

**Oracle® Business Intelligence Applications**

Installation and Configuration Guide

Version 7.9.4

**E10742-01**

December 2007

Copyright © 2007, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States 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, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

---

---

# Contents

<b>Preface</b> .....	xxv
Audience .....	xxv
Documentation Accessibility .....	xxv
Related Documents .....	xxvi
Conventions .....	xxvi
<b>1 What's New in This Version</b>	
1.1 What's New In Oracle Business Intelligence Applications Version 7.9.4? .....	1-1
<b>2 Overview of Oracle Business Intelligence Applications</b>	
2.1 What is Oracle Business Intelligence Applications? .....	2-1
2.2 Oracle Business Analytics Warehouse Overview .....	2-2
2.3 Oracle Business Analytics Warehouse Architecture.....	2-3
2.3.1 Oracle Business Analytics Warehouse Architecture Components.....	2-3
2.4 Roadmap To Installing, Configuring, and Customizing Oracle BI Applications .....	2-5
<b>3 Pre-installation and Predeployment Requirements for Oracle BI Applications</b>	
3.1 Oracle Business Analytics Warehouse Deployment Configuration.....	3-2
3.2 Operating System, Driver, and Connectivity Requirements for Oracle Business Analytics Warehouse .....	3-2
3.3 General Guidelines for Setting Up Oracle Business Analytics Warehouse.....	3-3
3.4 IBM DB2 UDB-Specific Database Guidelines for Oracle Business Analytics Warehouse	3-5
3.5 IBM DB2 UDB zOS and OS/390 and Z/os-Specific Database Guidelines for Oracle Business Analytics Warehouse .....	3-6
3.6 Oracle-Specific Database Guidelines for Oracle Business Analytics Warehouse.....	3-7
3.7 Additional Suggestions for Optimizing Oracle Performance in Oracle Business Analytics Warehouse .....	3-7
3.8 SQL Server-Specific Database Guidelines for Oracle Business Analytics Warehouse .....	3-8
3.8.1 Setting the ANSI NULL Option.....	3-9
3.8.2 Modifying the DB Library Options Setting.....	3-9
3.8.3 Recommended SQL Server Database Parameters .....	3-9
3.9 Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse .....	3-11
3.9.1 General Guidelines for Teradata Deployments .....	3-11
3.9.2 Best Practices For Teradata Deployments.....	3-11
3.9.2.1 LEFT OUTER JOIN Issue .....	3-12

3.9.2.2	Database Statistics .....	3-13
3.9.2.3	Additional Statistics Required .....	3-13
3.9.2.4	Collect Statistics As A Pre-requisite.....	3-13
3.9.2.5	Group By Vs Distinct .....	3-13
3.9.2.6	Pruning Of Mapping.....	3-13
3.9.2.7	Loader Configurations.....	3-13
3.9.2.8	Fastload .....	3-14
3.9.2.9	Mload .....	3-15
3.9.2.10	Tpump.....	3-15
3.10	Deploying Oracle Business Analytics Warehouse in Latin-1 General, Unicode and Non-English Environments.....	3-15
3.10.1	Code Page Overview.....	3-16
3.10.1.1	Choosing Code Pages .....	3-16
3.10.1.2	Code Page Compatibility .....	3-16
3.10.2	Source and Data Warehouse Code Pages for Latin-1 General (7-bit ASCII) to Latin-1 General (7-bit ASCII)	3-17
3.10.2.1	Informatica Server and Repository Server Running on Windows with OS ENU .....	3-17
3.10.2.2	Informatica Server and Repository Server Running on UNIX with OS ENU ..	3-17
3.10.2.3	Informatica Server on UNIX, Repository Server on Windows, both with OS ENU .....	3-18
3.10.3	Source and Data Warehouse Code Pages for Latin-1 General (8-bit ASCII) to Latin-1 General (8-bit ASCII)	3-18
3.10.3.1	Informatica Server and Repository Server Running on Windows with OS ENU .....	3-18
3.10.3.2	Informatica Server and Repository Server Running on UNIX with OS ENU ..	3-19
3.10.3.3	Informatica Server on UNIX, Repository Server on Windows, both with OS ENU .....	3-19
3.10.4	Source and Data Warehouse Code Pages for Unicode to Unicode .....	3-20
3.10.4.1	Informatica Server and Repository Server Running on Windows with OS ENU .....	3-20
3.10.4.2	Informatica Server and Repository Server Running on UNIX with OS ENU ..	3-20
3.10.4.3	Informatica Server on UNIX, Repository Server on Windows, both with OS ENU .....	3-21
3.10.5	Source and Data Warehouse Code Pages for Code Page to Unicode .....	3-21
3.10.5.1	Informatica Server and Repository Server Running on Windows and OS <LANG> .....	3-21
3.10.5.2	Informatica Server and Repository Server Running on UNIX with OS <LANG> .....	3-22
3.10.5.3	Informatica Server on UNIX, Repository Server on Windows, both with OS <LANG> .....	3-22
3.10.6	Source and Data Warehouse Code Pages for Code Page to Code Page .....	3-22
3.10.6.1	Informatica Server and Repository Server Running on Windows and OS <LANG> .....	3-23
3.10.6.2	Informatica Server and Repository Server Running on UNIX and OS <LANG> .....	3-23
3.10.6.3	Informatica Server on UNIX, Repository Server on Windows, Both with OS <LANG> .....	3-23
3.10.7	Setting the NLS_LANG Environment Variable for Oracle Databases.....	3-24
3.10.8	Setting the DB2CODEPAGE Environment Variable for DB2 Databases .....	3-25

3.11	Miscellaneous Information About Oracle Business Intelligence Applications Deployments.....	3-26
3.11.1	Preconfigured Mapping for the SA System Subject Area.....	3-26
3.11.2	Using Initialization Blocks.....	3-27
3.11.3	Performance Improvements for Change Capture for Incremental ETL Run .....	3-28
3.11.3.1	Implementing Transactional Database (OLTP) Indexes.....	3-28
3.11.3.2	An Example Of Change Capture SQL And Required Indexes.....	3-28
3.11.4	Running an ETL with a Non-English OLTP Data Source.....	3-29

## 4 Installing and Configuring Oracle BI Applications on Windows

4.1	Installation and Configuration Process Task List for Windows .....	4-2
4.2	How to Install the Java SDK.....	4-4
4.3	How to Install Oracle Business Intelligence Applications (Windows) .....	4-4
4.3.1	Prerequisites to Running The Oracle Business Intelligence Applications Installation Wizard .....	4-4
4.3.2	How to Run The Oracle Business Intelligence Applications Installation Wizard .....	4-5
4.4	How to Install Informatica PowerCenter and Required Patches (Windows).....	4-7
4.4.1	How to Install Informatica PowerCenter .....	4-8
4.4.2	How to Install the Informatica EBF Patch 2.....	4-10
4.4.3	(Teradata specific) How to Install Informatica Patch1 .....	4-10
4.5	How to Set Up the Informatica Server and Informatica Repository Server.....	4-10
4.5.1	About migrating the Informatica repository .....	4-11
4.5.2	How to Set Up Informatica Data Code Pages Validation.....	4-12
4.5.3	How to Set Up The Informatica Server .....	4-12
4.5.4	How to Set Up The Informatica Repository Server .....	4-14
4.5.5	How to Copy Source Files and Lookup Files .....	4-15
4.5.6	How to Start the Informatica Services .....	4-15
4.5.7	How to Create ODBC Database Connections.....	4-16
4.6	How to Set Environment Variables on Windows .....	4-18
4.6.1	How to Set the PATH Environment Variable for the Informatica Server and Informatica Repository Server .....	4-18
4.6.2	How to Set Siebel UnicodeDB Environment Variables on Windows .....	4-18
4.6.3	How to Set the Bulk Load Registry for DB2/390 Databases .....	4-19
4.7	How to Copy Hibernate Libraries.....	4-19
4.8	How to Configure the DAC Client.....	4-20
4.8.1	How to Configure the DAC Config.bat File .....	4-20
4.8.2	How to Install JDBC Drivers for DAC Database Connectivity.....	4-21
4.9	How to Create the Required Databases.....	4-21
4.10	How to Create the DAC Metadata Repository .....	4-22
4.10.1	How to Create a DAC Connection .....	4-23
4.10.2	How to Log Into the DAC .....	4-24
4.10.3	How to Import Metadata into the DAC repository .....	4-24
4.11	How to Configure the DAC Metadata Repository .....	4-26
4.11.1	How to Set Up DAC System Properties .....	4-26
4.11.2	How to Specify Transactional and Data Warehouse Data Sources in the DAC.....	4-27
4.11.3	How to Configure Email Recipients in the DAC Client.....	4-30
4.11.4	How to Activate Join Indices for Teradata Databases.....	4-30

4.12	How to Configure the Oracle Business Analytics Warehouse Database.....	4-30
4.12.1	How to Create the SSE Role .....	4-31
4.12.2	How to Create Data Warehouse Tables.....	4-31
4.12.3	How to Create Data Warehouse Tables on a Teradata Database .....	4-35
4.13	How to Register Informatica Servers in the DAC Client .....	4-36
4.13.1	How to Register Informatica Servers in the DAC Client .....	4-36
4.14	How to Configure and Manage the Informatica Repository .....	4-39
4.14.1	About the Versioned Informatica Repository.....	4-39
4.14.1.1	Summary of Versioning in the Informatica Repository .....	4-39
4.14.1.2	Applying the Team Based Development License to the Informatica Repository .....	4-40
4.14.1.3	Customization of Repository Objects Using Check Out and Check In .....	4-40
4.14.2	How to Register the Informatica Repository Server.....	4-41
4.14.3	How to Load the Pre-Built Repository Into Informatica.....	4-42
4.14.4	How to Configure the Informatica Repository in Workflow Manager .....	4-44
4.14.5	How to Stop and Start the Informatica Repository (Optional) .....	4-47
4.14.6	How to Back Up and Restore the Informatica Repository (Optional) .....	4-48
4.15	How to Set the Code Page File Property for Unicode Environments on Teradata Databases.....	4-50
4.16	How to Set Up the HOSTS File for Teradata Installations on Windows .....	4-50
4.17	How to Verify the Informatica Repository Configuration.....	4-50
4.18	How to Configure the DAC Server .....	4-51
4.18.1	How to Configure the Connection Between the DAC Server and DAC Repository .....	4-51
4.18.2	How to Configure Email in the DAC Server (Optional).....	4-53
4.18.3	How to Set DAC Source System Parameters .....	4-54
4.18.4	How to Start the DAC Server.....	4-55
4.19	How to Start the Informatica Server .....	4-55
4.20	How to Update Siebel Transactional Database Schema Definitions.....	4-56
4.20.1	How to Run the DDLIMP Tool From The Command Line.....	4-56
4.20.2	How to Apply Siebel (CRM) and Oracle E-Business Suite Schema Changes For Oracle, DB2/UDB, MSSQL.....	4-56
4.20.3	How to Apply CRM Schema Changes For DB2 on 390 or z/os .....	4-57
4.20.4	How to Verify the Siebel (CRM) and Oracle E-Business Suite Schema Changes ...	4-58
4.20.5	About Delete Triggers.....	4-58
4.21	How to Create Stored Procedures for DB2-UDB.....	4-59
4.21.1	How to Verify the existence of stored procedures.....	4-59
4.21.2	How to Create DB2 stored procedures.....	4-59
4.22	Teradata-Specific Installation and Configuration Checklist (Windows).....	4-60
4.23	Upgrade Issues for Oracle BI Applications.....	4-61
4.23.1	Moving From a Test Environment to a Production Environment .....	4-61
4.23.2	Migrating Seed Data from an Siebel Source System to the Oracle BI Repository ...	4-61

## 5 Installing Oracle BI Applications on Unix

5.1	Overview of Installing and Configuring Oracle BI Applications on UNIX.....	5-2
5.2	Installation and Configuration Process Task List for UNIX.....	5-3
5.3	How to Install Oracle BI Applications on UNIX .....	5-6
5.4	How to Install and Configure the Informatica Repository Server on UNIX.....	5-7

5.4.1	How to install and configure the Informatica Repository Server.....	5-7
5.4.2	How to specify the Informatica Repository Server as an environment variable .....	5-9
5.5	How to Install and Configure the Informatica Server on UNIX.....	5-9
5.6	How to Update the Informatica License Files on UNIX .....	5-15
5.7	How to Install the Informatica Patch on UNIX .....	5-16
5.8	How to Create Database Connections for Teradata Installations.....	5-16
5.9	How to Set Environment Variables for the Informatica Server on UNIX .....	5-16
5.9.1	How to Add setenv Variables to .cshrc .....	5-16
5.9.2	How to Set the Code Page for Non_ENU Operating Systems on UNIX.....	5-17
5.9.3	How to Set Siebel UnicodeDB on the Informatica Server if source and target databases are Unicode on UNIX.....	5-17
5.9.4	How to Disable the bulk loader for DB2/390 databases.....	5-17
5.10	How to Set Up the HOSTS File for Teradata Installations on UNIX.....	5-18
5.11	How to Start the Informatica Repository Server on UNIX .....	5-18
5.12	How to Stop the Informatica Repository Server on UNIX .....	5-18
5.13	About the DAC Server on UNIX .....	5-18
5.13.1	How to configure the DAC server on UNIX.....	5-19
5.14	How to Stop and Start the Informatica Server on UNIX.....	5-21
5.14.1	Requirements.....	5-21
5.14.2	How to Start the Informatica Server on UNIX .....	5-21
5.14.3	How to Start the Informatica Server On UNIX .....	5-22
5.15	How to Run the DAC Server on UNIX.....	5-22
5.15.1	How to Start the DAC server on UNIX .....	5-22
5.15.2	How to Run the DAC server in the background.....	5-22
5.16	How to Stop the DAC Server on UNIX .....	5-22
5.17	Teradata-Specific Installation and Configuration Checklist for UNIX .....	5-23

## 6 Customizing the Oracle Business Analytics Warehouse

6.1	Oracle Business Analytics Warehouse Customization Scenarios.....	6-1
6.1.1	Types of Customizations .....	6-2
6.1.2	Upgrade Consideration .....	6-2
6.2	Category 1 Customizations: Adding Columns to Existing Fact or Dimension Tables.....	6-2
6.2.1	About Extending Mappings.....	6-3
6.2.2	Impact of Customization on Upgrade .....	6-4
6.2.2.1	Points to Remember .....	6-5
6.2.3	Typical Steps to Extend Mappings in the Oracle Business Analytics Warehouse.....	6-6
6.2.4	Scenario for Extending the Oracle Business Analytics Warehouse.....	6-6
6.2.4.1	Tips for Modifying the SQL Override in a Source Qualifier .....	6-7
6.2.4.2	Example of Extracting Data from a Siebel Source .....	6-7
6.2.4.3	Including a Source Table for the Change Capture Process .....	6-9
6.2.4.4	Example of Extracting Data from a Non-Siebel Packaged Source .....	6-9
6.2.4.5	Example of Extracting Data from a Universal Source.....	6-11
6.2.4.6	Example of Loading Data into an Existing Target Table.....	6-12
6.2.4.7	Updating the DAC .....	6-13
6.3	Other Types of Customizations Requiring Special Handling.....	6-13
6.3.1	Modifying Category 2 SCD Triggers .....	6-13
6.3.2	Adding Dimensions to Existing Facts.....	6-14

6.3.3	Adding Date Dimensions to Existing Facts .....	6-16
6.3.4	Adding Currencies to an Existing Table .....	6-17
6.4	Category 2 Customizations: Adding Additional Tables .....	6-17
6.4.1	About Creating New Dimension or Fact Tables .....	6-17
6.4.1.1	Required Columns.....	6-18
6.4.1.2	About the Oracle Business Analytics Warehouse DATASRC_NUM_ID Column.....	6-18
6.4.2	Using Custom Folders in the Oracle Business Analytics Warehouse.....	6-19
6.4.3	Creating Custom Informatica Workflows.....	6-19
6.4.4	Important Notes About Customizing the Oracle Business Analytics Warehouse..	6-20
6.4.4.1	Additional Customization Instructions.....	6-20
6.4.5	Including a Source Table for Change Capture Process.....	6-22
6.4.6	Adding a New Dimension in the Oracle Business Analytics Warehouse.....	6-23
6.4.7	Adding a New Fact Table in the Oracle Business Analytics Warehouse .....	6-23
6.4.8	Adding a New Dimension Table for a New Fact Table in the Oracle Business Analytics Warehouse .....	6-24
6.5	Category 3 Customizations: Adding New Data as a Whole Row into a Standard Dimension Table.....	6-24
6.6	Configuring Extracts.....	6-25
6.6.1	Extracting Additional Data .....	6-25
6.6.1.1	Extracting New Data Using an Existing Source Table.....	6-26
6.6.1.2	Extracting Data from a New Source Table.....	6-26
6.6.2	Setting Up the Delimiter for Source Files.....	6-27
6.6.3	Configuring a Source Adapter Mapper.....	6-27
6.7	Configuring Loads.....	6-28
6.7.1	Filtering and Deleting Records .....	6-28
6.7.2	About Primary Extract and Delete Mappings Process.....	6-29
6.7.3	About Working with Primary Extract and Delete Mappings .....	6-30
6.7.3.1	Deleting the Configuration for Source-Archived Records .....	6-30
6.7.3.2	Enabling Delete and Primary Extract Sessions .....	6-31
6.8	Configuring Slowly Changing Dimensions.....	6-31
6.8.1	About Identifying Historically Significant Attributes.....	6-31
6.8.1.1	About the Extract View.....	6-32
6.8.2	Type I and Type II Slowly Changing Dimensions.....	6-32
6.8.2.1	Type I Slowly Changing Dimension.....	6-32
6.8.2.2	Type II Slowly Changing Dimension .....	6-32
6.8.2.3	Effective Dates.....	6-33
6.9	About Stored Lookups .....	6-35
6.10	Codes Lookup.....	6-35
6.10.1	W_CODES Table .....	6-35
6.10.2	Codes Mappings .....	6-35
6.10.3	Codes Mappers .....	6-36
6.10.4	Configuring Extension Column Code Description Lookups .....	6-36
6.11	About Resolving Dimension Keys .....	6-37
6.11.1	Resolving the Dimension Key Using Lookup .....	6-37
6.12	About Domain Values.....	6-38
6.12.1	About the Domain Value Conversion Process .....	6-38
6.12.1.1	Preparing to Define the Rules.....	6-39

6.12.2	About the Importance of Domain Values .....	6-40
6.12.2.1	Hire Count .....	6-40
6.12.2.2	Rehires Ratio .....	6-40
6.12.2.3	New Hire Count .....	6-40
6.12.2.4	Newly Separated Veterans - New Hires .....	6-40
6.12.2.5	Other Protected Veterans - New Hires.....	6-40
6.12.2.6	Special Disabled Veteran Head count - New Hires.....	6-41
6.12.2.7	Vietnam Era Veteran Head count - New Hires.....	6-41
6.12.3	About Extending the Domain Value Set .....	6-41
6.13	Configuring the Domain Value Set with CSV Worksheet Files .....	6-42
6.14	Configuring the Domain Value Set Using PowerCenter Designer .....	6-42
6.15	Configuring Conformed Dimensions .....	6-43
6.15.1	Configuring Conformed Dimensions for Universal Source.....	6-43
6.15.1.1	Product Effective Dates in the Products Dimension .....	6-43

## 7 Integrated Security for Oracle BI Applications

7.1	Types of Security in Oracle Business Intelligence Applications .....	7-1
7.1.1	Checking Oracle Business Intelligence Applications User Responsibilities .....	7-2
7.1.2	Registering a New User Responsibility in Oracle Business Intelligence.....	7-3
7.1.3	Default Security Settings in Oracle Business Intelligence.....	7-3
7.2	Integrating Data Security with Oracle E-Business Suite .....	7-3
7.2.1	Authorization .....	7-3
7.2.2	Operating Unit-Based Security With Oracle EBS.....	7-4
7.2.2.1	About Operating Unit-Based Security With Oracle EBS .....	7-4
7.2.2.2	Implementation Steps For Operating Unit-Based Security With Oracle EBS.....	7-6
7.2.3	Inventory Organization-Based Security With Oracle EBS.....	7-7
7.2.3.1	About Inventory Organization-Based Security With Oracle EBS .....	7-7
7.2.3.2	Implementation Steps For Inventory Organization-Based Security With Oracle EBS .....	7-8
7.2.4	Company Org-based Security Against Oracle EBS .....	7-9
7.2.4.1	About Company Org-based Security Against Oracle EBS.....	7-9
7.2.4.2	Implementation Steps For Company Org-based Security Against Oracle EBS 11i .....	7-9
7.2.5	Ledger Security Against Oracle EBS.....	7-10
7.2.6	Business Group Org-based Security Against Oracle EBS.....	7-12
7.2.6.1	About Business Group Org-based Security Against Oracle EBS .....	7-13
7.2.6.2	Implementation Steps For Business Group Org-based Security Against Oracle EBS .....	7-13
7.2.7	Employee Based Security with Oracle EBS.....	7-14
7.3	Integrating Data Security with Oracle's PeopleSoft Enterprise Applications.....	7-14
7.3.1	Authorization .....	7-14
7.3.2	About Operating Unit-based Security Against PeopleSoft Financials .....	7-15
7.3.3	About Company Org-based Security Against PeopleSoft Financials or PeopleSoft HR .....	7-16
7.3.4	About Ledger-based Security against Oracle's PeopleSoft Enterprise Applications.....	7-18
7.3.5	About HR Org-based Security Against PeopleSoft HR .....	7-18

7.3.6	About Payables Org-based security Against PeopleSoft Financials .....	7-19
7.3.7	About Receivables Org-based Security Against PeopleSoft Financials.....	7-20
7.3.8	About SET ID-based Security Against PeopleSoft Financials .....	7-21
7.3.9	About Human Resource Personnel Data Analyst Security Against PeopleSoft HR .....	7-22
7.3.10	Employee Based Security Against PeopleSoft Application.....	7-25
7.4	Metadata Object-Level Security in Oracle BI Applications .....	7-25
7.4.1	Metadata Object-Level Security (Repository Groups) .....	7-25
7.4.1.1	Where is Repository Groups Security Configured? .....	7-25
7.4.1.2	Metadata Object-Level (Presentation Services) Security in Oracle BI Applications .....	7-26
7.5	Integrating Data Security with the Siebel CRM Application .....	7-26
7.5.1	Primary Position based Security.....	7-26
7.5.1.1	Introduction.....	7-27
7.5.1.2	Primary Position based Security Group.....	7-27
7.5.1.3	Configuring Oracle BI Repository Table Joins for Position/Employee Based Security.....	7-29
7.5.2	Primary Position Based Security for CRM Vertical applications.....	7-30
7.5.2.1	Oracle Partner Analytics Security Settings .....	7-30
7.5.2.1.1	Partner Manager Role-Based Interactive Dashboards Mapping.....	7-31
7.5.2.1.2	Oracle's Siebel Partner Portal Role-Based Interactive Dashboards Mapping.....	7-32
7.5.2.1.3	PRM Analytics Subject Area Mapping.....	7-33
7.5.2.1.4	PRM Analytics Subject Area Visibility .....	7-34
7.5.2.1.5	PRM Analytics Data-Level Visibility.....	7-35
7.5.2.2	Consumer Sector Analytics Security Settings .....	7-36
7.5.2.3	Communications, Media, and Energy (CME) Analytics Security Settings.....	7-36
7.5.2.4	Financial Services Analytics Security Settings .....	7-37
7.5.2.4.1	Parent and Child Group Behavior .....	7-38
7.5.2.5	Oracle Pharma Sales Analytics and Oracle Pharma Marketing Analytics Security Settings .....	7-39
7.5.2.6	Usage Accelerator Analytics Security Settings.....	7-41
7.5.3	Team Based Visibility.....	7-42
7.5.3.1	Summary .....	7-43
7.5.3.2	Business Requirements Overview .....	7-43
7.5.3.3	Design Details .....	7-43
7.5.3.4	Data Model .....	7-43
7.5.3.5	Fact Security .....	7-45
7.5.3.6	ETL Design .....	7-46
7.5.3.6.1	Available Mappings.....	7-46
7.5.3.6.2	Creating Delete Triggers .....	7-47
7.5.3.6.3	DAC Metadata .....	7-48
7.5.3.7	DAC Metadata .....	7-48
7.5.3.8	Oracle BI Applications Repository Metadata.....	7-48
7.5.4	Embedded Oracle Business Intelligence support for Changing Positions.....	7-49
7.5.5	Business Unit based Security .....	7-51
7.5.6	Primary Owner Security .....	7-51
7.6	Data-Level Security In Oracle BI Applications.....	7-51

7.6.1	Data-Level Security in Oracle Business Intelligence Applications.....	7-52
7.6.1.1	Roadmap of Data Security in Oracle Business Intelligence Applications releases .....	7-52
7.6.2	Implementing Data-Level Security in the Oracle BI Applications Repository.....	7-53
7.6.3	Initialization Blocks Used For Security Oracle BI Applications .....	7-54
7.6.3.1	Data Security groups in Oracle Business Intelligence Applications.....	7-55

## 8 Configuring Common Areas and Dimensions

8.1	Source Independent Configuration Steps .....	8-1
8.1.1	Configuration Required Before A Full Load .....	8-1
8.1.1.1	How to Configure Initial Extract Date .....	8-1
8.1.1.2	How to Configure Global Currencies.....	8-1
8.1.1.3	How to Configure Exchange Rate Types .....	8-2
8.1.1.4	How to Configure Fiscal Calendars.....	8-3
8.1.1.4.1	Overview To Setting Up The Time Dimension.....	8-3
8.1.1.4.2	Setting Up Fiscal Calendar.....	8-4
8.1.1.4.3	Setting Up Fiscal Calendar By Fiscal Week .....	8-4
8.1.1.4.4	Setting Up Fiscal Calendar By Fiscal Month .....	8-4
8.1.1.4.5	Process of Setting Up The Fiscal Calendar .....	8-4
8.1.1.4.6	Reloading the Time Dimension Tables After Your Data Warehouse is Populated.....	8-5
8.1.1.4.7	Notes .....	8-6
8.1.2	Configuration Steps for Controlling Your Data Set for All Source Systems.....	8-6
8.1.2.1	How to Configure Data Source Num IDs .....	8-6
8.2	Oracle EBS-Specific Configuration Steps .....	8-7
8.2.1	Configuration Required Before A Full Load for Oracle EBS.....	8-7
8.2.1.1	Configuration of Product hierarchy (except for GL, HR modules).....	8-8
8.2.1.1.1	How to identify Category Sets from EBS.....	8-8
8.2.1.1.2	To configure product category sets extract from Oracle Applications.....	8-8
8.2.1.1.3	To configure the Category Sets for Product Hierarchy .....	8-9
8.2.1.1.4	To reconfigure the product hierarchy loads.....	8-9
8.2.1.1.5	To configure the Category Sets for Inventory Product Hierarchy .....	8-10
8.2.1.2	Configure the Master Inventory Organization in Product dimension Extract for Oracle 11i adapter (Except for GL & HR Modules).....	8-10
8.2.2	Configuration Steps for Controlling Your Data Set for Oracle EBS.....	8-10
8.2.2.1	Configuring the Country Region and State Region Name.....	8-10
8.2.2.2	Configuring the State Name .....	8-11
8.2.2.3	Configuring the Country Name .....	8-12
8.2.2.4	Configuring the Configuring the Make-Buy Indicator .....	8-13
8.3	PeopleSoft-Specific Configuration Steps .....	8-13
8.3.1	Configuration Required Before A Full Load for PeopleSoft .....	8-13
8.3.2	Configuration Steps for Controlling Your Data Set for PeopleSoft.....	8-13
8.4	Siebel-Specific Configuration Steps.....	8-13
8.4.1	Configuration Required Before A Full Load for Siebel Source Systems .....	8-14
8.4.2	Configuration Steps for Controlling Your Data Set for Siebel Source Systems.....	8-14

## 9 Configuring Oracle's Supply Chain Family of Products

9.1	Overview of Oracle's Supply Chain Analytics Family of Products.....	9-1
9.1.1	Procurement and Spend Analytics Module.....	9-2
9.1.2	Supplier Performance Analytics Module.....	9-2
9.1.3	Inventory Analytics Module.....	9-3
9.2	Configuration Required Before A Full Load for Supply Chain Analytics.....	9-4
9.2.1	Configuration Steps for Supply Chain For All Source Systems.....	9-4
9.2.1.1	How to Deploy Stored Procedures.....	9-4
9.2.2	Configuration Steps for Supply Chain For Oracle EBS.....	9-4
9.2.2.1	How to Configure the Bill of Materials Explosion for Oracle Version 11.5.9 and above.....	9-5
9.2.2.2	How to Configure the DAC parameter for Purchase Cycle Line.....	9-5
9.2.2.3	How to Configure Quantity Types for Product Transactions.....	9-6
9.2.2.4	Domain Values and CSV Worksheet Files for Supply Chain Analytics.....	9-7
9.2.2.4.1	List of Domain Values and CSV Worksheet Files for Supply Chain Analytics.....	9-7
9.2.2.4.2	To configure domainValues_Movement_Types_ora11i.csv.....	9-8
9.2.2.4.3	To configure domainValues_PaymentType_ora12.csv.....	9-8
9.2.2.4.4	To configure domainValues_Status_Purch_Approve_ora11i.csv.....	9-9
9.2.2.4.5	To configure domainValues_Status_Purch_Cycle_ora11i.csv.....	9-9
9.2.2.4.6	To configure domainValues_Xact_Types_Purch_Orders_ora11i.csv.....	9-9
9.2.2.4.7	To configure domainValues_Xact_Types_Purch_Requisitions_ora11i.csv.....	9-10
9.2.2.4.8	To configure domainValues_Xact_Types_PO_Line_Type_ora11i.csv.....	9-10
9.2.2.4.9	To configure domainValues_Xact_Types_PO_Line_Type_CODE1_ora11i.csv.....	9-11
9.2.2.4.10	To configure domainValues_ShipmentType_ora12.csv.....	9-11
9.2.3	Configuration Steps for Supply Chain For PeopleSoft.....	9-11
9.2.4	Configuration Steps for Supply Chain For Universal.....	9-12
9.2.5	Configuration Steps for Controlling Your Data Set.....	9-12
9.2.5.1	Configuration Steps for Supply Chain For All Source Systems.....	9-12
9.2.5.1.1	About Configuring the Inventory Monthly Balance Table.....	9-12
9.2.5.1.2	How to Configure the Inventory Monthly Balance Table.....	9-13
9.2.5.1.3	About Configuring the Product Transaction Aggregate Table.....	9-14
9.2.5.1.4	How to Configure the Product Transaction Aggregate Table.....	9-15
9.2.5.1.5	About Configuring the Purchase Receipts Aggregate Table.....	9-16
9.2.5.1.6	How to Configure the Purchase Receipts Aggregate Table.....	9-17
9.2.5.1.7	About Configuring the Purchase Cycle Lines Aggregate Table.....	9-17
9.2.5.1.8	How to Configure the Purchase Cycle Lines Aggregate Table.....	9-18
9.2.5.2	Configuration Steps for Supply Chain For Oracle EBS.....	9-18
9.2.5.2.1	How to Configure the Bill of Materials Explosion Options.....	9-18
9.2.5.2.2	How to Configure the Left Bound and Right Bound Calculation Option.....	9-23
9.2.5.2.3	How to Extract Particular Purchase Order Records.....	9-23
9.2.5.3	Configuration Steps for Supply Chain For PeopleSoft.....	9-24
9.2.5.4	Configuration Steps for Oracle Supply Chain For Universal.....	9-24
9.2.5.4.1	About Configuring Supply Chain Analytics for Universal Source.....	9-24
9.2.5.4.2	How to Configure the Preferred Merchant Flag.....	9-25
9.2.5.4.3	How to Configure the Customer Billable Indicator.....	9-26

9.2.5.4.4	How to Configure the Receipts Indicator .....	9-26
9.2.5.4.5	How to Configure Expense Payment Types .....	9-26
9.2.5.4.6	How to Configure the Default Expense Distribution Percentage .....	9-27
9.2.5.4.7	How to Configure Lookup Dates for Currency Conversion .....	9-27

## 10 Configuring Oracle Financial Analytics

10.1	Overview of Oracle Financial Analytics .....	10-1
10.2	Configuration Required Before A Full Load for Financial Analytics.....	10-2
10.2.1	Configuration Steps for Financial Analytics For All Source Systems .....	10-2
10.2.2	Configuration Steps for Financial Analytics For Oracle EBS .....	10-2
10.2.2.1	About Configuring Domain Values and CSV Worksheet Files for Oracle Financial Analytics .....	10-3
10.2.2.2	How to Map Oracle General Ledger Natural Accounts to Group Account Numbers .....	10-3
10.2.2.3	How to do Group Account correction for Oracle Application .....	10-6
10.2.2.4	How to configure domainValues_Xact_Types_DocTypes_ora12.csv for Oracle General Ledger and Profitability Analytics.....	10-7
10.2.2.5	How to configure domainValues_Xact_Types_DocTypes_ora11i.csv for Oracle General Ledger and Profitability Analytics.....	10-8
10.2.3	Configuration Steps for Financial Analytics For PeopleSoft .....	10-9
10.2.3.1	About Configuring Domain Values and CSV Worksheet Files for Oracle Financial Analytics .....	10-9
10.2.3.2	How to configure domainValues_Xact_Types_DocTypes_psft.csv for PeopleSoft Receivables .....	10-9
10.2.3.3	How to insert beginning GL Balances into Oracle BI Applications .....	10-10
10.2.4	Configuration Steps for Financial Analytics For Universal.....	10-12
10.2.5	Configuration Steps for Controlling Your Data Set.....	10-12
10.2.5.1	Configuration Steps for Financial Analytics For All Source Systems.....	10-12
10.2.5.1.1	How to Configure the AP/AR Aging Tables.....	10-12
10.2.5.2	Configuration Steps for Financial Analytics For Oracle EBS.....	10-13
10.2.5.2.1	How to Configure How GL Balances Is Populated in Oracle EBS.....	10-13
10.2.5.2.2	How to Configure Oracle Profitability Analytics Transaction Extracts...	10-14
10.2.5.2.3	How to Configure Cost Of Goods Extract (Oracle EBS 11i-specific) .....	10-14
10.2.5.2.4	How to Configure the General Ledger Account Hierarchies .....	10-15
10.2.5.2.5	Configuring General Ledger Account Hierarchies Using General Ledger Accounting Flexfield value sets definitions .....	10-16
10.2.5.2.6	How to Configure General Ledger Account Hierarchies Using Financial Statement Generator (FSG) Report Definition .....	10-20
10.2.5.2.7	How to Configure AP Balance ID for Oracle Payables Analytics.....	10-24
10.2.5.2.8	How to Configure AR Balance ID for Oracle Receivables Analytics and Oracle General Ledger and Profitability Analytics.....	10-25
10.2.5.2.9	How to Configure the AR Adjustments Extract for Oracle Receivables Analytics .....	10-25
10.2.5.2.10	How to Configure the AR Schedules Extract .....	10-26
10.2.5.2.11	How to Configure the AR Cash Receipt Application Extract for Oracle Receivables Analytics .....	10-26
10.2.5.2.12	How to Configure the AR Credit-Memo Application Extract for Oracle Receivables Analytics .....	10-27

10.2.5.2.13	How to Specify the Ledger or Set of Books for which GL Data is Extracted .....	10-27
10.2.5.2.14	How to Setup Drill Down in Oracle BI Answers from General Ledger to Subledger .....	10-29
10.2.5.3	Configuration Steps for Financial Analytics For PeopleSoft.....	10-30
10.2.5.3.1	About PeopleSoft Trees in Oracle Financial Analytics .....	10-30
10.2.5.3.2	GL Account Dimension, Chartfields and Hierarchy for PeopleSoft.....	10-31
10.2.5.3.3	Group Account Configuration .....	10-33
10.2.5.3.4	GL ChartField Hierarchy and Internal Organization Hierarchy Customization Process.....	10-34
10.2.5.3.5	General Ledger Chartfield Hierarchy Extraction Customization .....	10-37
10.2.5.3.6	Internal Organization Hierarchy Extraction Customization .....	10-40
10.2.5.4	Configuration Steps for Financial Analytics Chain For Universal.....	10-41
10.2.5.4.1	About the Customer Costs Lines and Product Costs Lines Tables for Oracle's Profitability Analytics.....	10-42
10.2.5.4.2	How to Configure the Customer Costs Lines and Product Costs Lines Tables for Oracle's Profitability Analytics.....	10-42

## 11 Configuring Oracle Order Management and Fulfillment Analytics

11.1	Overview of Oracle Order Management and Fulfillment Analytics.....	11-1
11.2	Configuration Required Before A Full Load for Oracle Order Management and Fulfillment Analytics .....	11-2
11.2.1	Configuration Steps for Oracle Order Management and Fulfillment Analytics For All Source Systems .....	11-2
11.2.2	Configuration Steps for Oracle Order Management and Fulfillment Analytics For Oracle EBS.....	11-2
11.2.2.1	About Configuring Domain Values and CSV Worksheet Files for Oracle Order Management and Fulfillment Analytics .....	11-2
11.2.2.2	How to Configure Invoice Type Domain Values .....	11-4
11.2.2.3	How to Configure Pick Types Domain Values .....	11-4
11.2.2.4	How to Configure Order Types Domain Values .....	11-5
11.2.2.5	How to Configure Pick Status Domain Values .....	11-5
11.2.2.6	How to Configure Invoice Status Domain Values.....	11-6
11.2.2.7	How to Configure Order Overall Status Domain Values.....	11-6
11.2.2.8	How to Configure Pay Method Domain Values.....	11-7
11.2.3	Configuration Steps for Oracle Order Management and Fulfillment Analytics For PeopleSoft .....	11-7
11.2.4	Configuration Steps for Oracle Order Management and Fulfillment Analytics For Universal .....	11-8
11.2.5	Configuration Steps for Controlling Your Data Set.....	11-8
11.2.5.1	Configuration Steps for Oracle Order Management and Fulfillment Analytics For All Source Systems .....	11-8
11.2.5.1.1	Tracking Multiple Attribute Changes in Bookings .....	11-8
11.2.5.1.2	Process of Aggregating Oracle Order Management and Fulfillment Analytics Tables .....	11-9
11.2.5.1.3	About Tracking Multiple Products for Oracle Order Management and Fulfillment Analytics .....	11-12
11.2.5.1.4	Adding Dates to the Order Cycle Time Table for Post-Load Processing.	11-12

11.2.5.1.5	About Configuring the Backlog Period Date for Oracle Order Management and Fulfillment Analytics.....	11-13
11.2.5.1.6	Example of How Backlog Data Is Stored in the Backlog History Table...	11-13
11.2.5.1.7	About Configuring the Customer Status History Fact for Post-Load Processing In Oracle Order Management and Fulfillment Analytics .....	11-15
11.2.5.1.8	Configuring the Customer Status History Fact table.....	11-15
11.2.5.1.9	How to Configure the Customer Status History Fact table .....	11-16
11.2.5.2	Configuration Steps for Oracle Order Management and Fulfillment Analytics For Oracle EBS .....	11-17
11.2.5.2.1	How to Configure Sales Order Lines Data Storage.....	11-17
11.2.5.2.2	About the Handling of Booked and Nonbooked Orders in the Order Lines and Bookings Table.....	11-17
11.2.5.2.3	How to Configure Sales Schedule Lines Data Storage .....	11-19
11.2.5.2.4	About the Handling of Booked and Nonbooked Orders in the Sales Schedule Lines Table.....	11-19
11.2.5.2.5	About Loading Bookings at the Schedule Line Level .....	11-19
11.2.5.2.6	How to Configure Early and Late Tolerances for Shipments.....	11-20
11.2.5.2.7	How to Configure Sales Invoice Lines Data Storage .....	11-21
11.2.5.2.8	How to Configure the Sales Invoice Extract.....	11-21
11.2.5.2.9	How to Configure the Order Lines Execution Plan and OTB ETL .....	11-22
11.2.5.2.10	How to Configure Supply Chain Analytics for Oracle Order Management and Fulfillment Analytics .....	11-23
11.2.5.2.11	How to Configure Oracle Financial Analytics for Oracle Order Management and Fulfillment Analytics.....	11-23
11.2.5.2.12	About Tracking Attribute Changes in Bookings .....	11-23
11.2.5.2.13	About Viewing the Data Warehouse Changes by Salesperson ID .....	11-24
11.2.5.2.14	How to Configure Different Types of Backlog Calculations .....	11-25
11.2.5.2.15	Adding Closed Orders to Backlog Calculations.....	11-25
11.2.5.3	Configuration Steps for Oracle Order Management and Fulfillment Analytics For PeopleSoft .....	11-26
11.2.5.4	Configuration Steps for Oracle Order Management and Fulfillment Analytics For Universal.....	11-26

## 12 Configuring Oracle HR Analytics

12.1	Overview of Oracle HR Analytics .....	12-1
12.2	Configuration Required Before A Full Load for Oracle HR Analytics .....	12-2
12.2.1	Configuration Steps for Oracle HR Analytics For All Source Systems .....	12-2
12.2.2	Configuration Steps for Oracle HR Analytics For Oracle EBS.....	12-2
12.2.2.1	About Domain Values and CSV Worksheet Files for Oracle HR Analytics .....	12-3
12.2.2.2	How to Configure the domainValues_Employee_Ethnic_Group_Code_ora11i.csv.....	12-8
12.2.2.3	How to Configure the domainValues_Employee_Sex_MF_ora11i.csv .....	12-8
12.2.2.4	How to Configure the domainValues_Employee_Veteran_Status_Code_ora11i.csv .....	12-9
12.2.2.5	How to Configure the domainValues_Employment_Cat_ora11i.csv.....	12-9
12.2.2.6	How to Configure the domainValues_Employment_Exempt_Flg_ora11i.csv .....	12-9
12.2.2.7	How to Configure the domainValues_Employment_Full_Time_Flg_ora11i.csv .....	12-10

12.2.2.8	How to Configure the domainValues_Employment_Status_ora11i.csv .....	12-10
12.2.2.9	How to Configure the domainValues_EventTypes_ora11i.csv .....	12-10
12.2.2.10	How to Configure the domainValues_HRPosition_Active_Pos_Flg_ora11i.csv .....	12-11
12.2.2.11	How to Configure the domainValues_HRPosition_Exempt_Flg_ora11i.csv ..	12-11
12.2.2.12	How to Configure the domainValues_Job_Eeo_Cat_Code_ora11i.csv .....	12-12
12.2.2.13	How to Configure the domainValues_Job_Flsa_Stat_Code_ora11i.csv .....	12-12
12.2.2.14	How to Configure the domainValues_Pay_Type_Grp_Code_ora11i.csv .....	12-12
12.2.2.15	How to Configure the domainValues_Pay_Type_Flg_ora11i.csv .....	12-13
12.2.2.16	How to Configure Address Types for HR Profile .....	12-13
12.2.2.17	How to Configure Phone Types for HR Profile .....	12-14
12.2.2.18	How to Configure Education Degree Codes for Employee Dimension.....	12-15
12.2.2.19	About Configuring Key Flexfields .....	12-15
12.2.2.20	How to Configure the Key Flexfields for the Job Dimension .....	12-16
12.2.2.21	How to Configure the Key Flexfields for the HR Position Dimension.....	12-16
12.2.2.22	How to Configure the Key Flexfields for the Pay Grade Dimension .....	12-17
12.2.2.23	How to Configure multi-segmented Flexfields .....	12-17
12.2.2.24	How to Configure Flags for the Pay Type Dimension.....	12-18
12.2.2.25	How to Configure Classification Names for Payroll.....	12-19
12.2.3	Configuration Steps for Oracle HR Analytics For PeopleSoft .....	12-20
12.2.3.1	Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft 8.8 HCM .....	12-21
12.2.3.2	How to configure the domainValues_Employee_Ethnic_Group_Code_psft.csv .....	12-23
12.2.3.3	How to configure the domainValues_Employee_Sex_MF_psft.csv .....	12-24
12.2.3.4	How to configure the domainValues_Employee_Veteran_Status_Code_psft.csv .....	12-24
12.2.3.5	How to configure the domainValues_Employment_Category_psft.csv .....	12-24
12.2.3.6	How to configure the domainValues_Employment_Exempt_Flg_psft.csv ....	12-25
12.2.3.7	How to configure the domainValues_Employment_Full_Time_Flg_psft.csv	12-25
12.2.3.8	How to configure the domainValues_Employment_Status_psft.csv .....	12-25
12.2.3.9	How to configure the domainValues_Emp_EventType_Grp_Code_psft.csv	12-26
12.2.3.10	How to configure the domainValues_Emp_EventType_SubG_Code_psft.csv .....	12-26
12.2.3.11	How to configure the domainValues_HRPosition_Active_Pos_Flg_psft.csv	12-27
12.2.3.12	How to configure the domainValues_HRPosition_Exempt_Flg_psft.csv .....	12-27
12.2.3.13	How to configure the domainValues_Job_Eeo_Cat_Code_psft.csv .....	12-27
12.2.3.14	How to configure the domainValues_Job_Flsa_Stat_Code_psft.csv .....	12-28
12.2.3.15	How to configure the domainValues_Pay_Type_Grp_Code_psft.csv .....	12-28
12.2.3.16	How to configure the domainValues_Emp_EventType_EventReason_Code_psft.csv .....	12-29
12.2.3.17	How to configure the domainValues_Emp_EventType_EventClass_Code_psft.csv .....	12-29
12.2.3.18	How to configure the file_pension_comp_flg.csv .....	12-30
12.2.3.19	How to Configure Compensation Flag and Pension Compensation Flag for Pay Type Dimension.....	12-30
12.2.3.20	How to Configure the Performance Review Type ETL runtime parameter...	12-30
12.2.4	Configuration Steps for Oracle HR Analytics For Universal .....	12-31
12.2.5	Configuration Steps for Controlling Your Data Set.....	12-31

12.2.5.1	Configuration Steps for Oracle HR Analytics For All Source Systems .....	12-31
12.2.5.1.1	How to Aggregate the Payroll Table for Oracle HR Analytics.....	12-31
12.2.5.1.2	About Configuring the Employee Snapshot Fact tables.....	12-32
12.2.5.1.3	How to control the rolling period for storing Employee Daily Snapshot data .....	12-34
12.2.5.1.4	How to control the granularity of the Employee Monthly Snapshot data .....	12-34
12.2.5.2	Configuration Steps for Oracle HR Analytics For Oracle EBS.....	12-34
12.2.5.3	Configuration Steps for Oracle HR Analytics For PeopleSoft .....	12-34
12.2.5.3.1	How to Tune Performance for the Payroll table for PeopleSoft 8.8 HCM HR Analytics .....	12-34
12.2.5.4	Configuration Steps for Oracle HR Analytics For Universal .....	12-35

### 13 Configuring Oracle Sales Analytics

13.1	Overview .....	13-1
13.2	Configuration Required Before A Full Load.....	13-1
13.2.1	How to Deploy Stored Procedures .....	13-1
13.2.2	Configuration Steps for Controlling Your Data Set .....	13-2
13.2.2.1	How to configure the snapshot frequency in Pipeline fact .....	13-2

### 14 Configuring Oracle Contact Center Telephony Analytics

14.1	Overview of Oracle Contact Center Telephony Analytics .....	14-1
14.2	Configuration Required Before A Full Load for Contact Center Telephony Analytics	14-1
14.2.1	About the Dimension Key Resolution Process for Universal Source .....	14-2
14.2.2	How to Load the Dimension Table .....	14-2
14.2.3	How to Load the Fact Table .....	14-3
14.2.4	Setting Up The DATASOURCE_NUM_ID Parameter For The Universal Adapter	14-3
14.2.5	About Configuring the Event Type Columns .....	14-3
14.2.5.1	The Activity Type Column.....	14-3
14.2.5.2	Call Type (or Contact Type) Column .....	14-4
14.2.5.3	Call Event Type Column .....	14-4
14.2.5.4	After-Call Work Activity (ACW) Type Column .....	14-5
14.2.6	About Configuring The Contact Reason Column.....	14-5
14.2.7	About Configuring Contact Status Type Column .....	14-6
14.2.8	Setting Up the Representative Activities Table.....	14-6
14.2.9	Setting Up the Contact Center Performance Table.....	14-7
14.2.10	Setting Up the Benchmarks and Targets Table .....	14-8
14.2.11	Logical Delete of Records From Fact Tables .....	14-10
14.2.12	How to Configure Dates and Times .....	14-11
14.2.13	Configuring Flags for Oracle Contact Center Telephony Analytics .....	14-12
14.2.14	How to Exclude Representative Data for Post-Load Processing.....	14-13
14.2.15	How to Configure Alternate Load Plan to Enable Direct Load to the Entry-Level Rep Activity Aggregates.....	14-13
14.2.16	How to Configure Alternate Load Plan to Enable Direct Load to the Entry-Level Contact Performance Aggregates.....	14-14
14.2.17	How to Set Up the Integration_Id of the Contact Center Telephony Facts in the Warehouse .....	14-15

14.2.18	Setting Up the IVR Menu and IVR Navigation Profile Tables.....	14-16
14.2.18.1	Determining Navigation Profile and Transfer Flag for a Call .....	14-16
14.2.18.2	Slowly Changing Dimension Behavior of IVR Menu and IVR Navigation Profile Dimensions .....	14-17
14.2.19	How to View Duration Metrics in hh:mm:ss Format in Oracle BI Answers .....	14-17
14.2.20	Configuration Steps for Controlling Your Data Set.....	14-18

## 15 Configuring the Oracle BI Applications Repository

15.1	Modifying the Oracle BI Repository for Siebel version 6.3 and version 7.5 applications .....	15-1
15.2	Configuring the Oracle BI Repository Connections .....	15-2
15.2.1	About The Predefined Connection Pools In The Oracle Business Analytics Warehouse .....	15-2
15.2.1.1	How to Configure Dynamic Data Source Names.....	15-3
15.2.2	How to Configure Oracle BI Applications Repository Connections.....	15-4
15.2.3	How to Configure Oracle BI Applications Repository Variables .....	15-5
15.3	Setting up Date Specific Metrics .....	15-5
15.4	Setting Up Additional Time Series Metrics for Oracle Business Analytics Warehouse	15-6
15.5	Setting Up Additional Dimension Tables for Oracle Business Analytics Warehouse...	15-7
15.6	About the Period Ago Keys for Oracle Business Analytics Warehouse .....	15-7
15.7	About Oracle BI Time Repository Variables .....	15-7
15.8	About User Authentication .....	15-9
15.9	About the Security or Visibility Configuration .....	15-9
15.10	About the Group Variable .....	15-12
15.11	About Configuring Usage Tracking for Oracle Business Analytics Warehouse .....	15-12
15.12	About the Incremental Deployment of the Oracle BI Applications Repository .....	15-13

## A Installing the Usage Accelerator Application

A.1	What is Usage Accelerator? .....	A-1
A.1.1	About Configuring Usage Accelerator .....	A-1
A.1.1.1	Tracking of Targeted Users .....	A-1
A.1.1.2	Other Required Responsibilities.....	A-1
A.1.2	About Deployment Date.....	A-2
A.1.3	About Metric Definitions and Variables .....	A-2
A.1.3.1	User Adoption Metrics .....	A-2
A.1.3.2	Opportunity Updates Metrics.....	A-3
A.1.3.3	Information Completeness Metrics.....	A-3
A.1.3.4	Coverage Metrics .....	A-4
A.2	About Installing the Usage Accelerator.....	A-4
A.3	Usage Accelerator Application Integration with Siebel 7.5.3 CRM Application.....	A-5
A.4	Usage Accelerator-Analytics Adapter Files .....	A-7

## B Localizing Oracle Business Intelligence Deployments

B.1	Process of Maintaining Translation Tables for Oracle BI.....	B-1
B.1.1	Upgrading Oracle Business Intelligence Seed Data for Non-English Locales.....	B-2
B.1.1.1	Importing Locale Seed Data Into The Translation Table (W_LOCALIZED _STRING_G).....	B-2

B.1.2	Externalizing Customer Metadata Strings .....	B-3
B.1.3	Adding Custom Translations to the W_LOCALIZED_STRING_G Table .....	B-4
B.1.3.1	Adding String Translations for Analytics Metadata .....	B-4
B.2	About Translating Presentation Services Strings .....	B-5
B.3	Changing the Default Currency in Analytics Applications.....	B-5

## **C Integrating Interactive Dashboards and Operational Applications Data**

C.1	Importing Oracle's Siebel Industry Applications Seed Data .....	C-1
C.2	Completing the Initialization in the Siebel Operational Application.....	C-2
C.2.1	Updating the Siebel Operational Application .....	C-2
C.2.2	Reapplying Customized Style Sheets .....	C-3
C.2.3	How to Configure Oracle Business Intelligence with Oracle's Siebel Web Extension (SWE).....	C-4
C.2.4	Creating a Virtual IP Address for the SWE and Oracle BI Presentation Services.....	C-5
C.2.5	Testing the Virtual IP Configuration for Oracle Business Intelligence and the SWE.....	C-5
C.2.6	How to Configure Oracle BI Action Links .....	C-6
C.2.7	Testing Action Links .....	C-6
C.2.8	Process of Integrating Analytics and Oracle's Siebel Operational Applications.....	C-6
C.2.9	Accessing Optional Analytics Applications .....	C-7
C.2.10	Configuring the Browser Settings for Operational Applications .....	C-7
C.2.11	Process of Customizing Oracle BI Application Home Page and Dashboards.....	C-8
C.2.12	Customizing Oracle BI Content on the Siebel Operational Application Home Page.....	C-8
C.2.13	Determining the Oracle BI Report Path Argument .....	C-10
C.2.14	Adding Views for Custom Oracle BI Interactive Dashboards .....	C-11
C.2.15	Verifying NQHOST Setup for Pharma Disconnected Analytics Client .....	C-11
C.2.16	Integrating the Pharma Application Home Page with the Pharma Disconnected Analytics Client.....	C-12
C.2.17	Importing Pharma Application Target Lists to the Pharma Disconnected Analytics Client.....	C-13
C.3	Viewing Information About Preconfigured Dashboards .....	C-14
C.3.1	Exposing Dashboards and Requests.....	C-14
C.3.2	Locating Dashboard and Page Names.....	C-14

## **D Configuring Metadata for Analytics Applications**

D.1	What Are Oracle's Siebel Operational Applications? .....	D-1
D.2	Updating Financial Analytics Logical Table Sources .....	D-2
D.3	Externalizing Financial Services Metadata Translation Strings .....	D-3
D.4	Disabling the Financial Analytics Logical Table Sources.....	D-3
D.5	Developing and Deploying Predictive Scores .....	D-4
D.6	Analytics Metadata Requirements for Oracle's Siebel Industry Applications.....	D-5
D.6.1	Oracle's Siebel Consumer Sector Dashboards and Pages .....	D-5
D.6.2	Oracle's Siebel Consumer Sector Data Requirements .....	D-7
D.6.3	Oracle's CME family of products (Oracle Communications, Media and Energy Sales Analytics) .....	D-7
D.6.4	Oracle's Siebel CME Dashboards and Pages .....	D-9

D.6.5	Oracle's Siebel CME Data Requirements.....	D-10
D.6.6	Oracle Pharma Sales Analytics Dimensions .....	D-10
D.6.6.1	Positions Dimension.....	D-10
D.6.6.2	Alignments Dimension.....	D-10
D.6.6.3	Products Dimension.....	D-11
D.6.6.4	Product Costs Dimension.....	D-11
D.6.7	Dimensions Specific to Subject Areas in Oracle Pharma Sales Analytics and Oracle Pharma Marketing Analytics .....	D-12
D.6.7.1	Pharma Sales Effectiveness .....	D-12
D.6.7.2	Pharma Product Categories .....	D-12
D.6.7.3	Pharma Promotional Effectiveness .....	D-12
D.6.7.4	Pharma Medical Education Effectiveness.....	D-13
D.6.7.5	Pharma Objectives Achievement` .....	D-13

## **E Oracle Business Analytics Warehouse for Life Sciences Data Considerations**

E.1	What's New for Pharma Analytics in Version 7.9.....	E-1
E.2	Oracle Business Analytics Warehouse for Life Sciences Overview.....	E-1
E.3	Importing Syndicated Data into Oracle Business Analytics Warehouse.....	E-2
E.3.1	Syndicated Loading Definitions .....	E-3
E.3.2	Data Types Supported in the Oracle Business Analytics Warehouse.....	E-3
E.3.3	Loading Syndicated Data into the Oracle Business Analytics Warehouse .....	E-4
E.3.3.1	Loading From a Flat File Source Using Syndicated Data Staging Tables and Cross-Referencing Tables .....	E-4
E.3.3.1.1	Flat File Data Population Rules for Populating TRx Data.....	E-6
E.3.3.2	Loading From a Flat File Source Using Syndicated Market Staging Tables.....	E-9
E.4	Syndicated Data Flat File Formats.....	E-10
E.4.1	Syndicated Data Flat File for Direct Sales Data.....	E-10
E.4.2	Syndicated Data Flat File for Indirect Sales Data.....	E-11
E.4.3	Syndicated Data Flat File for Prescription Data .....	E-13
E.4.4	Syndicated Data Flat File for Plan Level Rx Data .....	E-14
E.4.5	Syndicated Data Flat File for Physician Plan Level Rx Data .....	E-16
E.4.6	Syndicated Data Flat File for Weekly Early View Data .....	E-17
E.4.7	Syndicated Market Flat File for Indirect Sales Data .....	E-19
E.4.8	Syndicated Market Flat File for Prescription Data.....	E-20
E.4.9	Syndicated Market Flat File for Plan Level Rx Data.....	E-22
E.4.10	Syndicated Market Flat File for Physician Plan Level Rx Data.....	E-23
E.4.11	Syndicated Market Flat File for Weekly Early View Data .....	E-25
E.4.12	Flat File Cross-Reference Data Population Rules.....	E-26
E.5	Life Sciences Data Loading Issues with Oracle Business Analytics Warehouse .....	E-30
E.5.1	Known Issues with the Syndicated Data Fact Tables.....	E-30
E.6	Incremental Updates in the Oracle Business Analytics Warehouse LS Dimension Tables .....	E-31
E.6.1	W_ALIGNMT_DH .....	E-31
E.6.2	W_ALIGNVER_DH.....	E-31

## **F Pharma Disconnected Analytics Administration - Supplemental Information**

F.1	About Pharma Disconnected Analytics.....	F-1
-----	--	-----

F.2	Sourcing Reports for Oracle's Pharma Disconnected Analytics .....	F-2
-----	---	-----

## **G Using Oracle Business Analytics Warehouse Exception Reports**

G.1	Understanding Oracle Business Analytics Warehouse Exceptions.....	G-1
G.1.1	List of Values Exceptions.....	G-1
G.1.2	Cost List Exceptions .....	G-2
G.1.3	Products Without a Cost List .....	G-3
G.1.4	Exchange Rate Exceptions .....	G-3
G.1.5	Invalid Hierarchy Exceptions .....	G-3
G.1.6	Circular Hierarchy Exceptions.....	G-3
G.2	Executing Oracle Business Analytics Warehouse Exception Reports.....	G-4
G.2.1	Cleansing Data .....	G-4
G.2.2	Using the List of Values View.....	G-5
G.2.3	Using the Exchange Rates View .....	G-5
G.2.4	Using the Cost List View .....	G-6
G.2.5	Using the ETL History View .....	G-6
G.2.6	Additional Exceptions.....	G-6

## **H Troubleshooting Oracle Business Intelligence Applications**

H.1	Oracle Business Analytics Warehouse Installation and Initial Configuration Issues.....	H-2
H.2	Informatica and Loading Issues.....	H-2
H.3	Unicode Issues with Informatica and the Oracle Business Analytics Warehouse .....	H-6
H.4	UNIX Installation Issues with the Oracle Business Analytics Warehouse.....	H-8
H.5	Oracle Business Analytics Warehouse Upgrade Issues .....	H-10
H.6	Oracle BI Interactive Dashboard Issues.....	H-11
H.7	ETL Issues .....	H-12

## **Index**



---

---

# Preface

Oracle Business Intelligence Applications consists of components that were formerly available from Siebel Systems as Siebel Business Analytics Applications (both CRM and Enterprise) with a number of significant enhancements.

The Oracle Business Intelligence Applications Installation and Configuration Guide contains instructions on installing, configuring, and customizing Oracle Business Intelligence Applications. For a high level road map of the steps required to install, configure, and customize Oracle BI Applications, see [Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle BI Applications"](#).

Oracle recommends reading the *Oracle Business Intelligence Applications Release Notes* before installing, using, or upgrading Oracle Business Intelligence Applications. The *Oracle Business Intelligence Applications Release Notes* are available:

- On the Oracle Business Intelligence Applications CD-ROM.
- On the Oracle Technology Network at [http://www.oracle.com/technology/documentation/bi\\_apps.html](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 Business Intelligence Applications.

## Documentation Accessibility

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

### Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an

otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

### **Accessibility of Links to External Web Sites in Documentation**

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

### **TTY Access to Oracle Support Services**

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

## **Related Documents**

For more information, see the following documents in the Oracle Business Intelligence Applications Version 7.9.4 documentation set (available at [http://www.oracle.com/technology/documentation/bi\\_apps.html](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 Data Warehouse Administration Console Guide*
- *Oracle Business Intelligence Applications Upgrade Guide*

## **Conventions**

The following text conventions are used in this document:

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

# Part I

---

## Getting Started

Part I helps you get started with Oracle BI Applications, and contains the following chapters:

- [Chapter 1, "What's New in This Version"](#)
- [Chapter 2, "Overview of Oracle Business Intelligence Applications"](#)
- [Chapter 3, "Pre-installation and Predeployment Requirements for Oracle BI Applications"](#)

---

**Note:** For a high level road map for installation, configuration, and customization steps for Oracle BI Applications, see [Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle BI Applications"](#).

---



---

---

## What's New in This Version

This section summarizes the adapters that are supported in this version of Oracle Business Intelligence Applications.

### 1.1 What's New In Oracle Business Intelligence Applications Version 7.9.4?

The following list shows the main changes in Oracle Business Intelligence Applications Version 7.9.4:

- Adapters:
  - New Adapter for Oracle E-Business Suite R12.
  - Enhanced Universal Adapter to support all available Analytics Applications.
- Dimensions:
  - Ledger Dimension  
This dimension represents Ledger or Set of Books details and can be used to analyze many relevant measures.
  - Company Dimension (Updated)  
With the introduction of the Ledger dimension to capture Set of Books related detail, the Company dimension represents Legal Entity. (In previous releases, the Company dimension represented Set of Books detail.)
- Documentation. The Oracle Business Applications Installation and Configuration Guide has the following improvements:
  - A high level road map for installation, configuration, and customization steps (for more information, see [Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle BI Applications"](#)).
  - A new streamlined structure, based on the following new parts:
    - \* Part 1 - Getting Started
    - \* Part 2 - Pre-Installation and Installation Steps
    - \* Part 3 - Customization and Security in Oracle BI Applications
    - \* Part 4 - Configuring Your Analytical Applications
    - \* Part 5 - Appendices
  - The content in each chapter in Part 4 - Configuring Your Analytical Applications has been categorized as required before a full data load or as

additional configuration for controlling your data set, and is further broken down into source system.

- Languages:
  - 11 new languages (for a complete list of new languages, see *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*)
- Order Management and Fulfillment:
  - Support for COGS (Cost of Goods Sold) and Margin Analysis of ATO (Assembled To Order) Orders.
  - Support for COGS (Cost of Goods Sold) and Margin Analysis of PTO (Packaged To Order) Orders.

---

---

# Overview of Oracle Business Intelligence Applications

This chapter provides an overview of the Oracle Business Analytics Warehouse. It includes the following topics:

- [Section 2.1, "What is Oracle Business Intelligence Applications?"](#)
- [Section 2.2, "Oracle Business Analytics Warehouse Overview"](#)
- [Section 2.3, "Oracle Business Analytics Warehouse Architecture"](#)
- [Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle BI Applications"](#)

## 2.1 What is Oracle Business Intelligence Applications?

Oracle Business Intelligence Applications is a prebuilt business intelligence solution.

---

---

**Note:** Before you can install Oracle Business Intelligence Applications, you must install an Oracle Business Intelligence infrastructure. For more information about Oracle Business Intelligence Enterprise Edition Infrastructure Requirements, see *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*.

---

---

Oracle Business Intelligence Applications support Oracle sources such as Oracle E-Business Suite Applications, Oracle's Siebel Applications, Oracle's PeopleSoft Applications, and non-Oracle sources such as SAP Applications. If you already own one of the above applications, you can purchase Oracle Business Intelligence Infrastructure and Oracle Business Intelligence Applications to work with the application.

Oracle Business Intelligence Applications also provides complete support for enterprise data, including financial, supply chain, and workforce sources. These enterprise applications typically source from both Oracle data sources (such as E-Business Suite and PeopleSoft) and non-Oracle data sources (such as SAP).

Oracle Business Intelligence Applications consist of the components shown in the table below.

**Table 2–1 Oracle Business Intelligence Applications Components**

Component	Description
DAC metadata repository files	This content includes repository objects such as tables, subject areas, execution plans, and tasks, and is contained in XML files.
Embedded Informatica ETL Tool	This is a third-party application that performs the extract, transform, and load operations for the Data Warehouse.
Prebuilt Informatica content	This content includes Extract-Transform-Load (ETL) repository objects, such as mappings, sessions, and workflows, and is contained in the Informatica repository file (Oracle_BI_DW_Base.rep).
Prebuilt metadata content	This metadata content is contained in the Oracle Business Intelligence Applications repository file (OracleBIAnalyticsApps.rpd).
Prebuilt reports and dashboard content	This content is contained in the Oracle BI Presentation Services Catalog.
Oracle Business Analytics Warehouse	The prebuilt data warehouse that holds data extracted, transformed, and loaded from the transactional database.

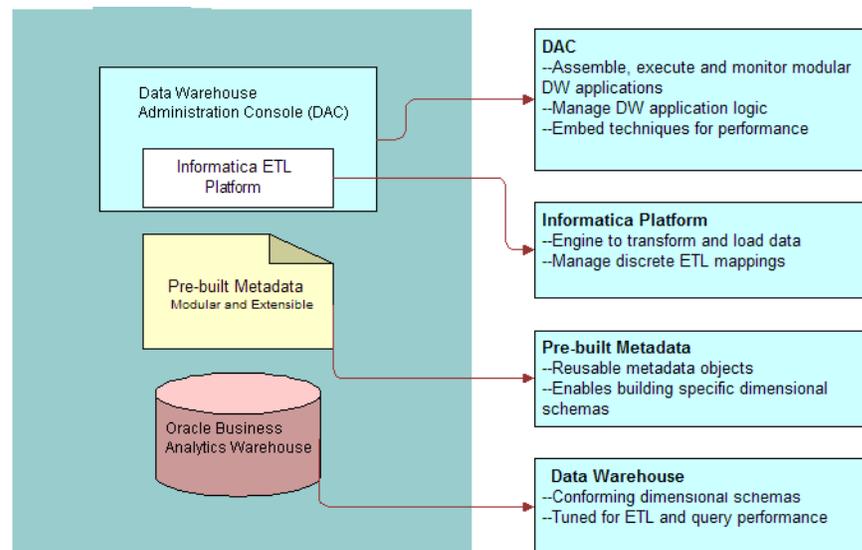
## 2.2 Oracle Business Analytics Warehouse Overview

The Oracle Business Analytics Warehouse is a unified data repository for all customer-centric data. The purpose of the Oracle Business Analytics Warehouse is to support the analytical requirements of Oracle E-Business Suite, Oracle's Siebel CRM, and PeopleSoft Applications.

The Oracle Business Analytics Warehouse includes the following:

- A complete relational enterprise data warehouse data model with numerous pre-built star schemas encompassing many conformed dimensions and several hundred fact tables.  
For more information about the data warehouse data model, please see the *Oracle Business Analytics Data Model Reference*.
- An open architecture to allow organizations to use third-party analytical tools in conjunction with the Oracle Business Analytics Warehouse using the Oracle Business Intelligence Server.
- Prebuilt data extractors to incorporate data from external applications into the Oracle Business Analytics Warehouse.
- A set of ETL (extract-transform-load) processes that takes data from Oracle E-Business Suite, Siebel CRM, PeopleSoft Enterprise and other transactional systems (OLTP), and creates the Oracle Business Analytics Warehouse tables.
- The Oracle Business Intelligence Data Warehouse Administration Console (DAC), a centralized console for the set up, configuration, administration, loading, and monitoring of the Oracle Business Analytics Warehouse. The figure below provides an overview of the Oracle Business Analytics Warehouse.

An overview of the Oracle Business Analytics Warehouse is shown in the figure below.

**Figure 2–1 Overview of the Oracle Business Analytics Warehouse**

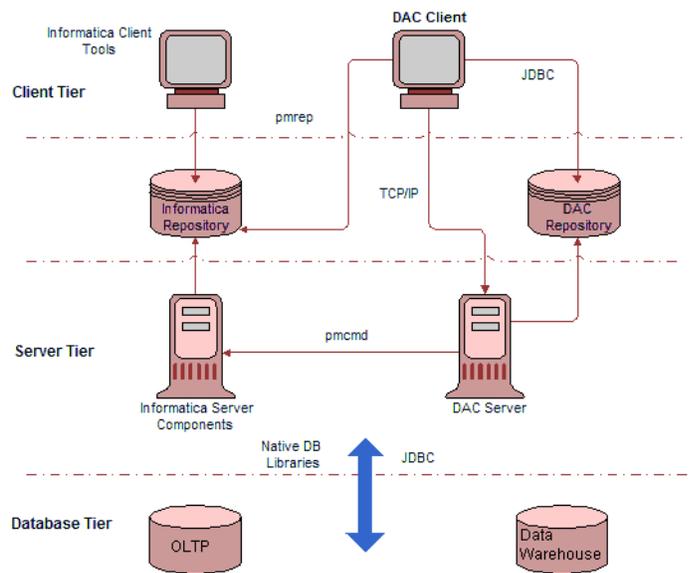
## 2.3 Oracle Business Analytics Warehouse Architecture

High-level analytical queries, like those commonly used in Oracle Business Intelligence, scan and analyze large volumes of data using complex formulas. This process can take a long time when querying a transactional database, which impacts overall system performance.

For this reason, the Oracle Business Analytics Warehouse was constructed using dimensional modeling techniques to allow for fast access to information required for decision making. The Oracle Business Analytics Warehouse derives its data from operational applications, and uses Informatica's data integration technology to extract, transform, and load data from various supported transactional database systems (OLTP) into the Oracle Business Analytics Warehouse.

### 2.3.1 Oracle Business Analytics Warehouse Architecture Components

The figure below illustrates the Oracle Business Analytics Warehouse architecture.

**Figure 2–2 An example Oracle Business Analytics Warehouse architecture**

The figure above shows the following Oracle Business Analytics Warehouse components:

- **DAC client.** A command and control interface for the data warehouse to allow for set up, configuration, administration, and monitoring of data warehouse processes.
- **DAC server.** Executes the instructions from the DAC client. The DAC server manages data warehouse processes, including scheduling, loading of the ETL, and configuring the subject areas to be loaded. It dynamically adjusts its actions based on information in the DAC repository. Depending on your business needs, you might incrementally refresh the Oracle Business Analytics Warehouse once a day, once a week, once a month, or on another similar schedule.
- **DAC repository.** Stores the metadata (semantics of the Oracle Business Analytics Warehouse) that represents the data warehouse processes.
- **Informatica Server Components:**
  - **Informatica Server.** When the server is loading or refreshing the Oracle Business Analytics Warehouse, it is recommended that the server be dedicated to that activity. However, when the ETL server is not loading or refreshing the Oracle Business Analytics Warehouse, you can use it for other purposes.
  - **Informatica Repository Server.** Manages the Informatica repository.
- **Informatica Repository.** Stores the metadata related to Informatica workflows.
- **Informatica client utilities.** Tools that allow you to create and manage the Informatica repository.

For information about using the Oracle Business Intelligence Data Warehouse Administration Console (DAC), see *Oracle Business Intelligence Data Warehouse Administration Console Guide*.

## 2.4 Roadmap To Installing, Configuring, and Customizing Oracle BI Applications

To install, configure, and customize Oracle BI Applications, do the following:

- For the source systems that you are using, follow the appropriate pre-installation steps in [Chapter 3, "Pre-installation and Predeployment Requirements for Oracle BI Applications"](#).
- Install and configure the Oracle BI Applications components as described in [Chapter 4, "Installing and Configuring Oracle BI Applications on Windows"](#) and/or [Chapter 5, "Installing Oracle BI Applications on Unix"](#).
- (Mandatory) Perform the source-independent steps that are required before a full data load, described in [Section 8.1.1, "Configuration Required Before A Full Load"](#), followed by the source system-specific steps that are required before a full data load for the appropriate source system that you are using, as follows:
  - [Section 8.2.1, "Configuration Required Before A Full Load for Oracle EBS"](#)
  - [Section 8.3.1, "Configuration Required Before A Full Load for PeopleSoft"](#)
  - [Section 8.4.1, "Configuration Required Before A Full Load for Siebel Source Systems"](#)
- (Optional) Perform any required additional source-independent steps described in [Section 8.1.2, "Configuration Steps for Controlling Your Data Set for All Source Systems"](#), followed by any required additional steps for the appropriate source system that you are using, as follows:
  - [Section 8.2.2, "Configuration Steps for Controlling Your Data Set for Oracle EBS"](#)
  - [Section 8.3.2, "Configuration Steps for Controlling Your Data Set for PeopleSoft"](#)
  - [Section 8.4.2, "Configuration Steps for Controlling Your Data Set for Siebel Source Systems"](#)
- (Mandatory) Perform the steps that are required before a full data load for all applications that you are deploying, from the appropriate chapters below:
  - [Chapter 9, "Configuring Oracle's Supply Chain Family of Products"](#)
  - [Chapter 10, "Configuring Oracle Financial Analytics"](#)
  - [Chapter 11, "Configuring Oracle Order Management and Fulfillment Analytics"](#)
  - [Chapter 12, "Configuring Oracle HR Analytics"](#)
  - [Chapter 13, "Configuring Oracle Sales Analytics"](#)
  - [Chapter 14, "Configuring Oracle Contact Center Telephony Analytics"](#)
- (Optional) Perform any required additional steps for all applications that you deploy, from the appropriate chapters below:
  - [Chapter 9, "Configuring Oracle's Supply Chain Family of Products"](#)
  - [Chapter 10, "Configuring Oracle Financial Analytics"](#)
  - [Chapter 11, "Configuring Oracle Order Management and Fulfillment Analytics"](#)
  - [Chapter 12, "Configuring Oracle HR Analytics"](#)

- [Chapter 13, "Configuring Oracle Sales Analytics"](#)
- [Chapter 14, "Configuring Oracle Contact Center Telephony Analytics"](#)
- (Optional) If you want to customize the out-of-the-box Oracle BI Applications functionality, follow the steps described in [Chapter 6, "Customizing the Oracle Business Analytics Warehouse"](#).
- (Optional) If you want to modify the out-of-the-box Oracle BI Applications security, follow the steps described in [Chapter 7, "Integrated Security for Oracle BI Applications"](#).

Once you have installed and configured Oracle BI Applications components, configured the modules (optional), and customized Oracle Business Intelligence Applications (optional), you are ready to start running ETLs (for more information about running ETLs, see *Oracle Business Intelligence Data Warehouse Administration Console Guide*).

---

---

## Pre-installation and Predeployment Requirements for Oracle BI Applications

This chapter provides information about preparing to install and deploy Oracle Business Intelligence Applications. You should review this information before you begin the installation and deployment process. For example, as a minimum you should read the general guidelines for setting up the Oracle Business Analytics Warehouse, and read the appropriate database-specific guidelines for the source OLTP databases that you are using.

---

---

**Note:** To install the server components, the computers need to meet the conditions specified in *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*.

---

---

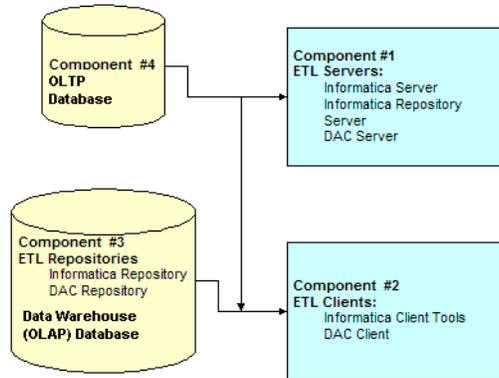
This chapter contains the following topics:

- [Section 3.1, "Oracle Business Analytics Warehouse Deployment Configuration"](#)
- [Section 3.2, "Operating System, Driver, and Connectivity Requirements for Oracle Business Analytics Warehouse"](#)
- [Section 3.3, "General Guidelines for Setting Up Oracle Business Analytics Warehouse"](#)
- [Section 3.4, "IBM DB2 UDB-Specific Database Guidelines for Oracle Business Analytics Warehouse"](#)
- [Section 3.5, "IBM DB2 UDB zOS and OS/390 and Z/os-Specific Database Guidelines for Oracle Business Analytics Warehouse"](#)
- [Section 3.6, "Oracle-Specific Database Guidelines for Oracle Business Analytics Warehouse"](#)
- [Section 3.7, "Additional Suggestions for Optimizing Oracle Performance in Oracle Business Analytics Warehouse"](#)
- [Section 3.8, "SQL Server-Specific Database Guidelines for Oracle Business Analytics Warehouse"](#)
- [Section 3.9, "Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse"](#)
- [Section 3.10, "Deploying Oracle Business Analytics Warehouse in Latin-1 General, Unicode and Non-English Environments"](#)
- [Section 3.11, "Miscellaneous Information About Oracle Business Intelligence Applications Deployments"](#)

## 3.1 Oracle Business Analytics Warehouse Deployment Configuration

The figure below shows the recommended Oracle Business Analytics Warehouse deployment configuration.

**Figure 3–1 Recommended Oracle Business Analytics Warehouse deployment**



In the figure above:

- Component 1 hosts the ETL servers (that is, the Informatica Server, the Informatica Repository Server, and the DAC server).

---



---

**Note:** You can install the Informatica Server on other machines as well to increase performance. The other ETL servers can also be hosted on other machines.

---



---

- Component 2 hosts all the ETL clients (that is, the Informatica client tools, and the DAC client).
- Components 3 (OLTP database) and 4 (Data Warehouse (OLAP) database) are database instances that can be hosted on one or more machines. The hardware requirements are entirely dependent on your usage and performance requirements. It is highly recommended that each of these components be defined in their own database instance to allow for instantiating parameters that are optimized for each component's usage.

---



---

**Note:** For more information about hardware and software requirements, see *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*.

---



---

## 3.2 Operating System, Driver, and Connectivity Requirements for Oracle Business Analytics Warehouse

Table 3–1 provides the operating system, driver, and connectivity software requirements for the Oracle Business Analytics Warehouse components.

---



---

**Note:** For information about the supported versions of the components listed in Table 3–1, see *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*.

---



---

**Table 3–1 OS, Driver and Connectivity Requirements for Oracle Business Analytics Warehouse**

Component	Operating System	Software	Connectivity and Driver
1 ETL Servers	<ul style="list-style-type: none"> <li>■ Solaris</li> <li>■ AIX</li> <li>■ HP</li> <li>■ Windows</li> <li>■ Linux</li> </ul>	<ul style="list-style-type: none"> <li>■ Informatica Server</li> <li>■ Informatica Repository Server</li> <li>■ DAC Server</li> </ul>	<ul style="list-style-type: none"> <li>■ Java SDK</li> <li>■ JDBC drivers</li> <li>■ Javamail</li> </ul>
2 ETL Clients	Windows	<ul style="list-style-type: none"> <li>■ Informatica client tools</li> <li>■ DAC client</li> </ul>	<ul style="list-style-type: none"> <li>■ Java SDK</li> <li>■ JDBC drivers</li> <li>■ ODBC drivers</li> </ul>
3 (A) Oracle Business Analytics Warehouse	<ul style="list-style-type: none"> <li>■ Solaris</li> <li>■ AIX</li> <li>■ HP</li> <li>■ Windows</li> <li>■ OS/390, Z/os</li> </ul>	Database software with Oracle Business Analytics Warehouse	Not applicable
3 (B) ETL Repositories	<ul style="list-style-type: none"> <li>■ Solaris</li> <li>■ AIX</li> <li>■ HP</li> <li>■ Windows</li> </ul>	Database software with ETL repositories	Not applicable

### 3.3 General Guidelines for Setting Up Oracle Business Analytics Warehouse

The Oracle Business Analytics Warehouse is a database that contains dimensional schemas. Although it is technically possible to put the Oracle Business Analytics Warehouse in the same database as the transactional database, it is not recommended for performance reasons. The transactional database is structured as an online transaction processing (OLTP) database, whereas the Oracle Business Analytics Warehouse is structured as an online analytical processing (OLAP) database, each optimized for its own purpose. The reasons for not combining the two databases are:

- The analytical queries interfere with normal use of the transactional database, which is entering and managing individual transactions.
- The data in a transactional database is normalized for update efficiency. Transactional queries join several normalized tables and will be slow (as opposed to pre-joined, de-normalized analytical tables).
- Historical data cannot be purged from a transactional database, even if not required for current transaction processing, because you need it for analysis. (By contrast, the analytical database is the warehouse for historical as well as current data.) This causes the transactional database to further slow down.
- Transactional databases are tuned for one specific application, and it is not productive to use these separate transactional databases for analytical queries that usually span more than one functional application.
- The analytical database can be specifically tuned for the analytical queries and Extract-Transform-Load (ETL) processing. These are quite different from transactional database requirements.

- On the transactional database, you should place the S\_ETL tables in a separate tablespace. These ETL tables are used by the Oracle Business Analytics Warehouse and should not be part of the routine backup processes.

A complete listing of these tables is available in *Oracle Business Analytics Warehouse Data Model Reference*.

- To maximize ETL performance on Siebel OLTP databases, create three indices on the Siebel OLTP database, using the following SQL commands:

```
CREATE INDEX S_AUDIT_ITEM_M100 ON S_AUDIT_ITEM (FIELD_NAME
ASC, BUSCOMP_NAME DESC) PCTFREE 10 ALLOW REVERSE SCANS
COLLECT DETAILED STATISTICS;
```

```
CREATE INDEX S_AUDIT_ITEM_M101 ON S_AUDIT_ITEM (RECORD_ID ASC,
FIELD_NAME DESC) PCTFREE 10 ALLOW REVERSE SCANS COLLECT
DETAILED STATISTICS;
```

```
CREATE INDEX S_OPTY_M102 ON S_OPTY (ROW_ID ASC, PR_POSTN_ID
DESC) PCTFREE 10 ALLOW REVERSE SCANS COLLECT DETAILED
STATISTICS;
```

The Informatica repository stores all of the Informatica object definitions for the ETL mappings that populate the Oracle Business Analytics Warehouse. It is a series of repository tables that are stored in a database, which can be transactional, analytical, or a separate database.

The Oracle Business Analytics Warehouse works with relational database management systems. In addition to the general requirements, there are additional database management systems (DBMS)-specific requirements depending on the DBMS you are using.

The following general guidelines will help you set up the data warehouse physical database for performance and growth:

- At a minimum, separate the data and index tablespaces. Create more tablespaces to separate heavily used tables and their indices.
- Use the maximum block and page size available for tablespaces ((for example, 32K), because it provides good overall performance and also does not impose low limits to the maximum size to which the tablespace can grow, as compared to 4K, 8K,16K sizes.
- If you are using multiple disk storage systems, stripe the tablespace containers and files across as many disks as possible.
- Raw devices for tablespaces provide better performance as compared to cooked file systems.
- RAID-5 is known to give a good balance of performance and availability.
- Size the buffer pools based on content and size (number of tables and their sizes) of tablespaces.
- Allocate about 75 percent of the total available server memory to the database, assuming no other application is running on the same server.

During the Oracle Business Analytics Warehouse configuration process, when you create the data warehouse tables using the procedure [Section 4.12.2, "How to Create Data Warehouse Tables"](#), you can create tables in one tablespace and indices in another tablespace. However, for performance reasons, it is recommended that you create tablespaces as described in [Table 3-2](#).

**Table 3–2 Recommended Tablespace Configuration**

Tablespace Name	List of Tables
DIM_STG	W_*DS
FACT_STG	W_*FS
DIM	W_*D and W_*MD
FACT	W_*F
AGG	W_*A
OTHER	Remaining W* tables
DIM_INDX	Indices of W_*D tables (for example, other tables would include W*G and W*GS tables)
FACT_INDX	Indices of W_*F tables
OTHER_INDX	Remaining Indices of W* tables

---

**Note:** To avoid fatal deadlocks during the ETL, make sure that you select the 'Session Level Retry on Deadlock' option in Informatica.

---

### 3.4 IBM DB2 UDB-Specific Database Guidelines for Oracle Business Analytics Warehouse

Table 3–3 provides guidelines for parameter settings for DB2 relational database management system (RDBMS) usage. Use these guidelines as a starting point. You will need to make changes based on your specific database sizes, data shape, server size (CPU and memory), and type of storage. The database administrator should make changes to the settings based on performance monitoring and tuning considerations.

