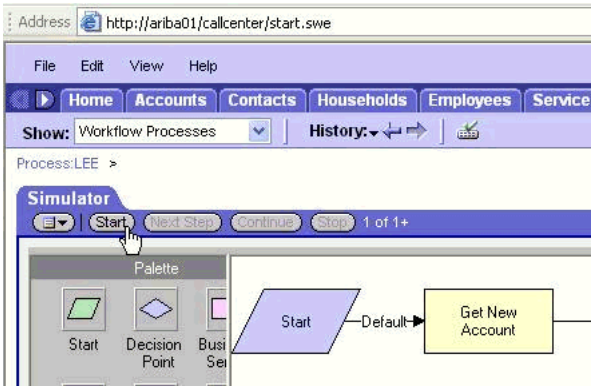
10. Click **Return To Designer**.



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

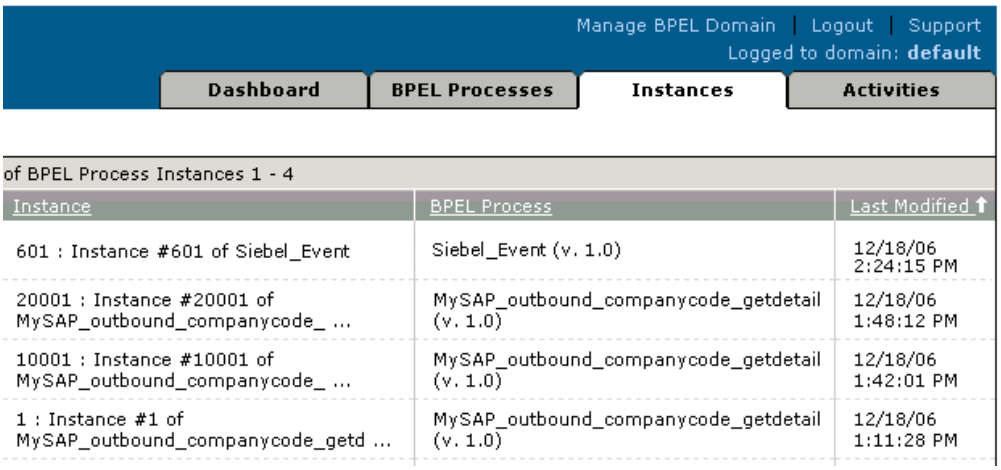
To verify your results:

1. Log in to Oracle BPEL Console at

   `http://host:port/BPELConsole`

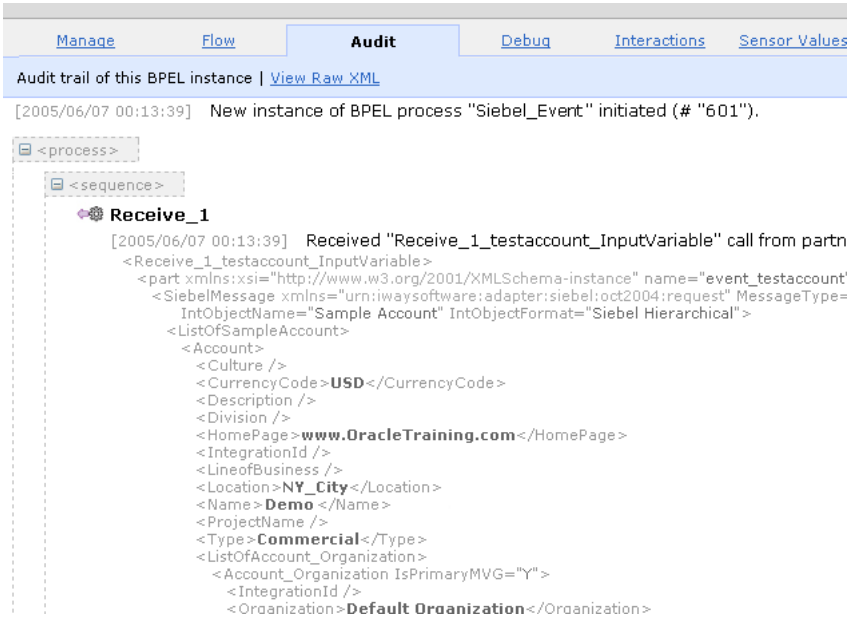
2. Provide a valid user name and password.
3. Click the **Instances** tab.

Recently received runtime events are displayed in the Instances tab.



4. Click the Siebel instance, then click **Audit** to see the event message.

   The message received from the Siebel system is displayed in the Audit tab.



6

ESB Integration Examples

This chapter contains the following examples:

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

The scenarios shown in this chapter require the following prerequisites.

Prerequisites

The following are installation and configuration requirements:

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

See Also: *Oracle Application Server Adapters Installation Guide*

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

- How to configure OracleAS Adapter for Siebel for services and events. See Chapter 2, "Configuring Oracle Application Server Adapter for Siebel".
- How to configure a new Application Server and Integration Server connection in Oracle JDeveloper. For more information, see Chapter 4, "Integration with Oracle BPEL Process Manager".
- How to use Siebel workflows. See Appendix A, "Using Siebel Workflows" for information on Siebel design requirements.

Overview of ESB Integration

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

Adapter integration with OracleAS ESB is a two-step process:

1. **Design Time:** OracleAS 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 runtime instructions to enable the communication between participating applications.

Namespace Requirements

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

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

1. Input XML for ESB based on unqualified namespaces:

```
<?xml version="1.0" encoding="UTF-8"?>
<Siebel location="S/BO/Account/Account/query">
<select>
<Name>A*</Name>
</select>
  <Field>Name</Field>
  <Field>Country</Field>
  <Field>State</Field>
  <Field>Account Status</Field>
  <Field>Employees</Field>
</Siebel>
```

2. Input XML for ESB based on qualified namespaces:

```
<Siebel xmlns="urn:iwaysoftware:adapter:siebel:oct2004:request"
location="S/BO/Account/Account/query">
            <select>
              <Name>ChennaiQA</Name>
            </select>
            <field>Account Competitors</field>
            <field>Name</field>
            <field>City</field>
            <field>Street Address</field>
            <field>Country</field>
            <field>Currency Code</field>
            <field>State</field>
            <field>Account Status</field>
</Siebel>
```

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

Configuring an ESB Outbound Process

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

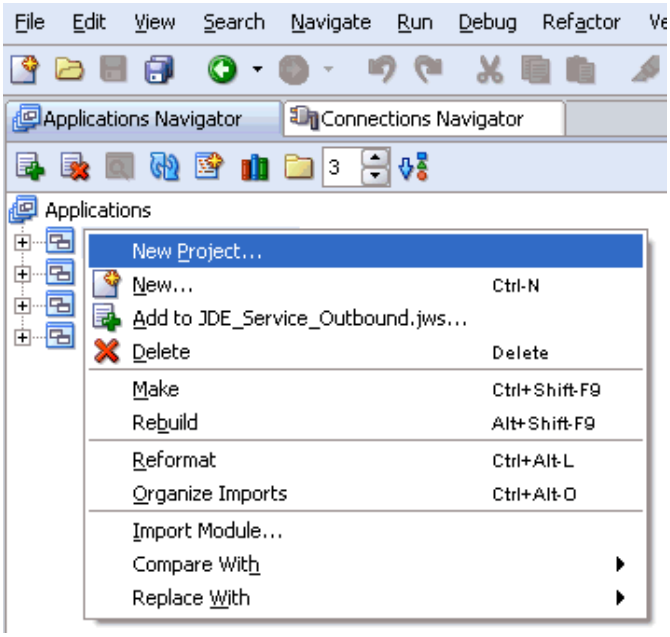
Prerequisites

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

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

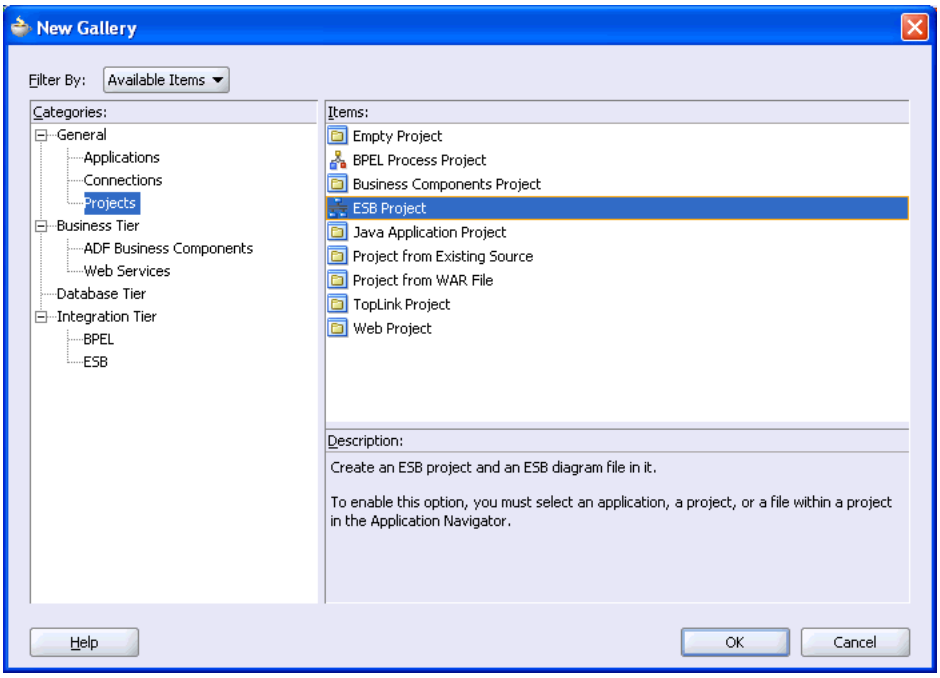
Creating an Outbound ESB Project and Assigning an Outbound WSDL File

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



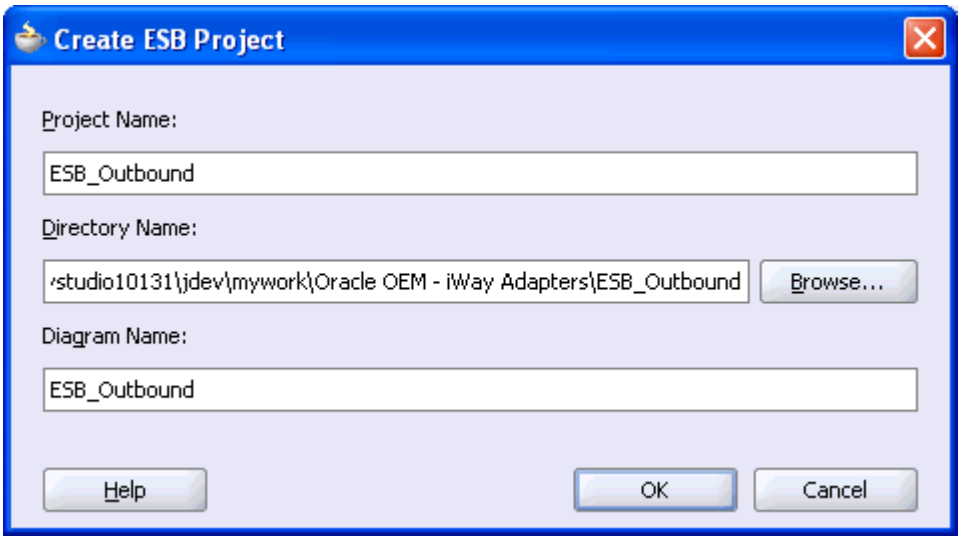
2. Right-click an application node that you created and select **New Project**.

   The New Gallery window is displayed.



3. From the Items list, select **ESB Project Project** and click **OK**.

   The Create ESB Project dialog box is displayed.



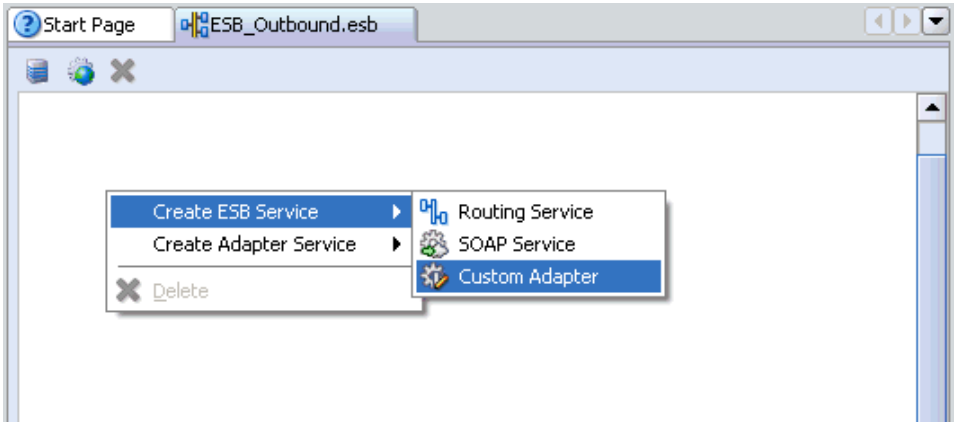
4. Perform the following steps:

   a. Specify a name for the ESB project.

      The Directory Name field and Diagram Name fields are updated automatically.

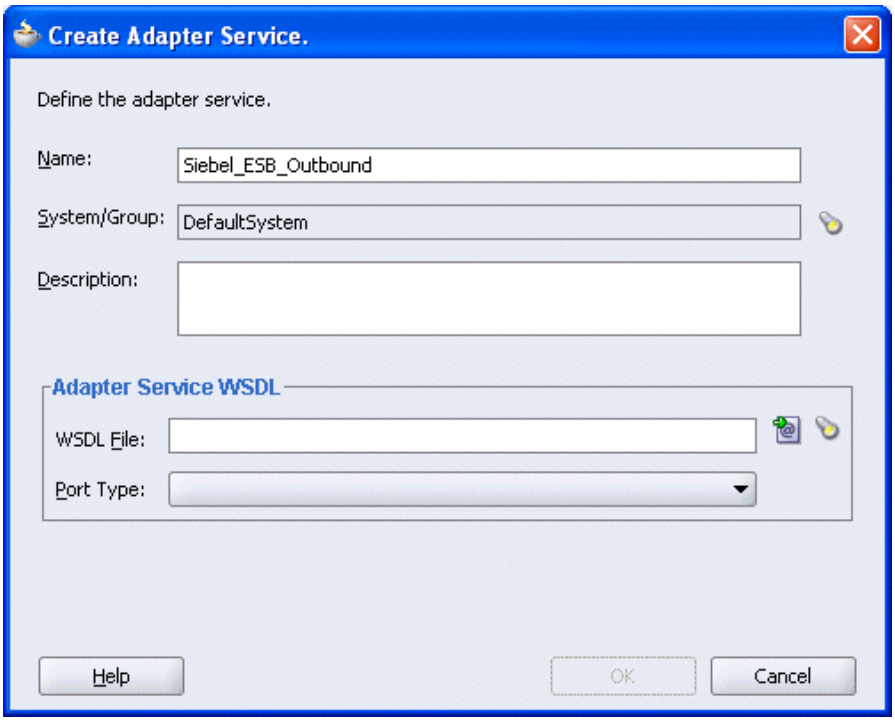
   b. Click **OK**.

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



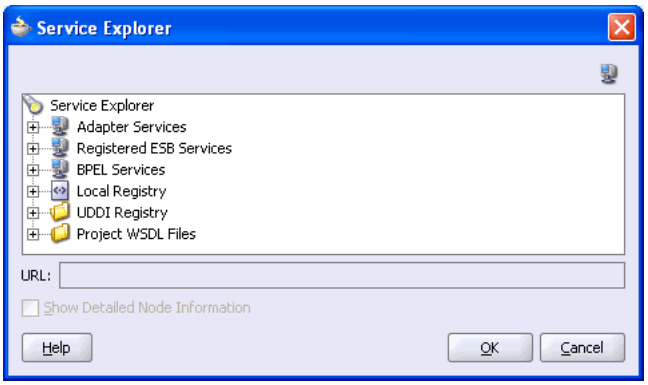
5. Right-click the ESB project in the middle pane, select **Create ESB Service** followed by **Custom Adapter**.

   The Create Adapter Service dialog box is displayed.



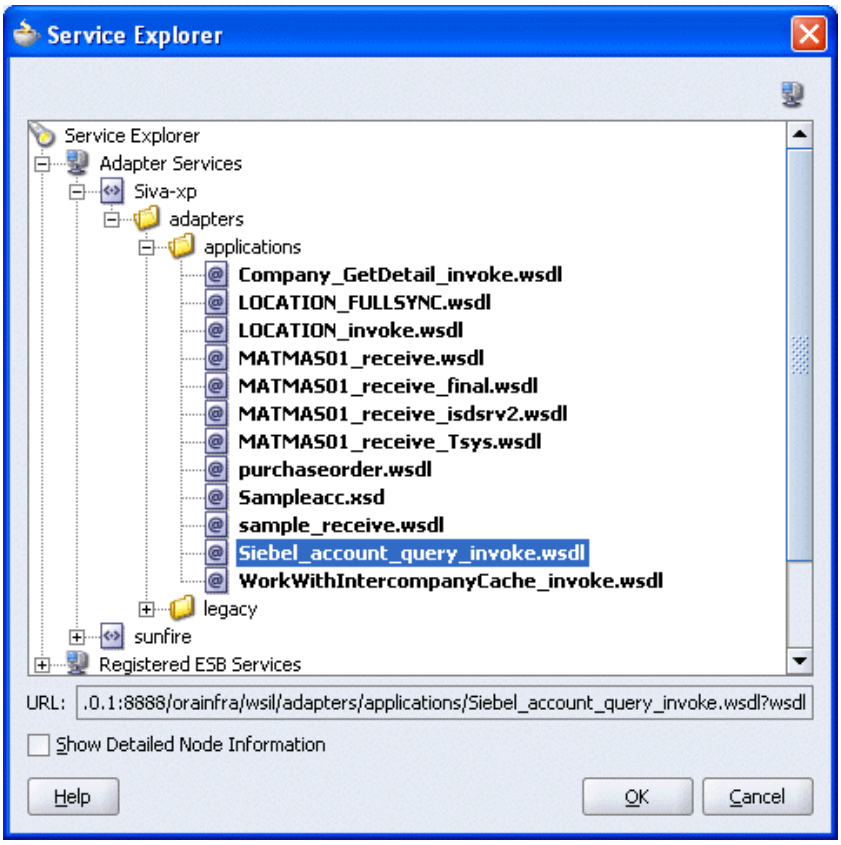
6. Enter a name for the adapter service and click the **Service Explorer** icon (second icon from the left preceding the **WSDL File** field).

The Service Explorer dialog box is displayed.



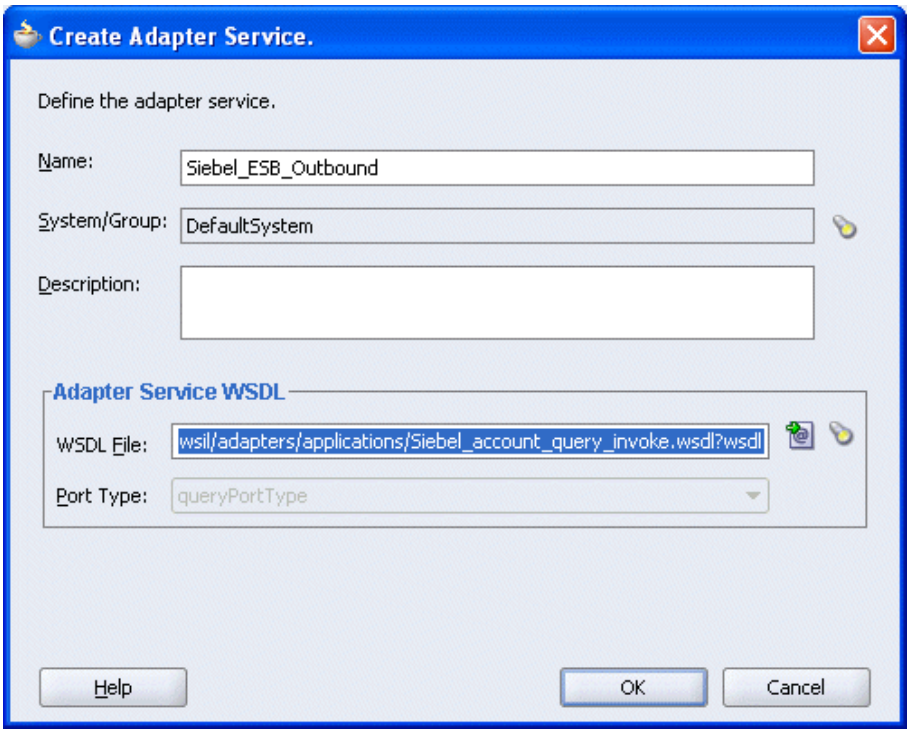
7. Expand your new connection under Adapter Services, followed by **adapters**, and then **applications**.

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



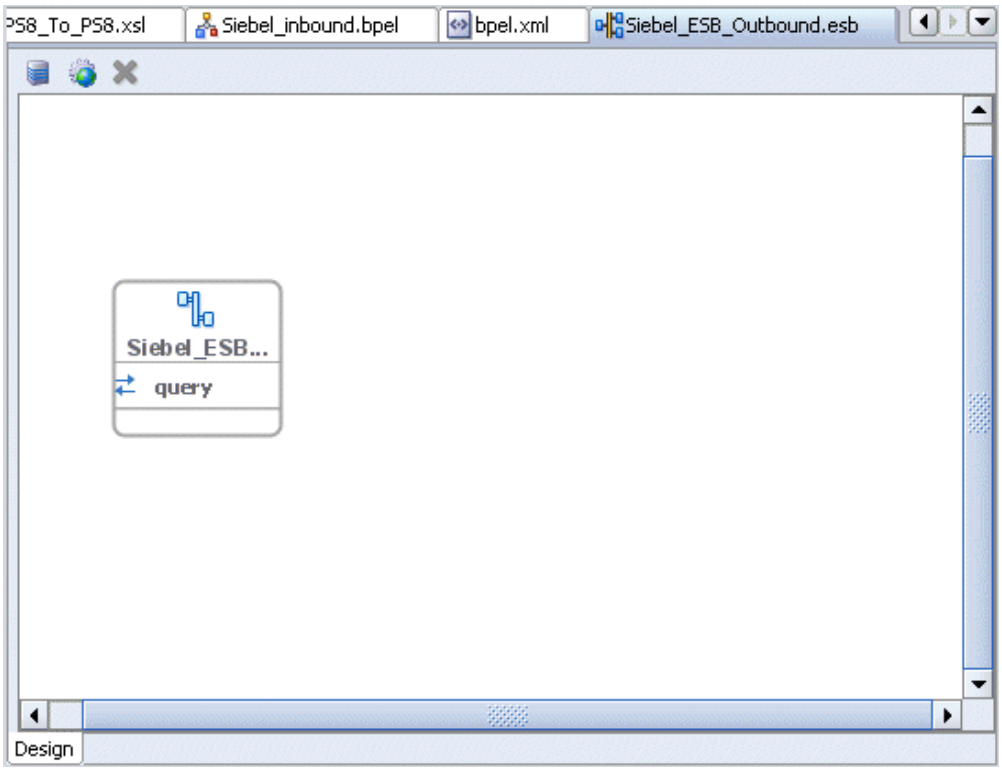
8. Select an outbound WSDL file that has been created using Application Explorer and click **OK**.

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



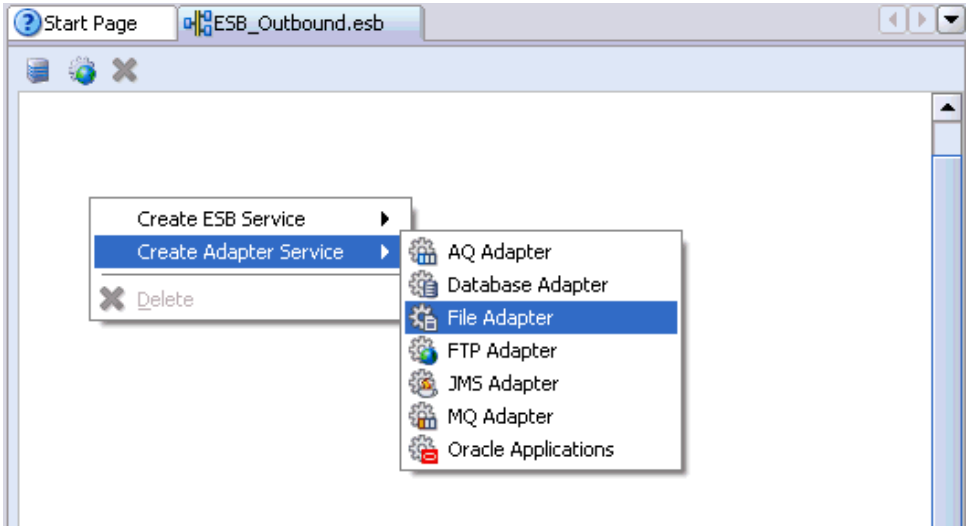
9. Click **OK**.

   The new ESB project appears in the visual editor.

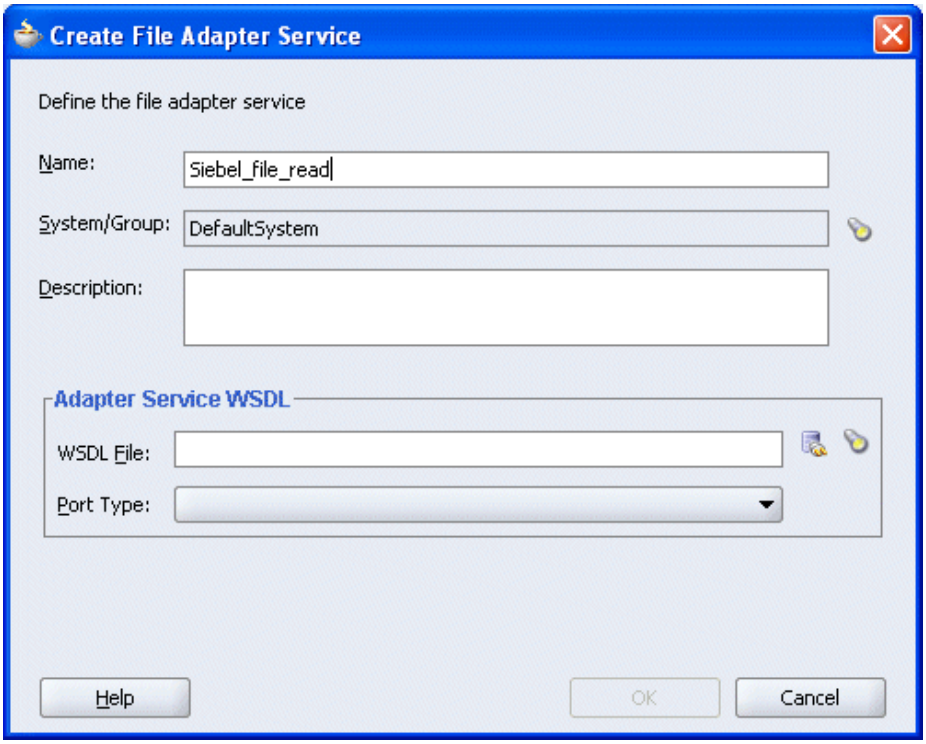


Creating a Read Process Operation Using the File Adapter

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

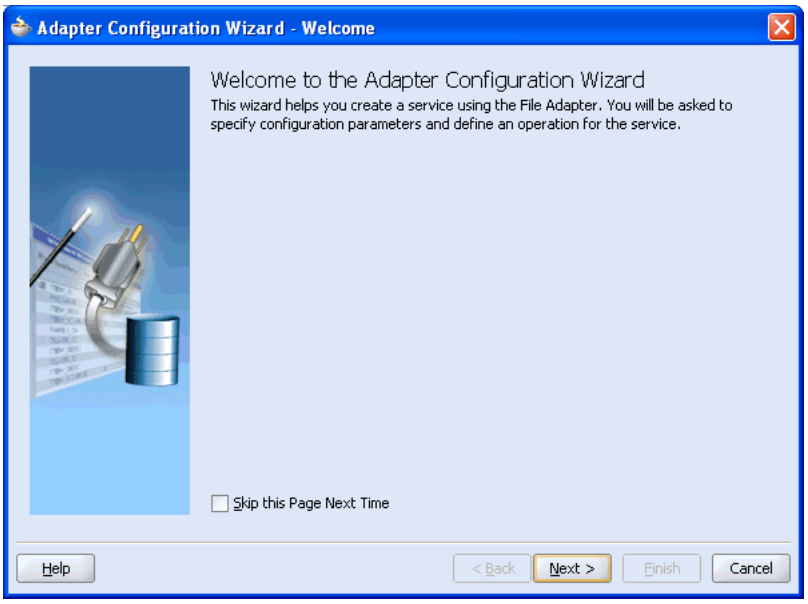


The Create File Adapter Service dialog box is displayed.



2. Enter a name for the File adapter and click the **Configure adapter service wsdl** icon next to the **WSDL File** field.

   The Adapter Configuration Wizard - Welcome window is displayed.

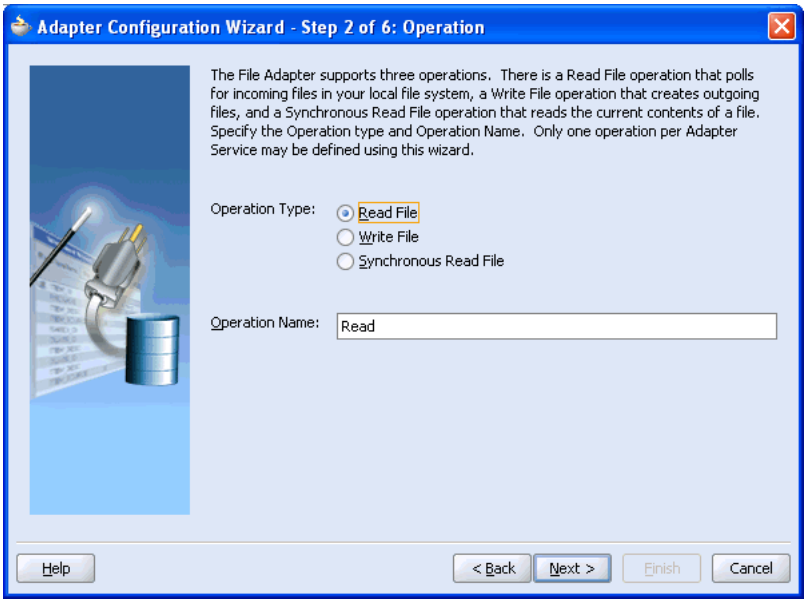


3. Click **Next**.

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

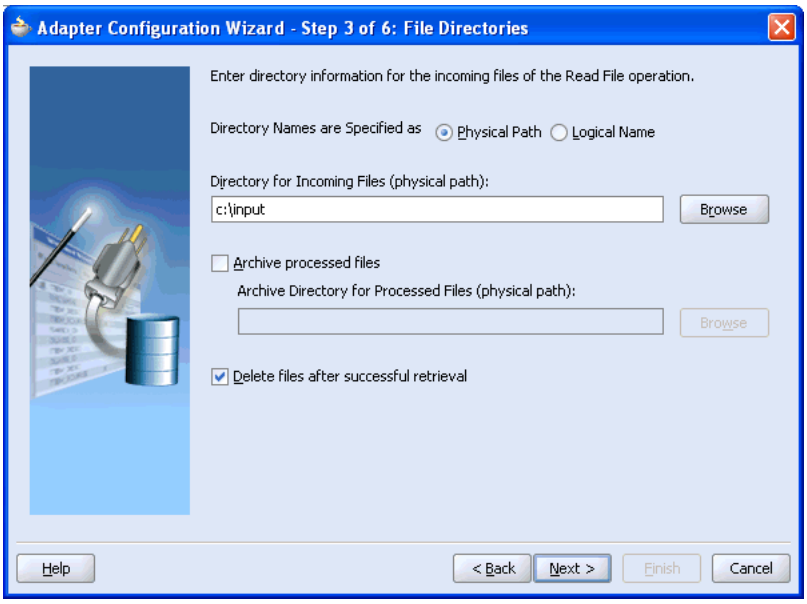
4. Click **Next**.

   The Adapter Configuration Wizard - Step 2 of 6: Operation window is displayed.



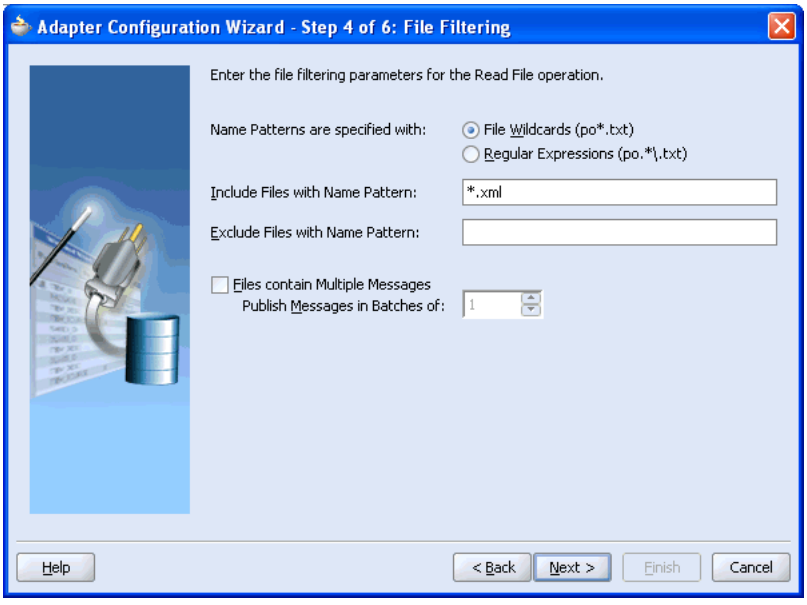
5. Click **Read File** as the Operation Type and click **Next**.

