Oracle® Fusion Middleware Application Adapters

Application Adapter Upgrade Guide for

12*c* (12.2.1.2.0)

E84209-01

December 2016

Provides information on how to upgrade Oracle Application Adapters for Oracle WebLogic Server from 11*g* PS6, 12*c* (12.1.3.0.0) to 12*c* (12.2.1.2.0).

ORACLE

Oracle Fusion Middleware Application Adapter 12*c* (12.2.1.2.0) Upgrade Guide for Oracle WebLogic Server, 12*c* (12.2.1.2.0)

E84209-01

Primary Author: Stefan Kostial

Contributors: Vikas Anand, Marian Jones, Sunil Gopal, Bo Stern

Contents

2 General Upgrade Guidelines

Index

Preface

Welcome to *Oracle Fusion Middleware Application Adapter Upgrade Guide for Oracle WebLogic Server*. This document provides information on how to upgrade Oracle Application Adapters for Oracle WebLogic Server from 11*g* PS6, 12*c* (12.1.3.0.0), 12*c* Cumulative-1 (12.1.3.0.0) to 12*c* (12.2.1.2.0).

Audience

This document is intended for system administrators who are upgrading Oracle Application Adapters for Oracle WebLogic Server from 11*g* PS6, 12*c* (12.1.3.0.0), 12*c* Cumulative-1 (12.1.3.0.0) to 12*c* (12.2.1.2.0).

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit
http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Enterprise Repository 12*c* (12.2.1.2.0) 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 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.
italic	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.

1 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. 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"

Note: <ORACLE_HOME> across this document refers the 12*c* (12.2.1.0.0) SOA/OSB installed home location.

<ADAPTER_HOME> across this document refers to the following:

For SOA:

```
<ORACLE_HOME>\soa\soa\thirdparty\ApplicationAdapters
```

For OSB:

```
<ORACLE_HOME>\osb\3rdparty\ApplicationAdapters
```

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, MSSQL, and DB2 databases as repositories.
- Migration between:
  - File to database repositories
  - Database to database repositories

This section contains the following topics:

1.1.1 Getting Started

This section contains the following topics:

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`

  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:

   `<ADAPTER_HOME>``/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 MSSQL and DB2 databases, use the appropriate connection URL and database driver. An example is provided for DB2 and MSSQL repositories in Section 1.2.2.1, "Copying the Repository" on page 1-12.

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:

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 an Oracle 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
   ```

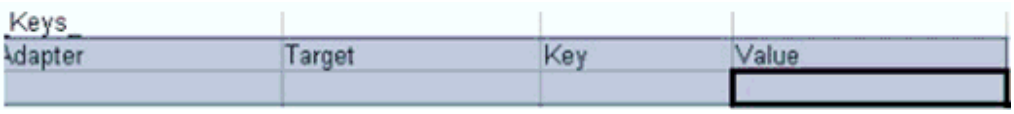
   On UNIX or Linux platforms:

   ```
   ./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



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]
   ```

2. 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
   ```

3. 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:

1. Using the diff utility to create a CSV file containing the objects that are in the source repository, but not in the target repository.
2. Removing any objects that are not desired in the target repository from the CSV file.
3. Editing the object parameters in the CSV file.
4. 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/ora117/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

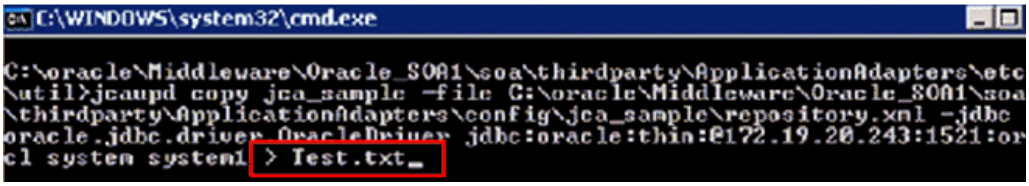
JDBC drivers must be placed in the `ApplicationAdapters\lib` folder where the migration utility is installed.

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.

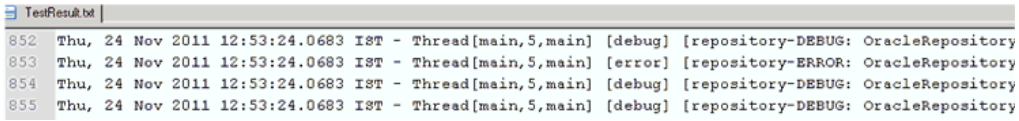
For example, in the command prompt, specify the command, `> Filename.txt`, as shown in Figure 1–2.

Figure 1–2 Command Prompt



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



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, MSSQL, and DB2 databases as repositories.
- Migration between:
  - File to database repositories
  - Database to database repositories

This section contains the following topics:

1.2.1 Getting Started

This section contains the following topics:

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`

  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:

   `<ADAPTER_HOME>``/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 MSSQL and DB2 databases, use the appropriate connection URL and database driver. An example has been given for DB2 and MSSQL repositories in Section 1.2.2.1, "Copying the Repository" on page 1-12.