**Table 3–3 Recommended DB2 Parameter Settings**

Parameter	DB2 UDB V7	DB2 UDB V8	Notes
SHEAPTHRES	400000	400000	
ASLHEAPSZ	15	15	
RQRIOBLK	65535	65535	
QUERY_HEAP_SZ	16384	16384	
JAVA_HEAP_SZ	2048	2048	
MAXAGENTS	400	400	
NUM_INITAGENTS	10	10	
NUM_POOLAGENTS	200	200	
INTRA_PARALLEL	YES	YES	
FCM_NUM_BUFFERS	12288	12288	
SHEAPTHRES_SHR	N/A	=SHEAPTHRES	
DBHEAP	16384	16384	
CATALOGCACHE_SZ	5558	5558	

**Table 3–3 (Cont.) Recommended DB2 Parameter Settings**

Parameter	DB2 UDB V7	DB2 UDB V8	Notes
LOGBUFSZ	2048	2048	
UTIL_HEAP_SZ	10000	10000	
NUM_ESTORE_SEGS	16	NIL	Restore is not needed in DB2 V8 64-bit because the V7 limit of 1.75 GB addressable memory has been lifted.
ESTORE_SEG_SZ	65536	NIL	
LOCKLIST	25000	25000	
APP_CTL_HEAP_SZ	5000	5000	
SORTHEAP	4000	4000	
STMTHEAP	40960	40960	
APPLHEAPSZ	2560	2560	
PCKCACHESZ	2560	2560	
STAT_HEAP_SZ	20000	20000	
DLCHKTIME	10000	10000	
MAXLOCKS	50	50	
LOCKTIMEOUT	1200	1200	
MAXAPPLS	500	500	
AVG_APPLS	10	10	
MAXFILOP	500	500	
GROUPHEAP_RATIO	N/A	70	New in V8
APPGROUP_MEM_SZ	N/A	30000	New in V8
DATABASE_MEMORY	N/A	AUTOMATIC	New in V8

---

**Note:** To avoid fatal deadlocks during the ETL, make sure that you select the 'Session Level Retry on Deadlock' option in Informatica.

---

### 3.5 IBM DB2 UDB zOS and OS/390 and Z/os-Specific Database Guidelines for Oracle Business Analytics Warehouse

The following requirements apply to IBM DB2 RDBMS usage for zOS and OS/390:

- The Analytics applications communicate with IBM DB2 UDB for z/OS and OS/390 (running on zSeries servers) through IBM DB2 Connect middleware.

The following editions of DB2 Connect are supported:

- **DB2 Connect Enterprise Edition (EE).** This edition is installed on a mid tier server such as an Informatica Server/Client, DAC, and Oracle Business Intelligence.

- **DB2 Connect Unlimited Edition (UE).** This edition provides the functionality of DB2 Connect Enterprise Edition but is priced differently.
- The ODBC driver for all connections must use the IBM DB2 ODBC Driver.
- Make the appropriate connections using the DB2 Client Configuration Assistant.
- Use the variable settings shown in [Table 3–4](#).

**Table 3–4 Variable Settings for IBM DB2 UDB zOS and OS/390 Databases**

Parameter	Recommended Setting	Notes
IDTHTOIN	1800	
CDSRDEF	Any	
STARJOIN	1	This setting indicates that star join is enabled. The one table with the largest cardinality is the fact table. However, if there is more than one table with this cardinality, star join is not enabled.

## 3.6 Oracle-Specific Database Guidelines for Oracle Business Analytics Warehouse

**Note:** Oracle Business Intelligence Applications Version 7.9.4 does not support Oracle 11g source systems.

To configure Business Analytics Data warehouses on Oracle databases more easily, refer to the following `init.ora` parameter template files, located in `<DRIVE>:\<BI Apps install directory>\dwrep\Documentation\`. For example, `C:\OracleBI\dwrep\Documentation\`.

- `init9iR2.ora` - `init.ora` template for Oracle RDBMS 9iR2
- `init10gR2.ora` - `init.ora` template for Oracle RDBMS 10gR2

The `init.ora` parameter template files provide parameter guidelines based on the rule-based optimizer for Oracle 8i and the cost-based optimizer for Oracle 9i and Oracle 10g. Use these guidelines as a starting point. You will need to make changes based on your specific database sizes, data shape, server size (CPU and memory), and type of storage. The database administrator should make changes to the settings based on performance monitoring and tuning.

Copy the template file corresponding to your database version into your `$ORACLE_HOME/dbs` directory, review the recommendations in the template file, and make the changes based on your specific database sizes, data shape, server size (CPU and memory), and type of storage. The database administrator should make changes to the settings based on performance monitoring and tuning considerations.

## 3.7 Additional Suggestions for Optimizing Oracle Performance in Oracle Business Analytics Warehouse

This section contains additional suggestions for optimizing performance.

- Oracle Business Intelligence Applications under Oracle support only binary sorting. If you are running an Oracle client, do one of the following:
  - Set the `NLS_SORT` parameter to `BINARY`.
  - Choose a `NLS_LANG` setting that includes binary.

These settings are required for adequate performance from the dedicated Web client.

- Make sure that cost-based optimization is enabled in the Oracle development, test, and production databases and that statistics are kept up to date. Otherwise, the rule-based optimizer may be used.
- Create foreign keys in the Oracle database, but configure Oracle to not enforce the foreign key relationship. The existence of foreign keys will allow Oracle to better optimize certain queries. By turning off enforcement, the database load should not be negatively affected.
- Analyze application for occurrences of highly skewed data that is indexed. Create histogram statistics for these indices to enable the optimizer to better perform queries.
- To increase data throughput between Oracle BI Server and Oracle, change SDU and TDU settings in listener.ora. The default is 2 KB and can be increased to 8 KB.
- On the server side, edit the listener.ora file. Under the particular SID\_LIST entry, modify SID\_DESC as follows:

```
SID_LIST_LISTENER =
  SID_LIST =
    SID_DESC = (SDU=16384)(TDU=16384)
              (ORACLE_HOME = /.....)
              (SID_NAME = SOLAP)
    )
  )
```

- Set the tablespace to at least the same as the transactional database size. Make sure the temporary tablespace has adequate space.
- Set the number of log file groups to 4.
- Set the size of each log file to 10 MB.
- The sga\_max\_size to 700 MB.
- On the client side, edit the tnsnames.ora file. Modify the TNS alias by adding SDU= and TDU= as follows:

```
myhost_orcl.world=
  DESCRIPTION=(SDU=16384)(TDU=16384)
  ADDRESS = (PROTOCOL = TCP)(HOST=myhost)(PORT=1521)
  CONNECT_DATA=(SID=ORCL)
```

## 3.8 SQL Server-Specific Database Guidelines for Oracle Business Analytics Warehouse

This section provides guidelines for SQL Server database usage.

---

**Note:** The SQL Server database must be created with a collation sequence that supports binary sort order or case-sensitive dictionary sort order. Case-insensitive dictionary sort order is not supported. For example, for binary sort order with the U.S English character set, use the collation "Latin1\_General\_BIN". If you use the default collation setting of "SQL\_Latin1\_General\_CP1\_CI\_AS", the database is set to case-insensitive, which is not supported, and causes index creation failures.

---

This section includes the following topics:

- [Section 3.8.1, "Setting the ANSI NULL Option"](#)
- [Section 3.8.2, "Modifying the DB Library Options Setting"](#)
- [Section 3.8.3, "Recommended SQL Server Database Parameters"](#)

### 3.8.1 Setting the ANSI NULL Option

Oracle Business Intelligence Applications requires that SQL Server databases be created with the ANSI NULL option selected.

#### To set the ANSI NULL option

1. In the SQL Server Enterprise Manager, right-click the appropriate database, and choose Properties.
2. Click the Options tab and select the box for ANSI NULL default.

### 3.8.2 Modifying the DB Library Options Setting

In a SQL Server 2000 environment, when loading Analytics tables with international data, or loading more than one language, you need to modify the DB Library Options setting.

#### To modify the DB Library Options setting

1. From the Microsoft SQL Server program menu, select Client Network Utility.
2. Select the DB Library Options tab.
3. Clear the option Automatic ANSI to OEM.

---



---

**Note:** SQL Server 2000 automatically tunes many of the server configuration options; therefore, an administrator is required to do little, if any, tuning. Although these configuration options can be modified, the general recommendation is that these options be left at their default values, allowing SQL Server to automatically tune itself based on run-time conditions.

---



---

### 3.8.3 Recommended SQL Server Database Parameters

If necessary, SQL Server components can be configured to optimize performance, as shown in [Table 3-5](#).

**Table 3-5 Recommended Variable Settings for SQL Server Databases**

Parameter	Recommended Setting	Notes
Affinity mask	0	
Allow updates	0	
Awe enabled	0	
C2 audit mode	0	
Cost threshold for parallelism	5	
Cursor threshold	-1	
Default full-text language	1033	

**Table 3–5 (Cont.) Recommended Variable Settings for SQL Server Databases**

Parameter	Recommended Setting	Notes
Default language	0	
Fill factor	95%	For insert-intensive transactions, set Fill Factor between 90 and 95%. For better query performance, set Fill factor to 95 or even 100%.
Index create memory	1024 KB	Default is 0.
Lightweight pooling	0	
Locks	0	
Max degree of parallelism	0	Default is 0. This turns off parallelism. Max degree of parallelism should be left at 0, which means use parallel plan generation. It should be set to 1 (use only 1 process) if you run multi threaded components (for example, several EIM threads).
Max server memory	2000 MB	Default is 2147483647.
Max text repl size	65536 B	
Max worker threads	100	Default is 255.
Media retention	0	
Min memory per query	1024 KB	
Min server memory	500 MB	Default is 0.
Nested triggers	1	
Network packet size	8192 B	Default is 4096.
Open objects	0	
Priority boost	0	
Query governor cost limit	0	Modify to 60 only if CPU is high.
Query wait	-1 sec	
Recovery interval	0 min	
Remote access	1	
Remote login timeout	20 sec	
Remote proc trans	0	
Remote query timeout	600 sec	
Scan for startup procs	0	
Set working set size	0	
Two-digit year cutoff	2049	
User connections	0	
User options	0	

- **SQL Server memory:** Make sure adequate memory is available.
- **Transaction logs and TempDB:** Reside on a separate disk from those used by database data.

- **Full load:** Full Recovery model for the database.
- **Incremental (Refresh) load:** Change from Full to Bulk-Logged Recovery model.

## 3.9 Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse

This sections contains the recommended best practices and guidelines for maximizing performance in Teradata deployments, and contains the following topics:

- [Section 3.9.1, "General Guidelines for Teradata Deployments"](#)
- [Section 3.9.2, "Best Practices For Teradata Deployments"](#)

### 3.9.1 General Guidelines for Teradata Deployments

The following requirements apply to Teradata database usage:

- Install the FastLoad and TPump Teradata utilities on the machine on which the Informatica Server is installed.
- When installing Informatica PowerCenter, make sure there are no spaces in the Informatica Server directory name or the directory path. The default directory contains spaces that you must remove manually.
- When you install Oracle BI Applications in a Teradata environment, you must load the pre-built Informatica repository file \dwrep\Oracle\_BI\_DW\_Teradata.rep. For more information, see [Section 4.14.3, "How to Load the Pre-Built Repository Into Informatica"](#).
- In the ODBC configuration for Teradata, set the session Mode to ANSI and DateFormat to AAA, before you create the schema. Tables must be created as case specific. If you do the ODBC configuration after the schema is created, tables might not be created as case specific.
- To avoid overflows if the arithmetic calculation involves any decimal data type more than 18,3 (precision, scale), add the following static Source System Parameters:
  - "as decimal(18,3))" for \$\$Hint\_Tera\_Post\_Cast
  - "Cast (" for \$\$Hint\_Tera\_Pre\_Cast    \$\$Hint\_Tera\_Post\_Cast    \$\$Hint\_Tera\_Pre\_Cast)

For more information about setting Source System Parameters in the DAC, refer to [Section 4.18.3, "How to Set DAC Source System Parameters"](#).

---



---

**Note:** To avoid fatal deadlocks during the ETL, make sure that you select the 'Session Level Retry on Deadlock' option in Informatica.

---



---

### 3.9.2 Best Practices For Teradata Deployments

This sections contains the recommended best practises for maximizing performance in Teradata deployments, and contains the following topics:

- [Section 3.9.2.1, "LEFT OUTER JOIN Issue"](#)
- [Section 3.9.2.2, "Database Statistics"](#)
- [Section 3.9.2.5, "Group By Vs Distinct"](#)

- [Section 3.9.2.6, "Pruning Of Mapping"](#)
- [Section 3.9.2.7, "Loader Configurations"](#)

---



---

**Note:** The following best practices should be treated as customizations. You should follow standard customization methodology, such as copying the mapping to a Custom directory. You should never make these changes directly in the OTB objects.

---



---

### 3.9.2.1 LEFT OUTER JOIN Issue

Teradata redistributes the OUTER table according to the join key, unless the INNER table is tiny. If the INNER table is tiny, the INNER table could be simply copied to all AMPs and the OUTER table would not be redistributed.

Many fact SIL mappings need to get the dimensional ROW\_WID from ROW\_ID/INTEGRATION\_ID. For example, W\_PER\_RANK\_FS.ACCNT\_ID needs to be converted to ACCNT\_WID before loading into W\_PER\_RANK\_F table. Since ACCT\_ID is nullable, the join between W\_PER\_RANK\_FS and W\_ORG\_D is defined as a LEFT OUTER JOIN.

However, the percentage of NULLs in ACCT\_ID column can be as high as 50% or more depending on the data set. When redistributing the W\_PER\_RANK\_FS according to ACCT\_ID, all rows with ACCT\_ID = NULL are put onto a single AMP.

Although a Teradata database usually has hundreds of gigabytes of spool space, the spool space is allocated across hundreds of AMPs. The spool space for each AMP is limited (for example, to two gigabytes).

When a large percentage of W\_PER\_RANK\_FS is distributed to a single AMP, this can result in insufficient spool space being available. This happens when too much data is spooled onto one AMP, not because the spool space is too small.

To work with Teradata's mechanism of parallel processing, and resolving the LEFT OUTER JOIN, SQL must be re-written.

As an example, refer to the following original SQL:

```
SELECT ... FROM
W_PER_RANK_FS FS LEFT OUTER JOIN W_ORG_D ORG ON
  FS.ACCNT_ID = ORG.INTEGRATION_ID AND
  FS.DATASOURCE_NUM_ID = ORG.DATASOURCE_NUM_ID
```

The above SQL should be re-coded as follows:

```
SELECT ... FROM
W_PER_RANK_FS FS LEFT OUTER JOIN
  (SELECT
    FS.INTEGRATION_ID, FS.DATASOURCE_NUM_ID, ORG.ROW_WID, ORG.GEO_WID
  FROM
    W_PER_RANK_FS FS, W_ORG_D ORG
  WHERE
    FS.ACCNT_ID = ORG.INTEGRATION_ID AND FS.DATASOURCE_NUM_ID =
    ORG.DATASOURCE_NUM_ID AND FS.ACCNT_ID IS NOT NULL)
ORG ON
  FS.DATASOURCE_NUM_ID = ORG.DATASOURCE_NUM_I AND
  FS.INTEGRATION_ID = ORG.INTEGRATION_ID
```

The same SQL re-coding method can be used for other Source Qualifiers that run into spool space problems.

In addition, there is another solution to the lack of spool space issue. This solution is to convert the NULLs to some evenly distributed but non-matched values, as shown in the following SQL example:

```
SELECT ... FROM
W_PER_RANK_FS FS LEFT OUTER JOIN W_ORG_D ORG ON
CASE WHEN FS.ACCNT_ID IS NOT NULL
THEN FS.ACCNT_ID
ELSE '#' || FS.INTEGRATION_ID END
= ORG.INTEGRATION_ID
AND FS.DATASOURCE_NUM_ID = ORG.DATASOURCE_NUM_ID
```

---



---

**Note:** The re-coded SQL is shown in bold type.

---



---

### 3.9.2.2 Database Statistics

This section describes database statistics.

### 3.9.2.3 Additional Statistics Required

Oracle Business Intelligence provides a list of column and index statistics collection statements. These are generally applicable for all situations, but should be evaluated for each site and situation. Additional statistics might be required, depending on your requirements. For example, some workflows need more statistics to avoid an out of spool space error on Teradata server.

### 3.9.2.4 Collect Statistics As A Pre-requisite

Once the tables have been created in the staging and target databases, you must run the supplied statistics collection. Failure to do so can affect ETL performance, and possibly result in a spool space error (error number 2646)

The DAC does the re-collection of statistics as part of the ETL processes. However, the DAC issues the collect statistics statement at the table level only (for example, collect statistics on w\_org\_d), and only for existing statistics.

### 3.9.2.5 Group By Vs Distinct

When there is a low number of distinct values, it is more efficient to use the GROUP BY phrase. Do not use the DISTINCT phrase, unless the number of distinct values is high.

### 3.9.2.6 Pruning Of Mapping

If you do not use all of the OTB fields supplied, you can improve performance by stripping the extraneous fields from the mapping and the tables.

### 3.9.2.7 Loader Configurations

This section explains the loaders that are available in Teradata, and how they are used in Oracle BI Applications.

Teradata has 3 different types of Teradata loader processes, as follows:

- Fastload
- Mload
- T pump

Each loader process can be used in two different modes, as follows:

- Staged Mode: The Informatica process does the following in this order:

- Reads from the source data.
- Creates a data file.
- Invokes the loader process to load the table using the data file created.

**Advantages:** In the event of failures, you can recover using the Teradata recovery process.

**Disadvantages:** Staged mode is slower than Piped mode, and you need more disk space, as it can create large data files.

- Piped Mode: The Informatica process reads from the source and simultaneously pipes that data to the loader to start loading the target table.

**Advantages:** Quicker than Staged mode, and you do not require large amounts of disk space because no data files are created.

**Disadvantages:** In the event of failures, you cannot recover using the Teradata recovery process (because tpump does row commits unlike fastload and mload).

### 3.9.2.8 Fastload

You use the Fastload process on empty tables, such as loading staging tables and in initial loads where the tables are empty.

When the Fastload process starts loading, it locks the target table, which means that processes (for example, lookups) cannot access that table. One solution to this problem is to specify dummy SQL for the look up overrides at the session level.

**Tip:** If a session fails during a Fastload process, use SQL Assistant to run a simple SQL command (for example, count(\*)), to determine whether the table is locked by a Fastload process.

If a table is locked (for example, (for W\_ORG\_DS), use the following script to release the lock:

```
LOGON DATABASEALIAS/USER,PASSWORD
BEGIN LOADING USER.W_ORG_DS
ERRORFILES USER.ET_W_ORG_DS,USER.UV_W_ORG_DS;
END LOADING;
```

If you save the above text in a file called test.ctl, you would run this process by entering the following command at a command prompt:

```
C:\fastload\test.ctl
```

**Tip:** To create a load script for a table, edit the test.ctl script above to change the login information, and replace all occurrences of W\_ORG\_DS with the required target table name.

After a load process script runs successfully, you should be able to run the command 'select count(\*)' on the target table. If you are not able release the lock, you might need to drop and re-create the table to remove the lock. If you do so, you must re-create the statistics.

**Tip:** Fastload is typically used in piped mode to load staging tables and initial loads. In the event of errors, reload the entire data.

### 3.9.2.9 Mload

The Mload process is slower than Fastload but quicker than Tmpump. The Mload process can work on both empty tables as well as on tables with data. In the event of errors when running in piped mode, you cannot recover the data.

### 3.9.2.10 Tmpump

The Tmpump process is slower than Mload but faster than ODBC. The Tmpump process does row commits, which enables you to recover processed operations, even if you use piping mode. In other words, if you re-start the process, Tmpump starts loading data from the last committed data.

Tmpump can be used in the following modes:

- **Tmpump\_Insert:** Use to do inserts.
- **Tmpump\_Update:** Use to do updates (this mode requires you to define the primary key in the Informatica target table definition).
- **Tmpump\_Upsert:** Use to do update otherwise insert (this mode requires you to define the primary key in the Informatica target table definition).
- **Tmpump\_Delete:** Use to do deletes (this mode requires you to define the primary key in the Informatica target table definition).

Informatica uses the actual target table name to generate the error table and log tables to be used as part of its control file generation. If you have two instances of Tmpump loading into same target table at the same time, you need to modify the session to use a different error table and log table name.

The Tmpump load process in piped mode is useful for incremental loads, and where the table is not empty. In the event of errors, restart the process and it starts re-loading from the last committed data.

Refer to Informatica documentation for information about configuring a session to use Teradata loaders.

## 3.10 Deploying Oracle Business Analytics Warehouse in Latin-1 General, Unicode and Non-English Environments

This section describes the different settings for Informatica servers and databases when deploying the Oracle Business Analytics Warehouse in Latin-1 General, Unicode, and non-English environments. When you configure Informatica, the Informatica repository, and the databases, you will need to refer to this section.

---

**Note:** If you are installing the Oracle Business Analytics Warehouse in a UNIX environment, you must use the Unicode character data mode.

---

The Oracle Business Analytics Warehouse can be deployed in various code page environments to support global deployments. The following source and data warehouse configurations are supported:

- **Latin-1 General (7-bit ASCII) to Latin-1 General (7-bit ASCII).** 7-bit ASCII is the character set used for English. See [Section 3.10.2, "Source and Data Warehouse Code Pages for Latin-1 General \(7-bit ASCII\) to Latin-1 General \(7-bit ASCII\)"](#).

- **Latin-1 General (8-bit ASCII) to Latin-1 General (8-bit ASCII).** 8-bit ASCII is the character set used for Western European languages that use accented Roman characters. See [Section 3.10.3, "Source and Data Warehouse Code Pages for Latin-1 General \(8-bit ASCII\) to Latin-1 General \(8-bit ASCII\)"](#).

---

**Note:** You must run the Informatica Server in Unicode mode if your source data contains multibyte or ISO 8859-1 (8-bit ASCII) data. For instructions on setting the data movement mode to Unicode, see [Section 4.5.3, "How to Set Up The Informatica Server"](#).

---

- **Unicode to Unicode.** See [Section 3.10.3, "Source and Data Warehouse Code Pages for Latin-1 General \(8-bit ASCII\) to Latin-1 General \(8-bit ASCII\)"](#).
- **Code page (multi- or single-byte) to Unicode.** See [Section 3.10.5, "Source and Data Warehouse Code Pages for Code Page to Unicode"](#).
- **Code page to code page (where the code pages are the same).** See [Section 3.10.6, "Source and Data Warehouse Code Pages for Code Page to Code Page"](#).

The following environment variables need to be set before you begin the installation process:

- **NLS\_LANG (for Oracle).** For instructions, see [Section 3.10.7, "Setting the NLS\\_LANG Environment Variable for Oracle Databases"](#).
- **DB2CODEPAGE (for DB2).** For instructions, see [Section 3.10.8, "Setting the DB2CODEPAGE Environment Variable for DB2 Databases"](#).

The following environment variable needs to be set during the configuration process:

- **Informatica Server Data Movement.** For instructions, see [Section 4.5.3, "How to Set Up The Informatica Server"](#).

## 3.10.1 Code Page Overview

A code page contains the encoding to specify characters in a set of one or more languages. An encoding is the assignment of a number to a character in the character set. You use code pages to identify data that might be in different languages. For example, if you are importing Japanese data into a mapping, you must select a Japanese code page for the source data.

When you set a code page, the application or program for which you set the code page refers to a specific set of data that describes the characters the application recognizes. This influences the way that application stores, receives, and sends character data.

### 3.10.1.1 Choosing Code Pages

Choose code pages based on the character data you are using in mappings. Character data can be represented by character modes based on the character size.

Character size is measured by the amount of storage space a character requires in the database. Database character can be one of the following: single byte, double byte or multibyte.

### 3.10.1.2 Code Page Compatibility

Compatibility between code pages is essential for accurate data movement when the Informatica Server runs in the Unicode data movement mode. When two code pages are compatible, the characters encoded in the two code pages are virtually identical.

For accurate data movement, the data warehouse code page must be a superset of the source code page. If the source code page is a superset of the data warehouse code page, the Informatica Server cannot process the characters because they cannot be encoded in the data warehouse code page. The data warehouse then results in incorrect or missing data.

### 3.10.2 Source and Data Warehouse Code Pages for Latin-1 General (7-bit ASCII) to Latin-1 General (7-bit ASCII)

This section provides the code pages for Latin-1 General (7-bit ASCII) to Latin-1 General (7-bit ASCII) configurations. 7-bit ASCII is the character set used for English.

#### 3.10.2.1 Informatica Server and Repository Server Running on Windows with OS ENU

Table 3–6 provides the code pages for the Informatica Server and Repository Server running on Windows with OS ENU.

**Table 3–6 Code Pages for Informatica Server and Repository Server on Windows with OS ENU**

Component Code Page	Code Page
Source code page	MS Windows Latin 1 (ANSI), superset of Latin 1
Data warehouse code page	MS Windows Latin 1 (ANSI), superset of Latin 1
Informatica Repository code page	MS Windows Latin 1 (ANSI), superset of Latin 1
Informatica Server code page	MS Windows Latin 1 (ANSI), superset of Latin 1

#### 3.10.2.2 Informatica Server and Repository Server Running on UNIX with OS ENU

The following environment variables need to be set before you begin the installation process:

- **NLS\_LANG (for Oracle).** For instructions, see [Section 3.10.7, "Setting the NLS\\_LANG Environment Variable for Oracle Databases"](#).
- **DB2CODEPAGE (for DB2).** For instructions, see [Section 3.10.8, "Setting the DB2CODEPAGE Environment Variable for DB2 Databases"](#).

The following environment variable needs to be set during the configuration process:

- **Informatica Server Data Movement.** For instructions, see [Section 4.5.3, "How to Set Up The Informatica Server"](#).

Table 3–7 provides the code pages for the Informatica Server and Repository Server running on UNIX with OS ENU.

**Table 3–7 Code Pages for Informatica Server and Repository Server on UNIX with OS ENU**

Component Code Page	Code Page
Source code page	MS Windows Latin 1 (ANSI), superset of Latin 1
Data warehouse code page	MS Windows Latin 1 (ANSI), superset of Latin 1
Informatica Repository code page	ISO 8859-1 Western European
Informatica Server code page	ISO 8859-1 Western European

### 3.10.2.3 Informatica Server on UNIX, Repository Server on Windows, both with OS ENU

The following environment variables need to be set before you begin the installation process:

- **NLS\_LANG (for Oracle).** For instructions, see [Section 3.10.7, "Setting the NLS\\_LANG Environment Variable for Oracle Databases"](#).
- **DB2CODEPAGE (for DB2).** For instructions, see [Section 3.10.8, "Setting the DB2CODEPAGE Environment Variable for DB2 Databases"](#).

The following environment variable needs to be set during the configuration process:

- **Informatica Server Data Movement.** For instructions, see [Section 4.5.3, "How to Set Up The Informatica Server"](#).

[Table 3–8](#) provides the code pages for the Informatica Server running on UNIX, and the Repository Server on Windows, both with OS ENU.

**Table 3–8 Code Pages for Informatica Server on UNIX, Repository Server on Windows with OS ENU**

Component Code Page	Code Page
Source code page	MS Windows Latin 1 (ANSI), superset of Latin 1
Data warehouse code page	MS Windows Latin 1 (ANSI), superset of Latin 1
Informatica Repository code page	MS Windows Latin 1 (ANSI), superset of Latin 1
Informatica Server code page	ISO 8859-1 Western European

## 3.10.3 Source and Data Warehouse Code Pages for Latin-1 General (8-bit ASCII) to Latin-1 General (8-bit ASCII)

This section provides the code pages for Latin-1 General (8-bit ASCII) to Latin-1 General (8-bit ASCII) configurations. 8-bit ASCII is the character set used for Western European languages that use accented Roman characters.

---

**Note:** You must run the Informatica Server in UNICODE mode if your source data contains multibyte or ISO 8859-1 (8-bit ASCII) data. For instructions on setting the data movement mode to UNICODE, see [Section 4.5.3, "How to Set Up The Informatica Server"](#).

---

### 3.10.3.1 Informatica Server and Repository Server Running on Windows with OS ENU

[Table 3–9](#) provides the code pages for the Informatica Server and Repository Server running on Windows with OS ENU.

**Table 3–9 Code Pages for Informatica Server and Repository Server on Windows with OS ENU**

Component Code Page	Code Page
Source code page	ISO 8859-1 Western European
Data warehouse code page	ISO 8859-1 Western European
Informatica Repository code page	MS Windows Latin 1 (ANSI), superset of Latin 1
Informatica Server code page	MS Windows Latin 1 (ANSI), superset of Latin 1

### 3.10.3.2 Informatica Server and Repository Server Running on UNIX with OS ENU

The following environment variables need to be set before you begin the installation process:

- **NLS\_LANG (for Oracle).** For instructions, see [Section 3.10.7, "Setting the NLS\\_LANG Environment Variable for Oracle Databases"](#).
- **DB2CODEPAGE (for DB2).** For instructions, see [Section 3.10.8, "Setting the DB2CODEPAGE Environment Variable for DB2 Databases"](#).

The following environment variable needs to be set during the configuration process:

- **Informatica Server Data Movement.** For instructions, see [Section 4.5.3, "How to Set Up The Informatica Server"](#).

[Table 3–10](#) provides the code pages for the Informatica Server and Repository Server running on UNIX with OS ENU.

**Table 3–10 Code Pages for Informatica Server and Repository Server on UNIX with OS ENU**

Component Code Page	Code Page
Source code page	ISO 8859-1 Western European
Data warehouse code page	ISO 8859-1 Western European
Informatica Repository code page	ISO 8859-1 Western European
Informatica Server code page	ISO 8859-1 Western European

### 3.10.3.3 Informatica Server on UNIX, Repository Server on Windows, both with OS ENU

The following environment variables need to be set before you begin the installation process:

- **NLS\_LANG (for Oracle).** For instructions, see [Section 3.10.7, "Setting the NLS\\_LANG Environment Variable for Oracle Databases"](#).
- **DB2CODEPAGE (for DB2).** For instructions, see [Section 3.10.8, "Setting the DB2CODEPAGE Environment Variable for DB2 Databases"](#).

The following environment variable needs to be set during the configuration process:

- **Informatica Server Data Movement.** For instructions, see [Section 4.5.3, "How to Set Up The Informatica Server"](#).

[Table 3–11](#) provides the code pages for the Informatica Server running on UNIX, and the Repository Server on Windows, both with OS ENU.

**Table 3–11 Code Pages for Informatica Server on UNIX, Repository Server on Windows with OS ENU**

Component Code Page	Code Page
Source code page	ISO 8859-1 Western European
Data warehouse code page	ISO 8859-1 Western European
Informatica Repository code page	MS Windows Latin 1 (ANSI), superset of Latin 1
Informatica Server code page	ISO 8859-1 Western European

### 3.10.4 Source and Data Warehouse Code Pages for Unicode to Unicode

For a list of the code pages that are supported for source and data warehouse databases, see System Requirements and Supported Platforms on Oracle's Siebel SupportWeb.

The following environment variables need to be set before you begin the installation process:

- **NLS\_LANG (for Oracle).** For instructions, see [Section 3.10.7, "Setting the NLS\\_LANG Environment Variable for Oracle Databases"](#).
- **DB2CODEPAGE (for DB2).** For instructions, see [Section 3.10.8, "Setting the DB2CODEPAGE Environment Variable for DB2 Databases"](#).

The following environment variables need to be set during the configuration process:

- **Informatica Server Data Movement.** For instructions, see [Section 4.5.3, "How to Set Up The Informatica Server"](#).
- **SiebelUnicodeDB.** For instructions on Windows, see [Section 4.6.2, "How to Set Siebel UnicodeDB Environment Variables on Windows"](#). For instructions on UNIX, see [Section 5.9, "How to Set Environment Variables for the Informatica Server on UNIX"](#).

#### 3.10.4.1 Informatica Server and Repository Server Running on Windows with OS ENU

[Table 3–12](#) provides the code pages for the Informatica Server and Repository Server running on Windows with OS ENU.

**Table 3–12 Code Pages for Informatica Server and Repository Server on Windows with OS ENU**

Component Code Page	Code Page
Source code page	UTF-8 encoding of Unicode
Data warehouse code page	UTF-8 encoding of Unicode
Informatica Repository code page	MS Windows Latin 1 (ANSI), superset of Latin 1
Informatica Server code page	MS Windows Latin 1 (ANSI), superset of Latin 1

#### 3.10.4.2 Informatica Server and Repository Server Running on UNIX with OS ENU

[Table 3–13](#) provides the code pages for the Informatica Server and Repository Server running on UNIX with OS ENU.

**Table 3–13 Code Pages for Informatica Server and Repository Server on UNIX with OS ENU**

Component Code Page	Code Page
Source code page	UTF-8 encoding of Unicode
Data warehouse code page	UTF-8 encoding of Unicode
Informatica Repository code page	ISO 8859-1 Western European
Informatica Server code page	ISO 8859-1 Western European

### 3.10.4.3 Informatica Server on UNIX, Repository Server on Windows, both with OS ENU

Table 3–14 provides the code pages for the Informatica Server running on UNIX, the Repository Server on Windows with OS ENU.

**Table 3–14 Code Pages for Informatica Server on UNIX, Repository Server on Windows with OS ENU**

Component Code Page	Code Page
Source code page	UTF-8 encoding of Unicode
Data warehouse code page	UTF-8 encoding of Unicode
Informatica Repository code page	MS Windows Latin 1 (ANSI), superset of Latin 1
Informatica Server code page	ISO 8859-1 Western European

## 3.10.5 Source and Data Warehouse Code Pages for Code Page to Unicode

For a list of the code pages that are supported for source and data warehouse databases, see System Requirements and Supported Platforms on Oracle's Siebel SupportWeb.

The following environment variables need to be set before you begin the installation process:

- NLS\_LANG (for Oracle). For instructions, see [Section 3.10.7, "Setting the NLS\\_LANG Environment Variable for Oracle Databases"](#).
- DB2CODEPAGE (for DB2). For instructions, see [Section 3.10.8, "Setting the DB2CODEPAGE Environment Variable for DB2 Databases"](#).

The following environment variables need to be set during the configuration process:

- Informatica Server Data Movement. For instructions, see [Section 4.5.3, "How to Set Up The Informatica Server"](#).
- SiebelUnicodeDB. For instructions on Windows, see [Section 4.6.2, "How to Set Siebel UnicodeDB Environment Variables on Windows"](#). For instructions on UNIX, see [Section 5.9, "How to Set Environment Variables for the Informatica Server on UNIX"](#).

If the Informatica Server is running on Unix, then you also need to set the PMREPCODEPAGE environment variable appropriately. For example:  
PMREPCODEPAGE=MS932.

---

**Note:** Informatica Server can only run <LANG> OS based on the source code page. For example, if the source code page is Japanese then the Informatica Server should be running on JPN OS.

---

The following information uses <LANG> = JPN as an example. If you are using a language other than Japanese, replace the code page with the appropriate language.

### 3.10.5.1 Informatica Server and Repository Server Running on Windows and OS <LANG>

Table 3–15 provides the code pages for the Informatica Server and Repository Server running on Windows with OS <LANG>.

**Table 3–15 Informatica Server and Repository Server on Windows with OS <LANG>**

Component Code Page	Code Page
Source code page	MS Windows Japanese, superset of shift JIS
Data warehouse code page	MS Windows Japanese, superset of shift JIS
Informatica Repository code page	MS Windows Japanese, superset of shift JIS
Informatica Server code page	MS Windows Japanese, superset of shift JIS

### 3.10.5.2 Informatica Server and Repository Server Running on UNIX with OS <LANG>

Table 3–16 provides the code pages for the Informatica Server and Repository Server running on UNIX with OS <LANG>.

**Table 3–16 Code Pages for Informatica Server and Repository Server on UNIX with OS <LANG>**

Component Code Page	Code Page
Source code page	MS Windows Japanese, superset of shift JIS
Data warehouse code page	MS Windows Japanese, superset of shift JIS
Informatica Repository code page	MS Windows Japanese, superset of shift JIS
Informatica Server code page	MS Windows Japanese, superset of shift JIS

### 3.10.5.3 Informatica Server on UNIX, Repository Server on Windows, both with OS <LANG>

Table 3–17 provides the code pages for the Informatica Server running on UNIX and the Repository Server running on Windows, both with OS <LANG>.

**Table 3–17 Code Pages for Informatica Server on UNIX, Repository Server on Windows, OS <LANG>**

Component Code Page	Code Page
Source code page	MS Windows Japanese, superset of shift JIS
Data warehouse code page	MS Windows Japanese, superset of shift JIS
Informatica Repository code page	MS Windows Japanese, superset of shift JIS
Informatica Server code page	MS Windows Japanese, superset of shift JIS

## 3.10.6 Source and Data Warehouse Code Pages for Code Page to Code Page

For a list of the code pages that are supported for source and data warehouse databases, see System Requirements and Supported Platforms on Oracle's Siebel SupportWeb.

The following environment variables need to be set before you begin the installation process:

- NLS\_LANG (for Oracle). For instructions, see [Section 3.10.7, "Setting the NLS\\_LANG Environment Variable for Oracle Databases"](#).
- DB2CODEPAGE (for DB2). For instructions, see [Section 3.10.8, "Setting the DB2CODEPAGE Environment Variable for DB2 Databases"](#).

The following environment variables need to be set during the configuration process:

- Informatica Server Data Movement. For instructions, see [Section 4.5.3, "How to Set Up The Informatica Server"](#).
- SiebelUnicodeDB. For instructions on Windows, see [Section 4.6.2, "How to Set Siebel UnicodeDB Environment Variables on Windows"](#). For instructions on UNIX, see [Section 5.9, "How to Set Environment Variables for the Informatica Server on UNIX"](#).
- If the Informatica Server is running on UNIX, then you also need to set the PMREPCODEPAGE environment variable appropriately. For example:  
PMREPCODEPAGE=MS932.

The following information uses <LANG> = JPN as an example. If you are using a language other than Japanese, replace the code page with the appropriate language.

### 3.10.6.1 Informatica Server and Repository Server Running on Windows and OS <LANG>

[Table 3–18](#) provides the code pages for the Informatica Server and the Repository Server running on Windows, both with OS <LANG>.

**Table 3–18 Code Pages for Informatica Server and Repository Server on Windows, OS <LANG>**

Component Code Page	Code Page
Source code page	MS Windows Japanese, superset of shift JIS
Data warehouse code page	MS Windows Japanese, superset of shift JIS
Informatica Repository code page	MS Windows Japanese, superset of shift JIS
Informatica Server code page	MS Windows Japanese, superset of shift JIS

### 3.10.6.2 Informatica Server and Repository Server Running on UNIX and OS <LANG>

[Table 3–19](#) provides the code pages for the Informatica Server and the Repository Server running on UNIX, both with OS <LANG>.

**Table 3–19 Code Pages for Informatica Server and Repository Server on Windows, OS <LANG>**

Component Code Page	Code Page
Source code page	MS Windows Japanese, superset of shift JIS
Data warehouse code page	MS Windows Japanese, superset of shift JIS
Informatica Repository code page	MS Windows Japanese, superset of shift JIS
Informatica Server code page	MS Windows Japanese, superset of shift JIS

### 3.10.6.3 Informatica Server on UNIX, Repository Server on Windows, Both with OS <LANG>

[Table 3–20](#) provides the code pages for the Informatica Server running on UNIX and the Repository Server running on Windows, both with OS <LANG>.

**Table 3–20 Code Pages for Informatica Server on UNIX, Repository Server on Windows, OS <LANG>**

Component Code Page	Code Page
Source code page	MS Windows Japanese, superset of shift JIS

**Table 3–20 (Cont.) Code Pages for Informatica Server on UNIX, Repository Server on Windows, OS <LANG>**

Component Code Page	Code Page
Data warehouse code page	MS Windows Japanese, superset of shift JIS
Informatica Repository code page	MS Windows Japanese, superset of shift JIS
Informatica Server code page	MS Windows Japanese, superset of shift JIS

### 3.10.7 Setting the NLS\_LANG Environment Variable for Oracle Databases

Follow this procedure to set the NLS\_LANG environment variable for Oracle databases.

---



---

**Note:** You need to set the NLS\_LANG environment variable on the machine that runs the Informatica server, and on the machine that runs the Oracle client.

---



---

#### To set the NLS\_LANG environment variable for Oracle databases

1. Determine the NLS\_LANG value.
  - a. In the data warehouse database, run the command
 

```
SELECT * FROM V$NLS_PARAMETERS
```
  - b. Make a note of the NLS\_LANG value, which is in the format [NLS\_LANGUAGE]\_[NLS\_TERRITORY].[NLS\_CHARACTERSET].  
For example: `American_America.UTF8`
2. For Windows:
  - a. Navigate to Control Panel > System and click the Advanced tab. Click Environment Variables.
  - b. In System variables section, click New.
  - c. In the Variable Name field, enter NLS\_LANG.
  - d. In the Variable Value field, enter the NLS\_LANG value that was returned in Step 1.

The format for the NLS\_LANG value should be [NLS\_LANGUAGE]\_[NLS\_TERRITORY].[NLS\_CHARACTERSET].

For example: `American_America.UTF8`.

---

**Note:** The NLS\_LANG character set should reflect the setting of the operating system character set of the client. For example, if the database character set is AL32UTF8 and the client is running on a Windows operating system, then you should not set AL32UTF8 as the client character set in the NLS\_LANG parameter because there are no UTF-8 WIN32 clients. Instead, the NLS\_LANG setting should reflect the code page of the client. For example, on an English Windows client, the code page is 1252. An appropriate setting for NLS\_LANG is AMERICAN\_AMERICA.WE8MSWIN1252.

Setting NLS\_LANG correctly allows proper conversion from the client operating system character set to the database character set. When these settings are the same, Oracle assumes that the data being sent or received is encoded in the same character set as the database character set, so character set validation or conversion may not be performed. This can lead to corrupt data if the client code page and the database character set are different and conversions are necessary

---

3. For UNIX, set the variable as shown below:

```
setenv NLS_LANG <NLS_LANG>
```

For example: `setenv NLS_LANG American_America.UTF8.`

If your data is 7-bit or 8-bit ASCII and the Informatica Server is running on UNIX, then set `NLS_LANG <NLS_LANGUAGE>_<NLS_TERRITORY>.WE8ISO8859P1`

---

**Caution:** Make sure you set the NLS\_LANG variable correctly, as stated in this procedure, or your data will not display correctly.

---

4. Reboot the machine after creating the variable.

### 3.10.8 Setting the DB2CODEPAGE Environment Variable for DB2 Databases

Follow this procedure to set the DB2CODEPAGE environment variable for DB2 databases.

#### To set the DB2CODEPAGE environment variable for DB2 databases

1. Determine the DB2CODEPAGE value.
  - a. Connect to the Source database, using the following command:
 

```
SELECT CODEPAGE FROM SYSCAT.DATATYPES WHERE TYPENAME = 'VARCHAR'
```
  - b. Make a note of the result.
 

For example: 1208
2. For Windows:
  - a. Navigate to Control Panel > System and click the Advanced tab. Click Environment Variables.
  - b. In System variables section, click New.
  - c. In the Variable Name field, DB2CODEPAGE.
  - d. In the Variable Value field, enter the value that was returned in Step 1.

3. For UNIX, set the variable as shown below:

```
setenv DB2CODEPAGE <DB2CODEPAGE value>
```

For example: `setenv 1208`.

4. Reboot the machine after creating the variable.

## 3.11 Miscellaneous Information About Oracle Business Intelligence Applications Deployments

This section contains the following topics:

- [Section 3.11.1, "Preconfigured Mapping for the SA System Subject Area"](#)
- [Section 3.11.2, "Using Initialization Blocks"](#)
- [Section 3.11.3, "Performance Improvements for Change Capture for Incremental ETL Run"](#)
- [Section 3.11.4, "Running an ETL with a Non-English OLTP Data Source"](#)

### 3.11.1 Preconfigured Mapping for the SA System Subject Area

For Oracle's Siebel Applications customers, [Table 3–21](#) describes the preconfigured mappings for the SA system subject area. Fields that are not available in Oracle's Siebel transactional database will default to values shown in the table.

- **Overriding Defaults.** You can add user-specific values for these fields, by creating an extension table to the S\_USER table. to store the user-specific defaults for these fields. Additionally, you can change any of the default values. The metadata for the following logical table can be modified to include any physical extension table.

SA User.(User)

For instructions, refer to the documentation about configuring tables and columns for Oracle's Siebel Business Applications.

- **Setting Provider Information.** Typically, the cell phone and the fax numbers in the Oracle Business Analytics Warehouse do not contain a provider name. Therefore, the Pager will typically be a numeric value such as 555-483-3843. To append a provider to this address, use the following guidelines:
  - If the entire company has the same provider, then you can append the provider in the column mappings.
  - If users can have different providers, you need to create an extension table. For instructions, refer to the documentation about configuring tables and columns for Oracle's Siebel business applications.

**Table 3–21 Preconfigured Mappings for the User Table in the SA System Subject Area**

Logical Column	Physical Table	Expression	Comments
Cell Phone		"	It might be mapped to S_CONTACT.CELL_PH_NUM if this field contains SMTP address.
Cell Phone Priority		"	Defaults to N

**Table 3–21 (Cont.) Preconfigured Mappings for the User Table in the SA System Subject**

Logical Column	Physical Table	Expression	Comments
Display Name	S_CONTACT	"Real Time OLTP"."SIEBEL.S_ CONTACT_User.FST_NAME   '    "Real Time OLTP"."SIEBEL.S_ CONTACT_User.LAST_ NAME	First Name concatenated with Last Name
Email	S_CONTACT	EMAIL_ADDR	
Email Priority		'HNL'	Defaults to N
Email Type		'html'	Defaults to HTML
Group Name	S_RESP	NAME	
Handheld		"	Defaults to an empty string
Handheld Priority		"	Defaults to an empty string
Language		'en'	Defaults to 'en'
Locale		'en'	Defaults to 'en'
Logon	S_USER	LOGIN	
Pager		"	It could be mapped to S_ CONTACT.PAGER_PH_ NUM if this field contains SMTP address
Pager Priority		"	Defaults to N
Time Zone	S_TIMEZONE	NAME	

### 3.11.2 Using Initialization Blocks

Table 3–22 lists some of the initialization blocks common to all Oracle Business Intelligence applications and their purposes. Initialization blocks that are specific to each Oracle BI applications area are not listed here.

**Table 3–22 Initialization Blocks and Their Purposes**

Initialization Block	Purpose
Authorization	Calculates user's responsibilities from the database.
Authentication	Authenticates against the database and verifies the user exists as a operational application user.
External Metadata Strings	Gets the translated value of metadata strings for the user's locale. This initialization block is critical to Intelligence Dashboards in international deployment.
LOCALE	Sets the user's locale specification in the Oracle BI Server.
Login Properties	Gets user's login properties, such as full name, from the database. This initialization block also sets the logging level of the users. By default, the log level is 0 for all users. If you wish to generate Oracle BI query logs for all users, this initialization block should be updated by changing the default value as well as the value in the initialization SQL.
Default System Language ID	Sets the variable OLTP_LANG_ID by querying the operational application database.

**Table 3–22 (Cont.) Initialization Blocks and Their Purposes**

Initialization Block	Purpose
Organizations for Org-based Security	Queries the operational application database to get the organization membership for each user. It sets the variable ORGANIZATION.
Primary Owner ID	Sets the primary owner ID based on the user login ID.
Primary Position ID	Queries the operational application database to set the variable PRIMARY_POSTN_ID.
Warehouse Refresh Date	Sets several time-based variables such as CURRENT_YEAR.
ETL Run Date	Retrieves the ETL run date.
ETL Default Currency	Retrieves the default currency.

### 3.11.3 Performance Improvements for Change Capture for Incremental ETL Run

To maximize performance for Siebel (CRM) and Oracle E-Business Suite OLTP Systems, you need to implement the indexes specified in this section.

#### 3.11.3.1 Implementing Transactional Database (OLTP) Indexes

To implement indexes, you use SQL files that are available in the \OracleBI\dwrep directory. [Table 3–23](#) describes the SQL files that are appropriate for the specified applications.

**Table 3–23 SQL files for Siebel transactional databases**

Application Name	SQL File Name
Horizontal Application	PerfIndex_Horizontal.sql
Industry Application	PerfIndex_Industry.sql

The SQL files generate indexes on all S\_.\* tables that are being used by the out-of-the-box applications.

---

**Note:** If you move from a test to a production environment, you need to drop the indexes and re-create them in the production environment.

---

#### 3.11.3.2 An Example Of Change Capture SQL And Required Indexes

Change capture SQL generates the following SQL:

```

Insert into S_ETL_I_IMG_XX (ROW_ID, LAST_UPD)
AS
SELECT ROW_ID, LAST_UPD, MODIFICATION_NUM
From
S_XXX
WHERE LAST_UPD > 'LAST_REFRESH_DATE - PRUNE_DAYS'
AND NOT EXISTS
(
SELECT 'X' FROM S_ETL_R_IMAGE
WHERE S_ETL_R_IMAGE.ROW_ID = S_XXX.ROW_ID AND
S_ETL_R_IMG_XX.MODIFICATION_NUM = S_XXX.MODIFICATION_NUM AND
S_ETL_R_IMG_XX.LAST_UPD = S_XXX.LAST_UPD
)

```

Table 3–24 shows the indexes that, based on the SQL above, are created on the S\_CONTACT table by the SQL generation scripts.

**Table 3–24 Indexes Created on the S\_CONTACT Table in Siebel transactional database**

Index	Index Column
S_CONTACT_W1	LAST_UPD, ROW_ID_MODIFICATION_NUM
S_CONTACT_W11	LAST_UPD

### 3.11.4 Running an ETL with a Non-English OLTP Data Source

To run an ETL with a non-English OLTP Data Source, you must make a copy of an appropriate source system container, and configure the language, country, and continent parameters.

To run an ETL with a non-English OLTP Data Source:

1. In Data Warehouse Administration Console, choose File, then New Source System to display the New Source System Container dialog.
2. Select the **Create as a Copy of Existing Container** radio button.
3. From the **Existing Containers** drop down list, select the container that you want to copy, then click OK.
4. Display the Design view.
5. Ensure that you have selected the correct container from the *containers* drop down list.
6. Select the Source System Parameters tab.
7. Use the Edit tab below the list of Source System Parameters to change the value of the following parameters in the list:
  - \$\$DFLT\_LANG (Default Language) – For example, for a Japanese data source, change this value to JPN.
  - (optional) \$\$DFLT\_COUNTRY (Default Country)
  - (optional) \$\$DFLT\_CONTINENT (Default Continent)

**Tip:** To find out the value to specify for the \$\$DFLT\_LANG parameter, issue the query 'select VAL from S\_SYS\_PREF where SYS\_PREF\_CD=<ETL value>' against the OLTP database. For example, to find out the default ETL language, issue the following command:

```
select VAL from S_SYS_PREF where SYS_PREF_CD='ETL Default Language' ;
```

8. Create a new ETL plan for the new source system container, and edit its parameters, as follows:
  - a. Display the Execute tab.
  - b. Click the Execution Plans sub-tab.
  - c. Click New to create a new blank execution tab and use the sub-tabs below (for example, Subject Areas, Parameters, Ordered Tasks) to specify the execution plan details.
  - d. Click Save.
9. Click Run Now to run the new ETL plan (or use the Schedule tab to specify when you want to run the new execution plan).



# Part II

---

## Installation and Post-Installation Steps For BI Applications

---

Part II explains how to install Oracle BI Applications, and contains the following chapters:

- [Chapter 4, "Installing and Configuring Oracle BI Applications on Windows"](#)
- [Chapter 5, "Installing Oracle BI Applications on Unix"](#)

---

**Note:** For a high level road map for installation, configuration, and customization steps for Oracle BI Applications, see [Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle BI Applications"](#).

---



---

---

# Installing and Configuring Oracle BI Applications on Windows

This chapter provides instructions for installing and configuring the Oracle Business Intelligence Applications components in the Windows environment. It contains the following main topics:

- Section 4.1, "Installation and Configuration Process Task List for Windows"
- Section 4.2, "How to Install the Java SDK"
- Section 4.3, "How to Install Oracle Business Intelligence Applications (Windows)"
- Section 4.4, "How to Install Informatica PowerCenter and Required Patches (Windows)"
- Section 4.5, "How to Set Up the Informatica Server and Informatica Repository Server"
- Section 4.6, "How to Set Environment Variables on Windows"
- Section 4.7, "How to Copy Hibernate Libraries"
- Section 4.8, "How to Configure the DAC Client"
- Section 4.9, "How to Create the Required Databases"
- Section 4.10, "How to Create the DAC Metadata Repository"
- Section 4.11, "How to Configure the DAC Metadata Repository"
- Section 4.12, "How to Configure the Oracle Business Analytics Warehouse Database"
- Section 4.13, "How to Register Informatica Servers in the DAC Client"
- Section 4.14, "How to Configure and Manage the Informatica Repository"
- Section 4.15, "How to Set the Code Page File Property for Unicode Environments on Teradata Databases"
- Section 4.16, "How to Set Up the HOSTS File for Teradata Installations on Windows"
- Section 4.17, "How to Verify the Informatica Repository Configuration"
- Section 4.18, "How to Configure the DAC Server"
- Section 4.19, "How to Start the Informatica Server"
- Section 4.21, "How to Create Stored Procedures for DB2-UDB"

- [Section 4.22, "Teradata-Specific Installation and Configuration Checklist \(Windows\)"](#)
- [Section 4.23, "Upgrade Issues for Oracle BI Applications"](#)

## 4.1 Installation and Configuration Process Task List for Windows

The Oracle BI Applications installation and configuration process consists of the following tasks. Complete each of these tasks in the order listed below.

1. Install Java SDK (if not already installed as part of the Oracle BI infrastructure installation (for more information, see [Section 4.2, "How to Install the Java SDK"](#)).
2. Install Oracle Business Intelligence Applications (for more information, see [Section 4.3, "How to Install Oracle Business Intelligence Applications \(Windows\)"](#)).
3. Install Informatica PowerCenter (for more information, see [Section 4.4, "How to Install Informatica PowerCenter and Required Patches \(Windows\)"](#)).
4. Set up the Informatica components (for more information, see [Section 4.5, "How to Set Up the Informatica Server and Informatica Repository Server"](#)):
  - a. [Section 4.5.2, "How to Set Up Informatica Data Code Pages Validation"](#).
  - b. [Section 4.5.3, "How to Set Up The Informatica Server"](#).
  - c. [Section 4.5.4, "How to Set Up The Informatica Repository Server"](#).
  - d. [Section 4.5.6, "How to Start the Informatica Services"](#).
5. Set the Windows environment variables (for more information, see [Section 4.6, "How to Set Environment Variables on Windows"](#)).
6. Install the required Hibernate libraries (for more information, see [Section 4.7, "How to Copy Hibernate Libraries"](#)).
7. Configure the DAC Client (for more information, see [Section 4.8, "How to Configure the DAC Client"](#)):
  - a. [Section 4.8.1, "How to Configure the DAC Config.bat File"](#).
  - b. [Section 4.8.2, "How to Install JDBC Drivers for DAC Database Connectivity"](#).
8. Creating the required OLTP and OLAP databases (for more information, see [Section 4.9, "How to Create the Required Databases"](#)).
9. Create the DAC Metadata Repository (for more information, see [Section 4.10, "How to Create the DAC Metadata Repository"](#)).
10. Configure the DAC Metadata Repository (for more information, see [Section 4.11, "How to Configure the DAC Metadata Repository"](#)):
  - a. [Section 4.11.1, "How to Set Up DAC System Properties"](#).
  - b. [Section 4.11.2, "How to Specify Transactional and Data Warehouse Data Sources in the DAC"](#).
  - c. [Section 4.11.3, "How to Configure Email Recipients in the DAC Client"](#).
  - d. [Section 4.11.4, "How to Activate Join Indices for Teradata Databases"](#).
11. Configure the Oracle Business Analytics Warehouse (for more information, see [Section 4.12, "How to Configure the Oracle Business Analytics Warehouse Database"](#)):
  - a. [Section 4.12.1, "How to Create the SSE Role"](#).

- b. [Section 4.12.2, "How to Create Data Warehouse Tables"](#).
  - c. [Section 4.12.3, "How to Create Data Warehouse Tables on a Teradata Database"](#).
12. Register the Informatica servers in the DAC (for more information, see [Section 4.13, "How to Register Informatica Servers in the DAC Client"](#)).
13. Configure the Informatica Repository (for more information, see [Section 4.14, "How to Configure and Manage the Informatica Repository"](#)):
  - a. [Section 4.14.2, "How to Register the Informatica Repository Server"](#).
  - b. [Section 4.14.3, "How to Load the Pre-Built Repository Into Informatica"](#).
  - c. [Section 4.14.4, "How to Configure the Informatica Repository in Workflow Manager"](#).
  - d. [Section 4.14.5, "How to Stop and Start the Informatica Repository \(Optional\)"](#).
  - e. [Section 4.14.6, "How to Back Up and Restore the Informatica Repository \(Optional\)"](#).
14. Verify the Informatica Repository configuration (for more information, see [Section 4.17, "How to Verify the Informatica Repository Configuration"](#)).
15. Start the Informatica Server (for more information, see [Section 4.19, "How to Start the Informatica Server"](#)).

---

---

**Note:** You must perform the following steps for each source database type.

---

---

16. Configure the DAC Server (for more information, see [Section 4.18, "How to Configure the DAC Server"](#)):
  - a. [Section 4.18.1, "How to Configure the Connection Between the DAC Server and DAC Repository"](#).
  - b. [Section 4.18.2, "How to Configure Email in the DAC Server \(Optional\)"](#).
  - c. [Section 4.18.3, "How to Set DAC Source System Parameters"](#).
  - d. [Section 4.18.4, "How to Start the DAC Server"](#).
17. Update the transactional database schema definitions (for more information, see [Section 4.20, "How to Update Siebel Transactional Database Schema Definitions"](#)):
  - a. [Section 4.20.2, "How to Apply Siebel \(CRM\) and Oracle E-Business Suite Schema Changes For Oracle, DB2/UDB, MSSQL"](#).
  - b. [Section 4.20.3, "How to Apply CRM Schema Changes For DB2 on 390 or z/os"](#).
  - c. [Section 4.20.4, "How to Verify the Siebel \(CRM\) and Oracle E-Business Suite Schema Changes"](#).
  - d. [Section 4.20.5, "About Delete Triggers"](#).

Other database specific tasks:

- If you are using Teradata as a target OLAP database, set the Code Page File Property (for more information, see [Section 4.15, "How to Set the Code Page File Property for Unicode Environments on Teradata Databases"](#)).

- If you are using Teradata as a target OLAP database, set up the HOSTS file (for more information, see [Section 4.16, "How to Set Up the HOSTS File for Teradata Installations on Windows"](#)).
- If you are using Teradata as a target OLAP database, perform the Teradata-specific configuration tasks (for more information, see [Section 4.22, "Teradata-Specific Installation and Configuration Checklist \(Windows\)"](#)).
- If you are using DB2-UDB, create stored procedures (for more information, see [Section 4.21, "How to Create Stored Procedures for DB2-UDB"](#)).

## 4.2 How to Install the Java SDK

This step can be optional or mandatory, depending on where you want to install the DAC server and the DAC client. Java SDK must be installed on the machines where both the DAC server and DAC client are installed. If a machine has the Oracle Business Intelligence infrastructure installed, the Java SDK will already be installed on that machine.

If required, you can have other versions of SDK installed on the same server to support other applications. For example, you might have Java SDK version 1.5 installed for the DAC, and Java SDK version 1.4.2 installed for a different application that is not supported on Java SDK version 1.5.

---

---

**Note:** For information about which version of the Java SDK is supported, see *System Requirements and Supported Platforms for Oracle Business Intelligence Applications*.

---

---

### To install Java SDK

1. Download the Java SDK from the Sun Web site, and install it on the machine where both the DAC server and DAC client are installed.

For example (on Windows), C:\j2sdk1.5.

---

---

**Note:** Make sure that there are no spaces in the directory path. If there are spaces in the directory path, the DAC will not start (the \OracleBI\DAC\startclient.bat file will fail).

---

---

## 4.3 How to Install Oracle Business Intelligence Applications (Windows)

This section explains how to install the Oracle BI Applications files using the Oracle BI Applications Installation Wizard, and contains the following sections:

- [Section 4.3.1, "Prerequisites to Running The Oracle Business Intelligence Applications Installation Wizard"](#)
- [Section 4.3.2, "How to Run The Oracle Business Intelligence Applications Installation Wizard"](#)

### 4.3.1 Prerequisites to Running The Oracle Business Intelligence Applications Installation Wizard

Before you install Oracle Business Intelligence Applications on a machine, you must first install the Oracle Business Intelligence infrastructure on that machine. For

instructions on installing the Oracle Business Intelligence infrastructure, see *Oracle Business Intelligence Platform Installation and Configuration Guide*.

If you are using Oracle client software for connection to an Oracle database, you must make sure that the Oracle client software has been installed before the latest verified version of Java SDK is installed when you install Oracle Business Intelligence. If you install the Oracle client software after Oracle Business Intelligence, you might change the Java SDK version for Oracle Business Intelligence, which might result in runtime errors in Oracle Business Intelligence.

### 4.3.2 How to Run The Oracle Business Intelligence Applications Installation Wizard

When you run the Oracle BI Applications Installation Wizard, the Oracle BI Applications files are installed into the existing Oracle BI infrastructure directory (for example, <DRIVE>:\OracleBI\).

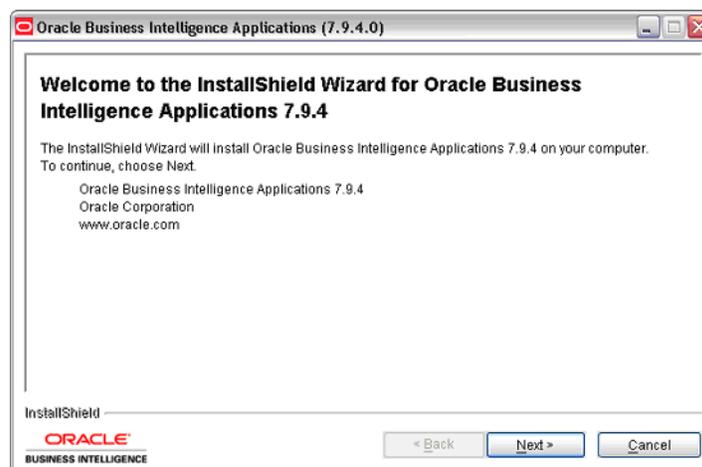
---

**Note:** During the installation, if you are prompted to create or upgrade the DAC repository, choose Yes.

---

#### To install Oracle BI Applications on Windows

1. Access the installation files on the installation CD-ROM, and then run the program setup.exe to display the Welcome page.

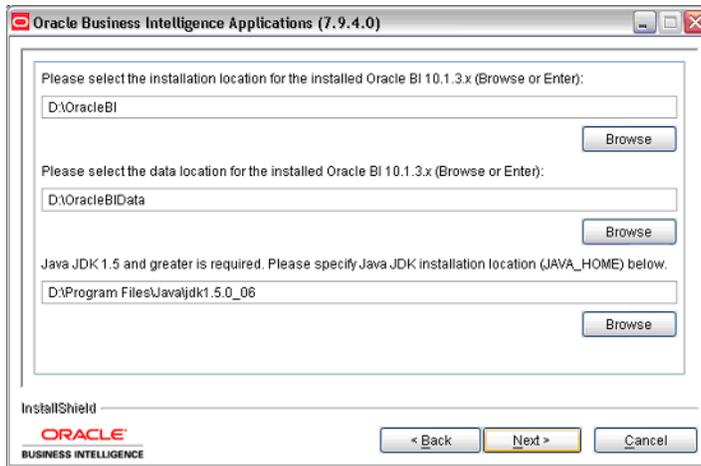



---

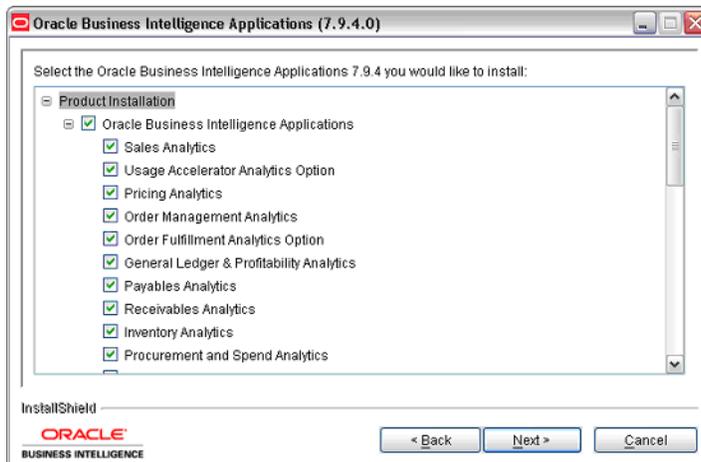
**Note:** To run the installer in console (or text) mode, run the command setup.exe -console. You do not see following dialogue screens in console installation mode. Instead, you enter input as plain text in the terminal window when prompted.

---

2. Click Next to display the Oracle BI infrastructure and Java SDK directory location page.

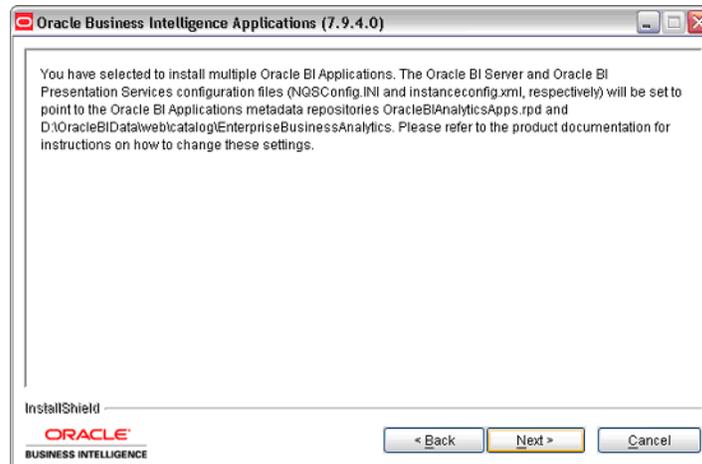


3. Enter or browse for the location for the Oracle BI infrastructure (for example, <DRIVE>:\OracleBI\), the Oracle BI infrastructure data (for example, <DRIVE>:\OracleBIData \), and the Java SDK (for example, <DRIVE>:\jdk1.5).
4. Click Next to display the Select Oracle Business Intelligence Applications page.
5. At the Select Oracle Business Intelligence Applications page, select the applications that you want to install, then click Next.

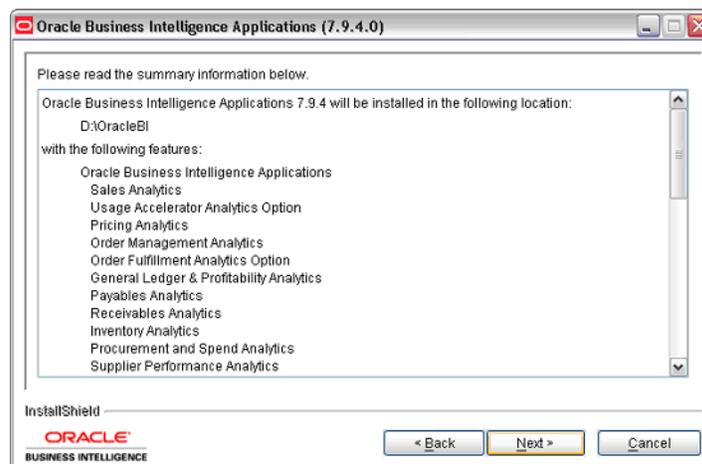


Once you have installed applications, you can deploy one or more of them selectively when you create the DAC metadata repository in a later step (for more information, see [Section 4.10, "How to Create the DAC Metadata Repository"](#)).

6. If you selected more than one application on the Select Oracle Business Intelligence Applications page, click Next at the informational page.



- At the summary page, review the summary information and click Next to start the installation.



The installer installs the Oracle BI Applications directories and files in the Oracle BI infrastructure installation (for example, <DRIVE>\OracleBI\), and installs the RPD Repository.

---

**Note:** Even when the progress bar on the installer reports 100% complete, you must wait until the **Finish** button is displayed.

---

- Click Finish.

**Tip:** Look in the <DRIVE>\OracleBI\Document\version\_apps.txt file in the Oracle BI infrastructure directory to check that you have installed the correct version of Oracle BI Applications.

## 4.4 How to Install Informatica PowerCenter and Required Patches (Windows)

To install Informatica PowerCenter:

- Install Informatica PowerCenter (for more information, see [Section 4.4.1, "How to Install Informatica PowerCenter"](#)).

2. Install Informatica EBF Patch 2 (for more information, see [Section 4.4.2, "How to Install the Informatica EBF Patch 2"](#)).
3. If you are using a Teradata database, install Informatica Patch 1 (for more information, see [Section 4.4.3, "\(Teradata specific\) How to Install Informatica Patch1"](#)).

For a list of the main files that are installed, see [Section 5.3, "How to Install Oracle BI Applications on UNIX"](#).

If you first want to find out more about the versioned Informatica repository that is included in Oracle Business Intelligence Applications, see [Section 4.14.1, "About the Versioned Informatica Repository"](#).

### 4.4.1 How to Install Informatica PowerCenter

Follow this procedure to install Informatica PowerCenter. The

The table below lists the default administrator usernames and passwords for Informatica:

**Table 4–1 Default Informatica Usernames and Passwords**

Component	Username	Password
Informatica Repository Server	Administrator	Administrator
Informatica Server	Administrator	Admin

---

**Note:** Even if you decide to run the Informatica server and the DAC server on two separate machines, you must still install Informatica on the DAC server machine, because the DAC server requires the Informatica libraries.

---

#### To install Informatica PowerCenter on Windows

1. Access the installation files on the installation CD-ROM, and locate the appropriate operating system directory.

For example, for Windows, use the directory `Informatica_7.1.4_PowerCenter_CD_for_Windows_and_Linux`.

---

**Note:** The Informatica installation directory is located in the `\ThirdPartySoftware` directory on the Oracle BI Applications Installation CD.

---

2. Run the program `launch.exe`.
3. The installation wizard window appears and prompts you through each screen, as shown in the following table.

To continue to the next screen, click `Next`. To return to a previous screen, click `Back`.

Screen	Your Action	Notes
Welcome	Click <code>Next</code> .	

Screen	Your Action	Notes
Customer Information	<p>Enter the user name, company name, and Product license key.</p> <p><b>Note:</b> Do not enter the Connectivity license key or Options license key during the product installation.</p>	<p>The Informatica license provides three different license keys:</p> <p><b>Product.</b> Allows you to install the Informatica Server and Repository Server. Enter during the product installation and the procedures <a href="#">Section 4.5, "How to Set Up the Informatica Server and Informatica Repository Server"</a> and <a href="#">Section 4.14.3, "How to Load the Pre-Built Repository Into Informatica"</a>.</p> <p><b>Connectivity.</b> Allows you to access sources and targets. Enter during the product installation and the procedures <a href="#">Section 4.5, "How to Set Up the Informatica Server and Informatica Repository Server"</a> and <a href="#">Section 4.14.3, "How to Load the Pre-Built Repository Into Informatica"</a>.</p> <p><b>Options.</b> This Team Based Development License key allows you to access specific options, such as Team-Based Development, Server Grid, and Partitioning. Enter during the product installation and the procedures <a href="#">Section 4.5, "How to Set Up the Informatica Server and Informatica Repository Server"</a> and <a href="#">Section 4.14.3, "How to Load the Pre-Built Repository Into Informatica"</a>.</p>
License Agreement	Select "I accept..." and click Next.	
Select Components and Destination directory	<p>Select the following components:</p> <ul style="list-style-type: none"> <li>■ Informatica PowerCenter 7.1.4 OEM for Oracle - Client</li> <li>■ Informatica PowerCenter 7.1.4 OEM for Oracle - Server</li> <li>■ Informatica PowerCenter 7.1.4 OEM for Oracle - Repository Server</li> </ul> <p>Browse for and select the appropriate destination directory, and then click Next.</p>	
Program Folder	Select a program folder or accept the default, and then click Next.	<b>Note:</b> If you are using a Teradata database, make sure there are no spaces in the Informatica Server directory name or the directory path. The default directory contains spaces that you must remove manually.
Review Settings	Review the selected components and destination directory information, and then click Next.	The Informatica components are now installed on the machine.
Setup	Click OK.	This screen informs you that you must specify Productivity and Options license keys in later steps.

Screen	Your Action	Notes
Setup Type	Remove the check marks from the Configure Informatica Server and Configure Informatica Repository Server check boxes.  (Optional) Select the View Readme check box to view the Read Me file.  Click Next.	You will configure the Informatica Server and Repository Server in later steps.
Wizard Complete	Click Finish.	

---

**Note:** Depending on how your Windows machine is set up, you might need to start the Informatica services manually (for more information, see [Section 4.5.6, "How to Start the Informatica Services"](#)).

---

## 4.4.2 How to Install the Informatica EBF Patch 2

Follow this procedure to install the required Informatica patches.

### To install the Informatica patches on Windows

1. Locate the \Third\_Party\_Software\Informatica\ directory on the Oracle BI Applications CD-ROM.
2. Locate the directory named Patch2, and install the patch by following the instructions in the EBF108240.html readme file.

---

**Note:** The instructions provided in the EBF108240.html file reference the regedit path for Informatica PowerCenter 7.1.4 on Windows machines as:

```
\hkey_local_
machine\system\currentcontrolset\services\powermart\
configuration
```

Depending on your machine and Informatica 7.1.4 installation, the path might be:

```
\hkey_local_
machine\system\currentcontrolset\services\powermart\
parameters\configuration
```

---

## 4.4.3 (Teradata specific) How to Install Informatica Patch1

If you are using Teradata as the target database, locate the directory named Patch1 and install the patch by following the instructions in the Readme.doc file.

## 4.5 How to Set Up the Informatica Server and Informatica Repository Server

After the Informatica Server is installed, you must configure the Informatica components by following the steps in this section.

You must configure Informatica Server and Informatica Repository Server before you can start them, by doing the following:

- Set up the Informatica Data Code Pages. For more information, see [Section 4.5.2, "How to Set Up Informatica Data Code Pages Validation"](#).
- Set up the Informatica Server in Informatica PowerCenter. For more information, see [Section 4.5.3, "How to Set Up The Informatica Server"](#).
- Set up the Informatica Repository Server in Informatica PowerCenter. For more information, see [Section 4.5.4, "How to Set Up The Informatica Repository Server"](#).
- Copy source files and lookup files from the Oracle BI Applications installation directory to the Informatica directory. For more information, see [Section 4.5.5, "How to Copy Source Files and Lookup Files"](#).
- After you have created the Informatica repository (for more information, see [Section 4.14, "How to Configure and Manage the Informatica Repository"](#)), start the Informatica Services. For more information, see [Section 4.5.6, "How to Start the Informatica Services"](#).
- Create the required ODBC Database Connections. For more information, see [Section 4.5.7, "How to Create ODBC Database Connections"](#).
- If you need to migrate the Informatica repository, you must initialize the Informatica sequence generator for incremental runs (for more information, see [Section 4.5.1, "About migrating the Informatica repository"](#)).

Make note of the settings you enter in the Server tab and Repository tab of the Configure Informatica Service window; you will need to enter the same settings in other configuration steps.

**Tip:** Keep a record of the Informatica Server details that you specify here, and use the same details when you register the Informatica servers in the DAC (for more information, see [Section 4.13, "How to Register Informatica Servers in the DAC Client"](#)).

---

**Note:** Throughout this guide, the name Oracle\_BI\_DW\_Base is used for the Informatica repository (the file \dwrep\Informatica\Repository\Oracle\_BI\_DW\_Base.rep). However, you can use a different name if required, and have more than one version of this file if you have more than one development environment.

---

### 4.5.1 About migrating the Informatica repository

If you migrate the Informatica repository from one environment to another environment (for example, from test to production), you need to initialize the Informatica sequence generator for incremental runs, as follows:

1. Navigate to the directory \OracleBI\dwrep\Upgrade\DbScripts\*<database type>*.  
For example, for an Oracle database, navigate to \OracleBI\dwrep\Upgrade\DbScripts\Oracle.
2. Open the Reset\_infa\_seq\_gen.bat file in a text editor.
3. Edit the parameter values in the section marked 'Login Settings for Informatica server and OLAP database' (for example, TARGET\_USERID\_PWD, INFA\_USERID\_PWD).
4. Save the file.

For more information about upgrading Oracle BI Applications, see *Oracle Business Intelligence Applications Upgrade Guide*.

### 4.5.2 How to Set Up Informatica Data Code Pages Validation

After installing the Informatica Server, Repository Server, and client tools, make sure that the ValidateDataCodePages parameter is set to 'No' in the powmart.ini file located in the \Informatica PowerCenter 7.1.4\Client directory, as follows:

```
[Code Pages]
ValidateDataCodePages=No
```

---

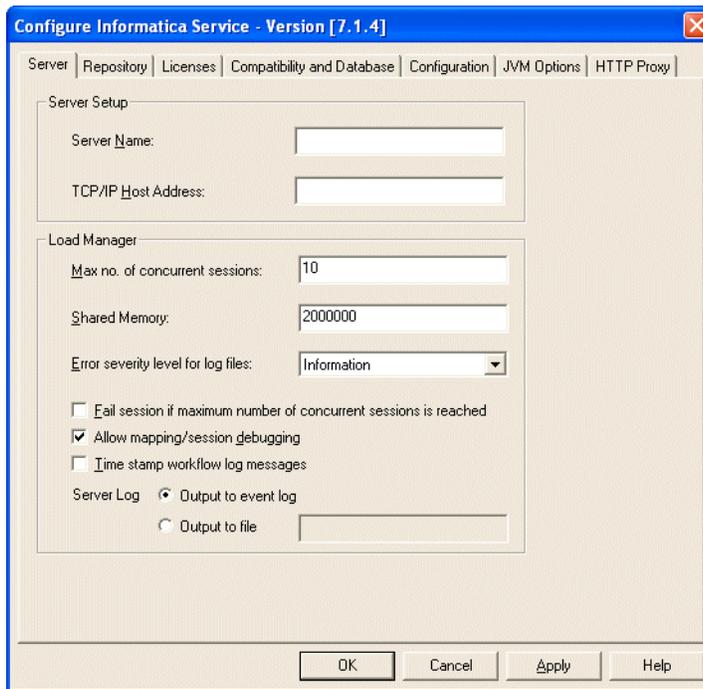
**Note:** If the powmart.ini file does not contain a ValidateDataCodePages parameter, add this parameter as specified above.

---

### 4.5.3 How to Set Up The Informatica Server

To set up the Informatica Server in Informatica PowerCenter

1. Navigate to Programs, then Informatica PowerCenter 7.1.4, then Informatica PowerCenter 7.1.4 OEM for Oracle - Server, then Informatica Server Setup.
2. In the Informatica Server Setup Application - Options window, select Configure Informatica Service, and then click Continue.



3. In the Server tab, enter the following:

Field	Description
Server Name	A logical name for the Informatica Server. Oracle recommends that you enter ORACLE_BI_DW_SERVER.
TCP/IP Host Address	The IP address or name of the machine on which you plan to run the Informatica Server.

Field	Description
Max no. of concurrent sessions	From 1 to 15 sessions.
Shared Memory	The default is 2,000,000. You should specify 200,000 for each session specified in the 'Max no. of concurrent sessions' field. For example, if you specify 5 in the 'Max no. of concurrent sessions' field, you should specify 1,000,000 in the Shared Memory field.

4. In the Repository tab, enter the following:

Field	Description
Repository Name	A logical name for the Informatica repository. Specify the name as follows: <ul style="list-style-type: none"> <li>■ For Teradata databases, Oracle recommends that you enter Oracle_BI_DW_Teradata.</li> <li>■ For all other databases, Oracle recommends that you enter Oracle_BI_DW_Base.</li> </ul>
Repository User	The account used to access the repository. Oracle recommends that you use Administrator as the username.
Repository Password	The password you use to connect to the Repository Server from the Repository Manager (the default password is Administrator).
Repository Server Host Name	The host machine name where the Informatica Repository Server is installed.
Repository Port Number	The port number the Repository Server uses to connect to repository client applications (the default is 5001). Oracle recommends that you use the default value.

5. In the Licenses tab:

- a. Leave the default License File Name as `pm.lic`.
- b. If you have a Connectivity license key, enter it in the **with Key** field, and click Update.
- c. If you have a Team Based Development license key, enter it in the **with Key** field, and click Update.

If you do not specify a Connectivity license key or Options license key (the Team Based Development license key) at this point, you can specify them when you load the pre-built repository into Informatica in a later step (for more information, see [Section 4.14.3, "How to Load the Pre-Built Repository Into Informatica"](#)).

6. In the Compatibility and Database tab:

- a. Make sure the "Treat CHAR as CHAR on read" option is selected.
- b. Make sure the maximum number of database connections is 100.

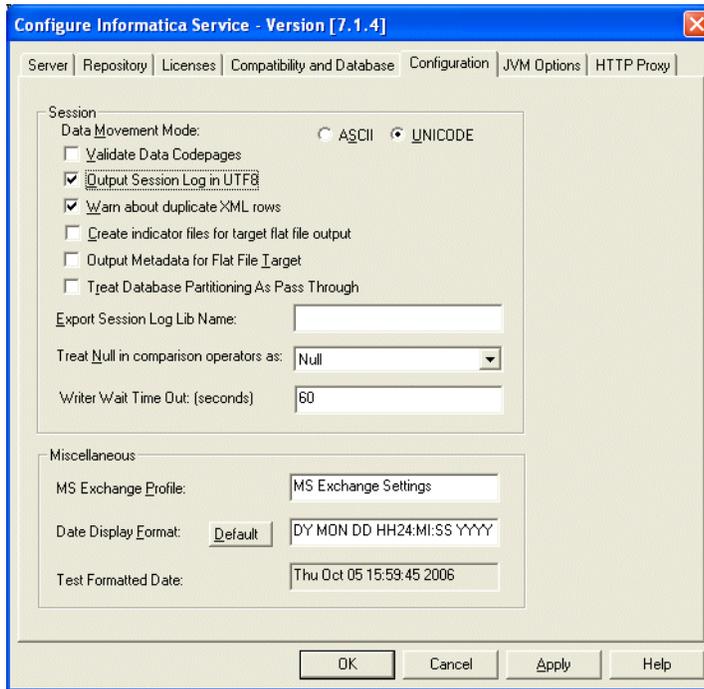
7. In the Configuration tab, set the Data Movement Mode.

---

**Note:** You must run the Informatica Server in Unicode mode if your source data contains multibyte or ISO 8859-1 (8-bit ASCII) data.

---

- a. Select the ASCII or UNICODE option.
- b. If you select the UNICODE option, remove the check from the Validate Data Codepages check box, and check the Output Session Log in UTF8 check box.



- c. Leave the default values in the remaining fields.
8. In the JVM Options tab, leave the default values in all fields.
9. Click OK in the Configure Informatica Service dialog box to complete the configuration of the Informatica Server service.

If you are using a Teradata database, you cannot start the Informatica Server until you have completed the procedures in the section [Section 4.15, "How to Set the Code Page File Property for Unicode Environments on Teradata Databases"](#).

#### 4.5.4 How to Set Up The Informatica Repository Server

To set up the Informatica Repository Server in Informatica PowerCenter

1. Navigate to Programs, then Informatica PowerCenter 7.1.4, then Informatica PowerCenter 7.1.4 OEM for Oracle - RepServer, then Informatica Repository Server Setup.

2. In the Configure Repository Server dialog box, enter the following information:

Field	Description
Server Port Number	<b>Note:</b> The port number the Repository Server uses to connect to repository client applications. By default, this value is set to 5001. It is recommended that you use the default value. This value must match the value you entered in <a href="#">Section 4.5, "How to Set Up the Informatica Server and Informatica Repository Server"</a> .
Administrator Password	The password you use to connect to the Repository Server from the Repository Manager. (the default password is Admin). <b>Note:</b> This value must match the value you entered in <a href="#">Section 4.5, "How to Set Up the Informatica Server and Informatica Repository Server"</a> .

3. Leave the default values in the remaining fields, and click OK.

#### 4.5.5 How to Copy Source Files and Lookup Files

You need to copy source files and lookup files from the Oracle BI Applications installation directory to the Informatica directory.

##### To copy source files and lookup files:

1. Copy the source files in OracleBI\dwrep\Informatica\SrcFiles to Informatica PowerCenter 7.1.4\Server\SrcFiles.
2. Copy the lookup files in OracleBI\dwrep\Informatica\LkpFiles to Informatica PowerCenter 7.1.4\Server\LkpFiles.

