

Oracle® Business Intelligence Applications

Upgrade Guide for Informatica PowerCenter Users

Version 7.9.6

E14218-01

April 2009

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

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	ix
Audience	x
Documentation Accessibility	x
Related Documents	x
Conventions	xi
 1 What's New in This Release	
1.1 What's New in Oracle BI Applications Release 7.9.6	1-1
 2 Overview of Oracle BI Applications Upgrade	
2.1 Supported Upgrade Paths	2-1
2.2 Preupgrade Considerations	2-2
2.3 Best Practices for Preparing to Upgrade	2-2
2.4 Major Stages of Oracle BI Applications Upgrade	2-3
2.4.1 Sequence of Upgrade Stages	2-4
 Part I Upgrading When Your Source System is Siebel CRM	
 3 Upgrading Siebel Analytics 7.5.x	
3.1 Upgrading Oracle BI Infrastructure	3-1
3.2 Upgrading Oracle BI Applications	3-2
3.3 Upgrading to Informatica PowerCenter Version 8.6 and Hotfix 4	3-2
3.4 Upgrading the Informatica Repository	3-3
3.5 Configuring Informatica PowerCenter Version 8.6 to Work with DAC	3-4
3.6 Installing and Configuring DAC	3-4
3.7 Overview of Upgrading the Data Warehouse Schema and Migrating Data	3-4
3.8 Upgrading the Data Warehouse Schema to Version 7.7 and Migrating Data	3-5
3.9 Upgrading the Data Warehouse Schema to Version 7.9.0	3-8
3.10 Importing New Schema Definitions into the Siebel Transactional Database	3-10
3.10.1 Verifying the Siebel Transactional Database Upgrade	3-10
3.11 Migrating Data into the Data Warehouse Upgraded to Version 7.9.0	3-11
3.12 Upgrading the Data Warehouse Schema to Version 7.9.4 and Migrating Data	3-14
3.13 Upgrading the Data Warehouse Schema to Version 7.9.5 and Migrating Data	3-16
3.14 Upgrading the Data Warehouse Schema to Version 7.9.5.1	3-19

3.15	Upgrading the Data Warehouse Schema to Version 7.9.6 and Migrating Data	3-20
3.15.1	Upgrading the Data Warehouse Schema to Version 7.9.6.....	3-20
3.15.2	Migrating Data into the Upgraded Data Warehouse	3-22
3.15.3	Verifying the Data Migrated Successfully	3-24
3.15.4	Resetting Refresh Dates	3-24
3.16	Upgrading the Siebel Analytics Repository	3-25
3.16.1	Preparing for the Oracle BI Repository Upgrade.....	3-25
3.16.2	Equalizing the Oracle BI Repositories	3-26
3.16.3	Comparing the Siebel Analytics and Oracle BI Repositories	3-27
3.16.4	Merging the Siebel Analytics and Oracle BI Repositories	3-28
3.16.5	Regression Testing the Oracle BI Repository Merge	3-29
3.17	Upgrading the Oracle BI Presentation Catalog	3-29
3.17.1	Trimming the Input Presentation Catalog	3-30
3.17.2	Upgrading the Oracle BI Presentation Catalog to a Newer Version.....	3-30
3.17.3	Testing the Results of the Presentation Catalog Upgrade	3-31

4 Upgrading Siebel Business Analytics 7.7, 7.7.x and 7.8.x

4.1	Upgrading Oracle BI Infrastructure	4-1
4.2	Upgrading Oracle BI Applications	4-2
4.3	Upgrading to Informatica PowerCenter Version 8.6 and Hotfix 4.....	4-2
4.4	Upgrading the Informatica Repository	4-3
4.5	Configuring Informatica PowerCenter Version 8.6 to Work with DAC	4-4
4.6	Upgrading and Configuring DAC	4-4
4.6.1	Installing the DAC Platform and Oracle BI Applications Metadata Repository Files	4-4
4.6.2	Configuring the DAC Client and Server to Work with Oracle BI Applications and Informatica	4-5
4.6.3	Upgrading the DAC Repository for Siebel Business Analytics	4-5
4.7	Overview of Upgrading the Data Warehouse Schema and Migrating Data	4-8
4.8	Upgrading the Data Warehouse Schema to Version 7.9.0	4-9
4.9	Importing New Schema Definitions into the Siebel Transactional Database	4-11
4.9.1	Verifying the Siebel Transactional Database Upgrade.....	4-11
4.10	Migrating Data into the Data Warehouse Upgraded to Version 7.9.0	4-12
4.11	Upgrading the Data Warehouse Schema to Versions 7.9.4 and Migrating Data	4-16
4.12	Upgrading the Data Warehouse Schema to Version 7.9.5 and Migrating Data	4-18
4.13	Upgrading the Data Warehouse Schema to Version 7.9.5.1	4-21
4.14	Upgrading the Data Warehouse Schema to Version 7.9.6 and Migrating Data	4-22
4.14.1	Upgrading the Data Warehouse Schema to Version 7.9.6.....	4-22
4.14.2	Migrating Data into the Upgraded Data Warehouse	4-24
4.14.3	Verifying the Data Migrated Successfully	4-26
4.14.4	Resetting Refresh Dates	4-26
4.15	Upgrading the Siebel Analytics Repository	4-27
4.15.1	Preparing for the Oracle BI Repository Upgrade.....	4-27
4.15.2	Equalizing the Oracle BI Repositories	4-28
4.15.3	Comparing the Siebel Analytics and Oracle BI Repositories	4-29
4.15.4	Merging the Siebel Analytics and Oracle BI Repositories	4-30
4.15.5	Regression Testing the Oracle BI Repository Merge	4-31

4.16	Upgrading the Oracle BI Presentation Catalog	4-31
4.16.1	Trimming the Input Presentation Catalog	4-32
4.16.2	Upgrading the Oracle BI Presentation Catalog to a Newer Version.....	4-32
4.16.3	Testing the Results of the Presentation Catalog Upgrade	4-33

5 Upgrading Oracle BI Applications 7.9.x for the Siebel Sources

5.1	Upgrading Oracle BI Infrastructure	5-1
5.2	Upgrading Oracle BI Applications	5-1
5.3	Upgrading to Informatica PowerCenter Version 8.6 and Hotfix 4.....	5-2
5.4	Upgrading the Informatica Repository	5-3
5.5	Configuring Informatica PowerCenter Version 8.6 Work with Oracle BI Applications and DAC	5-4
5.6	Upgrading and Configuring DAC	5-4
5.6.1	Installing the DAC Platform and Oracle BI Applications Metadata Repository Files	5-4
5.6.2	Configuring the DAC Client and Server to Work with Oracle BI Applications and Informatica	5-4
5.6.3	Upgrading the DAC Repository	5-5
5.7	Importing New Schema Definitions into the Siebel Transactional Database	5-5
5.7.1	Verifying the Siebel Transactional Database Upgrade.....	5-6
5.8	Upgrading the Data Warehouse Schema and Migrating Data	5-6
5.8.1	Upgrading the Data Warehouse Schema to Version 7.9.4 and Migrating Data	5-7
5.8.2	Upgrading the Data Warehouse Schema to Version 7.9.5 and Migrating Data	5-8
5.8.3	Upgrading the Data Warehouse Schema to Version 7.9.5.1	5-12
5.8.4	Upgrading the Data Warehouse Schema to Version 7.9.6 and Migrating Data	5-12
5.8.4.1	Upgrading the Data Warehouse Schema to Version 7.9.6.....	5-12
5.8.4.2	Migrating Data into the Upgraded Data Warehouse.....	5-15
5.8.4.3	Verifying the Data Migrated Successfully	5-16
5.8.4.4	Resetting Refresh Dates	5-17
5.9	Upgrading the Oracle BI Repository.....	5-17
5.9.1	Preparing for the Oracle BI Repository Upgrade.....	5-18
5.9.2	Equalizing the Oracle BI Repositories	5-18
5.9.3	Comparing the Oracle BI Repositories	5-20
5.9.4	Merging the Oracle BI Repositories	5-20
5.9.5	Regression Testing the Oracle BI Repository Merge	5-21
5.10	Upgrading the Oracle BI Presentation Catalog	5-22
5.10.1	Trimming the Input Presentation Catalog	5-22
5.10.2	Upgrading the Oracle BI Presentation Catalog to a Newer Version.....	5-23
5.10.3	Testing the Results of the Presentation Catalog Upgrade	5-24

Part II Upgrading When Your Source System Is Oracle EBS

6 Upgrading Oracle BI Applications for Oracle Source Systems

6.1	Upgrading Oracle BI Infrastructure	6-1
6.2	Upgrading Oracle BI Applications	6-1
6.3	Upgrading to Informatica PowerCenter Version 8.6 and Hotfix 4.....	6-2

6.4	Upgrading the Informatica Repository	6-3
6.5	Configuring Informatica PowerCenter Version 8.6 to Work with Oracle BI Applications and DAC	6-3
6.6	Upgrading and Configuring DAC	6-4
6.6.1	Installing the DAC Platform and Oracle BI Applications Metadata Repository Files	6-4
6.6.2	Configuring the DAC Client and Server to Work with Oracle BI Applications and Informatica	6-4
6.6.3	Upgrading the DAC Repository	6-5
6.7	Upgrading the Data Warehouse Schema and Migrating Data	6-5
6.7.1	Performing Data Warehouse Schema Upgrade Steps Common to All Previous Releases	6-6
6.7.2	Upgrading the Data Warehouse Schema to Version 7.9.3 and Migrating Data	6-6
6.7.3	Upgrading the Data Warehouse Schema to Version 7.9.4 and Migrating Data	6-9
6.7.4	Upgrading the Data Warehouse Schema to Version 7.9.5. and Migrating Data	6-12
6.7.5	Upgrading the Data Warehouse Schema to Version 7.9.5.1 and Migrating Data	6-15
6.7.6	Upgrading the Data Warehouse Schema to Version 7.9.6 and Migrating Data	6-19
6.7.6.1	Upgrading the Data Warehouse Schema to Version 7.9.6.....	6-19
6.7.6.2	Migrating Data into the Upgraded Data Warehouse	6-21
6.7.6.3	Verifying the Data Migrated Successfully	6-22
6.7.6.4	Resetting Refresh Dates	6-23
6.8	Upgrading the Oracle BI Repository	6-24
6.8.1	Preparing for the Oracle BI Repository Upgrade.....	6-24
6.8.2	Equalizing the Oracle BI Repositories	6-25
6.8.3	Comparing the Oracle BI Repositories	6-26
6.8.4	Merging the Oracle BI Repositories	6-27
6.8.5	Regression Testing the Oracle BI Repository Merge	6-28
6.9	Upgrading the Oracle BI Presentation Catalog	6-28
6.9.1	Trimming the Input Presentation Catalog	6-29
6.9.2	Upgrading the Oracle BI Presentation Catalog to a Newer Version.....	6-29
6.9.3	Testing the Results of the Presentation Catalog Upgrade	6-30

Part III Upgrading When Your Source System is Peoplesoft

7 Upgrading Oracle BI Applications for the PeopleSoft Source Systems

7.1	Upgrading Oracle BI Infrastructure	7-1
7.2	Upgrading Oracle BI Applications	7-1
7.3	Upgrading to Informatica PowerCenter Version 8.6 and Hotfix 4.....	7-2
7.4	Upgrading the Informatica Repository	7-3
7.5	Configuring Informatica PowerCenter Version 8.6 Work with Oracle BI Applications and DAC	7-3
7.6	Upgrading and Configuring DAC	7-4
7.6.1	Installing the DAC Platform and Oracle BI Applications Metadata Repository Files	7-4
7.6.2	Configuring the DAC Client and Server to Work with Oracle BI Applications and Informatica	7-4
7.6.3	Upgrading the DAC Repository	7-5
7.7	Upgrading the Data Warehouse Schema and Migrating Data	7-5

7.7.1	Performing Data Warehouse Schema Upgrade Steps Common to All Previous Releases	7-6
7.7.2	Upgrading the Data Warehouse Schema to Version 7.9.4 and Migrating Data	7-7
7.7.3	Upgrading the Data Warehouse Schema to Version 7.9.5 and Migrating Data	7-9
7.7.4	Upgrading the Data Warehouse Schema to Version 7.9.5.1 and Migrating Data ...	7-12
7.7.5	Upgrading the Data Warehouse Schema to Version 7.9.6 and Migrating Data	7-15
7.7.5.1	Upgrading the Data Warehouse Schema to Version 7.9.6.....	7-16
7.7.5.2	Migrating Data into the Upgraded Data Warehouse.....	7-18
7.7.5.3	Verifying the Data Migrated Successfully	7-19
7.7.5.4	Resetting Refresh Dates	7-19
7.8	Upgrading the Oracle BI Repository.....	7-20
7.8.1	Preparing for the Oracle BI Repository Upgrade.....	7-21
7.8.2	Equalizing the Oracle BI Repositories	7-21
7.8.3	Comparing the Oracle BI Repositories	7-23
7.8.4	Merging the Oracle BI Repositories	7-23
7.8.5	Regression Testing the Oracle BI Repository Merge	7-24
7.9	Upgrading the Oracle BI Presentation Catalog	7-25
7.9.1	Trimming the Input Presentation Catalog.....	7-25
7.9.2	Upgrading the Oracle BI Presentation Catalog to a Newer Version.....	7-25
7.9.3	Testing the Results of the Presentation Catalog Upgrade	7-26

Part IV Appendices

A Upgrading the Oracle BI Repository for Industry-Specific Analytics Applications

A.1	Common Dimensions.....	A-1
A.2	Merging Siebel Analytics and Oracle BI Repositories.....	A-2
A.2.1	Creating Working Folders.....	A-4
A.2.2	Trimming Repository Files.....	A-4
A.2.2.1	Trimming the Original Base Repository Files	A-5
A.2.2.2	Trimming the New Base Repository File	A-5
A.2.2.3	Trimming the Custom Repository Files	A-5
A.2.3	Renaming Objects in the Original Pharma Base Repository File.....	A-5
A.2.4	Equalizing the Oracle BI Repositories	A-6
A.2.5	Merging the Core Repository Files	A-7
A.2.6	Merging the Pharma Repository Files	A-8
A.2.7	Making Merge Decisions in the Server Admin Tool	A-9
A.3	Replacing Common Dimensions After the Repository Merge.....	A-9

B Configuring Common Parameters for Upgrading to Oracle BI Applications 7.9.6

B.1	Configuring Common Parameters for Oracle Source Systems	B-1
B.1.1	Configuring Common Parameters for Oracle EBS 11i Source Systems.....	B-1
B.1.2	Configuring Common Parameters for Oracle EBS R12 Source Systems	B-2
B.2	Configuring Common Parameters for Siebel Source Systems	B-2

C Configuring Application-Specific Parameters and Mappings

C.1	Configuring Parameters and Mappings for Oracle Financial Analytics	C-1
C.1.1	Configuring Parameters for Value Set Hierarchies and FSG Hierarchies.....	C-1
C.1.2	Configuring Parameters for GL Data Extraction	C-2
C.1.2.1	Configuring GL Data Extraction Parameters for Oracle EBS 11i Sources.....	C-2
C.1.2.2	Configuring GL Data Extraction Parameters for Oracle EBS R12 Sources	C-2
C.1.3	Configuring the COGS Fact Mapping for Oracle EBS R12.....	C-3
C.1.4	Configuring the \$\$Hint1 Parameter for Oracle Databases	C-3
C.2	Configuring Parameters for Oracle Supply Chain and Order Management Analytics ..	C-4
C.2.1	Configuring the TIME_GRAIN Parameter for Sales Order Lines Aggregate Fact and Invoice Lines Aggregate Fact Tables	C-4
C.2.2	Configuring the \$\$PERIOD Parameter for the Customer Status History Fact Table	C-4
C.3	Configuring Parameters for Oracle Projects Analytics.....	C-4

D Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File

D.1	Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File.....	D-1
-----	---	-----

Index

Preface

Oracle Business Intelligence Applications are comprehensive prebuilt solutions that deliver pervasive intelligence across an organization, empowering users at all levels — from front line operational users to senior management — with the key information they need to maximize effectiveness. Intuitive and role-based, these solutions transform and integrate data from a range of enterprise sources, including Siebel, Oracle, PeopleSoft, JD Edwards, and corporate data warehouses — into actionable insight that enables more effective actions, decisions, and processes.

Oracle BI Applications are built on Oracle Business Intelligence Suite Enterprise Edition, a comprehensive next-generation BI and analytics platform.

Oracle BI Applications includes the following:

- Oracle Financial Analytics
- Oracle Human Resources Analytics
- Oracle Supply Chain and Order Management Analytics
- Oracle Procurement and Spend Analytics
- Oracle Project Analytics
- Oracle Sales Analytics
- Oracle Service Analytics
- Oracle Contact Center Telephony Analytics
- Oracle Marketing Analytics
- Oracle Loyalty Analytics
- Oracle Price Analytics
- Oracle Pharma Marketing Analytics
- Oracle Pharma Sales Analytics

For more details on the applications included in this release of Oracle BI Applications, see the *Oracle Business Intelligence Applications Licensing and Packaging Guide*. This guide is included in the Oracle Business Intelligence Media Pack. Also, see the *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*, available at available at

http://www.oracle.com/technology/documentation/bi_apps.html.

The *Oracle Business Intelligence Applications Upgrade Guide for Informatica PowerCenter Users* is part of the documentation set for Oracle BI Applications. This guide contains information about upgrading to Oracle BI Applications from different versions of Siebel Business Analytics and Oracle Business Intelligence Applications.

Oracle recommends reading the *Oracle Business Intelligence Applications Release Notes* before installing, using, or upgrading Oracle Business Intelligence Applications. The most current version of the *Oracle Business Intelligence Applications Release Notes* is available:

- On the Oracle Technology Network at
http://www.oracle.com/technology/documentation/bi_apps.html
To register for a free account on the Oracle Technology Network, go to
<http://www.oracle.com/technology/about/index.html>.

Audience

This document is intended for BI managers and implementors of Oracle BI Applications.

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.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at <http://www.fcc.gov/cgb/consumerfacts/trs.html>, and a list of phone numbers is available at <http://www.fcc.gov/cgb/dro/trsphonebk.html>.

Related Documents

For more information, see the following Oracle BI Applications documentation set (available at http://www.oracle.com/technology/documentation/bi_apps.html):

- *Oracle Business Intelligence Applications Release Notes*

- *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*
- *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*
- *Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users*
- *Oracle Business Analytics Warehouse Data Model Reference*
- *Oracle Business Intelligence Applications Security Guide*

Also see the Oracle Business Intelligence Data Warehouse Administration Console documentation set (available at

http://www.oracle.com/technology/documentation/bi_dac.html):

- *Oracle Business Intelligence Data Warehouse Administration Console Release Notes*
- *System Requirements and Supported Platforms for Oracle Business Intelligence Data Warehouse Administration Console*
- *Oracle Business Intelligence Data Warehouse Administration Console User's Guide*
- *Oracle Business Intelligence Data Warehouse Administration Console Installation, Configuration and Upgrade Guide*

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.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

What's New in This Release

This section lists changes in this version of the documentation to support release 7.9.6 of the software.

Note: Some of the information about database platforms and source systems might not apply to this version of Oracle Business Intelligence Applications. For up-to-date information about supported databases and source systems in this version of Oracle Business Intelligence Applications, make sure you read *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*. Make sure that you also read the *Oracle Business Intelligence Applications Release Notes*. The most up-to-date versions of these documents are located on the Oracle Technology Network at http://www.oracle.com/technology/documentation/bi_apps.html. To register for a free account on the Oracle Technology Network, go to <http://www.oracle.com/technology/about/index.html>.

1.1 What's New in Oracle BI Applications Release 7.9.6

This guide includes the following changes:

- **Informatica PowerCenter version 8.6 and Hotfix 4.** All chapters related to upgrading Oracle BI Applications were updated to provide instructions for upgrading to Informatica PowerCenter Version 8.6 and Hotfix4.
- **DAC platform.** All chapters related to upgrading Oracle BI Applications were updated to provide instructions for upgrading the DAC platform.

Note: As of the Oracle BI Applications release 7.9.6, the DAC is no longer installed by the Oracle BI Applications installer.
- **Data warehouse schema.** All chapters related to upgrading Oracle BI Applications were updated to provide instructions for upgrading the data warehouse schema to version 7.9.6 and migrating data into the upgraded data warehouse.
- **Common parameters.** [Appendix B, "Configuring Common Parameters for Upgrading to Oracle BI Applications 7.9.6"](#) was added to provide instructions for configuring common parameters.
- **Application-specific parameters.** [Appendix C, "Configuring Application-Specific Parameters and Mappings"](#) was added to provide application-specific instructions for configuring application-specific parameters.

Overview of Oracle BI Applications Upgrade

This chapter provides an overview of the Oracle BI Applications upgrade process. It includes the following topics:

- [Section 2.1, "Supported Upgrade Paths"](#)
- [Section 2.2, "Preupgrade Considerations"](#)
- [Section 2.3, "Best Practices for Preparing to Upgrade"](#)
- [Section 2.4, "Major Stages of Oracle BI Applications Upgrade"](#)

2.1 Supported Upgrade Paths

The Oracle Business Intelligence Applications Upgrade Guide for Informatica PowerCenter Users provides instructions for upgrading to the current release of Oracle BI Applications from the previous releases listed below:

- **Siebel Analytics 7.5.3**

For instructions, see [Chapter 3, "Upgrading Siebel Analytics 7.5.x."](#)

- **Siebel Business Analytics 7.7, 7.7.x and 7.8.x**

For instructions, see [Chapter 4, "Upgrading Siebel Business Analytics 7.7, 7.7.x and 7.8.x."](#)

Note: Upgrade paths to the current release of Oracle BI Applications from Siebel Business Analytics Applications 6.x, 7.x, and Oracle Business Intelligence 7.9.x with Siebel Relationship Management Warehouse or Oracle Business Analytics Warehouse on NCR Teradata database are currently NOT available.

- **Siebel Business Analytics for Life Sciences 7.8.0, 7.8.1 and 7.8.2**

For instructions, see [Chapter 4, "Upgrading Siebel Business Analytics 7.7, 7.7.x and 7.8.x."](#)

- **Oracle Business Intelligence Applications 7.9.0 through 7.9.5.1**

For instructions, see one of the following, depending on your source system:

- [Chapter 5, "Upgrading Oracle BI Applications 7.9.x for the Siebel Sources"](#)
- [Chapter 6, "Upgrading Oracle BI Applications for Oracle Source Systems"](#)
- [Chapter 7, "Upgrading Oracle BI Applications for the PeopleSoft Source Systems"](#)

2.2 Preupgrade Considerations

You should consider the following points before you begin your upgrade:

- Review the *System Requirements and Supported Platforms for Oracle Business Intelligence Applications* to make sure your database platform versions are supported.
- The upgrade process is not only a technical process. You will also need to analyze the impact of the schema changes on your current custom implementation. The extent of your customizations will have an impact on the length of time required for the upgrade.

In addition, at different stages of the upgrade you will also need to analyze your organization's business requirements. For example, at the stage in which you are upgrading the Oracle Business Intelligence repository, you will need to know which repository objects are not in use so that they can be trimmed from the repository. To determine this information, you will need to know what reports are in use.

- It is highly recommended that you use side-by-side environments when performing each stage of the upgrade process. Enabling side-by-side instances of the entire Oracle Business Intelligence Applications environment is a critical success factor for upgrade.

For the other stages of the upgrade, you can upgrade your environments in place. However, for comparison and benchmarking purposes, it is recommended that you upgrade using side-by-side environments.

- You should determine what customizations were made to your existing DAC Repository before you begin the upgrade process. Depending on the release from which you are upgrading, moving the customizations in your existing DAC Repository into the new DAC Repository may be a manual process.
- For each stage of the upgrade process, you need to allocate a reasonable amount of time to validate the results of that stage and address any problems. In addition, final user acceptance testing must confirm that the entire upgrade process was successful.
- When you move from a development environment to test or production environments, you must perform the following stages of the upgrade process for each environment:
 - Upgrade the platform
 - Upgrade the applications
 - Migrate the data

You can migrate the DAC, Informatica, and Oracle BI repositories into the production environment after you merge and test them.

For a description of the major stages of the upgrade process, see [Section 2.4, "Major Stages of Oracle BI Applications Upgrade."](#)

2.3 Best Practices for Preparing to Upgrade

This section provides best practice steps you need to perform before you begin the upgrade process.

- Review this guide, *Oracle Business Intelligence Applications Upgrade Guide for Informatica PowerCenter Users*, in its entirety.

- Rename and copy into a different location your current repositories, Presentation Catalog (formerly known as the Web Catalog), and DAC installation folder from your existing environment. The file names and locations listed below represent the standard names and locations of files that shipped with previous versions of Siebel Business Analytics Applications and Oracle BI Applications.

When you backup the DAC Repository, you export the DAC metadata, in XML format (using the DAC's Export tool), into a different database, which can be used later as a reference. For instructions, see the *Oracle Business Intelligence Data Warehouse Administration Console User's Guide*.

Type	Name	Location
DAC metadata repository	*.xml	OracleBI\DAC\export
Oracle BI repository (formerly called Analytics repository)	OracleBIAnalyticsApps.rpd	OracleBI\Repository
Informatica Repository	Oracle_BI_DW_Base.rep	OracleBI\dwrep\Informatica\Repository
Presentation Catalog (formerly called Web Catalog)	<file_name>.webcat	OracleBIData\web\catalog
DAC installation folder	DAC	OracleBI\DAC

- Gather information that describes the current implementation, including the following:
 - Version of the transactional database to which you upgraded.
 - Version of the Oracle BI Infrastructure (platform) or Siebel Business Analytics platform from which you are upgrading.

To determine the platform version from which you are upgrading, launch the Administration Tool, and click Help, and then click Administration Tool in the toolbar. A message box provides the platform version.

- Version of Siebel Business Analytics Applications or Oracle BI Applications from which you are upgrading.

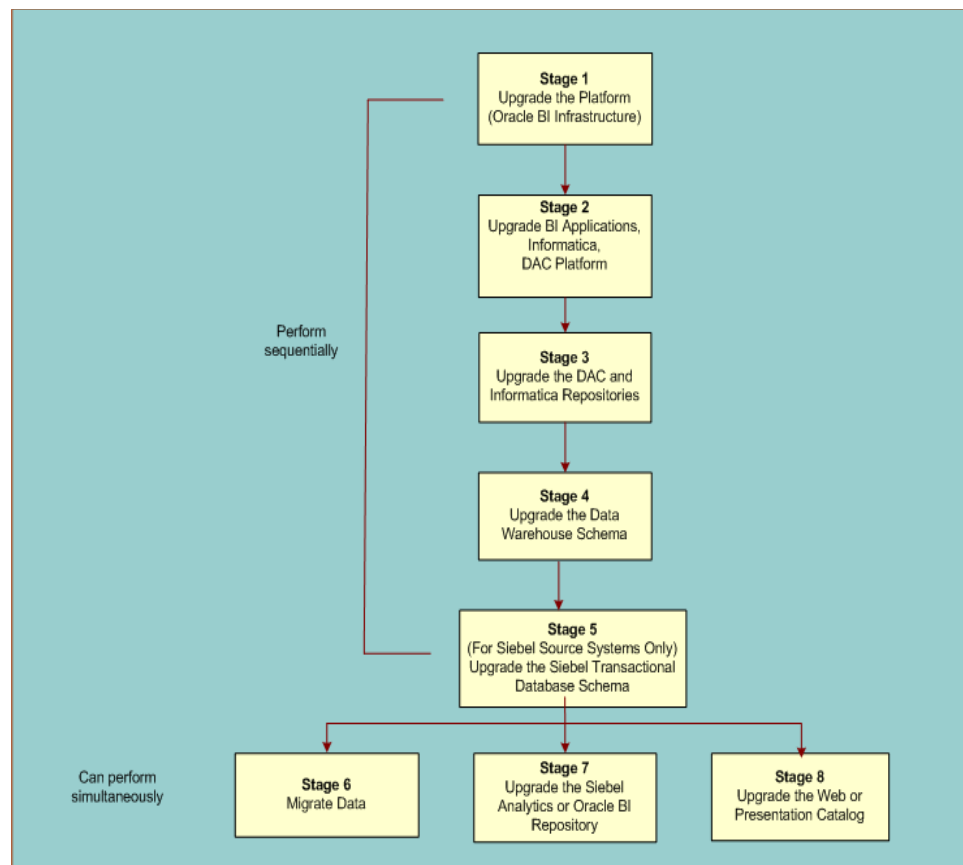
To determine the version of Siebel Business Analytics Applications or Oracle BI Applications from which you are upgrading, launch the Data Warehouse Administration Console (DAC), and click Help, and then click About DAC.

Note: If DAC is not a component of your Analytics installation, you are running Siebel Analytics 7.5.x.

2.4 Major Stages of Oracle BI Applications Upgrade

Figure 2–1 shows the upgrade flow for the major stages of the Siebel Business Analytics and Oracle BI Applications upgrade process. Depending on the prior release from which you are upgrading, you may find minor changes in the process shown in Figure 2–1.

Table 2–1 provides the documentation you should refer to for each stage of the upgrade.

Figure 2–1 Oracle BI Applications Upgrade Flow

2.4.1 Sequence of Upgrade Stages

In general, you must complete stages one through five sequentially, in the following order:

- Stage 1: Upgrade the Siebel Business Analytics platform or Oracle BI Infrastructure to the current release of Oracle BI Infrastructure.
- Stage 2: Upgrade the following:
 - Siebel Business Analytics Applications or Oracle BI Applications to the current release of Oracle BI Applications
 - Informatica PowerCenter
 - DAC Platform
- Stage 3: Upgrade the metadata in the DAC and Informatica repositories.
- Stage 4: Upgrade the Oracle Business Analytics Warehouse schema.
- Stage 5: (For Siebel source systems) upgrade the Siebel transactional database schema.

You can perform stages six through eight simultaneously; that is, you can migrate data into the upgraded data warehouse (stage six), upgrade the Siebel Analytics or Oracle BI repository (stage seven), and upgrade the Web (Presentation) catalog (stage eight) at the same time. However, you cannot test the results of your Web (Presentation) catalog upgrade until you have finished migrating your data.

Note: Depending on the prior release from which you are upgrading, you may find minor changes in the process described in this section.

Table 2–1 Major Stages of the Oracle BI Applications Upgrade and Related Documentation

Stage	Related Documentation
1. Upgrade the Siebel Business Analytics Platform or Oracle BI Infrastructure to the current release of the Oracle BI Infrastructure.	<ul style="list-style-type: none"> ■ <i>Oracle Business Intelligence Infrastructure Upgrade Guide</i>
Note: The Oracle Business Intelligence Infrastructure is code-independent from the Oracle BI Applications.	
2. Upgrade following: <ul style="list-style-type: none"> - Siebel Business Analytics Applications or Oracle BI Applications to the current release of the Oracle Business Intelligence Applications. - Informatica PowerCenter. - DAC platform. 	<ul style="list-style-type: none"> ■ <i>Oracle Business Intelligence Applications Upgrade Guide for Informatica PowerCenter Users</i>
3. Upgrade the DAC and Informatica repositories.	<ul style="list-style-type: none"> ■ <i>Oracle Business Intelligence Applications Upgrade Guide for Informatica PowerCenter Users</i> ■ <i>Oracle Business Intelligence Data Warehouse Administration Console User's Guide</i>
4. Upgrade the data warehouse schema.	<ul style="list-style-type: none"> ■ <i>Oracle Business Intelligence Applications Upgrade Guide for Informatica PowerCenter Users</i>
5. Upgrade the Siebel transactional database schema.	<ul style="list-style-type: none"> ■ <i>Oracle Business Intelligence Applications Upgrade Guide for Informatica PowerCenter Users</i>
6. Migrate data into the upgraded data warehouse.	<ul style="list-style-type: none"> ■ <i>Oracle Business Intelligence Applications Upgrade Guide for Informatica PowerCenter Users</i>
7. Upgrade the Siebel Analytics repository or Oracle BI repository to the current release of the Oracle BI repository.	<ul style="list-style-type: none"> ■ <i>Oracle Business Intelligence Applications Upgrade Guide for Informatica PowerCenter Users</i> ■ <i>Oracle Business Intelligence Server Administration Guide</i>
8. Upgrade the Siebel Business Analytics Web Catalog or Oracle BI Presentation Catalog to the current release of the Oracle BI Presentation Catalog.	<ul style="list-style-type: none"> ■ <i>Oracle Business Intelligence Applications Upgrade Guide for Informatica PowerCenter Users</i> ■ <i>Oracle Business Intelligence Infrastructure Upgrade Guide</i> ■ <i>Oracle Business Intelligence Presentation Services Administration Guide</i>

Caution: It is highly recommended that you consider obtaining support from Professional Services to assist with the end-to-end upgrade process.

Note: Upgrading Oracle's Siebel Customer Relationship Management applications is out of the scope of this document. Upgrading your current version of Siebel CRM applications is optional.

Part I

Upgrading When Your Source System is Siebel CRM

Part I contains instructions for upgrading to the current release of Oracle BI Applications from previous releases of Oracle BI Applications, Siebel Business Analytics, and Siebel Analytics.

Part I contains the following sections:

- [Chapter 3, "Upgrading Siebel Analytics 7.5.x"](#)
Follow the instructions in this section for upgrading to the current release of Oracle BI Applications from Siebel Analytics 7.5.x
- [Chapter 4, "Upgrading Siebel Business Analytics 7.7, 7.7.x and 7.8.x"](#)
Follow the instructions in this section for upgrading to the current release of Oracle BI Applications from Siebel Business Analytics 7.7, 7.7.x, and 7.8.x.
- [Chapter 5, "Upgrading Oracle BI Applications 7.9.x for the Siebel Sources"](#)
Follow the instructions in this section for upgrading to the current release from previous releases of Oracle BI Applications if your source system is Siebel CRM.

Note: Some of the information about database platforms and source systems might not apply to this version of Oracle Business Intelligence Applications. For up-to-date information about supported databases and source systems in this version of Oracle Business Intelligence Applications, make sure you read *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*. Make sure that you also read the *Oracle Business Intelligence Applications Release Notes*. The most up-to-date versions of these documents are located on the Oracle Technology Network at http://www.oracle.com/technology/documentation/bi_apps.html. To register for a free account on the Oracle Technology Network, go to <http://www.oracle.com/technology/about/index.html>.

Upgrading Siebel Analytics 7.5.x

This section contains instructions for upgrading Oracle's Siebel Analytics release 7.5.x.

This section includes the following topics:

- [Section 3.1, "Upgrading Oracle BI Infrastructure"](#)
- [Section 3.2, "Upgrading Oracle BI Applications"](#)
- [Section 3.3, "Upgrading to Informatica PowerCenter Version 8.6 and Hotfix 4"](#)
- [Section 3.4, "Upgrading the Informatica Repository"](#)
- [Section 3.5, "Configuring Informatica PowerCenter Version 8.6 to Work with DAC"](#)
- [Section 3.6, "Installing and Configuring DAC"](#)
- [Section 3.7, "Overview of Upgrading the Data Warehouse Schema and Migrating Data"](#)
- [Section 3.8, "Upgrading the Data Warehouse Schema to Version 7.7 and Migrating Data"](#)
- [Section 3.9, "Upgrading the Data Warehouse Schema to Version 7.9.0"](#)
- [Section 3.10, "Importing New Schema Definitions into the Siebel Transactional Database"](#)
- [Section 3.11, "Migrating Data into the Data Warehouse Upgraded to Version 7.9.0"](#)
- [Section 3.12, "Upgrading the Data Warehouse Schema to Version 7.9.4 and Migrating Data"](#)
- [Section 3.13, "Upgrading the Data Warehouse Schema to Version 7.9.5 and Migrating Data"](#)
- [Section 3.14, "Upgrading the Data Warehouse Schema to Version 7.9.5.1"](#)
- [Section 3.15, "Upgrading the Data Warehouse Schema to Version 7.9.6 and Migrating Data"](#)
- [Section 3.16, "Upgrading the Siebel Analytics Repository"](#)
- [Section 3.17, "Upgrading the Oracle BI Presentation Catalog"](#)

3.1 Upgrading Oracle BI Infrastructure

Upgrade the Oracle BI Infrastructure to the version that is supported for this release of Oracle BI Applications. See the *System Requirements and Supported Platforms for Oracle Business Intelligence Applications* for the current version that is supported. For

information on installing the supported version of Oracle BI Infrastructure, see the *Oracle Business Intelligence Infrastructure Upgrade Guide*.

3.2 Upgrading Oracle BI Applications

Run the Oracle BI Applications installer to upgrade your Oracle BI Applications environment to the current version. For instructions on running the installer, see *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

Note: Perform only the step that addresses the running of the installer. Do not perform the configuration instructions that follow the running of the installer.

Note: If you have a previous release of Oracle BI Applications installed, you must uninstall it before you run the installer for the current release. If you do not uninstall the old release, some folders from the current release will not be correctly installed.

3.3 Upgrading to Informatica PowerCenter Version 8.6 and Hotfix 4

You must install Informatica PowerCenter 8.6 and Hotfix 4 to run the current version of Oracle BI Applications.

The components and architecture for Informatica PowerCenter 8.6 differ significantly from PowerCenter 7.x versions. Oracle recommends that you carefully review the Informatica PowerCenter 8.6 documentation, which is included on the Informatica DVD provided with Oracle BI Applications.

For a summary of installation instructions for installing Informatica PowerCenter 8.6 on a single machine in an Oracle BI Applications deployment, see the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

For detailed information about deploying Informatica PowerCenter 8.6, refer to the *Informatica PowerCenter Installation Guide*, *Informatica PowerCenter Configuration Guide*, *Informatica PowerCenter Administrator Guide*, and related documentation.

For information about applying Hotfix 4, see *PowerCenter Version 8.6.0 Hotfix 4 Release Notes*. This PDF is included with the Informatica documentation on the Informatica PowerCenter DVD and is also included with the Hotfix 4 installation

To upgrade to Informatica PowerCenter 8.6 and Hotfix 4

1. Perform the pre-upgrade steps documented in the *Informatica PowerCenter Installation Guide*.
2. Install Informatica PowerCenter version 8.6 and Hotfix 4, by following the instructions in the *Informatica PowerCenter Installation Guide* and *PowerCenter Version 8.6.0 HotFix 4 Release Notes*.
3. Perform the post-upgrade steps documented in the *Informatica PowerCenter Installation Guide*.

Note: The Informatica PowerCenter 8.6 upgrade process includes upgrading your current Informatica Repository to the version 8.6 format. This process is necessary so that you will be able to access your current repository using version 8.6 client tools so that you can perform the procedure in [Section 3.4, "Upgrading the Informatica Repository."](#)

In [Section 3.4, "Upgrading the Informatica Repository,"](#) you back up and rename your current repository and then restore the Informatica Repository (Oracle_BI_DW_Base.rep) that is installed during the Oracle BI Applications installation. You then copy your custom folder from the backed up repository into the newly restored Oracle_BI_DW_Base repository.

3.4 Upgrading the Informatica Repository

Follow this procedure to upgrade the Informatica Repository.

For detailed instructions on backing up and restoring the Informatica Repository, see the topic titled, "Backing Up and Restoring the Repository," in "Chapter 8: Managing the Repository," in the *Informatica PowerCenter Administrator Guide*, which is included on the Informatica DVD provided with Oracle BI Applications.

To upgrade the Informatica Repository

1. Make sure you have backed up and renamed your current Informatica Repository.

Note: This repository must be upgraded to the version 8.6 format during the procedure in [Section 3.3, "Upgrading to Informatica PowerCenter Version 8.6 and Hotfix 4."](#) You must upgrade this repository to the version 8.6 format in order to move your custom folder from this repository into the new Oracle_BI_DW_Base repository that you restore in the steps below.

2. Copy the Oracle_BI_DW_Base.rep file from the folder OracleBI\dwrep\Informatica\Repository into the folder \Informatica\PowerCenter 8.6\server\infa_shared\Backup.

Note: The Oracle_BI_DW_Base.rep file is installed in the OracleBI root directory when you run the Oracle BI Applications installer, as described in [Section 3.2, "Upgrading Oracle BI Applications."](#)

3. Restore the Oracle_BI_DW_Base.rep repository.
4. Copy the custom folder from your previous Informatica Repository to the newly created Informatica Repository:
 - a. Launch the Informatica PowerCenter Repository Manager, and connect to both your previous and newly created Informatica repositories.
 - b. Copy the Custom folder in your previous repository into the newly created Informatica Repository.
 - c. Make sure there is an individual workflow for each of the mappings in the Custom folder.

3.5 Configuring Informatica PowerCenter Version 8.6 to Work with DAC

Informatica PowerCenter version 8.6 requires additional configuration steps to work with Oracle BI Applications and DAC. For instructions on performing these steps, see the following sections in the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*:

- Copying Source Files and Lookup Files
- Setting PowerCenter Integration Services Relaxed Code Page Validation
- Setting PowerCenter Integration Services Custom Properties
- Creating the Repository Administrator User in the Native Security Domain

3.6 Installing and Configuring DAC

Siebel Analytics release 7.5.x does not include the Data Warehouse Administration Console (DAC). In Oracle BI Applications release 7.9.6, you will use the DAC Client and Server and the DAC Repository. To install and configure DAC, follow the instructions in the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*. Make sure you perform the tasks in the following sections of the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*:

- Preinstallation Tasks
- Installing the DAC Platform
- Installing DAC Metadata Files
- Logging into DAC for the First Time and Importing Metadata into the DAC Repository

Note: After you complete this procedure, you will have the default DAC Repository for Oracle BI Applications version 7.9.6.

- Configuring the DAC Server
- Configuring DAC Integration Settings
- Configuring Relational Connections in Informatica Workflow Manager
- Configuring the SiebelUnicodeDB Custom Property
- Setting Up DAC to Receive Email Notification
- Additional Configuration Tasks

In this section, perform only the tasks that apply to your environment.

3.7 Overview of Upgrading the Data Warehouse Schema and Migrating Data

The process for upgrading the data warehouse schema and migrating data for Siebel Business Analytics releases 7.5.x involves multiple phases, as described below.

Note: The Oracle BI Applications upgrade logic is coded such that you must complete each of the phases listed below. To upgrade to the current release of Oracle BI Applications, you cannot skip any of the phases.

- **Phase 1**

In phase 1, you will upgrade the data warehouse schema to version 7.7 and migrate data into the upgraded data warehouse. For instructions, see [Section 3.8](#).

- **Phase 2**

In phase 2, you will perform the following steps:

1. Upgrade the data warehouse schema to version 7.9.0. For instructions, see [Section 3.9](#).
2. Import new schema definitions into the Siebel transactional database. For instructions, see [Section 3.10](#).
3. Migrate data into the data warehouse upgraded to version 7.9.0. For instructions, see [Section 3.11](#).