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
   ```

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 an Oracle 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
   ```

4. 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.

1. 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 WebService_Name 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. 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.

1.2.2.3 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
   ```

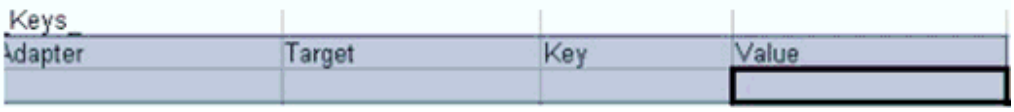
   On UNIX or Linux platforms:

```
./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



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]
   ```

2. 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
```

3. 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:

1. Using the diff utility to create a CSV file containing the objects that are in the source repository, but not in the target repository.
2. Removing any objects that are not desired in the target repository from the CSV file.
3. Editing the object parameters in the CSV file.
4. 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 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:

```
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

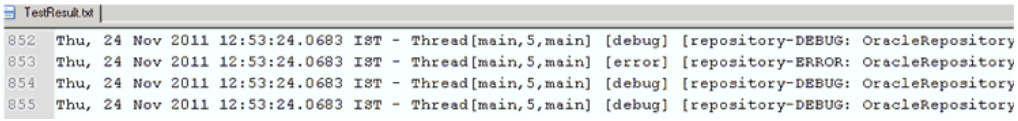
Log files are not be generated with the migration utility. All logging information is printed to standard output. To capture logging for review purposes, redirect the standard output to a file using the > character in the command window.

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

Figure 1–5 Command Prompt

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



2

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 11g PS6, 12c (12.1.3.0.0), and 12c Cumulative-1 (12.1.3.0.0) Outbound and Inbound BPEL, Mediator, and BPMN Processes to 12c (12.2.1.0.0)"