#### 4.5.6 How to Start the Informatica Services

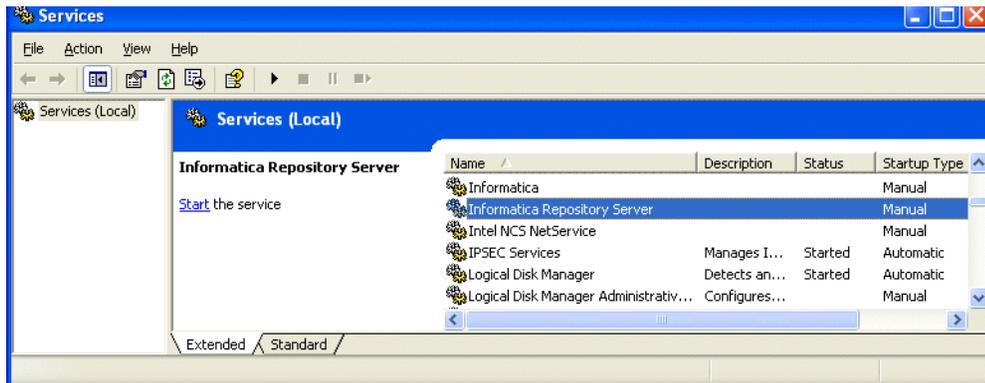
After you have installed Informatica PowerCenter, you might need to start the Informatica services manually. The Informatica services must be running to enable you to access the Oracle Business Analytics Warehouse using the Informatica tools.

You need to configure the Informatica repository before you start the Informatica services (for more information, see [Section 4.14, "How to Configure and Manage the Informatica Repository"](#)).

#### To start the Informatica Services

1. Display the Windows Services dialog.

For example, on Windows XP, display the Control Panel, then double click the Services icon to display the Services dialog.



2. Start the Informatica Repository Server service.
3. Start the Informatica service.

If you start the Informatica service and the Informatica Repository Server service is not running, the Informatica service will fail.

### 4.5.7 How to Create ODBC Database Connections

The Informatica Server uses native drivers to connect to the source and target databases to move data. Depending on the source and target database type, you can optionally configure the Informatica Server to use ODBC drivers supplied by your database vendors to connect to the source and target databases. The Informatica Repository Server uses native drivers to connect to the repository database.

You also need to specify an ODBC connections when you create the data warehouse tables in the DAC in a later configuration task (for more information, see [Section 4.12.2, "How to Create Data Warehouse Tables"](#)). In the case of Oracle database, use the Oracle Merant ODBC Driver that comes with the Oracle BI Applications install. For all other databases, you should use ODBC drivers supplied by your database vendor.

#### To create database connections for DB2 installations

1. Using the DB2 Client Configuration Assistant, create a database connection to the Oracle Business Analytics Warehouse, the transactional database, and the Informatica repository.

---

**Note:** If you use the DB2 Client Configuration Assistant to create database connections, you can omit step 2, because the DB2 Client Configuration Assistant automatically creates System DSNs (default behavior).

---

2. If necessary, in Windows, in the System DSN tab of the ODBC Data Source Administrator, create an ODBC connection to the Oracle Business Analytics Warehouse and transactional database using an ODBC driver.
3. Test the connections to make sure they work.

#### To create database connections for Oracle installations

1. Using Net8 Assistant or Oracle Net Service, create a native connect string (net service name) to the Oracle Business Analytics Warehouse, the transactional database, and the Informatica repository.
2. In Windows, in the System DSN tab of the ODBC Data Source Administrator, create an ODBC connection to the transactional database using the Oracle Merant ODBC driver that is supplied with Oracle BI Applications.

For example, you might create a database connection called Connect\_to\_OLTP.

3. In Windows, in the System DSN tab of the ODBC Data Source Administrator, create an ODBC connection to the Oracle Business Analytics Warehouse and transactional database using the Informatica-supplied Merant Closed driver.

For example, you might create a database connection called Connect\_to\_OLAP and specify 'Connect\_to\_OLAP' in the **ODBC Data Source** field in the Data Warehouse Configuration Wizard when you create the data warehouse tables in the DAC in a later configuration task (for more information, see [Section 4.12.2, "How to Create Data Warehouse Tables"](#)).

4. Test the connections to make sure they work.

#### To create database connections for SQL Server installations

1. In Windows, in the System DSN tab of the ODBC Data Source Administrator, create the following:
  - an ODBC connection to the Oracle Business Analytics Warehouse
  - an ODBC connection to the transactional database

---

**Note:** Select SQL Server as the ODBC driver.

---

2. Test the connections to make sure they work.

---

**Note:** When you use the ODBC Data Source Administrator to create a database connection, make sure that you select the SQL Server authentication option using a login ID and password entered by the user

---

#### To create database connections for Teradata installations

1. In Windows, in the System DSN tab of the ODBC Data Source Administrator, create an ODBC connection for the Teradata database.
2. Set the following parameters:

Field	Enter
DateFormat	AAA
SessionMode	ANSI

Field	Enter
NoScan	Yes

For Unicode environments, in the Teradata ODBC Driver Advanced Options dialog box, set the Character Set parameter to UTF8.

## 4.6 How to Set Environment Variables on Windows

This section includes instructions for setting the following environment variables:

- [Section 4.6.1, "How to Set the PATH Environment Variable for the Informatica Server and Informatica Repository Server"](#)
- [Section 4.6.2, "How to Set Siebel UnicodeDB Environment Variables on Windows"](#)
- [Section 4.6.3, "How to Set the Bulk Load Registry for DB2/390 Databases"](#)

---

**Note:** You need to reboot your machine after setting the environment variables for the change to take effect.

---

### 4.6.1 How to Set the PATH Environment Variable for the Informatica Server and Informatica Repository Server

The DAC server uses the pmcmd program to run the Informatica workflows on the Informatica servers. This requires the path to the \bin\ directory for the Informatica server and repository server to be added to the Windows system PATH environment variable.

#### To set the PATH environment variables for Informatica

- In the Windows System Properties window, add the path for the Informatica Server bin directory (for example, C:\Program Files\Informatica PowerCenter 7.1.4\Server\bin) to the PATH environment variable.
- In the Windows System Properties window, add the path for the Informatica Repository Server bin directory (for example, C:\Program Files\Informatica PowerCenter 7.1.4\RepositoryServer\bin) to the PATH environment variable.

### 4.6.2 How to Set Siebel UnicodeDB Environment Variables on Windows

If you are using the Unicode character data mode, you need to set the Siebel UnicodeDB environment variable.

#### To set Siebel UnicodeDB for Windows

1. In Windows, click Start, then Run.
2. In the Open field, enter `regedit`.
3. Navigate to HKEY\_LOCAL\_MACHINE, then SYSTEM, then CurrentControlSet, then Services, then PowerMart, then Parameters, then Configuration.
4. On the right window panel, right-click and select New, then String Value.
5. Rename the new string value SiebelUnicodeDB.
6. Double-click SiebelUnicodeDB.

7. In the Value data field, enter the connection string for your transactional database followed by the connect string for your data warehouse database, using the following format:

```
[user_OLTP]@[connectString_OLTP] [user_OLAP]@[ConnectString_OLAP].
```

For example, oltp.oltp@db204007.host.com olap.olap@db204008.host.com.

- Always leave a space between the connection strings.
- For MSSQL, use the system DSN name for [connectString\_OLTP] and [connectString\_OLAP].

---



---

**Note:** You must enter the user names and connect strings in the same case as you used in the Workflow Manager, then Connections, then Relational settings.

---



---

### 4.6.3 How to Set the Bulk Load Registry for DB2/390 Databases

DB2/390 databases do not support bulk loading; therefore, you need to disable bulk loading in the Windows registry: HKEY\_LOCAL\_MACHINE\..\System\CurrentControlSet\Services\Powermart\Configuration area.

#### To disable bulk loading for DB2/390 databases

- For DB2/390 databases, enter `DisableDB2BulkMode = Yes` in the Windows registry.

## 4.7 How to Copy Hibernate Libraries

To run the DAC Client or DAC Server, you need to have libraries from an open source software product called Hibernate. Hibernate libraries required to run the DAC Client or DAC server must be installed on each machine that runs the DAC client or DAC server. Hibernate libraries are not installed as part of Oracle BI Applications 7.9.x., but have to be downloaded from the Hibernate Web site.

Oracle recommends that you download Hibernate Core Package Version 3.2.x GA or later. Newer versions of Hibernate Core Package 3.2 are now generally available (for example, Hibernate Core Package Version 3.2.5 GA). DAC is supported on the libraries of these versions also. You can download Hibernate Core Package from <http://www.hibernate.org>.

#### To copy Hibernate libraries

1. Go to the Web site <http://hibernate.org>.
2. Navigate to the Download area and download Hibernate Core Package Version 3.2.x. GA to a local machine.
3. Unzip the files and directories in the hibernate-<3.2.x>.ga.zip file (for example, hibernate-3.2.2.ga.zip).

The files will unzip into a directory named hibernate-3.2.

4. For each machine that needs to run the DAC client or DAC server, copy the Hibernate files from the \hibernate-3.2 directory to the \OracleBI\DAC directory, as described in the table below.

**Table 4–2** *Hibernate library files that you need to copy*

Files	Copy from	Copy to
*.jar	\hibernate-3.2\lib	\DAC\lib
hibernate3.jar	\hibernate-3.2	\DAC\lib
hibernate-configuration-3.0.dtd	\hibernate-3.2\src\org\hibernate	\DAC
hibernate-mapping-3.0.dtd	\hibernate-3.2\src\org\hibernate	\DAC

---

**Note:** You do not need to copy any of the other files in the \hibernate-3.2 directory to the \OracleBI\DAC directory.

---

## 4.8 How to Configure the DAC Client

The DAC client is installed as part of an Oracle BI Analytics installation. You must run the DAC client from a Windows machine.

---

**Note:** You must have Java SDK installed on the machines where both the DAC server and DAC client are installed. Java SDK is also installed as part of the Oracle Business Intelligence infrastructure.

---

This section includes the following topics:

- [Section 4.8.1, "How to Configure the DAC Config.bat File"](#)
- [Section 4.8.2, "How to Install JDBC Drivers for DAC Database Connectivity"](#)

### 4.8.1 How to Configure the DAC Config.bat File

You configure the DAC config.bat file on the Windows machine that is used to run the DAC client. Follow this procedure to configure the DAC config.bat file.

#### To configure the DAC config.bat file

1. In the DAC directory, open the config.bat file.

This directory is usually in the OracleBI root directory (for example, <DRIVE>:\OracleBI\DAC).

2. Edit the JAVA\_HOME variable to point to the directory where you installed the Java SDK.

Make sure there are no spaces in the path reference.

For example:

```
set JAVA_HOME=d:\jdk1.5
```

3. Edit the DAC\_HOME variable to point to the directory where you installed the DAC.

For example:

```
set DAC_HOME=d:\OracleBI\DAC
```

## 4.8.2 How to Install JDBC Drivers for DAC Database Connectivity

This section provides instructions for installing the appropriate JDBC driver in the DAC\lib directory to enable DAC database connectivity. If the required JDBC drivers are not already installed, you need to install the JDBC driver on the machines where both the DAC server and DAC client are installed.

### To install JDBC drivers in the OracleBI\DAC\lib directory

- **Oracle.** If you are using an Oracle database (other than 8.x), find the directory where Oracle is installed. Copy the file named ojdbc14.jar in the jdbc\lib directory and paste it in the OracleBI\DAC\lib directory.
  - If you are using Oracle 8.x, copy the file named classes12.zip and paste it in the OracleBI\DAC\lib directory. Also, edit the ORACLELIB setting in the config.bat file as follows:

```
set ORACLELIB=%DAC_HOME%\lib\classes12.zip
```

- **DB2.** If you are using a DB2 database, find the directory where DB2 is installed. In the Java sub-directory copy the file named db2java.zip and paste it in the OracleBI\DAC\lib directory.

---

**Note:** If your source or target database is DB2-UDB, you also need to create DB2-UDB stored procedures to be used during the ETL process. For instructions, see [Section 4.21, "How to Create Stored Procedures for DB2-UDB"](#).

---

- **MSSQL.** Data Warehouse Administration Console is configured for Microsoft SQL Server 2005 JDBC drivers. If you are using a Microsoft SQL Server database, then download the Microsoft SQL Server 2005 JDBC Driver files and copy them to the <DRIVE>\OracleBI\DAC\lib\ directory. You can use the Microsoft SQL Server 2000 JDBC Driver files if you edit the \conf\connection\_templates.xml file and un-comment the section that starts "<!-- THIS SECTION IS FOR SQL SERVER 2000. Comment this section while using SQL Server 2005.-->

---

**Note:** You need the Microsoft SQL Server 2005 JDBC Driver 1.1 for SQL Server 2000 or SQL Server 2005.

---

- **Teradata.** If you are using a Teradata database, copy the files tdgssconfig.jar, TdgssUserConfigFile.xml, terajdbc4.jar, log4j.jar, and tdgssjava.jar from the Teradata installation directory to the \OracleBI\DAC\lib directory. Depending on the Teradata JDBC version, you might not have some of the above files.

## 4.9 How to Create the Required Databases

This procedure provides instructions for creating databases to hold the following:

- the DAC metadata repository.
- the Oracle Business Analytics Warehouse.

You can store both the DAC metadata repository and the Oracle Business Analytics Warehouse in separate databases, or in the same database. For example, if you are using an Oracle database, you might use Oracle SQL Plus to create one account for the

DAC metadata repository and one account for the Oracle Business Analytics Warehouse.

---

---

**Note:** You also need to know the database connection details for the OLTP database that stores your transactional data that you want to upload to the Oracle Business Analytics Warehouse. For example, you might have an Oracle database account called OLTP that stores the transactional data.

---

---

**To create the required databases**

1. (Optional) In your target database tool (for example, Oracle SQL Plus), create a database or database account for the DAC metadata repository. For example, in Oracle SQL Plus you might create a database account named 'DAC'.

If you perform step 1, you specify the name of the DAC metadata repository database (for example, DAC) when you log into the DAC (for more information about logging into the DAC, see [Section 4.10, "How to Create the DAC Metadata Repository"](#)).

2. In your target database tool (for example, Oracle SQL Plus), create a database account for the Oracle Business Analytics Warehouse tables. For example, in Oracle SQL Plus you might create a database account named 'OLAP'.

---

---

**Note:** Make sure that you specify this database account when you create the data warehouse tables in the DAC (for more information, see [Section 4.12.2, "How to Create Data Warehouse Tables"](#)).

---

---

If you omit step 1, you specify the name of the Oracle Business Analytics Warehouse database (for example, OLAP) when you log into the DAC (for more information about logging into the DAC, see [Section 4.10, "How to Create the DAC Metadata Repository"](#)).

## 4.10 How to Create the DAC Metadata Repository

This procedure provides instructions for logging into the DAC, and importing the required metadata into the DAC repository.

- [Section 4.10.1, "How to Create a DAC Connection"](#)
- [Section 4.10.2, "How to Log Into the DAC"](#)
- [Section 4.10.3, "How to Import Metadata into the DAC repository"](#)

---

---

**Note:** The DAC client can connect to only one DAC repository at a time.

---

---

---

---

**Note:** Before you start this procedure, you need to create a database for the DAC metadata repository. This database can be the same as the data warehouse database. For more information, see [Section 4.9, "How to Create the Required Databases"](#).

---

---

## 4.10.1 How to Create a DAC Connection

This section explains how to create a DAC Connection.

### To create a DAC connection

You create a DAC connection to enable you to log into the DAC using a stored set of login details.

1. Launch the DAC client by choosing Program Files\Oracle Business Intelligence\Oracle BI DAC\DAC Client.

---

**Note:** You can also start the DAC client by navigating to the OracleBI\DAC directory and double-clicking the startclient.bat file.

---

2. In the Login... dialog box, select Configure.
3. In the Configuring... dialog box, select Create Connection, and then click Next.
4. Enter the appropriate connection details.

---

**Note:** When you set up the DAC server later in the configuration (for more information, see [Section 4.18.1, "How to Configure the Connection Between the DAC Server and DAC Repository"](#)), you must use the same connection details that you specify here.

---

Field	Description
Name	A unique connection name that you specify here (for example, DAC).
Connection type	Type of database.
Database name	Database name or database account name in which you want to store the DAC metadata repository. If you are using... <ul style="list-style-type: none"> <li>■ Oracle (OCI8) use the tnsnames entry.</li> <li>■ Oracle (Thin) use the instance name.</li> <li>■ SQL Server use the database name.</li> <li>■ DB2-UDB use the connect string as defined in the DB2 configuration.</li> </ul>
Database Host	Machine name where the database resides.
Database Port	The port number on which the database listens. For example, for an Oracle database the default port is 1521, or for a SQL Server database the default port is 1433.

5. Select Test Connection to confirm that the login details are valid.
6. Select Apply to save the connection details and return to the login dialog.

The DAC repository is not supported on DB2-390 or Teradata databases. If your source or target database is DB2-390, you need to use a DB2-UDB, MSSQL, or Oracle database for both the DAC repository and the Informatica repository.

## 4.10.2 How to Log Into the DAC

This section explains how to log into the DAC.

### To log into the DAC

You log into the DAC to enable you to edit the DAC metadata repository. Before you can log into the DAC, you need to have created a DAC connection, which is a set of stored login details. For information about creating a DAC connection, see [Section 4.10.1, "How to Create a DAC Connection"](#).

1. Launch the DAC client by double-clicking the DAC Client icon on your desktop or navigating to the OracleBI\DAC directory and double-clicking the startclient.bat file.

The Login... dialog is displayed.



2. In the Login... dialog box, select a connection from the Connection drop down list.
3. In the Table owner name field, enter the database user name for the database in which you are creating the DAC metadata repository.

---

**Note:** For more information about creating a database for the DAC metadata repository, see [Section 4.9, "How to Create the Required Databases"](#).

---

4. In the Password field, enter the database password for the database user name that you specified in the previous step.
5. Click Login to start the DAC.

The DAC client starts and connects to the specified database.

6. Click Yes if you are prompted to create or upgrade the repository tables.

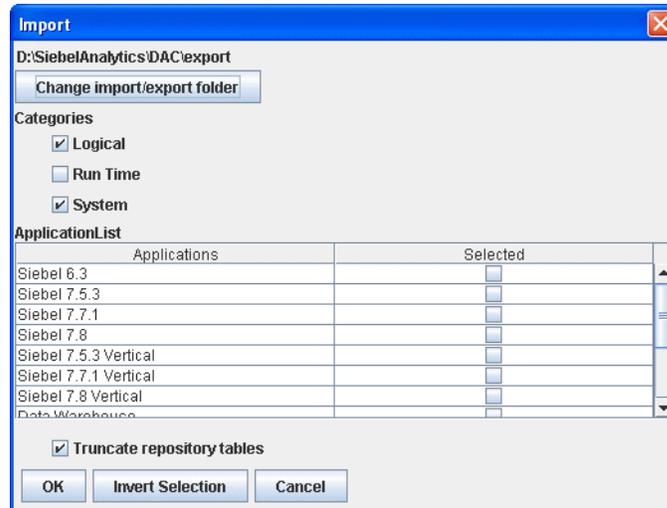
If the DAC repository schema does not exist in the database to which you are connecting, the schema will be automatically created. When a repository is created on Oracle or DB2 databases, you have the option of specifying a tablespace. For MSSQL and DB2 databases, you can indicate whether the environment is Unicode.

## 4.10.3 How to Import Metadata into the DAC repository

This section explains how to import metadata into the DAC repository.

### To import metadata into the DAC repository

1. If you are not logged into the DAC, log into the DAC (for more information, see [Section 4.10.2, "How to Log Into the DAC"](#)).
2. From the DAC menu bar, choose Tools, then DAC Repository Management, then Import to display the Import dialog.




---

**Note:** Make sure that the correct \DAC\export\ directory is displayed at the top of the dialog (for example, <drive letter>:\OracleBI\DAC\export). If necessary, use the 'Change import/export directory' button to select the \DAC\export\ directory. If the wrong directory is selected, the Applications list will be empty.

---

3. Using the check boxes in the **Selected** column, select the check box next to the business applications that you want to deploy.

---

**Note:** Oracle recommends that you always import the 'Universal' container by selecting the Universal check box in the **Selected** column.

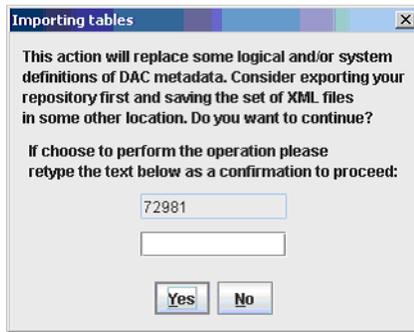
---

**Tip:** Make a note of the Application names that you select. When you create data warehouse tables later in the configuration process, you might need to type in the names exactly as they are displayed here (for more information, see [Section 4.12.2, "How to Create Data Warehouse Tables"](#)).

4. Select the Categories options as follows:

Categories Options	Description
Logical	Imports all information contained in the DAC Design view and the execution plan information for the DAC Execute view.
Run Time	Imports ETL Run History and the last refresh date information.
System	Imports all information contained in the DAC Setup view, except passwords for servers and database connections.

5. Click OK to display the Importing tables dialog.



6. To confirm that you want to import the seed data selected, re-type the text in the text box and click Yes.
7. If prompted to verify that you want to continue with the Import, click Yes.  
When the process is complete, the DAC displays a status box containing a success or failure message. If the process fails, use the `\OracleBI\DAC\log\import.log` file to diagnose errors.
8. Click OK.
9. If your source or target database is a DB2-390 database, run `DAC_DB2390_Source.sql` or `DAC_DB2390_Target.sql` immediately after importing the seed data by connecting to the database where the DAC repository resides. These files are stored in the `\OracleBI\DAC\` directory.

## 4.11 How to Configure the DAC Metadata Repository

This section includes the following topics:

- [Section 4.11.1, "How to Set Up DAC System Properties"](#)
- [Section 4.11.2, "How to Specify Transactional and Data Warehouse Data Sources in the DAC"](#)
- [Section 4.11.3, "How to Configure Email Recipients in the DAC Client"](#)
- [Section 4.11.4, "How to Activate Join Indices for Teradata Databases"](#)

---

**Note:** You will configure the DAC server to point to the DAC metadata repository later in the installation process, in the procedure [Section 4.18, "How to Configure the DAC Server"](#).

---

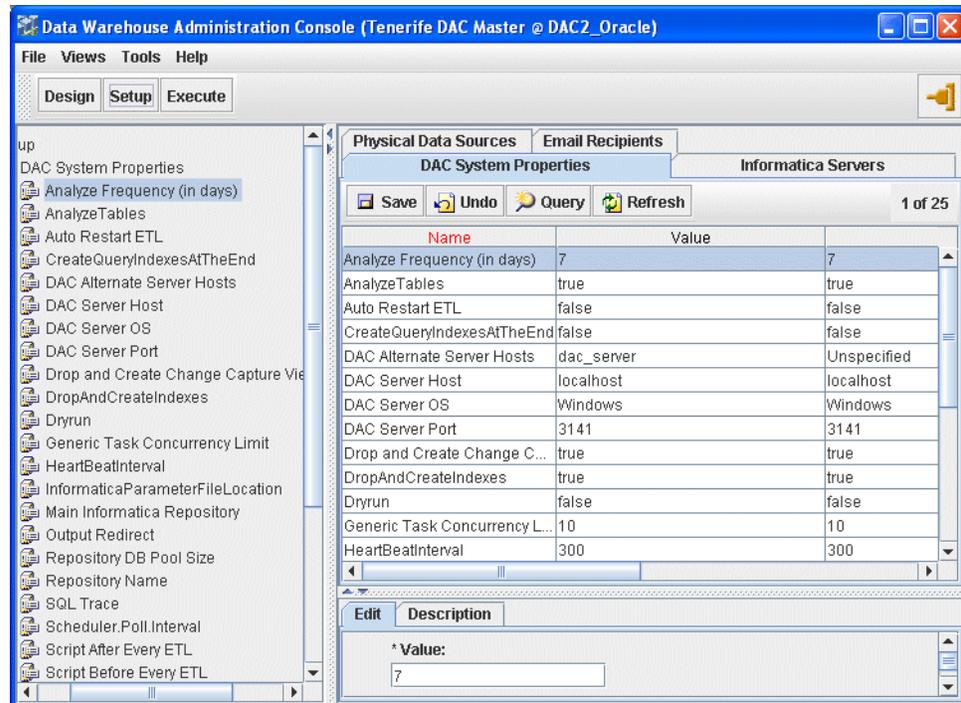
### 4.11.1 How to Set Up DAC System Properties

Follow this procedure to set up the DAC system properties, which determine the behavior of the DAC server. If you do not change a property value, Oracle BI Applications uses the property value that is automatically set during the Oracle BI Application installation.

#### To set up the DAC system properties

1. Launch the DAC client.
2. Click Setup on the DAC toolbar.

The DAC System Properties tab is active.



3. If required, change the property values, as described below.

---

**Note:** Possible values for all properties have to be keyed in exactly as described in the description tab (at the bottom) of each property. For example Auto Restart has possible values of false and true (case sensitive), and Server Log Level has the following possible values SEVERE, FINEST, FINER, FINE, INFO, WARNING (case sensitive).

---

#### 4.11.2 How to Specify Transactional and Data Warehouse Data Sources in the DAC

Follow this procedure to specify the transactional and data warehouse data sources in the DAC.

Oracle BI Applications is installed with a set of template data sources that you can edit to specify the data source details. Alternatively, you can specify new data sources without use a template.

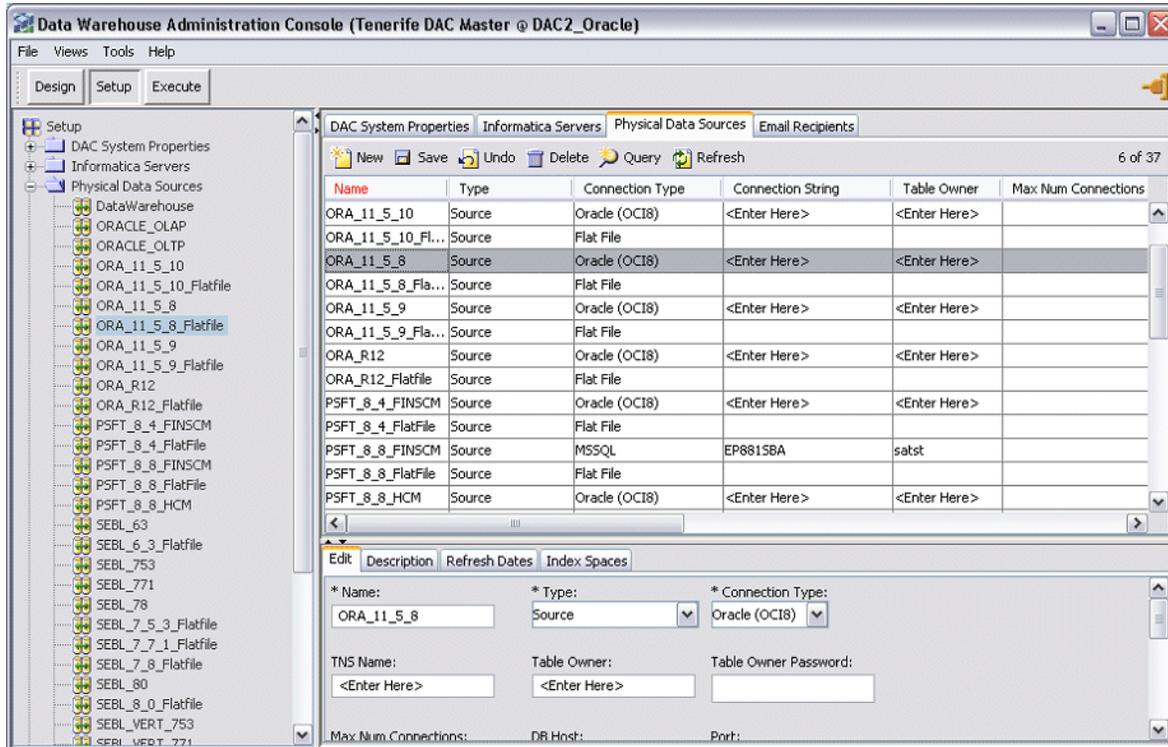
##### To specify transactional and data warehouse data sources

---

**Note:** You need to perform this procedure once for the transactional database and once for the data warehouse database. When you specify a database connection for the transactional database, select the 'Source' option from the Type drop down list on the Edit sub-tab. When you specify a database connection for the data warehouse database, select 'the Warehouse' option from the Type drop down list on the Edit sub-tab.

---

1. Launch the DAC client.
2. Click Setup on the toolbar, and then click the Physical Data Sources tab.



The Physical Data Sources tab displays a list of data source templates for many common database types, which you can use to specify your data sources.

3. Do one of the following:
  - Select an appropriate datasource template from the list, and edit the details in the Edit tab.
  - Click New, and enter the details in the Edit tab.
4. In the Edit subtab, enter the following:

Field	Description
Name	Logical name for the database connection. If you are using a pre-created data source template, do not change the default value.
Type	<ul style="list-style-type: none"> <li>■ Select 'Source' when you create the database connection for a transactional (OLTP) database.</li> <li>■ Select 'Warehouse' when you create the database connection for a data warehouse (OLAP) database.</li> </ul> <p>If you are using a pre-created data source template, do not change the default value.</p>
Connection Type	Type of database. Possible values are: <ul style="list-style-type: none"> <li>■ Oracle (OCI8)</li> <li>■ Oracle (Thin)</li> <li>■ DB2</li> <li>■ DB2-390</li> <li>■ MSSQL</li> <li>■ Teradata</li> <li>■ Flat File</li> </ul>

Field	Description
Instance or TNS Name	<p>For an Oracle database instance, enter the database SID (that is the SERVICE_NAME = value in the tnsnames.ora file in \network\admin\.</p> <p>For an Oracle TNS Name, enter the TNS entry name that is specified in the tnsnames.ora file in \network\admin\.</p> <p>For more information about which data sources you should specify, see <a href="#">Section 4.9, "How to Create the Required Databases"</a>.</p>
Table Owner	Valid database user.
Table Owner Password	Valid database user password.
Max Num Connections	Maximum number of database connections this connection pool can contain.
DB Host	Machine name or instance where the database resides. For example, databasename.us.company.com.
Port	Port number where the database listens (for example 1521 is the default for an Oracle database).
Priority	Number used to generate dependencies when designing execution plans.
Data Source Number	<p>Unique number assigned to the data source category so that the data can be identified in the data warehouse. For example, the value '9' is used for Siebel data sources.</p> <p>If you are editing a data source template for a data source type, Oracle recommends that you do not change the value that is set out-of-the-box. If you are specifying a data source without using a pre-defined template, you must use the correct value for that data source category. For example, if you specify an Oracle EBS data source, you must specify the DATASOURCE_NUM_ID value '9'. For a complete list of supported data sources and DATASOURCE_NUM_ID values, see <a href="#">Section 8.1.2.1, "How to Configure Data Source Num IDs"</a></p> <p>This value is passed as a parameter to the Informatica workflows. If you are using multiple sources, each data source has a unique number. Typically, all source dependent extracts will use this parameter to populate the DATASOURCE_NUM_ID column, and the source independent workflows will carry these values to the final dimension and fact tables.</p>
Default Index Space	<p>(Oracle specific) Specifies the table space in which the DAC drops and creates indices against this database connection</p> <p>To use the default table space, leave this field blank.</p>

---

**Note:** Alternatively, you can edit the template data sources that are installed and fill in the values for the data sources that you are using. For example, to specify the data warehouse data source, you can edit the DataWarehouse data source, fill in the data sources details, and click Save. Do not change the value in the Logical field.

---

5. Click Test Connection to make sure the connection works.
6. Click Save.

### 4.11.3 How to Configure Email Recipients in the DAC Client

Follow this procedure to configure email recipients.

#### To configure email recipients in the DAC client

1. Launch the DAC client.
2. Click Setup on the DAC toolbar, and then click the Email Recipients tab.
3. Click New.
4. In the Edit tab below, enter the following information:

Field	Description
Name	Logical name of the user to be notified.
Email Address	Email address where the notification is sent.
Notification Level	The notification levels are as follows: <ul style="list-style-type: none"> <li>■ 10 -- Notifies recipient of success or failure of each task.</li> <li>■ 5 -- Notifies recipient of success or failure of the entire ETL process.</li> <li>■ 1 -- Notifies recipient that ETL completed successfully.</li> </ul>
Inactive	Indicates whether the selected email activation is active or inactive.
Needs Authentication	Read only value that specifies whether the corporate email server requires authentication (this value is set by choosing Tools, then DAC Server Setup).

### 4.11.4 How to Activate Join Indices for Teradata Databases

For Teradata databases, the preconfigured tasks for creating and dropping join indices are inactive.

#### To activate join indices for Teradata databases

1. Create a new execution plan with the list of subject areas that you are interested for the ETL.
2. Query for all tasks whose name starts with 'Teradata Drop' and add them as preceding tasks.
3. Query for all tasks whose name start with 'Teradata Create' and add them as following tasks.
4. Assemble the execution plan parameters in the Parameters tab and configure the parameters.
5. Redesign the execution plans.

## 4.12 How to Configure the Oracle Business Analytics Warehouse Database

This section includes the following procedures for configuring the Oracle Business Analytics Warehouse database:

- [Section 4.12.1, "How to Create the SSE Role"](#)
- [Section 4.12.2, "How to Create Data Warehouse Tables"](#)
- [Section 4.12.3, "How to Create Data Warehouse Tables on a Teradata Database"](#)

---



---

**Note:** Before you start this procedure, you need to create a database for the data warehouse. This database can be the same as the DAC metadata repository database.

---



---

### 4.12.1 How to Create the SSE Role

Follow this procedure to create the SSE role.

#### To create the SSE Role

- Create a database role named SSE\_ROLE (SSEROLE for DB2 390 databases). Assign this role to the database user. For instructions on creating roles, see the documentation provided with your database.

For example, on an Oracle database you might use the 'CREATE ROLE' SQL command to create the role, and the GRANT command to assign privileges to the role, and to assign the role to the data warehouse database account.

Note the following:

- For an Oracle database, when you create the SSE\_ROLE role, you need to grant the following privileges:
  - CONNECT
  - RESOURCE
- For an Oracle database, if the DAC repository and the Informatica repository are stored in a different database to the data warehouse database, the SSE\_ROLE must have the following additional privileges:
  - READ
  - WRITE
  - UPDATE
  - DELETE
  - GRANT
- If you are using a Teradata database, you do not need to create the SSE role. If you are using a DB2 390 database, you can use an alternative SSE role name to SSEROLE if required. To specify a different role name, modify the 'set GRANTEE="SSEROLE"' line in the OracleBI\dwrep\createtables.bat file.

### 4.12.2 How to Create Data Warehouse Tables

Use this procedure to create the data warehouse tables. Before you perform this procedure make sure that the SSE role has been created for the Oracle Business Analytics Warehouse, and that the database user has been associated with the role (for more information, see [Section 4.12.1, "How to Create the SSE Role"](#)).

If you are using a Teradata database as a source transactional database, the DAC creates a SQL file to create the schema tables, not the tables themselves. Follow the steps in [Section 4.12.3, "How to Create Data Warehouse Tables on a Teradata Database"](#).

---



---

**Note:** Additional work by the database administrator is required if the data warehouse tables need to be moved to different tablespaces for performance or manageability reasons.

---



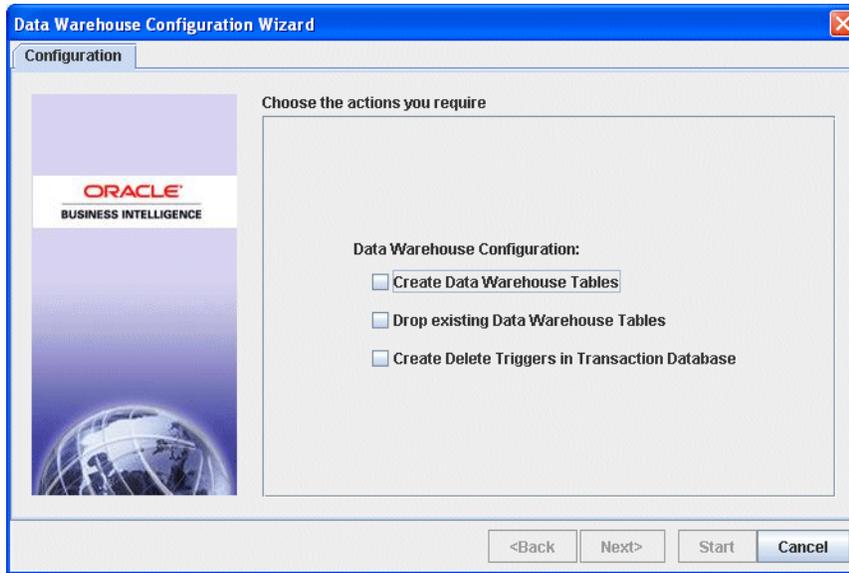
---

**To create data warehouse tables**

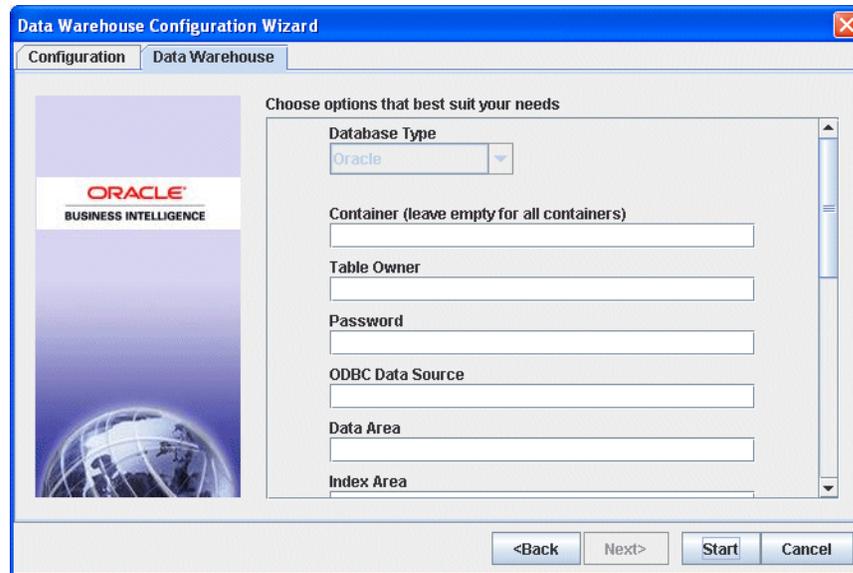
1. From the DAC menu bar, click Tools, then ETL Management, then Configure.
2. In the Sources dialog box, select the database platform for the target data warehouse and source transactional database.



3. Click OK to display the Data Warehouse Configuration Wizard.



4. Select the Create Data Warehouse Tables check box, and click Next.  
The Data Warehouse tab is active.



5. Enter the details of the database in which you want to store the data warehouse.

The information that you need to enter is dependent on the type of target database that you are using to store the data warehouse.

Field	Description
Database Type	(Read only) Type of database, as specified by the 'Target data warehouse database platform' field on the Sources dialog (see step 2). If the database type is DB2 390, then check the 390 Database check box.

Field	Description
Container	<p>The name of the source business applications that you want to deploy.</p> <p><b>Note:</b> Oracle recommends that you leave the <b>Container</b> field blank. If you leave the <b>Container</b> field blank, the DAC creates a container by default for all of the following:</p> <ul style="list-style-type: none"> <li>■ The source business applications that you selected when you imported the seed data into the DAC metadata repository earlier (for more information, see <a href="#">Section 4.10, "How to Create the DAC Metadata Repository"</a>).</li> <li>■ Any copies of those source system applications.</li> <li>■ Any additions made those source system applications.</li> <li>■ Any new source system applications that you have custom built.</li> </ul> <p>For example, you might do the following:</p> <ul style="list-style-type: none"> <li>■ Import the seed data for Oracle 11.5.9 and 11.5.10</li> <li>■ Make a copy of Oracle 11.5.10 and call it 'My Oracle 11.5.10'.</li> <li>■ Add two data warehouse tables to 'My Oracle 11.5.10'</li> <li>■ Create a new source system application called 'My new source' and add 10 data warehouse tables to 'my New Source'.</li> </ul> <p>If you leave the <b>Container</b> field blank, the The DAC will create the following containers:</p> <ul style="list-style-type: none"> <li>■ Oracle 11.5.9</li> <li>■ Oracle 11.5.10</li> <li>■ My Oracle 11.5.10</li> <li>■ My new Source</li> </ul> <p>If there tables that are common to them, then only one table will be created. For example, if there is a table called W_ORG_D in Oracle 11.5.9 and Oracle 11.5.10, then the DAC will create only one table called W_ORG_D. If columns are different for the same table across containers, then DAC will create a table that has all the columns in the same table.</p> <p>If you only want to deploy a sub-set of the source business applications for which you imported seed data earlier, then use this field to specify a container name. When you specify a container name, you must enter the names of the applications exactly as they are displayed on the seed data Import dialog.</p> <p>For example:</p> <ul style="list-style-type: none"> <li>■ If you want to deploy Oracle 11.5.9, enter: 'Oracle 11.5.9'.</li> <li>■ If you want to deploy Oracle 11.5.9 and 11.5.10, enter: 'Oracle 11.5.9,Oracle 11.5.10'.</li> </ul> <p>For more information about containers, see <i>Oracle Business Intelligence Data Warehouse Administration Console Guide</i>.</p>
Table Owner	Valid database owner, username, or account that you set up to hold the data warehouse (for more information, see <a href="#">Section 4.9, "How to Create the Required Databases"</a> ).
Password	Valid database user password for the database owner, username, or account that you specified in the Table Owner field (for more information, see <a href="#">Section 4.9, "How to Create the Required Databases"</a> ).
ODBC Data Source	Data Source Name (DSN) for the Oracle Business Analytics Warehouse. You need to specify the name of the ODBC connection (for example, Connect_to_OLAP) that you created for the data warehouse (for more information, see <a href="#">Section 4.5.7, "How to Create ODBC Database Connections"</a> ).

Field	Description
Data Area	(Optional) Tablespace where data warehouse tables are created.
Index Area	(Optional) Indexspace where data warehouse indices are created (applicable only to Oracle and DB2 databases).
Is Unicode	Specifies whether the data warehouse database is Unicode.

**6. Click Start.**

The Run Status tab displays information about the process, as follows:

- If a 'Success' message is displayed, the data warehouse tables have been created. If you want to see log information about the process, use the following log files.
  - \OracleBI\DAC\config\generate\_ctl.log - A log of the schema definition process, including details of any conflicts between containers.
  - \OracleBI\DAC\config\createtables.log - A log of the ddlimp process.
- If a 'Failure' message is displayed, the data warehouse tables have not been created. Use the log information in \OracleBI\DAC\config\generate\_ctl.log to diagnose the error. The createtables.log is not generated.

### 4.12.3 How to Create Data Warehouse Tables on a Teradata Database

This section includes the procedures for creating and dropping data warehouse tables on a Teradata database. Before you start, make sure that the tables that you create are case specific by setting the session mode to ANSI in your Teradata ODBC configuration (for more information, see [Section 3.9, "Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse"](#)).

---



---

**Note:** For Unicode environments, perform these procedures using a login that has the default character set UTF-8.

---



---

#### To create data warehouse tables on a Teradata database

1. From the DAC menu bar, click Tools, then ETL Management, then Configure.
2. In the Sources dialog box, select 'Teradata' as the database platform for the source data warehouse, and select the appropriate database type for the target transactional database.
3. Click OK to display the Data Warehouse Configuration Wizard.
4. Click the 'Generate create statements for Data Warehouse Tables' check box, then click Next to display the Data Warehouse SQL tab.
5. Use the Container field to specify individual containers, or leave blank to deploy all containers.
6. Click Start.

The Run Status tab displays information about the process, as follows:

- If a 'Success' message is displayed, the data warehouse tables have been created. If you want to see log information about the process, use the following log files.
  - \OracleBI\DAC\config\generate\_ctl.log - A log of the schema definition process, including details of any conflicts between containers.

\OracleBI\DAC\config\createtables.log - A log of the ddlimp process.

- If a 'Failure' message is displayed, the data warehouse tables have not been created. Use the log information in \OracleBI\DAC\config\generate\_ctl.log to diagnose the error. The createtables.log is not generated.
7. Copy the SQL file created in step 6 from \conf\sqlgen\sql\Teradata into SQL Assistant and execute the SQL.

## 4.13 How to Register Informatica Servers in the DAC Client

This section describes registering the Informatica Server and the Informatica Repository Server in the DAC and specifying the number of workflows that can be executed in parallel. The DAC server automatically load balances across the servers and does not run more than the value specified for each of them.

[Section 4.5, "How to Set Up the Informatica Server and Informatica Repository Server"](#).

**Tip:** Specify Informatica Server and Repository Server details that are consistent with the Informatica details that you specified when you set up the Informatica Server (for more information, see

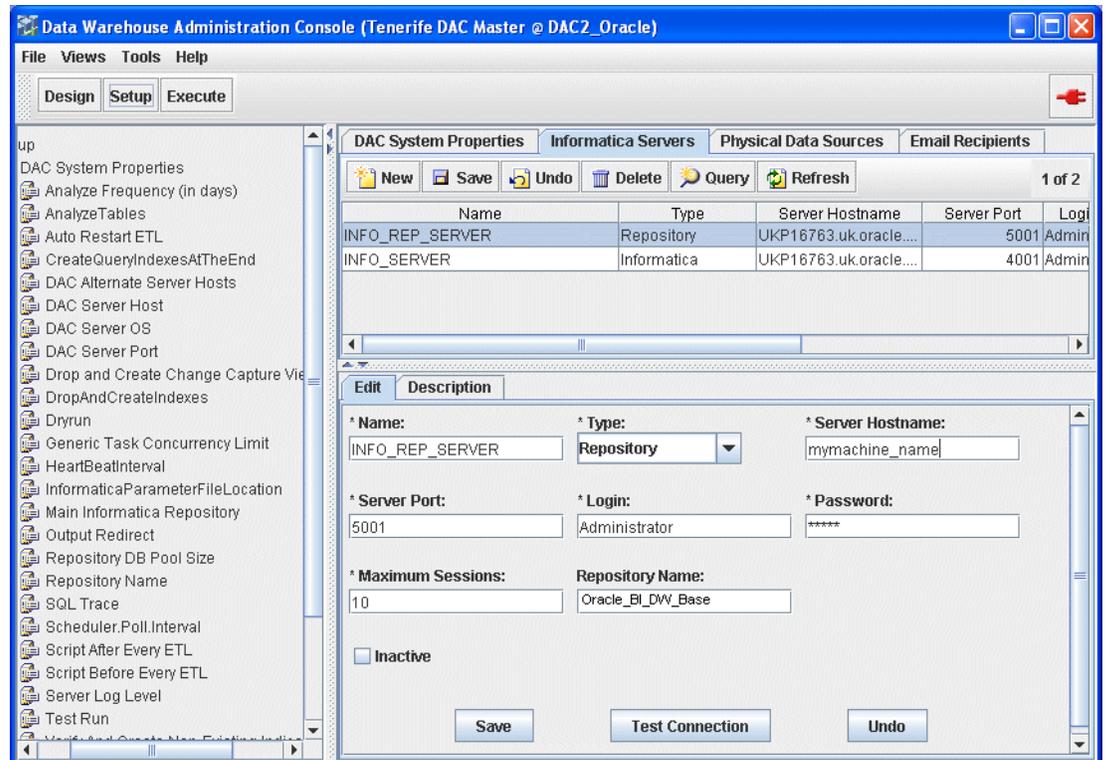
### 4.13.1 How to Register Informatica Servers in the DAC Client

When you register Informatica Servers in the DAC client, note the following:

- You must register at least one Informatica server in the DAC. You should register one Informatica server for each Informatica repository.
- You must not register more than one Informatica Repository server in the DAC.
- It is recommended that the Informatica Server be installed on the machine where the DAC server is installed. If a machine is running only the DAC client, it does not need an Informatica server installed.

#### To register Informatica servers in the DAC client

1. Launch the DAC client.
2. Click Setup on the DAC toolbar, and then click the Informatica Servers tab.
3. Click New to display the Edit tab below.



If you are configuring a new installation, the Informatica Servers tab will be empty. If you are upgrading an existing installation, the Informatica Servers tab might contain existing Informatica servers.

#### 4. Register the Informatica server, by entering the following information:

Field	Enter
Name	Logical name for the Informatica server (for example, INFO_SERVER).
Type	Choose Informatica.
Server Hostname	Informatica server host name or IP address.
Server Port	Port where the Informatica server connects for requests (the default port number is 4001). Oracle recommends that you use the default port number.
Login	Informatica user name who has appropriate privileges to execute workflows (for example, Administrator).
Password	Informatica user password (the default password is Administrator).
Maximum Sessions	Maximum number of workflows that can be executed in parallel on the Informatica server. If the number of sessions is zero or is not specified, the DAC server assigns the default value of 10.
Repository Name	Enter Oracle_BI_DW_Base (or Oracle_BI_DW_Teradata for Teradata installations). <b>Note:</b> If you need to verify the repository name, log into the Informatica Administration Console and connect to the repository server.
Inactive	Indicates whether the Informatica server is active or inactive.

---



---

**Note:** You need to specify Informatica Server and Informatica Repository Server details that are consistent with the Informatica details that you specified when you set up the Informatica Server (for more information, see [Section 4.5, "How to Set Up the Informatica Server and Informatica Repository Server"](#)).

---



---

- If you want to test the connection, click **Test Connection**.

---



---

**Note:** Before you can test a connection, you need to start the Informatica Repository (for more information, see [Section 4.19, "How to Start the Informatica Server"](#)).

---



---

- Click Save to save the details.
- Click New to display the Edit tab below
- Register the Informatica Repository server, by entering the following information:

Field	Enter
Name	Logical name for the Informatica Repository server (for example, INFO_REP_SERVER).
Type	Choose Repository.
Server Hostname	Informatica Repository server host name or IP address.
Server Port	Port where the Informatica Repository server connects for requests. (The default port number is 5001. It is recommended that you use the default port number.)
Login	Informatica user name who has appropriate privileges to execute workflows (for example, Administrator).
Password	Informatica user password (the default password is Admin).
Maximum Sessions	Maximum number of workflows that can be executed in parallel on the Informatica server. If the number of sessions is zero or is not specified, the DAC server assigns the default value of 10.
Repository Name	Type Oracle_BI_DW_Base (or Oracle_BI_DW_Teradata in a Teradata environment).  <b>Note:</b> If you need to verify the repository name, log into the Informatica Administration Console and connect to the repository server.
Inactive	Indicates whether the Informatica Repository server is active or inactive.

---



---

**Note:** You need to specify Informatica Server and Informatica Repository Server details that are consistent with the Informatica details that you specified when you set up the Informatica Server (for more information, see [Section 4.5, "How to Set Up the Informatica Server and Informatica Repository Server"](#)).

---



---

- Click Test Connection to make sure that the connection works.

---



---

**Note:** Before you can test a connection, you need to start the Informatica Repository (for more information, see [Section 4.14.5, "How to Stop and Start the Informatica Repository \(Optional\)"](#)).

---



---

10. Click Save to save the details.

If you have not already started the Informatica servers (for more information, see [Section 4.5.6, "How to Start the Informatica Services"](#)), start the Informatica servers now, then go back and test the connections using the Test Connection button.

## 4.14 How to Configure and Manage the Informatica Repository

This section contains instructions for configuring and managing the Informatica repository. For information about checking out and checking in objects in the Informatica repository, see [Section 4.14.1, "About the Versioned Informatica Repository"](#).

A pre-built Informatica repository called Oracle\_BI\_DW\_Base is copied into the OracleBI\dwrep\Informatica\Repository directory during the Oracle BI Applications installation.

**To configure the Informatica Repository for a new Oracle BI Applications installation:**

1. Use the Informatica Repository Administration Tool to register the Informatica Server machine. For more information, see [Section 4.14.2, "How to Register the Informatica Repository Server"](#).
2. Use the Informatica Repository Administration Tool to load the pre-built Oracle\_BI\_DW\_Base.rep repository. For more information, see [Section 4.14.3, "How to Load the Pre-Built Repository Into Informatica"](#).
3. Use the Informatica Workflow Manager tool to configure the Informatica Repository. For more information, see [Section 4.14.4, "How to Configure the Informatica Repository in Workflow Manager"](#).
4. Start the Informatica services. For more information, see [Section 4.5.6, "How to Start the Informatica Services"](#).

You can also use the Informatica Repository Administration Tool to do the following optional tasks:

- stop and start the repository. For more information, see [Section 4.14.5, "How to Stop and Start the Informatica Repository \(Optional\)"](#).
- create and restore backup versions. For more information, see [Section 4.14.6, "How to Back Up and Restore the Informatica Repository \(Optional\)"](#).

### 4.14.1 About the Versioned Informatica Repository

This section explains the versioned Informatica Repository that is included in Oracle Business Intelligence Applications.

#### 4.14.1.1 Summary of Versioning in the Informatica Repository

Oracle Business Intelligence Applications 7.9.4 ships the following two Informatica repository files:

- Oracle\_BI\_DW\_Base.rep

- Oracle\_BI\_DW\_Teradata.rep

In releases 7.9.0, 7.9.1 and 7.9.2, these were non-versioned files provided in the standard application. In Oracle Business Intelligence Applications version 7.9.3 and 7.9.4, these Informatica repository files are now versioned.

Because of this change, you need to enter the Team based Development license when loading the Informatica repository file supplied with Oracle BI Applications. Also, if you extend or modify the standard mappings, you need to check out and check changes to the Informatica repository. The Oracle Business Intelligence Data Warehouse Console (DAC) will now only pick up and execute valid and checked-in objects from the Informatica repository.

Oracle BI Applications now uses a standard way of labeling objects. For example, the standard Informatica objects have the label 'Oracle Business Intelligence Applications Release 7.9.x'. This label is useful when tracing back and comparing changes made from release to release. Note that Informatica does not allow a versioned repository file to be restored as a non-versioned repository somewhere else. Therefore, this change is uni-directional.

#### 4.14.1.2 Applying the Team Based Development License to the Informatica Repository

When you load the pre-built repository into Informatica, you must use a Team Based Development license for the Options connectivity key. For more information, see [Section 4.14.3, "How to Load the Pre-Built Repository Into Informatica"](#).

#### 4.14.1.3 Customization of Repository Objects Using Check Out and Check In

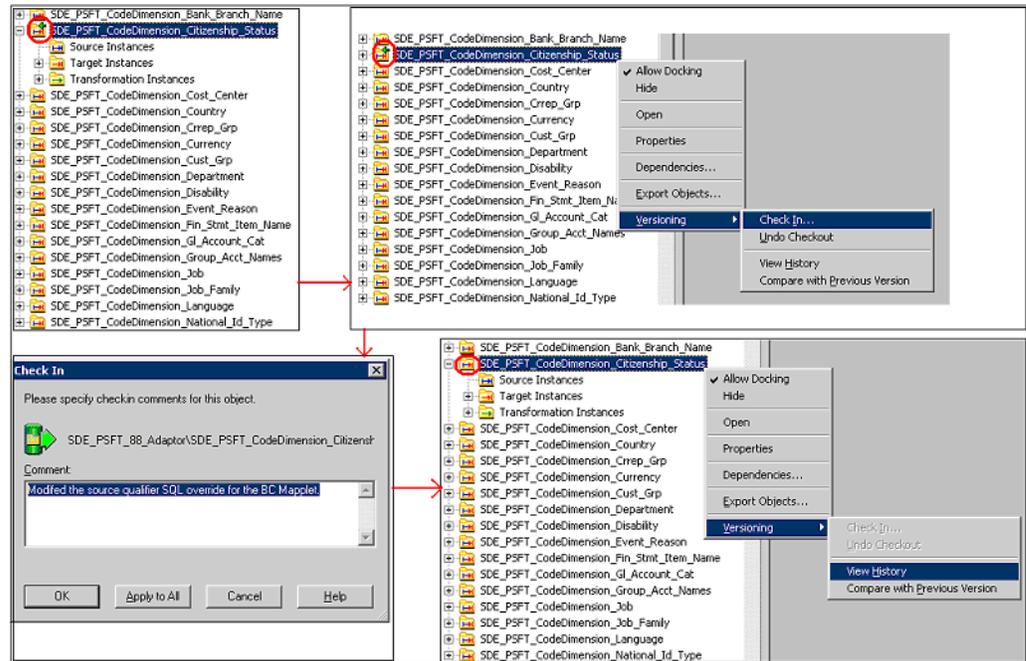
When the Team based Development option has been enabled for an Informatica repository, you must "check out" a repository object to modify and then "check in" the change. To extend or modify the standard mappings, you must check out a repository object that is to be modified, make the desired changes, and then check in the changes. This process is described below, and is illustrated in the screenshot below where the red arrow represents the action flow.

The workflow for checking out and checking in changes in the Informatica repository is described below:

- Developers check out an object for modification.  
Note the green arrow on the object in the screenshot below.
- Developers make changes, validate and save the object, and are ready to check their changes in.  
The green arrow is still displayed to indicate this is still a checked out object.
- Developers provide mandatory check in comments and finally check in.
- The object is checked in now.

Note that the green arrow disappears to indicate that the current version is now checked in and is usable/visible by other developers (for example, in the DAC). Developers can also compare versions, view history for this object now.

The version prior to this change also gets stored in the repository for tracking purposes. Apart from tracking changes in repository objects, Informatica provides additional useful features like "Query" and "labeling" on versioned repositories. Refer to the Informatica product guides for the more information on the topic "team based development".



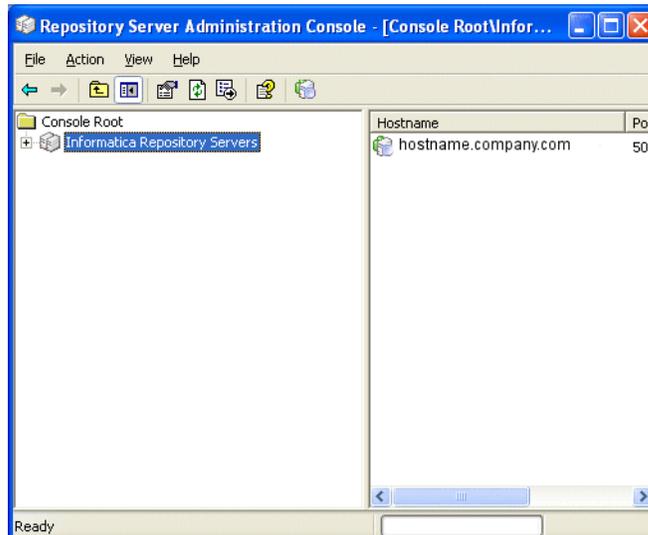
### 4.14.2 How to Register the Informatica Repository Server

Follow this procedure to register the Informatica Repository Server in the Informatica Repository Server Administration Console.

**Note:** Make sure the Informatica Repository Server is running before you begin this procedure.

#### To register the Informatica Repository Server

1. Launch the Repository Server Administration Console.
2. In the left pane, select the Informatica Repository Servers node, which appears under Console Root.



3. From the menu bar, select Action, then New Server Registration to display the 'Register a new Repository Server' dialog.
  - a. Enter the host name (the machine where the repository server resides).
  - b. Accept the default port number 5001 or enter the appropriate port number. Click OK.  
The Repository Server host name appears in the right pane under Hostname.
  - c. Right-click the server name and choose Connect.
  - d. In the Connecting to Repository Server dialog, enter the Administrator password (the default is Admin). Click OK.

These values should match the values you entered in [Section 4.5, "How to Set Up the Informatica Server and Informatica Repository Server"](#).

### 4.14.3 How to Load the Pre-Built Repository Into Informatica

You use the Restore option in the Informatica Repository Administration Tool to load the pre-built Oracle\_BI\_DW\_Base.rep repository or Oracle\_BI\_DW\_Teradata.rep (in a Teradata environment) into Informatica.

---

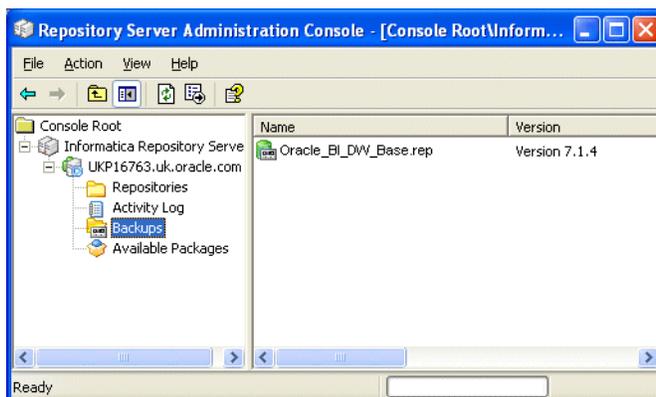
**Note:** The pre-built Oracle\_BI\_DW\_Base and Oracle\_BI\_DW\_Teradata repository files are versioned from BI Applications 7.9.3 and higher. You must use a The Team Based Development License key in step 13. For more information about the versioned repositories, see [Section 4.14.1, "About the Versioned Informatica Repository"](#)

---

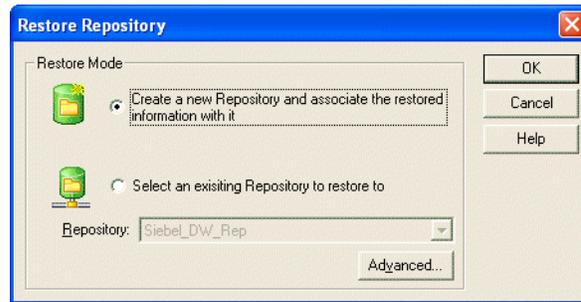
Before you start this procedure, you need to create a database to contain the Informatica Repository. You can use the same database as the data warehouse database. For more information, see [Section 4.9, "How to Create the Required Databases"](#).

#### To load the pre-built Oracle\_BI\_DW\_Base.rep repository into Informatica

1. Copy the Oracle\_BI\_DW\_Base.rep from the OracleBI\dwrep\Informatica\Repository directory to the \Informatica PowerCenter 7.1.4\RepositoryServer\bin\Backup directory.
2. Launch the Repository Server Administration Console and connect to the Informatica Repository machine.
3. Click Backups to display available backup files in the right hand pane.



- Right click on Oracle\_BI\_DW\_Base.rep and click Restore to display the Restore Repository dialog.

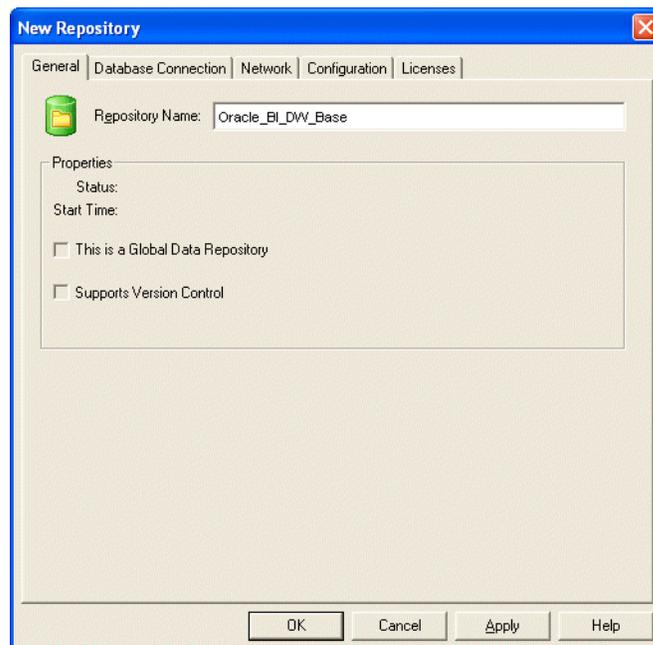


- Click the 'Create a new Repository and associate the restored information with it' check box.
- Click OK to start the Restore procedure and display the New Repository dialog.
- In the General tab, in the Repository Name field, enter the name for the Informatica repository (for example, Oracle\_BI\_DW\_Base).

---

**Note:** This value should match the value you entered in [Section 4.5, "How to Set Up the Informatica Server and Informatica Repository Server"](#) (for example, Oracle\_BI\_DW\_Base).

---



- In the Database Connection tab, enter the following information:

Field	Description
DatabaseType	The type of database storing the repository.
CodePage	The code page based on the operating system platform.

Field	Description
ConnectionString	The native connect string of the database that stores the data warehouse. For MSSQL Server databases, use <SERVERNAME@DATABASENAME>.
DBUser	The account used to access the DAC repository (for example, DAC).
DBPassword	The password for the DAC repository user (for example, DAC).

9. Click the Licenses tab.  
You may see a message stating the license file is empty.
10. Click OK to close the message dialog box.
11. In the License Key field, enter the Product license key, and then click Update.  
The license key information appears in the lower window of the Licenses tab.
12. Repeat the step above for the Connectivity license keys.
13. Repeat the step above for the Option license keys.  
You must enter the Team Based Development Licence key.
14. Click OK to save the repository details and create the repository.
15. Select the repository, right-click and then click Start to start the repository.

---



---

**Note:** The Informatica repository tables need to be analyzed for better performance. This procedure must be done manually by the database administrator.

---



---

#### 4.14.4 How to Configure the Informatica Repository in Workflow Manager

Follow this procedure to configure the Informatica repository in Informatica Workflow Manager.

##### To configure the server properties for the Informatica repository in Workflow Manager

1. Launch Informatica Workflow Manager.
2. In the Repositories list, select the Informatica repository.
3. Choose Repository, then Connect.  
The Connect to Repository window opens.
4. Log in using the username and password you entered in [Section 4.5, "How to Set Up the Informatica Server and Informatica Repository Server"](#) (for example, with the user name Administrator).
5. Click Connect.
6. Once connected, double-click the icon of the server name (ORACLE\_BI\_DW\_SERVER is the recommended server name) under the repository icon.  
The Server dialog box opens.
7. In the Host Name field, enter the computer name (as shown in the Network control panel) of the machine running the Informatica Server.
8. Click Resolve Server to confirm the computer name and update the IP address.

9. (Optional) Configure the Code Page setting.

---

**Note:** If the Informatica Server uses ENU Windows, leave the setting unchanged. For localized environments, choose the appropriate setting.

---

10. Click Advanced to expose the Server Variables list.

11. Make sure that the \$PMRootDir variable points to the directory where the Informatica Server is installed.

12. Make sure that the \$PMSourceFileDir points to the location mentioned in DAC, then Setup, then DAC System Properties, then InformaticaParameterFileLocation. Also make sure that this location has no spaces in its path (the default path the \SrcFiles\ directory in the Informatica installation directory).

---

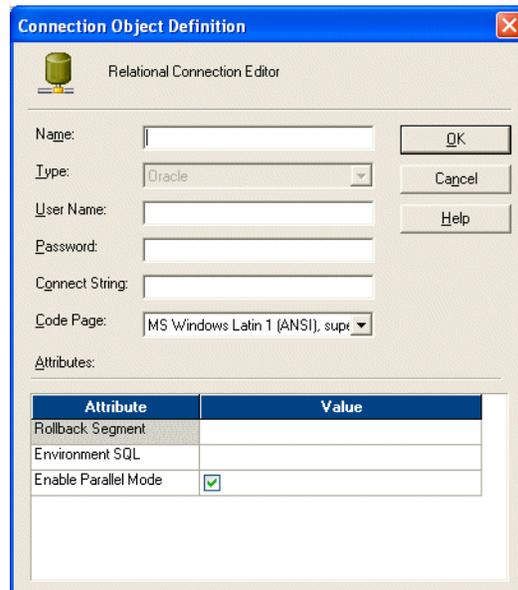
**Note:** If you install the Informatica Server on a machine different from the recommended configuration, make sure to update the \$PMRootDir variable with the new location.

---

13. Close the Server window.

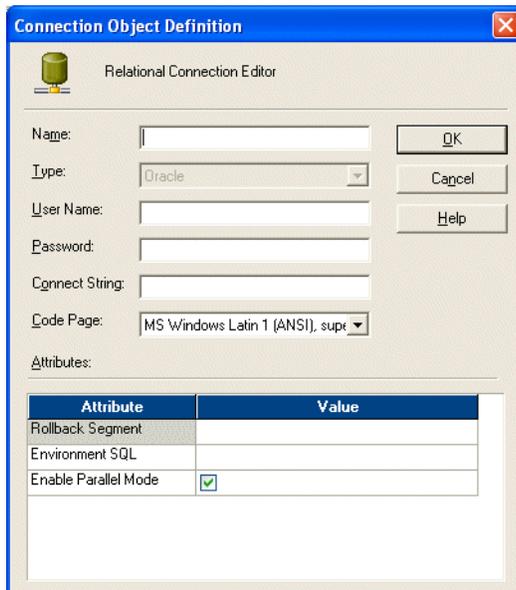
**To configure the database connections**

1. In Informatica Workflow Manager, choose Connections, then Relational to display the Relational Connection Browser.
2. Create a connection for the target data warehouse, as follows:
  - a. Click New, select the appropriate sub-type, then click OK to display the Relational Connection Editor.



- b. If the target database is Oracle, DB2, or Teradata, use the settings as follows:
      - Click New and select the type as Oracle, DB2, or Teradata, then click OK.

- Name: DataWarehouse (exactly as specified by selecting DAC, then setup, then Physical Data sources for Data Warehouse).
  - User Name: Table Owner.
  - Password: TBO password.
  - Connect string: Connect string for the connecting to the database.
  - Code Page: UTF-8 encoding of Unicode if the database is unicode.
- c. If the target database is SQL Server, use the settings as follows:
- Click New and select the type as ODBC, then click OK.
  - Name: DataWarehouse (type in the name exactly as specified by selecting DAC, then Setup, then Physical Data sources for your OLTP (for example, if your source system is Siebel ver 7.8.x then you name this connection as SEBL\_78).
  - User Name: Table Owner.
  - Password: TBO password.
  - Connect string: Connect string for the connecting to the database.
  - Code Page: UTF-8 encoding of Unicode if the database is unicode.
3. Create connections for the source database, as follows:
- a. Click New, select the appropriate sub-type, then click OK to display the Relational Connection Editor.



- b. If the target database is Oracle, DB2, use the settings as follows:
- Click New and select the type as Oracle or DB2, then click OK.
  - Name: Type in the name exactly as specified by selecting DAC, then Setup, then Physical Data sources for your OLTP (for example, if your source system is Siebel ver 7.8.x then you name this connection as SEBL\_78).
  - User Name: Table Owner.

- Password: TBO password.
  - Connect string: Connect string for the connecting to the database.
  - Code Page: UTF-8 encoding of Unicode if the database is unicode.
- c. If the target database is SQL Server, use the settings as follows:
- Click New and select the type as ODBC, then click OK.
  - Name: Type in the name exactly as specified by selecting DAC, then Setup, then Physical Data sources for your OLTP (for example, if your source system is Siebel ver 7.8.x then you name this connection as SEBL\_78).
  - User Name: Table Owner.
  - Password: TBO password.
  - Connect string: Connect string for the connecting to the database.
  - Code Page: UTF-8 encoding of Unicode if the database is unicode.
4. Click Close to close the Relational Connection Browser.

#### **To configure the Informatica repository in the Workflow Manager for Teradata external loader connections**

1. In Informatica Workflow Manager, choose Connections, then Loader to open the Loader Connection Browser.
2. In the Objects window, select Teradata\_Tpump\_Upsert, and then click Edit.
3. Edit the User Name, Password, TDPID, Database Name, and other attributes as necessary.
4. Repeat these steps for the following objects:
  - Teradata\_Tpump\_Update
  - Teradata\_Tpump\_Insert
  - Teradata\_Tpump\_Delete
  - Teradata\_FastLoad
5. For Unicode environments, append -c UTF8 to the value for the External Loader Executable attribute for each external loader.  
For example:
  - fastload -c UTF8
  - tpump -c UTF8
6. Click Close to close the Loader Connection Browser.

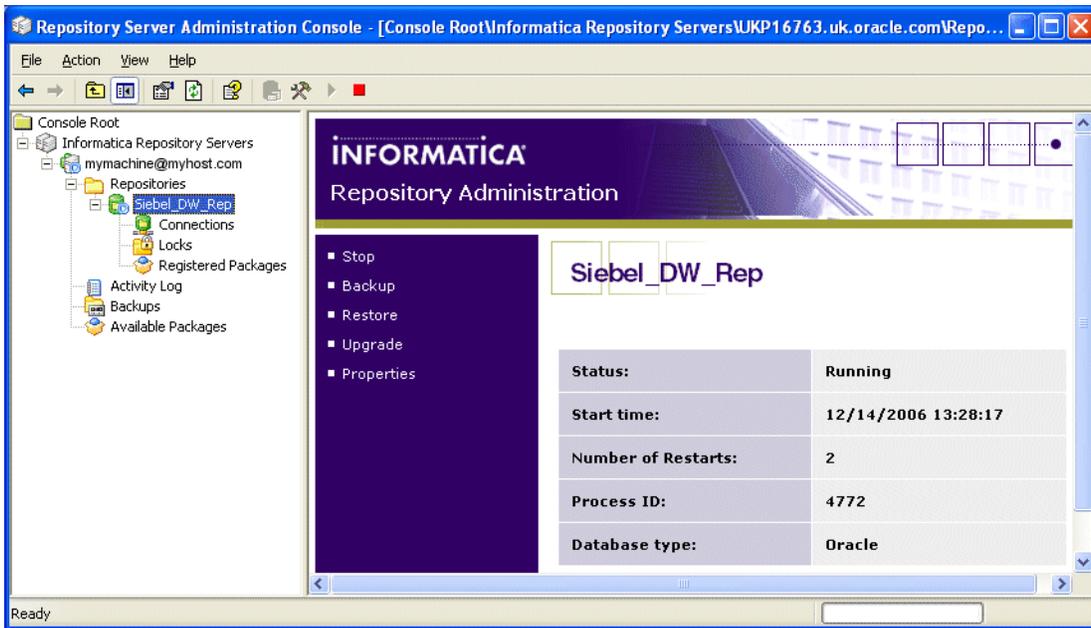
### **4.14.5 How to Stop and Start the Informatica Repository (Optional)**

This section includes instructions for stopping and starting the Informatica repository.

#### **To stop and start the Informatica repository**

1. Start the Repository Server Administration Console in Informatica PowerCenter.
2. Display the Informatica Repository Administration dialog, as follows:
  - a. In the console navigator at the left hand side, double click the Informatica Repository Server to expand this node.

- b. Right click on the server machine name that holds the Informatica Repository Server
- c. Choose Action, then Connect to display the Connecting to Repository Server <name> dialog.
- d. Enter the Administrator password for this machine (do not change the default port number 5001), and click OK.
- e. Double click the Repositories node.
- f. Double click the Oracle\_BI\_DW\_Base repository.

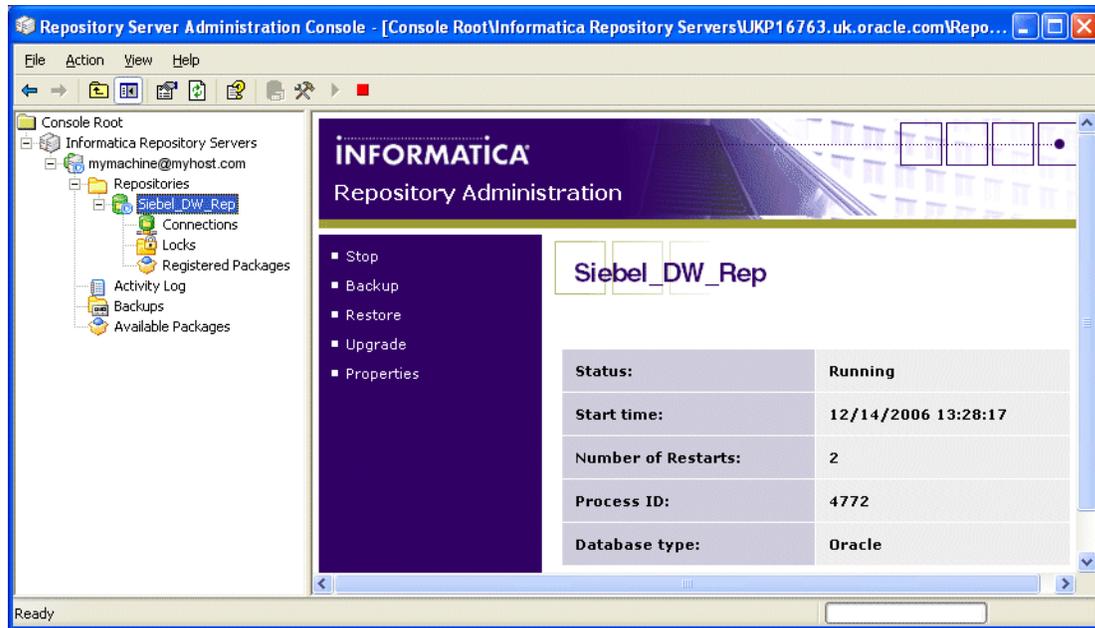


3. To stop the Informatica Repository, click the Stop option.
4. To start the Informatica Repository, click the Start option.

#### 4.14.6 How to Back Up and Restore the Informatica Repository (Optional)

To backup and restore the Informatica repository

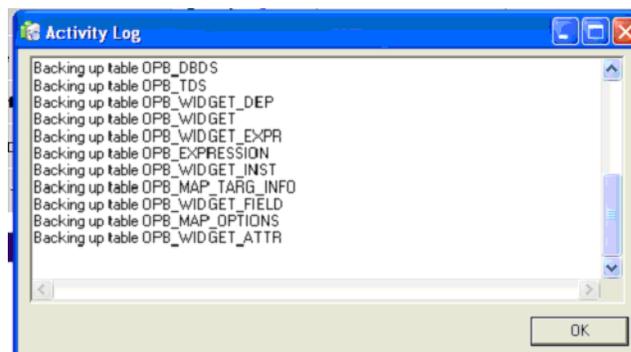
1. Start the Repository Server Administration Console in Informatica PowerCenter.
2. Display the Informatica Repository Administration dialog, as follows:
  - a. In the console navigator at the left hand side, double click the Informatica Repository Server to expand this node.
  - b. Right click on the server machine name that holds the Informatica Repository Server
  - c. Choose Action, then Connect to display the Connecting to Repository Server <name> dialog.
  - d. Enter the Administrator password for this machine (do not change the default port number 5001), and click OK.
  - e. Double click the Repositories node.
  - f. Double click the Oracle\_BI\_DW\_Base repository.



3. To backup the Informatica Repository, do the following:
  - a. Click the Backup option to display the Backup Repository dialog.
  - b. Enter the Informatica Repository Administrator username and password.
  - c. In the File Name field, enter a short name for the repository backup (for example, Inf\_rep\_backup).
  - d. Click OK to start the backup.

Informatica backs up the repository to the <DRIVE>:\Program Files\Informatica PowerCenter 7.1.4\RepositoryServer\bin\Backup directory.

4. To restore the Informatica Repository, do the following:
  - a. Right click on the Oracle\_BI\_DW\_Base node and click Delete to remove the Informatica Repository.
  - b. Click the Restore option to display the Restore Repository dialog.
  - c. Select a backup version of the Informatica Repository.
  - d. Click OK to start the procedure.
5. Check the Activity Log dialog to make sure that the backup or restore executes without errors.



## 4.15 How to Set the Code Page File Property for Unicode Environments on Teradata Databases

If you have a Unicode environment on a Teradata database, you need to set the code page file property of sessions that use the Teradata external loaders to use the code page UTF8. You need to follow this procedure to set the code page file property for each session that uses a Teradata external loader.

### To set the code page file property for a Unicode environment on a Teradata database

1. Launch the Informatica Workflow Manager.
2. Open the appropriate session.
3. In the Edit Tasks dialog box, select the Targets tab.
4. Click Set File Properties.
5. In the Flat Files - Targets dialog box, in the File Properties area, click Advanced.
6. In the Fixed Width Properties - Targets dialog box, in the Code Page area, select UTF-8 encoding of Unicode.

## 4.16 How to Set Up the HOSTS File for Teradata Installations on Windows

If you are using a Teradata database, you need to set the TDPID parameter on the machine where the Informatica Server is installed by adding an entry in the HOSTS file.

### To set up the HOSTS file for Teradata installations

1. On the machine where the Informatica Server is installed, go to the `\<SystemRoot>\system32\drivers\etc` directory and open the HOSTS file.
2. In the HOSTS file, enter a line in the following format:

```
<IP address of remote server> <remote server> <remote server>COP<n>
```

For example:

```
172.20.176.208 tdatsvr tdatsvrCOP1
```

where `tdatsvrCOP1` is the alias for the remote server. The alias must begin with an alphabetic string and end with the `COP n` suffix, where `n` is a number between 1 and the total number of applications processors that are associated with the Teradata communications processor.

3. Save the HOSTS file.

For more information about setting the TDPID parameter, see the Teradata documentation.

## 4.17 How to Verify the Informatica Repository Configuration

Follow this procedure to verify the Informatica repository configuration.

### To verify the Informatica repository configuration

1. Launch the Informatica Workflow Manager.
2. Double-click the `ORACLE_BI_DW_SERVER` icon that appears under the `Oracle_BI_DW_Base` icon.

- The Server dialog box opens.
3. Click Advanced to expose the Server Variables list.
  4. Verify the following settings:
    - Host Name
    - Port
    - Code Page
    - \$PMRootDir
  5. Close the Server dialog box.
  6. In the menu bar, select Connections, then Relational.  
The Relational Connection Browser window opens.
  7. Verify the DataWarehouse and OLTP connection settings.

## 4.18 How to Configure the DAC Server

This section provides the procedures for configuring the connection between the DAC server and the DAC metadata repository, and for setting up email recipients. You must perform this task on the machine hosting the DAC server. There should be one installation of the DAC server for each DAC repository.

This section includes the following procedures:

- [Section 4.18.1, "How to Configure the Connection Between the DAC Server and DAC Repository"](#)
- [Section 4.18.2, "How to Configure Email in the DAC Server \(Optional\)"](#)
- [Section 4.18.3, "How to Set DAC Source System Parameters"](#)
- [Section 4.18.4, "How to Start the DAC Server"](#)

### 4.18.1 How to Configure the Connection Between the DAC Server and DAC Repository

Follow this procedure to configure the connection between the DAC server and the DAC repository.

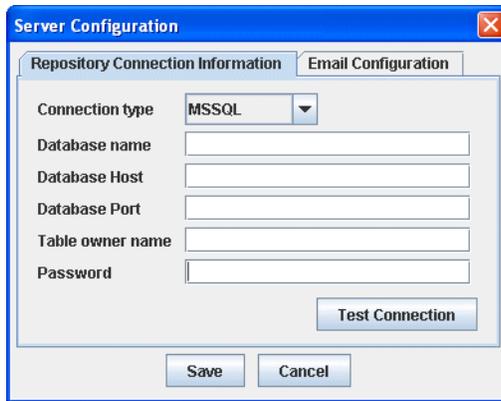
#### To configure the connection between the DAC server and the DAC repository

1. Launch the DAC client.
2. From the menu bar, click Tools, then DAC Server Management, then DAC Server Setup.

A confirmation dialog box asks you to confirm that you want to configure the DAC repository connection.

3. Click Yes.

The Server Configuration dialog box appears.



4. In the Repository Connection Information tab, enter the appropriate information, as described below.

---



---

**Note:** The DAC repository details that you specify here must match the DAC repository details that you specified in the following tasks:

When you created a database to store the DAC repository (for more information, see [Section 4.9, "How to Create the Required Databases"](#)).

When you created a DAC connection (for more information, see [Section 4.10.1, "How to Create a DAC Connection"](#)).

---



---

Field	Description
Connection type	Select the type of database in which you are storing the DAC metadata repository. Depending on what type you select, the connection details below change (see <i>Connection fields</i> below).

Field	Description
Connection fields (for example, Instance, TNS Name, Connection string/Database name).	<p>Specify connection details for the database that stores the DAC metadata repository.</p> <ul style="list-style-type: none"> <li>■ If you select Oracle (Thin), you are prompted below for the following information: <ul style="list-style-type: none"> <li>■ Instance (for example, mymachinename).</li> <li>■ Database Host (fully qualified, for example, mymachine.us.company.com).</li> <li>■ Database Port (for example, 1521).</li> <li>■ Table owner name, using the account that you created a database to store the DAC repository (for example, DAC). For more information, see <a href="#">Section 4.9, "How to Create the Required Databases"</a>.</li> <li>■ Password (that is, for the above database account).</li> </ul> </li> <li>■ If you select Oracle (OCI8), you are prompted below for a TNS name (for example, mymachinename@host.com).</li> <li>■ If you select DB2, you are prompted below for an Connection string.</li> <li>■ If you select MYSQL, you are prompted below for a Database name (for example, mydacdatabase).</li> </ul>
Table owner name	The database table name or instance (for example, DAC) that you created to store the DAC repository (for more information, see <a href="#">Section 4.9, "How to Create the Required Databases"</a> ).
Password	The database or instance password (for example, DAC).

5. Click Test Connection to make sure the DAC repository connection works.
6. Click Save.

#### 4.18.2 How to Configure Email in the DAC Server (Optional)

Follow this procedure to configure the email administrator account in the DAC, which enables the recipient to receive ETL status updates automatically. For example, the data warehousing administrator might want to be informed when an ETL routine has completed.

The DAC server has a built-in login-authentication based email (SMTP) client, which connects to any SMTP login-authenticating server to send emails.

---

**Note:** For the email functionality to work, you must be using an SMTP server in which the SMTP authentication mode LOGIN is enabled. For example, if you are using Microsoft Exchange Server, you must enable the SMTP module and enable authentication in the Basic Mode. The SMTP server outbound email authentication must be turned on for the DAC server to be able to send email notifications.