- **Phase 3**

In phase 3, you will upgrade your data warehouse schema to version 7.9.4 and migrate data into the upgraded data warehouse. For instructions, see [Section 3.12](#).

- **Phase 4**

In phase 4, you will upgrade your data warehouse schema to version 7.9.5 and migrate data into the upgraded data warehouse. For instructions, see [Section 3.13](#).

- **Phase 5**

In phase 5, you will upgrade your data warehouse schema to version 7.9.5.1. For instructions, see [Section 3.14](#).

- **Phase 6**

In phase 6, you will upgrade your data warehouse schema to version 7.9.6 and migrate data. For instructions, see [Section 3.15](#).

3.8 Upgrading the Data Warehouse Schema to Version 7.7 and Migrating Data

This procedure adds new tables, columns, and indexes to the existing data warehouse schema. It also modifies the existing data warehouse schema objects.

To upgrade the data warehouse schema

1. Run the schema upgrade script.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\CTLFiles, and locate the ddlsme_HOR_77.ctl file (Horizontal) or ddlsme_SIA_77.ctl file (Vertical).
 - b. If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)
 - c. Use the DDLimp utility to run one of the following scripts:

For Siebel Applications (Horizontal), run ddlsme_HOR_77.ctl.

For Siebel Industry Applications (Vertical), run ddlsme_SIA_77.ctl.

Use the following command, substituting the correct script name where appropriate.

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddlsme_HOR_77.ctl>
/L <..\oracleBI\dwrep\ddlsme_HOR_77.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C SIEBEL_OLTP /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\ddlsme_HOR_77.ctl /L
C:\OracleBI\dwrep\ddlsme_HOR_77.log
```

Notes:

- * /P <PASSWORD> - The password for the data warehouse.
- * /C <ODBC connect string> - The name of the ODBC connect string.
- * /I N - Tells DDLimp to ignore the indexes DDL defined in the CTL file if any exist. It does not change existing indexes. (The default is /I Y, which tells DDLimp to create and merge indexes from the CTL file with the indexes in the database.
- * For Oracle databases, use the Data Direct drivers.
- * In addition, you can use the following commands:
- * /W Y - If the OLAP database is Oracle and Unicode.
- * /Z Y - If the OLAP database is DB2 or SQL Server and Unicode.
- * /B <TABLE_SPACE_NAME> - If you want to create these tables in a separate table space. For DB2, This must be specified as 32K tablespace.
- * /X <INDEX_TABLE_SPACE_NAME> - If you want to create the indexes in a separate tablespace. For DB2, This must be specified as 32K tablespace.
- * /Y - Storage File for DB2/390.
- * /R - Regrant tables.

2. Copy the file Upgrade.rep file from the folder OracleBI\dwrep\Upgrade\Informatica\Repository into the folder Informatica PowerCenter 8.6\server\infa_shared\Backup.
3. Launch the Informatica PowerCenter Administration Console and restore Upgrade.rep (located in Informatica PowerCenter 8.6\server\infa_shared\Backup) into a database other than the database in which you restored Oracle_BI_DW_Base.rep.
4. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\ParameterFiles and copy the file 753_TENERIFE_UPG_PARAMS.txt into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
5. Rename 753_TENERIFE_UPG_PARAMS.txt to TENERIFE_UPG_PARAMS.txt.

6. In the Informatica Workflow Manager, open the Relational Connection Browser (in the menu bar, select Connections, and then select Relational), and edit the connect string, user and password for the relational connections as follows:
 - a. Edit the PARAM_OLTP_SIEBEL connection to point to the Siebel transactional database.
 - b. Edit the DataWarehouse connection to point to the newly upgraded data warehouse database.
 - c. Create or edit the PARAM_DAC_OLD connection to point to the previous DAC Repository database (the version from which you are upgrading).
 - d. Create or edit the PARAM_DAC_NEW connection to point to the DAC Repository database.

Note: If you are connected to an Oracle database, use the Oracle Native driver instead of ODBC.

If you are connected to a SQL Server database, use the ODBC driver rather than the native SQL Server driver.

7. For Siebel Applications (Horizontal), in Informatica Workflow Manager, navigate to the folder UPGRADE_753_to_770_HOR, and execute the following workflows in the order indicated:
 - a. Upgrade_R_Image
 - b. Upgrade_Unspecifieds
 - c. Upgrade_Agree
 - d. Upgrade_Asset
 - e. Upgrade_Opty
 - f. Upgrade_Order
 - g. Upgrade_Quote
 - h. Upgrade_Response
 - i. Upgrade_ServiceRequest
 - j. Upgrade_Others
 - k. Upgrade_Visibility
 - l. UpgradeSlowlyChangingDimensionStartDates
 - m. DAC_Metadata_Upgrade_Workflow
8. For Siebel Industry Applications (Vertical), in Informatica Workflow Manager, navigate to the folder UPGRADE_753_to_770_SIA, and execute the following workflows in the order indicated:
 - a. Upgrade_R_Image
 - b. Upgrade_Unspecifieds
 - c. Upgrade_Agree
 - d. Upgrade_Asset
 - e. Upgrade_Opty

- f. Upgrade_Order
- g. Upgrade_Quote
- h. Upgrade_Response
- i. Upgrade_ServiceRequest
- j. Upgrade_Others
- k. Upgrade_Visibility
- l. UpgradeSlowlyChangingDimensionStartDates
- m. Upgrade_Industry_R_Image
- n. Upgrade_Industry_Unspecified
- o. Upgrade_LS_ActivityProduct
- p. Upgrade_LS_Others
- q. Upgrade_Industry_Household
- r. Upgrade_FINS_Visibility
- s. UpgradeSlowlyChangingDimensionStartDates_Industry
- t. DAC_Metadata_Upgrade_Workflow

3.9 Upgrading the Data Warehouse Schema to Version 7.9.0

This procedure adds new tables, columns, and indexes to the existing data warehouse schema. It also modifies the existing data warehouse schema objects.

To upgrade the data warehouse schema

1. Run the UPGRADE.ctl script.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\CTLFiles, and locate the UPGRADE.ctl file.
 - b. If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)
 - c. Use the DDLimp utility to run the UPGRADE.ctl script. Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect  
string>  
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\UPGRADE.CTL>  
/L <..\OracleBI\dwrep\UPGRADE.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C SIEBEL_OLTP /G SSE_ROLE  
/I N /R Y /F C:\OracleBI\dwrep\UPGRADE.CTL /L C:\OracleBI\dwrep\UPGRADE.log
```

Notes:

- * /P <PASSWORD> - The password for the data warehouse.
- * /C <ODBC connect string> - The name of the ODBC connect string.
- * /I N - Tells DDLimp to ignore the indexes DDL defined in the CTL file if any exist. It does not change existing indexes. (The default is /I Y, which tells DDLimp to create and merge indexes from the CTL file with the indexes in the database.
- * For Oracle databases, use the Data Direct drivers.
In addition, you can use the following commands:
- * /W Y - If the OLAP database is Oracle and Unicode.
- * /Z Y - If the OLAP database is DB2 or SQL Server and Unicode.
- * /B <TABLE_SPACE_NAME> - If you want to create these tables in a separate table space. For DB2, This must be specified as 32K tablespace.
- * /X <INDEX_TABLE_SPACE_NAME> - If you want to create the indexes in a separate tablespace. For DB2, This must be specified as 32K tablespace.
- * /Y - Storage File for DB2/390.
- * /R - Regrant tables.

2. Run the 790_UPGRADE_PRE_CTL_SCRIPT.sql script.
 - a. Open the SQL client for your database type, for example, SQLPLUS for Oracle, Query Analyzer for SQL Server, or a command window for DB2.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 790_UPGRADE_PRE_CTL_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
3. Run the DW.ctl script.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\CTLFiles, and locate the DW.ctl file.
 - b. If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)
 - c. Use the DDLimp utility to run the DW.ctl script.
Use the command provided in Step 1, but substitute the correct script name.
4. Run the 790_UPGRADE_PRE_DIMENSION_SCRIPT.sql script.
 - a. Open the SQL client for your database type.

- b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
- c. Open the 790_UPGRADE_PRE_DIMENSION_SCRIPT.sql file, and copy the contents into the SQL client.
- d. Execute the script.

3.10 Importing New Schema Definitions into the Siebel Transactional Database

This procedure upgrades transactional database objects that relate to Oracle Business Intelligence Applications. It does not upgrade transactional database objects for Siebel CRM applications.

To import new schema definitions into the Siebel transactional database

1. Import schema definitions for non-image tables:
 - a. Access the DDLimp utility.
 - b. Run the DDL_OLTP.ctf script that is located in the OracleBI\dwrep\Upgrade\CTLFiles folder.
2. Import schema definitions for image tables:
 - a. In DAC, go to the Design view, and select your custom container from the drop-down list to the right of the Execute button.
 - b. Click the Tables tab.
 - c. Query for all tables for which the image suffix is not null.
 - d. Right-click in the list of tables returned by the query, and select Change Capture Scripts, and then Generate Image and Trigger Scripts.
 - e. In the Triggers and Image Tables dialog box, do the following:
 - Select the option All Tables in the List.
 - Select the option Generate Image Table Scripts.
 - Select the appropriate database type for the source system.
 - Click OK.
 - f. Open the SQL client for the source system database type, for example, SQL Plus for Oracle, Query Analyzer for SQL Server, or a command window for DB2.

The script may contain many lines; therefore, you can save the script file as a SQL file and execute it in a SQL client.
 - g. Copy the scripts generated by DAC into the SQL client and execute them.

3.10.1 Verifying the Siebel Transactional Database Upgrade

Follow this procedure to verify the following tables were created in the Siebel transactional database.

To verify the transactional database upgrade

- For all upgrade paths, verify the following tables were created in the Siebel transactional database:

- S_ETL_R_IMG_xxx
- S_ETL_I_IMG_xxx
- S_ETL_D_IMG_xxx
- S_ETL_PARAM
- S_ETL_PRD_ATTR
- S_ETL_PRD_REL

3.11 Migrating Data into the Data Warehouse Upgraded to Version 7.9.0

Follow this procedure to migrate data into the data warehouse upgraded to version 7.9.0.

To migrate data into the upgraded data warehouse

1. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\SrcFiles and copy the *.csv files into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
2. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\ParameterFiles and copy the file 770_TENERIFE_UPG_PARAMS.txt into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
3. Rename 770_TENERIFE_UPG_PARAMS.txt to TENERIFE_UPG_PARAMS.txt.
4. In the file TENERIFE_UPG_PARAMS.txt:
 - a. Search for the parameter \$\$SourceConnection.
 - b. Set the value to one of the following options, based on your Siebel applications (OLTP) version:
 - * SEBL_63
 - * SEBL_753
 - * SEBL_771
 - * SEBL_78
 - * SEBL_80 (use this value for OLTP versions 8.0 and 8.1.1)
 - * SEBL_VERT_753
 - * SEBL_VERT_771
 - * SEBL_VERT_78
 - * SEBL_VERT_80 (use this value for OLTP versions 8.0 and 8.1.1)

For example, if your Siebel applications (OLTP) version is Siebel Industry Applications 7.5.x, the parameter should appear as:

```
$$SourceConnection=SEBL_VERT_753
```

5. In the file TENERIFE_UPG_PARAMS.txt:
 - a. Search for the parameter \$\$Source_Container.
 - b. Set the value to one of the following options, based on your Siebel applications (OLTP) version:

- * Siebel 6.3
- * Siebel 7.5.3
- * Siebel 7.5.3 Vertical
- * Siebel 7.7.1
- * Siebel 7.7.1 Vertical
- * Siebel 7.8
- * Siebel 7.8 Vertical
- * Siebel 8.0 (use this value for OLTP versions 8.0 and 8.1.1)
- * Siebel 8.0 Vertical (use this value for OLTP versions 8.0 and 8.1.1)

For example, if your Siebel applications (OLTP) version is Siebel Industry Applications 7.5.x, the parameter should appear as:

```
$$Source_Container=Siebel 7.5.3 Vertical
```

6. In the file TENERIFE_UPG_PARAMS.txt, edit the ETL_PROC_WID parameter as follows:

```
MPLT_GET_ETL_PROC_WID.$$ETL_PROC_WID=<latest ETL_PROC_WID value from your database>
```

You can get this value from W_PARAM_G.ETL_PROC_WID.

7. If you are running Siebel Industry Applications (Vertical), in the file TENERIFE_UPG_PARAMS.txt, set the VERTICAL_UPGRADE parameter to 1. For example:

```
$$VERTICAL_UPGRADE=1
```

8. For Siebel Industry Applications (Vertical), define the alignment rule to be used for ETL loads.

- a. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\SrcFiles and copy the file AlignmentType_LS_782.csv into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
- b. Rename the file AlignmentType_LS_782.csv to AlignmentType.csv.
- c. Open the file AlignmentType.csv, and enter one of the following alignment item types under the ALIGN_TYPE field:
 - * Zipcode
 - * Account/Contact
 - * Brick
- d. Save the file.

The Oracle Business Analytics Warehouse allows only one alignment type to be used for ETL loads during upgrade.

9. In Informatica Workflow Manager, navigate to the folder UPGRADE_770_to_79, and execute the following workflows in the order indicated:

- a. MARKETING_LOAD
- b. Upgrade_Dimensions_Industry
- c. Update_Dimensions

- d. Update_Dimension_Unspecified
- e. Update_Facts
- f. (For Siebel Industry Applications only) Upgrade_LS_Dimensions
- g. (For Siebel Industry Applications only) Upgrade_LS_Facts
- h. SIL_PositionDimensionHierarchy_Full
- i. Load_INT_ORG_DH
- j. DIMENSION_LOAD

Note: If you are using the SCD version of the dimension, replace the corresponding TENN_UPG_W_XXX_D_784_To_W_XXX_D session with TENN_UPG_W_XXX_D_784_SCD_To_W_XXX_D. This will upgrade the data from the W_XXX_SCD version of the dimension to the new SCD-enabled W_XXX_D dimension.

- k. DIMENSION_UNSPECIFIED_UPDATE
- l. FACT_UPDATE
- m. DAC_Metadata_Upgrade_Workflow

Note: You need to run this workflow for upgrading to the data warehouse schema version 7.9.0 even though you ran a workflow with the same name during the upgrade to version 7.7.

10. If you upgraded your transactional database to Siebel Applications 8.0 or 8.1.1, navigate to the folder UPGRADE_790_to_791_SBL80UPG and run the following workflows in the order indicated:
 - a. UPGRADE_DIMENSIONS
 - b. UPGRADE_FACTS
11. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the main Informatica Repository (Oracle_BI_DW_Base.rep).
 - a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.
The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)
12. To verify the data migrated successfully:
 - a. Check whether any of the Informatica mapping log files stored in the directory \Informatica PowerCenter 8.6\Server\SessLogs indicates errors or failures.
 - b. Check whether the SQL scripts (790_UPGRADE_PRE_CTL_SCRIPT.sql and 790_UPGRADE_PRE_DIMENSION_SCRIPT.sql) that you ran in the SQL client of the database failed or errored out while executing.

- c. Check the log files for the CTL files (Upgrade.ctl and DW.ctl) that you ran using the DDLimp command to determine whether any of the runs failed.

If you did not detect any errors or failures in the steps above, then the data migration was successful.

Note: Ignore any error messages that indicate tables or indexes could not be dropped from the database. These error messages are displayed when the specified objects do not exist in the database. This can occur because each source system container holds a subset of all data warehouse tables.

13. If the data migration was successful, drop the tables that were created during the upgrade process, such as, W_xxxx_x_784 tables, LKP_xxxx_x, and W_ASSET_D_TMP.

This step frees the space occupied by these backup tables.

3.12 Upgrading the Data Warehouse Schema to Version 7.9.4 and Migrating Data

Follow this procedure to upgrade the data warehouse schema to version 7.9.4 and migrate data.

This procedure adds new tables, columns, and indexes to the existing data warehouse schema. It also modifies the existing data warehouse schema objects.

To upgrade the data warehouse schema to version 7.9.4 and migrate data

1. Run the 792_UPGRADE_PRE_CTL_SCRIPT.sql script.
 - a. Open the SQL client for your database type, for example, SQLPLUS for Oracle, Query Analyzer for SQL Server, or a command window for DB2.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 792_UPGRADE_PRE_CTL_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
2. Run the ddl_794.ctl script. Use the DDLimp utility to run the ddl_794.ctl script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder. Use the following command:
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\CTLFiles, and locate the ddl_794.ctl script.
 - b. If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

- c. Use the DDLimp utility to run ddl_794.ctf.

Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect
string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddl_794.ctf>
/L <..\OracleBI\dwrep\ddl_794.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\ddl_794.ctf /L C:\OracleBI\dwrep\ddl_794.log
```

3. Run the 792_UPGRADE_PRE_DIMENSION_SCRIPT.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 792_UPGRADE_PRE_DIMENSION_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
4. If you upgraded your transactional database to Siebel Applications 8.0 or 8.1.1, run the workflows to migrate your data into the upgraded data warehouse.
 - a. In Informatica Workflow Manager, navigate to the folder UPGRADE_790_TO_791_SBL80UPG.
 - b. Run the following workflows in the order indicated:

```
UPGRADE_DIMENSIONS
UPGRADE_FACTS
```
5. Verify the data migrated successfully by checking whether any of the Informatica mapping log files stored in the \Informatica PowerCenter 8.6\server\infa_shared\SessLogs directory indicate errors or failures.
6. If the data migration was successful, drop the tables that were created during the upgrade process, such as, W_xxxx_x_79x tables, LKP_xxxx_x and 79x_XXXX_TMP. This step frees the space occupied by these backup tables.
7. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the main Informatica Repository (Oracle_BI_DW_Base.rep).
 - a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.

The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)

3.13 Upgrading the Data Warehouse Schema to Version 7.9.5 and Migrating Data

Follow this procedure to upgrade the data warehouse schema to version 7.9.5 and migrate data.

This procedure adds new tables, columns, and indexes to the existing data warehouse schema. It also modifies the existing data warehouse schema objects.

To upgrade the data warehouse schema to version 7.9.5 and migrate data

1. Use the `reset_infa_seq_gen.bat` script to initialize the Informatica sequence generator for incremental runs on the Upgrade repository.
 - a. Navigate to `OracleBI\dwrep\Upgrade\DbScripts\<database type>`.
 - b. Open the `reset_infa_seq_gen.bat` file.

The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)
2. Run the `795_UPGRADE_PRE_CTL_SCRIPT.sql` script.
 - a. Open the SQL client for your database type, for example, SQLPLUS for Oracle, Query Analyzer for SQL Server, or a command window for DB2.
 - b. Navigate to the folder `OracleBI\dwrep\Upgrade\DbScripts\<database type>`.
 - c. Open the `795_UPGRADE_PRE_CTL_SCRIPT.sql` file, and copy the contents into the SQL client.
 - d. Execute the script.
3. Run the `UPGRADE_795.ctl` script.
 - a. Navigate to the folder `OracleBI\dwrep\Upgrade\CTLFiles`, and locate the `UPGRADE_795.ctl` file.
 - b. If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in `W_DAY_D`, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from `VARCHAR(50)` to `VARCHAR(100)`, and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of `VARCHAR(50)`, which could cause data to be truncated in some databases.)
 - c. Use the DDLimp utility to run the `UPGRADE_795.ctl` script. Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\UPGRADE_795.CTL>
/L <..\oracleBI\dwrep\UPGRADE.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\UPGRADE_795.CTL /L
C:\OracleBI\dwrep\UPGRADE.log
```

Notes:

- * /P <PASSWORD> - The password for the data warehouse.
- * /C <ODBC connect string> - The name of the ODBC connect string.
- * /I N - Tells DDLimp to ignore the indexes DDL defined in the CTL file if any exist. It does not change existing indexes. (The default is /I Y, which tells DDLimp to create and merge indexes from the CTL file with the indexes in the database.
- * For Oracle databases, use the Data Direct drivers.

In addition, you can use the following commands:

- * /W Y - If the OLAP database is Oracle and Unicode.
- * /Z Y - If the OLAP database is DB2 or SQL Server and Unicode.
- * /B <TABLE_SPACE_NAME> - If you want to create these tables in a separate table space. For DB2, This must be specified as 32K tablespace.
- * /X <INDEX_TABLE_SPACE_NAME> - If you want to create the indexes in a separate tablespace. For DB2, This must be specified as 32K tablespace.
- * /Y - Storage File for DB2/390.
- * /R - Regrant tables.

4. Use the DDLimp utility to run the ddl_795.ctl script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder.

If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddl_795.ctl>
/L <..\OracleBI\dwrep\ddl_795.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\ddl_795.ctl /L C:\OracleBI\dwrep\ddl_795.log
```

5. Run the 795_UPGRADE_PRE_DIMENSION_SCRIPT.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.

- c. Open the 795_UPGRADE_PRE_DIMENSION_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
6. Copy all of the domain value files in the folder \OracleBI\dwrep\Informatica\LkpFiles into the folder \Informatica PowerCenter 8.6\server\infa_shared\LkpFiles.
7. Migrate data into the upgraded data warehouse.
 - a. Copy the file Upgrade.rep file from the folder OracleBI\dwrep\Upgrade\Informatica\Repository into the folder Informatica PowerCenter 8.6\server\infa_shared\Backup.
 - b. Launch the Informatica PowerCenter Administration Console and restore Upgrade.rep (located in Informatica PowerCenter 8.6\server\infa_shared\Backup).
 - c. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\ParameterFiles and copy the file 795_UPG_PARAMS.txt into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
 - d. Set the parameter \$\$ETL_PROC_WID to the latest ETL_PROC_WID value from the database. You can get this value from W_PARAM_G.ETL_PROC_WID.
 - e. Set the parameter \$\$DATASOURCE_NUM_ID to the relevant value from the source system setup.
 - f. In Informatica Workflow Manager, open the Relational Connection Browser (in the menu bar, select Connections, and then select Relational), and edit the connect string, user and password for the relational connections as follows: create one relational connection based on the appropriate database platform for your OLTP database. Create the connection with the name PARAM_OLTP. Edit the PARAM_OLTP connection to match your OLTP environment. Edit the PARAM_OLAP connection to match your OLAP environment.

Note: If you are connected to an Oracle database, use the Oracle Native driver instead of ODBC.

If you are connected to a SQL Server database, use the ODBC driver rather than the native SQL Server driver.

- g. In Informatica Workflow Manager, navigate to the folder UPGRADE_794_to_795_SBL and execute the UPGRADE_DIMENSIONS workflow.
8. Verify the data migrated successfully.
 - a. Check whether any of the Informatica mapping log files stored in the \Informatica PowerCenter 8.6\serva\infa_shared\Sesslogs directory indicates errors or failures.
 - b. Check whether the script 795_UPGRADE_PRE_DIMENSION_SCRIPT.sql that you ran in the SQL client of the database failed or errored out while executing.
 - c. Check the log file for the script Upgrade_795.ctl that you ran using the DDLimp command to determine whether any of the runs failed.

If you did not detect any errors or failures in the steps above, then the data migration was successful.

Note: Ignore any error messages that indicate tables or indexes could not be dropped from the database. These error messages are displayed when the specified objects do not exist in the database. This can occur because each source system container holds a subset of all data warehouse tables

9. If the data migration was successful, drop the tables that were created during the upgrade process, such as W_XXXX_X_79X, LKP_XXXX_X and 79X_XXXX_TMP.

This step frees the space occupied by these backup tables.

10. Once the data migration steps above are complete, you can delete the Upgrade repository to avoid any accidental use or confusion.
 - a. In the Informatica PowerCenter Administration Console, select the Upgrade repository service.
 - b. In the General Properties area of the Properties tab, click Edit .
 - c. Make sure the operating mode of the repository service is set to Exclusive.
 - d. Click OK.
 - e. Choose Actions, and then click Delete Contents.
 - f. In the Delete contents for <repository name> dialog, enter the repository username and password (for example, Administrator/Administrator), then click OK.
11. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the main Informatica Repository (Oracle_BI_DW_Base.rep).
 - a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.
The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)

3.14 Upgrading the Data Warehouse Schema to Version 7.9.5.1

Follow this procedure to upgrade the data warehouse schema to version 7.9.5.1.

To upgrade the data warehouse schema

- Use the DDLimp utility to run the ddl_7951.ctl script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder.

If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the

preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>  
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddl_7951.ctl>  
/L <..\OracleBI\dwrep\ddl_7951.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE  
/I N /R Y /F C:\OracleBI\dwrep\ddl_7951.ctl /L C:\OracleBI\dwrep\ddl_7951.log
```

3.15 Upgrading the Data Warehouse Schema to Version 7.9.6 and Migrating Data

Follow the procedures in this section to upgrade the data warehouse schema to version 7.9.6 and migrate data.

This section contains the following topics:

- [Section 3.15.1, "Upgrading the Data Warehouse Schema to Version 7.9.6"](#)
- [Section 3.15.2, "Migrating Data into the Upgraded Data Warehouse"](#)
- [Section 3.15.3, "Verifying the Data Migrated Successfully"](#)
- [Section 3.15.4, "Resetting Refresh Dates"](#)

3.15.1 Upgrading the Data Warehouse Schema to Version 7.9.6

Follow this procedure to upgrade the data warehouse schema to version 7.9.6.

To upgrade the data warehouse schema to version 7.9.6

1. Run the 796_UPGRADE_DROP_INDEXES.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 796_UPGRADE_DROP_INDEXES.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
2. Run the UPGRADE_796.ctl script.

This script adds temp tables for the upgrade process.

- a. Navigate to the folder OracleBI\dwrep\Upgrade\CTLFiles, and locate the UPGRADE_796.ctl file.
- b. If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was

changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

- c. Use the DDLimp utility to run the UPGRADE_796.ctl script. Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect
string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\UPGRADE_796.CTL>
/L <..\OracleBI\dwrep\UPGRADE.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\UPGRADE_796.CTL /L
C:\OracleBI\dwrep\UPGRADE.log
```

Notes:

- * /P <PASSWORD> - The password for the data warehouse.
- * /C <ODBC connect string> - The name of the ODBC connect string.
- * /I N - Tells DDLimp to ignore the indexes DDL defined in the CTL file if any exist. It does not change existing indexes. (The default is /I Y, which tells DDLimp to create and merge indexes from the CTL file with the indexes in the database.
- * For Oracle databases, use the Data Direct drivers.
In addition, you can use the following commands:
 - * /W Y - If the OLAP database is Oracle and Unicode.
 - * /Z Y - If the OLAP database is DB2 or SQL Server and Unicode.
 - * /B <TABLE_SPACE_NAME> - If you want to create these tables in a separate table space. For DB2, This must be specified as 32K tablespace.
 - * /X <INDEX_TABLE_SPACE_NAME> - If you want to create the indexes in a separate tablespace. For DB2, This must be specified as 32K tablespace.
 - * /Y - Storage File for DB2/390.
 - * /R - Regrant tables.

3. Run the 796_UPGRADE_PRE_CTL_SCRIPT.sql script.
 - a. Open the SQL client for your database type, for example, SQLPLUS for Oracle, Query Analyzer for SQL Server, or a command window for DB2.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 796_UPGRADE_PRE_CTL_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Find and replace the Datasource_Num_id = -1 with the correct Datasource_Num_id from your existing implementation. (The value -1 is a dummy value.)
 - e. Execute the script.

4. Use the DDLimp utility to run the ddl_796.ctf script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder.

If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddl_796.ctf>
/L <..\OracleBI\dwrep\ddl_796.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\ddl_796.ctf /L C:\OracleBI\dwrep\ddl_796.log
```

5. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the Upgrade repository.
 - a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.

The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)
6. Run the 796_UPGRADE_PRE_DIMENSION_SCRIPT.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 796_UPGRADE_PRE_DIMENSION_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
7. Copy all of the domain value files in the folder \OracleBI\dwrep\Informatica\LkpFiles into the folder \Informatica PowerCenter 8.6\server\infa_shared\LkpFiles.

3.15.2 Migrating Data into the Upgraded Data Warehouse

Follow this procedure to migrate data into the upgraded data warehouse.

To migrate data into the upgraded data warehouse

1. Copy the file Upgrade.rep file from the folder OracleBI\dwrep\Upgrade\Informatica\Repository into the folder Informatica PowerCenter 8.6\server\infa_shared\Backup.

2. Launch the Informatica PowerCenter Administration Console and restore Upgrade.rep (located in Informatica PowerCenter 8.6\server\infa_shared\Backup).
3. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\ParameterFiles and copy the file 796_UPG_PARAMS.txt into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
4. In the 796_UPG_PARAMS.txt file, set the following parameters:
 - a. \$\$ETL_PROC_WID. Set this parameter to the relevant value from the source system setup. You can get this value from W_PARAM_G.ETL_PROC_WID
 - b. \$\$DATASOURCE_NUM_ID. Set this parameter to the relevant value from the source system setup.
 - c. \$\$INITIAL_EXTRACT_DATE. Set this parameter to the initial extraction data of the data warehouse.
 - d. \$\$WH_DATASOURCE_NUM_ID. Set this parameter to the data source number ID you have set up for the data warehouse.
 - e. \$\$MASTER_ORG. Get this value from the Source System Parameters tab in DAC.
 - f. \$\$INV_PROD_CAT_SET_ID1. Get this value from the Source System Parameters tab in DAC.
 - g. \$\$PROD_CAT_SET_ID1. Get this value from the Source System Parameters tab in DAC.
 - h. Set the parameter \$\$IS_SOURCE_PRE_80 to Y if your source OLTP application was on a version prior to Siebel 8.0 before you began the upgrade process. Otherwise, set this parameter to N.
5. Configure common parameters specific to Siebel source systems. For instructions, see [Section B.2, "Configuring Common Parameters for Siebel Source Systems."](#)
6. In Informatica Workflow Manager, open the Relational Connection Browser (in the menu bar, select Connections, and then select Relational), and edit the connect string, user and password for the relational connections as follows:
 - a. Edit the connection PARAM_OLTP_SIEBEL to match your OLTP environment.
 - b. Edit the connection PARAM_OLAP to match your OLAP environment.
 - c. Edit the connection PARAM_DAC to match your DAC database.

Note: If you are connected to an Oracle database, use the Oracle Native driver instead of ODBC.

If you are connected to a SQL Server database, use the ODBC driver rather than the native SQL Server driver.

7. In Informatica Workflow Manager, navigate to the folder UPGRADE_7951_to_796_SBL and execute the UPGRADE_DIMENSIONS and UPGRADE_FACTS workflows.
8. Run the 796_UPGRADE_POST_SCRIPT.sql.
 - a. Open the SQL client for your database type.

- b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
- c. Open the 796_UPGRADE_POST_SCRIPT.sql file, and copy the contents into the SQL client.
- d. Execute the script.

3.15.3 Verifying the Data Migrated Successfully

Follow this procedure to verify that the data was migrated successfully into the upgraded data warehouse.

To verify the data migrated successfully

1. Check whether any of the Informatica mapping log files stored in the \Informatica PowerCenter 8.6\serva\infa_shared\Sesslogs directory indicates errors or failures.
2. Check whether the following scripts that you ran in the SQL client of the database failed or errored out while executing:

796_UPGRADE_DROP_INDEXES.sql

796_UPGRADE_PRE_CTL_SCRIPT.sql

796_UPGRADE_PRE_DIMENSION.sql

796_UPGRADE_POST_SCRIPT.sql

3. Check the log files for the ddl_796.ctl and Upgrade_796.ctl scripts that you ran using the DDLimp command to determine whether any of the runs failed.

If you did not detect any errors or failures in the steps above, then the data migration was successful.

Note: Ignore any error messages that indicate tables or indexes could not be dropped from the database. These error messages are displayed when the specified objects do not exist in the database. This can occur because each source system container holds a subset of all data warehouse tables

4. If the data migration was successful, drop the tables that were created during the upgrade process, such as W_xxxx_x_79x, LKP_xxxx_x, 79x_xxxx_TMP, and TMP_xxxx_79x.

This step frees the space occupied by these backup tables.

3.15.4 Resetting Refresh Dates

After verifying the data was migrated successfully into the upgraded data warehouse, follow this procedure to reset refresh dates.

1. In Informatica Workflow Manager, navigate to the folder UPGRADE_7951_TO_796_SBL and execute the RESET_DAC_REFRESH_DATES workflow.
2. Once the data migration steps above are complete, you can delete the Upgrade repository to avoid any accidental use or confusion.
 - a. In the Informatica PowerCenter Administration Console, select the Upgrade repository service.

- b. In the General Properties area of the Properties tab, click Edit .
- c. Make sure the operating mode of the repository service is set to Exclusive.
- d. Click OK.
- e. Choose Actions, and then click Delete Contents.
- f. In the Delete contents for <repository name> dialog, enter the repository username and password (for example, Administrator/Administrator), then click OK.

3.16 Upgrading the Siebel Analytics Repository

This process merges your customizations of a prior release of the Siebel Analytics repository with the new version of the Oracle BI repository. Before you begin this process, make sure you have backed up and renamed your existing repository.

To upgrade the repository, perform the following tasks:

- [Section 3.16.1, "Preparing for the Oracle BI Repository Upgrade"](#)
- [Section 3.16.2, "Equalizing the Oracle BI Repositories"](#)
- [Section 3.16.3, "Comparing the Siebel Analytics and Oracle BI Repositories"](#)
- [Section 3.16.4, "Merging the Siebel Analytics and Oracle BI Repositories"](#)
- [Section 3.16.5, "Regression Testing the Oracle BI Repository Merge"](#)

The tasks in this section refer to multiple versions of the Siebel Analytics and Oracle BI repositories. [Table 3–1](#) provides the names and descriptions of the repositories used in the examples in this section.

Table 3–1 Names of Analytics Repositories used in Examples

Name of Repository	Description
SiebelAnalytics_7x.rpd	The standard Siebel Analytics repository for the version you are upgrading from. This repository is referred to as the "original" repository in the examples in this section. Note: Standard repositories from previous releases are available in the folder \OracleBI\Upgrade.
OracleBIAnalyticsApps.rpd	The standard Oracle BI repository for the version you are upgrading to.
Customer_SiebelAnalytics.rpd	The Siebel Analytics repository that contains your customizations for the version you are upgrading from.
Merged_Repository_OracleBI.rpd	The Oracle BI repository that contains your customizations for the version you are upgrading to.

3.16.1 Preparing for the Oracle BI Repository Upgrade

Follow this procedure to prepare for the repository upgrade.

To prepare for the Analytics repository upgrade

1. Set up a directory for the merge process, such as \OracleBIUpgrade, and create the following subfolders:
 - Original
 - AfterEqualize

- AfterMerge
 - AfterManualWork
 - AfterRegressions
2. Copy the original repository (for example, SiebelAnalytics_7x.rpd), the production repository (for example, Customer_SiebelAnalytics.rpd, and the repository from the latest installation (for example, OracleBIAnalyticsApps.rpd) into the folder \OracleBIUpgrade\Original.

If, in your current environment, you are running Siebel Analytics for one or more modules using a Siebel Analytics repository in which you extracted the corresponding projects for the modules from the standard Siebel Analytics repository file you received from the previous release, you need to extract the same projects from the SiebelAnalytics_7x.rpd file and use this as your original repository. (If you have the original repository that you extracted during the last upgrade, you can use it as the original repository file.) This will prevent you from losing any new metadata you would like to add in this upgrade.

Also, if you customized the Siebel Analytics repository by trimming a large number of objects and you would like to get those objects back during the current upgrade, you need to trim the SiebelAnalytics_7x.rpd file in the same way and use the modified version as the original repository file. This will prevent you from losing any new metadata you would like to add in this upgrade.

3.16.2 Equalizing the Oracle BI Repositories

The Merge feature in the Administration Tool relies on a change detection algorithm to determine the changes that need to be made to upgrade repositories correctly. For the algorithm to work correctly, it has to determine which objects in the three repositories (for example, SiebelAnalytics_7x.rpd, OracleBIAnalyticsApps.rpd, and Customer_SiebelAnalytics.rpd) are equivalent.

The point of this step is to determine for every object in the OracleBIAnalyticsApps.rpd and the Customer_SiebelAnalytics.rpd whether it is coming from the SiebelAnalytics_7x.rpd.

Equivalence between objects is established using the Administration Tool's Equalize feature. The file that you specify in the Output option (-O) is the only file that is modified during the equalization process.

The Equalize feature has several mechanisms for determining whether an object in two different repositories is semantically the same:

- **Fully Qualified Name.** If an object in one repository has the same fully qualified name as another object of the same class in another repository, then the two objects are declared equal.
- **Simple String Substitution.** Equivalence can be declared between two objects of the same class in two repositories whose only difference is that some key characters in their names differ. The equalizerpds executable file ignores those characters while checking fully qualified names. For example, "Core"."W_DAY_D" might be considered equivalent to "Core"."W DAY D" if the characters "_" and " " have been declared as equivalent.
- **Rename File.** When none of the preceding rules are applicable, equivalence can be manually declared using a script as input to the equalizerpds executable file. Oracle ships the rename files (MAP) for the major releases. The files are located in the OracleBI\Upgrade folder. You can also create your own rename files for

customizations not covered in the files that Oracle ships. You can open and edit the rename files in Microsoft Excel.

The syntax of the equalizerpds command is as follows:

```
equalizerpds.exe -A userid1 [-B [password1]] -C base_repository_name -D userid2
[-E [password2]] -F repository2_name [-J udml_utf8_file_name_equalization] [-O
ouput_repository_name] [-X] [-Y equalStringSet]
-X          Treat 'Factxxxx' as 'Fact' in Business Model.
-Y          Treat the characters as equals.
/?          Display this usage information and exit.
```

To equalize a repository

1. Copy the appropriate MAP file from the OracleBI\Upgrade folder into the folder where you will execute equalizerpds.exe, for example, \OracleBIUpgrade\Original.
2. Run equalizerpds.exe to equalize the repository from the latest installation (for example, OracleBIAnalyticsApps.rpd) with the original repository (for example, SiebelAnalytics_7x.rpd). An example of the equalizerpds command is as follows:

```
equalizerpds -A Administrator -B SADMIN
-C \\OracleBIUpgrade\Original\SiebelAnalytics_7x.rpd
-D Administrator -E SADMIN
-F \\OracleBIUpgrade\Original\OracleBIAnalyticsApps.rpd
-O \\OracleBIAnalyticsUpgrade\AfterEqualize\OracleBIAnalyticsApps.rpd
-X -J rename7x-79.map
```

The MAP files are located in the \OracleBI\Upgrade folder.

If the equalizerpds.exe executable file runs correctly, no errors are returned.

3. Run equalizerpds.exe to equalize your customized repository (for example, Customer_SiebelAnalytics.rpd) with the original repository (for example, SiebelAnalytics_7x.rpd). An example of the equalizerpds command is as follows:

```
equalizerpds -A Administrator -B SADMIN
-C \\OracleBIUpgrade\Original\SiebelAnalytics_7x.rpd
-D Administrator -E SADMIN
-F \\OracleBIUpgrade\Original\Customer_SiebelAnalytics.rpd
-O \\OracleBIUpgrade\AfterEqualize\Customer_OracleBIAnalyticsApps.rpd
```

The execution of equalizerpds that equalizes the customer repository with the original repository does not use the rename file.

Make sure that the original repository is copied unchanged into its new location so that after running the script, all three repositories are contained within the \OracleBIUpgrade\AfterEqualize directory.

4. To verify the process completed successfully, compare the size of the repositories. The output repository (-O) should be close to the same size as the repository you equalized (-F).

3.16.3 Comparing the Siebel Analytics and Oracle BI Repositories

Follow this procedure to compare your existing repository with the new version to which you are upgrading.

To compare the repositories

- Use the Administration Tool's Compare Repositories feature to analyze the differences between your existing repository and the new version of the repository

to which you are upgrading. Note where elements have been created, removed, or changed in the new version. Consider whether you can use the new metadata and retire customizations you made in the existing repository.

For instructions on how to use the Administration Tool's Compare Repositories feature, see *Oracle Business Intelligence Server Administration Guide*.

3.16.4 Merging the Siebel Analytics and Oracle BI Repositories

In this procedure, you execute the main algorithm to upgrade the repository. For more information on merging the repositories, see *Oracle Business Intelligence Server Administration Guide*.

To merge versions of the repositories

1. Copy the three repositories (for example, SiebelAnalytics_7x.rpd, OracleBIAnalyticsApps.rpd, and Customer_SiebelAnalytics.rpd) to the AfterMerge folder.
2. Open the repository from the latest installation (for example, OracleBIAnalyticsApps.rpd) in the \OracleBIUpgrade\AfterMerge folder.
3. Save the repository with a new name, for example, Merged_Repository_OracleBIAnalyticsApps.rpd.

This new repository will contain the final results of the upgrade.

4. From the Administration Tool menu bar, select File, then select Merge.
5. In the Select Original Repository dialog box, select the original repository (for example, SiebelAnalytics_7x.rpd).
6. Enter the password, and click OK.
7. Click Select for the Modified Repository field.
8. In the Select Modified Repository dialog box, select the repository that contains the customizations you made to the previous version of the Analytics repository.
9. Click Open, type the password, and then click OK.
10. In the Decision drop-down list, select the action you want to take regarding the repository change, or accept the default action.
11. To locate subsequent rows with empty Decision fields, click the Decision header cell.

When all rows have a value in the Decision field, the Merge button is enabled.

12. Click Merge.

This process can take up to 40 minutes, depending on the size of the repositories you are working with. A message will alert you when the merge is complete.

13. Click Yes when asked if you want to run a consistency check.

The number of errors returned by the consistency check is an indication of how successful the merge process was. If you receive many errors, for example, over 300, you should analyze the reason for the errors. If the merge process failed to recognize that two objects are the same, you may need to edit the rename file if the object is in the Current repository, or add your own rename file if you have renamed many of the objects and the upgrade engine failed to relate them to the original objects.

You also may need to change the actions you selected in the Decision drop-down list before rerunning the merge. This could save you time by reducing the number of errors that you will need to fix manually.

Once you are satisfied with the results of the merge, you should fix the remaining errors manually. It is important that you fix all errors before moving on to the next step. This repository serves as the input for the next stage.

You should also check that all of your customized objects are present and that no duplicate physical tables were introduced. To check for duplicate tables, search for physical tables using a query such as:

```
where name like '*#1'
```

14. Copy the repository to the folder \OracleBIUpgrade\AfterManualWork.

3.16.5 Regression Testing the Oracle BI Repository Merge

In performing a regression test for the repository merge, the objective is to collect a set of logical SQL statements that are used for reports and to verify that they continue to work with the new metadata. For this purpose, it is recommended that you perform the following procedure.