   The Adapter Configuration Wizard - Step 3 of 6: File Directories window is displayed.



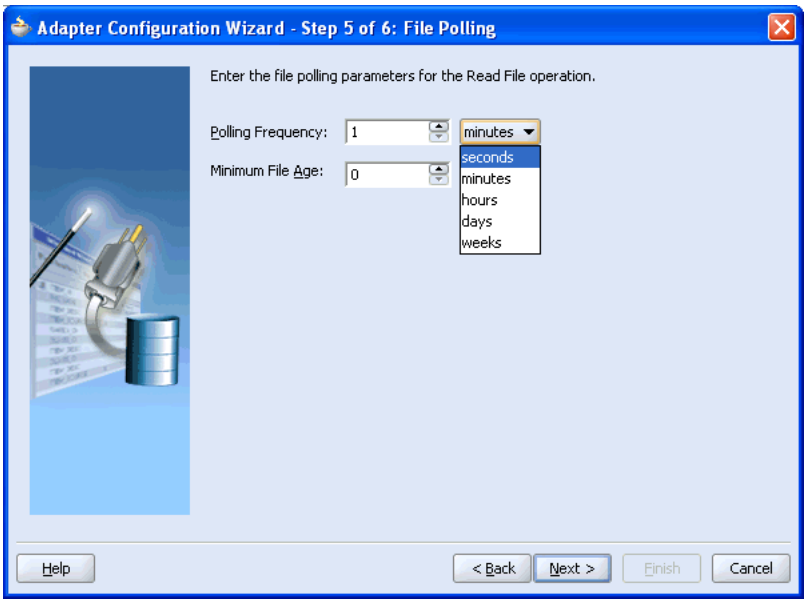
6. Enter the path of the input directory where you are placing the incoming XML file and click **Next**.

   The Adapter Configuration Wizard - Step 4 of 6: File Filtering window is displayed.



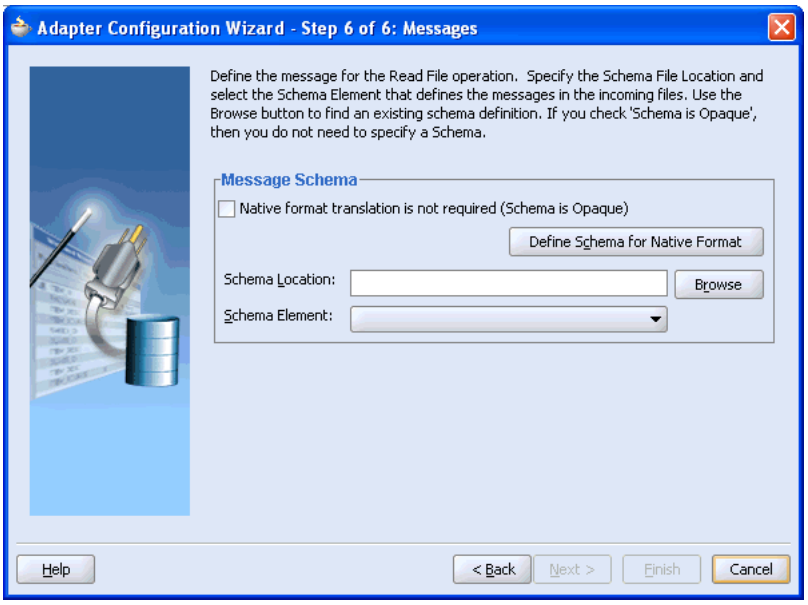
7. Enter the input file extension, for example ***.xml**, and click **Next**.

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



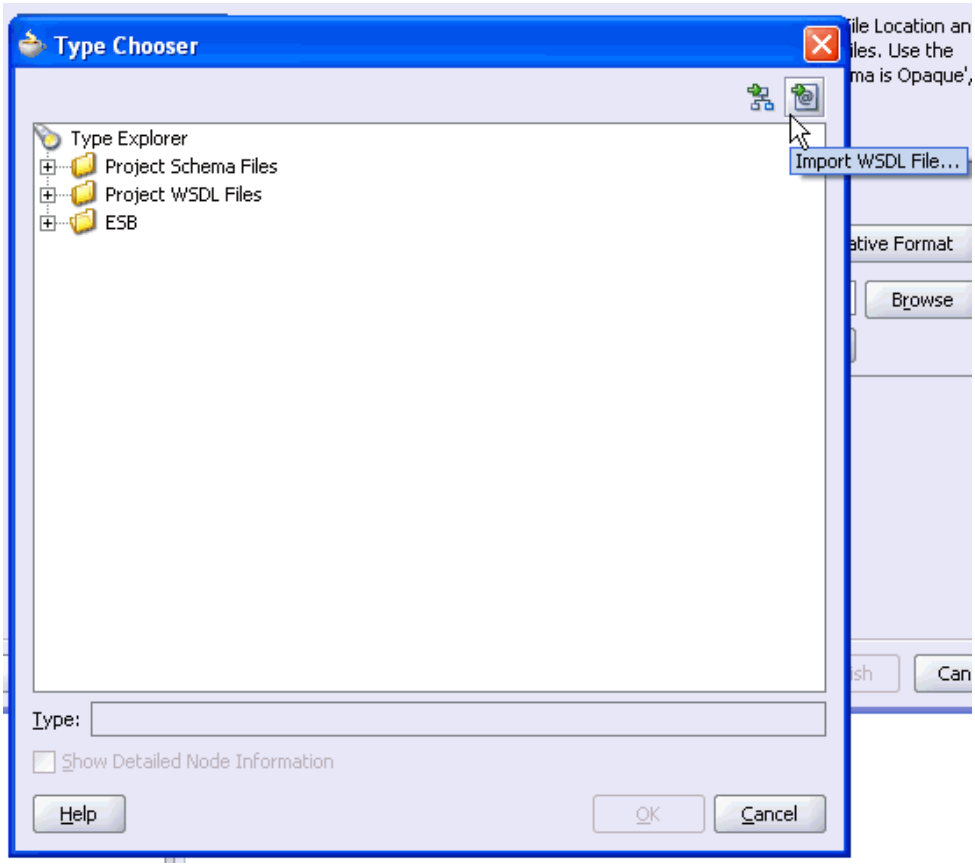
8. Change the Polling Frequency to seconds and click **Next**.

   The Adapter Configuration Wizard - Step 6 of 6: Messages window is displayed.



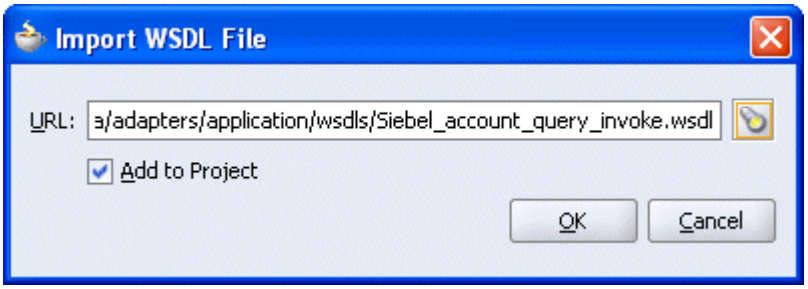
9. Click **Browse** to select the WSDL.

   The Type Chooser window is displayed.



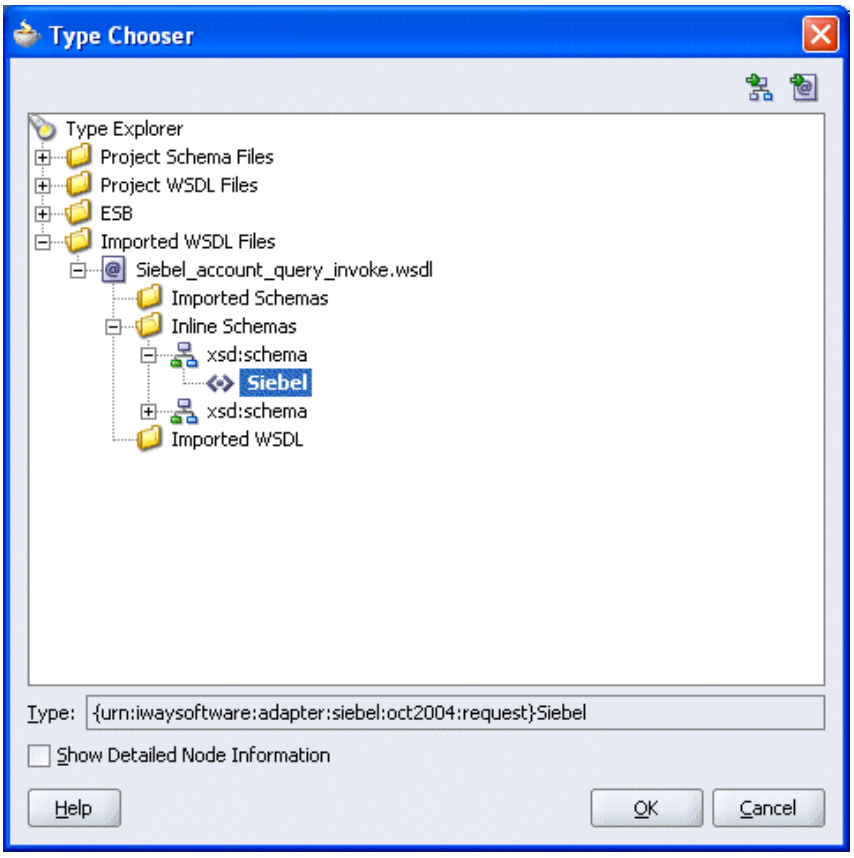
10. Click the **Import WSDL File** icon on the upper right corner of the dialog box.

    The Import WSDL File dialog box is displayed.



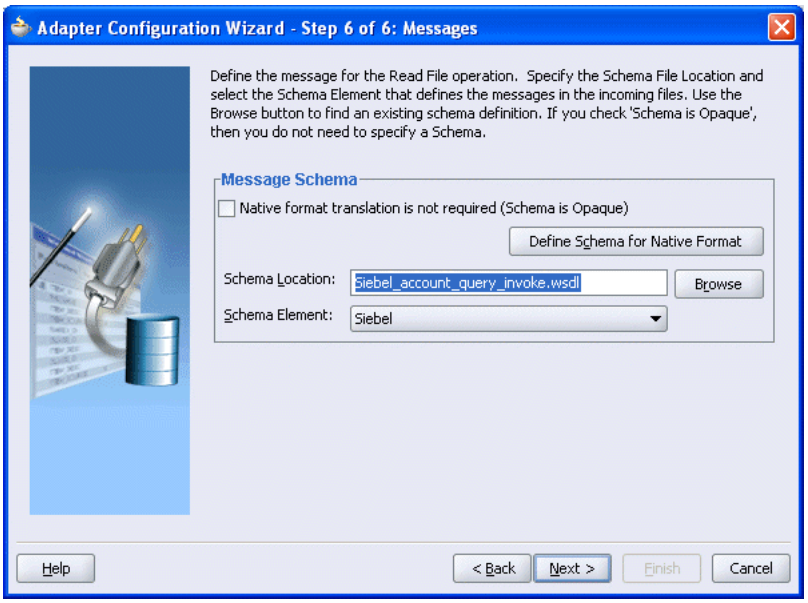
11. Select the WSDL file and click **OK**.

    The Imported WSDL Files folder is added.



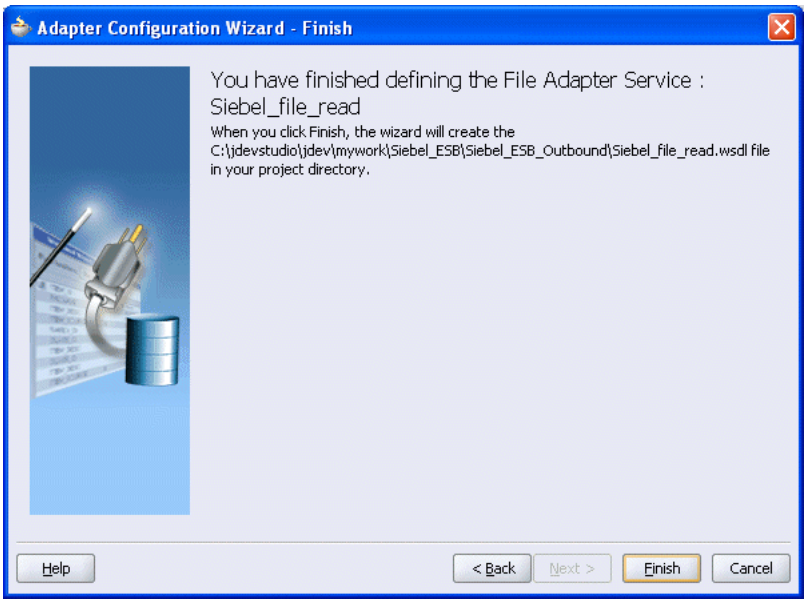
12. Expand the Imported WSDL Files folder, select an Inline Schema, for example, **PS8**, and click **OK**.

    You are returned to the Adapter Configuration Wizard - Step 6 of 6: Messages window.



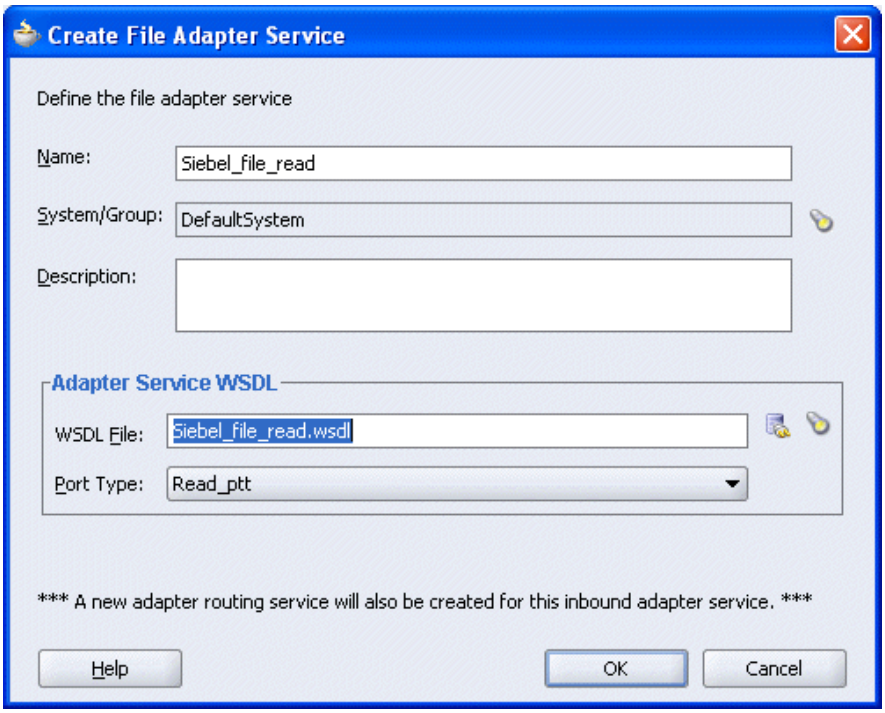
13. Click **Next**.

    The Adapter Configuration Wizard - Finish window is displayed.



14. Click **Finish**.

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

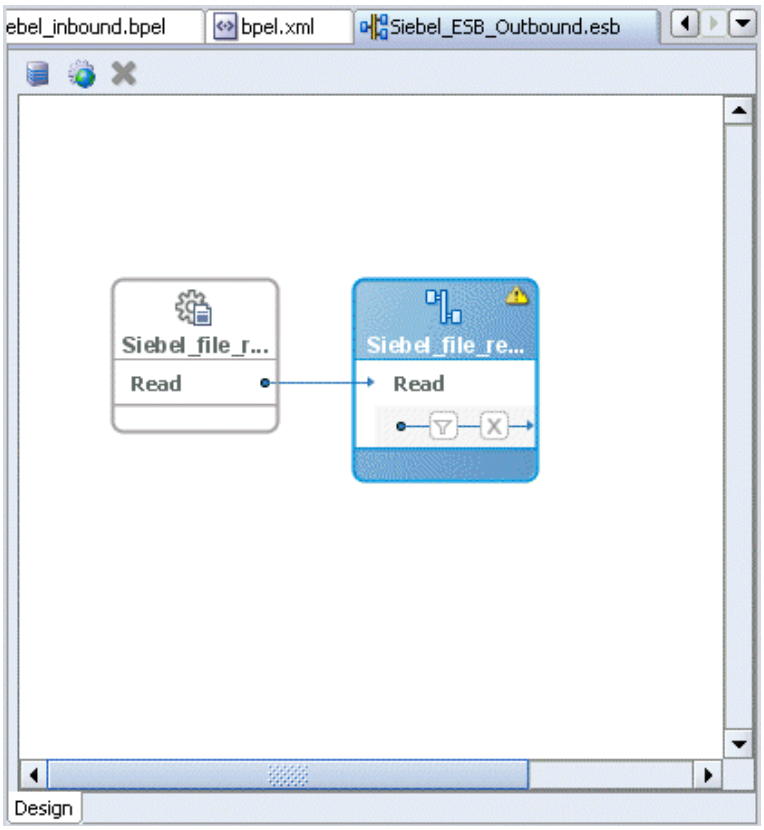


15. Click **OK**.

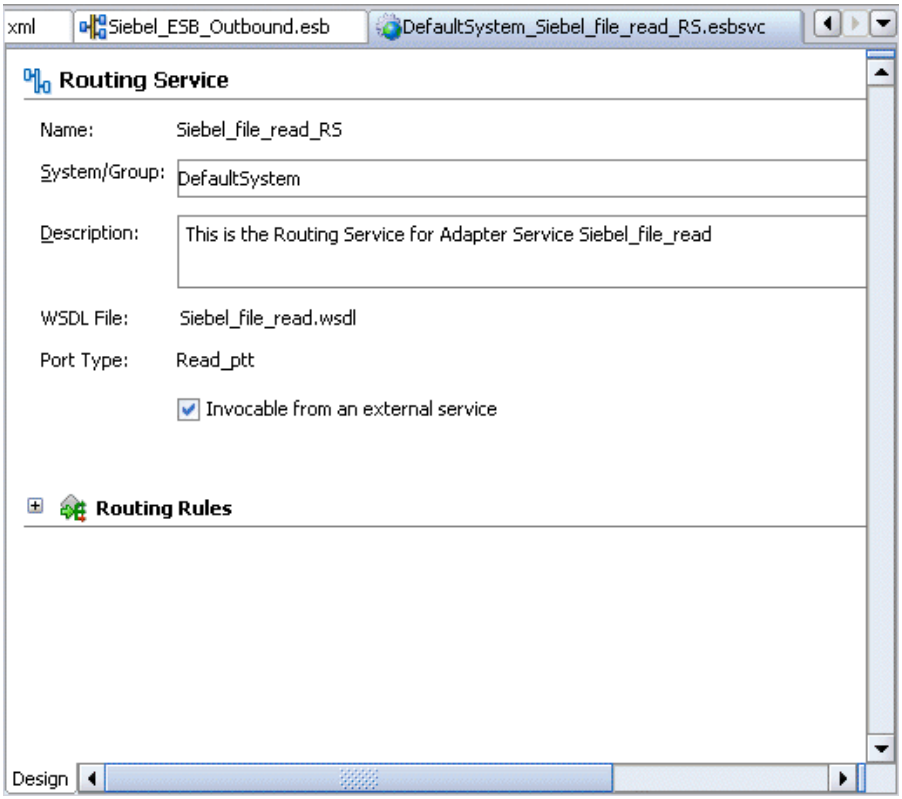
    The Read operation with a routing service is added to the ESB outbound project view.

Providing a Routing Service for the Read Operation

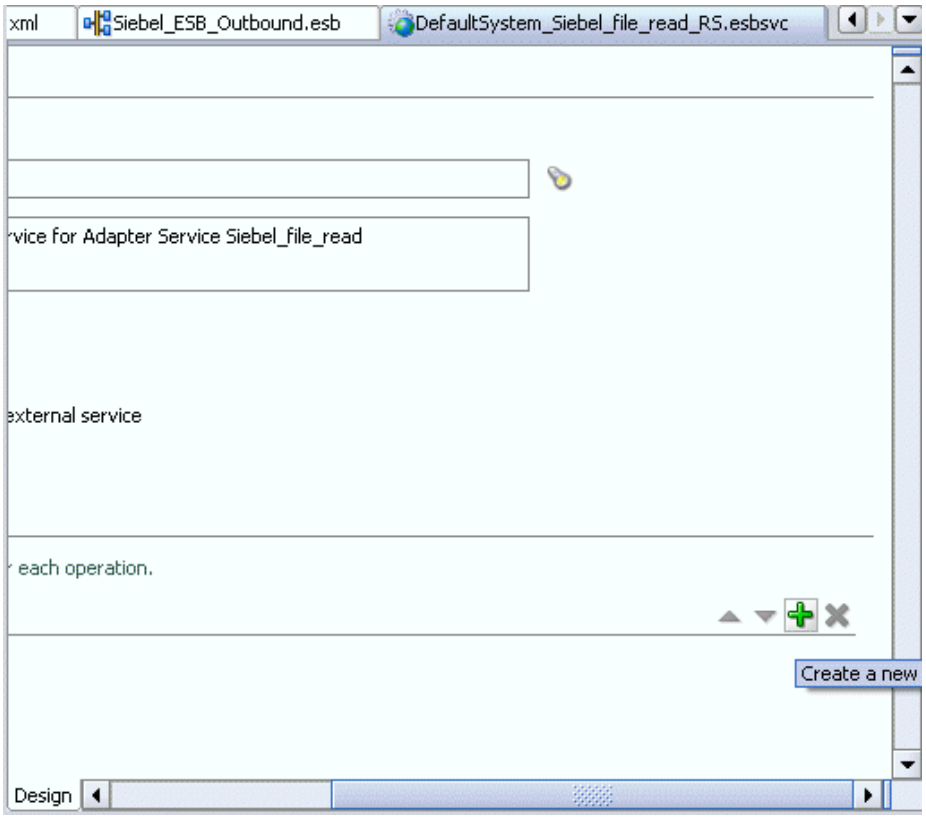
1. Double-click the routing service.



The Routing Service window is displayed.

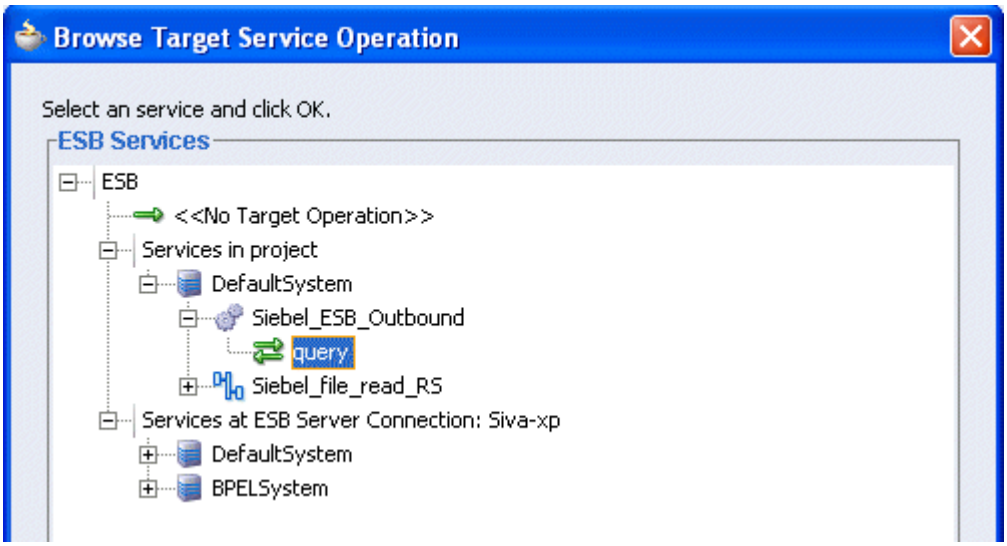


2. Expand the **Routing Rules**.



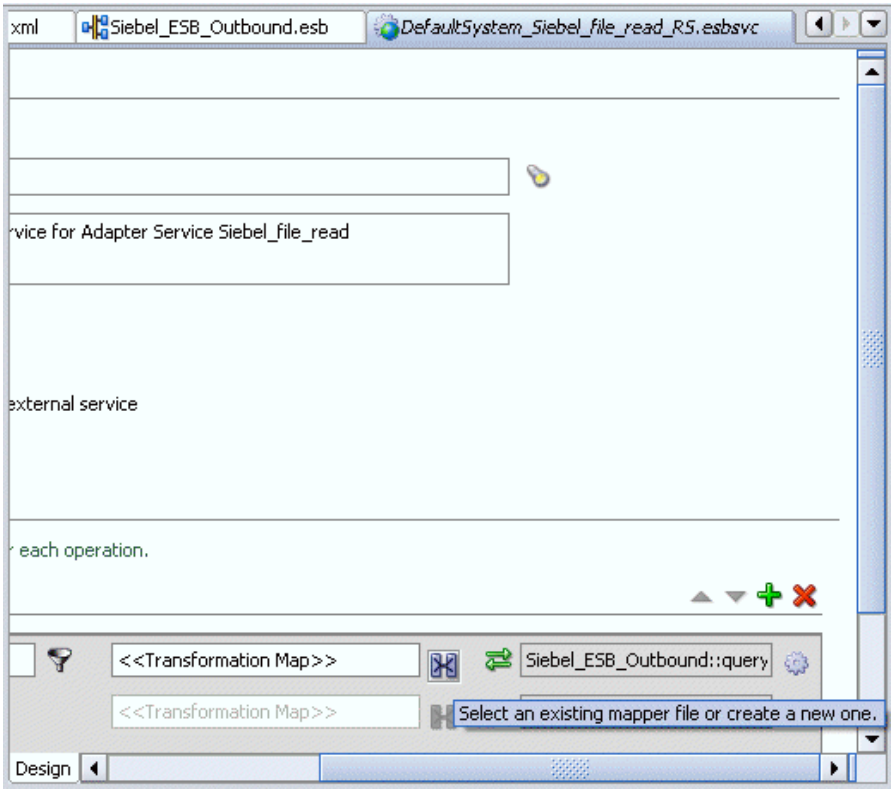
3. Click the green plus sign icon, which represents the option to **Create a new Routing Rule**.

   The Browse Target Service Operation window opens.



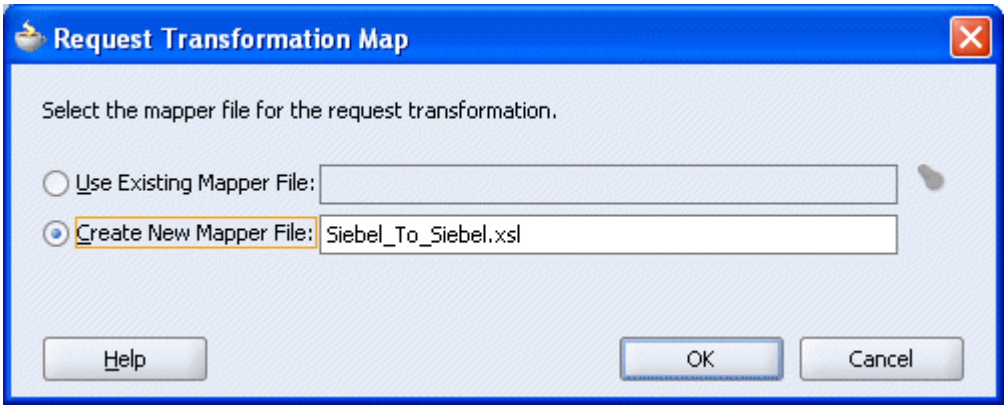
4. Expand **Services in project**, **Default System**, your adapter service node, for example, **CARRIER**, and select the service name, for example, **CARRIER**.
5. Click **OK**.

   You are returned to the Routing Rules window.



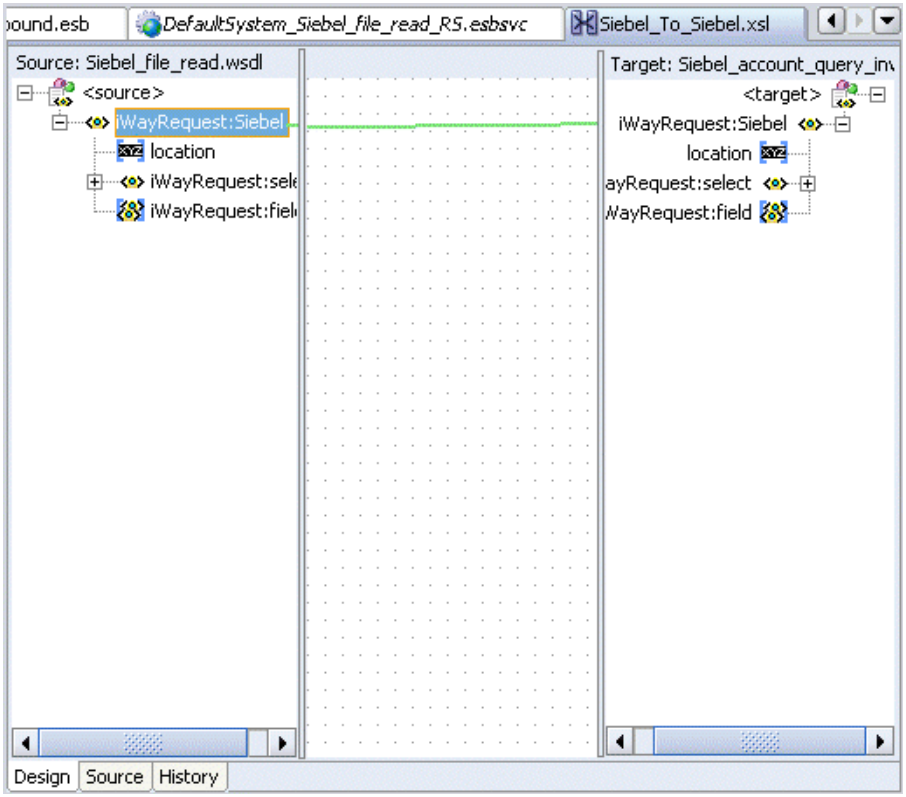
6. Click on the icon next to the <<Transformation Map>> field (**Select an existing mapper file or create a new one**).

   The Request Transformation Map dialog box is displayed.



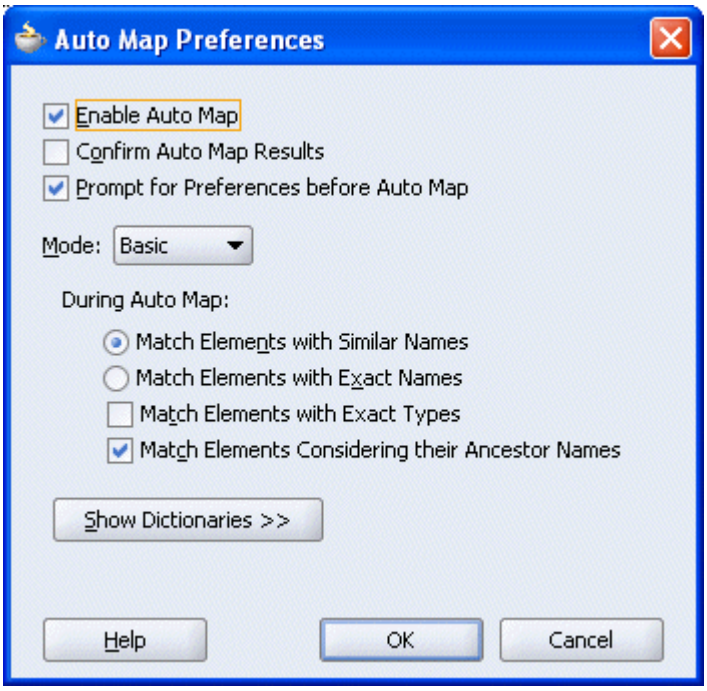
7. Select the **Create New Mapper File** option, specify the file name, and click **OK**.

   The following mapping window is displayed.



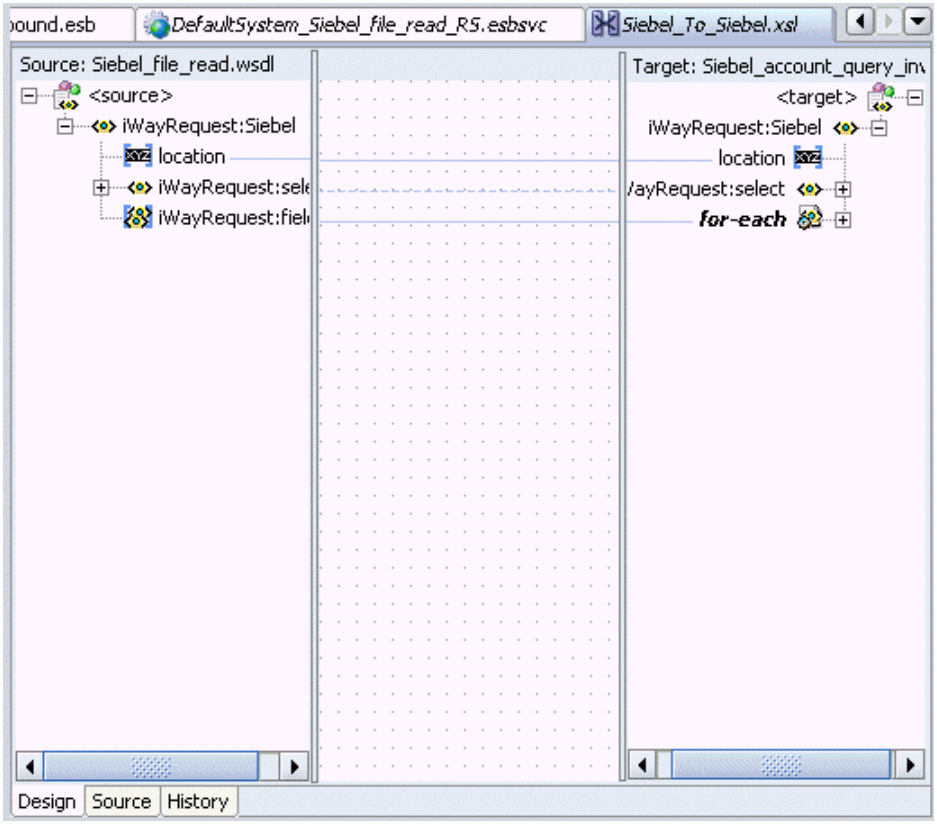
8. Select the WSDL file and map it to the Write operation.