---

**To configure the email administrator account in the DAC server**

1. Launch the DAC client.
2. From the menu bar, click Tools, then DAC Server Management, then DAC Server Setup.

A confirmation dialog box asks you to confirm that you want to configure the DAC repository connection.

3. Click Yes.

The Server Configuration dialog box appears.

4. In the Email Configuration tab, enter the appropriate information, and click Save. Once the configuration has been completed, you can start the DAC server.

**4.18.3 How to Set DAC Source System Parameters**

You set Source System Parameters (also known as DAC ETL Preferences) in the DAC to specify how the ETL routines process data for a container. For example, if operating in the United States, you might set the \$\$DLFT\_COUNTRY to 'USA' to identify your data.

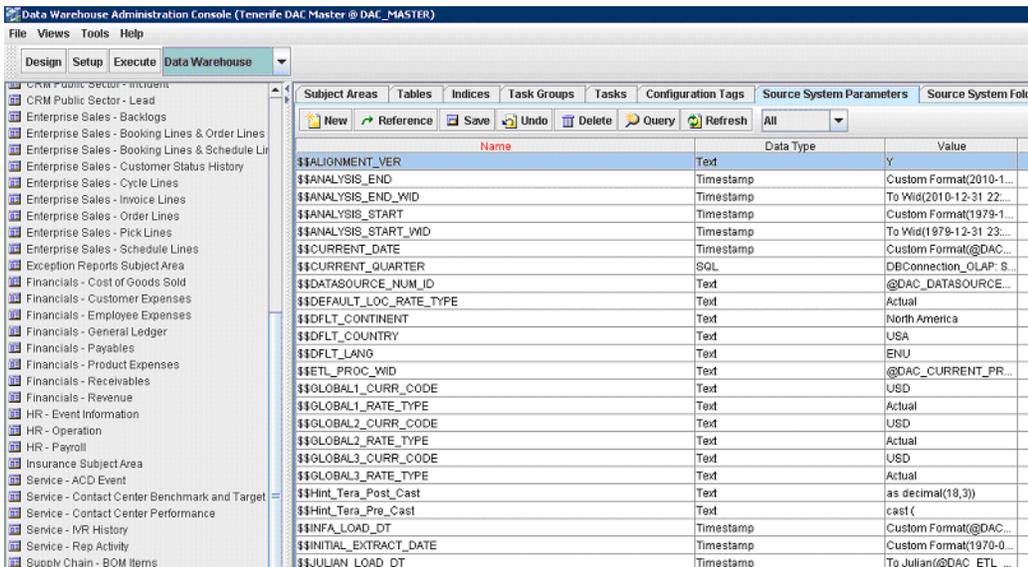
---

**Note:** The following preferences are applied to all tasks within a container. If extracting data from more than one source, these preferences will need to be re-applied to each associated container. They can be overridden at the task level by adding the parameter to the specific task and assigning a value there.

---

**To set DAC source system parameters**

1. Log into the DAC.
2. Display the Design view.
3. Make sure that you have selected the correct container from the *containers* drop down list.
4. Select the Source System Parameters tab.



---

**Note:** For information about recommended settings for specific databases, see [Chapter 3, "Pre-installation and Predeployment Requirements for Oracle BI Applications"](#).

---

5. Use the Edit tab below the list of Source System Parameters to change the value of parameters in the list.
6. Click Save.

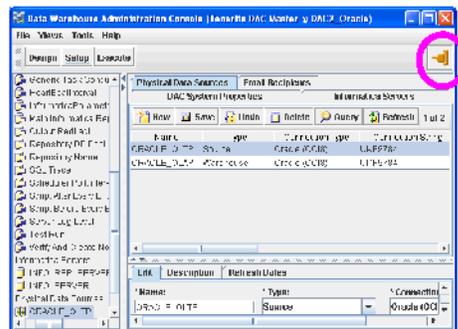
#### 4.18.4 How to Start the DAC Server

You start the DAC server on the Windows machine that is used to run the DAC client. Follow this procedure to start the DAC Server.

##### To start the DAC Server

1. Select Oracle Business Intelligence, then Oracle DAC, and then Start DAC Server.

**Tip:** Use the DAC Server status icon in the DAC console to make sure that the DAC Server has started. The DAC Server status icon should either be orange (idle) or green (active). The screen shot below shows the DAC Server status icon highlighted.



#### 4.19 How to Start the Informatica Server

At this point in the installation and configuration process, you can start the Informatica Server. It should now be able to communicate with the Informatica Repository Server and the Informatica repository.

##### To start the Informatica Server

1. Navigate to Start, then Settings, then Control Panel, then Administrative Tools, then Services.
2. Double-click Informatica.
 

The Informatica Properties dialog box opens.
3. In the General tab, in the Startup Type drop-down list, select Automatic.
4. In the Log On tab, if you plan on using email notification, enter the appropriate username and password. Select System Account if you do not plan on using email notification.
5. In the General tab, in the Server Status area, click Start.
6. Close the Services window.

## 4.20 How to Update Siebel Transactional Database Schema Definitions

---

**Note:** You must perform the following steps for each Siebel source database type.

---

This section explains how to update Transactional Database Schema Definitions, and contains the following sections:

- [Section 4.20.1, "How to Run the DDLIMP Tool From The Command Line"](#)
- [Section 4.20.2, "How to Apply Siebel \(CRM\) and Oracle E-Business Suite Schema Changes For Oracle, DB2/UDB, MSSQL"](#)
- [Section 4.20.3, "How to Apply CRM Schema Changes For DB2 on 390 or z/os"](#)
- [Section 4.20.4, "How to Verify the Siebel \(CRM\) and Oracle E-Business Suite Schema Changes"](#)
- [Section 4.20.5, "About Delete Triggers"](#)

### 4.20.1 How to Run the DDLIMP Tool From The Command Line

When you use the DDLIMP utility from a command line to update schema definitions, refer to the following notes:

To run DDLIMP from command line, run the following command:

```
\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC_CSN>
/GSSE_ROLE /F <..\OracleBI\dwrep\DDL_OLTP.CTL> /L
<..\oracleBI\dwrep\DDL_OLTP.log>
```

NOTE: For Oracle, please use the Siebel Merant ODBC Drivers. For MSSQL you need to use both flags /w and /Z. The syntax for MSSQL Server is:

```
\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC_CSN>
/GSSE_ROLE /W Y /Z Y /F <..\OracleBI\dwrep\DDL_OLTP.CTL>
/L<..\oracleBI\dwrep\DDL_OLTP.log>
```

Additionally you can use the following commands:

/W Y (if the OLTP database is Unicode).

/Z Y (if the OLTP database is DB2 and Unicode).

/B <TABLE\_SPACE\_NAME> if you want to create these table in a separate table space.

/X <INDEX\_TABLE\_SPACE\_NAME> if you want to create the indexes in a separate table space.

/Y Storage File for DB2/390.

### 4.20.2 How to Apply Siebel (CRM) and Oracle E-Business Suite Schema Changes For Oracle, DB2/UDB, MSSQL

To enable change capture for Oracle's Siebel adapters, you use the ddlimp control file to apply Siebel (CRM) and Oracle E-Business Suite schema changes, which updates the required image tables in the OLTP.

---



---

**Note:** Using the ddlimp control file to apply schema changes replaces the use of SIF files in Siebel Tools in previous product releases.

---



---

### How to Apply Oracle's CRM Schema Changes For Oracle, DB2/UDB, MSSQL

1. On the machine where Oracle BI Applications is installed, run the following command:

```
..\OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC connect string>
/G SSE_ROLE /F <..\OracleBI\dwrep\DDL_OLTP.CTL> /L <..\OracleBI\dwrep\DDL_
OLTP.log>
```

---



---

**Note:** If you are applying schema changes to a Siebel CRM Version 6.3, specify DDL\_OLTP\_63.CTL as the /F parameter instead of DDL\_OLTP.CTL.

---



---

For example:

```
DDLIMP /U SADMIN /P SADMIN /C SIEBEL_OLTP /G SSE_ROLE /F
<DRIVE>:\OracleBI\dwrep\DDL_OLTP.CTL /L <DRIVE>:\OracleBI\dwrep\DDL_OLTP.log
```

Notes:

- /P <PASSWORD> - The password for Oracle's CRM OLTP.
- /C <ODBC connect string> - The name of the ODBC connect string.
- For Oracle databases, use the Oracle Merant ODBC Drivers (installed with Oracle BI Applications).
- In addition, you can use the following commands:
  - /W Y - (if the OLTP database is Unicode).
  - /Z Y - (if the OLTP database is DB2 and Unicode or the OLTP database is MS SQL Server and Unicode).
  - /B <TABLE\_SPACE\_NAME> - If you want to create these table in a separate table space.
  - /X <INDEX\_TABLE\_SPACE\_NAME> - If you want to create the indexes in a separate table. space.
  - /Y - Storage File for DB2/390.

2. Restart all servers.

### 4.20.3 How to Apply CRM Schema Changes For DB2 on 390 or z/os

How to apply CRM schema changes for DB2 on 390 or z/os

1. Edit the following parameters in the Storage control files located in \OracleBI\dwrep\Storeage\_DDL\_OLTP.CTL:
  - %1 - Replace with a 2 character database name.
  - %indBufPool - Replace it with a index buffer pool name.
  - %4kBulfpool - Replace it with a 4k TBS buffer pool name.
  - %32kBulfpool - Replace it with a 32K TBS Buffer Pool name.

2. On the machine where Oracle BI Applications is installed, run the following command:

```
.. \OracleBI\dwrep\bin\DDLIMP /U <USER> /P <PASSWORD> /C <ODBC_CSN> >>> /G SSE_ROLE /F \OracleBI\dwrep\DDL_OLTP_DB2390.CTL> /L <..\oracleBI\dwrep\DDL_OLTP.log> /5 Y /A <SCHEMA_OWNER> /Y ..\OracleBI\dwrep\Storeage_DDL_OLTP.CTL
```

Notes:

- /P <PASSWORD> - The password for Oracle's CRM OLTP.
  - /C <ODBC connect string> - The name of the ODBC connect string.
  - For Oracle databases, use the Oracle Merant ODBC Drivers.
  - In addition, you can use the following commands:
    - /W Y - (if the OLTP database is Unicode).
    - /Z Y - (if the OLTP database is DB2 and Unicode or the OLTP database is MS SQL Server and Unicode).
    - /B <TABLE\_SPACE\_NAME> - If you want to create these table in a separate table space.
    - /X <INDEX\_TABLE\_SPACE\_NAME> - If you want to create the indexes in a separate table. space.
    - /Y - Storage File for DB2/390.
3. To create indices for the tables created, run the following SQL Script from the DB2 command line utility connected to your OLTP Schema:  

```
\OracleBI\Dwrep\Create_OLTP_Db2390_index.sql
```

#### 4.20.4 How to Verify the Siebel (CRM) and Oracle E-Business Suite Schema Changes

After applying Siebel (CRM) and Oracle E-Business Suite schema changes, you need to verify that appropriate tables were created in the transactional database.

**To verify Siebel (CRM) and Oracle E-Business Suite schema changes**

1. For Oracle's Siebel Business Applications, make sure that the following tables were created in the transactional database:
  - S\_ETL\_R\_IMG\_1 through S\_ETL\_R\_IMG\_166
  - S\_ETL\_I\_IMG\_1 through S\_ETL\_D\_IMG\_166
  - S\_ETL\_D\_IMG\_1 through S\_ETL\_D\_IMG\_166
  - S\_ETL\_PARAM
  - S\_ETL\_PRD\_ATTR
  - S\_ETL\_PRD\_REL

#### 4.20.5 About Delete Triggers

---

---

**Note:** Delete triggers are only used supported with Siebel CRM database.

---

---

Delete records in Siebel CRM sources are not propagated to the Data Warehouse tables. However the mechanism to identify the delete records is provided, as follows:

The DAC can create delete triggers on source tables (please refer to *Oracle Business Intelligence Data Warehouse Administration Console User Guide*). These triggers write the primary keys of deleted records with the flag "D" in the corresponding S\_ETL\_I\_IMG table. You need to write a custom SDE extract to pull these deleted primary keys from the image table and take corresponding action on the Datawarehouse table.

## 4.21 How to Create Stored Procedures for DB2-UDB

The DAC uses siebstat and siebtrun stored procedures when running ETL processes. Typically, these stored procedures are available in your transactional database; they might not be available on the data warehouse database.

If you need to install the DB2 stored procedures manually, they are located in the installation directory \OracleBI\dwrep\siebproc\db2udb\. There is a sub-directory for each platform. For example, stored procedures for the Windows platform are stored in the sub-directory \OracleBI\dwrep\siebproc\db2udb\win32\.

Each platform-specific directory contains the following sub-directories:

- \siebproc\ (containing stored procedures for a 32-bit DB2 environment)
- \siebproc64\ (containing stored procedures for a 64-bit DB2 environment)

These directories also contain the files siebproc.sql and sqlproc.ksh, which are used to create the function. For more information, see [Section 4.21.2, "How to Create DB2 stored procedures"](#).

### 4.21.1 How to Verify the existence of stored procedures

This section describes how to verify the existence of stored procedures.

**To verify the existence of these procedures, follow the steps below.**

- From a DB2 command prompt or control center, issue the following SQL scripts:

```
db2 => create table test_siebproc(id int);
DB20000I The SQL command completed successfully.

db2 => insert into test_siebproc(id) values (1);
DB20000I The SQL command completed successfully.

db2 => call siebtrun('TEST_SIEBPROC');
SQL0805N Package "NULLID.SYSSH200" was not found. SQLSTATE=51002
```

---

**Note:** If you get an error message, you do not have the required stored procedures.

---

Creating stored procedures must be performed by the database administrator (for more information, see [Section 4.21.2, "How to Create DB2 stored procedures"](#)).

### 4.21.2 How to Create DB2 stored procedures

This section describes how to create DB2 stored procedures.

**To create DB2 stored procedures**

1. Copy the DB2 stored procedure directory (i.e. \siebproc\ or \siebproc64\) from the appropriate platform-specific directory to a directory on the DB2 server side.

For example, for a 32-bit DB2 environment on a Windows platform, you might copy the directory \OracleBI\dwrep\siebproc\db2udb\win32\siebproc\ to the directory d:\Program Files\SQLLIB\function\ on the DB2 server side.

---

**Note:** For more information about the location of DB2 stored procedures, see [Section 4.21, "How to Create Stored Procedures for DB2-UDB"](#).

---

2. If you copied stored procedures for a 64-bit DB2 environment, on the DB2 server side, rename the \siebproc64\ directory to \siebproc\.

For example, if you copied stored procedures to d:\Program Files\SQLLIB\function\siebproc64\, rename this directory to d:\Program Files\SQLLIB\function\siebproc\.

Once these procedures are created, you can verify that they exist. After the test is complete, you can drop the table TEST\_SIEBPROC.

## 4.22 Teradata-Specific Installation and Configuration Checklist (Windows)

The table below provides a list of Teradata-specific installation and configuration steps that are performed during the Oracle BI Applications installation and configuration process. You should review this list to make sure that you have performed all of the required Teradata-specific steps.

**Table 4-3 Teradata-Specific Installation Checklist**

Action	Link to Topic
Review Teradata-specific database guidelines.	<a href="#">Section 3.9, "Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse"</a> .
Activate join indices.	<a href="#">Section 4.11.4, "How to Activate Join Indices for Teradata Databases"</a> .
Create or drop data warehouse tables.	<a href="#">Section 4.12.3, "How to Create Data Warehouse Tables on a Teradata Database"</a> .
Use recommended name for Informatica repository.	<a href="#">Section 4.5, "How to Set Up the Informatica Server and Informatica Repository Server"</a> .
Create or restore the Informatica repository using the Informatica Repository Manager. You cannot create or restore the Informatica repository using the DAC.	<a href="#">Section 4.14.2, "How to Register the Informatica Repository Server"</a> .
Configure the source and data warehouse database relational connections in the Informatica Workflow Manager. You cannot use the DAC's Data Warehouse Configurator to configure relational connections.	<a href="#">Section 3.9, "Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse"</a> .
Set the Code Page File Property for Unicode Environments.	<a href="#">Section 4.15, "How to Set the Code Page File Property for Unicode Environments on Teradata Databases"</a> .
Set up the HOSTS file.	<a href="#">Section 4.16, "How to Set Up the HOSTS File for Teradata Installations on Windows"</a> .

## 4.23 Upgrade Issues for Oracle BI Applications

This section contains miscellaneous upgrade issues and workarounds, and contains the following topics:

- [Section 4.23.1, "Moving From a Test Environment to a Production Environment"](#)
- [Section 4.23.2, "Migrating Seed Data from an Siebel Source System to the Oracle BI Repository"](#)

### 4.23.1 Moving From a Test Environment to a Production Environment

Oracle Business Intelligence Applications 7.9.x uses sequence generator transformation in Informatica rather than a re-usable expression generated ROW\_WID. The sequence generators in the 7.9.x mappings do not look up the target table to identify the Max(ROW\_WID), and instead store the current ROW\_WID value for each row inserted. To see how this works in 7.9.x, please review for example SIL\_OrderFact mapping. This contains a mapplet mplt\_SIL\_OrderFact, which in turn contains a sequence generator transformation Seq\_W\_Order\_F\_WID.

Using sequence generator transformation information in Informatica can cause Key index violation database error (on Max(ROW\_WID)) if an Informatica repository is moved from a Development/Test environment to a Production environment.

Since the data in the two different environments might not be in sync, the target tables in the OLAP database may contain different ROW\_WIDs for the same table in each environment. As a result, when the Informatica repository is moved for instance from Test to Production, the sequence generator transformations in the mappings hold on to the ROW\_WIDS generated in Test environment, when generating the next set of ROW\_WIDS for the Production environment. These could very well be duplicate values already existing in the Production environment, thus resulting in primary key violations (on ROW\_WID column) during data load.

Typical symptoms include a database error pointing to Primary Key violation on ROW\_WID column during data load.

The workaround for this issue is to run the Reset\_infa\_seq\_gen.bat script to initialize the Informatica sequence generator for incremental runs, as follows:

1. Navigate to the directory \OracleBI\dwrep\Upgrade\DbScripts\- 2. Open the Reset\_infa\_seq\_gen.bat script, and review/customize the parameters at the top of the file.

In circumstances where this step was omitted after migrating the repository, and the ETL has been run, the index violation prevents data corruption.

3. Re-queue the failed DAC Tasks, to allow the ETL to continue.

### 4.23.2 Migrating Seed Data from an Siebel Source System to the Oracle BI Repository

This section describes how to move localized string content from the S\_MSG table in a Siebel OLTP database to the W\_LOCALIZED\_STRING\_G table in the Business Analytics Warehouse database. Use the following steps:

1. Export the customized localization data in the S\_MSG table using the DATAEXP command from a command prompt, as shown in the following sample text:

```
DATAEXP /U <Username for OLTP read access> /P <Password> /C
<Siebel_OLTP odbc> /D <Table Owner> /F CUSTSTRINGS.DATA /W Y
/L <EXPCUST.LOG>
```

2. Respond to the prompts as follows:
  - When you are prompted for the TABLE\_NAME to export, enter S\_MSG and press ENTER.
  - When you are prompted for the WHERE CLAUSE for filtering records from the table, enter WHERE LAST\_UPD > '1/1/1980' and press ENTER twice.
3. A message is displayed that indicates the number of rows that were exported. Press ENTER to enter the Database Table Name and to end the process.
4. Import the data into the W\_LOCALIZED\_STRING\_G table using the data file that you generated in the previous step.

For information about importing localization data into the W\_LOCALIZED\_STRING\_G table, see [Section B.1.1, "Upgrading Oracle Business Intelligence Seed Data for Non-English Locales"](#).

Use the /R 1000000 argument in the DATAIMP command to insert the customized strings beyond Row Id 1000000.
5. Import the localization data into the W\_LOCALIZED\_STRING\_G table.

---

---

## Installing Oracle BI Applications on Unix

This chapter provides instructions for installing and configuring Oracle Business Intelligence Applications components in a UNIX environment. It contains the following topics:

- [Section 5.1, "Overview of Installing and Configuring Oracle BI Applications on UNIX"](#)
- [Section 5.2, "Installation and Configuration Process Task List for UNIX"](#)
- [Section 5.3, "How to Install Oracle BI Applications on UNIX"](#)
- [Section 5.4, "How to Install and Configure the Informatica Repository Server on UNIX"](#)
- [Section 5.5, "How to Install and Configure the Informatica Server on UNIX"](#)
- [Section 5.6, "How to Update the Informatica License Files on UNIX"](#)
- [Section 5.7, "How to Install the Informatica Patch on UNIX"](#)
- [Section 5.8, "How to Create Database Connections for Teradata Installations"](#)
- [Section 5.9, "How to Set Environment Variables for the Informatica Server on UNIX"](#)
- [Section 5.10, "How to Set Up the HOSTS File for Teradata Installations on UNIX"](#)
- [Section 5.11, "How to Start the Informatica Repository Server on UNIX"](#)
- [Section 5.12, "How to Stop the Informatica Repository Server on UNIX"](#)
- [Section 5.13, "About the DAC Server on UNIX"](#)
- [Section 5.14, "How to Stop and Start the Informatica Server on UNIX"](#)
- [Section 5.14.3, "How to Start the Informatica Server On UNIX"](#)
- [Section 5.15, "How to Run the DAC Server on UNIX"](#)
- [Section 5.16, "How to Stop the DAC Server on UNIX"](#)
- [Section 5.17, "Teradata-Specific Installation and Configuration Checklist for UNIX"](#)

---

---

**Note:** If you are installing Oracle Business Intelligence Applications in a UNIX environment, you must use the Unicode character data mode. For more information about Unicode, see [Section 3.10, "Deploying Oracle Business Analytics Warehouse in Latin-1 General, Unicode and Non-English Environments"](#).

---

---

## 5.1 Overview of Installing and Configuring Oracle BI Applications on UNIX

Before you begin the installation and configuration process on UNIX, you should review the following:

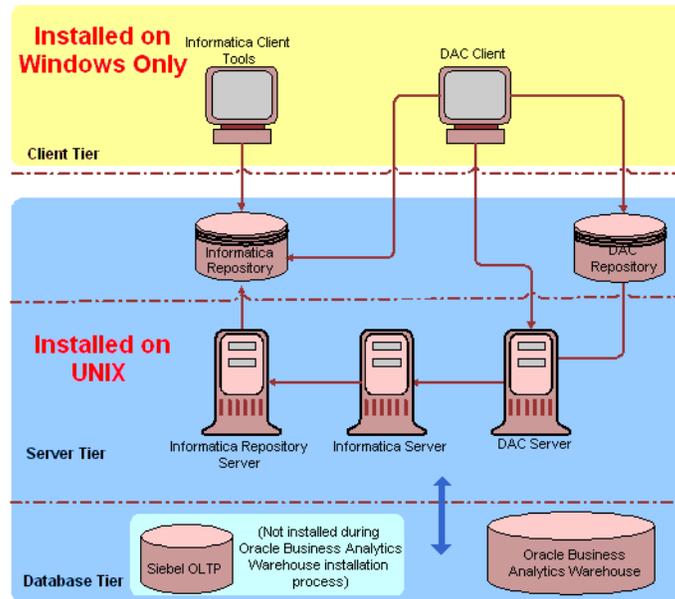
- [Chapter 3, "Pre-installation and Predeployment Requirements for Oracle BI Applications"](#)
- [Chapter 4, "Installing and Configuring Oracle BI Applications on Windows"](#)

When you install Oracle BI Applications in a UNIX environment, note the following:

- Some of the Oracle BI Applications and Informatica components can only be installed on Windows. For example, the Informatica client tools, and the DAC client tool. For more information, see the figure below.
- Although you can install Informatica server components and DAC server components on UNIX machines, Oracle does not provide a separate Oracle BI Applications installer for UNIX. Therefore, to install these component in a UNIX environment, you must first install Oracle BI Applications and the Oracle BI infrastructure onto a Windows machine, then manually copy over the installation directories to a UNIX machine. For more information, see [Section 5.2, "Installation and Configuration Process Task List for UNIX"](#).
- If your source or target database is Microsoft SQL Server, you will need to use DataDirect ODBC driver for Microsoft SQL Server supplied by Informatica. This can be downloaded directly from Informatica Web site or you can contact Oracle Support for further assistance.

The figure below shows a typical deployment of Oracle BI Applications in a UNIX installation. The figure shows how the Informatica client tools and the DAC client are installed on Windows only, and how the other Oracle BI Applications components can be installed on UNIX machines. You will complete some of the configuration tasks in the Windows environment. [Table 5-1](#) lists each task in the installation and configuration process and tells you in which environment to complete the task.

**Figure 5–1 Example hardware configuration for an Oracle BI Applications application on UNIX**



## 5.2 Installation and Configuration Process Task List for UNIX

The Oracle Business Intelligence Applications installation and configuration process on UNIX consists of the following steps. Complete each of the following tasks in the order listed below.

**Table 5–1 UNIX Installation and Configuration Process**

Task	Instructions	Environment
1. Install Oracle BI Applications on Windows.	Section 4.3, "How to Install Oracle Business Intelligence Applications (Windows)".	Windows
2. Install Oracle BI Applications on UNIX by copying across the installation files from a Windows installation directory.	Section 5.3, "How to Install Oracle BI Applications on UNIX".	UNIX
2. Install Oracle BI Applications on UNIX by copying across the installation files from a Windows installation directory.	Section 4.4, "How to Install Informatica PowerCenter and Required Patches (Windows)"	Windows
4. Download and copy Hibernate libraries. <b>Note:</b> Hibernate libraries must be copied on to each DAC Client machine (Windows only) and DAC Server machine (Windows or UNIX).	Section 4.7, "How to Copy Hibernate Libraries".	Windows (DAC client) UNIX (DAC server)
5. Install and Configure the Informatica Repository Server on UNIX (using the Informatica Installer).	Section 5.4, "How to Install and Configure the Informatica Repository Server on UNIX".	UNIX

**Table 5–1 (Cont.) UNIX Installation and Configuration Process**

<b>Task</b>	<b>Instructions</b>	<b>Environment</b>
6. Install and Configure the Informatica Server on UNIX (using the Informatica Installer).	<a href="#">Section 5.5, "How to Install and Configure the Informatica Server on UNIX"</a> .	UNIX
7. Update the Informatica license files.	<a href="#">Section 5.6, "How to Update the Informatica License Files on UNIX"</a> .	UNIX
8. Install the Informatica patch.	<a href="#">Section 5.7, "How to Install the Informatica Patch on UNIX"</a> .	UNIX
9. Set Informatica Data Code Pages Validation.	<a href="#">Section 4.5.2, "How to Set Up Informatica Data Code Pages Validation"</a> .	Windows
10. Install the Java SDK and JDBC drivers on the DAC client machine and the DAC server machine.	<a href="#">Section 4.8, "How to Configure the DAC Client"</a> .	Windows (DAC client) UNIX (DAC server)
11. Configure the DAC config.bat file.	<a href="#">Section 4.8.1, "How to Configure the DAC Config.bat File"</a> .	Windows
12. Create the DAC Metadata Repository (using the DAC client).	<a href="#">Section 4.10, "How to Create the DAC Metadata Repository"</a> .	Windows
13. Configure the DAC Metadata Repository (using the DAC client and ODBC Connections software).	<a href="#">Section 4.11.1, "How to Set Up DAC System Properties"</a> .	Windows
	<a href="#">Section 4.5.7, "How to Create ODBC Database Connections"</a> .	Windows
	<a href="#">Section 4.11.2, "How to Specify Transactional and Data Warehouse Data Sources in the DAC"</a> .	Windows
	<a href="#">Section 5.8, "How to Create Database Connections for Teradata Installations"</a> .	UNIX
	<a href="#">Section 4.11.3, "How to Configure Email Recipients in the DAC Client"</a> .	Windows
	<a href="#">Section 4.11.4, "How to Activate Join Indices for Teradata Databases"</a> .	Windows
14. Configure the Oracle Business Analytics Warehouse (using the DAC client).	<a href="#">Section 4.12.1, "How to Create the SSE Role"</a> .	Windows
	<a href="#">Section 4.12.2, "How to Create Data Warehouse Tables"</a> .	Windows
	<a href="#">Section 4.12.3, "How to Create Data Warehouse Tables on a Teradata Database"</a> .	Windows
15. Set environment variables for the Informatica Server.	<a href="#">Section 5.9, "How to Set Environment Variables for the Informatica Server on UNIX"</a> .	UNIX
16. Set the Windows PATH environment variable for the Informatica Server.	<a href="#">Section 4.6, "How to Set Environment Variables on Windows"</a> .	Windows

**Table 5–1 (Cont.) UNIX Installation and Configuration Process**

<b>Task</b>	<b>Instructions</b>	<b>Environment</b>
17. Start the Informatica Repository Server.	Section 5.11, "How to Start the Informatica Repository Server on UNIX".	UNIX
18. Create or restore and configure the Informatica repository (using the DAC).	Section 4.14.6, "How to Back Up and Restore the Informatica Repository (Optional)".  See also Section 4.14.5, "How to Stop and Start the Informatica Repository (Optional)".  Section 4.5, "How to Set Up the Informatica Server and Informatica Repository Server".	Windows
	Section 4.14.2, "How to Register the Informatica Repository Server".	Windows
	Section 4.14.5, "How to Stop and Start the Informatica Repository (Optional)".	Windows
	Section 4.14.2, "How to Register the Informatica Repository Server".	Windows
	Section 4.14.3, "How to Load the Pre-Built Repository Into Informatica".	Windows
	Section 4.14.4, "How to Configure the Informatica Repository in Workflow Manager".	Windows
19. Set the Code Page File Property for Unicode Environments on Teradata Databases.	Section 4.15, "How to Set the Code Page File Property for Unicode Environments on Teradata Databases".	Windows
20. Set Up the HOSTS File for Teradata Installations.	Section 5.10, "How to Set Up the HOSTS File for Teradata Installations on UNIX".	UNIX
21. Verify the Informatica Repository configuration (using Informatica tools).	Section 4.17, "How to Verify the Informatica Repository Configuration".	Windows
22. Start the Informatica Server.	Section 5.14, "How to Stop and Start the Informatica Server on UNIX".	UNIX
23. Set the Source System Parameters (also known as DAC ETL Preferences) using the DAC client.	Section 4.18.3, "How to Set DAC Source System Parameters".	Windows
24. Configure the DAC server. <b>Note:</b> Before you start, make sure that you have copied over the \OracleBI\DAC\ directory from the Windows machine where you installed Oracle BI Applications to a UNIX machine.	Section 5.13, "About the DAC Server on UNIX".	UNIX
25. Update the transactional database schema definition.	Section 4.20, "How to Update Siebel Transactional Database Schema Definitions".	Windows

## 5.3 How to Install Oracle BI Applications on UNIX

Before you install Oracle Business Intelligence Applications on UNIX you must first complete the following installations:

- Install the Oracle Business Intelligence infrastructure on Windows.  
For instructions, see *Oracle Business Intelligence Platform Installation and Configuration Guide*.
- Install the Oracle Business Intelligence infrastructure on UNIX.  
For instructions, see *Oracle Business Intelligence Platform Installation and Configuration Guide*.
- Install Oracle Business Intelligence Applications on Windows.  
For instructions, see [Section 4, "Installing and Configuring Oracle BI Applications on Windows"](#).

### To install Oracle BI Applications on UNIX

1. Copy the directory OracleBI\DAC\ from the Oracle Business Intelligence Applications installation directory in the Windows environment to the UNIX environment.

---



---

**Note:** You also need to copy the OracleBI\DAC\ directory onto any machine on which you want to run the DAC Server (for more information, see [Section 5.13, "About the DAC Server on UNIX"](#)).

---



---

2. Copy the directory OracleBI\dwrep\Informatica from the Oracle Business Intelligence Applications installation directory in the Windows environment to the UNIX environment. Use the same directory structure in the UNIX environment as the Windows environment.

This action copies pmrep, the Informatica repository file (Oracle\_BI\_DW\_Base.rep), and the parameter files to the UNIX environment.

3. Copy the following files from the Windows environment into the UNIX environment. Use the same directory structure in the UNIX environment as the Windows environment.

File Type	File Name	File Location
DAC repository	*.*	OracleBI\DAC\export
DAC application related	*.*	OracleBI\DAC\CustomSQLs
Delete trigger	DeleteTriggers.list	OracleBI\DAC\conf
Analytics repository	OracleBIAnalyticsApps.rpd	OracleBI\Server\Repository
Presentation Catalog	*.*	OracleBIData\Web\Catalog
		<b>Note:</b> You need to un-zip the ZIP files and copy the un-zipped files to the Unix machine.
Language	*.*	OracleBIData\Web\Res

File Type	File Name	File Location
Informatica seed files and lookup files	*.*	OracleBI\dwrep\Informatica\SrcFiles and OracleBI\dwrep\Informatica\LkpFiles  <b>Note:</b> You need to copy these files to the location specified as \$PMSourceFileDir in Informatica Workflow Manager.

## 5.4 How to Install and Configure the Informatica Repository Server on UNIX

If you are using a non-ENU operating system, make sure you have all the required locale files installed.

Install the Informatica software directly from the CD-ROM. If you cannot install the software directly from the CD-ROM, you must use FTP in binary mode; otherwise, the installation will fail. After the FTP process, use the `chmod` command to change the permission setting for all install files to read, write, and execute.

---

**Note:** If you want to perform ETL runs in Informatica on HP-UX Itanium 2 64-bit, you need to install one of the following patches:

HP-UX 11.11 Itanium 64-bit systems require Patch PHSS\_33033 or higher.

HP -UX 11.23 Itanium 64-bit systems require Patch PHSS\_33349 or higher.

If you do not install one of the patches listed above, ELT runs might fail.

---

To install and configure the Informatica Repository Server on Unix, do the following:

1. Install and configure the Informatica Repository Server (for more information, see [Section 5.4.1, "How to install and configure the Informatica Repository Server"](#)).
2. Specify the Informatica Repository Server as an environment variable (for more information, see [Section 5.4.2, "How to specify the Informatica Repository Server as an environment variable"](#)).

### 5.4.1 How to install and configure the Informatica Repository Server

To install and configure the Informatica Repository Server

1. Log onto the UNIX machine.
2. Change to the appropriate directory and run the installer by typing `./install`.
3. Select the language in which you want to perform the installation.
4. Enter the Product license key.
5. Accept the license agreement.
6. Select Informatica Repository Server from the list of components available to install.
7. Specify the directory where the Repository Server will be installed.

8. Enter the appropriate values for the options listed below.

Option	Required/Optional	Description
Configuration file name	Required	The name of the Repository Server configuration file.
ServerPort	Required	The port number the Repository Server uses for repository client connections. The default is 5001.
AdminPasswd	Required	The administrative password used to access the Repository Server. It is recommended that you use Administrator as the password.
RaMinPort	Required	The minimum port number the Repository Server can assign to the Repository Agent process. The default is 5002.
RaMaxPort	Required	The maximum port number the Repository Server can assign to the Repository Agent process. The default is 65535.
ConfigDir	Required	The name of the directory in which the Repository Server stores repository configuration information files. You can specify either a relative path or an absolute path. The default is Config.
BackupDir	Required	The name of the directory in which the Repository Server stores repository backup files. You can specify either a relative path or an absolute path. The default is Backup.
PluginDir	Required	The name of the directory in which the Repository Server stores repository plugin files. You can specify either a relative path or an absolute path. The default is Plugin.
ErrorSeverityLevel	Required	<p>The level of error messages written to the Repository Server log. Specify one of the following message levels:</p> <ul style="list-style-type: none"> <li>■ Error. Writes ERROR code messages to the log.</li> <li>■ Warning. Writes WARNING and ERROR code messages to the log.</li> <li>■ Information. Writes INFO, WARNING, and ERROR code messages to the log.</li> <li>■ Tracing. Writes TRACE, INFO, WARNING, and ERROR code messages to the log.</li> <li>■ Debug. Writes DEBUG, TRACE, INFO, WARNING, and ERROR code messages to the log.</li> </ul> <p>The default is Information</p>
LogFileName	Required	The name of the Repository Server log file. The default is pmrepsserver.log.

9. Type Y if you want to view the readme file, or N to read it later.

10. Press enter to continue to install the Informatica Server.

Do not exit from the installer. Proceed to [Section 5.5, "How to Install and Configure the Informatica Server on UNIX"](#).

## 5.4.2 How to specify the Informatica Repository Server as an environment variable

### To specify the Informatica Repository Server as an environment variable

1. From the \DAC\ directory, open the dac\_env.sh or dac\_env.csh file in a text editor.
2. Edit the value of the following environment variables by specifying the path of the Informatica Repository:
  - setenv PM\_HOME=<Informatica repository path>
  - setenv PM\_REPHOME=<Informatica repository path>
3. Save and close the dac\_env.sh or dac\_env.csh file.
4. Execute the 'source' command to populate the environment variables using the values specified in the dac\_env.sh or dac\_env.csh file. For example, at a command prompt, type 'source dac\_env.sh'.
5. Restart the Informatica Repository server.

## 5.5 How to Install and Configure the Informatica Server on UNIX

The instructions in this section for installing and configuring the Informatica Server are meant to be followed immediately after the Informatica Repository Server installation, without exiting the installer. If you exited the installer after installing the Repository Server, you may want to review [Section 5.4, "How to Install and Configure the Informatica Repository Server on UNIX"](#).

---

---

**Note:** You can install and configure two Informatica Server instances on the same UNIX machine. This configuration requires two copies of pmserver.cfg and pmrepserver.cfg files. Each file must have a unique name, for example, pmserverdev.cfg and pmserverQA.cfg. Each config file will define unique ports for each instance of the Informatica Server.

---

---

---

---

**Note:** Even if you run the Informatica server from a different machine to the DAC server machine, you still must install Informatica on the DAC server machine, because the DAC server requires the Informatica libraries (utility pmcmd).

---

---

---



---

**Note:** If you want to perform ETL runs in Informatica on HP-UX Itanium 2 64-bit, you need to install one of the following patches:

For HP-UX 11.11, apply patch PHSS\_33033.

For HP-UX 11.11, apply patch PHSS\_33033.

If you do not install one of the patches listed above, the ELT run might fail with one of the following errors:

```
FATAL ERROR: Caught a fatal signal [11]. Will abort
this server process shortly. INFO: CMN_1131:
(3644|1) System shared memory [360454] for [Load
Manager Shared Memory] removed.
```

```
FATAL ERROR: Aborting this server process due to a
fatal signal [11].
```

---



---

### To install and configure the Informatica Server

1. Select Informatica PowerCenter Server from the list of components available to install.
2. Specify the directory where the Informatica Server will be installed.
3. Enter the appropriate values for the options listed below.

Option	Required/Optional	Description
Configuration file name	Required	The name of the Informatica Server configuration file.
ServerName	Required	The name of the Informatica Server registered with the repository. It is recommended that you use ORACLE_BI_DW_SERVER.
RepositoryName	Required	The repository name entered when creating or upgrading a repository. It is recommended that you use Oracle_BI_DW_Base.
PMUser	Required	The repository user name used to connect to the repository. It is recommended that you use Administrator
PMPassword	Required	The password corresponding to the repository user (PMUser). It is recommended that you use Administrator.
RepServerHostName	Required	The host name of the machine hosting the Repository Server.
RepServerPortNumber	Required	The port number the Repository Server uses to communicate with repository client applications.
RepServerTimeout	Required	The maximum number of seconds the Informatica Server tries to establish a connection to the Repository Server.
LogFileNames	Optional	The filename of the Informatica Server log file.

Option	Required/Optional	Description
DataMovementMode	Required	<p>Possible values are ASCII and UNICODE. The default data movement mode is ASCII, which passes 7-bit ASCII character data. To pass 8-bit ASCII and multibyte character data from sources to targets, use the Unicode mode.</p> <p><b>Note:</b> If you use the UNICODE mode, you should set the ValidateDataCodePages option as No and the SessionLogInUTF8 option as Yes.</p>
ValidateDataCodePages	Required	<p>If set to Yes, the Informatica Server enforces data code page compatibility. If set to No, the Informatica Server lifts restrictions for source and target data code page selection, stored procedure and lookup database code page selection, and session sort order selection.</p> <p><b>Note:</b> If you set the DataMovementMode option to UNICODE, you should set the ValidateDataCodePages option as No.</p>
SessionLogInUTF8	Required	<p>If set to Yes, the Informatica Server writes to the session log using the UTF-8 character set. If set to No, the Informatica Server writes to the session log using the Informatica Server code page.</p> <p><b>Note:</b> If you set the DataMovementMode option to UNICODE, you should set the SessionLogInUTF8 option as Yes.</p>
MaxSessions	Required	<p>The maximum number of sessions stored in shared memory. This sets the maximum number of sessions the Informatica Server can run in parallel.</p>
LMSharedMem	Required	<p>The amount of shared memory available for use by the Informatica Server Load Manager process. For every 10 sessions in MaxSessions, you need at least 2,000,000 bytes reserved in LMSharedMem. The default is 2,000,000 bytes.</p>
TimeStampWorkflowLogMessages	Optional	<p>Set to Yes if you want to append a time stamp to messages written to the workflow log.</p>
FailSessionIfMaxSessionsReached	Optional	<p>Set to Yes to have the Informatica Server fail the session if the number of sessions already running is equal to the value configured for MaxSessions. If you set this option to No, the Informatica Server places the session in a ready queue until a session slot becomes available.</p>
ExportSessionLogLibName	Optional	<p>If you want the Informatica Server to write session log messages to an external library, enter the name of the library file.</p>

Option	Required/Optional	Description
ErrorSeverityLevel	Optional	<p>The level of error messages written to the Informatica Server log.</p> <p>Specify one of the following message levels:</p> <ul style="list-style-type: none"> <li>■ Error. Writes ERROR code messages to the log.</li> <li>■ Warning. Writes WARNING and ERROR code messages to the log.</li> <li>■ Information. Writes INFO, WARNING, and ERROR code messages to the log.</li> <li>■ Tracing. Writes TRACE, INFO, WARNING, and ERROR code messages to the log.</li> <li>■ Debug. Writes DEBUG, TRACE, INFO, WARNING, and ERROR code messages to the log.</li> </ul>
DateHandling40Compatibility	Optional	<p>Set to Yes if you want the Informatica Server to handle dates as in PowerCenter 1.0/PowerMart 4.0. Set to No if you want the Informatica Server to handle dates as defined in the current version of PowerCenter.</p> <p>Date handling significantly improved in PowerCenter 1.5 and PowerMart 4.5. If you need to revert to PowerCenter 1.0 or PowerMart 4.0 behavior, you can configure the Informatica Server to handle dates as in PowerCenter 1.0 and PowerMart 4.0.</p>
PMServer3XCompatibility	Optional	<p>Set to Yes if you want the Informatica Server to handle dates as in PowerCenter 1.0/PowerMart 4.0. Set to No if you want the Informatica Server to handle dates as defined in the current version of PowerCenter.</p> <p>Date handling significantly improved in PowerCenter 1.5 and PowerMart 4.5. If you need to revert to PowerCenter 1.0 or PowerMart 4.0 behavior, you can configure the Informatica Server to handle dates as in PowerCenter 1.0 and PowerMart 4.0.</p>
AggregateTreatNullAsZero	Optional	<p>Set to 1 to have the Informatica Server treat nulls as zero in Aggregator transformations. Set to 0 to have the Informatica Server treat nulls as nulls in aggregate calculations.</p>
AggregateTreatRowAsInsert	Optional	<p>Set to 1 to have the Informatica Server perform aggregate calculations before flagging records for insert, update, delete, or reject in Update Strategy expressions. Set to 0 to have the Informatica Server perform aggregate calculations based on the Update Strategy transformation.</p>
DateDisplayFormat	Required	<p>If specified, the Informatica Server validates the date display format and uses it in session log and server log entries. If the date display format is invalid, the Informatica Server uses the default date display format. The default date display format is DY MON DD HH 24:MI:SS YYYY.</p>

Option	Required/Optional	Description
JoinerSourceOrder6x Compatibility	Optional	If set to Yes, the Informatica Server processes master and detail pipelines sequentially as it did in versions prior to 7.0. The Informatica Server processes all data from the master pipeline before starting to process the detail pipeline. Also, if you enable this option, you cannot specify the Transaction level transformation scope for Joiner transformations. If set to No, the Informatica Server processes the master and detail pipelines concurrently.
TreatNullIn ComparisonOperator As	Required	Determines how the Informatica Server evaluates null values in comparison operations. Specify one of the following options: <ul style="list-style-type: none"> <li>Null. The Informatica Server evaluates null values as null in comparison expressions. If either operand is null, the result is null. This is the default behavior.</li> <li>High. The Informatica Server evaluates null values as greater than non-null values in comparison expressions. If both operands are null, the Informatica Server evaluates them as equal. When you choose High, comparison expressions never result in null.</li> <li>Low. The Informatica Server evaluates null values as less than non-null values in comparison expressions. If both operands are null, the Informatica Server treats them as equal. When you choose Low, comparison expressions never result in null.</li> </ul>
WriterWaitTimeOut	Optional	In target-based commit mode, the amount of time in seconds the writer remains idle before it issues a commit when the following conditions are true: <ul style="list-style-type: none"> <li>The Informatica Server has written data to the target.</li> <li>The Informatica Server has not issued a committed.</li> </ul> The Informatica Server may commit to the target before or after the configured commit interval. Default is 60 seconds. If you configure the timeout to be 0 or a negative number, the Informatica Server defaults to 60 seconds.
LoadManagerAllow Debugging	Optional	If set to Yes, you can run the Debugger.
CreateIndicatorFiles	Optional	If set to Yes, the Informatica Server creates indicator files when you run a workflow with a flat file target.
XMLWarnDupRows	Optional	If set to Yes, the Informatica Server writes duplicate row warnings and duplicate rows for XML targets to the session log.
OutputMetaDataFor FF	Optional	If you specify this option, the Informatica Server writes column headers to flat file targets. It writes the target definition port names to the flat file target in the first line, starting with the # symbol. By default, this option is disabled.

Option	Required/Optional	Description
MaxLookupSPDB Connections	Optional	<p>Allows you to specify a maximum number of connections to a lookup or stored procedure database when you start a session. If the number of connections needed exceeds this value, session threads must share connections. This can result in a performance loss. If you do not specify a value, the Informatica Server allows an unlimited number of connections to the lookup or stored procedure database.</p> <p>If the Informatica Server allows an unlimited number of connections, but the database user does not have permission for the number of connections required by the session, the session fails.</p>
MaxMSSQL Connections	Optional	<p>Allows you to specify a maximum number of connections to a MSSQL Server database when you start a session. If the number of connections required by the session is greater than this value, the session fails.</p>
NumOfDeadlock Retries	Optional	<p>Allows you to specify the number of times the Informatica Server retries a target write on a database deadlock.</p>
DeadlockSleep	Optional	<p>Allows you to specify the number of seconds before the Informatica Server retries a target write on database deadlock. If set to 0 seconds, the Informatica Server retries the target write immediately.</p>
LicenseFile	Required	<p>The name of the file that contains the license keys. The default file name is pm.lic. It is recommended that you use this file name.</p> <p>During the installation, the product license key is automatically added to this file. After the installation, you must add any additional Option and Connectivity license keys to this file. For instructions, see <a href="#">Section 5.6, "How to Update the Informatica License Files on UNIX"</a>.</p>
JVMDllPath	Optional	<p>The absolute path to the JVM library file.</p>
JVMClassPath	Optional	<p>You can set the CLASSPATH to any JAR files you need to run a session using a web service source, target, or transformation. The Informatica Server appends the values you set to the system CLASSPATH.</p>
JVMMinMemory	Optional	<p>Set this value to increase the minimum amount of memory for JVM to use during an Informatica Server session. The default value is 32 MB.</p> <p>If the session fails due to a lack of memory, you may want to increase this value.</p>
JVMMaxMemory	Optional	<p>Set this value to increase the maximum amount of memory for JVM to use during an Informatica Server session. The default value is 64 MB.</p> <p>If the session fails due to a lack of memory, you may want to increase this value.</p>
TrustStore	Optional	<p>Enter the value for TrustStore using the following syntax:</p> <pre>&lt;path&gt;/&lt;filename&gt;</pre>

Option	Required/Optional	Description
ClientStore	Optional	Enter the value for ClientStore using the following syntax: <code>&lt;path&gt;/&lt;filename&gt;</code>
JvmOptionName[n]	Optional	You can enter names and values for up to 10 custom VM options. Use the following syntax: <code>JvmOption[n]=-D&lt;VMOption_ prefix&gt;=&lt;VMOption_value&gt;</code>  n can range from 1 to 10 and must be unique for each option.
JCEProvider	Optional	Enter the JCEProvider class name to support NTLM authentication.
HttpProxyServer	Required	Name of the HTTP proxy server.
HttpProxyPort	Required	Port number of the HTTP proxy server. This must be a number.
HttpProxyUser	Optional	Authenticated username for the HTTP proxy server.
HttpProxyPassword	Optional	Password for the authenticated user.
HttpProxyDomain	Optional	Domain for authentication.

4. When the installation is complete, proceed to [Section 5.6, "How to Update the Informatica License Files on UNIX"](#).

## 5.6 How to Update the Informatica License Files on UNIX

When you install the Informatica PowerCenter, you enter a product license key. The installation process adds the product license key to a license file called pm.lic. Copies of this license file are stored in the installation directories for the Informatica Server and the Repository Server. You must add to the license file any additional Option and Connectivity license keys you have. You need a Team Based Development license (for more information, see [Section 4.14.1.1, "Summary of Versioning in the Informatica Repository"](#)).

### To update the Informatica license files

1. Navigate to the Informatica Server installation directory.
2. Update the license file with a license key using the following syntax:

```
pmlic update -key <keystring_to_add>
```

For example, if the license key is AAAA-AAAA-AAAA-AAAA-AAAA, enter the following:

```
pmlic update -key AAAA-AAAA-AAAA-AAAA-AAAA
```

3. Repeat Step 2 for every license key you need to add.
4. Navigate to the Repository Server installation directory, and repeat Step 2 for every license key you need to add.

---

---

**Note:** To execute pmlic, you must set the following environment variables: LD\_LIBRARY\_PATH = <InformaticaInstallroot>/server  
PATH=<InformaticaInstallroot>/server

---

---

## 5.7 How to Install the Informatica Patch on UNIX

After you install Informatica PowerCenter version 7.1.4, you need to install an Informatica patch.

### To install the Informatica patch

1. Access the patch files on the Oracle BI Applications CD-ROM.
2. Locate the directory named Patch1, and install the patch by following the instructions in the readme file.

## 5.8 How to Create Database Connections for Teradata Installations

If you are using a Teradata database, follow this procedure to create database connections.

### To create database connections for Teradata installations

1. Using the Teradata ODBC driver, create an ODBC connection for the Teradata database.
2. Set the following variables in the ODBC.INI file:

```
DateFormat=AAA  
SessionMode=ANSI  
NoScan=Yes
```

For UNICODE environments, add the following:

```
CharacterSet=UTF8
```

## 5.9 How to Set Environment Variables for the Informatica Server on UNIX

This section includes the following procedures:

- [Section 5.9.1, "How to Add setenv Variables to .cshrc"](#)
- [Section 5.9.2, "How to Set the Code Page for Non\\_ENU Operating Systems on UNIX"](#)
- [Section 5.9.3, "How to Set Siebel UnicodeDB on the Informatica Server if source and target databases are Unicode on UNIX"](#)
- [Section 5.9.4, "How to Disable the bulk loader for DB2/390 databases"](#)

### 5.9.1 How to Add setenv Variables to .cshrc

You should add setenv variables to .cshrc. Otherwise, you have to set the environment variables every time you log into the UNIX machine before configuring, starting, or stopping the Informatica servers.

To add setenv variables to .cshrc:

1. For Solaris, in the root directory, add these command lines to .cshrc:

```
setenv PATH /<INFA_home>/informatica/pm.:${PATH}
```

```
setenv LD_LIBRARY_PATH /<INFA_home>/informatica/pm:${LD_LIBRARY_PATH}
setenv PATH /<INFA_home>/informatica/repositoryserver:.${PATH}
setenv LD_LIBRARY_PATH /<INFA_home>/informatica/repositoryserver:${LD_LIBRARY_PATH}
```

2. For AIX, in the root directory, add these command lines to `.cshrc`:

```
setenv PATH ${PATH}:/<INFA_home>/informatica/pm:.
setenv LIBPATH ${LIBPATH}:/<INFA_home>/informatica/pm
setenv PATH ${PATH}:/<INFA_home>/informatica/repositoryserver:.
setenv LIBPATH ${LIBPATH}:/<INFA_home>/informatica/repositoryserver
```

3. For HP-UX, in the root directory, add these command lines to `.cshrc`:

```
setenv PATH /<INFA_home>/informatica/pm:.${PATH}
setenv SHLIB_PATH /<INFA_home>/informatica/pm:${SHLIB_PATH}
setenv PATH /<INFA_home>/informatica/repositoryserver:.${PATH}
setenv SHLIB_PATH /<INFA_home>/informatica/repositoryserver:${SHLIB_PATH}
```

## 5.9.2 How to Set the Code Page for Non\_ENU Operating Systems on UNIX

This section explains how to set the code page for non\_ENU operating systems on UNIX.

To set the code page for non\_ENU operating systems on UNIX:

1. If you are using a non-ENU operating system, you must set the environment variable `PMREPCODEPAGE` appropriately. For example: `PMREPCODEPAGE=MS932`.

## 5.9.3 How to Set Siebel UnicodeDB on the Informatica Server if source and target databases are Unicode on UNIX

This section explains how to set Siebel UnicodeDB on the Informatica Server if source and target databases are Unicode on UNIX.

To set Siebel UnicodeDB on the Informatica Server if source and target databases are Unicode on UNIX

1. For all UNIX platforms, if both the transactional and data warehouse data sources are Unicode, add the following line in `/<INFA_home>/informatica/pm/pmsserver.cfg` file.

```
SiebelUnicodeDB=<OLTP_TBO>@<OLTP_connectionString> <OLAP_TBO>@<OLAP_ConnectionString>
```

For example:

```
SiebelUnicodeDB=ora16221@sdcd1580i016_qa16b ora16222@sdcd1580i016_qa16b
```

## 5.9.4 How to Disable the bulk loader for DB2/390 databases

This section explains how to disable the bulk loader for DB2/390 databases.

To disable the bulk loader for DB2/390 databases:

1. For DB2/390 databases, add the following line in `/<INFA_home>/informatica/pm/pmsserver.cfg` file.

```
DisableDB2BulkMode = Yes
```

## 5.10 How to Set Up the HOSTS File for Teradata Installations on UNIX

If you are using a Teradata database, you need to set the TDPID parameter on the machine where the Informatica Server is installed by adding an entry in the HOSTS file.

### To set up the HOSTS file for Teradata installations

1. On the machine where the Informatica Server is installed, go to the /etc directory and open the HOSTS file.
2. In the HOSTS file, enter a line in the following format:

```
<IP address of remote server> <remote server> <remote server>COP<n>
```

For example:

```
172.20.176.208 tdatssvr tdatssvrCOP1
```

where tdatssvrCOP1 is the alias for the remote server. The alias must begin with an alphabetic string and end with the COP n suffix, where n is a number between 1 and the total number of applications processors that are associated with the Teradata communications processor.

3. Save the HOSTS file.

For more information about setting the TDPID parameter, see the Teradata documentation.

## 5.11 How to Start the Informatica Repository Server on UNIX

Follow this procedure to start the Informatica Repository Server on UNIX.

### To start the Informatica Repository Server

- Go to /<INFA\_home>/informatica/repositoryserver and enter ./pmrepserver.

Check the log file to verify the server started. If it started correctly, the log file will state: "Server Initialization Completed." If multiple instances are running, you need to specify the name of the configuration file (for example, enter ./pmsserver pmsserverdev.cfg or ./pmsserver mserverQA.cfg).

## 5.12 How to Stop the Informatica Repository Server on UNIX

Follow this procedure to stop the Informatica Repository Server on UNIX.

### To stop the Informatica Repository Server

- Go to /<INFA\_home>/informatica/repositoryserver, and enter the following command:

```
pmrepserver -s -h <hostname>:<port number>
```

Check the log file to verify the server stopped.

## 5.13 About the DAC Server on UNIX

Although you can run the DAC server on a UNIX machine, Oracle does not provide a separate Oracle BI Applications installer for UNIX. Therefore, to install the DAC server in a UNIX environment, you must first install Oracle BI Applications onto a Windows machine, then manually copy over the \OracleBI\DAC\ directory to a

UNIX machine, and then follow the steps below (see [Section 5.13.1, "How to configure the DAC server on UNIX"](#)).

Shell scripts are provided in the \*.sh and \*.csh formats. You must use the appropriate scripts in the appropriate shells, because their syntaxes are different. [Table 5–2](#) lists the available shell scripts and their usage. These files contain comments that provide information about how to configure the scripts.

**Table 5–2 Oracle Business Analytics Warehouse Scripts for Bash and C Shells**

Script	Purpose
config.sh config.csh	Configures the environment variables for DAC_HOME and JAVA_HOME.
dacCmdLine.sh dacCmdLine.csh	Invokes DAC commands on the DAC server. See the file for usage information.
dac_env.sh dac_env.csh	Configures environment variables, including Informatica's pmcmd utility, which enables the DAC server to invoke workflows. Other variables included are NLS_LANG and ORA_HOME.
serverSetupPrompt.sh serverSetupPrompt.csh	Configures DAC metadata repository connection information.
serverinfo.sh serverinfo.csh	Displays the DAC server version.
startserver.sh startserver.csh	Starts the DAC server on machines running Solaris and HP.
startserver_aix.sh startserver_aix.csh	Starts the DAC server on AIX machines.
stopserver.sh stopserver.csh	Shuts down the DAC server.

---



---

**Note:** All the files listed in [Table 5–2](#) need to have read, write, and execute permissions. If the files do not have these permissions, modify them using the chmod command.

---



---

### 5.13.1 How to configure the DAC server on UNIX

This section explains how to configure the DAC server on UNIX

1. Copy the \OracleBI\DAC\ directory from the Windows machine where you installed Oracle BI Applications to a UNIX machine.

---



---

**Note:** You can use FTP to copy the shell scripts under DAC using a dos2unix utility in an ascii mode. The other files can be copied in a binary mode.

---



---

2. Using a MS-DOS to UNIX conversion tool, convert the script files shown in [Table 5–2](#) to UNIX format (that is, remove the carriage return and line feed characters).

There are many MS-DOS to UNIX conversion tools that are freely available for download on the Internet. Alternatively, you can manually remove the carriage return and line feed characters from the script files.

3. Edit the `dac_env.sh` or `dac_env.csh` file to add the path of the directories for the Informatica Server and Informatica Repository Server to the appropriate environment variable:

- For Solaris: `LD_LIBRARY_PATH`
- For AIX: `LIBPATH`
- For HP: `SHLIB_PATH`

For example:

```
setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:/<INFA_HOME>/informatica<version>
```

4. If you are using an Oracle database and the OCI driver, configure the following variables and uncomment in the `dac_env.sh` or `dac_env.csh` file them by removing the # sign at the beginning of the line:

- `TNS_ADMIN`: To set the directory for Oracle to use `tnsnames.ora` files.
- `LD_LIBRARY_PATH` (Solaris), `LIBPATH` (AIX), or `SHLIB_PATH` (HP): To set the Oracle library path.

---

---

**Note:** Informatica does not support 64-bit libraries.

---

---

5. Edit the `config.sh` or `config.csh` file to set the `JAVA_HOME` directory.

---

---

**Caution:** Do not edit any other parameters in this file.

---

---

6. Copy the database-specific `.jar` files to the `$DAC/lib` directory:

- **Oracle.** If you are using an Oracle database (other than 8.x), find the directory where Oracle is installed. Copy the file named `ojdbc14.jar` in the `jdbc/lib` directory and paste it in the `$DAC/lib` directory.
  - If you are using Oracle 8.x, copy the file named `classes12.zip` and paste it in the `$DAC/lib` directory. Also, edit the `ORACLELIB` setting in the `config.sh` or `config.csh` file as follows:

```
setenv ORACLELIB=$DAC_Home/lib/classes12.zip
```

- **DB2.** If you are using a DB2 database, find the directory where DB2 is installed. In the Java subdirectory copy the file named `db2java.zip` and paste it in the `$DAC/lib` directory.

---

---

**Note:** If your source or target database is DB2-UDB, you also need to create stored procedures to be used during the ETL process. For instructions, see [Section 4.21, "How to Create Stored Procedures for DB2-UDB"](#).

---

---

- **MSSQL.** If you are using an MSSQL database, download the SQL Server JDBC drivers from Microsoft's Web site. Copy the files `msbase.jar`, `mssqlserver.jar`, and `msutil.jar` to the `$DAC/lib` directory.

7. Run `serverSetupPrompt.sh` or `serverSetupPrompt.csh` to set the repository connection information and the email settings:
  - a. At the initial prompt, enter 1 to set the repository connection information.
  - b. Enter 3 to test the repository connection.
  - c. Enter 4 to save the information you entered.
  - d. Enter 2 to set up the administrator email account.

---

**Note:** For the email functionality to work, you must be using an SMTP server in which the SMTP authentication mode LOGIN is enabled. For example, if you are using Microsoft Exchange Server, you must enable the SMTP module and enable authentication in the Basic Mode. The SMTP server outbound email authentication must be turned on for the DAC server to be able to send email notifications.

---

- e. Enter 5 to exit.
8. Copy content-specific XML files from the `OracleBI\DAC\CustomSQLs` directory in the Windows installation to the `OracleBI/DAC/CustomSQLs` directory in the UNIX installation.
9. Start the DAC server using the appropriate script (for more information, see [Table 5-2](#)).

For example, use the `startserver.sh` script on a Solaris or HP machine.

## 5.14 How to Stop and Start the Informatica Server on UNIX

This section explains how to stop and start the Informatica Server on UNIX.

### 5.14.1 Requirements

- From the UNIX machine, make sure you can connect to the transactional database and to the data warehouse database.
- The Informatica repository must have been configured using the Informatica Workflow Manager.
  - For instructions, see [Section 4.14.4, "How to Configure the Informatica Repository in Workflow Manager"](#).
- The environment variables for the Informatica Server on UNIX must be set.
- The appropriate parameter files should have been copied to the `SrcFiles` directory.
- For Unicode, the `SiebelUnicodeDB` variable must be set.

### 5.14.2 How to Start the Informatica Server on UNIX

This section explains how to start the Informatica Server.

To start the Informatica Server:

1. Go to the Informatica Server directory, and enter `./pmserver`.

For Teradata installations, enter `./pmserver ./pmserver.cfg>./pmserver.out`

Check the log file to verify the server started. If it started correctly, the log file will state: "Server Initialization Completed."

### 5.14.3 How to Start the Informatica Server On UNIX

This section explains how to stop the Informatica Server.

To stop the Informatica Server:

1. Go to the Informatica Server directory, and enter the following command:

```
./pmcmd stopserver <repository username> <repository password> <port #>.
```

For example:

```
./pmcmd stopserver Administrator Administrator 4001
```

You can check whether the Informatica Server is running by entering the command `ps -elf |grep pmserver`.

If it lists `pmserver` process, the server is running. If it lists `grep pmserver`, that is the process of the `ps` command.

## 5.15 How to Run the DAC Server on UNIX

Follow this procedure to start the DAC server on UNIX. Before you can start the DAC server, you must have already configured the `config.csh` or `config.sh` and `dac_env.csh` or `dac_env.sh` files.

### 5.15.1 How to Start the DAC server on UNIX

This section explains how to start the DAC server.

To start the DAC server:

1. In `csh`-related shells, issue the following command:

```
./startserver.csh
```

2. In `bash`-related shells, issue the following command:

```
./startserver.sh
```

### 5.15.2 How to Run the DAC server in the background

This section explains how to start the DAC server in the background.

To run the DAC server in the background:

1. In `csh`-related shells, issue the following command:

```
nohup startserver.csh 2>&1 &
```

2. In `bash`-related shells, issue the following command:

```
nohup startserver.sh 2>&1 &
```

The `nohup` command allows the DAC server to run in the background on UNIX even if the DAC client is disconnected.

## 5.16 How to Stop the DAC Server on UNIX

Follow this procedure to stop the DAC server on UNIX.

To stop the DAC server:

1. In `csh` related shells, issue the following command:

```
./stopserver.csh
```

2. In bash-related shells, issue the following command:

```
./stopserver.sh
```

---

**Note:** When you execute `./stopserver.csh` or `./stopserver.sh`, the server will issue a warning about the shutdown request. When the server receives the request, it will shut down even if there is an ETL in progress. The statuses for the ETL run will not be set correctly. The next time the DAC server starts, it will set the status to Failed for any uncompleted run.

---

## 5.17 Teradata-Specific Installation and Configuration Checklist for UNIX

Table 5–3 provides a list of Teradata-specific installation and configuration steps that are performed during the Oracle BI Applications installation and configuration process for UNIX. You should review this list to make sure that you have performed all of the required Teradata-specific steps. You should also review [Section 4.22, "Teradata-Specific Installation and Configuration Checklist \(Windows\)"](#) for Teradata-specific information related to the Windows environment.

**Table 5–3 Teradata-Specific Installation Checklist for UNIX**

Action	Link to Topic
Review Teradata-specific database guidelines	<a href="#">Section 3.9, "Teradata-Specific Database Guidelines for Oracle Business Analytics Warehouse"</a> .
Create database connections	<a href="#">Section 5.8, "How to Create Database Connections for Teradata Installations"</a> .
Set up the HOSTS file	<a href="#">Section 5.10, "How to Set Up the HOSTS File for Teradata Installations on UNIX"</a> .



# Part III

---

## Customization and Security in Oracle BI Applications

---

Part III explains how to customize the out-of-the-box functionality in Oracle BI Applications, and contains the following chapters:

- [Chapter 6, "Customizing the Oracle Business Analytics Warehouse"](#)
- [Chapter 7, "Integrated Security for Oracle BI Applications"](#)

---

**Note:** For a high level road map for installation, configuration, and customization steps for Oracle BI Applications, see [Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle BI Applications"](#).

---



---

---

# Customizing the Oracle Business Analytics Warehouse

This chapter describes concepts and techniques for customizing the Oracle Business Analytics Warehouse.

---

---

**Note:** Before implementing customizations, you should already be familiar with Informatica PowerCenter.

---

---

This chapter includes the following main topics:

- [Section 6.1, "Oracle Business Analytics Warehouse Customization Scenarios"](#)
- [Section 6.2, "Category 1 Customizations: Adding Columns to Existing Fact or Dimension Tables"](#)
- [Section 6.3, "Other Types of Customizations Requiring Special Handling"](#)
- [Section 6.4, "Category 2 Customizations: Adding Additional Tables"](#)
- [Section 6.5, "Category 3 Customizations: Adding New Data as a Whole Row into a Standard Dimension Table"](#)
- [Section 6.6, "Configuring Extracts"](#)
- [Section 6.7, "Configuring Loads"](#)
- [Section 6.8, "Configuring Slowly Changing Dimensions"](#)
- [Section 6.9, "About Stored Lookups"](#)
- [Section 6.10, "Codes Lookup"](#)
- [Section 6.11, "About Resolving Dimension Keys"](#)
- [Section 6.12, "About Domain Values"](#)
- [Section 6.13, "Configuring the Domain Value Set with CSV Worksheet Files"](#)
- [Section 6.14, "Configuring the Domain Value Set Using PowerCenter Designer"](#)
- [Section 6.15, "Configuring Conformed Dimensions"](#)

## 6.1 Oracle Business Analytics Warehouse Customization Scenarios

In customizing the Oracle Business Analytics Warehouse, various scenarios are available based on the type of your data source:

- Packaged applications (for example, Siebel CRM or Oracle EBS) use prepackaged adapters.
- Non-packaged data sources use the Universal adapter.

The figure below shows the categories of supported customization scenarios, based on the data source.

**Figure 6–1 Supported customizations based on data source**

		Data Warehouse Modifications		
		Add Additional Column to Existing Fact or Dimension Table	Add Additional Rows to Existing Fact or Dimension Tables	Add New Fact or Dimension Tables
Data Sources	Packaged Application (Uses Prepackaged Adaptor)	Category 1	Configure Filter	Category 2
	Non-Packaged Data (Uses Universal Adaptor)	Category 1	Category 3	Category 2

For detailed information about tables and naming conventions, see *Oracle Business Analytics Warehouse Data Model Reference*.

### 6.1.1 Types of Customizations

Figure 6–1, "Supported customizations based on data source" shows the following categories of customization:

- **Category 1.** In a Category 1 customization, you add additional columns from source systems that have pre-packaged adapters and load the data into existing data warehouse tables.
- **Category 2.** In a Category 2 customization, you use prepackaged adapters to add new fact or dimension tables to the data warehouse. Category 2 customizations normally require that you build new SDE and SIL mappings.
- **Category 3.** In a Category 3 customization, you use the Universal adapter to load data from sources that do not have pre-packaged adapters.

### 6.1.2 Upgrade Consideration

One of the most difficult aspects about working with customizations is handling the customizations at the time of an upgrade. Informatica does not provide a 'diff-merge' capability that would automatically detect changes introduced by customers and add them into upgraded mappings. Therefore, customizations must be reapplied manually to upgraded mappings. Oracle BI Applications attempts to minimize the amount of effort required to reapply customizations after an upgrade. As long as the customization methodology is followed, the effort at upgrade time should be minimal and in many cases there may be no manual effort required at all.

## 6.2 Category 1 Customizations: Adding Columns to Existing Fact or Dimension Tables

This section includes the following topics:

- [Section 6.2.1, "About Extending Mappings"](#)
- [Section 6.2.2, "Impact of Customization on Upgrade"](#)
- [Section 6.2.3, "Typical Steps to Extend Mappings in the Oracle Business Analytics Warehouse"](#)
- [Section 6.2.4, "Scenario for Extending the Oracle Business Analytics Warehouse"](#)

## 6.2.1 About Extending Mappings

Category 1 customizations involve extracting additional columns from source systems for which pre-packaged adapters are included (for example, Siebel or Oracle) and loading the data into existing data warehouse tables. For Category 1 customizations, data can also come from non-packaged sources, but this section assumes that the sources have already been mapped with a Universal adapter and only need to be extended to capture additional columns. (The initial mapping of a Universal adapter is considered a Category 3 customization. For information, see [Section 6.5, "Category 3 Customizations: Adding New Data as a Whole Row into a Standard Dimension Table"](#).)

In order to see additional columns in the data warehouse, the columns must first be passed through the ETL process. The existing mappings and tables are extensible. Sample placeholders demonstrate how to pass and store additional data. Oracle BI Applications provides a methodology to extend preconfigured mappings to include these additional columns and load the data into existing tables.

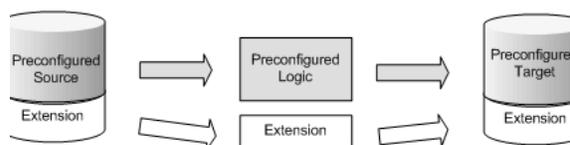
Oracle BI Applications recognizes two types of customization: extension and modification. The supported extension logic allows you to add to existing objects. For example, you can extract additional columns from a source, pass them through existing mappings, and populate new columns added to an existing table. Generally, Oracle BI Applications does not allow you to modify existing logic or columns. You should not change existing calculations to use different columns, and you should not remap existing columns to be loaded from different sources.

For example, if you want to calculate revenue differently from the existing logic, you should create a new transformation and connect that calculation to a new column, for example, X\_REVENUE. You can then remap the Oracle Business Intelligence repository to point to the new X\_REVENUE column.

Most mappings have a single placeholder column, named X\_CUSTOM, that marks a *safe path* through the mapping. All extension logic should follow the same route through the mapping as X\_CUSTOM. You can add additional transformations to the mapping, but they should follow the same route through the mapping as X\_CUSTOM.

In the following figure, the preconfigured logic is shaded in gray. You should not modify anything contained within these objects. You should add customizations to the existing mapping, which allows them to run parallel to the existing logic.

**Figure 6–2 Preconfigured logic and customizations**



Because some objects need to be modified in order to allow for extensions, Oracle BI Applications breaks down extensions into the following categories:

- **Exposed objects.** These objects allow changes, but the changes must be in the form of extensions (additive), and must never modify existing preconfigured logic. These objects are included in the mapping when shipped and are usually sources, targets, and nonreusable transformations.
- **Encapsulated objects.** These objects cannot be extended. They attempt to hide as much of the shipped transformation logic as possible to prevent breaking the preconfigured logic. These objects are included in the mapping when shipped and are usually mapplets and reusable transformations.
- **Custom objects.** You add custom objects to a mapping. (They are not shipped by Oracle.) Custom objects can be sources, transformations (reusable and nonreusable), or mapplets. Reusable transformations and mapplets that are shipped are considered encapsulated objects, but when you add such objects to an existing mapping, they are considered custom objects to that particular mapping. For example, if you want to add another amount to a fact table and that amount needs to be converted from its original currency to the data warehouse currency, you would normally add the existing Currency Exchange mapplet to the mapping to convert this new amount. In this case, the mapplet is considered a custom object to this particular mapping; however, it is also encapsulated, so the internal logic must not be changed.

---

**Note:** Targets should not be added to a mapping.

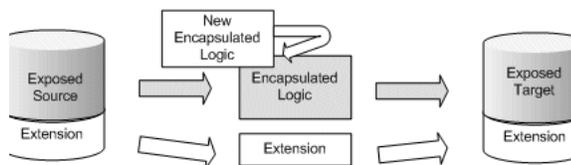
---

## 6.2.2 Impact of Customization on Upgrade

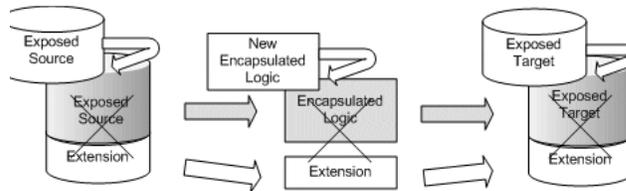
When upgrading, you will deploy customized mappings on an individual basis. Only the actual mappings that have changed will be applied in your existing environment. This means any mappings that have not changed will not be affected, so any customizations made to these mappings remain. Only the mappings that have actually changed will require some work to reapply customizations. If you follow the recommended approach, the amount of work required to reapply customizations should be minimal.

By encapsulating the logic as much as possible, any changes made to the preconfigured logic can be switched as either part of a patch release or upgrade without impacting any extension logic, as shown in the following figure.

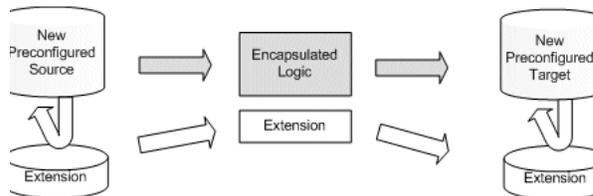
**Figure 6–3 Encapsulating logic**



If there is a change to an exposed object, the new logic will always take precedence over the extension logic. However, rather than losing all of the extensions, much of the extension logic is retained and only has to be reapplied to the exposed objects. For example, if you add an additional column from the source and load it into the target, during an upgrade, the upgraded mapping brings additional columns from the source and loads them into the target.

**Figure 6–4 Encapsulated logic and extension logic**

The source and target are completely replaced so any extensions to these are lost in Informatica (note that the columns will still exist in the database). However, the extension logic itself still exists after the upgrade. The source and target must be re-extended and then reconnected to the extension logic.

**Figure 6–5 Re-extending and reconnecting to extension logic**

If you extend a mapping and the mapping...

- Does not change during the upgrade, all extensions are retained.
- Experiences changes to the encapsulated logic, all extensions are retained.
- Experiences changes to the exposed objects, extensions to these objects are lost but the underlying extension logic is retained. Extensions to exposed objects must be manually reapplied.

### 6.2.2.1 Points to Remember

- Encapsulated objects must never be customized unless directed by Oracle. Encapsulated objects are usually mapplets and reusable transformations.
- Exposed objects can be extended but must never be otherwise modified. Exposed objects may be completely replaced at upgrade.
- Custom objects are never changed during an upgrade.
- To minimize the work required for upgrading, try to minimize the number of changes to exposed objects by using custom objects. For example, rather than adding a table to the Source Qualifier to bring in a column from a related table, add a lookup to that table in the mapping.
- In customizing objects, you must evaluate the options and determine the best approach for your environment. If you find the custom object approach allows the ETL to run in an acceptable amount of time, then this is the preferred approach. If the custom object causes the ETL process to take too long, you may want to consider incorporating the extension into an exposed object.
- When you add custom columns to the Oracle Business Analytics Data Warehouse, you do not need to manually add change capture views. The DAC automatically creates change capture views for all columns (including new columns) at runtime.

---

---

**Note:** Most SDE adapter folders use the concept of Business Component mapplets. These are extract mapplets that may contain relational, application, or flat file sources. The Siebel adapter folders do not use Business Component mapplets; the sources are exposed directly in the mapping. Usually, the Business Component mapplet can be treated as an exposed object and is the only mapplet object that should be modified.

---

---

### 6.2.3 Typical Steps to Extend Mappings in the Oracle Business Analytics Warehouse

The most common scenario for extending the data warehouse is to extract existing columns from a source and pass them through to an existing data warehouse table (either fact or dimension). This type of change generally requires extending an SIL mapping. If the data is coming from a packaged source, then you will also need to extend an appropriate SDE adapter mapping. If the data is coming from a non-packaged source, then you must use a Universal adapter mapping. (You will need to create a Universal adapter mapping if an appropriate one does not already exist).

#### To extend a mapping in the Oracle Business Analytics Warehouse

1. Copy the mapping to a custom folder.
2. Extend the source and target tables by making changes to the tables in the database. You then import the source and target definitions into the custom folder (which replaces the existing definitions) or manually edit the existing definition.

As a best practice, Oracle recommends that you rename custom columns with a "X\_" prefix to make it easier to distinguish custom columns that have been added to an existing table and to ensure there are no name conflicts with any columns Oracle may add later to that table.

3. Extend the SDE mapping by bringing in the additional columns.
  - a. Either modify the Source Qualifier (which is an exposed object) to include the column in the SQL Override or add a lookup (which is a custom object).
  - b. Connect objects to the placeholder transformation 'Exp\_Custom'.
  - c. Connect placeholder transformations to the target table.
4. Extend the SIL mapping by bringing in the additional columns.
  - a. Modify the Source Qualifier (which is an exposed object) to include the column in the SQL Override.
  - b. Add the column to the Source Qualifier and pass it through the filter to the 'Exp\_Custom' transformation, to the Update strategy, and to the target.
5. Copy the workflow to the custom folder.
6. Update the DAC with the necessary changes.

### 6.2.4 Scenario for Extending the Oracle Business Analytics Warehouse

In this scenario, data is passed from an existing source to an existing data warehouse table. The company in this example has identified additional fields in a base table that need to be added to the data warehouse table W\_ORG\_D. The company used an extension field to capture information related to organizations referred to as 'ACCOUNT\_LOG.' In addition, the company wants to include the name of the person who last updated the record as an attribute of the organization.

This scenario is applied to examples with the different source types that are supported: Prepackaged adapters for Siebel, non-Siebel prepackaged application sources, and non-packaged data.

This section includes the following topics:

- [Section 6.2.4.1, "Tips for Modifying the SQL Override in a Source Qualifier"](#)
- [Section 6.2.4.2, "Example of Extracting Data from a Siebel Source"](#)
- [Section 6.2.4.3, "Including a Source Table for the Change Capture Process"](#)
- [Section 6.2.4.4, "Example of Extracting Data from a Non-Siebel Packaged Source"](#)
- [Section 6.2.4.5, "Example of Extracting Data from a Universal Source"](#)
- [Section 6.2.4.6, "Example of Loading Data into an Existing Target Table"](#)
- [Section 6.2.4.7, "Updating the DAC"](#)

#### 6.2.4.1 Tips for Modifying the SQL Override in a Source Qualifier

- It is very important that the connected columns in the Source Qualifier appear in the same order in the SQL Override. A common mistake is to have the ports appear in the Source Qualifier in a different order than in the SQL Override.
- The column in the SELECT clause must reference the aliased name of the table if an alias is used. In the example [Section 6.2.4.2, "Example of Extracting Data from a Siebel Source"](#), the actual source for the new column LOGIN is S\_CONTACT but the SELECT clause references the alias LAST\_UPDATE\_BY.
- Make sure you include a comma before new columns in a SELECT clause or before a new table in a FROM clause.
- A new table should always be defined using LEFT OUTER join syntax. Do not use INNER join or RIGHT OUTER join syntax, because you could lose records as a result.
- Make sure you define joins to match on a unique set of values. If you do not define a join that ensures a unique relationship, you may get a cartesian product, which changes the granularity and will result in duplicate errors downstream. If you cannot define a unique join, then you should bring the data in with a Lookup Transformation, which guarantees that at most one record will be returned.
- As a best practice, you should comment custom code you have introduced. Comments should include at least the developer's name and the date the code was added.

#### 6.2.4.2 Example of Extracting Data from a Siebel Source

In this example, the company is using Siebel Business Analytics version 7.8 and has used the S\_ORG\_EXT\_X.ATTRIB\_04 field in the S\_ORG\_EXT extension table to capture data related to 'ACCOUNT\_LOG'. The name of the last person to update the record is retrieved by joining to the S\_CONTACT table on S\_ORG\_EXT.LAST\_UP\_BY.

---

**Note:** The Siebel adapter folders do not use Business Component mapplets; the sources are exposed directly in the mapping.

---

#### To extract data from a Siebel source

1. Create a new folder named CUSTOM\_SDE\_SBL\_78\_adapter.

2. Copy the SDE\_OrganizationDimension mapping and workflow to this folder.
3. Edit the target definition W\_ORG\_DS to include the following columns:

Column Name	Data Type
X_ACCOUNT_LOG	VARCHAR2(10)
X_LAST_LOGIN	VARCHAR2(10)

**Note:** If the source table has been customized, it would be necessary to re-import the source table into the custom folder, replacing the existing version. For this example, the source table has not changed.

4. The extension table S\_ORG\_EXT\_X is already joined in this mapping. Drag the ATTRIB\_04 column from the Source Definition to the Source Qualifier. The column should appear after the X\_CUSTOM column.
5. Since S\_CONTACT is not already joined for the last update person, drag a copy of S\_CONTACT into the mapping. (Delete the new Source Qualifier associated with this source if one exists.)
6. As a best practice, you should rename the source to indicate what it represents. In this case, rename the source to S\_CONTACT\_LAST\_UPDATE\_BY.
7. Drag the LOGIN column from the Source Definition to the Source Qualifier.
8. Drag ATTRIB\_04 and LOGIN to the EXP\_Custom Expression.
9. As a best practice, rename these ports to indicate both the table and column they come from.

In case the mapping is changed and the related exposed objects are replaced, this will make it easier to reconnect because the custom expression will not be replaced.

10. Connect the appropriate ports to the target definition
11. Edit the SQL override in the Source Qualifier.
  - a. Add the ATTRIB\_04 and LOGIN columns immediately after X\_CUSTOM in the SELECT clause.
  - b. Add the table to the FROM clause.
  - c. Add the join criteria.

The Siebel application can run on various database platforms. To maintain database independence, you should write the SQL using Informatica's join syntax, which will automatically be translated to the appropriate database syntax when run. When adding tables, be sure to follow the Informatica syntax when defining the join.

Below is an example of the SQL with the modified code in bold font.

```
S_ADDR_ORG.LAST_UPD,
S_ORG_EXT_T.LAST_UPD,
O AS X_CUSTOM
- Added by J.Smith on 1/10/2007
,S_ORG_EXT_X.ATTRIB_04
,LAST_UPDATE_BY.LOGIN
```

```

FROM
V_ORG_EXT S_ORG_EXT,
S_ORG_EXT BU_NAME,
...

S_ORG_EXT_T,
S_ORG_EXT_X,
S_ADDR_ORG,
...

S_MKT_SEG PRTNR_MKTSEG,
S_MKT_SEG TRGT_MKTSEG
-Added by J.Smith on 1/10/2007
,S_CONTACT LAST_UPDATE_BY

WHERE
{
V_ORG_EXT S_ORG_EXT
LEFT OUTER JOIN S_ORG_EXT_X ON
S_ORG_EXT.ROW_ID = S_ORG_EXT_X.PAR_ROW_ID
...

LEFT OUTER JOIN S_MKT_SEG TRGT_MKTSEG ON
ORG.PR_ORG_TRGT_MKT_ID = TRGT_MKTSEG.ROW_ID

- Added by J.Smith on 1/10/2007
LEFT OUTER JOIN S_CONTACT LAST_UPDATE_BY ON
S_ORG_EXT.LAST_UPD_BY = LAST_UPDATE_BY.ROW_ID
}

```

12. Save your changes.

13. Refresh and validate the session in the Workflow Manager.

This is necessary because it is possible that changes made to the mapping may invalidate the session.

### 6.2.4.3 Including a Source Table for the Change Capture Process

If you are bringing in data from a new table that was not previously included in an existing SDE mapping, it may be necessary to create an auxiliary change capture mapping so that when a row changes in the new table, the corresponding row in the main table is marked as changed. If you do not create an auxiliary process, it is possible that the new column from the new table is changed but if the base table does not experience any changes, this event will not be picked up. Keep in mind that auxiliary processing can have a negative impact on ETL performance. So, if it is not necessary to flag the main record as changed when there is a change in the related table, you may want to avoid building this mapping.