To perform regression testing

1. Run the reports that are necessary to include in the regression suite. These reports might be a subset of the reports in the Presentation Catalog.
2. Collect the logical SQL generated in the previous step. You can do this using Usage Tracking or by parsing the query log file.

For information about Usage Tracking, see *Oracle Business Intelligence Server Administration Guide*.

3. Execute the logical SQL against the old repository using the command line utility nQCmd.exe located in \OracleBI\server\bin, and save the results to a file.

For information about the nQCmd.exe utility, see *Oracle Business Intelligence Server Administration Guide*.

4. Edit the logical SQL test scripts to account for the name changes or modifications resulting from the upgrade.
5. Execute the edited logical SQL against the merged repository, and save the results.
6. Compare the results from the steps above and try to explain the differences. If it is determined that these differences are due to the upgrade process, then you have to correct them manually.

This repository now contains the merged content from the new OracleBIAnalyticsApps.rpd and the production repository.

3.17 Upgrading the Oracle BI Presentation Catalog

You will need to upgrade your current Oracle BI Presentation Catalog if your organization:

- Has prebuilt applications already installed, and
- Has customized the current Oracle BI Presentation Catalog

If you made no changes to the previous Presentation Catalog distributed with previous versions of prebuilt applications, you do not need to upgrade the catalog. You can begin using the newer version of the catalog.

This process includes the following tasks:

- [Section 3.17.1, "Trimming the Input Presentation Catalog"](#)
- [Section 3.17.2, "Upgrading the Oracle BI Presentation Catalog to a Newer Version"](#)
- [Section 3.17.3, "Testing the Results of the Presentation Catalog Upgrade"](#)

Caution: In releases of Oracle BI Applications previous to 7.9, the Presentation Catalog (formerly known as the Siebel Analytics Web Catalog) was stored in a single file rather than in a directory structure of individual files. If you have a previous version of the Presentation Catalog, you will need to convert it to the new format. For more information about how to convert the Presentation Catalog to the new format, see the *Oracle Business Intelligence Infrastructure Upgrade Guide*.

3.17.1 Trimming the Input Presentation Catalog

Before you upgrade and merge your current Presentation Catalog with the new Presentation Catalog, determine which of the existing content you want to keep and which new content you want to incorporate. Review your existing Presentation Catalog and determine the usage patterns of reports and dashboards. Note that some of the preconfigured content in the existing catalog may appear in the new version in a redesigned format. In addition, the new version includes completely new content. After you have decided the content that is to make up your enterprise Presentation Catalog, trim the input catalogs using the Catalog Manager. For information on trimming catalogs, see *Oracle Business Intelligence Presentation Services Administration Guide*.

3.17.2 Upgrading the Oracle BI Presentation Catalog to a Newer Version

The Presentation Catalog upgrade process makes use of three catalogs:

- The *original* Presentation Catalog. This is the unmodified Presentation Catalog that you received with the Oracle BI Applications release that you are upgrading from.
- The *new* Presentation Catalog. This is the Presentation Catalog that is installed in the OracleBIData\web\catalog folder with the installation of Oracle BI Applications.
- The *current* Presentation Catalog. This is the Presentation Catalog currently in use at your organization.

You use Catalog Manager for this upgrade process. Catalog Manager compares the content in both the *Current* Presentation Catalog and the *Modified* Presentation Catalog with the content in the Original Presentation Catalog, merges any changes into the *Current* Presentation Catalog, and produces a list of upgrade differences, which you must resolve by indicating how you want the differences handled. If the catalogs have conflicting content, you can choose which catalog the content should be taken from. The end result is a merged Presentation Catalog that contains the site-specific changes, as well as new metadata.

To upgrade your Presentation Catalog to a newer version

1. Make a backup copy of the current Presentation Catalog, rename the folder <catalogname>_old, and move it to a temporary location.
2. Copy the original Presentation Catalog into the folder that holds your current Presentation Catalog and rename it <catalogname>_Original.
3. Start Catalog Manager and open the new Presentation Catalog in offline mode.
4. Select Tools, then select Upgrade Catalog.
5. In the original Presentation Catalog field, browse to locate the original Presentation Catalog.
6. In the Current Presentation Catalog field, browse to locate your current Web Catalog, <catalogname>_old.
7. Click OK.
8. Resolve any upgrade differences as follows:
 - a. Review each unresolved difference in the Unresolved differences list.
 - b. For each unresolved difference, select the version that you want to keep.
 - c. Click OK.

The log file SiebelAnalyticsMigrationLog.txt holds information about the merge process. This log file is written to \OracleBI\web\catalogmanager. If you get an error logged in the file, this means that the path in question had a problem that did not allow the merge mechanism to resolve the merge. No action was taken. To merge that particular item, go into your original Presentation Catalog and merge it manually.

9. Review the upgraded Presentation Catalog, and, if necessary, set permissions for objects.
10. Save the new Presentation Catalog.

3.17.3 Testing the Results of the Presentation Catalog Upgrade

Note: Before you perform this step, you must first migrate the data into the upgraded data warehouse.

The Presentation Catalog upgrade functionality does not automatically carry over object permissions; therefore, you should review the Presentation Catalog object permissions before you perform this step.

This step ensures that the upgraded reports and the new preconfigured reports are functional and render correct results within the new, merged Presentation Catalog. This step is typically performed by visually inspecting the final results of the complete end-to-end upgrade process.

For upgraded reports, the preferred approach for comparison purposes is to have side-by-side environments, and have users review specific dashboard content between the two environments. Examine not only the look and feel of the application but also the data contained in the reports to make sure the content remains the same. It is recommended that you request users to use various elements of the user interface to validate results, such as global prompts, column selectors, report filters, drills, and navigations, as they normally do on a day-to-day basis.

Also review the overall visibility and administrative settings in the new Presentation Catalog to ensure they are correct. Pay careful attention to the visibility rules that are established for any content that was migrated during the upgrade. You might have to manually adjust these settings.

Upgrading Siebel Business Analytics 7.7, 7.7.x and 7.8.x

This section contains instructions for upgrading Oracle's Siebel Business Analytics releases 7.7, 7.7.x, and 7.8.x.

This section includes the following topics:

- [Section 4.1, "Upgrading Oracle BI Infrastructure"](#)
- [Section 4.2, "Upgrading Oracle BI Applications"](#)
- [Section 4.3, "Upgrading to Informatica PowerCenter Version 8.6 and Hotfix 4"](#)
- [Section 4.4, "Upgrading the Informatica Repository"](#)
- [Section 4.5, "Configuring Informatica PowerCenter Version 8.6 to Work with DAC"](#)
- [Section 4.6, "Upgrading and Configuring DAC"](#)
- [Section 4.7, "Overview of Upgrading the Data Warehouse Schema and Migrating Data"](#)
- [Section 4.8, "Upgrading the Data Warehouse Schema to Version 7.9.0"](#)
- [Section 4.9, "Importing New Schema Definitions into the Siebel Transactional Database"](#)
- [Section 4.10, "Migrating Data into the Data Warehouse Upgraded to Version 7.9.0"](#)
- [Section 4.11, "Upgrading the Data Warehouse Schema to Versions 7.9.4 and Migrating Data"](#)
- [Section 4.12, "Upgrading the Data Warehouse Schema to Version 7.9.5 and Migrating Data"](#)
- [Section 4.13, "Upgrading the Data Warehouse Schema to Version 7.9.5.1"](#)
- [Section 4.14, "Upgrading the Data Warehouse Schema to Version 7.9.6 and Migrating Data"](#)
- [Section 4.15, "Upgrading the Siebel Analytics Repository"](#)
- [Section 4.16, "Upgrading the Oracle BI Presentation Catalog"](#)

4.1 Upgrading Oracle BI Infrastructure

Upgrade the Oracle BI Infrastructure to the version that is supported for this release of Oracle BI Applications. See the *System Requirements and Supported Platforms for Oracle Business Intelligence Applications* for the current version that is supported. For

information on installing the supported version of Oracle BI Infrastructure, see the *Oracle Business Intelligence Infrastructure Upgrade Guide*.

4.2 Upgrading Oracle BI Applications

Run the Oracle BI Applications installer to upgrade your Oracle BI Applications environment to the current version. For instructions on running the installer, see *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

Note: Perform only the step that addresses the running of the installer. Do not perform the configuration instructions that follow the running of the installer.

Note: If you have a previous release of Oracle BI Applications installed, you must uninstall it before you run the installer for the current release. If you do not uninstall the old release, some folders from the current release will not be correctly installed. (Make a back-up of your DAC folder before you uninstall the old release. This will be the backup of your DAC Client and Server and the DAC metadata files.)

4.3 Upgrading to Informatica PowerCenter Version 8.6 and Hotfix 4

You must install Informatica PowerCenter 8.6 and Hotfix 4 to run the current version of Oracle BI Applications.

The components and architecture for Informatica PowerCenter 8.6 differ significantly from PowerCenter 7.x versions. Oracle recommends that you carefully review the Informatica PowerCenter 8.6 documentation, which is included on the Informatica DVD provided with Oracle BI Applications.

For a summary of installation instructions for installing Informatica PowerCenter 8.6 on a single machine in an Oracle BI Applications deployment, see the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

For detailed information about deploying Informatica PowerCenter 8.6, refer to the *Informatica PowerCenter Installation Guide*, *Informatica PowerCenter Configuration Guide*, *Informatica PowerCenter Administrator Guide*, and related documentation.

For information about applying Hotfix 4, see *PowerCenter Version 8.6.0 Hotfix 4 Release Notes*. This PDF is included with the Informatica documentation on the Informatica PowerCenter DVD and is also included with the Hotfix 4 installation

To upgrade to Informatica PowerCenter 8.6

1. Perform the pre-upgrade steps documented in the *Informatica PowerCenter Installation Guide*.
2. Install Informatica PowerCenter version 8.6 and Hotfix 4, by following the instructions in the *Informatica PowerCenter Installation Guide* and *PowerCenter Version 8.6.0 HotFix 4 Release Notes*.
3. Perform the post-upgrade steps documented in the *Informatica PowerCenter Installation Guide*.

Note: The Informatica PowerCenter 8.6 installation process includes upgrading your current Informatica Repository to the version 8.6 format. This process is necessary so that you will be able to access your current repository using version 8.6 client tools so that you can perform the procedure in [Section 4.4, "Upgrading the Informatica Repository."](#)

In [Section 4.4, "Upgrading the Informatica Repository,"](#) you back up and rename your current repository and then restore the Informatica Repository (Oracle_BI_DW_Base.rep) that is installed during the Oracle BI Applications installation. You then copy your custom folder from the backed up repository into the newly restored Oracle_BI_DW_Base repository.

4.4 Upgrading the Informatica Repository

Follow this procedure to upgrade the Informatica Repository.

For detailed instructions on backing up and restoring the Informatica Repository, see the topic titled, "Backing Up and Restoring the Repository," in "Chapter 8: Managing the Repository," in the *Informatica PowerCenter Administrator Guide*, which is included on the Informatica DVD provided with Oracle BI Applications.

To upgrade the Informatica Repository

1. Make sure you have backed up and renamed your current Informatica Repository.

Note: This repository must be upgraded to the version 8.6 format during the procedure in [Section 4.3, "Upgrading to Informatica PowerCenter Version 8.6 and Hotfix 4."](#) You must upgrade this repository to the version 8.6 format in order to move your custom folder from this repository into the new Oracle_BI_DW_Base repository that you restore in the steps below.

2. Copy the Oracle_BI_DW_Base.rep file from the folder OracleBI\dwrep\Informatica\Repository into the folder \Informatica\PowerCenter 8.6\server\infa_shared\Backup.

Note: The Oracle_BI_DW_Base.rep file is installed in the OracleBI root directory when you run the Oracle BI Applications installer, as described in [Section 4.2, "Upgrading Oracle BI Applications."](#)

3. Restore the Oracle_BI_DW_Base.rep repository.
4. Copy the custom folder from your previous Informatica Repository to the newly created Informatica Repository:
 - a. Launch the Informatica PowerCenter Repository Manager, and connect to both your previous and newly created Informatica repositories.
 - b. Copy the Custom folder in your previous repository into the newly created Informatica Repository.
 - c. Make sure there is an individual workflow for each of the mappings in the Custom folder.

4.5 Configuring Informatica PowerCenter Version 8.6 to Work with DAC

Informatica PowerCenter version 8.6 requires additional configuration steps to work with Oracle BI Applications and DAC. For instructions on performing these steps, see the following sections in the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*:

- Copying Source Files and Lookup Files
- Setting PowerCenter Integration Services Relaxed Code Page Validation
- Setting PowerCenter Integration Services Custom Properties
- Creating the Repository Administrator User in the Native Security Domain

4.6 Upgrading and Configuring DAC

This section includes information you must follow to upgrade and configure DAC.

This section includes the following topics:

- [Section 4.6.1, "Installing the DAC Platform and Oracle BI Applications Metadata Repository Files"](#)
- [Section 4.6.2, "Configuring the DAC Client and Server to Work with Oracle BI Applications and Informatica"](#)
- [Section 4.6.3, "Upgrading the DAC Repository for Siebel Business Analytics"](#)

4.6.1 Installing the DAC Platform and Oracle BI Applications Metadata Repository Files

Note: You must retain your current DAC environment (Siebel Business Analytics 7.7, 7.7.x, or 7.8.x), including the DAC Client and Server and Repository. This is necessary because you cannot open the version 7.7, 7.7.x or 7.8.x DAC Repository with DAC version 10.1.3.4.1.

The current release of Oracle BI Applications requires DAC version 10.1.3.4.1. This version of DAC is installed by its own installer and not the Oracle BI Applications installer. After you install the current version of DAC, you then need to copy metadata files from the machine hosting Oracle BI Applications to the machines hosting the DAC Client and Server. You then need to import the new metadata into the DAC Repository. For instructions on performing these tasks, see the following sections in the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*:

- Installing the DAC Platform

Note: You must perform all of the procedures in this section, including installing JDBC drivers and creating ODBC database connections.

- Installing DAC Metadata Files
- Logging into DAC for the First Time and Importing Metadata into the DAC Repository

After you complete this procedure, you will have the default DAC Repository for Oracle BI Applications version 7.9.6.

4.6.2 Configuring the DAC Client and Server to Work with Oracle BI Applications and Informatica

You need to perform certain configuration tasks to enable the DAC Client and Server to work with Oracle BI Applications and Informatica PowerCenter. For instructions on performing these tasks, see the following sections in the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*:

- Configuring the DAC Server
- Configuring DAC Integration Settings
- Configuring the SiebelUnicodeDB Custom Property

Note: This procedure is required only if your source to target data movement configuration is Unicode to Unicode.

- Setting Up DAC to Receive Email Notification
- Additional Configuration Tasks

In this section, perform only the tasks that apply to your environment.

4.6.3 Upgrading the DAC Repository for Siebel Business Analytics

This section applies to upgrades of Siebel Business Analytics 7.7, 7.7.x, and 7.8.x.

Note: For Siebel Business Analytics 7.8.x, if your DAC Repository customizations are limited and you can identify them, use the procedure in this section, which is a manual procedure. If your customizations are extensive and you cannot identify them, you can use the procedure in [Section 5.6.3, "Upgrading the DAC Repository,"](#) which is an automated process that uses the DAC version 10.1.3.4.1 Upgrade/Merge Wizard.

Before you begin this procedure, do the following:

- Determine what customizations were made to your existing DAC Repository. Moving the customizations in your existing DAC Repository into the new DAC Repository is a manual process.
- Make sure you have renamed and backed up your existing DAC Repository into a different database. When you backup the DAC Repository, you export the DAC metadata, in XML format (using the DAC's Export tool), into a folder other than the standard DAC export folder where backups are stored (DAC\export). For instructions on exporting DAC metadata, see the *Oracle Business Intelligence Data Warehouse Administration Console User's Guide*.
- Make sure you have set up the DAC system properties, Informatica PowerCenter Services, and database connections according to the instructions in the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

In the procedure below you will manually move the customizations in the existing DAC Repository into the new, default version 7.9.6 DAC Repository.

To manually move your customizations from the previous DAC Repository into the default version 7.9.6 DAC Repository

1. Make a copy of the default source system container. (You cannot change the metadata in the default container. You must make a copy of it in order to be able to modify the objects it contains.) For instructions, see the *Oracle Business Intelligence Data Warehouse Administration Console User's Guide*.
2. In the copy of the source system container, import the custom data warehouse tables:
 - a. Navigate to the Tables tab in the Design view.
 - b. Right-click and select Import from Database, and then click Import Database Tables.
 - c. In the Data Sources drop-down list, select DataWarehouse.
 - d. Click Read Tables.
 - e. Select the tables you want to import, and then click Import Tables.
3. Import the custom columns for the tables you imported in Step 3:
 - a. In the Tables tab, query for the tables you imported in Step 3.
 - b. Right-click and select Import from Database, and then click Import Database Columns.
 - c. In the Importing Columns... dialog box, select "All records in the list," and click OK.
 - d. In the Data Sources drop-down list, select DataWarehouse.
 - e. Click Read Columns.
 - f. Select the columns you want to import, and then click Import Columns.
 - g. Navigate to the Tables tab, and then click the Columns child tab, and add the Foreign Key to Table and Foreign Key to Column attributes for the newly imported columns.
4. Import the indices for the custom data warehouse tables.
 - a. In the Tables tab, query for the tables you imported in Step 3.
 - b. Right-click and select Import Indices.
 - c. In the Importing Indices... dialog box, select "All records in the list," and click OK.
 - d. In the Data Sources drop-down list, select DataWarehouse.
 - e. Click Read Indices.
 - f. Select the indices you want to import, and then click Import Indices.
5. Import the custom source tables:
 - a. Navigate to the Tables tab in the Design view.
 - b. Right-click and select Import from Database, and then click Import Database Tables.
 - c. In the Data Sources drop-down list, select the appropriate source.

- d. Click Read Tables.
- e. Select the tables you want to import, and then click Import Tables.
You do not have to import columns for the custom source tables. Columns for source tables are not required.
6. Import columns for any standard data warehouse tables that were extended, and add the appropriate attributes in the DAC.
7. Create new custom logical and physical task folders.
 - a. In the DAC, from the Tools menu, select Seed Data, then select Task Folders.
 - b. To create a custom logical folder, click New.
 - c. In the Name field, enter a name for the custom logical folder, for example, Custom Logical.
 - d. In the Type field, select Logical.
 - e. To create a custom physical folder, click New.
 - f. In the Name field, enter a name for the custom physical folder, for example, Custom Physical.
 - g. In the Type field, select Physical.
8. Register the folders you created in Step 8 in the Source System Folders tab.
 - a. In the Design view, select the Source System Folders tab.
 - b. Click New.
 - c. Enter the name of the Custom Logical folder in the Logical Folder field.
 - d. Enter the name of the Custom Physical folder in the Physical Folder field, and save the record.
9. Modify the task attributes for workflows in the custom folder that are modified standard mappings, that is, standard (out-of-the-box) mappings from the previous release that you copied into the Informatica custom folder.
 - a. Navigate to the Tasks tab and query for the Informatica workflow names that are in the Informatica custom folder under Command for Incremental Load or Command for Full Load.
You must review the workflows in the custom folder in Informatica Workflow Manager.
 - b. For each task, change the Folder Name (in the Edit child tab) to the Custom Logical folder name.
 - c. For each task, right-click and select Synchronize Tasks.
 - d. In the Task Synchronization dialog box, select Selected Record Only, and click OK.
This step adds source and target tables to the task.
 - e. Click Yes in the Synchronizing Task(s)... dialog box to proceed.
An informational message will indicate the results of the process.
 - f. Click OK.
10. Add tasks that were created as new tasks in the current implementation.
 - a. In the Design view, select the Tasks tab, and then select the Edit child tab.

- b. For each new task, copy the names of the Informatica workflows into the fields Command for Incremental Load and Command for Full Load.
- c. Assign the appropriate values for the remaining fields in the Edit child tab.
- d. In the Tasks tab, query for the tasks you entered in Step b.
- e. Right-click the list of query results, and in the Task Synchronization dialog box, select "All records in the list," and click OK.

This step adds source tables and target tables to the task.

- f. Click Yes in the Synchronizing Task(s)... dialog box to proceed.
An informational message will indicate the results of the process.
- g. Click OK.

Note: If your customizations included new fact tables, you will need to create and assemble new subject areas as well as create and build new execution plans. If your customizations included extending dimension tables, you will need to reassemble your existing subject areas and rebuild your existing execution plans. You will also need to set the appropriate execution plan attributes, such as Prune Days. For instructions, see *Oracle Business Intelligence Data Warehouse Administration Console User's Guide*.

4.7 Overview of Upgrading the Data Warehouse Schema and Migrating Data

The process for upgrading the data warehouse schema and migrating data for Siebel Business Analytics releases 7.7, 7.7.x, and 7.8.x involves multiple phases, as described below.

Note: The Oracle BI Applications upgrade logic is coded such that you must complete each of the phases listed below. To upgrade to the current release of Oracle BI Applications, you cannot skip any of the phases.

■ Phase 1

In phase 1, you will perform the following steps:

- 1. Upgrade the data warehouse schema to version 7.9.0. For instructions, see [Section 4.8](#).
- 2. Import new schema definitions into the Siebel transactional database. For instructions, see [Section 4.9](#).
- 3. Migrate data into the data warehouse upgraded to version 7.9.0. For instructions, see [Section 4.10](#).

■ Phase 2

In phase 2, you will upgrade your data warehouse schema to version 7.9.4 and migrate data into the upgraded data warehouse. For instructions, see [Section 4.11](#).

■ Phase 3

In phase 3, you will upgrade your data warehouse schema to version 7.9.5 and migrate data into the upgraded data warehouse. For instructions, see [Section 4.12](#).

- **Phase 4**

In phase 4, you will upgrade your data warehouse schema to version 7.9.5.1. For instructions, see [Section 4.13](#).

- **Phase 5**

In phase 5, you will upgrade your data warehouse schema to version 7.9.6 and migrate data. For instructions, see [Section 4.14](#).

4.8 Upgrading the Data Warehouse Schema to Version 7.9.0

This procedure adds new tables, columns, and indexes to the existing data warehouse schema. It also modifies the existing data warehouse schema objects.

To upgrade the data warehouse schema

1. Run the UPGRADE.ctl script.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\CTLFiles, and locate the UPGRADE.ctl file.
 - b. If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)
 - c. Use the DDLimp utility to run the UPGRADE.ctl script. Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\UPGRADE.CTL>
/L <..\oracleBI\dwrep\UPGRADE.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C SIEBEL_OLAP /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\UPGRADE.CTL /L C:\OracleBI\dwrep\UPGRADE.log
```

Notes:

- * /P <PASSWORD> - The password for the data warehouse.
- * /C <ODBC connect string> - The name of the ODBC connect string.
- * /I N - Tells DDLimp to ignore the indexes DDL defined in the CTL file if any exist. It does not change existing indexes. (The default is /I Y, which tells DDLimp to create and merge indexes from the CTL file with the indexes in the database.
- * For Oracle databases, use the Data Direct drivers.

In addition, you can use the following commands:

- * /W Y - If the OLAP database is Oracle and Unicode.
 - * /Z Y - If the OLAP database is DB2 or SQL Server and Unicode.
 - * /B <TABLE_SPACE_NAME> - If you want to create these tables in a separate table space. For DB2, This must be specified as 32K tablespace.
 - * /X <INDEX_TABLE_SPACE_NAME> - If you want to create the indexes in a separate tablespace. For DB2, This must be specified as 32K tablespace.
 - * /Y - Storage File for DB2/390.
 - * /R - Regrant tables.
2. Run the 790_UPGRADE_PRE_CTL_SCRIPT.sql script.
 - a. Open the SQL client for your database type, for example, SQLPLUS for Oracle, Query Analyzer for SQL Server, or a command window for DB2.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 790_UPGRADE_PRE_CTL_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
 3. Run the DW.ctl script.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\CTLFiles, and locate the DW.ctl file.
 - b. If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)
 - c. Use the DDLimp utility to run the DW.ctl script.
Use the command provided in Step 1, but substitute the correct script name.
 4. Run the 790_UPGRADE_PRE_DIMENSION_SCRIPT.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 790_UPGRADE_PRE_DIMENSION_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.

4.9 Importing New Schema Definitions into the Siebel Transactional Database

This procedure upgrades transactional database objects that relate to Oracle Business Intelligence Applications. It does not upgrade transactional database objects for Siebel CRM applications.

To import new schema definitions into the Siebel transactional database

1. Import schema definitions for non-image tables:
 - a. Access the DDLimp utility.
 - b. Run the DDL_OLTP.ctl script that is located in the OracleBI\dwrep\Upgrade\CTLFiles folder.
2. Import schema definitions for image tables:
 - a. In DAC, go to the Design view, and select your custom container from the drop-down list to the right of the Execute button.
 - b. Click the Tables tab.
 - c. Query for all tables for which the image suffix is not null.
 - d. Right-click in the list of tables returned by the query, and select Change Capture Scripts, and then Generate Image and Trigger Scripts.
 - e. In the Triggers and Image Tables dialog box, do the following:
 - Select the option All Tables in the List.
 - Select the option Generate Image Table Scripts.
 - Select the appropriate database type for the source system.
 - Click OK.
 - f. Open the SQL client for the source system database type, for example, SQL Plus for Oracle, Query Analyzer for SQL Server, or a command window for DB2.

The script may contain many lines; therefore, you can save the script file as a SQL file and execute it in a SQL client.
 - g. Copy the scripts generated by DAC into the SQL client and execute them.

4.9.1 Verifying the Siebel Transactional Database Upgrade

Follow this procedure to verify the following tables were created in the Siebel transactional database.

To verify the transactional database upgrade

- For all upgrade paths, verify the following tables were created in the Siebel transactional database:
 - S_ETL_R_IMG_XXX
 - S_ETL_I_IMG_XXX
 - S_ETL_D_IMG_XXX
 - S_ETL_PARAM
 - S_ETL_PRD_ATTR
 - S_ETL_PRD_REL

4.10 Migrating Data into the Data Warehouse Upgraded to Version 7.9.0

Follow this procedure to migrate data into the upgraded data warehouse.

To migrate data into the upgraded data warehouse

1. Copy the file Upgrade.rep file from the folder OracleBI\dwrep\Upgrade\Informatica\Repository into the folder Informatica PowerCenter 8.6\server\infa_shared\Backup.
2. Launch the Informatica PowerCenter Administration Console and restore Upgrade.rep (located in Informatica PowerCenter 8.6\server\infa_shared\Backup) into a database other than the database in which you restored Oracle_BI_DW_Base.rep.
3. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\SrcFiles and copy the *.csv files into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
4. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\ParameterFiles and copy the file <previous version of Siebel Analytics>_TENERIFE_UPG_PARAMS.txt into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
5. Rename the file <previous version of Siebel Analytics>_TENERIFE_UPG_PARAMS.txt to TENERIFE_UPG_PARAMS.txt.
6. In the file TENERIFE_UPG_PARAMS.txt:
 - a. Search for the parameter \$\$SourceConnection.
 - b. Set the value to match your Siebel applications (OLTP) version:
 - * SEBL_63
 - * SEBL_753
 - * SEBL_771
 - * SEBL_78
 - * SEBL_80 (use this value for OLTP versions 8.0 and 8.1.1)
 - * SEBL_VERT_753
 - * SEBL_VERT_771
 - * SEBL_VERT_78
 - * SEBL_VERT_80 (use this value for OLTP versions 8.0 and 8.1.1)

For example, if your Siebel applications (OLTP) version is Siebel Industry Applications 7.8, the parameter should appear as:

```
$$SourceConnection=SEBL_VERT_78
```
7. In the file TENERIFE_UPG_PARAMS.txt:
 - a. Search for the parameter \$\$Source_Container.
 - b. Set the value to match your Siebel applications (OLTP) version:
 - * Siebel 6.3
 - * Siebel 7.5.3

- * Siebel 7.7.1
- * Siebel 7.8
- * Siebel 8.0 (use this value for OLTP versions 8.0 and 8.1.1)
- * Siebel 7.5.3 Vertical
- * Siebel 7.7.1 Vertical
- * Siebel 7.8 Vertical
- * Siebel 8.0 Vertical (use this value for OLTP versions 8.0 and 8.1.1)

For example, if your Siebel applications (OLTP) version is Siebel Industry Applications 7.8, the parameter should appear as:

```
$$Source_Container=Siebel 7.8 Vertical
```

8. In the file TENERIFE_UPG_PARAMS.txt, edit the ETL_PROC_WID parameter as follows:

```
MPLT_GET_ETL_PROC_WID.$$ETL_PROC_WID=<latest ETL_PROC_WID value from your database>
```

You can get this value from W_PARAM_G.ETL_PROC_WID.

9. If you are running Siebel Industry Applications (Vertical), in the file TENERIFE_UPG_PARAMS.txt, set the VERTICAL_UPGRADE parameter to 1. For example:

```
$$VERTICAL_UPGRADE=1
```

10. In the Informatica Workflow Manager, open the Relational Connection Browser (in the menu bar, select Connections, and then select Relational), and edit the connect string, user and password for the relational connections as follows:
 - a. Edit the PARAM_OLTP_SIEBEL connection to point to the Siebel transactional database.
 - b. Edit the DataWarehouse connection to point to the newly upgraded data warehouse database.
 - c. Create or edit the PARAM_DAC_OLD connection to point to the previous DAC Repository database (the version from which you are upgrading).
 - d. Create or edit the PARAM_DAC_NEW connection to point to the DAC Repository database.

Note: If you are connected to an Oracle database, use the Oracle Native driver instead of ODBC.

If you are connected to a SQL Server database, use the ODBC driver rather than the native SQL Server driver.

11. If you are upgrading from Siebel Industry Applications (Vertical), define the alignment rule to be used for ETL loads.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\SrcFiles and copy the file AlignmentType_LS_782.csv into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
 - b. Rename the file AlignmentType_LS_782.csv to AlignmentType.csv.

- c. Open the file AlignmentType.csv, and enter one of the following alignment item types under the ALIGN_TYPE field:

- * Zipcode
- * Account/Contact
- * Brick

- d. Save the file.

The Oracle Business Analytics Warehouse allows only one alignment type to be used for ETL loads during upgrade.

12. In Informatica Workflow Manager, navigate to the appropriate folder and execute the workflows in the order specified:

Application Version	Folder Name	Workflow
7.7 to 7.9	UPGRADE_770_to_79	<ul style="list-style-type: none"> ■ MARKETING_LOAD ■ Upgrade_Dimensions_Industry ■ Update_Dimensions ■ Update_Dimension_Unspecified ■ Update_Facts ■ Upgrade_LS_Dimensions ■ Upgrade_LS_Facts ■ SIL_PositionDimensionHierarchy_Full ■ Load_INT_ORG_DH ■ DIMENSION_LOAD <p>Note: If you are using the SCD version of the dimension, replace the corresponding TENN_UPG_W_XXX_D_784_To_W_XXX_D session with TENN_UPG_W_XXX_D_784_SCD_To_W_XXX_D. This will upgrade the data from the W_XXX_SCD version of the dimension to the new SCD-enabled W_XXX_D dimension.</p> <ul style="list-style-type: none"> ■ DIMENSION_UNSPECIFIED_UPDATE ■ FACT_UPDATE ■ DAC_Metadata_Upgrade_Workflow

Application Version	Folder Name	Workflow
7.7.1.x to 7.9	UPGRADE_771_to_79	<ul style="list-style-type: none"> Update_Dimensions Update_Dimension_Unspecified Update_Facts Upgrade_LS_Dimensions Upgrade_LS_Facts SIL_PositionDimensionHierarchy_Full Load_INT_ORG_DH DIMENSION_LOAD <p>Note: If you are using the SCD version of the dimension, replace the corresponding TENN_UPG_W_XXX_D_784_To_W_XXX_D session with TENN_UPG_W_XXX_D_784_SCD_To_W_XXX_D. This will upgrade the data from the W_XXX_SCD version of the dimension to the new SCD-enabled W_XXX_D dimension.</p> <ul style="list-style-type: none"> DIMENSION_UNSPECIFIED_UPDATE FACT_UPDATE DAC_Metadata_Upgrade_Workflow
7.8 to 7.9	UPGRADE_78_to_79	<ul style="list-style-type: none"> SIL_PositionDimensionHierarchy_Full Load_INT_ORG_DH DIMENSION_LOAD <p>Note: If you are using the SCD version of the dimension, replace the corresponding TENN_UPG_W_XXX_D_784_To_W_XXX_D session with TENN_UPG_W_XXX_D_784_SCD_To_W_XXX_D. This will upgrade the data from W_XXX_SCD version of the dimension to the new SCD enabled W_XXX_D dimension.</p> <ul style="list-style-type: none"> DIMENSION_UNSPECIFIED_UPDATE FACT_UPDATE DAC_Metadata_Upgrade_Workflow

13. If you upgraded your transactional database to Siebel Applications 8.0 or 8.1.1, navigate to the folder UPGRADE_790_to_791_SBL80UPG and run the following workflows in the order indicated:

- a. UPGRADE_DIMENSIONS
- b. UPGRADE_FACTS

14. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the main Informatica Repository (Oracle_BI_DW_Base.rep).

- a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
- b. Open the reset_infa_seq_gen.bat file.

The file opens a command prompt, which will prompt you for the parameters listed below.

- c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)
15. To verify the data migrated successfully:
 - a. Check whether any of the Informatica mapping log files stored in the directory \Informatica PowerCenter 8.6\server\infa_shared\SessLogs indicates errors or failures.
 - b. Check whether the SQL scripts (790_UPGRADE_PRE_CTL_SCRIPT.sql and 790_UPGRADE_PRE_DIMENSION_SCRIPT.sql) that you ran in the SQL client of the database failed or errored out while executing.
 - c. Check the log files for the CTL files (Upgrade.ctl and DW.ctl) that you ran using the DDLimp command to determine whether any of the runs failed.

If you did not detect any errors or failures in the steps above, then the data migration was successful.

Note: Ignore any error messages that indicate tables or indexes could not be dropped from the database. These error messages are displayed when the specified objects do not exist in the database. This can occur because each source system container holds a subset of all data warehouse tables.

16. If the data migration was successful, drop the tables that were created during the upgrade process, such as, W_XXXX_X_784 tables, LKP_XXXX_X, and W_ASSET_D_TMP.

This step frees the space occupied by these backup tables.

4.11 Upgrading the Data Warehouse Schema to Versions 7.9.4 and Migrating Data

Follow this procedure to upgrade the data warehouse schema to version 7.9.4 and migrate data.

To upgrade the data warehouse schema and migrate data

1. Run the 792_UPGRADE_PRE_CTL_SCRIPT.sql script.
 - a. Open the SQL client for your database type, for example, SQLPLUS for Oracle, Query Analyzer for SQL Server, or a command window for DB2.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 792_UPGRADE_PRE_CTL_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
2. Use the DDLimp utility to run the ddl_794.ctl script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder. :

If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to

make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddl_794.ctl>
/L <..\OracleBI\dwrep\ddl_794.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\ddl_794.ctl /L C:\OracleBI\dwrep\ddl_794.log
```

3. Run the 792_UPGRADE_PRE_DIMENSION_SCRIPT.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 792_UPGRADE_PRE_DIMENSION_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
4. If you upgraded your transactional database to Siebel Applications 8.0 or 8.1.1, run the workflows to migrate your data into the upgraded data warehouse.
 - a. In Informatica Workflow Manager, navigate to the folder UPGRADE_790_TO_791_SBL80UPG.
 - b. Run the following workflows in the order indicated:
 - UPGRADE_DIMENSIONS
 - UPGRADE_FACTS
5. Verify the data migrated successfully by checking whether any of the Informatica mapping log files stored in the \Informatica PowerCenter 8.6\server\infa_shared\SessLogs directory indicate errors or failures.
6. If the data migration was successful, drop the tables that were created during the upgrade process, such as, W_xxxx_x_79x tables, LKP_xxxx_x and 79x_XXXX_TMP. This step frees the space occupied by these backup tables.
7. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the main Informatica Repository (Oracle_BI_DW_Base.rep).
 - a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.

The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)

4.12 Upgrading the Data Warehouse Schema to Version 7.9.5 and Migrating Data

Follow this procedure to upgrade the data warehouse schema to version 7.9.5 and migrate data.

To upgrade the data warehouse schema and migrate data

1. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the Upgrade repository.
 - a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.

The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)
2. Run the 795_UPGRADE_PRE_CTL_SCRIPT.sql script.
 - a. Open the SQL client for your database type, for example, SQLPLUS for Oracle, Query Analyzer for SQL Server, or a command window for DB2.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 795_UPGRADE_PRE_CTL_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
3. Run the UPGRADE_795.ctl script.

This script adds temp tables for the upgrade process.

- a. Navigate to the folder OracleBI\dwrep\Upgrade\CTLFiles, and locate the UPGRADE_795.ctl file.
- b. If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)
- c. Use the DDLimp utility to run the UPGRADE_795.ctl script. Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\UPGRADE_795.CTL>
/L <..\OracleBI\dwrep\UPGRADE.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
```



```
/I N /R Y /F C:\OracleBI\dwrep\UPGRADE_795.CTL /L
C:\OracleBI\dwrep\UPGRADE.log
```

Notes:

- * /P <PASSWORD> - The password for the data warehouse.
 - * /C <ODBC connect string> - The name of the ODBC connect string.
 - * /I N - Tells DDLimp to ignore the indexes DDL defined in the CTL file if any exist. It does not change existing indexes. (The default is /I Y, which tells DDLimp to create and merge indexes from the CTL file with the indexes in the database.
 - * For Oracle databases, use the Data Direct drivers.
- In addition, you can use the following commands:
- * /W Y - If the OLAP database is Oracle and Unicode.
 - * /Z Y - If the OLAP database is DB2 or SQL Server and Unicode.
 - * /B <TABLE_SPACE_NAME> - If you want to create these tables in a separate table space. For DB2, This must be specified as 32K tablespace.
 - * /X <INDEX_TABLE_SPACE_NAME> - If you want to create the indexes in a separate tablespace. For DB2, This must be specified as 32K tablespace.
 - * /Y - Storage File for DB2/390.
 - * /R - Regrant tables.

4. Use the DDLimp utility to run the ddl_795.ctl script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder.

If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddl_795.ctl>
/L <..\OracleBI\dwrep\ddl_795.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\ddl_795.ctl /L C:\OracleBI\dwrep\ddl_795.log
```

5. Run the 795_UPGRADE_PRE_DIMENSION_SCRIPT.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 795_UPGRADE_PRE_DIMENSION_SCRIPT.sql file, and copy the contents into the SQL client.

- d. Execute the script.
6. Copy all of the domain value files in the folder \OracleBI\dwrep\Informatica\LkpFiles into the folder \Informatica PowerCenter 8.6\server\infa_shared\LkpFiles.
7. Migrate data into the upgraded data warehouse.
 - a. Copy the file Upgrade.rep file from the folder OracleBI\dwrep\Upgrade\Informatica\Repository into the folder Informatica PowerCenter 8.6\server\infa_shared\Backup.
 - b. Launch the Informatica PowerCenter Administration Console and restore Upgrade.rep (located in Informatica PowerCenter 8.6\server\infa_shared\Backup).
 - c. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\ParameterFiles and copy the file 795_UPG_PARAMS.txt into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
 - d. Set the parameter \$\$ETL_PROC_WID to the latest ETL_PROC_WID value from the database. You can get this value from W_PARAM_G.ETL_PROC_WID.
 - e. Set the parameter \$\$DATASOURCE_NUM_ID to the relevant value from the source system setup.
 - f. In Informatica Workflow Manager, open the Relational Connection Browser (in the menu bar, select Connections, and then select Relational), and edit the connect string, user and password for the relational connections as follows: create one relational connection based on the appropriate database platform for your OLTP database. Create the connection with the name PARAM_OLTP. Edit PARAM_OLTP connection to match your OLTP environment. Edit the PARAM_OLAP connection to match your OLAP environment.

Note: If you are connected to an Oracle database, use the Oracle Native driver instead of ODBC.

If you are connected to a SQL Server database, use the ODBC driver rather than the native SQL Server driver.

- g. In Informatica Workflow Manager, navigate to the folder UPGRADE_794_to_795_SBL and execute the UPGRADE_DIMENSIONS workflow.
8. Verify the data migrated successfully.
 - a. Check whether any of the Informatica mapping log files stored in the \Informatica PowerCenter 8.6\serva\infa_shared\Sesslogs directory indicates errors or failures.
 - b. Check whether the script 795_UPGRADE_PRE_DIMENSION_SCRIPT.sql that you ran in the SQL client of the database failed or errored out while executing.
 - c. Check the log file for the script Upgrade_795.ctf that you ran using the DDLimp command to determine whether any of the runs failed.

If you did not detect any errors or failures in the steps above, then the data migration was successful.

Note: Ignore any error messages that indicate tables or indexes could not be dropped from the database. These error messages are displayed when the specified objects do not exist in the database. This can occur because each source system container holds a subset of all data warehouse tables

9. If the data migration was successful, drop the tables that were created during the upgrade process, such as W_xxxx_x_79x, LKP_xxxx_x and 79x_xxxx_TMP.

This step frees the space occupied by these backup tables.

10. Once the data migration steps above are complete, you can delete the Upgrade repository to avoid any accidental use or confusion.
 - a. In the Informatica PowerCenter Administration Console, select the Upgrade repository service.
 - b. In the General Properties area of the Properties tab, click Edit .
 - c. Make sure the operating mode of the repository service is set to Exclusive.
 - d. Click OK.
 - e. Choose Actions, and then click Delete Contents.
 - f. In the Delete contents for <repository name> dialog, enter the repository username and password (for example, Administrator/Administrator), then click OK.
11. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the main Informatica Repository (Oracle_BI_DW_Base.rep).
 - a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.
The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)

4.13 Upgrading the Data Warehouse Schema to Version 7.9.5.1

Follow this procedure to upgrade the data warehouse schema to version 7.9.5.1.

To upgrade the data warehouse schema

- Use the DDLimp utility to run the ddl_7951.ctl script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder.