- Section 2.2, "Upgrading 12c (12.1.3.0.0) and 12c Cumulative-1 (12.1.3.0.0) Outbound and Inbound OSB Jdeveloper Processes to 12c (12.2.1.0.0)"
- Section 2.3, "Upgrading 11g PS6, 12c (12.1.3.0.0), and 12c Cumulative-1 (12.1.3.0.0) Outbound and Inbound OSB Sbconsole Processes to 12c (12.2.1.0.0)"

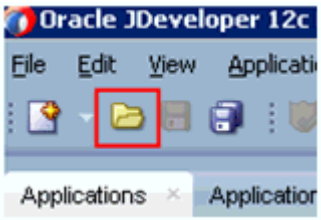
2.1 Upgrading 11*g* PS6, 12*c* (12.1.3.0.0), and 12*c* Cumulative-1 (12.1.3.0.0) Outbound and Inbound BPEL, Mediator, and BPMN Processes to 12*c* (12.2.1.0.0)

As a prerequisite, ensure that the adapter targets and channels created using Application Explorer for J2CA configurations in the Oracle 12*c* (12.2.1.0.0) environment are identical to those created in the previous releases. For BSE configurations, ensure that the adapter targets and Business Services created using Application Explorer in the Oracle 12*c* (12.2.1.0.0) environment are identical to those created in the previous releases.

In addition, ensure that you copy the previous releases of BPEL, Mediator, and BPMN processes for BSE and J2CA to the Oracle 12*c* (12.2.1.0.0) upgraded system location.

1. Open BPMN QuickStart JDeveloper 12*c* (12.2.1.0.0).
2. Select an available application and click **Open** from the toolbar, as shown in Figure 2–1.

Figure 2–1 Oracle JDeveloper Toolbar

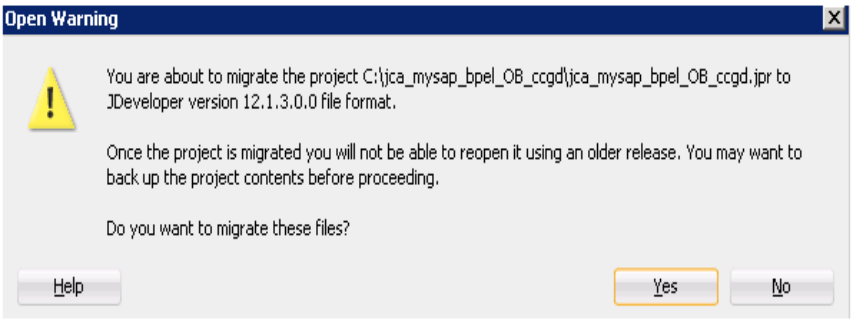


The Open dialog is displayed.

3. Open the previous release project and then select the .jpr extension file.
4. Click **Open**.

   The Open Warning dialog is displayed, as shown in Figure 2–2.

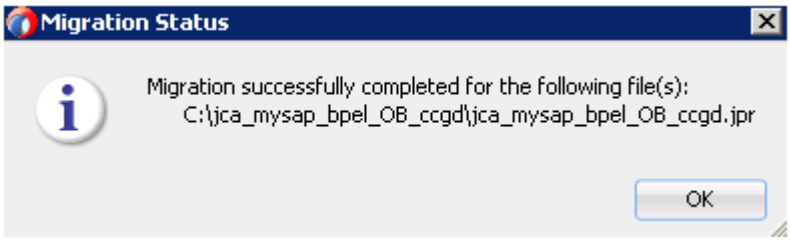
Figure 2–2 Open Warning Dialog



5. Click **Yes**.

   The Migration Status message is displayed, as shown in Figure 2–3.

Figure 2–3 Migration Status Message

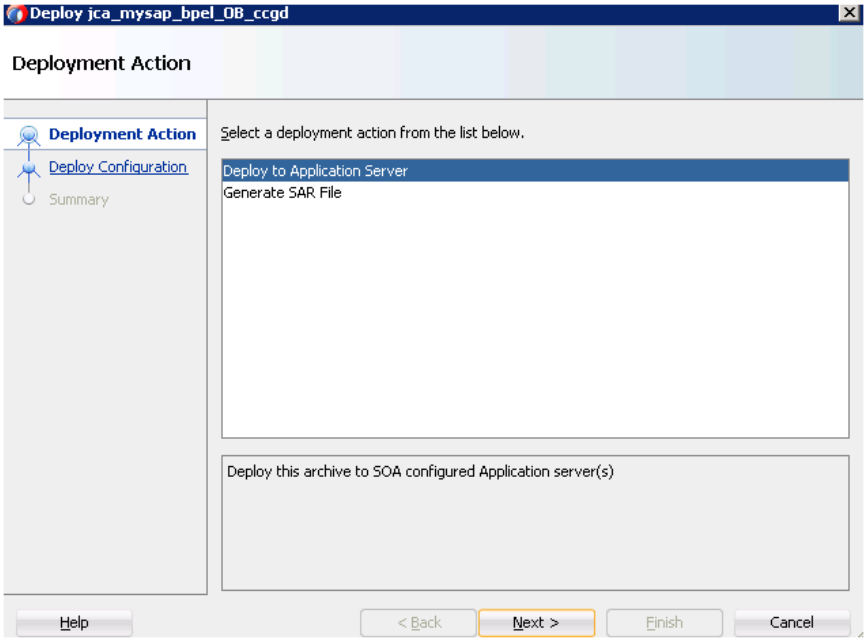


6. Click **OK**.