### 6.2.4.4 Example of Extracting Data from a Non-Siebel Packaged Source

In this example, the company is using Oracle Applications version 11.5.8 and has used the HZ\_CUST\_ACCOUNTS.ATTRIBUTE1 field to capture data related to 'ACCOUNT\_LOG.' The name of the last person to update the record is already stored in the HZ\_CUST\_ACCOUNTS.LAST\_UPDATE\_LOGIN field. There is no need to join to any additional tables.

Because Oracle Applications always runs on an Oracle database, you do not need to use Informatica's SQL syntax when defining joins in a SQL Override. If you need to add additional tables, you can define the join using standard Oracle syntax.

If you add another table as a source, in addition to defining the join, you must also include the table's 'LAST\_UPDATE\_DATE' in the WHERE clause using the following syntax:

```
OR TABLE_NAME.LAST_UPDATE_DATE > TO_DATE(' $$LAST_EXTRACT_DATE', 'MM/DD/YYYY
HH24:MI:SS' )
)
AND
...
```

This ensures that changes to a record in that table will trigger an extract. If this were the only table to have an update and the other tables were not updated, then this change would not be detected.

---



---

**Note:** Most SDE adapter folders use the concept of Business Component mapplets. These are extract mapplets that contain relational, application, or flat file sources. Generally, the Business Component mapplet can be treated as an exposed object and is the only mapplet object that should be modified. Keep in mind that you can modify exposed objects, but there is a risk that these changes may be lost at upgrade time.

---



---

**To extract data from a non-Siebel packaged source**

1. Create a new folder named CUSTOM\_SDE\_ORA1158\_adapter.
2. Copy the SDE\_ORA\_OrganizationDimension\_Customer mapping and workflow to this folder.
3. Edit the target definition W\_ORG\_DS to include the following columns:

Column Name	Data Type
X_ACCOUNT_LOG	VARCHAR2(10)
X_LAST_LOGIN	VARCHAR2(10)

---



---

**Note:** If the source table has been customized, it would be necessary to re-import the source table into the custom folder, replacing the existing version. For this example, the source table has not changed.

---



---

4. Open the mapping.
5. Edit the Business Component 'mplt\_BC\_ORA\_OrganizationDimension\_Customer' by right-clicking the mapplet and selecting 'Open Mapplet.'

Remember, the Business Component mapplets are the only mapplets you can normally edit. You should not edit any other mapplets unless directed by Oracle.

6. Drag the columns LAST\_UPDATE\_LOGIN and ATTRIBUTE1 to the Source Qualifier, and then drag these columns to the Mapplet Output.
7. Edit the Source Qualifier to include the new columns, as shown below:

```
SELECT
...
HZ_PARTIES.SIC_CODE
```

```
- Added by J.Smith on 1/10/2007
, HZ_CUST_ACCOUNTS.LAST_UPDATE_LOGIN
, HZ_CUST_ACCOUNTS.ATTRIBUTE1
```

```
FROM
HZ_CUST_ACCOUNTS, HZ_PARTIES
WHERE
...
```

8. Return to the mapping.
9. Add a new expression and rename it to 'X\_CUSTOM.'
10. Connect the new columns from the Business Component mapplet to this expression.
11. As a best practice, you should rename these ports to indicate both the table and column they came from. If the mapping is changed and the related exposed objects are replaced, this will make it easier to reconnect, because the custom expression will not be replaced
12. Connect these columns to the appropriate columns in the target definition.
13. Save your changes.
14. Refresh and validate the session in the Workflow Manager.

This is necessary because it is possible that changes made to the mapping may invalidate the session.

#### 6.2.4.5 Example of Extracting Data from a Universal Source

In this example, the company has a legacy mainframe with data they would like to incorporate into the data warehouse. This requires that data be preformatted to match the source definition for FILE\_ORG. Since the preexisting source definition does not include columns for the additional data the company desires, the source definition must be modified to include these columns.

---



---

**Note:** The Universal adapter folders do not use Business Component mapplets; the sources are exposed directly in the mapping.

---



---

#### To extract data from a Universal source

1. Create a new folder named CUSTOM\_SDE\_Universal\_adapter.
2. Copy the SDE\_Universal\_OrganizationDimension mapping and workflow to this folder.
3. Edit the source definition to include the following columns:

Column Name	Data Type
ACCOUNT_LOG	String(10)
LAST_LOGIN	String(10)

4. Edit the target definition W\_ORG\_DS to include the following columns:

Column Name	Data Type
X_ACCOUNT_LOG	VARCHAR2(10)

Column Name	Data Type
X_LAST_LOGIN	VARCHAR2(10)

5. Open the mapping.
6. Drag the columns LAST\_UPDATE\_LOGIN and ATTRIBUTE1 to the Source Qualifier.
7. Add a new expression and rename it to 'EXP\_CUSTOM.'
8. Connect the new columns from the Source Qualifier to this expression.
9. Connect these columns to the appropriate columns in the target definition.
10. Save your changes.
11. Refresh and validate the session in the Workflow Manager.

This is necessary because it is possible that changes made to the mapping may invalidate the session.

#### 6.2.4.6 Example of Loading Data into an Existing Target Table

Once the required data has been extracted and staged, it must be loaded into an existing target table in the data warehouse.

##### To load data into an existing target table in the data warehouse

1. Create a new folder named CUSTOM\_SILOS.
2. Copy the SIL\_OrganizationDimension mapping and workflow to this folder.
3. Edit the source definition W\_ORG\_DS to include the following columns:

Column Name	Data Type
X_ACCOUNT_LOG	VARCHAR2(10)
X_LAST_LOGIN	VARCHAR2(10)

4. Edit the target definition W\_ORG\_D to include the following columns:

Column Name	Data Type
X_ACCOUNT_LOG	VARCHAR2(10)
X_LAST_LOGIN	VARCHAR2(10)

5. Open the mapping.
6. Drag the columns X\_ACCOUNT\_LOG and X\_LAST\_LOGIN to the Source Qualifier.
7. Drag the columns X\_ACCOUNT\_LOG and X\_LAST\_LOGIN from the Source Qualifier to the Filter.

Normally, existing transformations should not be modified. Filters are active transformations, and it is not possible to route data around an active transformation and bring it back to the same data flow. In this case, the filter is considered an exposed object and may be modified, but any changes are at risk of being lost at upgrade time.

8. Drag the columns X\_ACCOUNT\_LOG and X\_LAST\_LOGIN from the Filter to the Expression EXP\_Custom. If you need to apply any transformations, you should do so in this expression.
9. Drag the columns X\_ACCOUNT\_LOG and X\_LAST\_LOGIN from the Expression to the Update Strategy.

The Update Strategy is another active transformation and is, therefore, considered an to be an exposed object, just like the Filter.

10. Connect these columns to the appropriate columns in the target definition.
11. Save your changes.
12. Refresh and validate the session in the Workflow Manager.

This is necessary because it is possible that changes made to the mapping may invalidate the session.

#### 6.2.4.7 Updating the DAC

After making these changes to the mappings, you need to register the changes in the DAC. You need to include the table definition, with any additional columns or indexes, and the required changes so the tasks execute the modified sessions in the new custom folders. For information about registering data warehouse objects in the DAC, see the Oracle Business Intelligence Data Warehouse Administration Console Guide.

## 6.3 Other Types of Customizations Requiring Special Handling

This section includes the following topics:

- [Section 6.3.1, "Modifying Category 2 SCD Triggers"](#)
- [Section 6.3.2, "Adding Dimensions to Existing Facts"](#)
- [Section 6.3.3, "Adding Date Dimensions to Existing Facts"](#)
- [Section 6.3.4, "Adding Currencies to an Existing Table"](#)

### 6.3.1 Modifying Category 2 SCD Triggers

All dimensions are configured as Category 1 dimensions when first installed. If you want to capture historical changes, the dimension must be enabled to capture Category 2 changes. However, a common form of customization is to change the criteria that triggers a Category 2 change in a dimension. Most changes in a dimension are treated as Category 1 changes in that the existing column is simply overwritten with the new value. Once enabled, there are only a small number of columns that will trigger a Category 2 change. You can extend the logic that triggers a Category 2 change by adding additional columns to the logic that tracks Category 2 changes. In addition, you can remove columns from this logic in case you do not want these types of changes to trigger a Category 2 change. Modifying the Category 2 tracking logic is one of the only exceptions to the rule that you should not make changes to shipped logic. The logic that tracks Category 2 changes is contained in exposed objects in each SIL dimension mapping that supports Category 2 changes.

There is a lookup between the Source Qualifier and the Filter. This lookup is used to determine if the record already exists in the target and, therefore, needs to be updated in addition to other system columns. Columns that track Category 2 changes are returned in this lookup and passed to the next expression. The columns returned by

the lookup are compared with the columns passed from the staging table. If any of these columns are different, the record is flagged for a Category 2 change.

This expression contains a variable port named 'TYPE2\_COLS\_DIFF'. If this port is flagged as 'Y' then a Category 2 change will be triggered. If it is flagged as 'N' then a Category 1 change will be triggered.

To change the columns used to determine a Category 2 change, modify the lookup to pass any additional columns you want to be evaluated for Category 2 changes. Then, modify the variable port 'TYPE2\_COLS\_DIFF' to include this column when being evaluated.

For example, the SIL\_BOMHeaderDimension mapping compares the following columns:

- BOM\_HEADER
- BOM\_VERSION
- BASE\_QTY
- ACTIVE\_FLG

If you wanted to include BOM\_VERSION as part of Category 2 logic, you would change the logic for 'TYPE2\_COLS\_DIFF' from the following:

```
IIF(BOM_NUMBER != LKP_BOM_NUMBER, 'Y',
IIF(BOM_VERSION != LKP_BOM_VERSION, 'Y',
IIF(BASE_QTY != LKP_BASE_QTY, 'Y',
IIF(ACTIVE_FLG != LKP_ACTIVE_FLG, 'Y',
'N'))))
```

To this:

```
IIF(BOM_NUMBER != LKP_BOM_NUMBER, 'Y',
IIF(BOM_VERSION != LKP_BOM_VERSION, 'Y',
IIF(BASE_QTY != LKP_BASE_QTY, 'Y',
IIF(ACTIVE_FLG != LKP_ACTIVE_FLG, 'Y',
IIF(BOM_VERSION != LKP_BOM_VERSION, 'Y',
'N')))))
```

### 6.3.2 Adding Dimensions to Existing Facts

This section covers adding a dimension (preexisting or custom) to an existing fact. It assumes you have already built the required process to populate this dimension.

This process involves extending both the fact staging table and the fact data warehouse table to include the new column. In Informatica, remember to define the tables using the Oracle database type. The staging table should be defined as a varchar2(80) field and named with an \_ID suffix. The data warehouse table column should be defined as an integer and named with a \_WID suffix.

The SDE fact mapping must be modified to pass through the unique identifier of the dimension key. This assumes that there is some relationship between the base table and this unique identifier. It may already be stored in the base table or stored by joining to a related table. Depending on the source system, this identifier may be based on a single column or derived from multiple columns. [Table 6-1](#) depicts various formats used to derive the INTEGRATION\_ID, which is used to identify a dimension key. The INTEGRATION\_ID value should be passed to the fact staging table.

**Table 6–1 Formats to Derive INTEGRATION\_ID**

<b>Dimension</b>	<b>Foreign Key</b>	<b>When Source is Oracle Application</b>	<b>When Source is Siebel Application</b>
W_AP_TERMS_D		TO_CHAR(TERM_ID)	Not applicable
W_BUSN_LOCATION_D	ASSET_LOC_WID	ASSET_LOC~'    LOCATION_ID	Not applicable
W_BUSN_LOCATION_D	EMP_LOC_WID	EMP_LOC~'    LOCATION_ID	Not applicable
W_BUSN_LOCATION_D	INVENTORY_LOC_WID	STORAGE_LOC'    '~'    ORGANIZATION_ID    '~'    SUBINVENTORY_CODE    '~'    INVENTORY_LOCATION_ID	Not applicable
W_BUSN_LOCATION_D	PLANT_LOC_WID	'PLANT'    '~'    TO_CHAR(ORGANIZATION_ID)	Not applicable
W_BUSN_LOCATION_D	RECEIVING_LOC_WID	'RECIPIENT_LOC'    '~'    TO_CHAR(LOCATION_ID)	Not applicable
W_BUSN_LOCATION_D	STORAGE_LOC_WID	'STORAGE_LOC'    '~'    ORGANIZATION_ID    '~'    SECONDARY_INVENTORY_NAME    '~'	Not applicable
W_CUSTOMER_FIN_PROFL_D	CUSTOMER_FIN_PROFL_WID	P'    '~'    TO_CHAR(CUSTOMER_ID)    '~'    TO_CHAR(SITE_USE_ID)    '~'    CURRENCY_CODE - CUSTOMER_ID is CUST_ACCOUNT_ID from HZ_CUST_ACCOUNTS and CURRENCY_CODE is from HZ_CUST_PROF_CLASS_AMTS	Not applicable
W_CUSTOMER_LOC_D		To get the customer location key, look up W_CUSTOMER_LOC_USE_D as below	Not applicable
W_CUSTOMER_LOC_USE_D		TO_CHAR(SITE_USE_ID) - Get Site Use Id from HZ_CUST_ACCOUNT_ROLES	Not applicable
W_FREIGHT_TERMS_D		LOOKUP_CODE	Not applicable
W_GL_ACCOUNT_D		to_char(ccid)	Not applicable
W_INT_ORG_D	COMPANY_ORG_KEY	COMPANY'    '~'    TO_CHAR(SET_OF_BOOKS_ID)	S_ORG_EXT.ROW_ID
W_INT_ORG_D	*_ORG_KEY	Remove any prefixes and use TO_CHAR()	S_ORG_EXT.ROW_ID

**Table 6–1 (Cont.) Formats to Derive INTEGRATION\_ID**

Dimension	Foreign Key	When Source is Oracle Application	When Source is Siebel Application
W_ORG_D	CUSTOMER_WID	TO_CHAR(CUSTOMER_ID) - CUSTOMER_ID is CUST_ACCOUNT_ID from HZ_CUST_ACCOUNTS	UNION OF S_ORG_EXT AND S_CONTACT. When source is S_ORG_EXT, ROW_ID is passed. When source is S_CONTACT, use 'C-'    ROW_ID - the ROW_ID is the ROW_ID from the S_PERSON (not from S_ORG_EXT). This is the new value passed to lookup the Contact Customers in W_ORG_D for resolving the ACCOUNT_WID (read as CUSTOMER_WID)
W_PAYMENT_METHOD_D		LOOKUP_CODE	Not applicable
W_PAYMENT_METHOD_D		TO_CHAR(TERM_ID)	Not applicable
W_PERSON_D	CUST_CONTACT_WID	TO_CHAR(PARTY_ID) - PARTY_ID from HZ_PARTY_RELATIONS	S_CONTACT.ROW_ID
W_PRODUCT_D	PRODUCT_WID	TO_CHAR(INVENTORY_ITEM_ID)	S_PROD_INT.ROW_ID
W_SALES_PRODUCT_D		TO_CHAR(INVENTORY_ITEM_ID)    '~'    TO_CHAR(ORGANIZATION_ID)	Not applicable

If you are adding an existing dimension, the SIL mapping should be extended to include the preexisting reusable Lookup transformation to that dimension. Pass the dimension's INTEGRATION\_ID through the mapping along the path identified by the X\_CUSTOM column and connect it to the Lookup after the Filter transformation. Also, connect the DATASOURCE\_NUM\_ID to the Lookup. If the dimension is a slowly changing dimension, the fact table's standard or 'canonical' date should be passed to the lookup as well, even if the dimension has not been enabled to capture Category 2 changes.

Remember to connect the ROW\_WID of the Lookup to the X\_CUSTOM transformation and include logic to default this value to 0 if no record is returned from the Lookup. Pass this column on to the Update strategy, and then on to the target.

Update the DAC to include the foreign key to this dimension in the fact table's definition. You should reassemble the subject Area and rebuild the execution plan to ensure that the DAC populates this dimension table before this fact table starts to load.

### 6.3.3 Adding Date Dimensions to Existing Facts

If adding a date dimension to a fact table, you merely have to pass the date itself through the SDE mapping to the stage table. In the SIL mapping, pass the date along

the same path as X\_CUSTOM. Add the reusable expression EXP\_DAY\_DIMENSION\_FK\_RESOLUTION after the Filter. Connect the date to any input and connect the appropriate output to the EXP\_Custom transformation, then on to the Update Strategy and finally to the target.

### 6.3.4 Adding Currencies to an Existing Table

Amounts must be converted from the original currency to the data warehouse currency. Along with the amount, you must pass the currency code if it is not already connected in the mapping. Depending on the source system, there may be more than one currency code

If the source is Siebel, there is generally only one currency type

Other sources may have several currency types. Be sure to read the section on configuring currency codes to get a better understanding of how these work.

If the SIL mapping does not already include it, add the maplet MPLT\_CURCY\_CONVERSION\_RATES after the Filter and connect all required input ports.

Connect the appropriate exchange rate(s) to the EXP\_Custom expression. Use the appropriate exchange rate to convert the amount to the data warehouse currency. Pass the converted currency to the Update strategy then onto the target.

## 6.4 Category 2 Customizations: Adding Additional Tables

This section includes the following topics:

- [Section 6.4.1, "About Creating New Dimension or Fact Tables"](#)
- [Section 6.4.2, "Using Custom Folders in the Oracle Business Analytics Warehouse"](#)
- [Section 6.4.3, "Creating Custom Informatica Workflows"](#)
- [Section 6.4.4, "Important Notes About Customizing the Oracle Business Analytics Warehouse"](#)
- [Section 6.4.5, "Including a Source Table for Change Capture Process"](#)
- [Section 6.4.6, "Adding a New Dimension in the Oracle Business Analytics Warehouse"](#)
- [Section 6.4.7, "Adding a New Fact Table in the Oracle Business Analytics Warehouse"](#)
- [Section 6.4.8, "Adding a New Dimension Table for a New Fact Table in the Oracle Business Analytics Warehouse"](#)

### 6.4.1 About Creating New Dimension or Fact Tables

This section relates to building entirely new tables that will be loaded with data from a source table that is not already extracted from. For example, you may want to create a new "Project" dimension table. In this case, you create new dimension and staging tables as well as new extract and load ETL mappings.

When creating a new custom table, use the prefix "WC\_" to help distinguish custom tables from tables provided by Oracle as well as to avoid naming conflicts in case Oracle later releases a table with a similar name. For example, for your "Project" dimension you may create a "WC\_PROJECT\_DS" and a "WC\_PROJECT\_D" table.

When you create a new dimension or fact table, use the required system columns that are part of each of the data warehouse tables to maintain consistency and the ability to

reference existing table structures. When you create a new table, you need to register the tables and indices in the DAC. You will also have to register in the DAC the new tasks for new Informatica workflows and then reassemble the appropriate subject area and rebuild the appropriate execution plan. For information about assembling subject areas and building execution plans, see the Oracle Business Intelligence Data Warehouse Administration Console Guide.

---

---

**Note:** If you are creating tables on a DB2-UDB database, make sure when you register the tables in the DAC the Not Logged Initially option is enabled.

---

---

#### 6.4.1.1 Required Columns

For custom staging tables, the following columns are required:

- **INTEGRATION\_ID.** Stores the primary key or the unique identifier of a record as in the source table.
- **DATASRC\_NUM\_ID.** Stores the data source from which the data is extracted.

For dimension and fact tables, the required columns are the **INTEGRATION\_ID** and **DATASRC\_NUM\_ID** columns as well as the following:

- **ROW\_WID.** A sequence number generated during the ETL process, which is used as a unique identifier for the data warehouse.
- **ETL\_PROC\_WID.** Stores the ID of the ETL process information. The details of the ETL process are stored in the **W\_ETL\_RUN\_S** table on the data warehouse side. This is also the Process ID on Current Run/Run History screen in the DAC.

#### 6.4.1.2 About the Oracle Business Analytics Warehouse DATASRC\_NUM\_ID Column

All the tables in the Oracle Business Analytics Warehouse schema have **DATASRC\_NUM\_ID** as part of their unique user key. While the transactional application normally ensures that a primary key is unique, it is possible that a primary key is duplicated between transactional systems. To avoid problems when loading this data into the data warehouse, uniqueness is ensured by including the **DATASOURCE\_NUM\_ID** as part of the user key. This means that the rows can be loaded in the same data warehouse tables from different sources if this column is given a different value for each data source.

---

---

**Note:** The **DATASRC\_NUM\_ID** is maintained by the DAC. Make sure that each source system has a unique value assigned. It is possible to have multiple instances of the same source system (for example, a U.S.-based and a European-based Siebel transactional database both loading into the same data warehouse). The two different transactional database systems should be assigned different **DATASOURCE\_NUM\_ID** values in the DAC. The DAC is predefined with one entry for Siebel and the **DATASOURCE\_NUM\_ID** is assigned the value of 1. If you are going to extract from additional Siebel transactional database systems and load the data into the same data warehouse, a different **DATASOURCE\_NUM\_ID** must be assigned to each Siebel transactional database system.

---

---

## 6.4.2 Using Custom Folders in the Oracle Business Analytics Warehouse

If you want to make changes to the Oracle Business Analytics Warehouse, you must create a custom folder and make the changes in it. Do not change objects in any of the shipped folders unless explicitly directed by Oracle. The shipped folders and the objects within them may be overwritten in future upgrades.

The deployed repository does not include any custom folders; you must create your own. You should create a custom folder for each SDE folder you have deployed that will have customizations. These folders hold the extract mappings to various sources. You should also create a separate custom folder for customizations you want to make to the SILOS folder. Do not store customized extract and load mappings in the same folder.

The easiest way to modify an object is to copy an existing object from the shipped folder into the corresponding custom folder and to reuse existing business components, source and target definitions, transformations, mapplets, and mappings.

---

**Note:** If source tables are extended, then the source tables require manual editing in Informatica Designer. Do not import the tables from the database into the repository, because it changes the source table definitions for the entire Oracle Business Analytics Warehouse.

---

When importing new tables from the database into the custom folder, use the Oracle Business Analytics Warehouse and transactional database ODBC database connections (using the ODBC drivers supplied by your database vendor) to connect to the source and target databases.

After importing new table definitions, change the database type to Oracle in the Informatica repository, regardless of the database platform you are using. This has no effect on the choice of a relational database. This step is very important because in Informatica if the database type of the source tables are not identical, all mappings and workflows that refer to the source tables will be invalid.

## 6.4.3 Creating Custom Informatica Workflows

You must create custom workflows for all customized mappings. The general requirements for creating custom workflows are as follows:

- Create the workflow such that each loads only one table. This helps to integrate the workflow with the DAC.
- The workflow name should match a session name that is used inside the workflow. This helps the DAC to collect certain statistics.
- The flag "Fail parent if this task fails" should be checked for all the sessions within a workflow.
- The flag "Fail parent if this task does not run" should be checked for all the sessions within a workflow.
- The Stop on Errors parameter should be set to 1 for all sessions within a workflow. This parameter is located in the Error Handling area of the Config Object tab in Informatica Designer.
- Set the appropriate source and target connection values in Informatica Designer.
- If the workflow is going to be used for full load command, you can choose to load in bulk mode (applicable only to Oracle and DB2-UDB databases). If the workflow

is going to be used for the full load command in the DAC, then in the Properties tab of Informatica Designer, set the Target Load type to Bulk. This requires that the target table does not have any indices during the load. The DAC will drop the indices automatically; no action is required on your part.

- For all entities, such as dimension and fact tables, create two workflows, one to be used for a full load and the other to be used for an incremental load. Both workflows are based on the same mapping. The same mapping gets executed during both full and incremental loads. This provides an opportunity to tune each of these load scenarios.
- If the workflow is designed for loading a dimension in a full mode, make sure the workflow also includes a session for the unspecified row creation.
- When defining a task in the DAC, you must choose the appropriate truncate option. This helps the DAC to decide whether to drop and create indices on the table.
- You should not truncate target tables using the Informatica 'truncate target' option. It is especially important that the DAC handle truncating tables when extracting from more than a single source system and loading into the same data warehouse. The DAC will dynamically determine when tables need to be truncated. By setting the truncate option in the Informatica workflow, the table will always be truncated, limiting your ability to extract and load data from multiple sources. For example, you cannot extract data from two systems and load the data into the same staging table in parallel as the Informatica session will truncate the staging table when another session is running.
- If some sessions have to be run in serial order and if all the sessions need to be rerun upon failure of the workflow, then you should consider designing a single workflow that has sessions run in serial. If upon failure, not all of the sessions need to be run again, then consider designing separate workflows and define dependencies in the DAC.
- The custom workflows can be plugged into an ETL process by registering them in the DAC. All new tasks need to be registered in the DAC, with the appropriate properties set. Also, you must register in the DAC, source and target tables, task definitions, and dependencies.

## 6.4.4 Important Notes About Customizing the Oracle Business Analytics Warehouse

All custom work, unless stated otherwise, must be done in the Custom folder so that the custom work can be preserved during an Informatica repository upgrade. Doing work on the standard folder should be avoided whenever possible. An upgrade of the Informatica repository overrides any changes to the standard folder.

### 6.4.4.1 Additional Customization Instructions

- **Table definitions in Informatica.** Make sure that the SQL style is set to Oracle while importing the table definitions from external data sources. Even if the actual data source is of another database type, such as DB2 or MSSQL, it does not affect the logic of how data gets loaded.
- **Update strategy.** For loading new fact and dimension tables, design a custom process on the source side to detect the new and modified records. The SDE process should be designed to load only the changed data (new and modified). If the data is loaded without the incremental process, the data that was previously loaded will be updated again, which is a costly process. For example, the logic in the preconfigured SIL mappings looks up the destination tables based on the

INTEGRATION\_ID and DATASRC\_NUM\_ID and returns the ROW\_WID if the combination exists, in which case it updates the record. If the lookup returns null, it inserts the record instead. In some cases, last update date(s) stored in target tables are also compared in addition to the columns mentioned above to determine insert or update. Look at the similar mappings in the preconfigured folder for more details.

- **ETL process.** When using multiple sources for the data warehouse, you can decide to load from all of them at the same time or at different time frequencies using different execution plans.
- **Truncating target tables.** Truncating should be done through the DAC. A single task has place holders for a full load, and one for an incremental load.
  - For the SDE workflows, the commands for full load and incremental load are the same. They should have the Truncate Always flag checked in the DAC. For these kinds of tasks, the command for full load and incremental load are based on the same mapping.
  - For SIL workflows, the command can be different for full and incremental loads. They should have the Truncate For Full Load option checked in the DAC. When a table gets truncated, the indices are automatically dropped and created after the data is loaded. The workflow associated with the full load command can have the Bulk Load option turned on for an optimized version of the mapping that quickly inserts data. Note that if there are indexes on the table, the bulk load may fail, so it is very important that the indices are registered in the DAC and that you drop all of the indexes on this table during a full load if you use the bulk load option.
  - If a source requires an auxiliary task, it needs to be run only during incremental mode. So, for these tasks, the full load command is empty. No truncate options should be set.
- **ETL\_PROC\_WID.** Use the same ETL\_PROC\_WID in W\_PARAM\_G table in custom mappings. ETL\_PROC\_WID is a reference key to Run History in the DAC. To use the same ETL\_PROC\_WID, copy the reusable lookup (called LKP\_ETL\_PROC\_WID) defined in the SILOS folder. The input to the lookup is a constant (hard coded to 1).
- **DATASRC\_NUM\_ID.** Use a parameter to define this value in the mapping. The DAC will automatically create a parameter file with the correct DATASOURCE\_NUM\_ID, which will be picked up by the parameter in the mapping. This allows you to make multiple copies of the same mapping when you have multiple instances of the same transactional database type. You do not have to do any additional hard-coding other than to register the sources in the DAC.
- **Creating indices and naming conventions.** Staging tables typically do not require any indices. Use care to determine if indices are required on staging tables. Create indices on all the columns that the ETL will use for dimensions and facts (for example, ROW\_WIDs of Dimensions and Facts, INTEGRATION\_ID and DATASRC\_NUM\_ID, and flags). Carefully consider which columns or combination of columns filter conditions should exist, and define indices to improve query performance. Inspect the OTB objects for guidance. Name all the newly created tables as WC\_. This helps visually isolate the new tables from the OTB tables. Keep good documentation of the customizations done; this helps when upgrading your data warehouse. Once the indices are decided upon, they should be registered in the DAC, either manually or by right-clicking on the certain table and invoking the Import Indices command.

- **Currency.** For currency-related data, populate the table with the base currency and exchange date field (in order to convert the data appropriately). The data for currency conversion should be maintained in the main data source. (The currency data is maintained by converting all the currency information into a single Base Currency code specified in the DAC.)
- **Day Dimension.** For the data that is related to `W_DAY_D`, use the reusable transformation `EXP_DAY_DIMENSION_FK_RESOLUTION`. Note that this transformation will take a date as input and return the foreign key to the Day dimension as output in the appropriate format (YYYYMMDD) avoiding the need of costly join or lookup to the `W_DAY_D` dimension table each time for resolution. Copy the reusable transformation and use it.
- **List Of Values.** This applies in particular to Category 1 and II. The preconfigured columns that depend on a list of values have a language-dependent column and a language-independent column. Use the maplet `MPLT_LOV_TRANSLATION` to populate the language-dependent and independent columns in the dimension tables. For fact tables, use `MPLT_LOV_D_ROW_WID` to create a new foreign key to the LOV dimension. You could also deal with translations directly in SQL overrides to improve performance.

## 6.4.5 Including a Source Table for Change Capture Process

This procedure applies to Siebel source tables only.

### To include a source table for the change capture process

1. Verify whether the source table is registered in the DAC.
  - a. If an entry is not in the DAC, create a new record for the table and assign an image suffix to it.
  - b. If the table is registered, make sure there is an image suffix assigned for this table.
2. If an image suffix does not exist for the source table, assign one.

The image suffix should be three characters long. The recommended naming convention is to start with C. For example, use C1, C2, CA1, CA2, and so on.
3. Make sure the image suffix is not used by other tables by querying the tables list in the DAC for that image suffix.

The DAC client does not validate this information when entering data.
4. Create Image tables in the Siebel transactional database.
  - a. Right-click on the table record in the DAC client, and choose Generate Change Capture Scripts.

This will assist you in creating the image tables, necessary indices, and triggers if you plan on tracking deletes on the transactional database.
  - b. Execute these scripts on the transactional database if you have the appropriate permission, otherwise ask the OLTP DBA to create it.
5. For the task that is created for the extract process, set the Build Image flag to True and select the new table as an auxiliary table or primary table.

---



---

**Note:** At the end of the change capture process, the DAC creates a view on the actual source table. Use this view as the main source table in all the extract procedures. For example, if your new source table is S\_COMPENSATION, then the default view name is V\_COMPENSATION.

---



---

## 6.4.6 Adding a New Dimension in the Oracle Business Analytics Warehouse

Follow this procedure to add a new dimension in the Oracle Business Analytics Warehouse.

### To add a new dimension and use it with an existing fact table

1. Create a DDL for the new dimension based on the standard structure (with appropriate system columns). Create a staging table for this dimension.
2. Register the new source table and its staging table (if it does not already exist) in the DAC repository and associate it with the appropriate database connection.
3. Create a new custom map SDE\_XYZ to populate the dimension stage. Instead of the actual source table (for example S\_ABC), use the view that will be generated by the change capture process (for example V\_ABC) in the SQL so that it extracts only the incremental data. Use existing reference maps as examples of how to populate the system columns. Make sure you truncate the stage table in corresponding tasks.
4. Create a new custom map SIL\_XYZ to populate the new dimension from the stage table. Use the above referenced map as example for how to populate the system columns.
5. Register the new dimension table in the DAC and associate it with the appropriate database connection.  
  
If you are planning to build a new dimension incrementally, assign an image suffix to the source table.
6. Register the workflows as tasks in the DAC.
7. For SDE mapping of the dimension make sure you set the Build Image flag to True, and the Truncate Always option to True. And in the list of source tables, mark the primary/auxiliary source(s) of this dimension.
8. For SIL workflows of the dimension make sure you set only Truncate for Full Load option to True.
9. Make sure the target table of the SDE\_XYZ is defined as source table for SIL\_XYZ.

## 6.4.7 Adding a New Fact Table in the Oracle Business Analytics Warehouse

Follow this procedure to add a new fact table in the Oracle Business Analytics Warehouse.

### To add a new fact table

1. Create a DDL for the new fact based on the standard structure (with appropriate system columns). Create a staging table for this fact.
2. Register the new source table (if it does not already exist) in the DAC repository and associate it with a database connection.

3. Create the change capture tables by right-clicking and selecting Generate Change Capture Scripts. For instructions, see [Section 6.4.5, "Including a Source Table for Change Capture Process"](#).
4. Create SDE mappings to populate the custom stage table. Use the view created by change capture as the main table in the SQL so that it extracts only the incremental data. Use the reference maps (above) as examples of how to populate the system columns. Be sure to truncate the stage table in corresponding workflows.
5. Create SIL mapping to populate the custom fact table. Use reference maps as examples of how to populate the system columns.
6. Use lookups or SQL override joins to dimension tables for populating dimension foreign keys (ROW\_WIDS) pointing to the existing dimension.
7. In the DAC, register the target tables.
8. Create new tasks for the workflows.
9. For the SDE task, make sure you have the Build Image flag set to True, and list all the source tables that it queries from. Choose one or more tables as primary or auxiliary. For the target tables choose the staging table. Set the Truncate Always flag to True.
10. For the SIL task, list all the dimensions that will be required under source tables.

### 6.4.8 Adding a New Dimension Table for a New Fact Table in the Oracle Business Analytics Warehouse

The steps for creating a new dimension table are similar to the steps for incremental change capture.

#### To add a new dimension table for a new fact table

1. In the new custom fact loading mapping (SIL), use lookups for getting foreign keys to the new dimension.
2. Use existing maps as examples.

## 6.5 Category 3 Customizations: Adding New Data as a Whole Row into a Standard Dimension Table

Follow this procedure to add new data as a whole row into a standard dimension table in the Oracle Business Analytics Warehouse.

#### To add new data as a whole row into the standard dimension table

1. Identify and understand the existing structure of staging tables. Refer to *Oracle Business Analytics Warehouse Data Model Reference* for the table structure. Non-system columns can include the null value.
2. Create a custom SDE mapping to load the data into the staging table in the custom folder for this purpose. The staging table needs to be populated with incremental data (rows that have been added or changed since the last Refresh ETL process), for performance reasons.
3. Populate the INTEGRATION\_ID column with the unique identifier for the record.

The combination of INTEGRATION\_ID and DATASRC\_NUM\_ID is unique. When importing the data, make sure that a unique identifier for the external data source is inserted in the DATASRC\_NUM\_ID column. The DATASRC\_NUM\_ID is set to 1 for mappings that source data from the Siebel transactional database. This is a

reserved value and is used in all standard mappings. For example, a value of 2 can be defined for DATASRC\_NUM\_ID in the custom SDE mapping. The standard SDE mappings populate the INTEGRATION\_ID column of the dimension staging table (used for resolving the dimension's Siebel transactional database ROW\_ID value). The custom process must be used to populate the same column with a unique identifier from the external data source.

4. After the data is populated in the staging table, use the standard SIL mappings to populate the dimension target tables.
5. Modify the SDE and SIL mappings of all the related fact tables (fact tables that need to be linked to this dimension).

The custom fact SDE mappings must populate the foreign key column of the changed dimension (using a custom map table process to convert from Siebel's row IDs to the external data source row IDs). The custom SIL mapping should be modified to use the appropriate DATASRC\_NUM\_ID, because the standard SIL mappings assume DATASRC\_NUM\_ID for the dimensions are the same as the fact table's DATASRC\_NUM\_ID.

It is very important to decide when the data is going to be loaded. If it is going to be loaded along with the Siebel source data, you must be careful with how you handle failure recovery. The preconfigured workflows truncate the target staging table prior to loading. Upon failure, when the DAC server restarts the task, all the data is truncated and all the data is loaded again.

If the data from the external source gets loaded into the same staging table, be careful with how you handle this situation, since you cannot use the truncate table functionality. The data migrating into the staging table is not incrementally loaded, and, therefore, should be cleaned up prior to attempting to load this table again.

In such a case, it is recommended that you encapsulate the extract part from both the sources inside an Informatica workflow. Upon failure of either of the extracts, the whole workflow gets rerun. Note that the data from both the sources should be run at the same time, all the time.

If it is decided that the data is going to be loaded at different time frequencies, then the new SDE workflows need not depend on the preconfigured SDE workflows and can use the Truncate Table option for failure recovery. In this case, in the DAC Design view, define a new execution plan in the Execution Plans tab, and define the new data source in the Database Connections child tab. Make sure the shared SIL process depends on the SDE processes from both sources.

## 6.6 Configuring Extracts

Each application has prepackaged logic to extract particular data from a particular source. This section discusses how to capture all data relevant to your reports and ad hoc queries by addressing what type of records you want and do not want to load into the data warehouse, and includes the following topics:

- [Section 6.6.1, "Extracting Additional Data"](#)
- [Section 6.6.2, "Setting Up the Delimiter for Source Files"](#)

### 6.6.1 Extracting Additional Data

You can configure extract mappings and mapplets in the Oracle Business Analytics Warehouse to accommodate additional source data. For example, if your business

divides customer information into separate tables based on region, then you would have to set up the extract mapping to include data from these tables.

### 6.6.1.1 Extracting New Data Using an Existing Source Table

Extract mappings generally consist of a source table or Business Component, an Expression transformation, and a staging table. If you want to extract new data using the existing mapping, you have to modify the extract mapping to include the new data by performing the following tasks:

#### To modify an existing mapping to include new data

1. Modify the existing Business Component to extract information from the source, and add it to an appropriate extension column.

**Tip:** You can perform calculation transformations in the Business Component mapplet or the Source Adapter mapplet in the extract mapping. However, do not use performance-expensive calculations in the extract that could tie up your source transaction system. For these types of calculations, it is recommended that you perform them in the Source Adapter mapplet.

2. Modify the Expression transformation to perform any necessary transformations.
3. Connect all input and output ports within the extract mapping so that the data moves from the source or Business Component to the Expression transformation and through the Source Adapter mapplet, and finally to the staging table's appropriate extension column.

You have to determine which type of extension column to map the data to in the staging table. After you modified the extract mapping, you would also have to modify the corresponding load mapping to make sure the extension columns that you added are connected all the way from the staging table to the warehouse table.

### 6.6.1.2 Extracting Data from a New Source Table

Business Components are packaged as mapplets, which reside in source-specific folders within the repository. Business Components are used to extract data from the source system. You can configure these mapplets to perform the following:

- **Extract data from a new source table**
- **Set incremental extraction logic**

The following procedure contains instructions for adding a new table to the Business Component. The procedure includes adding a new source definition, connecting the ports to the Source Qualifier, editing the Source Qualifier, connecting the ports to the Output transformation, and editing the Output transformation.

#### To add a new source table to an existing Business Component mapplet

1. In PowerCenter Designer, open the applicable source system configuration folder.
2. Open Mapplet Designer tool.
3. Drag the Business Component mapplet into Mapplet Designer to view the transformations that comprise the Business Component.
4. Expand the Sources folder, and copy a source table into the mapplet by dragging and dropping the table into Mapplet Designer.

5. Connect the applicable ports from the new source definition to the Source Qualifier by clicking on the port in the new source table and dragging it to the connecting port in the Source Qualifier.
6. Double-click the Source Qualifier to open the Edit Transformations box.  
In the Ports tab, make any changes to the new ports for data type, precision, scale, or all these values, as necessary.
7. Connect the applicable ports from the Source Qualifier to the Maplet Output transformation (MAPO).

---



---

**Note:** In some cases, the Business Component contains an Expression transformation between the Source Qualifier and the MAPO.

---



---

8. In the Properties tab, make changes to the SQL statement as necessary.
9. Validate and save your changes to the repository.

### 6.6.2 Setting Up the Delimiter for Source Files

You must make sure that the delimiter used in the CSV files is not present in the source data. Oracle Business Analytics Warehouse is preconfigured with a comma as the source file delimiter. If your data contains commas, then you need to enclose the data fields using a character that is not present in the data set. For example, common enclosing characters include single quotes and double quotes.

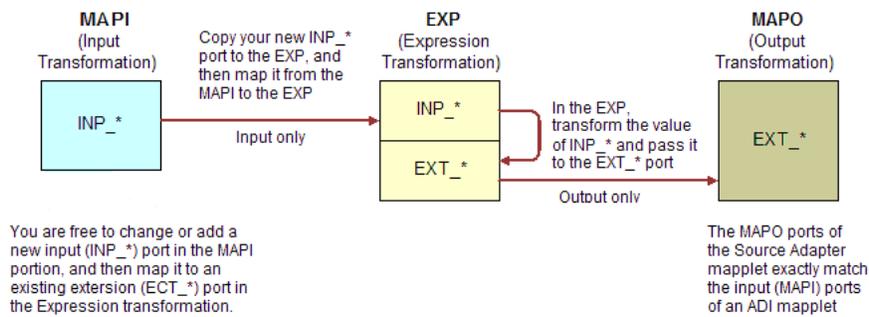
#### To set up the delimiter for source files

1. Open the CSV file.
2. Enclose the data fields with the enclosing character that you have identified.  
You can identify an enclosing character by identifying a character that is not present in the source data. Common enclosing characters include single quotes and double quotes.
3. Save and close the CSV file.
4. Identify all the source definitions associated with the modified files.
5. Change the properties for each of these source definitions to use the enclosing character.
6. Validate and save your changes to the repository.

### 6.6.3 Configuring a Source Adapter Maplet

The majority of all source-specific transformations occur in the Source Adapter maplet; source-independent transformations generally take place in the Analytic Data Interface (load mapping). The Source Adapter maplet converts source-specific data elements into standard formats and then stores them in a staging table. The source independent loading mapping then picks up these records, which are already transformed into standard format.

The figure below illustrates the three components of the Source Adapter maplet that allow transformations of data to occur. The three components are Maplet Input (MAPI), Expression transformation (EXP), and Maplet Output (MAPO).

**Figure 6–6 Components of the Source adapter mapplet**

In the figure above, if the input data is transformed, the data is passed to the Expression transformation (EXP) as input only. After the data is transformed, it is output through a new port, which is prefixed with EXT\_. If the data is not transformed, it comes in as input-only and leaves through an output-only port.

If you want to add a new transformation, you must add a new port to contain the expression that is used to transform the data.

#### To add a new port to the Source Adapter mapplet

1. In PowerCenter Designer, open the applicable source system configuration folder.
2. Open the applicable Source Adapter mapplet.
3. Double-click the MAPI component of the mapplet, and add a new input port following the INP\_\* naming convention.
4. Copy the new input port from the MAPI to the Expression transformation.
5. Connect the new port from the MAPI to the Expression transformation.
6. In the Expression transformation, uncheck the Output indicator for the new input port; you use the value from this port in an transformation expression.
7. Perform any necessary transformations within the Expression transformation.  
The transformed data is passed out of an EXT\_\* output-only port.
8. Connect the port from the Expression transformation to the MAPO.
9. Validate and save your repository.

## 6.7 Configuring Loads

The Oracle Business Analytics Warehouse prepackages load mappings for every data warehouse table.

### 6.7.1 Filtering and Deleting Records

---

**Note:** This section does not apply to Oracle's Siebel sources.

---

In a typical implementation, records that are deleted from your source system are not removed from the Oracle Business Analytics Warehouse. If you want to mark these records as deleted in the data warehouse, which were removed from the source system's database and archived in a separate database, you must enable the primary extract and delete mappings.

Primary extract mappings flag records that are deleted from the data warehouse. Delete mappings sets the DELETE\_FLG column to 'Y' for these records in the warehouse tables. When enabled, primary extract and delete mappings by default look for any records removed from the source system's database. If these mappings find that the records no longer exist in that database, the mappings mark them as deleted in the data ware

---

**Caution:** It is important to note that delete and primary extract mappings must always be disabled together; you may not disable only one type.

---

## 6.7.2 About Primary Extract and Delete Mappings Process

---

**Note:** This section does not apply to Oracle's Siebel adapter, which does not have Primary Extract mappings.

---

Before you decide to enable primary extract and delete sessions, it is important to understand their function within the Oracle Business Analytics Warehouse. Primary extract and delete mappings allow your analytics system to determine which records are removed from the source system by comparing primary extract staging tables with the most current Oracle Business Analytics Warehouse table.

The primary extract mappings perform a full extract of the primary keys from the source system. Although many rows are generated from this extract, the data only extracts the Key ID and Source ID information from the source table. The primary extract mappings load these two columns into staging tables that are marked with a \*\_PE suffix.

The figure below provides an example of the beginning of the extract process. It shows the sequence of events over a two day period during which the information in the source table has changed. On day one, the data is extracted from a source table and loaded into the Oracle Business Analytics Warehouse table. On day two, Sales Order number three is deleted and a new sales order is received, creating a disparity between the Sales Order information in the two tables.

**Figure 6–7 Extract and load mappings**

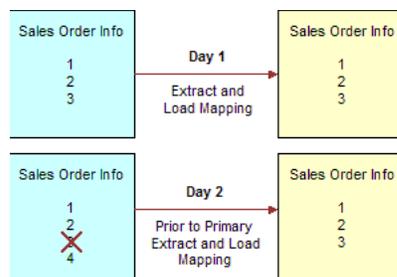
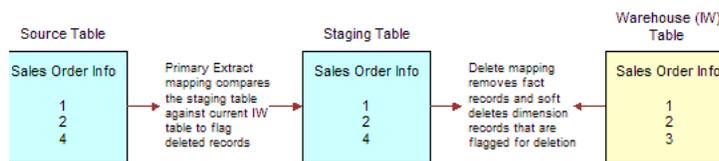


Figure 6–8 shows the primary extract and delete process that occurs when day two's information is extracted and loaded into the Oracle Business Analytics Warehouse from the source. The initial extract brings record four into the Oracle Business Analytics Warehouse. Then, using a primary extract mapping, the system extracts the Key IDs and the Source IDs from the source table and loads them into a primary extract staging table.

The extract mapping compares the keys in the primary extract staging table with the keys in the most current the Oracle Business Analytics Warehouse table. It looks for records that exist in the Oracle Business Analytics Warehouse but do not exist in the staging table (in the preceding example, record three), and sets the delete flag to Y in the Source Adapter mapplet, causing the corresponding record to be marked as deleted.

The extract mapping also looks for any new records that have been added to the source, and which do not already exist in the Oracle Business Analytics Warehouse; in this case, record four. Based on the information in the staging table, Sales Order number three is physically deleted from Oracle Business Analytics Warehouse, as shown in [Figure 6–8](#). When the extract and load mappings run, the new sales order is added to the warehouse.

**Figure 6–8 Primary Extract and Delete Mappings**



### 6.7.3 About Working with Primary Extract and Delete Mappings

The primary extract (\*\_Primary) and delete mappings (\*\_IdentifyDelete and \*\_Softdelete) serve a critical role in identifying which records have been physically deleted from the source system. However, there are some instances when you can disable or remove the primary extract and delete mappings, such as when you want to retain records in the data warehouse that were removed from the source systems' database and archived in a separate database.

Because delete mappings use Source IDs and Key IDs to identify purged data, if you are using multiple source systems, you must modify the SQL Query statement to verify that the proper Source ID is used in the delete mapping. In addition to the primary extract and delete mappings, the configuration of the delete flag in the load mapping also determines how record deletion is handled.

You can manage the extraction and deletion of data in the following ways:

- Deleting the configuration for source-archived records
- Deleting records from a particular source
- Enabling delete and primary-extract sessions
- Configuring the Record Deletion flag
- Configuring the Record Reject flag

This topic provides procedures for these management tasks.

#### 6.7.3.1 Deleting the Configuration for Source-Archived Records

Some sources archive records in separate databases and retain only the current information in the main database. If you have enabled the delete mappings, you must reconfigure the delete mappings in the Oracle Business Analytics Warehouse to retain the archived data.

To retain source-archived records in the Oracle Business Analytics Warehouse, make sure the `$$LAST_ARCHIVE_DATE` parameter value is set properly to reflect your

archive date. The delete mappings will not mark the archived records as 'deleted'. For more information about extract and delete mappings, see [Section 6.7.3, "About Working with Primary Extract and Delete Mappings"](#).

### 6.7.3.2 Enabling Delete and Primary Extract Sessions

If you want to mark your source-deleted records as deleted in the Oracle Business Analytics Warehouse, you need to enable the delete and primary extract tasks for your application.

#### To enable primary extract and delete sessions

1. Log into DAC
2. Go to your container
3. Select the 'Tasks' tab
4. Query for all tasks containing the string 'Delete' or 'Primary'
5. Uncheck the 'Inactive' check box for those tasks.

## 6.8 Configuring Slowly Changing Dimensions

The Oracle Business Analytics Warehouse provides Type II slowly changing dimension (SCD) functionality, which allows you to track the history of updates to dimension records. When a record in the Oracle Business Analytics Warehouse has an update, the updated information is posted into a new row and the old information is kept for historical reporting purposes.

The Oracle Business Analytics Warehouse identifies and applies the slowly changing dimension logic chosen by the user after data has been extracted and transformed to be source-independent. Users may configure Oracle BI Applications to support both Type I SCDs, in which data is overwritten with updates, and Type II SCDs, in which the original records are maintained while a new record stores the updated data. Choosing Type I or Type II SCDs depends on identifying your historically significant attributes.

By default, all dimensions are using Type I updates. If you need change a dimension to Type II SCD update, follow the procedure below.

#### To turn a dimension to Type II SCD update enabled

1. In the parameter file `parameterfileDW.txt` located under Informatica PowerCenter 7.1.4\Server\SrcFiles
2. Locate the parameter section for the `SIL_*` mapping which load this dimension.
3. Set the `$$TYPE2_FLG` to 'Y'.

### 6.8.1 About Identifying Historically Significant Attributes

You may want to retain a history of all the updates to a particular dimension so that you can use them in reports. These dimensions are known as *historically significant* attributes. For example, if a customer moves to a different region and you assign that customer a new regional salesperson and territory ID, you may want to keep records of that customer's account history with the original salesperson and territory ID. In this case, the salesperson and territory IDs are *historically significant* attributes. In contrast, you may have a load that populates the telephone number field. If your business does not perform data analysis on phone number history, then this information may be considered a *historically insignificant* attribute.

Identifying attributes as significant or insignificant allows you to determine the type of SCD you require. However, before you can select the appropriate type of SCD, you must understand their differences.

### 6.8.1.1 About the Extract View

The extract view of any given table in the Staging Area consists of four types of records:

- New records
- Changed records with data that is historically insignificant
- Changed records having historical significance
- Changed records whose changes have no significance of any kind and are ignored altogether

Of the four kinds of records, only the first three are of interest for the data mart. Of those three, brand new records and records whose changes are tracked as SCDs are both treated as new and become inserts into the data warehouse. Records with changes that are important but not historically tracked are overwritten in the data warehouse, based on the primary key.

## 6.8.2 Type I and Type II Slowly Changing Dimensions

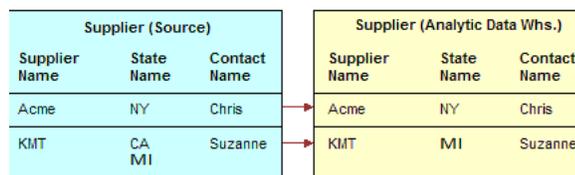
After you have correctly identified your significant and insignificant attributes, you can configure the Oracle Business Analytics Warehouse based on the type of slowly changing dimension (SCD) that best fits your needs—Type I or Type II.

### 6.8.2.1 Type I Slowly Changing Dimension

A Type I SCD overwrites the column's value and is the default SCD for the Oracle Business Analytics Warehouse. Although a Type I does not maintain history, it is the simplest and fastest way to load dimension data. Type I is used when the old value of the changed dimension is not deemed important for tracking or is an historically insignificant attribute. For example, you may want to use Type I when changing incorrect values in a column.

In the figure below, the State Name column for the supplier KMT is changed in the source table Suppliers, because it was incorrectly entered as California. When the data is loaded into the data warehouse table, no historical data is retained and the value is overwritten. If you look up supplier values for California, records for KMT do not appear; they only appear for Michigan, as they have from the beginning.

**Figure 6–9 An example Type 1 Slowly Changing Dimension**



### 6.8.2.2 Type II Slowly Changing Dimension

A Type II SCD creates another record and leaves the old record intact. Type II is the most common SCD because it allows you to track historically significant attributes. The old records point to all history prior to the latest change, and the new record maintains the most current information.

Slowly changing dimensions work in different parts of a star schema (the fact table and the dimension table). The figure below shows how an extract table (SOURCE\_CUSTOMERS) becomes a data warehouse dimension table (W\_ORG\_D). Although there are other attributes that are tracked, such as Customer Contact, in this example there is only one *historically tracked attribute*, Sales Territory. This attribute is of historical importance because businesses frequently compare territory statistics to determine performance and compensation. Then, if a customer changes region, the sales activity is recorded with the region that earned it.

This example deals specifically with a single day's extract, which brings in a new record for each customer. The extracted data from SOURCE\_CUSTOMERS is loaded into the target table W\_ORG\_D, and each record is assigned a unique primary key (ROW\_WID).

**Figure 6–10 An example Type 2 Slowly Changing Dimension**

SOURCE_CUSTOMERS			W_ORG_D			
Customer Name	Sales Territory	Customer Contact	Customer KEY	Customer Name	Sales Territory	Customer Contact
ABC Co.	East	Mary	101	ABC Co.	East	Mary
XYZ Inc.	West	John	102	XYZ Inc.	West	John

However, this data is not static; the next time a data extract shows a change for your customers in W\_ORG\_D, the records must change. This situation occurs when slowly changing dimensions are invoked. The figure below shows that records for the two customers, ABC Co., and XYZ inc. have changed when compared with the figure below. Notice that ABC's Customer Contact has changed from Mary to Jane, and XYZ's Sales Territory has changed from West to North.

As discussed earlier in this example, the Customer Contact column is historically insignificant; therefore a Type I SCD is applied and Mary is overwritten with Jane. Because the change in ABC's record was a Type I SCD, there was no reason to create a new customer record. In contrast, the change in XYZ's record shows a change of sales territory, an attribute that is historically significant. In this example, the Type II slowly changing dimension is required.

As shown in the figure below, instead of overwriting the Sales Territory column in the XYZ's record, a new record is added, assigning a new ROW\_WID, 172, to XYZ in W\_ORG\_D. XYZ's original record, 102, remains and is linked to all the sales that occurred when XYZ was located in the West sales territory. However, new sales records coming in are now attributed to ROW\_WID 172 in the North sales territory.

**Figure 6–11 An example Type 2 Slowly Changing Dimension**

TS_CUSTOMERS			IA_CUSTOMERS			
Customer Name	Sales Territory	Customer Contact	Customer KEY	Customer Name	Sales Territory	Customer Contact
ABC Co.	East	<del>Mary</del> Jane	101	ABC Co.	East	Jane
XYZ Inc.	<del>West</del> North	John	102	XYZ Inc.	West	John
			172	XYZ Inc.	North	John

### 6.8.2.3 Effective Dates

Effective dates specify when a record was effective. For example, if you load a new customer's address on January 10, 2003 and that customer moves locations on January 20, 2003, the address is only effective between these dates. Effective Dates are handled in the following manner:

- If the source supplies both effective dates, these dates are used in the warehouse table.
- If the source does not supply both the effective to and effective from dates, then the Type II logic creates effective dates.
- If the source supplies one of the two effective dates, then you can set up the Oracle Business Analytics Warehouse to populate the missing effective dates using a wrapper mapping. This situation is discussed in this section. By default, these wrapper sessions are disabled and need to be enabled in order to be executed.

For example, in the W\_ORG\_D table previously discussed, XYZ moved to a new sales territory.

If your source system supplied historical data on the location changes, your table may contain a record for XYZ in the West sales territory with an effective from date of January 1, 2001 and an effective to date of January 1, 3714. If the next year your source indicates XYZ has moved to the North sales territory, then a second record is inserted with an effective from date of January 1, 2002, and an effective to date of January 1, 3714, as shown in [Table 6-2](#).

**Table 6-2 Records Before a Wrapper Session in W\_CUSTOMER**

Customer Name	Sales Territory	Customer Contact	Effective From	Effective To	Current
ABC	East	Jane	1/1/2001	1/1/3714	Y
XYZ	West	John	1/1/2001	1/1/3714	Y
XYZ	North	John	1/1/2002	1/1/3714	Y

Note your first record for XYZ still shows as effective from January 1, 2001 to January 1, 3714, while a second record has been added for XYZ in the North territory with the new effective from date of January 1, 2002. In this second record the effective to date remains the same, January 1, 3714.

When you schedule a wrapper session to execute, the effective dates for the first XYZ are corrected (January 1, 2001-January 1, 2002), and the Current Flag is adjusted in the Analytic Data Interface (load mapping) so that only the second record (January 1, 2002-January 1, 3714) is set to Y. After the wrapper session completes its work, you have Type II information for XYZ in your data warehouse rather than two disparate records, as shown in [Table 6-3](#).

**Table 6-3 Records After a Wrapper Session in W\_CUSTOMER**

Customer Name	Sales Territory	Customer Contact	Effective From	Effective To	Current
ABC	East	Jane	1/1/2001	1/1/3714	Y
XYZ	West	John	1/1/2001	1/1/2002	N
XYZ	North	John	1/1/2002	1/1/3714	Y

In the previous paragraph, the wrapper session corrected the effective to dates and current flag. However, if the record's dates had been correct, the wrapper mapping would simply have set the current flag as needed, because its logic is set to check dates and flags and only adjust columns that contain discrepancies. Finally, if your source system does not supply any Type II information, you may disable the wrapper session completely; in this case all Type II work is handled by the Analytics Data Interface maplet.

## 6.9 About Stored Lookups

A lookup transformation allows you to specify a reference table, and then retrieve information such as code descriptions, exchange rates, and currency codes. The main types of preconfigured lookups within the Oracle Business Analytics Warehouse are:

- [Section 6.10, "Codes Lookup"](#).
- Dimension Key Lookups. For more information, see [Section 6.11, "About Resolving Dimension Keys"](#).

## 6.10 Codes Lookup

Some source systems use intelligent codes that are intuitively descriptive, such as HD for hard disks, while other systems use non-intuitive codes (like numbers, or other vague descriptors), such as 16 for hard disks. While codes are an important tool with which to analyze information, the variety of codes and code descriptions used poses a problem when performing an analysis across source systems. The lack of uniformity in source system codes must be resolved to integrate data for the Oracle Business Analytics Warehouse.

The code lookup in the load mapping integrates both intelligent and non-intuitive codes by performing a separate extract for codes, and inserting the codes and their description into a codes table. The codes table provides the load mapping with a resource from which it can automatically perform a lookup for code descriptions.

The Analytic Data Interface's architecture uses components, as well as both fact and dimension tables, to facilitate lookup functionality. The following components and process are used in a lookup:

### 6.10.1 W\_CODES Table

The load control table `W_CODES` consolidates all codes for future reference and assigns them a category and a single language for efficient lookup capability.

### 6.10.2 Codes Mappings

The Oracle Business Analytics Warehouse uses mappings designed to extract codes from source systems and populate the `W_CODES` table in preparation for use by the load mapping.

To understand how codes mappings function, it is helpful to first understand the columns within `W_CODES`. [Table 6–4](#) describes these columns.

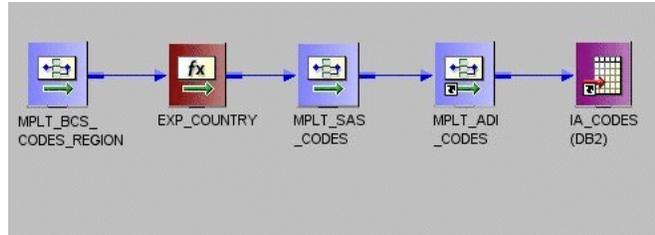
**Table 6–4 Columns in Code Mapplet**

Column	Description
DATASOURCE_NUM_ID	Unique identifier of the source system from which data was extracted
SOURCE_CODE1	The first code in the hierarchy of the various source system codes used to identify a particular code and description combinations
SOURCE_CODE2	The second code in the hierarchy of the various source system codes used to identify a particular code and description combinations
SOURCE_CODE3	The third code in the hierarchy of the various source system codes used to identify a particular code and description combinations
SOURCE_DESC_1	Short description of the source system code

**Table 6–4 (Cont.) Columns in Code Mapplet**

Column	Description
SOURCE_DESC_2	Long description for code

The naming convention for mappings designed for codes lookup is `SDE_[SOURCE]_CodeDimension_[CATEGORY]`. The figure below shows an example of a code mapping in PowerCenter Mapping Designer.

**Figure 6–12 Example of a code mapping in PowerCenter Mapping Designer**

### 6.10.3 Codes Mapplets

There are several mapplets that support the codes mappings in preparation for the source-independent load mapping. They are as follows:

- Source Adapter mapplets.** The Source Adapter mapplet connects the source-specific input attributes of CODES and the attributes from control or warehouse tables to the expression transform for mapping them. The naming convention for the Source Adapter codes mapplet is `MPLT_SA_CODES`.
- Business Component mapplets.** The Business Component mapplet makes the source system attributes of `CODES_CUST_CLASS` available to the extract mapping. The naming convention for the Business Component codes mapplet is `MPLT_BC_CODES_[CATEGORY]`.
- ADI Mapplet.** The Analytic Data Interface (load mapping) mapplet is source system independent and resolves the codes for the target table. The naming convention for the load mapping codes mapplet is `MPLT_ADI_CODES`.

The load mapping integrates multiple source system codes by designating one source system instance as a master in a mapping. All other source system codes are then mapped to the master. When the load mapping encounters a code that requires definition, it references the load control lookup table to match the source system code to an Oracle Business Analytics Warehouse source-independent code, which retains all the source system codes' original functionality.

The following columns are used to designate a source system instance as the master source system:

- MASTER\_ID.** Code for the source system designated as master.
- DATASOURCE\_NUM\_ID.** Unique identifier for the source system.

### 6.10.4 Configuring Extension Column Code Description Lookups

You can configure dimension and fact load sessions to perform specific lookups by editing the category of the data to be extracted from the `W_CODES` table and loading the code information into a target table. If the code and code name do not exist in the

W\_CODES table, then you must add them to the table. To configure the lookup, create a session override; do not modify the load mapping in the load mapping.

#### To configure sessions for lookups

1. In PowerCenter Workflow Manager, open the applicable source system configuration folder.
2. Open the Edit Tasks box.
3. In the Transformations tab, edit the SQL statement for the lookup.

For example, you may wish to edit the following lookup:

```
MPLT_ADI_SUPPLIERS.LKP_SPLR_ATTR1
```

4. Edit the SQL statement to use the desired code category.
5. Edit the SQL statement from 'GENERIC' to the category you wish to use for the lookup.

## 6.11 About Resolving Dimension Keys

By default, dimension key resolution is performed by the Oracle Business Analytics Warehouse in the load mapping. The load mapping uses prepackaged, reusable lookup transformations to provide prepackaged dimension key resolution. This section describes how dimension keys are looked up and resolved.

There are two commonly used methods for resolving dimension keys. The first method, which is the primary method used, is to perform a lookup for the dimension key. The second method is to supply the dimension key directly into the fact load mapping.

### 6.11.1 Resolving the Dimension Key Using Lookup

If the dimension key is not provided to the load mapping through database joins, the load mapping performs the lookup in the dimension table. The load mapping does this using prepackaged lookup transformations.

The load mapping uses the integration ID, the DATASOURCE\_NUM\_ID and Lookup date in looking up the dimension key. All these columns are necessary for the load mapping to return the dimension key. The ports are described in [Table 6–5](#).

**Table 6–5 Columns Used in the load mapping Dimension Key Lookup**

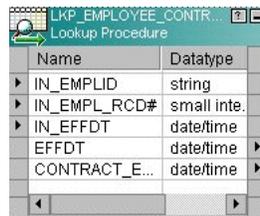
Port	Description
INTEGRATION ID	Uniquely identifies the dimension entity within its source system. Formed from the transaction in the Source Adapter of the fact table.
DATASOURCE_NUM_ID	Unique identifier of the source system instance.
Lookup Date	The primary date of the transaction; for example, receipt date, sales date, and so on.

In the figure below, the Supplier Products Key Lookup transformation illustrates the three input columns needed for the load mapping lookup—the INTEGRATION ID, DATASOURCE\_NUM\_ID, and Date (lookup date). The transformation then outputs the Supplier Product key (the dimension key) to the data warehouse table W\_SUPPLIER\_PRODUCT\_D.

If Type II slowly changing dimensions are enabled, the load mapping uses the unique effective dates for each update of the dimension records. When a dimension key is looked up, it uses the fact's primary date to resolve the appropriate dimension key.

The effective date range gives the effective period for the dimension record. The same entity can have multiple records in the dimension table with different effective periods due to Type II slowly changing dimensions. This effective date range is used to exactly identify a record in its dimension, representing the information in a historically accurate manner. In the lookup for Employee Contract Data shown in the figure below, you can see the effective dates used to provide the effective period of employee contracts.

**Figure 6–13** *Lookup for Employee Contract Data*



Name	Datatype
IN_EMPLID	string
IN_EMPL_RCD#	small inte.
IN_EFFDT	date/time
EFFDT	date/time
CONTRACT_E...	date/time

## 6.12 About Domain Values

The Oracle Business Analytics Warehouse foundation comprises a data model that accommodates data from disparate source systems. Data is sourced from operational systems and systematically molded into a source-independent format. After the data is source independent, it can then be used to create key metrics for analytic reporting, so that metric calculations are not source dependent. This clear separation allows you to swap source systems or integrate additional source systems without having to reconfigure the metric calculations to accommodate each source system's requirements.

One method for transforming source data into a source-independent format is to convert the source-supplied values to domain values. Domain values are a set of distinct values used to calculate prepackaged metrics. These values are provided by the Oracle Business Analytics Warehouse to allow you to create metric calculations independent of source system values.

### 6.12.1 About the Domain Value Conversion Process

To best understand the domain value conversion process, consider an example of two source systems—Source System A and Source System B. Each source system stores two types of employee events—hire and rehire. Source system A uses H to denote a hire event and R to denote a rehire event, whereas source system B uses 1 to denote a hire event and 2 to denote a rehire event. When the Oracle Business Analytics Warehouse extracts data from both systems, it ports those source values through the extract mapping until the data reaches the `W_EVENT_GRP_CODE` column in the `TF_EVENT_TYPES` Staging table.

The load mapping then ports the extracted source values (H and R from source system A, and 1 and 2 from source system B) into the Source Adapter mapplet. Within the Source Adapter, source values are translated into domain values (HIR and REH) based on a set of rules that are particular to your business practices.

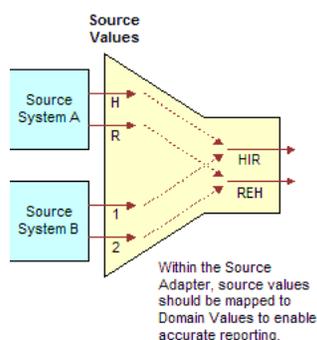
### 6.12.1.1 Preparing to Define the Rules

You must define the rules so that the Source Adapter knows how to map your specific source values to the given set of domain values. Before you set up the rules you must:

1. Analyze all of your source values and how they map to the prepackaged domain values. You may find that you need to create additional domain values for particular columns. The result of this preparation work is a list of each source value and how it is mapped to a domain value.
2. Implement this logic in the applicable Source Adapter mapplet. To set up the logic, modify the Expression transformation in the Source Adapter mapplet for each affected column. For information on setting up the rules for domain values, see [Section 6.14, "Configuring the Domain Value Set Using PowerCenter Designer"](#).

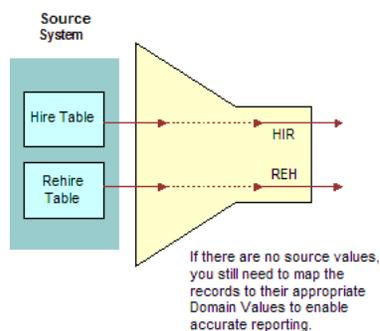
Figure 6–14 illustrates how the source values are converted to the domain values—HIR and REH.

**Figure 6–14 Source Values Translated to Domain Values**

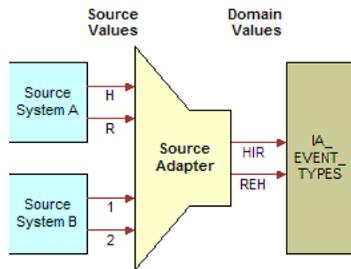


The figure below illustrates a different situation where the records may not contain a source value that flags the record as Hire or Rehire. In this case, the source system stores hires in one table and rehires in another table. To make this work, one possible solution is to modify the extract mappings to populate the `W_EVENT_GRP_CODE` column with HIR or REH. If the field is populated in the extract mapping, you can then carry those same values through the Source Adapter mapplet.

**Figure 6–15 Source Values in Different Tables Translated to Domain Values**



After the Source Adapter mapplet converts the source-specific values to domain values, the domain values are inserted into an Oracle Business Analytics Warehouse table. In this example, the HIR and REH values populate the `W_EVENT_TYPES` table, as illustrated in the figure below.

**Figure 6–16 HIR and REH Values Populating the W\_EVENT\_TYPES table**

## 6.12.2 About the Importance of Domain Values

Values in the W\_EVENT\_TYPES table are used to create metrics in the front end. Some metrics are defined using domain values. For example, seven metrics use the HIR and REH event group code in their calculation. The following are the seven metrics, along with their descriptions and calculations:

### 6.12.2.1 Hire Count

This metric counts all hires for a specified period. The calculation is:

```
SUM(CASE WHEN (CMMNEVTP.W_EVENT_GRP_CODE IN ('HIR', 'REH')) THEN EVNT.EVENT_CNT
ELSE 0 END)
```

### 6.12.2.2 Rehires Ratio

This metric determines the ratio of rehires to all employees hired during a specified period. The calculation is:

```
CASE WHEN SUM(CASE WHEN CMMNEVTP.W_EVENT_GRP_CODE IN ('REH', 'HIR') THEN
EVNT.EVENT_CNT ELSE 0 END) = 0 THEN 0 ELSE SUM(CASE WHEN CMMNEVTP.W_EVENT_GRP_CODE
IN ('REH') THEN EVNT.EVENT_CNT ELSE 0 END)/SUM(CASE WHEN CMMNEVTP.W_EVENT_GRP_CODE
IN ('REH', 'HIR') THEN EVNT.EVENT_CNT ELSE 0 END) END
```

### 6.12.2.3 New Hire Count

This metric counts the head count hired for regular full-time positions. The calculation is:

```
SUM(CASE WHEN CMMNEMPT.FULL_TIME_FLAG = 'Y' AND CMMNEMPT.EMP_CAT_CODE = 'R' AND
(CMMNEVTP.W_EVENT_GRP_CODE = 'HIR' OR CMMNEVTP.W_EVENT_GRP_CODE = 'REH') AND
EVNT.EVENT_DK >= (CMMNDATE.DATE_KEY - 365) AND EVNT.EVENT_DK <= CMMNDATE.DATE_KEY
THEN EVNT.EVENT_CNT ELSE 0 END)
```

### 6.12.2.4 Newly Separated Veterans - New Hires

This metric counts the regular full-time and part-time employees who belong to this category of veterans and were hired during the previous 12 months. The calculation is:

```
SUM(CASE WHEN CMMNEMPD.VETERAN_STAT_CODE = '4' AND CMMNEMPT.EMP_CAT_CODE = 'R' AND
(CMMNEVTP.W_EVENT_GRP_CODE = 'HIR' OR CMMNEVTP.W_EVENT_GRP_CODE = 'REH') AND
EVNT.EVENT_DK >= (CMMNDATE.DATE_KEY - 365) AND EVNT.EVENT_DK <= CMMNDATE.DATE_KEY
THEN EVNT.EVENT_CNT ELSE 0 END)
```

### 6.12.2.5 Other Protected Veterans - New Hires

This metric counts regular full-time and part-time employees who belong to this category of veterans. The calculation is:

```
SUM(CASE WHEN CMMNEMPD.VETERAN_STAT_CODE = '3' AND CMMNEMPT.EMP_CAT_CODE = 'R' AND
```

```
(CMMNEVTP.W_EVENT_GRP_CODE = 'HIR' OR CMMNEVTP.W_EVENT_GRP_CODE = 'REH') AND
EVNT.EVENT_DK >= (CMMNDATE.DATE_KEY - 365) AND EVNT.EVENT_DK <= CMMNDATE.DATE_KEY
THEN EVNT.EVENT_CNT ELSE 0 END)
```

### 6.12.2.6 Special Disabled Veteran Head count - New Hires

This metric counts regular full-time and part-time employees who belong to this category of veterans and were hired during the previous 12 months. The calculation is:

```
SUM(CASE WHEN CMMNEMPD.VETERAN_STAT_CODE = '1' AND CMMNEMPT.EMP_CAT_CODE = 'R' AND
(CMMNEVTP.W_EVENT_GRP_CODE = 'HIR' OR CMMNEVTP.W_EVENT_GRP_CODE = 'REH') AND
EVNT.EVENT_DK >= (CMMNDATE.DATE_KEY - 365) AND EVNT.EVENT_DK <= CMMNDATE.DATE_KEY
THEN EVNT.EVENT_CNT ELSE 0 END)
```

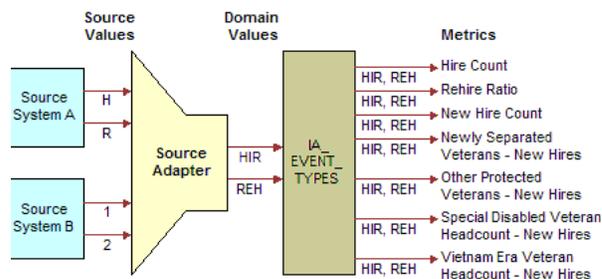
### 6.12.2.7 Vietnam Era Veteran Head count - New Hires

This metric counts regular full-time and part-time employees who belong to this category of veterans and were hired during the previous 12 months. The calculation is:

```
SUM(CASE WHEN CMMNEMPD.VETERAN_STAT_CODE = '2' AND CMMNEMPT.EMP_CAT_CODE = 'R' AND
(CMMNEVTP.W_EVENT_GRP_CODE = 'HIR' OR CMMNEVTP.W_EVENT_GRP_CODE = 'REH') AND
EVNT.EVENT_DK >= (CMMNDATE.DATE_KEY - 365) AND EVNT.EVENT_DK <= CMMNDATE.DATE_KEY
THEN EVNT.EVENT_CNT ELSE 0 END)
```

Each of these metric calculations is based on the domain values HIR and REH. All records whose source values are converted to one of these domain values are included in the metric calculations, as shown in the figure below.

**Figure 6–17 Metric Values From HIR and REH Domain Values**



## 6.12.3 About Extending the Domain Value Set

The Oracle Business Analytics Warehouse is also extensible in that you can create additional domain values for those columns that do not fit into the existing domain value definitions. However, before you modify the domain value set for a particular column, you first perform impact analysis on existing metrics. For example, the Oracle Business Analytics Warehouse prepackages the following two events:

- **New Hire.** This event occurs when a new person is hired.
- **New Position.** This event occurs when a position is created, but an existing employee may be hired internally.

If you have an event that represents both a New Hire and a New Position, you may have to create a third event that depicts both. If you create this new event type domain value, you need to include it in the applicable metric definitions so as to account for all hires and positions.

## 6.13 Configuring the Domain Value Set with CSV Worksheet Files

Domain values are a set of distinct values used to calculate prepackaged metrics. These values are provided by Oracle Business Analytics Warehouse to allow you to create metric calculations independent of source system values. Oracle Business Analytics Warehouse provides CSV worksheet files to map source system values to domain values.

You can add to these worksheet files if you need extra source system values and map them to domain values. You can also modify the worksheet files if you need to customize the domain values. You can use an existing domain value if you want to change the preconfigured metrics. Otherwise you can create a new domain value and create new metrics based on this domain value.

The source system values that are not mapped to a domain values in the CSV worksheet files have a question mark (?) as the domain value in the Oracle Business Analytics Warehouse. These values do not affect the domain values metrics.

If there are no worksheet files to map the source system values to the domain values, you need to modify the domain values using PowerCenter Designer. For more information on configuring domain values using PowerCenter Designer, see [Section 6.14, "Configuring the Domain Value Set Using PowerCenter Designer"](#).

For a list of CSV worksheet files and their domain values for your application, see your application configuration chapter.

### To map source values to domain values using CSV worksheet files

1. Identify all the Oracle Business Analytics Warehouse table columns that use domain values.

For a list of columns that use domain values, see the Siebel Customer-Centric Enterprise Warehouse Data Model Reference.

2. List all of your source values that qualify for conversion to one of the domain values.
3. Map each source value to a domain value.

If any of your source system values do not map to a prepackaged domain value, and you may modify the list of domain values, then create a list of new domain values and map your orphaned source system values to your newly created domain values.

You cannot modify all domain value sets. Also, you must check which metrics are affected by the modified domain value set. For more information, see the Siebel Customer-Centric Enterprise Warehouse Data Model Reference.

4. Open the CSV worksheet file in the `... \Informatica \SrcFiles` folder.
5. Edit the file to map your source values to the existing domain values.

Alternately, if you want to add additional domain values, add them in this worksheet file.

6. Save and close the worksheet file.

## 6.14 Configuring the Domain Value Set Using PowerCenter Designer

If there are no worksheet files to map the source system values to the domain values, you need to modify the values using PowerCenter Designer. For more information on configuring the domain value set with CSV worksheet files, see [Section 6.13, "Configuring the Domain Value Set with CSV Worksheet Files"](#).

Configuring the domain value set for a particular column, using PowerCenter Designer, entails one or both of the following activities:

- Mapping source-specific values to domain values
- Adding more domain values to the prepackaged set of values

Regardless of which activity you choose, the configuration occurs in the Expression transformation of the applicable Source Adapter mapplet. The following procedure shows how to configure the Expression transformation to change the domain values.

#### To map source values to domain values using PowerCenter Designer

1. Identify all the Oracle Business Analytics Warehouse table columns that use domain values.

For a list of columns that use domain values, see the Siebel Customer-Centric Enterprise Warehouse Data Model Reference.

2. List all of your source values that qualify for conversion to one of the domain values.
3. Map each source value to a domain value.

If any of your source system values do not map to a prepackaged domain value, and you may modify the list of domain values, then create a list of new domain values and map your orphaned source system values to your newly created domain values.

You cannot modify all domain value sets. Also, you must check which metrics are affected by the modified domain value set. For more information, see the Siebel Customer-Centric Enterprise Warehouse Data Model Reference.

4. In PowerCenter Designer, open the applicable Source Adapter mapplet.
5. Open the Expression transformation.
6. Locate the applicable port's expression so that you can modify it.
7. Edit the port's expression to map your source values to the existing domain values.

Alternately, if you want to add additional domain values, add them in this same expression.

8. Save and validate your changes to the repository.

## 6.15 Configuring Conformed Dimensions

This section provides procedures on configuring objects that apply to more than one application and contains the following topics:

- [Section 6.15.1, "Configuring Conformed Dimensions for Universal Source"](#)

### 6.15.1 Configuring Conformed Dimensions for Universal Source

This section provides configuration procedures for modifying dimensions that are loaded using a universal business adapter.

#### 6.15.1.1 Product Effective Dates in the Products Dimension

The Oracle Business Analytics Warehouse stores product Effective To (SRC\_EFF\_TO\_DT) and From (SRC\_EFF\_FROM\_DT) dates in the Products dimension table, w\_

PRODUCTS. In addition, the Products dimension stores a Support Withdrawn Date column, `SPRT_WITHDRAWN_DT`.

By default, the Support Withdrawn Date takes precedence over the product Effective To Date. This prioritization means that if you supply a value for the Support Withdrawn Date column in your flat file upload, the Oracle Business Analytics Warehouse uses that value as the product Effective To value as well, overwriting anything in the `SRC_EFF_TO_DT` column. You can change this default behavior by modifying the Products Expression in the Universal Source Products Extract mapping.

**To modify the product Effective To Date logic for a flat file extract**

1. In PowerCenter Designer, open the Configuration for Universal Source folder.
2. In the `SDE_Universal_ProductDimension` mapping, open the `W_PRODUCT_D` expression.
3. Edit the logic for the `SRC_EFF_TO_DT_OUT` port.
4. Validate and save your changes.

---

---

# Integrated Security for Oracle BI Applications

This chapter describes the security features in Oracle Business Intelligence Applications, and includes the following main topics:

- [Section 7.1, "Types of Security in Oracle Business Intelligence Applications"](#)
- [Section 7.2, "Integrating Data Security with Oracle E-Business Suite"](#)
- [Section 7.3, "Integrating Data Security with Oracle's PeopleSoft Enterprise Applications"](#)
- [Section 7.4, "Metadata Object-Level Security in Oracle BI Applications"](#)
- [Section 7.5, "Integrating Data Security with the Siebel CRM Application"](#)
- [Section 7.6, "Data-Level Security In Oracle BI Applications"](#)

## 7.1 Types of Security in Oracle Business Intelligence Applications

Oracle Business Intelligence Applications integrate tightly with the security model of the operational source system to allow the right content to be shown to the right user. Oracle BI Applications are embedded with a host of securing options that an administrator can leverage to authenticate and show critical business data to the right people.

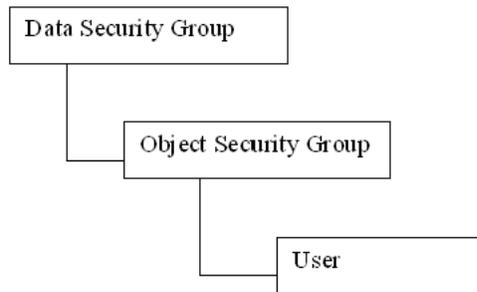
Security in Oracle BI Applications can be classified broadly into three different categories:

- User Security (Authentication of users)
- Object Security
- Data Security

User security refers to authentication and confirmation of the identity of a user based on the credentials provided. Object security controls the visibility to business logical objects (like Subject Areas, Presentation Tables, etc. defined in the metadata repository) and web objects (like Dashboard, Answers, etc. defined in the Presentation Catalog) based on a user's role. An example of a user's role is "Financial Analyst". Data security controls the visibility of data (content rendered in Subject Areas, Dashboards, Answers) based on the user's association to data in the OLTP system. An example is "Business Unit based Security".

Data and Object security are implemented in Oracle BI Applications using Security Groups. These security groups are defined using the Security Manager in Oracle Business Intelligence Administration Tool (Menu: Manage > Security). The standard

structure of security groups and users in Oracle BI Applications has the hierarchy Data Security Group, then Object Security Group, then User, as shown in the following figure.



The detail wirings between security groups and users can vary, and needs to be decided on actual implementation.

Oracle BI Applications aligns a user's security profile in Oracle Business Intelligence consistent with their security profiles in the source applications. The alignment is done by one of the following:

1. Creating Security groups in the Oracle Business Intelligence application repository with the same names as some existing Responsibilities or groups in the source applications. These security groups are added as members to some Oracle Business Intelligence specific security groups (for Position based or Org based security), and the users will inherit this membership based on their own responsibilities or roles in the OLTP application.
2. Adding new Oracle Business Intelligence specific Responsibilities (Oracle EBS and Siebel CRM Applications) or Roles (PeopleSoft application) in the source applications, making sure their names match the object security groups in Oracle BI Applications, and assigning OLTP users to these new groups. The users will then inherit the security group membership the same way as described in the first scenario.

**Note:** Users should always be created in the operational application databases or directory services such as LDAP for the Security mechanism to work, never in the Oracle Business Intelligence repository. If the users are created in the RPD, the security mechanism does not work. For details on integration with Oracle E-Business Suite and PeopleSoft applications, please refer to [Section 7.2, "Integrating Data Security with Oracle E-Business Suite"](#) and [Section 7.3, "Integrating Data Security with Oracle's PeopleSoft Enterprise Applications"](#).

### 7.1.1 Checking Oracle Business Intelligence Applications User Responsibilities

There are two ways for an Administrator to check a user's responsibility:

- In the Siebel or Oracle E-Business Suite operational application, go to the Responsibilities view.
- In the PeopleSoft application, go to the Roles View to check a user's roles.
- In the Oracle Business Intelligence application, click on Settings/My Account link. The Presentation Services group membership for the user is shown near the bottom of the Web page. These are the Presentation Services groups, defined in the Presentation Services Catalog only, and which are usually used to control the ability to perform actions (privileges). If a Presentation Services group has the same name as an Oracle BI Server security group, and the user is a member of the

latter, than he will become automatically a member of the corresponding Presentation group.

## 7.1.2 Registering a New User Responsibility in Oracle Business Intelligence

When you add a new responsibility to a user in Oracle BI Web, the change is not immediately reflected in the Oracle BI environment. In order to register the new user responsibility, both the Administrator and the user must perform a task:

- The Oracle BI Administrator must reload the Oracle BI Server metadata through Oracle BI Presentation Services.
- Then the user must log out from the Oracle Business Intelligence application, or from Siebel or Oracle E-Business Suite operational application in case the user is looking at the Oracle Business Intelligence dashboards using an embedded application, and then log in again.