If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL

file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>  
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddl_7951.ctl>  
/L <..\OracleBI\dwrep\ddl_7951.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE  
/I N /R Y /F C:\OracleBI\dwrep\ddl_7951.ctl /L C:\OracleBI\dwrep\ddl_7951.log
```

4.14 Upgrading the Data Warehouse Schema to Version 7.9.6 and Migrating Data

Follow this procedure to upgrade the data warehouse schema to version 7.9.6 and migrate data.

This section includes the following procedures:

- [Section 4.14.1, "Upgrading the Data Warehouse Schema to Version 7.9.6"](#)
- [Section 4.14.2, "Migrating Data into the Upgraded Data Warehouse"](#)
- [Section 4.14.3, "Verifying the Data Migrated Successfully"](#)
- [Section 4.14.4, "Resetting Refresh Dates"](#)

4.14.1 Upgrading the Data Warehouse Schema to Version 7.9.6

Follow this procedure to migrate data into the upgraded data warehouse.

To migrate data into the upgraded data warehouse

1. Run the 796_UPGRADE_DROP_INDEXES.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 796_UPGRADE_DROP_INDEXES.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
2. Run the UPGRADE_796.ctl script.

This script adds temp tables for the upgrade process.

- a. Navigate to the folder OracleBI\dwrep\Upgrade\CTLFiles, and locate the UPGRADE_796.ctl file.
- b. If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the

preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

- c. Use the DDLimp utility to run the UPGRADE_796.ctf script. Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect
string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\UPGRADE_796.CTL>
/L <..\oracleBI\dwrep\UPGRADE.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\UPGRADE_796.CTL /L
C:\OracleBI\dwrep\UPGRADE.log
```

Notes:

- * /P <PASSWORD> - The password for the data warehouse.
 - * /C <ODBC connect string> - The name of the ODBC connect string.
 - * /I N - Tells DDLimp to ignore the indexes DDL defined in the CTL file if any exist. It does not change existing indexes. (The default is /I Y, which tells DDLimp to create and merge indexes from the CTL file with the indexes in the database.
 - * For Oracle databases, use the Data Direct drivers.
- In addition, you can use the following commands:
- * /W Y - If the OLAP database is Oracle and Unicode.
 - * /Z Y - If the OLAP database is DB2 or SQL Server and Unicode.
 - * /B <TABLE_SPACE_NAME> - If you want to create these tables in a separate table space. For DB2, This must be specified as 32K tablespace.
 - * /X <INDEX_TABLE_SPACE_NAME> - If you want to create the indexes in a separate tablespace. For DB2, This must be specified as 32K tablespace.
 - * /Y - Storage File for DB2/390.
 - * /R - Regrant tables.

3. Run the 796_UPGRADE_PRE_CTL_SCRIPT.sql script.
 - a. Open the SQL client for your database type, for example, SQLPLUS for Oracle, Query Analyzer for SQL Server, or a command window for DB2.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 796_UPGRADE_PRE_CTL_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Find and replace the Datasource_Num_id = -1 with the correct Datasource_Num_id from your existing implementation. (The value -1 is a dummy value.)
 - e. Execute the script.
4. Use the DDLimp utility to run the ddl_796.ctf script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder.

If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddl_796.ctl>
/L <..\OracleBI\dwrep\ddl_796.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\ddl_796.ctl /L C:\OracleBI\dwrep\ddl_796.log
```

5. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the Upgrade repository.
 - a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.

The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)
6. Run the 796_UPGRADE_PRE_DIMENSION_SCRIPT.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 796_UPGRADE_PRE_DIMENSION_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
7. Copy all of the domain value files in the folder \OracleBI\dwrep\Informatica\LkpFiles into the folder \Informatica PowerCenter 8.6\server\infa_shared\LkpFiles.

4.14.2 Migrating Data into the Upgraded Data Warehouse

Follow this procedure to migrate data into the upgraded data warehouse.

To migrate data into the upgraded data warehouse

1. Copy the file Upgrade.rep file from the folder OracleBI\dwrep\Upgrade\Informatica\Repository into the folder Informatica PowerCenter 8.6\server\infa_shared\Backup.

2. Launch the Informatica PowerCenter Administration Console and restore Upgrade.rep (located in Informatica PowerCenter 8.6\server\infa_shared\Backup).
3. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\ParameterFiles and copy the file 796_UPG_PARAMS.txt into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
4. In the 796_UPG_PARAMS.txt file, set the following parameters:
 - a. \$\$ETL_PROC_WID. Set this parameter to the relevant value from the source system setup. You can get this value from W_PARAM_G.ETL_PROC_WID
 - b. \$\$DATASOURCE_NUM_ID. Set this parameter to the relevant value from the source system setup.
 - c. \$\$INITIAL_EXTRACT_DATE. Set this parameter to the initial extraction data of the data warehouse.
 - d. \$\$WH_DATASOURCE_NUM_ID. Set this parameter to the data source number ID you have set up for the data warehouse.
 - e. \$\$MASTER_ORG. Get this value from the Source System Parameters tab in DAC.
 - f. \$\$INV_PROD_CAT_SET_ID1. Get this value from the Source System Parameters tab in DAC.
 - g. \$\$PROD_CAT_SET_ID1. Get this value from the Source System Parameters tab in DAC.
 - h. Set the parameter \$\$IS_SOURCE_PRE_80 to Y if your source OLTP application was on a version prior to Siebel 8.0 before you began the upgrade process. Otherwise, set this parameter to N.
5. Configure common parameters specific to Siebel source systems. For instructions, see [Section B.2, "Configuring Common Parameters for Siebel Source Systems."](#)
6. In Informatica Workflow Manager, open the Relational Connection Browser (in the menu bar, select Connections, and then select Relational), and edit the connect string, user and password for the relational connections as follows:
 - a. Edit the connection PARAM_OLTP_SIEBEL to match your OLTP environment.
 - b. Edit the connection PARAM_OLAP to match your OLAP environment.
 - c. Edit the connection PARAM_DAC to match your DAC database.

Note: If you are connected to an Oracle database, use the Oracle Native driver instead of ODBC.

If you are connected to a SQL Server database, use the ODBC driver rather than the native SQL Server driver.

7. In Informatica Workflow Manager, navigate to the folder UPGRADE_7951_to_796_SBL and execute the UPGRADE_DIMENSIONS and UPGRADE_FACTS workflows.
8. Run the 796_UPGRADE_POST_SCRIPT.sql.
 - a. Open the SQL client for your database type.

- b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
- c. Open the 796_UPGRADE_POST_SCRIPT.sql file, and copy the contents into the SQL client.
- d. Execute the script.

4.14.3 Verifying the Data Migrated Successfully

Follow this procedure to verify that the data was migrated successfully into the upgraded data warehouse.

To verify the data migrated successfully

1. Check whether any of the Informatica mapping log files stored in the \Informatica PowerCenter 8.6\serva\infa_shared\Sesslogs directory indicates errors or failures.
2. Check whether the following scripts that you ran in the SQL client of the database failed or errored out while executing:
796_UPGRADE_DROP_INDEXES.sql
796_UPGRADE_PRE_CTL_SCRIPT.sql
796_UPGRADE_PRE_DIMENSION.sql
796_UPGRADE_POST_SCRIPT.sql
3. Check the log files for the ddl_796.ctl and Upgrade_796.ctl scripts that you ran using the DDLimp command to determine whether any of the runs failed.

If you did not detect any errors or failures in the steps above, then the data migration was successful.

Note: Ignore any error messages that indicate tables or indexes could not be dropped from the database. These error messages are displayed when the specified objects do not exist in the database. This can occur because each source system container holds a subset of all data warehouse tables

4. If the data migration was successful, drop the tables that were created during the upgrade process, such as W_XXXX_X_79X, LKP_XXXX_X, 79X_XXXX_TMP, and TMP_XXXX_79X.

This step frees the space occupied by these backup tables.

4.14.4 Resetting Refresh Dates

After verifying the data was migrated successfully into the upgraded data warehouse, follow this procedure to reset refresh dates.

1. In Informatica Workflow Manager, navigate to the folder UPGRADE_7951_TO_796_SBL and execute the RESET_DAC_REFRESH_DATES workflow.
2. Once the data migration steps above are complete, you can delete the Upgrade repository to avoid any accidental use or confusion.
 - a. In the Informatica PowerCenter Administration Console, select the Upgrade repository service.

- b. In the General Properties area of the Properties tab, click Edit .
- c. Make sure the operating mode of the repository service is set to Exclusive.
- d. Click OK.
- e. Choose Actions, and then click Delete Contents.
- f. In the Delete contents for <repository name> dialog, enter the repository username and password (for example, Administrator/Administrator), then click OK.

4.15 Upgrading the Siebel Analytics Repository

This process merges your customizations of a prior release of the Siebel Analytics repository with the new version of the Oracle BI repository. Before you begin this process, make sure you have backed up and renamed your existing repository.

To upgrade the repository, perform the following tasks:

- [Section 4.15.1, "Preparing for the Oracle BI Repository Upgrade"](#)
- [Section 4.15.2, "Equalizing the Oracle BI Repositories"](#)
- [Section 4.15.3, "Comparing the Siebel Analytics and Oracle BI Repositories"](#)
- [Section 4.15.4, "Merging the Siebel Analytics and Oracle BI Repositories"](#)
- [Section 4.15.5, "Regression Testing the Oracle BI Repository Merge"](#)

The tasks in this section refer to multiple versions of the Siebel Analytics and Oracle BI repositories. [Table 4–1](#) provides the names and descriptions of the repositories used in the examples in this section.

Table 4–1 Names of Analytics Repositories used in Examples

Name of Repository	Description
SiebelAnalytics_7x.rpd	The standard Siebel Analytics repository for the version you are upgrading from. This repository is referred to as the "original" repository in the examples in this section. Note: Standard repositories from previous releases are available in the folder \OracleBI\Upgrade.
OracleBIAnalyticsApps.rpd	The standard Oracle BI repository for the version you are upgrading to.
Customer_SiebelAnalytics.rpd	The Siebel Analytics repository that contains your customizations for the version you are upgrading from.
Merged_Repository_OracleBI.rpd	The Oracle BI repository that contains your customizations for the version you are upgrading to.

4.15.1 Preparing for the Oracle BI Repository Upgrade

Follow this procedure to prepare for the repository upgrade.

To prepare for the Analytics repository upgrade

1. Set up a directory for the merge process, such as \OracleBIUpgrade, and create the following subfolders:
 - Original
 - AfterEqualize

- AfterMerge
 - AfterManualWork
 - AfterRegressions
2. Copy the original repository (for example, SiebelAnalytics_7x.rpd), the production repository (for example, Customer_SiebelAnalytics.rpd, and the repository from the latest installation (for example, OracleBIAnalyticsApps.rpd) into the folder \OracleBIUpgrade\Original.

If, in your current environment, you are running Siebel Analytics for one or more modules using a Siebel Analytics repository in which you extracted the corresponding projects for the modules from the standard Siebel Analytics repository file you received from the previous release, you need to extract the same projects from the SiebelAnalytics_7x.rpd file and use this as your original repository. (If you have the original repository that you extracted during the last upgrade, you can use it as the original repository file.) This will prevent you from losing any new metadata you would like to add in this upgrade.

Also, if you customized the Siebel Analytics repository by trimming a large number of objects and you would like to get those objects back during the current upgrade, you need to trim the SiebelAnalytics_7x.rpd file in the same way and use the modified version as the original repository file. This will prevent you from losing any new metadata you would like to add in this upgrade.

4.15.2 Equalizing the Oracle BI Repositories

The Merge feature in the Administration Tool relies on a change detection algorithm to determine the changes that need to be made to upgrade repositories correctly. For the algorithm to work correctly, it has to determine which objects in the three repositories (for example, SiebelAnalytics_7x.rpd, OracleBIAnalyticsApps.rpd, and Customer_SiebelAnalytics.rpd) are equivalent.

The point of this step is to determine for every object in the OracleBIAnalyticsApps.rpd and the Customer_SiebelAnalytics.rpd whether it is coming from the SiebelAnalytics_7x.rpd.

Equivalence between objects is established using the Administration Tool's Equalize feature. The file that you specify in the Output option (-O) is the only file that is modified during the equalization process.

The Equalize feature has several mechanisms for determining whether an object in two different repositories is semantically the same:

- **Fully Qualified Name.** If an object in one repository has the same fully qualified name as another object of the same class in another repository, then the two objects are declared equal.
- **Simple String Substitution.** Equivalence can be declared between two objects of the same class in two repositories whose only difference is that some key characters in their names differ. The equalizerpds executable file ignores those characters while checking fully qualified names. For example, "Core"."W_DAY_D" might be considered equivalent to "Core"."W DAY D" if the characters "_" and " " have been declared as equivalent.
- **Rename File.** When none of the preceding rules are applicable, equivalence can be manually declared using a script as input to the equalizerpds executable file. Oracle ships the rename files (MAP) for the major releases. The files are located in the OracleBI\Upgrade folder. You can also create your own rename files for

customizations not covered in the files that Oracle ships. You can open and edit the rename files in Microsoft Excel.

The syntax of the equalizerpds command is as follows:

```
equalizerpds.exe -A userid1 [-B [password1]] -C base_repository_name -D userid2
[-E [password2]] -F repository2_name [-J udml_utf8_file_name_equalization] [-O
ouput_repository_name] [-X] [-Y equalStringSet]
-X          Treat 'Factxxxx' as 'Fact' in Business Model.
-Y          Treat the characters as equals.
/?          Display this usage information and exit.
```

To equalize a repository

1. Copy the appropriate MAP file from the OracleBI\Upgrade folder into the folder where you will execute equalizerpds.exe, for example, \OracleBIUpgrade\Original.
2. Run equalizerpds.exe to equalize the repository from the latest installation (for example, OracleBIAnalyticsApps.rpd) with the original repository (for example, SiebelAnalytics_7x.rpd). An example of the equalizerpds command is as follows:

```
equalizerpds -A Administrator -B SADMIN
-C \\OracleBIUpgrade\Original\SiebelAnalytics_7x.rpd
-D Administrator -E SADMIN
-F \\OracleBIUpgrade\Original\OracleBIAnalyticsApps.rpd
-O \\OracleBIAnalyticsUpgrade\AfterEqualize\OracleBIAnalyticsApps.rpd
-X -J rename7x-79.map
```

The MAP files are located in the \OracleBI\Upgrade folder.

If the equalizerpds.exe executable file runs correctly, no errors are returned.

3. Run equalizerpds.exe to equalize your customized repository (for example, Customer_SiebelAnalytics.rpd) with the original repository (for example, SiebelAnalytics_7x.rpd). An example of the equalizerpds command is as follows:

```
equalizerpds -A Administrator -B SADMIN
-C \\OracleBIUpgrade\Original\SiebelAnalytics_7x.rpd
-D Administrator -E SADMIN
-F \\OracleBIUpgrade\Original\Customer_SiebelAnalytics.rpd
-O \\OracleBIUpgrade\AfterEqualize\Customer_OracleBIAnalyticsApps.rpd
```

The execution of equalizerpds that equalizes the customer repository with the original repository does not use the rename file.

Make sure that the original repository is copied unchanged into its new location so that after running the script, all three repositories are contained within the \OracleBIUpgrade\AfterEqualize directory.

4. To verify the process completed successfully, compare the size of the repositories. The output repository (-O) should be close to the same size as the repository you equalized (-F).

4.15.3 Comparing the Siebel Analytics and Oracle BI Repositories

Follow this procedure to compare your existing repository with the new version to which you are upgrading.

To compare the repositories

- Use the Administration Tool's Compare Repositories feature to analyze the differences between your existing repository and the new version of the repository

to which you are upgrading. Note where elements have been created, removed, or changed in the new version. Consider whether you can use the new metadata and retire customizations you made in the existing repository.

For instructions on how to use the Administration Tool's Compare Repositories feature, see *Oracle Business Intelligence Server Administration Guide*.

4.15.4 Merging the Siebel Analytics and Oracle BI Repositories

In this procedure, you execute the main algorithm to upgrade the repository. For more information on merging the repositories, see *Oracle Business Intelligence Server Administration Guide*.

To merge versions of the repositories

1. Copy the three repositories (for example, SiebelAnalytics_7x.rpd, OracleBIAnalyticsApps.rpd, and Customer_SiebelAnalytics.rpd) to the AfterMerge folder.
2. Open the repository from the latest installation (for example, OracleBIAnalyticsApps.rpd) in the \OracleBIUpgrade\AfterMerge folder.
3. Save the repository with a new name, for example, Merged_Repository_OracleBIAnalyticsApps.rpd.

This new repository will contain the final results of the upgrade.

4. From the Administration Tool menu bar, select File, then select Merge.
5. In the Select Original Repository dialog box, select the original repository (for example, SiebelAnalytics_7x.rpd).
6. Enter the password, and click OK.
7. Click Select for the Modified Repository field.
8. In the Select Modified Repository dialog box, select the repository that contains the customizations you made to the previous version of the Analytics repository.
9. Click Open, type the password, and then click OK.
10. In the Decision drop-down list, select the action you want to take regarding the repository change, or accept the default action.
11. To locate subsequent rows with empty Decision fields, click the Decision header cell.

When all rows have a value in the Decision field, the Merge button is enabled.

12. Click Merge.

This process can take up to 40 minutes, depending on the size of the repositories you are working with. A message will alert you when the merge is complete.

13. Click Yes when asked if you want to run a consistency check.

The number of errors returned by the consistency check is an indication of how successful the merge process was. If you receive many errors, for example, over 300, you should analyze the reason for the errors. If the merge process failed to recognize that two objects are the same, you may need to edit the rename file if the object is in the Current repository, or add your own rename file if you have renamed many of the objects and the upgrade engine failed to relate them to the original objects.

You also may need to change the actions you selected in the Decision drop-down list before rerunning the merge. This could save you time by reducing the number of errors that you will need to fix manually.

Once you are satisfied with the results of the merge, you should fix the remaining errors manually. It is important that you fix all errors before moving on to the next step. This repository serves as the input for the next stage.

You should also check that all of your customized objects are present and that no duplicate physical tables were introduced. To check for duplicate tables, search for physical tables using a query such as:

```
where name like '*#1'
```

14. Copy the repository to the folder \OracleBIUpgrade\AfterManualWork.

4.15.5 Regression Testing the Oracle BI Repository Merge

In performing a regression test for the repository merge, the objective is to collect a set of logical SQL statements that are used for reports and to verify that they continue to work with the new metadata. For this purpose, it is recommended that you perform the following procedure.

To perform regression testing

1. Run the reports that are necessary to include in the regression suite. These reports might be a subset of the reports in the Presentation Catalog.
2. Collect the logical SQL generated in the previous step. You can do this using Usage Tracking or by parsing the query log file.

For information about Usage Tracking, see *Oracle Business Intelligence Server Administration Guide*.

3. Execute the logical SQL against the old repository using the command line utility nQCmd.exe located in \OracleBI\server\bin, and save the results to a file.

For information about the nQCmd.exe utility, see *Oracle Business Intelligence Server Administration Guide*.

4. Edit the logical SQL test scripts to account for the name changes or modifications resulting from the upgrade.
5. Execute the edited logical SQL against the merged repository, and save the results.
6. Compare the results from the steps above and try to explain the differences. If it is determined that these differences are due to the upgrade process, then you have to correct them manually.

This repository now contains the merged content from the new OracleBIAnalyticsApps.rpd and the production repository.

4.16 Upgrading the Oracle BI Presentation Catalog

You will need to upgrade your current Oracle BI Presentation Catalog if your organization:

- Has prebuilt applications already installed, and
- Has customized the current Oracle BI Presentation Catalog

If you made no changes to the previous Presentation Catalog distributed with previous versions of prebuilt applications, you do not need to upgrade the catalog. You can begin using the newer version of the catalog.

This process includes the following tasks:

- [Section 4.16.2, "Upgrading the Oracle BI Presentation Catalog to a Newer Version"](#)
- [Section 4.16.3, "Testing the Results of the Presentation Catalog Upgrade"](#)

Caution: In releases of Oracle BI Applications previous to 7.9, the Presentation Catalog (formerly known as the Siebel Analytics Web Catalog) was stored in a single file rather than in a directory structure of individual files. If you have a previous version of the Presentation Catalog, you will need to convert it to the new format. For more information about how to convert the Presentation Catalog to the new format, see the *Oracle Business Intelligence Infrastructure Upgrade Guide*

4.16.1 Trimming the Input Presentation Catalog

Before you upgrade and merge your current Presentation Catalog with the new Presentation Catalog, determine which of the existing content you want to keep and which new content you want to incorporate. Review your existing Presentation Catalog and determine the usage patterns of reports and dashboards. Note that some of the preconfigured content in the existing catalog may appear in the new version in a redesigned format. In addition, the new version includes completely new content. After you have decided the content that is to make up your enterprise Presentation Catalog, trim the input catalogs using the Catalog Manager. For information on trimming catalogs, see *Oracle Business Intelligence Presentation Services Administration Guide*.

4.16.2 Upgrading the Oracle BI Presentation Catalog to a Newer Version

The Presentation Catalog upgrade process makes use of three catalogs:

- The *original* Presentation Catalog. This is the unmodified Presentation Catalog that you received with the Oracle BI Applications release that you are upgrading from.
- The *new* Presentation Catalog. This is the Presentation Catalog that is installed in the OracleBIData\web\catalog folder with the installation of Oracle BI Applications.
- The *current* Presentation Catalog. This is the Presentation Catalog currently in use at your organization.

You use Catalog Manager for this upgrade process. Catalog Manager compares the content in both the *Current* Presentation Catalog and the *Modified* Presentation Catalog with the content in the Original Presentation Catalog, merges any changes into the *Current* Presentation Catalog, and produces a list of upgrade differences, which you must resolve by indicating how you want the differences handled. If the catalogs have conflicting content, you can choose which catalog the content should be taken from. The end result is a merged Presentation Catalog that contains the site-specific changes, as well as new metadata.

To upgrade your Presentation Catalog to a newer version

1. Make a backup copy of the current Presentation Catalog, rename the folder <catalogname>_old, and move it to a temporary location.

2. Copy the original Presentation Catalog into the folder that holds your current Presentation Catalog and rename it <catalogname>_Original.
3. Start Catalog Manager and open the new Presentation Catalog in offline mode.
4. Select Tools, then select Upgrade Catalog.
5. In the original Presentation Catalog field, browse to locate the original Presentation Catalog.
6. In the Current Presentation Catalog field, browse to locate your current Web Catalog, <catalogname>_old.
7. Click OK.
8. Resolve any upgrade differences as follows:
 - a. Review each unresolved difference in the Unresolved differences list.
 - b. For each unresolved difference, select the version that you want to keep.
 - c. Click OK.

The log file SiebelAnalyticsMigrationLog.txt holds information about the merge process. This log file is written to \OracleBI\web\catalogmanager. If you get an error logged in the file, this means that the path in question had a problem that did not allow the merge mechanism to resolve the merge. No action was taken. To merge that particular item, go into your original Presentation Catalog and merge it manually.

9. Review the upgraded Presentation Catalog, and, if necessary, set permissions for objects.
10. Save the new Presentation Catalog.

4.16.3 Testing the Results of the Presentation Catalog Upgrade

Note: Before you perform this step, you must first migrate the data into the upgraded data warehouse.

The Presentation Catalog upgrade functionality does not automatically carry over object permissions; therefore, you should review the Presentation Catalog object permissions before you perform this step.

This step ensures that the upgraded reports and the new preconfigured reports are functional and render correct results within the new, merged Presentation Catalog. This step is typically performed by visually inspecting the final results of the complete end-to-end upgrade process.

For upgraded reports, the preferred approach for comparison purposes is to have side-by-side environments, and have users review specific dashboard content between the two environments. Examine not only the look and feel of the application but also the data contained in the reports to make sure the content remains the same. It is recommended that you request users to use various elements of the user interface to validate results, such as global prompts, column selectors, report filters, drills, and navigations, as they normally do on a day-to-day basis.

Also review the overall visibility and administrative settings in the new Presentation Catalog to ensure they are correct. Pay careful attention to the visibility rules that are established for any content that was migrated during the upgrade. You might have to manually adjust these settings.

Upgrading Oracle BI Applications 7.9.x for the Siebel Sources

This section contains instructions for upgrading Oracle BI Applications releases when you are running Siebel CRM Applications as your source system.

This section includes the following topics:

- [Section 5.1, "Upgrading Oracle BI Infrastructure"](#)
- [Section 5.2, "Upgrading Oracle BI Applications"](#)
- [Section 5.3, "Upgrading to Informatica PowerCenter Version 8.6 and Hotfix 4"](#)
- [Section 5.4, "Upgrading the Informatica Repository"](#)
- [Section 5.5, "Configuring Informatica PowerCenter Version 8.6 Work with Oracle BI Applications and DAC"](#)
- [Section 5.6, "Upgrading and Configuring DAC"](#)
- [Section 5.7, "Importing New Schema Definitions into the Siebel Transactional Database"](#)
- [Section 5.8, "Upgrading the Data Warehouse Schema and Migrating Data"](#)
- [Section 5.9, "Upgrading the Oracle BI Repository"](#)
- [Section 5.10, "Upgrading the Oracle BI Presentation Catalog"](#)

5.1 Upgrading Oracle BI Infrastructure

Upgrade the Oracle BI Infrastructure to the version that is supported for this release of Oracle BI Applications. See the *System Requirements and Supported Platforms for Oracle Business Intelligence Applications* for the current version that is supported. For information on installing the supported version of Oracle BI Infrastructure, see the *Oracle Business Intelligence Infrastructure Upgrade Guide*.

5.2 Upgrading Oracle BI Applications

Run the Oracle BI Applications installer to upgrade your Oracle BI Applications environment to the current version. For instructions on running the installer, see the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

Note: Perform only the step that addresses the running of the installer. Do not perform the configuration instructions that follow the running of the installer.

Note: If you have a previous release of Oracle BI Applications installed, you must uninstall it before you run the installer for the current release. If you do not uninstall the old release, some folders from the current release will not be correctly installed. (Make a back-up of your DAC folder before you uninstall the old release. This will be the backup of your DAC Client and Server and the DAC metadata files.)

5.3 Upgrading to Informatica PowerCenter Version 8.6 and Hotfix 4

You must install Informatica PowerCenter 8.6 and Hotfix 4 to run the current version of Oracle BI Applications.

The components and architecture for Informatica PowerCenter 8.6 differ significantly from PowerCenter 7.x versions. Oracle recommends that you carefully review the Informatica PowerCenter 8.6 documentation, which is included on the Informatica DVD provided with Oracle BI Applications.

For a summary of installation instructions for installing Informatica PowerCenter 8.6 on a single machine in an Oracle BI Applications deployment, see the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

For detailed information about deploying Informatica PowerCenter 8.6, refer to the *Informatica PowerCenter Installation Guide*, *Informatica PowerCenter Configuration Guide*, *Informatica PowerCenter Administrator Guide*, and related documentation.

For information about applying Hotfix 4, see *PowerCenter Version 8.6.0 Hotfix 4 Release Notes*. This PDF is included with the Informatica documentation on the Informatica PowerCenter DVD and is also included with the Hotfix 4 installation

To upgrade to Informatica PowerCenter 8.6

1. Perform the pre-upgrade steps documented in the *Informatica PowerCenter Installation Guide*.
2. Install Informatica PowerCenter version 8.6 and Hotfix 4, by following the instructions in the *Informatica PowerCenter Installation Guide* and *PowerCenter Version 8.6.0 HotFix 4 Release Notes*.
3. Perform the post-upgrade steps documented in the *Informatica PowerCenter Installation Guide*.

Note: The Informatica PowerCenter 8.6 installation process includes upgrading your current Informatica Repository to the version 8.6 format. This process is necessary so that you will be able to access your current repository using version 8.6 client tools so that you can perform the procedure in [Section 5.4, "Upgrading the Informatica Repository."](#)

In [Section 5.4, "Upgrading the Informatica Repository,"](#) you back up and rename your current repository and then restore the Informatica Repository (Oracle_BI_DW_Base.rep) that is installed during the Oracle BI Applications installation. You then copy your custom folder from the backed up repository into the newly restored Oracle_BI_DW_Base repository.

5.4 Upgrading the Informatica Repository

Follow this procedure to upgrade the Informatica Repository.

For detailed instructions on backing up and restoring the Informatica Repository, see the topic titled, "Backing Up and Restoring the Repository," in "Chapter 8: Managing the Repository," in the *Informatica PowerCenter Administrator Guide*, which is included on the Informatica DVD provided with Oracle BI Applications.

To upgrade the Informatica Repository

1. Make sure you have backed up and renamed your current Informatica Repository.

Note: This repository must be upgraded to the version 8.6 format during the procedure in [Section 5.3, "Upgrading to Informatica PowerCenter Version 8.6 and Hotfix 4."](#) You must upgrade this repository to the version 8.6 format in order to move your custom folder from this repository into the new Oracle_BI_DW_Base repository that you restore in the steps below.

2. Copy the Oracle_BI_DW_Base.rep file from the folder OracleBI\dwrep\Informatica\Repository into the folder \Informatica\PowerCenter 8.6\server\infa_shared\Backup.

Note: The Oracle_BI_DW_Base.rep file is installed in the OracleBI root directory when you run the Oracle BI Applications installer, as described in [Section 5.2, "Upgrading Oracle BI Applications."](#)

3. Restore the Oracle_BI_DW_Base.rep repository.
4. Copy the custom folder from your previous Informatica Repository to the newly created Informatica Repository:
 - a. Launch the Informatica PowerCenter Repository Manager, and connect to both your previous and newly created Informatica repositories.
 - b. Copy the Custom folder in your previous repository into the newly created Informatica Repository.
 - c. Make sure there is an individual workflow for each of the mappings in the Custom folder.

5.5 Configuring Informatica PowerCenter Version 8.6 Work with Oracle BI Applications and DAC

Informatica PowerCenter version 8.6 requires additional configuration steps to work with Oracle BI Applications and DAC. For instructions on performing these steps, see the following sections in the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*:

- Copying Source Files and Lookup Files
- Setting PowerCenter Integration Services Relaxed Code Page Validation
- Setting PowerCenter Integration Services Custom Properties
- Creating the Repository Administrator User in the Native Security Domain

5.6 Upgrading and Configuring DAC

This section includes information you must follow to upgrade and configure DAC.

This section includes the following topics:

- [Section 5.6.1, "Installing the DAC Platform and Oracle BI Applications Metadata Repository Files"](#)
- [Section 5.6.2, "Configuring the DAC Client and Server to Work with Oracle BI Applications and Informatica"](#)
- [Section 5.6.3, "Upgrading the DAC Repository"](#)

5.6.1 Installing the DAC Platform and Oracle BI Applications Metadata Repository Files

The current release of Oracle BI Applications requires DAC version 10.1.3.4.1. This version of DAC is installed by its own installer and not the Oracle BI Applications installer. After you install DAC, you then need to copy metadata files from the machine hosting Oracle BI Applications to the machines hosting the DAC Client and Server. You then need to import the new metadata into the DAC Repository. For instructions on performing these tasks, see the following sections in the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*:

- Installing the DAC Platform

Note: You must perform all of the procedures in this section, including installing JDBC drivers and creating ODBC database connections.

- Installing DAC Metadata Files
- Logging into DAC for the First Time and Importing Metadata into the DAC Repository

After you complete this procedure, you will have the default DAC Repository for Oracle BI Applications version 7.9.6.

5.6.2 Configuring the DAC Client and Server to Work with Oracle BI Applications and Informatica

You need to perform certain configuration tasks to enable the DAC Client and Server to work with Oracle BI Applications and Informatica PowerCenter. For instructions on

performing these tasks, see the following sections in the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*:

- Configuring the DAC Server
- Configuring DAC Integration Settings
- Configuring the SiebelUnicodeDB Custom Property

Note: This procedure is required only if your source to target data movement configuration is Unicode to Unicode.

- Setting Up DAC to Receive Email Notification
- Additional Configuration Tasks

In this section, perform only the tasks that apply to your environment.

5.6.3 Upgrading the DAC Repository

You will use the Refresh Base option of the DAC's Upgrade/Merge Wizard to complete the upgrade of your existing DAC Repository. For information about how to use the Refresh Base option of the Upgrade/Merge Wizard, see the topic titled, "About the Refresh Base Option," in "Chapter 10: Upgrading, Comparing and Merging DAC Repositories," in the *Oracle Business Intelligence Data Warehouse Administration Console User's Guide*.

The *Oracle Business Intelligence Data Warehouse Administration Console User's Guide* is available in the Oracle Business Intelligence Data Warehouse Administration Console Documentation Library on the Oracle Technology Network.

Before you upgrade the DAC Repository, do the following:

- Make a copy of your default source system container. (You cannot change the metadata in the preconfigured container. You must make a copy of it in order to be able to modify the objects it contains.) For instructions, see the *Oracle Business Intelligence Data Warehouse Administration Console User's Guide*.

5.7 Importing New Schema Definitions into the Siebel Transactional Database

This procedure upgrades transactional database objects that relate to Oracle Business Intelligence Applications. It does not upgrade transactional database objects for Siebel CRM applications.

To import new schema definitions into the Siebel transactional database

1. Import schema definitions for non-image tables:
 - a. Access the DDLimp utility.
 - b. Run the DDL_OLTP.ctl script that is located in the OracleBI\dwrep\Upgrade\CTLFiles folder.
2. Import schema definitions for image tables:
 - a. In DAC, go to the Design view, and select your custom container from the drop-down list to the right of the Execute button.
 - b. Click the Tables tab.

- c. Query for all tables for which the image suffix is not null.
- d. Right-click in the list of tables returned by the query, and select Change Capture Scripts, and then Generate Image and Trigger Scripts.
- e. In the Triggers and Image Tables dialog box, do the following:
 - Select the option All Tables in the List.
 - Select the option Generate Image Table Scripts.
 - Select the appropriate database type for the source system.
 - Click OK.
- f. Open the SQL client for the source system database type, for example, SQL Plus for Oracle, Query Analyzer for SQL Server, or a command window for DB2.

The script may contain many lines; therefore, you can save the script file as a SQL file and execute it in a SQL client.
- g. Copy the scripts generated by DAC into the SQL client and execute them.

5.7.1 Verifying the Siebel Transactional Database Upgrade

Follow this procedure to verify the following tables were created in the Siebel transactional database.

To verify the transactional database upgrade

- For all upgrade paths, verify the following tables were created in the Siebel transactional database:
 - S_ETL_R_IMG_XXX
 - S_ETL_I_IMG_XXX
 - S_ETL_D_IMG_XXX
 - S_ETL_PARAM
 - S_ETL_PRD_ATTR
 - S_ETL_PRD_REL

5.8 Upgrading the Data Warehouse Schema and Migrating Data

This section contains the following topics:

- [Section 5.8.1, "Upgrading the Data Warehouse Schema to Version 7.9.4 and Migrating Data"](#)
- [Section 5.8.2, "Upgrading the Data Warehouse Schema to Version 7.9.5 and Migrating Data"](#)
- [Section 5.8.3, "Upgrading the Data Warehouse Schema to Version 7.9.5.1"](#)
- [Section 5.8.4, "Upgrading the Data Warehouse Schema to Version 7.9.6 and Migrating Data"](#)

If you are currently running Oracle BI Applications release 7.9.0, 7.9.1, 7.9.2, or 7.9.3, you need to perform the procedures in [Section 5.8.1](#), [Section 5.8.2](#), and [Section 5.8.3](#) and [Section 5.8.4](#).

If you are currently running Oracle BI Applications release 7.9.4, you need to perform the procedures in [Section 5.8.2](#), [Section 5.8.3](#) and [Section 5.8.4](#).

If you are currently running Oracle BI Applications release 7.9.5, you need to perform the procedures in [Section 5.8.3](#) and [Section 5.8.4](#).

If you are currently running Oracle BI Applications release 7.9.5.1, you need to perform the procedures in [Section 5.8.4](#).

These procedures add new tables, columns, and indexes to the existing data warehouse schema. They also modify the existing data warehouse schema objects.

5.8.1 Upgrading the Data Warehouse Schema to Version 7.9.4 and Migrating Data

You need to perform this procedure if you are upgrading from Oracle BI Applications releases 7.9.0, 7.9.1, 7.9.2, or 7.9.3.

To upgrade the data warehouse schema to version 7.9.4 and migrate data

1. Run the 792_UPGRADE_PRE_CTL_SCRIPT.sql script.
 - a. Open the SQL client for your database type, for example, SQLPLUS for Oracle, Query Analyzer for SQL Server, or a command window for DB2.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 792_UPGRADE_PRE_CTL_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
2. Use the DDLimp utility to run the ddl_794.ctf script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder.

If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddl_794.ctf>
/L <..\OracleBI\dwrep\ddl_794.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\ddl_794.ctf /L C:\OracleBI\dwrep\ddl_794.log
```

3. Run the 792_UPGRADE_PRE_DIMENSION_SCRIPT.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.

- c. Open the 792_UPGRADE_PRE_DIMENSION_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
4. If you upgraded your transactional database to Siebel Applications 8.0 or 8.1.1, run the workflows to migrate your data into the upgraded data warehouse.
 - a. In Informatica Workflow Manager, navigate to the folder UPGRADE_790_TO_791_SBL80UPG.
 - b. Run the following workflows in the order indicated:
 UPGRADE_DIMENSIONS
 UPGRADE_FACTS
5. Verify the data migrated successfully by checking whether any of the Informatica mapping log files stored in the \Informatica PowerCenter 8.6\server\infa_shared\SessLogs directory indicate errors or failures.
6. If the data migration was successful, drop the tables that were created during the upgrade process, such as, W_xxxx_x_79x tables, LKP_xxxx_x and 79x_XXXX_TMP. This step frees the space occupied by these backup tables.
7. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the main Informatica Repository (Oracle_BI_DW_Base.rep).
 - a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.
 The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)

5.8.2 Upgrading the Data Warehouse Schema to Version 7.9.5 and Migrating Data

You need to perform this procedure if you are upgrading from Oracle BI Applications releases 7.9.0, 7.9.1, 7.9.2, 7.9.3, or 7.9.4.

To upgrade the data warehouse schema to version 7.9.5 and migrate data

1. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the Upgrade repository.
 - a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.
 The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)
2. Run the 795_UPGRADE_PRE_CTL_SCRIPT.sql script.

- a. Open the SQL client for your database type, for example, SQLPLUS for Oracle, Query Analyzer for SQL Server, or a command window for DB2.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 795_UPGRADE_PRE_CTL_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
3. Run the UPGRADE_795.ctl script.

This script adds temp tables for the upgrade process.

- a. Navigate to the folder OracleBI\dwrep\Upgrade\CTLFiles, and locate the UPGRADE_795.ctl file.
- b. If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)
- c. Use the DDLimp utility to run the UPGRADE_795.ctl script. Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\UPGRADE_795.CTL>
/L <..\oracleBI\dwrep\UPGRADE.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\UPGRADE_795.CTL /L
C:\OracleBI\dwrep\UPGRADE.log
```

Notes:

- * /P <PASSWORD> - The password for the data warehouse.
- * /C <ODBC connect string> - The name of the ODBC connect string.
- * /I N - Tells DDLimp to ignore the indexes DDL defined in the CTL file if any exist. It does not change existing indexes. (The default is /I Y, which tells DDLimp to create and merge indexes from the CTL file with the indexes in the database.
- * For Oracle databases, use the Data Direct drivers.

In addition, you can use the following commands:

- * /W Y - If the OLAP database is Oracle and Unicode.
- * /Z Y - If the OLAP database is DB2 or SQL Server and Unicode.
- * /B <TABLE_SPACE_NAME> - If you want to create these tables in a separate table space. For DB2, This must be specified as 32K tablespace.

- * /X <INDEX_TABLE_SPACE_NAME> - If you want to create the indexes in a separate tablespace. For DB2, This must be specified as 32K tablespace.
- * /Y - Storage File for DB2/390.
- * /R - Regrant tables.

4. Use the DDLimp utility to run the ddl_795.ctl script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder.

If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>  
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddl_795.ctl>  
/L <..\OracleBI\dwrep\ddl_795.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE  
/I N /R Y /F C:\OracleBI\dwrep\ddl_795.ctl /L C:\OracleBI\dwrep\ddl_795.log
```

5. Run the 795_UPGRADE_PRE_DIMENSION_SCRIPT.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 795_UPGRADE_PRE_DIMENSION_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
6. Copy all of the domain value files in the folder \OracleBI\dwrep\Informatica\LkpFiles into the folder \Informatica PowerCenter 8.6\server\infa_shared\LkpFiles.
7. Migrate data into the upgraded data warehouse.
 - a. Copy the file Upgrade.rep file from the folder OracleBI\dwrep\Upgrade\Informatica\Repository into the folder Informatica PowerCenter 8.6\server\infa_shared\Backup.
 - b. Launch the Informatica PowerCenter Administration Console and restore Upgrade.rep (located in Informatica PowerCenter 8.6\server\infa_shared\Backup).
 - c. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\ParameterFiles and copy the file 795_UPG_PARAMS.txt into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.

- d. Set the parameter \$\$ETL_PROC_WID to the latest ETL_PROC_WID value from the database. You can get this value from W_PARAM_G.ETL_PROC_WID.
- e. Set the parameter \$\$DATASOURCE_NUM_ID to the relevant value from the source system setup.
- f. In Informatica Workflow Manager, open the Relational Connection Browser (in the menu bar, select Connections, and then select Relational), and edit the connect string, user and password for the relational connections as follows:
 Edit the PARAM_OLTP connection to match your OLTP environment.
 Edit the PARAM_OLAP connection to match your OLAP environment.

Note: If you are connected to an Oracle database, use the Oracle Native driver instead of ODBC.

If you are connected to a SQL Server database, use the ODBC driver rather than the native SQL Server driver.

- g. In Informatica Workflow Manager, navigate to the folder UPGRADE_794_to_795_SBL and execute the UPGRADE_DIMENSIONS workflow.
8. Verify the data migrated successfully.
- a. Check whether any of the Informatica mapping log files stored in the \Informatica PowerCenter 8.6\serva\infa_shared\Sesslogs directory indicates errors or failures.
 - b. Check whether the script 795_UPGRADE_PRE_DIMENSION_SCRIPT.sql that you ran in the SQL client of the database failed or errored out while executing.
 - c. Check the log file for the script Upgrade_795.ctl that you ran using the DDLimp command to determine whether any of the runs failed.

If you did not detect any errors or failures in the steps above, then the data migration was successful.

Note: Ignore any error messages that indicate tables or indexes could not be dropped from the database. These error messages are displayed when the specified objects do not exist in the database. This can occur because each source system container holds a subset of all data warehouse tables

9. If the data migration was successful, drop the tables that were created during the upgrade process, such as W_xxxx_x_79x, LKP_xxxx_x and 79x_xxxx_TMP.
 This step frees the space occupied by these backup tables.
10. Once the data migration steps above are complete, you can delete the Upgrade repository to avoid any accidental use or confusion.
- a. In the Informatica PowerCenter Administration Console, select the Upgrade repository service.
 - b. In the General Properties area of the Properties tab, click Edit .
 - c. Make sure the operating mode of the repository service is set to Exclusive.
 - d. Click OK.

- e. Choose Actions, and then click Delete Contents.
 - f. In the Delete contents for <repository name> dialog, enter the repository username and password (for example, Administrator/Administrator), then click OK.
11. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the main Informatica Repository (Oracle_BI_DW_Base.rep).
- a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.
The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)

5.8.3 Upgrading the Data Warehouse Schema to Version 7.9.5.1

You need to perform this procedure if you are upgrading from Oracle BI Applications releases 7.9.0, 7.9.1, 7.9.2, 7.9.3, 7.9.4, or 7.9.5.