   The previous release project is now available in your Oracle 12*c* (12.2.1.0.0) environment.
7. Expand the migrated project in the left pane 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.

   The Deployment Action page is displayed, as shown in Figure 2–4.

Figure 2–4 Deployment Action Page

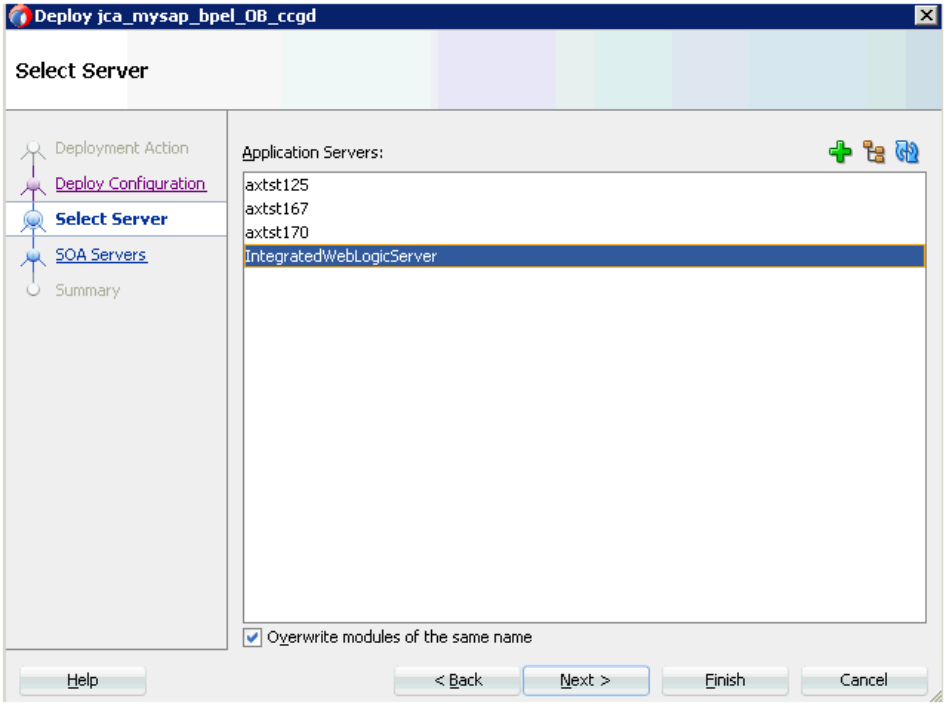


10. Ensure that **Deploy to Application Server** is selected.
11. Click **Next**.

    The Deploy Configuration page is displayed.
12. Leave the default values selected and click **Next**.

    The Select Server page is displayed, as shown in Figure 2–5.

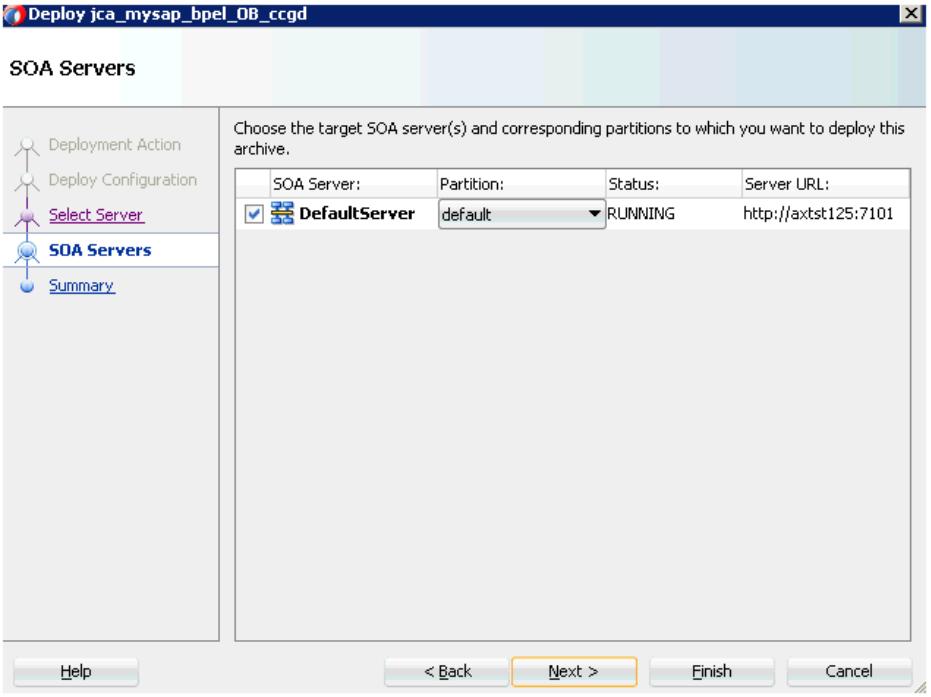
Figure 2–5 Select Server Page



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

    The SOA Servers page is displayed, as shown in Figure 2–6.

Figure 2–6 SOA Servers Page



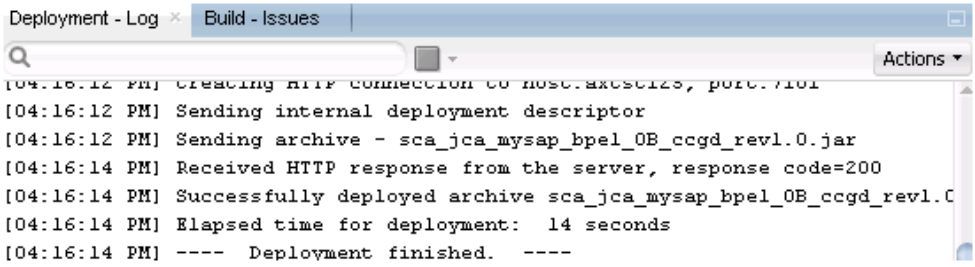
14. Select a partition from the Partition column list and click **Next**.

    The Summary page is displayed.

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 2–7.

Figure 2–7 Deployment Log



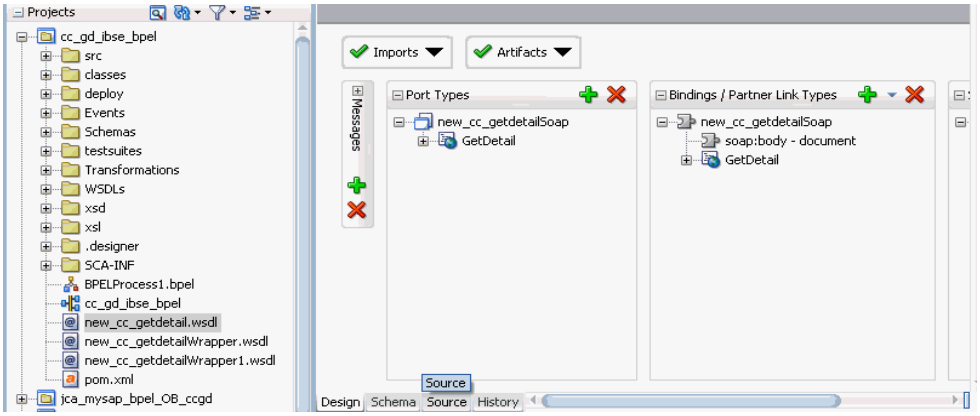
17. After the deployment is successful, open the Oracle Enterprise Manager console and execute the deployed process either in Tree View or XML View to receive a successful response.

2.1.1 Additional Modifications for Migrated Processes in 12*c* (12.2.1.0.0)

Upgraded J2CA outbound and inbound processes in 12*c* (12.2.1.0.0) 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 12*c* (12.2.1.0.0), double-click the **composite.xml** file to open the migrated project.
2. Double-click the BSE outbound WSDL file and then click the **Source** tab, as shown in Figure 2–8.

Figure 2–8 BSE Outbound WSDL File



3. Change the `<soap:address location>` element to point to the system and port number where 12*c* (12.2.1.0.0) is running.

   For example:

```
<service name="mysap_isdsrv2_compcode_getdetail">
<soap:address
location="http://172.19.95.190:7003/ibse/IBSEServlet/XDSOAPRouter"/>
        </port>
    </service>
</definitions>
```

4. Save and deploy the process.

2.2 Upgrading 12*c* (12.1.3.0.0) and 12*c* Cumulative-1 (12.1.3.0.0) Outbound and Inbound OSB Jdeveloper Processes to 12*c* (12.2.1.0.0)

This section describes how to migrate outbound and inbound J2CA processes and outbound BSE processes of previous releases from 12*c* (12.1.3.0.0) Jdeveloper to 12*c* (12.2.1.0.0) Jdeveloper. It contains the following topics:

- Section 2.2.1, "Exporting the Configured OSB Processes From 12c (12.1.3.0.0) Jdeveloper"
- Section 2.2.2, "Importing the Exported Processes to 12c (12.2.1.0.0) Jdeveloper"
- Section 2.2.3, "Additional Modifications for the Imported Process in 12c (12.2.1.0.0) Jdeveloper"

2.2.1 Exporting the Configured OSB Processes From 12*c* (12.1.3.0.0) Jdeveloper

This section describes how to export the configured OSB processes of previous releases from 12*c* (12.1.3.0.0) Jdeveloper.

Make sure all the configured OSB processes of previous releases are available in 12*c* (12.1.3.0.0) Jdeveloper before proceeding.

1. In the File menu, click **Export**.
2. From the Export window that appears, select **Service Bus Resources**.

   The Export Service Bus Resources window is displayed.
3. Select **Configuration Jar** and click **Next**.
4. In the displayed Source window, select the projects that need to be exported.
5. Provide the name and location in the export destination section and click **Finish**.

   The project is exported to the provided location.

2.2.2 Importing the Exported Processes to 12*c* (12.2.1.0.0) Jdeveloper

This section describes how to import the exported processes to 12*c* (12.2.1.0.0) Jdeveloper.

Prerequisites

The following are the prerequisites before importing the exported processes.

- All adapter targets and channels that were configured using Application Explorer in previous releases must match those in an Oracle Service Bus 12*c* (12.2.1.0.0) environment.
- All the input and output locations configured in an Oracle Service Bus 12*c* (12.2.1.0.0) environment must match those in an Oracle Service Bus previous releases.

Perform the following steps to import the exported processes:

1. Start the Oracle WebLogic Server for the domain you have configured.
2. Open Oracle 12*c* (12.2.1.0.0) SOA Quickstart JDeveloper.
3. Create a new Service Bus Application by providing an application name.
4. In the File menu, click **Import**.
5. From the Import window, select the **Service Bus Resources** option.