   Once you map the WSDL file, the Auto Map Preferences dialog box is displayed.



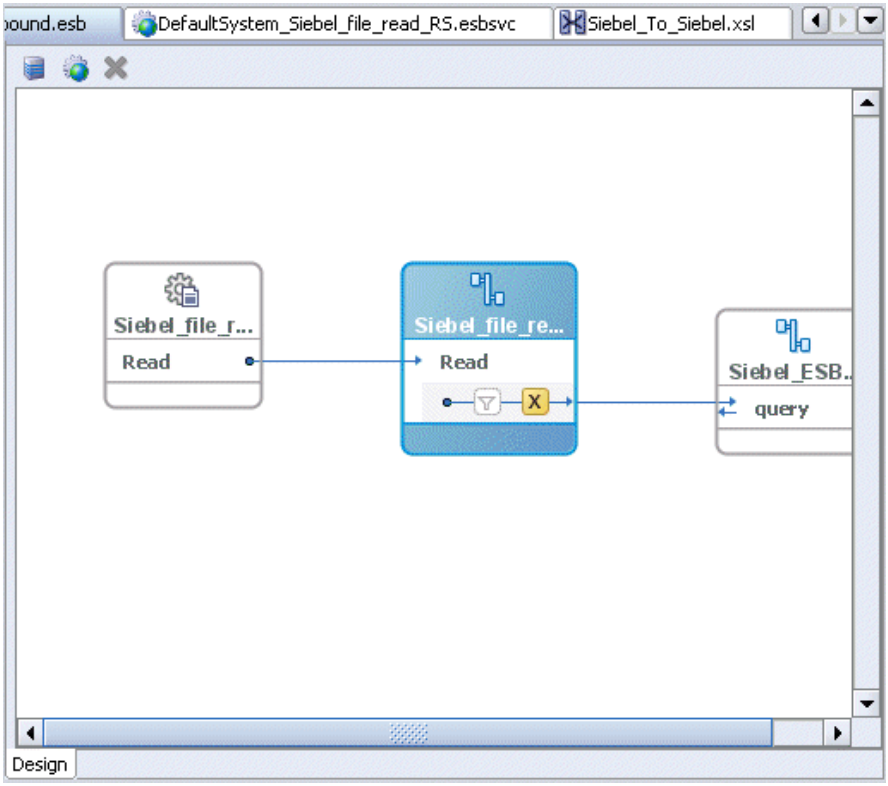
9. Click **OK**.

   The mapping is completed as shown in the following window.



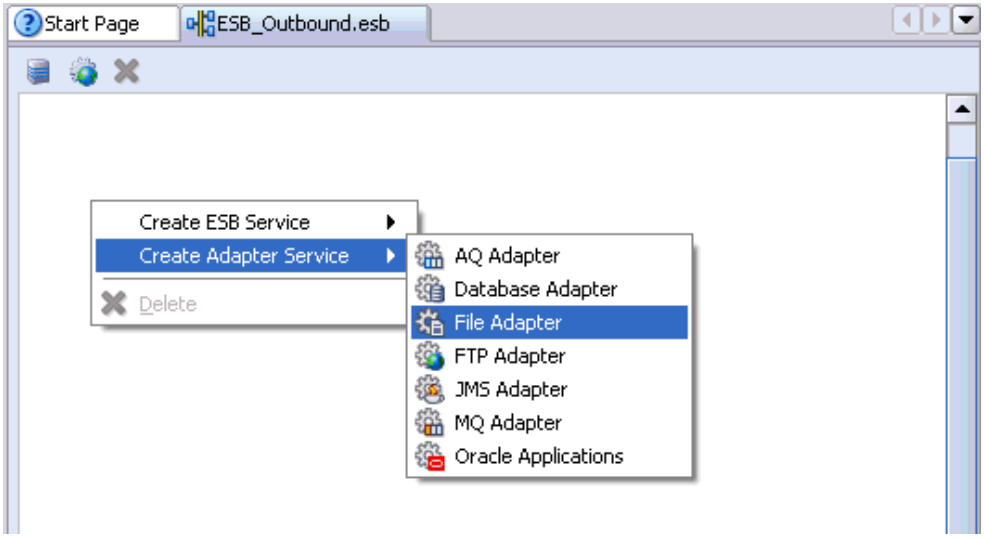
10. Double-click the ESB outbound project file in the left pane, for example, **ESB_Outbound.esb**.

    Notice that the Routing service is now created for the Read operation.

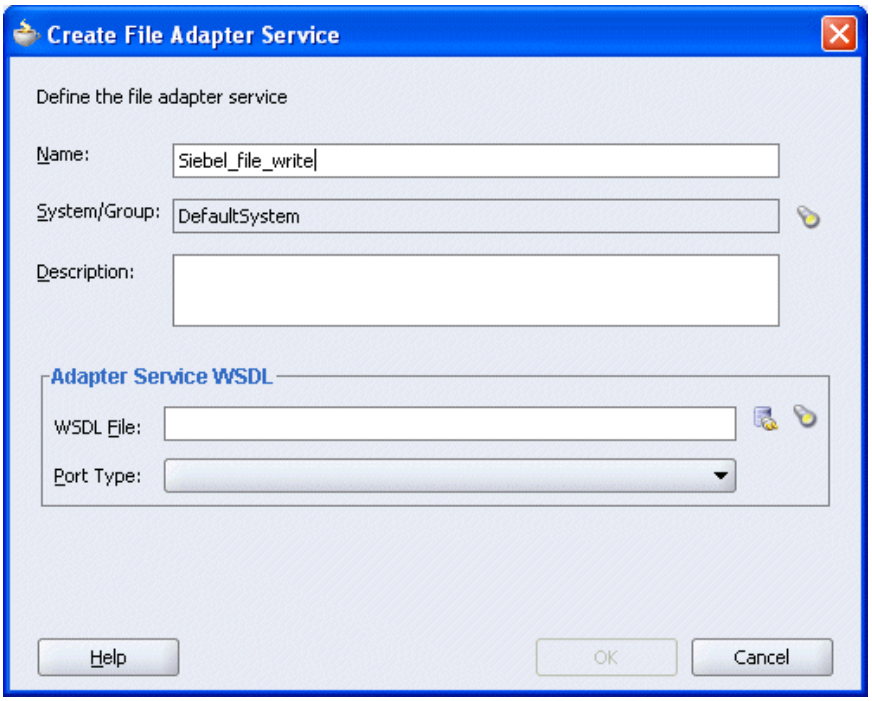


Creating a Write Process Operation Using the File Adapter

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

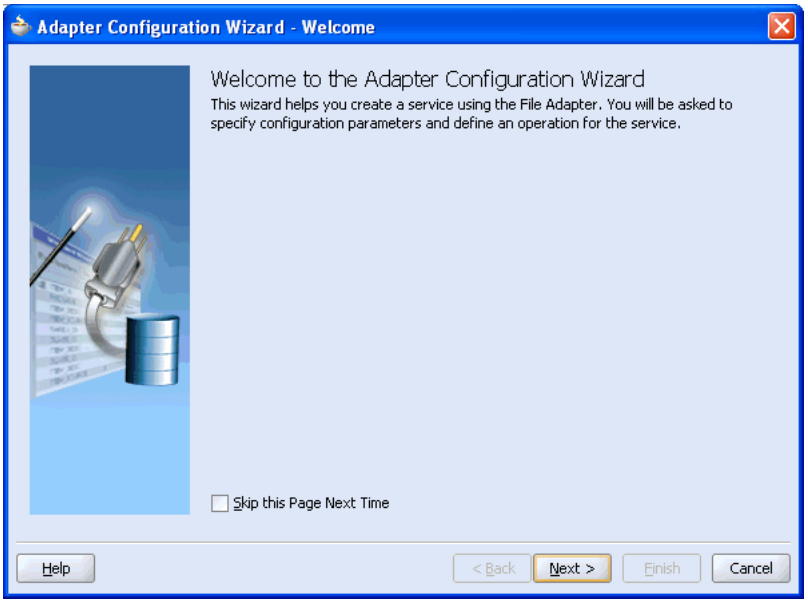


The Create File Adapter Service dialog box is displayed.



2. Enter a name for the File adapter and click the **Configure adapter service wsdl** icon next to the **WSDL File** field.

   The Adapter Configuration Wizard - Welcome window is displayed.



3. Click **Next**.

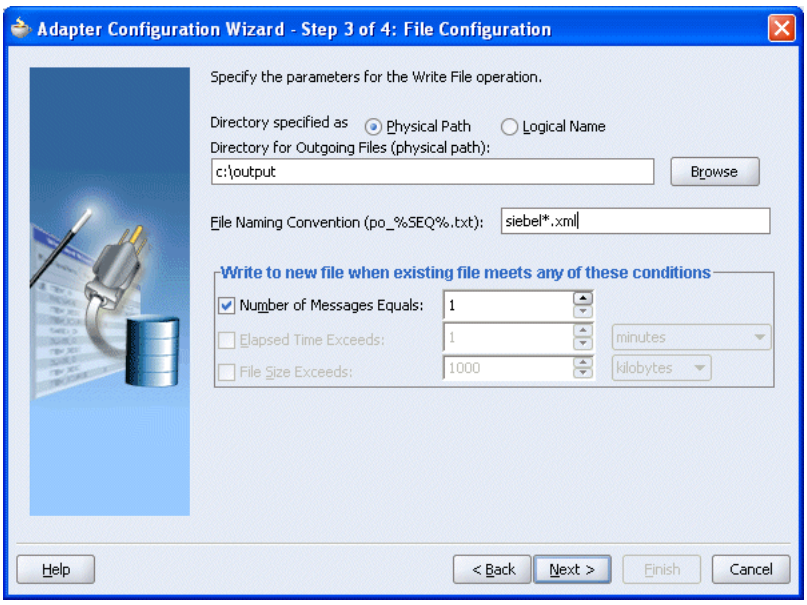
   The Adapter Configuration Wizard - Step 1 of 4: Service Name window is displayed.

4. Click **Next**.

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

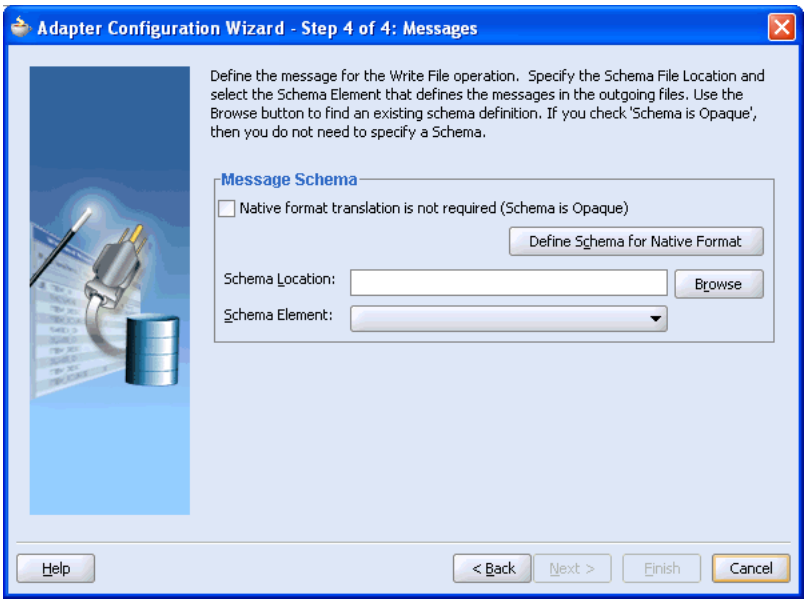
5. Click **Write File** as the Operation Type and click **Next**.

   The Adapter Configuration Wizard - Step 3 of 4: File Configuration window is displayed.



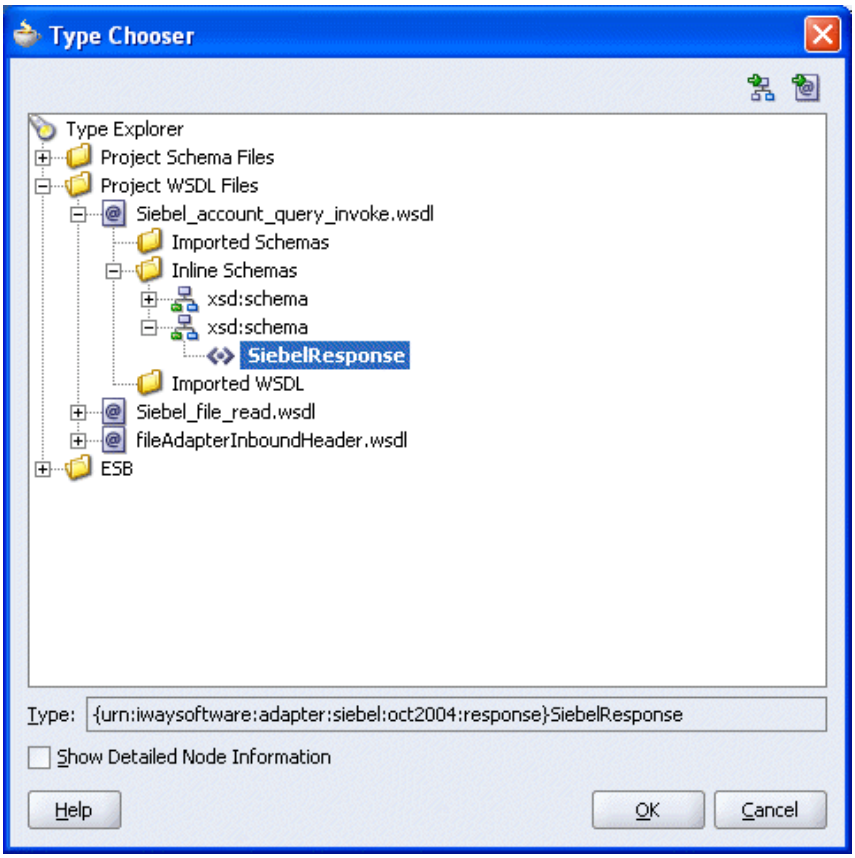
6. Enter the path of the output directory and name of the output file and click **Next**.

   The Adapter Configuration Wizard - Step 4 of 4: Messages window is displayed.



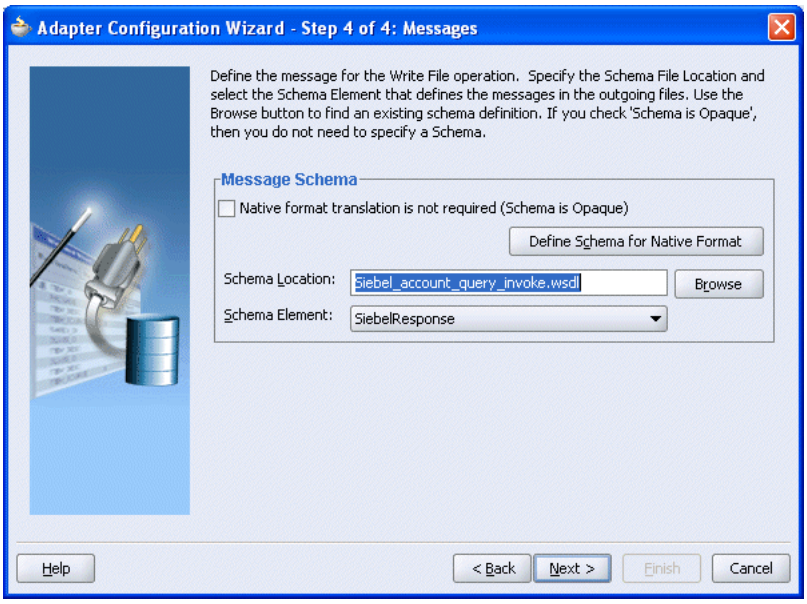
7. Click **Browse** to select the WSDL.

   The Type Chooser window is displayed.



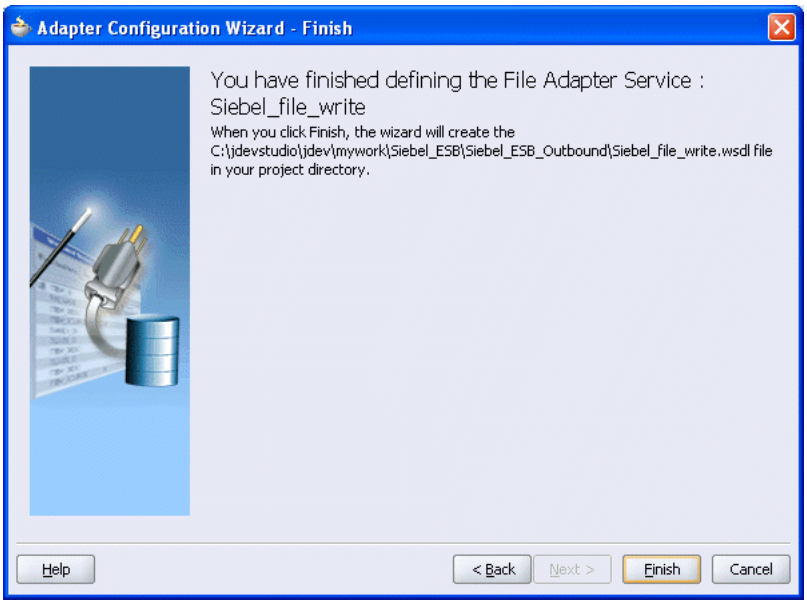
8. Expand the **Project WSDL Files** folder, select an Inline Schema, for example, **PS8**, and click **OK**.

   You are returned to the Adapter Configuration Wizard - Step 4 of 4: Messages window.



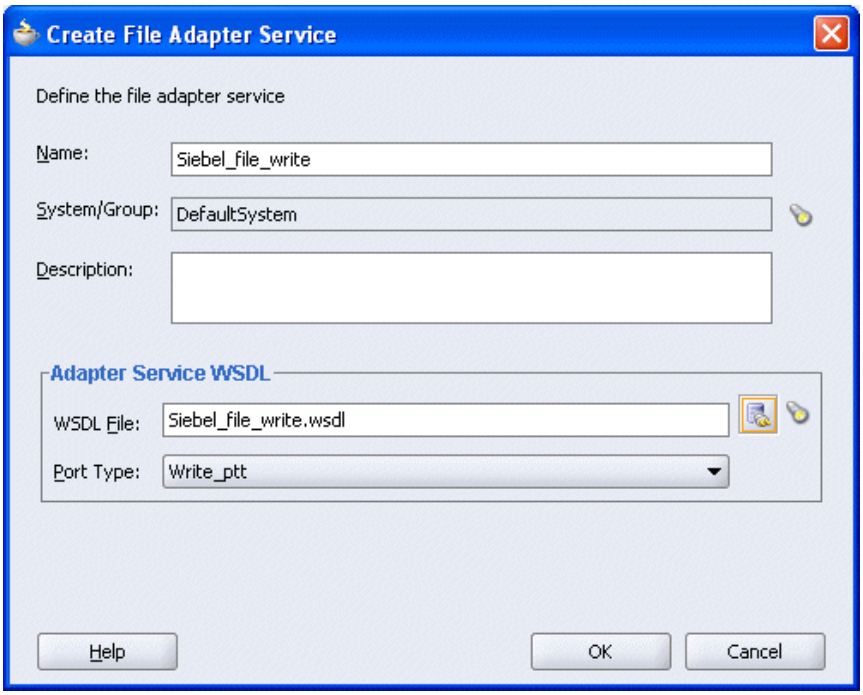
9. Click **Next**.

   The Adapter Configuration Wizard - Finish window is displayed.



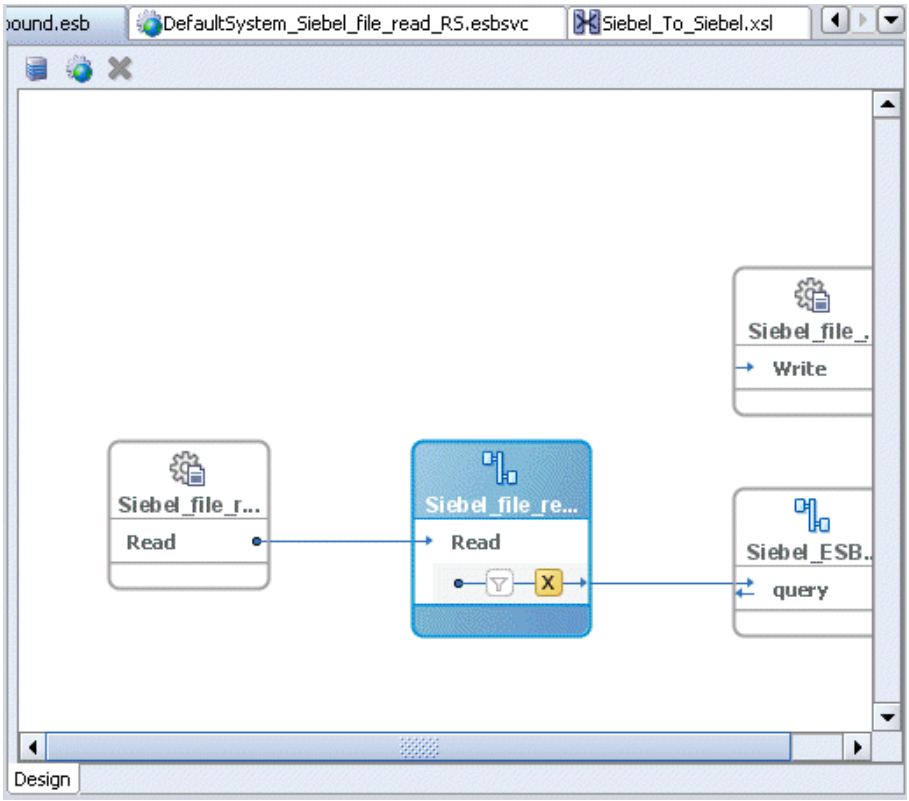
10. Click **Finish**.

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



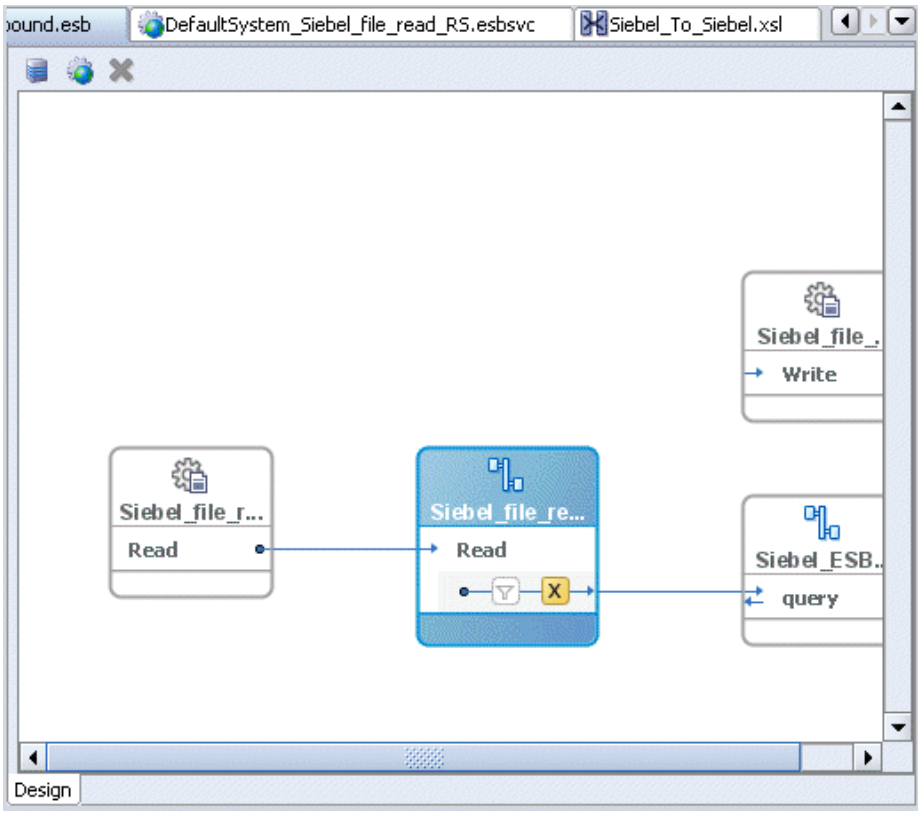
11. Click **OK**.

    The Write operation is added to the ESB outbound project view.



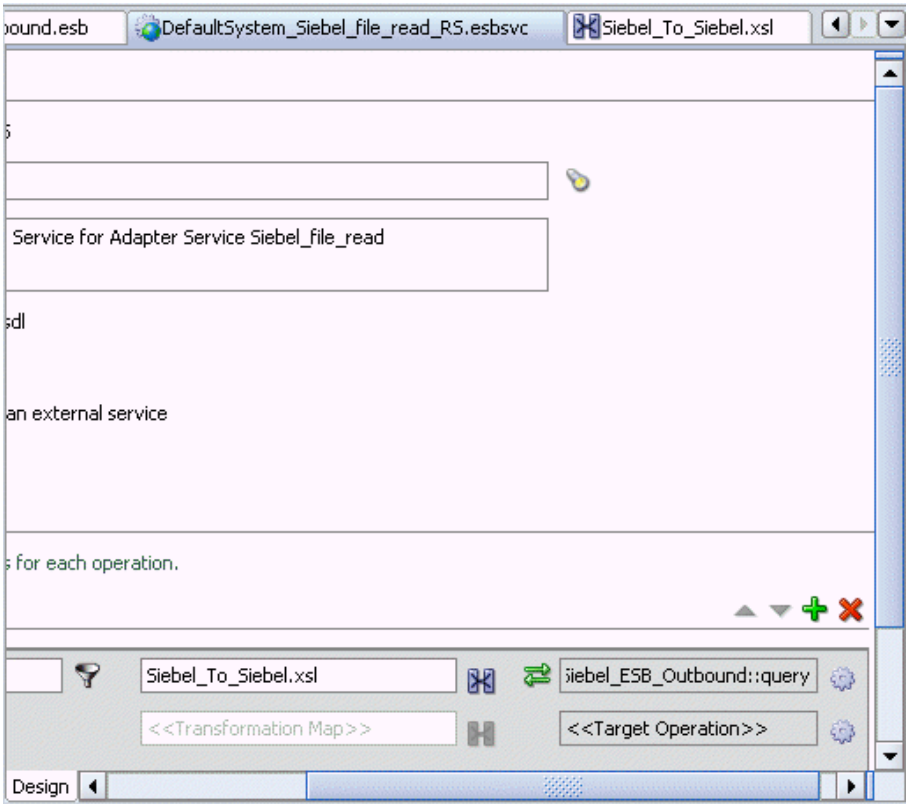
Providing a Routing Service for the Write Operation

1. Double-click the routing service.



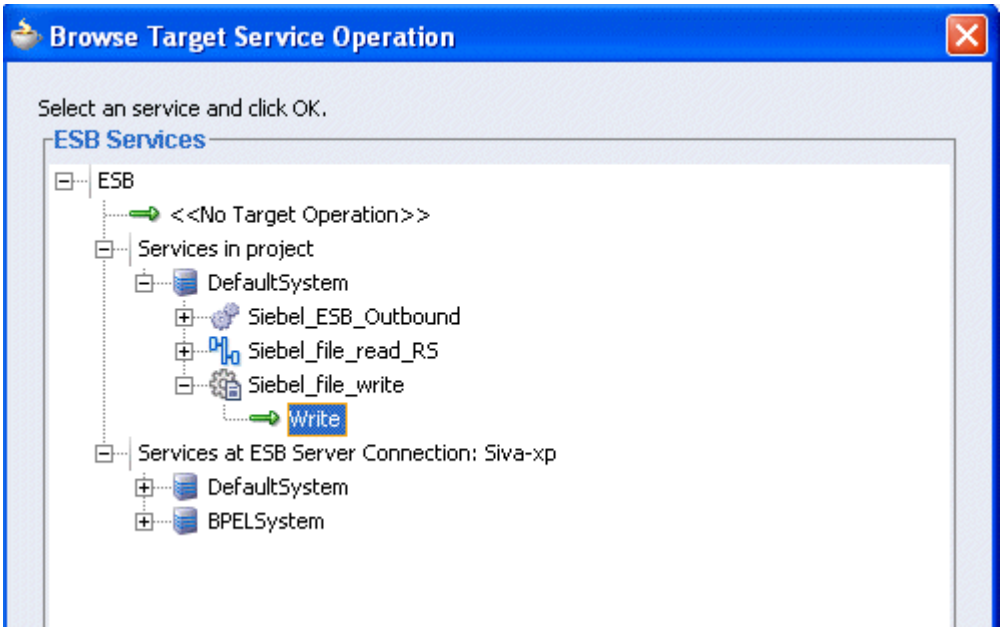
The Routing Service window is displayed.

2. Expand the **Routing Rules**.



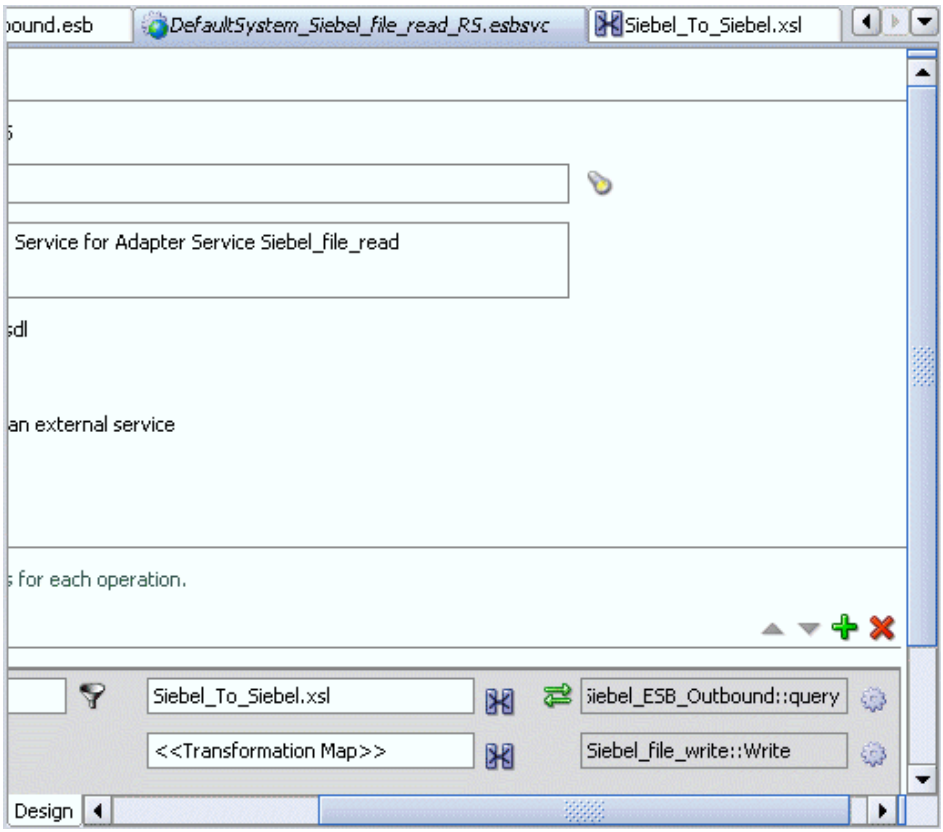
3. Click on the icon next to the <<Target Operation>> field (**Browse for target service operations**).

   The Browse Target Service Operation window opens.



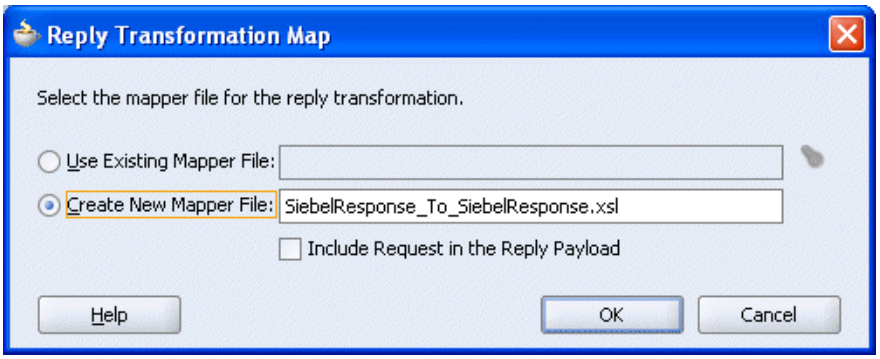
4. Expand **Services in project**, **Default System**, your adapter service node, for example, **write_operation**, and select the service name, for example, **Write**.
5. Click **OK**.