To upgrade the data warehouse schema to version 7.9.5.1

- Use the DDLimp utility to run the ddl_7951.ctf script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder. Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddl_7951.ctf>
/L <..\OracleBI\dwrep\ddl_7951.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\ddl_7951.ctf /L C:\OracleBI\dwrep\ddl_7951.log
```

5.8.4 Upgrading the Data Warehouse Schema to Version 7.9.6 and Migrating Data

You need to perform this procedure if you are upgrading from Oracle BI Applications releases 7.9.0, 7.9.1, 7.9.2, 7.9.3, 7.9.4, 7.9.5, or 7.9.5.1.

This procedure adds new tables, columns, and indexes to the existing data warehouse schema. It also modifies the existing data warehouse schema objects.

This section includes the following procedures:

- [Section 5.8.4.1, "Upgrading the Data Warehouse Schema to Version 7.9.6"](#)
- [Section 5.8.4.2, "Migrating Data into the Upgraded Data Warehouse"](#)
- [Section 5.8.4.3, "Verifying the Data Migrated Successfully"](#)
- [Section 5.8.4.4, "Resetting Refresh Dates"](#)

5.8.4.1 Upgrading the Data Warehouse Schema to Version 7.9.6

Follow this procedure to upgrade the data warehouse schema to version 7.9.6.

To upgrade the data warehouse schema to version 7.9.6 and migrate data

1. Run the 796_UPGRADE_DROP_INDEXES.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 796_UPGRADE_DROP_INDEXES. sql file, and copy the contents into the SQL client.
 - d. Execute the script.
2. Run the UPGRADE_796.ctl script.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\CTLFiles, and locate the UPGRADE_796.ctl file.
 - b. If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)
 - c. Use the DDLimp utility to run the UPGRADE_796.ctl script. Use the following command:

```

..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect
string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\Upgrade\CTLFiles\UPGRADE_
796.CTL>
/L <..\oracleBI\dwrep\UPGRADE.log>

```

For example:

```

DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\Upgrade\CTLFiles\UPGRADE_796.CTL /L
C:\OracleBI\dwrep\UPGRADE.log

```

Notes:

- * /P <PASSWORD> - The password for the data warehouse.
- * /C <ODBC connect string> - The name of the ODBC connect string.
- * /I N - Tells DDLimp to ignore the indexes DDL defined in the CTL file if any exist. It does not change existing indexes. (The default is /I Y, which tells DDLimp to create and merge indexes from the CTL file with the indexes in the database.
- * For Oracle databases, use the Data Direct drivers.

In addition, you can use the following commands:

- * /W Y - If the OLAP database is Oracle and Unicode.
- * /Z Y - If the OLAP database is DB2 or SQL Server and Unicode.
- * /B <TABLE_SPACE_NAME> - If you want to create these tables in a separate table space. For DB2, This must be specified as 32K tablespace.

- * /X <INDEX_TABLE_SPACE_NAME> - If you want to create the indexes in a separate tablespace. For DB2, This must be specified as 32K tablespace.
- * /Y - Storage File for DB2/390.
- * /R - Regrant tables.

3. Run the 796_UPGRADE_PRE_CTL_SCRIPT.sql script.
 - a. Open the SQL client for your database type, for example, SQLPLUS for Oracle, Query Analyzer for SQL Server, or a command window for DB2.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 796_UPGRADE_PRE_CTL_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Find and replace the Datasource_Num_id = -1 with the correct Datasource_Num_id from your existing implementation. (The value -1 is a dummy value.)
 - e. Execute the script.
4. Use the DDLimp utility to run the ddl_796.ctl script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder.

If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>  
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\Upgrade\CTLFiles\ddl_796.ctl>  
/L <..\OracleBI\dwrep\ddl_796.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE  
/I N /R Y /F C:\OracleBI\dwrep\Upgrade\CTLFiles\ddl_796.ctl /L  
C:\OracleBI\dwrep\ddl_796.log
```

5. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the main Informatica Repository (Oracle_BI_DW_Base.rep).
 - a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.

The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)

6. Run the 796_UPGRADE_PRE_DIMENSION_SCRIPT.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 796_UPGRADE_PRE_DIMENSION_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
7. Copy all of the domain value files in the folder \OracleBI\dwrep\Informatica\LkpFiles into the folder \Informatica PowerCenter 8.6\server\infa_shared\LkpFiles.

5.8.4.2 Migrating Data into the Upgraded Data Warehouse

Follow this procedure to migrate data into the upgraded data warehouse.

To migrate data into the upgraded data warehouse

1. Copy the file Upgrade.rep file from the folder OracleBI\dwrep\Upgrade\Informatica\Repository into the folder Informatica PowerCenter 8.6\server\infa_shared\Backup.
2. Launch the Informatica PowerCenter Administration Console and restore Upgrade.rep (located in Informatica PowerCenter 8.6\server\infa_shared\Backup).
3. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\ParameterFiles and copy the file 796_UPG_PARAMS.txt into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
4. In the 796_UPG_PARAMS.txt file, set the following parameters:
 - a. \$\$ETL_PROC_WID. Set this parameter to the relevant value from the source system setup. You can get this value from W_PARAM_G.ETL_PROC_WID
 - b. \$\$DATASOURCE_NUM_ID. Set this parameter to the relevant value from the source system setup.
 - c. \$\$INITIAL_EXTRACT_DATE. Set this parameter to the initial extraction data of the data warehouse.
 - d. \$\$WH_DATASOURCE_NUM_ID. Set this parameter to the data source number ID you have set up for the data warehouse.
 - e. \$\$MASTER_ORG. Get this value from the Source System Parameters tab in DAC.
 - f. \$\$INV_PROD_CAT_SET_ID1. Get this value from the Source System Parameters tab in DAC.
 - g. \$\$PROD_CAT_SET_ID1. Get this value from the Source System Parameters tab in DAC.
 - h. Set the parameter \$\$IS_SOURCE_PRE_80 to Y if your source OLTP application was on a version prior to Siebel 8.0 before you began the upgrade process. Otherwise, set this parameter to N.
5. Configure common parameters specific to Siebel source systems. For instructions, see [Section B.2, "Configuring Common Parameters for Siebel Source Systems."](#)

6. In Informatica Workflow Manager, open the Relational Connection Browser (in the menu bar, select Connections, and then select Relational), and edit the connect string, user and password for the relational connections as follows:
 - a. Edit the connection PARAM_OLTP_SIEBEL to match your OLTP environment.
 - b. Edit the connection PARAM_OLAP to match your OLAP environment.
 - c. Edit the connection PARAM_DAC to match your DAC database.

Note: If you are connected to an Oracle database, use the Oracle Native driver instead of ODBC.

If you are connected to a SQL Server database, use the ODBC driver rather than the native SQL Server driver.

7. In Informatica Workflow Manager, navigate to the folder UPGRADE_7951_to_796_SBL and execute the UPGRADE_DIMENSIONS and UPGRADE_FACTS workflows.
8. Run the 796_UPGRADE_POST_SCRIPT.sql.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 796_UPGRADE_POST_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.

5.8.4.3 Verifying the Data Migrated Successfully

Follow this procedure to verify that the data was migrated successfully into the upgraded data warehouse.

To verify the data migrated successfully

1. Check whether any of the Informatica mapping log files stored in the \Informatica PowerCenter 8.6\serva\infa_shared\Sesslogs directory indicates errors or failures.
2. Check whether the following scripts that you ran in the SQL client of the database failed or errored out while executing:
 - 796_UPGRADE_DROP_INDEXES.sql
 - 796_UPGRADE_PRE_CTL_SCRIPT.sql
 - 796_UPGRADE_PRE_DIMENSION.sql
 - 796_UPGRADE_POST_SCRIPT.sql
3. Check the log files for the ddl_796.ctl and Upgrade_796.ctl scripts that you ran using the DDLimp command to determine whether any of the runs failed.

If you did not detect any errors or failures in the steps above, then the data migration was successful.

Note: Ignore any error messages that indicate tables or indexes could not be dropped from the database. These error messages are displayed when the specified objects do not exist in the database. This can occur because each source system container holds a subset of all data warehouse tables

4. If the data migration was successful, drop the tables that were created during the upgrade process, such as W_XXXX_X_79X, LKP_XXXX_X, 79X_XXXX_TMP, and TMP_XXXX_79X.

This step frees the space occupied by these backup tables.

5.8.4.4 Resetting Refresh Dates

After verifying the data was migrated successfully into the upgraded data warehouse, follow this procedure to reset refresh dates.

1. In Informatica Workflow Manager, navigate to the folder UPGRADE_7951_TO_796_SBL and execute the RESET_DAC_REFRESH_DATES workflow.
2. Once the data migration steps above are complete, you can delete the Upgrade repository to avoid any accidental use or confusion.
 - a. In the Informatica PowerCenter Administration Console, select the Upgrade repository service.
 - b. In the General Properties area of the Properties tab, click Edit .
 - c. Make sure the operating mode of the repository service is set to Exclusive.
 - d. Click OK.
 - e. Choose Actions, and then click Delete Contents.
 - f. In the Delete contents for <repository name> dialog, enter the repository username and password (for example, Administrator/Administrator), then click OK.

5.9 Upgrading the Oracle BI Repository

This process merges your customizations of a prior release of the Oracle BI repository with the new version of the Oracle BI repository. Before you begin this process, make sure you have backed up and renamed your existing repository.

To upgrade the repository, perform the following tasks:

- [Section 5.9.1, "Preparing for the Oracle BI Repository Upgrade"](#)
- [Section 5.9.2, "Equalizing the Oracle BI Repositories"](#)
- [Section 5.9.3, "Comparing the Oracle BI Repositories"](#)
- [Section 5.9.4, "Merging the Oracle BI Repositories"](#)
- [Section 5.9.5, "Regression Testing the Oracle BI Repository Merge"](#)

The tasks in this section refer to multiple versions of the Oracle BI repository. [Table 5–1](#) provides the names and descriptions of the repositories used in the examples in this section.

Table 5–1 Names of Analytics Repositories used in Examples

Name of Repository	Description
OracleBIAnalyticsApps_79x.rpd	The standard Oracle BI repository for the version you are upgrading from. Note: Standard repositories from previous releases are available in the folder \OracleBI\Upgrade.
OracleBIAnalyticsApps.rpd	The standard Oracle BI repository for the version you are upgrading to.
Customer_OracleBIAnalyticsApps.rpd	The Oracle BI repository that contains your customizations for the version you are upgrading from.
Merged_Repository_OracleBI.rpd	The Oracle BI repository that contains your customizations for the version you are upgrading to.

5.9.1 Preparing for the Oracle BI Repository Upgrade

Follow this procedure to prepare for the repository upgrade.

To prepare for the Analytics repository upgrade

1. Set up a directory for the merge process, such as \OracleBIUpgrade, and create the following subfolders:
 - Original
 - AfterEqualize
 - AfterMerge
 - AfterManualWork
 - AfterRegressions
2. Copy the original repository (for example, OracleBIAnalyticsApps_79x.rpd), the production repository (for example, Customer_OracleBIAnalyticsApps.rpd, and the repository from the latest installation (for example, OracleBIAnalyticsApps.rpd) into the folder \OracleBIUpgrade\Original.

If, in your current environment, you are running Oracle BI Applications for one or more modules using a Oracle BI repository in which you extracted the corresponding projects for the modules from the standard Oracle BI repository file you received from the previous release, you need to extract the same projects from the OracleBIAnalyticsApps_79x.rpd file and use this as your original repository. (If you have the original repository that you extracted during the last upgrade, you can use it as the original repository file.) This will prevent you from losing any new metadata you would like to add in this upgrade.

Also, if you customized the Oracle BI repository by trimming a large number of objects and you would like to get those objects back during the current upgrade, you need to trim the OracleBIAnalyticsApps_79x.rpd file in the same way and use the modified version as the original repository file. This will prevent you from losing any new metadata you would like to add in this upgrade.

5.9.2 Equalizing the Oracle BI Repositories

The Merge feature in the Administration Tool relies on a change detection algorithm to determine the changes that need to be made to upgrade repositories correctly. For the algorithm to work correctly, it has to determine which objects in the three repositories

(for example, OracleBIAnalyticsApps_79x.rpd, OracleBIAnalyticsApps.rpd, and Customer_OracleBIAnalyticsApps.rpd) are equivalent.

The point of this step is to determine for every object in the OracleBIAnalyticsApps.rpd and the Customer_OracleBIAnalyticsApps.rpd whether it is coming from the OracleBIAnalyticsApps_79x.rpd.

Equivalence between objects is established using the Administration Tool's Equalize feature. The file that you specify in the Output option (-O) is the only file that is modified during the equalization process.

The Equalize feature has several mechanisms for determining whether an object in two different repositories is semantically the same:

- **Fully Qualified Name.** If an object in one repository has the same fully qualified name as another object of the same class in another repository, then the two objects are declared equal.
- **Simple String Substitution.** Equivalence can be declared between two objects of the same class in two repositories whose only difference is that some key characters in their names differ. The equalizerpds executable file ignores those characters while checking fully qualified names. For example, "Core"."W_DAY_D" might be considered equivalent to "Core"."W DAY D" if the characters "_" and " " have been declared as equivalent.
- **Rename File.** When none of the preceding rules are applicable, equivalence can be manually declared using a script as input to the equalizerpds executable file. Oracle ships the rename files (MAP) for the major releases. The files are located in the OracleBI\Upgrade folder. You can also create your own rename files for customizations not covered in the files that Oracle ships. You can open and edit the rename files in Microsoft Excel.

The syntax of the equalizerpds command is as follows:

```
equalizerpds.exe -A userid1 [-B [password1]] -C base_repository_name -D userid2
[-E [password2]] -F repository2_name [-J udml_utf8_file_name_equalization] [-O
output_repository_name] [-X] [-Y equalStringSet]
-X          Treat 'Factxxxx' as 'Fact' in Business Model.
-Y          Treat the characters as equals.
/?          Display this usage information and exit.
```

To equalize a repository

1. Copy the appropriate MAP file from the OracleBI\Upgrade folder into the folder where you will execute equalizerpds.exe, for example, \OracleBIUpgrade\Original.
2. Run equalizerpds.exe to equalize the repository from the latest installation (for example, OracleBIAnalyticsApps.rpd) with the original repository (for example, OracleBIAnalyticsAppss_79x.rpd). An example of the equalizerpds command is as follows:

```
equalizerpds -A Administrator -B SADMIN
-C \\OracleBIUpgrade\Original\OracleBIAnalyticsApps_791.rpd
-D Administrator -E SADMIN
-F \\OracleBIUpgrade\Original\OracleBIAnalyticsApps.rpd
-O \\OracleBIAnalyticsUpgrade\AfterEqualize\OracleBIAnalyticsApps.rpd
-X -J rename7x-79.map
```

The MAP files are located in the \OracleBI\Upgrade folder.

If the equalizerpds.exe executable file runs correctly, no errors are returned.

3. Run `equalizerpds.exe` to equalize your customized repository (for example, `Customer_OracleBIAnalyticsApps.rpd`) with the original repository (for example, `OracleBIAnalyticsApps_79x.rpd`). An example of the `equalizerpds` command is as follows:

```
equalizerpds -A Administrator -B SADMIN  
-C \\OracleBIUpgrade\Original\OracleBIAnalyticsApps_791.rpd  
-D Administrator -E SADMIN  
-F \\OracleBIUpgrade\Original\Customer_OracleBIAnalyticsApps.rpd  
-O \\OracleBIUpgrade\AfterEqualize\Customer_OracleBIAnalyticsApps.rpd
```

The execution of `equalizerpds` that equalizes the customer repository with the original repository does not use the `rename` file.

Make sure that the original repository is copied unchanged into its new location so that after running the script, all three repositories are contained within the `\OracleBIUpgrade\AfterEqualize` directory.

4. To verify the process completed successfully, compare the size of the repositories. The output repository (-O) should be close to the same size as the repository you equalized (-F).

5.9.3 Comparing the Oracle BI Repositories

Follow this procedure to compare your existing repository with the new version to which you are upgrading.

To compare the Analytics repositories

- Use the Administration Tool's Compare Repositories feature to analyze the differences between your existing repository and the new version of the repository to which you are upgrading. Note where elements have been created, removed, or changed in the new version. Consider whether you can use the new metadata and retire customizations you made in the existing repository.

For instructions on how to use the Administration Tool's Compare Repositories feature, see *Oracle Business Intelligence Server Administration Guide*.

5.9.4 Merging the Oracle BI Repositories

In this procedure, you execute the main algorithm to upgrade the repository. For more information on merging the repositories, see *Oracle Business Intelligence Server Administration Guide*.

To merge versions of the Oracle BI repositories

1. Copy the three repositories (for example, `OracleBIAnalyticsApps_79x.rpd`, `OracleBIAnalyticsApps.rpd`, and `Customer_OracleBIAnalyticsApps.rpd`) to the `AfterMerge` folder.
2. Open the repository from the latest installation (for example, `OracleBIAnalyticsApps.rpd`) in the `\OracleBIUpgrade\AfterMerge` folder.
3. Save the repository with a new name, for example, `Merged_Repository_OracleBIAnalyticsApps.rpd`.

This new repository will contain the final results of the upgrade.

4. From the Administration Tool menu bar, select `File`, then select `Merge`.
5. In the `Select Original Repository` dialog box, select the original repository (for example, `OracleBIAnalyticsApps_79x.rpd`).

6. Enter the password, and click OK.
7. Click Select for the Modified Repository field.
8. In the Select Modified Repository dialog box, select the repository that contains the customizations you made to the previous version of the Analytics repository.
9. Click Open, type the password, and then click OK.
10. In the Decision drop-down list, select the action you want to take regarding the repository change, or accept the default action.
11. To locate subsequent rows with empty Decision fields, click the Decision header cell.

When all rows have a value in the Decision field, the Merge button is enabled.

12. Click Merge.

This process can take up to 40 minutes, depending on the size of the repositories you are working with. A message will alert you when the merge is complete.

13. Click Yes when asked if you want to run a consistency check.

The number of errors returned by the consistency check is an indication of how successful the merge process was. If you receive many errors, for example, over 300, you should analyze the reason for the errors. If the merge process failed to recognize that two objects are the same, you may need to edit the rename file if the object is in the Current repository, or add your own rename file if you have renamed many of the objects and the upgrade engine failed to relate them to the original objects.

You also may need to change the actions you selected in the Decision drop-down list before rerunning the merge. This could save you time by reducing the number of errors that you will need to fix manually.

Once you are satisfied with the results of the merge, you should fix the remaining errors manually. It is important that you fix all errors before moving on to the next step. This repository serves as the input for the next stage.

You should also check that all of your customized objects are present and that no duplicate physical tables were introduced. To check for duplicate tables, search for physical tables using a query such as:

```
where name like '*#1'
```

14. Copy the repository to the folder \OracleBIUpgrade\AfterManualWork.

5.9.5 Regression Testing the Oracle BI Repository Merge

In performing a regression test for the repository merge, the objective is to collect a set of logical SQL statements that are used for reports and to verify that they continue to work with the new metadata. For this purpose, it is recommended that you perform the following procedure.

To perform regression testing

1. Run the reports that are necessary to include in the regression suite. These reports might be a subset of the reports in the Presentation Catalog.
2. Collect the logical SQL generated in the previous step. You can do this using Usage Tracking or by parsing the query log file.

For information about Usage Tracking, see *Oracle Business Intelligence Server Administration Guide*.

3. Execute the logical SQL against the old repository using the command line utility nQCmd.exe located in \OracleBI\server\bin, and save the results to a file.

For information about the nQCmd.exe utility, see *Oracle Business Intelligence Server Administration Guide*.

4. Edit the logical SQL test scripts to account for the name changes or modifications resulting from the upgrade.
5. Execute the edited logical SQL against the merged repository, and save the results.
6. Compare the results from the steps above and try to explain the differences. If it is determined that these differences are due to the upgrade process, then you have to correct them manually.

This repository now contains the merged content from the new OracleBIAnalyticsApps.rpd and the production repository.

5.10 Upgrading the Oracle BI Presentation Catalog

You will need to upgrade your current Oracle BI Presentation Catalog if your organization:

- Has prebuilt applications already installed, and
- Has customized the current Oracle BI Presentation Catalog

If you made no changes to the previous Presentation Catalog distributed with previous versions of prebuilt applications, you do not need to upgrade the catalog. You can begin using the newer version of the catalog.

This process includes the following tasks:

- [Section 5.10.1, "Trimming the Input Presentation Catalog"](#)
- [Section 5.10.2, "Upgrading the Oracle BI Presentation Catalog to a Newer Version"](#)
- [Section 5.10.3, "Testing the Results of the Presentation Catalog Upgrade"](#)

Caution: In releases of Oracle BI Applications previous to 7.9, the Presentation Catalog (formerly known as the Siebel Analytics Web Catalog) was stored in a single file rather than in a directory structure of individual files. If you have a previous version of the Presentation Catalog, you will need to convert it to the new format. For more information about how to convert the Presentation Catalog to the new format, see the *Oracle Business Intelligence Infrastructure Upgrade Guide*

5.10.1 Trimming the Input Presentation Catalog

Before you upgrade and merge your current Presentation Catalog with the new Presentation Catalog, determine which of the existing content you want to keep and which new content you want to incorporate. Review your existing Presentation Catalog and determine the usage patterns of reports and dashboards. Note that some of the preconfigured content in the existing catalog may appear in the new version in a redesigned format. In addition, the new version includes completely new content. After you have decided the content that is to make up your enterprise Presentation Catalog, trim the input catalogs using the Catalog Manager. For information on

trimming catalogs, see *Oracle Business Intelligence Presentation Services Administration Guide*.

5.10.2 Upgrading the Oracle BI Presentation Catalog to a Newer Version

The Presentation Catalog upgrade process makes use of three catalogs:

- The *original* Presentation Catalog. This is the unmodified Presentation Catalog that you received with the Oracle BI Applications release that you are upgrading from.
- The *new* Presentation Catalog. This is the Presentation Catalog that is installed in the OracleBIData\web\catalog folder with the installation of Oracle BI Applications.
- The *current* Presentation Catalog. This is the Presentation Catalog currently in use at your organization.

You use Catalog Manager for this upgrade process. Catalog Manager compares the content in both the *Current* Presentation Catalog and the *Modified* Presentation Catalog with the content in the Original Presentation Catalog, merges any changes into the *Current* Presentation Catalog, and produces a list of upgrade differences, which you must resolve by indicating how you want the differences handled. If the catalogs have conflicting content, you can choose which catalog the content should be taken from. The end result is a merged Presentation Catalog that contains the site-specific changes, as well as new metadata.

To upgrade your Presentation Catalog to a newer version

1. Make a backup copy of the current Presentation Catalog, rename the folder <catalogname>_old, and move it to a temporary location.
2. Copy the original Presentation Catalog into the folder that holds your current Presentation Catalog and rename it <catalogname>_Original.
3. Start Catalog Manager and open the new Presentation Catalog in offline mode.
4. Select Tools, then select Upgrade Catalog.
5. In the original Presentation Catalog field, browse to locate the original Presentation Catalog.
6. In the Current Presentation Catalog field, browse to locate your current Web Catalog, <catalogname>_old.
7. Click OK.
8. Resolve any upgrade differences as follows:
 - a. Review each unresolved difference in the Unresolved differences list.
 - b. For each unresolved difference, select the version that you want to keep.
 - c. Click OK.

The log file SiebelAnalyticsMigrationLog.txt holds information about the merge process. This log file is written to \OracleBI\web\catalogmanager. If you get an error logged in the file, this means that the path in question had a problem that did not allow the merge mechanism to resolve the merge. No action was taken. To merge that particular item, go into your original Presentation Catalog and merge it manually.

9. Review the upgraded Presentation Catalog, and, if necessary, set permissions for objects.
10. Save the new Presentation Catalog.

5.10.3 Testing the Results of the Presentation Catalog Upgrade

Note: Before you perform this step, you must first migrate the data into the upgraded data warehouse.

The Presentation Catalog upgrade functionality does not automatically carry over object permissions; therefore, you should review the Presentation Catalog object permissions before you perform this step.

This step ensures that the upgraded reports and the new preconfigured reports are functional and render correct results within the new, merged Presentation Catalog. This step is typically performed by visually inspecting the final results of the complete end-to-end upgrade process.

For upgraded reports, the preferred approach for comparison purposes is to have side-by-side environments, and have users review specific dashboard content between the two environments. Examine not only the look and feel of the application but also the data contained in the reports to make sure the content remains the same. It is recommended that you request users to use various elements of the user interface to validate results, such as global prompts, column selectors, report filters, drills, and navigations, as they normally do on a day-to-day basis.

Also review the overall visibility and administrative settings in the new Presentation Catalog to ensure they are correct. Pay careful attention to the visibility rules that are established for any content that was migrated during the upgrade. You might have to manually adjust these settings.

Part II

Upgrading When Your Source System Is Oracle EBS

Part II contains instructions for upgrading to the current release of Oracle BI Applications from previous releases when you are running Oracle EBS as your source system.

Part II contains [Chapter 6, "Upgrading Oracle BI Applications for Oracle Source Systems."](#)

Note: Some of the information about database platforms and source systems might not apply to this version of Oracle Business Intelligence Applications. For up-to-date information about supported databases and source systems in this version of Oracle Business Intelligence Applications, make sure you read *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*. Make sure that you also read the *Oracle Business Intelligence Applications Release Notes*. The most up-to-date versions of these documents are located on the Oracle Technology Network at http://www.oracle.com/technology/documentation/bi_apps.html. To register for a free account on the Oracle Technology Network, go to <http://www.oracle.com/technology/about/index.html>.

Upgrading Oracle BI Applications for Oracle Source Systems

This section contains instructions for upgrading Oracle BI Applications when you are running Oracle EBS as your source system.

This section includes the following topics:

- [Section 6.1, "Upgrading Oracle BI Infrastructure"](#)
- [Section 6.2, "Upgrading Oracle BI Applications"](#)
- [Section 6.3, "Upgrading to Informatica PowerCenter Version 8.6 and Hotfix 4"](#)
- [Section 6.4, "Upgrading the Informatica Repository"](#)
- [Section 6.5, "Configuring Informatica PowerCenter Version 8.6 to Work with Oracle BI Applications and DAC"](#)
- [Section 6.6, "Upgrading and Configuring DAC"](#)
- [Section 6.7, "Upgrading the Data Warehouse Schema and Migrating Data"](#)
- [Section 6.8, "Upgrading the Oracle BI Repository"](#)
- [Section 6.9, "Upgrading the Oracle BI Presentation Catalog"](#)

6.1 Upgrading Oracle BI Infrastructure

Upgrade the Oracle BI Infrastructure to the version that is supported for this release of Oracle BI Applications. See the *System Requirements and Supported Platforms for Oracle Business Intelligence Applications* for the current version that is supported. For information on installing the supported version of Oracle BI Infrastructure, see the *Oracle Business Intelligence Infrastructure Upgrade Guide*.

6.2 Upgrading Oracle BI Applications

Run the Oracle BI Applications installer to upgrade your Oracle BI Applications environment to the current version. For instructions on running the installer, see the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

Note: Perform only the step that addresses the running of the installer. Do not perform the configuration instructions that follow the running of the installer.

Note: If you have a previous release of Oracle BI Applications installed, you must uninstall it before you run the installer for the current release. If you do not uninstall the old release, some folders from the current release will not be correctly installed. (Make a back-up of your DAC folder before you uninstall the old release. This will be the backup of your DAC Client and Server and the DAC metadata files.)

6.3 Upgrading to Informatica PowerCenter Version 8.6 and Hotfix 4

You must install Informatica PowerCenter 8.6 and Hotfix 4 to run the current version of Oracle BI Applications.

The components and architecture for Informatica PowerCenter 8.6 differ significantly from PowerCenter 7.x versions. Oracle recommends that you carefully review the Informatica PowerCenter 8.6 documentation, which is included on the Informatica DVD provided with Oracle BI Applications.

For a summary of installation instructions for installing Informatica PowerCenter 8.6 on a single machine in an Oracle BI Applications deployment, see the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

For detailed information about deploying Informatica PowerCenter 8.6, refer to the *Informatica PowerCenter Installation Guide*, *Informatica PowerCenter Configuration Guide*, *Informatica PowerCenter Administrator Guide*, and related documentation.

For information about applying Hotfix 4, see *PowerCenter Version 8.6.0 Hotfix 4 Release Notes*. This PDF is included with the Informatica documentation on the Informatica PowerCenter DVD and is also included with the Hotfix 4 installation

To upgrade to Informatica PowerCenter 8.6

1. Perform the pre-upgrade steps documented in the *Informatica PowerCenter Installation Guide*.
2. Install Informatica PowerCenter version 8.6 and Hotfix 4, by following the instructions in the *Informatica PowerCenter Installation Guide* and *PowerCenter Version 8.6.0 HotFix 4 Release Notes*.
3. Perform the post-upgrade steps documented in the *Informatica PowerCenter Installation Guide*.

Note: The Informatica PowerCenter 8.6 installation process includes upgrading your current Informatica Repository to the version 8.6 format. This process is necessary so that you will be able to access your current repository using version 8.6 client tools so that you can perform the procedure [Section 6.4, "Upgrading the Informatica Repository."](#)

In [Section 6.4, "Upgrading the Informatica Repository,"](#) you back up and rename your current repository and then restore the Informatica Repository (Oracle_BI_DW_Base.rep) that is installed during the Oracle BI Applications installation. You then copy your custom folder from the backed up repository into the newly restored Oracle_BI_DW_Base repository.

6.4 Upgrading the Informatica Repository

Follow this procedure to upgrade the Informatica Repository.

For detailed instructions on backing up and restoring the Informatica Repository, see the topic titled, "Backing Up and Restoring the Repository," in "Chapter 8: Managing the Repository," in the *Informatica PowerCenter Administrator Guide*, which is included on the Informatica DVD provided with Oracle BI Applications.

To upgrade the Informatica Repository

1. Make sure you have backed up and renamed your current Informatica Repository.

Note: This repository must be upgraded to the version 8.6 format during the procedure in [Section 6.3, "Upgrading to Informatica PowerCenter Version 8.6 and Hotfix 4."](#) You must upgrade this repository to the version 8.6 format in order to move your custom folder from this repository into the new Oracle_BI_DW_Base repository that you restore in the steps below.

2. Copy the Oracle_BI_DW_Base.rep file from the folder OracleBI\dwrep\Informatica\Repository into the folder \Informatica\PowerCenter 8.6\server\infa_shared\Backup.

Note: The Oracle_BI_DW_Base.rep file is installed in the OracleBI root directory when you run the Oracle BI Applications installer, as described in [Section 6.2, "Upgrading Oracle BI Applications."](#)

3. Restore the Oracle_BI_DW_Base.rep repository.
4. Copy the custom folder from your previous Informatica Repository to the newly created Informatica Repository:
 - a. Launch the Informatica PowerCenter Repository Manager, and connect to both your previous and newly created Informatica repositories.
 - b. Copy the Custom folder in your previous repository into the newly created Informatica Repository.
 - c. Make sure there is an individual workflow for each of the mappings in the Custom folder.

6.5 Configuring Informatica PowerCenter Version 8.6 to Work with Oracle BI Applications and DAC

Informatica PowerCenter version 8.6 requires additional configuration steps to work with Oracle BI Applications and DAC. For instructions on performing these steps, see the following sections in the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*:

- Copying Source Files and Lookup Files
- Setting PowerCenter Integration Services Relaxed Code Page Validation
- Setting PowerCenter Integration Services Custom Properties
- Creating the Repository Administrator User in the Native Security Domain

6.6 Upgrading and Configuring DAC

This section includes information you must follow to upgrade and configure DAC.

This section includes the following topics:

- [Section 6.6.1, "Installing the DAC Platform and Oracle BI Applications Metadata Repository Files"](#)
- [Section 6.6.2, "Configuring the DAC Client and Server to Work with Oracle BI Applications and Informatica"](#)
- [Section 6.6.3, "Upgrading the DAC Repository"](#)

6.6.1 Installing the DAC Platform and Oracle BI Applications Metadata Repository Files

The current release of Oracle BI Applications requires DAC version 10.1.3.4.1. This version of DAC is installed by its own installer and not the Oracle BI Applications installer. After you install DAC, you then need to copy metadata files from the machine hosting Oracle BI Applications to the machines hosting the DAC Client and Server. You then need to import the new metadata into the DAC Repository. For instructions on performing these tasks, see the following sections in the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*:

- Installing the DAC Platform

Note: You must perform all of the procedures in this section, including installing JDBC drivers and creating ODBC database connections.

- Installing DAC Metadata Files
- Logging into DAC for the First Time and Importing Metadata into the DAC Repository

After you complete this procedure, you will have the default DAC Repository for Oracle BI Applications version 7.9.6.

6.6.2 Configuring the DAC Client and Server to Work with Oracle BI Applications and Informatica

You need to perform certain configuration tasks to enable the DAC Client and Server to work with Oracle BI Applications and Informatica PowerCenter. For instructions on performing these tasks, see the following sections in the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*:

- Configuring the DAC Server
- Configuring DAC Integration Settings
- Configuring the SiebelUnicodeDB Custom Property

Note: This procedure is required only if your source to target data movement configuration is Unicode to Unicode.

- Setting Up DAC to Receive Email Notification

- Additional Configuration Tasks

In this section, perform only the tasks that apply to your environment.

6.6.3 Upgrading the DAC Repository

You will use the Refresh Base option of the DAC's Upgrade/Merge Wizard to complete the upgrade of your existing DAC Repository. For information about how to use the Refresh Base option of the Upgrade/Merge Wizard, see the topic titled, "About the Refresh Base Option," in "Chapter 10: Upgrading, Comparing and Merging DAC Repositories," in the *Oracle Business Intelligence Data Warehouse Administration Console User's Guide*.

The *Oracle Business Intelligence Data Warehouse Administration Console User's Guide* is available in the Oracle Business Intelligence Data Warehouse Administration Console Documentation Library on the Oracle Technology Network.

Before you upgrade the DAC Repository, do the following:

- Make a copy of your default source system container. (You cannot change the metadata in the preconfigured container. You must make a copy of it in order to be able to modify the objects it contains.) For instructions, see the *Oracle Business Intelligence Data Warehouse Administration Console User's Guide*.

6.7 Upgrading the Data Warehouse Schema and Migrating Data

This section contains the following topics:

- [Section 6.7.1, "Performing Data Warehouse Schema Upgrade Steps Common to All Previous Releases"](#)
- [Section 6.7.2, "Upgrading the Data Warehouse Schema to Version 7.9.3 and Migrating Data"](#)
- [Section 6.7.3, "Upgrading the Data Warehouse Schema to Version 7.9.4 and Migrating Data"](#)
- [Section 6.7.4, "Upgrading the Data Warehouse Schema to Version 7.9.5. and Migrating Data"](#)
- [Section 6.7.5, "Upgrading the Data Warehouse Schema to Version 7.9.5.1 and Migrating Data"](#)
- [Section 6.7.6, "Upgrading the Data Warehouse Schema to Version 7.9.6 and Migrating Data"](#)

If you are currently running Oracle BI Applications release 7.9.0, 7.9.1 or 7.9.2, you need to perform the procedures in [Section 6.7.1](#), [Section 6.7.2](#), [Section 6.7.3](#), [Section 6.7.4](#), [Section 6.7.5](#), and [Section 6.7.6](#).

If you are currently running Oracle BI Applications release 7.9.3, you need to perform the procedures in [Section 6.7.1](#), [Section 6.7.3](#), [Section 6.7.4](#), [Section 6.7.5](#), and [Section 6.7.6](#).

If you are currently running Oracle BI Applications release 7.9.4, you need to perform the procedures in [Section 6.7.1](#), [Section 6.7.4](#), [Section 6.7.5](#), and [Section 6.7.6](#).

If you are currently running Oracle BI Applications release 7.9.5, you need to perform the procedures in [Section 6.7.1](#) and [Section 6.7.5](#) and [Section 6.7.6](#).

If you are currently running Oracle BI Applications release 7.9.5.1, you need to perform the procedures in [Section 6.7.1](#) and [Section 6.7.6](#).

These procedures add new tables, columns, and indexes to the existing data warehouse schema. They also modify the existing data warehouse schema objects.

6.7.1 Performing Data Warehouse Schema Upgrade Steps Common to All Previous Releases

The steps in this procedure are required for upgrading to Oracle BI Applications release 7.9.6 from all previous releases.

To perform common data warehouse upgrade steps

1. Copy all of the domain value files in the folder
\OracleBI\dwrep\Informatica\LkpFiles into the folder \Informatica
PowerCenter 8.6\server\infa_shared\LkpFiles.
2. Restore the Upgrade repository
 - a. Copy the file Upgrade.rep file from the folder
OracleBI\dwrep\Upgrade\Informatica\Repository into the folder
Informatica PowerCenter 8.6\server\infa_shared\Backup.
 - b. Launch the Informatica PowerCenter Administration Console and restore
Upgrade.rep (located in Informatica PowerCenter 8.6\server\infa_
shared\Backup).
3. In Informatica Workflow Manager, open the Relational Connection Browser (in the
menu bar, select Connections, and then select Relational), and edit the connect
string, user and password for the relational connections as follows:
 - a. For the OLTP connection:

If you are using the Oracle EBS 11i source system, edit the connection
PARAM_OLTP_ORA11i.

If you are using the Oracle EBS R12 source system, edit the connection
PARAM_OLTP_ORAR12.
 - b. For the OLAP connection, edit the connection PARAM_OLAP.
 - c. For the DAC database connection, edit the connection PARAM_DAC.

Note: If you are connected to an Oracle database, use the Oracle
native driver instead of ODBC.

If you are connected to a SQL Server database, use the ODBC driver
rather than the native SQL Server driver.

6.7.2 Upgrading the Data Warehouse Schema to Version 7.9.3 and Migrating Data

You need to perform this procedure if you are upgrading from Oracle BI Applications releases 7.9.0, 7.9.1, or 7.9.2.

To upgrade the data warehouse schema and migrate data into the upgraded data warehouse

1. Run the UPGRADE_793.ctl script.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\CTLFiles, and locate the
UPGRADE_793.ct file.
 - b. If you changed any of the preconfigured column definitions in the CTL file for
your implementation, you need to edit the CTL file to reflect the change. For

example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

- c. Use the DDLimp utility to run the UPGRADE_793.ctl script. Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect
string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\UPGRADE.CTL>
/L <..\oracleBI\dwrep\UPGRADE.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\UPGRADE_793.CTL /L
C:\OracleBI\dwrep\UPGRADE.log
```

Notes:

- * /P <PASSWORD> - The password for the data warehouse.
 - * /C <ODBC connect string> - The name of the ODBC connect string.
 - * /I N - Tells DDLimp to ignore the indexes DDL defined in the CTL file if any exist. It does not change existing indexes. (The default is /I Y, which tells DDLimp to create and merge indexes from the CTL file with the indexes in the database.
 - * For Oracle databases, use the Data Direct drivers.
- In addition, you can use the following commands:
- * /W Y - If the OLAP database is Oracle and Unicode.
 - * /Z Y - If the OLAP database is DB2 or SQL Server and Unicode.
 - * /B <TABLE_SPACE_NAME> - If you want to create these tables in a separate table space. For DB2, This must be specified as 32K tablespace.
 - * /X <INDEX_TABLE_SPACE_NAME> - If you want to create the indexes in a separate tablespace. For DB2, This must be specified as 32K tablespace.
 - * /Y - Storage File for DB2/390.
 - * /R - Regrant tables.

2. Use the DDLimp utility to run the ddl_793.ctl script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder.

If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL

file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

Use the following command:

```
.\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>  
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddl_793.ctl>  
/L <..\OracleBI\dwrep\ddl_793.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE  
/I N /R Y /F C:\OracleBI\dwrep\ddl_793.ctl /L C:\OracleBI\dwrep\ddl_793.log
```

3. Run the 793_UPGRADE_PRE_DIMENSION_SCRIPT.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 793_UPGRADE_PRE_DIMENSION_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
4. Migrate data into the upgraded data warehouse.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\ParameterFiles and copy the file 793_UPG_PARAMS.txt into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
 - b. Set the parameter \$\$ETL_PROC_WID to the latest ETL_PROC_WID value from the database. You can get this value from W_PARAM_G.ETL_PROC_WID.
 - c. Set the parameter \$\$DATASOURCE_NUM_ID to the relevant value from the source system setup.
 - d. In Informatica Workflow Manager, navigate to the folder UPGRADE_790_to_793 and run the Update_Dimensions workflow, and then run the Update_Facts workflow.
 - e. If you are using the Oracle EBS 11.5.10 source system, navigate to the folder UPGRADE_790_to_793_ORA11510, and run the SIL_PurchaseAmount_Patch workflow.
5. Verify the data migrated successfully.
 - a. Check whether any of the Informatica mapping log files stored in the \Informatica PowerCenter 8.6\server\infa_shared\SessLogs directory indicate errors or failures.
 - b. Check whether the script 793_UPGRADE_PRE_DIMENSION_SCRIPT.sql that you ran in the SQL client of the database failed or errored out while executing.
 - c. Check the log file for the script Upgrade_793.ctl that you ran using the DDLimp command to determine whether any of the runs failed.

If you did not detect any errors or failures in the steps above, then the data migration was successful.

Note: Ignore any error messages that indicate tables or indexes could not be dropped from the database. These error messages are displayed when the specified objects do not exist in the database. This can occur because each source system container holds a subset of all data warehouse tables

6. If the data migration was successful, drop the tables that were created during the upgrade process, such as, W_xxxx_x_79x tables, LKP_xxxx_x and 79x_XXXX_TMP.

This step frees the space occupied by these backup tables.

6.7.3 Upgrading the Data Warehouse Schema to Version 7.9.4 and Migrating Data

You need to perform this procedure if you are upgrading from Oracle BI Applications releases 7.9.0, 7.9.1, 7.9.2, or 7.9.3.

To upgrade the data warehouse schema and migrate data into the upgraded data warehouse

1. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the Upgrade repository.
 - a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.
The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)
2. Run the 794_UPGRADE_PRE_CTL_SCRIPT.sql.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 794_UPGRADE_PRE_CTL_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
3. Run the UPGRADE_794.ctl script.
This script adds temp tables for the upgrade process.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\CTLFiles, and locate the UPGRADE_794.ctl file.
 - b. If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the

preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

- c. Use the DDLimp utility to run the UPGRADE_794.ctf script. Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\UPGRADE.CTL>
/L <..\oracleBI\dwrep\UPGRADE.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\UPGRADE_794.CTL /L
C:\OracleBI\dwrep\UPGRADE.log
```

Notes:

- * /P <PASSWORD> - The password for the data warehouse.
- * /C <ODBC connect string> - The name of the ODBC connect string.
- * /I N - Tells DDLimp to ignore the indexes DDL defined in the CTL file if any exist. It does not change existing indexes. (The default is /I Y, which tells DDLimp to create and merge indexes from the CTL file with the indexes in the database.
- * For Oracle databases, use the Data Direct drivers.