   The Import Service Bus Resources window is displayed.
6. Select **Configuration Jar** and click **Next**.
7. In the displayed Source window, click the **Browse** icon, select the **sbconfig.jar** file that was exported and copied to the local machine, and then click **Next**.
8. In the displayed Configuration window, verify that all the imported items are selected in the Resource Summary area by default, and then click **Finish**.
9. Save the imported OSB project.

2.2.3 Additional Modifications for the Imported Process in 12*c* (12.2.1.0.0) Jdeveloper

This section describes the additional modifications for the imported process in 12*c* (12.2.1.0.0) Jdeveloper.

Upgraded J2CA outbound and inbound processes in 12*c* (12.2.1.0.0) function properly and do not require any additional modifications. Only BSE outbound processes require additional modifications, which are described in the following steps.

1. Once the BSE outbound process is migrated successfully to 12*c* (12.2.1.0.0), double-click the **composite.xml** file to open the migrated project.
2. Double-click the **BSE outbound** WSDL file and then click the Source tab.
3. Change the `<soap:address location>` element to indicate the system and port number where 12*c* (12.2.1.0.0) is running.

   For example:

   ```
   <service name="sap_getdetail">
   <soap:address
   location="http://172.19.95.190:7003/ibse/IBSEServlet/XDSOAPRouter"/>
           </port>
       </service>
   </definitions>
   ```

4. Save and deploy the process.

2.3 Upgrading 11*g* PS6, 12*c* (12.1.3.0.0), and 12*c* Cumulative-1 (12.1.3.0.0) Outbound and Inbound OSB Sbconsole Processes to 12*c* (12.2.1.0.0)

This section describes how to migrate outbound and inbound J2CA processes and outbound BSE processes from Oracle Service Bus previous releases to 12*c* (12.2.1.0.0). It contains the following topics:

- Section 2.3.1, "Exporting the Configured Processes From Oracle Service Bus 11g PS6, 12c (12.1.3.0.0), and 12c Cumulative-1 (12.1.3.0.0)"
- Section 2.3.2, "Importing the Exported Processes to Oracle Service Bus 12c (12.2.1.0.0)"
- Section 2.3.3, "Additional Modifications for Imported Processes in Oracle Service Bus 12c (12.2.1.0.0)"

2.3.1 Exporting the Configured Processes From Oracle Service Bus 11*g* PS6, 12*c* (12.1.3.0.0), and 12*c* Cumulative-1 (12.1.3.0.0)

This section describes how to export the configured processes from Oracle Service Bus previous releases to 12*c* (12.2.1.0.0).

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 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.
4. Click the **Export Resources** button on the top left pane.