   You are returned to the Routing Rules window.



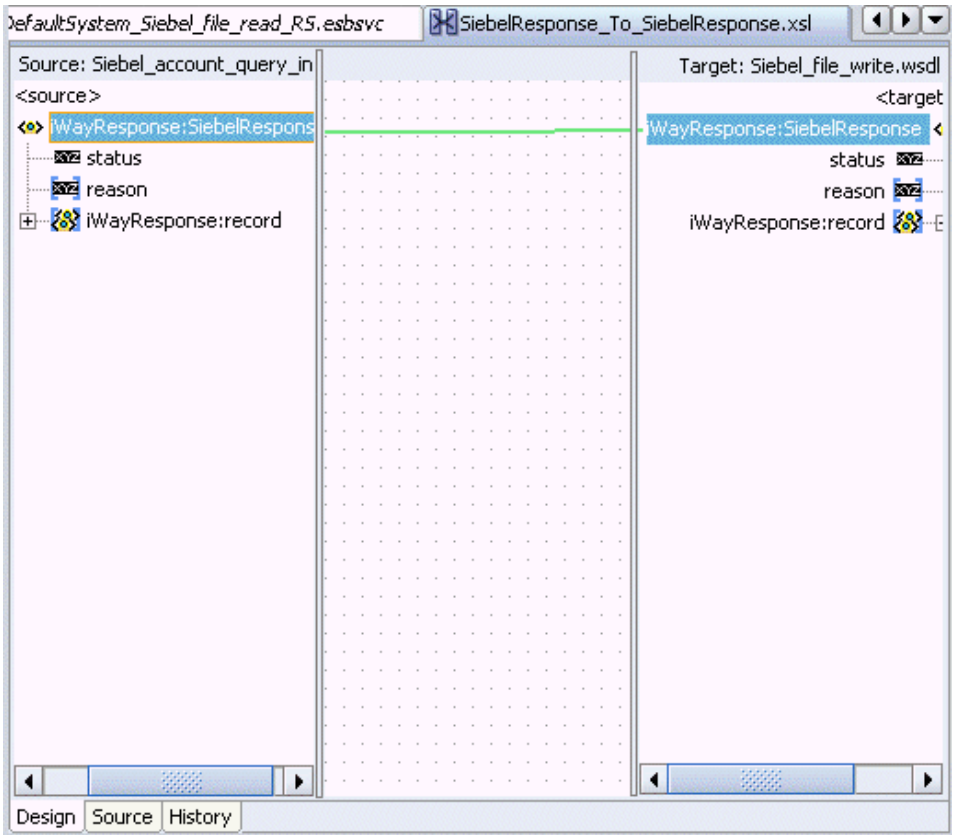
6. Click on the icon next to the <<Transformation Map>> field (**Select an existing mapper file or create a new one**).

   The Reply Transformation Map dialog box is displayed.



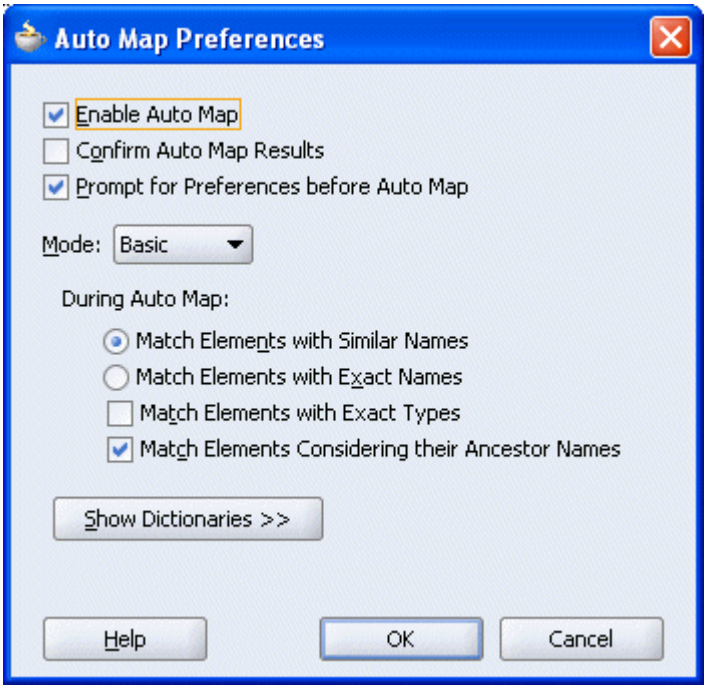
7. Select the **Create New Mapper File** option, specify the file name, and click **OK**.

   The following mapping window is displayed.



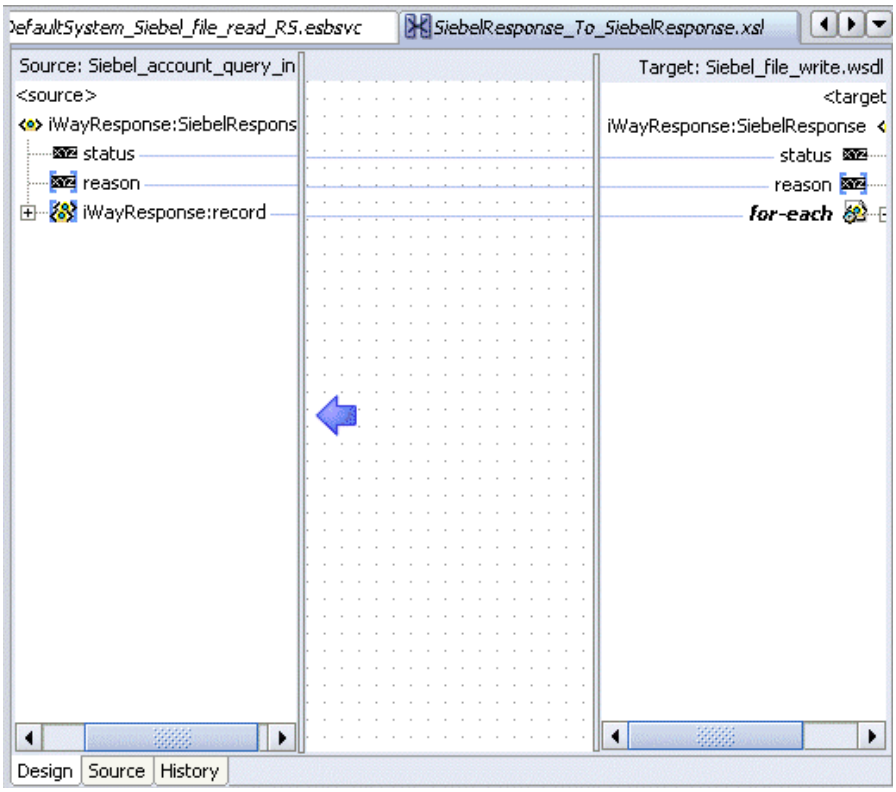
8. Select the WSDL file and map it to the Write operation.

   Once you map the WSDL file, the Auto Map Preferences dialog box is displayed.



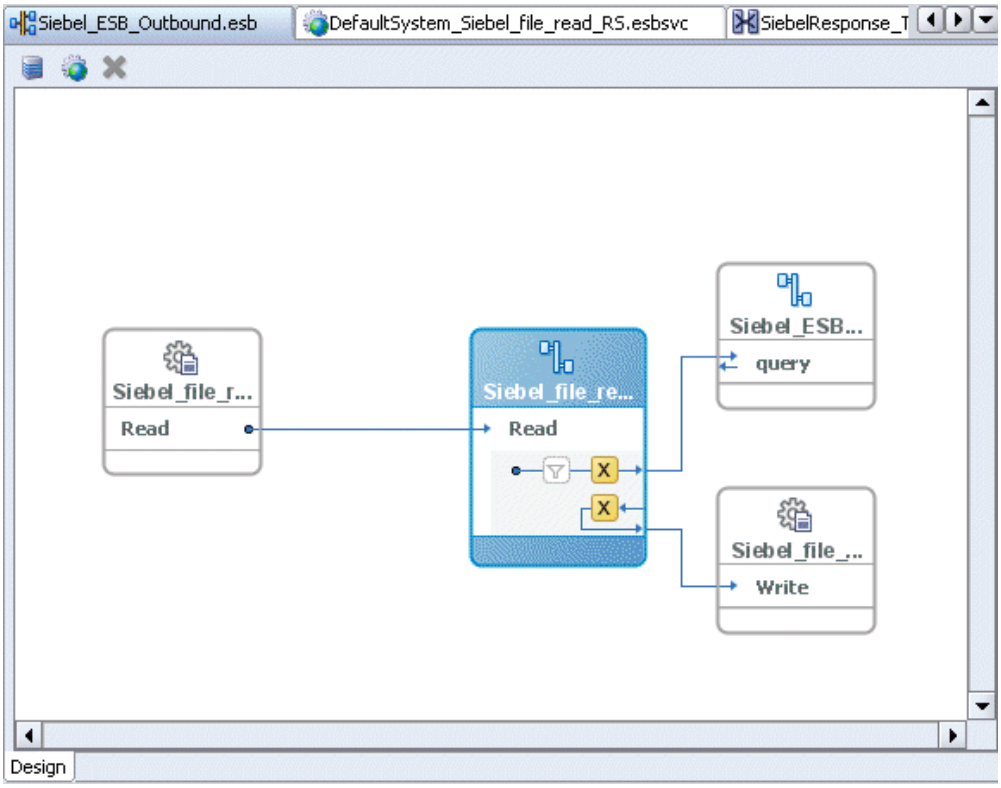
9. Click **OK**.

   The mapping is completed as shown in the following window.



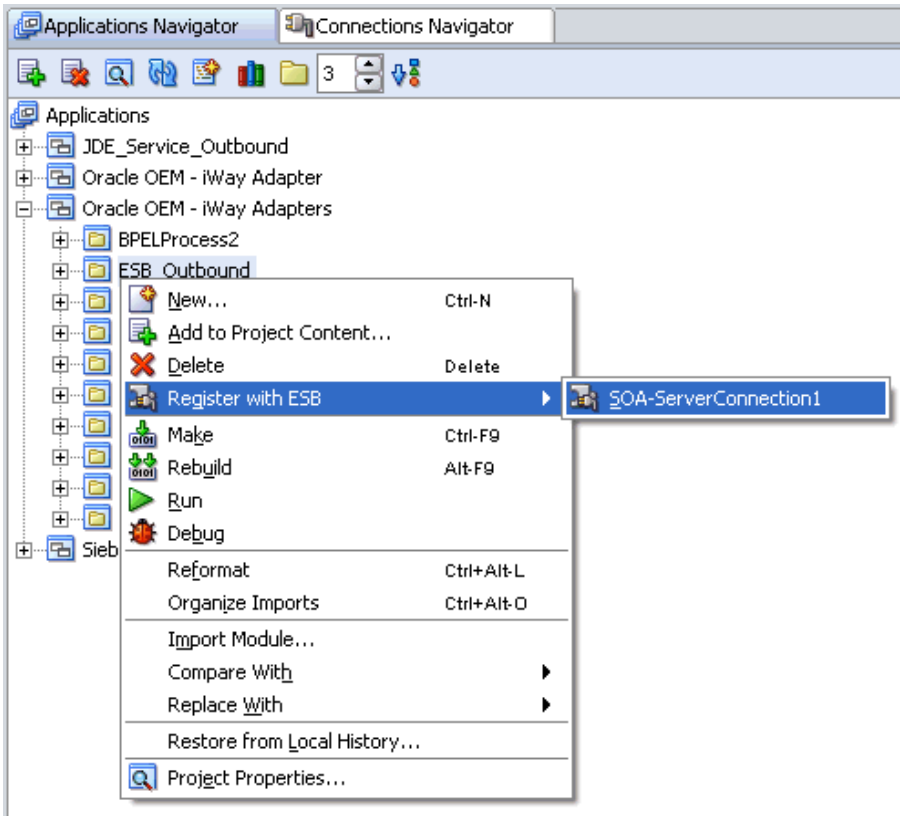
10. Double-click the ESB outbound project file in the left pane, for example, **ESB_Outbound.esb**.

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

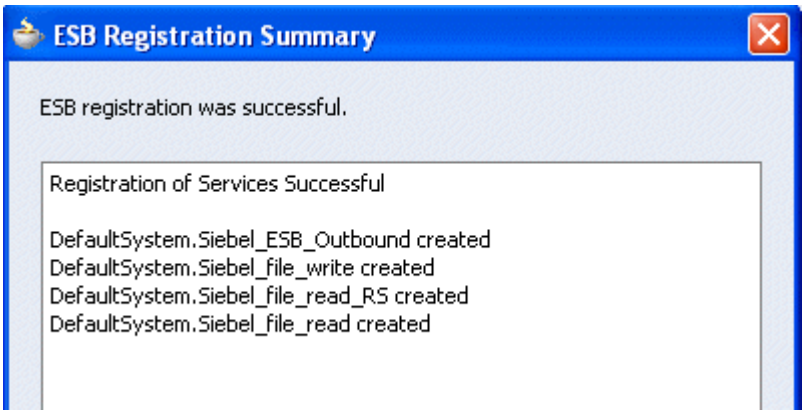


Deploying the Project

1. Right-click the created project, for example, **ESB_Outbound**, select **Register with ESB**, and the server connection, for example, **ServerConnection1**.



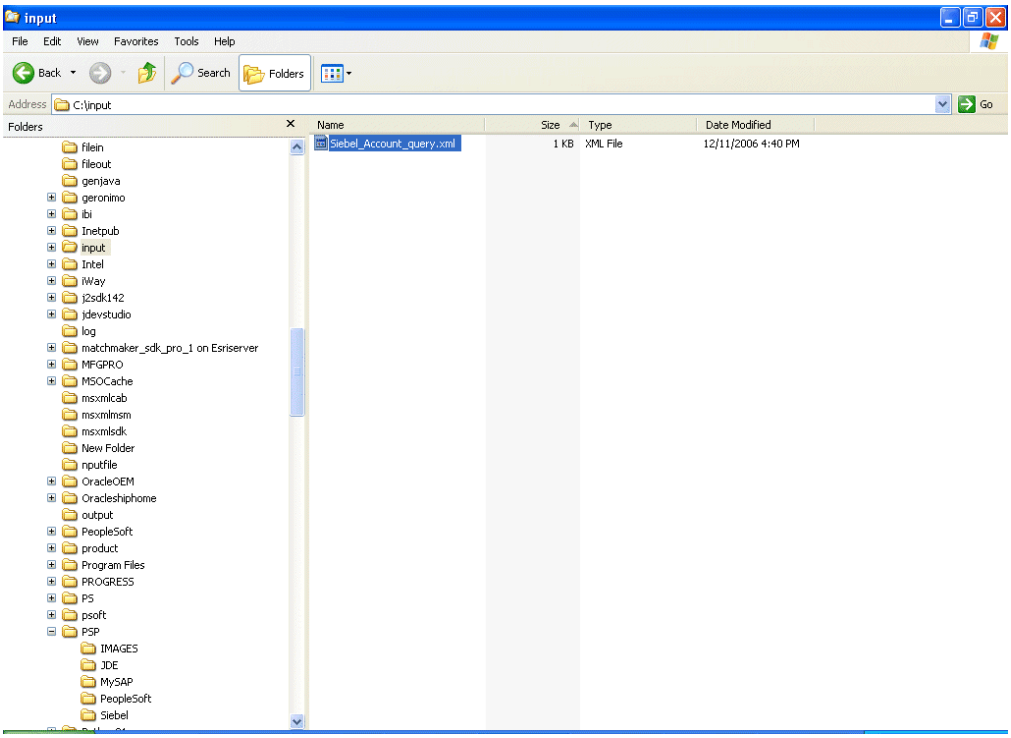
After successful deployment, the **Registration of services Successful** message is displayed.



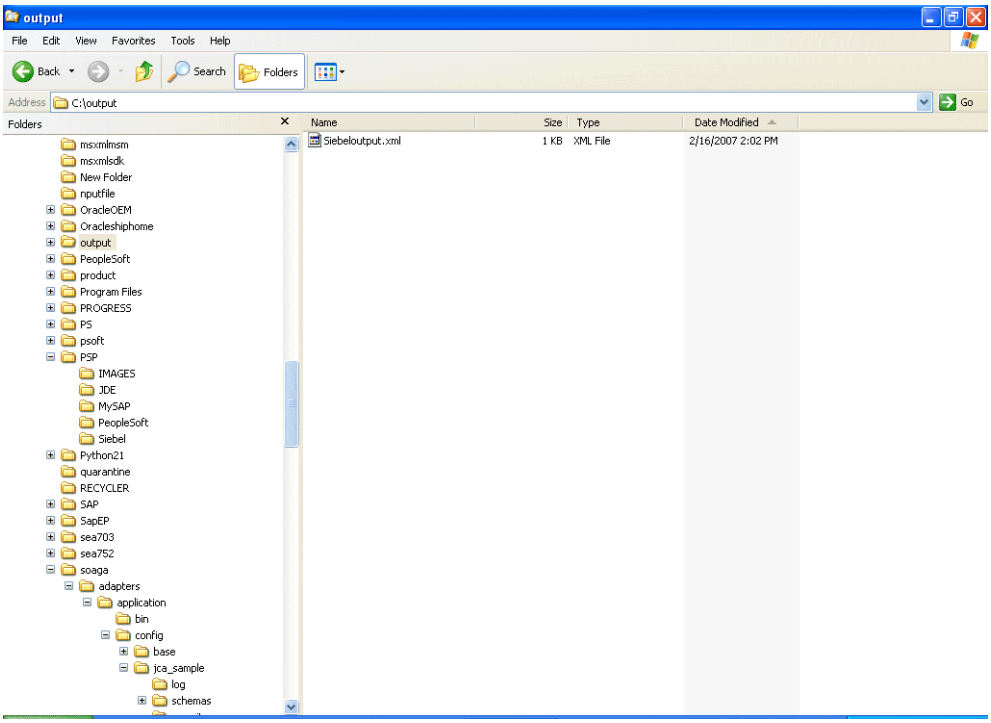
2. Logon to the ESB Control console to check whether the project has been successfully deployed.

The deployed process is listed under the **Default System** node.

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



4. Check whether you are receiving the response in the output folder, which you have specified during the creation of the write operation and also the corresponding instance in the ESB Control console.



5. If the response is not received in the output folder, check the instance and the logs for the corresponding errors in the ESB Control console.

Configuring an ESB Inbound Process

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

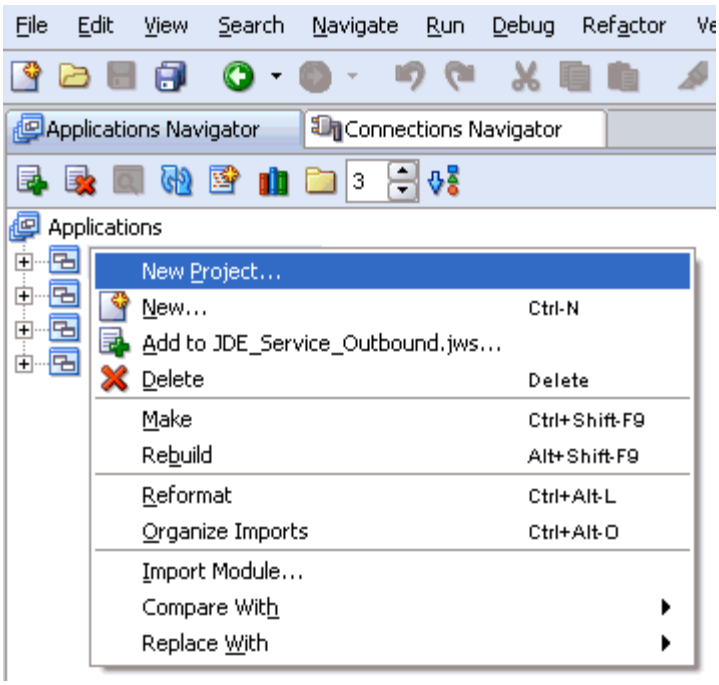
Prerequisites

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

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

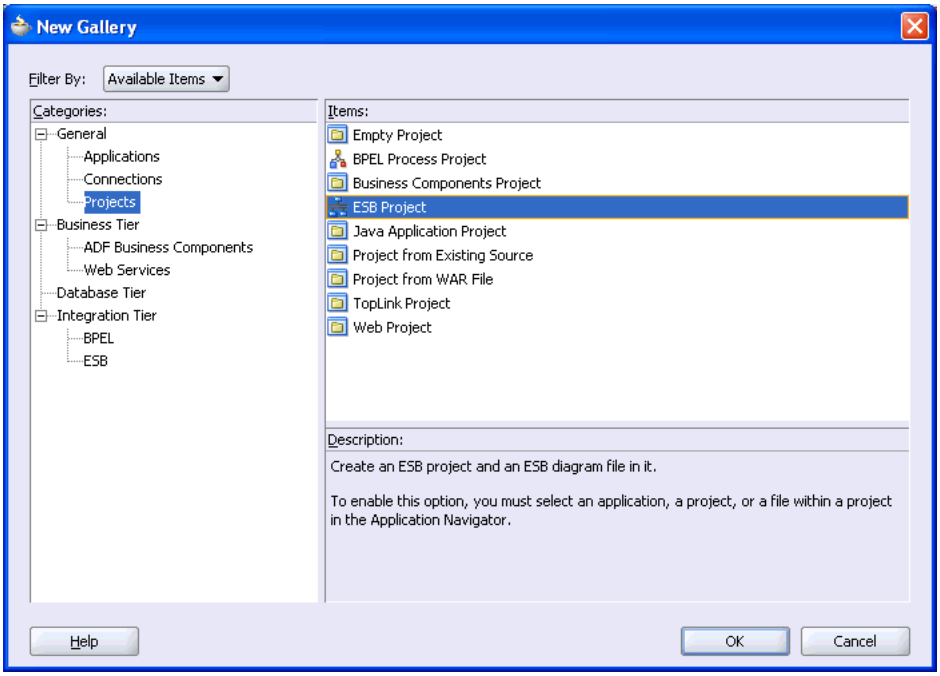
Creating an Inbound ESB Project and Assigning an Inbound WSDL File

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



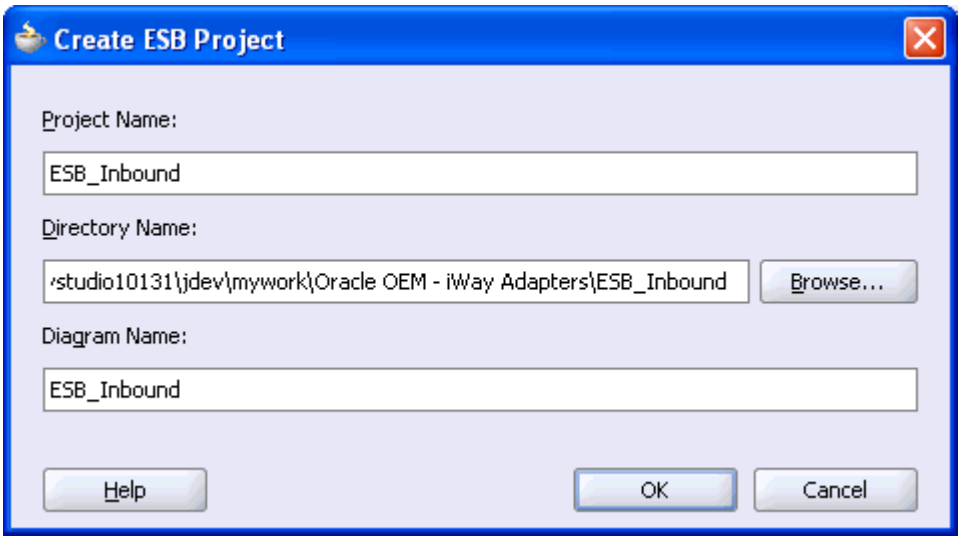
2. Right-click the application node you created and select **New Project**.

   The New Gallery window is displayed.



3. From the Items list, select **ESB Project Project** and click **OK**.

The Create ESB Project dialog box is displayed.



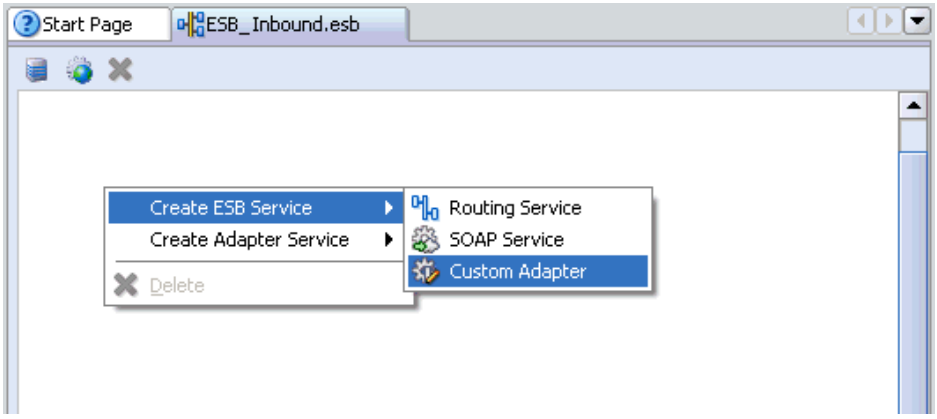
4. Perform the following steps:

   a. Specify a name for the ESB project.

      The Directory Name field and Diagram Name fields are updated automatically.

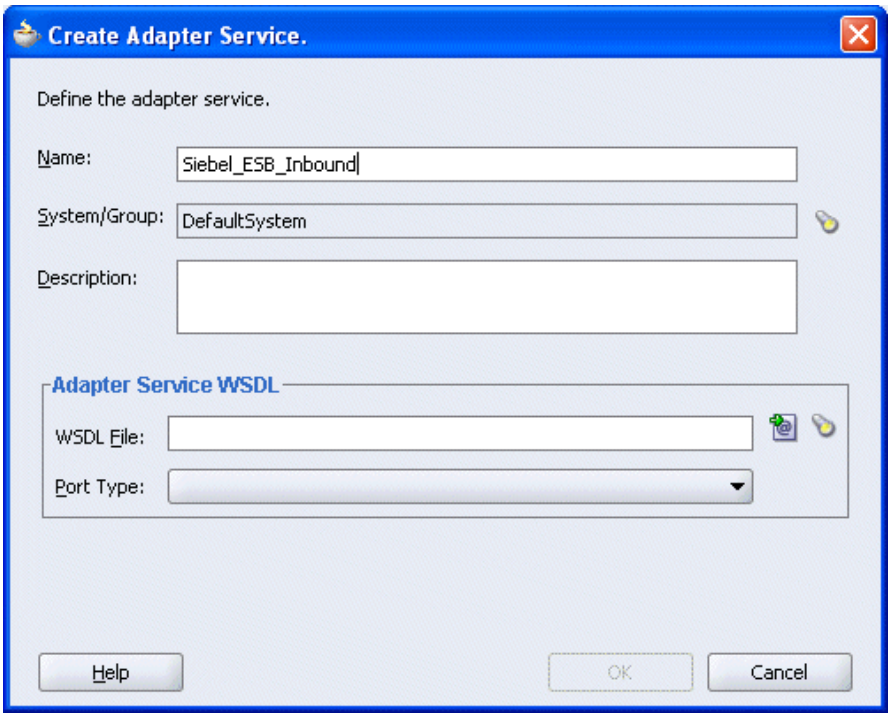
   b. Click **OK**.

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



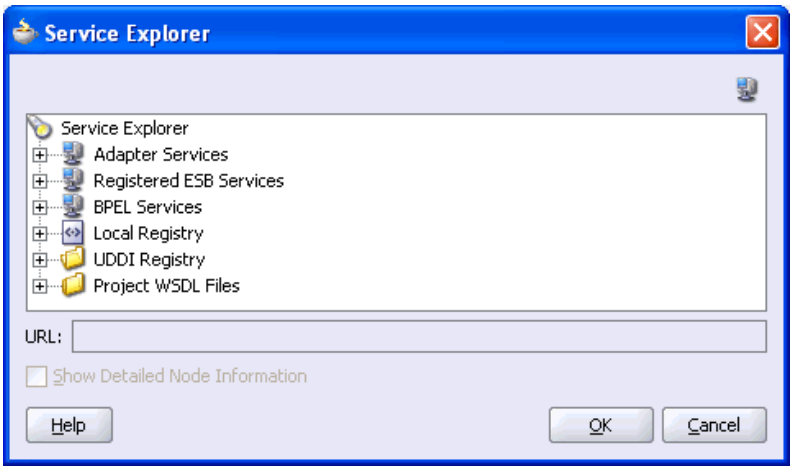
5. Right-click the ESB project in the middle pane, select **Create ESB Service** followed by **Custom Adapter**.

   The Create Adapter Service dialog box is displayed.



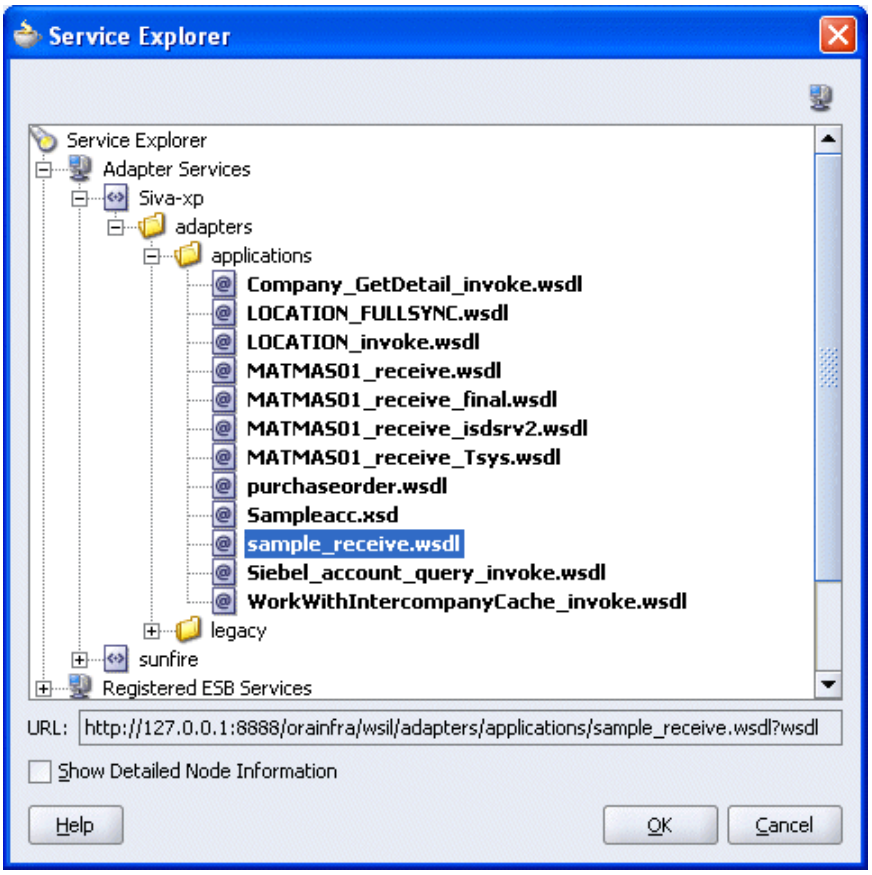
6. Enter a name for the adapter service and click the **Service Explorer** icon (second icon from the left preceding the **WSDL File** field).

   The Service Explorer dialog box is displayed.



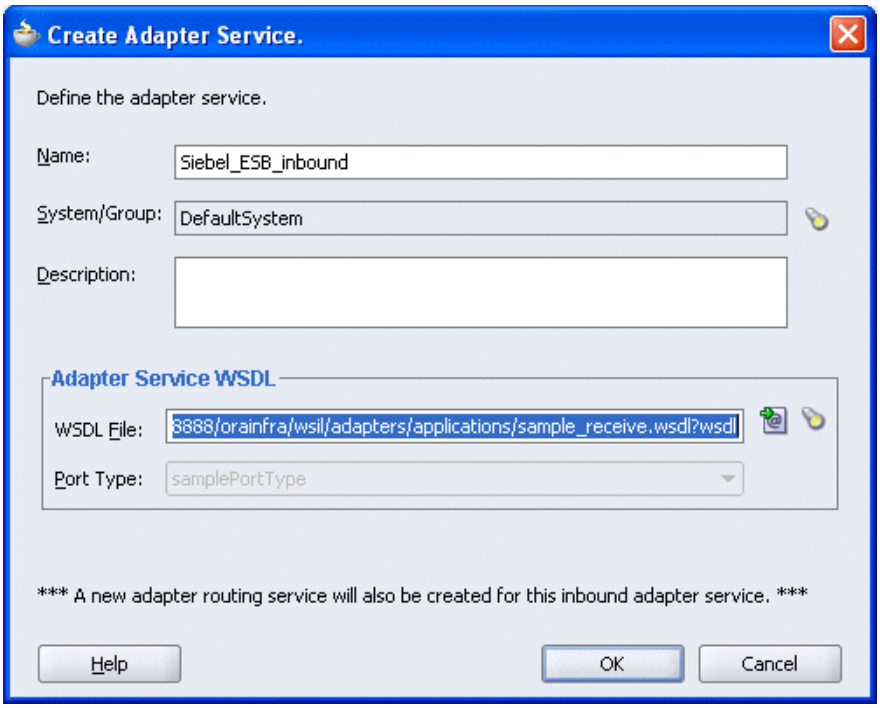
7. Expand your new connection under Adapter Services, followed by **adapters**, and then **applications**.

