

Oracle® Fusion Middleware

Application Adapter Upgrade Guide for Oracle WebLogic Server

11g Release 1 (11.1.1.7.0)

E17392-07

April 2013

Oracle Fusion Middleware Application Adapter Upgrade Guide for Oracle WebLogic Server, 11g Release 1 (11.1.1.7.0)

E17392-07

Copyright © 2009, 2013, Oracle and/or its affiliates. All rights reserved.

Primary Author: Stefan Kostial

Contributing Authors: Sunil Gopal, Marian Jones, Vikas Anand, Sunil Wadhwa, Vishal Saxena, Vimmika Dinesh

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	vii
Audience	vii
Documentation Accessibility	vii
Related Documents	viii
Conventions	viii
1 Migration Utility	
J2CA Migration Utility	1-1
Getting Started	1-2
Structure of the J2CA Migration Utility	1-2
Starting the J2CA Migration Utility	1-2
Migration Modes	1-2
Configuring a Complete Migration	1-3
Copying the Repository	1-3
Deleting From the New Repository	1-4
Using the Dump Utility	1-5
Using the Upload Utility	1-6
Configuring a Partial Migration	1-7
Using the Diff Utility	1-7
Editing the CSV File	1-8
Using the Insert Utility	1-8
Supported Scenarios	1-9
Usage Notes for Complete and Partial Migrations	1-10
BSE Migration Utility	1-10
Getting Started	1-11
Structure of the BSE Migration Utility	1-11
Starting the BSE Migration Utility	1-12
Migration Modes	1-12
Configuring a Complete Migration	1-12
Copying the Repository	1-12
Deleting From the New Repository	1-13
Using the Dump Utility	1-14
Using the Upload Utility	1-15
Configuring a Partial Migration	1-16
Using the Diff Utility	1-16

Editing the CSV File.....	1-17
Using the Insert Utility.....	1-17
Supported Scenarios.....	1-18
Usage Notes for Complete and Partial Migrations.....	1-18
SAP2_SAP3 J2CA Migration Utility.....	1-19
Introduction.....	1-19
Supported Releases.....	1-20
Migration Utility Installation Overview.....	1-20
Working With the Migration Utility.....	1-20
Use Case Scenario.....	1-23
Useful Considerations.....	1-24
Best Practices.....	1-24
Troubleshooting.....	1-25
SAP2_SAP3 BSE Migration Utility.....	1-30
Introduction.....	1-30
Supported Releases.....	1-30
Migration Utility Installation Overview.....	1-31
Working With the Migration Utility.....	1-31
Use Case Scenario.....	1-34
Useful Considerations.....	1-35
Best Practices.....	1-36
Troubleshooting.....	1-37

2 General Upgrade Guidelines

Upgrading a 10.1.3.x Outbound Process to 11g.....	2-1
Prerequisites.....	2-1
Overview of 10.1.3.x BPEL JCA Outbound Workflow Process.....	2-2
Creating an Outbound BPEL Process Project.....	2-2
Migrating the 10.1.3.x BPEL JCA Outbound Process to a 11g Workflow Process.....	2-10
Prerequisites.....	2-10
Overview of Migration to 11g BPEL JCA Outbound Workflow Process.....	2-10
Migrating an Outbound BPEL Process Project.....	2-10
Post-Upgrade Requirement.....	2-16
Upgrading a 10.1.3.x Inbound Process to 11g.....	2-17
Overview of 10.1.3.x BPEL JCA Inbound Workflow Process.....	2-18
Creating an Inbound BPEL Process Project.....	2-18
Migrating an Inbound BPEL Process Project.....	2-21
Post-Upgrade Requirement.....	2-26
Migrating Oracle Service Bus 10.1.3.x/PS2/PS3/PS4/PS5 Outbound and Inbound Processes to 11g PS6.....	2-27
Exporting the Configured Processes From Oracle Service Bus 10.1.3.x/PS2/PS3/PS4/PS5.....	2-27
Importing the Exported Processes to Oracle Service Bus 11g PS6.....	2-30
Additional Modifications for Imported Processes in Oracle Service Bus 11g PS6.....	2-37
Imported J2CA Inbound Processes.....	2-37
Imported BSE Outbound Processes.....	2-40

Upgrading 11g PS1/PS2/PS3/PS4/PS5 Outbound and Inbound BPEL and Mediator Processes to 11g PS6	2-42
Workaround for PS1 J.D. Edwards OneWorld Outbound Mediator Process When Migrating to 11g PS2/PS3/PS4/PS5/PS6	2-49
Additional Modifications for Migrated Processes in 11g PS6	2-54
Upgrading 11g PS2/PS3/PS4/PS5 Outbound and Inbound BPM Processes to 11g PS6	2-55
Workaround for J2CA BPM Processes	2-62
Additional Modifications for Migrated Processes in 11g PS6	2-73

3 Oracle Application Adapter for SAP R/3 Upgrade Guidelines

Upgrading a 10.1.3.x ESB J2CA Outbound Process to 11g	3-1
Upgrading a 10.1.3.x ESB J2CA Inbound Process to 11g	3-3
Upgrading a 10.1.3.x BPEL J2CA Outbound Process to 11g	3-5
Upgrading a 10.1.3.x BPEL J2CA Inbound Process to 11g	3-7

4 Oracle Application Adapter for Siebel Upgrade Guidelines

Upgrading a 10.1.3.x ESB J2CA Outbound Process to 11g	4-1
Upgrading a 10.1.3.x ESB J2CA Inbound Process to 11g	4-1
Upgrading a 10.1.3.x BPEL J2CA Outbound Process to 11g	4-8
Upgrading a 10.1.3.x BPEL J2CA Inbound Process to 11g	4-9

5 Oracle Application Adapter for PeopleSoft Upgrade Guidelines

Upgrading a 10.1.3.x ESB J2CA Outbound Process to 11g	5-1
Upgrading a 10.1.3.x ESB J2CA Inbound Process to 11g	5-1
Upgrading a 10.1.3.x BPEL J2CA Outbound Process to 11g	5-4
Upgrading a 10.1.3.x BPEL J2CA Inbound Process to 11g	5-5

6 Oracle Application Adapter for J.D. Edwards OneWorld Upgrade Guidelines

Upgrading a 10.1.3.x ESB J2CA Outbound Process to 11g	6-1
Upgrading a 10.1.3.x ESB J2CA Inbound Process to 11g	6-12
Upgrading a 10.1.3.x BPEL J2CA Outbound Process to 11g	6-16
Upgrading a 10.1.3.x BPEL J2CA Inbound Process to 11g	6-16

Index

Preface

Oracle Fusion Middleware Application Adapter Upgrade Guide for Oracle WebLogic Server describes how to upgrade Oracle Application Adapters for Oracle WebLogic Server from 10.1.3.x to 11g.

Audience

The *Oracle Fusion Middleware Application Adapter Upgrade Guide for Oracle WebLogic Server* is intended for system administrators who are upgrading Oracle Application Adapters for Oracle WebLogic Server from 10.1.3.x to 11g.

Documentation Accessibility

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

Accessibility of Code Examples in Documentation

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

Accessibility of Links to External Web Sites in Documentation

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

Access to Oracle Support

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

Related Documents

For more information, see the following documents in the Oracle Enterprise Repository 11g Release 1 (11.1.1.7.0) documentation set:

- *Oracle Fusion Middleware Application Adapters Installation Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter Best Practices Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter for SAP R/3 (SAP JCo 3.0) User's Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter for Siebel User's Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter for PeopleSoft User's Guide for Oracle WebLogic Server*
- *Oracle Fusion Middleware Application Adapter for J.D. Edwards OneWorld User's Guide for Oracle WebLogic Server*
- Oracle's Unified Method (OUM)

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

For more information about OUM, see the OUM FAQ at

http://my.oracle.com/portal/page/myo/ROOTCORNER/KNOWLEDGEAREAS1/BUSINESS_PRACTICE/Methods/Learn_about_OUM.html

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Migration Utility

This chapter describes how to configure and use the migration utility to migrate Oracle Application Adapter targets, channels, and Web services between J2CA and BSE environments. In addition, a migration utility for the Oracle Application Adapter for SAP R/3 to migrate between SAP JCo 2.x and SAP JCo 3.x versions of the adapter in J2CA or BSE configurations is available. The migration utility can be used to migrate an adapter repository from development, test, and production environments. The repositories can be migrated completely or partially for the artifacts. This chapter contains the following topics:

- [Section 1.1, "J2CA Migration Utility"](#)
- [Section 1.2, "BSE Migration Utility"](#)
- [Section 1.3, "SAP2_SAP3 J2CA Migration Utility"](#)
- [Section 1.4, "SAP2_SAP3 BSE Migration Utility"](#)

1.1 J2CA Migration Utility

The J2CA migration utility is used to migrate Oracle Application Adapter targets and channels between environments. This migration utility can be used to migrate an adapter repository in the J2CA container from development, test, and production environments. The repositories can be migrated completely or partially for these artifacts.

The J2CA migration utility supports:

- Only J2CA configurations
- The following Oracle Application Adapters:
 - SAP R/3
 - Siebel
 - PeopleSoft
 - J.D. Edwards
- Oracle and DB2 databases as repositories.
- Migration between:
 - File to database repositories
 - Database to database repositories

This section contains the following topics:

- [Section 1.1.1, "Getting Started"](#)

- [Section 1.1.2, "Configuring a Complete Migration"](#)
- [Section 1.1.3, "Configuring a Partial Migration"](#)
- [Section 1.1.4, "Usage Notes for Complete and Partial Migrations"](#)

1.1.1 Getting Started

This section contains the following topics:

- [Section 1.1.1.1, "Structure of the J2CA Migration Utility"](#)
- [Section 1.1.1.2, "Starting the J2CA Migration Utility"](#)
- [Section 1.1.1.3, "Migration Modes"](#)

1.1.1.1 Structure of the J2CA Migration Utility

The J2CA migration utility is structured as follows:

- Under `<ADAPTER_HOME>\etc\util`, the following files are available:
 - `iwrepocmd.jar` - The .jar file for the migration utility.

Note: This .jar file should not be used for any other purposes other than migration. This file should not be part of the classpath or any other path when the adapter running during design time or run time.

- `jcaupd.bat` - The script that must be used for Windows platforms.
- `jcaupd.sh` - The script that must be used for UNIX and Linux platforms.
- Copy the database JDBC driver files to the following directory:

`<ADAPTER_HOME>/lib`

Where `<ADAPTER_HOME>` is:

For Oracle SOA Suite:

`<SOA_HOME>\Oracle_SOA1\soa\thirdparty\ApplicationAdapters`

For OSB:

`<OSB_HOME>\Oracle_OSB1\3rdparty\ApplicationAdapters`

The database JDBC driver files will enable the migration utility to connect to the source and target database repositories.

1.1.1.2 Starting the J2CA Migration Utility

To begin using the J2CA migration utility:

1. Open a command window and navigate to the following directory:
`ApplicationAdapters/etc/util`
2. Execute the `jcaupd` commands as described.

1.1.1.3 Migration Modes

There are two migration modes available for the migration utility:

- **Complete.** This mode migrates all of the targets and channels from the source repository to the target repository. For more information, see [Section 1.1.2, "Configuring a Complete Migration"](#) on page 1-3.
- **Partial.** This mode migrates a partial list of targets and channels from the source repository to the target repository. For more information, see [Section 1.1.3, "Configuring a Partial Migration"](#) on page 1-7.

Note: All of the database repository commands used in this guide refer to the Oracle database. For DB2 databases, use the appropriate connection URL and database driver.

When executing the migration utility commands (throughout the migration process), ensure that Application Explorer and Oracle Application Server are not running and are shut down. This is applicable for the source and target environments.

1.1.2 Configuring a Complete Migration

This section describes how to configure a complete migration and contains the following topics:

- [Section 1.1.2.1, "Copying the Repository"](#)
- [Section 1.1.2.2, "Deleting From the New Repository"](#)
- [Section 1.1.2.3, "Using the Dump Utility"](#)
- [Section 1.1.2.4, "Using the Upload Utility"](#)

A complete migration inserts all of the targets and channels from a source repository to a target repository. This process consists of the following steps:

1. Copying the repository.
2. Deleting any targets and channels that are not required from the new repository.
3. Dumping the contents of the new repository to a CSV or XML file.
4. Editing the repository contents.
5. Uploading the changes to the new repository.

1.1.2.1 Copying the Repository

This section describes how to copy the source repository to the target repository.

1. Use the following syntax for the copy command:

```
jcaupd copy jca fromrepo torepo
```

Where *jca* is the name of a J2CA configuration created in Application Explorer.

2. Each repository argument may take the following form:

```
[-jdbc driver url user password | -file repofile]
```

3. For example, to copy a file-based repository to a database repository, the following syntax is used:

On Windows platforms:

```
jcaupd copy jca_sample -file C:\repository.xml -jdbc
oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway scott
```

tiger

On UNIX or Linux platforms:

```
./jcaupd.sh copy jca_sample -file /rdbms/ora117/repository.xml -jdbc  
oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway scott  
tiger
```

4. For example, to copy a database repository to another database repository, the following syntax is used:

On Windows platforms:

```
jcaupd copy jca_sample  
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway  
scott tiger  
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway2  
scott tiger
```

On UNIX or Linux platforms:

```
./jcaupd.sh copy jca_sample  
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway  
scott tiger  
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway2  
scott tiger
```

Note: If the destination repository is a database, then it must be a newly created database with no data. If a file, then the file should not exist prior to running the copy command.

In the case of a database repository, ensure that the same credentials that were used to create the repository and are also configured with J2CA. Do not use separate credentials for migration and other tasks.

1.1.2.2 Deleting From the New Repository

After a new repository is created from the source repository, you can remove any adapter targets and channels that are no longer required.

1. Use the following syntax to remove the entries from a repository:

To remove an adapter target:

```
jcaupd deltarget jca adapter target repo
```

To remove a channel:

```
jcaupd delchannel jca adapter channel repo
```

2. The repository argument may take the following form:

```
[-jdbc driver url user password | -file repofile]
```

3. For example, to remove the entries from a file repository, the following syntax is used:

On Windows platforms:

```
jcaupd deltarget jca_sample MySAP sap_target -file C:\repository.xml  
jcaupd delchannel jca_sample MySAP sap_ch -file C:\repository.xml
```

On UNIX or Linux platforms:

```
./jcaupd.sh deltarget jca_sample MySAP sap_target -file
/rdbms/ora117/repository.xml
./jcaupd.sh delchannel jca_sample MySAP sap_ch -file
/rdbms/ora117/repository.xml
```

4. For example, to remove the entries from a DB repository, the following syntax is used:

On Windows platforms:

```
jcaupd deltarget jca_sample MySAP sap_target -jdbc
oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway scott
tiger
jcaupd delchannel jca_sample MySAP sap_ch -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@localhost:1521:iway scott tiger
```

On UNIX or Linux platforms:

```
./jcaupd.sh deltarget jca_sample MySAP sap_target -jdbc
oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway scott
tiger
./jcaupd.sh delchannel jca_sample MySAP sap_ch -jdbc
oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway scott
tiger
```

Note: The commands in this procedure remove only the single record specified. Commands for deleting adapter keys are not provided, due to the potential for unintended side effects.

1.1.2.3 Using the Dump Utility

The dump utility writes the contents of a J2CA repository into a comma-separated value (CSV) file. The CSV file can be opened by Microsoft Excel.

1. Use the following syntax for the dump utility:

```
jcaupd dump jca file [-jdbc driver url user password | -file repofile]
```

Note: If the file ends with a .xml extension, then an XML file will be produced. Otherwise, a tab-delimited file will be produced.

2. For example, to dump the details from a file repository, the following syntax is used:

On Windows platforms:

```
jcaupd dump jca_sample repo.csv -file C:\repository.xml
```

On UNIX or Linux platforms:

```
./jcaupd.sh dump jca_sample repo.csv -file /rdbms/ora117/repository.xml
```

3. For example, to dump the details from a database repository, the following syntax is used:

On Windows platforms:

```
jcaupd dump jca_sample repo.csv -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@localhost:1521:iway scott tiger
```

On UNIX or Linux platforms:

```
./jcaupd.sh dump jca_sample repo.csv -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@localhost:1521:iway scott tiger
```

4. The program dumps the following information:

- Name and connection parameters for all adapter targets.
- Names and connection parameters for all channels.
- All ports.
- Adapter keys and values for all adapters.

The dump utility generally does not drop anything into the Keys row for Adapter, Target, Key, and Value, as shown in [Figure 1-1](#).

Figure 1-1 Keys Row

Keys			
Adapter	Target	Key	Value

Typically, the only time you will see anything in the Keys table is if you add a node to the metadata tree using adapter interactions.

The resulting file can be displayed and edited using a spreadsheet program, such as Microsoft Excel. Ensure that the following actions are not performed when editing the file:

- Changing target names, channel names, and so on. Only field values should be edited.
- Add new rows or columns to the file.
- Delete rows or columns from the file. Objects can only be deleted using the command line tool.
- If you want to do changes for the drop-down list parameters, only provide the supported values. Otherwise, the target parameters will not be listed in Application Explorer when the file is uploaded.
- Changing Boolean values to anything other than true or false is not recommended.

Password values appear in the file as encrypted strings. You may enter new passwords as plain text or leave the old encrypted passwords. If the password is plain text, the upload tool automatically encrypts the password when the file is uploaded.

1.1.2.4 Using the Upload Utility

By using the upload utility, details from the CSV file can be loaded back into the repository.

1. Use the following syntax for the upload utility:

```
jcaupd load jca file [-jdbc driver url user password | -file repofile]
```

- For example, to upload the entries to a file repository, the following syntax is used:

On Windows platforms:

```
jcaupd load jca_sample repo.csv -file C:\repository.xml
```

On UNIX or Linux platforms:

```
./jcaupd.sh load jca_sample repo.csv -file /rdbms/ora117/repository.xml
```

- For example, to upload the entries to a database repository, the following syntax is used:

On Windows platforms:

```
jcaupd load jca_sample repo.csv -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@localhost:1521:iway scott tiger
```

On UNIX or Linux platforms:

```
./jcaupd.sh load jca_sample repo.csv -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@localhost:1521:iway scott tiger
```

Note: The utility modifies existing records with values as provided in the CSV file. It cannot create or delete records.

1.1.3 Configuring a Partial Migration

This section describes how to configure a partial migration and contains the following topics:

- [Section 1.1.3.1, "Using the Diff Utility"](#)
- [Section 1.1.3.2, "Editing the CSV File"](#)
- [Section 1.1.3.3, "Using the Insert Utility"](#)
- [Section 1.1.3.4, "Supported Scenarios"](#)

Partial migration inserts selected targets, channels, and ports from a source repository to a target repository. The target repository may or may not be empty. This process consists of the following steps:

- Using the diff utility to create a CSV file containing the objects that are in the source repository, but not in the target repository.
- Removing any objects that are not desired in the target repository from the CSV file.
- Editing the object parameters in the CSV file.
- Using the insert utility to add the records from the source repository to the target repository.

1.1.3.1 Using the Diff Utility

Use the following syntax for the diff utility:

```
jcaupd diff jca file source target
```

The *source* and *target* designations use the following format:

```
[-jdbc driver url user password | -file repofile]
```

The diff utility produces a file identical in structure to that produced by the dump utility, but contains only objects that are in the source repository but not in the destination repository.

The following example shows how the diff utility can be executed between the source and target repositories.

Diff utility between a file repository (source) and a database repository (target).

On Windows platforms:

```
jcaupd diff jca_sample jca_diff.csv -file C:\repository.xml -jdbc
oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway2 scott tiger
```

On UNIX or Linux platforms:

```
./jcaupd.sh diff jca_sample jca_diff.csv -file /rdbms/ora117/repository.xml
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway2 scott
tiger
```

Diff utility between a database repository (source) and another database repository (target).

On Windows platforms:

```
jcaupd diff jca_sample jca_diff.csv
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost1:1521:iway scott
tiger
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost2:1521:iway2
scott tiger
```

On UNIX or Linux platforms:

```
./jcaupd.sh diff jca_sample jca_diff.csv
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost1:1521:iway scott
tiger
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost2:1521:iway2
scott tiger
```

1.1.3.2 Editing the CSV File

The CSV file can be modified to change the values. Ensure that you carefully modify the values for the adapter targets and channels. If you are removing any artifacts, then ensure that they are also removed with the dependencies. The following actions are allowed in the CSV file:

- Changing the values for adapter target connection parameters and channels parameters.
- Removing the artifacts, such as adapter targets and channels.

1.1.3.3 Using the Insert Utility

Use the following syntax for the insert utility:

```
jcaupd insert jca file source target
```

The *source* and *target* designations use the following format:

```
[-jdbc driver url user password | -file repofile]
```

The following example shows how the insert utility can be executed between the source and target repository.

Insert utility between file repository (source) and database repository (target).

On Windows platforms:

```
jcaupd insert jca_sample jca_diff.csv -file C:\repository.xml
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway2 scott
tiger
```

On UNIX or Linux platforms:

```
./jcaupd.sh insert jca_sample jca_diff.csv -file /rdbms/oral17/repository.xml
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway2 scott
tiger
```

Insert utility between database repository (source) and database repository (target).

On Windows platforms:

```
jcaupd insert jca_sample jca_diff.csv
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost1:1521:iway scott
tiger
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost2:1521:iway2
scott tiger
```

On UNIX or Linux platforms:

```
./jcaupd.sh insert jca_sample jca_diff.csv
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost1:1521:iway scott
tiger
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost2:1521:iway2
scott tiger
```

The insert utility copies all objects identified in the CSV file from the source repository to the destination repository. It then updates the object fields with the values specified in the CSV file. All objects in the CSV file must exist in the source repository.

1.1.3.4 Supported Scenarios

The following scenarios are supported with partial migration. If you have any questions about usage scenarios other than the ones that are mentioned in this section, then contact customer support.

- The source repository has artifacts for adapter targets and channels for each adapter. They are migrated to the target repository. If new artifacts are created in the source repository then, only the newly created artifacts can be migrated to the target repository.
- The source repository has artifacts for adapter targets and channels for each adapter. They are migrated to the target repository. If new artifacts are created in the source repository then, all the artifacts (old and new) can be migrated to a new target repository.
- The source repository can be Oracle Enterprise Edition and the target repository can be Oracle RAC. Migration can be performed in the other direction. For example, a migration from an Oracle RAC repository to an Oracle Enterprise Edition repository can also be performed.

1.1.4 Usage Notes for Complete and Partial Migrations

This section describes the usage notes for partial and complete migration.

File Paths

When specifying a file repository, the full path to the file must be provided. In addition, file paths containing spaces (for example, `C:\Program Files\iway60`), cannot be used with the utility.

JDBC Drivers

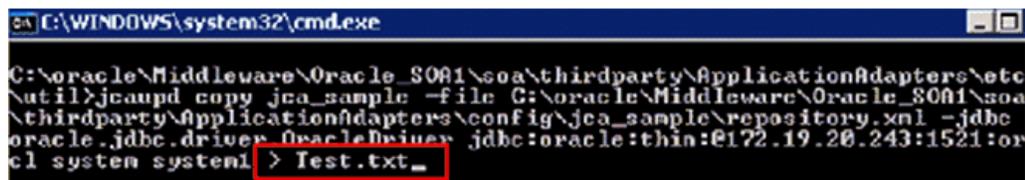
JDBC drivers must be placed in the `ApplicationAdapters\lib` folder where the migration utility is installed.

Log Files

Log files are not be generated with the migration utility. All logging information is printed to standard output. To capture logging for review purposes, redirect the standard output to a file using the `>` character in the command window.

For example, in the command prompt, specify the command, `> Filename.txt`, as shown in [Figure 1-2](#).

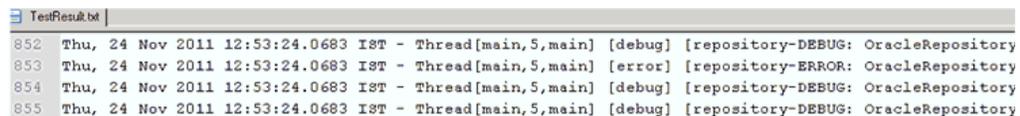
Figure 1-2 Command Prompt



```
C:\WINDOWS\system32\cmd.exe
C:\oracle\Middleware\Oracle_S001\soa\thirdparty\ApplicationAdapters\etc
\util>jcaupd copy jca_sample -file C:\oracle\Middleware\Oracle_S001\soa
\thirdparty\ApplicationAdapters\config\jca_sample\repository.xml -jdbc
oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@172.19.20.243:1521:or
cl system system > Test.txt
```

After the command is executed successfully, a text file is created in the specified location where you can review the captured text, as shown in [Figure 1-3](#).

Figure 1-3 Text File



```
TestResult.txt
852 Thu, 24 Nov 2011 12:53:24.0683 IST - Thread[main,5,main] [debug] [repository-DEBUG: OracleRepository
853 Thu, 24 Nov 2011 12:53:24.0683 IST - Thread[main,5,main] [error] [repository-ERROR: OracleRepository
854 Thu, 24 Nov 2011 12:53:24.0683 IST - Thread[main,5,main] [debug] [repository-DEBUG: OracleRepository
855 Thu, 24 Nov 2011 12:53:24.0683 IST - Thread[main,5,main] [debug] [repository-DEBUG: OracleRepository
```

1.2 BSE Migration Utility

The BSE migration utility is used to migrate Oracle Application Adapter targets and Web services between environments. This migration utility can be used to migrate the repository configuration details from development, test, and production environments. The repositories can be migrated completely or partially for these artifacts.

The BSE migration utility supports:

- Only BSE configurations
- The following Oracle Application Adapters:
 - SAP R/3
 - Siebel
 - PeopleSoft

- J.D. Edwards
- Oracle and DB2 databases as repositories.
- Migration between:
 - File to database repositories
 - Database to database repositories

This section contains the following topics:

- [Section 1.2.1, "Getting Started"](#)
- [Section 1.2.2, "Configuring a Complete Migration"](#)
- [Section 1.2.3, "Configuring a Partial Migration"](#)
- [Section 1.2.4, "Usage Notes for Complete and Partial Migrations"](#)

1.2.1 Getting Started

This section contains the following topics:

- [Section 1.2.1.1, "Structure of the BSE Migration Utility"](#)
- [Section 1.2.1.2, "Starting the BSE Migration Utility"](#)
- [Section 1.2.1.3, "Migration Modes"](#)

1.2.1.1 Structure of the BSE Migration Utility

The BSE migration utility is structured as follows:

- Under <ADAPTER_HOME>\etc\util, the following files are available:
 - *iwrepocmd.jar* - The .jar file for the migration utility.

Note: This .jar file should not be used for any other purposes other than migration. This file should not be part of the classpath or any other path when the adapter running during design time or run time.

- *ibspupd.bat* - The script that must be used for Windows platforms.
- *ibspupd.sh* - The script that must be used for UNIX and Linux platforms.
- Copy the database JDBC driver files to the following directory:

<ADAPTER_HOME>/lib

Where <ADAPTER_HOME> is:

For Oracle SOA Suite:

<SOA_HOME>\Oracle_SOA1\soa\thirdparty\ApplicationAdapters

For OSB:

<OSB_HOME>\Oracle_OSB1\3rdparty\ApplicationAdapters

The database JDBC driver files will enable the migration utility to connect to the source and target database repositories.

1.2.1.2 Starting the BSE Migration Utility

To begin using the BSE migration utility:

1. Open a command window and navigate to the following directory:
`ApplicationAdapters/etc/util`
2. Execute the `ibspupd` commands as described.

1.2.1.3 Migration Modes

There are two migration modes available for the migration utility:

- **Complete.** This mode migrates all of the targets and Web services from the source repository to the target repository. For more information, see [Section 1.2.2, "Configuring a Complete Migration"](#) on page 1-12.
- **Partial.** This mode migrates a partial list of targets and Web services from the source repository to the target repository. For more information, see [Section 1.2.3, "Configuring a Partial Migration"](#) on page 1-16.

Note: All of the database repository commands used in this guide refer to the Oracle database. For DB2 databases, use the appropriate connection URL and database driver.

When executing the migration utility commands (throughout the migration process), ensure that Application Explorer and Oracle Application Server are not running and are shut down. This is applicable for the source and target environments.

1.2.2 Configuring a Complete Migration

This section describes how to configure a complete migration and contains the following topics:

- [Section 1.2.2.1, "Copying the Repository"](#)
- [Section 1.2.2.2, "Deleting From the New Repository"](#)
- [Section 1.2.2.3, "Using the Dump Utility"](#)
- [Section 1.2.2.4, "Using the Upload Utility"](#)

A complete migration inserts all of the targets and Web services from a source repository to a target repository. This process consists of the following steps:

1. Copying the repository.
2. Deleting any targets and Web services that are not required from the new repository.
3. Dumping the contents of the new repository to a CSV or XML file.
4. Editing the repository contents.
5. Uploading the changes to the new repository.

1.2.2.1 Copying the Repository

This section describes how to copy the source repository to the target repository.

1. Use the following syntax for the copy command:

```
ibspupd copy fromrepo torepo
```

- Each repository argument may take the following form:

```
[-jdbc driver url user password | -file repofile]
```

- For example, to copy a file-based repository to a database repository, the following syntax is used:

On Windows platforms:

```
ibspupd copy -file C:\ibse_repository.xml -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@localhost:1521:iway scott tiger
```

On UNIX or Linux platforms:

```
./ibspupd.sh copy -file /rdbms/ora117/ibse_repository.xml -jdbc
oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway scott
tiger
```

- For example, to copy a database repository to another database repository, the following syntax is used:

On Windows platforms:

```
ibspupd copy
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway
scott tiger
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway2
scott tiger
```

On UNIX or Linux platforms:

```
./ibspupd.sh copy
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway
scott tiger
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway2
scott tiger
```

Note: If the destination repository is a database, then it must be a newly created database with no data.

In the case of a database repository, ensure that the same credentials that were used to create the repository and are also configured with BSE. Do not use separate credentials for migration and other tasks.

1.2.2.2 Deleting From the New Repository

After a new repository is created from the source repository, you can remove any adapter targets and Web services that are no longer required.

- Use the following syntax to remove the entries from a repository:

To remove an adapter target:

```
ibspupd deltarget adapter target repo
```

To remove a Web service:

```
ibspupd delservice adapter channel repo
```

2. The repository argument may take the following form:

```
[-jdbc driver url user password | -file repofile]
```

3. For example, to remove the entries from a file repository, the following syntax is used:

On Windows platforms:

```
ibspupd delservice sap_service -file C:\ibse_repository.xml  
ibspupd deltarget MySAP sap_target -file C:\ibse_repository.xml
```

On UNIX or Linux platforms:

```
./ibspupd.sh delservice sap_service -file /rdbms/ora117/ibse_repository.xml  
./ibspupd.sh deltarget MySAP sap_target -file /rdbms/ora117/ibse_repository.xml
```

4. For example, to remove the entries from a DB repository, the following syntax is used:

On Windows platforms:

```
ibspupd delservice sap_service -jdbc oracle.jdbc.driver.OracleDriver  
jdbc:oracle:thin:@localhost:1521:orcl scott tiger  
ibspupd deltarget MySAP sap_target -jdbc oracle.jdbc.driver.OracleDriver  
jdbc:oracle:thin:@localhost:1521:orcl scott tiger
```

On UNIX or Linux platforms:

```
./ibspupd.sh delservice sap_service -jdbc oracle.jdbc.driver.OracleDriver  
jdbc:oracle:thin:@localhost:1521:orcl scott tiger  
./ibspupd.sh deltarget MySAP sap_target -jdbc oracle.jdbc.driver.OracleDriver  
jdbc:oracle:thin:@localhost:1521:orcl scott tiger
```

Note: The `delservice` command will remove entries for the Web service from the service and method tables. Other commands remove only the single record specified. Commands for deleting adapter keys are not provided, due to the potential for unintended side effects.

1.2.2.3 Using the Dump Utility

The dump utility writes the contents of a BSE repository into a comma-separated value (CSV) file. The CSV file can be opened by Microsoft Excel.

1. Use the following syntax for the dump utility:

```
ibspupd dump csvfile [-jdbc driver url user password | -file repofile]
```

2. For example, to dump the details from a file repository, the following syntax is used:

On Windows platforms:

```
ibspupd dump repo.csv -file C:\ibse_repository.xml
```

On UNIX or Linux platforms:

```
./ibspupd.sh dump repo.csv -file /rdbms/ora117/ibse_repository.xml
```

3. For example, to dump the details from a database repository, the following syntax is used:

On Windows platforms:

```
ibspupd dump repo.csv -jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway scott tiger
```

On UNIX or Linux platforms:

```
./ibspupd.sh dump repo.csv -jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway scott tiger
```

4. The program dumps the following information into the CSV file:

- The service name, adapter, target, and target parameters for each service.
- Name and connection parameters for all adapter targets.
- Adapter keys and values for all adapters.

The dump utility generally does not drop anything into the Keys row for Adapter, Target, Key, and Value, as shown in [Figure 1-4](#).

Figure 1-4 Keys Row

Keys			
Adapter	Target	Key	Value

Typically, the only time you will see anything in the Keys table is if you add a node to the metadata tree using adapter interactions.

The resulting file can be displayed and edited using a spreadsheet program, such as Microsoft Excel. Ensure that the following actions are not performed when editing the file:

- Changing target names, service names, and so on. Only field values should be edited.
- Add new rows or columns to the file.
- Delete rows or columns from the file. Objects can only be deleted using the command line tool.
- If you want to do changes for the drop-down list parameters, only provide the supported values. Otherwise, the target parameters will not be listed in Application Explorer when the file is uploaded.
- Changing Boolean values to anything other than true or false is not recommended.

Password values appear in the file as encrypted strings. You may enter new passwords as plain text or leave the old encrypted passwords. If the password is plain text, the upload tool automatically encrypts the password when the file is uploaded.

1.2.2.4 Using the Upload Utility

By using the upload utility, details from the CSV file can be loaded back into the repository.

1. Use the following syntax for the upload utility:

```
ibspupd load csvfile [-jdbc driver url user password | -file repofile]
```

- For example, to upload the entries to a file repository, the following syntax is used:

On Windows platforms:

```
ibspupd load repo.csv -file C:\ibse_repository.xml
```

On UNIX or Linux platforms:

```
./ibspupd.sh load repo.csv -file /rdbms/ora117/ibse_repository.xml
```

- For example, to upload the entries to a database repository, the following syntax is used:

On Windows platforms:

```
ibspupd load repo.csv -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@localhost:1521:iway scott tiger
```

On UNIX or Linux platforms:

```
./ibspupd.sh load repo.csv -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@localhost:1521:iway scott tiger
```

Note: The utility modifies existing records with values as provided in the CSV file. It cannot create or delete records.

1.2.3 Configuring a Partial Migration

This section describes how to configure a partial migration and contains the following topics:

- [Section 1.2.3.1, "Using the Diff Utility"](#)
- [Section 1.2.3.2, "Editing the CSV File"](#)
- [Section 1.2.3.3, "Using the Insert Utility"](#)
- [Section 1.2.3.4, "Supported Scenarios"](#)

Partial migration inserts selected Web services and targets from a source repository to a target repository. The target repository may or may not be empty. This process consists of the following steps:

- Using the diff utility to create a CSV file containing the objects that are in the source repository, but not in the target repository.
- Removing any objects that are not desired in the target repository from the CSV file.
- Editing the object parameters in the CSV file.
- Using the insert utility to add the records from the source repository to the target repository.

1.2.3.1 Using the Diff Utility

Use the following syntax for the diff utility:

```
ibspupd diff csvfile source target
```

The *source* and *target* designations use the following format:

```
[-jdbc driver url user password | -file repofile]
```

The diff utility produces a CSV file identical in structure to that produced by the dump utility, but contains only objects that are in the source repository but not in the destination repository.

The following example shows how the diff utility can be executed between the source and target repositories.

Diff utility between a file repository (source) and a database repository (target).

On Windows platforms:

```
ibspupd diff phase1.csv -file C:\ibse_repository.xml -jdbc
oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway2 scott tiger
```

On UNIX or Linux platforms:

```
./ibspupd.sh diff phase1.csv -file /rdbms/ora117/ibse_repository.xml
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway2 scott
tiger
```

Diff utility between a database repository (source) and another database repository (target).

On Windows platforms:

```
ibspupd diff phase1.csv
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost1:1521:iway scott
tiger
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost2:1521:iway2
scott tiger
```

On UNIX or Linux platforms:

```
./ibspupd.sh diff phase1.csv
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost1:1521:iway scott
tiger
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost2:1521:iway2
scott tiger
```

1.2.3.2 Editing the CSV File

The CSV file can be modified to change the values. Ensure that you carefully modify the values for the adapter targets and Web services. If you are removing any artifacts, then ensure that they are also removed with the dependencies. The following actions are allowed in the CSV file:

- Changing the values for adapter target connection parameters and Web services parameters.
- Removing the artifacts, such as adapter targets and Web services.

1.2.3.3 Using the Insert Utility

Use the following syntax for the insert utility:

```
ibspupd insert csvfile source target
```

The *source* and *target* designations use the following format:

```
[-jdbc driver url user password | -file repopfile]
```

The following example shows how the insert utility can be executed between the source and target repository.

Insert utility between file repository (source) and database repository (target).

On Windows platforms:

```
ibspupd insert phase1.csv -file C:\ibse_repository.xml
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway2 scott
tiger
```

On UNIX or Linux platforms:

```
./ibspupd.sh insert phase1.csv -file /rdbms/ora117/ibse_repository.xml
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway2 scott
tiger
```

Insert utility between database repository (source) and database repository (target).

On Windows platforms:

```
ibspupd insert phase1.csv
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost1:1521:iway scott
tiger
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost2:1521:iway2
scott tiger
```

On UNIX or Linux platforms:

```
./ibspupd.sh insert phase1.csv
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost1:1521:iway scott
tiger
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost2:1521:iway2
scott tiger
```

The insert utility copies all objects identified in the CSV file from the source repository to the destination repository. It then updates the object fields with the values specified in the CSV file. All objects in the CSV file must exist in the source repository.

1.2.3.4 Supported Scenarios

The following scenarios are supported with partial migration. If you have any questions about usage scenarios other than the ones that are mentioned in this section, then contact customer support.

- The source repository has artifacts for adapter targets and Web services for each adapter. They are migrated to the target repository. If new artifacts are created in the source repository then, only the newly created artifacts can be migrated to the target repository.
- The source repository has artifacts for adapter targets and Web services for each adapter. They are migrated to the target repository. If new artifacts are created in the source repository then, all the artifacts (old and new) can be migrated to a new target repository.
- The source repository can be Oracle Enterprise Edition and the target repository can be Oracle RAC. Migration can be performed in the other direction. For example, a migration from an Oracle RAC repository to an Oracle Enterprise Edition repository can also be performed.

1.2.4 Usage Notes for Complete and Partial Migrations

This section describes the usage notes for partial and complete migration.

File Paths

When specifying a file repository, the full path to the file must be provided. In addition, file paths containing spaces (for example, C:\Program Files\iway60), cannot be used with the utility.

JDBC Drivers

JDBC drivers must be placed in the ApplicationAdapters\lib folder where the migration utility is installed.

Log Files

Log files are not be generated with the migration utility. All logging information is printed to standard output. To capture logging for review purposes, redirect the standard output to a file using the > character in the command window.

For example, in the command prompt, specify the command, > *Filename.txt*, as shown in Figure 1-5.

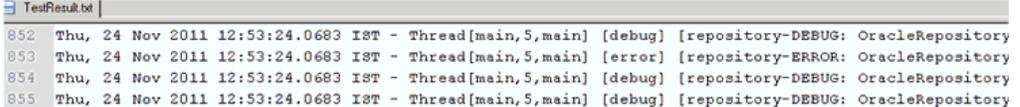
Figure 1-5 Command Prompt



```
C:\WINDOWS\system32\cmd.exe
C:\TEMP\iwrepopcmd>ibspupd copy -file C:\TEMP\ibserepo_result.xml -jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@localhost:1521:iway scott tiger > testResult.txt
```

After the command is executed successfully, a text file is created in the specified location where you can review the captured text, as shown in Figure 1-6.

Figure 1-6 Text File



```
TestResult.txt
852 Thu, 24 Nov 2011 12:53:24.0683 IST - Thread[main,5,main] [debug] [repository-DEBUG: OracleRepository
853 Thu, 24 Nov 2011 12:53:24.0683 IST - Thread[main,5,main] [error] [repository-ERROR: OracleRepository
854 Thu, 24 Nov 2011 12:53:24.0683 IST - Thread[main,5,main] [debug] [repository-DEBUG: OracleRepository
855 Thu, 24 Nov 2011 12:53:24.0683 IST - Thread[main,5,main] [debug] [repository-DEBUG: OracleRepository
```

1.3 SAP2_SAP3 J2CA Migration Utility

This section describes the SAP2_SAP3 J2CA migration utility. It contains the following topics:

- Section 1.3.1, "Introduction"
- Section 1.3.2, "Supported Releases"
- Section 1.3.3, "Migration Utility Installation Overview"
- Section 1.3.4, "Working With the Migration Utility"
- Section 1.3.5, "Use Case Scenario"
- Section 1.3.6, "Useful Considerations"
- Section 1.3.7, "Best Practices"
- Section 1.3.8, "Troubleshooting"

1.3.1 Introduction

A new version of the Oracle Fusion Middleware Application Adapter for SAP R/3 is available that supports SAP Java Connector (SAP JCo) version 3.x. The SAP JCo API has changed by providing a different set of connection parameters to connect with the SAP R/3 server. As a result, the new Oracle Fusion Middleware Application Adapter

for SAP R/3 has a different set of connection parameters than the previous version, which supported SAP JCo version 2.x. This introduced a backward compatibility issue by preventing users to reuse adapter targets and channels that were created by the previous version of the SAP R/3 adapter with the new version of the adapter that supports SAP JCo version 3.x. To enable the transition from the previous version of the SAP R/3 adapter, a command line migration utility is available that can be used to migrate the adapter targets and channels from the SAP R/3 adapter (using SAP JCo 2.x) to the SAP R/3 adapter (using SAP JCo 3.x).

This migration utility cannot completely automate the migration of the targets and channels between the SAP R/3 adapters because of the incompatibility between the SAP JCo versions. As a result, some manual operation is required. The scope of the migration utility is to provide adapter target and channel migration capabilities between both versions of the SAP R/3 adapter. The migration utility does not provide any other functionality and cannot be used for any other purposes. In this appendix, the SAP R/3 adapter (using SAP JCo 2.x) is referred to as the sap2 adapter and the SAP R/3 adapter (using SAP JCo 3.x) is referred to as the sap3 adapter.

1.3.2 Supported Releases

The following Oracle releases are supported:

- 10.1.3.4
- 11g PS1
- 11g PS2

Only J2CA configurations are supported for migration purpose. BSE configurations are not supported by the migration utility. The migration utility only supports Oracle database repositories for migration purposes. No other database repositories are supported. The utility works with a file repository, but migrating file repositories is not supported.

1.3.3 Migration Utility Installation Overview

The migration utility is located in the following directory:

```
<ADAPTER_HOME>\etc\util
```

The contents of the `util` folder include:

- The `iwrepocmd.jar` file, which is the required .jar file for the migration utility.

Note: The `iwrepocmd.jar` file must not be used for any other purpose other than the migration. This file must not be part of the classpath or any other path when the adapter is running for design time or runtime purposes.

- The `sapupd.bat` file, which is the migration utility .bat file for Windows platforms.
- The `sapupd.sh` file, which is the migration utility .sh file for UNIX/Linux platforms.

1.3.4 Working With the Migration Utility

This section describes the syntax that must be used with the migration utility.

Syntax for Listing the Targets or Channels

```
sapupd list config [targets | channels] [sap2 | sap3 | sap2to3] [-jdbc driver url
user password | -file file] > [output file]
```

Table 1–1 Syntax for Listing the Targets or Channels

Syntax	Description
sapupd list	Command to list the adapter targets or channels.
config	The name of the J2CA configuration that was specified by the user in Application Explorer.
targets channels	List of targets or channels.
sap2 sap3 sap2to3	List of sap2 objects, sap3 objects, or sap2 objects as mapped to sap3 parameters.
-jdbc driver url user password -file file	Repository to connect to, which can be a file or Oracle database repository. The JDBC URL is used to connect to the Oracle database repository.
output file	List of adapter targets or channels are redirected to the output file, which must use the .tab extension.

Usage considerations:

- Use the `sapupd list` command to list the targets or channels from a file or Oracle database repository (J2CA configuration). The output consists of tab-delimited text with targets or channels as columns and parameters as rows.
- The `sap2` and `sap3` options are provided to allow you to list the respective SAP R/3 objects from the repository.
- When you run the migration utility, you can list only the adapter targets or channels. Adapter targets or channels cannot be combined to create a single Excel spreadsheet.
- The output file name must use the .tab extension, since that extension is recognized by Excel for the migration. In this appendix, the output file uses the `sap2to3.tab` naming convention.
- To replace sap2 objects with sap3 objects, the `sap2to3` option can be used.
In the following examples, the `sapupd list` command lists all sap2 targets mapped to sap3 parameters into the `sap2to3.tab` file.

File Repository Example

```
sapupd list jca targets sap2to3 -file $ADAPTER_HOME\config\jca_
sample\repository.xml > sap2to3.tab
```

In this example:

- `ADAPTER_HOME` is the location where the adapters are installed.
- `jca` represents the name of the J2CA configuration that was specified in Application Explorer.

Database Repository Example

```
sapupd list jca targets sap2to3 -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@192.168.128.164:1521:orcl scott scott1 > sap2to3.tab
```

For UNIX/Linux platforms, execute the `sapupd.sh` file. For example, the following command lists all sap2 channels mapped to sap3 parameters into the `dbchn.tab` file for a database repository:

```
./sapupd.sh list jca_sample channels sap2to3 -jdbc
oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@172.19.20.242:1521:orcl system
welcome1 > dbchn.tab
```

Syntax for Updating the Targets or Channels

```
sapupd update config [targets | channels] sap2to3[source file] [-jdbc driver url
user password | -file file]
```

Table 1–2 Syntax for Updating the Targets or Channels

Syntax	Description
<code>sapupd update</code>	Command to update the adapter targets or channels.
<code>config</code>	The name of the J2CA configuration that was specified in Application Explorer.
<code>targets channels</code>	List of targets or channels to update.
<code>source file</code>	The source Excel file that contains the mappings between sap2 and sap3 parameters.
<code>-jdbc driver url</code> <code>user password </code> <code>-file file</code>	Repository to connect to, which can be a file or Oracle database repository. The JDBC URL is used to connect to the Oracle database repository.

Use the `sapupd update` command to update the targets or channels in a file or Oracle database repository (J2CA configuration).

File Repository Example

```
sapupd update jca targets sap2to3 sap2to3.tab -file $ADAPTER_HOME\config\jca_
sample\repository.xml
```

Database Repository Example

```
sapupd update jca targets sap2to3 sap2to3.tab -jdbc
oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@192.168.128.164:1521:orcl scott
scott1
```

For UNIX/Linux platforms, run the `sapupd.sh` file. For example:

```
./sapupd.sh update jca targets sap2to3 sap2to3.tab -jdbc
oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@172.19.20.242:1521:orcl system
welcome1
```

A manual step must be performed before you create the list and update the J2CA repository. This is the process of mapping the values between sap2 and sap3 adapter targets and channels. At this stage, you must take a look at the values underneath each adapter target/channel column and update the output file that has been created accordingly. While editing, do not remove or add rows and columns to the file. Also, do not change the SAP R/3 connection parameters in the file. The only action allowed is to update the parameter values.

Figure 1–7 shows sample output from a J2CA repository with two SAP R/3 application server targets, "asserver2" and "isdsrv2", and one SAP R/3 message server target, "msgsrvr", as opened in an Excel spreadsheet. The SAP R/3 parameters have

been pre-populated with the matching sap2 parameter values. If no matching value was present in the sap2 target, then the sap3 parameter default is used.

Figure 1–7 Sample Mapping Document

	A	B	C
1	Application Server		
2		asserver2	isdsrv2
3	jco.client.client		800
4	user	iwayqa	
5	password	tom21cat	
6	jco.client.lang	EN	EN
7	jco.client.codepage		
8	authentication	Password	Password
9	jco.client.ashost	asserver2	isdsrv2
10	jco.client.sysnr		0
11	jco.destination.peak_limit		2
12	jco.destination.expiration_time		10
13	jco.destination.max_get_client		30
14	edi_version		3
15	edi_release		
16	edi_release_option	IDOC DOCREL field	IDOC DOCREL field
17	exception	Throws Exception	Throws Exception
18	commitwait	FALSE	FALSE
19	jco.client.trace	FALSE	FALSE
20	trace_level		0
21	jco.client.snc_mode		0
22	jco.client.snc_partnername		
23	jco.client.snc_qop		3
24	jco.client.snc_myname		
25	jco.client.snc_lib		
26			
27	Message Server		
28		msgsrvr	
29	jco.client.client		800
30	user	iwayqa	
31	password	tom21cat	
32	jco.client.lang	EN	
33	jco.client.codepage		

You can add or modify parameter values as required, then save the file and use it to replace existing sap2 objects with sap3 objects.

The update command can be used only to replace existing sap2 targets or channels with their sap3 equivalents. It cannot be used to create new targets or channels or to edit the parameter values of existing targets and channels. Also, you cannot change the adapter target or channel names in the file.

1.3.5 Use Case Scenario

This section provides a use case scenario for the migration utility.

1. Close Application Explorer. Ensure that no instance of Application Explorer is running and that no updates are being made in the J2CA repository.
2. Use the `sapupd list` command to list the connection parameters into an Excel spreadsheet. This command inserts the sap2 targets or channels into the spreadsheet. Running this command only extracts targets or channels. As a result, first run the command to list the adapter targets and then repeat the command to

list the channels. Ensure to save the contents in separate files for adapter targets and channels.

3. Make an required updates to the Excel spreadsheet. This is the step where the Excel spreadsheet is updated with the sap3 targets and channels.
4. Stop Oracle WebLogic Server if it is running. Remove the SAP R/3 adapter (using SAP JCo 2.x) files and any SAP JCo 2.x library files from the adapter environment.
5. Install the SAP R/3 adapter (using SAP JCo 3.x) in the environment. This adapter is delivered as a patch for 10.1.3.4 and 11g PS1 releases. It is a part of the application adapters in the 11g PS2/PS3/PS4/PS5/PS6 releases.
6. If you are using a 10.1.3.4 adapter environment, then check the installation documentation when making any changes to configuration files (for example, server.xml).
7. Use the `sapupd update` command to update the connection parameters for sap3 targets or channels. Running this command only updates adapter targets or channels from different files. As a result, repeat the steps accordingly with the correct input file. As a best practice, first update the adapter targets and then update the adapter channels.
8. Start Application Explorer and connect to an SAP R/3 adapter (using SAP JCo 3.x) target and channel to ensure that the design time is working.
9. Restart the Oracle WebLogic Server and execute runtime test cases to verify successfully functionality.

1.3.6 Useful Considerations

This section includes useful considerations for the migration utility.

The following is list of actions that are recommended:

- Close Application Explorer before executing the update command. The target or channel parameters are not updated if Application Explorer is open.
- Running the utility remains the same with a single instance of the Oracle database or with the RAC database. In the RAC database environment, running the utility on one instance updates the repository for any other instances that belong to the environment.