   The Export Resources window is opened.

5. Select the required projects and click **Export**.
6. Verify that the *sbconfig.jar* is saved successfully.

> **Note:** The *sbconfig.jar* file that is exported from Oracle Service Bus previous releases must be copied to the system where Oracle Service Bus 12*c* (12.2.1.0.0) is installed.

2.3.2 Importing the Exported Processes to Oracle Service Bus 12*c* (12.2.1.0.0)

This section describes how to import the exported processes to Oracle Service Bus 12*c* (12.2.1.0.0).

Prerequisites

- All the adapter targets and channels that are configured using Application Explorer in an Oracle Service Bus 12*c* (12.2.1.0.0) environment must match those in an Oracle Service Bus previous releases.
- All the input and output locations configured for the processes in an Oracle Service Bus 12*c* (12.2.1.0.0) environment must match those in an Oracle Service Bus previous releases.

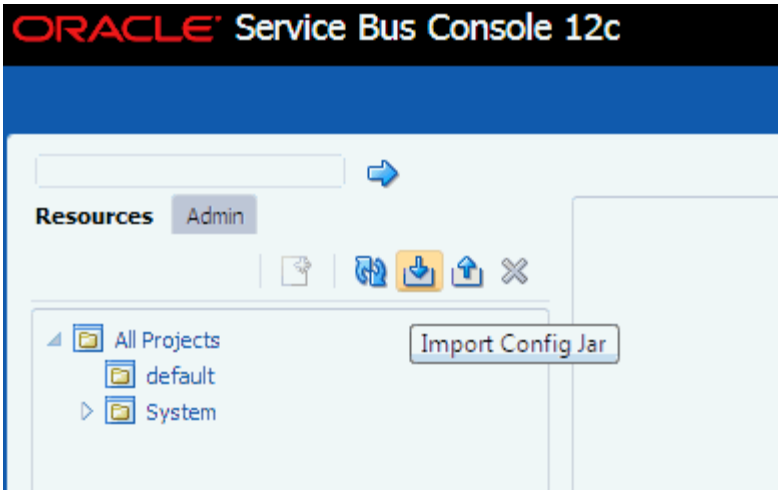
To import the exported processes to Oracle Service Bus 12*c* (12.2.1.0.0):

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`

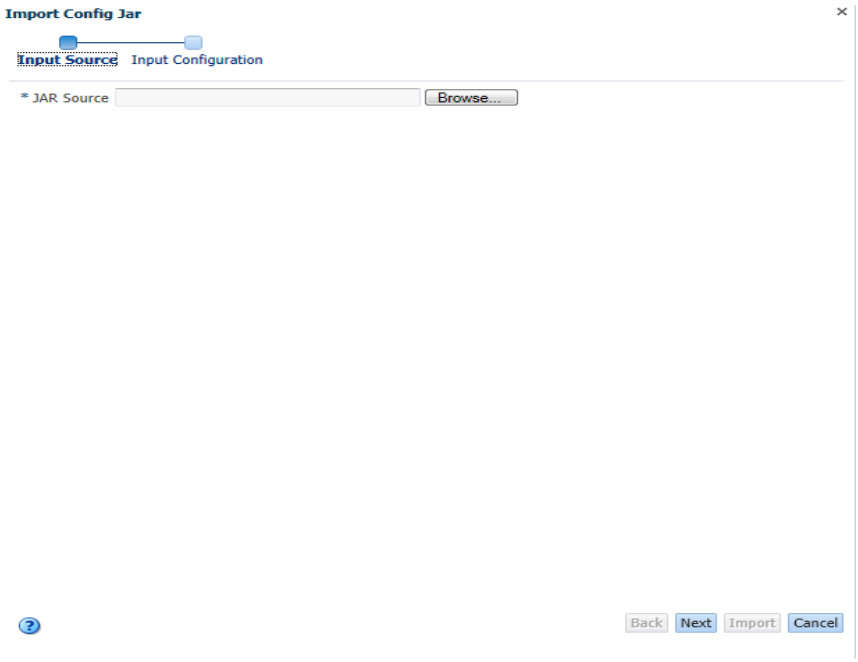
   where *host name* is the name of the system where Oracle WebLogic Server is running and *port* is the port for the domain you are using. The port for the default domain is 7001.
3. Log in to the Oracle Service Bus console using a valid user name and password.
4. Click **Create** in the top left pane to start a new Oracle Service Bus session.
5. Click the **Import Config Jar** icon in the left pane, as shown in Figure 2–9.

Figure 2–9 Import Config Jar Icon



The Import Config Jar window is displayed, as shown in Figure 2–10.

Figure 2–10 Import Config Jar Window



6. Click the **Browse** button and select the sbconfig.jar file, which was exported from Oracle Service Bus previous releases and copied to the local machine.
7. Click **Next**.
8. Verify that all imported items are selected in the Resource area by default, and click **Import**.

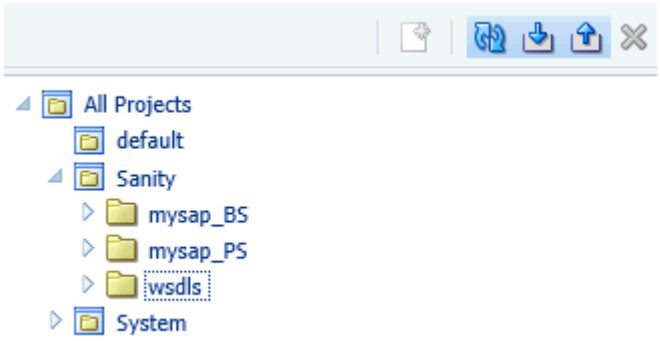
   The detailed status of imported items is displayed.
9. Make sure there are no issues and click **Close**.

   You are returned to the SB Console page.
10. Click the **Activate** button in the top right pane.

    The Confirm Session Activation window is displayed.
11. Click **Activate**.
12. Ensure that all project folders are imported and displayed in the left pane.

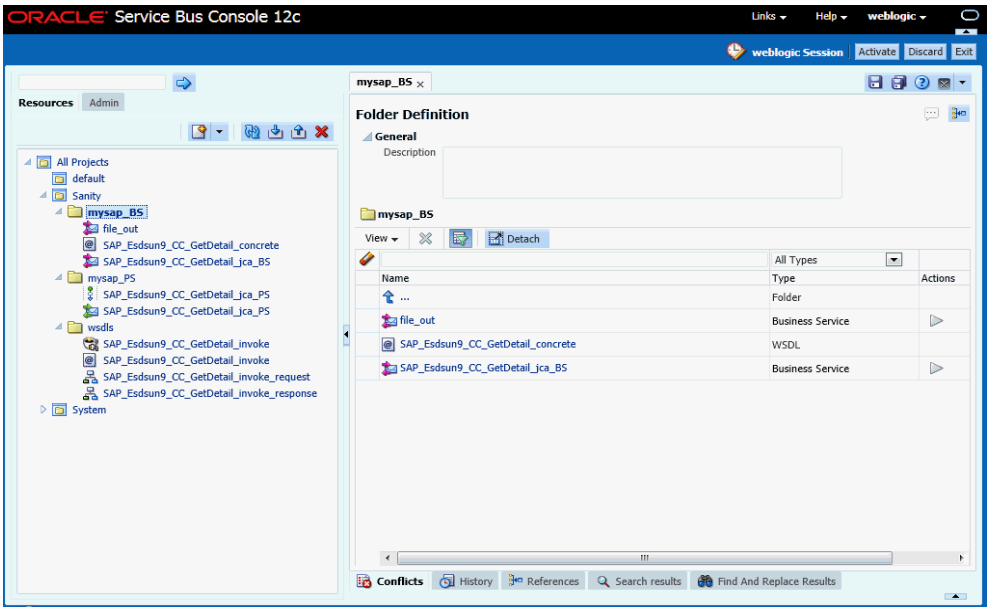
    The project folder structure must match the one from Oracle Service Bus 11g PS6, since you imported those settings from that environment.
13. Expand a project folder, for example, Sanity, as shown in Figure 2–11.