In addition, you can use the following commands:

- * /W Y - If the OLAP database is Oracle and Unicode.
- * /Z Y - If the OLAP database is DB2 or SQL Server and Unicode.
- * /B <TABLE_SPACE_NAME> - If you want to create these tables in a separate table space. For DB2, This must be specified as 32K tablespace.
- * /X <INDEX_TABLE_SPACE_NAME> - If you want to create the indexes in a separate tablespace. For DB2, This must be specified as 32K tablespace.
- * /Y - Storage File for DB2/390.
- * /R - Regrant tables.

4. Use the DDLimp utility to run the ddl_794.ctf script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder.

If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddl_794.ctf>
/L <..\oracleBI\dwrep\ddl_794.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\ddl_794.ctl /L C:\OracleBI\dwrep\ddl_794.log
```

5. Run the 794_UPGRADE_PRE_DIMENSION_SCRIPT.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 794_UPGRADE_PRE_DIMENSION_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
6. Migrate data into the upgraded data warehouse.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\ParameterFiles and copy the file 794_UPG_PARAMS.txt into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
 - b. Set the parameter \$\$ETL_PROC_WID to the latest ETL_PROC_WID value from the database. You can get this value from W_PARAM_G.ETL_PROC_WID.
 - c. Set the parameter \$\$DATASOURCE_NUM_ID to the relevant value from the source system setup.
 - d. In Informatica Workflow Manager, navigate to the folder UPGRADE_793_to_794_ORA11i and execute first the UPGRADE_DIMENSIONS workflow and then the UPGRADE_FACTS workflow.
7. Verify the data migrated successfully.
 - a. Check whether any of the Informatica mapping log files stored in the \Informatica PowerCenter 8.6\server\infa_shared\SessLogs directory indicate errors or failures.
 - b. Check whether the script 794_UPGRADE_PRE_DIMENSION_SCRIPT.sql that you ran in the SQL client of the database failed or errored out while executing.
 - c. Check the log file for the script Upgrade_794.ctl that you ran using the DDLimp command to determine whether any of the runs failed.

If you did not detect any errors or failures in the steps above, then the data migration was successful.

Note: Ignore any error messages that indicate tables or indexes could not be dropped from the database. These error messages are displayed when the specified objects do not exist in the database. This can occur because each source system container holds a subset of all data warehouse tables

8. If the data migration was successful, drop the tables that were created during the upgrade process, such as, W_xxxx_x_79x tables, LKP_xxxx_x and 79x_XXXX_TMP. This step frees the space occupied by these backup tables.

9. Use the `reset_infa_seq_gen.bat` script to initialize the Informatica sequence generator for incremental runs on the main Informatica Repository (Oracle_BI_DW_Base.rep).
 - a. Navigate to `OracleBI\dwrep\Upgrade\DbScripts\<database type>`.
 - b. Open the `reset_infa_seq_gen.bat` file.

The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)

6.7.4 Upgrading the Data Warehouse Schema to Version 7.9.5. and Migrating Data

You need to perform this procedure if you are upgrading from Oracle BI Applications releases 7.9.0, 7.9.1, 7.9.2, 7.9.3, or 7.9.4.

To upgrade the data warehouse schema and migrate data into the upgraded data warehouse

1. Use the `reset_infa_seq_gen.bat` script to initialize the Informatica sequence generator for incremental runs on the Upgrade repository.
 - a. Navigate to `OracleBI\dwrep\Upgrade\DbScripts\<database type>`.
 - b. Open the `reset_infa_seq_gen.bat` file.

The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)
2. Run the `795_UPGRADE_PRE_CTL_SCRIPT.sql`.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder `OracleBI\dwrep\Upgrade\DbScripts\<database type>`.
 - c. Open the `795_UPGRADE_PRE_CTL_SCRIPT.sql` file, and copy the contents into the SQL client.
 - d. Execute the script.
3. Run the `UPGRADE_795.ctl` script.

This script adds temp tables for the upgrade process.

 - a. Navigate to the folder `OracleBI\dwrep\Upgrade\CTLFiles`, and locate the `UPGRADE_795.ctl`.
 - b. If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in `W_DAY_D`, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from `VARCHAR(50)` to `VARCHAR(100)`, and the same change was

not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.

- c. Use the DDLimp utility to run the UPGRADE_795.ctl script. Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect
string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\UPGRADE_795.CTL>
/L <..\OracleBI\dwrep\UPGRADE.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\UPGRADE_795.CTL /L
C:\OracleBI\dwrep\UPGRADE.log
```

Notes:

- * /P <PASSWORD> - The password for the data warehouse.
- * /C <ODBC connect string> - The name of the ODBC connect string.
- * /I N - Tells DDLimp to ignore the indexes DDL defined in the CTL file if any exist. It does not change existing indexes. (The default is /I Y, which tells DDLimp to create and merge indexes from the CTL file with the indexes in the database.
- * For Oracle databases, use the Data Direct drivers.

In addition, you can use the following commands:

- * /W Y - If the OLAP database is Oracle and Unicode.
- * /Z Y - If the OLAP database is DB2 or SQL Server and Unicode.
- * /B <TABLE_SPACE_NAME> - If you want to create these tables in a separate table space. For DB2, This must be specified as 32K tablespace.
- * /X <INDEX_TABLE_SPACE_NAME> - If you want to create the indexes in a separate tablespace. For DB2, This must be specified as 32K tablespace.
- * /Y - Storage File for DB2/390.
- * /R - Regrant tables.

4. Use the DDLimp utility to run the ddl_795.ctl script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder.

If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddl_795.ctl>
```

```
/L <..\oracleBI\dwrep\ddl_795.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE  
/I N /R Y /F C:\OracleBI\dwrep\ddl_795.ct1 /L C:\OracleBI\dwrep\ddl_795.log
```

5. Run the 795_UPGRADE_PRE_DIMENSION_SCRIPT.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 795_UPGRADE_PRE_DIMENSION_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
6. Migrate data into the upgraded data warehouse.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\ParameterFiles and copy the file 795_UPG_PARAMS.txt into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
 - b. Set the parameter \$\$ETL_PROC_WID to the latest ETL_PROC_WID value from the database. You can get this value from W_PARAM_G.ETL_PROC_WID.
 - c. Set the parameter \$\$DATASOURCE_NUM_ID to the relevant value from the source system setup.
 - d. If you are using Oracle EBS 11i, in Informatica Workflow Manager, navigate to the folder UPGRADE_794_to_795_ORA11i and execute first the UPGRADE_DIMENSIONS workflow and then the UPGRADE_FACTS workflow.

If you are using Oracle EBS R12, in Informatica Workflow Manager, navigate to the folder UPGRADE_794_to_795_ORAR12 and execute first the UPGRADE_DIMENSIONS workflow and then the UPGRADE_FACTS workflow.
7. Verify the data migrated successfully.
 - a. Check whether any of the Informatica mapping log files stored in the \Informatica PowerCenter 8.6\serva\infa_shared\Sesslogs directory indicates errors or failures.
 - b. Check whether the script 795_UPGRADE_PRE_DIMENSION_SCRIPT.sql that you ran in the SQL client of the database failed or errored out while executing.
 - c. Check the log file for the script Upgrade_795.ct1 that you ran using the DDLimp command to determine whether any of the runs failed.

If you did not detect any errors or failures in the steps above, then the data migration was successful.

Note: Ignore any error messages that indicate tables or indexes could not be dropped from the database. These error messages are displayed when the specified objects do not exist in the database. This can occur because each source system container holds a subset of all data warehouse tables

8. If the data migration was successful, drop the tables that were created during the upgrade process, such as W_XXXX_X_79X, LKP_XXXX_X and 79X_XXXX_TMP.

This step frees the space occupied by these backup tables.

9. Once the data migration steps above are complete, you can delete the Upgrade repository to avoid any accidental use or confusion.
 - a. In the Informatica PowerCenter Administration Console, select the Upgrade repository service.
 - b. In the General Properties area of the Properties tab, click Edit .
 - c. Make sure the operating mode of the repository service is set to Exclusive.
 - d. Click OK.
 - e. Choose Actions, and then click Delete Contents.
 - f. In the Delete contents for <repository name> dialog, enter the repository username and password (for example, Administrator/Administrator), then click OK.
10. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the main Informatica Repository (Oracle_BI_DW_Base.rep).
 - a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.
The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)

6.7.5 Upgrading the Data Warehouse Schema to Version 7.9.5.1 and Migrating Data

You need to perform this procedure if you are upgrading from Oracle BI Applications releases 7.9.0, 7.9.1, 7.9.2, 7.9.3, 7.9.4, or 7.9.5.

To upgrade the data warehouse schema and migrate data into the upgraded data warehouse

1. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the Upgrade repository.
 - a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.
The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)
2. Run the 7951_UPGRADE_PRE_CTL_SCRIPT.sql.
 - a. Open the SQL client for your database type.

- b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 7951_UPGRADE_PRE_CTL_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
 3. Run the UPGRADE_7951.ctf script.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\CTLFiles, and locate the UPGRADE_7951.ctf file.
 - b. If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)
 - c. Use the DDLimp utility to run the UPGRADE_7951.ctf script. Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\UPGRADE_7951.CTL>
/L <..\OracleBI\dwrep\UPGRADE.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\UPGRADE_7951.CTL /L
C:\OracleBI\dwrep\UPGRADE.log
```

Notes:

- * /P <PASSWORD> - The password for the data warehouse.
- * /C <ODBC connect string> - The name of the ODBC connect string.
- * /I N - Tells DDLimp to ignore the indexes DDL defined in the CTL file if any exist. It does not change existing indexes. (The default is /I Y, which tells DDLimp to create and merge indexes from the CTL file with the indexes in the database.
- * For Oracle databases, use the Data Direct drivers.

In addition, you can use the following commands:

- * /W Y - If the OLAP database is Oracle and Unicode.
- * /Z Y - If the OLAP database is DB2 or SQL Server and Unicode.
- * /B <TABLE_SPACE_NAME> - If you want to create these tables in a separate table space. For DB2, This must be specified as 32K tablespace.
- * /X <INDEX_TABLE_SPACE_NAME> - If you want to create the indexes in a separate tablespace. For DB2, This must be specified as 32K tablespace.
- * /Y - Storage File for DB2/390.

* /R - Regrant tables.

4. Use the DDLimp utility to run the ddl_7951.ctf script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder.

If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddl_7951.ctf>
/L <..\OracleBI\dwrep\ddl_7951.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\ddl_7951.ctf /L C:\OracleBI\dwrep\ddl_7951.log
```

5. Run the 7951_UPGRADE_PRE_DIMENSION_SCRIPT.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 7951_UPGRADE_PRE_DIMENSION_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
6. Migrate data into the upgraded data warehouse.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\ParameterFiles and copy the file 7951_UPG_PARAMS.txt into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
 - b. Set the parameter \$\$ETL_PROC_WID to the latest ETL_PROC_WID value from the database. You can get this value from W_PARAM_G.ETL_PROC_WID.
 - c. Set the parameter \$\$DATASOURCE_NUM_ID to the relevant value from the source system setup.
 - d. If you are using Oracle EBS 11i, in Informatica Workflow Manager, navigate to the folder UPGRADE_794_to_7951_ORA11i and execute first the UPGRADE_DIMENSIONS workflow and then the UPGRADE_FACTS workflow.

If you are using Oracle EBS R12, in Informatica Workflow Manager, navigate to the folder UPGRADE_794_to_7951_ORAR12 and execute first the UPGRADE_DIMENSIONS workflow and then the UPGRADE_FACTS workflow.

7. Verify the data migrated successfully.

- a. Check whether any of the Informatica mapping log files stored in the \Informatica PowerCenter 8.6\serva\infa_shared\Sesslogs directory indicates errors or failures.
- b. Check whether the script 7951_UPGRADE_PRE_DIMENSION_SCRIPT.sql that you ran in the SQL client of the database failed or errored out while executing.
- c. Check the log file for the script Upgrade_7951.ctf that you ran using the DDLimp command to determine whether any of the runs failed.

If you did not detect any errors or failures in the steps above, then the data migration was successful.

Note: Ignore any error messages that indicate tables or indexes could not be dropped from the database. These error messages are displayed when the specified objects do not exist in the database. This can occur because each source system container holds a subset of all data warehouse tables

8. If the data migration was successful, drop the tables that were created during the upgrade process, such as W_xxxx_x_79x, LKP_xxxx_x and 79x_xxxx_TMP.
This step frees the space occupied by these backup tables.
9. Once the data migration steps above are complete, you can delete the Upgrade repository to avoid any accidental use or confusion.
 - a. In the Informatica PowerCenter Administration Console, select the Upgrade repository service.
 - b. In the General Properties area of the Properties tab, click Edit .
 - c. Make sure the operating mode of the repository service is set to Exclusive.
 - d. Click OK.
 - e. Choose Actions, and then click Delete Contents.
 - f. In the Delete contents for <repository name> dialog, enter the repository username and password (for example, Administrator/Administrator), then click OK.
10. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the main Informatica Repository (Oracle_BI_DW_Base.rep).
 - a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.
The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)

6.7.6 Upgrading the Data Warehouse Schema to Version 7.9.6 and Migrating Data

You need to perform the procedures in this section if you are upgrading from Oracle BI Applications releases 7.9.0, 7.9.1, 7.9.2, 7.9.3, 7.9.4, 7.9.5 or 7.9.5.1.

This section includes the following procedures:

- [Section 6.7.6.1, "Upgrading the Data Warehouse Schema to Version 7.9.6"](#)
- [Section 6.7.6.2, "Migrating Data into the Upgraded Data Warehouse"](#)
- [Section 6.7.6.3, "Verifying the Data Migrated Successfully"](#)
- [Section 6.7.6.4, "Resetting Refresh Dates"](#)

6.7.6.1 Upgrading the Data Warehouse Schema to Version 7.9.6

Follow this procedure to upgrade the data warehouse schema to version 7.9.6.

To upgrade the data warehouse schema to version 7.9.6

1. Run the 796_UPGRADE_DROP_INDEXES.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 796_UPGRADE_DROP_INDEXES.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
2. Run the UPGRADE_796.ctl script.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\CTLFiles, and locate the UPGRADE_796.ctl file.
 - b. If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)
 - c. Use the DDLimp utility to run the UPGRADE_796.ctl script. Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect
string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\UPGRADE_796.CTL>
/L <..\oracleBI\dwrep\UPGRADE.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\UPGRADE_796.CTL /L
C:\OracleBI\dwrep\UPGRADE.log
```

Notes:

- * /P <PASSWORD> - The password for the data warehouse.

- * /C <ODBC connect string> - The name of the ODBC connect string.
- * /I N - Tells DDLimp to ignore the indexes DDL defined in the CTL file if any exist. It does not change existing indexes. (The default is /I Y, which tells DDLimp to create and merge indexes from the CTL file with the indexes in the database.
- * For Oracle databases, use the Data Direct drivers.
In addition, you can use the following commands:
- * /W Y - If the OLAP database is Oracle and Unicode.
- * /Z Y - If the OLAP database is DB2 or SQL Server and Unicode.
- * /B <TABLE_SPACE_NAME> - If you want to create these tables in a separate table space. For DB2, This must be specified as 32K tablespace.
- * /X <INDEX_TABLE_SPACE_NAME> - If you want to create the indexes in a separate tablespace. For DB2, This must be specified as 32K tablespace.
- * /Y - Storage File for DB2/390.
- * /R - Regrant tables.

3. Run the 796_UPGRADE_PRE_CTL_SCRIPT.sql.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 796_UPGRADE_PRE_CTL_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Find and replace the Datasource_Num_id = -1 with the correct Datasource_Num_id from your existing implementation. (The value -1 is a dummy value.)
 - e. Execute the script.
4. Use the DDLimp utility to run the ddl_796.ctl script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder.

If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddl_796.ctl>
/L <..\oracleBI\dwrep\ddl_796.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\ddl_796.ctl /L C:\OracleBI\dwrep\ddl_796.log
```

5. Use the `reset_infa_seq_gen.bat` script to initialize the Informatica sequence generator for incremental runs on the Upgrade repository.
 - a. Navigate to `OracleBI\dwrep\Upgrade\DbScripts\<database type>`.
 - b. Open the `reset_infa_seq_gen.bat` file.
The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)
6. Run the `796_UPGRADE_PRE_DIMENSION_SCRIPT.sql` script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder `OracleBI\dwrep\Upgrade\DbScripts\<database type>`.
 - c. Open the `796_UPGRADE_PRE_DIMENSION_SCRIPT.sql` file, and copy the contents into the SQL client.
 - d. Execute the script.

6.7.6.2 Migrating Data into the Upgraded Data Warehouse

Follow this procedure to migrate data into the upgraded data warehouse.

To migrate data into the upgraded data warehouse

1. Navigate to the folder `OracleBI\dwrep\Upgrade\Informatica\ParameterFiles` and copy the file `796_UPG_PARAMS.txt` into the `SrcFiles` folder on the Informatica Server machine, for example, `C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles`.
2. In the `796_UPG_PARAMS.txt` file, set the following parameters:
 - a. `$$ETL_PROC_WID`. Set this parameter to the relevant value from the source system setup. You can get this value from `W_PARAM_G.ETL_PROC_WID`
 - b. `$$DATASOURCE_NUM_ID`. Set this parameter to the relevant value from the source system setup.
 - c. `$$INITIAL_EXTRACT_DATE`. Set this parameter to the initial extraction data of the data warehouse.
 - d. `$$WH_DATASOURCE_NUM_ID`. Set this parameter to the data source number ID you have set up for the data warehouse.
 - e. `$$MASTER_ORG`. Get this value from the Source System Parameters tab in DAC.
 - f. `$$INV_PROD_CAT_SET_ID1`. Get this value from the Source System Parameters tab in DAC.
 - g. `$$PROD_CAT_SET_ID1`. Get this value from the Source System Parameters tab in DAC.
3. Configure common parameters specific to Oracle source systems. For instructions, see [Section B.1, "Configuring Common Parameters for Oracle Source Systems."](#)

4. If you are upgrading Oracle Financial Analytics, you need to configure specific parameters and update mappings. For more information, see [Section C.1, "Configuring Parameters and Mappings for Oracle Financial Analytics."](#)
5. If you are upgrading Oracle Supply Chain and Order Management Analytics, you need to configure specific parameters. For more information, see [Section C.2, "Configuring Parameters for Oracle Supply Chain and Order Management Analytics."](#)
6. If you are deploying Oracle Projects Analytics, you need to configure specific parameters. For more information, see [Section C.3, "Configuring Parameters for Oracle Projects Analytics."](#)
7. If you are using Oracle EBS 11i, in Informatica Workflow Manager, navigate to the folder UPGRADE_7951_to_796_ORA11i and execute first the UPGRADE_DIMENSIONS workflow and then the UPGRADE_FACTS workflow.

If you are using Oracle EBS R12, in Informatica Workflow Manager, navigate to the folder UPGRADE_7951_to_796_ORAR12 and execute first the UPGRADE_DIMENSIONS workflow and then the UPGRADE_FACTS workflow.

8. Run the 796_UPGRADE_POST_SCRIPT.sql.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 796_UPGRADE_POST_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.

6.7.6.3 Verifying the Data Migrated Successfully

Follow this procedure to verify that the data was migrated successfully into the upgraded data warehouse.

To verify the data migrated successfully

1. Check whether any of the Informatica mapping log files stored in the \Informatica PowerCenter 8.6\serva\infa_shared\Sesslogs directory indicates errors or failures.
2. Check whether the following scripts that you ran in the SQL client of the database failed or errored out while executing:

796_UPGRADE_DROP_INDEXES.sql

796_UPGRADE_PRE_CTL_SCRIPT.sql

796_UPGRADE_PRE_DIMENSION.sql

796_UPGRADE_POST_SCRIPT.sql

3. Check the log files for the ddl_796.ctl and Upgrade_796.ctl scripts that you ran using the DDLimp command to determine whether any of the runs failed.

If you did not detect any errors or failures in the steps above, then the data migration was successful.

Note: Ignore any error messages that indicate tables or indexes could not be dropped from the database. These error messages are displayed when the specified objects do not exist in the database. This can occur because each source system container holds a subset of all data warehouse tables

4. If the data migration was successful, drop the tables that were created during the upgrade process, such as W_XXXX_X_79X, LKP_XXXX_X and 79X_XXXX_TMP.

This step frees the space occupied by these backup tables.

6.7.6.4 Resetting Refresh Dates

After verifying the data was migrated successfully into the upgraded data warehouse, follow this procedure to reset refresh dates.

1. If you are using Oracle EBS 11i, in Informatica Workflow Manager, navigate to the folder UPGRADE_7951_TO_796_ORA11i and execute the RESET_DAC_REFRESH_DATES workflow.
2. If you are using Oracle EBS R12, in Informatica Workflow Manager, navigate to the folder UPGRADE_7951_TO_796_ORAR12 and execute the RESET_DAC_REFRESH_DATES workflow.
3. In DAC, do the following:
 - a. Navigate to the Setup view, and click the Physical Data Sources tab.
 - b. In the top pane list, select DataWarehouse. (If you customized the name of the connection for the data warehouse, select the appropriate connection for the data warehouse database.)
 - c. Click the Refresh Dates subtab (in the lower pane).
 - d. Query for the following tables, and, for each table, set the value in the Refresh Date column to NULL:
 - W_POSITION_D
 - W_POSITION_DS
 - W_POSITION_DH
4. Once the data migration steps above are complete, you can delete the Upgrade repository to avoid any accidental use or confusion.
 - a. In the Informatica PowerCenter Administration Console, select the Upgrade repository service.
 - b. In the General Properties area of the Properties tab, click Edit .
 - c. Make sure the operating mode of the repository service is set to Exclusive.
 - d. Click OK.
 - e. Choose Actions, and then click Delete Contents.
 - f. In the Delete contents for <repository name> dialog, enter the repository username and password (for example, Administrator/Administrator), then click OK.

6.8 Upgrading the Oracle BI Repository

This process merges your customizations of a prior release of the Oracle BI repository with the new release of the Oracle BI repository. Before you begin this process, make sure you have backed up and renamed your existing repository.

To upgrade the repository, perform the following tasks:

- [Section 6.8.1, "Preparing for the Oracle BI Repository Upgrade"](#)
- [Section 6.8.2, "Equalizing the Oracle BI Repositories"](#)
- [Section 6.8.3, "Comparing the Oracle BI Repositories"](#)
- [Section 6.8.4, "Merging the Oracle BI Repositories"](#)
- [Section 6.8.5, "Regression Testing the Oracle BI Repository Merge"](#)

The tasks in this section refer to multiple releases of the Oracle BI repository. [Table 6–1](#) provides the names and descriptions of the repositories used in the examples in this section.

Table 6–1 Names of Analytics Repositories used in Examples

Name of Repository	Description
OracleBIAnalyticsApps_79x.rpd	The standard Oracle BI repository for the version you are upgrading from. Note: Standard repositories from previous releases are available in the folder \OracleBI\Upgrade.
OracleBIAnalyticsApps.rpd	The standard Oracle BI repository for the version you are upgrading to.
Customer_OracleBIAnalyticsApps.rpd	The Oracle BI repository that contains your customizations for the version you are upgrading from.
Merged_Repository_OracleBI.rpd	The Oracle BI repository that contains your customizations for the version you are upgrading to.

6.8.1 Preparing for the Oracle BI Repository Upgrade

Follow this procedure to prepare for the repository upgrade.

To prepare for the Analytics repository upgrade

1. Set up a directory for the merge process, such as \OracleBIUpgrade, and create the following subfolders:
 - Original
 - AfterEqualize
 - AfterMerge
 - AfterManualWork
 - AfterRegressions
2. Copy the original repository (for example, OracleBIAnalyticsApps_79x.rpd), the production repository (for example, Customer_OracleBIAnalyticsApps.rpd, and the repository from the latest installation (for example, OracleBIAnalyticsApps.rpd) into the folder \OracleBIUpgrade\Original.

If, in your current environment, you are running Oracle BI Applications for one or more modules using a Oracle BI repository in which you extracted the corresponding projects for the modules from the standard Oracle BI repository file

you received from the previous release, you need to extract the same projects from the OracleBIAnalyticsApps_79x.rpd file and use this as your original repository. (If you have the original repository that you extracted during the last upgrade, you can use it as the original repository file.) This will prevent you from losing any new metadata you would like to add in this upgrade.

Also, if you customized the Oracle BI repository by trimming a large number of objects and you would like to get those objects back during the current upgrade, you need to trim the OracleBIAnalyticsApps_79x.rpd file in the same way and use the modified version as the original repository file. This will prevent you from losing any new metadata you would like to add in this upgrade.

6.8.2 Equalizing the Oracle BI Repositories

The Merge feature in the Administration Tool relies on a change detection algorithm to determine the changes that need to be made to upgrade repositories correctly. For the algorithm to work correctly, it has to determine which objects in the three repositories (for example, OracleBIAnalyticsApps_79x.rpd, OracleBIAnalyticsApps.rpd, and Customer_OracleBIAnalyticsApps.rpd) are equivalent.

The point of this step is to determine for every object in the OracleBIAnalyticsApps.rpd and the Customer_OracleBIAnalyticsApps.rpd whether it is coming from the OracleBIAnalyticsApps_79x.rpd.

Equivalence between objects is established using the Administration Tool's Equalize feature. The file that you specify in the Output option (-O) is the only file that is modified during the equalization process.

The Equalize feature has several mechanisms for determining whether an object in two different repositories is semantically the same:

- **Fully Qualified Name.** If an object in one repository has the same fully qualified name as another object of the same class in another repository, then the two objects are declared equal.
- **Simple String Substitution.** Equivalence can be declared between two objects of the same class in two repositories whose only difference is that some key characters in their names differ. The equalizerpds executable file ignores those characters while checking fully qualified names. For example, "Core"."W_DAY_D" might be considered equivalent to "Core"."W DAY D" if the characters "_" and " " have been declared as equivalent.
- **Rename File.** When none of the preceding rules are applicable, equivalence can be manually declared using a script as input to the equalizerpds executable file. Oracle ships the rename files (MAP) for the major releases. The files are located in the OracleBI\Upgrade folder. You can also create your own rename files for customizations not covered in the files that Oracle ships. You can open and edit the rename files in Microsoft Excel.

The syntax of the equalizerpds command is as follows:

```
equalizerpds.exe -A userid1 [-B [password1]] -C base_repository_name -D userid2
[-E [password2]] -F repository2_name [-J udml_utf8_file_name_equalization] [-O
ouput_repository_name] [-X] [-Y equalStringSet]
-X          Treat 'Factxxxx' as 'Fact' in Business Model.
-Y          Treat the characters as equals.
/?          Display this usage information and exit.
```

To equalize a repository

1. Copy the appropriate MAP file from the OracleBI\Upgrade folder into the folder where you will execute equalizerpds.exe, for example, \OracleBIUpgrade\Original.
2. Run equalizerpds.exe to equalize the repository from the latest installation (for example, OracleBIAnalyticsApps.rpd) with the original repository (for example, OracleBIAnalyticsAppss_793.rpd). An example of the equalizerpds command is as follows:

```
equalizerpds -A Administrator -B SADMIN  
-C \\OracleBIUpgrade\Original\OracleBIAnalyticsApps_791.rpd  
-D Administrator -E SADMIN  
-F \\OracleBIUpgrade\Original\OracleBIAnalyticsApps.rpd  
-O \\OracleBIAnalyticsUpgrade\AfterEqualize\OracleBIAnalyticsApps.rpd  
-X -J rename7x-79.map
```

The MAP files are located in the \OracleBI\Upgrade folder.

If the equalizerpds.exe executable file runs correctly, no errors are returned.

3. Run equalizerpds.exe to equalize your customized repository (for example, Customer_OracleBIAnalyticsApps.rpd) with the original repository (for example, OracleBIAnalyticsApps_79x.rpd). An example of the equalizerpds command is as follows:

```
equalizerpds -A Administrator -B SADMIN  
-C \\OracleBIUpgrade\Original\OracleBIAnalyticsApps_791.rpd  
-D Administrator -E SADMIN  
-F \\OracleBIUpgrade\Original\Customer_OracleBIAnalyticsApps.rpd  
-O \\OracleBIUpgrade\AfterEqualize\Customer_OracleBIAnalyticsApps.rpd
```

The execution of equalizerpds that equalizes the customer repository with the original repository does not use the rename file.

Make sure that the original repository is copied unchanged into its new location so that after running the script, all three repositories are contained within the \OracleBIUpgrade\AfterEqualize directory.

4. To verify the process completed successfully, compare the size of the repositories. The output repository (-O) should be close to the same size as the repository you equalized (-F).

6.8.3 Comparing the Oracle BI Repositories

Follow this procedure to compare your existing repository with the new version to which you are upgrading.

To compare the Analytics repositories

- Use the Administration Tool's Compare Repositories feature to analyze the differences between your existing repository and the new version of the repository to which you are upgrading. Note where elements have been created, removed, or changed in the new version. Consider whether you can use the new metadata and retire customizations you made in the existing repository.

For instructions on how to use the Administration Tool's Compare Repositories feature, see *Oracle Business Intelligence Server Administration Guide*.

6.8.4 Merging the Oracle BI Repositories

In this procedure, you execute the main algorithm to upgrade the repository. For more information on merging the repositories, see *Oracle Business Intelligence Server Administration Guide*.

To merge versions of the Oracle BI repositories

1. Copy the three repositories (for example, OracleBIAnalyticsApps_79x.rpd, OracleBIAnalyticsApps.rpd, and Customer_OracleBIAnalyticsApps.rpd) to the AfterMerge folder.
2. Open the repository from the latest installation (for example, OracleBIAnalyticsApps.rpd) in the \OracleBIUpgrade\AfterMerge folder.
3. Save the repository with a new name, for example, Merged_Repository_OracleBIAnalyticsApps.rpd.

This new repository will contain the final results of the upgrade.

4. From the Administration Tool menu bar, select File, then select Merge.
5. In the Select Original Repository dialog box, select the original repository (for example, OracleBIAnalyticsApps_79x.rpd).
6. Enter the password, and click OK.
7. Click Select for the Modified Repository field.
8. In the Select Modified Repository dialog box, select the repository that contains the customizations you made to the previous version of the Analytics repository.
9. Click Open, type the password, and then click OK.
10. In the Decision drop-down list, select the action you want to take regarding the repository change, or accept the default action.
11. To locate subsequent rows with empty Decision fields, click the Decision header cell.

When all rows have a value in the Decision field, the Merge button is enabled.

12. Click Merge.

This process can take up to 40 minutes, depending on the size of the repositories you are working with. A message will alert you when the merge is complete.

13. Click Yes when asked if you want to run a consistency check.

The number of errors returned by the consistency check is an indication of how successful the merge process was. If you receive many errors, for example, over 300, you should analyze the reason for the errors. If the merge process failed to recognize that two objects are the same, you may need to edit the rename file if the object is in the Current repository, or add your own rename file if you have renamed many of the objects and the upgrade engine failed to relate them to the original objects.

You also may need to change the actions you selected in the Decision drop-down list before rerunning the merge. This could save you time by reducing the number of errors that you will need to fix manually.

Once you are satisfied with the results of the merge, you should fix the remaining errors manually. It is important that you fix all errors before moving on to the next step. This repository serves as the input for the next stage.

You should also check that all of your customized objects are present and that no duplicate physical tables were introduced. To check for duplicate tables, search for physical tables using a query such as:

```
where name like '*#1'
```

14. Copy the repository to the folder \OracleBIUpgrade\AfterManualWork.

6.8.5 Regression Testing the Oracle BI Repository Merge

In performing a regression test for the repository merge, the objective is to collect a set of logical SQL statements that are used for reports and to verify that they continue to work with the new metadata. For this purpose, it is recommended that you perform the following procedure.

To perform regression testing

1. Run the reports that are necessary to include in the regression suite. These reports might be a subset of the reports in the Presentation Catalog.
2. Collect the logical SQL generated in the previous step. You can do this using Usage Tracking or by parsing the query log file.

For information about Usage Tracking, see *Oracle Business Intelligence Server Administration Guide*.

3. Execute the logical SQL against the old repository using the command line utility nQCmd.exe located in \OracleBI\server\bin, and save the results to a file.

For information about the nQCmd.exe utility, see *Oracle Business Intelligence Server Administration Guide*.

4. Edit the logical SQL test scripts to account for the name changes or modifications resulting from the upgrade.
5. Execute the edited logical SQL against the merged repository, and save the results.
6. Compare the results from the steps above and try to explain the differences. If it is determined that these differences are due to the upgrade process, then you have to correct them manually.

This repository now contains the merged content from the new OracleBIAnalyticsApps.rpd and the production repository.

6.9 Upgrading the Oracle BI Presentation Catalog

You will need to upgrade your current Oracle BI Presentation Catalog if your organization:

- Has prebuilt applications already installed, and
- Has customized the current Oracle BI Presentation Catalog

If you made no changes to the previous Presentation Catalog distributed with previous versions of prebuilt applications, you do not need to upgrade the catalog. You can begin using the newer version of the catalog.

This process includes the following tasks:

- [Section 6.9.1, "Trimming the Input Presentation Catalog"](#)
- [Section 6.9.2, "Upgrading the Oracle BI Presentation Catalog to a Newer Version"](#)
- [Section 6.9.3, "Testing the Results of the Presentation Catalog Upgrade"](#)

Caution: In releases of Oracle BI Applications previous to 7.9, the Presentation Catalog (formerly known as the Siebel Analytics Web Catalog) was stored in a single file rather than in a directory structure of individual files. If you have a previous version of the Presentation Catalog, you will need to convert it to the new format. For more information about how to convert the Presentation Catalog to the new format, see the *Oracle Business Intelligence Infrastructure Upgrade Guide*.

6.9.1 Trimming the Input Presentation Catalog

Before you upgrade and merge your current Presentation Catalog with the new Presentation Catalog, determine which of the existing content you want to keep and which new content you want to incorporate. Review your existing Presentation Catalog and determine the usage patterns of reports and dashboards. Note that some of the preconfigured content in the existing catalog may appear in the new version in a redesigned format. In addition, the new version includes completely new content. After you have decided the content that is to make up your enterprise Presentation Catalog, trim the input catalogs using the Catalog Manager. For information on trimming catalogs, see *Oracle Business Intelligence Presentation Services Administration Guide*.

6.9.2 Upgrading the Oracle BI Presentation Catalog to a Newer Version

The Presentation Catalog upgrade process makes use of three catalogs:

- The *original* Presentation Catalog. This is the unmodified Presentation Catalog that you received with the Oracle BI Applications release that you are upgrading from.
- The *new* Presentation Catalog. This is the Presentation Catalog that is installed in the OracleBIData\web\catalog folder with the installation of Oracle BI Applications.
- The *current* Presentation Catalog. This is the Presentation Catalog currently in use at your organization.

You use Catalog Manager for this upgrade process. Catalog Manager compares the content in both the *Current* Presentation Catalog and the *Modified* Presentation Catalog with the content in the Original Presentation Catalog, merges any changes into the *Current* Presentation Catalog, and produces a list of upgrade differences, which you must resolve by indicating how you want the differences handled. If the catalogs have conflicting content, you can choose which catalog the content should be taken from. The end result is a merged Presentation Catalog that contains the site-specific changes, as well as new metadata.

To upgrade your Presentation Catalog to a newer version

1. Make a backup copy of the current Presentation Catalog, rename the folder <catalogname>_old, and move it to a temporary location.
2. Copy the original Presentation Catalog into the folder that holds your current Presentation Catalog and rename it <catalogname>_Original.
3. Start Catalog Manager and open the new Presentation Catalog in offline mode.
4. Select Tools, then select Upgrade Catalog.
5. In the original Presentation Catalog field, browse to locate the original Presentation Catalog.

6. In the Current Presentation Catalog field, browse to locate your current Web Catalog, <catalogname>_old.
7. Click OK.
8. Resolve any upgrade differences as follows:
 - a. Review each unresolved difference in the Unresolved differences list.
 - b. For each unresolved difference, select the version that you want to keep.
 - c. Click OK.

The log file SiebelAnalyticsMigrationLog.txt holds information about the merge process. This log file is written to \OracleBI\web\catalogmanager. If you get an error logged in the file, this means that the path in question had a problem that did not allow the merge mechanism to resolve the merge. No action was taken. To merge that particular item, go into your original Presentation Catalog and merge it manually.

9. Review the upgraded Presentation Catalog, and, if necessary, set permissions for objects.
10. Save the new Presentation Catalog.

6.9.3 Testing the Results of the Presentation Catalog Upgrade

Note: Before you perform this step, you must first migrate the data into the upgraded data warehouse.

The Presentation Catalog upgrade functionality does not automatically carry over object permissions; therefore, you should review the Presentation Catalog object permissions before you perform this step.

This step ensures that the upgraded reports and the new preconfigured reports are functional and render correct results within the new, merged Presentation Catalog. This step is typically performed by visually inspecting the final results of the complete end-to-end upgrade process.

For upgraded reports, the preferred approach for comparison purposes is to have side-by-side environments, and have users review specific dashboard content between the two environments. Examine not only the look and feel of the application but also the data contained in the reports to make sure the content remains the same. It is recommended that you request users to use various elements of the user interface to validate results, such as global prompts, column selectors, report filters, drills, and navigations, as they normally do on a day-to-day basis.

Also review the overall visibility and administrative settings in the new Presentation Catalog to ensure they are correct. Pay careful attention to the visibility rules that are established for any content that was migrated during the upgrade. You might have to manually adjust these settings.

Part III

Upgrading When Your Source System is Peoplesoft

Part III contains instructions for upgrading to the current release of Oracle BI Applications when you are running PeopleSoft Enterprise Applications as your source system.

Part III contains [Chapter 7, "Upgrading Oracle BI Applications for the PeopleSoft Source Systems."](#)

Note: Some of the information about database platforms and source systems might not apply to this version of Oracle Business Intelligence Applications. For up-to-date information about supported databases and source systems in this version of Oracle Business Intelligence Applications, make sure you read *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*. Make sure that you also read the *Oracle Business Intelligence Applications Release Notes*. The most up-to-date versions of these documents are located on the Oracle Technology Network at http://www.oracle.com/technology/documentation/bi_apps.html. To register for a free account on the Oracle Technology Network, go to <http://www.oracle.com/technology/about/index.html>.

Upgrading Oracle BI Applications for the PeopleSoft Source Systems

This section contains instructions for upgrading Oracle BI Applications when you are running PeopleSoft Enterprise Applications as your source system.

This section includes the following topics:

- [Section 7.1, "Upgrading Oracle BI Infrastructure"](#)
- [Section 7.2, "Upgrading Oracle BI Applications"](#)
- [Section 7.3, "Upgrading to Informatica PowerCenter Version 8.6 and Hotfix 4"](#)
- [Section 7.4, "Upgrading the Informatica Repository"](#)
- [Section 7.5, "Configuring Informatica PowerCenter Version 8.6 Work with Oracle BI Applications and DAC"](#)
- [Section 7.6, "Upgrading and Configuring DAC"](#)
- [Section 7.7, "Upgrading the Data Warehouse Schema and Migrating Data"](#)
- [Section 7.8, "Upgrading the Oracle BI Repository"](#)
- [Section 7.9, "Upgrading the Oracle BI Presentation Catalog"](#)

7.1 Upgrading Oracle BI Infrastructure

Upgrade the Oracle BI Infrastructure to the version that is supported for this release of Oracle BI Applications. See the *System Requirements and Supported Platforms for Oracle Business Intelligence Applications* for the current version that is supported. For information on installing the supported version of Oracle BI Infrastructure, see the *Oracle Business Intelligence Infrastructure Upgrade Guide*.

7.2 Upgrading Oracle BI Applications

Run the Oracle BI Applications installer to upgrade your Oracle BI Applications environment to the current version. For instructions on running the installer, see *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

Note: Perform only the step that addresses the running of the installer. Do not perform the configuration instructions that follow the running of the installer.

Note: If you have a previous release of Oracle BI Applications installed, you must uninstall it before you run the installer for the current release. If you do not uninstall the old release, some folders from the current release will not be correctly installed. (Make a back-up of your DAC folder before you uninstall the old release. This will be the backup of your DAC Client and Server and the DAC metadata files.)

7.3 Upgrading to Informatica PowerCenter Version 8.6 and Hotfix 4

You must install Informatica PowerCenter 8.6 and Hotfix 4 to run the current version of Oracle BI Applications.

The components and architecture for Informatica PowerCenter 8.6 differ significantly from PowerCenter 7.x versions. Oracle recommends that you carefully review the Informatica PowerCenter 8.6 documentation, which is included on the Informatica DVD provided with Oracle BI Applications.

For a summary of installation instructions for installing Informatica PowerCenter 8.6 on a single machine in an Oracle BI Applications deployment, see the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*.

For detailed information about deploying Informatica PowerCenter 8.6, refer to the *Informatica PowerCenter Installation Guide*, *Informatica PowerCenter Configuration Guide*, *Informatica PowerCenter Administrator Guide*, and related documentation.

For information about applying Hotfix 4, see *PowerCenter Version 8.6.0 Hotfix 4 Release Notes*. This PDF is included with the Informatica documentation on the Informatica PowerCenter DVD and is also included with the Hotfix 4 installation

To upgrade to Informatica PowerCenter 8.6

1. Perform the pre-upgrade steps documented in the *Informatica PowerCenter Installation Guide*.
2. Install Informatica PowerCenter version 8.6 and Hotfix 4, by following the instructions in the *Informatica PowerCenter Installation Guide* and *PowerCenter Version 8.6.0 HotFix 4 Release Notes*.
3. Perform the post-upgrade steps documented in the *Informatica PowerCenter Installation Guide*.

Note: The Informatica PowerCenter 8.6 installation process includes upgrading your current Informatica Repository to the version 8.6 format. This process is necessary so that you will be able to access your current repository using version 8.6 client tools and so that you can perform the procedure in [Section 7.4, "Upgrading the Informatica Repository."](#)

In [Section 7.4, "Upgrading the Informatica Repository,"](#) you back up and rename your current repository and then restore the Informatica Repository (Oracle_BI_DW_Base.rep) that is installed during the Oracle BI Applications installation. You then copy your custom folder from the backed up repository into the newly restored Oracle_BI_DW_Base repository.

7.4 Upgrading the Informatica Repository

Follow this procedure to upgrade the Informatica Repository.

For detailed instructions on backing up and restoring the Informatica Repository, see the topic titled, "Backing Up and Restoring the Repository," in "Chapter 8: Managing the Repository," in the *Informatica PowerCenter Administrator Guide*, which is included on the Informatica DVD provided with Oracle BI Applications.

To upgrade the Informatica Repository

1. Make sure you have backed up and renamed your current Informatica Repository.

Note: This repository must be upgraded to the version 8.6 format during the procedure in [Section 7.3, "Upgrading to Informatica PowerCenter Version 8.6 and Hotfix 4."](#) You must upgrade this repository to the version 8.6 format in order to move your custom folder from this repository into the new Oracle_BI_DW_Base repository that you restore in the steps below.

2. Copy the Oracle_BI_DW_Base.rep file from the folder OracleBI\dwrep\Informatica\Repository into the folder \Informatica\PowerCenter 8.6\server\infa_shared\Backup.

Note: The Oracle_BI_DW_Base.rep file is installed in the OracleBI root directory when you run the Oracle BI Applications installer, as described in [Section 7.2, "Upgrading Oracle BI Applications."](#)

3. Restore the Oracle_BI_DW_Base.rep repository.
4. Copy the custom folder from your previous Informatica Repository to the newly created Informatica Repository:
 - a. Launch the Informatica PowerCenter Repository Manager, and connect to both your previous and newly created Informatica repositories.
 - b. Copy the Custom folder in your previous repository into the newly created Informatica Repository.
 - c. Make sure there is an individual workflow for each of the mappings in the Custom folder.

7.5 Configuring Informatica PowerCenter Version 8.6 Work with Oracle BI Applications and DAC

Informatica PowerCenter version 8.6 requires additional configuration steps to work with Oracle BI Applications and DAC. For instructions on performing these steps, see the following sections in the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*:

- Copying Source Files and Lookup Files
- Setting PowerCenter Integration Services Relaxed Code Page Validation
- Setting PowerCenter Integration Services Custom Properties
- Creating the Repository Administrator User in the Native Security Domain

7.6 Upgrading and Configuring DAC

This section includes information you must follow to upgrade and configure DAC.

This section includes the following topics:

- [Section 7.6.1, "Installing the DAC Platform and Oracle BI Applications Metadata Repository Files"](#)
- [Section 7.6.2, "Configuring the DAC Client and Server to Work with Oracle BI Applications and Informatica"](#)
- [Section 7.6.3, "Upgrading the DAC Repository"](#)

7.6.1 Installing the DAC Platform and Oracle BI Applications Metadata Repository Files

The current release of Oracle BI Applications requires DAC version 10.1.3.4.1. This version of DAC is installed by its own installer and not the Oracle BI Applications installer. After you install DAC, you then need to copy metadata files from the machine hosting Oracle BI Applications to the machines hosting the DAC Client and Server. You then need to import the new metadata into the DAC Repository. For instructions on performing these tasks, see the following sections in the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*:

- Installing the DAC Platform

Note: You must perform all of the procedures in this section, including installing JDBC drivers and creating ODBC database connections.

- Installing DAC Metadata Files
- Logging into DAC for the First Time and Importing Metadata into the DAC Repository

After you complete this procedure, you will have the default DAC Repository for Oracle BI Applications version 7.9.6.

7.6.2 Configuring the DAC Client and Server to Work with Oracle BI Applications and Informatica

You need to perform certain configuration tasks to enable the DAC Client and Server to work with Oracle BI Applications and Informatica PowerCenter. For instructions on performing these tasks, see the following sections in the *Oracle Business Intelligence Applications Installation Guide for Informatica PowerCenter Users*:

- Configuring the DAC Server
- Configuring DAC Integration Settings
- Configuring the SiebelUnicodeDB Custom Property

Note: This procedure is required only if your source to target data movement configuration is Unicode to Unicode.

- Setting Up DAC to Receive Email Notification

- Additional Configuration Tasks

In this section, perform only the tasks that apply to your environment.

7.6.3 Upgrading the DAC Repository

You will use the Refresh Base option of the DAC's Upgrade/Merge Wizard to complete the upgrade of your existing DAC Repository. For information about how to use the Refresh Base option of the Upgrade/Merge Wizard, see the topic titled, "About the Refresh Base Option," in "Chapter 10: Upgrading, Comparing and Merging DAC Repositories," in the *Oracle Business Intelligence Data Warehouse Administration Console User's Guide*.

The *Oracle Business Intelligence Data Warehouse Administration Console User's Guide* is available in the Oracle Business Intelligence Data Warehouse Administration Console Documentation Library on the Oracle Technology Network.

Before you upgrade the DAC Repository, do the following:

- Make a copy of your default source system container. (You cannot change the metadata in the preconfigured container. You must make a copy of it in order to be able to modify the objects it contains.) For instructions, see the *Oracle Business Intelligence Data Warehouse Administration Console User's Guide*.

7.7 Upgrading the Data Warehouse Schema and Migrating Data

This section contains the following topics:

- [Section 7.7.1, "Performing Data Warehouse Schema Upgrade Steps Common to All Previous Releases"](#)
- [Section 7.7.2, "Upgrading the Data Warehouse Schema to Version 7.9.4 and Migrating Data"](#)
- [Section 7.7.3, "Upgrading the Data Warehouse Schema to Version 7.9.5 and Migrating Data"](#)
- [Section 7.7.4, "Upgrading the Data Warehouse Schema to Version 7.9.5.1 and Migrating Data"](#)
- [Section 7.7.5, "Upgrading the Data Warehouse Schema to Version 7.9.6 and Migrating Data"](#)

If you are currently running Oracle BI Applications version 7.9.3, you need to perform the procedures in [Section 7.7.1](#), [Section 7.7.2](#), [Section 7.7.3](#), [Section 7.7.4](#) and [Section 7.7.5](#).

If you are currently running Oracle BI Applications version 7.9.4, you need to perform the procedures in [Section 7.7.1](#), [Section 7.7.3](#), [Section 7.7.4](#) and [Section 7.7.5](#).

If you are currently running Oracle BI Applications version 7.9.5, you need to perform the procedures in [Section 7.7.1](#), [Section 7.7.4](#) and [Section 7.7.5](#).

If you are currently running Oracle BI Applications version 7.9.5.1, you need to perform the procedures in [Section 7.7.1](#) and [Section 7.7.5](#).

These procedures add new tables, columns, and indexes to the existing data warehouse schema. They also modify the existing data warehouse schema objects.

7.7.1 Performing Data Warehouse Schema Upgrade Steps Common to All Previous Releases

The steps in this procedure are required for upgrading to Oracle BI Applications release 7.9.6 from all previous releases.

To perform common data warehouse upgrade steps

1. Copy all of the domain value files in the folder
 \OracleBI\dwrep\Informatica\LkpFiles into the folder \Informatica
 PowerCenter 8.6\server\infa_shared\LkpFiles.
2. Restore the Upgrade repository.
 - a. Copy the file Upgrade.rep file from the folder
 OracleBI\dwrep\Upgrade\Informatica\Repository into the folder
 Informatica PowerCenter 8.6\server\infa_shared\Backup.
 - b. Launch the Informatica PowerCenter Administration Console and restore
 Upgrade.rep (located in Informatica PowerCenter 8.6\server\infa_
 shared\Backup).
3. Configure application connections for the OLAP and OLTP databases.

Note: Make sure all the PeopleSoft source connections are created as application connections and not relational connections.

- a. In Informatica Workflow Manager, open the Application Connection Browser (in the menu bar, select Connections, and then select Application).
- b. Configure the connection PARAM_OLAP_PFST to match your OLAP environment.
- c. Configure the parameter PARAM_OLTP_PFST to match your OLTP environment.
- d. If you are upgrading Oracle Financial Analytics, configure PARAM_OLTP_PFST_FIN to point to the PeopleSoft Financials Pillar database.

Note: This connection is present as both an Application connection and Relational connection. You must configure both of them to point to the PeopleSoft Financial Pillar database.
- e. If you are upgrading Oracle Human Resources Analytics, configure PARAM_OLTP_PSFT_HR to point to the PeopleSoft HRMS Pillar database.
- f. Some of the upgrade sessions apply to both the PeopleSoft HRMS and Financial pillars. Replace the string PARAM_OLTP_PSFT_XXX with either PARAM_OLTP_PSFT_FIN or PARAM_OLTP_PSFT_HR, depending on your implementation.
- g. Configure the PARAM_DAC connection to match your DAC database connection.

Note: If you are connected to an Oracle database, use the Oracle Native driver instead of ODBC.

If you are connected to a SQL Server database, use the ODBC driver rather than the native SQL Server driver.

4. Make sure all the date parameters are set up with an appropriate format.

For example, the format for \$\$LOW_DATE is as follows:

```
SIL_EmployeeDimension_Full]
$$TYPE2_FLG=Y
$DBConnection_OLAP=adevuni_f11
MPLT_GET_ETL_PROC_WID.$$ETL_PROC_WID=1
$$LOW_DATE=to_date('01/01/1970','MM/DD/YYYY')
mplt_SIL_EmployeeDimension.$$LANGUAGE_CODE=E
mplt_SIL_EmployeeDimension.$$MASTER_CODE_NOT_FOUND=NA
mplt_SIL_EmployeeDimension.$$SOURCE_CODE_NOT_SUPPLIED=NA
```

7.7.2 Upgrading the Data Warehouse Schema to Version 7.9.4 and Migrating Data

You need to perform this procedure if you are upgrading from Oracle BI Applications release 7.9.3.

To upgrade the data warehouse schema and migrate data into the upgraded data warehouse

1. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the Upgrade repository.
 - a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.
The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)
2. Run the UPGRADE_794.ctl script.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\CTLFiles, and locate the UPGRADE_794.ctl file.
 - b. If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)
 - c. Use the DDLimp utility to run the UPGRADE_794.ctl script. Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\UPGRADE.CTL>
/L <..\oracleBI\dwrep\UPGRADE.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
```

```
/I N /R Y /F C:\OracleBI\dwrep\UPGRADE_794.CTL /L
C:\OracleBI\dwrep\UPGRADE.log
```

Notes:

- * /P <PASSWORD> - The password for the data warehouse.
- * /C <ODBC connect string> - The name of the ODBC connect string.
- * /I N - Tells DDLimp to ignore the indexes DDL defined in the CTL file if any exist. It does not change existing indexes. (The default is /I Y, which tells DDLimp to create and merge indexes from the CTL file with the indexes in the database.
- * For Oracle databases, use the Data Direct drivers.
In addition, you can use the following commands:
- * /W Y - If the OLAP database is Oracle and Unicode.
- * /Z Y - If the OLAP database is DB2 or SQL Server and Unicode.
- * /B <TABLE_SPACE_NAME> - If you want to create these tables in a separate table space. For DB2, This must be specified as 32K tablespace.
- * /X <INDEX_TABLE_SPACE_NAME> - If you want to create the indexes in a separate tablespace. For DB2, This must be specified as 32K tablespace.
- * /Y - Storage File for DB2/390.
- * /R - Regrant tables.

3. Use the DDLimp utility to run the ddl_794.ctl script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder.

If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddl_794.ctl>
/L <..\OracleBI\dwrep\ddl_794.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\ddl_794.ctl /L C:\OracleBI\dwrep\ddl_794.log
```

4. Run the 794_UPGRADE_PRE_DIMENSION_SCRIPT.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 794_UPGRADE_PRE_DIMENSION_SCRIPT.sql file, and copy the contents into the SQL client.

- d. Execute the script.
5. Migrate data into the upgraded data warehouse.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\ParameterFiles and copy the file 794_UPG_PARAMS.txt into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
 - b. Set the parameter \$\$ETL_PROC_WID to the latest ETL_PROC_WID value from the database. You can get this value from W_PARAM_G.ETL_PROC_WID.
 - c. Set the parameter \$\$DATASOURCE_NUM_ID to the relevant value from the source system setup.
 - d. In Informatica Workflow Manager, navigate to the folder UPGRADE_793_to_794_PSFT88, and execute first the UPGRADE_DIMENSIONS workflow and then the UPGRADE_FACTS workflow.
6. Verify the data migrated successfully.
 - a. Check whether any of the Informatica mapping log files stored in the \Informatica PowerCenter 8.6\server\infa_shared\SessLogs directory indicate errors or failures.
 - b. Check whether the script 794_UPGRADE_PRE_DIMENSION_SCRIPT.sql that you ran in the SQL client of the database failed or errored out while executing.
 - c. Check the log file for the script Upgrade_794.ctl that you ran using the DDLimp command to determine whether any of the runs failed.

If you did not detect any errors or failures in the steps above, then the data migration was successful.

Note: Ignore any error messages that indicate tables or indexes could not be dropped from the database. These error messages are displayed when the specified objects do not exist in the database. This can occur because each source system container holds a subset of all data warehouse tables

7. If the data migration was successful, drop the tables that were created during the upgrade process, such as, W_xxxx_x_79x tables, LKP_xxxx_x and 79x_XXXX_TMP.

This step frees the space occupied by these backup tables.

7.7.3 Upgrading the Data Warehouse Schema to Version 7.9.5 and Migrating Data

You need to perform this procedure if you are upgrading from Oracle BI Applications releases 7.9.3 or 7.9.4.

To upgrade the data warehouse schema and migrate data into the upgraded data warehouse

1. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the Upgrade repository.
 - a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.

The file opens a command prompt, which will prompt you for the parameters listed below.

- c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)
2. Run the 795_UPGRADE_PRE_CTL_SCRIPT.sql.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 795_UPGRADE_PRE_CTL_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
3. Run the UPGRADE_795.ctl script.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\CTLFiles, and locate the UPGRADE_795.ctl file.
 - b. If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)
 - c. Use the DDLimp utility to run the UPGRADE_795.ctl script. Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\UPGRADE_795.CTL>
/L <..\OracleBI\dwrep\UPGRADE.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\UPGRADE_795.CTL /L
C:\OracleBI\dwrep\UPGRADE.log
```

Notes:

- * /P <PASSWORD> - The password for the data warehouse.
- * /C <ODBC connect string> - The name of the ODBC connect string.
- * /I N - Tells DDLimp to ignore the indexes DDL defined in the CTL file if any exist. It does not change existing indexes. (The default is /I Y, which tells DDLimp to create and merge indexes from the CTL file with the indexes in the database.
- * For Oracle databases, use the Data Direct drivers.

In addition, you can use the following commands:

- * /W Y - If the OLAP database is Oracle and Unicode.
- * /Z Y - If the OLAP database is DB2 or SQL Server and Unicode.
- * /B <TABLE_SPACE_NAME> - If you want to create these tables in a separate table space. For DB2, This must be specified as 32K tablespace.
- * /X <INDEX_TABLE_SPACE_NAME> - If you want to create the indexes in a separate tablespace. For DB2, This must be specified as 32K tablespace.
- * /Y - Storage File for DB2/390.
- * /R - Regrant tables.

4. Use the DDLimp utility to run the ddl_795.ctf script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder.

If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddl_795.ctf>
/L <..\OracleBI\dwrep\ddl_795.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\ddl_795.ctf /L C:\OracleBI\dwrep\ddl_795.log
```

5. Run the 795_UPGRADE_PRE_DIMENSION_SCRIPT.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 795_UPGRADE_PRE_DIMENSION_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
6. Migrate data into the upgraded data warehouse.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\ParameterFiles and copy the file 795_UPG_PARAMS.txt into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
 - b. Set the parameter \$\$ETL_PROC_WID to the latest ETL_PROC_WID value from the database. You can get this value from W_PARAM_G.ETL_PROC_WID.
 - c. Set the parameter \$\$DATASOURCE_NUM_ID to the relevant value from the source system setup.

- d. In Informatica Workflow Manager, navigate to the folder UPGRADE_794_to_795_PSFT88 and execute first the UPGRADE_DIMENSIONS workflow and then the UPGRADE_FACTS workflow.
7. Verify the data migrated successfully.
 - a. Check whether any of the Informatica mapping log files stored in the \Informatica PowerCenter 8.6\serva\infa_shared\Sesslogs directory indicates errors or failures.
 - b. Check whether the script 795_UPGRADE_PRE_DIMENSION_SCRIPT.sql that you ran in the SQL client of the database failed or errored out while executing.
 - c. Check the log file for the script Upgrade_795.ctf that you ran using the DDLimp command to determine whether any of the runs failed.

If you did not detect any errors or failures in the steps above, then the data migration was successful.
8. If the data migration was successful, drop the tables that were created during the upgrade process, such as W_xxxx_x_79x, LKP_xxxx_x and 79x_xxxx_TMP.
This step frees the space occupied by these backup tables.
9. Once the data migration steps above are complete, you can delete the Upgrade repository to avoid any accidental use or confusion.
 - a. In the Informatica PowerCenter Administration Console, select the Upgrade repository service.
 - b. In the General Properties area of the Properties tab, click Edit .
 - c. Make sure the operating mode of the repository service is set to Exclusive.
 - d. Click OK.
 - e. Choose Actions, and then click Delete Contents.
 - f. In the Delete contents for <repository name> dialog, enter the repository username and password (for example, Administrator/Administrator), then click OK.
10. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the main Informatica Repository (Oracle_BI_DW_Base.rep).
 - a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.

The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)

7.7.4 Upgrading the Data Warehouse Schema to Version 7.9.5.1 and Migrating Data

You need to perform this procedure if you are upgrading from Oracle BI Applications releases 7.9.3, 7.9.4, or 7.9.5.

To upgrade the data warehouse schema and migrate data

1. Use the `reset_infa_seq_gen.bat` script to initialize the Informatica sequence generator for incremental runs on the Upgrade repository.
 - a. Navigate to `OracleBI\dwrep\Upgrade\DbScripts\<database type>`.
 - b. Open the `reset_infa_seq_gen.bat` file.
The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)
2. Run the `7951_UPGRADE_PRE_CTL_SCRIPT.sql`.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder `OracleBI\dwrep\Upgrade\DbScripts\<database type>`.
 - c. Open the `7951_UPGRADE_PRE_CTL_SCRIPT.sql` file, and copy the contents into the SQL client.
 - d. Execute the script.
3. Run the `UPGRADE_7951.ctf` script.

This script adds temp tables for the upgrade process.

- a. Navigate to the folder `OracleBI\dwrep\Upgrade\CTLFiles`.
- b. Use the DDLimp utility to run the `UPGRADE_7951.ctf` script. Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\UPGRADE_7951.CTL>
/L <..\oracleBI\dwrep\UPGRADE.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\UPGRADE_7951.CTL /L
C:\OracleBI\dwrep\UPGRADE.log
```

Notes:

- * `/P <PASSWORD>` - The password for the data warehouse.
- * `/C <ODBC connect string>` - The name of the ODBC connect string.
- * `/I N` - Tells DDLimp to ignore the indexes DDL defined in the CTL file if any exist. It does not change existing indexes. (The default is `/I Y`, which tells DDLimp to create and merge indexes from the CTL file with the indexes in the database.
- * For Oracle databases, use the Data Direct drivers.
In addition, you can use the following commands:
 - * `/W Y` - If the OLAP database is Oracle and Unicode.
 - * `/Z Y` - If the OLAP database is DB2 or SQL Server and Unicode.

- * /B <TABLE_SPACE_NAME> - If you want to create these tables in a separate table space. For DB2, This must be specified as 32K tablespace.
- * /X <INDEX_TABLE_SPACE_NAME> - If you want to create the indexes in a separate tablespace. For DB2, This must be specified as 32K tablespace.
- * /Y - Storage File for DB2/390.
- * /R - Regrant tables.

4. Use the DDLimp utility to run the ddl_7951.ctl script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder. Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>  
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddl_7951.ctl>  
/L <..\OracleBI\dwrep\ddl_7951.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE  
/I N /R Y /F C:\OracleBI\dwrep\ddl_7951.ctl /L C:\OracleBI\dwrep\ddl_7951.log
```

5. Run the 7951_UPGRADE_PRE_DIMENSION_SCRIPT.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 7951_UPGRADE_PRE_DIMENSION_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
6. Migrate data into the upgraded data warehouse.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\ParameterFiles and copy the file 7951_UPG_PARAMS.txt into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
 - b. Set the parameter \$\$ETL_PROC_WID to the latest ETL_PROC_WID value from the database. You can get this value from W_PARAM_G.ETL_PROC_WID.
 - c. Set the parameter \$\$DATASOURCE_NUM_ID to the relevant value from the source system setup.
 - d. In Informatica Workflow Manager, navigate to the folder UPGRADE_795_to_7951_PSFT88 and execute first the UPGRADE_DIMENSIONS workflow and then the UPGRADE_FACTS workflow.
7. Verify the data migrated successfully.
 - a. Check whether any of the Informatica mapping log files stored in the \Informatica PowerCenter 8.6\serva\infa_shared\Sesslogs directory indicates errors or failures.
 - b. Check whether the script 7951_UPGRADE_PRE_DIMENSION_SCRIPT.sql that you ran in the SQL client of the database failed or errored out while executing.
 - c. Check the log file for the script Upgrade_7951.ctl that you ran using the DDLimp command to determine whether any of the runs failed.

If you did not detect any errors or failures in the steps above, then the data migration was successful.

Note: Ignore any error messages that indicate tables or indexes could not be dropped from the database. These error messages are displayed when the specified objects do not exist in the database. This can occur because each source system container holds a subset of all data warehouse tables

8. If the data migration was successful, drop the tables that were created during the upgrade process, such as W_XXXX_X_79X, LKP_XXXX_X and 79X_XXXX_TMP.

This step frees the space occupied by these backup tables.

9. Once the data migration steps above are complete, you can delete the Upgrade repository to avoid any accidental use or confusion.
 - a. In the Informatica PowerCenter Administration Console, select the Upgrade repository service.
 - b. In the General Properties area of the Properties tab, click Edit .
 - c. Make sure the operating mode of the repository service is set to Exclusive.
 - d. Click OK.
 - e. Choose Actions, and then click Delete Contents.
 - f. In the Delete contents for <repository name> dialog, enter the repository username and password (for example, Administrator/Administrator), then click OK.
10. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the main Informatica Repository (Oracle_BI_DW_Base.rep).
 - a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.
The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)

7.7.5 Upgrading the Data Warehouse Schema to Version 7.9.6 and Migrating Data

You need to perform the procedures in this section if you are upgrading from Oracle BI Applications releases 7.9.3, 7.9.4, 7.9.5, or 7.9.5.1.

This section includes the following topics:

- [Section 7.7.5.1, "Upgrading the Data Warehouse Schema to Version 7.9.6"](#)
- [Section 7.7.5.2, "Migrating Data into the Upgraded Data Warehouse"](#)
- [Section 7.7.5.3, "Verifying the Data Migrated Successfully"](#)
- [Section 7.7.5.4, "Resetting Refresh Dates"](#)

7.7.5.1 Upgrading the Data Warehouse Schema to Version 7.9.6

Follow this procedure to upgrade the data warehouse schema to version 7.9.6.

To upgrade the data warehouse schema to version 7.9.6

1. Run the 796_UPGRADE_DROP_INDEXES.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 796_UPGRADE_DROP_INDEXES.sql file, and copy the contents into the SQL client.
 - d. Execute the script.
2. Run the UPGRADE_796.ctf script.
 - a. Navigate to the folder OracleBI\dwrep\Upgrade\CTLFiles, and locate the UPGRADE_796.ctf file.
 - b. If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)
 - c. Use the DDLimp utility to run the UPGRADE_796.ctf script. Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\UPGRADE_796.CTL>
/L <..\OracleBI\dwrep\UPGRADE.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\UPGRADE_796.CTL /L
C:\OracleBI\dwrep\UPGRADE.log
```

Notes:

- * /P <PASSWORD> - The password for the data warehouse.
- * /C <ODBC connect string> - The name of the ODBC connect string.
- * /I N - Tells DDLimp to ignore the indexes DDL defined in the CTL file if any exist. It does not change existing indexes. (The default is /I Y, which tells DDLimp to create and merge indexes from the CTL file with the indexes in the database.
- * For Oracle databases, use the Data Direct drivers.
In addition, you can use the following commands:
 - * /W Y - If the OLAP database is Oracle and Unicode.
 - * /Z Y - If the OLAP database is DB2 or SQL Server and Unicode.

- * /B <TABLE_SPACE_NAME> - If you want to create these tables in a separate table space. For DB2, This must be specified as 32K tablespace.
 - * /X <INDEX_TABLE_SPACE_NAME> - If you want to create the indexes in a separate tablespace. For DB2, This must be specified as 32K tablespace.
 - * /Y - Storage File for DB2/390.
 - * /R - Regrant tables.
3. Run the 796_UPGRADE_PRE_CTL_SCRIPT.sql.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 796_UPGRADE_PRE_CTL_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Find and replace the Datasource_Num_id = -1 with the correct Datasource_Num_id from your existing implementation. (The value -1 is a dummy value.)
 - e. Execute the script.
 4. Use the DDLimp utility to run the ddl_796.ctf script, which is located in the OracleBI\dwrep\Upgrade\CTLFiles folder.

If you changed any of the preconfigured column definitions in the CTL file for your implementation, you need to edit the CTL file to reflect the change. For example, if you renamed or resized an existing column in W_DAY_D, you need to make the same change in the CTL file. If you do not make the change in the CTL file, when the CTL file is run, the column definition will revert to the preconfigured definition. (As a customization, if a column size was changed from VARCHAR(50) to VARCHAR(100), and the same change was not made in the CTL file before running, the column will revert to the preconfigured value of VARCHAR(50), which could cause data to be truncated in some databases.)

Use the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /I N /R Y /F <..\OracleBI\dwrep\ddl_796.ctf>
/L <..\OracleBI\dwrep\ddl_796.log>
```

For example:

```
DDLIMP /U SADMIN /P SADMIN /C OBIA /G SSE_ROLE
/I N /R Y /F C:\OracleBI\dwrep\ddl_796.ctf /L C:\OracleBI\dwrep\ddl_796.log
```

5. Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the main Informatica Repository (Oracle_BI_DW_Base.rep).
 - a. Navigate to OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - b. Open the reset_infa_seq_gen.bat file.

The file opens a command prompt, which will prompt you for the parameters listed below.
 - c. Enter the appropriate parameter values to reflect your environment (the parameters are database specific). For information about the settings to use, see [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File."](#)

6. Run the 796_UPGRADE_PRE_DIMENSION_SCRIPT.sql script.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.
 - c. Open the 796_UPGRADE_PRE_DIMENSION_SCRIPT.sql file, and copy the contents into the SQL client.
 - d. Execute the script.

7.7.5.2 Migrating Data into the Upgraded Data Warehouse

Follow this procedure to migrate data into the upgraded data warehouse.

To migrate data into the upgraded data warehouse

1. Navigate to the folder OracleBI\dwrep\Upgrade\Informatica\ParameterFiles and copy the file 796_UPG_PARAMS.txt into the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
2. In the 796_UPG_PARAMS.txt file, set the following parameters:
 - a. \$\$ETL_PROC_WID. Set this parameter to the relevant value from the source system setup. You can get this value from W_PARAM_G.ETL_PROC_WID
 - b. \$\$DATASOURCE_NUM_ID. Set this parameter to the relevant value from the source system setup.
 - c. \$\$INITIAL_EXTRACT_DATE. Set this parameter to the initial extraction data of the data warehouse.
 - d. \$\$WH_DATASOURCE_NUM_ID. Set this parameter to the data source number ID you have set up for the data warehouse.
 - e. \$\$MASTER_ORG. Get this value from the Source System Parameters tab in DAC.
 - f. \$\$INV_PROD_CAT_SET_ID1. Get this value from the Source System Parameters tab in DAC.
 - g. \$\$PROD_CAT_SET_ID1. Get this value from the Source System Parameters tab in DAC.
3. If you are upgrading Oracle Financial Analytics, you need to configure specific parameters and update mappings. For more information, see [Section C.1, "Configuring Parameters and Mappings for Oracle Financial Analytics."](#)
4. If you are upgrading Oracle Projects Analytics, you need to configure specific parameters. For more information, see [Section C.3, "Configuring Parameters for Oracle Projects Analytics."](#)
5. In Informatica Workflow Manager, navigate to the folder UPGRADE_7951_to_796_PSFT90 and execute first the UPGRADE_DIMENSIONS workflow and then the UPGRADE_FACTS workflow.
Use this folder for PeopleSoft versions 8.9 and 9.0.
6. Run the 796_UPGRADE_POST_SCRIPT.sql.
 - a. Open the SQL client for your database type.
 - b. Navigate to the folder OracleBI\dwrep\Upgrade\DbScripts\<database type>.

- c. Open the 796_UPGRADE_POST_SCRIPT.sql file, and copy the contents into the SQL client.
- d. Execute the script.

7.7.5.3 Verifying the Data Migrated Successfully

Follow this procedure to verify that the data was migrated successfully into the upgraded data warehouse.

To verify the data migrated successfully

1. Check whether any of the Informatica mapping log files stored in the \Informatica PowerCenter 8.6\serva\infa_shared\Sesslogs directory indicates errors or failures.
2. Check whether the following scripts that you ran in the SQL client of the database failed or errored out while executing:
796_UPGRADE_DROP_INDEXES.sql
796_UPGRADE_PRE_CTL_SCRIPT.sql
796_UPGRADE_PRE_DIMENSION.sql
796_UPGRADE_POST_SCRIPT.sql
3. Check the log files for the ddl_796.ctl and Upgrade_796.ctl scripts that you ran using the DDLimp command to determine whether any of the runs failed.

If you did not detect any errors or failures in the steps above, then the data migration was successful.

Note: Ignore any error messages that indicate tables or indexes could not be dropped from the database. These error messages are displayed when the specified objects do not exist in the database. This can occur because each source system container holds a subset of all data warehouse tables

4. If the data migration was successful, drop the tables that were created during the upgrade process, such as W_XXXX_X_79X, LKP_XXXX_X and 79X_XXXX_TMP.

This step frees the space occupied by these backup tables.

7.7.5.4 Resetting Refresh Dates

After verifying the data was migrated successfully into the upgraded data warehouse, follow this procedure to reset refresh dates.

1. Navigate to the folder UPGRADE_7951_to_796_PSFT90, and execute the RESET_DAC_REFRESH_DATES workflow.
Use this folder for PeopleSoft versions 8.9 and 9.0.
2. In DAC, do the following:
 - a. Navigate to the Setup view, and click the Physical Data Sources tab.
 - b. In the top pane list, select DataWarehouse. (If you customized the name of the connection for the data warehouse, select the appropriate connection for the data warehouse database.)
 - c. Click the Refresh Dates subtab (in the lower pane).

- d. Query for the following tables, and, for each table, set the value in the Refresh Date column to NULL:
 - W_POSITION_D
 - W_POSITION_DS
 - W_POSITION_DH
3. Once the data migration steps above are complete, you can delete the Upgrade repository to avoid any accidental use or confusion.
 - a. In the Informatica PowerCenter Administration Console, select the Upgrade repository service.
 - b. In the General Properties area of the Properties tab, click Edit .
 - c. Make sure the operating mode of the repository service is set to Exclusive.
 - d. Click OK.
 - e. Choose Actions, and then click Delete Contents.
 - f. In the Delete contents for <repository name> dialog, enter the repository username and password (for example, Administrator/Administrator), then click OK.

7.8 Upgrading the Oracle BI Repository

This process merges your customizations of a prior release of the Oracle BI repository with the new version of the Oracle BI repository. Before you begin this process, make sure you have backed up and renamed your existing repository.

To upgrade the repository, perform the following tasks:

- [Section 7.8.1, "Preparing for the Oracle BI Repository Upgrade"](#)
- [Section 7.8.2, "Equalizing the Oracle BI Repositories"](#)
- [Section 7.8.3, "Comparing the Oracle BI Repositories"](#)
- [Section 7.8.4, "Merging the Oracle BI Repositories"](#)
- [Section 7.8.5, "Regression Testing the Oracle BI Repository Merge"](#)

The tasks in this section refer to multiple versions of the Oracle BI repository. [Table 7–1](#) provides the names and descriptions of the repositories used in the examples in this section.

Table 7–1 Names of Analytics Repositories used in Examples

Name of Repository	Description
OracleBIAnalyticsApps_793.rpd	The standard Oracle BI repository for the version you are upgrading from. Note: Standard repositories from previous releases are available in the folder \OracleBI\Upgrade.
OracleBIAnalyticsApps.rpd	The standard Oracle BI repository for the version you are upgrading to.
Customer_OracleBIAnalyticsApps.rpd	The Oracle BI repository that contains your customizations for the version you are upgrading from.
Merged_Repository_OracleBI.rpd	The Oracle BI repository that contains your customizations for the version you are upgrading to.

7.8.1 Preparing for the Oracle BI Repository Upgrade

Follow this procedure to prepare for the repository upgrade.

To prepare for the Analytics repository upgrade

1. Set up a directory for the merge process, such as \OracleBIUpgrade, and create the following subfolders:
 - Original
 - AfterEqualize
 - AfterMerge
 - AfterManualWork
 - AfterRegressions
2. Copy the original repository (for example, OracleBIAnalyticsApps_793.rpd), the production repository (for example, Customer_OracleBIAnalyticsApps.rpd, and the repository from the latest installation (for example, OracleBIAnalyticsApps.rpd) into the folder \OracleBIUpgrade\Original.

If, in your current environment, you are running Oracle BI Applications for one or more modules using a Oracle BI repository in which you extracted the corresponding projects for the modules from the standard Oracle BI repository file you received from the previous release, you need to extract the same projects from the OracleBIAnalyticsApps_793.rpd file and use this as your original repository. (If you have the original repository that you extracted during the last upgrade, you can use it as the original repository file.) This will prevent you from losing any new metadata you would like to add in this upgrade.

Also, if you customized the Oracle BI repository by trimming a large number of objects and you would like to get those objects back during the current upgrade, you need to trim the OracleBIAnalyticsApps_793.rpd file in the same way and use the modified version as the original repository file. This will prevent you from losing any new metadata you would like to add in this upgrade.

7.8.2 Equalizing the Oracle BI Repositories

The Merge feature in the Administration Tool relies on a change detection algorithm to determine the changes that need to be made to upgrade repositories correctly. For the algorithm to work correctly, it has to determine which objects in the three repositories (for example, OracleBIAnalyticsApps_793.rpd, OracleBIAnalyticsApps.rpd, and Customer_OracleBIAnalyticsApps.rpd) are equivalent.

The point of this step is to determine for every object in the OracleBIAnalyticsApps.rpd and the Customer_OracleBIAnalyticsApps.rpd whether it is coming from the OracleBIAnalyticsApps_793.rpd.

Equivalence between objects is established using the Administration Tool's Equalize feature. The file that you specify in the Output option (-O) is the only file that is modified during the equalization process.

The Equalize feature has several mechanisms for determining whether an object in two different repositories is semantically the same:

- **Fully Qualified Name.** If an object in one repository has the same fully qualified name as another object of the same class in another repository, then the two objects are declared equal.

- **Simple String Substitution.** Equivalence can be declared between two objects of the same class in two repositories whose only difference is that some key characters in their names differ. The equalizerpds executable file ignores those characters while checking fully qualified names. For example, "Core"."W_DAY_D" might be considered equivalent to "Core"."W DAY D" if the characters "_" and " " have been declared as equivalent.
- **Rename File.** When none of the preceding rules are applicable, equivalence can be manually declared using a script as input to the equalizerpds executable file. Oracle ships the rename files (MAP) for the major releases. The files are located in the OracleBI\Upgrade folder. You can also create your own rename files for customizations not covered in the files that Oracle ships. You can open and edit the rename files in Microsoft Excel.

The syntax of the equalizerpds command is as follows:

```
equalizerpds.exe -A userid1 [-B [password1]] -C base_repository_name -D userid2
[-E [password2]] -F repository2_name [-J udml_utf8_file_name_equalization] [-O
ouput_repository_name] [-X] [-Y equalStringSet]
-X          Treat 'Factxxxx' as 'Fact' in Business Model.
-Y          Treat the characters as equals.
/?          Display this usage information and exit.
```

To equalize a repository

1. Copy the appropriate MAP file from the OracleBI\Upgrade folder into the folder where you will execute equalizerpds.exe, for example, \OracleBIUpgrade\Original.
2. Run equalizerpds.exe to equalize the repository from the latest installation (for example, OracleBIAnalyticsApps.rpd) with the original repository (for example, OracleBIAnalyticsAppss_793.rpd). An example of the equalizerpds command is as follows:

```
equalizerpds -A Administrator -B SADMIN
-C \\OracleBIUpgrade\Original\OracleBIAnalyticsApps_793.rpd
-D Administrator -E SADMIN
-F \\OracleBIUpgrade\Original\OracleBIAnalyticsApps.rpd
-O \\OracleBIAnalyticsUpgrade\AfterEqualize\OracleBIAnalyticsApps.rpd
-X -J rename7x-79.map
```

The MAP files are located in the \OracleBI\Upgrade folder.

If the equalizerpds.exe executable file runs correctly, no errors are returned.

3. Run equalizerpds.exe to equalize your customized repository (for example, Customer_OracleBIAnalyticsApps.rpd) with the original repository (for example, OracleBIAnalyticsApps_7x.rpd). An example of the equalizerpds command is as follows:

```
equalizerpds -A Administrator -B SADMIN
-C \\OracleBIUpgrade\Original\OracleBIAnalyticsApps_7x.rpd
-D Administrator -E SADMIN
-F \\OracleBIUpgrade\Original\Customer_OracleBIAnalyticsApps.rpd
-O \\OracleBIUpgrade\AfterEqualize\Customer_OracleBIAnalyticsApps.rpd
```

The execution of equalizerpds that equalizes the customer repository with the original repository does not use the rename file.

Make sure that the original repository is copied unchanged into its new location so that after running the script, all three repositories are contained within the \OracleBIUpgrade\AfterEqualize directory.

4. To verify the process completed successfully, compare the size of the repositories. The output repository (-O) should be close to the same size as the repository you equalized (-F).

7.8.3 Comparing the Oracle BI Repositories

Follow this procedure to compare your existing repository with the new version to which you are upgrading.

To compare the Analytics repositories

- Use the Administration Tool's Compare Repositories feature to analyze the differences between your existing repository and the new version of the repository to which you are upgrading. Note where elements have been created, removed, or changed in the new version. Consider whether you can use the new metadata and retire customizations you made in the existing repository.

For instructions on how to use the Administration Tool's Compare Repositories feature, see *Oracle Business Intelligence Server Administration Guide*.

7.8.4 Merging the Oracle BI Repositories

In this procedure, you execute the main algorithm to upgrade the repository. For more information on merging the repositories, see *Oracle Business Intelligence Server Administration Guide*.

To merge versions of the Oracle BI repositories

1. Copy the three repositories (for example, OracleBIAnalyticsApps_793.rpd, OracleBIAnalyticsApps.rpd, and Customer_OracleBIAnalyticsApps.rpd) to the AfterMerge folder.
2. Open the repository from the latest installation (for example, OracleBIAnalyticsApps.rpd) in the \OracleBIUpgrade\AfterMerge folder.
3. Save the repository with a new name, for example, Merged_Repository_OracleBIAnalyticsApps.rpd.

This new repository will contain the final results of the upgrade.

4. From the Administration Tool menu bar, select File, then select Merge.
5. In the Select Original Repository dialog box, select the original repository (for example, OracleBIAnalyticsApps_793.rpd).
6. Enter the password, and click OK.
7. Click Select for the Modified Repository field.
8. In the Select Modified Repository dialog box, select the repository that contains the customizations you made to the previous version of the Analytics repository.
9. Click Open, type the password, and then click OK.
10. In the Decision drop-down list, select the action you want to take regarding the repository change, or accept the default action.
11. To locate subsequent rows with empty Decision fields, click the Decision header cell.

When all rows have a value in the Decision field, the Merge button is enabled.

12. Click Merge.

This process can take up to 40 minutes, depending on the size of the repositories you are working with. A message will alert you when the merge is complete.

13. Click Yes when asked if you want to run a consistency check.

The number of errors returned by the consistency check is an indication of how successful the merge process was. If you receive many errors, for example, over 300, you should analyze the reason for the errors. If the merge process failed to recognize that two objects are the same, you may need to edit the rename file if the object is in the Current repository, or add your own rename file if you have renamed many of the objects and the upgrade engine failed to relate them to the original objects.

You also may need to change the actions you selected in the Decision drop-down list before rerunning the merge. This could save you time by reducing the number of errors that you will need to fix manually.

Once you are satisfied with the results of the merge, you should fix the remaining errors manually. It is important that you fix all errors before moving on to the next step. This repository serves as the input for the next stage.

You should also check that all of your customized objects are present and that no duplicate physical tables were introduced. To check for duplicate tables, search for physical tables using a query such as:

```
where name like '*#1'
```

14. Copy the repository to the folder \OracleBIUpgrade\AfterManualWork.

7.8.5 Regression Testing the Oracle BI Repository Merge

In performing a regression test for the repository merge, the objective is to collect a set of logical SQL statements that are used for reports and to verify that they continue to work with the new metadata. For this purpose, it is recommended that you perform the following procedure.

To perform regression testing

1. Run the reports that are necessary to include in the regression suite. These reports might be a subset of the reports in the Presentation Catalog.
2. Collect the logical SQL generated in the previous step. You can do this using Usage Tracking or by parsing the query log file.

For information about Usage Tracking, see *Oracle Business Intelligence Server Administration Guide*.

3. Execute the logical SQL against the old repository using the command line utility nQCmd.exe located in \OracleBI\server\bin, and save the results to a file.

For information about the nQCmd.exe utility, see *Oracle Business Intelligence Server Administration Guide*.

4. Edit the logical SQL test scripts to account for the name changes or modifications resulting from the upgrade.
5. Execute the edited logical SQL against the merged repository, and save the results.
6. Compare the results from the steps above and try to explain the differences. If it is determined that these differences are due to the upgrade process, then you have to correct them manually.

This repository now contains the merged content from the new OracleBIAnalyticsApps.rpd and the production repository.

7.9 Upgrading the Oracle BI Presentation Catalog

You will need to upgrade your current Oracle BI Presentation Catalog if your organization:

- Has prebuilt applications already installed, and
- Has customized the current Oracle BI Presentation Catalog

If you made no changes to the previous Presentation Catalog distributed with previous versions of prebuilt applications, you do not need to upgrade the catalog. You can begin using the newer version of the catalog.

This process includes the following tasks:

- [Section 7.9.1, "Trimming the Input Presentation Catalog"](#)
- [Section 7.9.2, "Upgrading the Oracle BI Presentation Catalog to a Newer Version"](#)
- [Section 7.9.3, "Testing the Results of the Presentation Catalog Upgrade"](#)

Caution: In releases of Oracle BI Applications previous to 7.9, the Presentation Catalog (formerly known as the Siebel Analytics Web Catalog) was stored in a single file rather than in a directory structure of individual files. If you have a previous version of the Presentation Catalog, you will need to convert it to the new format. For more information about how to convert the Presentation Catalog to the new format, see the *Oracle Business Intelligence Infrastructure Upgrade Guide*.

7.9.1 Trimming the Input Presentation Catalog

Before you upgrade and merge your current Presentation Catalog with the new Presentation Catalog, determine which of the existing content you want to keep and which new content you want to incorporate. Review your existing Presentation Catalog and determine the usage patterns of reports and dashboards. Note that some of the preconfigured content in the existing catalog may appear in the new version in a redesigned format. In addition, the new version includes completely new content. After you have decided the content that is to make up your enterprise Presentation Catalog, trim the input catalogs using the Catalog Manager. For information on trimming catalogs, see *Oracle Business Intelligence Presentation Services Administration Guide*.

7.9.2 Upgrading the Oracle BI Presentation Catalog to a Newer Version

The Presentation Catalog upgrade process makes use of three catalogs:

- The *original* Presentation Catalog. This is the unmodified Presentation Catalog that you received with the Oracle BI Applications release that you are upgrading from.
- The *new* Presentation Catalog. This is the Presentation Catalog that is installed in the OracleBIData\web\catalog folder with the installation of Oracle BI Applications.
- The *current* Presentation Catalog. This is the Presentation Catalog currently in use at your organization.

You use Catalog Manager for this upgrade process. Catalog Manager compares the content in both the *Current* Presentation Catalog and the *Modified* Presentation Catalog with the content in the Original Presentation Catalog, merges any changes into the *Current* Presentation Catalog, and produces a list of upgrade differences, which you must resolve by indicating how you want the differences handled. If the catalogs have conflicting content, you can choose which catalog the content should be taken from. The end result is a merged Presentation Catalog that contains the site-specific changes, as well as new metadata.

To upgrade your Presentation Catalog to a newer version

1. Make a backup copy of the current Presentation Catalog, rename the folder <catalogname>_old, and move it to a temporary location.
2. Copy the original Presentation Catalog into the folder that holds your current Presentation Catalog and rename it <catalogname>_Original.
3. Start Catalog Manager and open the new Presentation Catalog in offline mode.
4. Select Tools, then select Upgrade Catalog.
5. In the original Presentation Catalog field, browse to locate the original Presentation Catalog.
6. In the Current Presentation Catalog field, browse to locate your current Web Catalog, <catalogname>_old.
7. Click OK.
8. Resolve any upgrade differences as follows:
 - a. Review each unresolved difference in the Unresolved differences list.
 - b. For each unresolved difference, select the version that you want to keep.
 - c. Click OK.

The log file SiebelAnalyticsMigrationLog.txt holds information about the merge process. This log file is written to \OracleBI\web\catalogmanager. If you get an error logged in the file, this means that the path in question had a problem that did not allow the merge mechanism to resolve the merge. No action was taken. To merge that particular item, go into your original Presentation Catalog and merge it manually.

9. Review the upgraded Presentation Catalog, and, if necessary, set permissions for objects.
10. Save the new Presentation Catalog.

7.9.3 Testing the Results of the Presentation Catalog Upgrade

Note: Before you perform this step, you must first migrate the data into the upgraded data warehouse.

The Presentation Catalog upgrade functionality does not automatically carry over object permissions; therefore, you should review the Presentation Catalog object permissions before you perform this step.

This step ensures that the upgraded reports and the new preconfigured reports are functional and render correct results within the new, merged Presentation Catalog.

This step is typically performed by visually inspecting the final results of the complete end-to-end upgrade process.

For upgraded reports, the preferred approach for comparison purposes is to have side-by-side environments, and have users review specific dashboard content between the two environments. Examine not only the look and feel of the application but also the data contained in the reports to make sure the content remains the same. It is recommended that you request users to use various elements of the user interface to validate results, such as global prompts, column selectors, report filters, drills, and navigations, as they normally do on a day-to-day basis.

Also review the overall visibility and administrative settings in the new Presentation Catalog to ensure they are correct. Pay careful attention to the visibility rules that are established for any content that was migrated during the upgrade. You might have to manually adjust these settings.

Part IV

Appendices

Part IV contains the following appendixes:

- [Appendix A, "Upgrading the Oracle BI Repository for Industry-Specific Analytics Applications"](#)
- [Appendix B, "Configuring Common Parameters for Upgrading to Oracle BI Applications 7.9.6"](#)
- [Appendix C, "Configuring Application-Specific Parameters and Mappings"](#)
- [Appendix D, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File"](#)

Upgrading the Oracle BI Repository for Industry-Specific Analytics Applications

This appendix contains instructions for upgrading the Oracle BI repository for Pharma Analytics, Consumer Sector, and Vehicle Sales.

In Oracle BI Applications release 7.9.4, the Pharma Analytics business model and Core business model were merged. In Oracle BI Applications release 7.9.5, the Consumer Sector and Vehicle Sales Analytics business models and Core business model were merged. Because of this merge, the process of upgrading the Oracle BI repository has some steps that differ from the standard Oracle BI repository upgrade process.

This section includes the following topics:

- [Section A.1, "Common Dimensions"](#)
- [Section A.2, "Merging Siebel Analytics and Oracle BI Repositories"](#)
- [Section A.3, "Replacing Common Dimensions After the Repository Merge"](#)

A.1 Common Dimensions

[Table A-1](#), [Table A-2](#), and [Table A-3](#) list the common dimensions of Pharma, Consumer Sector, and Vehicle Sales Analytics and their statuses. Some of the dimensions are not shared with other Core modules because of specific requirements.

Table A-1 Common Pharma Analytics Dimensions

Dimensions in Pre-7.9.4 Releases	Dimensions in Release 7.9.4	Status	Comments
Dim - Accounts	Dim - Customer	Shared	None
Dim - Contacts	Dim - Contact	Shared	New name is singular
Dim - Security Dimension	Dim - Security Dimension	Shared	None
Dim - Time Period	Dim - Date	Shared	None
Dim - Geography	Dim - Pharma Geography	Not shared	Use Pharma-specific dimension in Core
Dim - Geography_Account	Dim - Pharma Geography_Account	Not shared	Use Pharma-specific dimension in Core
Dim - Geography_Contact	Dim - Pharma Geography_Contact	Not shared	Use Pharma-specific dimension in Core
Dim - Products	Dim - Pharma Products	Not shared	Use Pharma-specific dimension in Core

Table A-2 Common Consumer Sector Analytics Dimensions

Dimensions in Pre-7.9.5 Releases	Dimensions in Release 7.9.5	Status	Comments
Dim - Account Geography	Dim - Account Geography	Shared	None
Dim - Accounts	Dim - Customer	Shared	None
Dim - Accounts Hierarchy	Dim - Accounts Hierarchy	Shared	None
Dim - Employees	Dim - Employees	Shared	None
Dim - End Date	Dim - End Date	Shared	None
Dim - Position Hierarchies	Dim - Position	Shared	Dim - Position in 7.9.5 combines both Position and Position Hierarchy
Dim - Positions	Dim - Position	Shared	Dim - Position in 7.9.5 combines both Position and Position Hierarchy
Dim - Product Categories Hierarchy	Dim - CS Product Category Hierarchy	Not shared	Use CS-specific dimension in Core
Dim - Products	Dim - CS Product	Not shared	Use CS-specific dimension in Core
Dim - Start Date/Date	Dim - Start Date	Shared	None

Table A-3 Common Vehicle Sales Analytics Dimensions

Dimensions in Pre-7.9.5 Releases	Dimensions in Release 7.9.5	Status	Comments
Dim - Accounts	Dim - Customer	Shared	None
Dim - Accounts Geography	Dim - Account Geography	Shared	None
Dim - Contacts Geography	Dim - Person Geography	Shared	None
Dim - Date	Dim - Date	Shared	None
Dim - Households	Dim - Households	Shared	None
Dim - Individuals	Dim - Contact	Shared	None
Dim - Lease/Loan Expiry	Dim - End Date	Shared	None
Dim - Product Hierarchy	Dim - Product Hierarchy	Shared	None
Dim - Products	Dim - Product	Shared	None
Dim - Vehicle	Dim - Asset	Shared	None

A.2 Merging Siebel Analytics and Oracle BI Repositories

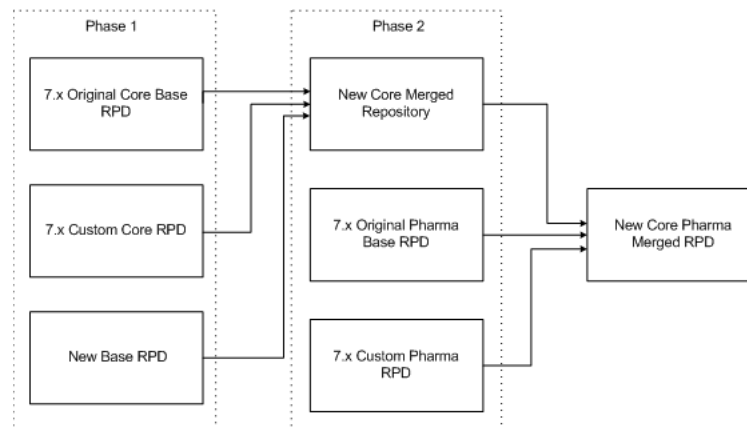
The Pharma, Consumer Sector, and Vehicle Sales Analytics upgrade process involves merging customizations of prior releases of the Siebel Analytics or Oracle BI

repositories into the current version of the Oracle BI repository. This process follows the same principle as the Core upgrade process but includes some unique steps.

In this section, the upgrade process for Pharma Analytics from pre-7.9.4 releases to release 7.9.4 and higher will be used as an example. The upgrade process for Consumer Sector and Vehicle Sales Analytics from pre-7.9.5 releases to release 7.9.5 and higher are the same as that for the upgrade to Pharma Analytics release 7.9.4 and higher.

As shown in [Figure A-1](#), there are two phases to the Pharma Analytics upgrade process.

Figure A-1 Phases of Repository Merge for Pharma Analytics



Phase 1

As shown in [Figure A-1](#), there are two phases in the Pharma Analytics upgrade process. Phase 1 involves merging the 7.x original Core base repository (the repository that shipped with the release of Siebel Analytics or Oracle BI you are currently running), the 7.x custom Core repository, and the new base repository into one output repository.

Phase 2

Phase 2 involves merging the output repository of Phase 1 with the 7.x original Pharma Analytics base repository and the 7.x custom Pharma Analytics repository. The output repository of Phase 2 is a merged Pharma and Core repository that includes your customizations from prior releases and the new data model.

Note: You need to perform both phases of the upgrade process if your current, customized repository has both Core and Pharma models and the Core model is customized.

If your current, customized repository does not include the Core model or includes the Core model but the Core model is not customized, you only need to perform Phase 2.

If you only perform Phase 2, then you will not have a new Core merged RPD (as shown in [Figure A-1](#)). Instead, you will use the new base RPD in place of the new Core merged RPD.

See [Table A-4](#) and [Table A-5](#) for a description of the different RPD file names.

The tasks in this section refer to multiple versions of the Siebel Analytics or Oracle BI repository. [Table A-4](#) provides the names and descriptions of the repositories used in Phase 1. [Table A-5](#) provides the names and descriptions of the repositories used in Phase 2.

Table A-4 Phase 1 Repository Files

Repository File	Description
7x_original_Core_base.rpd	The trimmed, original (standard) repository for the version you are upgrading from.
7x_custom_Core.rpd	The trimmed, custom 7.x repository that includes the Core model customizations.
New_base.rpd	The trimmed, new repository that includes related Core and Pharma modules in one Core model (logical folder).

Table A-5 Phase 2 Repository Files

Repository File	Description
New_Core_merged.rpd	The output repository from Phase 1. If you do not perform Phase 1, then use the new base RPD in place of the new Core merged RPD.
7x_original_Pharma_base.rpd	The trimmed, original 7.x repository that includes the Pharma model, which was not customized.
7x_custom_Pharma.rpd	The trimmed, customized repository that includes a customized Pharma model.
New_Core_Pharma_merged.rpd	The final, merged new repository that includes a single Core model (logical folder), including all the customized content.

A.2.1 Creating Working Folders

You will use the working folders to hold the repository files after the different stages of the merge process.

Create a folder for the merge process, such as \OracleBIPharmaUpgrade, and then create the following subfolders:

- Original
- AfterTrimDown
- AfterEqualize
- AfterMerge
- AfterManualWork
- AfterRegressions

A.2.2 Trimming Repository Files

Trimming the repositories so that you upgrade only the content that is in use can narrow the upgrade scope and eliminate unnecessary complexities during the merge process.

You can trim down repository files by extracting projects or manually trimming down repository objects.

Note: The Security Group property controls the presentation catalogs that will be extracted. If a presentation catalog is added to a project for extract but none of the groups in the project has visibility to see the catalog, it will not show up in the extracted repository file.

A.2.2.1 Trimming the Original Base Repository Files

You should trim the original base repository file to meet your business requirements.

To trim the original base repository files

1. If you are performing Phase 1 of the upgrade process, trim the original Core base repository file.
2. Save the file as 7x_original_Core_base.rpd (where 7x represents the release of Siebel Analytics or Oracle BI) in the AfterTrimDown subfolder.
3. Trim the original Pharma base repository file.
4. Save the file as 7x_original_Pharma_base.rpd in the AfterTrimDown subfolder.

A.2.2.2 Trimming the New Base Repository File

The new base repository file that you received with the current Oracle BI Applications release, contains one Core model (logical folder), which holds Pharma and Core repository objects.

To trim the new base repository file

1. Trim the new base repository file.
2. Save the file as New_base.rpd in the AfterTrimDown subfolder.

A.2.2.3 Trimming the Custom Repository Files

If you are performing Phase 1 of the upgrade process, you will have a 7x custom Core repository file (7x_custom_core.rpd) that contains the 7x original Core base contents and your customizations.

You will also have a 7x custom Pharma repository file (7x_custom_Pharma.rpd) that contains the 7x original Pharma base contents and your customizations.

Save these files in the AfterTrimDown subfolder.

A.2.3 Renaming Objects in the Original Pharma Base Repository File

In the 7x original Pharma base repository file and the 7x custom Pharma repository file, you need to rename some repository objects to be compatible with the new data model.

To rename objects in the Pharma repository files

1. In the Server Administration Tool, open the 7x_original_Pharma_base.rpd file.
2. In the Business Model and Mapping layer, rename Business Model Pharma to Core.
3. In the Physical layer, do the following:
 - a. Rename Database Pharma Data Warehouse to Oracle Data Warehouse.
 - b. Make sure the Connection Pool is named as Oracle Data Warehouse Connection Pool.

- c. Make sure the Catalog entry (below the Connection Pool entry) is named as Catalog.
 - d. Make sure the Schema entry (below the Catalog entry) is named as dbo.
4. Repeat Steps 1 through 3 using the 7x_custom_Pharma.rpd file.

A.2.4 Equalizing the Oracle BI Repositories

The equalization process in the standard Oracle BI repository upgrade uses the original base repository from a previous release as a starting point. This type of equalization is referred to as "backward" equalization. The Pharma upgrade uses what is called "forward" equalization, in which you use the repository file from the current release as a starting point and equalize the 7x base repository file and the 7x custom repository file to it.

You will first need to prepare the MAP file for "forward" equalization before you execute the equalization.

To prepare the MAP file for "forward" equalization

1. Copy the appropriate MAP file from the \OracleBI\Upgrade folder into the folder where you will execute equalizerpds.exe.

Note: The naming convention of the MAP files is rename<release from which you are upgrading>-<release to which you are upgrading>.map.

For example if you are upgrading from release 7.8.4 to release 7.9.5, you should use the file rename784_795.map.

2. Open the MAP file in Excel.
3. In the Text Import Wizard, accept the default in the Step 1 dialog by clicking Next.
4. In the Step 2 dialog of the Text Import Wizard, do the following:
 - a. Make sure the Tab check box is selected in the Delimiters region.
 - b. In the Text qualifier list, select None.
 - c. Click Next.
5. In the Step 3 dialog, click Finish.

The file opens in an Excel spreadsheet window and contains three columns.

6. Switch the order of the second and third columns by cutting the third column and inserting it as the second column.
7. Save the file.
8. Check to make sure the string format is correct.

To equalize the repository files

1. Run equalizerpds.exe.

An example of the equalizerpds command is as follows:

```
equalizerpds -A Administrator -B SADMIN  
-C \OracleBIPharmaUpgrade\AfterTrimDown\New_base.rpd  
-D Administrator -E SADMIN  
-F \OracleBIPharmaUpgrade\AfterTrimDown\7x_custom_Pharma.rpd
```

```
-O \OracleBIPharmaUpgrade\AfterEqualize\7x_custom_Pharma.rpd
-X -J rename784-795.map
```

If the equalizerpds.exe executable file runs correctly, no errors are returned.

2. Repeat Step 1 to equalize each of the files in AfterTrimDown to New_base.rpd.
3. To verify the process completed successfully, compare the size of the repositories. The output repository (-O) should be close to the same size as the repository you equalized (-F).

A.2.5 Merging the Core Repository Files

Note: You need to perform this procedure if your customized repository has both Core and Pharma models and the Core model is customized. If the customized repository does not include the Core model or includes the Core model but the Core model is not customized, you do not need to perform this step.

This step involves a three-way merge on the Core repository. For a description of the files that will be merged in this procedure, see [Section A.2, "Merging Siebel Analytics and Oracle BI Repositories."](#)

To merge the core repository files

1. Copy 7x_original_Core_base.rpd, 7x_custom_Core.rpd, and New_base.rpd to the AfterMerge folder.
2. In the Server Administration Tool, open the New_base.rpd file.
3. From the Administration Tool menu bar, select File, then select Merge.
4. In the Select Original Repository dialog, select 7x_original_Core_base.
5. Enter the password, and click OK.
6. Click Select for the Modified Repository field.
7. In the Select Modified Repository dialog, select 7x_custom_Core.rpd.
8. Click Open, type the password, and then click OK.
9. In the Decision list, select the action you want to take regarding the repository change, or accept the default action.

For information about making decisions, see [Section A.2.7, "Making Merge Decisions in the Server Admin Tool."](#)

10. To locate subsequent rows with empty Decision fields, click the Decision header cell.

When all rows have a value in the Decision field, the Merge button is enabled.

11. Click Merge.

This process can take up to 40 minutes, depending on the size of the repositories you are working with. A message will alert you when the merge is complete.

12. Click Yes when asked if you want to run a consistency check.

The number of errors returned by the consistency check is an indication of how successful the merge process was. If you receive many errors, for example, over 300 you should analyze the reason for the errors. If the merge process failed to

recognize that two objects are the same, you may need to edit the rename file or add your own rename file if you have renamed many of the objects and the upgrade engine failed to relate them to the original objects.

You also may need to change the actions you selected in the Decision drop-down list before rerunning the merge. This could save you time by reducing the number of errors that you will need to fix manually.

Once you are satisfied with the results of the merge, you should fix the remaining errors manually. It is important that you fix all errors before moving on to the next step. This repository serves as the input for the next stage.

You should also check that all of your customized objects are present and that no duplicate physical tables were introduced. To check for duplicate tables, search for physical tables using a query such as:

```
where name like '*#1'
```

13. Save the merged repository as New_Core_merged.rpd .

A.2.6 Merging the Pharma Repository Files

This step involves a three-way merge on the Pharma repository. For a description of the files that will be merged in this procedure, see [Section A.2, "Merging Siebel Analytics and Oracle BI Repositories."](#)

Note: If you did not perform Phase 1 of the upgrade process you will not have a New_Core_merged.rpd file. In place of the New_Core_merged.rpd file, you should use the New_new_base.rpd file.

To merge the Pharma repository files

1. Copy New_Core_merged.rpd, 7x_original_Pharma_base.rpd, and 7x_custom_Pharma.rpd to the AfterMerge folder.
2. In the Server Administration Tool, open the New_Core_merged.rpd file.
3. From the Administration Tool menu bar, select File, then select Merge.
4. In the Select Original Repository dialog, select 7x_original_Pharma_base.
5. Enter the password, and click OK.
6. Click Select for the Modified Repository field.
7. In the Select Modified Repository dialog, select 7x_custom_Pharma.rpd.
8. Click Open, type the password, and then click OK.
9. In the Decision list, select the action you want to take regarding the repository change, or accept the default action.

For information about making decisions, see [Section A.2.7, "Making Merge Decisions in the Server Admin Tool."](#)
10. To locate subsequent rows with empty Decision fields, click the Decision header cell.

When all rows have a value in the Decision field, the Merge button is enabled.
11. Click Merge.

This process can take up to 40 minutes, depending on the size of the repositories you are working with. A message will alert you when the merge is complete.

12. Click Yes when asked if you want to run a consistency check.

The number of errors returned by the consistency check is an indication of how successful the merge process was. If you receive many errors, for example, over 300, you should analyze the reason for the errors. If the merge process failed to recognize that two objects are the same, you may need to edit the rename file or add your own rename file if you have renamed many of the objects and the upgrade engine failed to relate them to the original objects.

You also may need to change the actions you selected in the Decision drop-down list before rerunning the merge. This could save you time by reducing the number of errors that you will need to fix manually.

Once you are satisfied with the results of the merge, you should fix the remaining errors manually. It is important that you fix all errors before moving on to the next step. This repository serves as the input for the next stage.

You should also check that all of your customized objects are present and that no duplicate physical tables were introduced. To check for duplicate tables, search for physical tables using a query such as:

```
where name like '*#1'
```

13. Save the merged repository as New_Core_Pharma_merged.rpd .

A.2.7 Making Merge Decisions in the Server Admin Tool

When making decisions about merging repository objects in the Server Admin Tool, you should consider the following points:

- For objects that do not appear in the new base repository, you should normally choose "current," which will incorporate the changes into the new base repository.
- For objects added to the new base repository, you should normally choose "current," which will keep the changes.
- For objects in the original repository that were replaced with new objects, you may see decisions for removing the old objects from the current repository and adding the new objects. Choosing "current" will replace the old objects.
- For new customizations that you added to a repository, choose "modified" to keep the changes.
- If an object is changed in both the customized repository and the new base repository, the description "changed in both," may appear. In such cases, choose "current" to keep the object as it is in the new base repository, or choose "modified," to keep the object as it is in the customized repository.

A.3 Replacing Common Dimensions After the Repository Merge

After the repository file merge, all preconfigured presentation dimension tables and columns should be merged properly and sourcing from the new logical dimensions in the Core model. For example, Account Name in the Account presentation table should source from "Core"."Dim - Customer"."Account Name," and Gross Margin in the Product presentation table should source from "Core"."Dim - Pharma Products"."Gross Margin."

However, customized presentation tables and columns may still source from old dimensions. [Table A-6](#), [Table A-7](#), and [Table A-8](#) list the common dimensions that have new names in the new release. If you customized presentation tables or columns from the old dimensions listed in [Table A-6](#), [Table A-7](#), and [Table A-8](#) you need to replace the old logical source tables or table columns with the new ones.

A way to quickly allocate these problematic presentation tables or columns would be to do the following:

1. In the Core logical folder in the Server Admin Tool, find the old dimension name as listed in [Table A-6](#), [Table A-7](#), or [Table A-8](#), for example, "Dim - Accounts."
2. Right-click the old dimension name, then click Display Related, and then click Presentation Column.
3. Replace the presentation columns with the same columns from the new logical dimension, for example, "Dim - Customer"
4. To verify you replaced the presentation columns correctly, search for presentation columns on the Core logical dimension, such as "Core - Accounts". If the return is empty, then it is safe to delete the old dimension, for example, "Dim - Accounts"

The following common dimensions in 7.x need to be reviewed and removed.

Table A-6 Names for Pharma Analytics Common Dimensions in Pre-7.9.4 Releases

Name of Dimension in Pre-7.9.4 Releases	Name of Dimension in Release 7.9.4	Status	Comments
Dim - Accounts	Dim - Customer	Shared	None
Dim - Contacts	Dim - Contact	Shared	New name is singular
Dim - Time Period	Dim - Date	Shared	None

Table A-7 Names for Consumer Sector Analytics Common Dimensions in Pre-7.9.5 Releases

Name of Dimension in Pre-7.9.5 Releases	Name of Dimension in Release 7.9.5	Status	Comments
Dim - Accounts	Dim - Customer	Shared	None
Dim - Position Hierarchies	Not applicable	Shared	Merged into Dim - Position in Core
Dim - Positions	Dim - Position	Shared	Dim - Position combines both Position and Position Hierarchy
Dim - Start Date/Date	Dim - Start Date	Shared	None

Table A-8 Names for Vehicle Sales Analytics Common Dimensions in Pre-7.9.5 Releases

Name of Dimension in Pre-7.9.5 Releases	Name of Dimension in Release 7.9.5	Status	Comments
Dim - Accounts	Dim - Customer	Shared	None
Dim - Accounts Geography	Dim - Account Geography	Shared	None
Dim - Contacts Geography	Dim - Person Geography	Shared	None

Table A–8 (Cont.) Names for Vehicle Sales Analytics Common Dimensions in Pre-7.9.5 Releases

Name of Dimension in Pre-7.9.5 Releases	Name of Dimension in Release 7.9.5	Status	Comments
Dim - Individuals	Dim - Contact	Shared	None
Dim - Lease/Loan Expiry	Dim - End Date	Shared	None
Dim - Products	Dim - Product	Shared	None
Dim - Vehicle	Dim - Asset	Shared	None

Configuring Common Parameters for Upgrading to Oracle BI Applications 7.9.6

This appendix contains instructions for configuring parameters that are specific to the various source systems and that are common across all Oracle BI Applications. It is mandatory that you set the appropriate parameters for your specific source system.

The procedure for setting common parameters entails looking up the parameter value in the DAC's Source System Parameters tab and entering this value in the 796_UPG_PARAMS.txt file, which is stored in the SrcFiles folder on the Informatica Server machine.

This appendix contains the following topics:

- [Section B.1, "Configuring Common Parameters for Oracle Source Systems"](#)
- [Section B.2, "Configuring Common Parameters for Siebel Source Systems"](#)

B.1 Configuring Common Parameters for Oracle Source Systems

This section contains instructions for configuring parameters specific to Oracle source systems.

This section contains the following topics:

- [Section B.1.1, "Configuring Common Parameters for Oracle EBS 11i Source Systems"](#)
- [Section B.1.2, "Configuring Common Parameters for Oracle EBS R12 Source Systems"](#)

B.1.1 Configuring Common Parameters for Oracle EBS 11i Source Systems

This section contains instructions for configuring parameters for Oracle EBS 11i source systems.

To configure parameters for Oracle EBS 11i source systems

1. Navigate to the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
2. Open the file 796_UPG_PARAMS.txt.
3. Note the values for the following parameters:
 - \$\$ORA_DATASOURCE_NUM_ID_LIST
 - \$\$GRAIN
 - \$\$GBL_DATASOURCE_NUM_ID

- \$\$QUALIFICATION_CATEGORY_LIST
- 4. In DAC, go to the Design view, and select the appropriate custom container.
- 5. Select the Source System Parameters tab.
- 6. Query for the parameters listed in Step 3, and compare the values.
- 7. If necessary, change the values for the parameters in the 796_UPG_PARAMS.txt file to match the values for the parameters in the DAC Source System Parameters tab.
- 8. Save the 796_UPG_PARAMS.txt file.

B.1.2 Configuring Common Parameters for Oracle EBS R12 Source Systems

This section contains instructions for configuring parameters for Oracle EBS R12 source systems.

To configure parameters for Oracle EBS R12 source systems

1. Navigate to the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
2. Open the file 796_UPG_PARAMS.txt.
3. Note the values for the following parameters:
 - \$\$START_DATE
 - \$\$END_DATE
4. In DAC, go to the Design view, and select the appropriate custom container.
5. Select the Source System Parameters tab.
6. Query for the parameters listed in Step 3, and compare the values.
7. If necessary, change the values for the parameters in the 796_UPG_PARAMS.txt file to match the values for the parameters in the DAC Source System Parameters tab.
8. Save the 796_UPG_PARAMS.txt file.

B.2 Configuring Common Parameters for Siebel Source Systems

This section contains instructions for configuring parameters specific to Siebel source systems.

To configure parameters for Siebel source systems

1. Navigate to the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
2. Open the file 796_UPG_PARAMS.txt.
3. Note the values for the following parameters:
 - \$\$HI_DT
 - \$\$LOW_DT
 - \$\$NAME_ORDER_WITH_FIRSTNAME
4. In DAC, go to the Design view, and select the appropriate custom container.
5. Select the Source System Parameters tab.

6. Query for the parameters listed in Step 3, and compare the values.
7. If necessary, change the values for the parameters in the 796_UPG_PARAMS.txt file to match the values for the parameters in the DAC Source System Parameters tab.
8. Save the 796_UPG_PARAMS.txt file.

Configuring Application-Specific Parameters and Mappings

This appendix provides information about application-specific parameters and mappings that you may need to configure or update depending on your environment.

This appendix includes the following topics:

- [Section C.1, "Configuring Parameters and Mappings for Oracle Financial Analytics"](#)
- [Section C.2, "Configuring Parameters for Oracle Supply Chain and Order Management Analytics"](#)
- [Section C.3, "Configuring Parameters for Oracle Projects Analytics"](#)

C.1 Configuring Parameters and Mappings for Oracle Financial Analytics

This section provides information about parameters and mappings specific to Oracle Financial Analytics that you may need to configure or update depending on your environment.

This section includes the following topics:

- [Section C.1.1, "Configuring Parameters for Value Set Hierarchies and FSG Hierarchies"](#)
- [Section C.1.2, "Configuring Parameters for GL Data Extraction"](#)
- [Section C.1.3, "Configuring the COGS Fact Mapping for Oracle EBS R12"](#)
- [Section C.1.4, "Configuring the \\$\\$Hint1 Parameter for Oracle Databases"](#)

C.1.1 Configuring Parameters for Value Set Hierarchies and FSG Hierarchies

In Oracle Financial Analytics, the default behavior is for Value Set Hierarchies to be enabled and Financial Statement Generator (FSG) Hierarchies to be disabled.

If you have changed this behavior by disabling Value Set Hierarchies and enabling FSG Hierarchies, then you need to configure the parameters that control this behavior in the 796_UPG_PARAMS.txt file and the DAC configuration tags.

To configure the FSG Hierarchies and Value Set Hierarchies parameters

1. Navigate to the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
2. Open the 796_UPG_PARAMS.txt file.

3. Search for the parameter `$$IS_FSGHIERARCHY_INSTALLED`, and set the value to Y.

For example:

```
$$IS_FSGHIERARCHY_INSTALLED=Y
```

4. Search for the parameter `$$IS_VALUESETHIERARCHY_INSTALLED`, and set the value to N.

For example:

```
$$IS_VALUESETHIERARCHY_INSTALLED=N
```

C.1.2 Configuring Parameters for GL Data Extraction

If you have configured your GL data extraction to limit the set of books or ledgers extracted, you need to set the appropriate parameters in the `796_UPG_PARAMS.txt` file to match the same parameters in the DAC source system parameters.

This section includes the following topics:

- [Section C.1.2.1, "Configuring GL Data Extraction Parameters for Oracle EBS 11i Sources"](#)
- [Section C.1.2.2, "Configuring GL Data Extraction Parameters for Oracle EBS R12 Sources"](#)

C.1.2.1 Configuring GL Data Extraction Parameters for Oracle EBS 11i Sources

Follow this procedure to configure GL data extraction parameters for Oracle EBS 11i sources.

To configure GL data extraction parameters

1. Navigate to the `SrcFiles` folder on the Informatica Server machine, for example, `C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles`.
2. Open the `796_UPG_PARAMS.txt` file.
3. Note the values for the following parameters:
 - `$$FILTER_BY_SET_OF_BOOKS_ID`
 - `$$FILTER_BY_SET_OF_BOOKS_TYPE`
 - `$$SET_OF_BOOKS_ID_LIST`
 - `$$SET_OF_BOOKS_TYPE_LIST`
4. In DAC, go to the Design view, and select the appropriate custom container.
5. Select the Source System Parameters tab.
6. Query for the parameters listed in Step 3, and compare the values.
7. If necessary, change the values for the parameters in the `796_UPG_PARAMS.txt` file to match the values for the parameters in the DAC Source System Parameters tab.
8. Save the `796_UPG_PARAMS.txt` file.

C.1.2.2 Configuring GL Data Extraction Parameters for Oracle EBS R12 Sources

Follow this procedure to configure GL data extraction parameters for Oracle EBS R12 sources.

To configure GL data extraction parameters

1. Navigate to the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
2. Open the 796_UPG_PARAMS.txt file.
3. Note the values for the following parameters:
 - \$\$FILTER_BY_LEDGER_ID
 - \$\$FILTER_BY_LEDGER_TYPE
 - \$\$LEDGER_ID_LIST
 - \$\$LEDGER_TYPE_LIST
4. In DAC, go to the Design view, and select the appropriate custom container.
5. Select the Source System Parameters tab.
6. Query for the parameters listed in Step 3, and compare the values.
7. If necessary, change the values for the parameters in the 796_UPG_PARAMS.txt file to match the values for the parameters in the DAC Source System Parameters tab.
8. Save the 796_UPG_PARAMS.txt file.

C.1.3 Configuring the COGS Fact Mapping for Oracle EBS R12

For Oracle EBS R12 sources, follow this procedure to update the COGS fact mapping.

To configure the COGS fact mapping

1. Launch Informatica PowerCenter Designer.
2. Navigate to the folder UPGRADE_7951_TO_796_ORA12.
3. Open the mapping SDE_ORA_GLCOGSFact_UPG796.
4. Open the maplet mplt_BC_ORA_GLCOGSFact.
5. Open the Source Qualifier Transformation, do the following:
 - a. Open the SQL Query property.
 - b. In the WHERE clause of the query locate the hard-coded filter on the Transaction Type ID and Transaction Action ID.

For example:

```
MMT.TRANSACTION_TYPE_ID IN (15, 33, 10008) AND
(MMT.TRANSACTION_ACTION_ID, MTA.ACCOUNTING_LINE_TYPE ) IN
((27, 2), (1, 36), (36, 35))
```

6. Change the values to the actual values you used in the SDE_ORA_GLCogsFact mapping in the main Informatica Repository code.

C.1.4 Configuring the \$\$Hint1 Parameter for Oracle Databases

If your target data warehouse is an Oracle database server, follow this procedure to set the \$\$Hint1 parameter.

To set the \$\$Hint1 parameter

1. Navigate to the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.

2. Open the 796_UPG_PARAMS.txt file.
3. Locate the \$\$Hint1 parameter for the appropriate version of the Oracle database, and enter the following value:

```
/*+ USE_HASH(W_GL_BALANCE_F, W_GL_ACCOUNT_D, W_GLACCT_SEG_CONFIG_TMP) */
```

For example:

```
mp1t_GLBalanceAggrByAcctSegCodes.$$Hint1=/*+ USE_HASH(W_GL_BALANCE_F, W_GL_ACCOUNT_D, W_GLACCT_SEG_CONFIG_TMP) */
```

4. Save the 796_UPG_PARAMS.txt file.

C.2 Configuring Parameters for Oracle Supply Chain and Order Management Analytics

This section provides information about parameters specific to Oracle Supply Chain and Order Management Analytics that you need to configure.

This section includes the following topics:

- [Section C.2.1, "Configuring the TIME_GRAIN Parameter for Sales Order Lines Aggregate Fact and Invoice Lines Aggregate Fact Tables"](#)
- [Section C.2.2, "Configuring the \\$\\$PERIOD Parameter for the Customer Status History Fact Table"](#)

C.2.1 Configuring the TIME_GRAIN Parameter for Sales Order Lines Aggregate Fact and Invoice Lines Aggregate Fact Tables

If you are deploying Oracle Supply Chain and Order Management Analytics, you need to configure the TIME_GRAIN parameter for the Sales Order Lines Aggregate Fact table and for the Invoice Lines Aggregate Fact table. For instructions, see the section titled, "Process of Aggregating Oracle Supply Chain and Order Management," in *Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users*.

C.2.2 Configuring the \$\$PERIOD Parameter for the Customer Status History Fact Table

If you are deploying Oracle Supply Chain and Order Management Analytics, you need to configure the \$\$PERIOD parameter for the Customer Status History Fact table. For instructions, see the section titled, "How to Configure the Customer Status History Fact Table," in *Oracle Business Intelligence Applications Configuration Guide for Informatica PowerCenter Users*.

C.3 Configuring Parameters for Oracle Projects Analytics

If you are deploying Oracle Projects Analytics, follow the procedure in this section to configure the IS_PROJECTS_INSTALLED parameter.

To configure the IS_PROJECTS_INSTALLED parameter

1. Navigate to the SrcFiles folder on the Informatica Server machine, for example, C:\Program Files\Informatica PowerCenter 8.6\server\infa_shared\SrcFiles.
2. Open the 796_UPG_PARAMS.txt file.
3. Search for the parameter \$\$IS_PROJECTS_INSTALLED, and set the value to Y.

For example:

```
$$IS_PROJECTS_INSTALLED=Y
```

4. Save the 796_UPG_PARAMS.txt file.

Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File

This appendix contains the following topic:

- [Section D.1, "Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File"](#)

D.1 Database-Specific Parameter Settings for the reset_infa_seq_gen.bat File

Use the reset_infa_seq_gen.bat script to initialize the Informatica sequence generator for incremental runs on the Upgrade repository.

The reset_infa_seq_gen.bat file requires you to enter database-specific parameter values. This section includes parameter settings for the following databases:

- [Table D-1, "Oracle Database Parameter Settings for reset_infa_seq_gen.bat"](#)
- [Table D-2, "SQL Server Database Parameter Settings for reset_infa_seq_gen.bat"](#)
- [Table D-3, "IBM DB2 Database Parameter Settings for reset_infa_seq_gen.bat"](#)

Oracle Database Parameter Settings

Table D-1 Oracle Database Parameter Settings for reset_infa_seq_gen.bat

Parameter	Setting
OLAP USERNAME	Enter the data warehouse database user ID.
OLAP PASSWORD	Enter the user ID/password@connection string of the data warehouse database.
INFORMATICA DB USERNAME	Enter the user ID of the Upgrade Informatica Repository database.
INFORMATICA DB PASSWORD	Enter the password@connection string of the Upgrade Informatica Repository database.
INFORMATICA REPOSITORY	Enter the name of the Upgrade Informatica Repository.
INFORMATICA REPOSITORY USERNAME	Enter the user ID of the Upgrade Informatica Repository.

Table D–1 (Cont.) Oracle Database Parameter Settings for reset_infa_seq_gen.bat

Parameter	Setting
INFORMATICA REPOSITORY PASSWORD	Enter the password for the Upgrade Informatica Repository.
INFORMATICA REPOSITORY SERVER ADDRESS	Enter the name of the machine that runs the Repository Service for the Upgrade Repository.
INFA_PORT	Enter the port number for the Repository Service. The default is 6001.
INFA_FODLER	Enter the value UPGRADE.

SQL Server Database Parameter Settings

Table D–2 SQL Server Database Parameter Settings for reset_infa_seq_gen.bat

Parameter	Setting
OLAP_SERVER	Enter the name of the SQL Server for the OLAP database.
OLAP_DATABASE	Enter the name of the database on which OLAP data is available.
OLAP_USERNAME	Enter the user ID of the data warehouse database.
OLAP_PASSWORD	Enter the user ID/password@connection string of the data warehouse database.
INFORMATICA SERVER ADDRESS	Enter the name of the SQL Server for the Informatica Repository database.
INFORMATICA DATABASE	Enter the name of the database in which the Informatica Repository metadata is available.
INFORMATICA DB USERNAME	Enter the user ID of the Upgrade Informatica Repository database.
INFORMATICA DB PASSWORD	Enter the password @connection string of the Upgrade Informatica Repository database.
INFORMATICA REPOSITORY	Enter the name of the Upgrade Informatica Repository.
INFORMATICA REPOSITORY USERNAME	Enter the user ID of the Upgrade Informatica Repository.
INFORMATICA REPOSITORY PASSWORD	Enter the password of the Upgrade Informatica Repository.
INFORMATICA REPOSITORY SERVER ADDRESS	Enter the name of the machine that runs the Repository Service for the Upgrade Repository.
INFA_PORT	Enter the port number for the Repository Service. The default is 6001.
INFA_FOLDER	Enter the value UPGRADE.

IBM DB2 Database Parameter Settings**Table D-3 IBM DB2 Database Parameter Settings for reset_infa_seq_gen.bat**

Parameter	Setting
OLAP_USERNAME	Enter the user ID of the data warehouse database.
OLAP_PASSWORD	Enter the user ID/password@connection string of the data warehouse database.
INFORMATICA_DB_USERNAME	Enter the user ID of the Upgrade Informatica Repository database.
INFORMATICA_DB_PASSWORD	Enter the password @connection string of the Upgrade Informatica Repository database.
INFA_DATABASE_ALIAS	Enter the name of the database in which Informatica Repository metadata is available.
INFORMATICA_REPOSITORY	Enter the name of the Upgrade Informatica Repository.
INFORMATICA_REPOSITORY_USERNAME	Enter the user ID of the Upgrade Informatica Repository.
INFORMATICA_REPOSITORY_PASSWORD	Enter the password for the Informatica Repository.
INFORMATICA_REPOSITORY_SERVER_ADDRESS	Enter the name of the machine that runs the Repository Service for the Upgrade Repository.
INFA_PORT	Enter the port number for the Repository Service. The default is 6001.
INFA_FOLDER	Enter the value UPGRADE.

Index

B

best practices
 preparing to upgrade, 2-2

C

Consumer Sector
 upgrading Oracle BI repository, A-1

O

Oracle EBS
 upgrading, 7-1

P

PeopleSoft
 upgrading, 7-1
Pharma Analytics
 upgrading Oracle BI repository, A-1
preupgrade
 best practices, 2-2
 points to consider, 2-2
process flow, 2-3

S

Siebel Analytics
 upgrading from release 7.5.x, 3-1
Siebel Business Analytics
 upgrading from releases 7.7, 7.7.x and 7.8.x, 4-1
stages of upgrade
 process flow, 2-3
 sequence of stages, 2-4
supported upgrade paths, 2-1

U

upgrade
 preparing, 2-2
 preupgrade considerations, 2-2
 process flow, 2-3
 sequence of steps, 2-4
 supported paths, 2-1
 upgrading for PeopleSoft, 7-1
 upgrading from Oracle BI Applications 7.9.x for

Siebel Sources, 5-1
upgrading from Siebel Analytics 7.5.x, 3-1
upgrading from Siebel Business Analytics 7.7,
 7.7.x, 7.8.x, 4-1
upgrading older releases of Oracle BI for Oracle
 EBS, 6-1

V

Vehicle Sales
 upgrading Oracle BI repository, A-1