The following is list of actions that must be avoided:

- Do not add, delete or change the rows in the Excel spreadsheet that was created by the upgrade utility.
- Do not rename the targets or channels in the Excel spreadsheet that was created by the upgrade utility.
- Do not add or delete the column in the Excel spreadsheet that was created by the upgrade utility.
- This utility cannot be used to migrate the adapter targets and channels from one environment to another environment. For example, migration between development, testing, and production environments cannot be performed. You must use the migration utility in each of the environments separately.

1.3.7 Best Practices

This section describes best practices for the migration utility.

- Create a backup of the Oracle database which contains the J2CA repository. This applies irrespective of development, testing, and production environments. In case of a file repository, create a backup of the file repository. This also applies irrespective of development, testing, and production environments. The file repository is located in the adapter folder structure. For example:

```
$ADAPTER_HOME\config\jca
```

- When you use the migration utility with K shell, the following warning may be generated, which can be ignored:

```
./sapupd.sh: /dev/null: bad number
```

- When the migration utility extracts information into the Excel spreadsheet, the Excel spreadsheet takes 00 as 0 for the system number parameter value for the sap2 target. When the sap2 target values are mapped to sap3 from the Excel spreadsheet, the adapter target has the system number parameter value set to 0 in Application Explorer.

There is no issue with the adapter to work with this target in design and run time. However, it is recommended that you to manually update the target for the correct value. For example, replace "0" with "00".

- There is no command to migrate both targets and channels. This must be done separately
- When performing the upgrade in a production environment, consider the following guidelines:
 1. Extract the sap2 adapter targets and channels well in advance of the production migration date. Perhaps two to four weeks earlier. This date is just an approximation.
 2. Make changes to the Excel spreadsheet with the new parameter values for the sap3 adapter.
 3. Upload the details from the Excel spreadsheet into the environment which should be an exact replica of the production environment. Ensure that the design time (Application Explorer), run time (BPEL, ESB, Mediator) is working. Note that if there is a change in the environment, then the runtime scenarios based on BPEL, ESB, and Mediator process flows would not work.
 4. If everything is working accordingly, then keep the Excel spreadsheet intact until the production migration.
 5. Do not make any changes to the adapter targets and channels in the production environment after extracting the details. If you make any changes, then repeat steps 1 through 4.
 6. On the day of the production environment migration, use the Excel spreadsheet to update the sap3 adapter targets and channels.
This approach minimizes the downtime during the production migration.
 7. If your production environment is a high availability (cluster) environment, then perform this procedure for any node that belongs to the cluster.

1.3.8 Troubleshooting

This section provides troubleshooting information for the migration utility.

- Executing the `sapupd update` command to update the adapter targets with the `sap2to3.tab` file, which contains the adapter channel parameters, generates the following exception:

```
C:\soadp1\adapters\application\etc\util>sapupd update jca_sample targets
sap2to3 sap2to3chn.tab -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@192.168.128.164:1521:orcl scott scott1
```

```
Exception in thread "main"
com.iwaysoftware.iwrepository.PrimaryKeyNotExistsException: Primary key does
not exist: xpath = //AF_CONFIG[cfg_adapter='MySAP' and cfg_name='jca_sample'
and cfg_target='service_mysap_isdsrv2_ch2']
at com.iwaysoftware.iwrepository.xml.file.FileDataObjectFactory.getDataObject
(FileDataObjectFactory.java:86)
```

- Executing the `sapupd update` command to update the adapter channels with the `sap2to3tgt.tab` file, which contains the adapter target parameters, generates the following exception:

```
C:\soadp1\adapters\application\etc\util>sapupd update jca_sample channels
sap2to3 sap2to3tgt.tab -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@192.168.128.164:1521:orcl scott scott1
```

```
Exception in thread "main"
com.iwaysoftware.iwrepository.PrimaryKeyNotExistsException: Primary key does
not exist: xpath = //AF_CONFIG[cfg_adapter='MySAP' and cfg_name='jca_sample'
and cfg_channel='service_mysap_isdsrv2_tgt']
at com.iwaysoftware.iwrepository.xml.file.FileDataObjectFactory.getDataObject
(FileDataObjectFactory.java:86)
```

- After the connection parameters are updated successfully for the Oracle Fusion Middleware Application Adapter for SAP R/3 (SAP JCo 3.x), if you try to connect to the J2CA configuration using the SAPJCO 2.x libraries, the following exception is generated:

```
Jca could not initialize
```

- Executing the `sapupd list` command to list the targets when no targets are available results in the creation of an empty output file (0 KB in size).
- Executing the `sapupd list` command to list the channels when no channels are available results in the creation of an empty output file (0 KB in size).
- Executing the `sapupd update` command without specifying a file name generates the following exception:

```
C:\soadp1\adapters\application\etc\util>sapupd update jca_sample targets
sap2to3 -file
C:\soadp1\adapters\application\config\jca_sample\repository.xml
```

```
Exception in thread "main" java.lang.Exception: File -file does not exist.
at com.iwaysoftware.iwrepocmd.sap.Script.update(Script.java:67)
at com.iwaysoftware.iwrepocmd.sap.Script.run(Script.java:83)
at com.iwaysoftware.iwrepocmd.CommandBase.run(CommandBase.java:86)
at com.iwaysoftware.iwrepocmd.sap.Command.main(Command.java:8)
```

- Executing the `sapupd update` command to update the targets in a repository where no targets are available in the J2CA configuration (for example, empty `repository.xml`), generates the following exception:

```
C:\soadp1\adapters\application\etc\util>sapupd update jca_sample targets
sap2to3 sap2to3tgt.tab -file
```

```
C:\soadp1\adapters\application\config\jca_sample\repository.xml
```

```
Exception in thread "main"
com.iwaysoftware.iwrepository.PrimaryKeyNotExsistsException: Primary key does
not exist: xpath = //AF_CONFIG[cfg_adapter='MySAP' and cfg_name='jca_sample'
and cfg_target='service_isdsrv2_tgt']
    at
com.iwaysoftware.iwrepository.xml.file.FileDataObjectFactory.getDataObject(File
DataObjectFactory.java:86)
    at
com.iwaysoftware.iwrepocmd.sap.Sap2to3.updateDescriptor(Sap2to3.java:273)
    at
com.iwaysoftware.iwrepocmd.sap.Sap2to3.updateTargets(Sap2to3.java:303)
```

- Executing the `sapupd update` command to update the channels in a repository where no channels are available in the J2CA configuration (for example, empty `repository.xml`), generates the following exception:

```
C:\soadp1\adapters\application\etc\util>sapupd update jca_sample channels
sap2to3 sap2to3chn.tab -file
C:\soadp1\adapters\application\config\jca_sample\repository.xml
```

```
Exception in thread "main"
com.iwaysoftware.iwrepository.PrimaryKeyNotExsistsException: Primary key does
not exist: xpath = //AF_CONFIG[cfg_adapter='MySAP' and cfg_name='jca_sample'
and cfg_channel='channel_isdsrv2_chn1']
    at
com.iwaysoftware.iwrepository.xml.file.FileDataObjectFactory.getDataObject(File
DataObjectFactory.java:86)
    at
com.iwaysoftware.iwrepocmd.sap.Sap2to3.updateDescriptor(Sap2to3.java:273)
    at
com.iwaysoftware.iwrepocmd.sap.Sap2to3.updateChannels(Sap2to3.java:325)
```

- Executing the `sapupd list` command using the incorrect user name to the database repository, generates the following exception:

```
C:\soadp1\adapters\application\etc\util>sapupd list jca_sample channels sap2to3
-jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@192.168.128.164:1521:orcl sct scott1 > sap2to3.tab
java.sql.SQLException: ORA-01017: invalid username/password; logon denied
at oracle.jdbc.driver.DatabaseError.throwSQLException(DatabaseError.java:112)
    at oracle.jdbc.driver.T4CTTIoer.processError(T4CTTIoer.java:331)
```

- Executing the `sapupd list` command using the incorrect password to the database repository, generates the following exception:

```
C:\soadp1\adapters\application\etc\util>sapupd list jca_sample channels sap2to3
-jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@192.168.128.164:1521:orcl scott ott > sap2to3.tab
java.sql.SQLException: ORA-01017: invalid username/password; logon denied
    at
oracle.jdbc.driver.DatabaseError.throwSQLException(DatabaseError.java:112)
    at oracle.jdbc.driver.T4CTTIoer.processError(T4CTTIoer.java:331)
    at oracle.jdbc.driver.T4CTTIoer.processError(T4CTTIoer.java:283)
    at oracle.jdbc.driver.T4CTTIoer.processError(T4CTTIoer.java:278)
```

- Executing the `sapupd list` command using the incorrect URL to the database repository, generates the following exception:

```
C:\soadp1\adapters\application\etc\util>sapupd list jca_sample channels sap2to3
```

```
-jdbc oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@192.168.128.164:1521:rl
scott scott1 > sapt.tab
java.sql.SQLException: Listener refused the connection with the following
error:
ORA-12505, TNS:listener does not currently know of SID given in connect
descriptor
The Connection descriptor used by the client was:192.168.128.164:1521:rl
```

- Executing the `sapupd list` command using the incorrect driver to the database repository, generates the following exception:

```
C:\soadp1\adapters\application\etc\util>sapupd list jca_sample channels sap2to3
-jdbc oracle.jdbc.driverr jdbc:oracle:thin:@192.168.128.164:1521:orcl scott
scott1 > sapt.tab
java.lang.ClassNotFoundException: oracle.jdbc.driverr
    at java.net.URLClassLoader$1.run(URLClassLoader.java:200)
    at java.security.AccessController.doPrivileged(Native Method)
```

- Executing the `sapupd list` command without specifying a driver to the database repository, generates the following exception:

```
C:\soadp1\adapters\application\etc\util>sapupd list jca_sample channels sap2to3
-jdbc jdbc:oracle:thin:@192.168.128.164:1521:orcl scott scott1 > sapt.tab
Exception in thread "main" java.lang.NullPointerException
    at
com.iwaysoftware.iwrepository.IWRepositoryFactory.getRepositoryClassName(IWRepo
sitoryFactory.java:196)
    at
com.iwaysoftware.iwrepository.IWRepositoryFactory.getRepository(IWRepositoryFac
tory.java:163)
    at
com.iwaysoftware.iwrepcmd.CommandBase.readOptions(CommandBase.java:67)
```

- Executing the `sapupd list` command without specifying a user name to list sap2 / sap3 in the database repository, generates the following exception:

```
C:\soadp1\adapters\application\etc\util>sapupd list jca_sample channels sap2to3
-jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@192.168.128.164:1521:orcl scott1 > sap2to3.tab
java.lang.NullPointerException
    at java.util.Hashtable.put(Hashtable.java:394)
    at
com.iwaysoftware.iwrepository.rdbms.oracle.OracleRepository.open(OracleReposito
ry.java:144)
```

- Executing the `sapupd list` command without specifying a password to list sap2 / sap3 in the database repository, generates the following exception:

```
C:\soadp1\adapters\application\etc\util>sapupd list jca_sample channels sap2to3
-jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@192.168.128.164:1521:orcl scott > sapt.tab
java.lang.NullPointerException
    at java.util.Hashtable.put(Hashtable.java:394)
    at
com.iwaysoftware.iwrepository.rdbms.oracle.OracleRepository.open(OracleReposito
ry.java:144)
    at
com.iwaysoftware.iwrepository.IWRepositoryFactory.getRepo(IWRepositoryFactory.j
ava:286)
```

- Executing the `sapupd` update command with an incorrect user name to update sap2 to sap3 in the database repository, generates the following exception:

```
C:\soadp1\adapters\application\etc\util>sapupd update jca_sample targets
sap2to3 saptgt.tab -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@192.168.128.164:1521:orcl sco scott1
java.sql.SQLException: ORA-01017: invalid username/password; logon denied
at
oracle.jdbc.driver.DatabaseError.throwSQLException(DatabaseError.java:112)
at oracle.jdbc.driver.T4CTTIoer.processError(T4CTTIoer.java:331)
at oracle.jdbc.driver.T4CTTIoer.processError(T4CTTIoer.java:283)
at oracle.jdbc.driver.T4CTTIoer.processError(T4CTTIoer.java:278)
```

- Executing the `sapupd` update command with an incorrect password to update sap2 to sap3 in the database repository, generates the following exception:

```
C:\soadp1\adapters\application\etc\util>sapupd update jca_sample targets
sap2to3 saptgt.tab -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@192.168.128.164:1521:orcl scott scpt1
java.sql.SQLException: ORA-01017: invalid username/password; logon denied
at
oracle.jdbc.driver.DatabaseError.throwSQLException(DatabaseError.java:112)
at oracle.jdbc.driver.T4CTTIoer.processError(T4CTTIoer.java:331)
at oracle.jdbc.driver.T4CTTIoer.processError(T4CTTIoer.java:283)
at oracle.jdbc.driver.T4CTTIoer.processError(T4CTTIoer.java:278)
```

- Executing the `sapupd` update command without specifying a user name to update sap2 to sap3 in the database repository, generates the following exception:

```
C:\soadp1\adapters\application\etc\util>sapupd update jca_sample channels
sap2to3 saptch.tab -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@192.168.128.164:1521:orcl scott1
java.lang.NullPointerException
at java.util.Hashtable.put(Hashtable.java:394)
at
com.iwaysoftware.iwrepository.rdbms.oracle.OracleRepository.open(OracleRepository.java:144)
```

- Executing the `sapupd` update command without specifying a file name to update sap2 to sap3 in the database repository, generates the following exception:

```
C:\soadp1\adapters\application\etc\util>sapupd update jca_sample targets
sap2to3
-jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@192.168.128.164:1521:orcl scott scott1
Exception in thread "main" java.lang.Exception: File -jdbc does not exist.
at com.iwaysoftware.iwrepocmd.sap.Script.update(Script.java:67)
at com.iwaysoftware.iwrepocmd.sap.Script.run(Script.java:83)
at com.iwaysoftware.iwrepocmd.CommandBase.run(CommandBase.java:86)
at com.iwaysoftware.iwrepocmd.sap.Command.main(Command.java:8)
```

- Executing the `sapupd` update command without specifying a password to update sap2 to sap3 in the database repository, generates the following exception:

```
C:\soadp1\adapters\application\etc\util>sapupd update jca_sample channels
sap2to3 sapch.tab -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@192.168.128.164:1521:orcl scott
java.lang.NullPointerException
at java.util.Hashtable.put(Hashtable.java:394)
at
com.iwaysoftware.iwrepository.rdbms.oracle.OracleRepository.open(OracleRepository.java:144)
```

```
        at  
        com.iwaysoftware.iwrepository.IWRepositoryFactory.getRepo(IWRepositoryFactory.j  
        ava:286)
```

1.4 SAP2_SAP3 BSE Migration Utility

This section describes the SAP2_SAP3 BSE migration utility. It contains the following topics:

- [Section 1.4.1, "Introduction"](#)
- [Section 1.4.2, "Supported Releases"](#)
- [Section 1.4.3, "Migration Utility Installation Overview"](#)
- [Section 1.4.4, "Working With the Migration Utility"](#)
- [Section 1.4.5, "Use Case Scenario"](#)
- [Section 1.4.6, "Useful Considerations"](#)
- [Section 1.4.7, "Best Practices"](#)
- [Section 1.4.8, "Troubleshooting"](#)

1.4.1 Introduction

A new version of the Oracle Fusion Middleware Application Adapter for SAP R/3 is available that supports SAP Java Connector (SAP JCo) version 3.x. The SAP JCo API has changed by providing a different set of connection parameters to connect with the SAP R/3 server. As a result, the new Oracle Fusion Middleware Application Adapter for SAP R/3 has a different set of connection parameters than the previous version, which supported SAP JCo version 2.x. This introduced a backward compatibility issue by preventing users to reuse adapter targets and Web services that were created by the previous version of the SAP R/3 adapter with the new version of the adapter that supports SAP JCo version 3.x. To enable the transition from the previous version of the SAP R/3 adapter, a command line migration utility is available that can be used to migrate the adapter targets and Web services from the SAP R/3 adapter (using SAP JCo 2.x) to the SAP R/3 adapter (using SAP JCo 3.x).

This migration utility cannot completely automate the migration of the targets between the SAP R/3 adapters because of the incompatibility between the SAP JCo versions. As a result, some manual operation is required. The scope of the migration utility is to provide adapter target and Web services migration capabilities between both versions of the SAP R/3 adapter. The migration utility does not provide any other functionality and cannot be used for any other purposes. In this appendix, the SAP R/3 adapter (using SAP JCo 2.x) is referred to as the MySAP2 or sap2 adapter and the SAP R/3 adapter (using SAP JCo 3.x) is referred to as the MySAP3 or sap3 adapter.

1.4.2 Supported Releases

The following Oracle releases are supported:

- 10.1.3.4
- 11g PS1
- 11g PS2

The BSE migration utility supports all databases. However, for illustration purposes, examples provided in this section use parameters related to the Oracle database. For

other databases, use the appropriate URL and JDBC driver. The utility works with file repositories, but migration of file repositories is not supported.

1.4.3 Migration Utility Installation Overview

The migration utility is located in the following directory:

```
<ADAPTER_HOME>\etc\util
```

The contents of the `util` folder include:

- The `iwrepocmd.jar` file, which is the required .jar file for the migration utility.

Note: The `iwrepocmd.jar` file must not be used for any other purpose other than the migration. This file must not be part of the classpath or any other path when the adapter is running for design time or runtime purposes.

- The `sapupd.bat` file, which is the migration utility .bat file for Windows platforms.
- The `sapupd.sh` file, which is the migration utility .sh file for UNIX/Linux platforms.

Copy the database JDBC driver files into the `<ADAPTER_HOME>\lib` folder or add it into the classpath. JDBC driver files are required by the migration utility to connect to the BSE repository in the database.

1.4.4 Working With the Migration Utility

This section describes the syntax that must be used with the migration utility.

Syntax for Listing the Targets or Web Services

```
sapupd list -ibse [targets | services] [sap2 | sap3 | sap2to3] [-jdbc driver url
user password | -file file] > [output file]
```

Table 1–3 Syntax for Listing the Targets or Web Services

Syntax	Description
<code>sapupd list</code>	Command to list the adapter targets or Web services.
<code>-ibse</code>	Tells the migration utility that the command is executed for BSE.
<code>targets services</code>	List of targets or Web services.
<code>sap2 sap3 sap2to3</code>	List of sap2 objects, sap3 objects, or sap2 objects as mapped to sap3 parameters.
<code>-jdbc driver url user password -file file</code>	Repository to connect to, which can be a file or Oracle database repository. The JDBC URL is used to connect to the Oracle database repository.
<code>output file</code>	List of adapter targets or Web services are redirected to the output file, which must use the .tab extension.

Usage considerations:

- Use the `sapupd list` command to list the targets or Web services from a file or Oracle database BSE repository. The output consists of tab-delimited text with targets or Web services as columns and parameters as rows.
- The `sap2` and `sap3` options are provided to allow you to list the respective SAP R/3 objects from the repository.
- When you run the migration utility, you can list only the adapter targets or Web services. Adapter targets or Web services cannot be combined to create a single Excel spreadsheet.
- The output file name must use the `.tab` extension, since that extension is recognized by Excel for the migration. In this appendix, the output file uses the `sap2to3.tab` naming convention.
- To replace `sap2` objects with `sap3` objects, the `sap2to3` option can be used.
In the following examples, the `sapupd list` command lists all `sap2` targets mapped to `sap3` parameters into the `sap2to3.tab` file.

File Repository Example

```
sapupd list -ibse targets sap2to3 -file <DomainHome>\servers\server_
name\stage\ibse\ibse.war\ibserrepo.xml > sap2to3_tgt.tab
sapupd list -ibse services sap2to3 -file <DomainHome>\servers\server_
name\stage\ibse\ibse.war\ibserrepo.xml > sap2to3_serv.tab
```

For UNIX/Linux platforms platforms, execute `sapupd.sh`. For example:

```
./sapupd.sh list -bse targets sap2to3 -file <DomainHome>\servers\server_
name\stage\ibse\ibse.war\ibserrepo.xml > sap2to3_tgt.tab
./sapupd.sh list -bse services sap2to3 -file <DomainHome>\servers\server_
name\stage\ibse\ibse.war\ibserrepo.xml > sap2to3_serv.tab
```

In this example:

- `DomainHome` is the location where the Oracle WebLogic domain is installed.
- `ibse` represents that the command is executed for BSE.

Database Repository Example

To list targets:

```
sapupd list -ibse targets sap2to3 -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@192.168.128.164:1521:orcl scott scott1 > dbtgt.tab
sapupd list -ibse services sap2to3 -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@192.168.128.164:1521:orcl scott scott1 > dbserv.tab
```

For UNIX/Linux platforms, execute the `sapupd.sh` file. For example, the following command will list all SAP2 targets/services mapped to SAP3 parameters into the file `dbser.tab` for a database repository.

```
./sapupd.sh list -bse targets sap2to3 -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@172.19.20.242:1521:orcl system welcome1 > dbtgt.tab
./sapupd.sh list -bse services sap2to3 -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@172.19.20.242:1521:orcl system welcome1 > dbserv.tab
```

Syntax for Updating the Targets or Web Services

```
sapupd update -ibse [targets | services] sap2to3[source file] [-jdbc driver url
user password | -file file]
```

Table 1–4 Syntax for Updating the Targets or Web Services

Syntax	Description
<code>sapupd update</code>	Command to update the adapter targets or Web services.
<code>-ibse</code>	Tells the utility that the command is executed for BSE.
<code>targets services</code>	List of targets or Web services to update.
<code>source file</code>	The source Excel file that contains the mappings between sap2 and sap3 parameters.
<code>-jdbc driver url</code> <code>user password </code> <code>-file file</code>	Repository to connect to, which can be a file or Oracle database repository. The JDBC URL is used to connect to the Oracle database repository.

Use the `sapupd update` command to update the targets or Web services in a file or Oracle database repository.

File Repository Example

```
sapupd update -ibse targets sap2to3 sap2to3_tgt.tab -file
<DomainHome>\servers\server_name\stage\ibse\ibse.war\ibserepo.xml
sapupd update -ibse services sap2to3 sap2to3_serv.tab -file
<DomainHome>\servers\server_name\stage\ibse\ibse.war\ibserepo.xml
```

For UNIX/Linux platforms, run the `sapupd.sh` file. For example:

```
./sapupd.sh update -ibse targets sap2to3 sap2to3_tgt.tab -file
<DomainHome>\servers\server_name\stage\ibse\ibse.war\ibserepo.xml
./sapupd.sh update -ibse services sap2to3 sap2to3_serv.tab -file
<DomainHome>\servers\server_name\stage\ibse\ibse.war\ibserepo.xml
```

Database Repository Example

```
sapupd update -ibse targets sap2to3 dbtgt.tab -jdbc
oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@192.168.128.164:1521:orcl scott
scott1
sapupd update -ibse services sap2to3 dbserv.tab -jdbc
oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@192.168.128.164:1521:orcl scott
scott1
```

For UNIX/Linux platforms, run the `sapupd.sh` file. For example:

```
./sapupd.sh update -ibse targets sap2to3 dbtgt.tab -jdbc
oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@172.19.20.242:1521:orcl system
welcome1
./sapupd.sh update -ibse services sap2to3 dbserv.tab -jdbc
oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@172.19.20.242:1521:orcl system
welcome1
```

A manual step must be performed before you create the list and update the BSE repository. This is the process of mapping the values between sap2 and sap3 adapter targets and Web services. At this stage, you must take a look at the values underneath each adapter target/service column and update the output file that has been created accordingly. While editing, do not remove or add rows and columns to the file. Also, do not change the SAP R/3 connection parameters in the file. The only action allowed is to update the parameter values.

Figure 1–7 shows sample output from a repository with one SAP application server target, `sap_new_tgt_app_serv`, and one SAP message server target, `sap_new_tgt_msg_serv`, as opened in Excel. The SAP R/3 parameters have been pre populated with the

matching SAP2 parameter values. If no matching value was present in the SAP2 target, the SAP3 parameter default is used.

Figure 1–8 Sample Mapping Document

A	B
Application Server	
	sap_new_tgt_app_serv
jco.dient.dient	800
user	iwayqa
password	ENCR(3253310131831131222333215313732323192322731773156)
jco.dient.lang	EN
jco.dient.codepage	
authentication	Password
jco.dient.ashost	172.30.162.52
jco.dient.sysnr	0
jco.destination.peak_limit	2
jco.destination.expiration_time	10
jco.destination.max_get_dient_time	30
edi_version	3
edi_release	
edi_release_option	IDOC DOCREL field
exception	Throws Exception
commitwait	FALSE
jco.dient.trace	TRUE
trace_level	0
jco.dient.snc_mode	0
jco.dient.snc_partnername	
jco.dient.snc_qop	3
jco.dient.snc_myname	
jco.dient.snc_lib	
Message Server	
	sap_new_tgt_msg_serv
jco.dient.dient	800
user	iwayqa
password	ENCR(3221324531831131222333215315332323192322731773188)
jco.dient.lang	EN
jco.dient.codepage	
authentication	Password

You can add or modify parameter values as required, then save the file and use it to replace existing sap2 objects with sap3 objects.

The update command can be used only to replace existing sap2 targets or Web services with their sap3 equivalents. It cannot be used to create new targets or Web services or to edit the parameter values of existing targets and Web services. Also, you cannot change the adapter target or Web service names in the file.

1.4.5 Use Case Scenario

This section provides a use case scenario for the migration utility.

1. Close Application Explorer. Ensure that no instance of Application Explorer is running and that no updates are being made in the BSE repository.
2. Use the `sapupd list` command to list the connection parameters into an Excel spreadsheet. This command inserts the sap2 targets or Web services into the spreadsheet. Running this command only extracts targets or Web services. As a result, first run the command to list the adapter targets and then repeat the

command to list the Web services. Ensure to save the contents in separate files for adapter targets and Web services.

3. Make an required updates to the Excel spreadsheet. This is the step where the Excel spreadsheet is updated with the sap3 targets and Web services.
4. Stop Oracle WebLogic Server if it is running. Remove the SAP R/3 adapter (using SAP JCo 2.x) files and any SAP JCo 2.x library files from the adapter environment.
5. Install the SAP R/3 adapter (using SAP JCo 3.x) in the environment. This adapter is delivered as a patch for 10.1.3.4 and 11g PS1 releases. It is a part of the application adapters in the 11g PS2/PS3/PS4/PS5/PS6 releases.
6. If you are using a 10.1.3.4 adapter environment, then check the installation documentation when making any changes to configuration files (for example, server.xml).
7. Use the `sapupd update` command to update the connection parameters for sap3 targets or Web services. Running this command only updates adapter targets or Web services from different files. As a result, repeat the steps accordingly with the correct input file. As a best practice, first update the adapter targets and then update the adapter Web services.
8. Start Application Explorer and connect to an SAP R/3 adapter (using SAP JCo 3.x) target and ensure that the design time is working.
9. Restart the Oracle WebLogic Server and execute runtime test cases to verify successfully functionality.

1.4.6 Useful Considerations

This section includes useful considerations for the migration utility.

The following is list of actions that are recommended:

- Close Application Explorer before executing the update command. The target or Web service parameters are not updated if Application Explorer is open.
- Running the utility remains the same with a single instance of the Oracle database or with the RAC database. In the RAC database environment, running the utility on one instance updates the repository for any other instances that belong to the environment.
- Modify the script as "2>&1" which is used ">&" in the `sapud.sh` file to work in the 'k' shell for non-windows as given below. (otherwise the utility works fine with the warning message `./sapupd.sh: /dev/null: bad number`)

Original script:

```
LOCAL_CLASSPATH=`find ../../lib/*.jar >& /dev/null | tr '\n' ':' `
```

Modified script:

```
LOCAL_CLASSPATH=`find ../../lib/*.jar 2>&1 /dev/null | tr '\n' ':' `
```

The following is list of actions that must be avoided:

- Do not add, delete or change the rows in the Excel spreadsheet that was created by the upgrade utility.
- Do not rename the targets or Web services in the Excel spreadsheet that was created by the upgrade utility.

- Do not add or delete the column in the Excel spreadsheet that was created by the upgrade utility.
- This utility cannot be used to migrate the adapter targets and Web services from one environment to another environment. For example, migration between development, testing, and production environments cannot be performed. You must use the migration utility in each of the environments separately.

1.4.7 Best Practices

This section describes best practices for the migration utility.

- Create a backup of the Oracle database which contains the BSE repository. This applies irrespective of development, testing, and production environments. In case of a file repository, create a backup of the file repository. This also applies irrespective of development, testing, and production environments. The file repository is located in the adapter folder structure. For example:

```
<DomainHome>\servers\server_name\stage\ibse\ibse.war
```

- When the migration utility extracts information into the Excel spreadsheet, the Excel spreadsheet takes 00 as 0 for the system number parameter value for the sap2 target. When the sap2 target values are mapped to sap3 from the Excel spreadsheet, the adapter target has the system number parameter value set to 0 in Application Explorer.

There is no issue with the adapter to work with this target in design and run time. However, it is recommended that you to manually update the target for the correct value. For example, replace "0" with "00".

- There is no command to migrate both targets and Web services. This must be done separately
- When performing the upgrade in a production environment, consider the following guidelines:
 1. Extract the sap2 adapter targets and Web services well in advance of the production migration date. Perhaps two to four weeks earlier. This date is just an approximation.
 2. Make changes to the Excel spreadsheet with the new parameter values for the sap3 adapter.
 3. Upload the details from the Excel spreadsheet into the environment which should be an exact replica of the production environment. Ensure that the design time (Application Explorer), run time (BPEL, ESB, Mediator) is working. Note that if there is a change in the environment, then the runtime scenarios based on BPEL, ESB, and Mediator process flows would not work.
 4. If everything is working accordingly, then keep the Excel spreadsheet intact until the production migration.
 5. Do not make any changes to the adapter targets and Web services in the production environment after extracting the details. If you make any changes, then repeat steps 1 through 4.
 6. On the day of the production environment migration, use the Excel spreadsheet to update the sap3 adapter targets and Web services.

This approach minimizes the downtime during the production migration.

7. If your production environment is a high availability (cluster) environment, then perform this procedure for any node that belongs to the cluster.

1.4.8 Troubleshooting

This section provides troubleshooting information for the migration utility.

- Executing the `sapupd update` command to update the adapter targets with the `sap2to3.tab` file, which contains the adapter Web service parameters, generates an exception. Ensure to use the appropriate file.
- After the connection parameters are updated successfully for the MySAP3 adapter, if you try to connect to the BSE configuration with the SAPJCO 2.x libraries, the following exception is generated:

```
ibSE could not initialize
```

- Executing the `sapupd list` command to list the targets when no targets are available results in the creation of an empty output file (0 KB in size).
- Executing the `sapupd list` command to list the Web services when no Web services are available results in the creation of an empty output file (0 KB in size).
- Executing the `sapupd update` command without specifying a file name generates the following exception:

```
sapupd update -ibse targets sap2to3 -file <DomainHome>\servers\server_
name\stage\ibse\ibse.war\ibserrepo.xml
Exception in thread "main" java.lang.Exception: File -file does not exist.
  at com.iwaysoftware.iwrepocmd.sap.Script.update(Script.java:67)
  at com.iwaysoftware.iwrepocmd.sap.Script.run(Script.java:83)
  at com.iwaysoftware.iwrepocmd.CommandBase.run(CommandBase.java:86)
  at com.iwaysoftware.iwrepocmd.sap.Command.main(Command.java:8)
```

- Execution of the `sapupd update` command to update the targets in the repository in which no targets are available or to update the web services where no web services are available in the BSE configuration (for example, empty `ibserrepo.xml`) will result in error.
- Executing the `sapupd list` command using the incorrect user name to the database repository, generates the following exception:

```
sapupd list -ibse targets sap2to3 -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@192.168.128.164:1521:orcl sct scott1 > sap2to3.tab
java.sql.SQLException: ORA-01017: invalid username/password; logon denied
  at oracle.jdbc.driver.DatabaseError.throwSQLException(DatabaseError.java:112)
  at oracle.jdbc.driver.T4CTTIoer.processError(T4CTTIoer.java:331)
```

- Executing the `sapupd list` command using the incorrect URL to the database repository, generates the following exception:

```
sapupd list -ibse targets sap2to3 -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@192.168.128.164:1521:rl scott scott1 > sapt.tab
java.sql.SQLException: Listener refused the connection with the following
error:
ORA-12505, TNS:listener does not currently know of SID given in connect
descriptor
The Connection descriptor used by the client was:192.168.128.164:1521:rl
```

- Executing the `sapupd list` command using the incorrect driver to the database repository, generates the following exception:

```

sapupd list -ibse targets sap2to3 -jdbc oracle.jdbc.driverr
jdbc:oracle:thin:@192.168.128.164:1521:orcl scott scott1 > sapt.tab
java.lang.ClassNotFoundException: oracle.jdbc.driverr
    at java.net.URLClassLoader$1.run(URLClassLoader.java:200)
    at java.security.AccessController.doPrivileged(Native Method)

```

- Executing the `sapupd list` command without specifying a driver to the database repository, generates the following exception:

```

sapupd list -ibse targets sap2to3 -jdbc
jdbc:oracle:thin:@192.168.128.164:1521:orcl scott scott1 > sapt.tab
Exception in thread "main" java.lang.NullPointerException
    at
com.iwaysoftware.iwrepository.IWRepositoryFactory.getRepositoryClassName(IWRepositoryFactory.java:196)
    at
com.iwaysoftware.iwrepository.IWRepositoryFactory.getRepository(IWRepositoryFactory.java:163)
    at
com.iwaysoftware.iwrepcmd.CommandBase.readOptions(CommandBase.java:67)

```

- Executing the `sapupd list` command without specifying a user name to list `sap2 / sap3` in the database repository, generates the following exception:

```

sapupd list -ibse targets sap2to3 -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@192.168.128.164:1521:orcl scott1 > sap2to3.tab
java.lang.NullPointerException
    at java.util.Hashtable.put(Hashtable.java:394)
    at
com.iwaysoftware.iwrepository.rdbms.oracle.OracleRepository.open(OracleRepository.java:144)

```

- Executing the `sapupd update` command with an incorrect user name to update `sap2` to `sap3` in the database repository, generates the following exception:

```

sapupd update -ibse targets sap2to3 saptgt.tab -jdbc
oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@192.168.128.164:1521:orcl sco
scott1
java.sql.SQLException: ORA-01017: invalid username/password; logon denied
    at
oracle.jdbc.driver.DatabaseError.throwSQLException(DatabaseError.java:112)
    at oracle.jdbc.driver.T4CTTIoer.processError(T4CTTIoer.java:331)
    at oracle.jdbc.driver.T4CTTIoer.processError(T4CTTIoer.java:283)
    at oracle.jdbc.driver.T4CTTIoer.processError(T4CTTIoer.java:278)

```

- Executing the `sapupd update` command without specifying a user name to update `sap2` to `sap3` in the database repository, generates the following exception:

```

sapupd update -ibse targets sap2to3 saptch.tab -jdbc
oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@192.168.128.164:1521:orcl
scott1
java.lang.NullPointerException
    at java.util.Hashtable.put(Hashtable.java:394)
    at
com.iwaysoftware.iwrepository.rdbms.oracle.OracleRepository.open(OracleRepository.java:144)

```

- Executing the `sapupd update` command without specifying a file name to update `sap2` to `sap3` in the database repository, generates the following exception:

```

sapupd update -ibse targets sap2to3 -jdbc oracle.jdbc.driver.OracleDriver
jdbc:oracle:thin:@192.168.128.164:1521:orcl scott scott1

```

```
Exception in thread "main" java.lang.Exception: File -jdbc does not exist.  
    at com.iwaysoftware.iwrepocmd.sap.Script.update(Script.java:67)  
    at com.iwaysoftware.iwrepocmd.sap.Script.run(Script.java:83)  
    at com.iwaysoftware.iwrepocmd.CommandBase.run(CommandBase.java:86)  
    at com.iwaysoftware.iwrepocmd.sap.Command.main(Command.java:8)
```

- Executing the `sapupd update` command without specifying a password to update sap2 to sap3 in the database repository, generates the following exception:

```
sapupd update -ibse targets sap2to3 sapch.tab -jdbc  
oracle.jdbc.driver.OracleDriver jdbc:oracle:thin:@192.168.128.164:1521:orcl  
scott  
java.lang.NullPointerException  
    at java.util.Hashtable.put(Hashtable.java:394)  
    at  
com.iwaysoftware.iwrepository.rdbms.oracle.OracleRepository.open(OracleRepository.java:144)  
    at  
com.iwaysoftware.iwrepository.IWRepositoryFactory.getRepo(IWRepositoryFactory.java:286)
```

General Upgrade Guidelines

This chapter lists and describes general upgrade guidelines that are common to all Oracle Application Adapters for Oracle WebLogic Server. It contains the following topics:

- [Section 2.1, "Upgrading a 10.1.3.x Outbound Process to 11g"](#)
- [Section 2.2, "Upgrading a 10.1.3.x Inbound Process to 11g"](#)
- [Section 2.3, "Migrating Oracle Service Bus 10.1.3.x/PS2/PS3/PS4/PS5 Outbound and Inbound Processes to 11g PS6"](#)
- [Section 2.4, "Upgrading 11g PS1/PS2/PS3/PS4/PS5 Outbound and Inbound BPEL and Mediator Processes to 11g PS6"](#)
- [Section 2.5, "Upgrading 11g PS2/PS3/PS4/PS5 Outbound and Inbound BPM Processes to 11g PS6"](#)

2.1 Upgrading a 10.1.3.x Outbound Process to 11g

This section describes how to upgrade a 10.1.3.x outbound process to 11g.

Note: For demonstration purposes, Oracle Application Adapter for SAP R/3 is used as an example.

The 10.1.3.x BPEL and Mediator projects for the application adapters should not have any warning or error messages during compilation and deployment with 10.1.3.x version. If any warning or error messages are present, then they must be corrected before migrating the projects to 11g. If you have followed the 10.1.3.x user guides for creating a BPEL or Mediator project, two warning messages appear in the Oracle JDeveloper at the time of compilation deployment. These warning messages are not acceptable in the 11g environment. Those warning messages are shown because the assign statements were assigning the BPEL/ESB project namespace to the adapter's namespace. This is an incompatible namespace assignment. As a result, any warning or error messages in 10.1.3.x projects must be corrected before migrating to 11g. This section describes how to create 10.1.3.x projects without the warning messages.

2.1.1 Prerequisites

Take a note of the adapter targets and channels that were created in the 10.1.3.x Application Explorer for your application adapters. For the migration to be successful, you must create the same adapter targets and channels in the 11g environment using the Application Explorer.

Before continuing, ensure that the following components are available:

1. Using Application Explorer, export the schemas for a particular business object for Oracle Application Adapter for SAP R/3 into the default location.
2. Using Application Explorer, generate a JCA outbound WSDL for a particular business object for Oracle Application Adapter for SAP R/3.

2.1.2 Overview of 10.1.3.x BPEL JCA Outbound Workflow Process

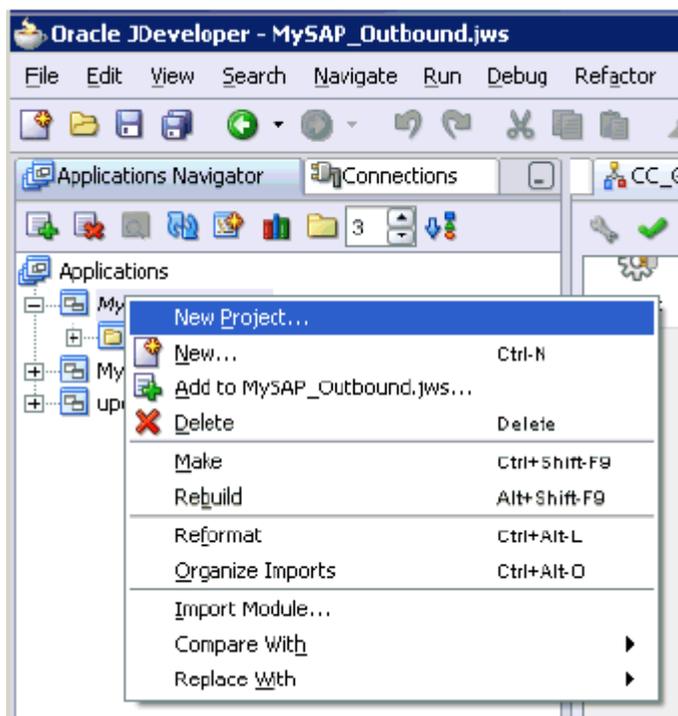
1. Create an outbound BPEL process project.
2. Configure an outbound BPEL process.
3. Deploy the outbound BPEL process project.
4. Invoke the input XML using the BPEL console.

2.1.3 Creating an Outbound BPEL Process Project

To create an outbound BPEL process project:

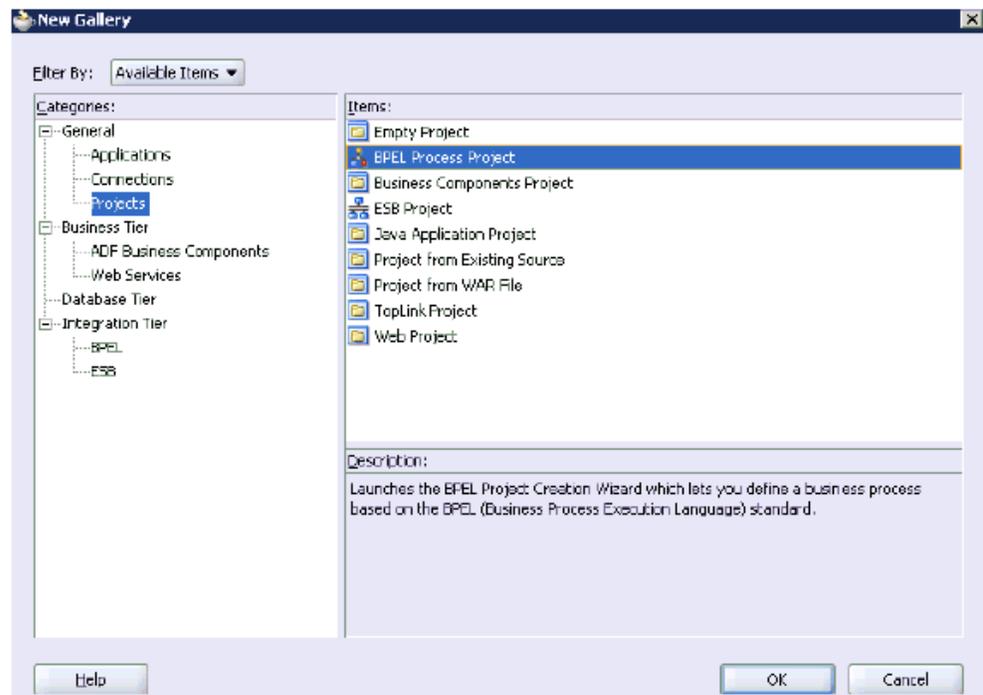
1. As shown in [Figure 2–1](#), start Oracle JDeveloper 10.1.3.4.

Figure 2–1 New Project Option



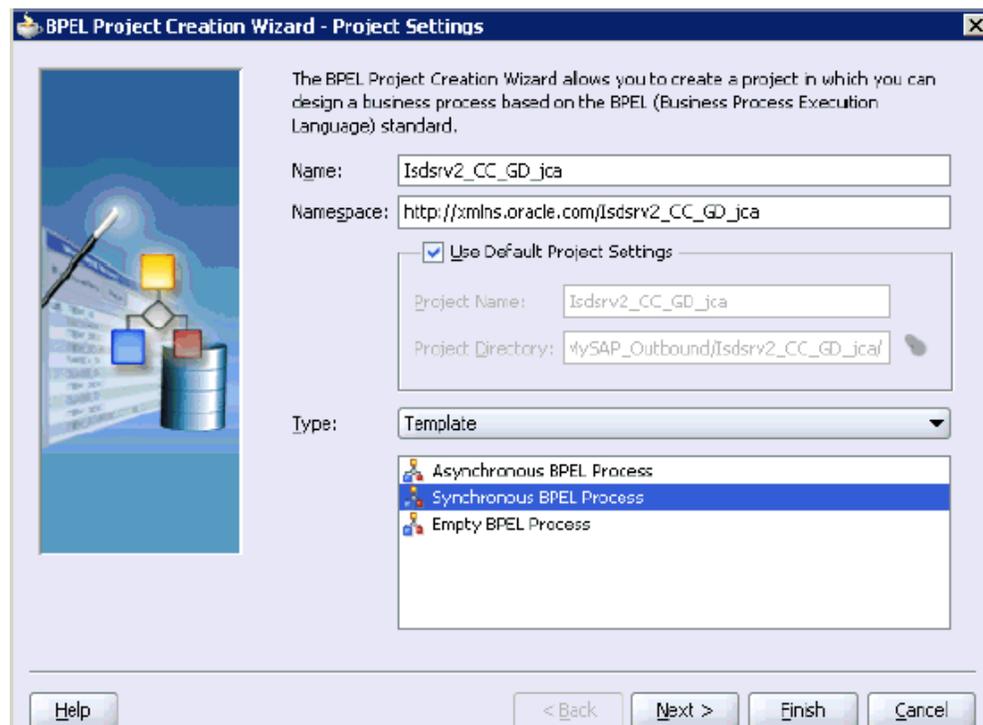
2. Click the **Application Navigator** tab and select **New Project** from the context menu.

The New Gallery dialog is displayed, as shown in [Figure 2–2](#).

Figure 2–2 New Gallery Dialog

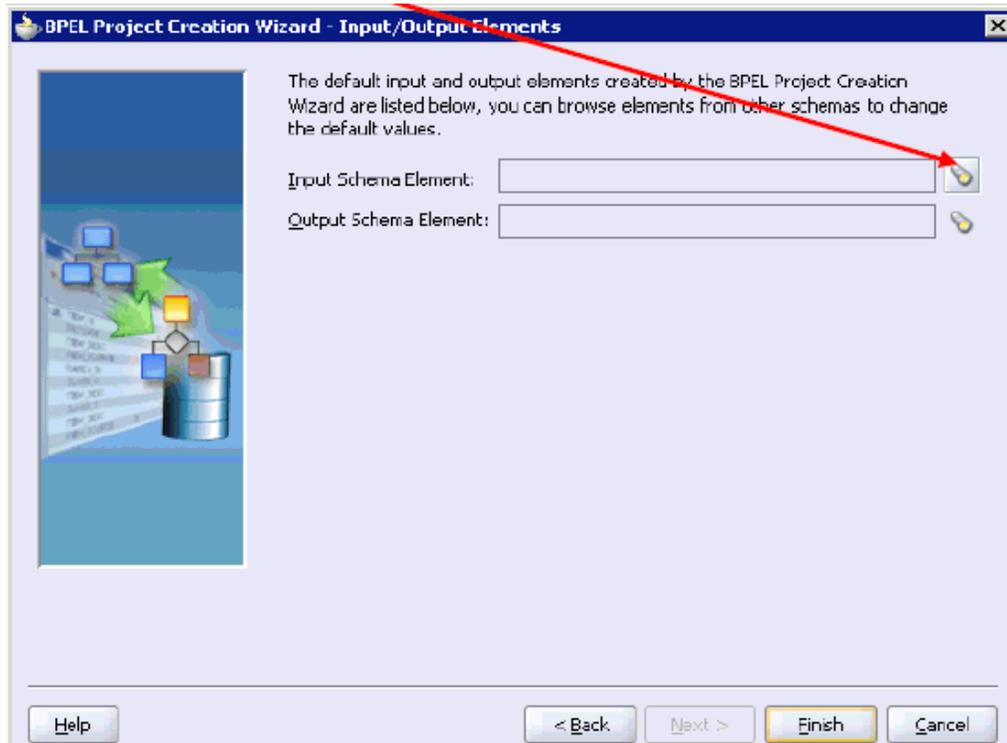
3. Select the **BPEL Process Project** and then click **OK**.

The BPEL Project Creation Wizard - Project Settings page is displayed, as shown in [Figure 2–3](#).

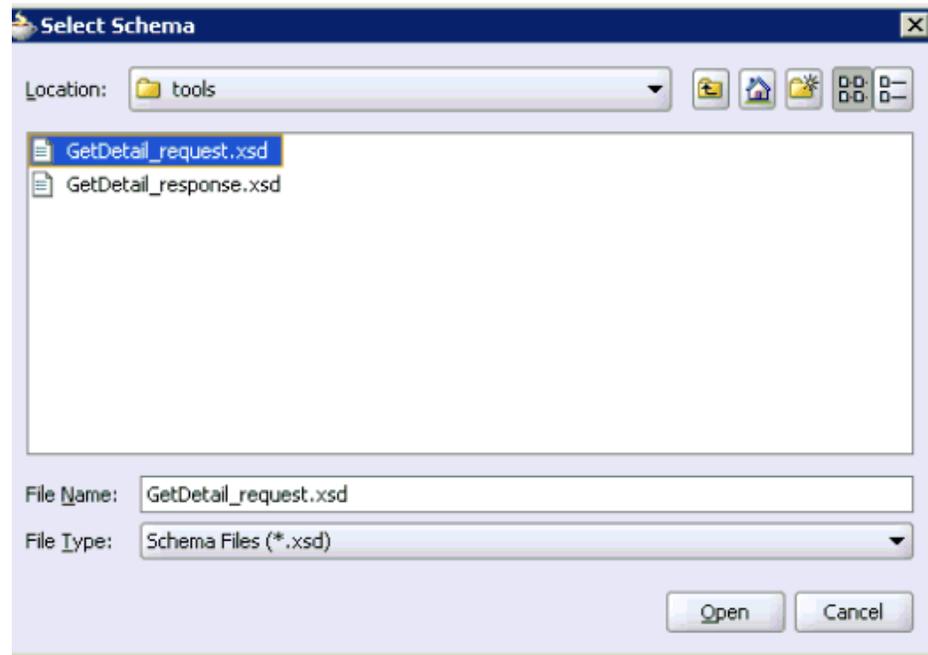
Figure 2–3 BPEL Project Creation Wizard - Project Settings Page

4. Enter a name for the new BPEL process project in the Name field.
5. Select **Synchronous BPEL Process** from the Template Type list and click **Next**.
The BPEL Project Creation Wizard - I/O Elements page is displayed, as shown in [Figure 2-4](#).