## 7.1.3 Default Security Settings in Oracle Business Intelligence

The User Administrator and the Group Administrators are a special user and group respectively, that do not have any restrictions and do not go through the Siebel or Oracle E-Business Suite database. The User SADMIN (password SADMIN) is also a special user, similar to Administrator.

The Administrator group is set up as a member of the Supergroup Administrators, so members of this group have no restrictions.

**Note:** Be sure to change the default password before migrating to production.

The Administrator group is set up as a member of the supergroup Administrators, so members of this group have no restrictions.

## 7.2 Integrating Data Security with Oracle E-Business Suite

This section explains how security in Oracle BI Applications is deployed with Oracle E-Business Suite (EBS). Read this section if you want to understand how security is configured out-of-the-box so that you can change the way that security is implemented if required. This section contains the following sections:

- [Section 7.2.1, "Authorization"](#)
- [Section 7.2.2, "Operating Unit-Based Security With Oracle EBS"](#)
- [Section 7.2.3, "Inventory Organization-Based Security With Oracle EBS"](#)
- [Section 7.2.4, "Company Org-based Security Against Oracle EBS"](#)
- [Section 7.2.5, "Ledger Security Against Oracle EBS"](#)
- [Section 7.2.6, "Business Group Org-based Security Against Oracle EBS"](#)
- [Section 7.2.7, "Employee Based Security with Oracle EBS"](#)

### 7.2.1 Authorization

The authorization process of Oracle BI Applications fetches a user's responsibilities from source Oracle EBS applications, matches them with all Oracle BI Applications security groups, and determine the user's applicable object security during a user's session. The initialization block "Authorization" is used to fetch roles and assign the result set to a special session variable called "GROUP". The initialization block SQL is:

```

SELECT DISTINCT 'GROUP', RESPONSIBILITY_NAME FROM
FND_USER ,FND_USER_RESP_GROUPS, FND_RESPONSIBILITY_VL
WHERE
FND_USER.user_id=FND_USER_RESP_GROUPS.user_id
AND FND_USER_RESP_GROUPS.RESPONSIBILITY_ID = FND_RESPONSIBILITY_
VL.RESPONSIBILITY_ID
AND FND_USER_RESP_GROUPS.RESPONSIBILITY_APPLICATION_ID = FND_
RESPONSIBILITY_VL.APPLICATION_ID AND
FND_USER_RESP_GROUPS.START_DATE < SYSDATE AND
(CASE WHEN FND_USER_RESP_GROUPS.END_DATE IS NULL THEN SYSDATE
ELSE TO_DATE(FND_USER_RESP_GROUPS.end_Date) END) >= SYSDATE
AND FND_USER.user_id = = (SELECT USER_ID FROM FND_USER WHERE
USER_NAME = ':USER')

```

This release of Oracle BI Applications supports the following types of data security integrating with Oracle EBS:

- Operating Unit-Based security against Oracle EBS (for more information, see [Section 7.2.2, "Operating Unit-Based Security With Oracle EBS"](#)).
- Inventory Organization-based security against Oracle EBS (for more information, see [Section 7.2.3, "Inventory Organization-Based Security With Oracle EBS"](#)).
- Company Org-based security against Oracle EBS (for more information, see [Section 7.2.4, "Company Org-based Security Against Oracle EBS"](#)).
- Ledger based security (for more information, see [Section 7.2.4, "Company Org-based Security Against Oracle EBS"](#)).
- Business Group Org-based security against Oracle EBS (for more information, see [Section 7.2.6, "Business Group Org-based Security Against Oracle EBS"](#)).
- Primary Employee/Position Hierarchy-based Security against Oracle EBS - HRMS application (for more information see [Section 7.5.1, "Primary Position based Security"](#)).

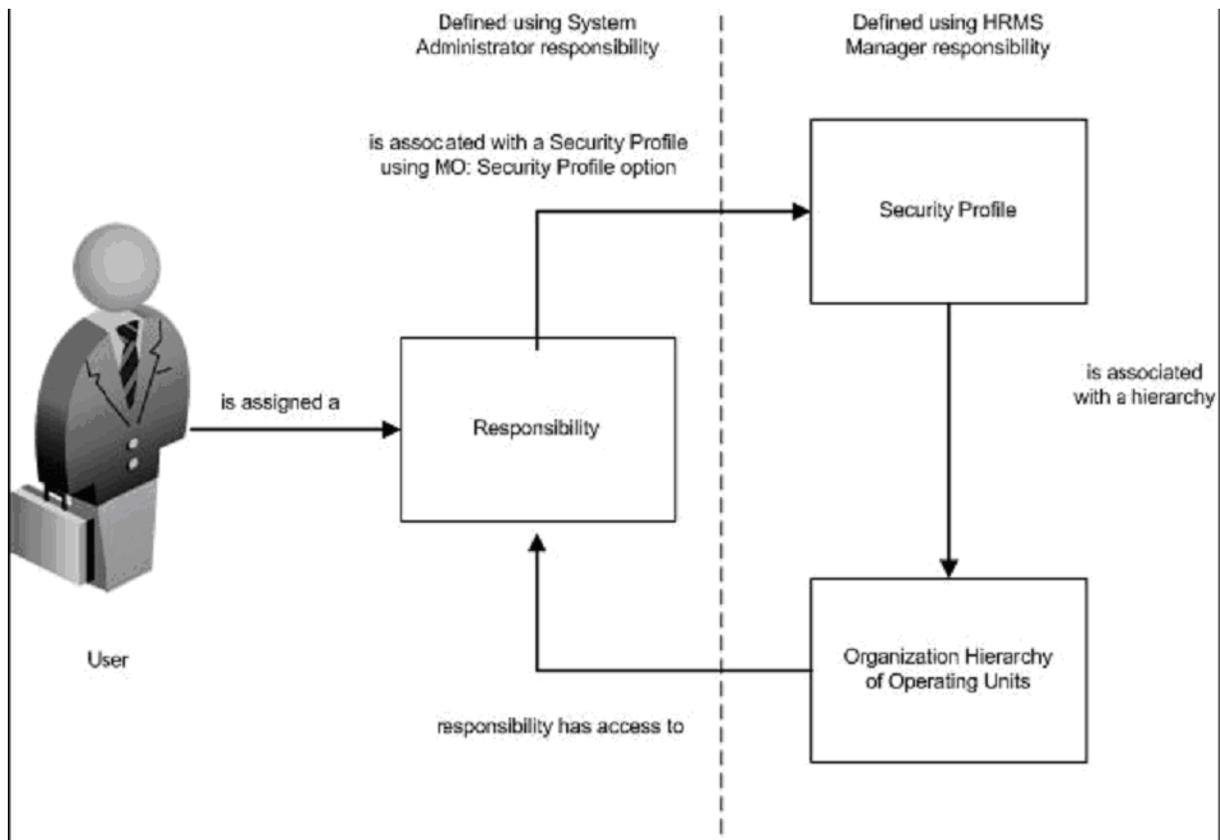
Oracle BI Applications Version 7.9.4 does not support GL flex field or HR security profile-based security against EBS.

## 7.2.2 Operating Unit-Based Security With Oracle EBS

This section covers Operating Unit-Based Security With Oracle EBS.

### 7.2.2.1 About Operating Unit-Based Security With Oracle EBS

Operating Units are secured by attaching a security profile to a user ID or responsibility. In turn, a security profile is associated with an organization hierarchy, which also has access to the user ID or responsibility (see figure below). The user ID or responsibility is defined using the System Administrator responsibility. The security profile and organization hierarchy are defined using the HRMS manager responsibility.

**Figure 7-1 Operating Unit-Based Security Against Oracle EBS**

Operating Unit Dimension is decided by looking at the profiles set at User/Responsibility/Application/Site Levels (in order). In other words, if a value is set in the profile at User Level and at Site Level, the value set at user level takes precedence.

The profiles used are:

- MO: Security Profile.

Depending on the Security Profile assigned, the Operating Units that are associated with that Security Profile become accessible to the User.

- MO: Operating Unit.

If (1) is set to NULL (not set at any of the 4 levels), the profile MO: Operating Unit is used. (One MO: Operating Unit can be assigned to only one Operating Unit, whereas MO: Security Profile contains a Security Profile, which can be assigned to multiple Operating Units.)

Operating Unit Dimension is essentially a User secured dimension, which does the following:

- It looks at the context of the user accessing a report.
- It determines which Operating Units the user can access.
- It displays the content.

From Oracle EBS R12 onwards, one Operating Unit is not fixed to just one responsibility. Users can still login as "Purchasing, Vision Operations (USA)" and

create a PO for "Vision Germany", as OU is now shown as a LOV in the PO screen and the LOV values that show up depend on the 2 profiles.

### 7.2.2.2 Implementation Steps For Operating Unit-Based Security With Oracle EBS

This section covers Operating Unit-Based Security With Oracle EBS.

The sequence for Operating Unit-Based Security With Oracle EBS is described below:

- When a user logs in to Oracle BI Applications, the session variable below is set automatically.

```
USER (System variable)
```

- The "EBS Single Sign-on Integration" session variable is initialized in the "EBS Single Sign-on Integration" initialization block:

```
EBS_SSO_INTEGRATION_MODE
```

This session can be initialized with two possible values, "Integrated" or "Not Integrated", to indicate whether Oracle BI Applications is integrated with EBS SSO or not.

- The "EBS Security Context" initialization block then populates these session variables:

```
OLTP_EBS_RESP_ID
```

The session variable is initialized with the responsibility of the user's session in Oracle EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

```
OLTP_EBS_RESP_APPL_ID
```

The session variable is initialized with the responsibility application of the user session in EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

- The Oracle Business Intelligence Server will get the set of books corresponding to the USER from FND\_USER\_RESP\_GROUPS. The following session variable is set automatically:

```
OU_ORG (Row-wise variable)
```

The initialization block "Operating Unit Org", which sets the value for this variable, is shown below.

#### Initialization block -- "Operating Unit Org"

The initialization block "Operating Unit Org" sets the value for variable OU\_ORG using the following SQL:

```
SELECT DISTINCT 'OU_ORG', TO_CHAR(PER_ORGANIZATION_
LIST.ORGANIZATION_ID)
FROM PER_ORGANIZATION_LIST,
(SELECT FND_PROFILE.VALUE_SPECIFIC('XLA_MO_SECURITY_PROFILE_
LEVEL', USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_
APPLICATION_ID) PROFILE_ID
FROM (SELECT USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_
APPLICATION_ID
FROM FND_USER_RESP_GROUPS
WHERE START_DATE < SYSDATE
```

```

AND (CASE WHEN END_DATE IS NULL THEN SYSDATE ELSE TO_
DATE(END_DATE) END) >= SYSDATE

AND USER_ID = (SELECT USER_ID FROM FND_USER WHERE USER_NAME =
':USER')

AND RESPONSIBILITY_ID = (CASE WHEN VALUEOF(NQ_SESSION.EBS_
SSO_INTEGRATION_MODE) = 'Integrated' THEN
VALUEOF(NQ_SESSION.OLTP_EBS_RESP_ID) ELSE RESPONSIBILITY_ID
END)

AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN VALUEOF(NQ_
SESSION.EBS_SSO_INTEGRATION_MODE) = 'Integrated' THEN
VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE
RESPONSIBILITY_APPLICATION_ID END))

WHERE PER_ORGANIZATION_LIST.SECURITY_PROFILE_ID = PROFILE_ID

UNION

SELECT DISTINCT 'OU_ORG', FND_PROFILE.VALUE_SPECIFIC('ORG_
ID', USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_
ID) ORGANIZATION_ID

FROM (SELECT USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_
APPLICATION_ID

FROM FND_USER_RESP_GROUPS

WHERE START_DATE < SYSDATE

AND (CASE WHEN END_DATE IS NULL THEN SYSDATE ELSE TO_
DATE(END_DATE) END) >= SYSDATE

AND USER_ID = (SELECT USER_ID FROM FND_USER WHERE USER_NAME =
':USER')

AND RESPONSIBILITY_ID = (CASE WHEN VALUEOF(NQ_SESSION.EBS_
SSO_INTEGRATION_MODE) = 'Integrated' THEN VALUEOF(NQ_
SESSION.OLTP_EBS_RESP_ID) ELSE RESPONSIBILITY_ID END)

AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN VALUEOF(NQ_
SESSION.EBS_SSO_INTEGRATION_MODE) = 'Integrated' THEN
VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE
RESPONSIBILITY_APPLICATION_ID END))

```

## 7.2.3 Inventory Organization-Based Security With Oracle EBS

This section explains Inventory Organization-Based Security With Oracle EBS.

### 7.2.3.1 About Inventory Organization-Based Security With Oracle EBS

Ideally, Inventory organization security is applied on the basis of the currently logged-in responsibility, rather than a current user.

In EBS however, if an Inventory Org V1 has been associated with Responsibility R1 and R2, then V1 is accessible only to those 2 responsibilities. If another Inventory Org V2 has not been defined at all in that screen, then all responsibilities have access to V2. Each record entry into this form inserts a row into ORG\_ACCESS table. This is the reason for a UNION based query of BIS\_ORGANIZATIONS\_V, and ideally, the Inventory Org secured view definition should have been:

```
SELECT 'INV_ORG', BIS_ORGANIZATIONS_V.ID FROM BIS_ORGANIZATIONS_V WHERE
```

```
RESPONSIBILITY_ID = :RESPONSIBILITY_ID
```

This would give us a list of Inventory Organizations that are accessible to the "logged in" responsibility. In case security is set as user level in BI EE, the BI query is secure because it simply checks all the responsibilities accessible to the user, and then queries the ORG\_ACCESS to check all the Inventory Organizations accessible to those responsibilities and all those Inventory Organizations that are not specifically allocated to any responsibility (in other words, accessible to everyone). This combined list of Inventory Orgs would be applied in the query. Therefore, it is possible that certain inventory organizations have been explicitly granted to certain responsibilities (using the ORG ACCESS form), and if the responsibility is not assigned to the logged-in user, then those inventory orgs would not be displayed to the user.

### 7.2.3.2 Implementation Steps For Inventory Organization-Based Security With Oracle EBS

The sequence for Inventory-Based Security With Oracle EBS is described below:

- When a user logs in to Oracle BI Applications, the session variable below is set automatically.

```
USER (System variable)
```

- The "EBS Single Sign-on Integration" session variable is initialized in the "EBS Single Sign-on Integration" initialization block:

```
EBS_SSO_INTEGRATION_MODE
```

This session can be initialized with two possible values, "Integrated" or "Not Integrated", to indicate whether Oracle BI Applications is integrated with EBS SSO or not.

- The "EBS Security Context" initialization block then populates these session variables:

```
OLTP_EBS_RESP_ID
```

The session variable is initialized with the responsibility of the user session in Oracle EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

```
OLTP_EBS_RESP_APPL_ID
```

The session variable is initialized with the responsibility application of the user session in EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

- The Oracle Business Intelligence Server will get the set of books corresponding to the USER from FND\_USER\_RESP\_GROUPS. The following session variable is set automatically:

```
INV_ORG (Row-wise variable)
```

The initialization block "Inventory Organizations", which sets the value for this variable, is shown below.

#### Initialization block -- "Inventory Organizations"

The initialization block "Inventory Organizations" sets the value for variable INV\_ORG using the following SQL:

```
SELECT DISTINCT 'INV_ORG' , BIS_ORGANIZATIONS_V.ID
FROM FND_USER_RESP_GROUPS , BIS_ORGANIZATIONS_V
```

```

WHERE FND_USER_RESP_GROUPS.RESPONSIBILITY_ID = BIS_
ORGANIZATIONS_V.RESPONSIBILITY_ID

AND FND_USER_RESP_GROUPS.START_DATE < SYSDATE

AND (CASE WHEN FND_USER_RESP_GROUPS.END_DATE IS NULL THEN
SYSDATE ELSE

AND FND_USER_RESP_GROUPS.USER_ID = (SELECT USER_ID FROM FND_
USER WHERE USER_NAME = ':USER' )

AND RESPONSIBILITY_ID = (CASE WHEN VALUEOF(NQ_SESSION.EBS_
SSO_INTEGRATION_MODE) = 'Integrated' THEN

VALUEOF(NQ_SESSION.OLTP_EBS_RESP_ID) ELSE RESPONSIBILITY_ID
END)

AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN VALUEOF(NQ_
SESSION.EBS_SSO_INTEGRATION_MODE) =

'Integrated' THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID)
ELSE RESPONSIBILITY_APPLICATION_ID END)

```

## 7.2.4 Company Org-based Security Against Oracle EBS

Company Org-based security against EBS was introduced in 7.9.3. It is replaced by Ledger Security in 7.9.4.

This section covers Ledger Org-based security against Oracle EBS.

### 7.2.4.1 About Company Org-based Security Against Oracle EBS

In Oracle E-Business Suite, a set of book is essentially a reporting organization that uses a common chart of accounts, functional currency, and fiscal calendar. The company org-based security filters data based on the set of books associated to the logged in user.

### 7.2.4.2 Implementation Steps For Company Org-based Security Against Oracle EBS 11i

The sequence for Company Org-based security against Oracle EBS is described below:

- When a user logs in to Oracle BI Applications, the session variable below is set automatically.

```
USER (System variable)
```

- The "EBS Single Sign-on Integration" session variable is initialized in the "EBS Single Sign-on Integration" initialization block:

```
EBS_SSO_INTEGRATION_MODE
```

This session can be initialized with two possible values, "Integrated" or "Not Integrated", to indicate whether Oracle BI Applications is integrated with EBS SSO or not.

- The "EBS Security Context" initialization block then populates these session variables:

```
OLTP_EBS_RESP_ID
```

The session variable is initialized with the responsibility of the user session in Oracle EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

OLTP\_EBS\_RESP\_APPL\_ID

The session variable is initialized with the responsibility application of the user session in EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

- The Oracle Business Intelligence Server will get the set of books corresponding to the USER and OLTP\_EBS\_RESP\_ID from FND\_USER\_RESP\_GROUPS. The following session variable is set automatically:

COMPANY (Row-wise variable)

The initialization block "Companies", which sets the value for this variable, is shown below.

#### Initialization block -- "Companies"

The initialization block "Companies" sets the value for variable COMPANY using the following SQL:

```
SELECT DISTINCT 'COMPANY', FND_PROFILE.VALUE_SPECIFIC('GL_
SET_OF_BKS_ID', USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_
APPLICATION_ID)

FROM (SELECT USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_
APPLICATION_ID FROM FND_USER_RESP_GROUPS WHERE START_DATE
< SYSDATE AND (CASE WHEN END_DATE IS NULL THEN SYSDATE ELSE
TO_DATE(END_DATE) END) >= SYSDATE AND USER_ID IN (SELECT
USER_ID FROM FND_USER WHERE USER_NAME = ':USER'))

AND RESPONSIBILITY_ID = (CASE WHEN VALUEOF(NQ_SESSION.EBS_
SSO_INTEGRATION_MODE) = 'Integrated' THEN VALUEOF(NQ_
SESSION.OLTP_EBS_RESP_ID) ELSE RESPONSIBILITY_ID END)

AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN VALUEOF(NQ_
SESSION.EBS_SSO_INTEGRATION_MODE) = 'Integrated' THEN
VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE
RESPONSIBILITY_APPLICATION_ID END))
```

- The "Company Org-based Security" security group contains all the data access permission filters. During implementation, customers need to decide which users or groups should be associated to this security group.

## 7.2.5 Ledger Security Against Oracle EBS

Ledger-based security against Oracle EBS is introduced in Oracle BI Applications Version 7.9.4. It replaces the Company-based security to support the EBS GL set of books security model in E-Business Suite release 11i and the EBS Data Access Set model in E-Business Suite release 12.

In Oracle EBS Release 11i, a set of books is essentially a reporting entity that defines the reporting context including a chart of accounts, a functional currency, and an accounting calendar. A set of books can be assigned to a user, a responsibility, or to the site as the default for all responsibilities. Each user is associated with a single set of books when they logon to the application under a given responsibility in Oracle Applications. The Ledger-based security filters data based on the set of books associated to the logged in user.

In Oracle EBS Release 12, the set of books is replaced by the ledger. A ledger determines the currency, chart of accounts, accounting calendar, ledger processing options and subledger accounting method. The data access set assigned to the user's responsibility controls what ledgers the user can access. A user may be able to access

multiple ledgers from a responsibility. The Ledger-based security filters data based on the ledgers associated to the logged in user.

Source-specific steps for Oracle EBS:

- When a user logs in to Oracle Business Intelligence Enterprise Edition, the session variable below is set automatically.

```
USER (System variable)
```

- The "EBS Single Sign-on Integration" session variable is initialized in the "EBS Single Sign-on Integration" initialization block:

```
EBS_SSO_INTEGRATION_MODE
```

This session can be initialized with two possible values, "Integrated" or "Not Integrated", to indicate whether Oracle BI Applications is integrated with EBS SSO or not.

- The "EBS Security Context" initialization block then populates these session variables:

```
OLTP_EBS_RESP_ID
```

The session variable is initialized with the responsibility of the user session in Oracle EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

```
OLTP_EBS_RESP_APPL_ID
```

The session variable is initialized with the responsibility application of the user session in EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

- Then this session variable would be initialized in another init block, "Ledgers", which gets the ledgers (which is essentially the set of books in EBS) corresponding to the USER and OLTP\_EBS\_RESP\_ID and OLTP\_EBS\_RESP\_APPL\_ID, via table FND\_USER\_RESP\_GROUPS and procedure FND\_PROFILE.

Row-wise variable:

```
LEDGER (Row-wise variable)
```

- The Oracle BI server gets the set of books or ledgers corresponding to the USER and OLTP\_EBS\_RESP\_ID from the OLTP. The "Ledgers" initialization block then populates these session variables.

The Ledgers initialization block should be set according to the Oracle EBS release, as follows:

- If you are using EBS release 12 or after, please use the following SQL as the data source in the initialization block:

```
SELECT DISTINCT 'LEDGER', TO_CHAR(GAL.LEDGER_ID)
FROM GL_ACCESS_SET_LEDGERS GAL, (SELECT FND_PROFILE.VALUE_
SPECIFIC('GL_ACCESS_SET_ID',USER_ID, RESPONSIBILITY_ID,
RESPONSIBILITY_APPLICATION_ID) PROFILE_VALUE
FROM (SELECT USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_
APPLICATION_ID
FROM FND_USER_RESP_GROUPS
WHERE START_DATE < SYSDATE AND (CASE WHEN END_DATE IS NULL
THEN SYSDATE ELSE
```

```

TO_DATE(END_DATE) END) >= SYSDATE AND USER_ID = (CASE WHEN
'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE)' = 'Integrated'
THEN VALUEOF(NQ_SESSION.OLTP_EBS_USER_ID) ELSE (SELECT USER_
ID FROM FND_USER WHERE
USER_NAME = 'OPERATIONS') END) AND RESPONSIBILITY_ID = (CASE
WHEN
'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE)' = 'Integrated'
THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_ID) ELSE
RESPONSIBILITY_ID END)
AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN
'VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE)' = 'Integrated'
THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE
RESPONSIBILITY_APPLICATION_ID END)
))WHERE GAL.ACCESS_SET_ID = PROFILE_VALUE

```

The repository file shipped in Oracle BI Applications 7.9.4 release include the EBS release 12 version of the SQL in the “Ledger” initialization block.

- If you are running BI applications against Oracle EBS 11i, please use the following SQL as the data source in “Ledger” initialization block:

```

SELECT DISTINCT 'LEDGER', FND_PROFILE.VALUE_SPECIFIC('GL_SET_
OF_BKS_ID', USER_ID,
RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_ID)
FROM (SELECT USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_
APPLICATION_ID FROM
FND_USER_RESP_GROUPS
WHERE START_DATE < SYSDATE
AND (CASE WHEN END_DATE IS NULL THEN SYSDATE ELSE TO_
DATE(END_DATE) END) >= SYSDATE
AND USER_ID IN (CASE WHEN VALUEOF(NQ_SESSION.EBS_SSO_
INTEGRATION_MODE) = 'Integrated'
THEN VALUEOF(NQ_SESSION.OLTP_EBS_USER_ID) ELSE (SELECT USER_
ID FROM FND_USER WHERE USER_NAME = ':USER') END)
AND RESPONSIBILITY_ID = (CASE WHEN VALUEOF(NQ_SESSION.EBS_
SSO_INTEGRATION_MODE) = 'Integrated'
THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_ID) ELSE
RESPONSIBILITY_ID END)
AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN
VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE) = 'Integrated'
THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE
RESPONSIBILITY_APPLICATION_ID END))

```

## 7.2.6 Business Group Org-based Security Against Oracle EBS

This section explains Business Group Org-based security against Oracle EBS.

### 7.2.6.1 About Business Group Org-based Security Against Oracle EBS

A Business Group is the highest level in the organization structure is usually used to represent the entire enterprise or a major division. A business group can have several sets of books.

### 7.2.6.2 Implementation Steps For Business Group Org-based Security Against Oracle EBS

The sequence for Business Group Org-based security against Oracle EBS is described below:

- When a user logs in to Oracle BI Applications, the session variable below is set automatically.

`USER` (System variable)

- The "EBS Single Sign-on Integration" session variable is initialized in the "EBS Single Sign-on Integration" initialization block:

`EBS_SSO_INTEGRATION_MODE`

This session can be initialized with two possible values, "Integrated" or "Not Integrated", to indicate whether Oracle BI Applications is integrated with EBS SSO or not.

- The "EBS Security Context" initialization block then populates these session variables:

`OLTP_EBS_RESP_ID`

The session variable is initialized with the responsibility of the user session in Oracle EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

`OLTP_EBS_RESP_APPL_ID`

The session variable is initialized with the responsibility application of the user session in EBS if Oracle BI Applications is integrated with EBS; otherwise it is defaulted to a random value, which will be ignored.

- The Oracle Business Intelligence Server will get the set of books corresponding to the `USER` and `OLTP_EBS_RESP_ID` from `FND_USER_RESP_GROUPS`. The following session variable is set automatically:

`BUSINESS_GROUP` (Row-wise variable)

The initialization block "Business Groups", which sets the value for this variable, is shown below.

#### Initialization block -- "Business Groups"

The initialization block "Business Groups" sets value for variable `INV_ORG` using the following SQL:

```
SELECT DISTINCT 'BUSINESS_GROUP' ,
TO_CHAR(FND_PROFILE.VALUE_SPECIFIC('PER_BUSINESS_GROUP_ID' ,
USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_APPLICATION_ID))
FROM (SELECT USER_ID, RESPONSIBILITY_ID, RESPONSIBILITY_
APPLICATION_ID FROM FND_USER_RESP_GROUPS WHERE START_DATE <
SYSDATE AND (CASE WHEN END_DATE IS NULL THEN SYSDATE ELSE TO_
DATE(END_DATE) END) >= SYSDATE AND USER_ID = (SELECT USER_ID
FROM FND_USER WHERE USER_NAME = ':USER'))
```

```
AND RESPONSIBILITY_ID = (CASE WHEN VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE) = 'Integrated' THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_ID) ELSE RESPONSIBILITY_ID END)

AND RESPONSIBILITY_APPLICATION_ID = (CASE WHEN VALUEOF(NQ_SESSION.EBS_SSO_INTEGRATION_MODE) = 'Integrated' THEN VALUEOF(NQ_SESSION.OLTP_EBS_RESP_APPL_ID) ELSE RESPONSIBILITY_APPLICATION_ID END)
```

- The "Business Group Org-based Security" security group contains all the data access permission filters. During implementation, customers need to decide which users or groups should be associated to this security group.

## 7.2.7 Employee Based Security with Oracle EBS

Employee based security restricts data visibility of the records to the Owner of that record, and all employees he/she reports to in the company's Employee hierarchy. This security mechanism uses data from the data warehouse database, and shares the metadata components with other supported applications (Siebel CRM and PeopleSoft). Out of the box, this type of security supports only HR Analytics facts. For more information on how this security mechanism works, see [Section 7.5.2, "Primary Position Based Security for CRM Vertical applications"](#).

## 7.3 Integrating Data Security with Oracle's PeopleSoft Enterprise Applications

This section explains how security is implemented for Oracle's PeopleSoft Enterprise Applications in Oracle Business Intelligence Applications. Read this section if you want to understand how security is configured out-of-the-box so that you can change the way that security is implemented if required. This section contains the following sections:

- [Appendix 7.3.1, "Authorization"](#)
- [Appendix 7.3.2, "About Operating Unit-based Security Against PeopleSoft Financials"](#)
- [Appendix 7.3.3, "About Company Org-based Security Against PeopleSoft Financials or PeopleSoft HR"](#)
- [Appendix 7.3.4, "About Ledger-based Security against Oracle's PeopleSoft Enterprise Applications"](#)
- [Appendix 7.3.5, "About HR Org-based Security Against PeopleSoft HR"](#)
- [Appendix 7.3.6, "About Payables Org-based security Against PeopleSoft Financials"](#)
- [Appendix 7.3.7, "About Receivables Org-based Security Against PeopleSoft Financials"](#)
- [Appendix 7.3.8, "About SET ID-based Security Against PeopleSoft Financials"](#)
- [Appendix 7.3.9, "About Human Resource Personnel Data Analyst Security Against PeopleSoft HR"](#)

### 7.3.1 Authorization

The authorization process of Oracle BI Applications fetches a user's role from source PeopleSoft applications, matches the role with all Oracle BI Applications security

groups, and determine the user's applicable object security during a user's session. The initialization block "Authorization" is used to fetch roles and assign the result set to a special session variable called "GROUP", which will be examined later by the Oracle Business Intelligence Server for matching. The initialization block SQL is:

```
SELECT DISTINCT
'GROUP' , ROLENAME
FROM
PSROLEUSER
WHERE
ROLEUSER = ':USER'
```

Oracle Business Intelligence Applications offers the following types of data security:

- ["About Operating Unit-based Security Against PeopleSoft Financials"](#)
- ["About Company Org-based Security Against PeopleSoft Financials or PeopleSoft HR"](#)
- ["About Ledger-based Security against Oracle's PeopleSoft Enterprise Applications"](#)
- ["About HR Org-based Security Against PeopleSoft HR"](#)
- ["About Payables Org-based security Against PeopleSoft Financials"](#)
- ["About Receivables Org-based Security Against PeopleSoft Financials"](#)
- ["About SET ID-based Security Against PeopleSoft Financials"](#)
- ["About Human Resource Personnel Data Analyst Security Against PeopleSoft HR"](#)

### 7.3.2 About Operating Unit-based Security Against PeopleSoft Financials

The sequence for Operating Unit-based Security against PeopleSoft Financials is described below:

- When a user logs in to Oracle BI Applications, the session variable below is set automatically.

```
USER (System variable)
```

- The Oracle BI Server then gets the operating units (or the general ledger business units in PeopleSoft Financials) corresponding to the USER from the following tables:
  - PS\_SEC\_BU\_OPR
  - PS\_BUS\_UNIT\_TBL\_GL
  - PS\_INSTALLATION\_FS

The following session variable is set automatically:

```
OU_ORG (Row-wise variable)
```

The initialization block "Operating Unit Organizations", which sets the value for this variable, is shown below.

#### **Initialization block -- "Operating Unit Org"**

The initialization block "Operating Unit Org" sets value for variable OU\_ORG using the following SQL:

```
SELECT DISTINCT 'OU_ORG', S1.BUSINESS_UNIT
FROM PS_SEC_BU_OPR S1, PS_BUS_UNIT_TBL_GL A, PS_INSTALLATION_
FS I
WHERE S1.OPRID = ':USER'
AND S1.BUSINESS_UNIT = A.BUSINESS_UNIT
AND I.SECURITY_TYPE = 'O'
AND I.BU_SECURITY = 'Y'
UNION
SELECT DISTINCT 'OU_ORG', S2.BUSINESS_UNIT
FROM PS_SEC_BU_CLS S2,
PS_BUS_UNIT_TBL_GL A,
PS_INSTALLATION_FS I2,
PSOPRDEFN P
WHERE P.OPRID = ':USER'
AND S2.BUSINESS_UNIT = A.BUSINESS_UNIT
AND P.OPRCLASS = S2.OPRCLASS
AND I2.SECURITY_TYPE = 'C'
AND I2.BU_SECURITY = 'Y'
```

- The "Operating Unit Org-based Security" security group contains all the data access permission filters. During implementation, customers need to decide which users or groups should be associated to this security group.

### 7.3.3 About Company Org-based Security Against PeopleSoft Financials or PeopleSoft HR

The sequence for Company Org-based Security Against PeopleSoft Financials or PeopleSoft HR is described below:

- When a user logs in to Oracle BI Applications, the session variable below is set automatically.  
USER (System variable)
- The Oracle BI Server then gets the companies or business units corresponding to the USER from the following tables:
  - PS\_SEC\_BU\_OPR
  - PS\_BUS\_UNIT\_TBL\_GL
  - PS\_INSTALLATION\_FS for PeopleSoft Financials
  - PS\_SCRTY\_TBL\_DEPT
  - PS\_BU\_DEPT\_VW
  - PS\_BUS\_UNIT\_TBL\_GL
  - PSOPRDEFN for PeopleSoft HR

The following session variable is set automatically:

COMPANY (Row-wise variable)

The initialization block "Companies", which sets the value for this variable, is shown below.

#### **Initialization block -- "Companies"**

The initialization block "Companies" sets value for variable COMPANY using the following SQL:

For PeopleSoft Financials:

```
SELECT DISTINCT 'COMPANY', S1.BUSINESS_UNIT
FROM PS_SEC_BU_OPR S1, PS_BUS_UNIT_TBL_GL A, PS_
INSTALLATION_FS I
WHERE S1.OPRID = ':USER'
AND S1.BUSINESS_UNIT = A.BUSINESS_UNIT
AND I.SECURITY_TYPE = 'O'
UNION
SELECT DISTINCT 'COMPANY', S2.BUSINESS_UNIT
FROM PS_SEC_BU_CLS S2,
PS_BUS_UNIT_TBL_GL A,
PS_INSTALLATION_FS I2,
PSOPRDEFN P
WHERE P.OPRID = ':USER'
AND S2.BUSINESS_UNIT = A.BUSINESS_UNIT
AND P.OPRCLASS = S2.OPRCLASS
AND I2.SECURITY_TYPE = 'C'
AND I2.BU_SECURITY = 'Y'
```

For PeopleSoft HR:

```
SELECT DISTINCT 'COMPANY', C.BUSINESS_UNIT
FROM PSOPRDEFN A, PS_SCRTY_TBL_DEPT B, PS_BU_DEPT_VW C, PS_
BUS_UNIT_TBL_GL D
WHERE
A.ROWSECCLASS = B.ROWSECCLASS AND
B.ACCESS_CD = 'Y' AND
B.DEPTID = C.DEPTID AND
C.BUSINESS_UNIT = D.BUSINESS_UNIT AND
A.OPRID = ':USER'
```

- The "Company Org-based Security" security group contains all the data access permission filters. During implementation, customers need to decide which users or groups should be associated to this security group.

### 7.3.4 About Ledger-based Security against Oracle's PeopleSoft Enterprise Applications

Ledger-based security against PeopleSoft Enterprise Applications is introduced in 7.9.4. Ledger in PeopleSoft is reference data that is secured by and shared by business units. The Ledger table has the setID field and leverages the TableSet feature from PeopleTool. In addition, the Ledger data access is controlled by the Row-level security. With row-level support, you can implement security to restrict individual users or permission lists from specific rows of data that are controlled by the Ledger. The Ledger-based security filters data based on the ledgers associated to the logged in user. The ledgers are restricted by PeopleSoft TableSet and Row-based security.

The sequence for Ledger-based security against PeopleSoft Financials is described below:

- When a user logs in to Oracle BI Applications, the session variable below is set automatically.

```
USER (System variable)
```

- The Oracle BI Server gets the ledgers corresponding to the USER.
- The following session variable is set automatically:

```
LEDGER (Row-wise variable)
```

- The initialization block "Ledgers", which sets the value for this variable, is set as below.

```
SELECT DISTINCT 'LEDGER', LG.SETID || SO.LEDGER
FROM PS_SEC_LEDGER_OPR SO, PS_LED_DEFN_TBL LG, PS_
INSTALLATION_FS IFS
WHERE SO.LEDGER = LG.LEDGER AND IFS.SECURITY_TYPE = 'O'
AND IFS.LEDGER_SECURITY = 'Y' AND SO.OPRID = ':USER'
UNION
SELECT distinct 'LEDGER', LG.SETID || SC.LEDGER
FROM PS_SEC_LEDGER_CLS SC, PS_LED_GRP_TBL LG, PSOPRDEFN OP,
PSROLEUSER ORL, PSROLECLASS RCL, PS_INSTALLATION_FS IFS
WHERE SC.LEDGER_GROUP = LG.LEDGER_GROUP AND SC.OPRCLASS =
RCL.CLASSID AND OP.OPRID = ORL.ROLEUSER
AND ORL.ROLENAME = RCL.ROLENAME and IFS.SECURITY_TYPE = 'C'
AND IFS.LEDGER_SECURITY = 'Y' AND OP.OPRID = ':USER'
```

When you set up Ledger security against a PeopleSoft application, you should also set up the Company Org-based security against PeopleSoft. The Ledger-based security does not automatically restrict the data by the GL business unit.

### 7.3.5 About HR Org-based Security Against PeopleSoft HR

The sequence for HR Org-based Security Against PeopleSoft HR is described below:

- When a user logs in to Oracle BI Applications, the session variable below is set automatically.

```
USER (System variable)
```

- The Oracle BI Server gets the HR business units corresponding to the USER from the following tables:
  - PSOPRDEFN
  - PS\_SCRTY\_TBL\_DEPT
  - PS\_BU\_DEPT\_VW
  - PS\_BUS\_UNIT\_TBL\_HR

The following session variable is set automatically:

HR\_ORG (Row-wise variable)

The initialization block "HR Organizations", which sets the value for this variable, is shown below.

#### **Initialization block -- "HR Organizations"**

The initialization block "HR Organizations" sets value for variable HR\_ORG using the following SQL:

```
SELECT DISTINCT 'HR_ORG', C.BUSINESS_UNIT
FROM PSOPRDEFN A, PS_SCRTY_TBL_DEPT B, PS_BU_DEPT_VW C, PS_
BUS_UNIT_TBL_HR D
WHERE
A.ROWSECCLASS = B.ROWSECCLASS AND
B.ACCESS_CD = 'Y' AND
B.DEPTID = C.DEPTID AND
C.BUSINESS_UNIT = D.BUSINESS_UNIT AND
A.OPRID = ':USER'
```

- The "HR Org-based Security" security group contains all the data access permission filters. During implementation, customers need to decide which users or groups should be associated to this security group.
- When users create ad-hoc reports, they see the data that is assigned with their permissions. For reports involved with the tables defined above, users are restricted to the data pertaining to their visibility in the organization structure.

### **7.3.6 About Payables Org-based security Against PeopleSoft Financials**

The sequence for Payables Org-based security Against PeopleSoft Financials is described below:

- When a user logs in to Oracle BI Applications, the session variable below is set automatically.
 

USER (System variable)
- The Oracle BI Server gets the HR business units corresponding to the USER from the following tables:
  - PSOPRDEFN
  - PS\_SEC\_BU\_OPR
  - PS\_SEC\_BU\_CLR
  - PS\_INSTALLATION\_FS

– PS\_BUS\_UNIT\_TBL\_AP

The following session variable is set automatically:

PAYABLES\_ORG (Row-wise variable)

The initialization block "Payables Organizations", which sets the value for this variable, is shown below.

#### **Initialization block -- "Payables Organizations"**

The initialization block "Payables Organizations" sets value for variable PAYABLES\_ORG using the following SQL:

```
SELECT DISTINCT 'PAYABLES_ORG', s1.BUSINESS_UNIT
FROM PS_SEC_BU_OPR s1, PS_BUS_UNIT_TBL_AP a, PS_INSTALLATION_
FS i
WHERE s1.OPRID = ':USER'
AND s1.BUSINESS_UNIT = a.BUSINESS_UNIT
AND i.SECURITY_TYPE = 'O'
AND i.BU_SECURITY = 'Y'
UNION
SELECT DISTINCT 'PAYABLES_ORG', s2.BUSINESS_UNIT
FROM PS_SEC_BU_CLS s2, PS_BUS_UNIT_TBL_AP a, PS_INSTALLATION_
FS i2, PSOPRDEFN p
WHERE p.OPRID = ':USER'
AND s2.BUSINESS_UNIT = a.BUSINESS_UNIT
AND p.OPRCLASS = s2.OPRCLASS
AND i2.SECURITY_TYPE = 'C'
AND i2.BU_SECURITY = 'Y'
```

- The "Payables Org-based Security" security group contains all the data access permission filters. During implementation, customers need to decide which users or groups should be associated to this security group.
- When users create ad-hoc reports, they see the data that is assigned with their permissions. For reports involved with the tables defined above, users are restricted to the data pertaining to their visibility in the organization structure.

### **7.3.7 About Receivables Org-based Security Against PeopleSoft Financials**

The sequence for Receivables Org-based Security Against PeopleSoft Financials is described below:

- When a user logs in to Oracle BI Applications, the session variable below is set automatically.  
USER (System variable)
- The Oracle BI Server gets the HR business units corresponding to the USER from the following tables:
  - PSOPRDEFN
  - PS\_SEC\_BU\_OPR

- PS\_SEC\_BU\_CLR
- PS\_INSTALLATION\_FS
- PS\_BUS\_UNIT\_TBL\_AP

The following session variable is set automatically:

RECEIVABLES\_ORG (Row-wise variable)

The initialization block "Receivables Organizations", which sets the value for this variable, is shown below.

#### **Initialization block -- "Receivables Organizations"**

The initialization block "Receivables Organizations" sets value for variable RECEIVABLES\_ORG using the following SQL:

```
SELECT DISTINCT 'RECEIVABLES_ORG', s1.BUSINESS_UNIT
FROM PS_SEC_BU_OPR s1, PS_BUS_UNIT_TBL_AR a, PS_INSTALLATION_
FS i
WHERE s1.OPRID = ':USER'
AND s1.BUSINESS_UNIT = a.BUSINESS_UNIT AND i.SECURITY_TYPE =
'O'
AND i.BU_SECURITY = 'Y'
UNION
SELECT DISTINCT 'RECEIVABLES_ORG', s2.BUSINESS_UNIT
FROM PS_SEC_BU_CLS s2, PS_BUS_UNIT_TBL_AR a, PS_INSTALLATION_
FS i2, PSOPRDEFN p
WHERE p.OPRID = ':USER'
AND s2.BUSINESS_UNIT = a.BUSINESS_UNIT AND p.OPRCLASS =
s2.OPRCLASS AND i2.SECURITY_TYPE = 'C'
AND i2.BU_SECURITY = 'Y'
```

- The "Receivables Org-based Security" security group contains all the data access permission filters. During implementation, customers need to decide which users or groups should be associated to this security group.
- When users create ad-hoc reports, they see the data that is assigned with their permissions. For reports involved with the tables defined above, users are restricted to the data pertaining to their visibility in the organization structure.

### **7.3.8 About SET ID-based Security Against PeopleSoft Financials**

The sequence for SET ID-based Security Against PeopleSoft Financials is described below:

- When a user logs in to Oracle BI Applications, the session variable below is set automatically.

USER (System variable)

- The Oracle BI Server gets the set IDs corresponding to the USER from the following tables:
  - PS\_SEC\_SETID\_OPR
  - PS\_SEC\_SETID\_CLS

- PS\_INSTALLATION\_FS
- PSOPRDEFN

The following session variable is set automatically:

SET\_ID (Row-wise variable)

The initialization block "Set ID" sets value for variable SET\_ID using the following SQL:

For PeopleSoft Financials:

```
SELECT DISTINCT 'SET_ID', s1.SETID
FROM PS_SEC_SETID_OPR s1, PS_INSTALLATION_FS i
WHERE s1.OPRID = ':USER'
AND i.SECURITY_TYPE = 'O'
AND i.SETID_SECURITY = 'Y'
UNION
SELECT DISTINCT 'SET_ID', s2.SETID
FROM PS_SEC_SETID_CLS s2, PS_INSTALLATION_FS i2, PSOPRDEFN p
WHERE p.OPRID = ':USER'
AND p.OPRCLASS = s2.OPRCLASS
AND i2.SECURITY_TYPE = 'C'
AND i2.SETID_SECURITY = 'Y'
```

- The "Set ID-based Security" security group contains all the data access permission filters. During implementation, customers need to decide which users or groups should be associated to this security group.

### 7.3.9 About Human Resource Personnel Data Analyst Security Against PeopleSoft HR

HR personnel need to see all data for the Internal Organizations for which they are responsible for and the data for their subordinates in their own organization. The Human Resource Personnel Data Security group supports this requirement. The security mechanism for this group uses the following metadata elements:

- Variable HR\_ORG, which is a variable defined by the Row-wise Initialization Block: HR Organizations. This data set stores all the Organizations the user is responsible for, plus the user's own organization (the same as the one selected in USER\_HR\_ORG). The query for populating this data set is:

```
SELECT DISTINCT
'HR_ORG',
C.BUSINESS_UNIT
FROM
PSOPRDEFN A, PS_SCRTY_TBL_DEPT B, PS_BU_DEPT_VW C, PS_BUS_
UNIT_TBL_HR D
WHERE
A.ROWSECCLASS = B.ROWSECCLASS AND
B.ACCESS_CD = 'Y' AND
```

```

B.DEPTID = C.DEPTID AND
C.BUSINESS_UNIT = D.BUSINESS_UNIT AND
A.OPRID = ':USER'
UNION
SELECT DISTINCT 'HR_ORG', FINAL_JOB.BUSINESS_UNIT
FROM (
SELECT X.EMPLID, MAX(X.BUSINESS_UNIT) BUSINESS_UNIT FROM
(
SELECT A.EMPLID, A.EMPL_RCD, A.EFFDT, EFFSEQ, A.JOB_
INDICATOR, A.EMPL_STATUS, A.BUSINESS_UNIT
FROM PS_JOB A ,
(SELECT EMPLID, MAX(EFFDT) MAX_EFFDT
FROM PS_JOB
WHERE
JOB_INDICATOR = 'P' AND EMPL_STATUS IN ('A', 'L', 'P', 'W')
GROUP BY EMPLID) B
WHERE
A.EMPLID = B.EMPLID
AND A.EFFDT = B.MAX_EFFDT
AND A.JOB_INDICATOR = 'P' AND A.EMPL_STATUS IN ('A', 'L',
'P', 'W')
AND A.EFFSEQ = (SELECT MAX (C.EFFSEQ)
FROM PS_JOB C
WHERE
C.EMPLID = A.EMPLID AND
C.EMPL_RCD = A.EMPL_RCD AND
C.EFFDT = A.EFFDT AND
C.JOB_INDICATOR = 'P' AND C.EMPL_STATUS IN ('A', 'L', 'P',
'W'))
) X
GROUP BY X.EMPLID
) FINAL_JOB, PSOPRDEFN
WHERE
FINAL_JOB.EMPLID = PSOPRDEFN.EMPLID AND
PSOPRDEFN.OPRID = ':USER'

```

- Variable USER\_HR\_ORG, defined using the Initialization block: User HR Organizations. This variable stores the user's own organization. The query for populating this variable is:

```

SELECT DISTINCT FINAL_JOB.BUSINESS_UNIT

```

```

FROM (
SELECT X.EMPLID, MAX(X.BUSINESS_UNIT) BUSINESS_UNIT FROM
(
SELECT A.EMPLID, A.EMPL_RCD, A.EFFDT, EFFSEQ, A.JOB_INDICATOR,
A.EMPL_STATUS, A.BUSINESS_UNIT
FROM PS_JOB A ,
(SELECT EMPLID, MAX(EFFDT) MAX_EFFDT
FROM PS_JOB
WHERE
JOB_INDICATOR = 'P' AND EMPL_STATUS IN ('A', 'L', 'P', 'W')
GROUP BY EMPLID) B
WHERE
A.EMPLID = B.EMPLID
AND A.EFFDT = B.MAX_EFFDT
AND A.JOB_INDICATOR = 'P' AND A.EMPL_STATUS IN ('A', 'L',
'P', 'W')
AND A.EFFSEQ = (SELECT MAX (C.EFFSEQ)
FROM PS_JOB C
WHERE
C.EMPLID = A.EMPLID AND
C.EMPL_RCD = A.EMPL_RCD AND
C.EFFDT = A.EFFDT AND
C.JOB_INDICATOR = 'P' AND C.EMPL_STATUS IN ('A', 'L', 'P',
'W'))
) X
GROUP BY X.EMPLID
) FINAL_JOB, PSOPRDEFN
WHERE
FINAL_JOB.EMPLID = PSOPRDEFN.EMPLID AND
PSOPRDEFN.OPRID = ':USER'

```

- A security group: Human Resources Analyst. The data filter defined for this group is:

```

Core."Dim - Employee Organization"."Employee Organization
Number" = VALUEOF(NQ_SESSION."HR_ORG") AND (Core."Dim -
Employee Organization"."Employee Organization Number" <>
VALUEOF(NQ_SESSION."USER_HR_ORG") OR Core."Dim - Security
Dimension"."Hierarchy Based Column" = VALUEOF(NQ_
SESSION."USER"))

```

This filter joins the fact used in the report to the Employee Organization dimension to get the Organization number for the Employee owner of the fact record. If this Organization is among the HR\_ORGs it will be compared next to the user's own

organization. If they are different there is no further check, the record is selected. If they are the same, an additional filter is applied based on Employee hierarchy, to make sure the employee owner of this fact record is one of the user's subordinates.

### 7.3.10 Employee Based Security Against PeopleSoft Application

Employee based security restricts data visibility of the records to the Owner of that record, and all employees that they report to in the company's Employee hierarchy. This security mechanism uses data from the data warehouse database, and shares the metadata components with other supported applications (for example, Oracle EBS, Siebel CRM). Out of the box, this type of security supports only HR Analytics facts. For more information on how this security mechanism works, see [Section 7.5.2, "Primary Position Based Security for CRM Vertical applications"](#).

## 7.4 Metadata Object-Level Security in Oracle BI Applications

Object-level security controls access to various Analytics objects, primarily metadata and Presentation Services objects.

### 7.4.1 Metadata Object-Level Security (Repository Groups)

Repository groups control access to metadata objects such as subject areas, tables and columns.

#### 7.4.1.1 Where is Repository Groups Security Configured?

Metadata object security is configured in the Analytics repository (OracleBIAnalyticsApps.rpd) using the Oracle BI Administration Tool. The User Group Everyone is denied access to each of the subject areas. Each subject area is configured to give explicit read access to selected related responsibilities. This access can be extended to table and column level.

---



---

**Note:** In the shipped product, only permissions at the subject area level have been configured.

---



---

The exceptions to the explicit configuration rule are the Communications and Financial Analytics industry applications, where there are tables and columns specific to these two industries scattered throughout the general Siebel operational application subject areas. These industry-specific metadata objects are hidden from other groups.

Oracle Business Intelligence supports hierarchies within the groups in the repository. In the Analytics repository there are certain groups that are parent groups, which define the behavior of all the child groups. Inheritance is used to let permissions ripple through to child groups. The parent groups and their purpose are shown in [Table 7-1](#).

**Table 7-1 Repository Parent Groups**

Parent Group	Permissions Inherited by
Finance	All Financial applications groups
Insurance	All Insurance applications groups
CM General	All Communications applications
Consumer Sector	Consumer Sector groups
Pharma	Life Sciences/Pharmaceuticals applications groups

**Table 7–1 (Cont.) Repository Parent Groups**

Parent Group	Permissions Inherited by
Channel Managers	All Channel applications groups
Partner Managers	All Partner application groups

### 7.4.1.2 Metadata Object-Level (Presentation Services) Security in Oracle BI Applications

Presentation Services objects, such as dashboards and pages, are controlled using Presentation Services groups, which have the same name as the Siebel responsibilities. Access to dashboards and pages is controlled using the Presentation Services groups. If you log on as a user who belongs to the Presentation Services group Field Sales Representative Analytics, then you see only the Overview, Forecasting, and Details pages within the Pipeline Dashboard. In a similar fashion, you see only dashboards that allow you access to at least one page within that dashboard. These groups are customized in the Oracle BI Web interface.

For Oracle Business Intelligence integrated with Oracle's Siebel operational applications, Presentation Services security makes use of the following principles:

- Security in Presentation Services has been pre-configured for the groups listed in [Table 7–1](#) for each application.
- Permissions to each dashboard in Presentation Services are matched with the permissions of each related Siebel operational application view. In the Siebel operational application, views are controlled through responsibilities. However, in Oracle Business Intelligence Presentation Services, access to dashboards for each group is controlled through Web Administration. If the two access setups do not match, both of the following situations can occur:
  - If users have access to a view in the Siebel operational application, but do not have access to the corresponding dashboard, then they receive an error message indicating that they do not have access to the dashboard.
  - If users try to access a dashboard containing reports based on a subject area to which they do not have access, they see a dashboard with no reports.

## 7.5 Integrating Data Security with the Siebel CRM Application

This section describes how Oracle BI Applications security integrates with Siebel CRM security, and contains the following topics:

- [Section 7.5.1, "Primary Position based Security"](#)
- [Section 7.5.2, "Primary Position Based Security for CRM Vertical applications"](#)
- [Section 7.5.3, "Team Based Visibility"](#)
- [Section 7.5.4, "Embedded Oracle Business Intelligence support for Changing Positions"](#)
- [Section 7.5.5, "Business Unit based Security"](#)
- [Section 7.5.6, "Primary Owner Security"](#)

### 7.5.1 Primary Position based Security

This section covers Primary Position based Security and contains the following sections:

- [Section 7.5.1.1, "Introduction"](#)
- [Section 7.5.1.2, "Primary Position based Security Group"](#)
- [Section 7.5.1.3, "Configuring Oracle BI Repository Table Joins for Position/Employee Based Security"](#)

### 7.5.1.1 Introduction

This type of security restricts data visibility for a fact (or dimension) record to the Primary owner of this record and those above him in the hierarchy. The primary owner of a record could be a Position (like for Opportunities and Revenue records in Siebel CRM application), or an Employee (like for Payroll records in Oracle EBS application). Primary Position based Security uses a flattened hierarchy table called W\_POSITION\_DH, which is based on W\_POSITION\_D, and is treated as a slowly changing dimension of type 2. For Siebel CRM based data, this table (W\_POSITION\_D) is populated from the Position table, so a new record is created for the same position, every time a new employee is associated with this position as the Primary Employee. For Oracle EBS and PeopleSoft sourced data, this table is populated from the Employee table (PER\_ALL\_PEOPLE\_F, JTF\_RESOURCES among others) for Oracle EBS), so a new record is created for the same Employee, every time the hierarchy structure of that employee, or his position changes.

Consequently, every record in the source table(s) can be represented by more than one record in this table (W\_POSITION\_DH), but only one record can have the value of CURRENT\_FLG as 'Y' at any time. The W\_POSITION\_DH table also contains one set of columns prefixed with CURRENT, and another set of columns not prefixed with CURRENT. The columns that are prefixed with CURRENT reflect the current hierarchy structure for the Position (or Employee record) at any time. The columns that are not prefixed with CURRENT reflect the hierarchy structure for the same Position (or Employee record) during the period between EFFECTIVE\_START\_DT and EFFECTIVE\_END\_DT. This latter set of columns is used for an AS WAS type of security, meaning the fact records are visible to the owner of a record, and his upper level managers at the time the record was created, even after he changes position or managers in the company hierarchy.

Facts join to this dimension via the record owner, for example W\_REVN\_F is joined using PR\_POSITION\_DH\_WID, where PR\_POSITION\_DH\_WID is the Primary Position on the Revenue in the source application. Another example is W\_PAYROLL\_F is joined using EMP\_POSTN\_DH\_WID, where EMP\_POSTN\_DH\_WID is the employee owner of this payroll record.

### 7.5.1.2 Primary Position based Security Group

Primary Position based Security allows only the record owner and any Employee up in his hierarchy chain to see the record. Primary Position based Security uses the following metadata elements in the repository:

1. Session variable: HIER\_LEVEL. It is populated by the Initialization block "User Hierarchy Level" using the following SQL:

```
Select round(FIXED_HIER_LEVEL) FROM VALUEOF(OLAPTBO).W_
POSITION_DH WHERE BASE_LOGIN= ':USER' AND CURRENT_FLG='Y'
```

HIER\_LEVEL value will take a number between 0 and 9, and will designate the level of the user in the Position (or Employee) hierarchy of the company. For example the CEO of the company is the only Employee who's HIER\_LEVEL takes the value 9, if the employee hierarchy is a full tree.

2. A logical dimension "Dim - Security" joined to the supported facts with Primary Position/Employee - based security. This logical dimension is defined using on the physical table W\_POSITION\_DH.
3. A logical column in this dimension, called "Hierarchy Based Column", and defined as follows:

```
"INDEXCOL(VALUEOF(NQ_SESSION."HIER_LEVEL"), "Core"."Dim -
Security Dimension"."Current Base Level Login", "Core"."Dim -
Security Dimension"."Current Level 1 Login", "Core"."Dim -
Security Dimension"."Current Level 2 Login", "Core"."Dim -
Security Dimension"."Current Level 3 Login", "Core"."Dim -
Security Dimension"."Current Level 4 Login", "Core"."Dim -
Security Dimension"."Current Level 5 Login", "Core"."Dim -
Security Dimension"."Current Level 6 Login", "Core"."Dim -
Security Dimension"."Current Level 7 Login", "Core"."Dim -
Security Dimension"."Current Level 8 Login", "Core"."Dim -
Security Dimension"."Current Top Level Login")".
```

The IndexCol function in this definition makes the Hierarchy Based Column default to one of the logical columns in the list based on the value of HIER\_LEVEL. So if HIER\_LEVEL is 0, the new column will default to the first column in the list, etc.

4. A filter in the security group "Primary Employee/Position Hierarchy-based Security" defined as follows: ("Core"."Dim - Security Dimension"."Hierarchy Based Column" = VALUEOF(NQ\_SESSION."USER")),

A user needs to be a member of the security group "Primary Employee/Position Hierarchy-based Security", through one of his responsibilities (for Siebel and Oracle EBS Applications) and Roles (for PeopleSoft application), for the data security filters to apply. Users are assigned to this security group based on their responsibilities, using the Initialization block: Authorization, defined in [Section 7.6.3, "Initialization Blocks Used For Security Oracle BI Applications"](#). Out of the box, this initialization block is populated using the following SQL:

```
Select 'GROUP', R.NAME
from VALUEOF(TBO).S_RESP R, VALUEOF(TBO).S_PER_RESP P,
VALUEOF(TBO).S_USER U
where U.LOGIN=Upper(':USER') and U.ROW_ID=P.PER_ID and
P.RESP_ID=R.ROW_ID
UNION
select 'GROUP', CASE VALUEOF(NQ_SESSION.HIER_LEVEL)
WHEN 0 THEN 'Hierarchy Level (Base)'
when 1 then 'Hierarchy Level 1'
when 2 then 'Hierarchy Level 2'
when 3 then 'Hierarchy Level 3'
when 4 then 'Hierarchy Level 4'
when 5 then 'Hierarchy Level 5'
when 6 then 'Hierarchy Level 6'
when 7 then 'Hierarchy Level 7'
when 8 then 'Hierarchy Level 8'
```

```
When 9 then 'Hierarchy Level (Top) '
ELSE 'NOGROUP' END from VALUEOF(TBO).S_DUAL
```

The first part of this SQL selects the user's responsibilities from the Siebel CRM application. The user will be assigned automatically to the Security groups with the same name in the Oracle Business Intelligence repository.

The second part of this sql assigns the user to one of the Oracle Business Intelligence specific security groups: "Hierarchy Level (Base)", ..., "Hierarchy Level (Top)", based on the variable `HIER_LEVEL`. These security groups are not used for data security purposes, but just for Presentation column purposes, in conjunction with the Web Choose function defined in some reports. The purpose of this function is to allow a multi-user report to show different Position columns to the user, based on his hierarchy level. This is very similar to the `IndexCol` function defined above.

### 7.5.1.3 Configuring Oracle BI Repository Table Joins for Position/Employee Based Security

You can add Primary Position based Security to a new dimension or fact in Oracle Business Intelligence. The following tasks use a dimension `W_AGREE_D` (Agreement) as an example.

#### To add security support to a dimension

1. Create an alias on `W_POSITION_DH` specifically to join to the underlying physical table.
2. Configure the join in the physical layer.
3. Add the `W_POSITION_DH` alias to the dimension's Logical table source.
4. Add new logical columns `CURRENT_BASE_LOGIN`, `CURRENT_LVL1ANC_LOGI`; etc. to the logical table, and map them to the corresponding physical columns.
5. Add the Hierarchy column "Hierarchy Based Column", as defined in described in section 3.2 above.
6. Open the security group screen using Manage/Security in Oracle BI Administrator.
  - a. Right-click the group "Primary Employee/Position Hierarchy-based Security".and choose Properties.
  - b. In the Properties dialog box, click the Permissions box and select the Filter tab.
  - c. To add a new filter, click on the Add button.
  - d. In the new window, select the Business Model tab, and find the logical table: `Dim - Agreement`.  
A new record will be added to the list of Filters automatically.
  - e. Click on the ellipsis box, and add the filter condition "`Core`"."`Dim - Customer`"."`Hierarchy Based Login`" = `VALUEOF(NQ_SESSION."USER")` in the Security Filter Expression Builder and click OK.

#### To add security support to a fact

1. Join the underlying physical table to `Dim_W_POSITION_DH_Position_Hierarchy`.  
This assumes you already created the appropriate foreign key in the fact table, and populated it correctly.

2. Join the logical table to the Dim - Security Dimension.
3. Open the security group screen using Manage/Security in Oracle BI Administrator.
  - a. Right-click the group "Primary Employee/Position Hierarchy-based Security".and choose Properties.
  - b. In the Properties dialog box, click the Permissions box and select the Filter tab.
  - c. To add a new filter, click on the Add button.
  - d. In the new window, select the Business Model tab, and find the logical table: Dim - Agreement.  
A new record will be added to the list of Filters automatically.
  - e. Click on the ellipsis box, and add the condition "Core"."Dim - Security Dimension"."Hierarchy Based Column" = VALUEOF(NQ\_SESSION."USER") in the Security Filter Expression Builder and click OK.

## 7.5.2 Primary Position Based Security for CRM Vertical applications

This section covers primary position based security for CRM vertical applications and contains the following sections:

- [Section 7.5.2.1, "Oracle Partner Analytics Security Settings"](#)
- [Section 7.5.2.2, "Consumer Sector Analytics Security Settings"](#)
- [Section 7.5.2.3, "Communications, Media, and Energy \(CME\) Analytics Security Settings"](#)
- [Section 7.5.2.4, "Financial Services Analytics Security Settings"](#)
- [Section 7.5.2.5, "Oracle Pharma Sales Analytics and Oracle Pharma Marketing Analytics Security Settings"](#)
- [Section 7.5.2.6, "Usage Accelerator Analytics Security Settings"](#)

### 7.5.2.1 Oracle Partner Analytics Security Settings

Oracle's Siebel operational applications include Oracle's Siebel Sales, Oracle's Siebel Service, and Oracle's Siebel Partner Relationship Management. This topic describes the additional security configurations that may be necessary for the Siebel operational application Siebel ERM Analytics, which is divided into HR, Partner Manager, and Partner Portal Analytics.

Oracle Partner Analytics incorporates the concept of role-based analytics. Role-based analytics provides brand owners the ability to display dashboards and pages to users based on their specific roles. For example, a sales manager would have the ability to view dashboards related to pipeline and sales effectiveness, whereas the marketing manager would have the ability to view dashboards related to campaigns. Oracle Partner Analytics also includes flexible security mechanisms to control access to subject areas and to data.

The Analytics roles map to Siebel Responsibilities in the Siebel operational application. This topic describes the roles and associated dashboards and pages for both partner Manager and Partner Portal applications. It also includes subject area and data-level security settings for responsibilities.

**7.5.2.1.1 Partner Manager Role-Based Interactive Dashboards Mapping** Table 7-2 provides the dashboard and page tab mapping for specific responsibilities in the Siebel PRM Partner Manager application.

**Table 7-2 Siebel Responsibilities for PRM Analytics**

<b>Responsibility</b>	<b>Dashboard</b>	<b>Page Tab Name</b>
Channel Account Manager Analytics User	Channel Customers	Overview
	Channel Customers	Sales
	Channel Sales	Products
	Channel Sales	Sales
	Channel Service	Products
	Channel Service	Service
	Channel Training	Training Profile
Channel Executive Analytics User	Channel Customers	Customer Profile
	Channel Executive	Customer Satisfaction
	Channel Executive	Pipeline
	Channel Executive	Product
	Channel Executive	Program
	Channel Executive	Revenue
	Channel Executive	Service
	Channel Segmentation	Channel Mix
	Channel Segmentation	Partner Territory
Channel Marketing Manager Analytics User	Channel Customers	Overview
	Channel Customers	Sales
	Customer Marketing	Effectiveness
	Customer Marketing	Responses
	Customer Marketing	ROI
Channel Operations Analytics User	Channel Commerce	Orders
	Channel Commerce	Overview
	Channel Commerce	Quotes
	Channel Commerce	Products
	Channel Customers	Overview
	Channel Customers	Sales

**Table 7–2 (Cont.) Siebel Responsibilities for PRM Analytics**

<b>Responsibility</b>	<b>Dashboard</b>	<b>Page Tab Name</b>
	Channel Customers	Service
	Channel Marketing	Effectiveness
	Channel Marketing	Overview
	Channel Sales	Margins
	Channel Sales	Pipeline
	Channel Sales	Revenue
	Channel Sales	Sales Cycle
	Channel Sales	Wins
	Channel Segmentation	Partner Territory
	Channel Segmentation	Partner Tier
	Channel Segmentation	Partner Type
	Channel Service	Customer Satisfaction
	Channel Service	Overview
	Channel Service	Products
	Channel Service	Resolution Time
	Channel Service	Service Requests
	Channel Training	Overview
	Channel Training	Performance

**7.5.2.1.2 Oracle's Siebel Partner Portal Role-Based Interactive Dashboards Mapping** The dashboard and page tab mapping for specific responsibilities in the Siebel PRM Partner Portal application are shown in [Table 7–3](#).

**Table 7–3 Responsibilities for PRM Partner Portal Analytics**

<b>Responsibility</b>	<b>Dashboard</b>	<b>Page Tab Name</b>
Partner Executive Analytics User	Partner Executive	Pipeline
	Partner Executive	Products
	Partner Executive	Sales Effectiveness
	Partner Executive	Service
Partner Operations Analytics User	Partner Commerce	Overview
	Partner Commerce	Products
	Partner Marketing	Overview
	Partner Marketing	ROI
	Partner Sales	Pipeline
	Partner Sales	Revenue
	Partner Service	Customer Sat
	Partner Service	Overview

**Table 7–3 (Cont.) Responsibilities for PRM Partner Portal Analytics**

<b>Responsibility</b>	<b>Dashboard</b>	<b>Page Tab Name</b>
Partner Sales Manager Analytics User	Partner Service	Service Requests
	Partner Training	Training
	Partner Commerce	Orders
	Partner Commerce	Overview
	Partner Commerce	Quotes
	Partner Sales	Pipeline
	Partner Sales	Revenue
Partner Sales Rep Analytics User	Partner Sales	Subordinates
	Partner Training	Subordinates
	Partner Commerce	Orders
	Partner Commerce	Overview
	Partner Commerce	Quotes
	Partner Sales	Pipeline
	Partner Sales	Revenue
Partner Service Manager Analytics User	Partner Sales	Subordinates
	Partner Training	Subordinates
	Partner Service	Customer Sat
	Partner Service	Overview
	Partner Service	Service Requests
	Partner Service	Subordinates
	Partner Training	Subordinates
Partner Service Rep Analytics User	Partner Service	Overview
	Partner Service	Service Requests
	Partner Service	Subordinates
	Partner Training	Subordinates

**7.5.2.1.3 PRM Analytics Subject Area Mapping** Ad hoc queries in Siebel PRM Analytics are built by the user, depending on user responsibilities and based on columns in subject areas in the Oracle Business Intelligence application. By restricting visibility to subject areas based on responsibilities, Oracle's Siebel PRM Analytics provides brand owners a flexible way to deploy role-based analytics.

The subject area visibility for responsibilities in Partner Manager are shown in [Table 7–4](#), where a X indicates that subject area is visible for the user holding that responsibility.

**Table 7-4 Responsibilities for PRM Partner Manager Analytics**

Subject Area	Channel Executive Analytics User	Channel Operations Analytics User	Channel Account Manager Analytics User	Channel Marketing Manager Analytics User
Activities	X	X	X	X
Assets	X	X	X	
Campaigns	X	X	X	X
Consumers	X	X	X	X
Customer Satisfaction	X	X	X	
Customers	X	X	X	X
Orders	X	X	X	X
Partner Training	X	X	X	
Partners	X	X	X	X
Pipeline	X	X	X	X
Pricing	X	X	X	X
Products	X	X	X	X
Real-Time Activity				
Real-Time Assets				
Service Requests	X	X	X	

**7.5.2.14 PRM Analytics Subject Area Visibility** The subject area visibility for roles in Partner Portal is shown in [Table 7-5](#), where a X indicates that subject area is visible for the user holding that responsibility.

**Table 7-5 Subject Area Visibility for PRM Partner Portal**

Subject Area	Partner Executive Analytics User	Partner Operations Manager Analytics User	Partner Sales Manager Analytics User	Partner Sales Rep Analytics User	Partner Service Manager Analytics User	Partner Service Rep Analytics User
Activities	X	X	X	X	X	X
Assets	X	X			X	X
Campaigns	X	X				
Consumers	X	X				
Customer Satisfaction	X	X			X	X
Customers	X	X	X	X	X	X
Orders	X	X	X	X	X	X
Partner Training	X	X	X	X	X	X
Partners	X	X				
Pipeline	X	X	X	X		
Pricing						

**Table 7–5 (Cont.) Subject Area Visibility for PRM Partner Portal**

Subject Area	Partner Executive Analytics User	Partner Operations Manager Analytics User	Partner Sales Manager Analytics User	Partner Sales Rep Analytics User	Partner Service Manager Analytics User	Partner Service Rep Analytics User
Products	X	X	X	X	X	X
Real-Time Activity						
Real-Time Assets						
Service Requests	X	X			X	X

**7.5.2.1.5 PRM Analytics Data-Level Visibility** Oracle PRM Analytics also provides brand owners the ability to restrict security based on the user's organization or position. This security mechanism makes sure that one user does not have access to another user's data. It also makes sure that one partner does not have access to another partner's data. Data-level security is administered for responsibilities. Details regarding setting up data-level visibility are provided in the topic [Section 7.6.2, "Implementing Data-Level Security in the Oracle BI Applications Repository"](#). To change Partner Service Representative and Sales Representative to Organization-based security, follow the procedure shown in the topic [Section 7.5.4, "Embedded Oracle Business Intelligence support for Changing Positions"](#).

[Table 7–6](#) shows the data-level security settings included for the responsibilities in Partner Manager and Partner Portal.

**Table 7–6 Oracle PRM Data-Level Security Settings**

Responsibility	Data-Level Security	Type	Comments
Channel Executive Analytics User	No	N/A	N/A
Channel Operations Analytics User	No	N/A	N/A
Channel Account Manager Analytics User	No	N/A	N/A
Channel Marketing Manager Analytics User	No	N/A	N/A
Partner Executive Analytics User	Yes	Organization	Displayed records should match organization of the user.
Partner Sales Manager Analytics User	Yes	Organization	Displayed records should match organization of the user.
Partner Sales Rep Analytics User	Yes	Position	Displayed records should match position of the user.
Partner Service Manager Analytics User	Yes	Organization	Displayed records should match organization of the user.
Partner Service Rep Analytics User	Yes	Position	Displayed records should match position of the user.

### 7.5.2.2 Consumer Sector Analytics Security Settings

Table 7-7 describes the consumer sector responsibilities associated with each CS Dashboard.

**Table 7-7 Consumer Sector Responsibilities Associated with Each CS Dashboard**

Responsibility	Dashboard	Pages
VP Sales	VP Sales	Business Overview, Product Overview
	Sales Performance	Sales Volume Planning, Hierarchy, Trends, Growth
	Promotion	Plan Year to Date, Corporate
Key Account Manager	Key Account Manager	Business, Category
	Promotion	Plan year to date, Key account
	Funds	Account
	Retail Audit	Last audit, Trends
	Sales Performance	Sales Volume Planning, Hierarchy, Trends, Growth

### 7.5.2.3 Communications, Media, and Energy (CME) Analytics Security Settings

Oracle's CME family of products (Oracle Communications, Media and Energy Sales Analytics, Oracle Communications, Media and Energy Service Analytics, Oracle Communications, Media and Energy Marketing Analytics) applies the Oracle's Siebel operational applications security model; that is, it uses Oracle's Siebel operational applications responsibilities (and corresponding repository and Presentation Services groups) for controlling access to Oracle's Siebel operational applications objects (both metadata and Presentation Services objects). This security model is described in the topic [Section 7.1, "Types of Security in Oracle Business Intelligence Applications"](#).

In addition to responsibilities provided by the operational applications, Oracle Communications, Media, and Energy (CME) provides additional responsibilities, and responsibility-specific security, as indicated in [Table 7-8](#).

**Table 7-8 CME Responsibilities Associated with Each CME Dashboard**

CME Responsibility	CME Dashboard	Dashboard Pages
CM Marketing Analytics User	Loyalty Management	Customer Lifetime Value
		Churn Propensity
		Selling Propensity
		Financial Risk
CM Marketing Analytics Administrator		Actual Churn
		Sales Portal
		Service Activations
CM Sales Analytics User	Revenue Management	Service Modifications
		Service Disconnections
CM Sales Analytics Administrator		

**Table 7–8 (Cont.) CME Responsibilities Associated with Each CME Dashboard**

<b>CME Responsibility</b>	<b>CME Dashboard</b>	<b>Dashboard Pages</b>
	Account Management	<ul style="list-style-type: none"> <li>■ Sales Portal</li> <li>■ Service Activations</li> <li>■ Service Modifications</li> <li>■ Service Disconnections</li> </ul>
CM Service Analytics User	Account Management	<ul style="list-style-type: none"> <li>■ Trouble Tickets</li> <li>■ Customer Satisfaction</li> </ul>
CM Service Analytics Administrator		

#### 7.5.2.4 Financial Services Analytics Security Settings

The following applications apply Oracle's Siebel operational applications security model:

- The Financial Analytics family of products (Finance Sales Analytics, Finance Service Analytics, Finance Marketing Analytics, Finance Institutional Analytics, Finance Retail Analytics).
- The Insurance Analytics family of products (Oracle Insurance Partner Manager Analytics, Oracle Insurance Sales Analytics, Oracle Insurance Service Analytics, Oracle Insurance Marketing Analytics, Oracle Insurance Partner Manager Analytics).
- The Healthcare Analytics family of products (Oracle Healthcare Sales Analytics, Oracle Healthcare Service Analytics, Oracle Healthcare Marketing Analytics, Oracle Healthcare Partner Manager Analytics).

In addition to responsibilities provided by the Siebel operational applications, these applications provide additional responsibilities, and responsibility-specific security, as indicated in [Table 7–9](#).

For the Financial Services products, the Oracle's Siebel operational applications security model has been extended in the following ways:

- **Financial Analytics user**  
A finance-specific responsibility (and corresponding repository and Presentation Services group) that must be used in conjunction with Oracle's Siebel operational applications responsibilities and groups to control access to Finance-specific objects in Financial Analytics.
- **A user in the Oracle Insurance Analytics family of products (Oracle Insurance Partner Manager Analytics, Oracle Insurance Sales Analytics, Oracle Insurance Service Analytics, Oracle Insurance Marketing Analytics, Oracle Insurance Partner Manager Analytics)**  
An insurance-specific responsibility (and corresponding repository and Presentation Services group) that must be used to control access to the Insurance and Healthcare-specific objects in Insurance and the Healthcare Analytics family of products (Oracle Healthcare Sales Analytics, Oracle Healthcare Service Analytics, Oracle Healthcare Marketing Analytics, Oracle Healthcare Partner Manager Analytics).

For example, when you give a salesperson all horizontal Sales responsibilities and also include the finance responsibility Financial Analytics User, this user is able to see, in

addition to all horizontal sales objects (Dashboards, Subject Areas, folders in the presentation Layer, and so on), all finance-specific Sales objects. Similarly, in order to see Insurance and Healthcare-specific objects, you need to add one of the Oracle Insurance Analytics family of products (Oracle Insurance Partner Manager Analytics, Oracle Insurance Sales Analytics, Oracle Insurance Service Analytics, Oracle Insurance Marketing Analytics, Oracle Insurance Partner Manager Analytics) user responsibilities to this user.

**7.5.2.4.1 Parent and Child Group Behavior** Oracle Business Intelligence supports hierarchies in the repository groups, and certain groups within the Oracle Business Intelligence repository are parent groups that define the behavior of all the child groups. For Financial Services Analytics, the parent groups are the following:

- **Finance**  
Parent group for all Financial applications groups. Financial Analytics User is a child group of Finance group.
- **Insurance**  
Parent group for all Insurance applications groups. Insurance Analytics User is a child group of Insurance group.

Inheritance is used to let permissions ripple through to child groups. The parent groups for Financial Services and their purpose are shown in [Table 7-9](#).

---

**Note:** A Financial Services Analytics user is provided as a child to both Finance and Insurance. Therefore this user has permissions available to both Finance and Insurance. If you have purchased both Financial Analytics and one of the Oracle Insurance Analytics family of products (Oracle Insurance Partner Manager Analytics, Oracle Insurance Sales Analytics, Oracle Insurance Service Analytics, Oracle Insurance Marketing Analytics, Oracle Insurance Partner Manager Analytics), you should use the Financial Services Analytics user responsibilities to view all relevant dashboards.

---

[Table 7-9](#) shows the additional responsibilities, and responsibility-specific security in Oracle's Financial Analytics family of products (Finance Sales Analytics, Finance Service Analytics, Finance Marketing Analytics, Finance Institutional Analytics, Finance Retail Analytics), the Oracle Insurance Analytics family of products (Oracle Insurance Partner Manager Analytics, Oracle Insurance Sales Analytics, Oracle Insurance Service Analytics, Oracle Insurance Marketing Analytics, Oracle Insurance Partner Manager Analytics), and the Healthcare Analytics family of products (Oracle Healthcare Sales Analytics, Oracle Healthcare Service Analytics, Oracle Healthcare Marketing Analytics, Oracle Healthcare Partner Manager Analytics).

If you are also deploying Usage Accelerator, Financial Services-specific Usage Accelerator responsibilities are shown in [Table 7-11](#).

**Table 7-9 Financial Services Responsibility Required to View FS Dashboards**

FS Responsibilities	Dashboards
Financial Analytics User	Credit
	Credit Card
	Private Banking
	Consumer Banking

**Table 7–9 (Cont.) Financial Services Responsibility Required to View FS Dashboards**

<b>FS Responsibilities</b>	<b>Dashboards</b>
User in one of the Oracle Insurance Analytics family of products (Oracle Insurance Partner Manager Analytics, Oracle Insurance Sales Analytics, Oracle Insurance Service Analytics, Oracle Insurance Marketing Analytics, Oracle Insurance Partner Manager Analytics)	Corporate and Commercial Banking
	Investment Holdings
	Separate Account Management
	Wealth Management
	Institutional Sales
	Investment Banking
	Finance Marketing
	Finance Executive
	Policy Sales
	Policy Service
	Insurance Marketing
	Insurance Executive
	Insurance Claims
	Health Plan Sales
	Health Plan Service
	Health Plan Marketing
Health Plan Executive	
Insurance Agents / Partners	

### 7.5.2.5 Oracle Pharma Sales Analytics and Oracle Pharma Marketing Analytics Security Settings

Data-level security in Oracle Pharma Sales Analytics and Oracle Pharma Marketing Analytics is based on Siebel Position ID for all Pharma Analytics responsibilities except PH Executive Analytics. Siebel Position ID is always resolved through the fact table.

Data visibility is unconstrained for administrative roles. For other roles, data visibility is controlled by Position ID. The Oracle Business Analytics Warehouse uses table W\_POSITION\_DH for user position based security control. address not being available. A user sees only the data that are available to that user's positions. This security model is enforced for all queries, with the exception of queries that deal exclusively with dimension data only, such as:

- Time Period
- Product
- Invitee Status

Table 7–10 shows Pharma Analytics Responsibilities and Functions.

**Table 7–10 Pharma Analytics Responsibilities and Functions**

<b>Responsibility</b>	<b>Use</b>
LS Administrator	Administrator privileges to all options on Pharma Analytics.
PH Call Activity Analytics Admin	Administrator privileges to Call Activity Analytics option.
PH EMEA Call Activity Analytics User	Enables brick-based metrics to be used in the Presentation Services for Pharma subject areas.  Note that in the 7.7 Analytics Release, all report columns use position-based hierarchies, where in earlier releases, report columns used alignment-based sales hierarchies. All brick-based alignment pages have been removed from the reports. Therefore, if you want to use brick-based position hierarchies, you must reconfigure the reports to maintain the alternate hierarchy.
PH EMEA Executive Analytics User	Enables brick-based metrics to be used in the Presentation Services for Pharma subject areas.  Note that in the 7.7 Analytics Release, all report columns use position-based hierarchies, where in earlier releases, report columns used alignment-based sales hierarchies. All brick-based alignment pages have been removed from the reports. Therefore, if you want to use brick-based position hierarchies, you must reconfigure the reports to maintain the alternate hierarchy.
PH EMEA Marketing Analytics User	Enables brick-based metrics to be used in the Presentation Services for Pharma subject areas.  Note that in the 7.7 Analytics Release, all report columns use position-based hierarchies, where in earlier releases, report columns used alignment-based sales hierarchies. All brick-based alignment pages have been removed from the reports. Therefore, if you want to use brick-based position hierarchies, you must reconfigure the reports to maintain the alternate hierarchy.
PH EMEA Sales Analytics User	Enables brick-based metrics to be used in the Presentation Services for Pharma subject areas.  Note that in the 7.7 Analytics Release, all report columns use position-based hierarchies, where in earlier releases, report columns used alignment-based sales hierarchies. All brick-based alignment pages have been removed from the reports. Therefore, if you want to use brick-based position hierarchies, you must reconfigure the reports to maintain the alternate hierarchy.
PH Executive Analytics Admin	Unrestricted access to all Pharma Analytics options with ZIP territories.
PH Marketing Analytics Administrator	Administrator privileges to Pharma ROI, Call Activity Profit & Loss Report, Pharma Promotional Effectiveness Subject Area, and Medical Education Effectiveness Subject Area.
PH Medical Education Analytics Admin	Administrator privileges to Medical Education Analytics option.
PH Medical Education Analytics User	Enables access to Medical Education Analytics option.
PH Disconnected Analytics Admin	Administrator privileges to the PH Disconnected Manager Analytics User and Sales Rep Analytics Dashboards.
PH Disconnected Analytics User	Enables the Pharma Disconnected Analytics Home Page. Allows access to Sales Rep Dashboard as part of the Sales Rep Analytics option.

**Table 7–10 (Cont.) Pharma Analytics Responsibilities and Functions**

<b>Responsibility</b>	<b>Use</b>
PH Disconnected Manager Analytics Admin	Administrator privilege to the PH Disconnected Manager Analytics User and District Manager Analytics Dashboards.
PH Disconnected Manager Analytics User	Enables the Pharma Disconnected Analytics Home Page. Allows access to the District Manager Dashboard as part of the Sales Rep Analytics option.
PH Sales Analytics Administrator	Administrator privileges to Rx Sales Analytics option.
PH US Call Activity Analytics User	Enables access to Call Activity Analytics Option for ZIP territory alignments.
PH US Executive Analytics User	Unrestricted access to all Pharma Disconnected Analytics options with ZIP-based territories.

### 7.5.2.6 Usage Accelerator Analytics Security Settings

Table 7–11 describes the additional security configurations that may be necessary and the particular responsibilities associated with the Oracle Usage Accelerator application dashboards.

**Table 7–11 Usage Accelerator Responsibilities and Dashboards**

<b>User Responsibility</b>	<b>Data Level Security</b>	<b>Dashboard Name (View)</b>	<b>Dashboard Page</b>
Usage Accelerator–Sales Rep	Primary Position Data Level Security	Score Card	Individual Scorecard
Usage Accelerator–Financial Services Sales Rep		Action Plan	Account Coverage Contact Coverage Opportunity Coverage Financial Account Coverage— Financial Services only Account Completeness Contact Completeness Opportunity Updates
Usage Accelerator–Sales Manager	No position-based Security	Score Card	Team Scorecard Individual Scorecard
		Action Plan	Account Coverage (Team) Contact Coverage (Team) Opportunity Coverage (Team) Financial Account Coverage (Team)— Financial Services only Account Completeness (Team) Contact Completeness (Team) Opportunity Updates (Team)
Usage Accelerator–Financial Services Sales Manager		Coverage	Account Coverage Account Coverage (Team) Contact Coverage Opportunity Coverage Financial Account Coverage— Financial Services only
		Completeness	Account Completeness Contact Completeness

**Table 7–11 (Cont.) Usage Accelerator Responsibilities and Dashboards**

User Responsibility	Data Level Security	Dashboard Name (View)	Dashboard Page
		Opportunity Updates	Opportunity Updates
		User Adoption	Active Users Application Usage—excluded for Financial Services Application Usage—Financial Services only*
Usage Accelerator–Sales Executive	No position-based Security	Scorecard	Organization Scorecard Individual Scorecard
Usage Accelerator–Financial Services Sales Executive		Action Plan	Account Coverage (Org) Contact Coverage (Org) Opportunity Coverage (Org) Financial Account Coverage (Org)—Financial Services only Account Completeness (Org) Contact Completeness (Org) Opportunity Updates (Org)
Usage Accelerator–Administrator		Coverage	Account Coverage Contact Coverage Opportunity Coverage Financial Account Coverage—Financial Services only
Usage Accelerator–Financial Services Administrator		Completeness	Account Completeness Contact Completeness
		Opportunity Updates	Opportunity Updates
		User Adoption	Active Users Application Usage (excluded for Financial Services) Application Usage—Financial Services only
			Although the Application Usage dashboard names appears similar, the Application Usage—Financial Services-only version of this dashboard is different from the other dashboard.