Figure 2–11 Expanded Sanity Project Folder



14. Expand folders for Business Service, Proxy Service and wsdls, as shown in Figure 2–12.

Figure 2–12 Expanded Folders



Ensure that all the Business Services, Proxy Services, and wsdls that were created in the Oracle Service Bus previous releases are included.

2.3.3 Additional Modifications for Imported Processes in Oracle Service Bus 12*c* (12.2.1.0.0)

This section describes additional modifications that are required for the imported processes in Oracle Service Bus 12*c* (12.2.1.0.0).

Note: Imported J2CA processes in Oracle Service Bus 12*c* (12.2.1.0.0) work properly and do not require any additional changes. Only BSE outbound processes require additional changes.

This section contains the following topics:

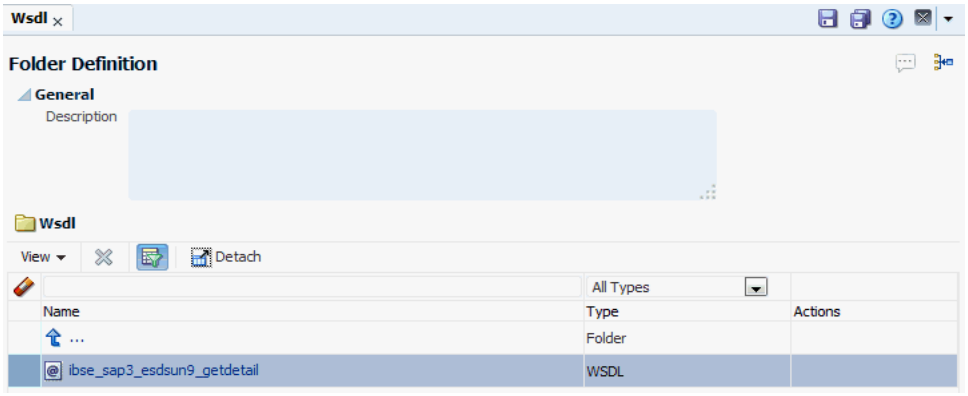
- Section 2.3.3.1, "Imported BSE Outbound Processes"

2.3.3.1 Imported BSE Outbound Processes

This section describes additional modifications that are required for imported BSE outbound processes in Oracle Service Bus 12*c* (12.2.1.0.0).

1. Select the folder that contains the WSDL files for your adapter in the Project folder.
2. Click the specific WSDL file for the outbound process that must be changed, as shown in Figure 2–13.

Figure 2–13 WSDL File Selected for the Outbound Process



3. Click the **View/Edit** source icon on the top right pane.

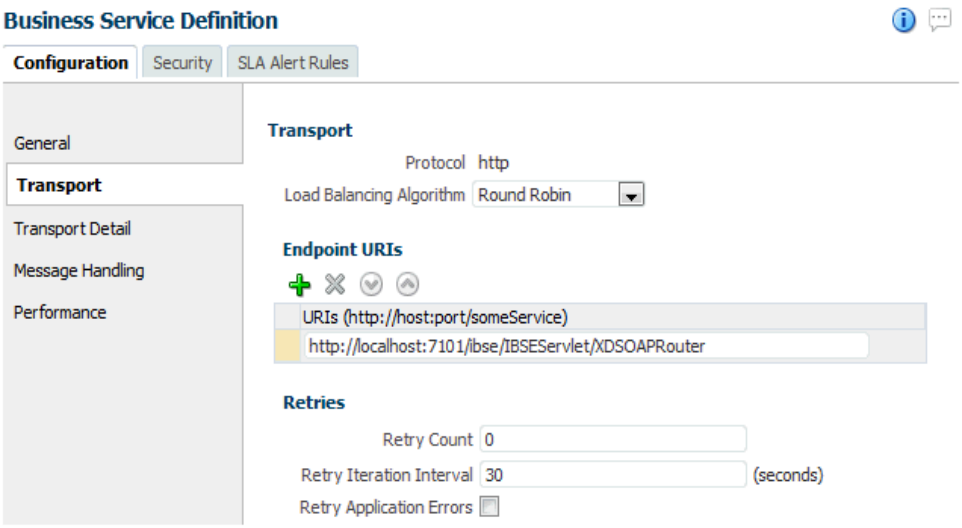
   The View/Edit source page is displayed, as shown in Figure 2–14.

Figure 2–14 The Edit a WSDL Source Page

4. Change the `<soap:address location>` element to point to the system and port number where OSB 12*c* (12.2.1.0.0) is running.
5. Click **Save**.
6. Select the Business Service that is created for the BSE outbound process in the Project Folder.
7. Click **Create** on the top right pane.

8. In the right pane, select the Transport tab and update the value for the Endpoint URIs property to point to the system and port number where OSB 12*c* (12.2.1.0.0) is running, as shown in Figure 2–15.

Figure 2–15 Transport Configuration Tab



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

Glossary

adapter

Provides universal connectivity by enabling an electronic interface to be accommodated (without loss of function) to another electronic interface.

agent

Supports service protocols in listeners and documents.

business service

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

channel

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

listener

A component that accepts requests from client applications.

port

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

Index