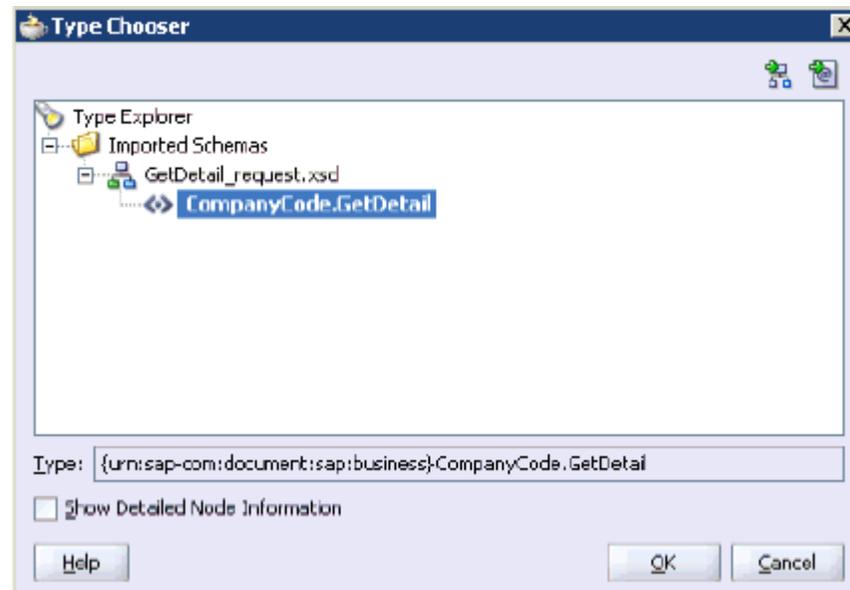
Figure 2-4 BPEL Project Creation Wizard - I/O Elements Page



6. Click the **Browse** icon to the right of the Input Schema Element field.
The Select Schema dialog is displayed, as shown in [Figure 2-5](#).

Figure 2–5 Select Schema Dialog

7. Navigate to the default exported schema location:
`<soadp1>\adapters\application\tools`
8. Select the request XML schema file (.xsd) for the corresponding business object and click **Open**.
 The Type Chooser dialog is displayed, as shown in [Figure 2–6](#).

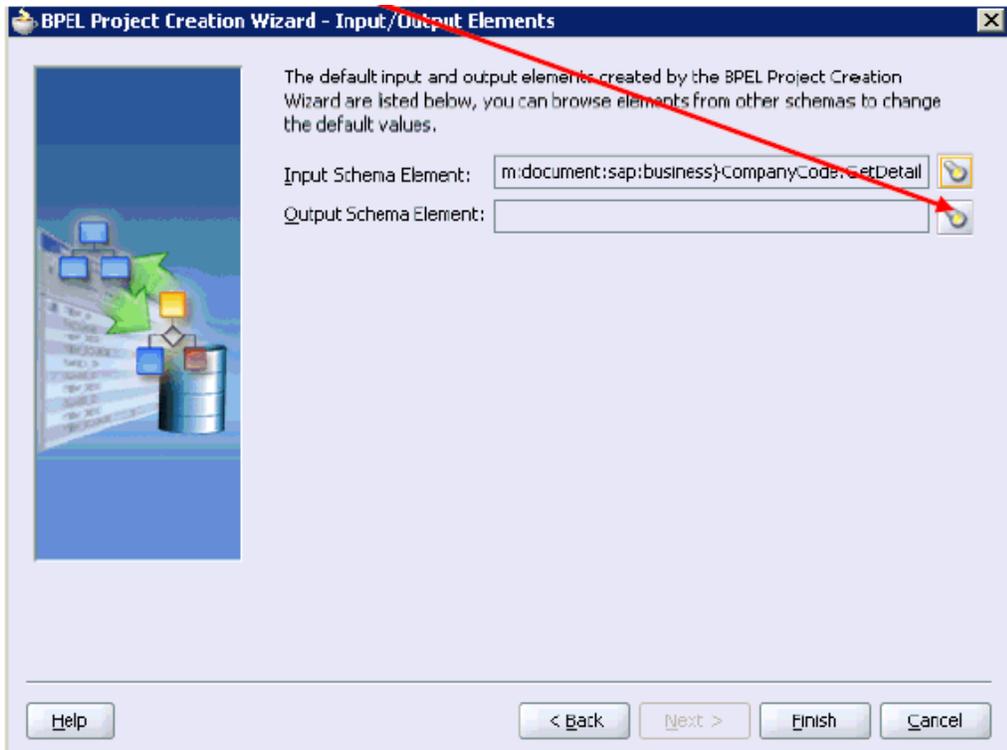
Figure 2–6 Type Chooser Dialog

9. Expand **Imported Schemas**, the request XML schema (for example, **GetDetail_request.xsd**), and then the method (for example, **CompanyCode.GetDetail**).

10. Click **OK**.

You are returned to the BPEL Project Creation Wizard - I/O Elements page, as shown in [Figure 2-7](#).

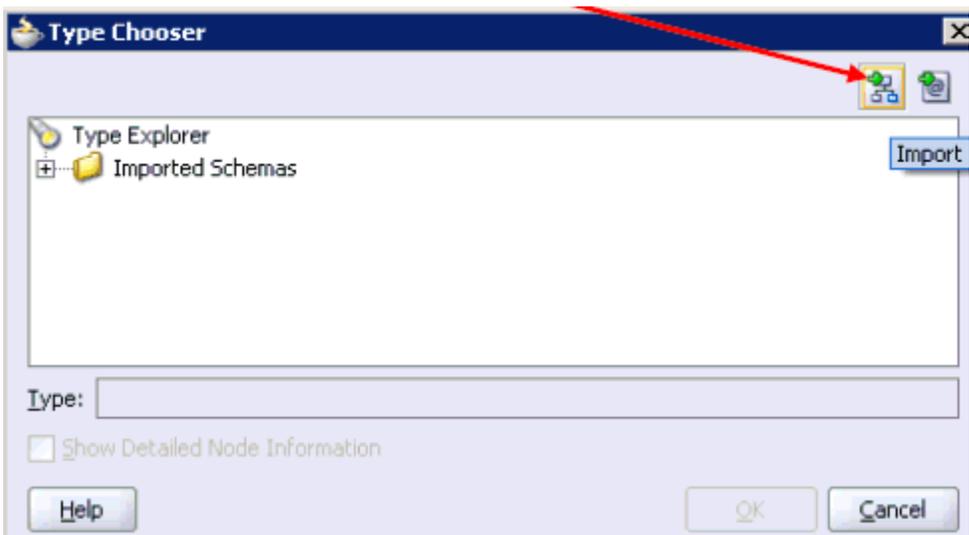
Figure 2-7 BPEL Project Creation Wizard - I/O Elements Page



11. Click the **Browse** icon to the right of the Output Schema Element field.

The Type Chooser dialog is displayed, as shown in [Figure 2-8](#).

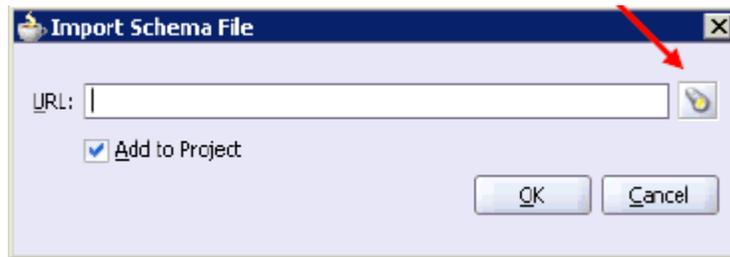
Figure 2-8 Type Chooser Dialog



12. Click the **Import Schema** icon.

The Import Schema File dialog is displayed, as shown in [Figure 2–9](#).

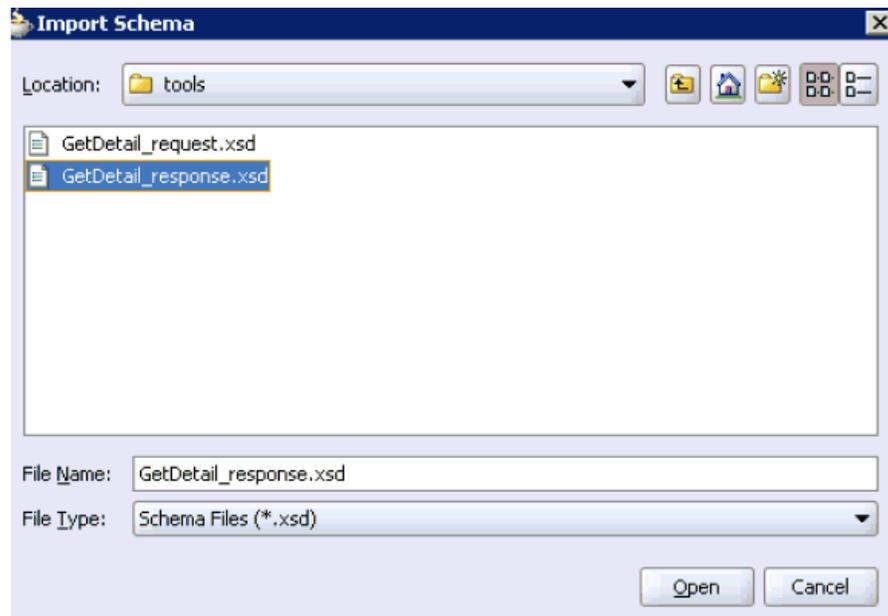
Figure 2–9 Import Schema File Dialog



13. Click the **Browse** icon.

The Import Schema dialog is displayed, as shown in [Figure 2–10](#).

Figure 2–10 Import Schema Dialog



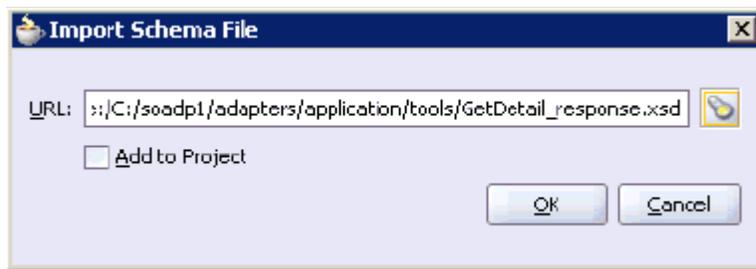
14. Navigate to the default exported schema location:

```
<soadpl>\adapters\application\tools
```

15. Select the response XML schema file (.xsd) for the corresponding business object and click **Open**.

You are returned to the Import Schema File dialog, as shown in [Figure 2–11](#).

Figure 2–11 Import Schema File Dialog

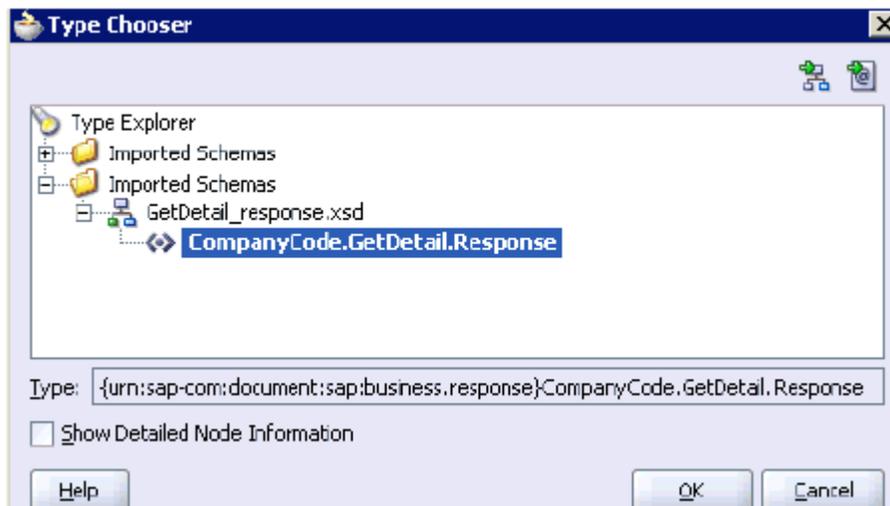


16. Uncheck the **Add to Project** option.

17. Click **OK**.

The Type Chooser dialog is displayed, as shown in [Figure 2–12](#).

Figure 2–12 Type Chooser Dialog

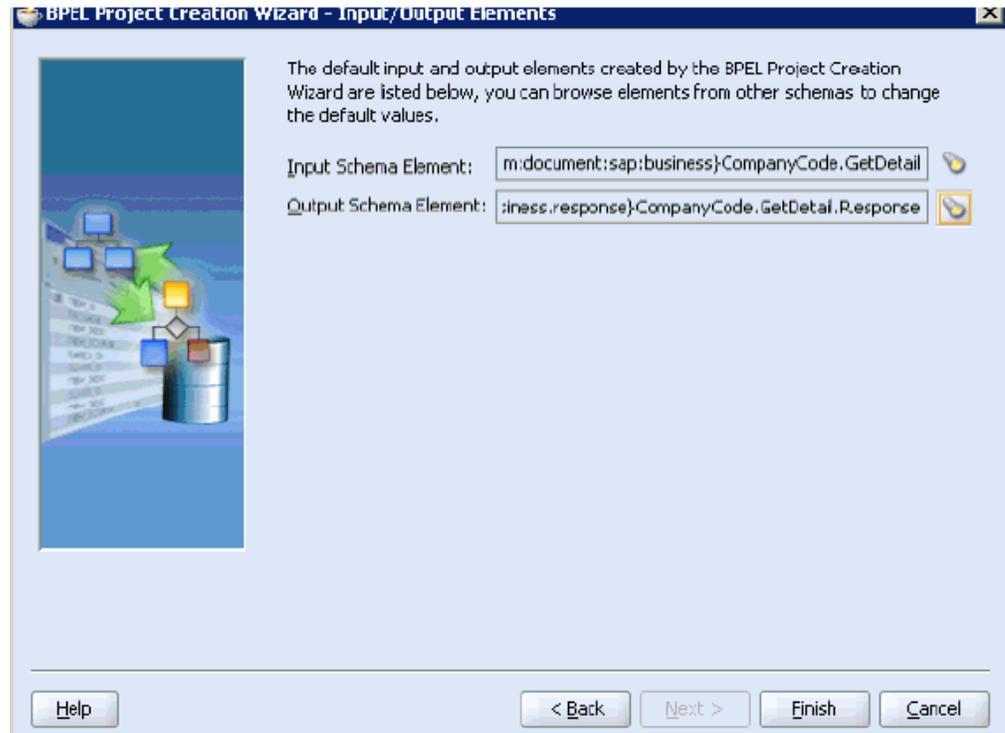


18. Expand **Imported Schemas**, the response XML schema (for example, **GetDetail_response.xsd**), and then the method (for example, **CompanyCode.GetDetail.Response**).

19. Click **OK**.

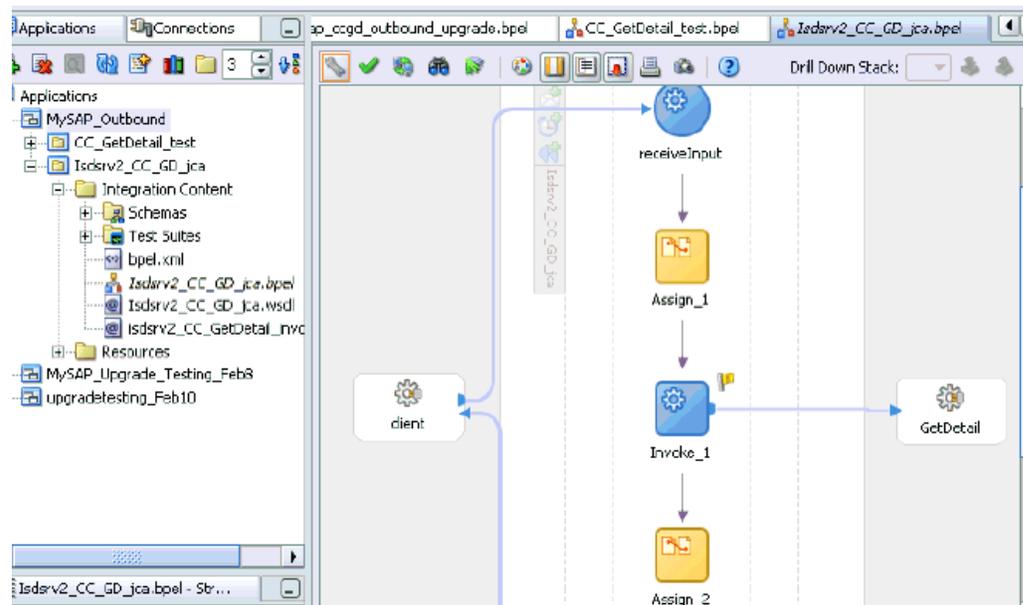
You are returned to the BPEL Project Creation Wizard - I/O Elements page, as shown [Figure 2–13](#).

Figure 2–13 BPEL Project Creation Wizard - I/O Elements Page



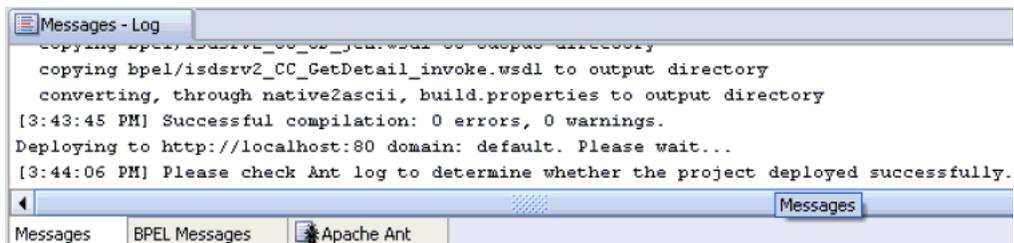
20. Click **Finish**.

Figure 2–14 Standard BPEL Outbound Process



21. Continue with the creation of a standard BPEL outbound process using Partnerlink, Invoke, and Assign components.

22. Compile the outbound BPEL process project, as shown in [Figure 2–15](#).

Figure 2–15 Outbound BPEL Process Project

23. Verify that there are no error or warning messages during compilation.
24. Deploy the outbound BPEL process project.
25. After the deployment is successful, go to the BPEL console and execute the deployed process.
 - A successful response is received.

2.1.4 Migrating the 10.1.3.x BPEL JCA Outbound Process to a 11g Workflow Process

This section describes how to migrate the 10.1.3.x BPEL JCA outbound process to a 11g workflow process.

Note: For demonstration purposes, Oracle Application Adapter for SAP R/3 is used as an example.

2.1.4.1 Prerequisites

Before continuing, ensure that you copy and paste the 10.1.3.x outbound BPEL Process Project to the 11g system location.

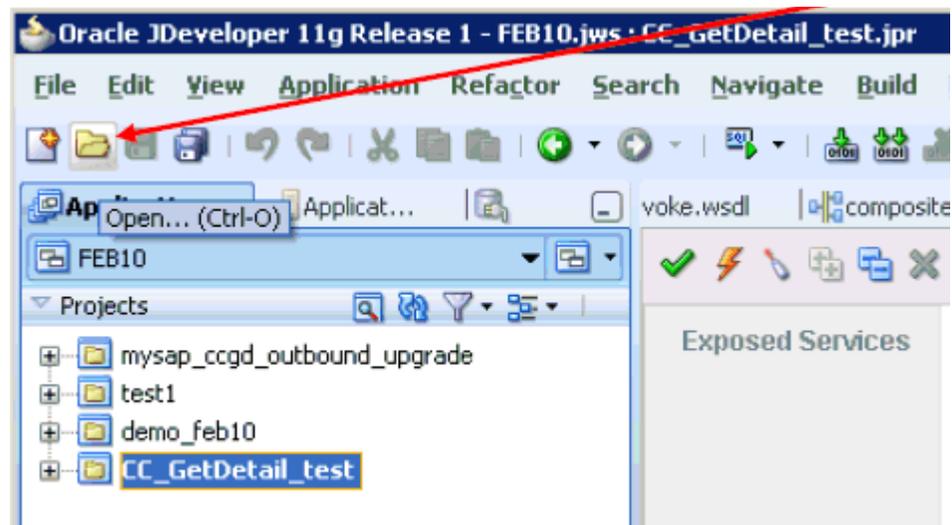
2.1.4.2 Overview of Migration to 11g BPEL JCA Outbound Workflow Process

1. Open an application.
2. Migrate the 10.1.3.x BPEL JCA outbound process to 11g.
3. Deploy the BPEL process project.
4. Invoke the input XML using the Oracle Enterprise Manager console.

2.1.4.3 Migrating an Outbound BPEL Process Project

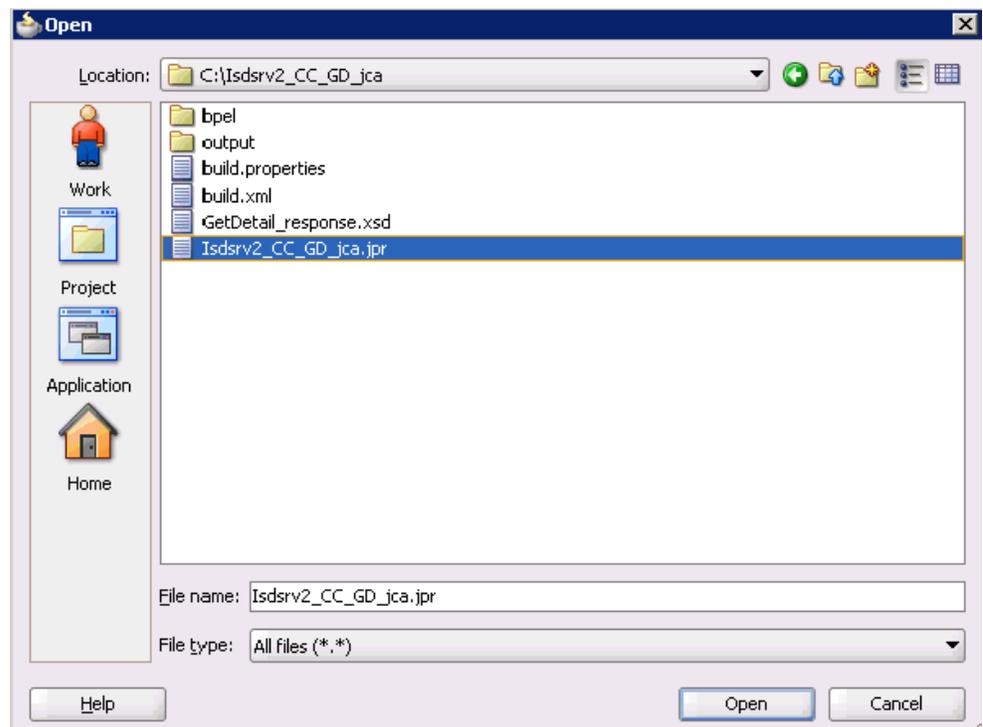
To migrate an outbound BPEL process project:

1. Start Oracle JDeveloper 11g, as shown in [Figure 2–16](#).

Figure 2–16 Open Icon in Oracle JDeveloper 11g

2. Select an available application (for example, FEB10) and click **Open** from the tool bar.

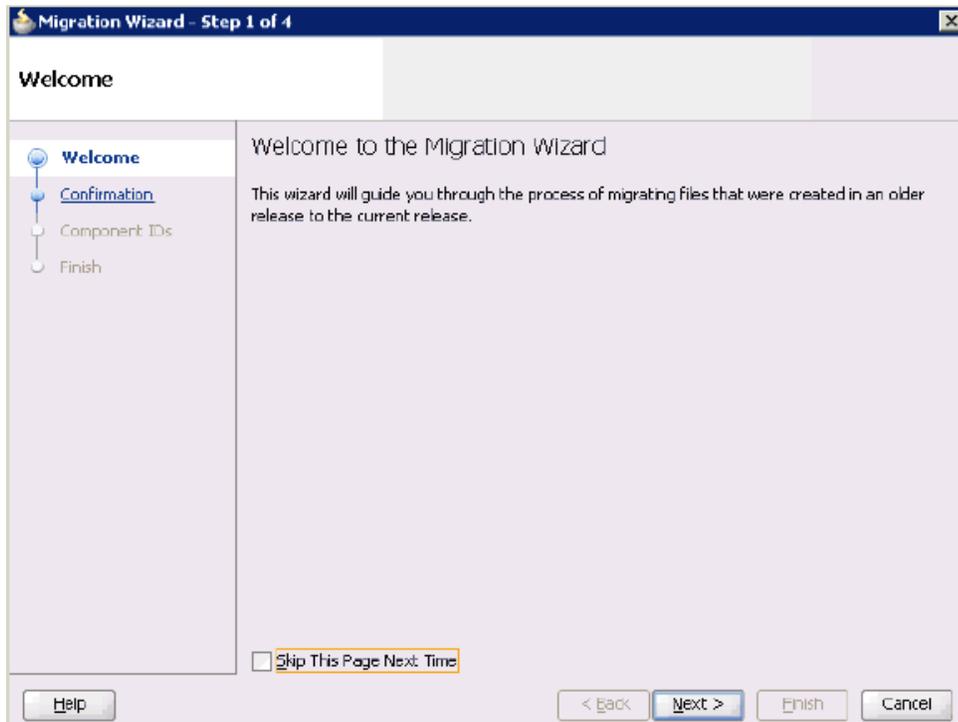
The Open dialog is displayed, as shown in [Figure 2–17](#).

Figure 2–17 Open Dialog

3. Open the 10.1.3.x project (for example, Isdsrv2_CC_GD_jca) and select the **.jpr** extension file (for example, Isdsrv2_CC_GD_jca.jpr).
4. Click **Open**.

The Migration Wizard is displayed, as shown in [Figure 2–18](#).

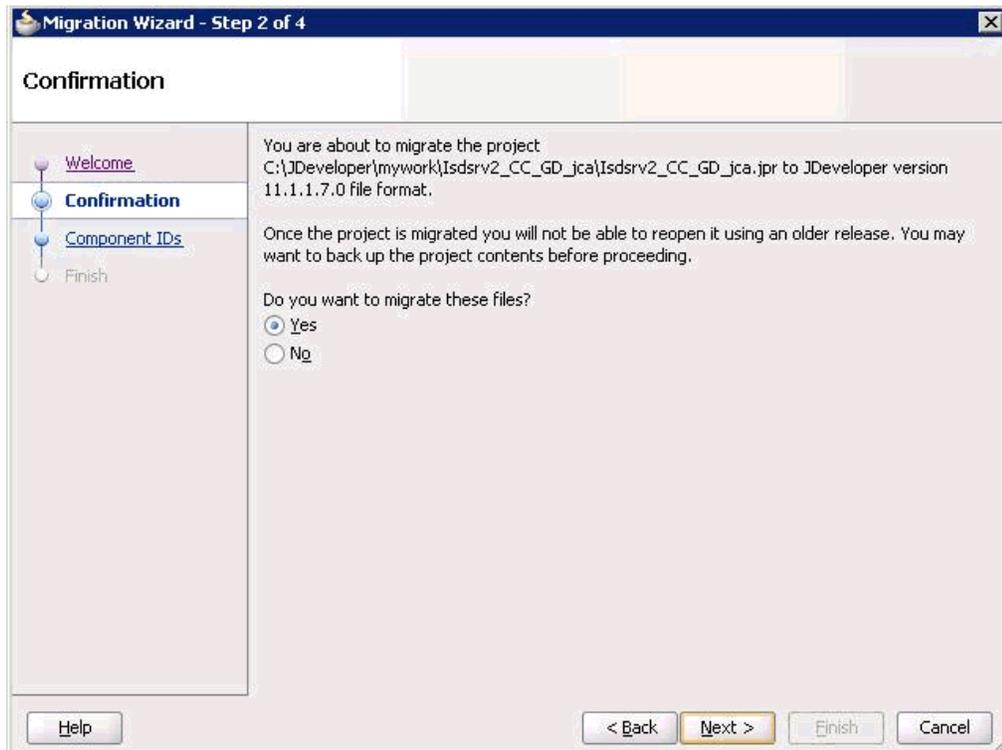
Figure 2–18 Welcome Page of the Migration Wizard



5. Click Next.

The Confirmation page is displayed, as shown in [Figure 2–19](#).

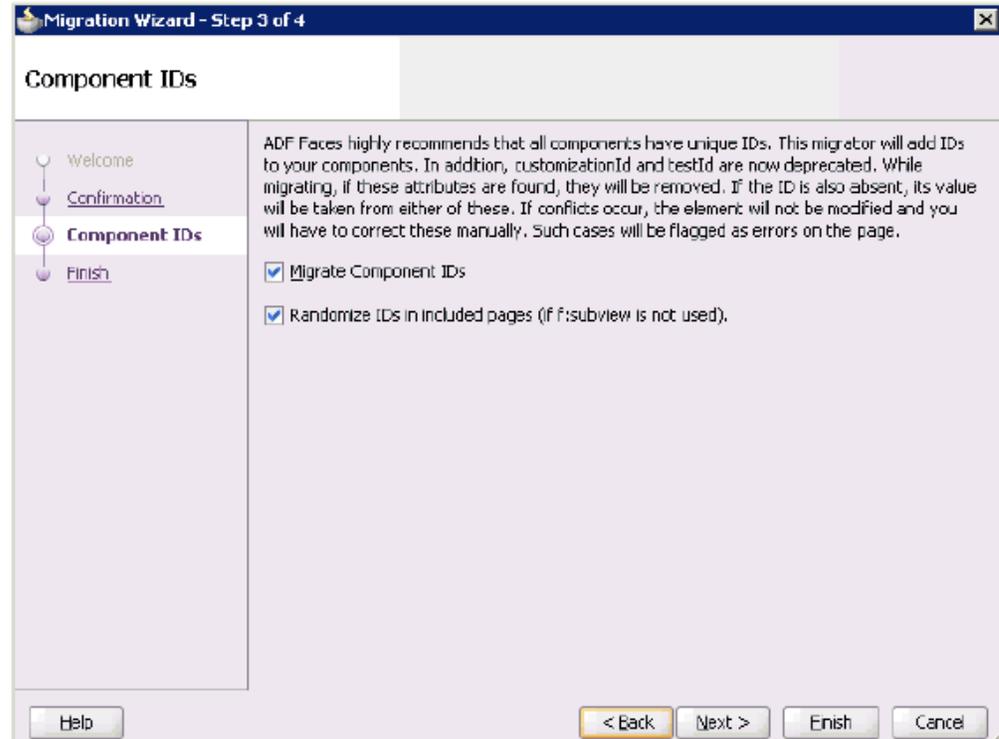
Figure 2–19 Confirmation Page



6. Verify that **Yes** is selected (default) and click **Next**.

The Component IDs page is displayed, as shown in [Figure 2–20](#).

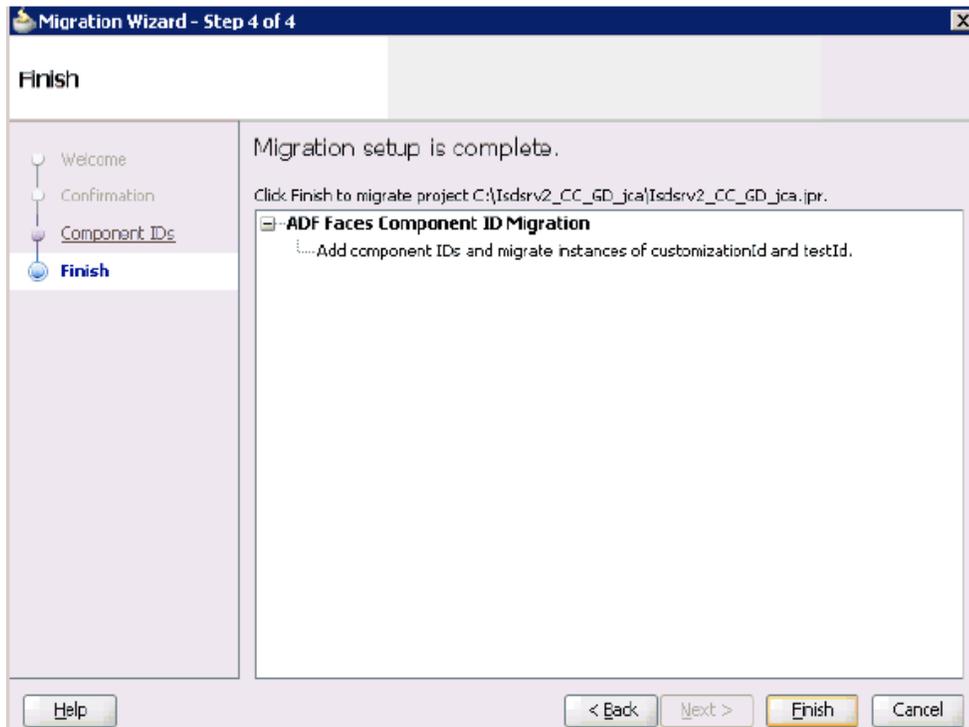
Figure 2–20 *Component IDs Page*



7. Accept the default values and click **Next**.

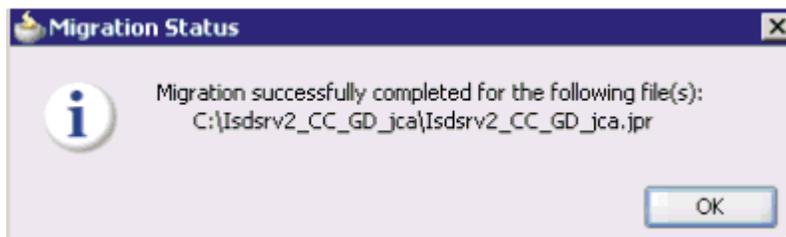
The Finish page is displayed, as shown in [Figure 2–21](#).

Figure 2–21 Finish Page of the Migration Wizard

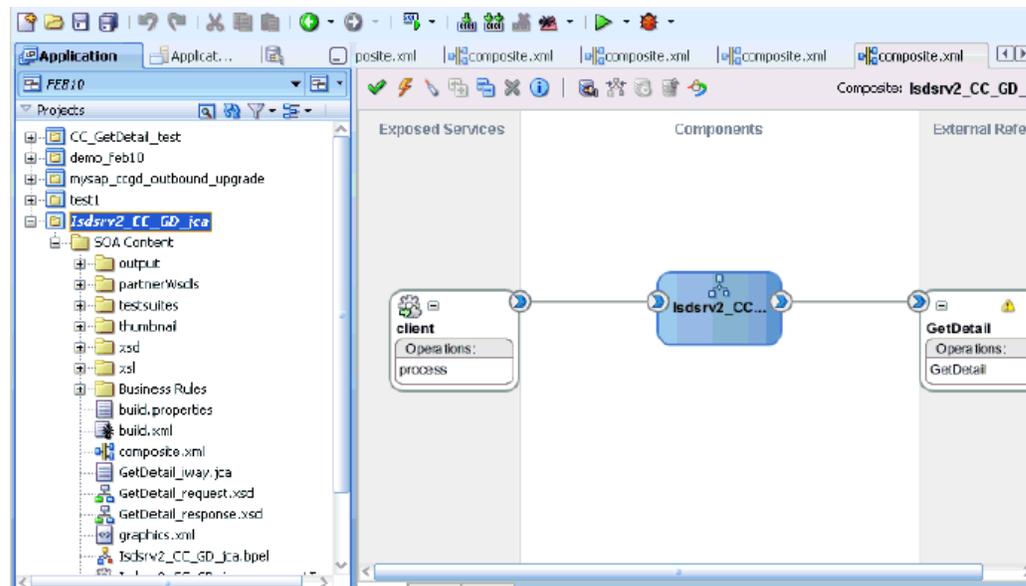


8. Click **Finish**. The following message is displayed, as shown in [Figure 2–22](#).

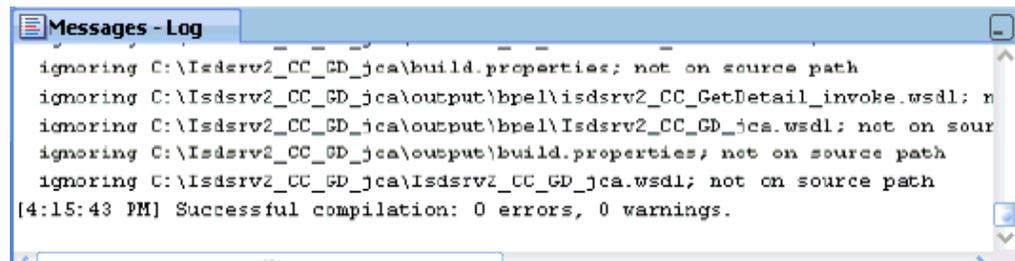
Figure 2–22 Successful Migration Status Message



9. Click **OK**.
As shown in [Figure 2–23](#), the 10.1.3.x project is now available in your 11g environment.

Figure 2–23 A Migrated 10.1.3.x Project in the 11g Environment

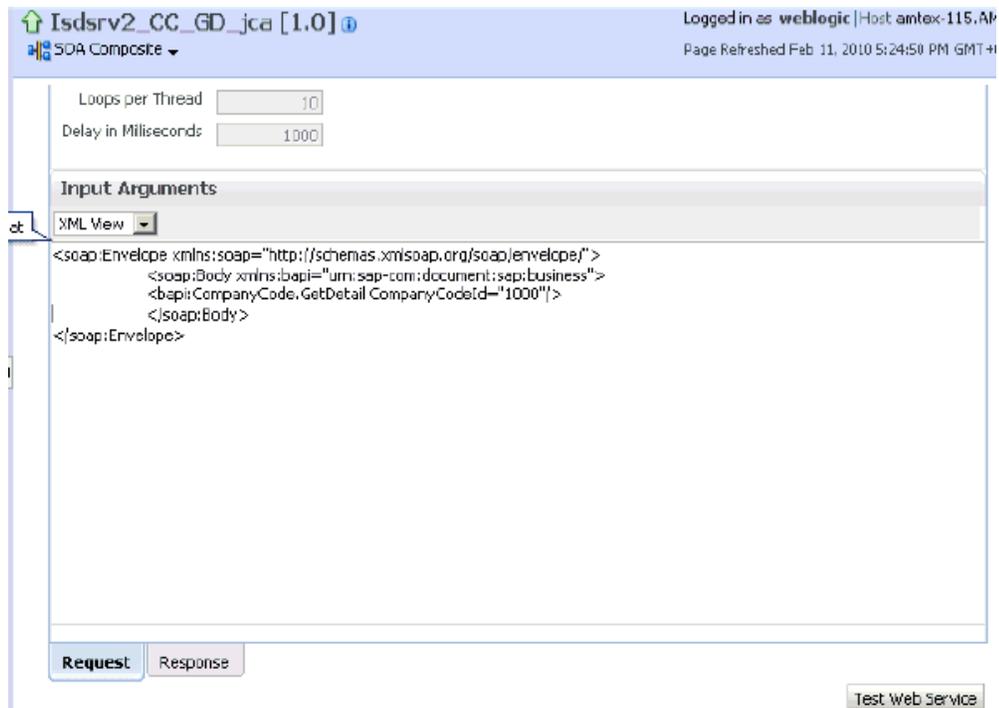
10. Expand the migrated project in the left page (for example, `Isdsrv2_CC_GD_jca`) and double-click the `composite.xml` file to verify that the project opens without any errors.
11. Click **Save**.
12. Compile and then deploy the migrated BPEL process project, as shown in [Figure 2–24](#).

Figure 2–24 Messages Log Tab

13. Verify that there are no error or warning messages during compilation and deployment.
14. After the deployment is successful, as shown in [Figure 2–25](#), go to the Oracle Enterprise Manager console and execute the deployed process either in Tree View or XML View and get the successful response.

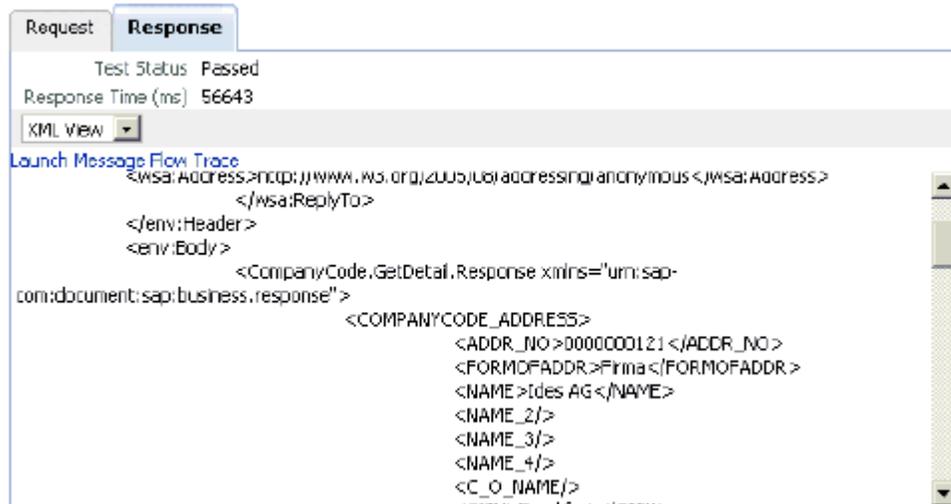
Input XML in XML View

Figure 2–25 Input XML in the Request Tab



Output XML that is received

Figure 2–26 Output XML Received in the Response Tab



2.1.5 Post-Upgrade Requirement

The following changes must be made for every outbound ESB and BPEL process after the corresponding project is migrated to 11g.

1. Expand the migrated project and double-click the **composite.xml** file.
2. Click **source**.

3. Navigate to the reference section, check for the .jca file that has the naming convention as xxxx_iway.jca and replace with xxxx_iway_3P.jca (adding _3P as a suffix).

Original:

```
<reference ui:wSDLLocation="isdsrv15_eaicreate_node_invoke.wsdl" name="create_node">
<interface.wsdl
interface="http://xmlns.oracle.com/pcbpel/iWay/wsdl/Siebel/isdsrv15/create_node#wsdl.interface(create_nodePortType)"
xmlns:ns="http://xmlns.oracle.com/sca/1.0"/>
<binding.jca config="create_node_iway.jca"/>
</reference>
```

Modified:

```
<reference ui:wSDLLocation="isdsrv15_eaicreate_node_invoke.wsdl" name="create_node">
<interface.wsdl
interface="http://xmlns.oracle.com/pcbpel/iWay/wsdl/Siebel/isdsrv15/create_node#wsdl.interface(create_nodePortType)" xmlns:ns="http://xmlns.oracle.com/sca/1.0"/>
<binding.jca config="create_node_iway_3P.jca"/>
</reference>
```

Figure 2–27 Source of the JCA Properties File



4. Open the project folder and rename the same .jca properties file by adding _3P as a suffix in the .jca file (for example, create_node_iway_3P.jca).

2.2 Upgrading a 10.1.3.x Inbound Process to 11g

This section describes how to upgrade a 10.1.3.x inbound process to 11g.

Note: For demonstration purposes, Oracle Application Adapter for SAP R/3 is used as an example.

As a requirement, you must create the same adapter targets and channels in the 11g environment using Application Explorer that you had created in the 10.1.3.x environment. Any change in the adapter target or channel can cause issues for the projects to work in 11g environment.

2.2.1 Overview of 10.1.3.x BPEL JCA Inbound Workflow Process

1. Using Application Explorer, generate a 10.1.3.x JCA inbound WSDL document for Oracle Application Adapter for SAP R/3.
2. Create a 10.1.3.x BPEL JCA inbound process using Oracle JDeveloper 10.1.3.x and deploy it successfully.
3. Trigger the event messages from the ERP system (for example, SAP GUI) and verify that successful instances are received for the deployed process in the BPEL console.
4. Migrate the 10.1.3.x BPEL JCA inbound process to 11g using Oracle JDeveloper 11g.
5. Remove the following line from the JCA properties file in the migrated project and deploy it successfully:

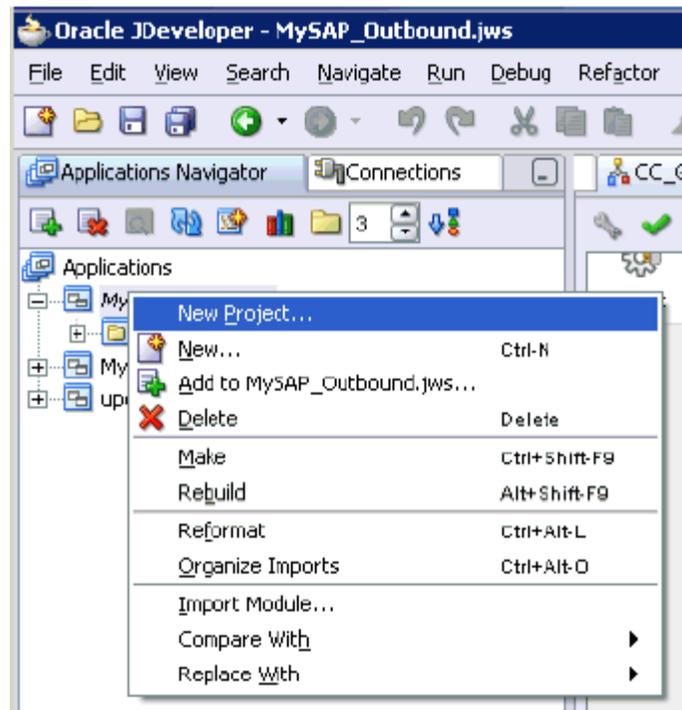
```
<record-converter  
  className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

6. Trigger the event messages from the ERP system (for example, SAP GUI) and ensure that successful instances are received in the Oracle Enterprise Manager console.

2.2.2 Creating an Inbound BPEL Process Project

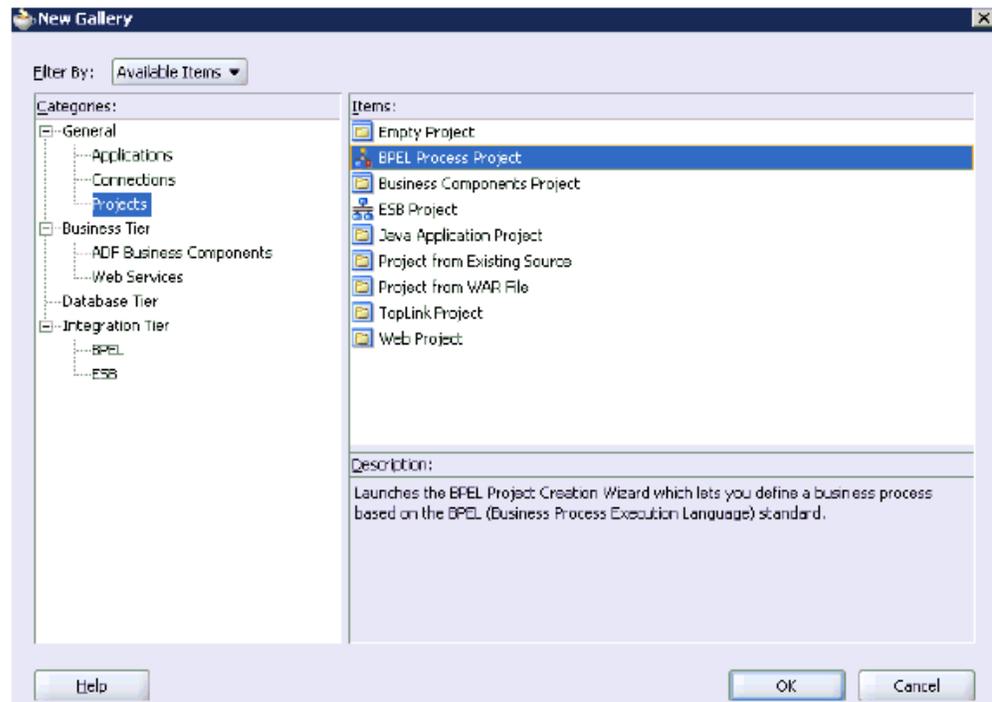
To create an inbound BPEL process project:

1. As shown in [Figure 2–28](#), start Oracle JDeveloper 10.1.3.4.

Figure 2–28 New Project Option

2. Click the **Application Navigator** tab and select **New Project** from the context menu.

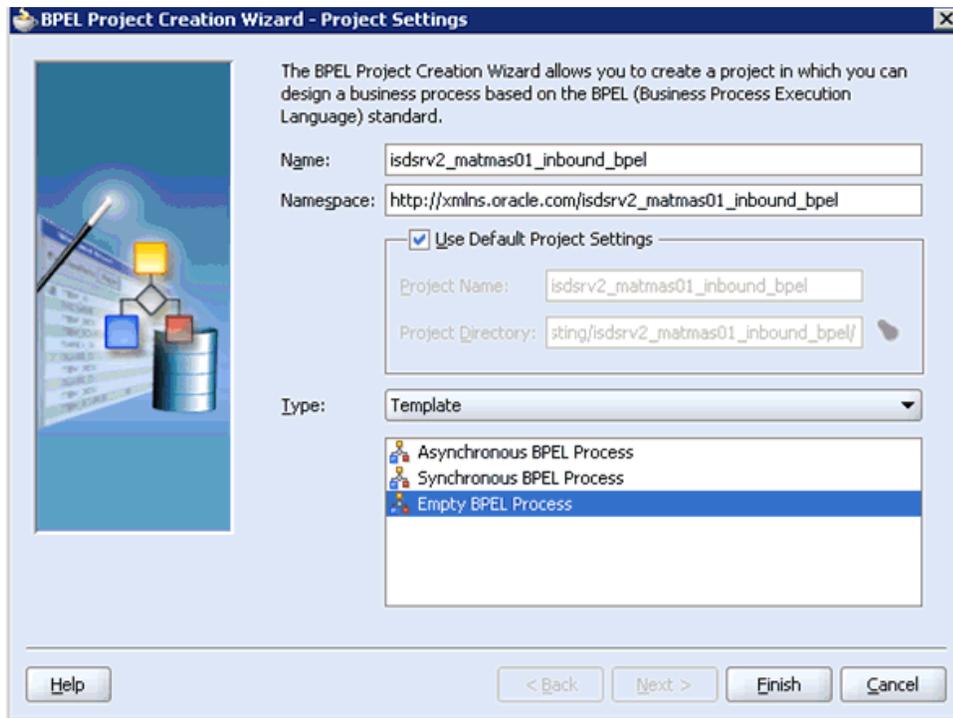
The New Gallery dialog is displayed, as shown in [Figure 2–29](#).

Figure 2–29 New Gallery Dialog

3. Select the **BPEL Process Project** and then click **OK**.

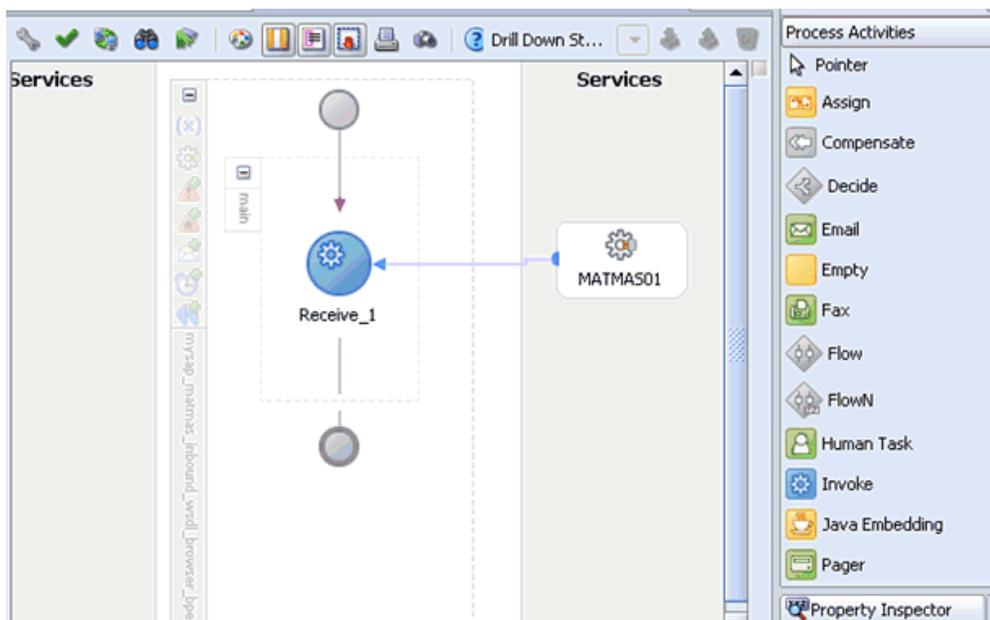
The BPEL Project Creation Wizard - Project Settings page is displayed, as shown in Figure 2-30.

Figure 2-30 BPEL Project Creation Wizard - Project Settings Page



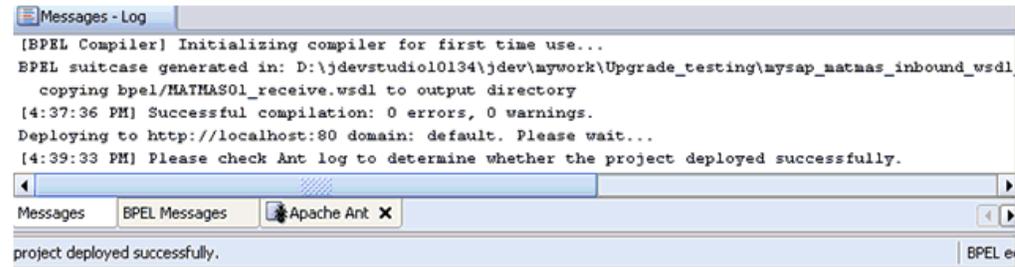
4. Enter a name for the new BPEL process project in the Name field.
5. Select **Empty BPEL Process** from the Template Type list and click **Finish**. Figure 2-31 shows the BPEL inbound process.

Figure 2-31 A Standard BPEL Inbound Process



6. Continue with the creation of a standard BPEL inbound process using Partnerlink and Receive components.
7. Compile the inbound BPEL process project, as shown in [Figure 2–32](#).

Figure 2–32 Messages Tab Showing Deployment Was Successful



8. Verify that there are no error or warning messages during compilation.
9. Deploy the inbound BPEL process project.
10. After the deployment is successful, trigger the event messages from the ERP system (for example, SAP GUI).

The successful instances for the deployed process are received in the BPEL console.

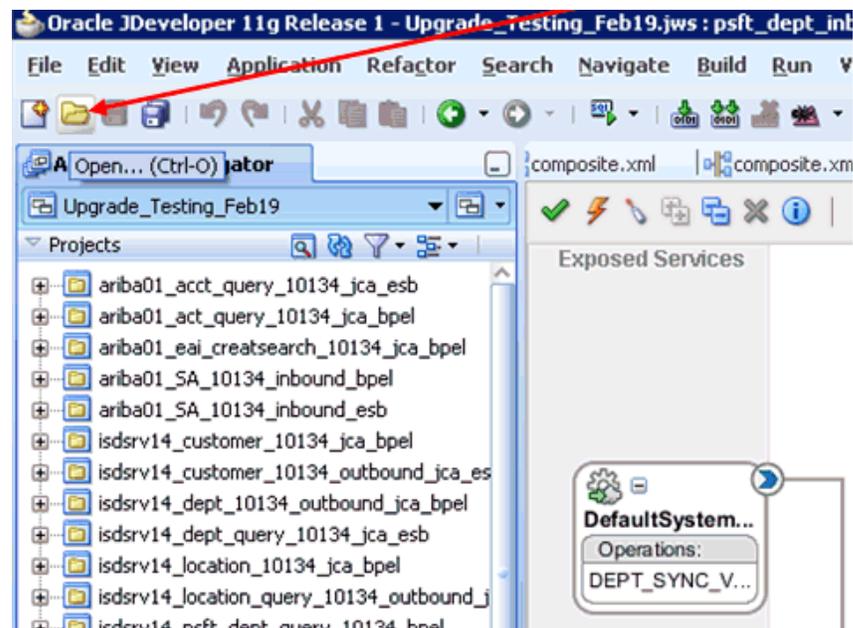
2.2.3 Migrating an Inbound BPEL Process Project

Before continuing, ensure that you copy and paste the 10.1.3.x inbound BPEL Process Project to the 11g system location.

To migrate an inbound BPEL process project:

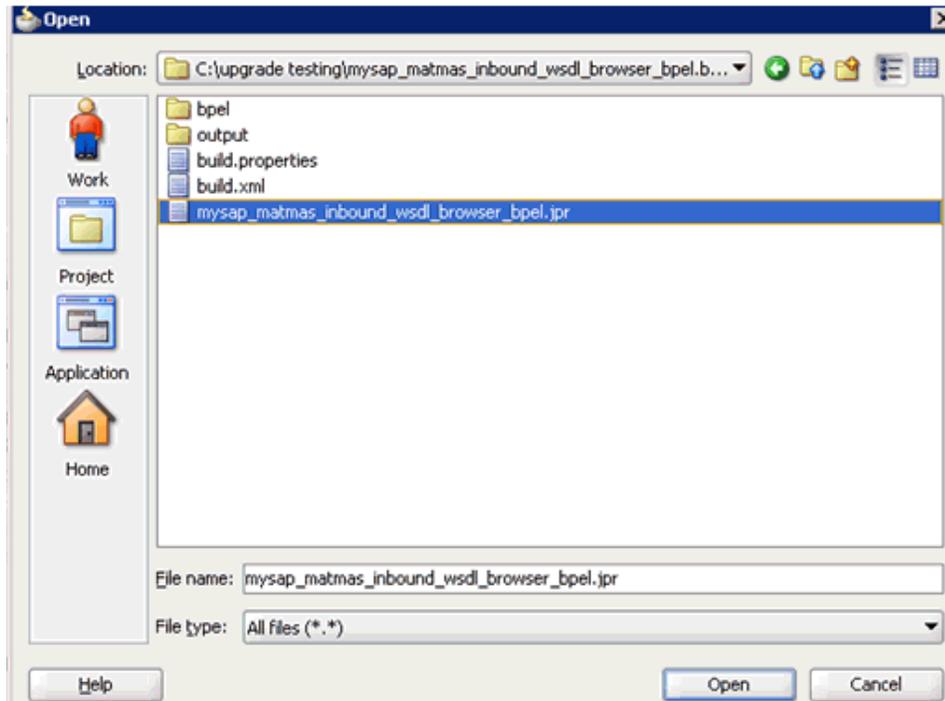
1. As shown in [Figure 2–33](#), start Oracle JDeveloper 11g.

Figure 2–33 Oracle JDeveloper 11g Open Icon

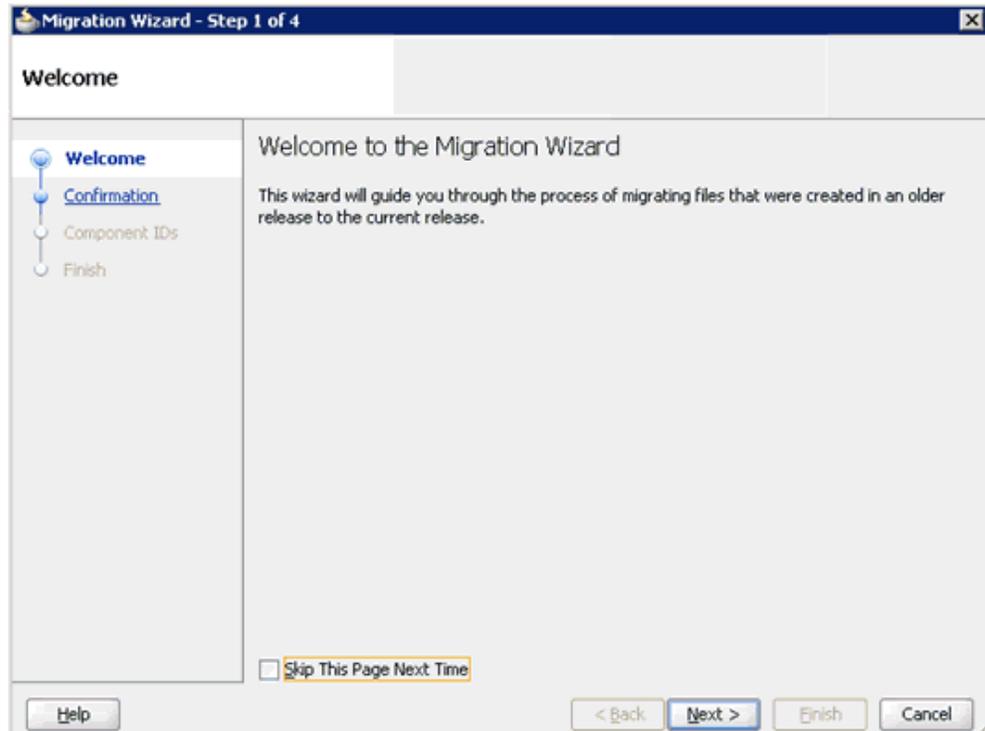


2. Select an available application and click **Open** from the tool bar.
The Open dialog is displayed, as shown in [Figure 2–34](#).

Figure 2–34 *Open Dialog*

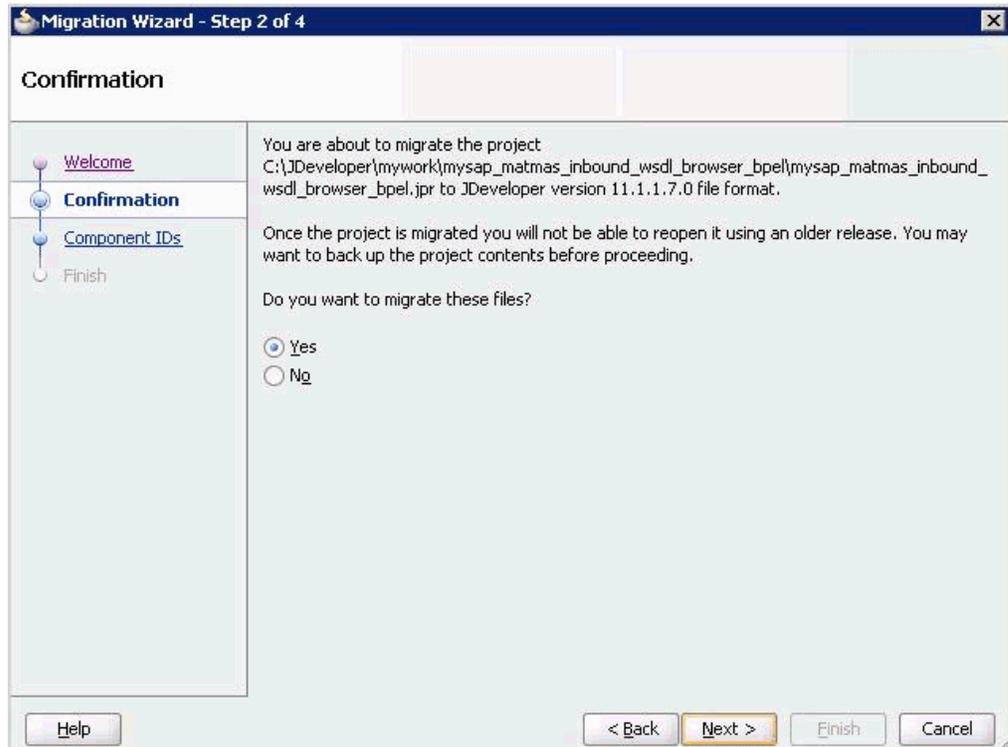


3. Open the 10.1.3.x project and select the **.jpr** extension file (for example, mysap_matmas_inbound_wsdl_browser_bpel.jpr).
4. Click **Open**.
The Migration Wizard is displayed, as shown in [Figure 2–35](#).

Figure 2–35 Migration Wizard

5. Click Next.

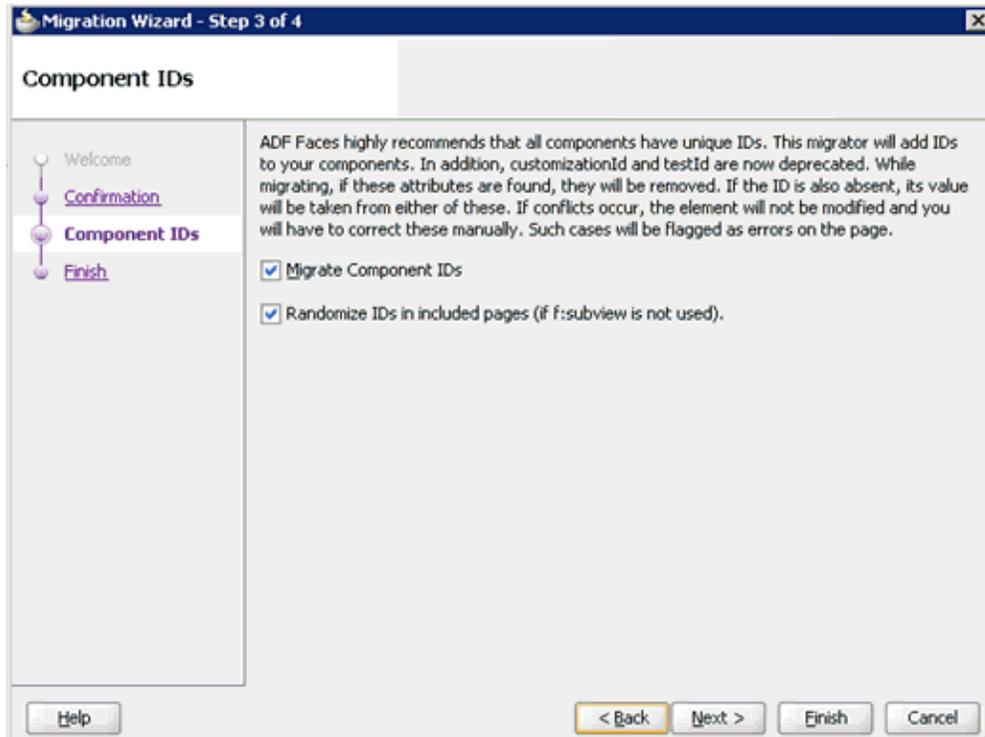
The Confirmation page is displayed, as shown in [Figure 2–36](#).

Figure 2–36 Confirmation Page

- Verify that **Yes** is selected (default) and click **Next**.

The Component IDs page is displayed, as shown in [Figure 2–37](#).

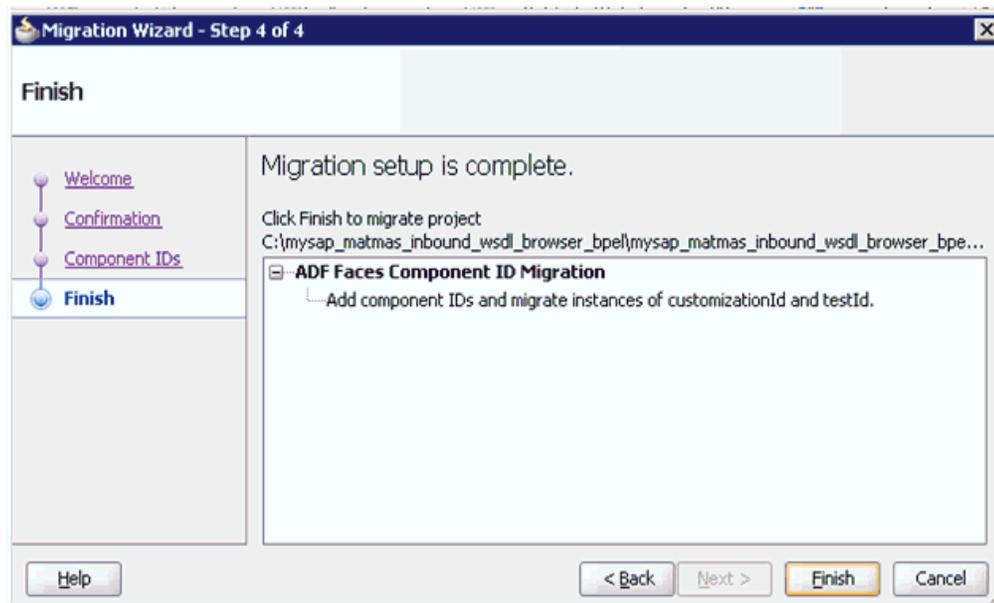
Figure 2–37 Component IDs Page



- Accept the default values and click **Next**.

The Finish page is displayed, as shown in [Figure 2–38](#).

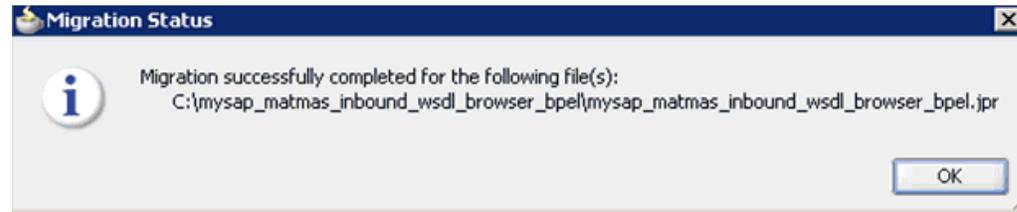
Figure 2–38 Finish Page



8. Click **Finish**.

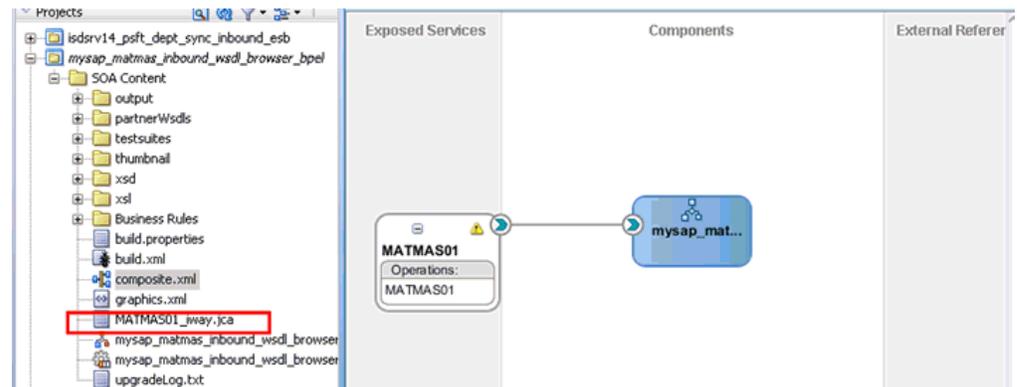
The following message is displayed, as shown in [Figure 2–39](#).

Figure 2–39 Migration Successfully Completed Message

9. Click **OK**.

As shown below in [Figure 2–40](#), the 10.1.3.x project is now available in your 11g environment.

Figure 2–40 Migrated 10.1.3.x Project in the 11g Environment

10. Expand the migrated project in the left page and double-click the **composite.xml** file to verify that the project opens without any errors.

11. Double-click the JCA properties file that is associated with this project (for example, MATMAS01_iway.jca).

12. Remove the XMLRecordConverter line, as shown in [Figure 2–41](#).

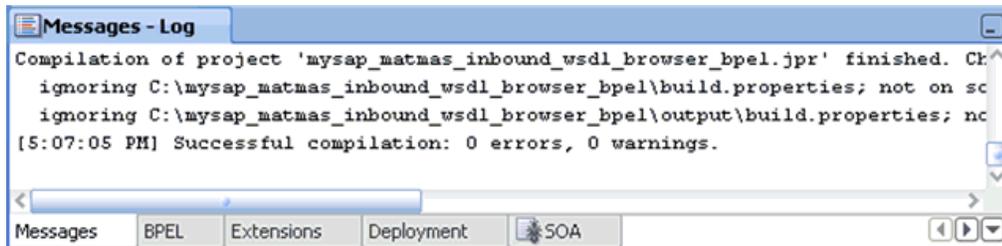
Figure 2–41 JCA Properties File Source

```
<adapter-config name="MATMAS01" adapter="iWay ERP Adapter" wsdlLocation="MATMAS01_receive.wsdl?w
<resource-adapter className="com.iwaysoftware.afjcal5.IWAFOracleResourceAdapter"/>
<record-converter className="com.iwaysoftware.afjcal5.oracle.InboundXMLRecordConverterImpl"/>
<connection-factory location="eis/OracleJCAAdapter/DefaultConnection" adapterRef="" />
<endpoint-activation portType="MATMAS01PortType" operation="MATMAS01">
  <use-record-converter addonNamespace="urn:sap-com:document:sap:idoc"/>
  <activation-spec className="com.iwaysoftware.afjcal5.IWAFOracleActivationSpec">
    <property name="AdapterName" value="MySAP"/>
    <property name="ChannelName" value="mysap_ch"/>
  </activation-spec>
</endpoint-activation>
</adapter-config>
```

13. Click **Save**.

14. Compile and then deploy the migrated BPEL process project, as shown in Figure 2–42.

Figure 2–42 Messages Tab - BPEL Process Object Compiled and Deployed Successfully



15. Verify that there are no error or warning messages during compilation and deployment.
16. After the deployment is successful, trigger the event messages from the ERP system (for example, SAP GUI).

The successful instances for the deployed process are received in the Oracle Enterprise Manager console.

2.2.4 Post-Upgrade Requirement

The following changes must be made for every inbound ESB and BPEL process after the corresponding project is migrated to 11g.

1. Expand the migrated project and double-click the **composite.xml** file.
2. Click **source**.
3. Navigate to the **service** section, check for the **.jca** file that has the naming convention as **xxxx_iway.jca** and replace with **xxxx_iway_3P.jca** (adding **_3P** as a suffix).

Original:

```
<service ui:wSDLLocation="isdsrv22_sampleAcct_receive.wsd1" name="samp_node">
<interface.wsd1
interface="http://xmlns.oracle.com/pcbpel/iWay/wsd1/Siebel/isdsrv22/samp_
node#wsd1.interface(samp_nodePortType) "
xmlns:ns="http://xmlns.oracle.com/sca/1.0"/>
<binding.jca config="samp_node_iway.jca"/>
</service>
```

Modified:

```
<service ui:wSDLLocation="isdsrv22_sampleAcct_receive.wsd1" name="samp_node">
<interface.wsd1
interface="http://xmlns.oracle.com/pcbpel/iWay/wsd1/Siebel/isdsrv22/samp_
node#wsd1.interface(samp_nodePortType) "
xmlns:ns="http://xmlns.oracle.com/sca/1.0"/>
<binding.jca config="samp_node_iway_3P.jca"/>
</service>
```

Figure 2–43 JCA Properties File Source

```

<?xml version="1.0" encoding="UTF-8"?>
<composite name="isdsrv22_SampleAccount_inbound_esb" applicationName="Upgrade_Testing_Feb19" mode="active" revision="
<import location="isdsrv22_sampleaccount_esb_receive_new.wsdl" namespace="http://xmlns.oracle.com/pcbpel/iWay/wsd
<import location="write_inbound_SA.wsdl" namespace="http://xmlns.oracle.com/pcbpel/adapter/file/write_inbound_SA/
<import location="isdsrv22_sampleaccount_esb_receive_new.wsdl" namespace="http://xmlns.oracle.com/pcbpel/iWay/wsd
<service ui:wSDLLocation="isdsrv22_sampleaccount_esb_receive_new.wsdl" name="DefaultSystem.isdsrv22_inbound_esb">
  <interface wsd:interface="http://xmlns.oracle.com/pcbpel/iWay/wsd1/Siebel/isdsrv22/samp_node#wsdl.interface{
    <binding.jca config="samp_node_iway_3P.jca"/>
  </service>
</service ui:wSDLLocation="isdsrv22_sampleaccount_esb_receive_new.wsdl" name="DefaultSystem.isdsrv22_inbound_esb_R

```

4. Open the project folder and rename the same .jca properties file by adding _3P as a suffix in the .jca file (for example, samp_node_iway_3P.jca).

2.3 Migrating Oracle Service Bus 10.1.3.x/PS2/PS3/PS4/PS5 Outbound and Inbound Processes to 11g PS6

This section describes how to migrate outbound and inbound J2CA processes and outbound BSE processes from Oracle Service Bus 10.1.3.x/PS2/PS3/PS4/PS5 to 11g PS6. It contains the following topics:

- Section 2.3.1, "Exporting the Configured Processes From Oracle Service Bus 10.1.3.x/PS2/PS3/PS4/PS5"
- Section 2.3.2, "Importing the Exported Processes to Oracle Service Bus 11g PS6"
- Section 2.3.3, "Additional Modifications for Imported Processes in Oracle Service Bus 11g PS6"

2.3.1 Exporting the Configured Processes From Oracle Service Bus 10.1.3.x/PS2/PS3/PS4/PS5

This section describes how to export the configured processes from Oracle Service Bus 10.1.3.x/PS2/PS3/PS4/PS5.

1. Start the Oracle WebLogic Server for the Oracle WebLogic Server domain that you have configured.
2. Open the Oracle Service Bus console in a Web browser by entering the following URL:

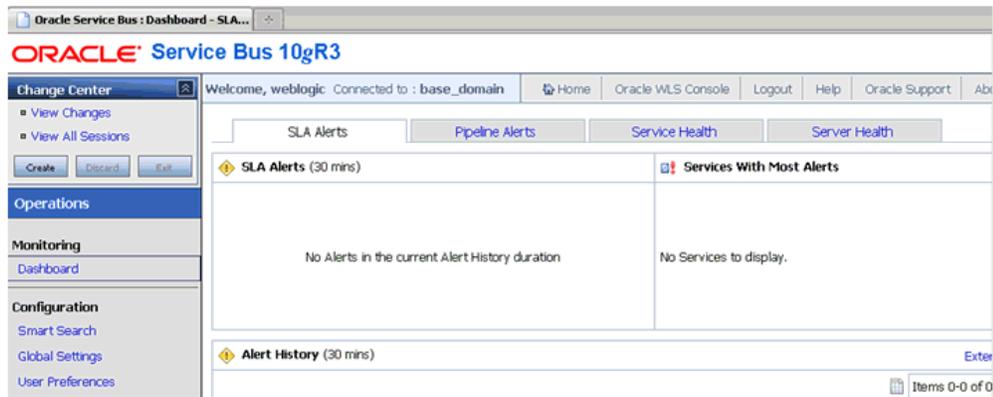
```
http://host name:port/sbconsole
```

where *host name* is the name of the system where Oracle WebLogic Server is running (Oracle Service Bus 10.1.3.x/PS2/PS3/PS4/PS5) and *port* is the port for the domain you are using. The port for the default domain is 7001.

3. Log in to the Oracle Service Bus console using a valid user name and password.

The Oracle Service Bus console home page is displayed, as shown in [Figure 2–44](#).

Figure 2–44 Oracle Service Bus Console Home Page



4. As shown in [Figure 2–45](#), click **Create** in the Change Center area to start a new Oracle Service Bus session.

Figure 2–45 Create Button in Change Center Area



5. Click **System Administration** in the left pane, as shown in [Figure 2–46](#).

Figure 2–46 System Administration Option



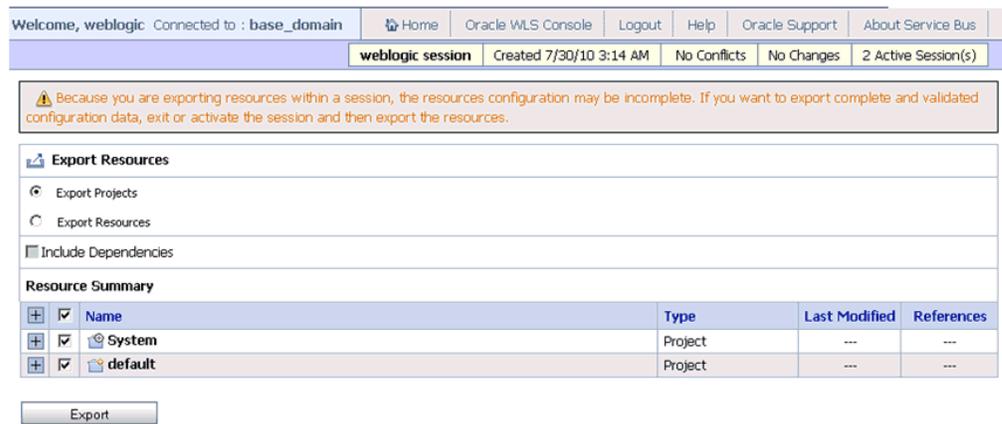
6. Click **Export Resources** in the Import/Export area, as shown in [Figure 2–47](#).

Figure 2–47 Export Resources Option



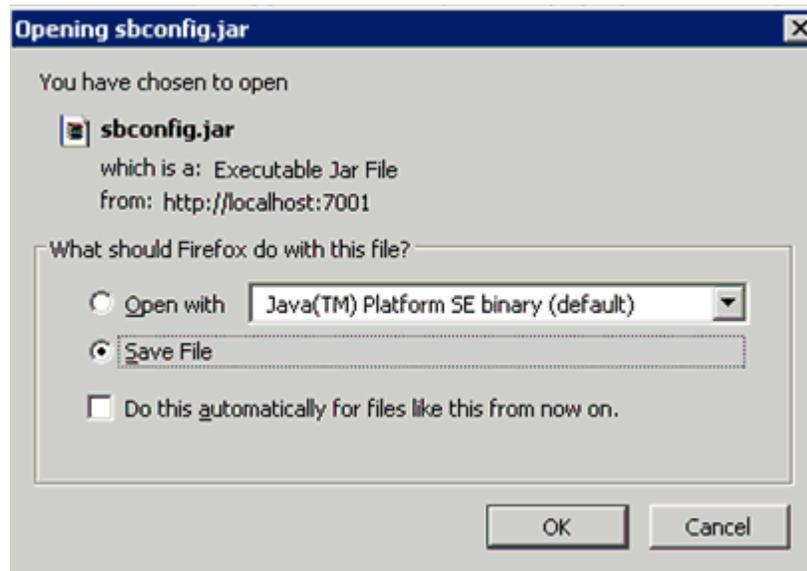
7. Ensure that all the available options under Resource Summary are selected (System and default), and then click **Export**, as shown in Figure 2–48.

Figure 2–48 Resource Summary Area and Export Button



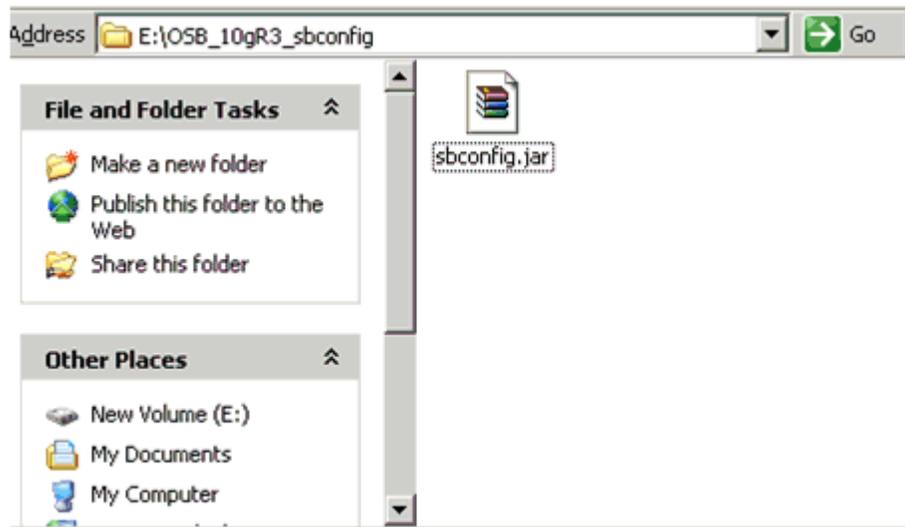
8. Save the *sbconfig.jar* file, as shown in Figure 2–49.

Figure 2–49 Opening sbconfig.jar Dialog



9. As shown in [Figure 2–50](#), verify that the *sbconfig.jar* is saved successfully.

Figure 2–50 The *sbconfig.jar* File Saved Successfully in a File System Directory Location



Note: The *sbconfig.jar* file that is exported from Oracle Service Bus 10.1.3.x/PS2/PS3/PS4/PS5 must be copied to the system where Oracle Service Bus 11g PS6 is installed.

2.3.2 Importing the Exported Processes to Oracle Service Bus 11g PS6

This section describes how to import the exported processes to Oracle Service Bus 11g PS6.

Prerequisites

- All the adapter targets and channels that are configured using Application Explorer in an Oracle Service Bus 11g PS6 environment must match those in an Oracle Service Bus 10.1.3.x/PS2/PS3/PS4/PS5 environment.
- All the input and output locations configured for the processes in an Oracle Service Bus 11g PS6 environment must match those in an Oracle Service Bus 10.1.3.x/PS2/PS3/PS4/PS5 environment.

To import the exported processes to Oracle Service Bus 11g PS6:

1. Start the Oracle WebLogic Server for the Oracle WebLogic Server domain that you have configured.
2. Open the Oracle Service Bus console in a Web browser by entering the following URL:

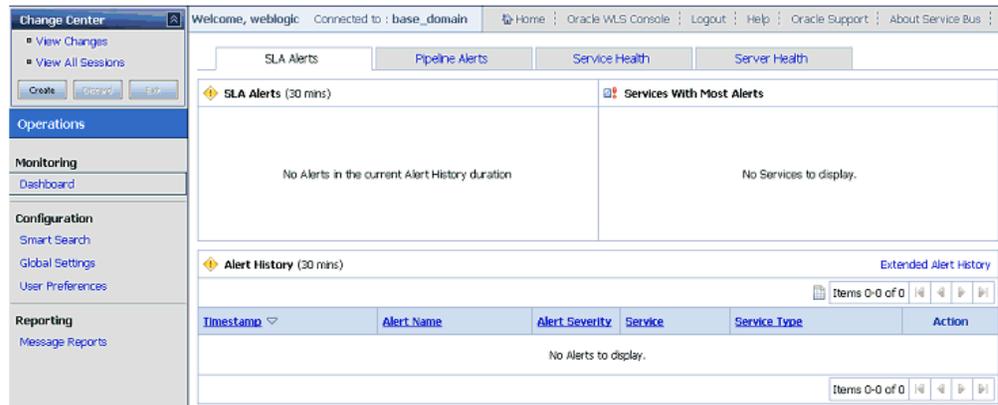
```
http://host name:port/sbconsole
```

where *host name* is the name of the system where Oracle WebLogic Server is running (Oracle Service Bus 11g PS6) and *port* is the port for the domain you are using. The port for the default domain is 7001.

3. Log in to the Oracle Service Bus console using a valid user name and password.

The Oracle Service Bus console home page is displayed, as shown in [Figure 2–51](#).

Figure 2–51 Oracle Service Bus Console Home Page



4. Click **Create** in the Change Center area to start a new Oracle Service Bus session, as shown below in [Figure 2–52](#).

Figure 2–52 Create Button in the Change Center Area



5. Click **System Administration** in the left pane, as shown in [Figure 2–53](#).

Figure 2–53 System Administration Option



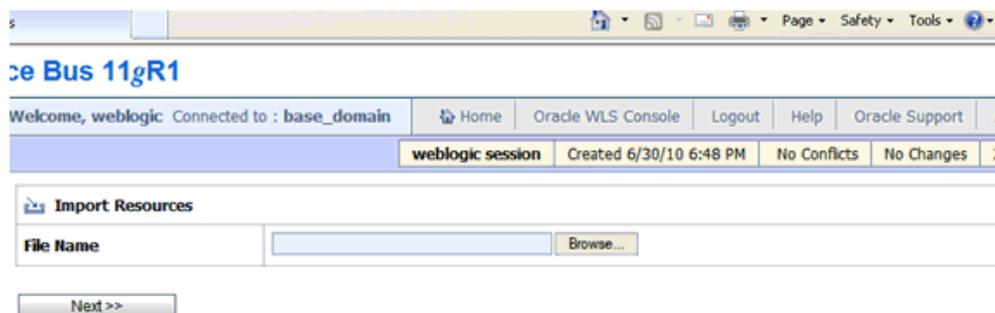
6. As shown in [Figure 2–54](#), click **Import Resources** in the Import/Export area.

Figure 2–54 Import Resources Option From Import/Export Area



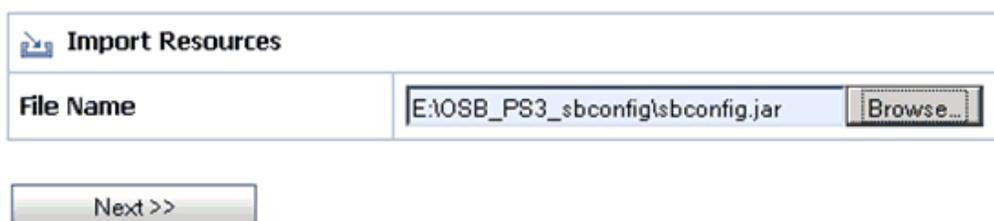
The Import Resources page is displayed, as shown in [Figure 2–55](#).

Figure 2–55 Import Resources Page



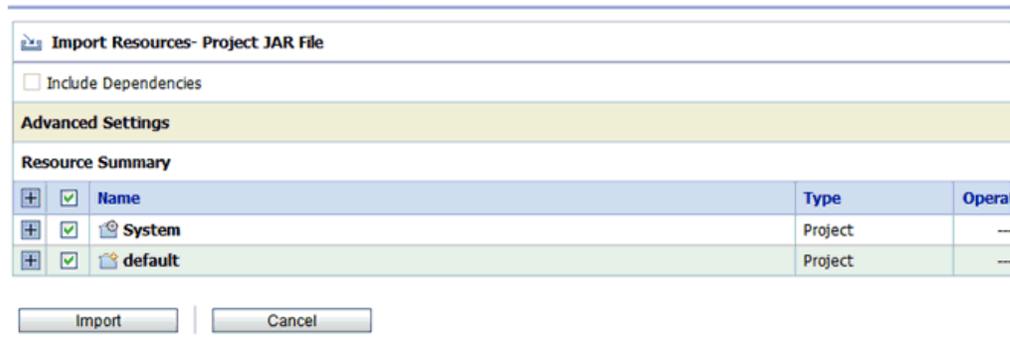
7. As shown in [Figure 2–56](#), click **Browse** and select the `sbconfig.jar` file, which was exported from Oracle Service Bus 10.1.3.x/PS2/PS3/PS4/PS5 and copied to the local system.

Figure 2–56 Browse Button on the Import Resources Page



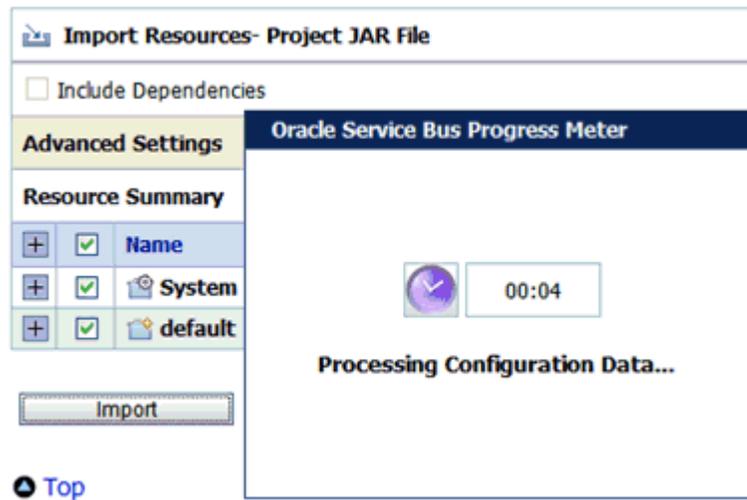
8. Click **Next**.
9. Select all the options listed in the Resource Summary area and click **Import**, as shown in [Figure 2–57](#).

Figure 2–57 Import Button in the Resource Summary Area



As shown below in [Figure 2–58](#), the processes are imported.

Figure 2–58 Processing Status Indicator



10. Verify that the success message is displayed and click **Activate**, as shown in [Figure 2–59](#).

Figure 2–59 Import Process Success Message and Activate Button



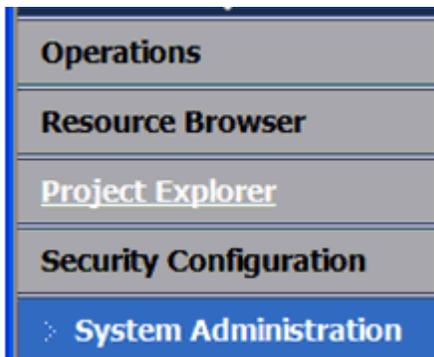
The Activate Session page is displayed, as shown in [Figure 2–60](#).

Figure 2–60 Activate Session Page

Activate Session	
Session Name	weblogic
User	weblogic
Description	<div style="border: 1px solid gray; height: 100px;"></div>
<input type="button" value="Submit"/>	

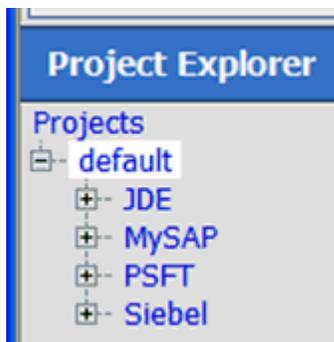
11. Click **Submit**.
12. Click **Project Explorer** after the session is successfully activated, as shown in [Figure 2–61](#).

Figure 2–61 Project Explorer Option



13. As shown below in [Figure 2–62](#), ensure that all the project folders are created.

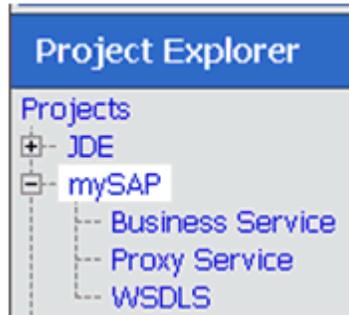
Figure 2–62 Imported Project Folders



The project folder structure must match the one from Oracle Service Bus 10.1.3.x/PS2/PS3/PS4/PS5, since you imported these settings from that environment.

14. Expand an adapter folder, for example, mySAP, as shown in [Figure 2–63](#).

Figure 2–63 Expanded mySAP Project Folder



15. As shown in Figure 2–64, select the **Business Service** folder.

Figure 2–64 Imported Business Services

The screenshot shows the 'Resources' panel in a software interface. At the top, there is a 'Create Resource' dropdown menu set to 'Select Resource Type'. Below this is a table listing imported resources. The table has columns for 'Name', 'Resource Type', and 'Actions'. There are 8 items listed, including Business Services and JCA Bindings.

Name	Resource Type	Actions
idsrv2_CC_GD_ibse_BS	Business Service	[Settings] [Info]
idsrv2_CC_GD_jca_BS	Business Service	[Settings] [Info]
idsrv2_CC_GD_jca_BS	JCA Binding	[Info]
idsrv2_Idoc_MATMAS_jca_BS	Business Service	[Settings] [Info]
idsrv2_Idoc_MATMAS_jca_BS	JCA Binding	[Info]
idsrv2_RFC_Holiday_Check_jca_BS	Business Service	[Settings] [Info]
idsrv2_RFC_Holiday_Check_jca_BS	JCA Binding	[Info]
MySAP_File_out	Business Service	[Settings]

Ensure that all the Business Services that were created in the Oracle Service Bus 10.1.3.x/PS2/PS3/PS4/PS5 environment are included.

16. As shown in Figure 2–65, select the **Proxy Service** folder.

Figure 2–65 Imported Proxy Services

Name	Resource Type	Actions
isdsrv2_BA_GD_IB_PS	JCA Binding	
isdsrv2_BA_GD_IB_PS	Proxy Service	
isdsrv2_CC_GD_ibse_PS	Proxy Service	
isdsrv2_CC_GD_jca_PS	Proxy Service	
isdsrv2_Idoc_MATMAS_jca_PS	Proxy Service	
isdsrv2_MATMAS01_IB_PS	JCA Binding	
isdsrv2_MATMAS01_IB_PS	Proxy Service	
isdsrv2_RFC_Holiday_Check_jca_PS	Proxy Service	

Ensure that all the Proxy Services that were created in the Oracle Service Bus 10.1.3.x/PS2/PS3/PS4/PS5 environment are included.

17. Select the **WSDLs** folder, as shown in [Figure 2–66](#).

Figure 2–66 Imported WSDL Files

Name	Resource Type	Actions
isdsrv2_BAPI_BUSINESSAREA_GETDETAIL_rece...	WSDL	
isdsrv2_BAPI_BUSINESSAREA_GETDETAIL_rece...	WSDL	
isdsrv2_CC_getdetail_ibse	WSDL	
isdsrv2_CC_GetDetail_invoke	WSDL	
isdsrv2_CC_GetDetail_invoke_abstract	WSDL	
isdsrv2_MATMAS01_invoke	WSDL	
isdsrv2_MATMAS01_invoke_abstract	WSDL	
isdsrv2_MATMAS01_receive	WSDL	
isdsrv2_MATMAS01_receive_abstract	WSDL	
isdsrv2_RFC_HOLIDAY_CHECK_AND_GET_INFO_L...	WSDL	
isdsrv2_RFC_HOLIDAY_CHECK_AND_GET_INFO_L...	WSDL	

Ensure that all the WSDL files that were created in the Oracle Service Bus 10.1.3.x/PS2/PS3/PS4/PS5 environment are included.

Note: For applications using Oracle Application Adapter for Siebel, the namespace declaration for the XML payload must use the version that is generated in Oracle Service Bus (OSB) 10g Release 3. The Oracle Application Adapter for Siebel namespace in OSB 10g Release 3 environments is different from Oracle Service Bus 11g PS6 environments.

For example:

OSB 10g Release 3:

```
<sbl:Siebel location="S/BO/Account/Account/query"
xmlns:sbl="urn:iwaysoftware:adapter:siebel:oct2004:request">
```

OSB 11g:

```
<sbl:Siebel location="S/BO/Account/Account/query"
xmlns:sbl="urn:iwaysoftware:adapter:siebel:request:S/BO/Account/Account/query">
```

The change that is described in this note is only required if you are migrating an OSB 10g Release 3 process to 11g PS6. This change is not required if you are migrating an OSB 11g PS2/PS3/PS4/PS5 process to 11g PS6.

2.3.3 Additional Modifications for Imported Processes in Oracle Service Bus 11g PS6

This section describes additional modifications that are required for the imported processes in Oracle Service Bus 11g PS6.

Note: Imported J2CA outbound processes in Oracle Service Bus 11g PS6 work properly and do not require any additional changes. Only BSE outbound processes and J2CA inbound processes require additional changes.

This section contains the following topics:

- [Section 2.3.3.1, "Imported J2CA Inbound Processes"](#)
- [Section 2.3.3.2, "Imported BSE Outbound Processes"](#)

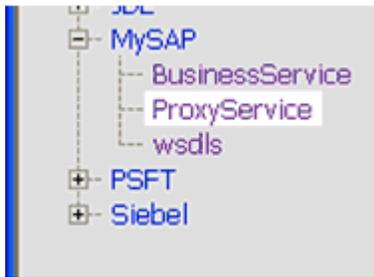
2.3.3.1 Imported J2CA Inbound Processes

This section describes additional modifications that are required for imported J2CA inbound processes in Oracle Service Bus 11g PS6.

Note: The changes that are described in this section are only required if you are migrating an OSB 10g Release 3 process to 11g PS6. If you are migrating an OSB 11g PS2/PS3/PS4/PS5 process to 11g PS6, then you can skip this section (1.3.3.1, "Imported J2CA Inbound Processes").

1. Select the folder that contains the Proxy Services for your adapter in the Project Explorer, as shown in [Figure 2–67](#).

Figure 2–67 ProxyService Folder



2. Select the appropriate JCA binding file for the inbound process that must be changed, as shown in [Figure 2–68](#).

Figure 2–68 JCA Binding File

A screenshot of a 'Resources' table. The table has columns for 'Name', 'Resource Type', and 'Actions'. The 'Name' column is expanded to show a search filter 'isdsrv2_MATMAS01_IB_PS'. The table contains several rows of resources, including JCA Binding and Proxy Service files. The row for 'isdsrv2_MATMAS01_IB_PS' is highlighted in yellow.

Name	Resource Type	Actions
isdsrv2_BA_GD_IB_PS	JCA Binding	[Icons]
isdsrv2_BA_GD_IB_PS	Proxy Service	[Icons]
isdsrv2_CC_GD_ibse_PS	Proxy Service	[Icons]
isdsrv2_CC_GD_jca_PS	Proxy Service	[Icons]
isdsrv2_Idoc_MATMAS_jca_PS	Proxy Service	[Icons]
isdsrv2_MATMAS01_IB_PS	JCA Binding	[Icons]
isdsrv2_MATMAS01_IB_PS	Proxy Service	[Icons]
isdsrv2_RFC_Holiday_Check_jca_PS	Proxy Service	[Icons]

The JCA Binding Text View is displayed, as shown in [Figure 2–69](#).

Figure 2–69 JCA Binding Text View

3. Click **Edit**.
4. Remove the `<record-converter>` element, as shown in [Figure 2–70](#).

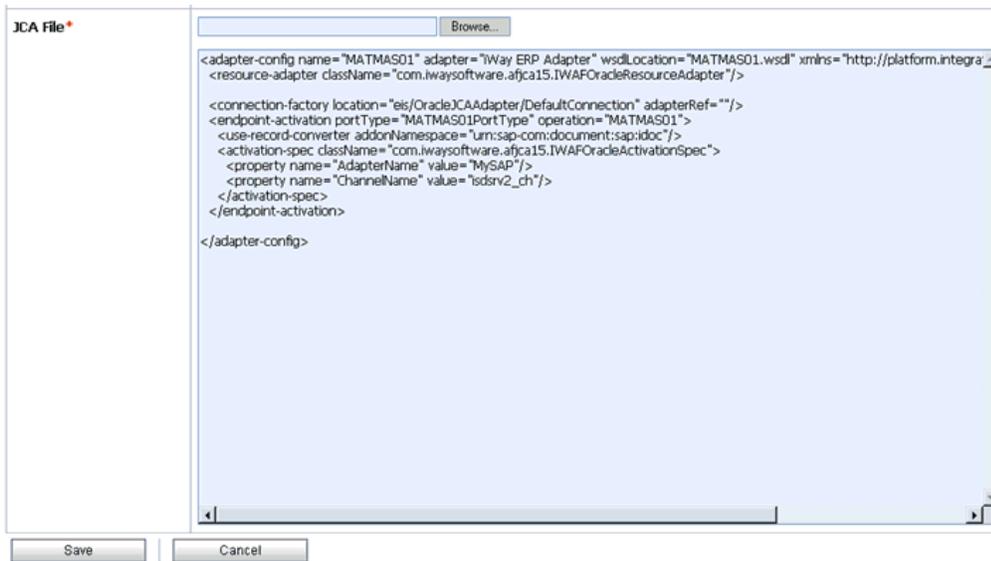
Figure 2–70 The `<record-converter>` Element