### 7.5.3 Team Based Visibility

This section covers team based visibility and contains the following sections:

- [Section 7.5.3.1, "Summary"](#)
- [Section 7.5.3.2, "Business Requirements Overview"](#)
- [Section 7.5.3.3, "Design Details"](#)
- [Section 7.5.3.4, "Data Model"](#)
- [Section 7.5.3.5, "Fact Security"](#)
- [Section 7.5.3.6, "ETL Design"](#)

- [Section 7.5.3.7, "DAC Metadata"](#)
- [Section 7.5.3.8, "Oracle BI Applications Repository Metadata"](#)

### 7.5.3.1 Summary

Team based visibility will enable the Oracle Business Intelligence users to have access to all records of an entity (for example, opportunity) based on the team that they or their subordinates are part of. This is very similar to the team visibility implemented in Siebel transactional application.

This section uses Opportunity as an example entity to explain how team based visibility can be implemented. The same concept can be extended to other entities.

---



---

**Note:** This type of data security is not supported in the metadata that is installed out-of-the-box. Oracle BI Applications provides only some of the required ETL mappings in the Informatica repository, as described in this section. The examples in this section should provide enough information to enable you to code the missing pieces and add them to your application.

---



---

### 7.5.3.2 Business Requirements Overview

In Siebel transactional application, "my team" view allows users to see those records of which they are members of the record's team. It has been noted that similar view is needed for Oracle Business Intelligence users when they query the Oracle BI Application Data Warehouse. In this visibility mode, the query should return records for which the user (or people reporting to the user) is part of the record's team.

Oracle BI Applications administrator will create special group for team-based visibility that will contain appropriate filters on select entities. The Administrator will assign users to this group who need this visibility filter to be applied. This configuration will force the visibility filter to be applied to all queries issued by the users on associated entities, causing only the right set of rows to be returned to the user.

### 7.5.3.3 Design Details

The design of team-based visibility implementation needs special visibility tables to be populated in the Oracle Business Analytics Warehouse. These tables will be populated by special ETL mappings and used to define the filter condition for analytics queries.

In the following sections, Opportunity is used as an example entity for which team-based visibility needs to be implemented. The same concept can be extended to other entities.

### 7.5.3.4 Data Model

The W\_PARTY\_LOGIN table (included in the out-of-the-box application) decodes the position hierarchy information that is used for all entities. The table below describes the structure of the W\_PARTY\_LOGIN table.

**Table 7-12 Structure of W\_PARTY\_LOGIN table**

Columns	Description
PARTY_ID	The party_id (position) for the user.
LOGIN	The login_id for the parent position (including any level parent).

**Table 7–12 (Cont.) Structure of W\_PARTY\_LOGIN table**

Columns	Description
PARTY_TYPE	'P' for Position party type, 'O' for Organization party type, 'E' for Employee party type.

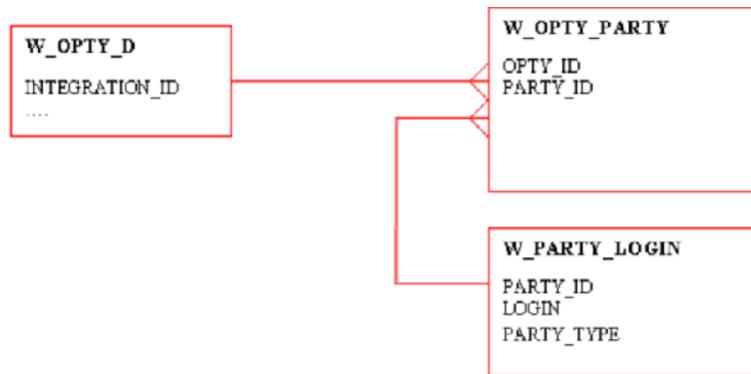
The transactional source for this table is S\_PARTY\_RPT\_REL. All possible pairs of position\_id and its parent's login at any level are selected from the source table during ETL and inserted in this table. The same table is being used to store hierarchy relationships for other party objects (for example, Organization). Therefore a party\_type column has been added to separate these different sets of records.

One table for each entity is needed to store the team members (Positions) associated with every object. The following example is for Opportunity. The table below describes the structure of the W\_OPTY\_PARTY table.

**Table 7–13 Structure of W\_OPTY\_PARTY table**

Columns	Description
OPTY_ID	The row_id for the opportunity.
PARTY_ID	The party_id (position) for the user associated to opportunity.

The transactional source for this table is S\_OPTY\_POSTN. The grain of this table is same as the grain of the source table. The relationship between these tables and the W\_OPTY\_D (the Oracle Business Analytics Warehouse table that requires visibility filtering) is shown in the following diagram.




---

**Note:** There is a M:M relationship between W\_OPTY\_D (the entity table which require visibility) and the W\_PARTY\_LOGIN table. Therefore a filter using the joins shown above will provide duplicate opportunity rows and cause errors in aggregation.

---

To get around this issue the party tables needs to be pre-joined and duplicate rows filtered. Both these are combined to provide a distinct set of (OPTY\_ID, LOGIN) in the table W\_OPTY\_LOGIN (not included in the out-of-the-box application). The table below describes the structure of the W\_OPTY\_LOGIN table.

**Table 7-14 Structure of W\_OPTY\_LOGIN table**

Columns	Description
OPTY_ID	The row_id for the opportunity.
LOGIN	The login id for the user who will have visibility access to the opportunity.

This is done using the following SQL:

```
SELECT W_PARTY_LOGIN.LOGIN, W_OPTY_PARTY.OPTY_ID
FROM W_PARTY_LOGIN, W_OPTY_PARTY
WHERE W_OPTY_PARTY.PARTY_ID = W_PARTY_LOGIN.PARTY_ID AND
W_PARTY_LOGIN.PARTY_TYPE = 'P' /*For only position records */
GROUP BY LOGIN, OPTY_ID
```

The relationship between the W\_OPTY\_D table and the W\_OPTY\_LOGIN table is a one-to-many relationship, and can be used for filtering. This is shown in the following diagram:



Similar intersection tables need to be created to support team visibility security for the other dimensions (or facts). Depending on your requirements, you must create one (or more) of the following tables, in a manner similar to W\_OPTY\_LOGIN.

- W\_ACT\_LOGIN (for Activity)
- W\_CON\_LOGIN (for Contacts)
- W\_HHOLD\_LOGIN (for Households)
- W\_ORDER\_LOGIN (for Orders)
- W\_ORG\_LOGIN (for Accounts)
- W\_PROG\_LOGIN (for Marketing Programs)
- W\_QUOTE\_LOGIN (for Quotes)
- W\_RESP\_LOGIN (for Responses)
- W\_SEG\_LOGIN (for Segments)
- W\_SR\_LOGIN (for Service Requests)

These tables should be created both in the DAC repository (under a custom container), and in the ETL repository, to be able to build the corresponding mappings.

### 7.5.3.5 Fact Security

It has been shown how to use the W\_OPTY\_LOGIN table to secure data in W\_OPTY\_D. If you want to secure the fact table W\_REVN\_F with team visibility, you can choose one of the following options:

- Define the filter on "Fact – Revenue", using the Opportunity Dimension (Dim – Opportunity).Login field. In this case, all Revenue reports will join to the Opportunity dimension, even if there is no field being selected from that table in the group by clause.
- Use W\_OPTY\_LOGIN to create a filter on the Revenue fact (Fact – Revenue) directly. But in this case you need to add an OPTY\_ID to W\_REVN\_F, so you can join directly to W\_OPTY\_LOGIN.

### 7.5.3.6 ETL Design

The sections covers ETL Design.

**7.5.3.6.1 Available Mappings** There will be a set of ETL routine to populate the tables mentioned in the data model section. In the standard application there are visibility mappings to populate the following tables:

Common party login table:

- W\_PARTY\_LOGIN

Entity Party tables:

- W\_OPTY\_PARTY (for Opportunity)
- W\_ACT\_PARTY (for Activity)
- W\_CON\_PARTY (for Contacts)
- W\_HHOLD\_PARTY (for Households)
- W\_ORDER\_PARTY (for Orders)
- W\_ORG\_PARTY (for Accounts)
- W\_PROG\_PARTY (for Marketing Programs)
- W\_QUOTE\_PARTY (for Quotes)
- W\_RESP\_PARTY (for Responses)
- W\_SEG\_PARTY (for Segments)
- W\_SR\_PARTY (for Service Requests)

The standard ETL mappings responsible for populating these tables, available in the Informatica repository, are:

- VisibilityAccountParty
- VisibilityActivityParty
- VisibilityContactParty
- VisibilityHouseHoldParty
- VisibilityOpportunityParty
- VisibilityOrderParty
- VisibilityProgramParty
- VisibilityQuoteParty
- VisibilityResponseParty
- VisibilitySegmentParty
- VisibilityServiceRequestParty

- VisibilityPartyLogin

It should be noted that the records in these tables need to be removed when a position is removed from a team. This has to be done by tracking deletes on the corresponding source table and using the deleted row information. Any removed rows need to be deleted by ETL process from the target tables.

The mappings to delete rows are available in the Informatica repository as follows:

- VisibilityAccountParty\_LoadDeletedRows
- VisibilityActivityParty\_LoadDeletedRows
- VisibilityContactParty\_LoadDeletedRows
- VisibilityHouseHoldParty\_LoadDeletedRows
- VisibilityOpportunityParty\_LoadDeletedRows
- VisibilityOrderParty\_LoadDeletedRows
- VisibilityProgramParty\_LoadDeletedRows
- VisibilityQuoteParty\_LoadDeletedRows
- VisibilityResponseParty\_LoadDeletedRows
- VisibilitySegmentParty\_LoadDeletedRows

The ETL mappings to populate W\_<entity>\_LOGIN do not exist in the standard Oracle Business Intelligence Applications and can be easily built using the SQL provided in the data model section.

**7.5.3.6.2 Creating Delete Triggers** Some Image tables need to be updated automatically when the OLTP data is changed. For this purpose, you need to enable some triggers as follows:

- In the Applications Data Warehouse, preconfigured visibility tables are activated by default. If your organization is not going to use any of the visibility-related data warehouse tables, you need to inactivate them in the Tables/Related Tables screen of the Design view on the DAC client. If you leave them as activate because you are planning to enable team visibility, you should also create delete triggers on the optional tables.
- When creating delete triggers using the Data Warehouse Configurator, you have the option of including the optional tables for which you want to enable the delete triggers. You also have the option of having the trigger statements be directly executed or written to a file, which can then be executed by a database administrator.

To create a delete trigger:

1. From the DAC menu bar, click Tools > ETL Management Configure.
2. Select the Create Delete Triggers in Transaction Database check box.  
The Delete Triggers tab is now active.
3. Select Create Triggers.
4. Select the transactional database where the triggers will be created.
5. (Optional) Select the Include optional triggers check box to include optional tables.
6. Click Start.

**7.5.3.6.3 DAC Metadata** All the Visibility tables and corresponding tasks mentioned above have already been imported into the DAC metadata. All tasks responsible for synchronizing the tables with OLTP tables (in case of Deletion) are also automatically included. Each of these tables (like W\_OPTY\_LOGIN) is listed as a Related table for the corresponding Fact table (W\_OPTY\_D in this case). If you are adding a fact to your Subject area, these tasks will be included in your Execution plan automatically. For the additional intersection tables (W\_OPTY\_LOGIN etc.), assuming that you already created them in DAC, you need to do the following steps to ensure they will be added to your execution plan, and populated during ETL:

1. Import the ETL workflows responsible for populating them to the Task list.
2. Add every table as a Related table to the corresponding fact or dimension.

### **7.5.3.7 DAC Metadata**

All the Visibility tables and corresponding tasks mentioned above have been already imported into the DAC metadata. All tasks responsible for synchronizing the tables with the OLTP (in case of Deletion.) are also automatically included. Each of these tables (like W\_OPTY\_LOGIN) is listed as a Related table for the corresponding Fact table (W\_OPTY\_D in this case). If you are adding a fact to your Subject area, these tasks will be included in your Execution plan automatically. For the additional intersection tables (W\_OPTY\_LOGIN,...), assuming that you already created them in DAC, you need to do the following steps to ensure they will be added to your execution plan, and populated during ETL:

1. Import the ETL workflows responsible for populating them to the Task list.
2. Add every table as a Related table to the corresponding fact or dimension.

### **7.5.3.8 Oracle BI Applications Repository Metadata**

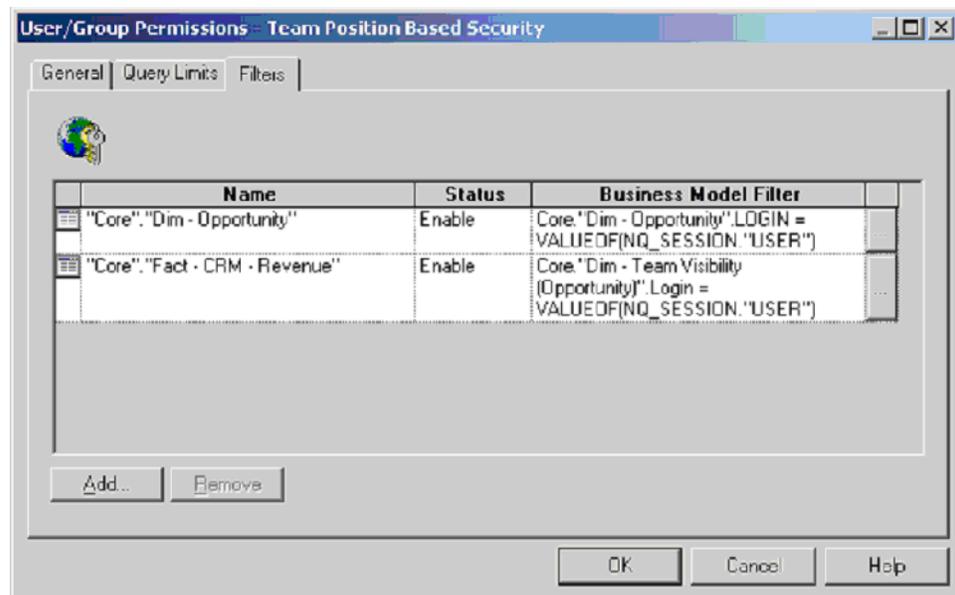
In the metadata that is provided with the Oracle Business Intelligence Applications, Primary Position based Security security is implemented using the security group: Primary Employee/Position Hierarchy-based Security. In order to apply this type of security to a user, you need to add at least one of the user's responsibilities (in OLTP) to this group. This is different from what was done in Oracle BI Applications 7.9.0, 7.9.1 and 7.9.2 releases and from what is described in the Oracle Business Intelligence Applications Installation and Configuration Guide version 7.9 > Integrated Security for Oracle BI Applications. This security group is still based on the Primary Position owner of the record (Fact or Dimension), and the Position hierarchy stored in W\_POSITION\_DH.

The team-based security is based on the W\_<entity>\_LOGIN tables that have been described above. To enable Team-based security, you need to add some metadata elements, and change some of the standard elements.

To enable team-based security:

1. In the Physical Layer:
  - a. Import the new W\_<entity>\_LOGIN table in physical layer.
  - b. Join the table W\_<entity>\_LOGIN to the corresponding dimension or fact. For example, join W\_OPTY\_LOGIN to W\_OPTY\_D.
2. In the Logical Layer:
  - a. If the table to be secured is a Dimension, add the W\_<entity>\_Login table to the logical table source for the corresponding dimension.

- b. If the table to be secured is a Fact, create a new Logical dimension based on the W\_<entity>\_LOGIN table, for example, Dim – Team Security (Opportunity).
  - c. Expose the LOGIN column in the logical layer as a logical column.
3. Security Filters:
- a. Create a new user group for team-based security (for example, Team Position Based Security).
  - b. Users to whom you want to apply team-based visibility security filters need to have at least one of their responsibilities as part of this group. You can add a responsibility to a group by double clicking on the group name, and then clicking the Add/Remove button.
  - c. Create filters on the respective fact or dimension tables. The screenshot below shows the Filters tab of the User/Group Positions dialog, with filters specified for both fact and dimension tables.



- d. Users who are members of this Team Visibility group should not be members of the existing Primary Employee/Position Hierarchy-based Security. The filter in this latter group is more restrictive and it will not add anything to the data set, since the two filters will be grouped with an OR condition.

## 7.5.4 Embedded Oracle Business Intelligence support for Changing Positions

In the Siebel CRM Application, a user can assume a new Position during the same session by clicking on the button: Change Position in User Preferences/Change Position screen. This screen allows the user to choose from a set of Positions already associated with him through Employee/Position screen. Currently, when the user changes positions in the OLTP application, the data in the embedded Oracle Business Intelligence reports is still filtered by his Primary Position, and not the new position he selected.

This section describes a solution that allows users to connect seamlessly to Oracle Business Intelligence dashboards from an OLTP application using the new Position, and see all the data available for the impersonated position. The data security in this case will filter on the Login of the Primary Owner for the new Position.

This enhancement requires the following steps to be implemented:

1. Configure the RunAs option in the Oracle Business Intelligence Enterprise Edition environment.

This step is explained in Chapter 8 of *Oracle Business Intelligence Presentation Services Administration Guide*, in the section: About Authorizing Users to Act for Others. This document can be obtained at the following address:

[http://www.oracle.com/technology/documentation/bi\\_ee.html](http://www.oracle.com/technology/documentation/bi_ee.html)

This feature allows a user to impersonate other Oracle Business Intelligence users, based on a table that lists the Proxy (the impersonating user) and the target user(s) with their login. This information can be stored in a physical table, or in a view populated dynamically using the association between Positions in S\_PARTY\_PER. You can find a definition of this view in the installation directory under (`\\installdirectory\EmbeddedAnalytics\Proxy_View.sql`).

2. Configure the Siebel CRM Application to communicate the new Position to the Oracle BI server.

This step requires adding two new functions to the Siebel CRM Application you are running (like Universal Agent, Marketing, etc.). The purpose of these functions is:

- a. Capture the new Position selected by the Change Position button.
- b. Find the login of the Primary Employee of the new position.
- c. Pass this login to a new session variable ActPosLogin.

The sif files containing these records are available in your installation directory. You should import these sif files to your Siebel Repository (using Siebel Tools), then recompile the corresponding projects into your srf file. The sif files are available for eScript (`\\installdirectory\EmbeddedAnalytics\Escript`) and VB Script (`\\installdirectory\EmbeddedAnalytics\VB Script`). You can use either one of them, based on your preference. If the application you are using is not Siebel Universal Agent, you can create a similar record directly in Siebel Tools for your own application. You need to do the following steps:

- a. Go to Siebel Tools/Detail/Application/<your application> /Application Server Scripts
  - b. Create a new script called Application\_Start as defined in the sample files available in `\\installdirectory\EmbeddedAnalytics` (under Escript and VB Script directories.)
3. Add a new Symbolic URL parameter to the existing embedded dashboards.

The last step is to add a new parameter under Symbolic URL screen for each of the embedded dashboards. The new parameter is called RUNAS.

- a. In the OLTP application, go to Integration Administration/Symbolic URL.
- b. Choose Symbolic URL Administration.
- c. Search for the dashboard you want to support with this new functionality.
- d. Under Parameters, add a new parameter called RUNAS as follows:
  - i. ArgType: Profile Attribute
  - ii. ArgumentValue: ActPosLogin

The RUNAS parameters is expected by the Proxy feature described in the first paragraph, and the ActPosLogin is the new variable added and populated by the new metadata added to the Siebel Repository in the second paragraph.

## 7.5.5 Business Unit based Security

The Business Unit security is supported through the Security group named Primary Org-Based Security. When installed out-of-the-box, only a few dimensions in the Core, Workforce Analytics and Forecasting business models are supporting this data security type. Other facts and dimensions can be added to this security group if they have the column VIS\_PR\_BU\_ID column populated.

The security filter in this security group is defined as:

```
"Core"."Dim - Order"."VIS_PR_BU_ID" = VALUEOF(NQ_SESSION."ORGANIZATION")
```

The session variable ORGANIZATION is a Row-wise variable, initialized using the Initialization block: Orgs for Org-Based Security. This Init Block runs the following SQL against the Siebel OLTP data source, to populate the ORGANIZATION variable:

```
select distinct 'ORGANIZATION', PRR.SUB_PARTY_ID
from VALUEOF(TBO).S_POSTN P, VALUEOF(TBO).S_USER U,
VALUEOF(TBO).S_PARTY_PER PP, VALUEOF(TBO).S_PARTY_RPT_REL PRR
where U.ROW_ID=PP.PERSON_ID and P.ROW_ID=PP.PARTY_ID and
PRR.PARTY_ID = P.BU_ID and PRR.PARTY_TYPE_CD = 'Organization'
and U.LOGIN = ':USER'
```

## 7.5.6 Primary Owner Security

Primary Owner security is supported through the security group named Primary Owner-Based Security. This type of security mechanism allows records to be visible only to their Primary Owner. When installed out-of-the-box, this type of security supports a few dimensions in the Core business model, but other tables can be added if they have a primary owner's source Integration Id column.

The security filter in this security group is defined as:

```
"Core"."Dim - Activity"."VIS_PR_OWN_ID" = VALUEOF(NQ_SESSION."PR_OWNER_ID")
```

The session variable PR\_OWNER\_ID is a single value variable, populated by the Initialization block: Primary Owner ID. This Init Block runs the following sql against the Siebel OLTP data source, to populate the variable:

```
select PAR_ROW_ID
from VALUEOF(TBO).S_USER
where LOGIN = ':USER'
```

## 7.6 Data-Level Security In Oracle BI Applications

This section describes the data-level security features in Oracle BI Applications, and contains the following topics:

- [Section 7.6.1, "Data-Level Security in Oracle Business Intelligence Applications"](#)
- [Section 7.6.2, "Implementing Data-Level Security in the Oracle BI Applications Repository"](#)

- [Section 7.6.3, "Initialization Blocks Used For Security Oracle BI Applications"](#)

## 7.6.1 Data-Level Security in Oracle Business Intelligence Applications

Data-level security defines what a user in an OLTP application (for example, Oracle E-Business Suite, Siebel (CRM), PeopleSoft) can access inside a report. The same report, when run by two different users, can bring up different data. This is similar to how the My Opportunities view in an operational application displays different data for different users. However, the structure of the report is the same for all users, (unless a user does not have access to a column in a report, in which case the column is not displayed for that user). As another example, the Procurement Manager of Vision USA should only see the Procurement details that happened in USA (assuming that the organization has clearly demarcated their procurement organization on the basis of geographical territories). Similarly, the Procurement Manager of Asia-Pacific should see procurement data of Asia-Pacific Region only.

Even if both these persons belong to the "Procurement Manager" group, the actual data seen by them varies, depending on the access controls they have. This is a critical requirement and spans across all pillars (HCM, SCM, FII, CRM) etc.

### 7.6.1.1 Roadmap of Data Security in Oracle Business Intelligence Applications releases

Since the early Siebel Analytics releases, the solution supported the following types of data security:

- Primary Position-based security for Siebel CRM Applications.
- Primary Owner based security for Siebel CRM Applications.
- Business Unit based security for Siebel CRM Applications.

In release 7.9.0, Oracle BI Applications have introduced the following types of data security:

- Operating Unit Org-based security for Oracle EBS.
- Inventory Org-based security for Oracle EBS.
- The Position-based security for Siebel CRM Applications has been extended to support Slowly changing position hierarchies, and AS IS/AS WAS type of analysis.

In release 7.9.3, Oracle BI Applications have expanded the scope to:

- Company Org-based security for EBS, PeopleSoft Financials, and PeopleSoft HCM.
- Business Group Org-based security for EBS.
- Operating Unit Org-based security for PeopleSoft Financials, and PeopleSoft HCM.
- SET ID-based security for PeopleSoft Financials, and PeopleSoft HCM.

In Version 7.9.4, Oracle BI Applications replaced the Company Org-based security with:

- Ledger security for EBS, PeopleSoft Financials, and PeopleSoft HCM.

The following data security type is not supported, but will be included in future releases:

- GL flex field (for EBS).

Here is the summary chart of these security groups and different source applications they support in different Oracle Business Intelligence Enterprise Edition Application releases. During installations and configuration, customers have to make sure the correct security group and initialization blocks are set up for different sources and different releases. A security group, which is out of the source-and-release matrix as below, should be disabled as well as its corresponding initialization blocks. The details of security group and its initialization blocks are addressed in parts after this chart.

**Table 7–15 Summary of security groups and different source applications supported in Oracle Business Intelligence Applications releases**

	Oracle EBS	PeopleSoft Financials	PeopleSoft HR	Siebel
Operating Unit Org-based Security	<i>Available since 7.9</i>	Available since 7.9		<i>Available since 7.5</i>
Inventory Org-based Security	Available since 7.9			
Company Org-based Security	<i>Available in 7.9.3 and obsolete in 7.9.4</i>	<i>Available in 7.9.3</i>	<i>Available in 7.9.3</i>	
Business Group Org-based Security	<i>Available since 7.9.3</i>			
HR Org-based Security			<i>Available since 7.9.3</i>	
Payables Org-based Security		<i>Available since 7.9.3</i>		
Receivables Org-based Security		<i>Available since 7.9.3</i>		
SET ID-based security		<i>Available since 7.9.3</i>	<i>Available since 7.9.3</i>	
Position-based security	<i>Available since 7.9.4 for HRMS</i>		<i>Available since 7.9.3</i>	<i>Available since 7.5</i>
Ledger-based security	<i>Available since 7.9.4</i>	<i>Available since 7.9.4</i>		

## 7.6.2 Implementing Data-Level Security in the Oracle BI Applications Repository

Data-level security in Oracle BI Applications is implemented in three major steps:

1. Set up initialization blocks that obtain specific security-related information when a user logs in—for example, the user's Hierarchy Level in the Organization hierarchy, or the User's responsibilities.  
See [Section 7.6.3, "Initialization Blocks Used For Security Oracle BI Applications"](#).
2. Set up the joins to the appropriate security tables in the metadata physical and logical layers.
3. Set up the filters for each security group on each logical table that needs to be secured.

See [Section 7.5.1, "Primary Position based Security"](#) for Employee/Position Based security, and [Section 7.2, "Integrating Data Security with Oracle E-Business Suite"](#) and [Section 7.3, "Integrating Data Security with Oracle's PeopleSoft Enterprise Applications"](#) for Organization based security.

### 7.6.3 Initialization Blocks Used For Security Oracle BI Applications

In the Oracle Business Intelligence repository, the initialization blocks are set up for obtaining a given user's primary position, primary organization, and the owner ID, as described below:

- **Authorization**

This initialization block is used to associate users with all Security groups to which they belong. It obtains a user's responsibilities or roles from the source OLTP application, matches them with Oracle BI applications security groups, and determine the user's applicable object security during the session. This initialization block populates a variable set called GROUP.

- **Business Groups**

This initialization block is used to retrieve the business groups from OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called BUSINESS\_GROUP, which is used to drive security permissions for Business Group Org-based security.

- **Companies**

This initialization block is used to retrieve the companies from OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called COMPANY, which is used to drive security permissions for Company Org-based security.

- **HR Organizations**

This initialization block is used to retrieve the HR organizations from OLTP application to which the corresponding login user has access. This initialization block populates a variable set called HR\_ORG, which is used to drive security permissions for HR Analysts.

- **Inventory Organizations**

This initialization block is used to retrieve the inventory organizations from OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called INV\_ORG, which is used to drive security permissions for Inventory Org-based security.

- **Ledgers**

This initialization block is used to retrieve the ledgers from the OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called LEDGER, which is used to drive security permissions for Ledger-based security.

- **Operating Unit Organizations**

This initialization block is used to retrieve the operating unit organizations from OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called OU\_ORG, which is used to drive security permissions for Operating Unit Org-based security.

- **Orgs for Org-Based Security**

This initialization block is used to retrieve the Organizations reporting to the current user's Business Unit, from the Siebel CRM OLTP application.

This initialization block populates a variable set called ORGANIZATION, which is used to drive Primary Org-Based Security.

- **Payable Organizations**

This initialization block is used to retrieve the payable organizations from OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called PAYABLE\_ORG, which is used to drive security permissions for Payable Org-based security.

- **Primary Owner ID**

This initialization block obtains the owner ID for the given user. It obtains this information from the Siebel OLTP and populates the PR\_OWNER\_ID variable.

- **Receivable Organizations**

This initialization block is used to retrieve the payable organizations from OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called PAYABLE\_ORG, which is used to drive security permissions for Payable Org-based security.

- **Set ID**

This initialization block is used to retrieve the Set IDs from OLTP application to which the corresponding login responsibility has access. This initialization block populates a variable set called SET\_ID, which is used to drive security permissions for Set ID-based security.

- **User Hierarchy Level**

This initialization block obtains the fixed hierarchy level of the given user, based on the user's login, from W\_POSITION\_DH. It populates the variable HIER\_LEVEL. The SQL used by the block is run against the data warehouse. Therefore it reflects the hierarchy level at the time of the last ETL run that populated this table (W\_POSITION\_DH).

- **User HR Organizations**

This initialization block is used to retrieve the current HR organization from OLTP application to which the current user belongs. This initialization block populates a variable called USER\_HR\_ORG.

### 7.6.3.1 Data Security groups in Oracle Business Intelligence Applications

The table below describes the security groups used in Oracle Business Intelligence Applications and the Application to which they apply. Some selected security groups, which share the same name as Responsibilities (for Siebel CRM and Oracle EBS applications) and Roles (for PeopleSoft application), have been added as members of some of these security groups. A user who has any of these responsibilities or roles in the Source application will be a member of the corresponding data security group automatically when he/she logs in to the Analytics application. Other security groups based on similar objects in the source applications can be added to the Analytics repository and added to these data security groups, if you need the corresponding data filters to apply to any additional group of users.

**Table 7–16 Data Security Groups in Oracle BI Applications**

Security Group Name	Application	Description
Business Group Org-based Security	Oracle E-Business Suite	A Business Group is the highest level in the organization structure, and is usually used to represent the entire enterprise or a major division. A business group can have several sets of books.
Company Org-based Security	PeopleSoft	The Company Org-based Security filters data based on the General Ledger business units associated to the logged in user. The Business Unit is the highest level key structure in PeopleSoft.
HR Org-based Security	PeopleSoft	The HR Org-based Security filters data based on the HR business units associated to the logged in user. The Business Unit is the highest level key structure in PeopleSoft.
Human Resources Analyst	PeopleSoft	HR analysts have a special visibility requirement: They need to see all data for the Internal Organizations they are responsible for plus the data for their subordinates in their own organization. This is a special security group to address this requirement.
Operating Unit Org-based Security	Oracle E-Business Suite	Monthly and yearly fiscal reports are created by Operating Units which can belong to only one Legal Entity. Operating Units are secured by attaching a security profile to a user ID or responsibility. In turn, a security profile is associated with an organization hierarchy, which also has access to the user ID or responsibility. The Operating Unit Org-based security filters data based on the OUs associated to the logged in user.
Payable Org-based Security	PeopleSoft	The Payables Org-based Security filters data based on the Payables business units associated to the logged in user. The Business Unit is the highest level key structure in PeopleSoft.
Primary Employee/Position Hierarchy-based Security	Not applicable.	The Employee/Position based security allows only the record owner, and any Employee up in his hierarchy chain to see the record.
Primary Org-Based Security	Siebel CRM	Filters data based on the organization(s) associated to the logged in user.
Primary Owner-Based Security	Siebel CRM	Filters data based on the logged in user.
Primary Position-Based Security (Obsolete)	Not applicable.	Use the new Primary Employee/Position Hierarchy-based Security.
Receivables Org-based Security	PeopleSoft	The Receivables Org-based Security filters data based on the Receivables business units associated to the logged in user. The Business Unit is the highest level key structure in PeopleSoft.
SET ID-based Security	PeopleSoft	The SET ID-based Security filters data based on the set IDs associated to the logged in user.
Inventory Org-based Security	Oracle E-Business Suite	An Inventory Organization tracks inventory transactions and balances, and/or manufactures or distributes products or components. The Inventory Org-based security filters data based on the inventory orgs associated to the logged in user.

**Table 7–16 (Cont.) Data Security Groups in Oracle BI Applications**

<b>Security Group Name</b>	<b>Application</b>	<b>Description</b>
Ledger-based Security	Oracle E-Business Suite	A ledger is essentially a reporting organization that uses a common chart of accounts, functional currency, fiscal calendar, and accounting method. The Ledger-based security filters data based on the ledgers associated to the logged in user.



# Part IV

---

## Configuring Your Analytical Applications

---

Part IV covers the steps required to configure Oracle BI Applications, and contains the following chapters:

- Chapter 8, "Configuring Common Areas and Dimensions"
- Chapter 9, "Configuring Oracle's Supply Chain Family of Products"
- Chapter 10, "Configuring Oracle Financial Analytics"
- Chapter 11, "Configuring Oracle Order Management and Fulfillment Analytics"
- Chapter 12, "Configuring Oracle HR Analytics"
- Chapter 13, "Configuring Oracle Sales Analytics"
- Chapter 14, "Configuring Oracle Contact Center Telephony Analytics"
- Chapter 15, "Configuring the Oracle BI Applications Repository"

---

---

**Note:** For a high level road map for installation, configuration, and customization steps for Oracle BI Applications, see [Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle BI Applications"](#).

---

---



---

---

# Configuring Common Areas and Dimensions

This chapter contains configuration steps for Oracle Business Intelligence that you need to follow for any applications you deploy (for example, Oracle Financial Analytics, Oracle Human Resources).

This chapter contains the following topics:

- [Section 8.1, "Source Independent Configuration Steps"](#)
- [Section 8.2, "Oracle EBS-Specific Configuration Steps"](#)
- [Section 8.3, "PeopleSoft-Specific Configuration Steps"](#)
- [Section 8.4, "Siebel-Specific Configuration Steps"](#)

## 8.1 Source Independent Configuration Steps

This section contains configuration steps that apply to Oracle Business Intelligence Applications deployed with any source system.

This section contains the following topics:

- [Section 8.1.1, "Configuration Required Before A Full Load"](#)
- [Section 8.1.2, "Configuration Steps for Controlling Your Data Set for All Source Systems"](#)

### 8.1.1 Configuration Required Before A Full Load

This section contains configuration steps required before a full data load that apply to Oracle BI Applications deployed with any source system, and contains the following topics:

- [Section 8.1.1.1, "How to Configure Initial Extract Date"](#)
- [Section 8.1.1.2, "How to Configure Global Currencies"](#)
- [Section 8.1.1.3, "How to Configure Exchange Rate Types"](#)
- [Section 8.1.1.4, "How to Configure Fiscal Calendars"](#)

#### 8.1.1.1 How to Configure Initial Extract Date

TBD.

#### 8.1.1.2 How to Configure Global Currencies

Currency conversions are required because your business might have transactions involving multiple currencies. To create a meaningful report, you have to use a

common currency. The Oracle Business Analytics Warehouse stores amounts in the following currencies:

- Document currency. The currency of the transaction. For example, if you purchase a chair from a supplier in Mexico, the document currency is probably the Mexican peso. Or you made a business trip to UK and filed an expense report for meal expenses in UK, the document currency of the expense report will most likely be in GBP.
- Local currency. This is the base currency of your ledger, the currency in which your accounting entries are recorded in.
- Global currencies. Out of the box, Oracle BI Applications provides three global currencies, which are the common currencies used by the Data Warehouse. For example, if your organization is a multinational enterprise that has its headquarters in the United States, you probably want to choose US dollars (USD) as one of the three global currencies. The global currency is useful when creating enterprise-wide reports. For example, a user might want to view enterprise-wide data in other currencies. For every monetary amount extracted from the source, the load mapping loads the document and local amounts into the target table. It also loads the exchange rates required to convert the document amount into each of the three global currencies. In the target table, there will be two amount columns, and three exchange rate columns. In most cases, the source system provides the document currency amount. This is the most common situation, and thus is the Oracle Business Analytics Warehouse's default for handling currency. If the source system provides only the document currency amount, the Source Adapter performs lookups to identify the local currency codes based on the source system. Based on the source system the appropriate currencies are assigned. After the lookups occur, the extract mapping provides the load mapping with the document currency amount and the document and local currency codes. The load mapping will then use the provided local currency codes and perform currency conversion to derive the local amount. The load mapping will also fetch the global currencies setup from DAC parameters and look up the corresponding exchange rates to each of the three global currencies.

To configure the global currencies you want to report in your warehouse:

1. Log into DAC (for more information, see [Section 4.10.2, "How to Log Into the DAC"](#)).
2. Open your custom container
3. Display the 'Source System Parameters' tab.
4. Locate the following parameters and set the currency code values for them in the 'Value' box:
  - \$\$GLOBAL1\_CURR\_CODE (for the document currency).
  - \$\$GLOBAL2\_CURR\_CODE (for the local currency).
  - \$\$GLOBAL3\_CURR\_CODE (for the global currency).

Make sure you spell these currency as they are spelled in your source OLTP system.

5. Save your changes

### 8.1.1.3 How to Configure Exchange Rate Types

When Oracle BI Applications converts your transaction records' amount from document currency to global currencies, it also requires the exchange rate types to use

to perform the conversion. For each of the global currencies, Oracle BI Applications also allows you to specify the exchange rate type to use to perform the conversion. Oracle BI Applications also provides three global exchange rate types for you to configure.

Oracle BI Applications also converts your transaction records' amount from document currency to local currency. Local currencies are the base currencies in which your accounting entries and accounting reports are recorded. In order to perform this conversion, Oracle BI Application also allows you to configure the rate type that you want to use when converting the document currency to the local currency.

To configure the currency rate types:

1. Log into DAC (for more information, see [Section 4.10.2, "How to Log Into the DAC"](#)).
2. Open your custom container
3. Display the 'Source System Parameters' tab.
4. Locate the following DAC parameters and set the exchange rate type values for them in the 'Value' box:
  - \$\$GLOBAL1\_RATE\_TYPE
  - \$\$GLOBAL2\_RATE\_TYPE
  - \$\$GLOBAL3\_RATE\_TYPE
  - \$\$DEFAULT\_LOC\_RATE\_TYPE (the conversion rate type for document currency to local currency conversion).

Make sure you spell the exchange rate type values as they are spelled in your source OLTP system.

5. Save your changes

#### 8.1.1.4 How to Configure Fiscal Calendars

This sections explains how you configure the time dimension in the Oracle Business Analytics Warehouse, and contains the following sections:

- [Section 8.1.1.4.1, "Overview To Setting Up The Time Dimension"](#)
- [Section 8.1.1.4.2, "Setting Up Fiscal Calendar"](#)
- [Section 8.1.1.4.3, "Setting Up Fiscal Calendar By Fiscal Week"](#)
- [Section 8.1.1.4.4, "Setting Up Fiscal Calendar By Fiscal Month"](#)
- [Section 8.1.1.4.5, "Process of Setting Up The Fiscal Calendar"](#)
- [Section 8.1.1.4.6, "Reloading the Time Dimension Tables After Your Data Warehouse is Populated"](#)
- [Section 8.1.1.4.7, "Notes"](#)

**8.1.1.4.1 Overview To Setting Up The Time Dimension** When you configure Time Dimension tables, W\_DAY\_D is the base table that represents the time dimension in the Oracle Business Analytics Warehouse. There are two lines of aggregate dimension tables built on this table, as follows:

- Regular Calendar tables.
- Fiscal Calendar tables.

[Table 8–1](#) shows the tables used to store calendar information.

**Table 8–1 Tables in base table W\_DAY\_D that are used to store calendar information**

Regular calendar tables in W_DAY_D	Fiscal calendar tables in W_DAY_D
W_WEEK_D	W_FSCL_WEEK_D
W_MONTH_D	W_FSCL_MONTH_D
W_QTR_D	W_FSCL_QTR_D
W_YEAR_D	W_FSCL_YEAR_D

There are two parameters \$\$Start\_Date and \$\$End\_date for the task SIL\_DayDimension that need to be setup to load the calendar data in W\_DAY\_D. The SIL mappings use standard time functions to create records for each calendar day falling within the boundary defined by these two parameters. Once the records are created in W\_DAY\_D, the aggregate calendar tables are loaded by their respective SIL mapping.

**8.1.1.4.2 Setting Up Fiscal Calendar** Installed out of the box, Oracle BI Applications supports one fiscal calendar. Fiscal data is first loaded in the W\_DAY\_D table and then the SIL mappings read data from W\_DAY\_D and load data into the aggregate Fiscal Time Dimension tables such as Fiscal Week, Fiscal Month, Fiscal Quarter and Fiscal Year.

You may choose to provide Fiscal calendar information in terms of the Fiscal Weeks of your organization or in terms of the Fiscal months of your organization. In either case, The SIL mappings are designed to derive the Fiscal Week from the Start and End Date of a Fiscal Month by grouping into periods of seven days each.

**8.1.1.4.3 Setting Up Fiscal Calendar By Fiscal Week** In this option you provide data for the Fiscal Year, Fiscal Month, Fiscal Week and Start Date of Fiscal Week. The Fiscal Month information is derived using the 4-4-5 rule for grouping weeks into months. The Fiscal Week End Date is derived based on the start date of the next week that is supplied in the data file. Fiscal Months are grouped into sets of 4 months each to determine the Fiscal Quarter.

**8.1.1.4.4 Setting Up Fiscal Calendar By Fiscal Month** In this option you can provide data at the level of Fiscal Month. The SIL\_DayDimension\_FiscalMonth\_Extract task divides the Fiscal Month into Fiscal Weeks of seven days each. If the number of days in the Fiscal Month is not in multiples of seven, the last week will have less number of days.

**8.1.1.4.5 Process of Setting Up The Fiscal Calendar To configure Fiscal Calendar by Fiscal Week**

1. Open the file fiscal\_week.csv using a text editor in the \$PMRootDir\SrcFiles directory. Enter the Fiscal Year, Fiscal Month, Fiscal Week and the Start Date of Fiscal Week in YYYYMMDD. Records must be inserted in ascending order of Fiscal Year, Fiscal Month, Fiscal Week. Save and close the file.

---

**Note:** Make sure you enter accurate Fiscal Data as there is no check done within the Informatica mappings.

---

2. Deactivate the task SIL\_DayDimension\_FiscalMonth\_Extract in the Data Warehouse container in DAC. Then right click and push to references.
3. Activate the task SIL\_DayDimension\_FiscalWeek\_Extract in the Datawarehouse container in DAC. Then right click and push to references.

### To configure Fiscal Calendar by Fiscal Month

1. Open the file `fiscal_month.csv` using a text editor in the `$PMRootDir\SrcFiles` directory. Enter the Fiscal Year, Fiscal Month and the Start Date of Fiscal\_Month in YYYYMMDD format. Save and close the file.

---

**Note:** Make sure that you enter accurate Fiscal Data as there is no check done within the Informatica mappings.

---

2. Activate the task `SIL_DayDimension_FiscalMonth_Extract` in Datawarehouse container in DAC. Then right click and push to references.
3. Deactivate the task `SIL_DayDimension_FiscalWeek_Extract` in Datawarehouse container. Then right click and push to references.

**8.1.1.4.6 Reloading the Time Dimension Tables After Your Data Warehouse is Populated** The data in time dimension is loaded once during the initial full load. Subsequently, the `SIL_*_UpdateFlag` mappings run everyday to update the domain value codes, which indicate whether a day, week, month, quarter or year is 'Current', 'Next' or 'Previous' as of the current day. The `SIL_Fiscal_UpdateFlag` mappings also update the flags that indicate whether a fiscal week, month, quarter or year is 'Current', 'Previous' or 'Next' with respect to the system date.

You might want to extend the range of data that you have in your time dimension sometime after the warehouse is in production. In order to achieve this, please follow the steps below that will kick off the full load ETL run of the `W_DAY_D` and all the aggregate time dimension tables.

#### To set up the load strategy of the time dimension table

1. In DAC, go to Setup, then Physical Data Sources and click on the connection DataWarehouse.
2. Go the Refresh Dates tab in the bottom panel.

The Refresh Date of all the tables will be given there. Double click on the Refresh Date of `W_DAY_D` and make it Null. Do the same for the following aggregate time dimension tables: `W_WEEK_D`, `W_QTR_D`, `W_MONTH_D`, `W_YEAR_D`, `W_FSCL_WEEK_D`, `W_FSCL_QTR_D`, `W_FSCL_MONTH_D`, and `W_FSCL_YEAR_D`.

3. Display the Design\Task tab and open the task `SIL_DayDimension` in DataWarehouse container.
4. Set the `$$Start_date` and `$$End_Date` parameters to the appropriate start date and end date of the new date range.
5. Save the task and push the changes to references.

---

**Note:** Make sure that the new date range defined by the parameters `$$Start_date` and `$$End_date` encompasses the old date range entirely. Otherwise records can be lost. Also make sure you have provided the fiscal information for the new date range also in the appropriate fiscal input file. The next time you run your ETL execution plan, the calendar dimension will be truncated and reloaded with the new date range that you have specified. Since the calendar dimensions use smart keys, none of the fact tables that refer to these tables need to be reloaded.

---

**8.1.1.4.7 Notes** The following notes pertain to the process of setting up the fiscal calendar:

- If there is a week (starting on a Sunday and ending on a Saturday) that falls across two Calendar years, the week is counted in both years. For example the week that starts on 12/30/2007 will be counted in both 2007 and 2008. In 2007 the week Start Date will 12/30/2007 and the End Date will be 12/31/2007. In 2008 this will be the first week with Start Date as 01/01/2008 and End Date as 01/05/2008.
- W\_DAY\_D stores 31 records for each month irrespective of whether the month actually has 31 days or not. If the month actually has less number of days, there will be records with Null values in the Calendar Date and Day Date columns. These extra records are loaded for the calculation of Period Ago metrics in the RPD, and will not affect the ETL or reporting.
- There are some attributes on the W\_DAY\_D table that are not mapped in the physical layer of the RPD. Therefore, before creating any new attribute in the RPD, check whether the attribute is already available in the physical layer and if it can be mapped directly.
- If your fiscal calendar contains more than 12 months, the extra months will be assigned a value of 0 for the Fiscal Quarter. The same holds for the Fiscal Trimester and Fiscal Half values.

## 8.1.2 Configuration Steps for Controlling Your Data Set for All Source Systems

This section contains additional configuration steps that apply to Oracle BI Applications deployed with any source system, and contains the following topics:

- [Section 8.1.2.1, "How to Configure Data Source Num IDs"](#)

### 8.1.2.1 How to Configure Data Source Num IDs

DATASOURCE\_NUM\_ID is a system column in the data warehouse that uniquely identifies a data source category and indicates which source systems the data comes from. For example, the value '1' indicates one of the Siebel data sources, and the value 2 indicates an Oracle 11.5.8. data source.

Oracle Business Intelligence is installed with a number of pre-defined data source templates that you can edit to specify OLTP and OLAP data sources. If you create a new data source without using one of the pre-defined templates, you must specify the correct DATASOURCE\_NUM\_ID for that data source category. For example, if you specify an Oracle EBS data source, you must specify the DATASOURCE\_NUM\_ID value '9'. For more information about specifying data sources, see [Section 4.11.2, "How to Specify Transactional and Data Warehouse Data Sources in the DAC"](#).

The table below shows the data sources that are supported by Oracle Business Intelligence and their associated DATASOURCE\_NUM\_ID values.

**Table 8–2 Data Sources and Associated DATASOURCE\_NUM\_ID Values**

Data Source Name	Data Source Number
ORA_11_5_8	2
ORA_11_5_9	5
ORA_11_5_10	4
ORACLE_R12	9

**Table 8–2 (Cont.) Data Sources and Associated DATASOURCE\_NUM\_ID Values**

Data Source Name	Data Source Number
PSFT_8_4_FINSCM	7
PSFT_8_8_FINSCM	8
PSFT_8_8_HCM	6
SEBL_63	1
SEBL_771	1
SEBL_753	1
SEBL_78	1
SEBL_80	1
SEBL_VERT_771	1
SEBL_VERT_753	1
SEBL_VERT_78	1
SEBL_VERT_80	1
SRV_SEBL_80	1
UNIV	3

To configure a DATASOURCE\_NUM\_ID value:

1. Log into DAC (for more information, see [Section 4.10.2, "How to Log Into the DAC"](#)).
2. Display the Set tab.
3. Display the 'Physical Data Sources' tab.
4. Select a data source from the list.
5. On the Edit tab, edit the value in the **Data Source Number** field.
6. Click Save.

If you do choose to modify the Data Source Number and if you implement Supply Chain Analytics, it is Mandatory that you finish the steps in section [Section 9.2.2.2, "How to Configure the DAC parameter for Purchase Cycle Line"](#).

## 8.2 Oracle EBS-Specific Configuration Steps

This section contains configuration steps that apply to Oracle Business Intelligence Applications deployed with Oracle EBS source systems.

This section contains the following topics:

- [Section 8.2.1, "Configuration Required Before A Full Load for Oracle EBS"](#)
- [Section 8.2.2, "Configuration Steps for Controlling Your Data Set for Oracle EBS"](#)

### 8.2.1 Configuration Required Before A Full Load for Oracle EBS

This section contains configuration steps required before a full data load that apply to Oracle BI Applications deployed with Oracle EBS source systems, and contains the following topics:

- [Section 8.2.1.1, "Configuration of Product hierarchy \(except for GL, HR modules\)"](#)
- [Section 8.2.1.2, "Configure the Master Inventory Organization in Product dimension Extract for Oracle 11i adapter \(Except for GL & HR Modules\)"](#)

### 8.2.1.1 Configuration of Product hierarchy (except for GL, HR modules)

There are two dimension tables that have built-in product hierarchies - the Product and Inventory Product dimension tables. These dimension tables share one Product category staging table.

ETL extracts the Product and Sales Product staging tables separately, and then joins these tables with the shared category staging table to load hierarchies. The category extract mapping controls the category sets that are used in the Product and Sales Product dimensions. The load mappings for the Product and Inventory Product dimensions specifies which Category Set is used to load to the hierarchy columns.

As initially configured, the Oracle Business Analytics Warehouse extracts product categories where the CATEGORY\_SET\_ID is 2 or 27. However, it is likely that the categories you extract from the source system are different from these prepackaged category sets. Therefore, you must reconfigure your product category sets by making the following changes:

- Identify and extract only the categories that you want to report against into the Product Hierarchy sets Staging table.
- Properly position the data so that it loads into the Oracle Business Analytics Warehouse.

**8.2.1.1.1 How to identify Category Sets from EBS** To find out the category sets that your organization is using:

1. Login to the instance.
2. Navigate to Setup > Items > Categories > Default Category Sets
3. Look for functional Area Inventory and place the cursor in the Category Set Field
4. Then click menu Help > Diagnostics > Examine, and provide the apps user password
5. Click the Field LOV button and select CATEGORY\_SET\_ID, note down the value
6. Repeat step 3-5 for Purchasing functional area.

**8.2.1.1.2 To configure product category sets extract from Oracle Applications** To configure product category sets extract from Oracle Applications:

1. Identify the categories that need to be mapped to the extension hierarchy columns. These categories are extracted from the source and placed in the hierarchy column specified.
2. In PowerCenter Workflow Manager, open the Configuration for Oracle Applications v11i.
3. Open the SDE\_ORA\_PRODUCT\_CATEGORY\_DERIVE session with the Task Developer.
4. In the Mappings tab, click MPLT\_BC\_ORA\_PRODUCT\_CATEGORY.SQ\_MTL\_CATEGORIES.
5. In the right pane, scroll down and click SQL Query to edit session SQL override.
6. Click the arrow to edit the WHERE clause.

The following statement is an example of how to structure the WHERE clause:

```
WHERE...MTL_CATEGORY_SETS_B.CATEGORY_SET_ID IN (27,2)
```

In this example, the WHERE clause extracts categories where the Category SetID is 27 or 2.

7. Click OK, and then click OK to close the Edit Tasks box.
8. Repeat steps 3 to 7 for the SDE\_ORA\_PRODUCT\_CATEGORY\_DERIVE\_FULL session.

---

**Note:** The default product hierarchy level is two. If you only have one product hierarchy level, you need to make the following modifications:

1. In the EXP\_PRODUCTS expression transformation in the mplt\_SA\_ORA\_ProductDimension mapplet, modify the EXT\_PROD\_HIER1\_CODE port to INP\_SEGMENT1.
  2. In the SDE\_ORA\_ProductDimension mapping, open the EXP\_W\_PRODUCT\_DS transformation expression and modify the PROD\_HIER1\_CODE port expression to IN\_PROD\_HIER1\_CODE.
- 

---

**Note:** The category set chosen as a value for this parameter must be a Category Set controlled at a Master Level but not at the Org level as the grain of the Product dimension is at the Master level.

---

**8.2.1.1.3 To configure the Category Sets for Product Hierarchy** To configure the Category Sets for Product Hierarchy:

1. Open the ParameterOLTP.TXT file in OracleBI\DAC\Informatica\parameters\input folder.
2. Replace the default parameter in SDE\_ORA\_Product\_Dimension:
  - \$\$CATEGORY\_SET\_ID=2
3. Save and Exit the Parameter file.
4. Restart the DAC Server to load the parameter changes.

**8.2.1.1.4 To reconfigure the product hierarchy loads** To reconfigure the product hierarchy loads:

1. In PowerCenter Designer, open the Configuration for Oracle Applications v11i folder, and expand the Mapplets folder.
2. Open the mplt\_SA\_ORA\_ProductDimension mapplet for Oracle 11i.
3. Double-click the EXP\_PRODUCTS expression transformation to open the Edit Transformation box.
4. In the Ports tab, scroll down to find the hierarchy code port.

Hierarchy levels are named with the following convention EXT\_PROD\_HIERX\_CODE, where X denotes the level within the hierarchy. For example, if you want to edit the first level of your hierarchy, you must edit the definition for EXT\_PROD\_HIER1\_CODE port.

The first two levels are preconfigured as follows:

```
EXT_PROD_HIER1_CODE = IIF(ISNULL(INP_SEGMENT1) OR ISNULL(INP_SEGMENT2), NULL, INP_SEGMENT1 || '~' || INP_SEGMENT2)
EXT_PROD_HIER2_CODE = INP_SEGMENT1
```

5. Modify the expression that defines your hierarchy code.
6. Validate and save your changes to the repository.

**8.2.1.1.5 To configure the Category Sets for Inventory Product Hierarchy** To configure the Category Sets for Inventory Product Hierarchy:

1. Open the ParameterOLTP.TXT file in OracleBI\DAC\Informatica\parameters\input folder.
2. Replace the default Category Set ID (27) with your new value.
3. Save and close the file.

### **8.2.1.2 Configure the Master Inventory Organization in Product dimension Extract for Oracle 11i adapter (Except for GL & HR Modules)**

In Oracle 11i applications, the products are defined in a Master Organization and then copied into the other Inventory Organizations for transactions. The Product dimension Extract mapping 'SDE\_ORA\_ProductDimension\_Derive' has been enabled for configuration of this Master Organization based on the configuration in the OLTP. This is enabled via a session parameter "MASTER\_ORG" in the parameter file (OracleBI\DAC\Informatica\parameters\input\ParameterOLTP.TXT) for the ETL. Out-of-the-box, this parameter is set to "\$\$MASTER\_ORG=204". This organization Id 204 needs to be changed based on the individual implementation of OLTP in your deployment.

---

---

**Note:** This ETL implementation supports the best practice prescribed by Oracle for the creation of Single Master Organization for defining the Product master. This ETL implementation does not support the multiple master Organizations if the same product is defined in multiple master organizations.

---

---

## **8.2.2 Configuration Steps for Controlling Your Data Set for Oracle EBS**

This section contains additional configuration steps that apply to Oracle BI Applications deployed with Oracle EBS source systems, and contains the following topics:

- [Section 8.2.2.1, "Configuring the Country Region and State Region Name"](#)
- [Section 8.2.2.2, "Configuring the State Name"](#)
- [Section 8.2.2.3, "Configuring the Country Name"](#)
- [Section 8.2.2.4, "Configuring the Configuring the Make-Buy Indicator"](#)

### **8.2.2.1 Configuring the Country Region and State Region Name**

For Oracle 11i, you can reconfigure the region, state, and country names. This configuration information applies only to plant, storage, and supplier locations. By default, the Region Name column (EXT\_REGION\_NAME) is populated using the same code value as the Region Code column (EXT\_REGION\_CODE). However, you can redefine the load mapping's Source Adapter mapplet to load a source-supplied region name instead of the code. If you want to reconfigure the load in this manner,

you can load the region code and region name into the W\_CODE\_D table. For information on loading codes and code names into the W\_CODE\_D table, see [Section 6.10, "Codes Lookup"](#).

When you have loaded the region code and region name into the W\_CODE\_D table, you can remove the expression in the Source Adapter that defines the Region Name column. By making the Region Name's expression blank, the SIL looks up the Region Name in the W\_CODE\_D table, using the supplied region code when the load occurs. The load mapping then inserts the region name and region code into the data warehouse table.

To configure the Country Region Name:

1. In Informatica PowerCenter Designer, open the Configuration for SDE\_ORA1158\_adapter folder.
2. Open the maplet you want to edit.  
The following is a list of all Source Adapter maplets that use the EXT\_COUNTRY\_REGION column:
  - mplt\_SA\_ORA\_SupplierDimension
  - mplt\_SA\_ORA\_BusinessLocationDimension\_Plant
  - mplt\_SA\_ORA\_BusinessLocationDimension\_StorageLocation
3. Double-click the Expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the EXT\_COUNTRY\_REGION port.
4. Edit the condition by removing the assigned value if you want the lookup to occur.
5. Click Apply.
6. Validate the maplet, and save your changes to the repository.

To configure the State Region Name

1. In Informatica PowerCenter Designer, open the SDE\_ORA<Ver>\_adapter.
2. Open the maplet you want to edit.  
The following is a list of all Source Adapter maplets that use the EXT\_STATE\_REGION column:
  - mplt\_SA\_ORA\_SupplierDimension
  - mplt\_SA\_ORA\_BusinessLocationDimension\_Plant
  - mplt\_SA\_ORA\_BusinessLocationDimension\_StorageLocation
3. Double-click the Expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the EXT\_STATE\_REGION port.
4. Edit the condition by removing the assigned value if you want the lookup to occur.
5. Click Apply.
6. Validate the maplet, and save your changes to the repository.

### 8.2.2.2 Configuring the State Name

For Oracle 11i, you can reconfigure the region, state, and country names that apply to the Supplier locations only. By default, the State Name column (EXT\_STATE\_NAME) is populated using the same code value as the State Code column (EXT\_STATE\_CODE). However, you can redefine the load mapping's Source Adapter maplet to load a source-supplied state name instead of the code. If you want to reconfigure the

load in this manner, you can load the state code and state name into the W\_CODE\_D table. For information on loading codes and code names into the W\_CODE\_D table, see [Section 6.10, "Codes Lookup"](#).

When you have loaded the state code and state name into the W\_CODE\_D table, you can remove the Expression in the Source Adapter that defines the State Name column. By setting the State Name's expression to null, the SIL looks up the state name in the W\_CODE\_D table using the supplied state code, during the load process. The load mapping then inserts the state name and state code into the data warehouse table.

To configure the State Name:

1. In Informatica PowerCenter Designer, open the SDE\_ORA<Ver>\_adapter.
2. Open one of the following mapplets:
  - mplt\_SA\_ORA\_SupplierDimension
  - mplt\_SA\_ORA\_BusinessLocationDimension\_Plant
  - mplt\_SA\_ORA\_BusinessLocationDimension\_StorageLocation
3. Double-click the Expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the EXT\_STATE\_NAME port.
4. Edit the condition by removing the assigned value if you want the lookup to occur.
5. Click Apply.
6. Validate the mapplet and save your changes to the repository.

### 8.2.2.3 Configuring the Country Name

For Oracle 11i, you can reconfigure the region, state, and country names that apply to supplier locations only. By default, the Country Name column (EXT\_COUNTRY\_NAME) is populated using the same code value as the Country Code column (EXT\_COUNTRY\_CODE). However, you can redefine the load mapping's Source Adapter mapplet to load a source-supplied country name instead of the code. If you want to reconfigure the load in this manner, you can load the country code and country name into the W\_CODE\_D table. For information on loading codes and code names into the W\_CODE\_D table, see [Section 6.10, "Codes Lookup"](#).

When you have loaded the country code and country name into the W\_CODE\_D table, you can remove the expression in the Source Adapter that defines the Country Name column. By setting the Country Name's expression to null, when the load occurs, the SIL looks up the country name in the W\_CODE\_D table, using the supplied country code. The load mapping then inserts the country name and country code into the data warehouse table.

To configure the Country Name:

1. In Informatica PowerCenter Designer, open the SDE\_ORA<Ver>\_adapter.
2. Open one of the following mapplets:
  - mplt\_SA\_ORA\_SupplierDimension
  - mplt\_SA\_ORA\_BusinessLocationDimension\_Plant
  - mplt\_SA\_ORA\_BusinessLocationDimension\_StorageLocation
3. Double-click the Expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the EXT\_COUNTRY\_NAME port.
4. Edit the condition by removing the assigned value if you want the lookup to occur.

5. Click Apply.
6. Validate the mapplet, and save your changes to the repository.

#### 8.2.2.4 Configuring the Configuring the Make-Buy Indicator

The Make-Buy indicator specifies whether a material that was used to manufacture a product was made in-house or bought from an outside vendor. By default, the indicator is set using the INP\_PLANNING\_MAKE\_BUY\_CODE. If the code is set to 1, then the indicator is set to M (for make). However, if the code is set to 2, then the indicator is set to B (for buy). Otherwise, the indicator is set to null.

Your organization may require different indicator codes. If so, you can modify the indicator logic by reconfiguring the condition in the mapplet `mplt_SA_ORA_ProductDimension`. For example, you may want your indicator code to be 0 for make, and 1 for buy.

To configure the Make-Buy Indicator:

1. In Informatica PowerCenter Designer, open the `SDE_ORA<Ver>_adapter`.
2. Open one of the following mapplets:
  - `mplt_SA_ORA_SupplierDimension`
  - `mplt_SA_ORA_BusinessLocationDimension_Plant`
  - `mplt_SA_ORA_BusinessLocationDimension_StorageLocation`
3. Double-click the Expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the `EXT_COUNTRY_NAME` port.
4. Edit the condition by removing the assigned value if you want the lookup to occur.
5. Click Apply.
6. Validate the mapplet, and save your changes to the repository.

## 8.3 PeopleSoft-Specific Configuration Steps

This section contains configuration steps that apply to Oracle Business Intelligence Applications deployed with PeopleSoft source systems.

This section contains the following topics:

- [Section 8.3.1, "Configuration Required Before A Full Load for PeopleSoft"](#)
- [Section 8.3.2, "Configuration Steps for Controlling Your Data Set for PeopleSoft"](#)

### 8.3.1 Configuration Required Before A Full Load for PeopleSoft

Not applicable to Oracle BI Applications Version 7.9.4.

### 8.3.2 Configuration Steps for Controlling Your Data Set for PeopleSoft

Not applicable to Oracle BI Applications Version 7.9.4.

## 8.4 Siebel-Specific Configuration Steps

This section contains configuration steps that apply to Oracle Business Intelligence Applications deployed with Siebel source systems.

This section contains the following topics:

- [Section 8.4.1, "Configuration Required Before A Full Load for Siebel Source Systems"](#)
- [Section 8.4.2, "Configuration Steps for Controlling Your Data Set for Siebel Source Systems"](#)

### **8.4.1 Configuration Required Before A Full Load for Siebel Source Systems**

Not applicable to Oracle BI Applications Version 7.9.4.

### **8.4.2 Configuration Steps for Controlling Your Data Set for Siebel Source Systems**

Not applicable to Oracle BI Applications Version 7.9.4.

---

---

# Configuring Oracle's Supply Chain Family of Products

This chapter describes how to configure Oracle's Supply Chain Analytics family of products for particular sources to meet your business needs.

It contains the following topics:

- [Section 9.1, "Overview of Oracle's Supply Chain Analytics Family of Products"](#)
- [Section 9.2, "Configuration Required Before A Full Load for Supply Chain Analytics"](#)

## 9.1 Overview of Oracle's Supply Chain Analytics Family of Products

Oracle's Supply Chain Analytics family of products consists of the following:

- Oracle Inventory Analytics
- Oracle Procurement and Spend Analytics
- Oracle Supplier Performance Analytics

Oracle's Supply Chain Analytics family of products enable organizations to optimize their supply side performance by integrating data from across the enterprise value chain and enabling executives, managers, and frontline employees to make more informed and actionable decisions. Organizations using Supply Chain Analytics benefit from increased visibility into the complete Supply chain process, including comprehensive procurement and spend analysis, supplier performance analysis, inventory analysis, and supplier payables analysis. Through complete end-to-end insight into the factors that impact supply chain performance, organizations can significantly reduce costs, enhance profitability, increase customer satisfaction, and gain competitive advantage. Supply Chain Analytics also integrate with the other applications in Oracle BI Applications product line. They deliver this insight across the organization to increase the company's effectiveness in managing its customers, suppliers, and financial decisions. Supply Chain Analytics consists of the following modules:

- [Section 9.1.1, "Procurement and Spend Analytics Module"](#)
- [Section 9.1.2, "Supplier Performance Analytics Module"](#)
- [Section 9.1.3, "Inventory Analytics Module"](#)

### 9.1.1 Procurement and Spend Analytics Module

Provides complete visibility into direct and indirect spend across the enterprise, payment, and employee expenses. Example analyses are spend by Commodity & Supplier, by Purchase Org, Cost Center, and expense by Employee, Buyer, etc.

The Procurement and Spend Analytics application is comprised of these subject areas:

- **Total Spend:** This is a summary subject area that provides the ability to do comparative analysis and report on requested spend, committed spend and actual spend across suppliers, company, products, commodities and associated hierarchies for both direct and indirect spend (indirect spend being MRO and employee expenses) in detail to allow complete visibility of spending across your organization.
- **Purchase Orders:** This is a detailed subject area that provides the ability to report on committed spend, and Purchase orders of the suppliers of an organization across suppliers, company, products, commodities and associated hierarchies at purchase order line level
- **Purchase Order Costs:** This is a detailed subject area that provides the ability to report on committed spend and Purchase orders of the suppliers of an organization across suppliers, company, products, and commodities and associated hierarchies at cost center (distribution line) level.
- **Purchase Cycle Lines:** This is a summary subject area that provides the ability to report cycle time performance such as Requisition to PO lead time, PO to Receipt lead time, P2P lead time of the Suppliers of an organization.
- **Purchase Schedules:** This is a detailed subject area that provides the ability to report on purchase order shipments of an organization across suppliers, company, products, commodities and associated hierarchies at purchase schedule line level
- **Purchase Requisitions:** This is a detailed subject area that provides the ability to report on requested spend and Purchase requisitions of the suppliers of an organization across suppliers, company, products, commodities and associated hierarchies at purchase requisition line level
- **Purchase Requisition Status:** This is a summary subject area that provides the ability to report on requisition status along the approval cycle of Purchase requisitions of the suppliers of an organization. It's populated only by Universal adapter.
- **Purchase Receipts:** This is a detailed subject area that provides the ability to report on actual spend and Purchase Receipts of the suppliers of an organization across suppliers, company, location, products, commodities and associated hierarchies at purchase receipt line level
- **Employee Spend:** This is a detailed subject area that provides the ability to report on employee spend of an organization across employees, company, cost center and associated hierarchies. The Expenses subject area contains targeted metrics and reports that examine travel and expense costs in relationship to your organization's overall spending patterns. In contrast to analyzing direct spending patterns, where you may review purchasing, Expenses examines indirect spending—the cost of employee related expenses. It's populated only by Universal adapter.

### 9.1.2 Supplier Performance Analytics Module

Enables organizations to have a complete picture of the performance of their suppliers, including complete supplier scorecards, procurement cycle times, supplier price

performance, delivery performance, product receipt quality, on-time payment ratings, payment activity and volume and payments due / overdue analysis.

The Supplier Performance Analytics application is comprised of these subject areas:

- **Supplier Performance.** The Suppliers functional area contains targeted reports and metrics that allow you to analyze the timeliness, reliability, cost, and quality of goods provided by your suppliers. It helps you to understand how well suppliers are contributing to success of your organization, and to evaluate the price, quality, and delivery timing in procuring materials
- **Supplier AP Transactions:** This is a summary subject area that provides the ability to analyze payment performance and payment due analysis of the suppliers of an organization across suppliers, company, location, products, commodities and associated hierarchies. In addition to monitoring supplier performance, it is important to monitor organization's performance of making on time payments. This will help the Organizations to maintain better relationships with their best suppliers.

### 9.1.3 Inventory Analytics Module

Incorporates analysis of inventory held by an organization; bill of materials; and inventory movements in, out, and through manufacturing plants, distribution centers, or storage locations. This enables companies to monitor inventory levels trend to Sales performance to improve cost exposure, increase turnover through inventory level reduction and increased velocity, properly deploy inventory at the right place / right time and better understand Customer and Supplier Returns to maintain quality.

The Inventory Analytics application is comprised of these subject areas:

- **Inventory Transactions.** The Inventory Transactions subject area allows you to analyze the various types of events and tasks that occur. Examples of these activities include tracking inventory by type of movement. For example, transfer, issues, receipts, returns, sales, and so on. It allows the user to understand the impact of these activities on business operations, and allows the identification of problematic trends early. For example, large quantities of product in-transit.
- **Inventory Balances.** The Inventory Balances subject area allows you to analyze the inventory held by an organization in relation to a number of different dimensions. For example, Product type, Product number, Storage location, Plant, Consigned Inventory, Restricted, and so on. It allows the user the ability to understand and determine the optimal distribution of assets as well as identify potential issues such as unnecessary build up of inventories.
- **Bill of Materials.** The Bill of Materials (BOM) subject area allows you to determine the profit margin of the components that comprise the finished goods. BOM allows you to keep up with the most viable vendors in terms of cost and profit, and to keep your sales organization aware of product delivery status, including shortages.
- **Customer and Supplier Returns.** The Customer and Supplier Returns subject area allows the user to specifically monitor the return of product by both Customers and Suppliers. At a Product level, it allows the user to identify early, potential, Customer-satisfaction issues relating to problematic Suppliers and Product.

## 9.2 Configuration Required Before A Full Load for Supply Chain Analytics

This section contains configuration steps that you need to perform on Oracle's Supply Chain Analytics family of products before you do a full data load, and contains the following topics:

- [Section 9.2.1, "Configuration Steps for Supply Chain For All Source Systems"](#)
- [Section 9.2.2, "Configuration Steps for Supply Chain For Oracle EBS"](#)
- [Section 9.2.3, "Configuration Steps for Supply Chain For PeopleSoft"](#)
- [Section 9.2.4, "Configuration Steps for Supply Chain For Universal"](#)
- [Section 9.2.5, "Configuration Steps for Controlling Your Data Set"](#)

### 9.2.1 Configuration Steps for Supply Chain For All Source Systems

This section contains configuration steps that apply to all source systems before you do a full data load, and contains the following topics:

- [Section 9.2.1.1, "How to Deploy Stored Procedures"](#)

---

---

**Note:** For configuration steps that apply to all analytics modules (for example, Oracle Financial Analytics, Oracle HR Analytics, Oracle Sales Analytics, see [Chapter 8, "Configuring Common Areas and Dimensions"](#)).

---

---

#### 9.2.1.1 How to Deploy Stored Procedures

Stored procedures are a group of SQL statements that perform particular tasks on the database. For example, stored procedures can help to improve the performance of the database.

You can deploy stored procedures by copying the stored procedure files from your Oracle BI Analytics installation and deploying them to the target data warehouse.

---

---

**Note:** Some sessions may fail if these procedures are not compiled in the database before running the workflows.

---

---

#### To deploy stored procedures

1. Navigate to the OracleBI\dwrep\Informatica\Stored\_Procedure\_Scripts folder. copy the source codes Compute\_Bounds\_Ora11i.sql into the target data warehouse schema.
2. Compile the stored procedures in the target data warehouse database.

---

---

**Note:** If you have problems deploying the stored procedures, see your database reference guide, or contact your database administrator.

---

---

### 9.2.2 Configuration Steps for Supply Chain For Oracle EBS

This section contains configuration steps that apply to Oracle EBS before you do a full data load, and contains the following topics:

- Section 9.2.2.1, "How to Configure the Bill of Materials Explosion for Oracle Version 11.5.9 and above"
- Section 9.2.2.2, "How to Configure the DAC parameter for Purchase Cycle Line"
- Section 9.2.2.3, "How to Configure Quantity Types for Product Transactions"
- Section 9.2.2.4, "Domain Values and CSV Worksheet Files for Supply Chain Analytics"

### 9.2.2.1 How to Configure the Bill of Materials Explosion for Oracle Version 11.5.9 and above

Starting from 11.5.9 version, the BOM\_EXPLOSION\_TEMP table is changed to temp table, duration = SYS\$SESSION. To be able to extract data from the BOM\_EXPLOSION\_TEMP table, you need to follow these configuration steps:

#### To configure the bill of materials explosion options for Oracle Version 11.5.9

1. On the Informatica Server machine, Click Start, click Run, type regedit, click OK.
2. Go to the following registry key:

```
HKEY_LOCAL_
MACHINE\SYSTEM\CurrentControlSet\Services\PowerMart\Parameters\Configuration
```

3. On the Edit menu, point to New, and then click String Value.
4. Enter the String Value as:
  - Name: ShareConnections\_<folder name>.SDE\_ORA\_BOMItemFact
  - Data: Yes
  - The folder name is one of the following:
    - For Oracle 11.5.9 Version, use SDE\_ORA1159\_adapter.
    - For Oracle 11.5.10 Version, use SDE\_ORA11510\_adapter.
    - For Oracle R12 Version, use SDE\_ORAR12\_adapter.
5. Re-start the PowerCenter server (Informatica Service).

#### To configure the bill of materials explosion options for Oracle 11.5.9 and above on a Unix machine

1. Add the following line in /<INFA\_home>/informatica/pm/pmsserver.cfg file:
 

```
ShareConnections_<folder name>.SDE_ORA_BOMItemFact=Yes
```
2. Enter the Folder Name value as one of the following:
  - For Oracle 11.5.9 Version, use SDE\_ORA1159\_adapter.
  - For Oracle R12 Version, use SDE\_ORAR12\_adapter.
3. Re-start the PowerCenter server (Informatica Service).

### 9.2.2.2 How to Configure the DAC parameter for Purchase Cycle Line

To load the purchase cycle line table (W\_PURCH\_CYCLE\_LINE\_F), the ETL tasks needs to distinguish data coming from Oracle 11i applications from data from other applications such as SAP and PeopleSoft. This configuration explains the detail steps.

#### To configuring the DAC parameter for purchase cycle line

1. Login to your DAC repository

2. Go to the Tasks Tab, query for task SIL\_PurchaseCycleLinesFact.
3. Click Parameters Child tab. Change the values for parameter \$\$ORA\_DATASOURCE\_NUM\_ID\_LIST from (2,4,5,9) to the list of Data Source NUM ID you defined for your Oracle connections
4. Repeat the same for task SIL\_PurchaseCycleLinesFact\_Extract

For more information on configuring DAC, please refer to *Oracle Business Intelligence Data Warehouse Administration Console Guide*.

### 9.2.2.3 How to Configure Quantity Types for Product Transactions

Oracle 11i categorize quantities into three different types:

- **Goods Received quantities.** *Goods Received quantity* refers to the number of goods received.
- **Delivery quantities.** *Delivery quantity* refers to the number of goods delivered.
- **Base quantities.** *Base quantity* refers to any transaction quantity.

The Oracle Business Analytics Warehouse extracts the transaction type and loads this value into the XACT\_SRC\_TYPE column. In this column, the value 1 denotes a Goods Received quantity, and 2 denotes a Delivery quantity.

To find out more about XACT\_SRC\_TYPE column, please execute the following SQL against your EBS instance:

```
select TRANSACTION_SOURCE_TYPE_ID, TRANSACTION_SOURCE_TYPE_NAME,
DESCRIPTION from MTL_TXN_SOURCE_TYPES order by 1
```

If you have rows equivalent to Purchase order (1), you should include the TRANSACTION\_SOURCE\_TYPE\_ID in the Goods Received quantity column (EXT\_GR\_QTY). If you have rows equivalent to Sales Order (2), you should include the TRANSACTION\_SOURCE\_TYPE\_ID in the Delivery quantity column (EXT\_DELIVERY\_QTY).

All quantities extracted from the source system are always loaded into the Base quantity column (EXT\_BASE\_QTY). However, only the receipt quantity is loaded into the Goods Received quantity column (EXT\_GR\_QTY), and only delivered quantities are loaded into the Delivery quantity column (EXT\_DELIVERY\_QTY).

If your definition of goods received or delivery quantity is different from the prepackaged condition, then you can edit the condition to suit your business needs.

#### To configure the Quantity type

1. In PowerCenter Designer, open the SDE\_ORA<Ver>\_adapter.
2. Open the mplt\_SA\_ORA\_ProductTransactionFact mapplet.
3. Double-click the Expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the EXT\_GR\_QTY and EXT\_DELIVERY\_QTY port.
4. Edit the quantity types by substituting your desired condition for the prepackaged expression.
5. Click Apply.
6. Validate the mapplet, and save your changes to the repository.

### 9.2.2.4 Domain Values and CSV Worksheet Files for Supply Chain Analytics

If you have modify or extend a seeded list of values, you must configure the CSV files for Oracle Supply Chain Analytics by mapping values from your source systems to the domain values.

This section explains how to extract the lists of values from your source system, which you then compare with the seeded values. If the lists of values are different to the seeded values, you need to follow the instructions to configure the Domain Values and CSV Worksheet Files.

**9.2.2.4.1 List of Domain Values and CSV Worksheet Files for Supply Chain Analytics** Table 9–1 lists the CSV worksheet files and the domain values for Supply Chain Analytics in the `$pmsserver\LkpFiles` folder.

**Table 9–1 Domain Values and CSV Worksheet Files for Supply Chain Analytics**

Worksheet File Name	Description	Session
domainValues_Movement_Types_ora11i.csv	List the Movement Type and the corresponding domain values for the Oracle 11i application. For information about how to edit this file, see <a href="#">Section 9.2.2.4.2, "To configure domainValues_Movement_Types_ora11i.csv"</a> .	SDE_ORA_MovementTypeDimension
domainValues_PaymentType_ora12.csv	Lists the Payment Type and the corresponding domain values for the Oracle 11i application. For information about how to edit this file, see <a href="#">Section 9.2.2.4.3, "To configure domainValues_PaymentType_ora12.csv"</a> .	SDE_ORA_TransactionTypeDimension_PO_Payment_Type
domainValues_ShipmentType_ora12.csv	List the Shipment Type and the corresponding domain values for the Oracle 11i application. For information about how to edit this file, see <a href="#">Section 9.2.2.4.10, "To configure domainValues_ShipmentType_ora12.csv"</a> .	SDE_ORA_TransactionTypeDimension_PO_Shipment_Type
domainValues_Status_Purch_Approve_ora11i.csv	Lists the Purchasing Approval Status column and the corresponding domain values for the Oracle 11i application. For information about how to edit this file, see <a href="#">Section 9.2.2.4.4, "To configure domainValues_Status_Purch_Approve_ora11i.csv"</a> .	SDE_ORA_StatusDimension_PurchaseApprove
domainValues_Status_Purch_Cycle_ora11i.csv	Lists the Purchasing Cycle Status column and the corresponding domain values for the Oracle 11i application. For information about how to edit this file, see <a href="#">Section 9.2.2.4.5, "To configure domainValues_Status_Purch_Cycle_ora11i.csv"</a> .	SDE_ORA_StatusDimension_PurchaseCycle
domainValues_Xact_Types_PO_Line_Type_CODE1_ora11i.csv	List the Purchase Basis Type and the corresponding domain values for the Oracle EBS application. For information about how to edit this file, see <a href="#">Section 9.2.2.4.9, "To configure domainValues_Xact_Types_PO_Line_Type_CODE1_ora11i.csv"</a> .	SDE_ORA_TransactionTypeDimension_PO_Line_Type

**Table 9–1 (Cont.) Domain Values and CSV Worksheet Files for Supply Chain Analytics**

Worksheet File Name	Description	Session
domainValues_Xact_Types_PO_Line_Type_ora11i.csv	Lists the Purchasing Line Type and the corresponding domain Values for the Oracle EBS application. For information about how to edit this file, see <a href="#">Section 9.2.2.4.8, "To configure domainValues_Xact_Types_PO_Line_Type_ora11i.csv"</a> .	SDE_ORA_TransactionTypeDimension_PO_Line_Type
domainValues_Xact_Types_Purch_Orders_ora11i.csv	Lists the Purchase Order Transaction Type column and the corresponding domain values for the Oracle EBS application. For information about how to edit this file, see <a href="#">Section 9.2.2.4.6, "To configure domainValues_Xact_Types_Purch_Orders_ora11i.csv"</a> .	SDE_ORA_TransactionTypeDimension_PurchaseOrder
domainValues_Xact_Types_Purch_Requisitions_ora11i.csv	Lists the Purchase Requisition Transaction Type column and the corresponding domain values for the Oracle EBS application. For information about how to edit this file, see <a href="#">Section 9.2.2.4.7, "To configure domainValues_Xact_Types_Purch_Requisitions_ora11i.csv"</a> .	SDE_ORA_TransactionTypeDimension_PurchaseRequest

**9.2.2.4.2 To configure domainValues\_Movement\_Types\_ora11i.csv**

This section explains how to configure domainValues\_Movement\_Types\_ora11i.csv.

1. Identify the Inventory Movement Type in your Oracle EBS source system by using the following SQL:

```
SELECT DISTINCT MTL_TRANSACTION_TYPES.TRANSACTION_TYPE_NAME FROM MTL_TRANSACTION_TYPES
```

2. From the \$pmsserver\lkpfiles folder, open the domainValues\_Movement\_Types\_ora11i.csv file in a text editor.
3. Copy the TRANSACTION\_TYPE\_NAME to the TRANSACTION\_TYPE\_NAME column in the file.

The data must be copied starting from the 2nd line.

4. Map each TRANSACTION\_TYPE\_NAME to one Inventory Movement Type domain value.

Use commas to separate the entries.

5. Save and close the file.

**9.2.2.4.3 To configure domainValues\_PaymentType\_ora12.csv**

This section explains how to configure domainValues\_PaymentType\_ora12.csv. This file is only applicable to Oracle R12.

1. Identify the Purchase Order Payment Type in your Oracle EBS source system by using the following SQL:

```
SELECT PO_LOOKUP_CODES.LOOKUP_CODE FROM PO_LOOKUP_CODES WHERE PO_LOOKUP_CODES.LOOKUP_TYPE = 'PAYMENT TYPE'
```

2. From the \$pmsserver\lkpfiles folder, open the domainValues\_PaymentType\_ora12.csv file in a text editor.

3. Copy the LOOKUP\_CODE to the XACT\_TYPE\_CODE column in the file.  
The data must be copied starting from the 2nd line.
4. Map each LOOKUP\_CODE to one PO Payment Type (PO\_PAYMENT\_TYPE) domain value.  
Use commas to separate the entries.
5. Save and close the file.

**9.2.2.4.4 To configure domainValues\_Status\_Purch\_Approve\_ora11i.csv** This section explains how to configure domainValues\_Status\_Purch\_Approve\_ora11i.csv.

1. Identify the Purchase Approval Status in your Oracle EBS source system by using the following SQL:

```
SELECT A.LOOKUP_CODE FROM FND_LOOKUP_VALUES A WHERE
A.LOOKUP_TYPE='AUTHORIZATION STATUS' AND A.LANGUAGE = 'US'
```

2. From the domainValues\_Status\_Purch\_Approve\_ora11i.csv file in a text editor.
3. Copy the LOOKUP\_CODE to the STATUS\_CODE column in the file.  
The data must be copied starting from the 8th line.
4. Map each LOOKUP\_CODE to one Purchase Approval (PURCH\_APPROVAL) domain value.  
Use commas to separate the entries.
5. Save and close the file.

**9.2.2.4.5 To configure domainValues\_Status\_Purch\_Cycle\_ora11i.csv** This section explains how to configure domainValues\_Status\_Purch\_Cycle\_ora11i.csv.