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



8. Select an inbound WSDL file that has been created using Application Explorer and click **OK**.

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

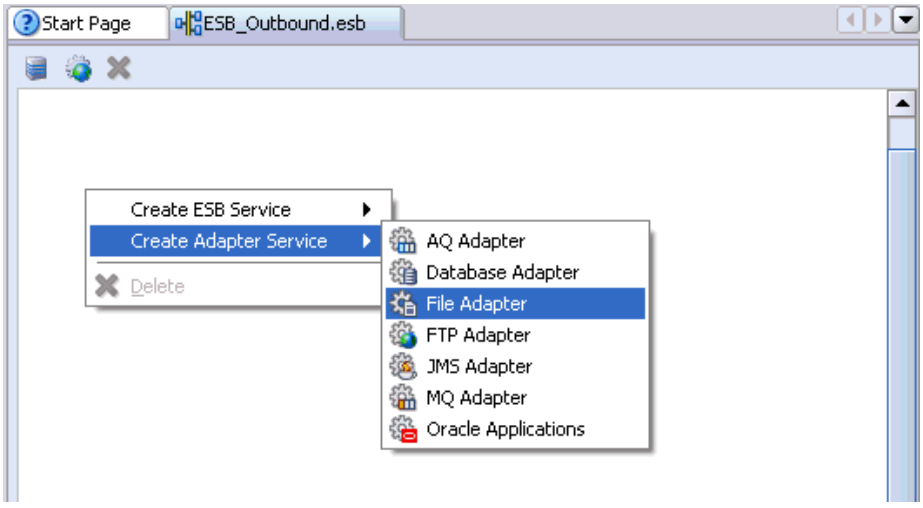


9. Click **OK**.

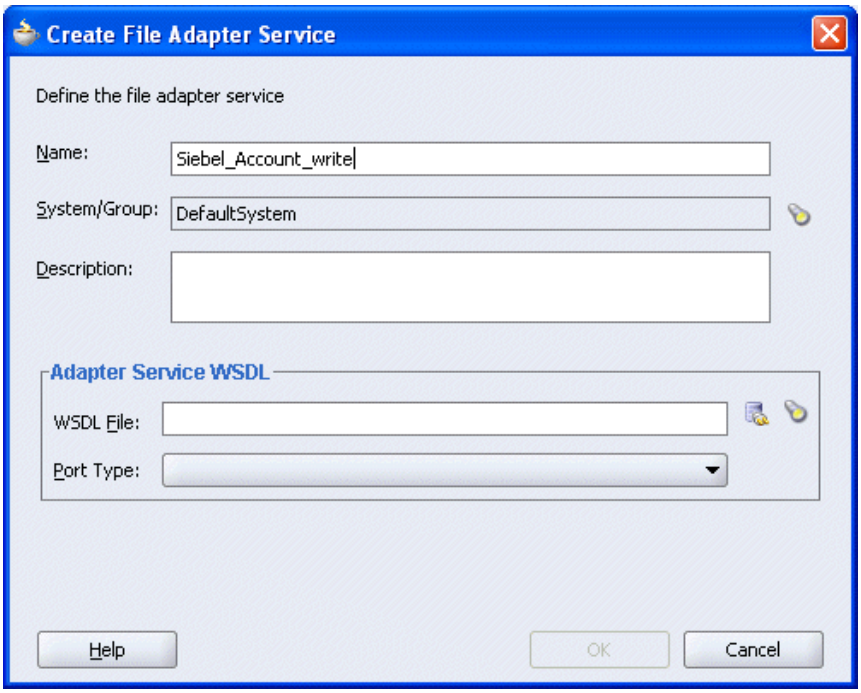
   The new ESB project appears in the visual editor.

Creating a Write Process Operation Using the File Adapter

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

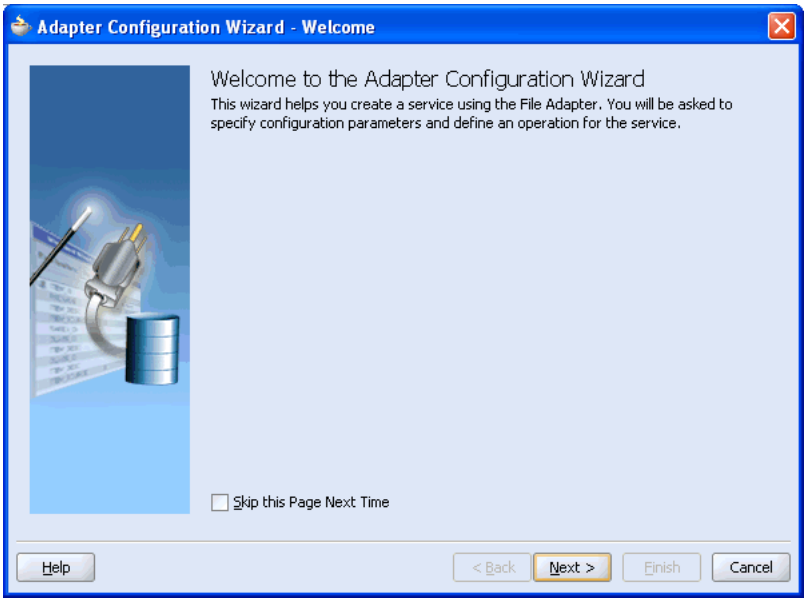


   The Create File Adapter Service dialog box is displayed.



2. Enter a name for the File adapter and click the **Configure adapter service wsdl** icon next to the **WSDL File** field.

   The Adapter Configuration Wizard - Welcome window is displayed.



3. Click **Next**.

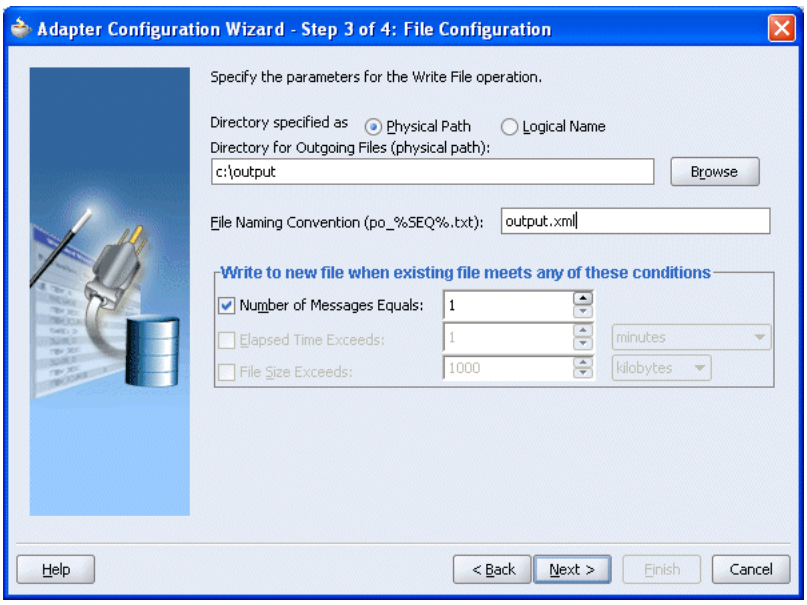
   The Adapter Configuration Wizard - Step 1 of 4: Service Name window is displayed.

4. Click **Next**.

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

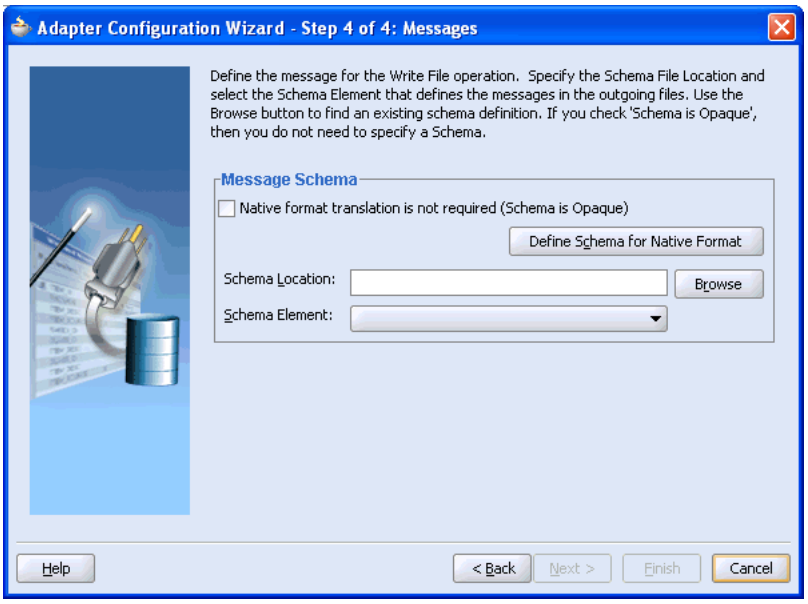
5. Click **Write File** as the Operation Type and click **Next**.

   The Adapter Configuration Wizard - Step 3 of 4: File Configuration window is displayed.



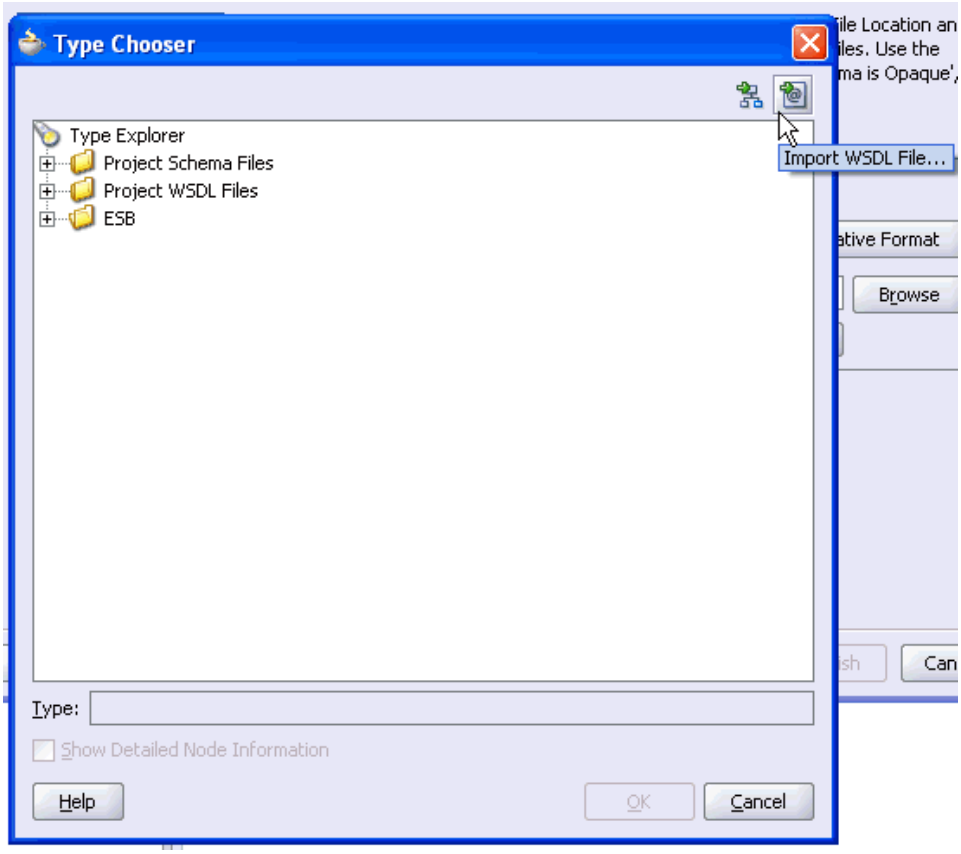
6. Enter the path of the output directory and name of the output file and click **Next**.

   The Adapter Configuration Wizard - Step 4 of 4: Messages window is displayed.



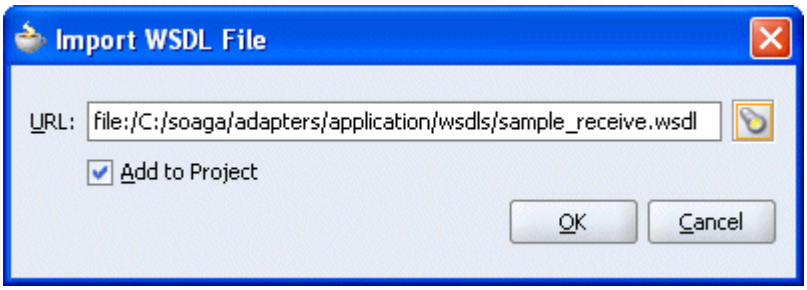
7. Click **Browse** to select the WSDL.

   The Type Chooser window is displayed.



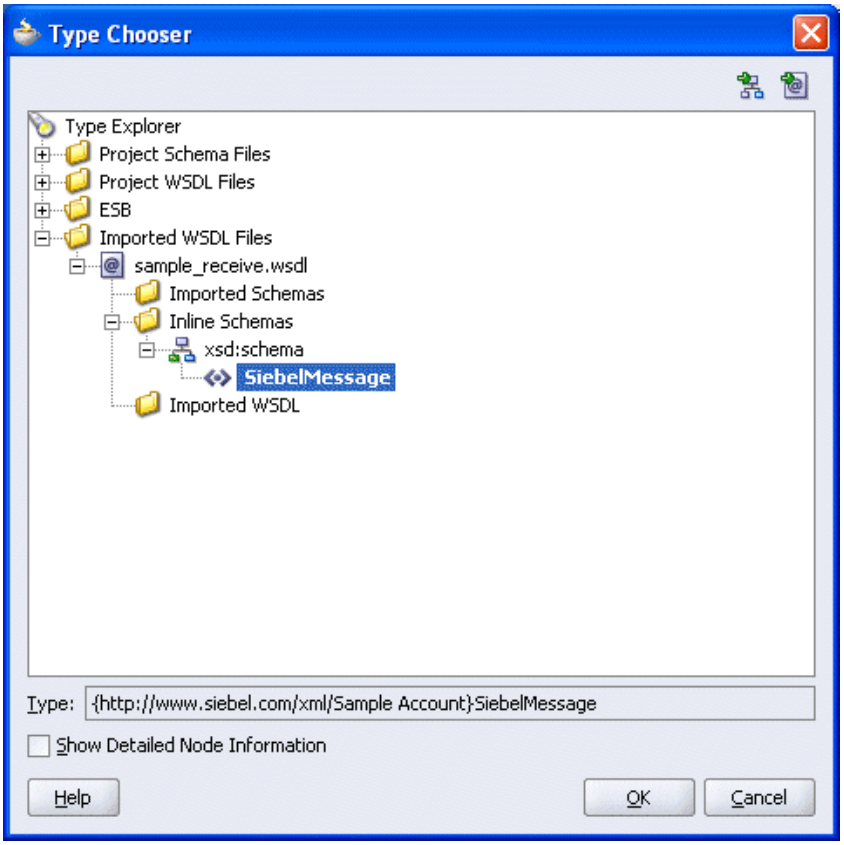
8. Click the **Import WSDL File** icon on the upper right corner of the dialog box.

   The Import WSDL File dialog box is displayed.



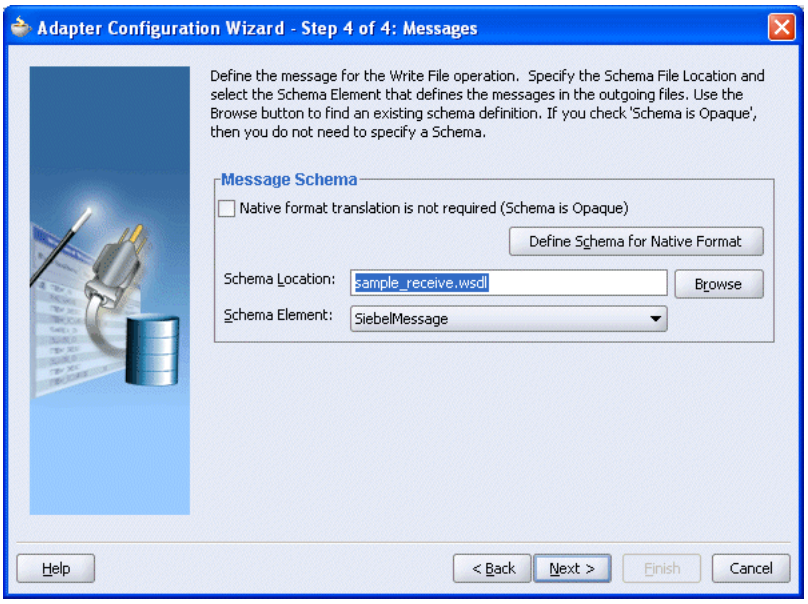
9. Select the WSDL file and click **OK**.

   The Imported WSDL Files folder is added.



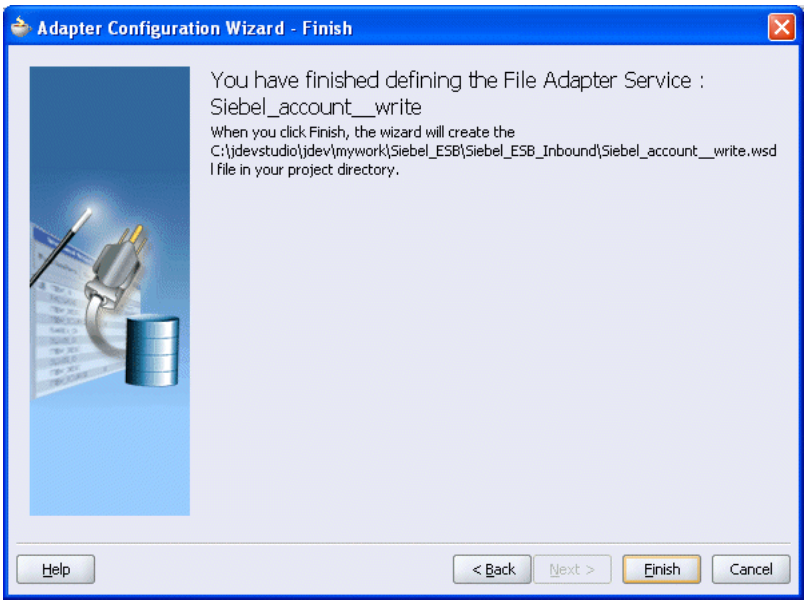
10. Expand the Imported WSDL Files folder, select an Inline Schema, for example, **DEBMAS05**, and click **OK**.

    You are returned to the Adapter Configuration Wizard - Step 4 of 4: Messages window.



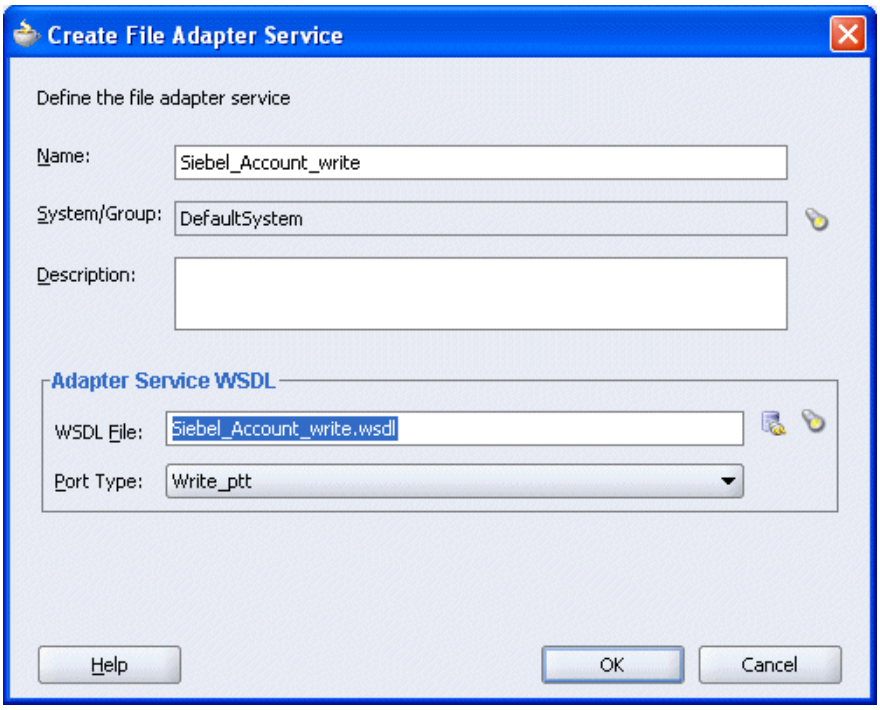
11. Click **Next**.

    The Adapter Configuration Wizard - Finish window is displayed.



12. Click **Finish**.

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

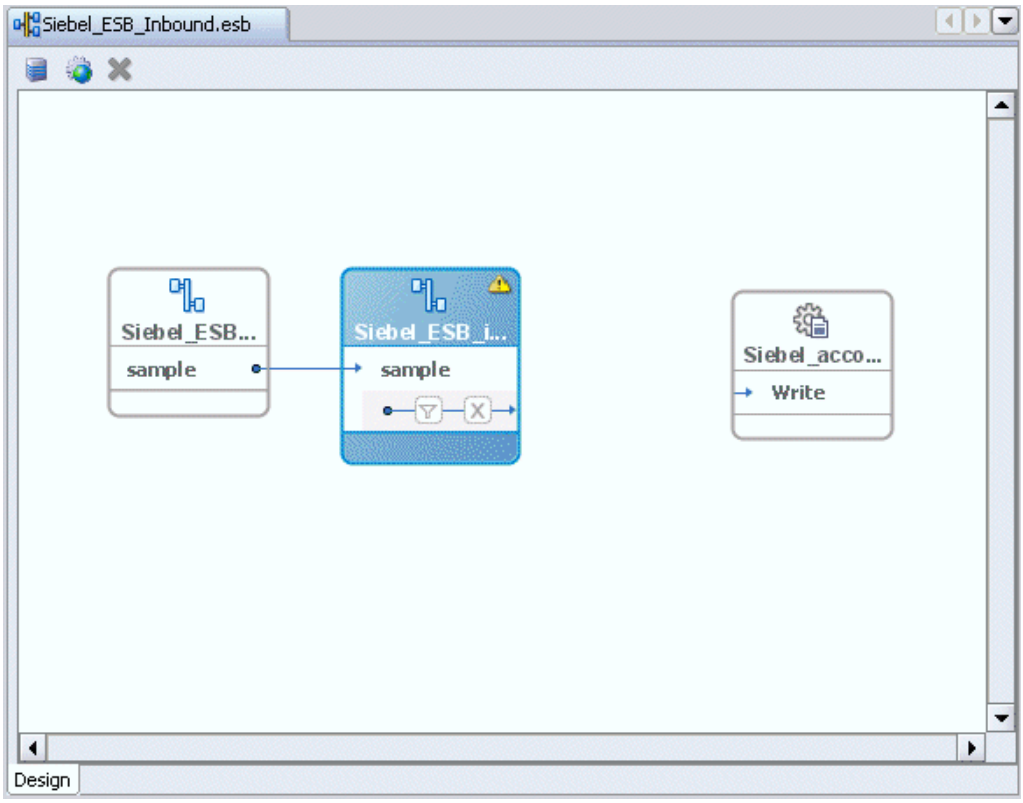


13. Click **OK**.

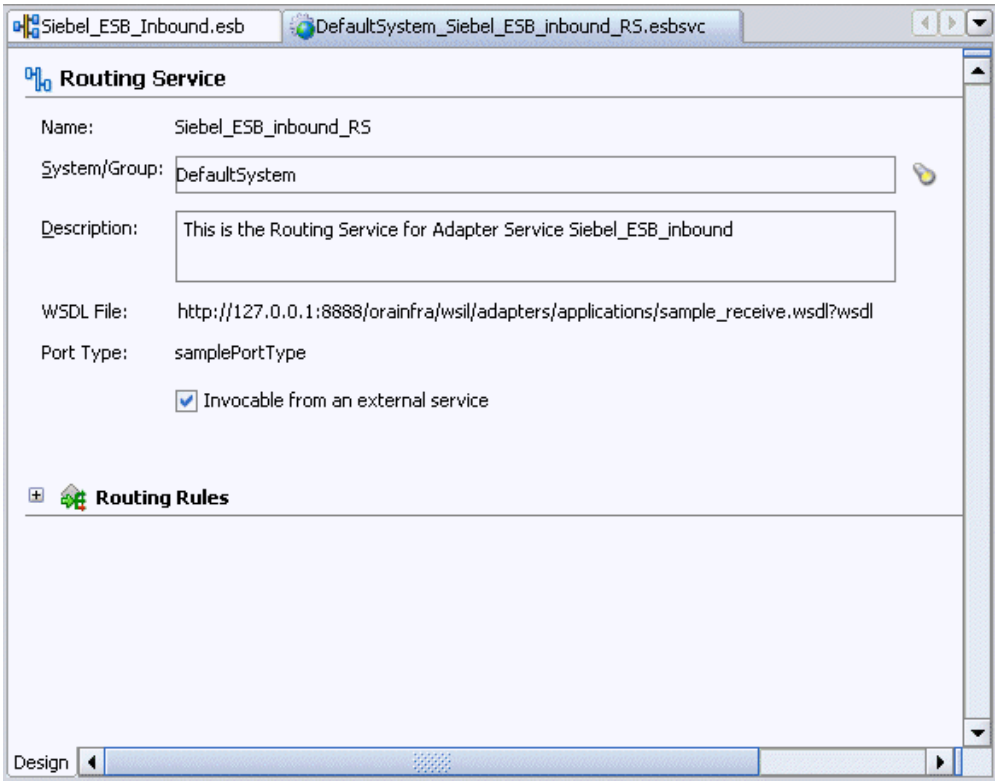
    The Write operation with a routing service is added to the ESB inbound project view.

Providing a Routing Service for the Write Operation

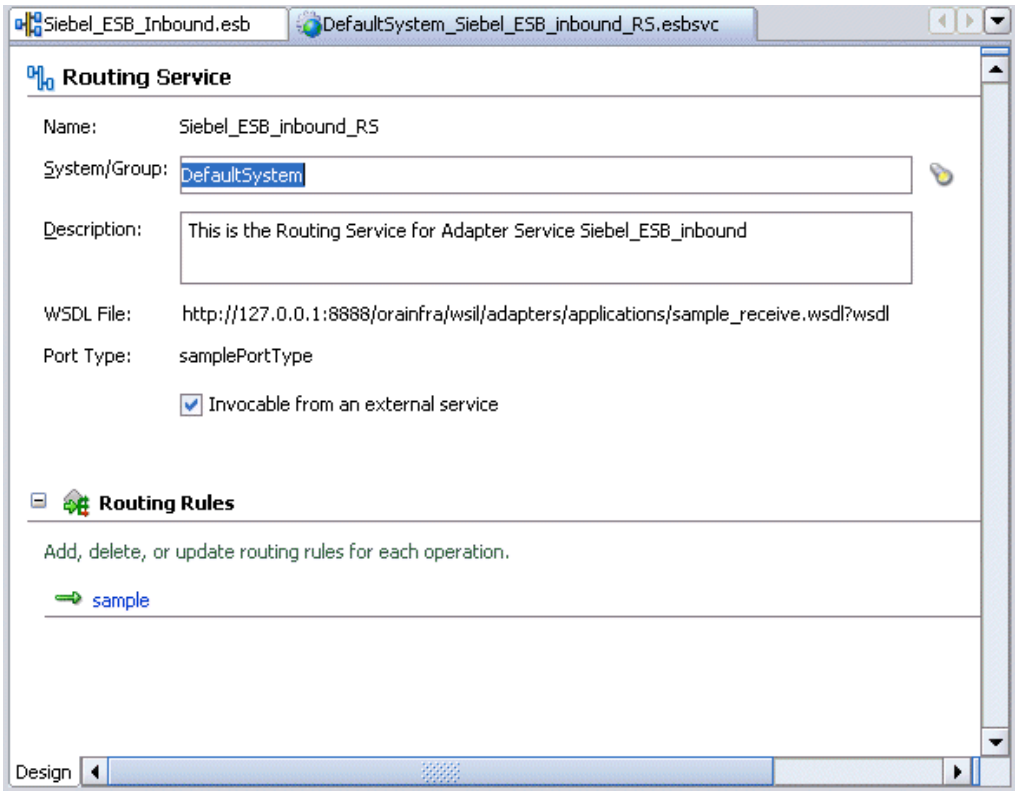
1. Double-click the routing service.



The Routing Service window is displayed.

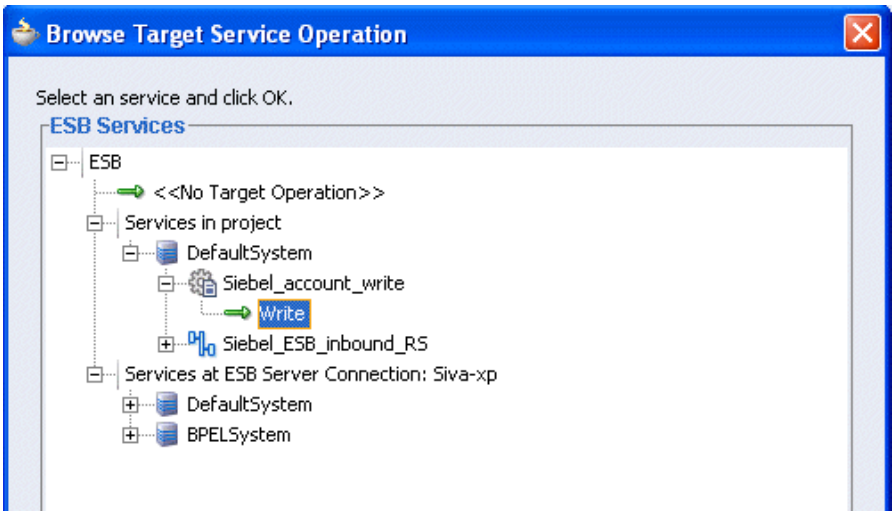


2. Expand the **Routing Rules**.



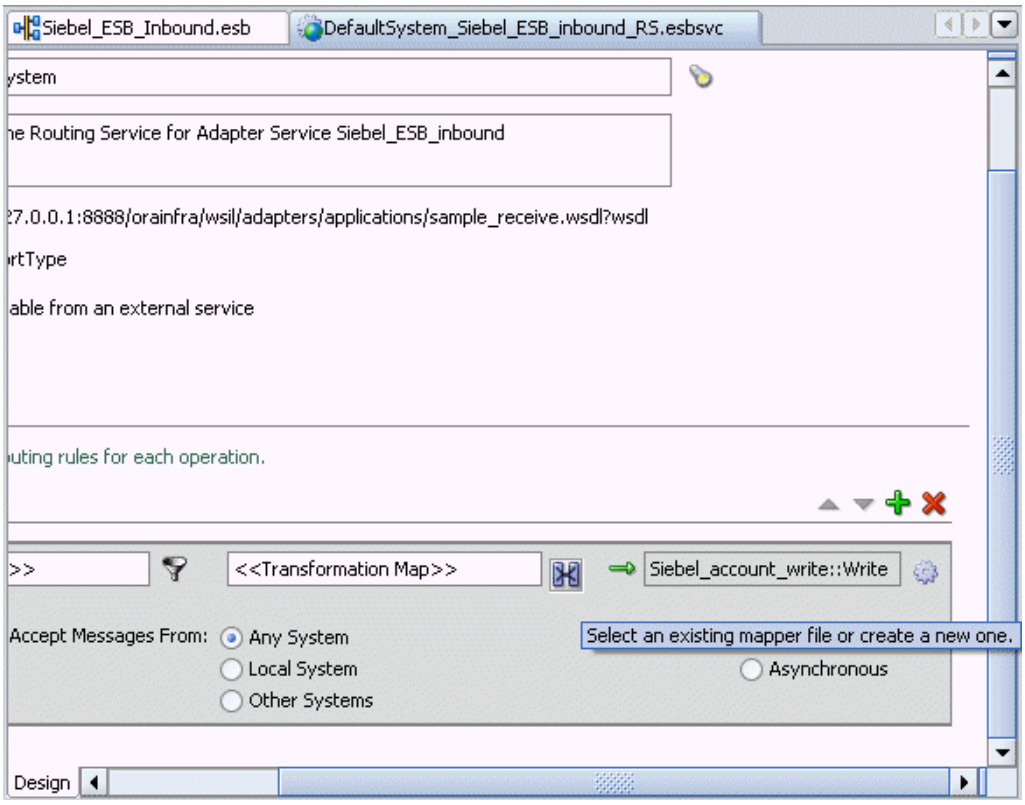
3. Click the green plus sign icon, which represents the option to **Create a new Routing Rule**.

   The Browse Target Service Operation window opens.



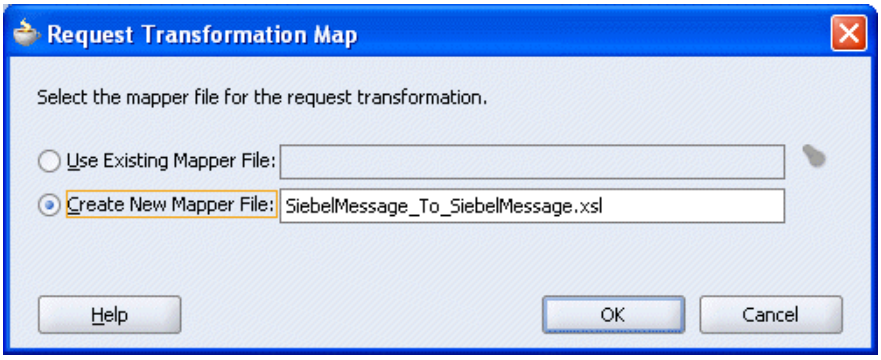
4. Expand **Services in project**, **Default System**, your adapter service node, for example, **ESB_write**, and select the service name, for example, **Write**.
5. Click **OK**.