```
<adapter-config name="MATMAS01" adapter="iWay ERP Adapter" wsdlLocation="MATMAS01.wsdl" xmlns="http://platform.integrations.com:1.0">
  <resource-adapter className="com.iwaysoftware.afjca15.IWAFOracleResourceAdapter"/>
  <record-converter className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
  <connection-factory location="eis/OracleJCAAdapter/DefaultConnection" adapterRef=""/>
  <endpoint-activation portType="MATMAS01PortType" operation="MATMAS01">
    <use-record-converter addonNamespace="urn:sap-com:document:sap:idoc"/>
    <activation-spec className="com.iwaysoftware.afjca15.IWAFOracleActivationSpec">
      <property name="AdapterName" value="MySAP"/>
      <property name="ChannelName" value="sdsrv2_ch"/>
    </activation-spec>
  </endpoint-activation>
</adapter-config>
```

5. As shown in [Figure 2–71](#), click **Save**.

General Upgrade Guidelines 2-39

Figure 2–71 Save Button



2.3.3.2 Imported BSE Outbound Processes

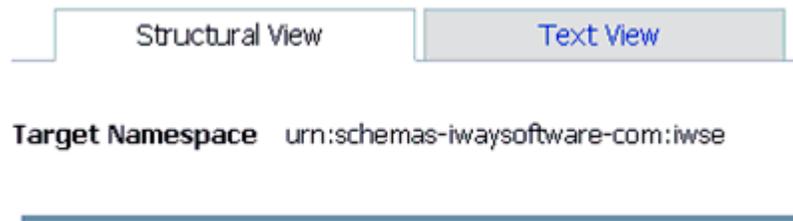
This section describes additional modifications that are required for imported BSE outbound processes in Oracle Service Bus 11g PS6.

1. Select the folder that contains the WSDL files for your adapter in the Project Explorer.
2. Select the specific WSDL file for the outbound process that must be changed, as shown in [Figure 2–72](#).

Figure 2–72 WSDL File Selected for the Outbound Process



3. Click the Text View tab, as shown in [Figure 2–73](#).

Figure 2–73 Text View Tab**4. Click Edit.**

The Edit a WSDL Resource page is displayed, as shown in [Figure 2–74](#).

Figure 2–74 The Edit a WSDL Resource Page

Resource Name*	default/MySAP/wsdl/isdsrv2_CC_getdetail_ibse
Resource Description	<input type="text"/>
WSDL*	<input type="text"/> <input type="button" value="Browse..."/> <pre><?xml encoding="UTF-8" xmlns:tns="urn:schemas-iwaysoftware-com:iwse" xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:mirr soap"><soap:address location="http://192.168.128.122:7001/ibse/IBSEServlet/XDSOAPRouter"/></port></service></definitions></pre>

5. Change the <soap:address location> element to point to the system where OSB 11g PS6 is running. For example:

```
<soap:address
location="http://OSB10gR3Machine:7001/ibse/IBSEServlet/XDSOAPRouter" />
```

must be changed to:

```
<soap:address
location="http://OSB11gMachine:7001/ibse/IBSEServlet/XDSOAPRouter" />
```

6. Click Save.**7. Select a Business Service that is created for a BSE outbound process in the Project Explorer.****8. Click Edit in the Configuration details tab.****9. In the Transport Configuration section, update the value for the Endpoint URI property to point to the system where OSB 11g PS6 is running, as shown in [Figure 2–75](#).**

Figure 2–75 Transport Configuration Section

Transport Configuration	
Protocol	http
Load Balancing Algorithm	round-robin
Endpoint URI	http://172.19.21.203:7001/ibse/IBSEServlet/XDSOAPRouter
Retry Count	0
Retry Iteration Interval	30
Retry Application Errors	Yes

For example:

```
<soap:address
location="http://OSB10gR3Machine:7001/ibse/IBSEServlet/XDSOAPRouter"/>
```

must be changed to:

```
<soap:address
location="http://OSB11gMachine:7001/ibse/IBSEServlet/XDSOAPRouter"/
```

Note: The best option for both changes is to use *localhost* instead of an IP address, which eliminates the need for this change.

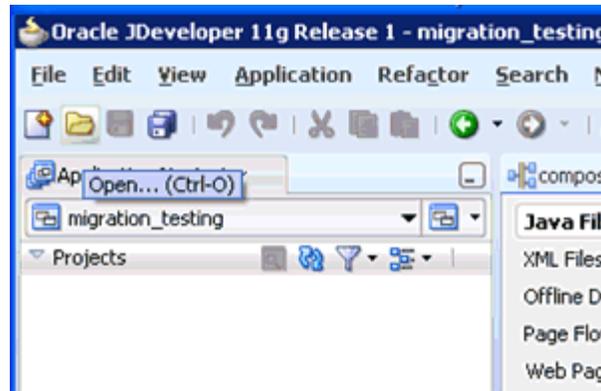
2.4 Upgrading 11g PS1/PS2/PS3/PS4/PS5 Outbound and Inbound BPEL and Mediator Processes to 11g PS6

As a prerequisite, ensure that the adapter targets and channels created using Application Explorer for J2CA configurations in the Oracle 11g PS6 environment are identical to those created in the Oracle 11g PS1/PS2/PS3/PS4/PS5 environment. For BSE configurations, ensure that the adapter targets and Business Services created using Application Explorer in the Oracle 11g PS6 environment are identical to those created in the Oracle 11g PS1/PS2/PS3/PS4/PS5 environment.

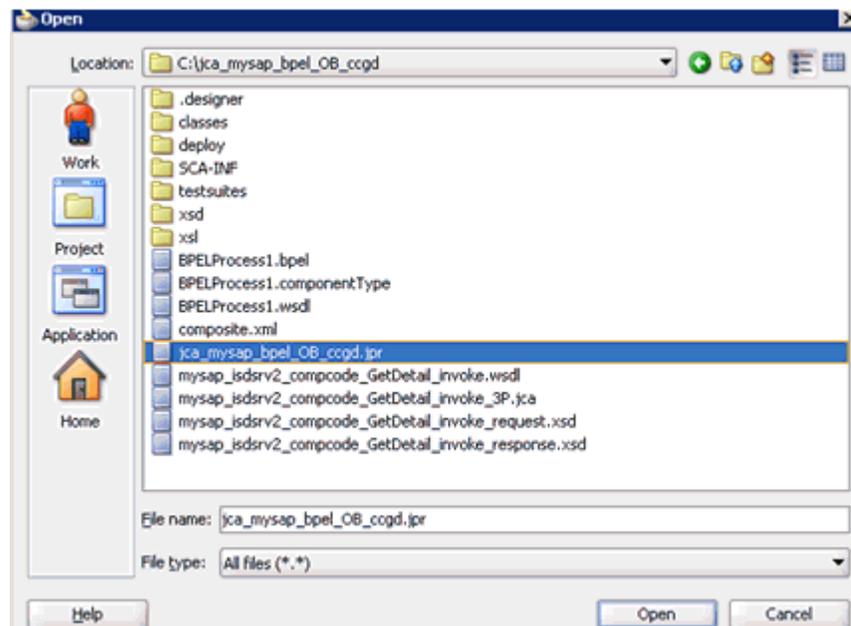
In addition, ensure that you copy the Oracle 11g PS1/PS2/PS3/PS4/PS5 BPEL and Mediator processes for BSE and J2CA to the Oracle 11g PS6 upgraded system location.

Note: 11g PS5 projects are compatible with Oracle JDeveloper 11g PS6, and no Migration Status messages are displayed while the project is opened.

1. Open Oracle JDeveloper 11g (11.1.1.7.0).
2. Select an available application (for example, migration-testing) and click **Open** from the toolbar, as shown in [Figure 2–76](#).

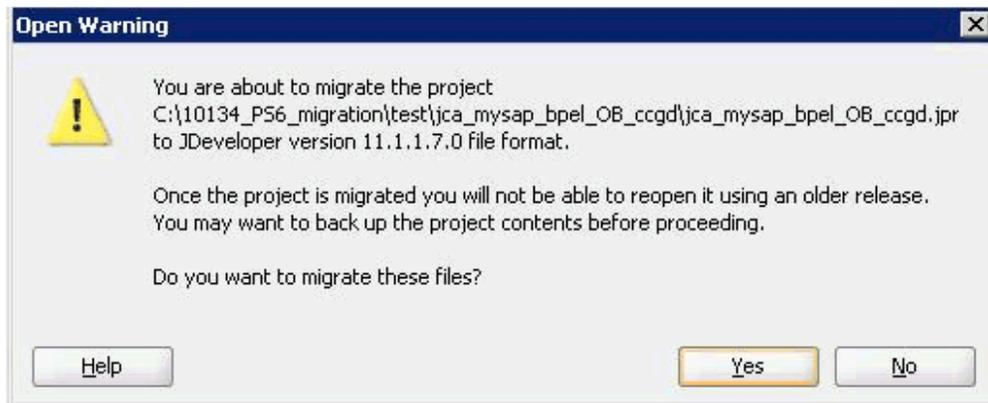
Figure 2–76 Oracle JDeveloper Toolbar

The Open dialog is displayed, as shown in [Figure 2–77](#).

Figure 2–77 Open Dialog

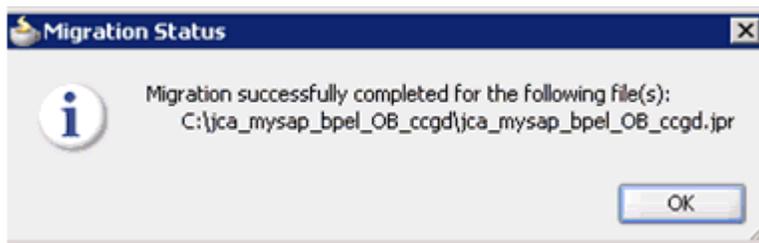
3. Open the 11g PS1/PS2/PS3/PS4/PS5 project (for example, jca_mysap_bpel_OB_ccgd) and then select the .jpr extension file (for example, jca_mysap_bpel_OB_ccgd.jpr).
4. Click **Open**.

The Open Warning dialog is displayed, as shown in [Figure 2–78](#).

Figure 2–78 Open Warning Dialog

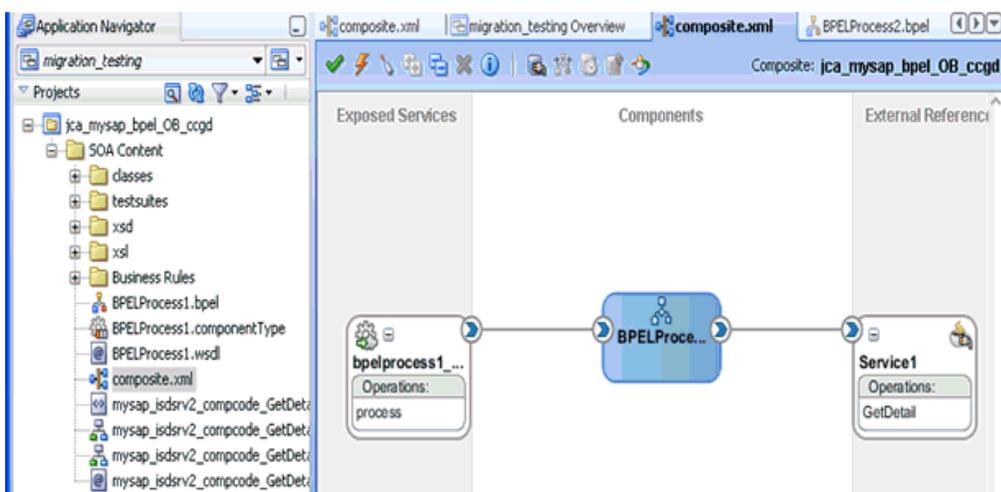
5. Click **Yes**.

The Migration Status message is displayed, as shown in [Figure 2–79](#).

Figure 2–79 Migration Status Message

6. Click **OK**.

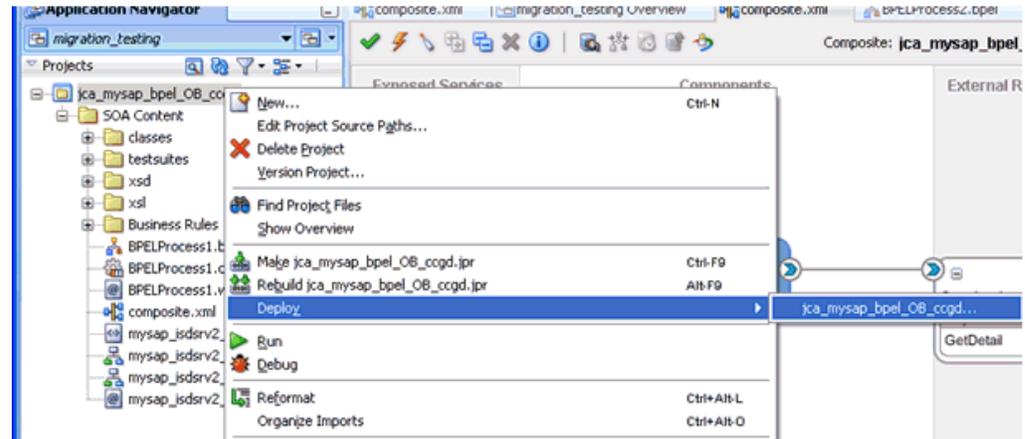
The Oracle 11g PS1/PS2/PS3/PS4/PS5 project is now available in your Oracle 11g PS6 environment, as shown in [Figure 2–80](#).

Figure 2–80 Migrated Sample Project

7. Expand the migrated project in the left pane (for example, `jca_mysap_bpel_ob_ccgd`) and then double-click the **composite.xml** file to verify that the project opens without any errors.

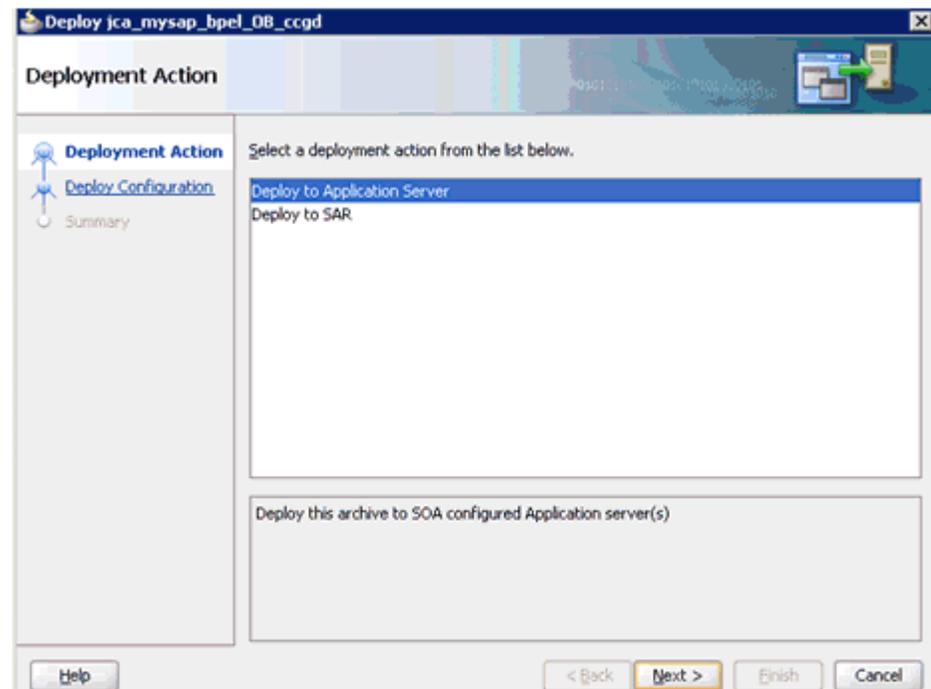
8. Click **Save**.
9. Right-click the migrated project, click **Deploy**, and select the project name from the menu (for example, `jca_mysap_bpel_OB_ccgd`), as shown in [Figure 2–81](#).

Figure 2–81 Deploy Menu Option



The Deployment Action page is displayed, as shown in [Figure 2–82](#).

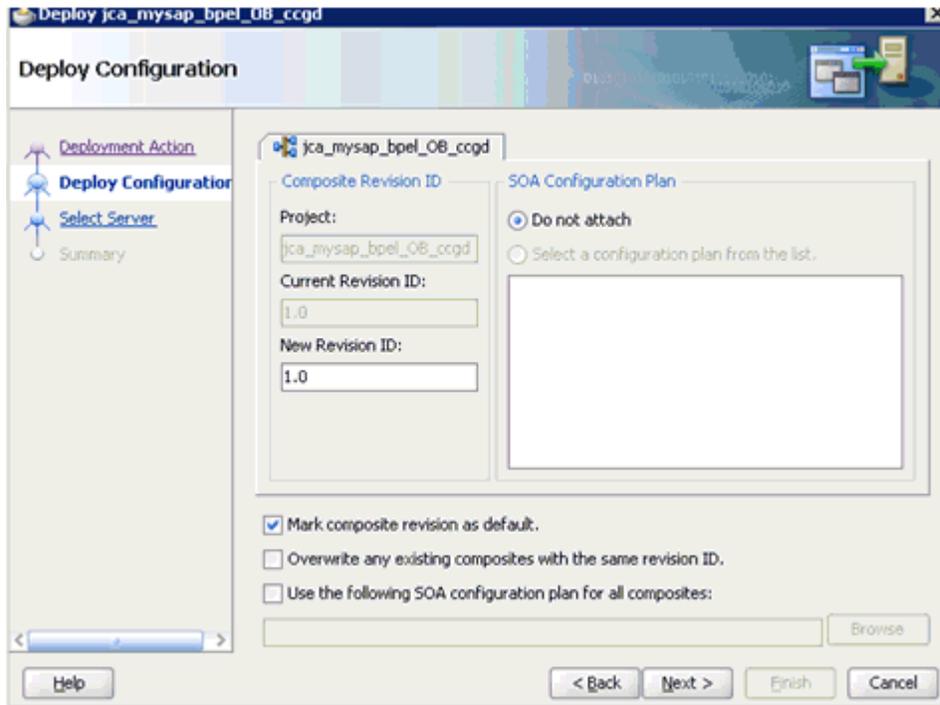
Figure 2–82 Deployment Action Page



10. Ensure that **Deploy to Application Server** is selected.
11. Click **Next**.

The Deploy Configuration page is displayed, as shown in [Figure 2–83](#).

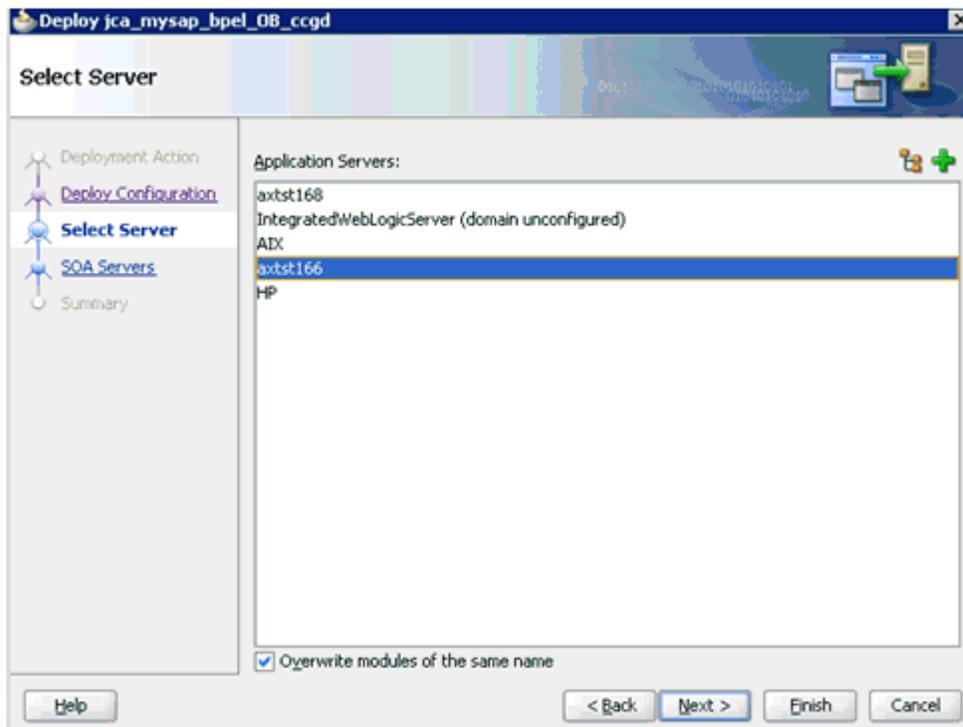
Figure 2–83 Deploy Configuration Page



12. Leave the default values selected and click **Next**.

The Select Server page is displayed, as shown in [Figure 2–84](#).

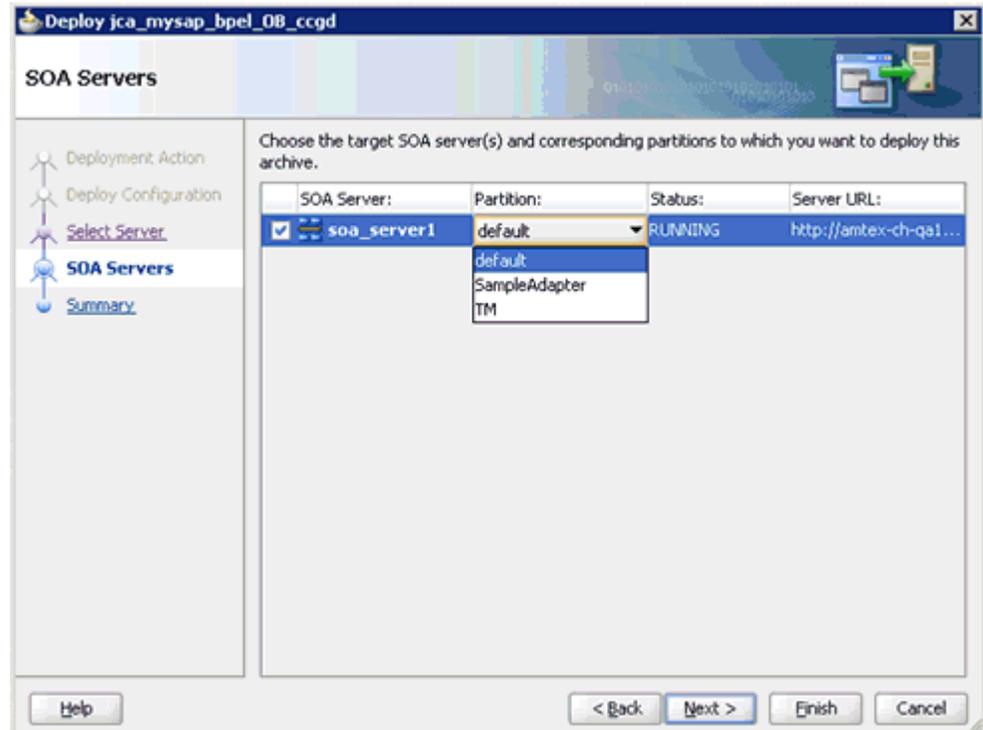
Figure 2–84 Select Server Page



13. Select the configured server and click **Next**.

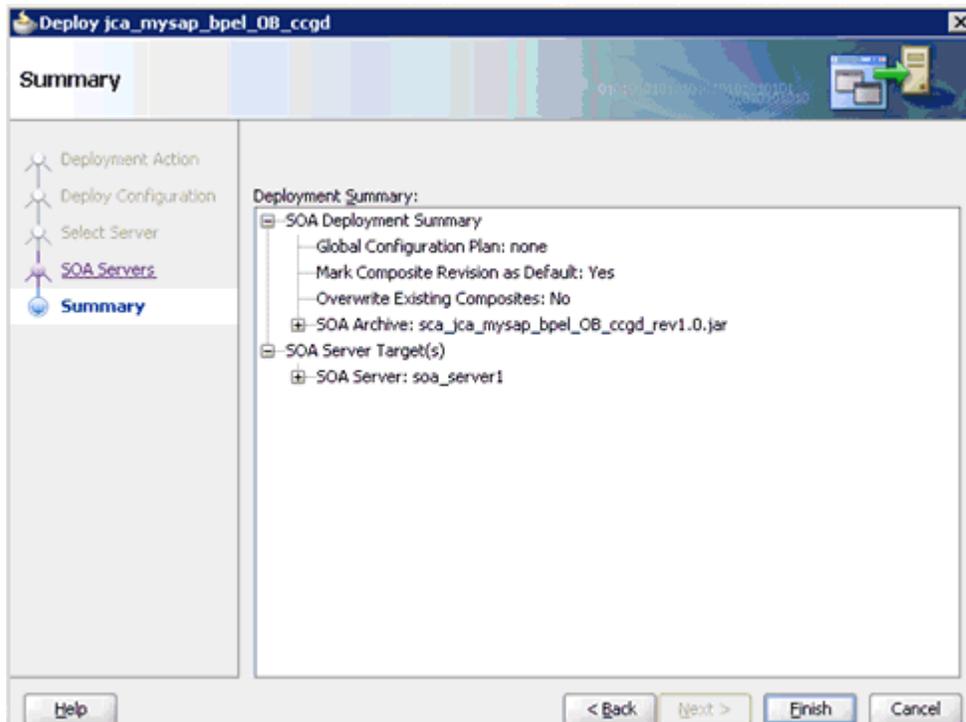
The SOA Servers page is displayed, as shown in [Figure 2–85](#).

Figure 2–85 SOA Servers Page

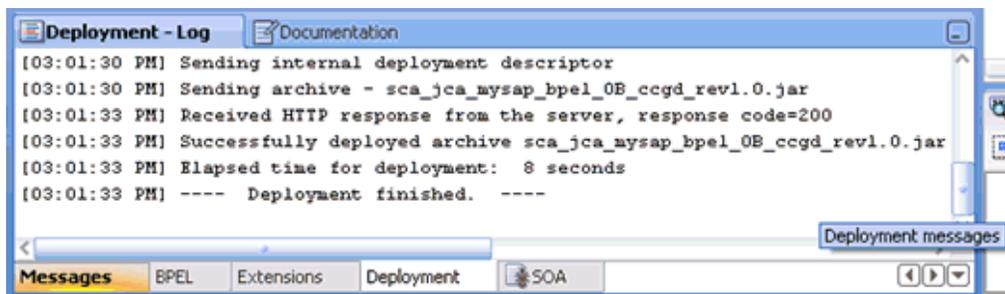


14. Select a partition from the Partition column list and click **Next**.

The Summary page is displayed, as shown in [Figure 2–86](#).

Figure 2–86 Summary Page

15. Review and verify all the available deployment information for your project and click **Finish**.
16. Verify that there are no error or warning messages during compilation and deployment in the deployment log, as shown in [Figure 2–87](#).

Figure 2–87 Deployment Log

17. After the deployment is successful, open the Oracle Enterprise Manager console and execute the deployed process either in Tree View or XML View to receive a successful response, as shown in [Figure 2–88](#) and [Figure 2–89](#).

Figure 2–88 Input XML in XML View

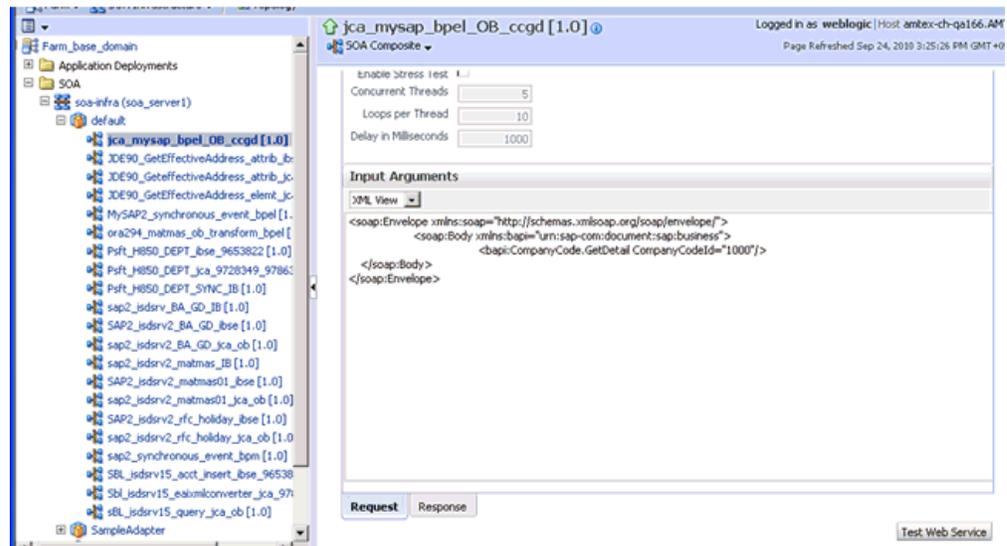
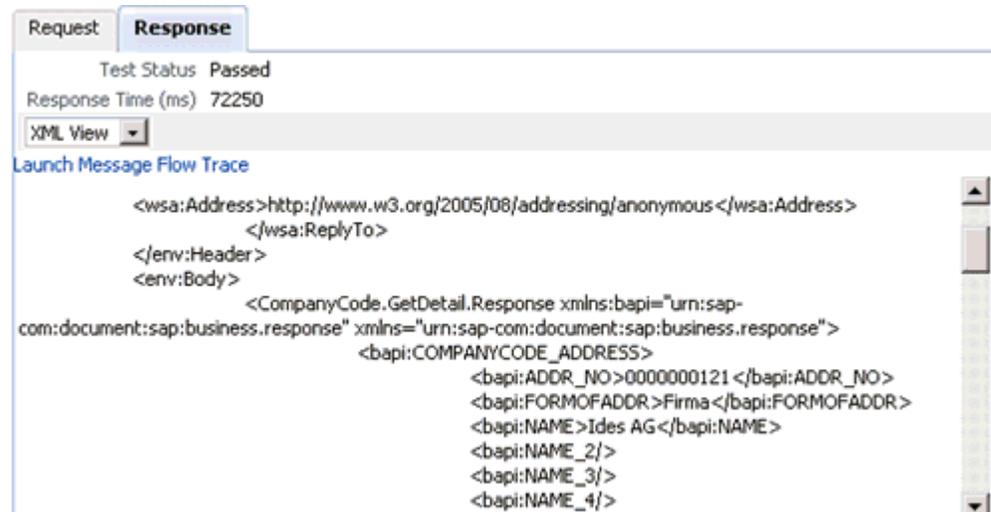


Figure 2–89 Received Output XML



2.4.1 Workaround for PS1 J.D. Edwards OneWorld Outbound Mediator Process When Migrating to 11g PS2/PS3/PS4/PS5/PS6

Once the J.D. Edwards OneWorld Mediator process is copied into 11g PS2/PS3/PS4/PS5/PS6 system, perform the following steps:

1. Open the PS1 project folder and open the WSDL file generated from Application Explorer to edit and perform the following changes:

Original

```
<definitions
name="GetPhone"
targetNamespace="http://xmlns.oracle.com/pcbpel/iWay/wsd1/JDEdwards/jde_90_attr_tgt/GetPhone"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:iWayRequest="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
xmlns:GetPhone="http://xmlns.oracle.com/pcbpel/iWay/wsd1/JDEdwards/jde_90_attr_
```

```

tgt/GetPhone"
xmlns:plt="http://schemas.xmlsoap.org/ws/2003/05/partner-link/">
.....
<schema xmlns="http://www.w3.org/2001/XMLSchema">
<import schemaLocation="jde90_GetPhone_invoke_response.xsd"
  namespace="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone" />
</schema>
.....
<message name="response">
<part name="output_GetPhone" element="iWayRequest:jdeResponse" />
</message>

```

Modified

```

<definitions
name="GetPhone"
targetNamespace="http://xmlns.oracle.com/pcbpel/iWay/wsd1/JDEdwards/jde_90_
attr_tgt/GetPhone"
xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:iWayResponse="urn:iwaysoftware:jd
e/services/CALLBSFN/Addressbook/GetPhone.response" xmlns:iWayRequest="urn:iwayso
ftware:jde/services/CALLBSFN/Addressbook/GetPhone"
xmlns:GetPhone="http://xmlns.oracle.com/pcbpel/iWay/wsd1/JDEdwards/jde_90_attr_
tgt/GetPhone"
xmlns:plt="http://schemas.xmlsoap.org/ws/2003/05/partner-link/">
.....
<schema xmlns="http://www.w3.org/2001/XMLSchema">
<import schemaLocation="jde90_GetPhone_invoke_response.xsd"
namespace="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone.response"
"/>
</schema>
.....
<message name="response">
<part name="output_GetPhone" element="iWayResponse:jdeResponse" />
</message>

```

2. Open the *response.xsd* file generated from Application Explorer to edit and *.response* to the end of the *targetNamespace* and *ns* declaration. For example:

Original

```

<xsd:schema xmlns:xsd=http://www.w3.org/2001/XMLSchema
targetNamespace="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
xmlns:ns="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
elementFormDefault="qualified">

```

Modified

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone.re
sponse"
xmlns:ns="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone.response"
elementFormDefault="qualified">

```

3. Open the *MediatorComponentName.wsdl* file (for example, *Mediator.wsdl*) to edit and perform the following changes:

Original

```

<wsdl:definitions
  name="Mediator1"
targetNamespace="http://xmlns.oracle.com/ps1_project/ps1_jca_jd90_medi_

```

```

gephone/Mediator1"
    xmlns:wSDL=http://schemas.xmlsoap.org/wSDL/
    xmlns:inp1="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
xmlns:tns=http://xmlns.oracle.com/ps1_project/ps1_jca_jd90_medi_
gephone/Mediator1"xmlns:out1="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook
/GetPhone">
<wSDL:import
namespace="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
location="jde90_GetPhone_invoke.wSDL"/>
<wSDL:import
namespace="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
location="jde90_GetPhone_invoke.wSDL"/>
.....
<wSDL:message name="replyMessage">
    <wSDL:part name="reply" element="inp1:jdeResponse"/>
</wSDL:message>

```

Modified

```

<wSDL:definitions
    name="Mediator1"
targetNamespace="http://xmlns.oracle.com/ps1_project/ps1_jca_jd90_medi_
gephone/Mediator1"
    xmlns:wSDL="http://schemas.xmlsoap.org/wSDL/"
    xmlns:inp1="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
xmlns:tns="http://xmlns.oracle.com/ps1_project/ps1_jca_jd90_medi_
gephone/Mediator1"xmlns:out1="urn:iwaysoftware:jde/services/CALLBSFN/Addressboo
k/GetPhone.response">
<wSDL:import
namespace="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
location="jde90_GetPhone_invoke.wSDL"/>
<wSDL:import
namespace="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone.response"
location="jde90_GetPhone_invoke.wSDL"/>
.....
<wSDL:message name="replyMessage">
    <wSDL:part name="reply" element="out1:jdeResponse"/>
</wSDL:message>

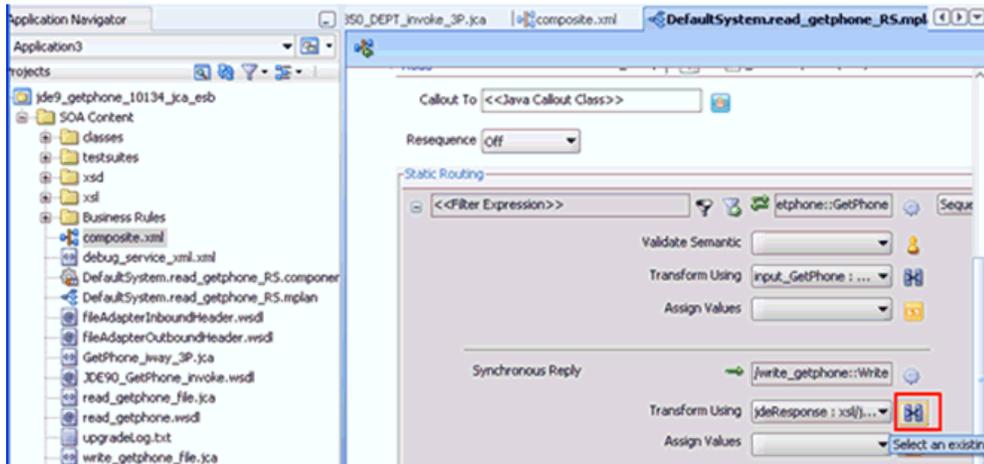
```

4. Start Oracle JDeveloper 11g and migrate the PS1 Mediator project.
5. Once the project is migrated, a successful message is received in Oracle JDeveloper.
6. Expand the migrated project and double-click the **composite.xml** file to ensure that the project opens without any errors.

Since changes were made to the WSDL file for the response section, a new mapper file must be created for *jdeResponse*.

7. Double-click the mediator component and go to the Synchronous Reply area.
8. Click the **Select an existing mapper file or create a new one** button in the Transform Using field, as shown in [Figure 2-90](#).

Figure 2–90 Transform Using Field



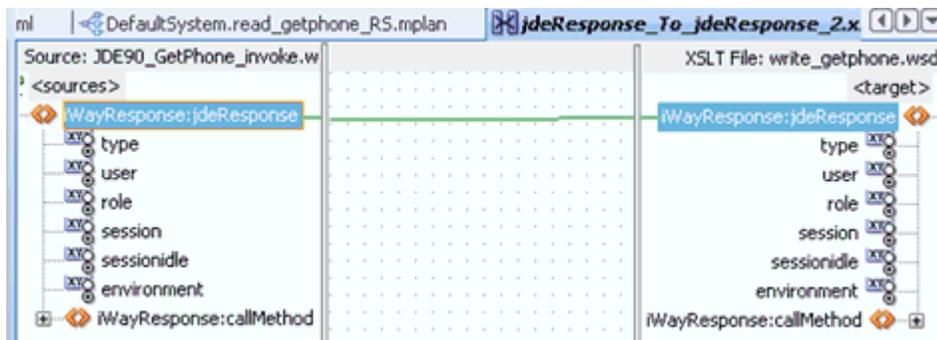
The Reply Transformation Map dialog is displayed, as shown in [Figure 2–91](#).

Figure 2–91 Reply Transformation Map Dialog

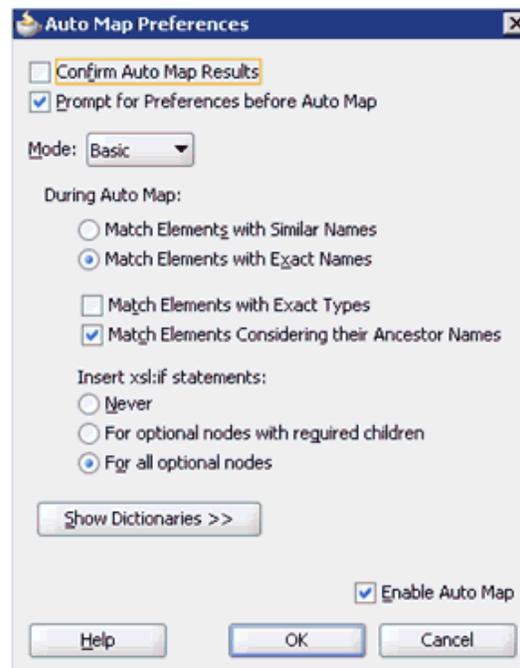


9. Click the **Create New Mapper File** option and click **OK**.
10. Automap the **iWayResponse;jdeResponse** in the source and **iWayResponse;jdeResponse** in the target, as shown in [Figure 2–92](#).

Figure 2–92 Mapping Source To Target

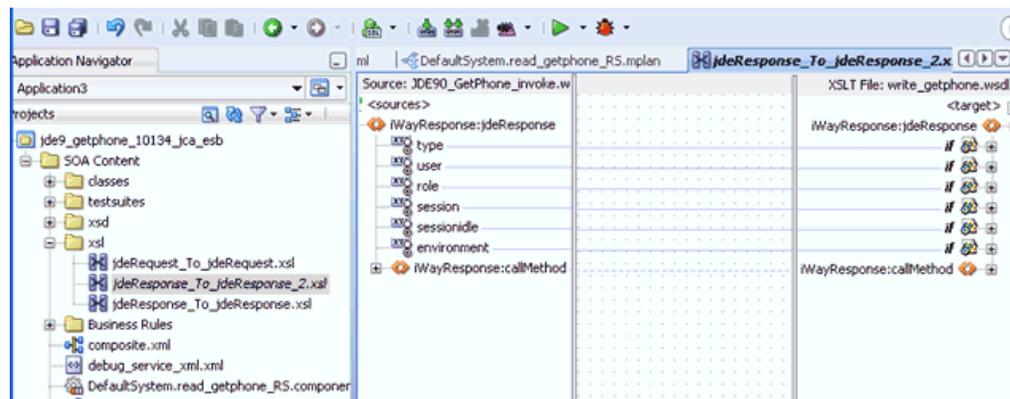


The Auto Map Preferences dialog is displayed, as shown in [Figure 2–93](#).

Figure 2–93 Auto Map Preferences Dialog

11. Accept the default values and click **OK**.

The automap is completed successfully, as shown in [Figure 2–94](#).

Figure 2–94 Completed Automap

12. Double click the **composite.xml** file.
13. Save and then deploy the migrated Mediator project.
14. Ensure that there are no error or warning messages during the deployment process.
15. Once the deployment is successful, navigate to the Oracle Enterprise Manager console and invoke the input XML file for the particular deployed project in the Request tab.

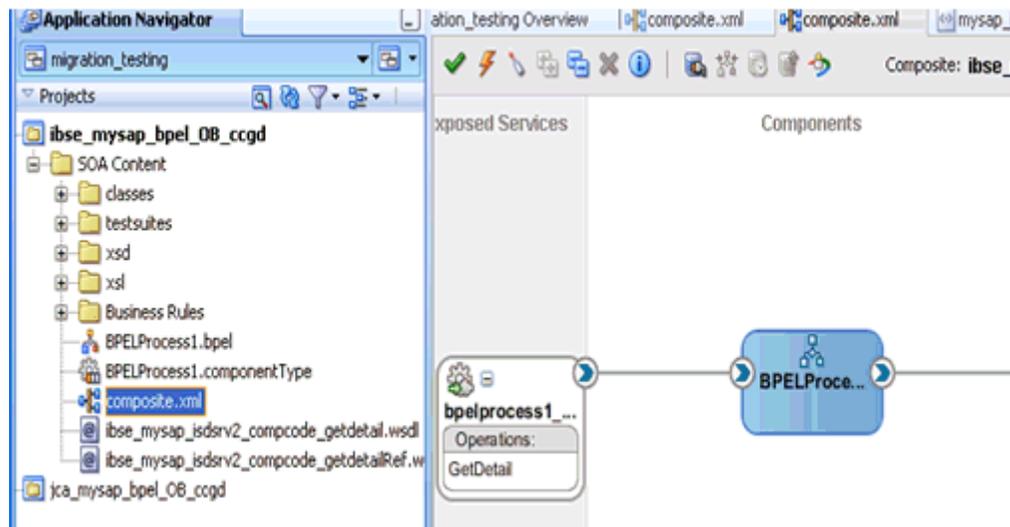
The successful response XML is received in the Response tab.

2.4.2 Additional Modifications for Migrated Processes in 11g PS6

Upgraded J2CA outbound and inbound processes in 11g PS6 function properly and do not require any additional modifications. Only BSE outbound processes require additional modifications, which are described in this section.

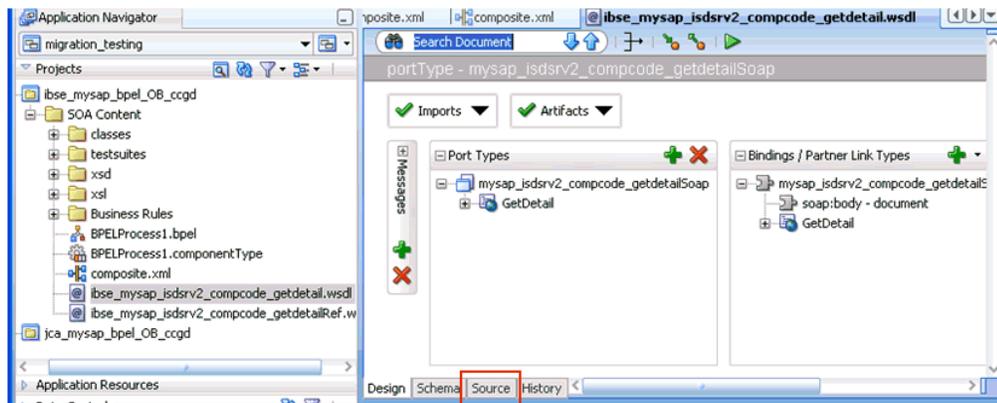
1. Once the BSE outbound process is migrated successfully to 11g PS6, double-click the `composite.xml` file to open the migrated project, as shown in [Figure 2-95](#)

Figure 2-95 Opened Migrated Project



2. Double-click the BSE outbound WSDL file and then click the **Source** tab, as shown in [Figure 2-96](#)

Figure 2-96 BSE Outbound WSDL File



3. Change the `<soap:address location>` element to point to the system where 11g PS6 is running.

For example:

```
<service name="mysap_isdsrv2_compcode_getdetail">
  <documentation/>
  <port name="mysap_isdsrv2_compcode_getdetailSoap1" binding="tns:mysap_isdsrv2_compcode_getdetailSoap">
```

```

<soap:address
location="http://172.19.95.190:8001/ibse/IBSEServlet/XDSOAPRouter" />
  </port>
</service>
</definitions>

```

Note: The best option for both changes is to use localhost instead of an IP address, which eliminates the need for this change.

4. Save and deploy the process.

2.5 Upgrading 11g PS2/PS3/PS4/PS5 Outbound and Inbound BPM Processes to 11g PS6

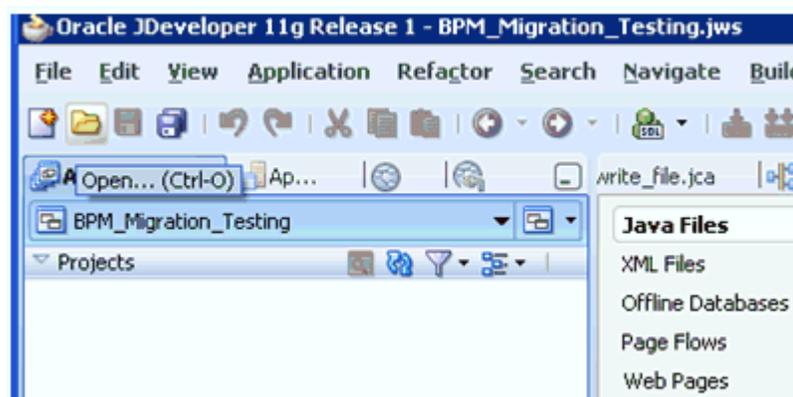
As a prerequisite, ensure that the adapter targets and channels created using Application Explorer for J2CA configurations in the Oracle 11g PS6 environment are identical to those created in the Oracle 11g PS2/PS3/PS4/PS5 environment. For BSE configurations, ensure that the adapter targets and Business Services created using Application Explorer in the Oracle 11g PS6 environment are identical to those created in the Oracle 11g PS2/PS3/PS4/PS5 environment.

In addition, ensure that you copy the Oracle 11g PS2/PS3/PS4/PS5 BPM processes for BSE and J2CA to the Oracle 11g PS6 upgraded system location.

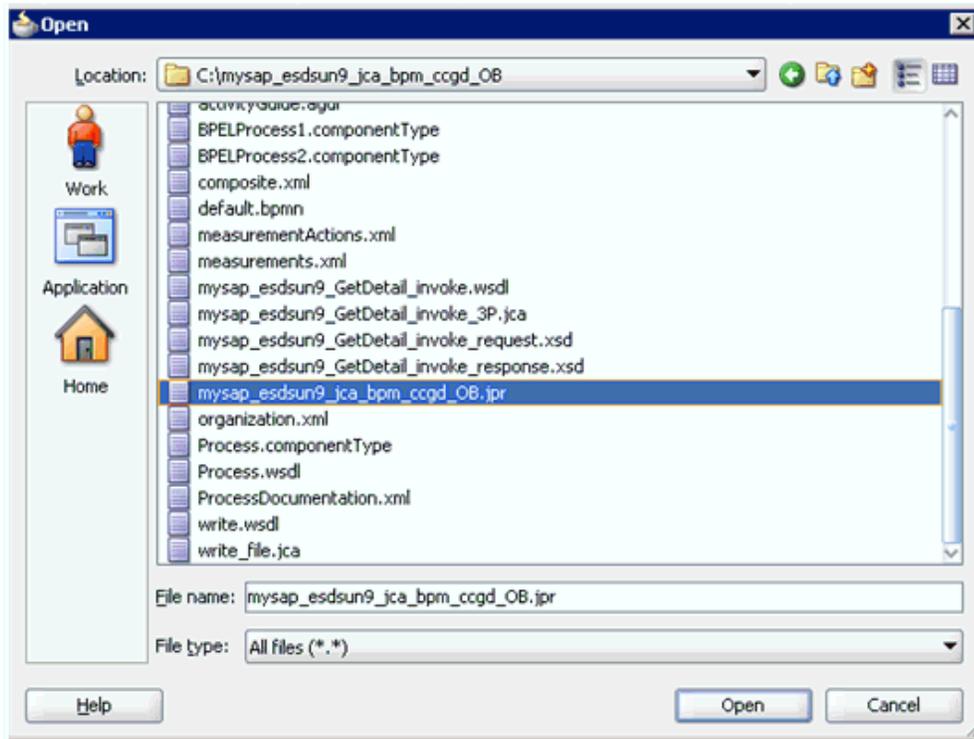
Note: 11g PS5 projects are compatible with Oracle JDeveloper 11g PS6, and no Migration Status messages are displayed while the project is opened.

1. Open Oracle JDeveloper 11g (11.1.1.7.0).
2. Select an available application (for example, BPM_Migration_Testing) and click **Open** from the toolbar, as shown in [Figure 2-97](#).

Figure 2-97 Oracle JDeveloper Toolbar

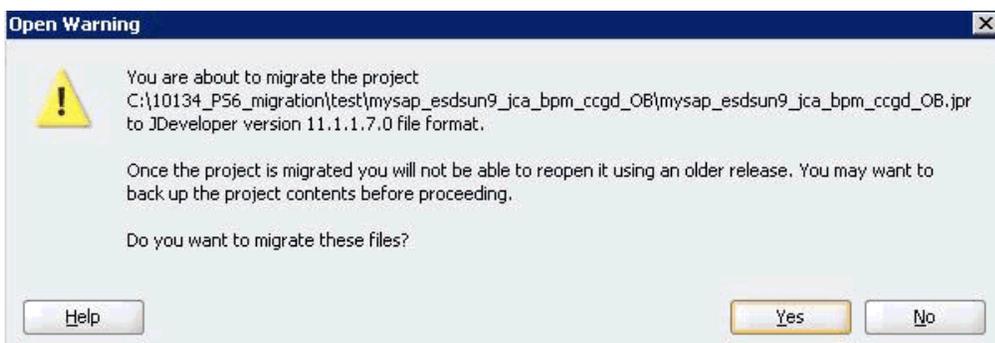


The Open dialog is displayed, as shown in [Figure 2-98](#).

Figure 2–98 Open Dialog

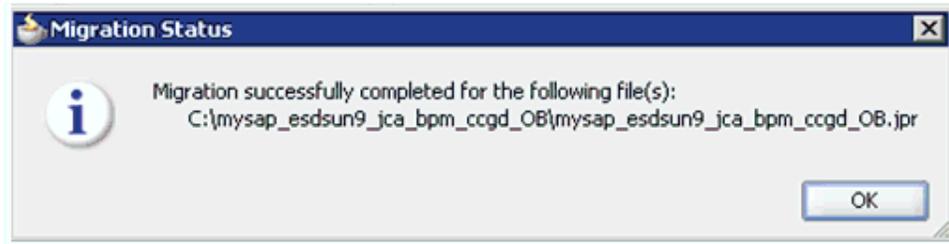
3. Open the 11g PS2/PS3/PS4/PS5 project (for example, mysap_esdsun9_jca_bpm_ccgd_OB) and then select the .jpr extension file (for example, mysap_esdsun9_jca_bpm_ccgd_OB.jpr).
4. Click **Open**.

The Open Warning dialog is displayed, as shown in [Figure 2–99](#).

Figure 2–99 Open Warning Dialog

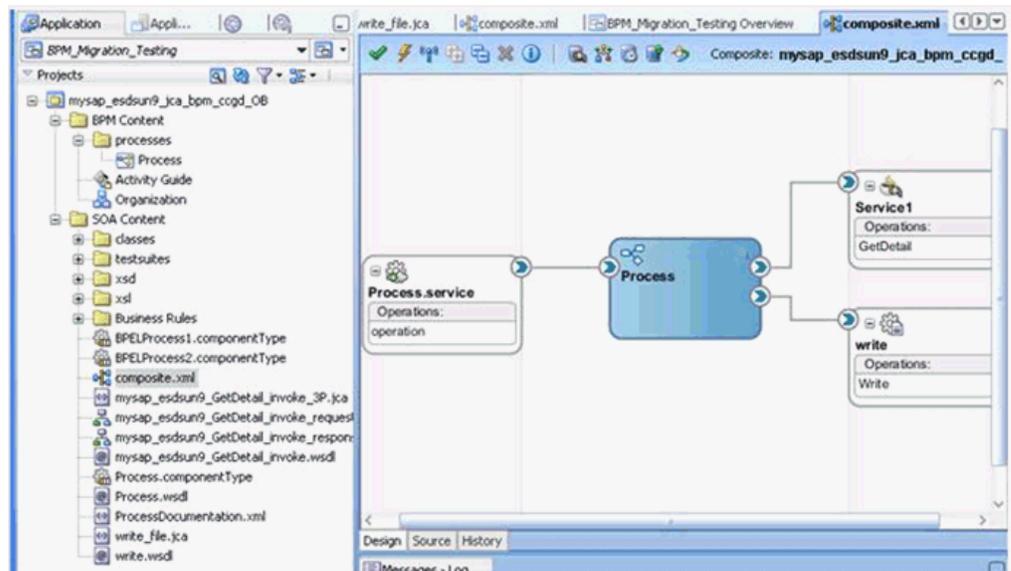
5. Click **Yes**.

The Migration Status message is displayed, as shown in [Figure 2–100](#).

Figure 2–100 Migration Status Message

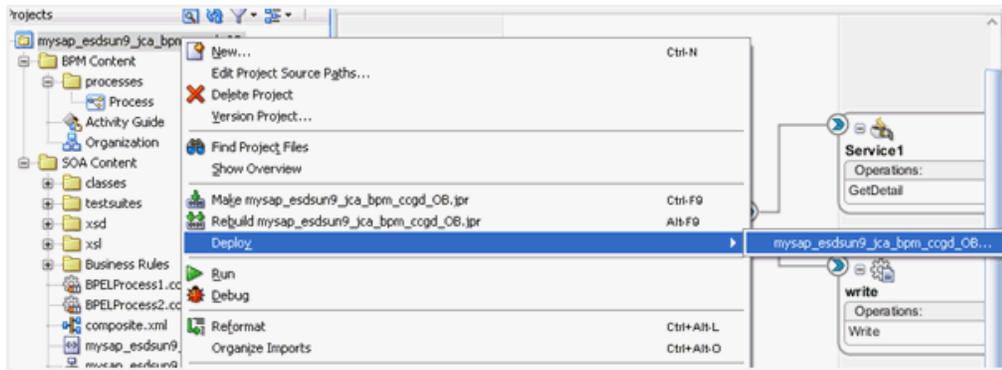
6. Click **OK**.

The Oracle 11g PS2/PS3/PS4/PS5 project is now available in your Oracle 11g PS6 environment, as shown in [Figure 2–101](#).

Figure 2–101 Migrated Sample Project

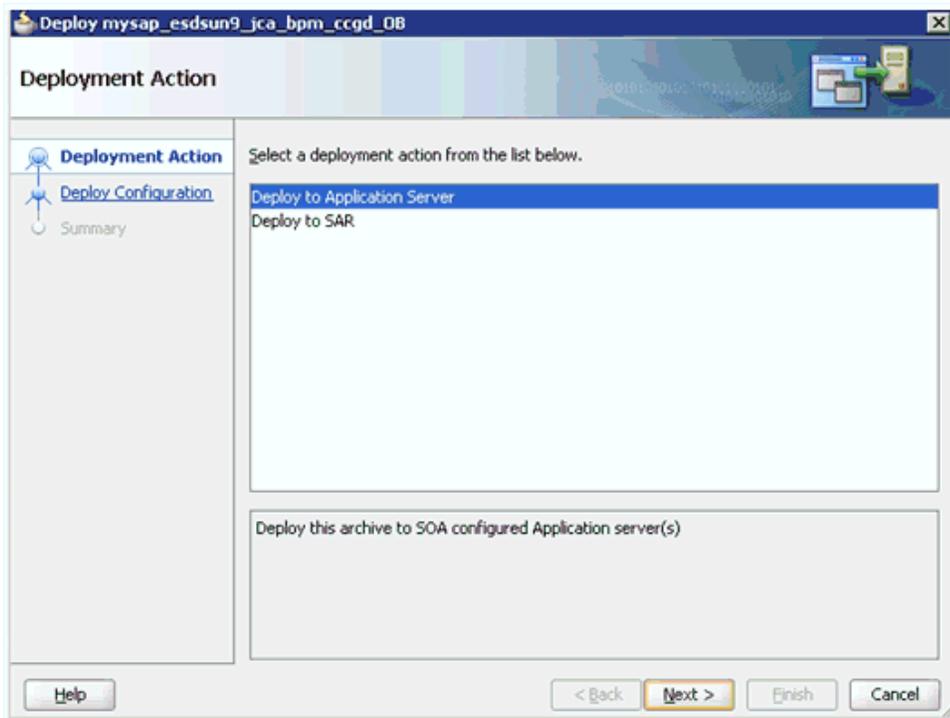
7. Expand the migrated project in the left pane (for example, `mysap_esdsun9_jca_bpm_ccgd_OB`) and then double-click the **composite.xml** file to verify that the project opens without any errors.
8. Click **Save**.
9. Right-click the migrated project, click **Deploy**, and select the project name from the menu (for example, `mysap_esdsun9_jca_bpm_ccgd_OB`), as shown in [Figure 2–102](#).

Figure 2–102 Deploy Menu Option



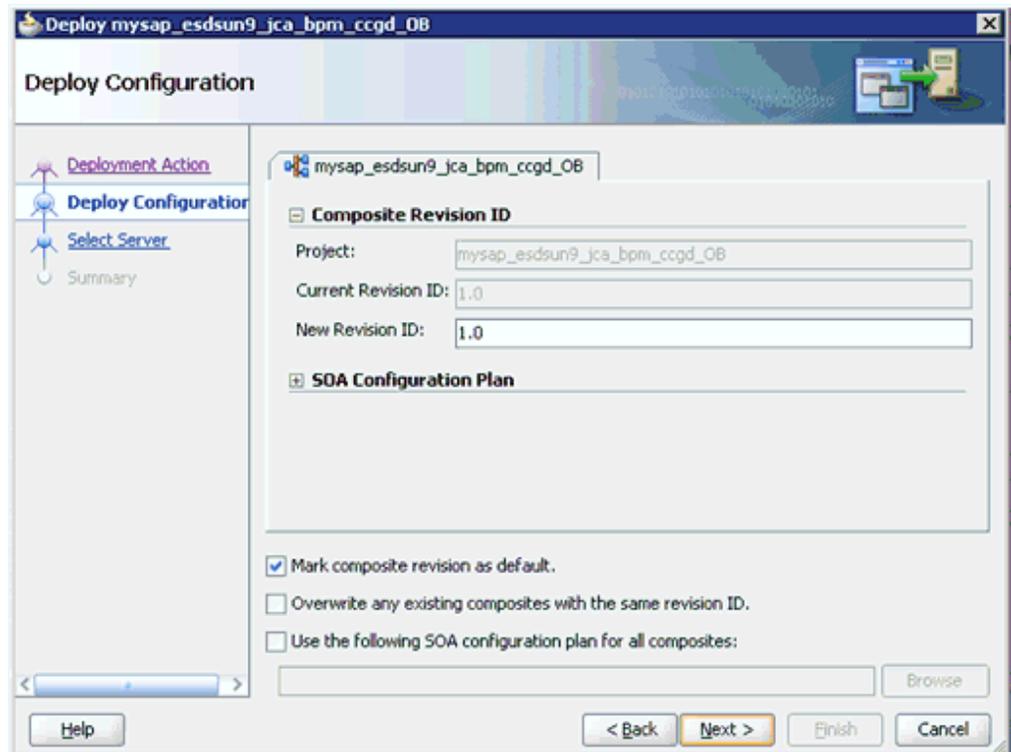
The Deployment Action page is displayed, as shown in [Figure 2–103](#).

Figure 2–103 Deployment Action Page



10. Ensure that **Deploy to Application Server** is selected.
11. Click **Next**.

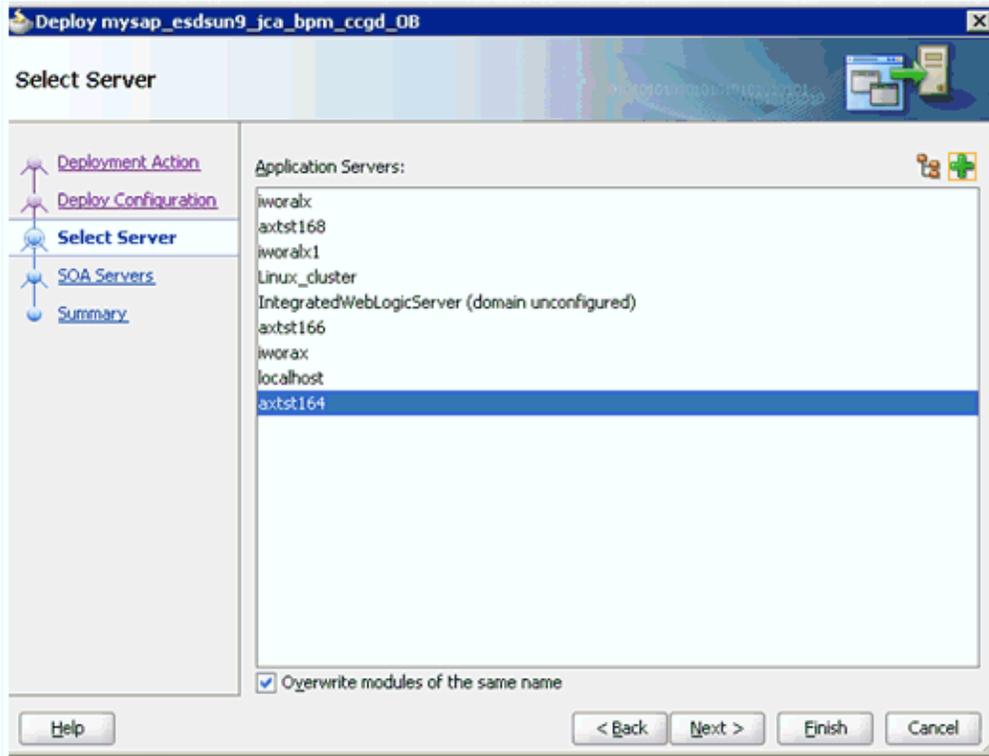
The Deploy Configuration page is displayed, as shown in [Figure 2–104](#).

Figure 2–104 Deploy Configuration Page

12. Leave the default values selected and click **Next**.

The Select Server page is displayed, as shown in [Figure 2–105](#).

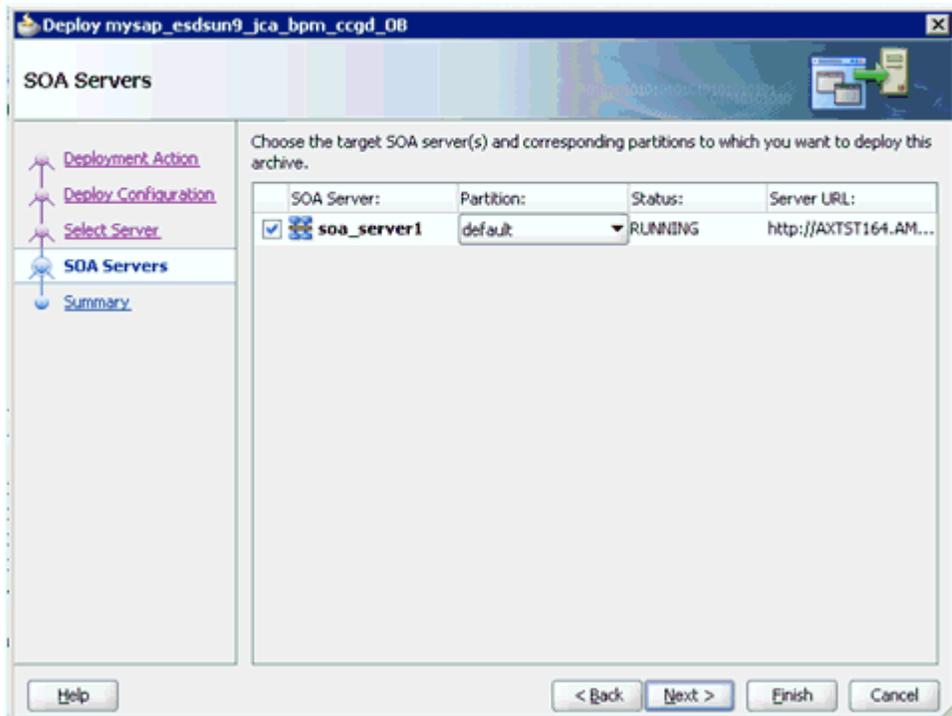
Figure 2–105 Select Server Page



13. Select the configured server and click **Next**.

The SOA Servers page is displayed, as shown in [Figure 2–106](#).

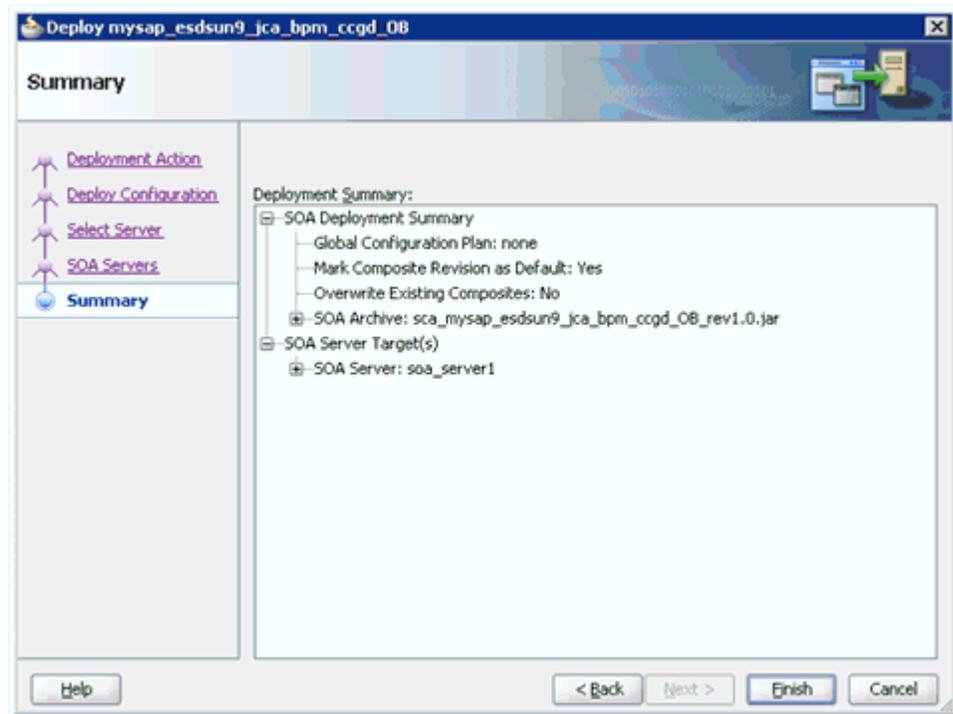
Figure 2–106 SOA Servers Page



14. Select a partition from the Partition column list and click **Next**.

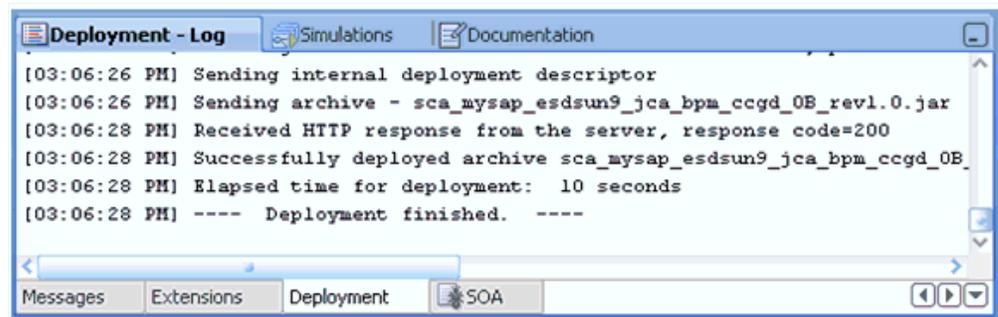
The Summary page is displayed, as shown in [Figure 2–107](#).

Figure 2–107 Summary Page



15. Review and verify all the available deployment information for your project and click **Finish**.
16. Verify that there are no error or warning messages during compilation and deployment in the deployment log, as shown in [Figure 2–108](#).

Figure 2–108 Deployment Log

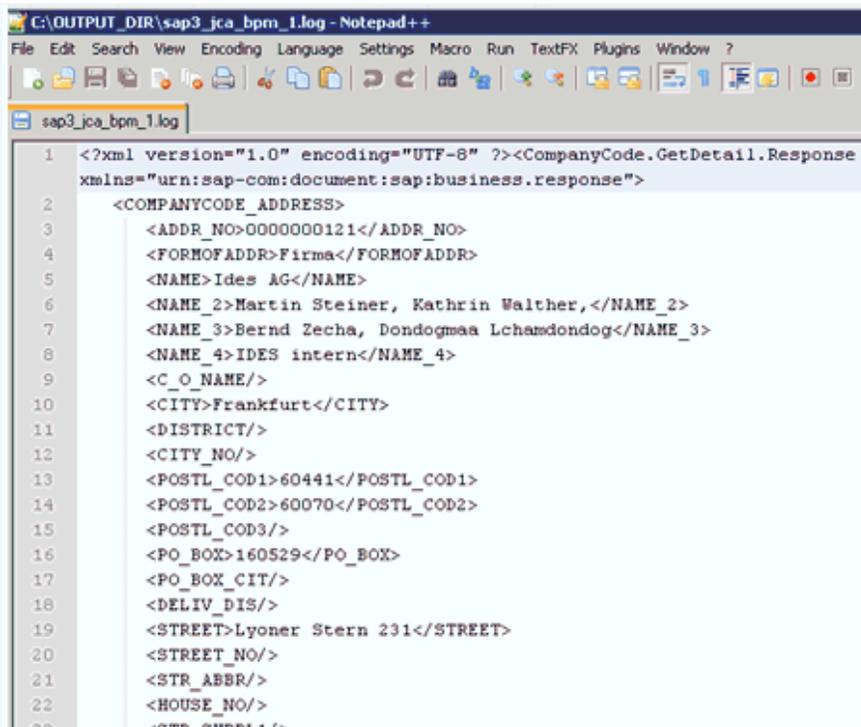


17. After the deployment is successful, open the Oracle Enterprise Manager console and execute the deployed process either in Tree View or XML View to receive a successful response, as shown in [Figure 2–109](#) and [Figure 2–110](#).

Figure 2–109 Input XML in XML View



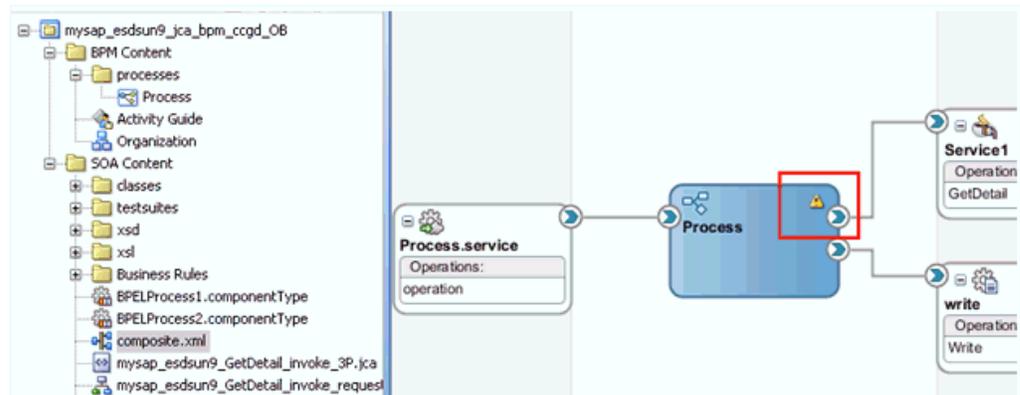
Figure 2–110 Received Output XML



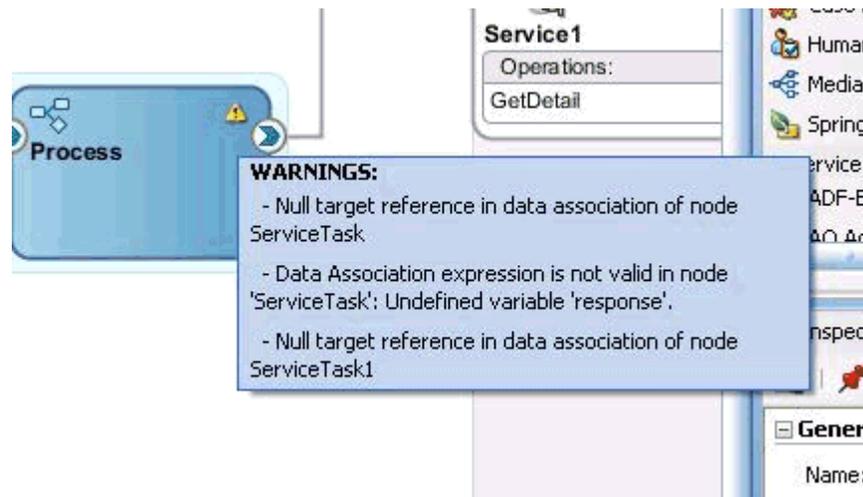
2.5.1 Workaround for J2CA BPM Processes

This section describes a workaround for J2CA BPM processes when migrating BAPI or RFC objects for MySAP adapter from 11g PS2/PS3/PS4/PS5 to 11g PS6.

When a 11g PS2/PS3/PS4/PS5 J2CA BPM outbound process for the CompanyCode GetDetail object is migrated, a warning symbol is displayed in the 11g PS6 version of Oracle JDeveloper, as shown in [Figure 2–111](#).

Figure 2–111 Warning Symbol

When you move your pointer over the warning symbol, warning messages are displayed, as shown in [Figure 2–112](#).

Figure 2–112 Warning Messages

For example:

WARNINGS:

- Null target reference in data association of node ServiceTask
- Data Association expression is not valid in node 'ServiceTask': Undefined variable 'response'.
- Null target reference in data association of node ServiceTask1

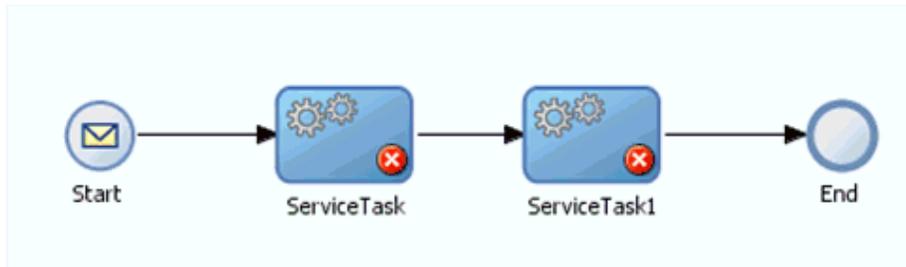
This section provides instructions on how to apply a workaround that resolves this issue.

Note: This workaround is applicable to Oracle BUG 11798374.

1. Open the **composite.xml** file from the project folder and double-click the BPM **Process** component.

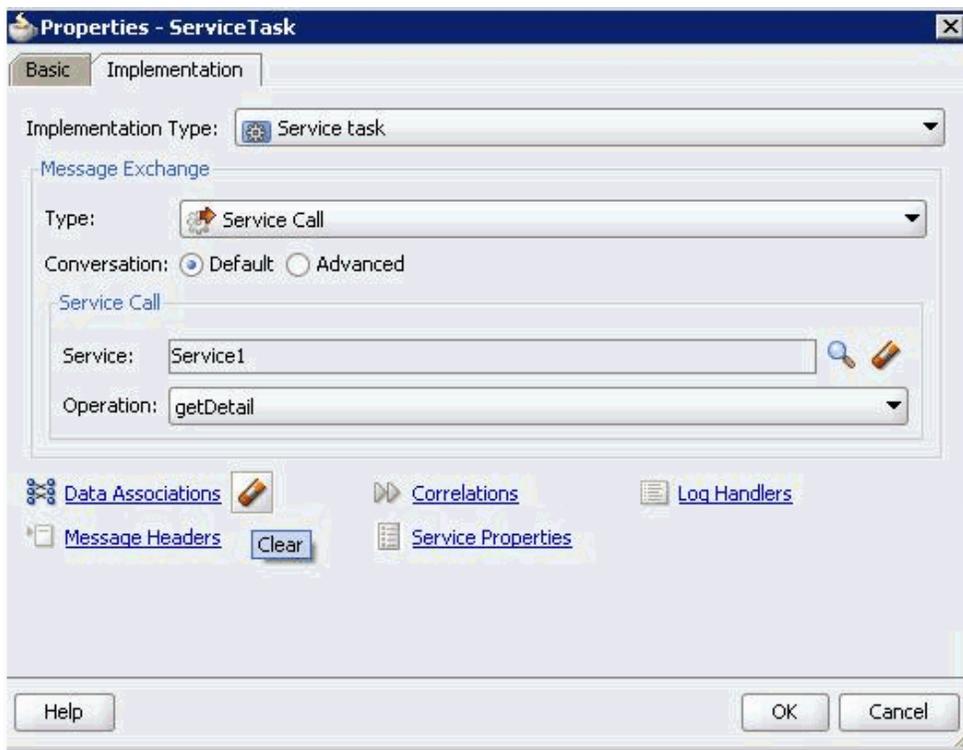
The BPM process is opened and you are able to see the ServiceTask and ServiceTask1 activities, including the error symbols, as shown in [Figure 2–113](#).

Figure 2–113 ServiceTask Activities

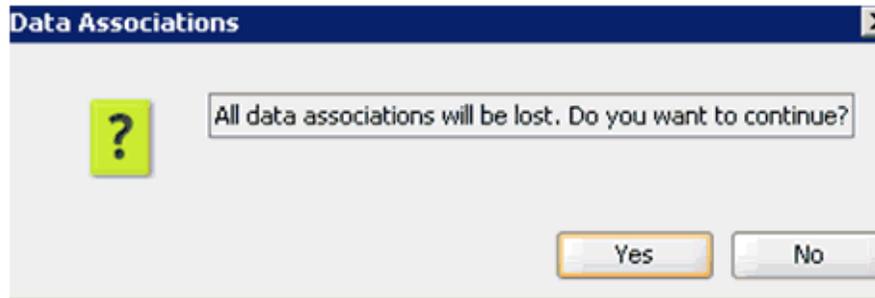


2. Double-click the **ServiceTask** activity.
The Properties - ServiceTask dialog is displayed, as shown in [Figure 2–114](#).

Figure 2–114 Properties - ServiceTask Dialog

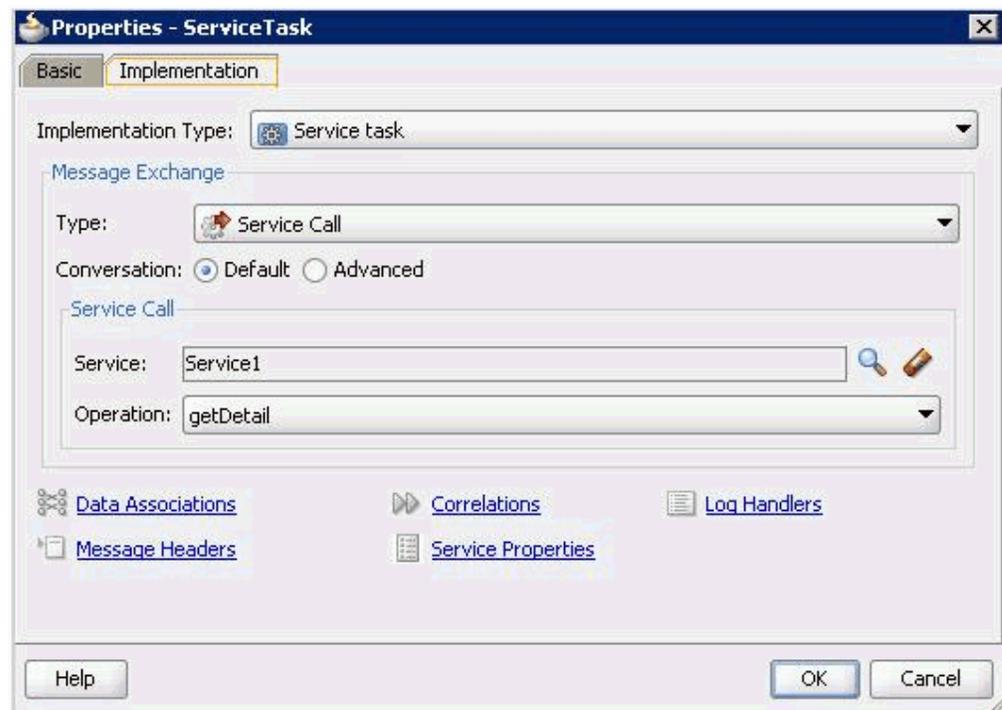


3. Click the **Implementation** tab.
4. Click the **Clear** icon next to the Data Associations icon.
A confirmation message is displayed in the Data Associations dialog, as shown in [Figure 2–115](#).

Figure 2–115 Confirmation Message

5. Click **Yes**.

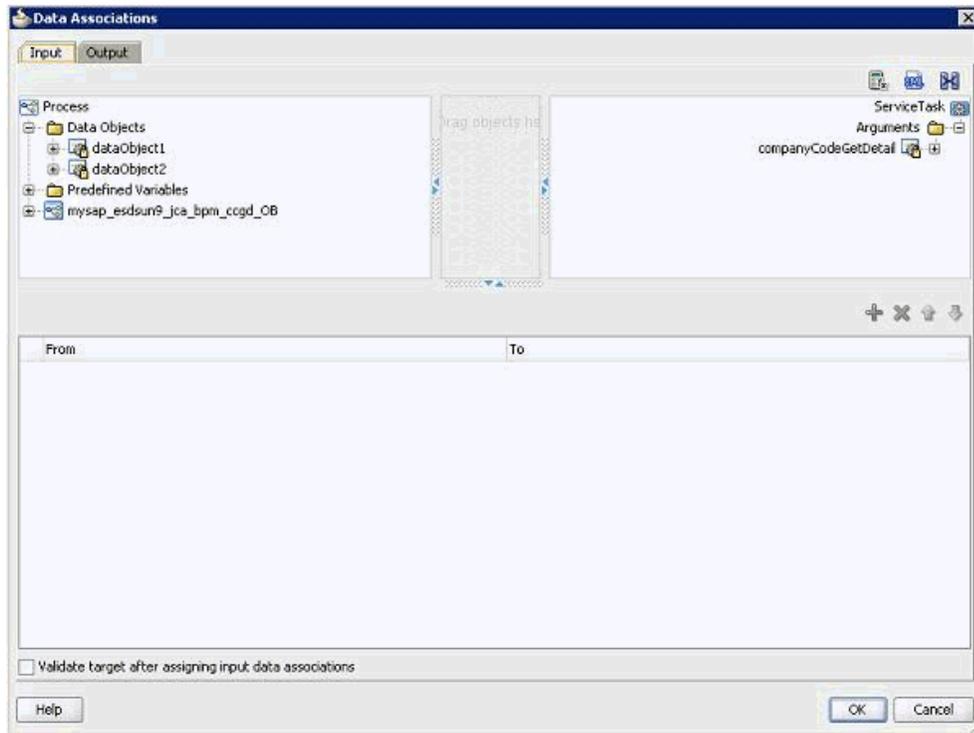
You are returned to the Properties - ServiceTask dialog, as shown in [Figure 2–116](#).

Figure 2–116 Properties - ServiceTask Dialog

6. Click the **Data Associations** icon.

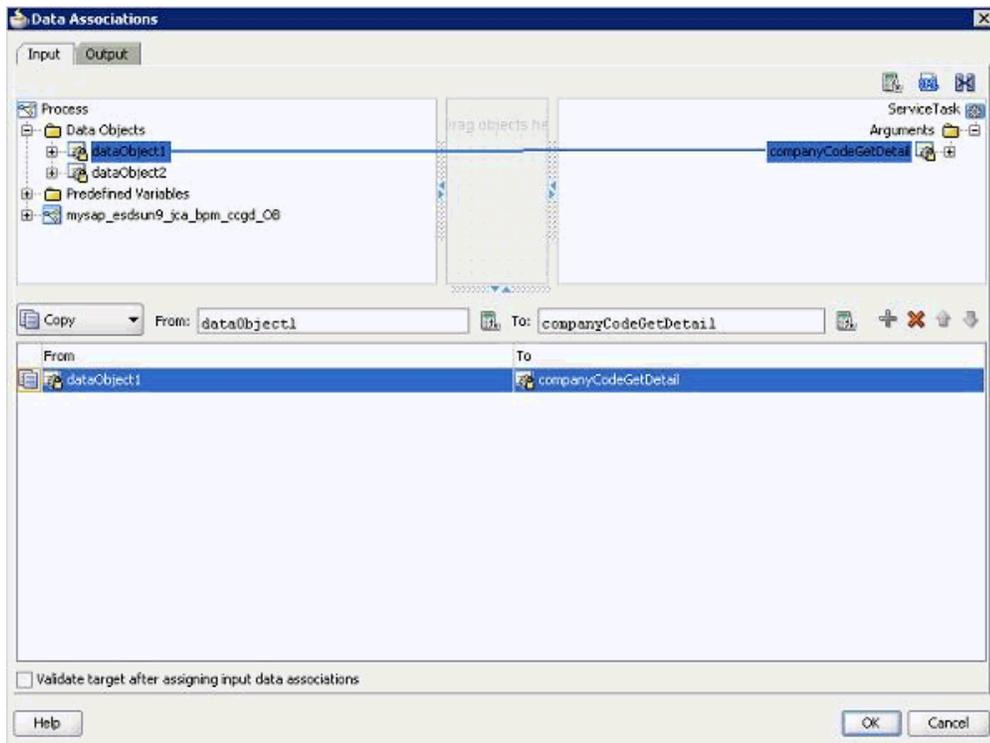
The Data Associations window is displayed, as shown in [Figure 2–117](#).

Figure 2–117 Data Associations Window



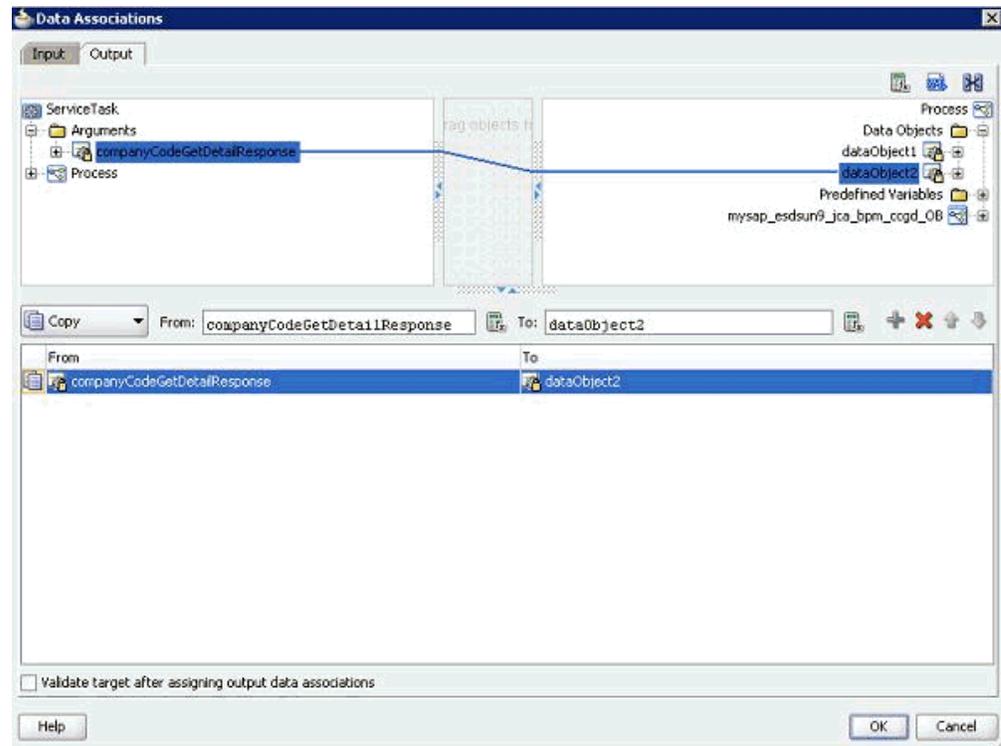
7. In the Input tab, drag and connect **dataobject1** (request) under Data Objects to **companyCodeGetDetail** under Arguments, as shown in [Figure 2–118](#).

Figure 2–118 Data Associations Window



8. In the Output tab, drag and connect **companyCodeGetDetailResponse** under Arguments to **dataobject2** (response) under Data Objects, as shown in [Figure 2–119](#).

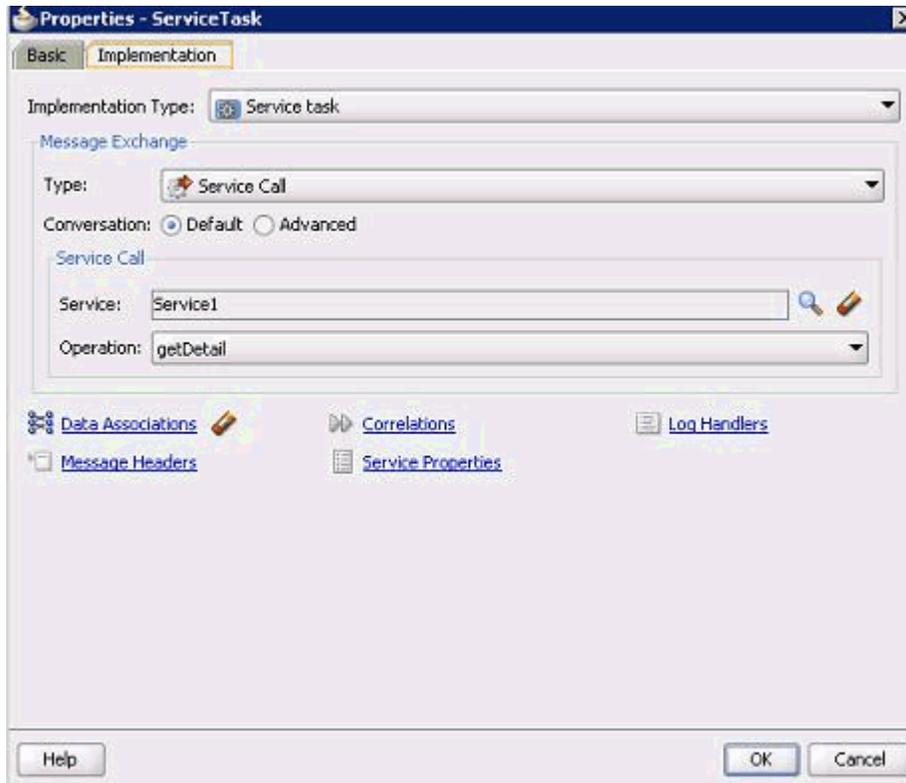
Figure 2–119 Data Associations Window



9. Click **OK**.

You are returned to the Properties - ServiceTask dialog, as shown in [Figure 2–120](#).

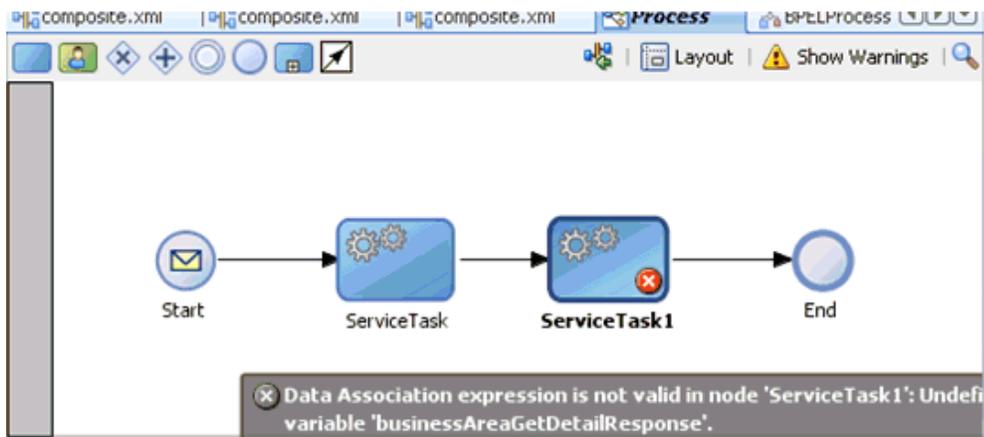
Figure 2–120 Properties - ServiceTask Dialog



10. Click OK.

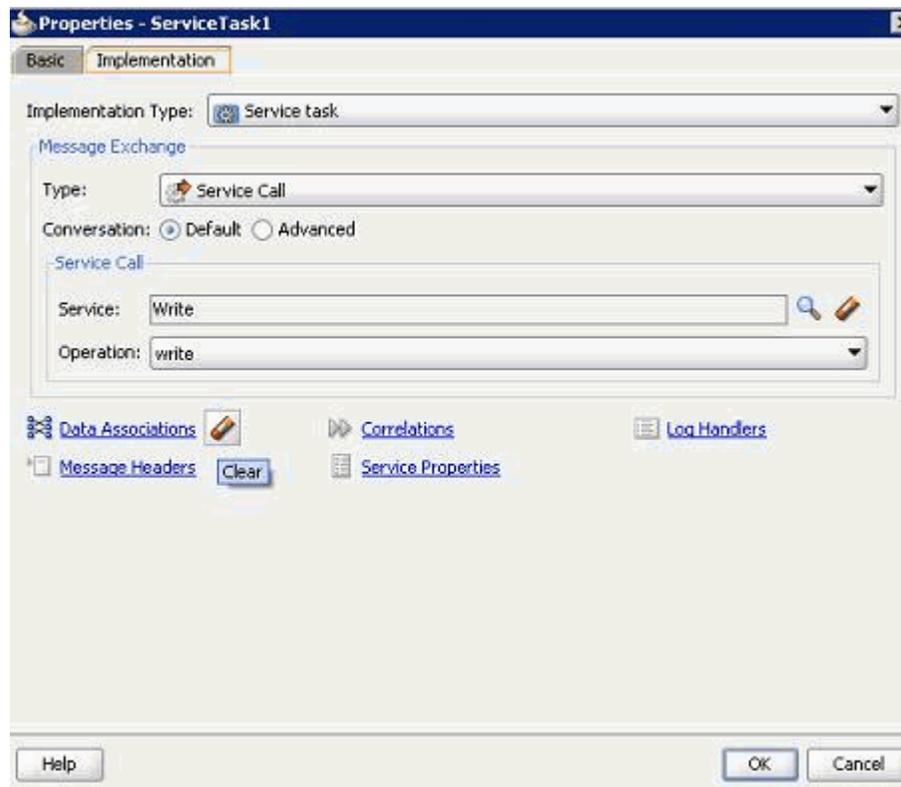
You are returned to the Process tab, as shown in [Figure 2–121](#).

Figure 2–121 Process Tab



11. Double-click the **ServiceTask1** activity.

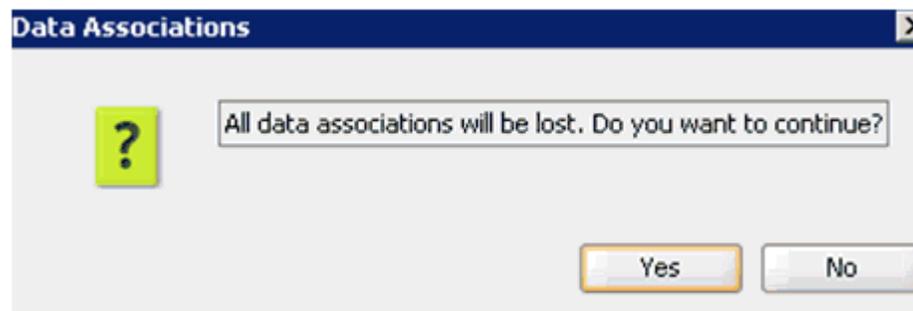
The Properties - ServiceTask1 dialog is displayed, as shown in [Figure 2–122](#).

Figure 2–122 Properties - ServiceTask1 Dialog

12. Click the **Implementation** tab.

13. Click the **Clear** icon next to the Data Associations icon.

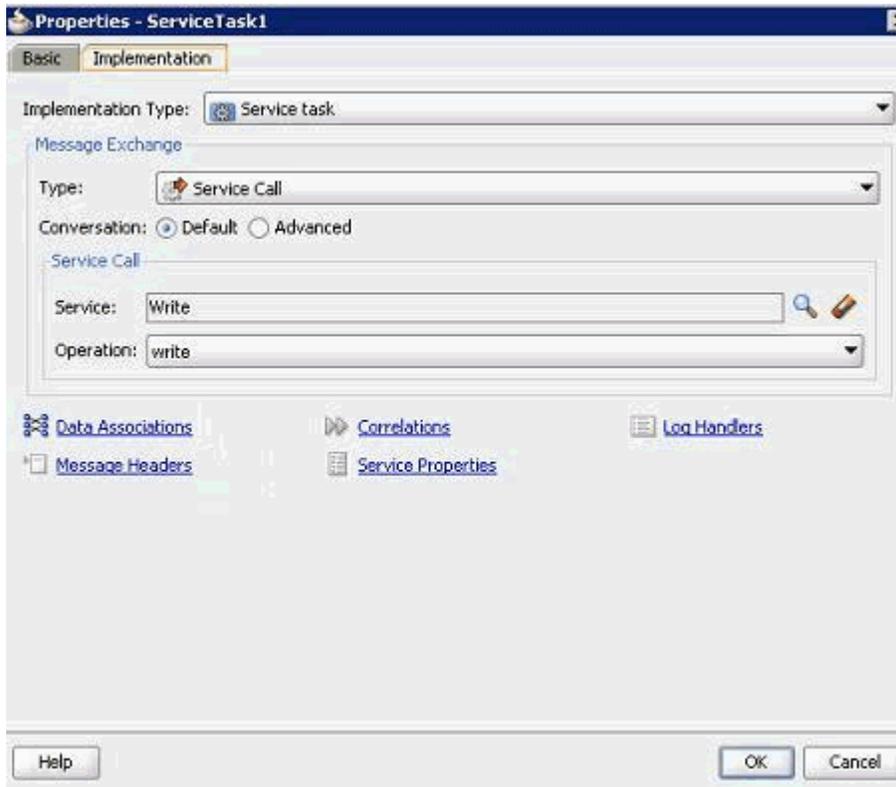
A confirmation message is displayed in the Data Associations dialog, as shown in [Figure 2–123](#).

Figure 2–123 Confirmation Message

14. Click **Yes**.

You are returned to the Properties - ServiceTask1 dialog, as shown in [Figure 2–124](#).

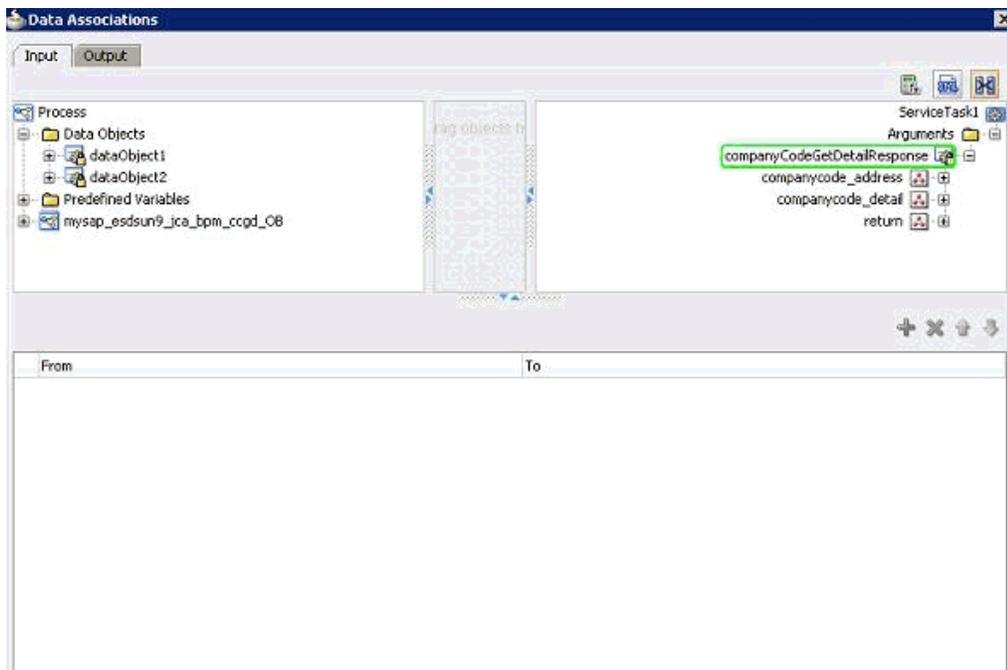
Figure 2–124 Properties - ServiceTask1 Dialog



15. Click the **Data Associations** icon.

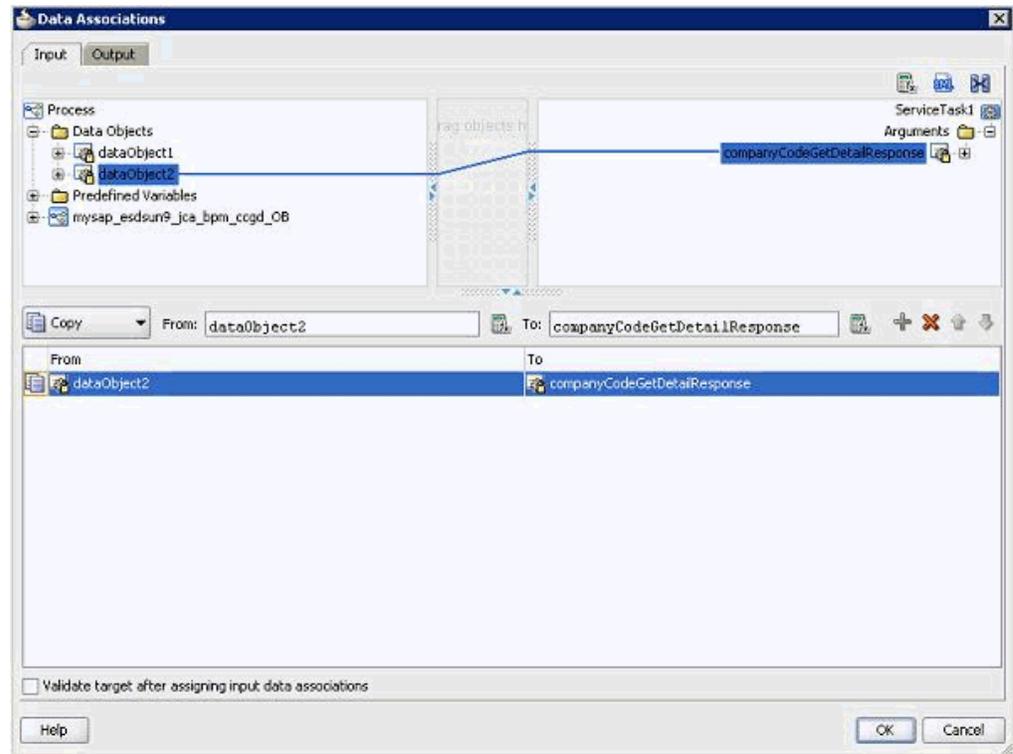
The Data Associations window is displayed, as shown in [Figure 2–125](#).