   You are returned to the Routing Rules window.



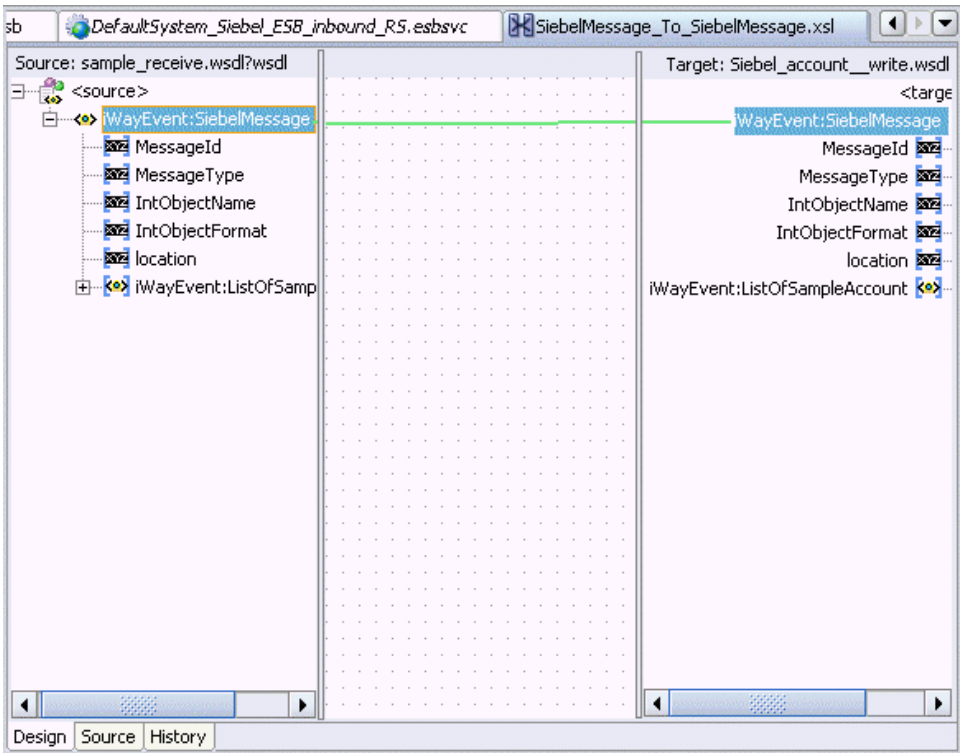
6. Click on the icon next to the <<Transformation Map>> field (**Select an existing mapper file or create a new one**).

   The Request Transformation Map dialog box is displayed.



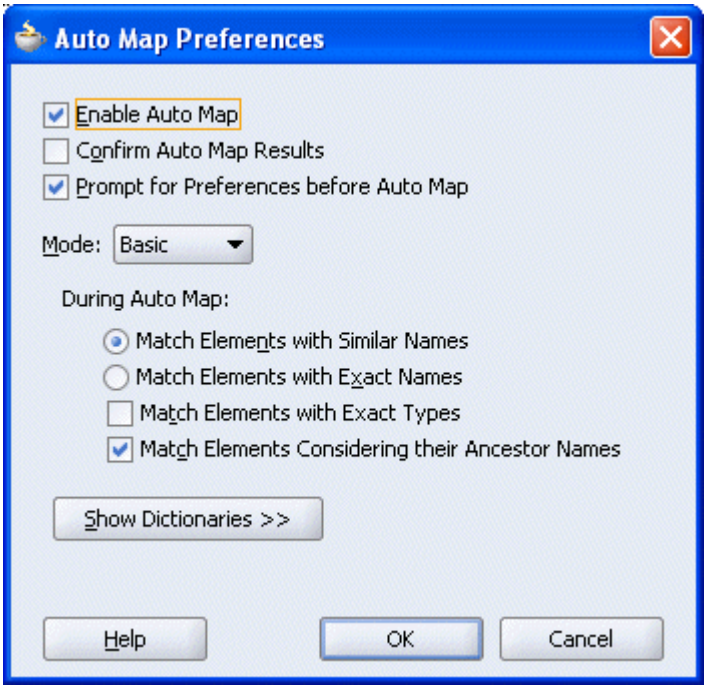
7. Select the **Create New Mapper File** option, specify the file name, and click **OK**.

   The following mapping window is displayed.



8. Select the WSDL file and map it to the Write operation.

   Once you map the WSDL file, the Auto Map Preferences dialog box is displayed.



9. Click **OK**.

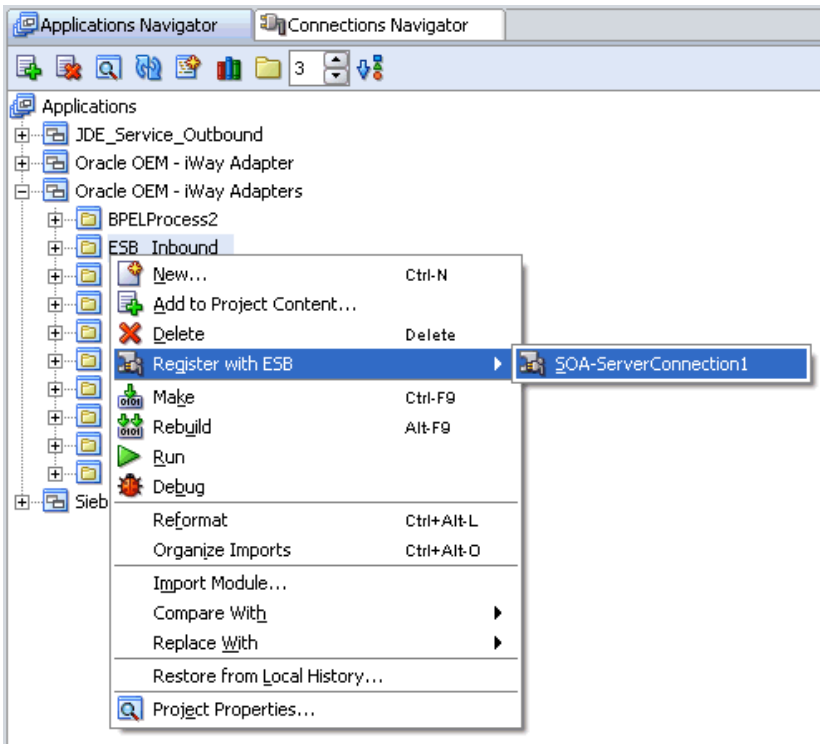
   The mapping is completed.

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

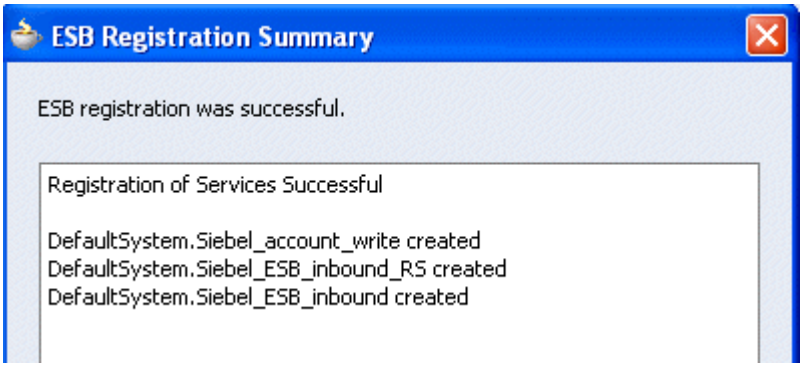
Notice that the Routing service is now created for the Write operation in the middle pane.

Deploying the Project

1. Right-click the created project, for example, **ESB_Inbound**, select **Register with ESB**, and the server connection, for example, **ServerConnection1**.

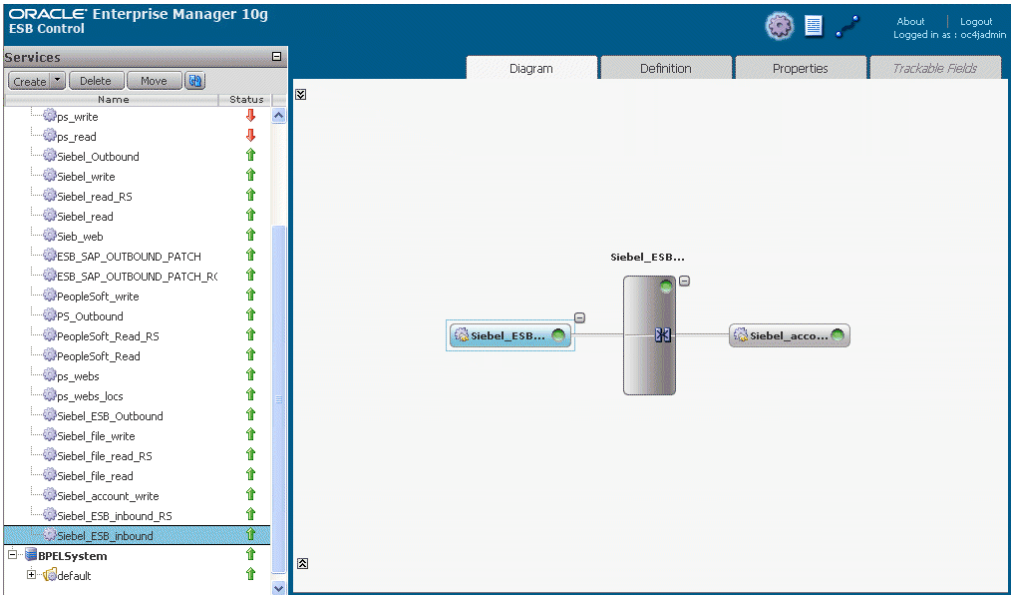


After successful deployment, the **Registration of services Successful** message is displayed.

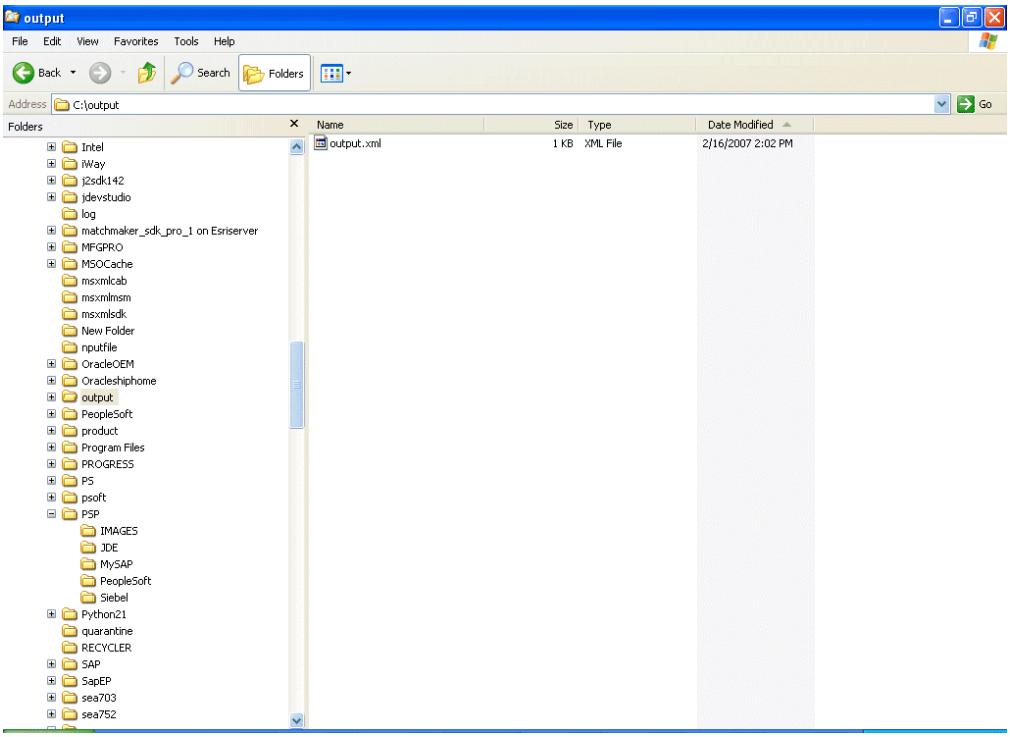


2. Logon to the ESB Control console to check whether the project has been successfully deployed.

The deployed process is listed under the **Default System** node.



3. Trigger the event.
4. Check whether you are receiving the response in the output folder, which you have specified during the creation of the write operation.



5. If the response is not received in the output folder, check the instance and the logs for the corresponding errors in the ESB Control console.

7

Troubleshooting and Error Messages

This chapter explains the limitations and workarounds when connecting to Siebel. The following topics are discussed:

Troubleshooting

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

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

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

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

General Usage Notes for the OracleAS Adapter for Siebel

The OracleAS 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, load the batch script `ae.bat`, found under:

OracleAS_home`\adapters\application\tools`

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

OracleAS_home`/adapters/application/tools`

Important Prerequisite:

Before starting OracleAS Adapter Application Explorer (Application Explorer) and using Oracle Application Server Adapter for Siebel (OracleAS Adapter for Siebel), you must create `\endorsed` directories under your *OracleAS_home* directory and place a copy of the `xalan.jar` file in those directories. Otherwise, you will receive a transformation error when adding an IO node under an Integration Object in Application Explorer. See "Starting Application Explorer" on page 2-1 for more information.

Error	Solution
Siebel does not appear in the Application Explorer Adapter node list.	Ensure that the Siebel jar files supplied with your Siebel distribution media have been placed in the *OracleAS_home*`\adapters\application\lib` directory. For example, for Siebel 7.03 environments, the `SiebelJI_Common.jar` and `SiebelJI_enu.jar` should be placed in the *OracleAS_home*`\adapters\application\lib` directory.
Target Type drop down contains only Java Data Bean Connection and COM connection type is desired.	Ensure that the Siebel thin client is installed correctly on the machine hosting Application Explorer so that appropriate COM environment is available.
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 (<server_name>). Check logs for more information.`	Ensure that the name of the Siebel Gateway server is correctly defined for the target you are using.
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.`	Ensure that the User ID and password parameter values to connect to your Siebel system are correct.
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.`	Check on network connectivity to Siebel environment. Correct networking problem and retry connection.

Error	Solution
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.`	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
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.SSAMessages_enux). Check logs for more information`	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).
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.	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.
Logon failure error at runtime.	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 will be 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.
The following exception occurs when you start Application Explorer by activating `ae.bat` (not `iaexplorer.exe`): `java.lang.ClassNotFoundException: org.bouncycastle.jce.provider.BouncyCastleProvider`	This is a benign exception. It does not affect adapter functionality. Download BouncyCastle files from: `ftp://ftp.bouncycastle.org/pub`

Error	Solution
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)` `Could not create the connection factory.`	`JAVACMD` is not set on the user system. Before starting Application Explorer, export `JAVACMD` as follows: `JAVACMD=/<jdk_home>/bin/java`, where `<jdk_home>` is the directory where JDK is installed on your machine.

Siebel

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

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.	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.
When executing a request, the following error message appears: `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: `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.

Error	Solution
When connecting to releases prior to Siebel 7.7 using the Java Data Bean Interface, you cannot reconnect after initial connection loss. This might occur when Application Explorer experiences a brief loss of network connection or if the Siebel Server or Gateway Service is restarted while Application Explorer is logged into the Siebel application.	Restart OC4J and Application Explorer in order to log in successfully to the Siebel application. This is a known Siebel API issue. See Siebel Alert 984 for more information.
The following error may occur when adding a service node for a Business Service that includes methods containing method arguments having hierarchy data types. If you enter a valid `XMLCharEncoding` value such as UTF-8 or UTF-16, you will get the following error: `Invocation of Service failed.`	The method argument `XMLCharEncoding` is not supported. Leave this element blank in the XML payload.

OracleAS Adapter J2CA

Error	Solution
In Application Explorer, the following error message appears when you attempt to connect to an OracleAS Adapter J2CA configuration: `Could not initialize JCA`	In the Details tab in the right pane, ensure that the directory specified in the Home field points to the correct directory, for example, *`OracleAS_home`*\adapters\application

BPEL Process Manager

Error	Solution
Endpoint activation error on deployment of Siebel event handling project (inbound) in JDeveloper	Verify that the channel used for this inbound J2CA service is stopped in Application Explorer. If you have started this channel for testing or debugging purposes, you must stop it before starting BPEL PM Server. Endpoint activation is managed by BPEL Process Manager.

Error	Solution
The following error message appears in BPEL PM Server Console: `Process "TestSiebel" (revision "1.0") compilation failed. <2005-05-18 10:49:53,285> <ERROR><default.collaxa.cube.engine.deployment> <Cube ProcessLoader::create> Failed to read wsdl. Error happened when reading wsdl at "http://127.0.0.1:7777/BPELConsole/wsil/adapters/applications/QueryWithView_invoke.wsdl?wsdl", because "WSDLException: faultCode=INVALID_WSDL: Invalid XML in document at: http://127.0.0.1:7777/BPELConsole/wsil/adapters/applications/QueryWithView_invoke.wsdl?wsdl: The element type "P" must be terminated by the matching end-tag "</P>".`	Verify that the specified WSDL file exists at that URL and that the file is valid. Workaround: Change the WSDL location to `localhost:7777`. The default is `127.0.0.1:7777`. Alternative workaround: Add the IP address to the `Dhttp.nonProxyHosts` list found in `obsetenv.bat` (Windows) or `obsetenv.sh` (Unix)
Second message invocation fails at runtime.	Verify that you have all the required patches installed. The required patches are listed and updated on the Oracle Technology Network Web site, http://www.oracle.com/technology/index.html.
The following exception is thrown in JDeveloper during deployment of the BPEL process: `java.io.FileNotFoundException: \BPELConsole\wsil\adapters\applications\SampleAccount_receive.wsdl?wsdl (The system cannot find the path specified)`	Verify that you have all the required patches installed. The required patches are listed and updated on the Oracle Technology Network Web site, http://www.oracle.com/technology/index.html

BSE Error Messages

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

General Error Handling in BSE

BSE serves as both a SOAP gateway into the adapter framework and as the engine for some of the adapters. In both design time and runtime, various conditions can cause errors in BSE when Web services that use adapters are running. 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 request, a SOAP fault element is generated in the SOAP

response. The SOAP fault element contains fault string and fault code elements. The fault code contains a description of the SOAP agent error.

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

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

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

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

Adapter-Specific Error Handling

When an adapter raises an exception during runtime, 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 or not an exception is raised depends on how the target systems interface or API treats the error condition. If a SOAP request message is passed to an adapter by the SOAP agent in BSE, and that request is invalid based on the WSDL for that service, the adapter may raise an exception yielding a SOAP fault.

While it is almost impossible to anticipate every error condition that an adapter may encounter, the following is a description of how adapters handle common error conditions and how they are then exposed to the Web services consumer application.

OracleAS Adapter for Siebel Invalid SOAP Request

If OracleAS 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:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<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 OracleAS Adapter for Siebel cannot connect to Siebel when executing a Web service, then the following SOAP response is generated.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
 <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></faultstring>
  </SOAP-ENV:Fault>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Oracle Application Server Integration Adapters

OracleAS 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 Application Server Integration Adapter when running Web services.

Invalid SOAP Request

If OracleAS 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:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
<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 Application Server Adapter Request

If OracleAS 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.

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

8

Advanced User Tools

This chapter includes the following topics:

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. This enables easy application integration but raises the issue of controlling the use and implementation of critical and sensitive business logic that is run as a Web service.

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 will not be governed by it. However, if a policy is applied to the iBS, all methods are governed by it. At runtime, 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.

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.

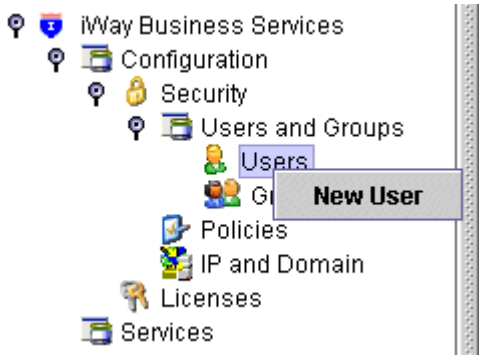
1. Start Application Explorer.
2. Right-click the configuration to which you want to connect, for example, SampleConfig. See Chapter 2, "Configuring Oracle Application Server Adapter for Siebel" for information on creating a new configuration.
3. Select **Connect**.

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

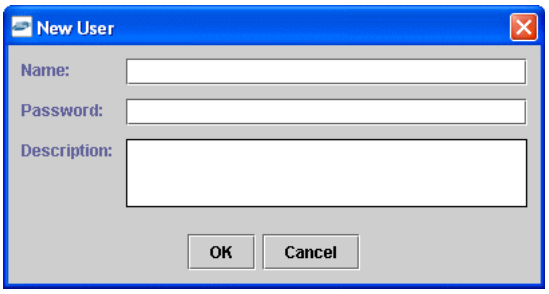
   Perform the following steps:

   a. Expand the **Business Services** node.
   b. Expand the **Configuration** node.
   c. Expand the **Security** node.
   d. Expand the **Users and Groups** node.

4. Right-click **Users** and 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**, **Events**, and **Business Services** (also known as Web services).

Perform the following steps:

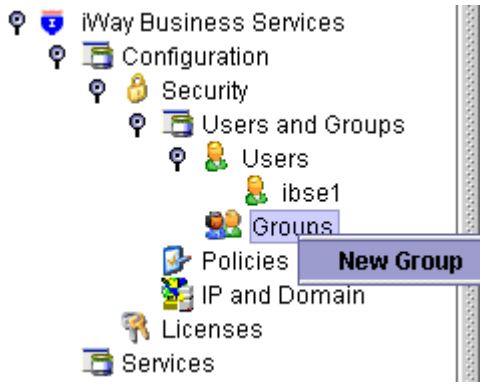
a. Expand the **Business Services** node.

b. Expand the **Configurations** node.

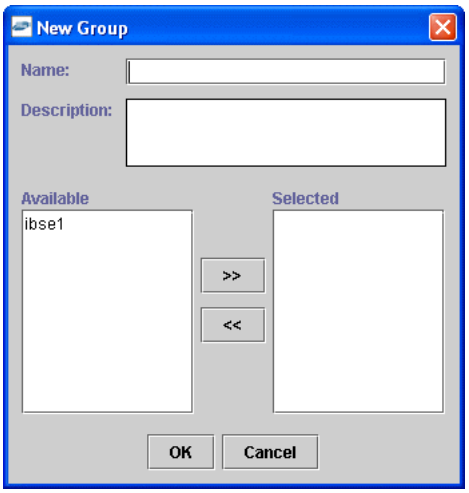
c. Expand the **Security** node.

d. Expand the **Users and Groups** node.

4. Right-click **Groups** and select **New Group**.



The New Group dialog box is displayed.



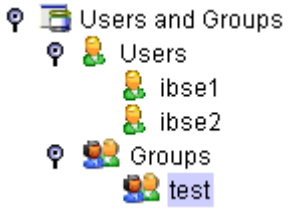
Perform the following steps:

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**.

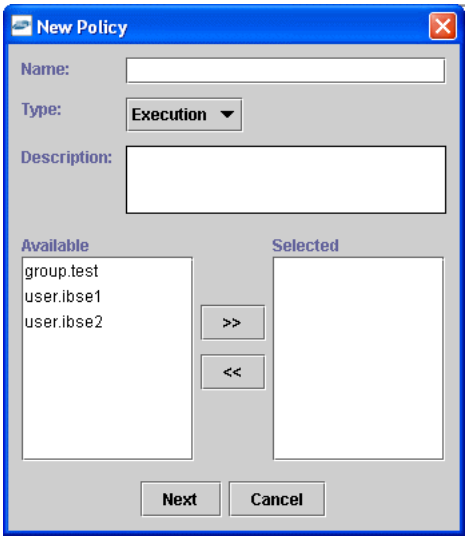
   Nodes appear for **Adapters**, **Events**, and **Business Services** (also known as Web services).

   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.

- 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.

7. Click **OK**.

   The following pane summarizes your configuration.

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

Using the IP and Domain Restrictions Policy Type

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

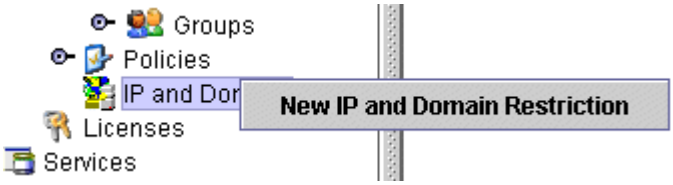
1. Start Application Explorer.
2. Right-click the configuration to which you want to connect, for example, SampleConfig. See Chapter 2, "Configuring Oracle Application Server Adapter for Siebel" for information on creating a new configuration.
3. Select **Connect**.

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

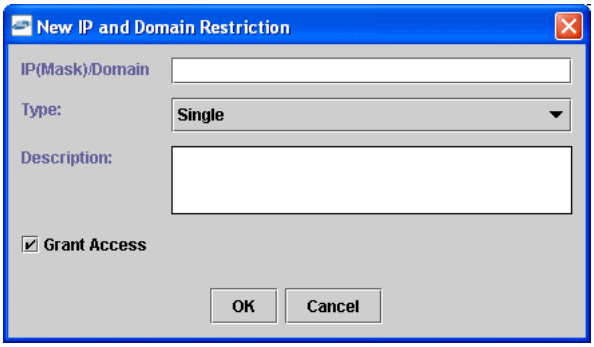
   Perform the following steps:

   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.



   Perform the following steps:

   a. In the **IP(Mask)/Domain** field, enter the IP or domain name using the following guidelines:

      If you select **Single** (Computer) from the **Type** list, you must provide the IP address for that computer. If you only know the DNS name for the computer, click **DNS Lookup** to obtain the IP Address based on the DNS name.

If you select **Group** (of Computers), you must provide the IP address and subnet mask for the computer group.

If you select **Domain**, you must provide the domain name.

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

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

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

5. Click **OK**.

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

The following pane summarizes your configuration.

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

Migrating Repositories

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

Migrating a BSE Repository

To migrate a BSE repository:

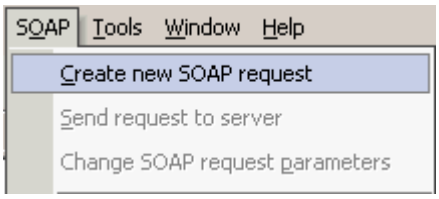
1. Copy the BSE control service URL, for example:

```
http://localhost:7777/ibse/IBSEServlet/admin/iwcontrol.ibs
```

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

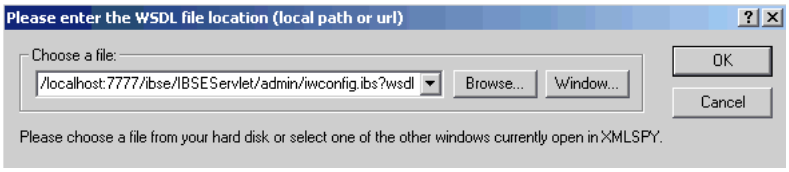
3. From the menu bar, click **SOAP**.

A list of options appears.



4. Select **Create new SOAP request**.

The WSDL file location dialog box is displayed.



Perform the following steps:

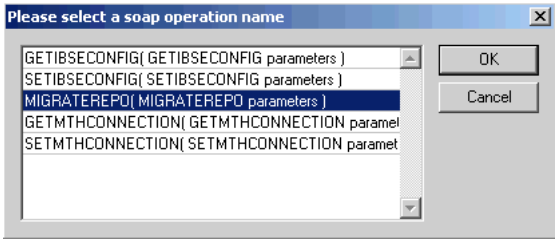
a. In the **Choose a file** field, paste the BSE control service URL.

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

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

5. Click **OK**.

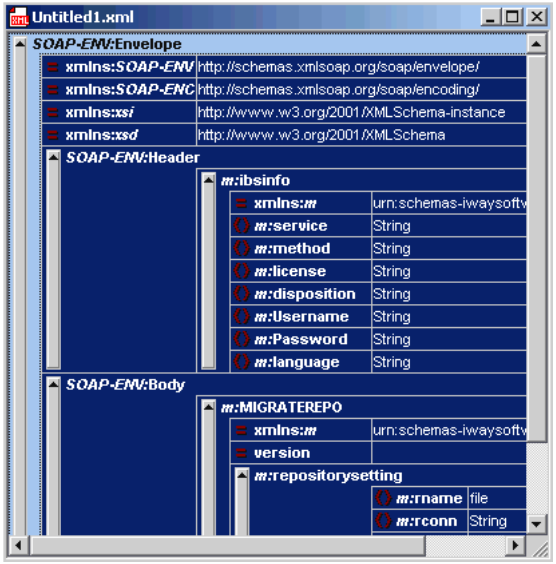
The soap operation name dialog box displays the available control methods.



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:

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

   Perform the following steps:

   a. For the <m:rconn> tag, replace the String placeholder with a repository URL where you want to migrate your existing BSE repository.

      The Oracle repository URL has the following format:

      ```
      jdbc:oracle:thin:@[host]:[port]:[sid]
      ```

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

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

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

10. Perform one of the following migration options.

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

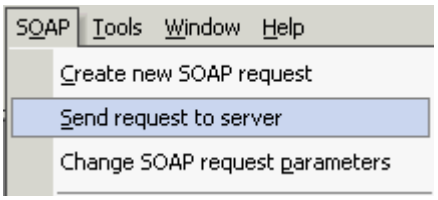
     ```
     <m:servicename>SiebelService1</m:servicename>
     ```

   - 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:

   OracleAS_home\adapters\application\config*JCA_CONFIG*

   Where `JCA_CONFIG` is the name of your J2CA configuration.

2. Locate and copy the `repository.xml` file.

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

   Your J2CA repository is migrated to the new J2CA configuration directory.

A

Using Siebel Workflows

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 OracleAS 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.

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.

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 runtime 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.

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.

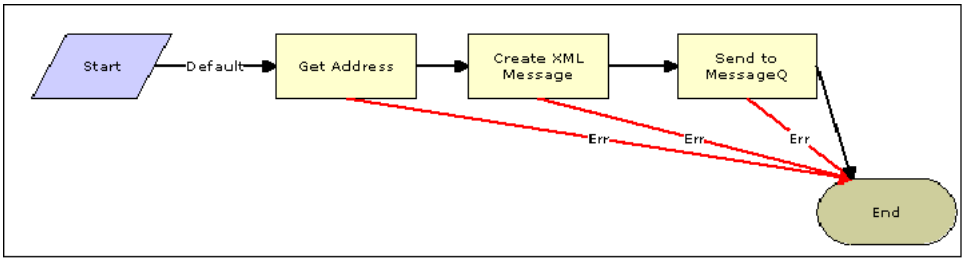
Siebel Workflow - Outbound

When a Siebel Workflow is triggered based on a Siebel policy, runtime, 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 OracleAS Adapter for Siebel 6.3 and higher. The following Workflow sequence illustrates the previous steps.



Siebel Workflow - Inbound

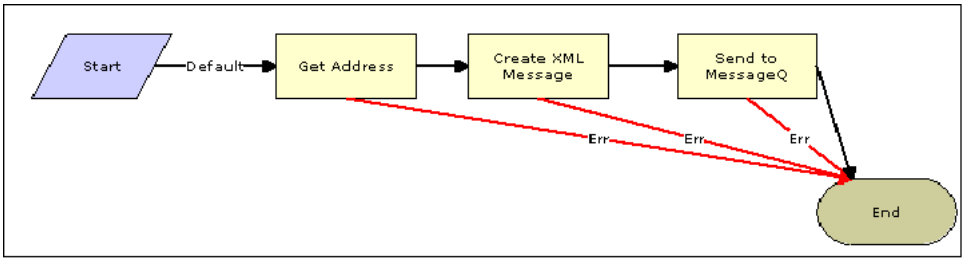
A Siebel Workflow that is triggered by an external event begins by receiving a Siebel XML document placed on one of its transports. The result might be the update of a Siebel record using the XML as input, for example, when a new account is added in another CRM system but also must be updated in the 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.



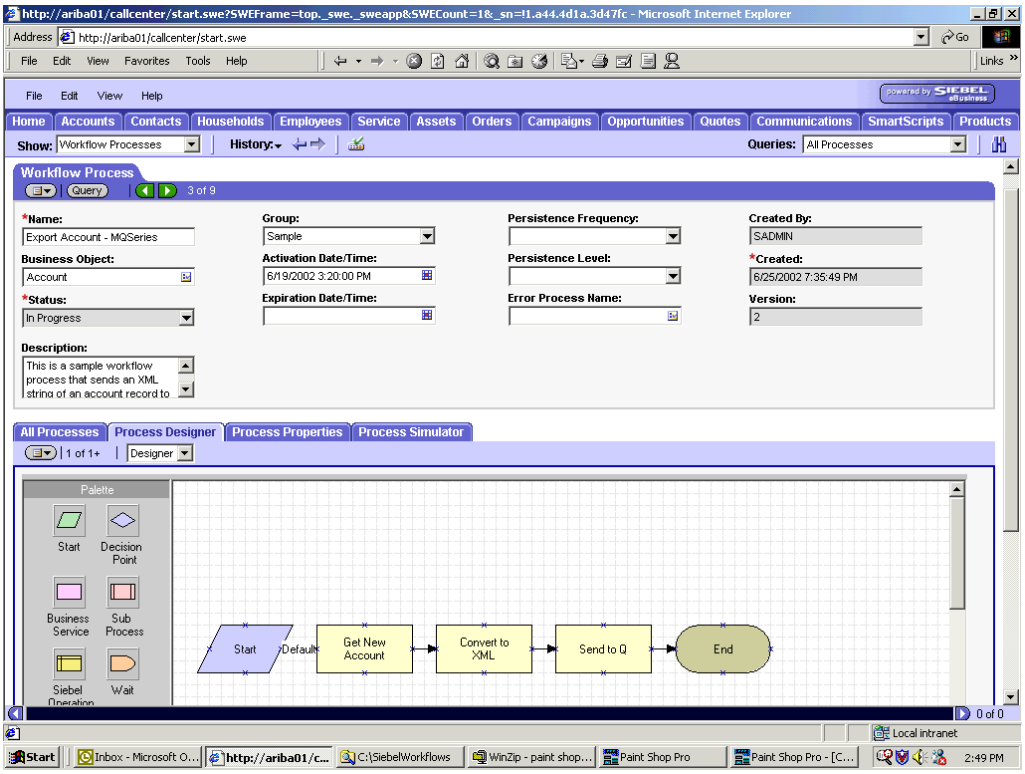
Creating a Siebel Workflow

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

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.