Figure 2–125 Data Associations Window



16. Drag and connect **dataObject2** (response) to **companyCodeGetDetailResponse**, as shown in [Figure 2-126](#).

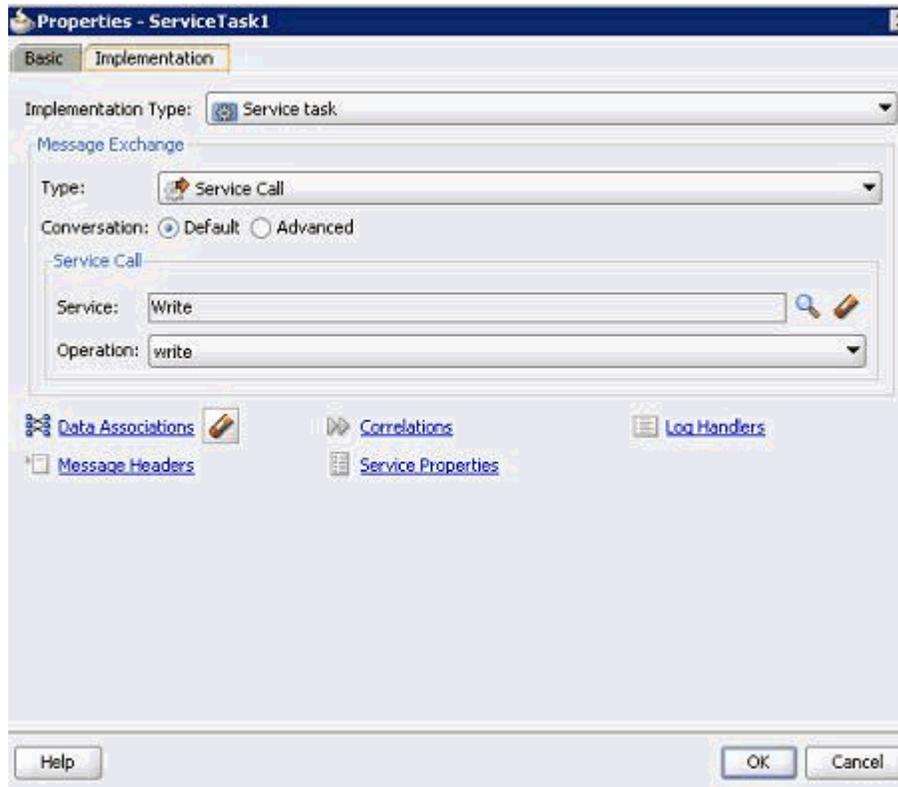
Figure 2-126 *Data Associations Window*



17. Click **OK**.

You are returned to the Properties - ServiceTask1 dialog, as shown in [Figure 2-127](#).

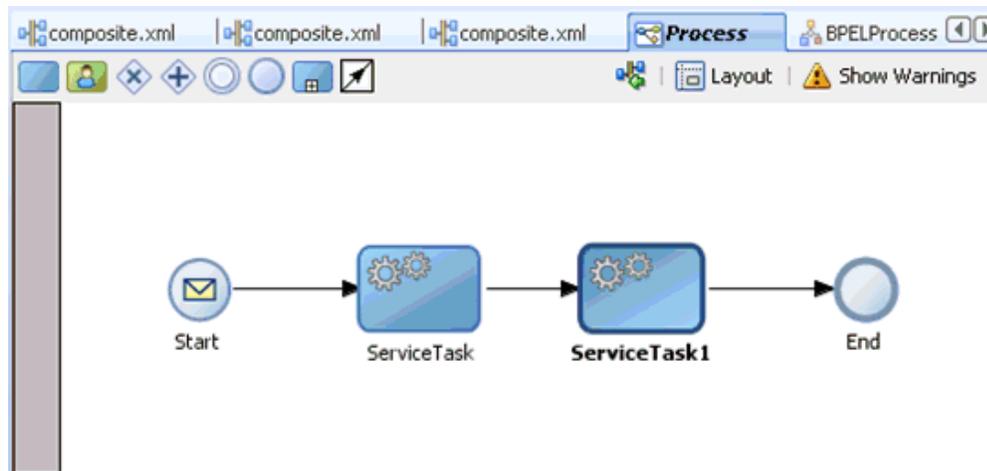
Figure 2–127 Properties - ServiceTask1 Dialog



18. Click OK.

You are returned to the Process tab, as shown in [Figure 2–128](#).

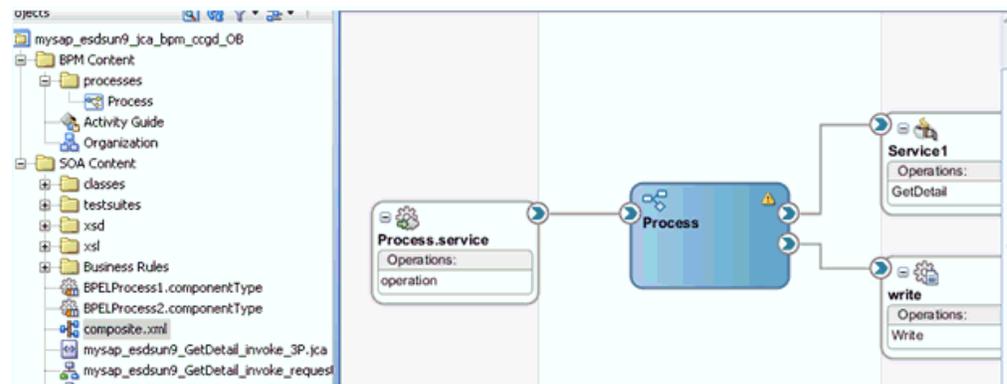
Figure 2–128 Process Tab



Notice that no errors or warnings are being indicated.

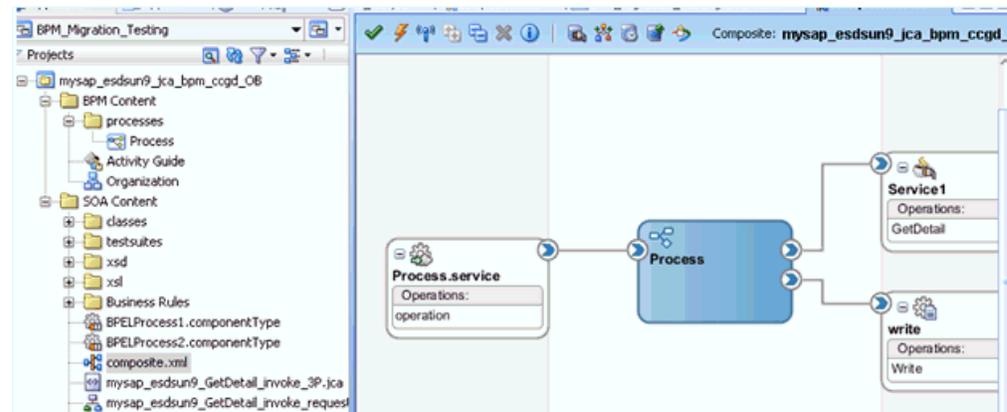
19. Save the process.
20. Double-click the **composite.xml** file.

Notice that the warning symbol is still indicated in the BPM Process component, as shown in [Figure 2–129](#).

Figure 2–129 BPM Process Component

- Restart Oracle JDeveloper.

Once Oracle JDeveloper has restarted, the warning symbol is not indicated in the BPM Process component, as shown in [Figure 2–130](#).

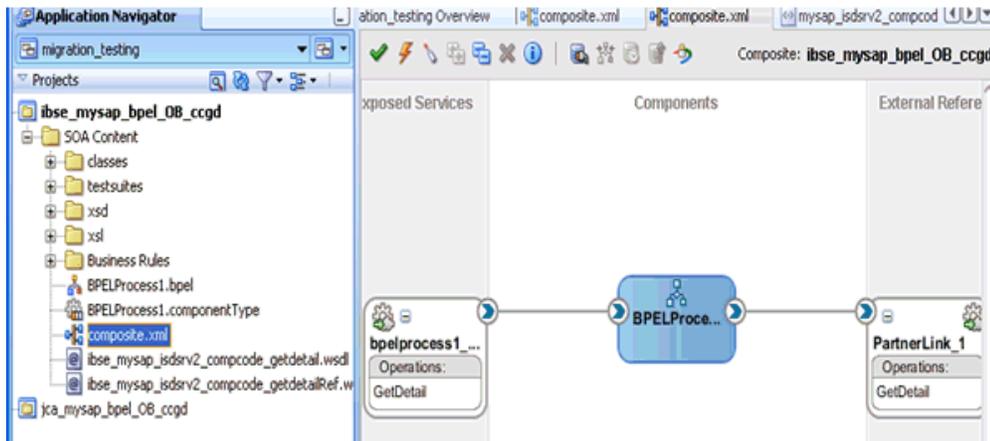
Figure 2–130 BPM Process Component

2.5.2 Additional Modifications for Migrated Processes in 11g PS6

Upgraded J2CA outbound and inbound processes in 11g PS6 function properly and do not require any additional modifications. Only BSE outbound processes require additional modifications, which are described in this section.

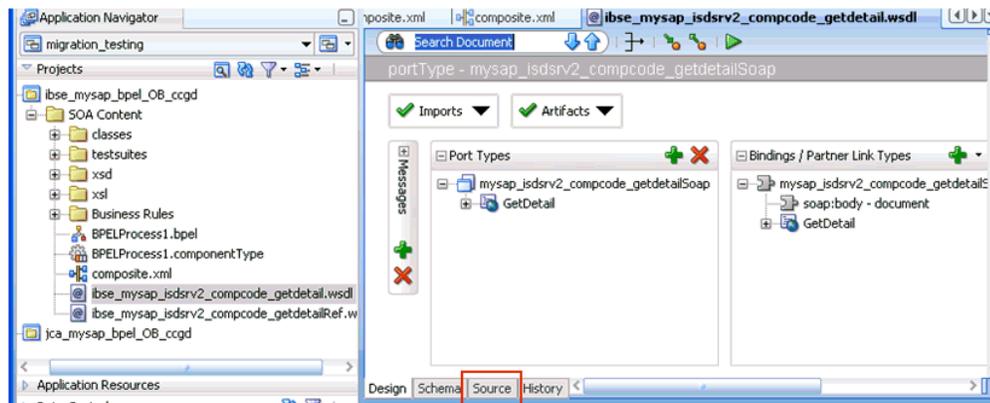
- Once the BSE outbound process is migrated successfully to 11g PS6, double-click the **composite.xml** file to open the migrated project, as shown in [Figure 2–131](#)

Figure 2–131 Opened Migrated Project



2. Double-click the BSE outbound WSDL file and then click the **Source** tab, as shown in Figure 2–132

Figure 2–132 BSE Outbound WSDL File



3. Change the `<soap:address location>` element to point to the system where 11g PS6 is running.

For example:

```
<service name="mysap_isdsrv2_compcode_getdetail">
  <documentation/>
  <port name="mysap_isdsrv2_compcode_getdetailSoap1" binding="tns:mysap_isdsrv2_compcode_getdetailSoap">
    <soap:address
      location="http://172.19.95.190:8001/ibse/IBSEServlet/XDSOAPRouter" />
  </port>
</service>
</definitions>
```

Note: The best option for both changes is to use localhost instead of an IP address, which eliminates the need for this change.

4. Save and deploy the process.

Oracle Application Adapter for SAP R/3 Upgrade Guidelines

This chapter lists and describes upgrade guidelines that are specific to the Oracle Application Adapter for SAP R/3. It contains the following topics:

- [Section 3.1, "Upgrading a 10.1.3.x ESB J2CA Outbound Process to 11g"](#)
- [Section 3.2, "Upgrading a 10.1.3.x ESB J2CA Inbound Process to 11g"](#)
- [Section 3.3, "Upgrading a 10.1.3.x BPEL J2CA Outbound Process to 11g"](#)
- [Section 3.4, "Upgrading a 10.1.3.x BPEL J2CA Inbound Process to 11g"](#)

3.1 Upgrading a 10.1.3.x ESB J2CA Outbound Process to 11g

Selecting a WSDL From the Local File System

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for SAP R/3 using Application Explorer and connect to the target.
2. Browse to a specific business object and generate an outbound WSDL for this object.
3. Restart the server.
4. Start Oracle JDeveloper 10.1.3.4 and create a JCA outbound ESB Project.
5. In the Custom Adapter service, select the outbound WSDL from the local file system.
6. Deploy the ESB project successfully and ensure the registration of the ESB project is successful.
7. On the 11g system, create a target using the same name that was provided on the 10.1.3.4 system.
8. Copy the deployed ESB project into the 11g system.
9. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 ESB project.
10. Once the project is migrated, a successful message is received in Oracle JDeveloper.
11. Expand the migrated project and double-click the composite.xml file to ensure that the project opens without any errors.
12. Save and then deploy the migrated ESB project.

13. Ensure that there are no error or warning messages during the deployment process.
14. Once the deployment is successful, navigate to the input folder and paste the input XML file.

The successful response XML is received in the specified output folder.

Selecting a WSDL Using Service Explorer

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for SAP R/3 using Application Explorer and connect to the target.
2. Browse to a specific business object and generate an outbound WSDL for this object.
3. Restart the server.
4. Start Oracle JDeveloper 10.1.3.4 and create a JCA outbound ESB Project.
5. In the Custom Adapter service, select the outbound WSDL using Service Explorer.
6. Deploy the ESB project successfully and ensure the registration of the ESB project is successful.
7. On the 11g system, create a target using the same name that was provided on the 10.1.3.4 system.
8. Copy the deployed ESB project into the 11g system.
9. Open the project folder and edit the **DefaultSystem_CustomAdapterServiceName.esbsvc** file (for example, `DefaultSystem_isdsrv2_cc_gd.esbsvc`) by providing the system IP address and port number (for example, 192.168.128.122:80) of the 10.1.3.4 WSDL generated system in the `<wsdlURL>` section that is located within `<serviceDefinition>`. For example:

Original

```
<serviceDefinition>
<wsdlURL>http://127.0.0.1:80/orainfra/wsdl/adapters/applications/isdsrv2_CC_
GetDetail_invoke.wsdl?wsdl</wsdlURL>
-----
</endpointDefinition>
</serviceDefinition>
```

Modified

```
<serviceDefinition>
<wsdlURL>http://192.168.128.122:80/orainfra/wsdl/adapters/applications/isdsrv2_
CC_GetDetail_invoke.wsdl?wsdl</wsdlURL>
-----
</endpointDefinition>
</serviceDefinition>
```

10. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 ESB project.
11. Once the project is migrated, a successful message is received in Oracle JDeveloper.
12. Expand the migrated project and double-click the `composite.xml` file to ensure that the project opens without any errors.
13. Save and then deploy the migrated ESB project.
14. Ensure that there are no error or warning messages during the deployment process.

15. Once the deployment is successful, navigate to the input folder and paste the input XML file.

The successful response XML is received in the specified output folder.

3.2 Upgrading a 10.1.3.x ESB J2CA Inbound Process to 11g

Selecting a WSDL From the Local File System

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for SAP R/3 using Application Explorer and connect to the target.
2. Create a new channel for Oracle Application Adapter for SAP R/3.
3. Browse to a specific business object and generate an inbound WSDL for this object.
4. Restart the server.
5. Start Oracle JDeveloper 10.1.3.4 and create a JCA inbound ESB Project.
6. In the Custom Adapter service, select the inbound WSDL from the local file system.
7. Deploy the ESB project successfully and ensure the registration of the ESB project is successful.
8. On the 11g system, create a target and channel using the same name that was provided on the 10.1.3.4 system.
9. Copy the deployed ESB project into the 11g system.
10. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 ESB project.
11. Once the project is migrated, a successful message is received in Oracle JDeveloper.
12. Expand the migrated project and double-click the composite.xml file to ensure that the project opens without any errors.
13. Double-click the J2CA properties file under the migrated project in Oracle JDeveloper and remove the following line:

```
<record-converter
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

For example:

```
<adapter-config name="MATMAS01" adapter="iWay ERP Adapter"
wsdlLocation="MATMAS01_receive.wsdl?wsdl"
xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">
<resource-adapter
className="com.iwaysoftware.afjca15.IWAFOracleResourceAdapter"/><record-converter
er className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

14. Save and then deploy the migrated ESB project.
15. Ensure that there are no error or warning messages during the deployment process.
16. Trigger the event messages from the ERP system (for example, SAP GUI) and verify that successful response XML files are received in the specified output folder.

Selecting a WSDL Using Service Explorer

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for SAP R/3 using Application Explorer and connect to the target.
2. Create a new channel for Oracle Application Adapter for SAP R/3.
3. Browse to a specific business object and generate an inbound WSDL for this object.
4. Restart the server.
5. Start Oracle JDeveloper 10.1.3.4 and create a JCA inbound ESB Project.
6. In the Custom Adapter service, select the inbound WSDL using Service Explorer.
7. Deploy the ESB project successfully and ensure the registration of the ESB project is successful.
8. On the 11g system, create a target and channel using the same name that was provided on the 10.1.3.4 system.
9. Copy the deployed ESB project into the 11g system.
10. Open the project folder and edit the files **DefaultSystem_CustomAdapterServiceName.esbsvc** (for example, DefaultSystem_matmas01_esb.esbsvc) and **DefaultSystem_CustomAdapterServiceName_RS.esbsvc** (for example, DefaultSystem_matmas01_esb_RS.esbsvc) by providing the system IP address and port number (for example, 192.168.128.122:80) of the 10.1.3.4 WSDL generated system in the <wsdlURL> sections. For example:

1. Original (DefaultSystem_matmas01_esb.esbsvc)

```
<interface>
<wsdlURL>http://127.0.0.1:80/orainfra/wsdl/adapters/applications/
MATMAS01_receive.wsdl?wsdl</wsdlURL>
-----
</interface>
```

Modified (DefaultSystem_matmas01_esb.esbsvc)

```
<interface>
<wsdlURL>http://192.168.128.122:80/orainfra/wsdl/adapters/applications/
MATMAS01_receive.wsdl?wsdl</wsdlURL>-----
</interface>
```

2. Original (DefaultSystem_matmas01_esb_RS.esbsvc)

```
<serviceDefinition>
<wsdlURL>http://127.0.0.1:80/orainfra/wsdl/adapters/applications/ MATMAS01_
receive.wsdl?wsdl</wsdlURL>
-----
</endpointDefinition>
</serviceDefinition>
```

Modified (DefaultSystem_matmas01_esb_RS.esbsvc)

```
<serviceDefinition>
<wsdlURL>http://192.168.128.122:80/orainfra/wsdl/adapters/applications/
MATMAS01_receive.wsdl?wsdl</wsdlURL>
-----
</endpointDefinition>
</serviceDefinition>
```

11. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 ESB project.

12. Once the project is migrated, a successful message is received in Oracle JDeveloper.
13. Expand the migrated project and double-click the composite.xml file to ensure that the project opens without any errors.
14. Double-click the J2CA properties file under the migrated project in Oracle JDeveloper and remove the following line:

```
<record-converter
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

For example:

```
<adapter-config name="MATMAS01" adapter="iWay ERP Adapter"
wsdlLocation="MATMAS01_receive.wsdl?wsdl"
xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">
<resource-adapter
className="com.iwaysoftware.afjca15.IWAF0racleResourceAdapter"/><record-convert
er className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

15. Save and then deploy the migrated ESB project.
16. Ensure that there are no error or warning messages during the deployment process.
17. Trigger the event messages from the ERP system (for example, SAP GUI) and verify that successful response XML files are received in the specified output folder.

3.3 Upgrading a 10.1.3.x BPEL J2CA Outbound Process to 11g

Selecting a WSDL From the Local File System

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for SAP R/3 using Application Explorer and connect to the target.
2. Browse to a specific business object and export the schemas into the default location.
3. Generate the outbound WSDL for the specific business object.
4. Restart the server.
5. Start Oracle JDeveloper 10.1.3.4 and create a JCA outbound BPEL Project by uploading the request and response schema generated from Application Explorer in step 2.
6. In the Partner Link, select the outbound WSDL from the local file system.
7. Deploy the BPEL project successfully and ensure that there are no error or warning messages during the deployment process.
8. On the 11g system, create a target using the same name that was provided on the 10.1.3.4 system.
9. Copy the deployed BPEL project into the 11g system.
10. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 BPEL project.
11. Once the project is migrated, a successful message is received in Oracle JDeveloper.

12. Expand the migrated project and double-click the composite.xml file to ensure that the project opens without any errors.
13. Save and then deploy the migrated BPEL project.
14. Ensure that there are no error or warning messages during the deployment process.
15. Open the Oracle Enterprise Manager console and invoke the input XML in XML / Tree view to receive a successful response XML.

Selecting a WSDL Using Service Explorer

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for SAP R/3 using Application Explorer and connect to the target.
2. Browse to a specific business object and export the schemas into the default location.
3. Generate the outbound WSDL for the specific business object.
4. Restart the server.
5. Start Oracle JDeveloper 10.1.3.4 and create a JCA outbound BPEL Project by uploading the request and response schema generated from Application Explorer in step 2.
6. In the Partner Link, select the outbound WSDL from Service Explorer.
7. Deploy the BPEL project successfully and ensure that there are no error or warning messages during the deployment process.
8. On the 11g system, create a target using the same name that was provided on the 10.1.3.4 system.
9. Copy the deployed BPEL project into the 11g system.
10. Edit the bpel.xml file for the WSDL location in the 10.1.3.4 project folder (Project_Name\bpel):

```
<partnerLinkBinding name="GetDetail">
<property
name="wsdlLocation">http://192.168.128.125:80/orainfra/wsdl/adapters/applicatio
ns/isdsrv2_CC_GetDetail_invoke.wsdl?wsdl</property>
</partnerLinkBinding>
```

Where the IP address and port number refer to the 10.1.3.4 system where the WSDL was generated.

11. Ensure that the 10.1.3.4 server is up and running.
12. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 BPEL project.
13. Once the project is migrated, a successful message is received in Oracle JDeveloper.
14. Expand the migrated project and double-click the composite.xml file to ensure that the project opens without any errors.
15. Save and then deploy the migrated BPEL project.
16. Ensure that there are no error or warning messages during the deployment process.
17. Open the Oracle Enterprise Manager console and invoke the input XML in XML / Tree view to receive a successful response XML.

3.4 Upgrading a 10.1.3.x BPEL J2CA Inbound Process to 11g

Selecting a WSDL From the Local File System

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for SAP R/3 using Application Explorer and connect to the target.
2. Create a new channel for Oracle Application Adapter for SAP R/3.
3. Browse to a specific business object.
4. Generate the inbound WSDL for the specific business object.
5. Restart the server.
6. Start Oracle JDeveloper 10.1.3.4 and create a JCA inbound BPEL Project by selecting the inbound WSDL from the local file system in the Partner Link
7. Deploy the BPEL project successfully and ensure that there are no error or warning messages during the deployment process.
8. Trigger the event messages from the SAP R/3 system and ensure that successful instances are received in the BPEL Console.
9. On the 11g system, create a target and channel using the same name that was provided on the 10.1.3.4 system.
10. Copy the deployed BPEL project into the 11g system.
11. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 BPEL project.
12. Once the project is migrated, a successful message is received in Oracle JDeveloper.
13. Expand the migrated project and double-click the composite.xml file to ensure that the project opens without any errors.
14. Double-click the J2CA properties file under the migrated project in Oracle JDeveloper and remove the following line:

```
<record-converter
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

For example:

```
<adapter-config name="MATMAS01" adapter="iWay ERP Adapter"
wsdlLocation="MATMAS01_receive.wsdl?wsdl"
xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">
<resource-adapter
className="com.iwaysoftware.afjca15.IWAFOracleResourceAdapter"/><record-converter
er className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

15. Save and then deploy the migrated BPEL project.
16. Ensure that there are no error or warning messages during the deployment process.
17. Trigger the event messages from the SAP R/3 system and ensure that successful instances are received in the Oracle Enterprise Manager console.

Selecting a WSDL Using Service Explorer

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for SAP R/3 using Application Explorer and connect to the target.
2. Create a new channel for Oracle Application Adapter for SAP R/3.

3. Browse to a specific business object.
4. Generate the inbound WSDL for the specific business object.
5. Restart the server.
6. Start Oracle JDeveloper 10.1.3.4 and create a JCA inbound BPEL Project by selecting the inbound WSDL from the Service Explorer in the Partner Link.
7. Deploy the BPEL project successfully and ensure that there are no error or warning messages during the deployment process.
8. Trigger the event messages from the SAP R/3 system and ensure that successful instances are received in the BPEL Console.
9. On the 11g system, create a target and channel using the same name that was provided on the 10.1.3.4 system.
10. Copy the deployed BPEL project into the 11g system.
11. Edit the `bpel.xml` file for the WSDL location in the 10.1.3.4 project folder (Project_Name\bpel):

```
<partnerLinkBinding name="MATMAS01">
<property
name="wsdlLocation">http://192.168.128.122:80/orainfra/wsdl/adapters/applicatio
ns/MATMAS01_receive.wsdl?wsdl</property>
</partnerLinkBinding>
```

Where the IP address and port number refer to the 10.1.3.4 system where the WSDL was generated.

12. Ensure that the 10.1.3.4 server is up and running.
13. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 BPEL project.
14. Once the project is migrated, a successful message is received in Oracle JDeveloper.
15. Expand the migrated project and double-click the `composite.xml` file to ensure that the project opens without any errors.
16. Double-click the J2CA properties file under the migrated project in Oracle JDeveloper and remove the following line:

```
<record-converter
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

For example:

```
<adapter-config name="MATMAS01" adapter="iWay ERP Adapter"
wsdlLocation="MATMAS01_receive.wsdl?wsdl"
xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">
<resource-adapter
className="com.iwaysoftware.afjca15.IWAF0racleResourceAdapter"/><record-convert
er className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

17. Save and then deploy the migrated BPEL project.
18. Ensure that there are no error or warning messages during the deployment process.
19. Trigger the event messages from the SAP R/3 system and ensure that successful instances are received in the Oracle Enterprise Manager console.

Oracle Application Adapter for Siebel Upgrade Guidelines

This chapter lists and describes upgrade guidelines that are specific to the Oracle Application Adapter for Siebel. It contains the following topics:

- [Section 4.1, "Upgrading a 10.1.3.x ESB J2CA Outbound Process to 11g"](#)
- [Section 4.2, "Upgrading a 10.1.3.x ESB J2CA Inbound Process to 11g"](#)
- [Section 4.3, "Upgrading a 10.1.3.x BPEL J2CA Outbound Process to 11g"](#)
- [Section 4.4, "Upgrading a 10.1.3.x BPEL J2CA Inbound Process to 11g"](#)

4.1 Upgrading a 10.1.3.x ESB J2CA Outbound Process to 11g

Selecting a WSDL From the Local File System

You can follow the same procedure in ["Selecting a WSDL From the Local File System"](#) on page 3-1.

Selecting a WSDL Using Service Explorer

You can follow the same procedure in ["Selecting a WSDL Using Service Explorer"](#) on page 3-2.

4.2 Upgrading a 10.1.3.x ESB J2CA Inbound Process to 11g

Selecting a WSDL From the Local File System

This section applies to **Siebel 7.7** systems.

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for Siebel using Application Explorer and connect to the target.
2. Browse to a specific business object.
3. Create an Integration Object node by selecting a Siebel generated schema file (.xsd) using Application Explorer.
4. Create a new HTTP channel for Oracle Application Adapter for Siebel.
5. Generate an inbound WSDL for this object.
6. Modify the created 10.1.3.4 WSDL by replacing **www.siebel.com** with **www.iwaysoftware.com** in the three lines as shown in the following example:

Original 10.1.3.4 WSDL

```

<definitions name="samp_node"
targetNamespace="http://xmlns.oracle.com/pcbpel/iWay/wsd1/Siebel/isdsrv22/samp_
node"
xmlns:iWayEvent="http://www.siebel.com/xml/Sample%20Account"
-----
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.siebel.com/xml/Sample%20Account"
attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xsdLocal="http://www.siebel.com/xml/Sample%20Account">
<xsd:element type="xsdLocal:SiebelMessage" name="SiebelMessage"/>
<xsd:complexType name="SiebelMessage">

```

Modified 10.1.3.4 WSDL

```

<definitions name="samp_node"
targetNamespace="http://xmlns.oracle.com/pcbpel/iWay/wsd1/Siebel/isdsrv22/samp_
node"
xmlns:iWayEvent="http://www.iwaysoftware.com/xml/Sample%20Account"
-----
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.iwaysoftware.com/xml/Sample%20Account"
xmlns:xsdLocal="http://www.iwaysoftware.com/xml/Sample%20Account"
attributeFormDefault="unqualified" elementFormDefault="qualified">
<xsd:element name="SiebelMessage" type="xsdLocal:SiebelMessage"/>

```

7. Restart the server.
8. Start Oracle JDeveloper 10.1.3.4 and create a JCA inbound ESB Project.
9. In the Custom Adapter service, select the inbound WSDL from the local file system.
10. Deploy the ESB project successfully and ensure the registration of the ESB project is successful.
11. On the 11g system, ensure the target, Integration Object node, and channel are created, and named the same as on the 10.1.3.4 system. While configuring the channel, in the PreParser tab, provide the event schema location generated from Application Explorer (right-click the Integration Object node and export the schema in 11g).
12. Copy the deployed ESB project into the 11g system.
13. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 ESB project.
14. Once the project is migrated, a successful message is received in Oracle JDeveloper.
15. Expand the migrated project and double-click the composite.xml file to ensure that the project opens without any errors.
16. Double-click the J2CA properties file under the migrated project in Oracle JDeveloper and remove the following line:

```

<record-converter
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>

```

For example:

```

<adapter-config name="samp_node" adapter="iWay ERP Adapter"
wsdlLocation="isdsrv22_sampleAcct_receive.wsdl"
xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">
<resource-adapter
className="com.iwaysoftware.afjca15.IWAFOracleResourceAdapter"/>

```

```
<record-converter
  className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

17. Save and then deploy the migrated ESB project.
18. Ensure that there are no error or warning messages during the deployment process.
19. Trigger the event messages from the Siebel system and verify that successful response XML files are received in the specified output folder.

This section applies to **Siebel 7.0** systems.

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for Siebel using Application Explorer and connect to the target.
2. Browse to a specific business object.
3. Create an Integration Object node by selecting a Siebel generated schema file (.xdr) using Application Explorer.
4. Create a new HTTP channel for Oracle Application Adapter for Siebel.
5. Generate an inbound WSDL for this object.
6. Modify the created 10.1.3.4 WSDL by replacing **urn:iwaysoftware:adapter:siebel:oct2004:request** with **http://www.iwaysoftware.com/xml/Sample%20Account**, where **Sample%20Account** is the name of the specific Integration Object that is displayed in Application Explorer, in the following three lines:

Original 10.1.3.4 WSDL

```
<definitions name="samp_node"
targetNamespace="http://xmlns.oracle.com/pcbpel/iWay/wsd1/Siebel/ariba01/samp_
node"
xmlns:iWayEvent="urn:iwaysoftware:adapter:siebel:oct2004:request"
-----
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:iwaysoftware:adapter:siebel:oct2004:request"
elementFormDefault="qualified"
xmlns:sbl="urn:iwaysoftware:adapter:siebel:oct2004:request">
<xsd:element name="SiebelMessage">
```

Modified 10.1.3.4 WSDL

```
<definitions name="samp_node"
targetNamespace="http://xmlns.oracle.com/pcbpel/iWay/wsd1/Siebel/ariba01/samp_
node"
xmlns:iWayEvent="http://www.iwaysoftware.com/xml/Sample%20Account"
-----
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.iwaysoftware.com/xml/Sample%20Account"
xmlns:xsdLocal="http://www.iwaysoftware.com/xml/Sample%20Account"
attributeFormDefault="unqualified" elementFormDefault="qualified">
<xsd:element name="SiebelMessage" type="xsdLocal:SiebelMessage"/>
```

7. Restart the server.
8. Start Oracle JDeveloper 10.1.3.4 and create a JCA inbound ESB Project.
9. In the Custom Adapter service, select the inbound WSDL from the local file system.

10. Deploy the ESB project successfully and ensure the registration of the ESB project is successful.
11. On the 11g system, ensure the target, Integration Object node, and channel are created, and named the same as on the 10.1.3.4 system. While configuring the channel, in the PreParser tab, provide the event schema location generated from Application Explorer (right-click the Integration Object node and export the schema in 11g).
12. Copy the deployed ESB project into the 11g system.
13. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 ESB project.
14. Once the project is migrated, a successful message is received in Oracle JDeveloper.
15. Expand the migrated project and double-click the composite.xml file to ensure that the project opens without any errors.
16. Double-click the J2CA properties file under the migrated project in Oracle JDeveloper and remove the following line:

```
<record-converter  
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

For example:

```
<adapter-config name="samp_node" adapter="iWay ERP Adapter"  
wsdlLocation="isdsrv22_sampleAcct_receive.wsdl"  
xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">  
<resource-adapter  
className="com.iwaysoftware.afjca15.IWAFOracleResourceAdapter"/>  
  <record-converter  
  className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

17. Save and then deploy the migrated ESB project.
18. Ensure that there are no error or warning messages during the deployment process.
19. Trigger the event messages from the Siebel system and verify that successful response XML files are received in the specified output folder.

Selecting a WSDL Using Service Explorer

This section applies to Siebel 7.7 systems.

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for Siebel using Application Explorer and connect to the target.
2. Browse to a specific business object.
3. Create an Integration Object node by selecting a Siebel generated schema file (.xsd) using Application Explorer.
4. Create a new HTTP channel for Oracle Application Adapter for Siebel.
5. Generate an inbound WSDL for this object.
6. Modify the created 10.1.3.4 WSDL by replacing **www.siebel.com** with **www.iwaysoftware.com** in the three lines as shown in the following example:

Original 10.1.3.4 WSDL

```
<definitions name="samp_node"  
targetNamespace="http://xmlns.oracle.com/pcbepel/iWay/wsdl/Siebel/isdsrv22/samp_
```

```

node"
xmlns:iWayEvent="http://www.siebel.com/xml/Sample%20Account"
-----
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.siebel.com/xml/Sample%20Account"
attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xsdLocal="http://www.siebel.com/xml/Sample%20Account">
<xsd:element type="xsdLocal:SiebelMessage" name="SiebelMessage"/>
<xsd:complexType name="SiebelMessage">

```

Modified 10.1.3.4 WSDL

```

<definitions name="samp_node"
targetNamespace="http://xmlns.oracle.com/pcbpel/iWay/wsd/Siebel/isdsrv22/samp_
node"
xmlns:iWayEvent="http://www.iwaysoftware.com/xml/Sample%20Account"
-----
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.iwaysoftware.com/xml/Sample%20Account"
xmlns:xsdLocal="http://www.iwaysoftware.com/xml/Sample%20Account"
attributeFormDefault="unqualified" elementFormDefault="qualified">
<xsd:element name="SiebelMessage" type="xsdLocal:SiebelMessage"/>

```

7. Restart the server.
8. Start Oracle JDeveloper 10.1.3.4 and create a JCA inbound ESB Project.
9. In the Custom Adapter service, select the inbound WSDL using Service Explorer.
10. Deploy the ESB project successfully and ensure the registration of the ESB project is successful.
11. On the 11g system, ensure the target, Integration Object node, and channel are created, and named the same as on the 10.1.3.4 system. While configuring the channel, in the PreParser tab, provide the event schema location generated from Application Explorer (right-click the Integration Object node and export the schema in 11g).
12. Copy the deployed ESB project into the 11g system.
13. Open the project folder and edit the files **DefaultSystem_CustomAdapterServiceName.esbsvc** (for example, **DefaultSystem_SA.esbsvc**) and **DefaultSystem_CustomAdapterServiceName_RS.esbsvc** (for example, **DefaultSystem_SA_RS.esbsvc**) by providing the system IP address and port number (for example, 192.168.128.122:80) of the 10.1.3.4 WSDL generated system in the <wsdlURL> sections. For example:

1. Original (DefaultSystem_SA.esbsvc)

```

<interface> <wsdlURL>http://127.0.0.1:80/orainfra/wsdl/adapters/applications/
isdsrv22_SA_receive.wsdl?wsdl</wsdlURL>
-----
</interface>

```

Modified (DefaultSystem_SA.esbsvc)

```

<interface>
<wsdlURL>http://192.168.128.122:80/orainfra/wsdl/adapters/applications/
isdsrv22_SA_receive.wsdl?wsdl</wsdlURL>
-----
</interface>

```

2. Original (DefaultSystem_SA_RS.esbsvc)

```
<serviceDefinition>
<wsdlURL>http://127.0.0.1:80/orainfra/wsdl/adapters/applications/ isdsrv22_SA_
receive.wsdl?wsdl</wsdlURL>
-----
</endpointDefinition>
</serviceDefinition>
```

Modified (DefaultSystem_SA_RS.esbsvc)

```
<serviceDefinition>
<wsdlURL>http://192.168.128.122:80/orainfra/wsdl/adapters/applications/
isdsrv22_SA_receive.wsdl?wsdl</wsdlURL>
-----
</endpointDefinition>
</serviceDefinition>
```

14. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 ESB project.
15. Once the project is migrated, a successful message is received in Oracle JDeveloper.
16. Expand the migrated project and double-click the composite.xml file to ensure that the project opens without any errors.
17. Double-click the J2CA properties file under the migrated project in Oracle JDeveloper and remove the following line:

```
<record-converter
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl" />
```

For example:

```
<adapter-config name="samp_node" adapter="iWay ERP Adapter"
wsdlLocation="isdsrv22_sampleAcct_receive.wsdl"
xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">
<resource-adapter
className="com.iwaysoftware.afjca15.IWAFOracleResourceAdapter" />
<record-converter
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl" />
```

18. Save and then deploy the migrated ESB project.
19. Ensure that there are no error or warning messages during the deployment process.
20. Trigger the event messages from the Siebel system and verify that successful response XML files are received in the specified output folder.

This section applies to **Siebel 7.0** systems.

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for Siebel using Application Explorer and connect to the target.
2. Browse to a specific business object.
3. Create an Integration Object node by selecting a Siebel generated schema file (.xdr) using Application Explorer.
4. Create a new HTTP channel for Oracle Application Adapter for Siebel.
5. Generate an inbound WSDL for this object.
6. Modify the created 10.1.3.4 WSDL by replacing **urn:iwaysoftware:adapter:siebel:oct2004:request** with **http://www.iwaysoftware.com/xml/Sample%20Account**, where

Sample%20Account is the name of the specific Integration Object that is displayed in Application Explorer, in the following three lines:

Original 10.1.3.4 WSDL

```
<definitions name="samp_node"
targetNamespace="http://xmlns.oracle.com/pcbpel/iWay/wsd1/Siebel/ariba01/samp_
node"
xmlns:iWayEvent="urn:iwaysoftware:adapter:siebel:oct2004:request"
-----
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:iwaysoftware:adapter:siebel:oct2004:request"
elementFormDefault="qualified"
xmlns:sbl="urn:iwaysoftware:adapter:siebel:oct2004:request">
<xsd:element name="SiebelMessage">
```

Modified 10.1.3.4 WSDL

```
<definitions name="samp_node"
targetNamespace="http://xmlns.oracle.com/pcbpel/iWay/wsd1/Siebel/ariba01/samp_
node"
xmlns:iWayEvent="http://www.iwaysoftware.com/xml/Sample%20Account"
-----
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.iwaysoftware.com/xml/Sample%20Account"
xmlns:xsdLocal="http://www.iwaysoftware.com/xml/Sample%20Account"
attributeFormDefault="unqualified" elementFormDefault="qualified">
<xsd:element name="SiebelMessage" type="xsdLocal:SiebelMessage"/>
```

7. Restart the server.
8. Start Oracle JDeveloper 10.1.3.4 and create a JCA inbound ESB Project.
9. In the Custom Adapter service, select the inbound WSDL using Service Explorer.
10. Deploy the ESB project successfully and ensure the registration of the ESB project is successful.
11. On the 11g system, ensure the target, Integration Object node, and channel are created, and named the same as on the 10.1.3.4 system. While configuring the channel, in the PreParser tab, provide the event schema location generated from Application Explorer (right-click the Integration Object node and export the schema in 11g).
12. Copy the deployed ESB project into the 11g system.
13. Open the project folder and edit the files **DefaultSystem_CustomAdapterServiceName.esbsvc** (for example, **DefaultSystem_SA.esbsvc**) and **DefaultSystem_CustomAdapterServiceName_RS.esbsvc** (for example, **DefaultSystem_SA_RS.esbsvc**) by providing the system IP address and port number (for example, 192.168.128.122:80) of the 10.1.3.4 WSDL generated system in the `<wsdlURL>` sections. For example:

1. Original (DefaultSystem_SA.esbsvc)

```
<interface> <wsdlURL>http://127.0.0.1:80/orainfra/wsdl/adapters/applications/
idsrv22_SA_receive.wsdl?wsdl</wsdlURL>
-----
</interface>
```

Modified (DefaultSystem_SA.esbsvc)

```
<interface>
<wsdlURL>http://192.168.128.122:80/orainfra/wsdl/adapters/applications/
```

```
isdsrv22_SA_receive.wsdl?wsdl</wsdlURL>
-----
</interface>
```

2. Original (DefaultSystem_SA_RS.esbsvc)

```
<serviceDefinition>
<wsdlURL>http://127.0.0.1:80/orainfra/wsdl/adapters/applications/ isdsrv22_SA_
receive.wsdl?wsdl</wsdlURL>
-----
</endpointDefinition>
</serviceDefinition>
```

Modified (DefaultSystem_SA_RS.esbsvc)

```
<serviceDefinition>
<wsdlURL>http://192.168.128.122:80/orainfra/wsdl/adapters/applications/
isdsrv22_SA_receive.wsdl?wsdl</wsdlURL>
-----
</endpointDefinition>
</serviceDefinition>
```

14. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 ESB project.
15. Once the project is migrated, a successful message is received in Oracle JDeveloper.
16. Expand the migrated project and double-click the composite.xml file to ensure that the project opens without any errors.
17. Double-click the J2CA properties file under the migrated project in Oracle JDeveloper and remove the following line:

```
<record-converter
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl" />
```

For example:

```
<adapter-config name="samp_node" adapter="iWay ERP Adapter"
wsdlLocation="ariba01_sampleAcct_receive.wsdl"
xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">
<resource-adapter
className="com.iwaysoftware.afjca15.IWAFOracleResourceAdapter" />
  <record-converter
    className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl" />
```

18. Save and then deploy the migrated ESB project.
19. Ensure that there are no error or warning messages during the deployment process.
20. Trigger the event messages from the Siebel system and verify that successful response XML files are received in the specified output folder.

4.3 Upgrading a 10.1.3.x BPEL J2CA Outbound Process to 11g

Selecting a WSDL From the Local File System

You can follow the same procedure in ["Selecting a WSDL From the Local File System"](#) on page 3-5.

Selecting a WSDL Using Service Explorer

You can follow the same procedure in ["Selecting a WSDL Using Service Explorer"](#) on page 3-6.

4.4 Upgrading a 10.1.3.x BPEL J2CA Inbound Process to 11g

Selecting a WSDL From the Local File System

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for Siebel using Application Explorer and connect to the target.
2. Browse to the particular business object. Create an Integration Object node by selecting the Siebel generated XSD/XDR file using Application Explorer.
3. Create a new HTTP channel for Oracle Application Adapter for Siebel.
4. Generate the inbound WSDL for the specific business object.
5. Restart the server.
6. Start Oracle JDeveloper 10.1.3.4 and create a JCA inbound BPEL Project by selecting the inbound WSDL from the local file system in the Partner Link
7. Deploy the BPEL project successfully and ensure that there are no error or warning messages during the deployment process.
8. Trigger the event messages from the Siebel system and ensure that successful instances are received in the BPEL Console.
9. On the 11g system, ensure the target, Integration Object node, and channel are created, and named the same as on the 10.1.3.4 system. While configuring the channel, in the PreParser tab, provide the event schema location generated from Application Explorer (right-click the Integration Object node and export the schema in 11g).
10. Copy the deployed BPEL project into the 11g system.
11. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 BPEL project.
12. Once the project is migrated, a successful message is received in Oracle JDeveloper.
13. Expand the migrated project and double-click the composite.xml file to ensure that the project opens without any errors.
14. Double-click the J2CA properties file under the migrated project in Oracle JDeveloper and remove the following line:

```
<record-converter
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

For example:

```
<adapter-config name="samp_node" adapter="iWay ERP Adapter"
wsdlLocation="isdsrv22_sampleAcct_receive.wsdl"
xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">
<resource-adapter
className="com.iwaysoftware.afjca15.IWAFOracleResourceAdapter"/>
<record-converter
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

15. Save and then deploy the migrated BPEL project.

16. Ensure that there are no error or warning messages during the deployment process.
17. Trigger the event messages from the Siebel system and ensure that successful instances are received in the Oracle Enterprise Manager console.

Selecting a WSDL Using Service Explorer

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for Siebel using Application Explorer and connect to the target.
2. Browse to the particular business object. Create an Integration Object node by selecting the Siebel generated XSD/XDR file using Application Explorer.
3. Create a new HTTP channel for Oracle Application Adapter for Siebel.
4. Generate the inbound WSDL for the specific business object.
5. Restart the server.
6. Start Oracle JDeveloper 10.1.3.4 and create a JCA inbound BPEL Project by selecting the inbound WSDL from the Service Explorer in the Partner Link
7. Deploy the BPEL project successfully and ensure that there are no error or warning messages during the deployment process.
8. Trigger the event messages from the Siebel system and ensure that successful instances are received in the BPEL Console.
9. On the 11g system, ensure the target, Integration Object node, and channel are created, and named the same as on the 10.1.3.4 system. While configuring the channel, in the PreParser tab, provide the event schema location generated from Application Explorer (right-click the Integration Object node and export the schema in 11g).
10. Copy the deployed BPEL project into the 11g system.
11. Edit the **bpel.xml** file for the WSDL location in the 10.1.3.4 project folder (Project_Name\bpel):

```
<partnerLinkBinding name="samp_node">
<property
name="wsdlLocation">http://192.168.128.125:80/orainfra/wsdl/adapters/applicatio
ns/isdsrv22_sampleAcct_receive.wsdl?wsdl</property>
</partnerLinkBinding>
```

Where the IP address and port number refer to the 10.1.3.4 system where the WSDL was generated.

12. Ensure that the 10.1.3.4 server is up and running.
13. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 BPEL project.
14. Once the project is migrated, a successful message is received in Oracle JDeveloper.
15. Expand the migrated project and double-click the composite.xml file to ensure that the project opens without any errors.
16. Double-click the J2CA properties file under the migrated project in Oracle JDeveloper and remove the following line:

```
<record-converter
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

For example:

```
<adapter-config name="samp_node" adapter="iWay ERP Adapter"
wsdlLocation="isdsrv22_sampleAcct_receive.wsdl"
xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">
<resource-adapter
className="com.iwaysoftware.afjca15.IWAFOracleResourceAdapter"/>
  <record-converter
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

17. Save and then deploy the migrated BPEL project.
18. Ensure that there are no error or warning messages during the deployment process.
19. Trigger the event messages from the Siebel system and ensure that successful instances are received in the Oracle Enterprise Manager console.

Oracle Application Adapter for PeopleSoft Upgrade Guidelines

This chapter lists and describes upgrade guidelines that are specific to the Oracle Application Adapter for PeopleSoft. It contains the following topics:

- [Section 5.1, "Upgrading a 10.1.3.x ESB J2CA Outbound Process to 11g"](#)
- [Section 5.2, "Upgrading a 10.1.3.x ESB J2CA Inbound Process to 11g"](#)
- [Section 5.3, "Upgrading a 10.1.3.x BPEL J2CA Outbound Process to 11g"](#)
- [Section 5.4, "Upgrading a 10.1.3.x BPEL J2CA Inbound Process to 11g"](#)

5.1 Upgrading a 10.1.3.x ESB J2CA Outbound Process to 11g

Selecting a WSDL From the Local File System

You can follow the same procedure in ["Selecting a WSDL From the Local File System"](#) on page 3-1.

Selecting a WSDL Using Service Explorer

You can follow the same procedure in ["Selecting a WSDL Using Service Explorer"](#) on page 3-2.

5.2 Upgrading a 10.1.3.x ESB J2CA Inbound Process to 11g

Selecting a WSDL From the Local File System

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for PeopleSoft using Application Explorer and connect to the target.
2. Create a new channel for Oracle Application Adapter for PeopleSoft.
3. Browse to a specific business object and generate an inbound WSDL for this object.
4. Restart the server.
5. Start Oracle JDeveloper 10.1.3.4 and create a JCA inbound ESB Project.
6. In the Custom Adapter service, select the inbound WSDL from the local file system.
7. Deploy the ESB project successfully and ensure the registration of the ESB project is successful.

8. On the 11g system, create a target and channel using the same name that was provided on the 10.1.3.4 system.
9. Copy the deployed ESB project into the 11g system.
10. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 ESB project.
11. Once the project is migrated, a successful message is received in Oracle JDeveloper.
12. Expand the migrated project and double-click the composite.xml file to ensure that the project opens without any errors.
13. Double-click the J2CA properties file under the migrated project in Oracle JDeveloper and remove the following line:

```
<record-converter
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

For example:

```
<adapter-config name="DEPT_SYNC_VERSION_1" adapter="iWay ERP Adapter"
wsdlLocation="isdsrv14_DEPT_SYNC_receive.wsdl"
xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">
<resource-adapter
className="com.iwaysoftware.afjca15.IWAF0racleResourceAdapter"/><record-convert
er className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

14. Save and then deploy the migrated ESB project.
15. Ensure that there are no error or warning messages during the deployment process.
16. Trigger the event messages from the ERP system (for example, PeopleTools) and verify that successful response XML files are received in the specified output folder.

Selecting a WSDL Using Service Explorer

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for PeopleSoft using Application Explorer and connect to the target.
2. Create a new channel for Oracle Application Adapter for PeopleSoft.
3. Browse to a specific business object and generate an inbound WSDL for this object.
4. Restart the server.
5. Start Oracle JDeveloper 10.1.3.4 and create a JCA inbound ESB Project.
6. In the Custom Adapter service, select the inbound WSDL using Service Explorer.
7. Deploy the ESB project successfully and ensure the registration of the ESB project is successful.
8. On the 11g system, create a target and channel using the same name that was provided on the 10.1.3.4 system.
9. Copy the deployed ESB project into the 11g system.
10. Open the project folder and edit the **DefaultSystem_CustomAdapterServiceName.esbsvc** file (for example, DefaultSystem_dept_sync_wsdl_brows.esbsvc) and **DefaultSystem_CustomAdapterServiceName_RS.esbsvc** file (for example, DefaultSystem_dept_sync_wsdl_brows_RS.esbsvc) by providing the system IP address and port number (for example,

192.168.128.122:80) of the 10.1.3.4 WSDL generated system in the <wsdlURL> sections. For example:

1. Original (DefaultSystem_dept_sync_wsdl_brows.esbsvc)

```
<interface> <wsdlURL>http://127.0.0.1:80/orainfra/wsdl/adapters/applications/
isdsrv14_DEPT_SYNC_receive_esb.wsdl?wsdl</wsdlURL>
-----
</interface>
```

Modified (DefaultSystem_dept_sync_wsdl_brows.esbsvc)

```
<interface>
<wsdlURL>http://192.168.128.122:80/orainfra/wsdl/adapters/applications/
isdsrv14_DEPT_SYNC_receive_esb.wsdl?wsdl</wsdlURL>
-----
</interface>
```

2. Original (DefaultSystem_dept_sync_wsdl_brows_RS.esbsvc)

```
<serviceDefinition>
<wsdlURL>http://127.0.0.1:80/orainfra/wsdl/adapters/applications/ isdsrv14_
DEPT_SYNC_receive_esb.wsdl?wsdl</wsdlURL>
-----
</endpointDefinition>
</serviceDefinition>
```

Modified (DefaultSystem_dept_sync_wsdl_brows_RS.esbsvc)

```
<serviceDefinition>
<wsdlURL>http://192.168.128.122:80/orainfra/wsdl/adapters/applications/
isdsrv14_DEPT_SYNC_receive_esb.wsdl?wsdl</wsdlURL>
-----
</endpointDefinition>
</serviceDefinition>
```

11. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 ESB project.
12. Once the project is migrated, a successful message is received in Oracle JDeveloper.
13. Expand the migrated project and double-click the composite.xml file to ensure that the project opens without any errors.
14. Double-click the J2CA properties file under the migrated project in Oracle JDeveloper and remove the following line:

```
<record-converter
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

For example:

```
<adapter-config name="DEPT_SYNC_VERSION_1" adapter="iWay ERP Adapter"
wsdlLocation="isdsrv14_DEPT_SYNC_receive.wsdl"
xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">
<resource-adapter
className="com.iwaysoftware.afjca15.IWAFOracleResourceAdapter"/><record-converter
er className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

15. Save and then deploy the migrated ESB project.
16. Ensure that there are no error or warning messages during the deployment process.

17. Trigger the event messages from the ERP system (for example, PeopleTools) and verify that successful response XML files are received in the specified output folder.

5.3 Upgrading a 10.1.3.x BPEL J2CA Outbound Process to 11g

Selecting a WSDL From the Local File System

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for PeopleSoft using Application Explorer and connect to the target.
2. Browse to a specific business object and export the schemas into the default location.
3. Generate the outbound WSDL for the specific business object.
4. Modify the request and response schema that was exported from Application Explorer by changing the **elementFormDefault** value from **unqualified** to **qualified**.
5. Restart the server.
6. Start Oracle JDeveloper 10.1.3.4 and create a JCA outbound BPEL Project by uploading the request and response schema that is modified in step 4.
7. In the Partner Link, select the outbound WSDL from the local file system.
8. Deploy the BPEL project successfully and ensure that there are no error or warning messages during the deployment process.
9. On the 11g system, create a target using the same name that was provided on the 10.1.3.4 system.
10. Copy the deployed BPEL project into the 11g system.
11. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 BPEL project.
12. Once the project is migrated, a successful message is received in Oracle JDeveloper.
13. Expand the migrated project and double-click the composite.xml file to ensure that the project opens without any errors.
14. Save and then deploy the migrated BPEL project.
15. Ensure that there are no error or warning messages during the deployment process.
16. Open the Oracle Enterprise Manager console and invoke the input XML in XML / Tree view to receive a successful response XML.

Selecting a WSDL Using Service Explorer

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for PeopleSoft using Application Explorer and connect to the target.
2. Browse to a specific business object and export the schemas into the default location.
3. Generate the outbound WSDL for the specific business object.
4. Modify the request and response schema that was exported from Application Explorer by changing the **elementFormDefault** value from **unqualified** to **qualified**.

5. Restart the server.
6. Start Oracle JDeveloper 10.1.3.4 and create a JCA outbound BPEL Project by uploading the request and response schema that is modified in step 4.
7. In the Partner Link, select the outbound WSDL from Service Explorer.
8. Deploy the BPEL project successfully and ensure that there are no error or warning messages during the deployment process.
9. On the 11g system, create a target using the same name that was provided on the 10.1.3.4 system.
10. Copy the deployed BPEL project into the 11g system.
11. Edit the bpel.xml file for the WSDL location in the 10.1.3.4 project folder (Project_Name\bpel):

```
<partnerLinkBinding name="DEPT">
  <property
name="wsdlLocation">http://192.168.128.125:80/orainfra/wsdl/adapters/applicatio
ns/isdsrv14_DEPT_invoke.wsdl?wsdl </property>
</partnerLinkBinding>
```

Where the IP address and port number refer to the 10.1.3.4 system where the WSDL was generated.

12. Ensure that the 10.1.3.4 server is up and running.
13. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 BPEL project.
14. Once the project is migrated, a successful message is received in Oracle JDeveloper.
15. Expand the migrated project and double-click the composite.xml file to ensure that the project opens without any errors.
16. Save and then deploy the migrated BPEL project.
17. Ensure that there are no error or warning messages during the deployment process.
18. Open the Oracle Enterprise Manager console and invoke the input XML in XML / Tree view to receive a successful response XML.

5.4 Upgrading a 10.1.3.x BPEL J2CA Inbound Process to 11g

Selecting a WSDL From the Local File System

You can follow the same procedure in ["Selecting a WSDL From the Local File System"](#) on page 3-7.

Selecting a WSDL Using Service Explorer

You can follow the same procedure in ["Selecting a WSDL Using Service Explorer"](#) on page 3-7.

Oracle Application Adapter for J.D. Edwards OneWorld Upgrade Guidelines

This chapter lists and describes upgrade guidelines that are specific to the Oracle Application Adapter for J.D. Edwards OneWorld. It contains the following topics:

- [Section 6.1, "Upgrading a 10.1.3.x ESB J2CA Outbound Process to 11g"](#)
- [Section 6.2, "Upgrading a 10.1.3.x ESB J2CA Inbound Process to 11g"](#)
- [Section 6.3, "Upgrading a 10.1.3.x BPEL J2CA Outbound Process to 11g"](#)
- [Section 6.4, "Upgrading a 10.1.3.x BPEL J2CA Inbound Process to 11g"](#)

6.1 Upgrading a 10.1.3.x ESB J2CA Outbound Process to 11g

Selecting a WSDL From the Local File System

This workaround must be performed for a J.D. Edwards outbound ESB process when migrating from 10.1.3.x to PS2/PS3/PS4/PS5/PS6. If you are migrating 10.1.3.x to PS1, then you can follow the same procedure in ["Selecting a WSDL From the Local File System"](#) on page 3-1.

1. On the 10.1.3.4 system, create a new target for the Oracle Application Adapter for J.D. Edwards OneWorld using Application Explorer and connect to the target.
2. Browse to a specific business object and generate an outbound WSDL file for this object.
3. Restart the server.
4. Start Oracle JDeveloper 10.1.3.4 and create a JCA outbound ESB project.
5. In the Custom Adapter service, select the outbound WSDL file from the local file system.
6. Deploy the ESB project successfully and ensure that the registration of the ESB project is successful.
7. On the 11g system, create a target using the same name that was provided on the 10.1.3.4 system.
8. Copy the deployed ESB project into the 11g system
9. Open the 10.1.3.4 project and edit the associated WSDL file with the following changes:

Change 1

Add the following line in the `<definitions>` section for `iWayResponse`:

```
xmlns:iWayResponse="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone
.response"
```

For example:

```
<definitions name="GetPhone"
targetNamespace="http://xmlns.oracle.com/pcbpel/iWay/wsd1/JDEdwards/jde9/GetPho
ne"
xmlns:plt="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
xmlns:GetPhone="http://xmlns.oracle.com/pcbpel/iWay/wsd1/JDEdwards/jde9/GetPhon
e"
xmlns:jca="http://xmlns.oracle.com/pcbpel/wsd1/jca/"
xmlns:iWay="http://xmlns.oracle.com/pcbpel/adaptor/iWay/"
xmlns:pc="http://xmlns.oracle.com/pcbpel/"
xmlns:iWayRequest="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
```

```
xmlns:iWayResponse="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone
.response"
xmlns="http://schemas.xmlsoap.org/wsd1/">
```

Change 2

Add `.response` at the end of `targetNamespace` and `ns` declaration in the schema section above the `<xsd:element name="jdeResponse">` line. For example:

Original

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
xmlns:ns="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
elementFormDefault="qualified">
<xsd:element name="jdeResponse">
```

Modified

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone.re
sponse"
xmlns:ns="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone.response"
elementFormDefault="qualified">
<xsd:element name="jdeResponse">
```

Change 3

Change `iWayRequest` to `iWayResponse` in the element field for the response message section. For example:

Original

```
<message name="response">
  <part name="output_GetPhone" element="iWayRequest:jdeResponse"/>
</message>
```

Modified

```
<message name="response">
  <part name="output_GetPhone" element="iWayResponse:jdeResponse"/>
</message>
```

10. Open and edit the `DefaultSystem_CustomAdapterServiceName.esbsvc` file (for example, `DefaultSystem_jde9_getphone.esbsvc`) in the reply validate section by adding `.response` in the `tns` namespace. For example:

Original

```
<reply validate="false"
xmlns:tns="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
element="tns:jdeResponse" schemaLocation="esb:///ESB_Projects/Upgrade_testing_
jde9_getphone_10134_jca_esb/JDE90_GetPhone_invoke.wsdl"/>
```

Modified

```
<reply validate="false"
xmlns:tns="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone.response"
" element="tns:jdeResponse" schemaLocation="esb:///ESB_Projects/Upgrade_
testing_jde9_getphone_10134_jca_esb/JDE90_GetPhone_invoke.wsdl"/>
```

11. Open and edit the *DefaultSystem_FileAdapterName.esbsvc* file created for the write operation (for example, *DefaultSystem_write_getphone.esbsvc*) in the request validate section by adding *.response* in the *tns* namespace. For example:

Original

```
<request validate="false"
xmlns:tns="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
element="tns:jdeResponse"/
```

Modified

```
<request validate="false"
xmlns:tns="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone.response"
" element="tns:jdeResponse"/>
```

12. Open and edit the *FileAdapterName.wsdl* file created for write operation (for example, *write_getphone.wsdl*) by adding *.response*. For example:

Original

```
<definitions name="write_getphone"
targetNamespace="http://xmlns.oracle.com/pcbpel/adapter/file/write_getphone/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="http://xmlns.oracle.com/pcbpel/adapter/file/write_getphone/"
xmlns:plt="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
xmlns:jca="http://xmlns.oracle.com/pcbpel/wsdl/jca/"
xmlns:impl="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
xmlns:hdr="http://xmlns.oracle.com/pcbpel/adapter/file/"
<import namespace="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
location="JDE90_GetPhone_invoke.wsdl"/>
```

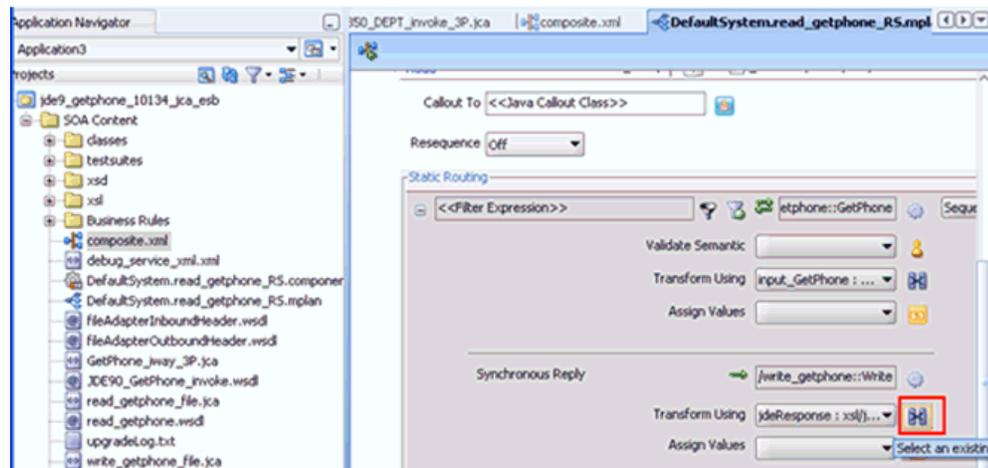
Modified

```
<definitions name="write_getphone"
targetNamespace="http://xmlns.oracle.com/pcbpel/adapter/file/write_getphone/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="http://xmlns.oracle.com/pcbpel/adapter/file/write_getphone/"
xmlns:plt="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
xmlns:jca="http://xmlns.oracle.com/pcbpel/wsdl/jca/"
xmlns:impl="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone.response"
xmlns:hdr="http://xmlns.oracle.com/pcbpel/adapter/file/"
<import
namespace="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone.response"
" location="JDE90_GetPhone_invoke.wsdl"/>
```

13. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 ESB project.

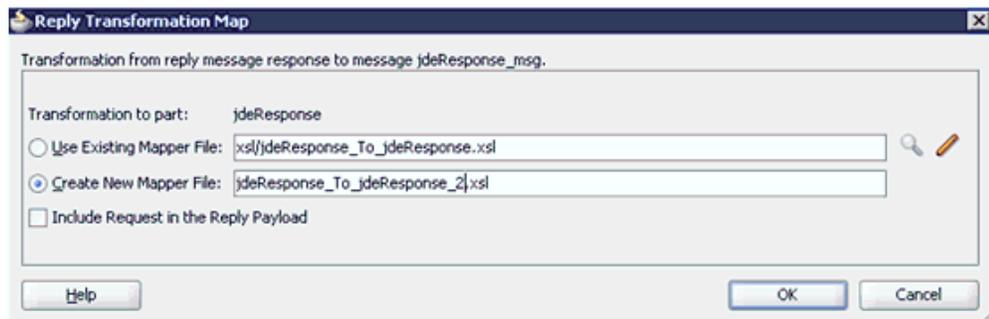
14. Once the project is migrated, a successful message is received in Oracle JDeveloper.
15. Expand the migrated project and double-click the *composite.xml* file to ensure that the project opens without any errors.
 Since changes were made to the WSDL file for the response section, a new mapper file must be created for *jdeResponse*.
16. Double-click the mediator component and go to the Synchronous Reply area.
17. Click the **Select an existing mapper file or create a new one** button in the Transform Using field, as shown in [Figure 6-1](#).

Figure 6-1 Transform Using Field

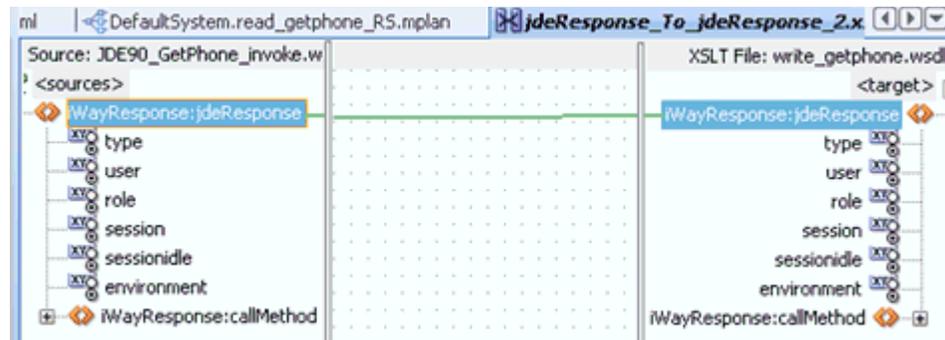


The Reply Transformation Map dialog is displayed, as shown in [Figure 6-2](#).

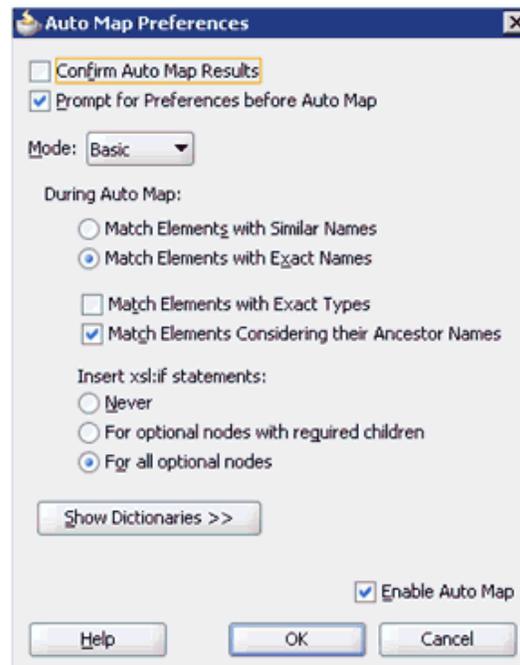
Figure 6-2 Reply Transformation Map Dialog



18. Click the **Create New Mapper File** option and click **OK**.
19. Automap the **iWayResponse:jdeResponse** in the source and **iWayResponse:jdeResponse** in the target, as shown in [Figure 6-3](#).

Figure 6–3 Mapping Source To Target

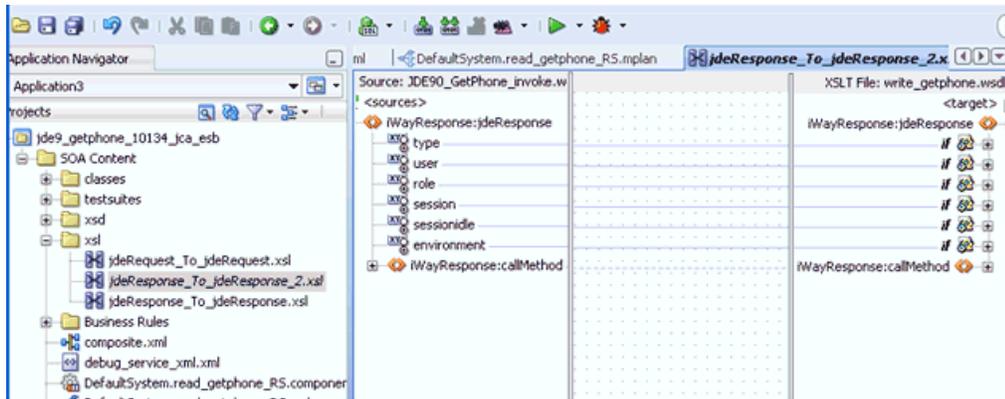
The Auto Map Preferences dialog is displayed, as shown in [Figure 6–4](#).

Figure 6–4 Auto Map Preferences Dialog

20. Accept the default values and click **OK**.

The automap is completed successfully, as shown in [Figure 6–5](#).

Figure 6–5 Completed Automap



21. Double click the **composite.xml** file.
22. Save and then deploy the migrated ESB project.
23. Ensure that there are no error or warning messages during the deployment process.
24. Once the deployment is successful, navigate to the input folder and paste the input XML file.

The successful response XML is received in the specified output folder.

Selecting a WSDL Using Service Explorer

This workaround must be performed for a J.D. Edwards outbound ESB process when migrating from 10.1.3.x to PS2/PS3/PS4/PS5/PS6. If you are migrating 10.1.3.x to PS1, then you can follow the same procedure in ["Selecting a WSDL Using Service Explorer"](#) on page 3-2.

1. On the 10.1.3.4 system, create a new target for the Oracle Application Adapter for J.D. Edwards OneWorld using Application Explorer and connect to the target.
2. Browse to a specific business object and generate an outbound WSDL file for this object.
3. Restart the server.
4. Start Oracle JDeveloper 10.1.3.4 and create a JCA outbound ESB project.
5. In the Custom Adapter service, select the outbound WSDL using Service Explorer.
6. Deploy the ESB project successfully and ensure that the registration of the ESB project is successful.
7. Navigate to the location where the 10.1.3.4 WSDL file is generated and edit the WSDL file with the following changes:

Change 1

Add the following line in the `<definitions>` section for `iWayResponse`:

```
xmlns:iWayResponse="urn:iwaysoftware:jde/services/CALLBFSN/Addressbook/GetPhone.response"
```

For example:

```
<definitions name="GetPhone"
targetNamespace="http://xmlns.oracle.com/pcbpel/iWay/wsd1/JDEdwards/jde9/GetPho
```

```

ne"
xmlns:plt="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
xmlns:GetPhone="http://xmlns.oracle.com/pcbpel/iWay/wsd/JDEdwards/jde9/GetPhone"
xmlns:jca="http://xmlns.oracle.com/pcbpel/wsd/jca/"
xmlns:iWay="http://xmlns.oracle.com/pcbpel/adaptor/iWay/"
xmlns:pc="http://xmlns.oracle.com/pcbpel/"
xmlns:iWayRequest="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"

xmlns:iWayResponse="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone
.response"
xmlns="http://schemas.xmlsoap.org/wsd/">

```

Change 2

Add `.response` at the end of the `targetNamespace` and `ns` declaration in the schema section above the `<xsd:element name="jdeResponse">` line. For example:

Original

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
xmlns:ns="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
elementFormDefault="qualified">
<xsd:element name="jdeResponse">

```

Modified

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone.re
sponse"
xmlns:ns="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone.response"
elementFormDefault="qualified">
<xsd:element name="jdeResponse">

```

Change 3

Change `iWayRequest` to `iWayResponse` in the element field for the response message section. For example:

Original

```

<message name="response">
  <part name="output_GetPhone" element="iWayRequest:jdeResponse"/>
</message>

```

Modified

```

<message name="response">
  <part name="output_GetPhone" element="iWayResponse:jdeResponse"/>
</message>

```

8. On the 11g system, create a target using the same name that was provided on the 10.1.3.4 system.
9. Copy the deployed ESB project into the 11g system
10. Open the project folder and edit the `DefaultSystem_CustomAdapterServiceName.esbsvc` file (for example, `DefaultSystem_jde9_getphone.esbsvc`) by providing the system IP address and port number (for example, `192.168.128.122:80`) of the 10.1.3.4 WSDL generated system in

the `<wsdlURL>` section that is located within `<serviceDefinition>`. For example:

Original

```
<serviceDefinition>
<wsdlURL>http://127.0.0.1:80/orainfra/wsdl/adapters/applications/ JDE90_
GetPhone_invoke.wsdl?wsdl</wsdlURL>
-----
</endpointDefinition>
</serviceDefinition>
```

Modified

```
<serviceDefinition>
<wsdlURL>http://192.168.128.122:80/orainfra/wsdl/adapters/applications/ JDE90_
GetPhone_invoke.wsdl?wsdl</wsdlURL>
-----
</endpointDefinition>
</serviceDefinition>
```

- Open and edit the `DefaultSystem_CustomAdapterServiceName.esbsvc` file (for example, `DefaultSystem_jde9_getphone.esbsvc`) in the reply validate section by adding `.response` in the `tns` namespace. For example:

Original

```
<reply validate="false"
xmlns:tns="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
element="tns:jdeResponse" schemaLocation="esb:///ESB_Projects/Upgrade_testing_
jde9_getphone_10134_jca_esb/JDE90_GetPhone_invoke.wsdl"/>
```

Modified

```
<reply validate="false"
xmlns:tns="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone..response"
" element="tns:jdeResponse" schemaLocation="esb:///ESB_Projects/Upgrade_
testing_jde9_getphone_10134_jca_esb/JDE90_GetPhone_invoke.wsdl"/>
```

- Open and edit the `DefaultSystem_FileAdapterName.esbsvc` file created for the write operation (for example, `DefaultSystem_write_getphone.esbsvc`) in the request validate section by adding `.response` in the `tns` namespace. For example:

Original

```
<request validate="false"
xmlns:tns="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
element="tns:jdeResponse"/>
```

Modified

```
<request validate="false"
xmlns:tns="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone..response"
" element="tns:jdeResponse"/>
```

- Open and edit the `FileAdapterName.wsdl` file created for the write operation (for example, `write_getphone.wsdl`) by adding `.response`. For example:

Original

```
<definitions name="write_getphone"
targetNamespace="http://xmlns.oracle.com/pcbpel/adapter/file/write_getphone/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
```

```

xmlns:tns="http://xmlns.oracle.com/pcbpel/adapter/file/write_getphone/"
xmlns:plt="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
xmlns:jca="http://xmlns.oracle.com/pcbpel/wsd/jca/"
xmlns:impl="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
xmlns:hdr="http://xmlns.oracle.com/pcbpel/adapter/file/"
<import namespace="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone"
  location="JDE90_GetPhone_invoke.wsdl" />

```

Modified

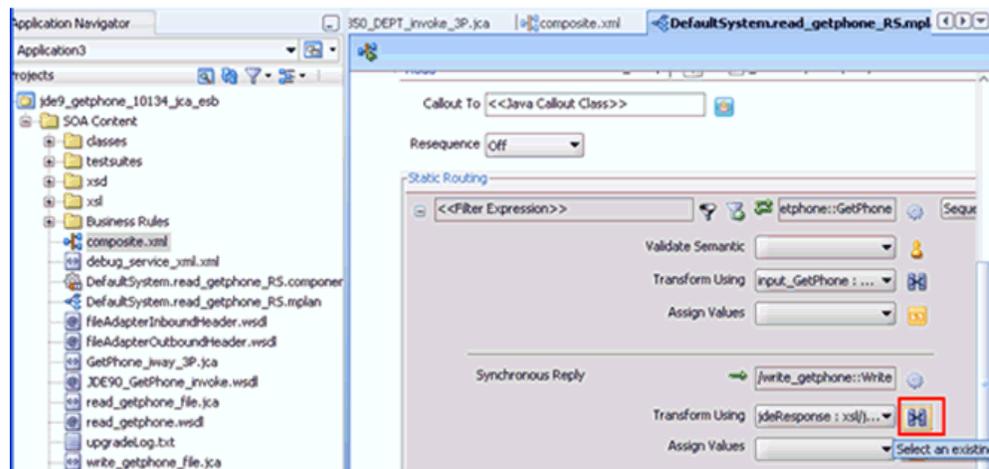
```

<definitions name="write_getphone"
targetNamespace="http://xmlns.oracle.com/pcbpel/adapter/file/write_getphone/"
  xmlns="http://schemas.xmlsoap.org/wsd/"
  xmlns:tns="http://xmlns.oracle.com/pcbpel/adapter/file/write_getphone/"
  xmlns:plt="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
  xmlns:jca="http://xmlns.oracle.com/pcbpel/wsd/jca/"
  xmlns:impl="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone.response"
  xmlns:hdr="http://xmlns.oracle.com/pcbpel/adapter/file/">
<import
namespace="urn:iwaysoftware:jde/services/CALLBSFN/Addressbook/GetPhone.response"
location="JDE90_GetPhone_invoke.wsdl" />

```

14. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 ESB project.
15. Once the project is migrated, a successful message is received in Oracle JDeveloper.
16. Expand the migrated project and double-click the *composite.xml* file to ensure that the project opens without any errors.
 Since changes were made to the WSDL file for the response section, a new mapper file must be created for *jdeResponse*.
17. Double-click the mediator component and go to the Synchronous Reply area.
18. Click the **Select an existing mapper file or create a new one** button in the Transform Using field, as shown in Figure 6–6.

Figure 6–6 Transform Using Field



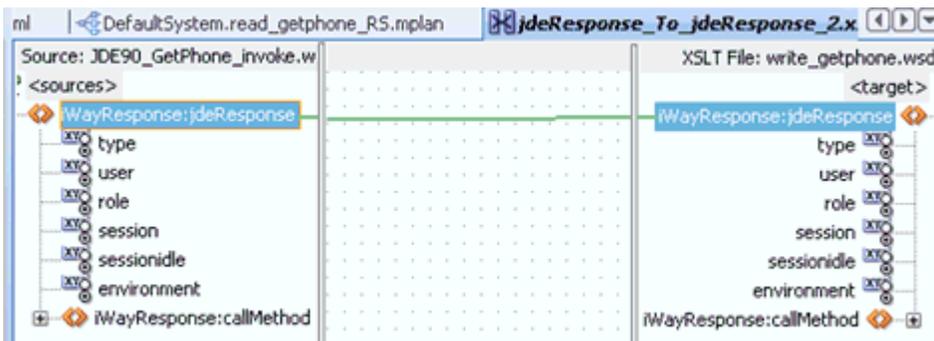
The Reply Transformation Map dialog is displayed, as shown in Figure 6–7.

Figure 6–7 Reply Transformation Map Dialog

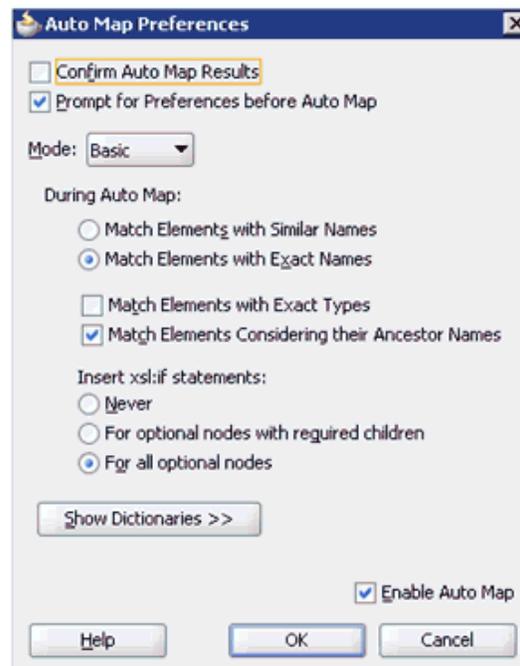


19. Click the **Create New Mapper File** option and click **OK**.
20. Automap the **iWayResponse:jdeResponse** in the source and **iWayResponse:jdeResponse** in the target, as shown in [Figure 6–8](#).

Figure 6–8 Mapping Source To Target

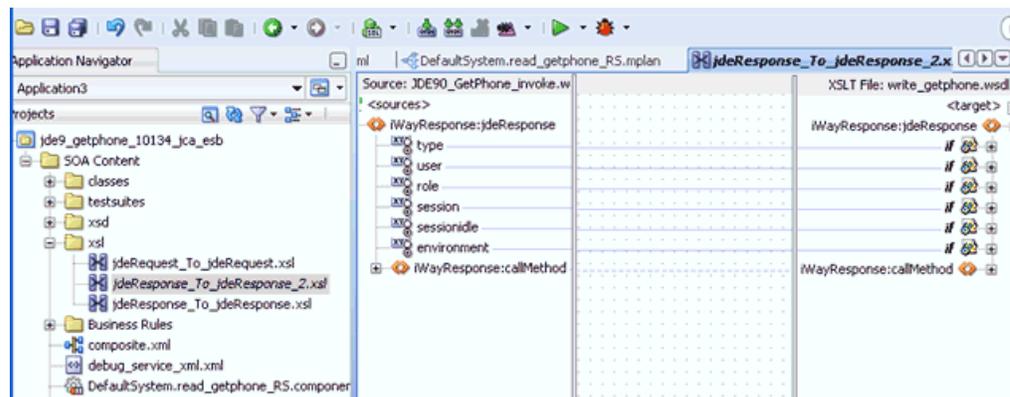


The Auto Map Preferences dialog is displayed, as shown in [Figure 6–9](#).

Figure 6–9 Auto Map Preferences Dialog

21. Accept the default values and click **OK**.

The automap is completed successfully, as shown in [Figure 6–10](#).

Figure 6–10 Completed Automap

22. Double click the **composite.xml** file.

23. Save and then deploy the migrated ESB project.

24. Ensure that there are no error or warning messages during the deployment process.

25. Once the deployment is successful, navigate to the input folder and paste the input XML file.

The successful response XML is received in the specified output folder.

6.2 Upgrading a 10.1.3.x ESB J2CA Inbound Process to 11g

Selecting a WSDL From the Local File System

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for J.D. Edwards OneWorld using Application Explorer and connect to the target.
2. Browse to a specific business object.
3. Create a new port and TCP channel for Oracle Application Adapter for J.D. Edwards OneWorld.
4. Trigger from the back-end and generate a schema with the response XML that was received in the port location using XMLSpy.
5. Copy the generated schema (.xsd file) in the following location:

```
<soadp1>\adapters\application\config\jca_sample\schemas\JDEdwards\target
```

6. Generate an inbound WSDL manually using the generated schema.
7. Make the following changes to the generated WSDL, as shown in the following examples:

Change 1

Replace **Schemas-jdedwards-com** to **iwaysoftware** in the target namespace:

```
<xs:schema
targetNamespace="urn:Schemas-jdedwards-com:trans.response.JDES00OUT"
  xmlns="urn:Schemas-jdedwards-com:trans.response.JDES00OUT"
  xmlns:xs=http://www.w3.org/2001/XMLSchema elementFormDefault="qualified">
```

to:

```
<xs:schema
  targetNamespace="urn:iwaysoftware:trans.response.JDES00OUT"
  xmlns="urn:iwaysoftware:trans.response.JDES00OUT"
  xmlns:xs=http://www.w3.org/2001/XMLSchema elementFormDefault="qualified">
```

Change 2

Cut and place the `<xs:element name="jdeResponse"></xs:element>` section before the `<xs:element name="transaction">` section in the WSDL as shown in the following example:

```
<xs:element name="jdeResponse">
  <xs:complexType>
    <xs:sequence>
      -----
</xs:element>
  <xs:element name="transaction">
    -----
```

Change 3

Replace **iWayEvent:transaction** to **iWayEvent:jdeResponse**, which is located in the `<message name="event">` section:

```
<message name="event">
  <part name="event_jde_inbound" element="iWayEvent:jdeResponse" />
</message>
```

to:

```
<message name="event">
  <part name="event_jde_inbound" element="iWayEvent:jdeResponse"/>
</message>
```

8. Restart the server.
9. Start Oracle JDeveloper 10.1.3.4 and create a JCA inbound ESB Project.
10. In the Custom Adapter service, select the inbound WSDL from the local file system.
11. Deploy the ESB project successfully and ensure the registration of the ESB project is successful.
12. On the 11g system, ensure that the target and channel that are created use the same name as specified on the 10.1.3.4 system. Copy the 10.1.3.4 schema and modify the 10.1.3.4 schema (jdeschema.xsd) as described in step 7 (only change 1 and change 2), which is used for 10.1.3.4 inbound WSDL creation. While creating the channel on the 11g system, in the PreParser tab, provide the modified event schema location (for example, c:\jdeschema.xsd).
13. Copy the deployed ESB project into the 11g system.
14. Start the Oracle JDeveloper 11g and migrate the 10.1.3.4 ESB project.
15. Once the project is migrated, a successful message is received in Oracle JDeveloper.
16. Expand the migrated project and double-click the composite.xml file to ensure that the project opens without any errors.
17. Double-click the J2CA properties file under the migrated project in Oracle JDeveloper and remove the following line:

```
<record-converter
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

For example:

```
<adapter-config name="jde90_schema_ori" adapter="iWay ERP Adapter"
wsdlLocation="jde9_salesorder_tcp_Feb27.wsdl"
xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">
<resource-adapter
className="com.iwaysoftware.afjca15.IWAFOracleResourceAdapter"/>
<record-converter
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

18. Save and then deploy the migrated ESB project.
19. Ensure that there are no error or warning messages during the deployment process.
20. Trigger the event messages from the J.D. Edwards OneWorld system and verify that successful response XML files are received in the specified output folder.

Selecting a WSDL Using Service Explorer

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for J.D. Edwards OneWorld using Application Explorer and connect to the target.
2. Browse to a specific business object.
3. Create a new port and TCP channel for Oracle Application Adapter for J.D. Edwards OneWorld.

4. Trigger from the back-end and generate a schema with the response XML that was received in the port location using XMLSpy.
5. Copy the generated schema (.xsd file) in the following location:

```
<soadpI>\adapters\application\config\jca_sample\schemas\JDEdwards\target
```

6. Generate an inbound WSDL manually using the generated schema.
7. Make the following changes to the generated WSDL, as shown in the following examples:

Change 1

Replace **Schemas-jdedwards-com** to **iwaysoftware** in the target namespace:

```
<xs:schema
targetNamespace="urn:Schemas-jdedwards-com:trans.response.JDES00UT"
  xmlns="urn:Schemas-jdedwards-com:trans.response.JDES00UT"
  xmlns:xs=http://www.w3.org/2001/XMLSchema elementFormDefault="qualified">
```

to:

```
<xs:schema
  targetNamespace="urn:iwaysoftware:trans.response.JDES00UT"
  xmlns="urn:iwaysoftware:trans.response.JDES00UT"
  xmlns:xs=http://www.w3.org/2001/XMLSchema elementFormDefault="qualified">
```

Change 2

Cut and place the **<xs:element name="jdeResponse"></xs:element>** section before the **<xs:element name="transaction">** section in the WSDL as shown in the following example:

```
<xs:element name="jdeResponse">
  <xs:complexType>
    <xs:sequence>
      -----
</xs:element>
  <xs:element name="transaction">
    -----
```

Change 3

Replace **iWayEvent:transaction** to **iWayEvent:jdeResponse**, which is located in the **<message name="event">** section:

```
<message name="event">
  <part name="event_jde_inbound" element="iWayEvent:transaction" />
</message>
```

to:

```
<message name="event">
  <part name="event_jde_inbound" element="iWayEvent:jdeResponse" />
</message>
```

8. Restart the server.
9. Start Oracle JDeveloper 10.1.3.4 and create a JCA inbound ESB Project.
10. In the Custom Adapter service, select the inbound WSDL using Service Explorer.
11. Deploy the ESB project successfully and ensure the registration of the ESB project is successful.

12. On the 11g system, ensure that the target and channel that are created use the same name as specified on the 10.1.3.4 system. Copy the 10.1.3.4 schema and modify the 10.1.3.4 schema (jdeschema.xsd) as described in step 7 (only change 1 and change 2), which is used for 10.1.3.4 inbound WSDL creation. While creating the channel on the 11g system, in the PreParser tab, provide the modified event schema location (for example, c:\jdeschema.xsd).
13. Copy the deployed ESB project into the 11g system.
14. Open the project folder and edit the **DefaultSystem_CustomAdapterServiceName.esbsvc** file (for example, DefaultSystem_SalesOrder.esbsvc) and **DefaultSystem_CustomAdapterServiceName_RS.esbsvc** file (for example, DefaultSystem_SalesOrder_RS.esbsvc) by providing the system IP address and port number (for example, 192.168.128.122:80) of the 10.1.3.4 WSDL generated system in the <wsdlURL> sections. For example:

1. Original (DefaultSystem_SalesOrder.esbsvc)

```
<interface> <wsdlURL>http://127.0.0.1:80/orainfra/wsdl/adapters/applications/
jde90_SalesOrder_receive.wsdl?wsdl</wsdlURL>
-----
</interface>
```

Modified (DefaultSystem_SalesOrder.esbsvc)

```
<interface>
<wsdlURL>http://192.168.128.122:80/orainfra/wsdl/adapters/applications/
jde90_SalesOrder_receive.wsdl?wsdl</wsdlURL>
-----
</interface>
```

2. Original (DefaultSystem_SalesOrder_RS.esbsvc)

```
<serviceDefinition>
<wsdlURL>http://127.0.0.1:80/orainfra/wsdl/adapters/applications/ jde90_
SalesOrder_receive.wsdl?wsdl</wsdlURL>
-----
</endpointDefinition>
</serviceDefinition>
```

Modified (DefaultSystem_SalesOrder_RS.esbsvc)

```
<serviceDefinition>
<wsdlURL>http://192.168.128.122:80/orainfra/wsdl/adapters/applications/ jde90_
SalesOrder_receive.wsdl?wsdl</wsdlURL>
-----
</endpointDefinition>
</serviceDefinition>
```

15. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 ESB project.
16. Once the project is migrated, a successful message is received in Oracle JDeveloper.
17. Expand the migrated project and double-click the composite.xml file to ensure that the project opens without any errors.
18. Double-click the J2CA properties file under the migrated project in Oracle JDeveloper and remove the following line:

```
<record-converter
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

For example:

```
<adapter-config name="jde90_schema_ori" adapter="iWay ERP Adapter"
wsdlLocation="jde9_salesorder_tcp_Feb27.wsdl"
xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">
<resource-adapter
className="com.iwaysoftware.afjca15.IWAFOracleResourceAdapter"/>
<record-converter
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

19. Save and then deploy the migrated ESB project.
20. Ensure that there are no error or warning messages during the deployment process.
21. Trigger the event messages from the J.D. Edwards OneWorld system and verify that successful response XML files are received in the specified output folder.

6.3 Upgrading a 10.1.3.x BPEL J2CA Outbound Process to 11g

Selecting a WSDL From the Local File System

You can follow the same procedure in ["Selecting a WSDL From the Local File System"](#) on page 3-5.

Selecting a WSDL Using Service Explorer

You can follow the same procedure in ["Selecting a WSDL Using Service Explorer"](#) on page 3-6.

6.4 Upgrading a 10.1.3.x BPEL J2CA Inbound Process to 11g

Selecting a WSDL From the Local File System

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for J.D. Edwards OneWorld using Application Explorer and connect to the target.
2. Browse to a specific business object.
3. Create a new port and TCP channel for Oracle Application Adapter for J.D. Edwards OneWorld.
4. Trigger from the back-end and generate a schema with the response XML that was received in the port location using XMLSpy.
5. Copy the generated schema (.xsd file) in the following location:

```
<soadp1>\adapters\application\config\jca_sample\schemas\JDEdwards\target
```

6. Generate an inbound WSDL manually using the generated schema.
7. Make the following changes to the generated WSDL, as shown in the following examples:

Change 1

Replace **Schemas-jdedwards-com** to **iwaysoftware** in the target namespace:

```
<xs:schema
targetNamespace="urn:Schemas-jdedwards-com:trans.response.JDESOUT"
xmlns="urn:Schemas-jdedwards-com:trans.response.JDESOUT"
xmlns:xs=http://www.w3.org/2001/XMLSchema elementFormDefault="qualified">
```

to:

```
<xs:schema
  targetNamespace="urn:iwaysoftware:trans.response.JDES00UT"
  xmlns="urn:iwaysoftware:trans.response.JDES00UT"
  xmlns:xs=http://www.w3.org/2001/XMLSchema elementFormDefault="qualified">
```

Change 2

Cut and place the `<xs:element name="jdeResponse"></xs:element>` section before the `<xs:element name="transaction">` section in the WSDL as shown in the following example:

```
  <xs:element name="jdeResponse">
    <xs:complexType>
      <xs:sequence>
        -----
      </xs:element>
    <xs:element name="transaction">
      -----
```

Change 3

Replace `iWayEvent:transaction` to `iWayEvent:jdeResponse`, which is located in the `<message name="event">` section:

```
<message name="event">
  <part name="event_jde_inbound" element="iWayEvent:transaction"/>
</message>
```

to:

```
<message name="event">
  <part name="event_jde_inbound" element="iWayEvent:jdeResponse"/>
</message>
```

8. Restart the server.
9. Start Oracle JDeveloper 10.1.3.4 and create the inbound BPEL Process by selecting the inbound WSDL from the local file system in the Partner Link.
10. Deploy the BPEL process successfully and ensure there are no error or warning messages during deployment.
11. Trigger the event messages from the J.D. Edwards OneWorld system and ensure that successful instances are received in the BPEL Console.
12. On the 11g system, ensure that the target and channel that are created use the same name as specified on the 10.1.3.4 system. Copy the 10.1.3.4 schema and modify the 10.1.3.4 schema (jdeschema.xsd) as described in step 7 (only change 1 and change 2), which is used for 10.1.3.4 inbound WSDL creation. While creating the channel on the 11g system, in the PreParser tab, provide the modified event schema location (for example, c:\jdeschema.xsd).
13. Copy the deployed BPEL project into the 11g system.
14. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 BPEL project.
15. Once the project is migrated, a successful message is received in Oracle JDeveloper.
16. Expand the migrated project and double-click the composite.xml file to ensure that the project opens without any errors.

17. Double-click the J2CA properties file under the migrated project in Oracle JDeveloper and remove the following line:

```
<record-converter
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

For example:

```
<adapter-config name="jde90_schema_ori" adapter="iWay ERP Adapter"
wsdlLocation="jde9_salesorder_tcp_Feb27.wsdl"
xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">
<resource-adapter
className="com.iwaysoftware.afjca15.IWAFOracleResourceAdapter"/>
<record-converter
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

18. Save and then deploy the migrated BPEL project.
19. Ensure that there are no error or warning messages during the deployment process.
20. Trigger the event messages from the J.D. Edwards OneWorld system and ensure that successful instances are received in the Oracle Enterprise Manager console.

Selecting a WSDL Using Service Explorer

1. On the 10.1.3.4 system, create a new target for Oracle Application Adapter for J.D. Edwards OneWorld using Application Explorer and connect to the target.
2. Browse to a specific business object.
3. Create a new port and TCP channel for Oracle Application Adapter for J.D. Edwards OneWorld.
4. Trigger from the back-end and generate a schema with the response XML that was received in the port location using XMLSpy.
5. Copy the generated schema (.xsd file) in the following location:

```
<soadpl>\adapters\application\config\jca_sample\schemas\JDEdwards\target
```

6. Generate an inbound WSDL manually using the generated schema.
7. Make the following changes to the generated WSDL, as shown in the following examples:

Change 1

Replace **Schemas-jdedwards-com** to **iwaysoftware** in the target namespace:

```
<xs:schema
targetNamespace="urn:Schemas-jdedwards-com:trans.response.JDES00T"
xmlns="urn:Schemas-jdedwards-com:trans.response.JDES00T"
xmlns:xs=http://www.w3.org/2001/XMLSchema elementFormDefault="qualified">
```

to:

```
<xs:schema
targetNamespace="urn:iwaysoftware:trans.response.JDES00T"
xmlns="urn:iwaysoftware:trans.response.JDES00T"
xmlns:xs=http://www.w3.org/2001/XMLSchema elementFormDefault="qualified">
```

Change 2

Cut and place the `<xs:element name="jdeResponse"></xs:element>` section before the `<xs:element name="transaction">` section in the WSDL as shown in the following example:

```
<xs:element name="jdeResponse">
  <xs:complexType>
    <xs:sequence>
      -----
</xs:element>
  <xs:element name="transaction">
    -----
```

Change 3

Replace `iWayEvent:transaction` to `iWayEvent:jdeResponse`, which is located in the `<message name="event">` section:

```
<message name="event">
  <part name="event_jde_inbound" element="iWayEvent:transaction"/>
</message>
```

to:

```
<message name="event">
  <part name="event_jde_inbound" element="iWayEvent:jdeResponse"/>
</message>
```

8. Restart the server.
9. Start Oracle JDeveloper 10.1.3.4 and create the inbound BPEL Process by selecting the inbound WSDL from the Service Explorer in the Partner Link.
10. Deploy the BPEL process successfully and ensure there are no error or warning messages during deployment.
11. Trigger the event messages from the J.D. Edwards OneWorld system and ensure that successful instances are received in the BPEL Console.
12. On the 11g system, ensure that the target and channel that are created use the same name as specified on the 10.1.3.4 system. Copy the 10.1.3.4 schema and modify the 10.1.3.4 schema (jdeschema.xsd) as described in step 7 (only change 1 and change 2), which is used for 10.1.3.4 inbound WSDL creation. While creating the channel on the 11g system, in the PreParser tab, provide the modified event schema location (for example, c:\jdeschema.xsd).
13. Copy the deployed BPEL project into the 11g system.
14. Edit the `bpel.xml` file for the WSDL location in the 10.1.3.4 project folder (Project_Name\bpel):

```
<partnerLinkBinding name=" jde90_schema_ori">
  <property
name="wsdlLocation">http://192.168.128.125:80/orainfra/wsdl/adapters/applicatio
ns/ jde9_salesorder_tcp_Feb27.wsdl?wsdl</property>
</partnerLinkBinding>
```

Where the IP address and port number refer to the 10.1.3.4 system where the WSDL was generated.

15. Ensure that the 10.1.3.4 server is up and running.
16. Start Oracle JDeveloper 11g and migrate the 10.1.3.4 BPEL project.

17. Once the project is migrated, a successful message is received in Oracle JDeveloper.
18. Expand the migrated project and double-click the composite.xml file to ensure that the project opens without any errors.
19. Double-click the J2CA properties file under the migrated project in Oracle JDeveloper and remove the following line:

```
<record-converter  
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

For example:

```
<adapter-config name="jde90_schema_ori" adapter="iWay ERP Adapter"  
wsdlLocation="jde9_salesorder_tcp_Feb27.wsdl"  
xmlns="http://platform.integration.oracle/blocks/adapter/fw/metadata">  
<resource-adapter  
className="com.iwaysoftware.afjca15.IWAFOracleResourceAdapter"/>  
<record-converter  
className="com.iwaysoftware.afjca15.oracle.InboundXMLRecordConverterImpl"/>
```

20. Save and then deploy the migrated BPEL project.
21. Ensure that there are no error or warning messages during the deployment process.
22. Trigger the event messages from the J.D. Edwards OneWorld system and ensure that successful instances are received in the Oracle Enterprise Manager console.

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.

G

general upgrade guidelines, 2-1

I

inbound ESB and BPEL process
 post-upgrade requirement, 2-26
inbound process
 upgrading, 2-17

O

Oracle Application Adapter for J.D. Edwards
 OneWorld
 BPEL J2CA inbound process
 upgrading, 6-16
 BPEL J2CA outbound process
 upgrading, 6-16
 ESB J2CA inbound process
 upgrading, 6-12
 ESB J2CA outbound process
 upgrading, 6-1
Oracle Application Adapter for PeopleSoft
 BPEL J2CA inbound process
 upgrading, 5-5
 BPEL J2CA outbound process
 upgrading, 5-4
 ESB J2CA inbound process
 upgrading, 5-1
 ESB J2CA outbound process
 upgrading, 5-1
Oracle Application Adapter for SAP R/3
 BPEL J2CA inbound process
 upgrading, 3-7
 BPEL J2CA outbound process
 upgrading, 3-5
 ESB J2CA inbound process
 upgrading, 3-3
 ESB J2CA outbound process
 upgrading, 3-1
Oracle Application Adapter for Siebel
 BPEL J2CA inbound process
 upgrading, 4-9
 BPEL J2CA outbound process
 upgrading, 4-8

ESB J2CA inbound process
 upgrading, 4-1

ESB J2CA outbound process
 upgrading, 4-1

Oracle's Unified Method (OUM), viii
outbound ESB and BPEL process
 post-upgrade requirement, 2-16
outbound process
 upgrading, 2-1

U

upgrade guidelines
 general, 2-1
 Oracle Application Adapter for J.D. Edwards
 OneWorld, 6-1
 Oracle Application Adapter for PeopleSoft, 5-1
 Oracle Application Adapter for SAP R/3, 3-1
 Oracle Application Adapter for Siebel, 4-1