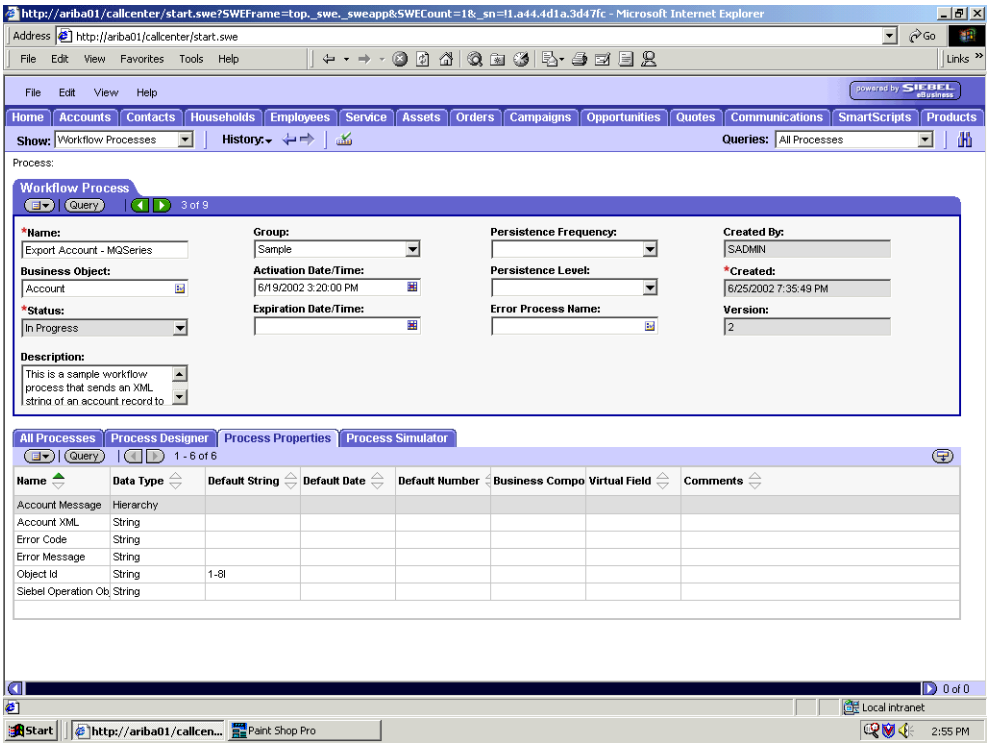
To create a Siebel Workflow:

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.

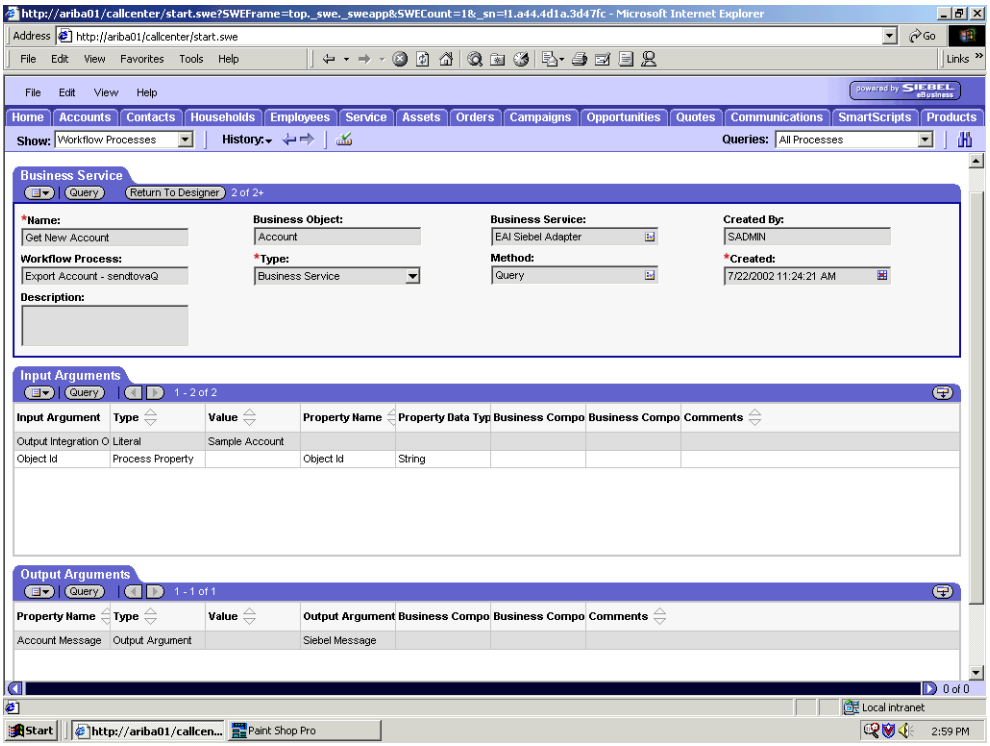
The following window is displayed, showing the Process Properties tab active.



2. Use the **Siebel Workflow Administration** windows to create a Workflow.
3. Define an **EAI Siebel Adapter Business Service** step to receive an instance of Account data and call it `Get New Account`.

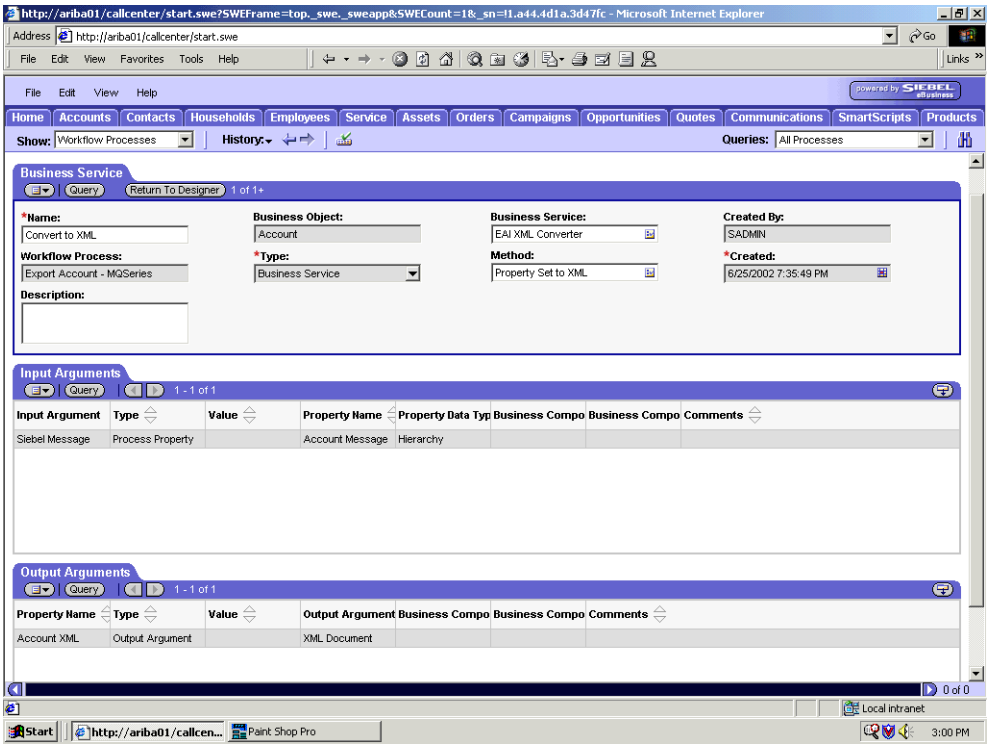
   The Business Service obtains the Account information from Siebel using the Query method.

Output from this Business Service is generated in hierarchical format.



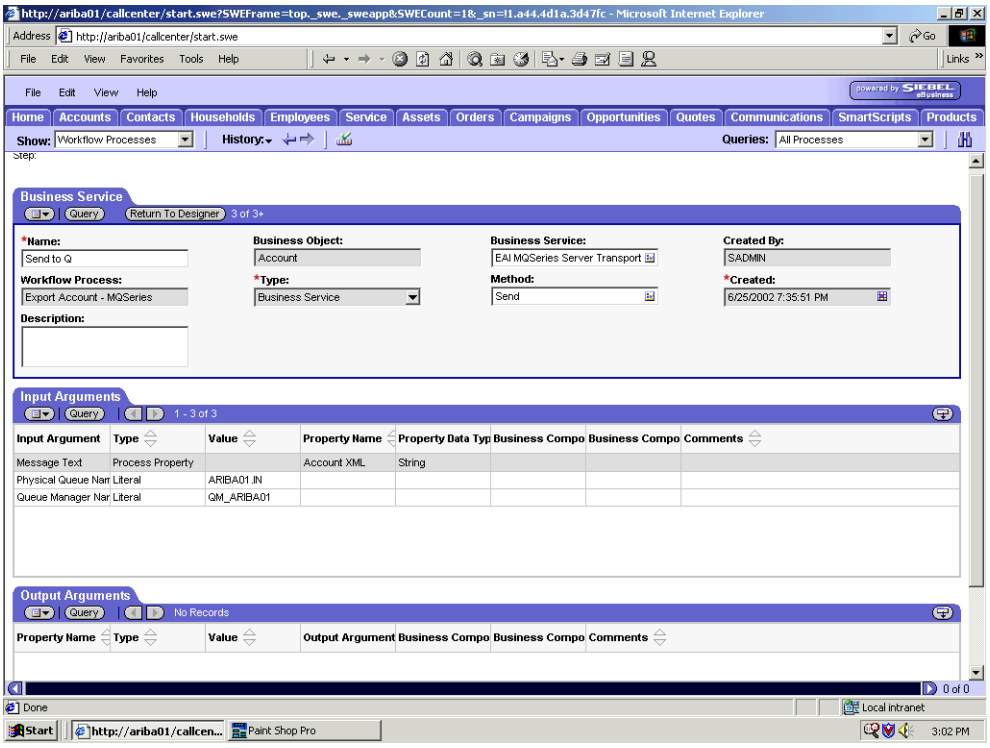
4. Define an **EAI XML Converter Business Service** step and call it `Convert to XML`.

   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`.

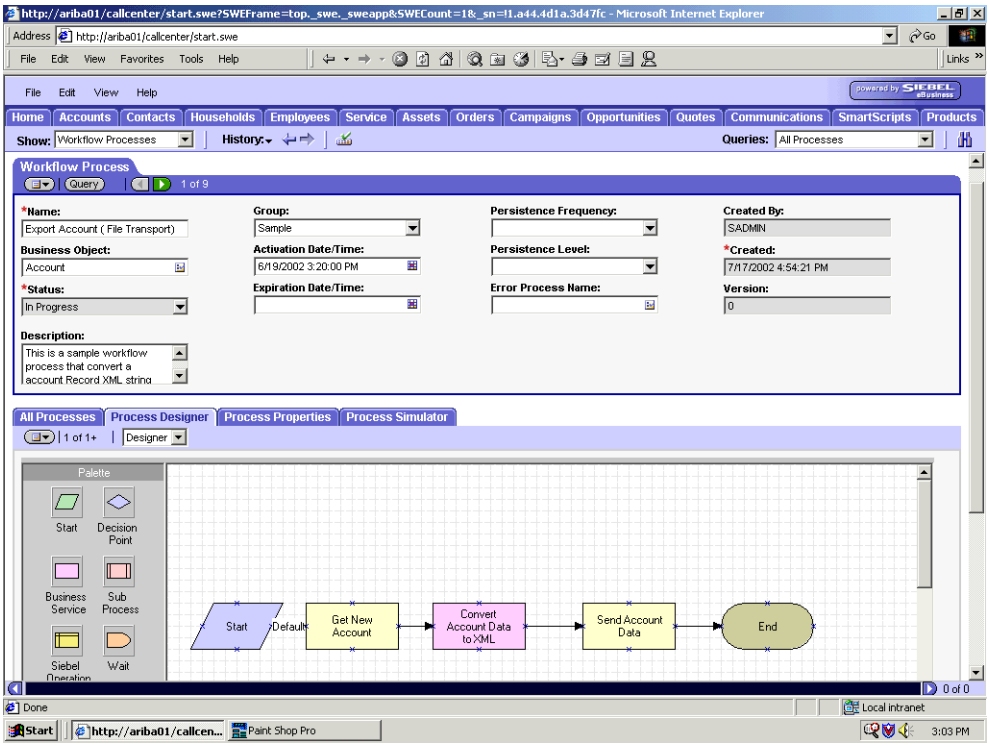
   It is defined to receive the Account data from the EAI XML Converter Business Service in Siebel XML format and send the Account XML to MQSeries using the Send method.



Creating a Siebel Workflow for an Event Using File Transport

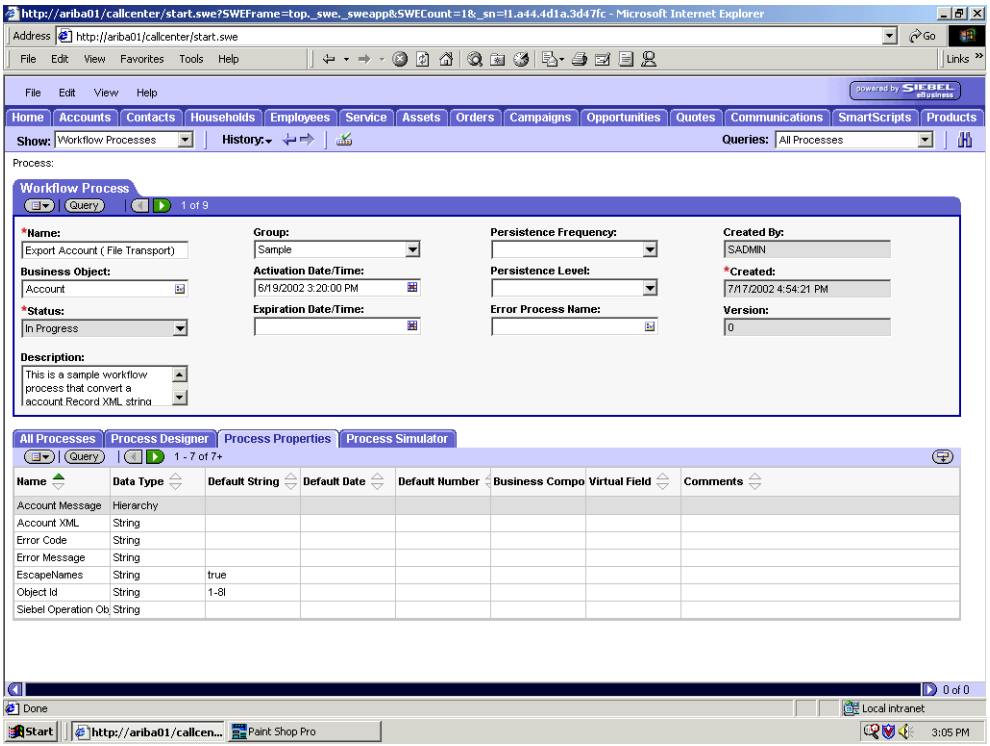
The following procedure is an example of a Siebel Workflow illustrated in the Siebel Workflow Administration window. The Workflow was designed for exporting Siebel Account record information using the File transport.

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:



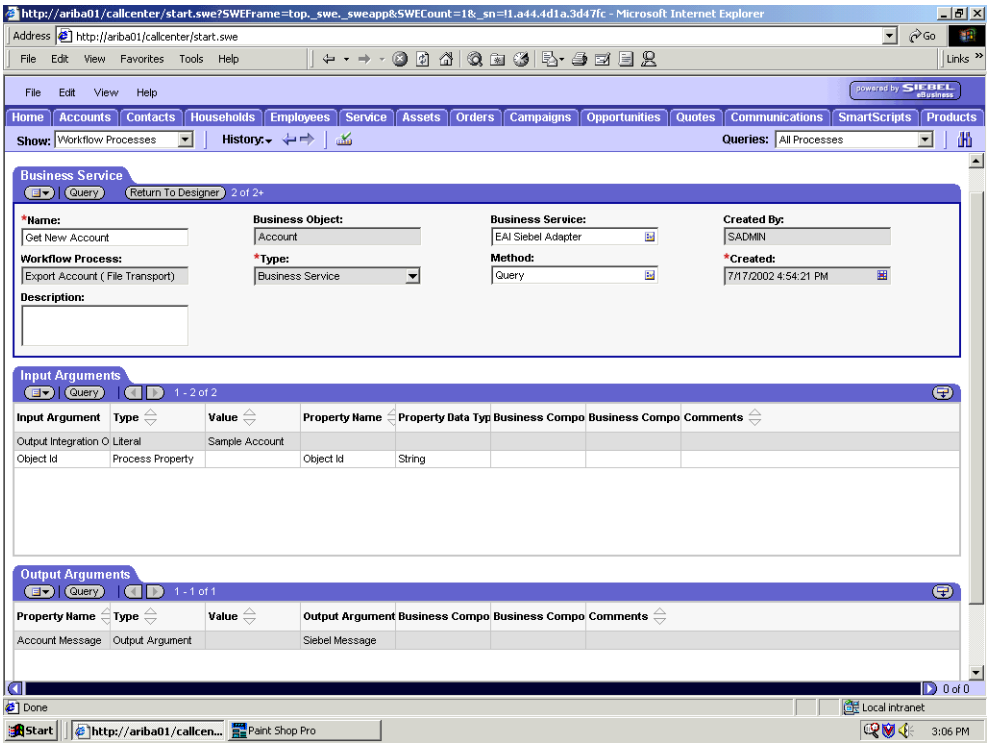
1. On the **Process Properties** tab of the **Workflow Process** window, define the Account message and Account XML process properties.

   Account message contains the Siebel Account data in hierarchical format.

   Account XML specifies which Siebel Account data the Workflow converted to XML.

2. Use the **Siebel Workflow Administration** windows to create a Workflow.

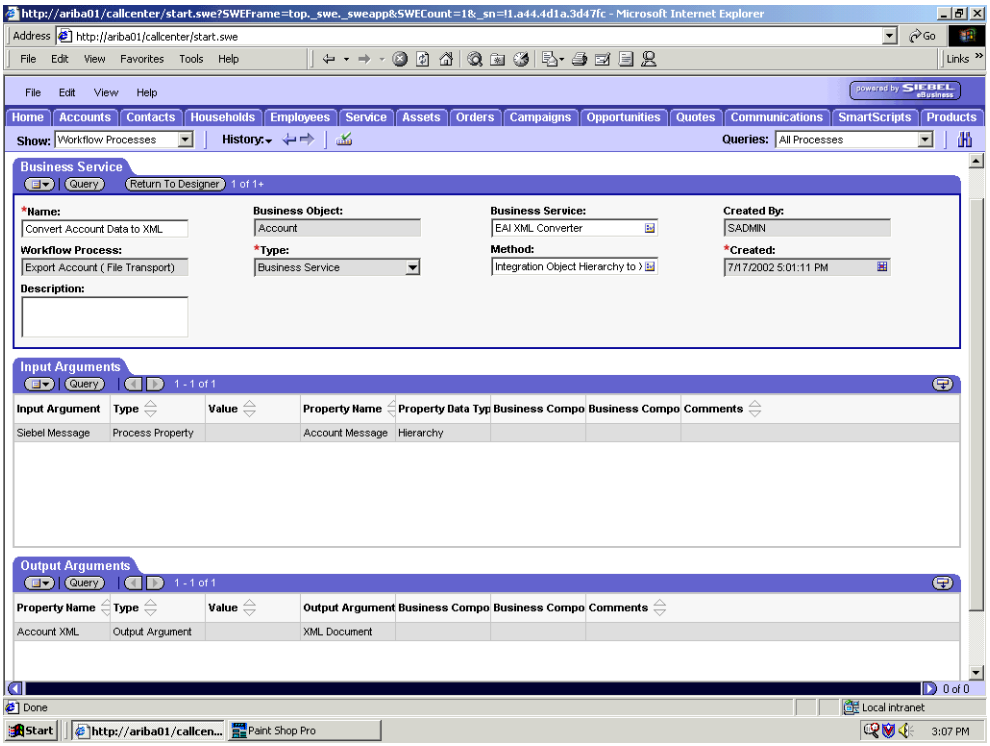
   The following is an example of a Siebel Workflow Administration window.



3. Define an **EAI Siebel Adapter Business Service** step to receive an instance of Account data and call it `Get New Account`.

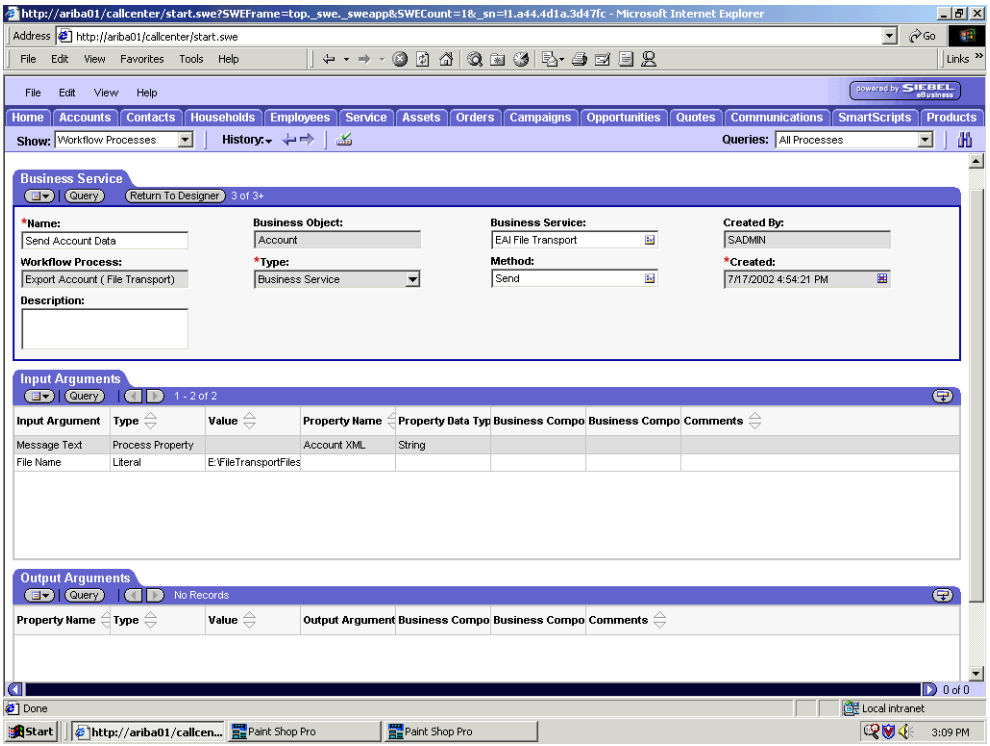
   The Business Service obtains the Account information from Siebel using the Query method.

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`.

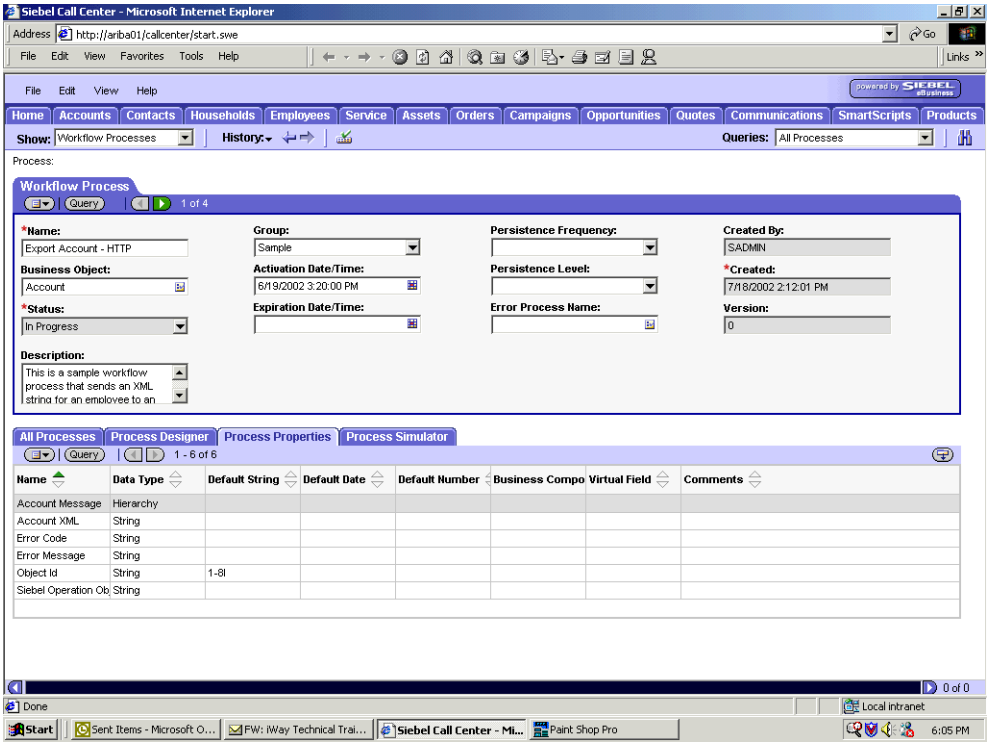
   This Business Service is defined to receive the Account data from the EAI XML Converter Business Service in Siebel XML format and send the Account XML to the file system in a specified directory using the Send method.

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:

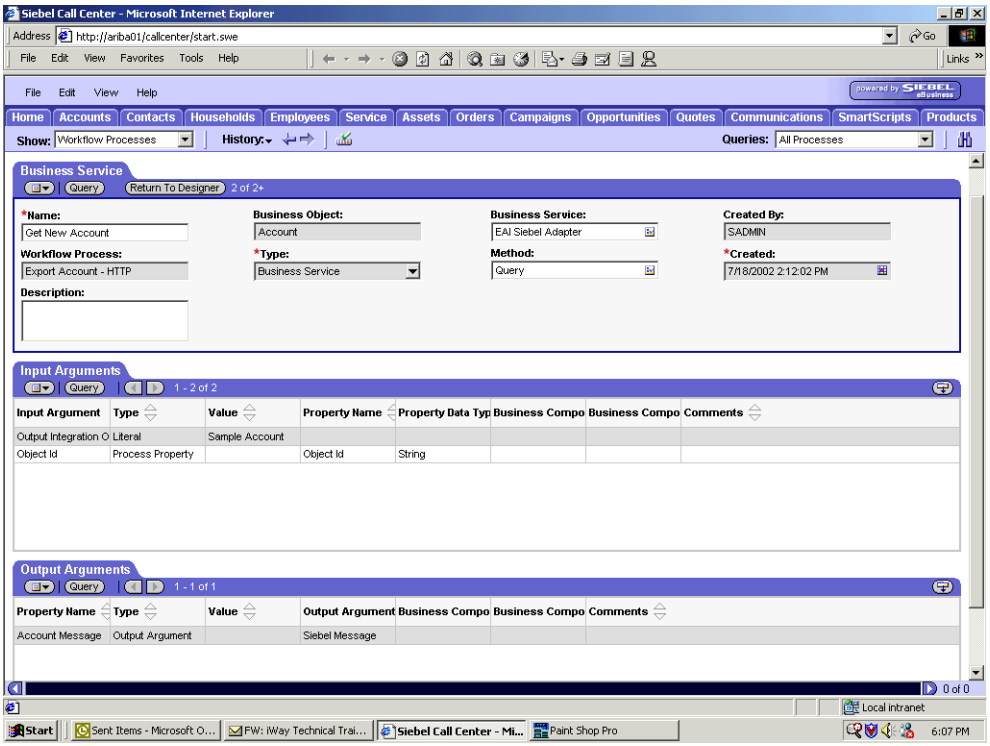


1. In the **Process Properties** tab of the **Workflow Process** window, define the Account message and Account XML process properties.

   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.



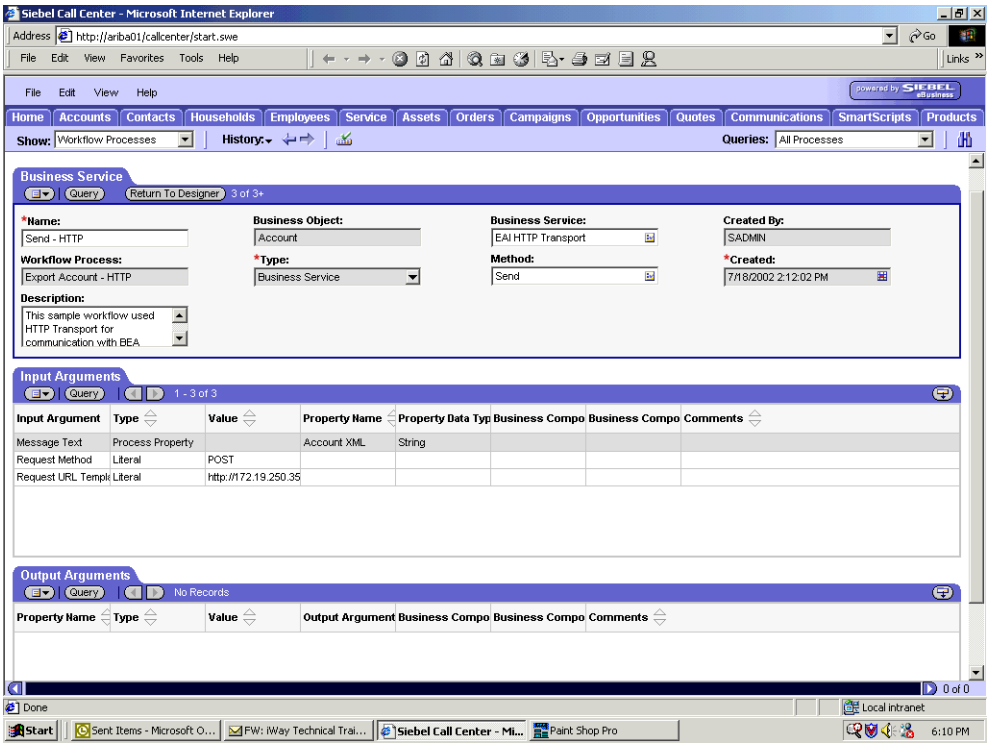
3. Define an **EAI Siebel Adapter Business Service** step to receive an instance of Account data and call it `Get New Account`.

   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.



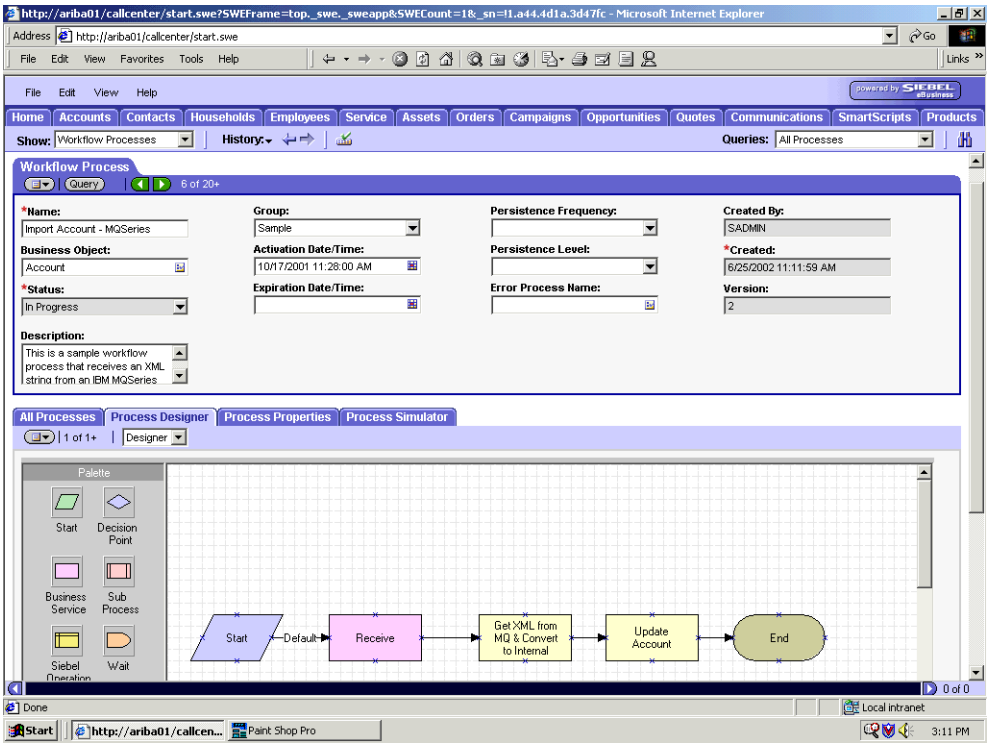
5. Define an **EAI HTTP Transport Business Service** step and call it `Send - HTTP`.

   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.

Creating a Siebel Workflow for a Service Using MQSeries Transport

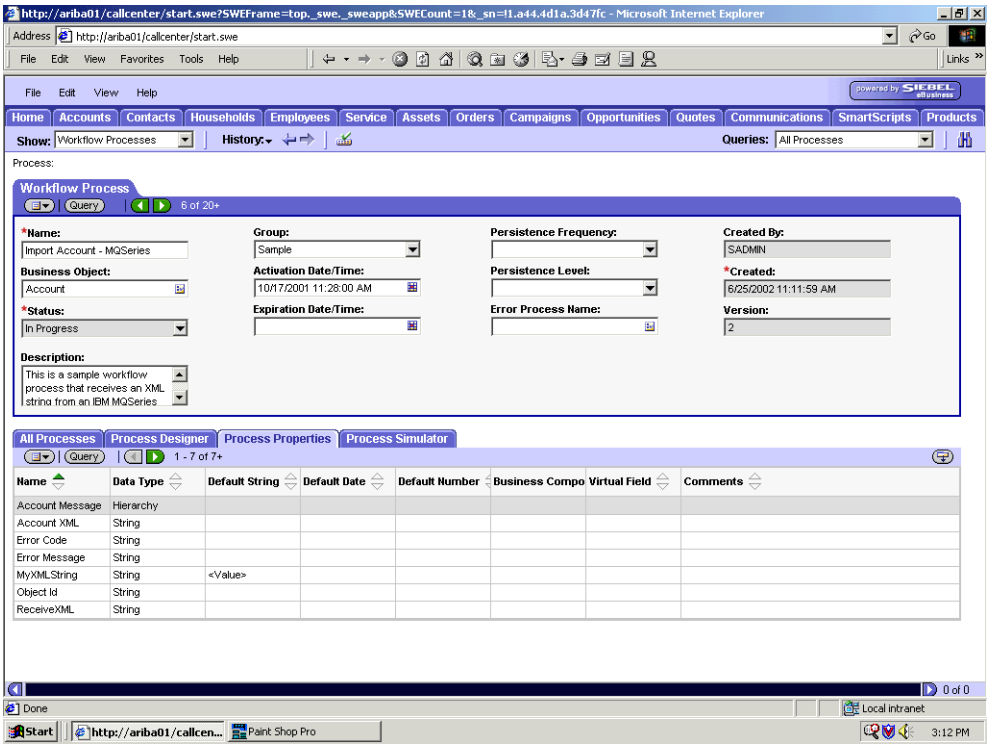
The following procedure is an example of a Siebel Workflow illustrated in the Siebel Workflow Administration window. The Workflow was designed for importing Siebel Account record information through the MQSeries Transport.

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.

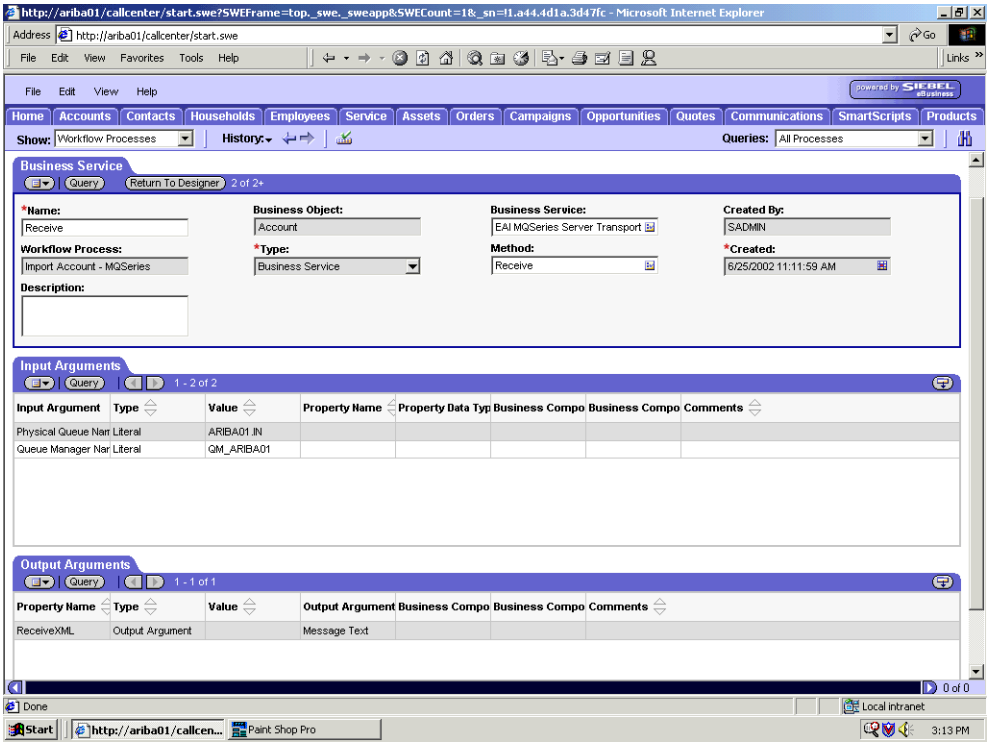
To create a Siebel Workflow:



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.

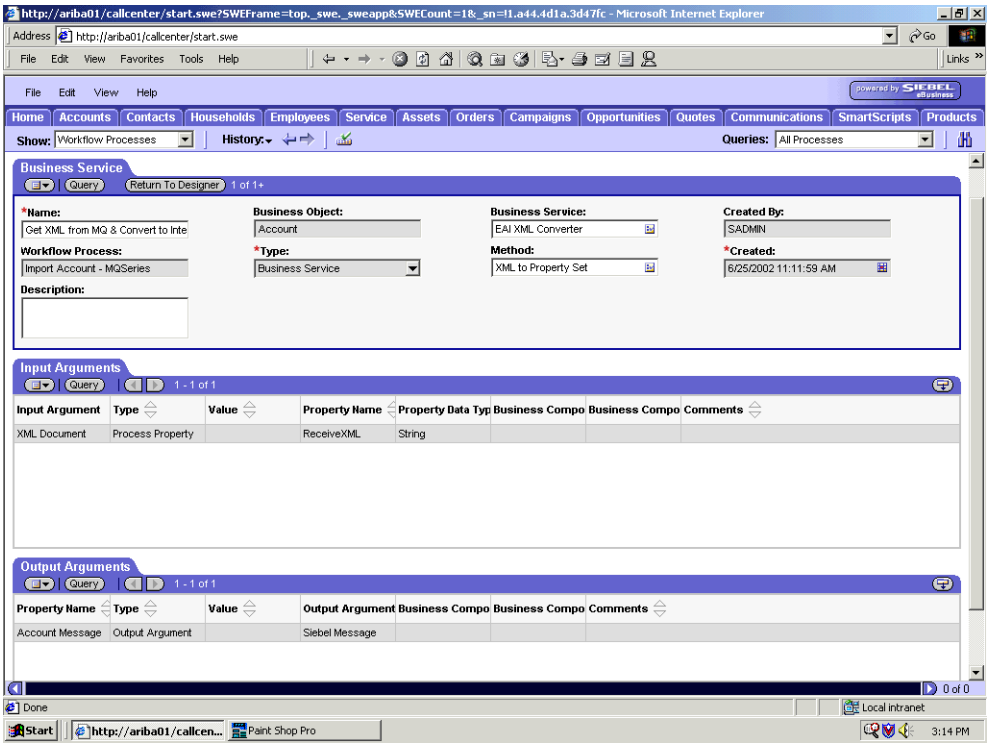
   Account XML specifies the Siebel Account data that the Workflow converted to XML.



2. Define an **EAI MQSeries Server Transport Business Service** step and call it `Receive`.

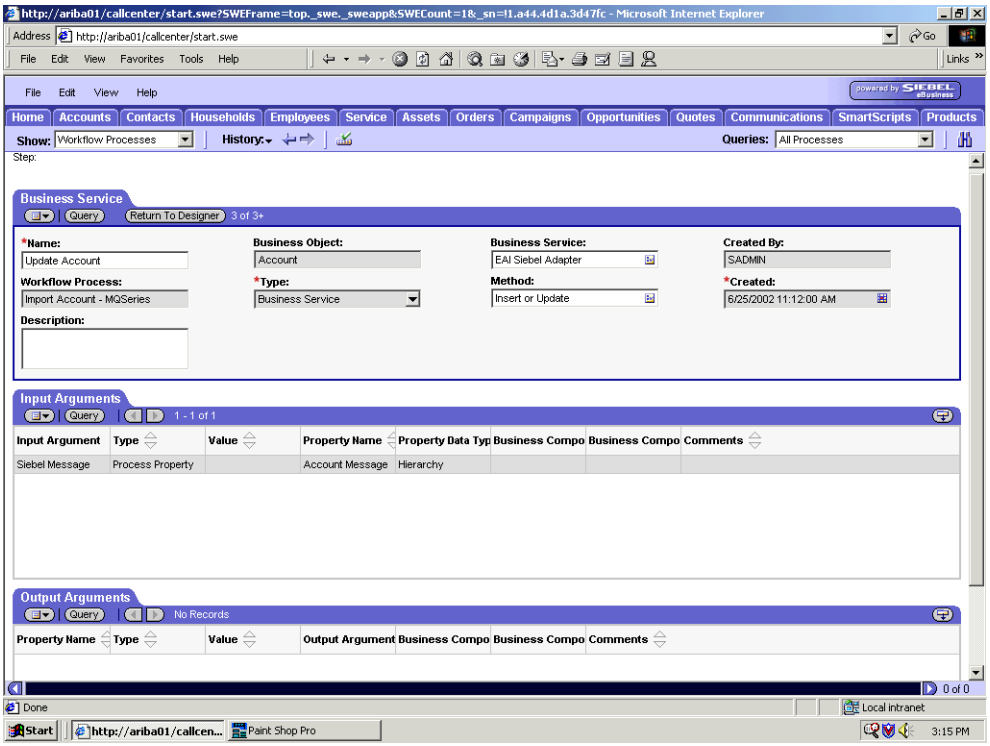
   The Business Service is defined to receive the Account data from the MQSeries message queue.

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.

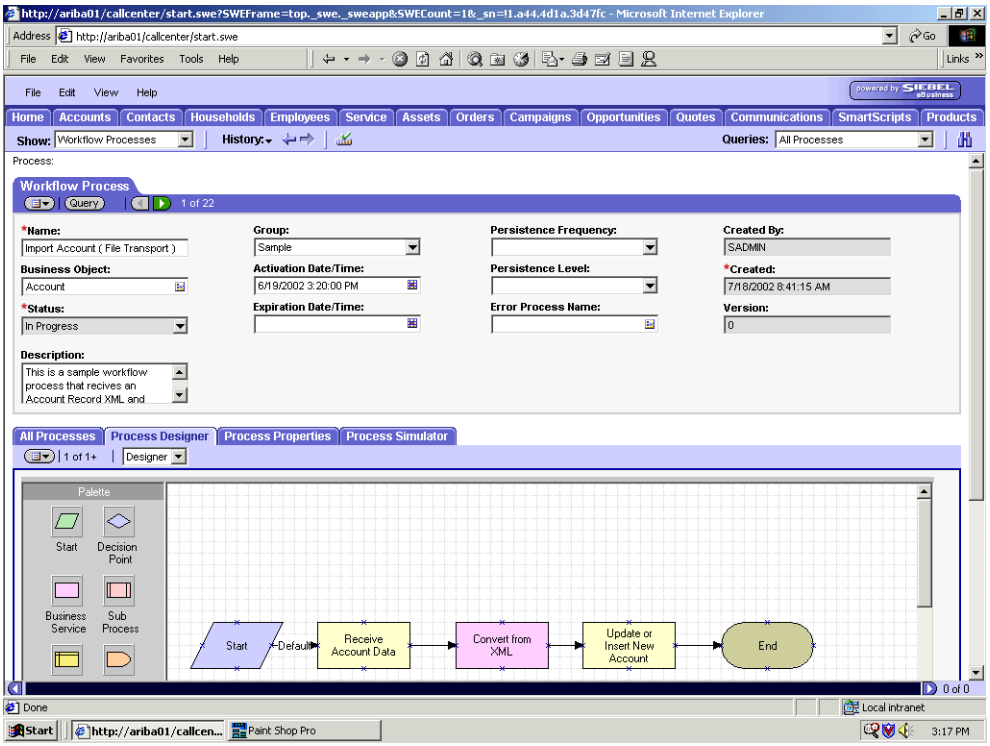
   The Business Service applies the Account information into Siebel using the Insert or Update method.

Creating a Siebel Workflow for a Service Using File Transport

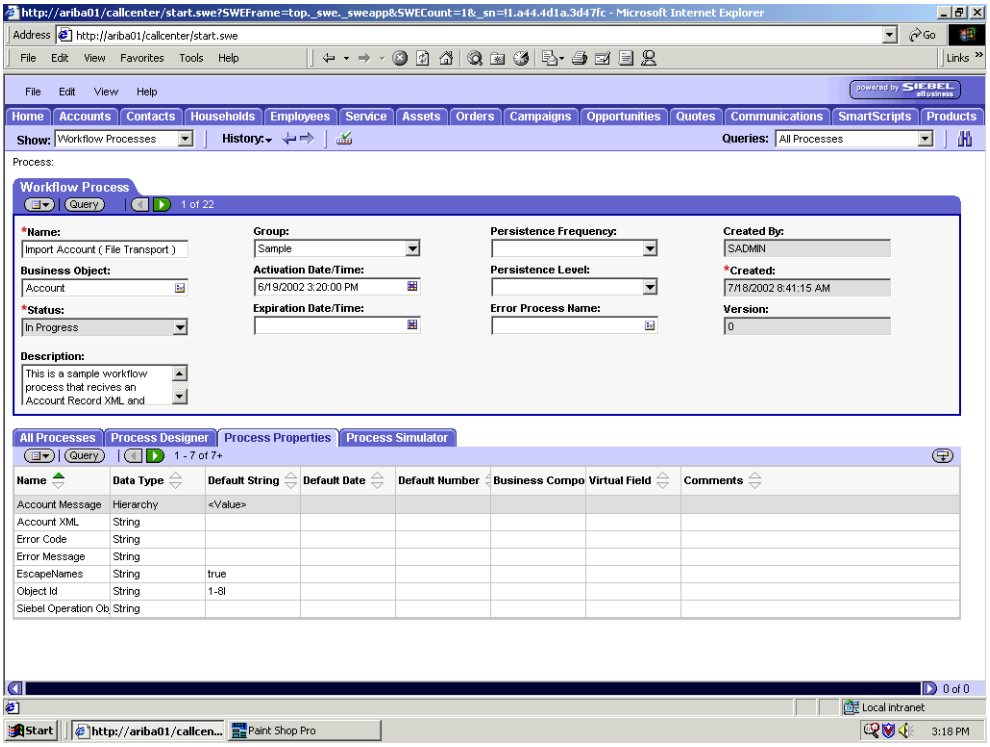
The following procedure is an example of a Siebel Workflow illustrated in the Siebel Workflow Administration window. The workflow was designed for importing Siebel Account record information through the File transport

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.



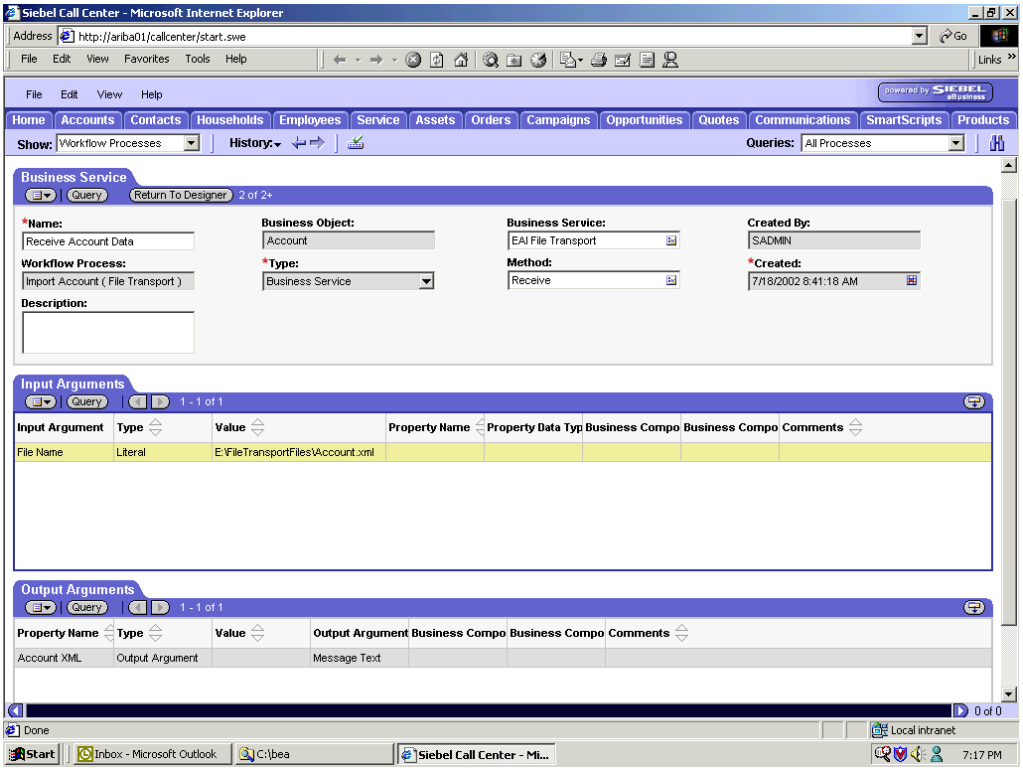
To create a Siebel Workflow:



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.

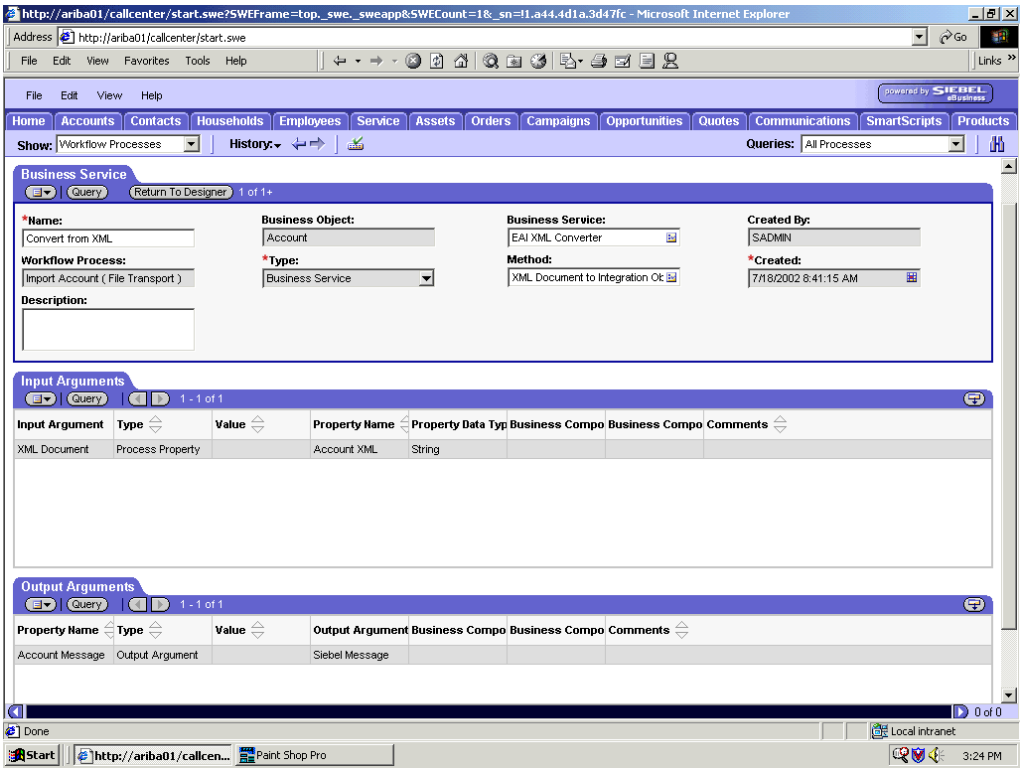
Account XML specifies the Siebel Account data that the workflow converted to XML.



2. Define an **EAI FileTransport Business Service** step and call it `Receive Account Data`.

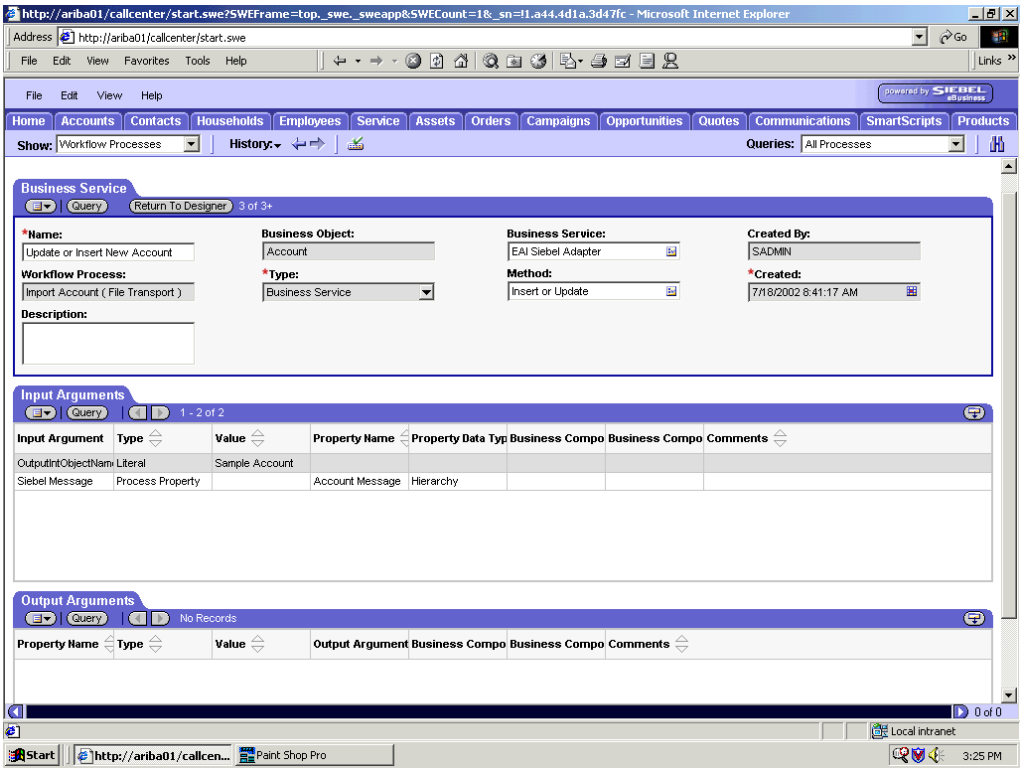
   The Business Service is defined to receive the Account data from the file system.

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.



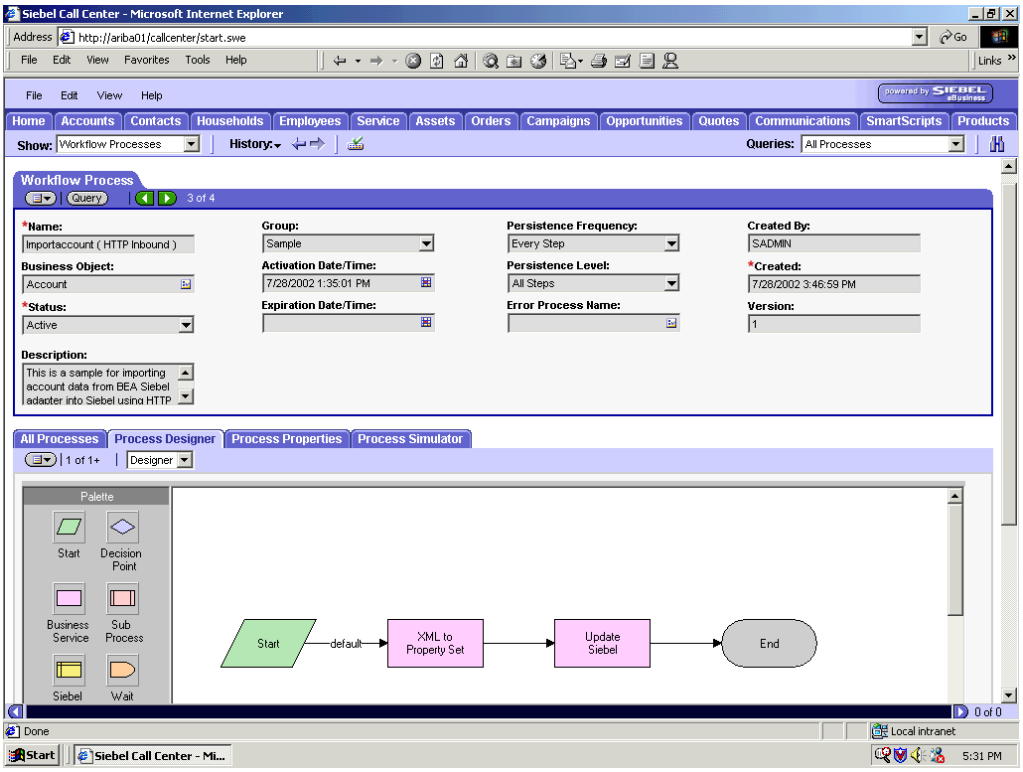
4. Define an **EAI Siebel Adapter Business Service** step and call it `Update` or `Insert New Account`.

   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.

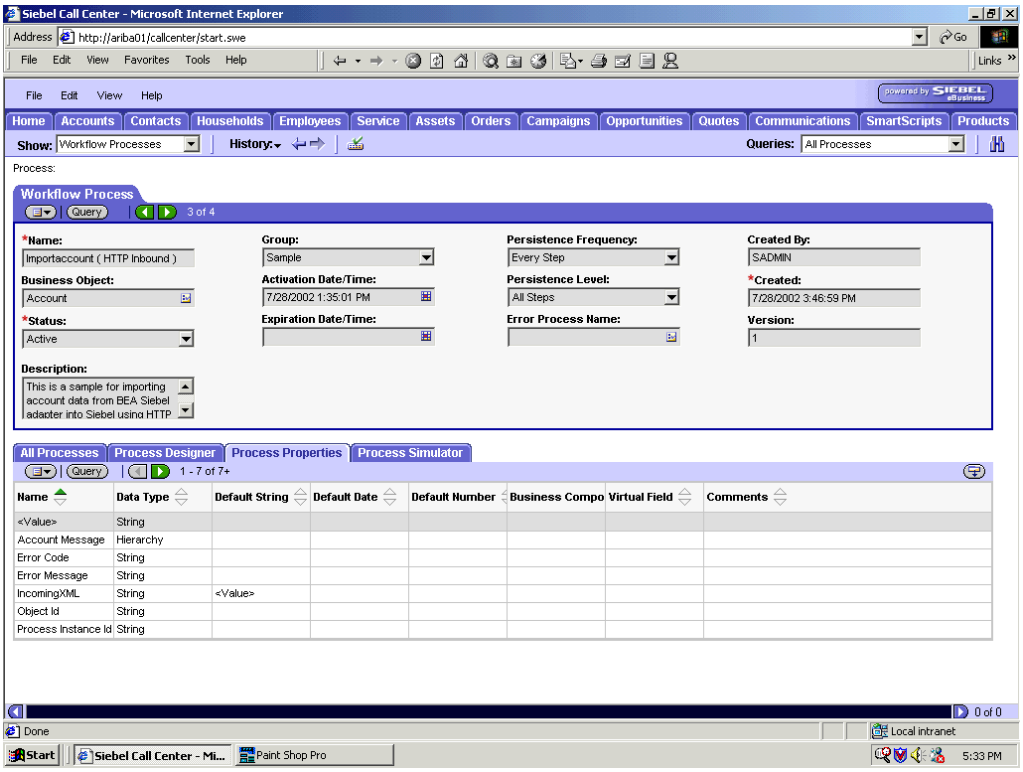
Creating a Siebel Workflow for a Service Using HTTP Transport

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.

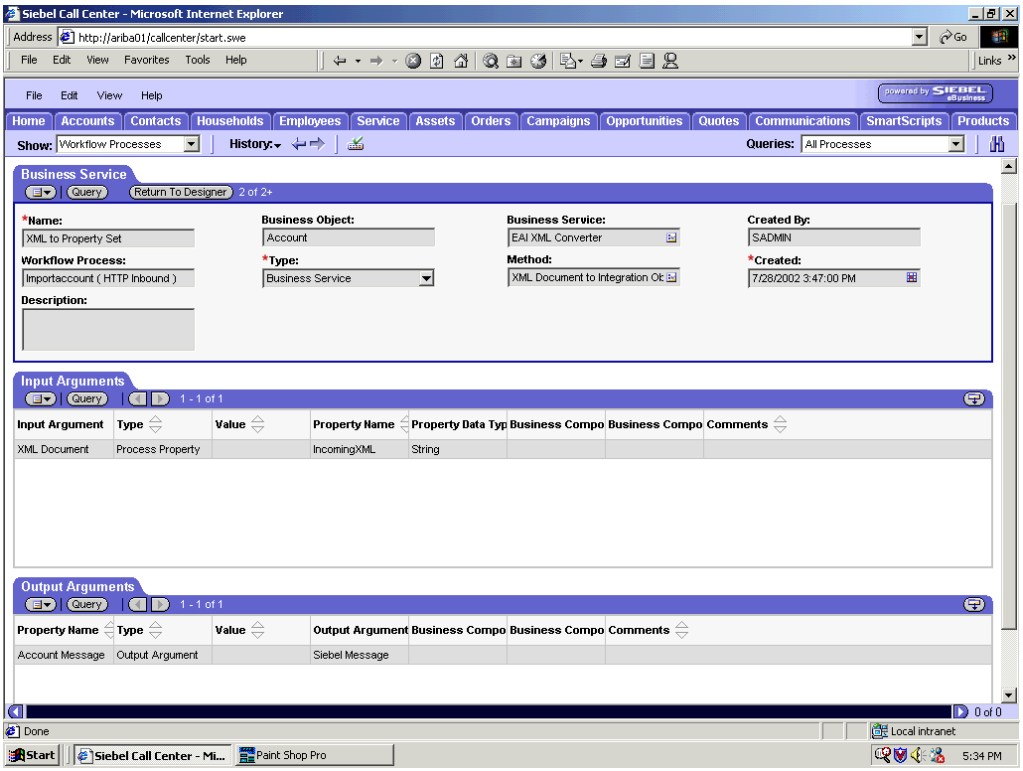
To create a Siebel Workflow:



1. In the **Process Properties** tab of the **Workflow Process** window, define the Account message and Account XML process properties.

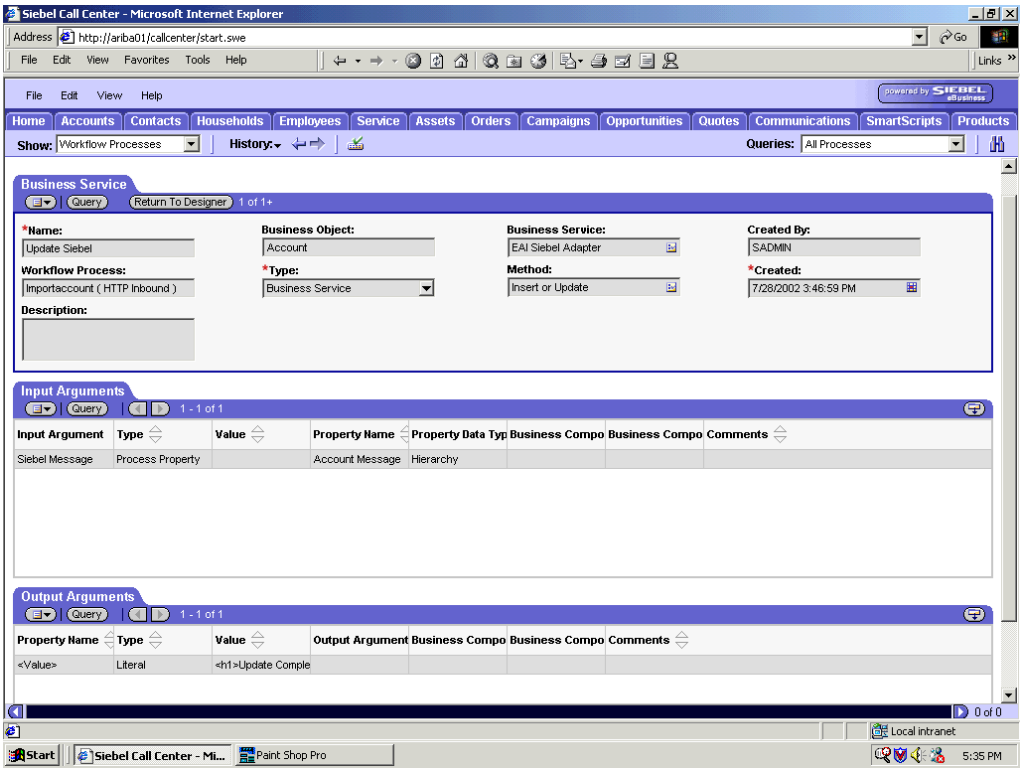
   Account message contains the Siebel Account data in hierarchical format.

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.

Glossary

adapter

Provides universal connectivity by enabling an electronic interface to be accommodated (without loss of function) to another electronic interface.

agent

Supports service protocols in listeners and documents.

business service

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

channel

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

listener

A component that accepts requests from client applications.

port

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

Index

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

X