1. Identify the Purchase Cycle Status in your Oracle EBS source system by using the following SQL:

```
SELECT A.LOOKUP_CODE FROM FND_LOOKUP_VALUES A WHERE
A.LOOKUP_TYPE='DOCUMENT STATE' AND A.LANGUAGE = 'US'
```

2. From the \$pmsserver\lkpfiles folder, open domainValues\_Status\_Purch\_Cycle\_ora11i.csv file in a text editor.
3. Copy the LOOKUP\_CODE to the STATUS\_CODE column in the file.  
The data must be copied starting from the 8th line.
4. Map each LOOKUP\_CODE to one Purchase Cycle (PURCH\_CYCLE) domain value.  
Use commas to separate the entries.
5. Save and close the file.

**9.2.2.4.6 To configure domainValues\_Xact\_Types\_Purch\_Orders\_ora11i.csv** This section explains how to configure domainValues\_Xact\_Types\_Purch\_Orders\_ora11i.csv.

1. Identify the Purchase Order Types in your Oracle EBS source system by using the following SQL:

```
SELECT DISTINCT PO_DOCUMENT_TYPES_ALL_TL.DOCUMENT_SUBTYPE
FROM PO_DOCUMENT_TYPES_ALL_TL
```

```
WHERE PO_DOCUMENT_TYPES_ALL_TL.LANGUAGE='US' AND PO_DOCUMENT_
TYPES_ALL_TL.DOCUMENT_TYPE_CODE IN ('PO', 'PA') AND PO_
DOCUMENT_TYPES_ALL_TL.DOCUMENT_SUBTYPE <> 'CONTRACT'
```

```
UNION SELECT 'COMPLEXWORK_ACTUAL' FROM DUAL
```

```
UNION SELECT 'COMPLEXWORK_FINANCING' FROM DUAL;
```

2. From the \$pmsserver\lkpfiles folder, open domainValues\_Xact\_Types\_Purch\_Orders\_ora11i.csv file in a text editor.
3. Copy the DOCUMENT\_SUBTYPE to the XACT\_SUBTYPE\_CODE column in the file.

The data must be copied starting from the 8th line.

4. Map each DOCUMENT\_SUBTYPE to one Purchase Order transaction type (PURCH\_ORDERS) domain value.  
Use commas to separate the entries.
5. Save and close the file.

**9.2.2.4.7 To configure domainValues\_Xact\_Types\_Purch\_Requisitions\_ora11i.csv** This section explains how to configure domainValues\_Xact\_Types\_Purch\_Requisitions\_ora11i.csv.

1. Identify the Purchase Requisition Type in your Oracle EBS source system by using the following SQL:

```
SELECT DISTINCT PO_DOCUMENT_TYPES_ALL_TL.DOCUMENT_SUBTYPE
FROM PO_DOCUMENT_TYPES_ALL_TL
```

```
WHERE PO_DOCUMENT_TYPES_ALL_TL.LANGUAGE='US' AND PO_DOCUMENT_
TYPES_ALL_TL.DOCUMENT_TYPE_CODE = 'REQUISITION'
```

2. From the \$pmsserver\lkpfiles folder, open domainValues\_Xact\_Types\_Purch\_Requisitions\_ora11i.csv file in a text editor.
3. Copy the DOCUMENT\_SUBTYPE to the XACT\_SUBTYPE\_CODE column in the file.

The data must be copied starting from the 7th line.

4. Map each DOCUMENT\_SUBTYPE to one Purchase Requisition Type (PURCH\_RQLNS) domain value.  
Use commas to separate the entries.
5. Save and close the file.

**9.2.2.4.8 To configure domainValues\_Xact\_Types\_PO\_Line\_Type\_ora11i.csv** This section explains how to configure domainValues\_Xact\_Types\_PO\_Line\_Type\_ora11i.csv.

1. Identify the Purchase Order Line Type in your Oracle EBS source system by using the following SQL:

```
SELECT DISTINCT PO_LINE_TYPES_V.ORDER_TYPE_LOOKUP_CODE FROM
PO_LINE_TYPES_V
```

2. From the \$pmsserver\lkpfiles folder, open domainValues\_Xact\_Types\_PO\_Line\_Type\_ora11i.csv file in a text editor.
3. Copy the ORDER\_TYPE\_LOOKUP\_CODE to the XACT\_TYPE\_CODE column in the file.

The data must be copied starting from the 8th line.

4. Map each ORDER\_TYPE\_LOOKUP\_CODE to one PO Line Type transaction type (PO\_LINE\_TYPE) domain value.

Use commas to separate the entries.

5. Save and close the file.

**9.2.2.4.9 To configure domainValues\_Xact\_Types\_PO\_Line\_Type\_CODE1\_ora11i.csv** This section explains how to configure domainValues\_Xact\_Types\_Code1\_PO\_Line\_Type\_ora11i.csv.

1. Identify the Purchase Order Line Purchase Basis in your Oracle EBS source system by using the following SQL:

```
SELECT DISTINCT PO_LINE_TYPES_V.PURCHASE_BASIS FROM PO_
LINE_TYPES_V
```

2. From the \$pmsserver\lkpfiles folder, open domainValues\_Xact\_Types\_Code1\_PO\_Line\_Type\_ora11i.csv file in a text editor.

3. Copy the PURCHASE\_BASIS to the XACT\_TYPE\_CODE1 column in the file.

The data must be copied starting from the 8th line.

4. Map each PURCHASE\_BASIS to one PO Line Type Purchase Basis Type (PO\_LINE\_TYPE) domain value.

Use commas to separate the entries.

5. Save and close the file.

**9.2.2.4.10 To configure domainValues\_ShipmentType\_ora12.csv** This section explains how to configure domainValues\_ShipmentType\_ora12.csv. This file is only applicable to Oracle R12.

1. Identify the Purchase Order Shipment Type in your Oracle EBS source system by using the following SQL:

```
SELECT DISTINCT PO_LOOKUP_CODES.LOOKUP_CODE FROM PO_LOOKUP_
CODES
WHERE PO_LOOKUP_CODES.LOOKUP_TYPE = 'SHIPMENT TYPE' AND
PO_LOOKUP_CODES.LOOKUP_CODE IN ( 'STANDARD' , 'PREPAYMENT' ,
'SCHEDULED' , 'BLANKET' )
```

2. From the \$pmsserver\lkpfiles folder, open domainValues\_PaymentType\_ora12.csv file in a text editor.

3. Copy the LOOKUP\_CODE to the XACT\_TYPE\_CODE column in the file. The data must be copied starting from the 2nd line.

4. Map each LOOKUP\_CODE to one PO Shipment Type (PO\_SHIPMENT\_TYPE) domain value. Use commas to separate the entries.

5. Save and close the file.

### 9.2.3 Configuration Steps for Supply Chain For PeopleSoft

This section contains configuration steps that apply to PeopleSoft before you do a full data load.

Not applicable to Oracle BI Applications Version 7.9.4.

## 9.2.4 Configuration Steps for Supply Chain For Universal

This section contains configuration steps that apply to Universal before you do a full data load.

Not applicable to Oracle BI Applications Version 7.9.4.

## 9.2.5 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Supply Chain Analytics.

- [Section 9.2.5.1, "Configuration Steps for Supply Chain For All Source Systems"](#)
- [Section 9.2.5.2, "Configuration Steps for Supply Chain For Oracle EBS"](#)
- [Section 9.2.5.3, "Configuration Steps for Supply Chain For PeopleSoft"](#)
- [Section 9.2.5.4, "Configuration Steps for Oracle Supply Chain For Universal"](#)

### 9.2.5.1 Configuration Steps for Supply Chain For All Source Systems

This section contains configuration steps that apply to all source systems, and contains the following topics:

- [Section 9.2.5.1.1, "About Configuring the Inventory Monthly Balance Table"](#)
- [Section 9.2.5.1.2, "How to Configure the Inventory Monthly Balance Table"](#)
- [Section 9.2.5.1.3, "About Configuring the Product Transaction Aggregate Table"](#)
- [Section 9.2.5.1.4, "How to Configure the Product Transaction Aggregate Table"](#)
- [Section 9.2.5.1.5, "About Configuring the Purchase Receipts Aggregate Table"](#)
- [Section 9.2.5.1.6, "How to Configure the Purchase Receipts Aggregate Table"](#)
- [Section 9.2.5.1.7, "About Configuring the Purchase Cycle Lines Aggregate Table"](#)
- [Section 9.2.5.1.8, "How to Configure the Purchase Cycle Lines Aggregate Table"](#)

**9.2.5.1.1 About Configuring the Inventory Monthly Balance Table** To configure the Inventory Monthly Balance (W\_INVENTORY\_DAILY\_BALANCE\_F\_A1) aggregate table, you need to consider the aggregation level, the time period to update the aggregation, and the time period to keep records in the Inventory Balance table.

You need to configure three parameters to configure the Inventory Monthly Balance table:

- GRAIN
- KEEP\_PERIOD
- NUM\_OF\_PERIOD

The GRAIN parameter has a preconfigured value of Month. The possible values for the GRAIN parameter are:

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

The KEEP\_PERIOD parameter has a preconfigured value of Month. Values for the KEEP\_PERIOD parameter include:

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

The NUM\_OF\_PERIOD parameter has a preconfigured value of 3. The value for the NUM\_OF\_PERIOD parameter is a positive integer, for example, 1, 2, 3, and so on.

**9.2.5.1.2 How to Configure the Inventory Monthly Balance Table** You need to configure the parameterfileDW.txt parameters file, and run the initial ETL session or incremental ETL sessions to load the Inventory Monthly Balance table.

**To configure the parameterfileDW.txt parameters file**

1. Open the parameterfileDW.txt file using a text editor in the OracleBI\DAC\Informatica\parameters\input folder.
2. Replace the default parameter values with your new values.

The default values for the parameterfileDW.txt file are shown in the following table.

**Table 9–2 Default values in the parameterfileDW.txt file**

SESSION	Parameter	Value
PLP_InventoryMonthlyBalance	\$\$GRAIN	'MONTH'
PLP_InventoryDailyBalance_Trim	\$\$KEEP_PERIOD	'MONTH'
PLP_InventoryDailyBalance_Trim	\$\$NUM_OF_PERIOD	3

**Note:** You need to use single quotes for the values of the KEEP\_PERIOD and GRAIN parameters.

3. Save and close the file.

**To incrementally refresh the Inventory Monthly Balance table**

1. Delete the records from the Monthly Balance (W\_INVENTORY\_MONTHLY\_BAL\_F) aggregate table for a certain time.

The GRAIN parameter determines the time period for the deletion. For example, if GRAIN=MONTH, and the date is May 15, 2005, then all records for April and the current month (May) are deleted in the Monthly Balance (W\_INVENTORY\_MONTHLY\_BAL\_F) table.

Running the PLP\_InventoryMonthlyBalance workflow mapping implements this step.

2. Retrieve the records in the Inventory Balance (W\_INVENTORY\_DAILY\_BALANCE\_F) fact table and load the records to the Monthly Balance (W\_INVENTORY\_MONTHLY\_BAL\_F) table at a certain grain level.

For example, if GRAIN=MONTH, then the month end balance records in the W\_INVENTORY\_DAILY\_BALANCE\_F fact table are stored in and aggregated to the Monthly Balance (W\_INVENTORY\_MONTHLY\_BAL\_F).

Running the S\_M\_PLP\_INV\_BALANCE\_A1\_AGG session, and the M\_PLP\_INV\_BALANCE\_A1\_AGG mapping implements this step. For the current month balance, balance records of the previous day (if it is in the same month) are deleted from W\_INVENTORY\_MONTHLY\_BAL\_F, and balance records of the current day will be loaded from W\_INVENTORY\_BALANCE\_F to W\_INVENTORY\_MONTHLY\_BAL\_F.

Running the PLP\_InventoryMonthlyBalance workflow implements this step.

3. Remove the old records from the W\_INVENTORY\_DAILY\_BALANCE\_F fact table.

To remove old records you need to use the KEEP\_PERIOD and the NUM\_OF\_PERIOD parameters. For example, if KEEP\_PERIOD=MONTH, NUM\_OF\_PERIOD=1, and the date is May 15, 2005, then the records for April and the current month (May) are kept and the older records are deleted.

Running the PLP\_InventoryDailyBalance\_Trim workflow implements this step.

---



---

**Note:** The trimming process is to reduce data size in the table. It is important to emphasize that you will not be able to see the old daily balance records. But you will still be able to see the month-end balance. Therefore, please make sure that you adjust the NUM\_OF\_PERIOD values to reflect your data volume and data recency requirement.

---



---

**9.2.5.1.3 About Configuring the Product Transaction Aggregate Table** There are two aggregation scenarios to configure the Product Transaction aggregate (W\_PRODUCT\_XACT\_A) table—the initial ETL run and then the incremental ETL run.

For your initial ETL run, you need to configure the aggregation level, and the length of history kept in the Product Transaction fact table.

For your initial ETL run, you need to configure the aggregation grain, using the GRAIN parameter.

For the incremental ETL run, you need to configure the aggregation level, the update period in aggregation, and the length of history kept in the Product Transaction fact table, using the following parameters:

- **GRAIN**  
The GRAIN parameter specifies the aggregation level. Valid values are DAY, WEEK, MONTH (preconfigured value), QUARTER, YEAR.
- **REFRESH\_PERIOD**  
The REFRESH\_PERIOD parameter, together with NUM\_OF\_PERIOD, indicates the number of period of records that will be refresh from the transaction table to the aggregate table. Valid values are DAY, WEEK, MONTH (preconfigured value), QUARTER, YEAR.
- **NUM\_OF\_PERIOD**  
The NUM\_OF\_PERIOD parameter, together with REFRESH\_METHOD, indicates the number of period of records that will be refresh from the transaction table to the aggregate table. Valid values are positive integers, for example, 1, 2, 3 (preconfigured value).

**9.2.5.1.4 How to Configure the Product Transaction Aggregate Table** You need to configure the `parameterfileDW.txt` parameters file, and run the initial ETL and then the incremental ETL to load the Product Transaction aggregate table.

**To configure the parameterfileDW.txt parameters file**

1. Open the `parameterfileDW.txt` file using a text editor in the `OracleBI\DAC\Informatica\parameters\input` folder.
2. Replace the default parameter values with your new values.

The default values are shown in the following table.

**Table 9–3 Default values in the parameterfileDW.txt file**

SESSION	PARAMETER	VALUE
PLP_ProductTransactionAggregate	\$\$REFRESH_PERIOD	'MONTH'
PLP_ProductTransactionAggregate	\$\$GRAIN	'MONTH'
PLP_ProductTransactionAggregate	\$\$NUM_OF_PERIOD	3
PLP_ProductTransactionAggregate_Full	\$\$GRAIN	'MONTH'

**Note:** You need to use single quotes for the values of the `KEEP_PERIOD`, `GRAIN`, and `REFRESH_PERIOD` parameters. The `KEEP_PERIOD` value must be equal to or greater than the `GRAIN` value. The `REFRESH_PERIOD` value must equal the `GRAIN` value.

3. Save and close the file.

**To configure the Product Transaction aggregate table for the initial ETL run**

1. Retrieve the records in the Product Transaction fact (`W_PRODUCT_XACT_F`) table, and aggregate the records to the Product Transaction aggregate (`W_PRODUCT_XACT_A`) table at a certain grain level.

For example, if `GRAIN=MONTH` then the records in the `W_PRODUCT_XACT_F` fact table are retrieved and aggregated to the `W_PRODUCT_XACT_A` table at a monthly level.

Running the `PLP_ProductTransactionAggregate_Full` workflow implements this step.

**To configure the Product Transaction aggregate table for the incremental ETL run**

1. Delete the refreshed records from the Product Transaction aggregate (`W_PRODUCT_XACT_A`) table for a certain time.

The `REFRESH_PERIOD` and the `NUM_OF_PERIOD` parameters determine the time period for the deletion.

For example, if `REFRESH_PERIOD=MONTH`, `NUM_OF_PERIOD=1`, and the date is May 15, 2005, then all records for April and the current month (May) are deleted in the `W_PRODUCT_XACT_A` table.

Running the `PLP_ProductTransactionAggregate` workflow implements this step.

2. Retrieve the records in the Product Transaction fact (`W_PRODUCT_XACT_F`) table, and aggregate the records to the `W_PRODUCT_XACT_A` table at a certain grain level.

For example, if GRAIN=MONTH then the records in the W\_PRODUCT\_XACT\_F fact table are retrieved and aggregated to the W\_PRODUCT\_XACT\_A table at a monthly level.

Running the PLP\_ProductTransactionAggregate workflow implements this step.

**9.2.5.1.5 About Configuring the Purchase Receipts Aggregate Table** The Purchase Receipts aggregate table (W\_PURCH\_RCPT\_A) is used to capture information about the product receipts received from your suppliers and the purchase orders placed on them by your purchasing organization.

For your initial ETL run, you need to configure the GRAIN parameter for the time aggregation level in the Purchase Receipts Aggregate fact table.

For the incremental ETL run, you need to configure the time aggregation level and the source identification. The source identification value represents the source system you are sourcing data from.

You need to configure two parameters to aggregate the Purchase Receipts table for your incremental run:

- GRAIN
- \$\$TIME\_GRAIN

These parameters have a preconfigured value of Month. The possible values for the parameter are:

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

The Purchase Receipt Lines aggregate table is fully loaded from the base table in the initial ETL run. The table can grow to millions of records. Thus, the Purchase Receipts aggregate table is not fully reloaded from the base table after each incremental ETL run. Oracle Business Analytics Warehouse minimizes the incremental aggregation effort, by modifying the aggregate table incrementally as the base table is updated. This process is done in four steps:

1. Oracle Business Analytics Warehouse finds the records to be deleted in the base table since the last ETL run, and loads them into the W\_PURCH\_RCPT\_TMP table. The measures in these records are multiplied by (-1). The mapping responsible for this task is suffixed with \_Derive\_PreSoftDeleteImage, and it is run before the records are deleted from the base table. The mapping is run in the source-specific workflow.
2. Oracle Business Analytics Warehouse finds the records to be updated in the base table since the last ETL run, and loads them into the W\_PURCH\_RCPT\_TMP table. The measures in these records are multiplied by (-1). The mapping responsible for this task is suffixed with \_Derive\_PreLoadImage, and it is run before the records are updated in the base table. It is run in the source-specific workflow.
3. Oracle Business Analytics Warehouse finds the inserted or updated records in the base table since the last ETL run, and loads them into the W\_PURCH\_RCPT\_TMP table, without changing their sign. The mapping responsible for this task is suffixed with \_Derive\_PostLoadImage, and it is run after the records are updated or inserted into the base table. It is run in the post load-processing workflow.

4. Oracle Business Analytics Warehouse aggregates the W\_PURCH\_RCPT\_TMP table, and joins it with the W\_PURCH\_RCPT\_A aggregate table to insert new or update existing buckets to the aggregate table. This step is part of the post load-processing workflow, and the mapping is suffixed with \_Derive.

**9.2.5.1.6 How to Configure the Purchase Receipts Aggregate Table** To load the Purchase Receipts aggregate table (W\_PURCH\_RCPT\_A), you need to configure the post-load-processing parameter file and the source system parameter files, and run the initial workflow and then the incremental workflow.

**To configure the parameterfiledw.txt parameters file**

1. Open the parameterfiledw.txt file using a text editor in the OracleBI\DAC\Informatica\parameters\input folder.
2. Replace the default parameter values with your new values.
3. Save and close the file.

**To configure the Purchase Receipts aggregate table for Universal Source**

1. Open the parameterfileDW.txt file using a text editor in the OracleBI\DAC\Informatica\parameters\input folder.
2. Replace the default parameter values with your new values.
3. Save and close the file.

**9.2.5.1.7 About Configuring the Purchase Cycle Lines Aggregate Table** To aggregate the Purchase Cycle Lines table (W\_PURCH\_CYCLE\_LINE\_A), you need to configure the parameterfiledw.txt parameters file and the source system parameters file, and run the initial ETL workflow and then the incremental ETL workflow.

For your initial ETL run, you need to configure the GRAIN parameter for the time aggregation level in the Purchase Cycle Lines Aggregate fact table.

For the incremental ETL run, you need to configure the time aggregation level and the source identification. The source identification value represents the source system you are sourcing data from.

You need to configure two parameters to aggregate the Purchase Cycle Lines table for your incremental run:

- GRAIN
- \$TIME\_GRAIN

These parameters has a preconfigured value of Month. The possible values for parameters are:

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

The Purchase Cycle Lines aggregate table is fully loaded from the base table in the initial ETL run. The table can grow to millions of records. The Purchase Cycle Lines aggregate table is not fully reloaded from the base table after an ETL run. Oracle Business Analytics Warehouse minimize the incremental aggregation effort, by

modifying the aggregate table incrementally as the base table gets updated. This process is done in four steps:

1. Oracle Business Analytics Warehouse finds the records to be deleted in the base table since the last ETL run, and loads them into the `W_PURCH_CYCLE_LINE_TMP` table. The measures in these records are multiplied by (-1). The mapping responsible for this task is suffixed with `_Derive_PreSoftDeleteImage`, and it is run before the records are deleted from the base table. It is run in the source-specific workflow.
2. Oracle Business Analytics Warehouse finds the records to be updated in the base table since the last ETL run, and loads them into the `W_PURCH_CYCLE_LINE_TMP` table. The measures in these records are multiplied by (-1). The mapping responsible for this task is suffixed with `_Derive_PreLoadImage`, and it is run before the records are updated in the base table. It is run in the source-specific workflow.
3. Oracle Business Analytics Warehouse finds the inserted or updated records in the base table since the last ETL run, and loads them into the `W_PURCH_CYCLE_LINE_TMP` table, without changing their sign. The mapping responsible for this task is suffixed with `_Derive_PostLoadImage`, and it is run after the records are updated or inserted into the base table. It is run in the post load-processing workflow.
4. Oracle Business Analytics Warehouse aggregates the `W_PURCH_CYCLE_LINE_TMP` table, and joins it with the `W_PURCH_CYCLE_LINE_A` aggregate table to insert new or update existing buckets to the aggregate table. This step is part of the post load-processing workflow, and the mapping is suffixed with `_Derive`.

**9.2.5.1.8 How to Configure the Purchase Cycle Lines Aggregate Table** To load the Purchase Cycle Lines aggregate table (`W_PURCH_CYCLE_LINE_A`), you need to configure the post-load-processing parameter file and the source system parameter files, and run the initial workflow and then the incremental workflow.

**To configure the Purchase Cycle Lines Aggregate Table**

1. Open the `parameterfiledw.txt` file using a text editor in the `OracleBI\DAC\Informatica\parameters\input` folder.
2. Replace the default parameters values with your new values.
3. Save and close the file.

**9.2.5.2 Configuration Steps for Supply Chain For Oracle EBS**

This section contains configuration steps that apply to Oracle EBS, and contains the following topics:

- [Section 9.2.5.2.1, "How to Configure the Bill of Materials Explosion Options"](#)
- [Section 9.2.5.2.2, "How to Configure the Left Bound and Right Bound Calculation Option"](#)
- [Section 9.2.5.2.3, "How to Extract Particular Purchase Order Records"](#)

**9.2.5.2.1 How to Configure the Bill of Materials Explosion Options** The Bill of Materials (BOM) functional area allows you to determine the profit margin of the components that comprise the finished goods. BOM allows you to keep up with the most viable vendors in terms of cost and profit, and to keep your sales organization aware of product delivery status, including shortages.

You can explode the BOM structure with three different options:

- **All.** All the BOM components are exploded regardless of their effective date or disable date. To *explode* a BOM component is to expand the BOM tree structure.
- **Current.** The incremental extract logic considers any changed components that are currently effective, any components that are effective after the last extraction date, or any components that are disabled after the last extraction date.
- **Current and Future.** All the BOM components that are effective now or in the future are exploded. The disabled components are left out.

These options are controlled by the `EXPLODE_OPTION` variable. The `EXPLODE_OPTION` variable is preconfigured with a value of 2, explode Current BOM structure.

In the `SDE_ORA_BomItemFact` mapping, the `bompexpl.exploder_userexit` stored procedure explodes the BOM structure. The table below lists the variables for the `bompexpl.exploder_userexit` stored procedure.

**Table 9-4 Variables for the `bompexpl.exploder_userexit` Stored Procedure**

Input Variable	Preconfigured Value	Description
<code>VERIFY_FLAG</code>	0	A verify flag with a value of 1 only applies to standard BOM.
<code>ORG_ID</code>	<code>ORGANIZATION_ID</code>	Organization ID
<code>ORDER_BY</code>	1	Controls the order of the records. 1—Operation Sequence Number, Item Number 2—Item Number, Operation Sequence Number
<code>GRP_ID</code>	Negative Sequence ID -1, -2, and so on.	Unique value to identify current explosion.
<code>SESSION_ID</code>	Negative Sequence ID -1, -2, and so on.	Unique value to identify current session.
<code>LEVELS_TO_EXPLODE</code>	10	Levels to explode.
<code>BOM_OR_ENG</code>	1	1—BOM 2—ENG
<code>IMPL_FLAG</code>	1	1—Implemented Only 2—Implemented and Non-implemented
<code>PLAN_FACTOR</code>	2	1—Yes 2—No
<code>EXPLODE_OPTION</code>	2	1—All 2—Current 3—Current and Future
<code>MODULE</code>	2	1—Costing 2—BOM 3—Order Entry 4—ATO 5—WSM
<code>CST_TYPE_ID</code>	0	Cost type ID for cost explosion.

**Table 9–4 (Cont.) Variables for the `bompexpl.exploder_userexit` Stored Procedure**

Input Variable	Preconfigured Value	Description
STD_COMP_FLAG	0	1—Explode only standard components 2—All components
EXPL_QTY	1	Explosion quantity
ITEM_ID	ROUND(TO_DECIMAL(PRODUCT_ID))	Item ID of assembly to explode.
ALT_DESG	ALTERNATE_BOM_DESIGNATOR	Alternate routing designator
COMP_CODE	NULL	Concatenated component code
REV_DATE	TO_CHAR(CREATION_DT, 'YYYY/MM/DD HH24:MI')	Explosion date YYYY/MM/DD HH24:MI

There are five different BOM types in a source system—1- Model, 2 - Option Class, 3 - Planning, 4 - Standard, and 5 - Product Family. By default, only the Standard BOM type is extracted and exploded.

**To configure the BOM explosion to the All option**

1. In PowerCenter Designer, open the SDE\_ORA<Ver>\_adapter.
2. Open the SDE\_ORA\_BomItemFact mapping.
3. Double-click the EXP\_BOMPexpl\_EXPLoder\_USEREXIT expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the EXPLODE\_OPTION port.
4. Change the value to 1, and click Apply.
5. Validate the mapping, and save your changes to the repository.
6. Go to Mapplet Designer, and open mplt\_BC\_ORA\_BOMHeaderDimension.
7. Double-click the SQL qualifier SQ\_BOM\_INVENTORY\_COMPONENTS to open the Edit Transformations dialog box, and click on Properties tab, open value for SQL Query.
8. Modify the following default Where condition from:

```
((
    /* CURRENT valid component changed */

    INV.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY
    HH24:MI:SS') AND (INV.EFFECTIVITY_DATE <= TO_DATE('$$CURRENT_DATE', 'MM/DD/YYYY
    HH24:MI:SS') and (INV.DISABLE_DATE > TO_DATE('$$CURRENT_DATE', 'MM/DD/YYYY
    HH24:MI:SS') OR INV.DISABLE_DATE IS NULL))

    OR

    /* Component that became effective after last extract date and before
    today's extract, for CURRENT Option*/

    INV.EFFECTIVITY_DATE between TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY
    HH24:MI:SS') and TO_DATE('$$CURRENT_DATE', 'MM/DD/YYYY HH24:MI:SS')

    OR
```

```

/* Component that become disabled after last extract date and before today's
extract, for CURRENT and CURRENT-FUTURE Option*/

```

```

INV.DISABLE_DATE between TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY
HH24:MI:SS') and TO_DATE('$$CURRENT_DATE', 'MM/DD/YYYY HH24:MI:SS')

```

```

)

```

```

OR BOM.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY
HH24:MI:SS')

```

```

GROUP BY

```

To:

```

(INV.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY
HH24:MI:SS')

```

```

OR BOM.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY
HH24:MI:SS')

```

```

GROUP BY

```

9. Click Apply, and Validate the mapping and save your changes to the repository.

#### To configure the BOM explosion to the Current and Future option

1. In PowerCenter Designer, open the SDE\_ORA<Ver>\_adapter.
2. Open the SDE\_ORA\_BOMItemFact mapping.
3. Double-click the EXP\_BOMPEXPL\_EXPLODER\_USEREXIT expression transformation to open the Edit Transformations dialog box, and click the Port tab to display the EXPLODE\_OPTION port.
4. Change the value to 3, and click Apply.
5. Validate the mapping, and save your changes to the repository.
6. Go to Mapplet Designer, and open mplt\_BC\_ORA\_BOMHeaderDimension.
7. Double-click the SQL qualifier SQ\_BOM\_INVENTORY\_COMPONENTS to open the Edit Transformations dialog box, and click on Properties tab, open value for SQL Query.
8. Modify the following default Where condition from:

```

((

```

```

/* CURRENT valid component changed */

```

```

INV.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY HH24:MI:SS')
AND (INV.EFFECTIVITY_DATE <= TO_DATE('$$CURRENT_DATE', 'MM/DD/YYYY HH24:MI:SS')
and (INV.DISABLE_DATE > TO_DATE('$$CURRENT_DATE', 'MM/DD/YYYY HH24:MI:SS') OR
INV.DISABLE_DATE IS NULL))

```

```

OR

```

```

/* Component that became effective after last extract date and before today's
extract, for CURRENT Option*/

```

```

INV.EFFECTIVITY_DATE between TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY
HH24:MI:SS') and TO_DATE('$$CURRENT_DATE', 'MM/DD/YYYY HH24:MI:SS')

```

```

OR

/* Component that become disabled after last extract date and before today's
extract, for CURRENT and CURRENT-FUTURE Option*/

INV.DISABLE_DATE between TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY HH24:MI:SS')
and TO_DATE('$$CURRENT_DATE', 'MM/DD/YYYY HH24:MI:SS')

)

OR BOM.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY
HH24:MI:SS')

GROUP BY

To:

((

INV.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY HH24:MI:SS')

AND ((INV.DISABLE_DATE > TO_DATE('$$CURRENT_DATE', 'MM/DD/YYYY HH24:MI:SS') OR
INV.DISABLE_DATE IS NULL))

OR

INV.DISABLE_DATE between TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY HH24:MI:SS')
and TO_DATE('$$CURRENT_DATE', 'MM/DD/YYYY HH24:MI:SS')

)

OR BOM.LAST_UPDATE_DATE > TO_DATE('$$LAST_EXTRACT_DATE', 'MM/DD/YYYY
HH24:MI:SS')

GROUP BY

```

9. Click Apply, and Validate the mapping and save your changes to the repository.

**To configure the BOM type**

1. In PowerCenter Designer, open the SDE\_ORA<Ver>\_adapter.
2. Open the mplt\_BC\_ORA\_BOMHeaderDimension mapplet.
3. Double-click the SQL qualifier SQ\_BOM\_INVENTORY\_COMPONENTS to open the Edit Transformations dialog box, and click on Properties tab, open value for SQL Query.
4. Modify BOM\_ITEM\_TYPE section in Where statement.

Change the number to your BOM type. For example, change the number to 3 for a Planning BOM type.

```

Where INV.BOM_ITEM_TYPE = 3 AND
M.BOM_ITEM_TYPE = 3 AND

```

---

**Note:** You can also remove these two filters to extract all types of BOM.

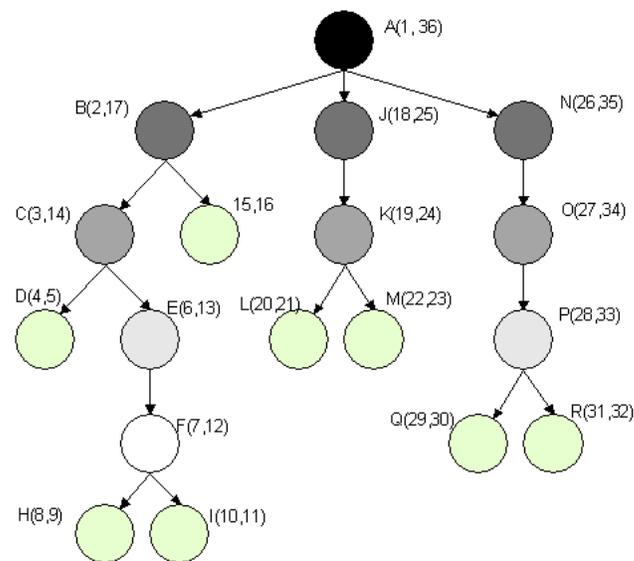
---

5. Click Apply, and Validate the mapping and save your changes to the repository.

**9.2.5.2.2 How to Configure the Left Bound and Right Bound Calculation Option** You can use the left bound and the right bound calculation to expedite some reporting requirements. For example, you can find the components in a subassembly within a finished product. Left bound and right bound are stored in the `W_BOM_ITEM_F` table for each BOM node, and they have one row of data in the `W_BOM_ITEM_F` table. The `COMPUTE_BOUNDS` stored procedure traverses the exploded BOM tree structure and calculates the left bound and right bound. By default, the `COMPUTE_BOUNDS` stored procedure is off and the `W_BOM_ITEM_F.LEFT_BOUNDS` and `W_BOM_ITEM_F.RIGHT_BOUNDS` columns are empty.

The figure below illustrates a sample BOM structure with the left bound and right bound values listed for each node. To find all the components under node B, you select the components with a top product key value of A, the left bound value is greater than 2, and the right bound value is less than 17.

**Figure 9–1 Sample BOM Structure**



You can use the following procedure to turn on the left bound and the right bound calculation and populate the `W_BOM_ITEM_F.LEFT_BOUNDS` and `W_BOM_ITEM_F.RIGHT_BOUNDS` columns.

**To configure the left bound and right bound calculation option**

1. In PowerCenter Designer, open the `SDE_ORA<Ver>_adapter`.
2. Open the `SIL_BOMItemFact` mapping.
3. Double-click the `COMPUTE_BOUNDS` stored procedure transformation to open the Edit Transformations dialog box, and click the Properties tab.
4. Change the value for Transformation Attributes Call Text to `compute_bounds_ora11i(1)`.
5. Click Apply.
6. Validate the mapping, and save your changes to the repository.

**9.2.5.2.3 How to Extract Particular Purchase Order Records** This configuration also applies to the Spend functional area.

You may not want to extract particular types of records from purchase orders in your source system. In these cases, you can modify the filter condition in the Source Qualifier of the maplet. By default, the filter condition is set to PLANNED, BLANKET, or STANDARD. However, you can change this value to some conditional statement that only allows particular types of records to be extracted.

**To extract particular types of purchase order records**

1. In PowerCenter Designer, open the SDE\_ORA115<Ver>\_adapter.
2. Open the MPLT\_BC\_ORA\_PURCHASEORDERFACT maplet.
3. Double-click the Source Qualifier to open the Edit Transformations box, and select the Properties tab to display the SQL Query.
4. Double-click the value in the SQL Query to open the SQL Editor box and edit the statement.
5. Replace the prepackaged filter condition with the new filter statement that reflects your business needs.
6. Edit the WHERE clause of the statement.
7. Click Apply to save the changes, and click OK to exit.
8. Validate the expression, and save your changes to the repository.
9. Repeat Step 2 to Step 8 for the MPLT\_BC\_ORA\_PURCHASESCHEDULELINESFACT maplet.

**9.2.5.3 Configuration Steps for Supply Chain For PeopleSoft**

This section contains configuration steps that apply to PeopleSoft.

Not applicable to Oracle BI Applications Version 7.9.4.

**9.2.5.4 Configuration Steps for Oracle Supply Chain For Universal**

This section contains configuration steps that apply to Universal, and contains the following topics:

- [Section 9.2.5.4.1, "About Configuring Supply Chain Analytics for Universal Source"](#)
- [Section 9.2.5.4.2, "How to Configure the Preferred Merchant Flag"](#)
- [Section 9.2.5.4.3, "How to Configure the Customer Billable Indicator"](#)
- [Section 9.2.5.4.4, "How to Configure the Receipts Indicator"](#)
- [Section 9.2.5.4.5, "How to Configure Expense Payment Types"](#)
- [Section 9.2.5.4.6, "How to Configure the Default Expense Distribution Percentage"](#)
- [Section 9.2.5.4.7, "How to Configure Lookup Dates for Currency Conversion"](#)

**9.2.5.4.1 About Configuring Supply Chain Analytics for Universal Source** Expenses has one fact table (W\_EXPENSE\_F) that supports metrics and reports for examining employee expenses. Several mappings populate these tables to complete extracts, loads and updates; you may configure these to suit your organization's business rules. The following sections discuss decisions you must make before you begin adapting individual PowerCenter objects, and provide specific configuration procedures for the universal source.

Universal Source Adapter mapplets extract data from a flat file interface to populate the Oracle Business Analytics Warehouse. In this phase of your project, you can configure the following:

- **System Flags and Indicators.** You may configure various system flags to indicate record rejection settings, as well as to indicate if your employees are using your preferred vendors, if you can forward expenses to your customers, and if receipts are available for expensed items.
- **Currency and Payment Options.** You may configure the date used to establish your exchange rates, determine if you allow expenses to be distributed across multiple cost centers, and define payment types in your data warehouse.

Before you begin, you must make the following decisions:

- **Cash Advances.** Cash advance records have a unique expense item number. If your system allows multiple cash advance records for one expense report, each of these advances must have their own identifiers.
- **Violations.** Many organizations capture violations of company expense policies at the item level (for example, the line item airfare exceeds \$2000), cash advance level (for example, cash advance exceeds \$500) and at the expense report level (for example, the report's total expenses exceed \$5000). Currently the Oracle Business Analytics Warehouse stores item level violations within the corresponding item record, but the cash advance record stores both cash advance and report-level violations. Furthermore, each record has a `VIOLATION_WID` that can point to `W_REASON_D`, where violation details are stored. Depending on how you want your analytic system to perform, you must edit your universal business adapter file to reflect the violation counts and keys appropriately. For example:
  - If a requestor violates a cash advance policy, but there are no other violations at the report level, the `VIOLATION_ID` refers to the cash advance violation only. The violation count equals the cash advance violation counts.
  - If a requestor violates company policy with their expense report, but has not taken a cash advance, you must add a dummy record in the flat file for a cash advance and set the cash advance amount to zero, and enter the violation count as the total number of expense report violations. In this scenario, `VIOLATION_ID` refers to the expense report violation data only.
  - If a requestor violates a cash advance policy and an expense report policy, you must total the violation counts and enter them in your flat file record, and the `VIOLATION_ID` has no value. However, if your organization wants to prioritize the violations and have the `VIOLATION_ID` point to that which is most important, you may point it to the appropriate entry in `W_REASON_D`.

**9.2.5.4.2 How to Configure the Preferred Merchant Flag** The Oracle Business Analytics Warehouse provides a preferred merchant flag to indicate whether the requestor used a preferred merchant for an expensed item. The flag can have only one value—Y (item acquired from a preferred merchant) or N (item acquired from a merchant not recorded). If you use custom logic to determine merchant status, you must include that logic in the expenses Source Adapter.

#### To configure the preferred merchant flag

1. In PowerCenter Designer, open the Configuration for Universal Source folder.
2. Open the `SDE_Universal_ExpenseFact` mapping.
3. Select the Expression transformation to open the Edit Transformations box and select the Port tab.

4. Select the Add Port icon to add the `PREF_MERCHANT_FLAG` port.
5. Enter preferred merchant flag logic.
6. Validate the mapplet and click OK to exit.
7. Save your changes to the repository.

**9.2.5.4.3 How to Configure the Customer Billable Indicator** The Oracle Business Analytics Warehouse provides a customer billable indicator that registers whether an expense item is billed to a customer or paid by your organization. The flag can have only one value—Y (cost is passed to the customer) or N (cost is paid by your organization). If you use custom logic to determine customer billable status, you must include that logic in the expenses Source Adapter.

**To configure the customer billable indicator**

1. In PowerCenter Designer, open the Configuration for Universal Source folder.
2. Open the `SDE_Universal_ExpenseFact` mapping.
3. Select the Expression transformation to open the Edit Transformations box, and select the Port tab.
4. Add a port called `CUST_BILLABLE_FLG = <insert your expression here>`.
5. Validate the mapplet and click OK to exit.
6. Save your changes to the repository.

**9.2.5.4.4 How to Configure the Receipts Indicator** The Oracle Business Analytics Warehouse provides a receipts indicator that registers whether requestors have submitted a receipt for a line item in their expense report. The flag can have only one value—Y (receipts are available) or N (receipts are not available). If you use custom logic to indicate receipt availability, you must include that logic in the expenses Source Adapter.

**To configure the receipts indicator**

1. In PowerCenter Designer, open the `SDE_Universal_ExpenseFact` mapping.
2. Open the `MPLT_SAF_EXPENSES` Source Adapter mapplet.
3. Select the Expression transformation to open the Edit Transformations box, and select the Port tab.
4. Add a port called `RECEIPT_FLG = <insert your expression here>`.
5. Validate the mapplet and click OK to exit.
6. Save your changes to the repository.

**9.2.5.4.5 How to Configure Expense Payment Types** The Oracle Business Analytics Warehouse supports analysis on three types of payment—Reimbursable Expense (type E), expenses prepaid by your company (type P), and cash advance (type C). All of your organization's payment types must be mapped to one of these types described earlier; do this by modifying `MPLT_SAF_EXPENSES`.

**To configure additional payment types**

1. In PowerCenter Designer, open the `SDE_Universal_ExpenseFact` mapping.
2. Open the `MPLT_SAF_EXPENSES` Source Adapter mapplet.

3. Select the Expression transformation to open the Edit Transformations box, and select the Port tab to display the EXPENSE\_PAY\_TYPE\_ID port.
4. Copy and paste this port, rename it EXPENSE\_PAY\_TYPE\_ID.
5. Select O to make your new port an output port.
6. Add a decode logic in the expression to decode source-supplied values to the Oracle Business Analytics Warehouse payment type of your choice.
7. Validate your mapping, and save your changes to the repository.

**9.2.5.4.6 How to Configure the Default Expense Distribution Percentage** At times, employee expenses may be distributed across multiple cost centers. For example, technical support associates frequently travel to work in an office with many cost centers; their expenses could be split between those who used their services. This cost center distribution is expected as a percentage from the source system or file; if it is not present a null value is returned. However, this prevents further calculations, so it is preferable to configure the default to be 100% if only one cost center is charged, rather than allow the system to return a null value.

**To configure the default expense distribution percentage**

1. In PowerCenter Designer, open the Configuration for Universal Source folder.
2. Open the SDE\_Universal\_ExpenseFact mapping.
3. Select the Expression transformation to open the Edit Transformations box and select the Port tab.
4. Add a port named DIST\_PERCENTAGE = <expression that sets this to 100%>.
5. Validate your mapplet.
6. Save your changes.

**9.2.5.4.7 How to Configure Lookup Dates for Currency Conversion** The Oracle Business Analytics Warehouse supports conversion of currency to document (transactional, or source, currency) and group (corporate umbrella currency) for exchange rates. The Oracle Business Analytics Warehouse uses a specific lookup date to determine the rate of exchange on the date an expense was incurred (ACTUAL\_EXP\_DT). If you decide to use a different date as your currency conversion lookup, you must use the following procedure.

**To configure the exchange rate lookup date**

1. In PowerCenter Designer, open the Configuration for Universal Source folder.
2. Open the SDE\_Universal\_ExpenseFact mapping.
3. Select the Expression transformation to open the Edit Transformations box, and select the Port tab to display the EXT\_XRATE\_LKP\_DATE port.  
The expression contains the exchange rate lookup date logic.
4. Select the expression in the EXT\_XRATE\_LOOKUP\_DATE port to open the Expression Editor box and edit the expression.
5. Edit the lookup date logic by substituting your lookup date for the prepackaged expression.
6. Validate the mapplet and click OK to exit.
7. Save your changes.



---

## Configuring Oracle Financial Analytics

This chapter describes how to configure Oracle Financial Analytics for particular sources to meet your business needs.

This chapter contains the following topics:

- [Section 10.1, "Overview of Oracle Financial Analytics"](#)
- [Section 10.2, "Configuration Required Before A Full Load for Financial Analytics"](#)

### 10.1 Overview of Oracle Financial Analytics

Oracle Financial Analytics consists of the following:

- **Oracle General Ledger and Profitability Analytics.** The General Ledger Analytics application provides information to support your enterprise's balance sheet and provides a detailed analysis on the overall health of your company. The default configuration for the General Ledger Analytics application is based on what is identified as the most-common level of detail or granularity. However, you can configure and modify the extracts to best meet your business requirements.

The Profitability Analytics application provides cost analysis, revenue trends, and sales performance to provide an accurate picture of profit and loss. The information found in the Profitability Analytics application pertains to data found in the revenue and expense account groupings of your financial statements and chart of accounts. The Profitability Analytics application is designed to provide insight into your enterprise's revenue and profitability information, which ties into your accounts receivable. The default configuration for the Profitability Analytics application is based on what is identified as the most-common level of detail, or granularity. However, the extracts are configurable and you can modify the extracts to meet your business requirements. The Profitability Analytics application provides cost analysis, revenue trends, and profitability analysis at the products and customer level, and the income statement at the company and business divisions level.

- **Oracle Payables Analytics.** The Oracle Payables Analytics application provides information about your enterprise's accounts payable information and identifies the cash requirements to meet your obligations.

The information found in the Oracle Payables Analytics application pertains to data found exclusively under Accounts Payable (AP) in your financial statements and chart of accounts. Analysis of your payables allows you to evaluate the efficiency of your cash outflows. The need for analysis is increasingly important because suppliers are becoming strategic business partners with the focus on increased efficiency for just in time, and quality purchasing relationships.

The default configuration for the Oracle Payables Analytics application is based on what is identified as the most- common level of detail, or granularity. However, you can configure or modify the extracts to best meet your business requirements.

- **Oracle Receivables Analytics.** The Oracle Receivables Analytics application provides information to support your credit and collection activities, and to monitor and identify potential, receivables problems.

The information found in the Oracle Receivables Analytics application pertains to data found exclusively in the Accounts Receivable (AR) account grouping of your financial statements and chart of accounts. Each day that your receivables are past the due date represents a significant, opportunity-cost to your company. Keeping a close eye on the trends, and clearing of AR is one way to assess the efficiency of your sales operations, the quality of your receivables, and the value of key customers.

The default configuration for the Oracle Receivables Analytics application is based on what is identified as the most-common level of detail or granularity. However, you may configure and modify the extracts to best meet your business requirements.

## 10.2 Configuration Required Before A Full Load for Financial Analytics

This section contains configuration steps that you need to perform on Oracle Financial Analytics before you do a full data load, and contains the following topics:

- [Section 10.2.1, "Configuration Steps for Financial Analytics For All Source Systems"](#)
- [Section 10.2.2, "Configuration Steps for Financial Analytics For Oracle EBS"](#)
- [Section 10.2.3, "Configuration Steps for Financial Analytics For PeopleSoft"](#)
- [Section 10.2.4, "Configuration Steps for Financial Analytics For Universal"](#)
- [Section 10.2.5, "Configuration Steps for Controlling Your Data Set"](#)

### 10.2.1 Configuration Steps for Financial Analytics For All Source Systems

This section contains configuration steps before you do a full data load that apply to all source systems.

---

---

**Note:** For configuration steps that apply to all analytics modules (for example, Oracle Financial Analytics, Oracle HR Analytics, Oracle Sales Analytics, see [Chapter 8, "Configuring Common Areas and Dimensions"](#)).

---

---

### 10.2.2 Configuration Steps for Financial Analytics For Oracle EBS

This section contains configuration steps before you do a full data load that apply to Oracle EBS, and contains the following topics:

- [Section 10.2.2.1, "About Configuring Domain Values and CSV Worksheet Files for Oracle Financial Analytics"](#)
- [Section 10.2.2.2, "How to Map Oracle General Ledger Natural Accounts to Group Account Numbers"](#)
- [Section 10.2.2.3, "How to do Group Account correction for Oracle Application"](#)

- [Section 10.2.2.4, "How to configure domainValues\\_Xact\\_Types\\_DocTypes\\_ora12.csv for Oracle General Ledger and Profitability Analytics"](#)
- [Section 10.2.2.5, "How to configure domainValues\\_Xact\\_Types\\_DocTypes\\_ora11i.csv for Oracle General Ledger and Profitability Analytics"](#)

### 10.2.2.1 About Configuring Domain Values and CSV Worksheet Files for Oracle Financial Analytics

If you modify or extend a seeded list of values, you must configure the CSV files for Oracle Financial Analytics by mapping values from your source systems to the domain values.

The following three sections explains how to extract the lists of values from your source system, which you then compare with the seeded values. If the lists of values are different to the seeded values, you need to follow the instructions to configure the Domain Values and CSV Worksheet Files.

[Table 10–3](#) lists the CSV worksheet files and the domain values for Financial Analytics and Oracle EBS in the \$pmsserver\LkpFiles folder.

**Table 10–1 Domain Values and CSV Worksheet Files for Oracle Financial Analytics and Oracle EBS**

Worksheet File Name	Description	Session
domainValues_Xact_Types_DocTypes_ora12.csv	List the Transaction types and the corresponding domain values for the Oracle EBS R12 application.  For more information about updating the values in this file, see <a href="#">Section 10.2.2.4, "How to configure domainValues_Xact_Types_DocTypes_ora12.csv for Oracle General Ledger and Profitability Analytics"</a> .	SDE_ORA_TransactionTypeDimension_GLCOGSDerive
domainValues_Xact_Types_DocTypes_ora11i.csv	List the Transaction types and the corresponding domain values for the Oracle EBS 11i application.  For more information about updating the values in this file, see <a href="#">Section 10.2.2.5, "How to configure domainValues_Xact_Types_DocTypes_ora11i.csv for Oracle General Ledger and Profitability Analytics"</a> .	SDE_ORA_TransactionTypeDimension_GLCOGSDerive

### 10.2.2.2 How to Map Oracle General Ledger Natural Accounts to Group Account Numbers

---

**Note:** It is critical that the General Ledger Account Numbers are mapped to the Group Account Numbers (or domain values) as the metrics in the General Ledger reporting layer uses these values. For a list of domain values for General Ledger Account Numbers, see *Oracle Business Analytics Warehouse Naming Conventions and Domain Values*.

---

You can categorize your Oracle General Ledger accounts into specific group account numbers. The group account number is used during data extraction as well as

front-end reporting. The GROUP\_ACCT\_NUM field in the GL account dimension table W\_GL\_ACCOUNT\_D denotes the nature of the General Ledger accounts (for example, cash account, payroll account). Refer to the master\_code column in the file\_group\_acct\_names\_ora.csv file for values you can use. For a list of the Group Account Number domain values, see *Oracle Business Analytics Warehouse Naming Conventions and Domain Values*. The mappings to General Ledger Accounts Numbers are important for both Profitability analysis and General Ledger analysis (for example, Balance Sheets).

The logic for assigning the accounts is located in the file\_group\_acct\_codes\_ora.csv file. The table below shows an example configuration of the file\_group\_acct\_codes\_ora.csv file.

**Table 10-2 Example configuration of file\_group\_acct\_codes\_ora.csv**

COA ID	FROM ACCT	TO ACCT	GROUP_ACCT_NUM
1	101010	101099	CA
1	131010	131939	FG INV
1	152121	152401	RM INV
1	171101	171901	WIP INV
1	173001	173001	PPE
1	240100	240120	ACC DEPCN
1	261000	261100	INT EXP
1	181011	181918	CASH
1	251100	251120	ST BORR

In the table above, in the first row, all accounts within the account number range from 101010 to 101099 that have a Chart of Account (COA) ID equal to 1 are assigned to Current Asset. Each row maps all accounts within the specified account number range and within the given chart of account ID.

If you need to create a new group of account numbers, you can create new rows in the file\_group\_acct\_names\_EBS.csv file. You can then assign GL accounts to the new group of account numbers in the file\_group\_acct\_codes\_ora11i.csv file.

You also need to add a new row in the file\_grpact\_fstmt.csv file. This file specifies the relationship between a Group Account Number and a Financial Statement Item Code. You must map the new Group Account Number to one of the following Financial Statement Item codes:

- AP
- AR
- COGS
- REVENUE
- TAX
- OTHERS

These Financial Statement Item codes correspond to the following six base fact tables in the Financial Analytics product.

- AP base fact (W\_AP\_XACT\_F)
- AR base fact (W\_AR\_XACT\_F)

- Revenue base fact (W\_GL\_REVN\_F)
- Cost of Goods Sold base fact (W\_GL\_COGS\_F)
- Tax base fact (W\_TAX\_XACT\_F)
- GL Journal base fact (W\_GL\_OTHER\_F)

By mapping your GL accounts against the Group Account Numbers and then associating the Group Account Number to a Financial Statement Item code, you have indirectly associated the GL account numbers to Financial Statement Item codes as well.

Financial Statement Item codes are internal codes used by ETL process to process the GL journal records during the GL reconciliation process against the subledgers. When the ETL process reconciles a GL journal record, it looks at the Financial Statement Item code associated with the GL account that the journal is charging against, and then uses the value of the Financial Statement item code to decide which base fact the GL journal should reconcile against. For example, when processing a GL journal that charges to a GL account which is associate to 'AP' Financial Statement Item code, then the ETL process will try to go against AP base fact table (W\_AP\_XACT\_F), and try to locate the corresponding matching AP accounting entry. If that GL account is associated with the 'REVENUE' Financial Statement Item code, then the ETL program will try to go against the Revenue base fact table (W\_GL\_REVN\_F), and try to locate the corresponding matching Revenue accounting entry.

---

**Note:** When you specify the Group Account Number, you must capitalize the letters and use the values in the `master_code` column of the `file_group_acct_names_ora.csv` file.

---

### To map Oracle General Ledger account numbers to Group Account Numbers

1. Open the `file_group_acct_names_ora.csv` file with a text editor in the `$pmserver\srcfiles` folder.
2. Edit the fields in the following table:

Field Name	Description
COA ID	The ID of the General Ledger chart of account.
FROM ACCT and TO ACCT	The natural account range. This is based on the natural account segment of your GL accounts.
GROUP_ACCT_NUM	This field denotes the nature of the Oracle General Ledger accounts. For example, Cash account, Payroll account, and so on. Refer to the <code>file_group_acct_names_ora.csv</code> file for values you can use.

---

**Note:** It is important that you do not edit any other fields in the CSV files.

---

3. Save and close the CSV file.

### 10.2.2.3 How to do Group Account correction for Oracle Application

**Note:** Refer to the section [Section 10.2.2.2, "How to Map Oracle General Ledger Natural Accounts to Group Account Numbers"](#) for general concepts about Group Account Number and Financial Statement Item Code.

When a user mistakenly maps a GL natural account number to an incorrect Group Account Number, incorrect accounting entries might be inserted into the fact table. For example, natural account 1210 is classified as belonging to 'AR' Group Account Number when it should be classified as having 'AP' Group Account Number. When this happens, the ETL program will get all the journal lines charging to account 100 and try to reconcile these journal lines against subledger accounting records in the AR fact table (W\_AR\_XACT\_F). Since these journal lines did not come from AR, the ETL program will not be able to find the corresponding matching subledger accounting records for these journal lines. In this case, the ETL program will insert 'Manual' records into the AR fact table because it thinks that these GL journal lines are 'Manual' journal entries created directly in GL system charging against the AR accounts. This entire process is call GL Reconciliation process.

In order to revert these 'Manual' entries in the AR fact, you will need to utilize the 'Group Account Number Cleanup' program provided in Oracle BI Applications. This program will revert the 'Manual' entries in the fact table (in this case, AR fact table); and then try to do the GL reconciliation process again. This time, the ETL program will try to look for the corresponding matching subledger accounting records in the AP fact (W\_AP\_XACT\_F); provided that you've re-assign the natural account 1210 to the 'AP' group account number in the file\_group\_acct\_codes\_ora.csv file.

To do Group Account correction:

1. Correct the mapping of GL Natural account to the group account in the input 'csv' file "file\_group\_acct\_codes\_ora.csv".

For example, before correction, a CSV file has the following values:

```
CHART OF ACCOUNTS ID = 101
FROM ACCT = 1110
TO ACCT = 1110
GROUP_ACCT_NUM = CASH
```

After correction, if the account '1210' originally belonged to the 'AP' Group Acct Num and after correcting the GL Natural account to the group account, the CSV file would have the following values:

```
CHART OF ACCOUNTS ID = 101
FROM ACCT = 1210
TO ACCT = 1210
GROUP_ACCT_UM = AR
```

2. In the DAC, do the following:
  - a. Navigate to the appropriate container.
  - b. Go to 'Design' and open up the 'Subject Areas' tab.
  - c. Open up the 'Financials – General Ledger' subject area and verify which of the following configuration tag is marked as 'inactive':
    - Financials – Calculate GL Balance
    - Oracle – Extract GL Balance

Out of the box, 'Financials – Calculate GL Balance' should be marked as inactive.

- d. Open up the 'Financials – Group Account Number Cleanup' subject area and make sure the followings is configured:
  - If in step 3, 'Financials – Calculate GL Balance' is marked as 'inactive', then the configuration tag 'Financials – Calculate GL Balance Reverse' should be marked as 'inactive' as well.
  - If in step 3, 'Oracle – Extract GL Balance' is marked as 'inactive', then the configuration tag 'Financials – Calculate GL Balance Reverse' should be unmarked in the 'inactive' check box.
3. If you need to make any change in step b above, then you need to reassemble the 'Financials – Group Account Number Clean Up' subject area and then redesign the one of these execution plans, depending on which version of Oracle EBS you have:
  - Financials – Group Account Number Clean Up ORA1158
  - Financials – Group Account Number Clean Up ORA1159
  - Financials – Group Account Number Clean Up ORA11510
  - Financials – Group Account Number Clean Up R12
4. Depending on the Oracle EBS version you have, run the corresponding Group Account Cleanup execution plan:
  - Financials – Group Account Number Clean Up ORA1158
  - Financials – Group Account Number Clean Up ORA1159
  - Financials – Group Account Number Clean Up ORA11510
  - Financials – Group Account Number Clean Up R12

#### 10.2.2.4 How to configure domainValues\_Xact\_Types\_DocTypes\_ora12.csv for Oracle General Ledger and Profitability Analytics

This section explains how to configure domainValues\_Xact\_Types\_DocTypes\_ora12.csv for Oracle General Ledger and Profitability Analytics, Release 12.

To configure domainValues\_Xact\_Types\_DocTypes\_psft.csv for PeopleSoft Receivables:

1. Identify the Entry Types in your Oracle Inventory Application by using the following SQL:

```
select mtt.transaction_type_id, mtt.description from mtl_transaction_types mtt;
```

This query gives the transaction type codes in your Oracle Inventory Application and their corresponding descriptions.

2. Open the domainValues\_Xact\_Types\_DocTypes\_ora12.csv file using a text editor in the \$pmsserver\l\kfiles folder.

Look for only the rows in the file which has xact\_cat\_code = 'COGS' or xact\_cat\_code = 'DEFERRED\_COGS'. From among these records, look for those with W\_XACT\_TYPE\_CODE = 'DELIVERY' or 'RETURN'. The column XACT\_TYPE\_CODE will contain the entry type codes from Oracle Inventory application, and the column W\_XACT\_TYPE\_CODE is the corresponding domain value to which this entry type code will be mapped to.

3. Map the entry type codes returned by the query above from your Oracle Inventory Application to the domain values.

Make sure the entry type codes go into the XACT\_TYPE\_CODE column in the file. Note that it is not necessarily a one-to-one mapping between the domain values and the entry types. For example, you can have multiple entry types mapped into the DELIVERY/RETURN domain value. Be careful not to map the same entry type to two different domain values.

4. Save and close the file.
5. In Informatica Designer, open the mapping SDE\_ORA\_GLCOGSFact in the folder SDE\_ORAxxx\_adapter.
6. Open the mapplet mplt\_BC\_ORA\_GLCOGSFact contained inside this mapping.
7. Open the source qualifier SQ\_MTL\_TRANSACTION\_ACCOUNTS.
8. Modify the sql so that it extracts the additional transaction type ids, from step 1, that need to be extracted.
9. Make the same change in Informatica Workflow Manager to the SQL Override in the full load sessions SDE\_PSFT\_ARTransactionFact\_Item\_Full and SDE\_PSFT\_ARTransactionFact\_ItemActivity\_Full".

### 10.2.2.5 How to configure domainValues\_Xact\_Types\_DocTypes\_ora11i.csv for Oracle General Ledger and Profitability Analytics

This section explains how to configure domainValues\_Xact\_Types\_DocTypes\_ora11i.csv for Oracle General Ledger and Profitability Analytics, Release 11i.

To configure domainValues\_Xact\_Types\_DocTypes\_ora11i.csv for Oracle General Ledger and Profitability Analytics:

1. Identify the Entry Types in your Oracle Inventory Application by using the following SQL:

```
select mtt.transaction_type_id, mtt.description from mtl_transaction_types mtt;
```

This query gives the transaction type codes in your Oracle Inventory Application and their corresponding descriptions.

2. Open the domainValues\_Xact\_Types\_DocTypes\_ora11i.csv file using a text editor in the \$pmsserver\lkpfiles folder.

Look for only the rows in the file which has xact\_cat\_code = 'COGS'. From among these records, look for those with W\_XACT\_TYPE\_CODE = 'DELIVERY' or 'RETURN'. The column XACT\_TYPE\_CODE will contain the entry type codes from Oracle Inventory application, and the column W\_XACT\_TYPE\_CODE is the corresponding domain value to which this entry type code will be mapped to.

3. Map the entry type codes returned by the query above from your Oracle Inventory Application to the domain values.

Make sure the entry type codes go into the XACT\_TYPE\_CODE column in the file. Note that it is not necessarily a one-to-one mapping between the domain values and the entry types. For example, you can have multiple entry types mapped into the DELIVERY/RETURN domain value. Be careful not to map the same entry type to two different domain values.

4. Save and close the file.

5. In Informatica Designer, open the mapping SDE\_ORA\_GLCOGSFact in the folder SDE\_ORAxxx\_adapter.
6. Open the maplet mplt\_BC\_ORA\_GLCOGSFact contained inside this mapping.
7. Open the source qualifier SQ\_MTL\_TRANSACTION\_ACCOUNTS.
8. Modify the sql so that it extracts the additional transaction type ids, from step 1, that need to be extracted.
9. Make the same change in Informatica Workflow Manager to the full load sessions that correspond to these mappings: SDE\_ORA\_GLCOGSFact\_Full.

## 10.2.3 Configuration Steps for Financial Analytics For PeopleSoft

This section contains configuration steps before you do a full data load that apply to PeopleSoft, and contains the following topics:

- [Section 10.2.3.1, "About Configuring Domain Values and CSV Worksheet Files for Oracle Financial Analytics"](#)
- [Section 10.2.3.2, "How to configure domainValues\\_Xact\\_Types\\_DocTypes\\_psft.csv for PeopleSoft Receivables"](#)
- [Section 10.2.3.3, "How to insert beginning GL Balances into Oracle BI Applications"](#)

### 10.2.3.1 About Configuring Domain Values and CSV Worksheet Files for Oracle Financial Analytics

If you modify or extend a seeded list of values, you must configure the CSV files for Oracle Financial Analytics by mapping values from your source systems to the domain values.

The section [Section 10.2.3.2, "How to configure domainValues\\_Xact\\_Types\\_DocTypes\\_psft.csv for PeopleSoft Receivables"](#) explains how to extract the lists of values from your source system, which you then compare with the seeded values. If the lists of values are different to the seeded values, you need to follow the instructions to configure the Domain Values and CSV Worksheet Files.

[Table 10–3](#) lists the CSV worksheet files and the domain values for Financial Analytics and PeopleSoft in the \$pmsserver\LkpFiles folder.

**Table 10–3 Domain Values and CSV Worksheet Files for Oracle Financial Analytics**

Worksheet File Name	Description	Session
domainValues_Xact_Types_DocTypes_psft.csv	List the Transaction types and the corresponding domain values for the PeopleSoft application.  For more information about updating the values in this file, see <a href="#">Section 10.2.3.2, "How to configure domainValues_Xact_Types_DocTypes_psft.csv for PeopleSoft Receivables"</a> .	SDE_PSFT_TransactionTypeDimension_ARDerive

### 10.2.3.2 How to configure domainValues\_Xact\_Types\_DocTypes\_psft.csv for PeopleSoft Receivables

This section explains how to configure domainValues\_Xact\_Types\_DocTypes\_psft.csv for PeopleSoft Receivables.

To configure domainValues\_Xact\_Types\_DocTypes\_psft.csv for PeopleSoft Receivables:

1. Identify the Entry Types in your PeopleSoft Receivables Application by using the following SQL:

```
SELECT DISTINCT ENTRY_TYPE, DESCR FROM PS_ENTRY_TYPE_TBL
```

This query gives the entry type codes in your PeopleSoft Receivables Application and their corresponding descriptions.

2. Open the domainValues\_Xact\_Types\_DocTypes\_psft.csv file using a text editor in the \$pmserver\lkpfiles folder.

Look for only the rows in the file that have xact\_cat\_code = 'RECEIVABLE' and W\_XACT\_TYPE\_CODE = 'ORIGINAL' or 'APPLICATION'. The column XACT\_SUBTYPE\_CODE will contain the entry type codes from PeopleSoft Receivables and the columns W\_XACT\_TYPE\_CODE and W\_XACT\_SUBTYPE\_CODE are the corresponding domain values these entry type codes will be mapped to.

3. Copy the TYPE column to the XACT\_TYPE\_CODE column in the file.

The data must be copied starting from the 2nd line. The first line is the column header.

4. Map the entry type codes returned by the query above from your PeopleSoft Receivable Application to the domain values.

Make sure the entry type codes go into the XACT\_TYPE\_CODE column in the file. Note that it is not necessarily a one-to-one mapping between the domain values and the entry types. For example, you can have multiple entry types mapped into the ORIGINAL/INVOICE domain value. Be careful not to map the same entry type to two different domain values.

5. Save and close the file.

6. In Informatica Designer, open mapplets mplt\_BC\_PSFT\_ARTransactionFact\_Item and mplt\_BC\_PSFT\_ARTransactionFact\_ItemActivity in the SDE\_PSFTxx\_adapter folder.

7. Open the source qualifier transformation.

At the end of the SQL, there is a "IN" clause that should include all the entry type codes that map into W\_XACT\_TYPE\_CODE = 'ORIGINAL' in the csv file prepared above.

8. Make the same change in Informatica Workflow Manager to the SQL Override in the full load sessions SDE\_PSFT\_ARTransactionFact\_Item\_Full and SDE\_PSFT\_ARTransactionFact\_ItemActivity\_Full".

### 10.2.3.3 How to insert beginning GL Balances into Oracle BI Applications

Oracle BI Application computes GL Balances based on the GL Journals extracted into the data warehouse. It assumes that the beginning balance is zero and calculates the ending GL Balance for a month by adding up all the GL Journals from the beginning of time to that month (for Balance Sheet Accounts) or from the beginning of the year to that month (for Non-Balance Sheet Accounts). But since it assumes the beginning balance is zero, there is a problem if you want to extract Journals from your PeopleSoft system only from a certain year. For example, if you have 10 years worth of Journals in the OLTP system and you extract only last 3 years data into the data warehouse, the GL Balance computed will not take into account the first 7 years' journals, and hence the balance will not be correct.

To address the issue, Oracle BI Applications provides a facility for you to enter a beginning balance for every account. So instead of starting from zero, the GL Balance computation will start from this beginning balance number and start adding up journals to it to calculate ending GL Balances. The facility is through the universal adapter for the GL Journals Staging table, W\_GL\_OTHER\_FS. There is a csv file, file\_gl\_other.csv, in the srcfiles directory of the informatica server. In this file, you enter a beginning balance for each GL Account and that will be automatically loaded into the staging table W\_GL\_OTHER\_FS. Note that the file needs to be ready before the first full load ETL is started.

To insert beginning GL Balances

1. Open the file\_gl\_other.csv file using a text editor in the \$pmserver\lkpfiles folder.
2. Populate the following columns:
  - GL\_ACCOUNT\_ID - FK to GL Account dimension
  - DOC\_STATUS\_ID - FK to Status dimension. Set this to "ACCT\_DOC\_STATUS~~POSTED" for all rows.
  - POSTED\_ON\_DT - Posted Date in YYYYMMDDHH24MISS format. Set this to the date for which you are entering beginning balances. For example, you are decide to extract journals from Jan 01 of 2005, set this date to 20050101000000.
  - ACCT\_PERIOD\_END\_DT - Accounting Period Date in YYYYMMDDHH24MISS format. Set this to the date for which you are entering beginning balances. For example, you are decide to extract journals from Jan 01 of 2005, set this date to 20050101000000.
  - OTHER\_DOC\_AMT - Amount in document currency.
  - OTHER\_LOC\_AMT - Amount in local/Business Unit currency
  - GLOBAL1\_EXCHANGE\_RATE - Exchange rate for Doc Currency to Global1 Currency Code
  - GLOBAL2\_EXCHANGE\_RATE - Exchange rate for Doc Currency to Global2 Currency Code
  - GLOBAL3\_EXCHANGE\_RATE - Exchange rate for Doc Currency to Global3 Currency Code.
  - BALANCE\_ID - Give a unique value for this row
3. Save and close the file.
4. In Informatica Designer, open mapplets mplt\_BC\_PSFT\_ARTransactionFact\_Item and mplt\_BC\_PSFT\_ARTransactionFact\_ItemActivity in the SDE\_PSFTxx\_adapter folder.
5. Open the source qualifier transformation.
 

At the end of the SQL, there is a "IN" clause that should include all the entry type codes that map into W\_XACT\_TYPE\_CODE = 'ORIGINAL' in the csv file prepared above.
6. Make the same change in Informatica Workflow Manager to the SQL Override in the full load sessions SDE\_PSFT\_ARTransactionFact\_Item\_Full and SDE\_PSFT\_ARTransactionFact\_ItemActivity\_Full".

## 10.2.4 Configuration Steps for Financial Analytics For Universal

This section contains configuration steps before you do a full data load that apply to Universal.

Not applicable to Oracle BI Applications Version 7.9.4.

## 10.2.5 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle Financial Analytics, and contains the following topics:

- [Section 10.2.5.1, "Configuration Steps for Financial Analytics For All Source Systems"](#)
- [Section 10.2.5.2, "Configuration Steps for Financial Analytics For Oracle EBS"](#)
- [Section 10.2.5.3, "Configuration Steps for Financial Analytics For PeopleSoft"](#)
- [Section 10.2.5.4, "Configuration Steps for Financial Analytics Chain For Universal"](#)

### 10.2.5.1 Configuration Steps for Financial Analytics For All Source Systems

This section contains additional configuration steps that apply to all source systems, and contains the following topics:

- [Section 10.2.5.1.1, "How to Configure the AP/AR Aging Tables"](#)

**10.2.5.1.1 How to Configure the AP/AR Aging Tables** This section explains how to control the lengths of the aging buckets in the AP and AR aging snapshot tables. These tables are:

- W\_AP\_AGING\_INVOICE\_A
- W\_AR\_AGING\_INVOICE\_A
- W\_AP\_AGING\_SUPPLIER\_A
- W\_AR\_AGING\_CUSTOMER\_A

In these four tables, outstanding AP/AR balance information is broken out into rows. Each row represents the outstanding balance information for a particular aging bucket. Four aging buckets are provided out-of-the-box, with the following durations:

- Bucket 1: 0 – 30 days
- Bucket 2: 31 – 60 days
- Bucket 3: 61 – 90 days
- Bucket 4: 90+ days

Use the DAC to configure the lengths of these aging buckets, as described below.

#### To configure the length of aging buckets

1. Click on the 'Tasks' tab.
2. Query for the following tasks
  - PLP\_APSnapshotInvoiceAging
  - PLP\_ARSnapshotInvoiceAging
3. For each of these tasks, click on the 'Parameters' tab and update the parameters with names starting with '\$\$BUCKET'.
4. Save your work.

These aging tables are snapshot tables. You can also control how many historic month end snapshots you want to keep in these tables (the default value is 24 months). You can increase or decrease the number by adjusting the `$$HISTORY_MONTHS` parameter.

### 10.2.5.2 Configuration Steps for Financial Analytics For Oracle EBS

This section contains additional configuration steps that apply to Oracle EBS, and contains the following topics:

- [Section 10.2.5.2.1, "How to Configure How GL Balances Is Populated in Oracle EBS"](#)
- [Section 10.2.5.2.2, "How to Configure Oracle Profitability Analytics Transaction Extracts"](#)
- [Section 10.2.5.2.3, "How to Configure Cost Of Goods Extract \(Oracle EBS 11i-specific\)"](#)
- [Section 10.2.5.2.4, "How to Configure the General Ledger Account Hierarchies"](#)
- [Section 10.2.5.2.5, "Configuring General Ledger Account Hierarchies Using General Ledger Accounting Flexfield value sets definitions"](#)
- [Section 10.2.5.2.6, "How to Configure General Ledger Account Hierarchies Using Financial Statement Generator \(FSG\) Report Definition"](#)
- [Section 10.2.5.2.7, "How to Configure AP Balance ID for Oracle Payables Analytics"](#)
- [Section 10.2.5.2.8, "How to Configure AR Balance ID for Oracle Receivables Analytics and Oracle General Ledger and Profitability Analytics"](#)
- [Section 10.2.5.2.9, "How to Configure the AR Adjustments Extract for Oracle Receivables Analytics"](#)
- [Section 10.2.5.2.10, "How to Configure the AR Schedules Extract"](#)
- [Section 10.2.5.2.11, "How to Configure the AR Cash Receipt Application Extract for Oracle Receivables Analytics"](#)
- [Section 10.2.5.2.12, "How to Configure the AR Credit-Memo Application Extract for Oracle Receivables Analytics"](#)
- [Section 10.2.5.2.13, "How to Specify the Ledger or Set of Books for which GL Data is Extracted"](#)
- [Section 10.2.5.2.14, "How to Setup Drill Down in Oracle BI Answers from General Ledger to Subledger"](#)

**10.2.5.2.1 How to Configure How GL Balances Is Populated in Oracle EBS** Oracle BI Applications provides two ways to populate the GL balances (stored in the `W_GL_BALANCE_F` table), as follows:

1. By extracting the GL balances directly from Oracle General Ledger, as follows:
  - a. In DAC, for the Subject Area 'Financials – General Ledger', in the 'Configuration Tag' tab, make sure that the tag 'Oracle – Extract GL Balance' is unchecked for the 'Inactive' checkbox.
  - b. Make sure the tag 'Financials – Calculate GL Balance' is checked in the 'Inactive' checkbox.
  - c. Click Assemble to redesign the subject area.

- d. After the subject area has been redesigned, redesign the execution plans that contain this subject area.
2. By calculating the GL balances based on the records in the W\_GL\_OTHER\_F table, which stores all journal lines, as follows:
  - a. In DAC, for the Subject Area 'Financials – General Ledger', in the 'Configuration Tag' tab, make sure that the tag 'Financials – Calculate GL Balance' is unchecked for the 'Inactive' checkbox.
  - b. Make sure the tag 'Oracle – Extract GL Balance' is checked in the 'Inactive' checkbox.
  - c. Click Assemble to redesign the subject area.
  - d. After the subject area has been redesigned, redesign the execution plans that contain this subject area.

**10.2.5.2.2 How to Configure Oracle Profitability Analytics Transaction Extracts** There are two separate transaction extracts for Oracle General Ledger and Profitability Analytics—General Ledger Revenue and General Ledger COGS. By default, the Oracle General Ledger Analytics application extracts only Completed revenue and COGS that have been posted to the general ledger. Completed revenue transactions are those where the RA\_CUSTOMER\_TRX\_ALL.COMPLETE\_FLAG = Y. If you want to extract incomplete revenue transactions, you can remove the filter in the Business Component.

**To modify the extract filter for Oracle General Ledger and Profitability Analytics Revenue**

1. In PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE\_ORA11510\_adapter).
2. In Maplet Designer, open the mp1t\_BC\_ORA\_GLRevenueFact maplet.
3. Double-click the Source Qualifier to open the Edit Transformations window, and click the Properties tab.

In the User Defined Join field and in the SQL Query field, remove the statement:

```
AND RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG = Y
```

4. Validate and save your changes to the repository.

**10.2.5.2.3 How to Configure Cost Of Goods Extract (Oracle EBS 11i-specific)** For customers of EBS 11i, 11.5.8, 11.5.9, and 11.5.10, the Oracle General Ledger Analytics application extracts only COGS transactions that have been posted to the general ledger, by default. All COGS transactions that have been transferred satisfy the following condition— MTL\_TRANSACTION\_ACCOUNTS.GL\_BATCH\_ID <> -1. If you want to extract all transactions, you can remove the filter in the Business Component maplet. You need to modify the Business Component maplet for both the regular extract maplet as well as the primary extract maplet. This does not apply to EBS R12; by default all transactions are extracted. The following section applies only to EBS 11i customers.

**To modify the extract filter for General Ledger COGS**

1. In PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE\_ORA11510\_adapter).
2. In Maplet Designer, open mp1t\_BC\_ORA\_GLCOGSFact.

3. Double-click the Source Qualifier to open the Edit Transformations window, and click the Properties tab.

In the User Defined Join field and in the SQL Query field, remove the statement:

```
AND MTL_TRANSACTION_ACCOUNTS.GL_BATCH_ID <> -1
```

4. In Mapplet Designer, open `mp1t_BC_ORA_GLCOGSFact_PRIMARY`.
5. Double-click the Source Qualifier to open the Edit Transformations window, and click the Properties tab.

In the User Defined Join field and in the SQL Query field, remove the statement:

```
AND MTL_TRANSACTION_ACCOUNTS.GL_BATCH_ID <> -1
```

6. Validate and save your changes to the repository.
7. Make appropriate changes to the sql in the informatica session, "SDE\_ORA\_GLCOGSFact\_Full", which is present in the corresponding folder in Workflow Manager.
8. If you customized the mapping SDE\_ORA\_GLCOGSFact to extract from Oracle EBS versions 11.5.8, 11.5.9, or 11.5.10, modify the domain values file `domainValues_Xact_Types_DocTypes_ora11i.csv`.
9. To extract from Oracle EBS version R12, modify the file `domainValues_Xact_Types_DocTypes_ora12.csv`.

**10.2.5.2.4 How to Configure the General Ledger Account Hierarchies** There are two ways to set up hierarchies in Oracle Financial Analytics:

- Using General Ledger Accounting Flexfield Value Sets Definitions
- Using the Financial Statement Generator (FSG) Report Definition

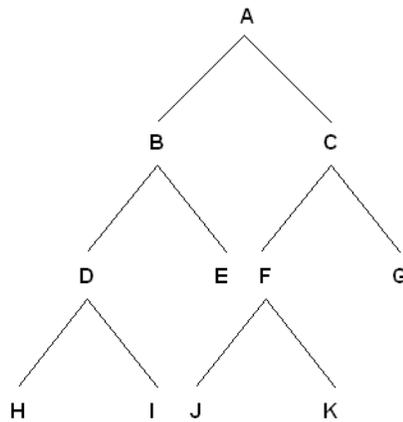
Whichever method you choose to set up General Ledger Account hierarchies, you store the hierarchy information in the `W_HIERARCHY_D` table.

As an example, the hierarchy for US Acct might have the following structure:

- Node A has child nodes B and C.
- Node B has child nodes D and E.
- Node C has child nodes F and G.
- Node D has child nodes H and I.
- Node F has child nodes J and K.

The figure below shows the example hierarchy for US Acct.

**Figure 10–1 Example hierarchy US Acct**



The table below shows how the hierarchy US Acct would be stored in the W\_HIERARCHY\_D table as follows:

**Table 10–4 Example hierarchy US Acct stored in W\_HIERARCHY\_D**

HIER_KEY	HIER_NAME	HIER1_CODE	HIER2_CODE	HIER3_CODE	HIER4_CODE	HIER5_CODE	6 - 19	HIER20_CODE
1	US Acct	A	B	D	H	H	H	H
2	US Acct	A	B	D	I	I	I	I
3	US Acct	A	B	E	E	E	E	E
4	US Acct	A	C	F	J	J	J	J
5	US Acct	A	C	F	K	K	K	K
6	US Acct	A	C	G	G	G	G	G

**10.2.5.2.5 Configuring General Ledger Account Hierarchies Using General Ledger Accounting Flexfield value sets definitions** Oracle EBS supports up to 30 segments in which to store accounting flex fields. Flex fields are flexible enough to support complex data configurations, for example:

- You can store data in any segments.
- You can use more or fewer segments per chart of account, as required.
- You can specify multiple segments for the same chart of account.

**An Example Data Configuration for a Chart of Accounts**

A single company might have a US Chart of Account and an APAC Chart of Account with the following data configuration:

**Table 10–5 Example Chart of Accounts**

Segment Type	US Chart of Account (4256) value	APAC Chart of Account (4257) value
Company	Stores in segment 3	Stores in segment 1
Natural Account	Stores in segment 4	Stores in segment 3
Cost Center	Stores in segment 5	Stores in segment 2
Geography	Stores in segment 2	Stores in segment 2

**Table 10–5 (Cont.) Example Chart of Accounts**

Segment Type	US Chart of Account (4256) value	APAC Chart of Account (4257) value
Line of Business (LOB)	Stores in segment 1	Stores in segment 4

The example shows that in Chart Of Account 4256, Company is stored in the segment3 column in the Oracle EBS table GL\_CODE\_COMBINATIONS\_ALL. In Chart Of Account COA4257, Company is stored in segment1 column in GL\_CODE\_COMBINATIONS\_ALL table. The objective of this configuration file is to make sure that when segment information is extracted into the warehouse table W\_GL\_ACCOUNT\_D, segments with the same nature from different chart of accounts are stored in the same column in W\_GL\_ACCOUNT\_D.

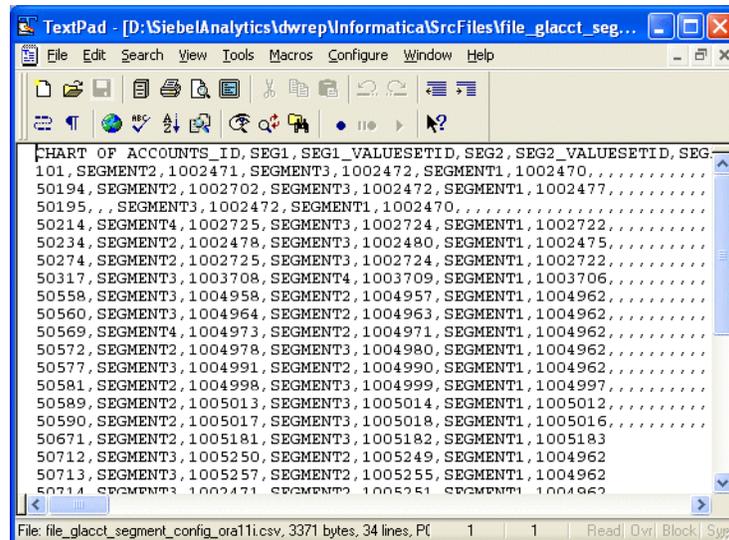
For example, we can store Company segments from COA 4256 and 4257 in segment1 column in W\_GL\_ACCOUNT\_D; and Cost Center segments from COA 4256 and 4257 in segment2 column in W\_GL\_ACCOUNT\_D, and so forth.

**About Configuring the ETL Process for GL Accounting Flex fields**

Before you run the ETL process for General Ledger Accounts, you need to specify the segments that you want to analyze. To specify the segments that you want to analyze, use the following ETL configuration file:

\$PMSEVER/srcfile/file\_glacct\_segment\_config\_ora.csv

**Figure 10–2 Screen Shot of file\_glacct\_segment\_config\_ora11i.csv File Opened in a Text Editor**



In the file\_glacct\_segment\_config\_ora11i.csv file, you need to specify the segments of the same type in the same column. For example, you might store all Cost Center segments from all chart of accounts in one column, and all Company segments from all chart of accounts in another column.

For example, you might want to do the following:

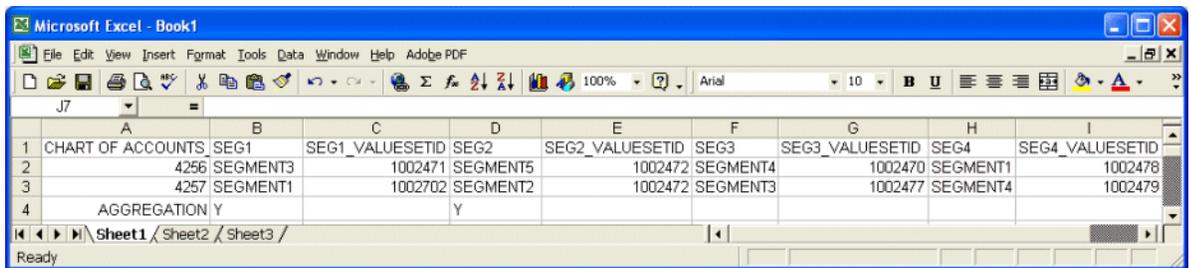
- Analyze GL account hierarchies using only Company, Cost Center, Natural Account, and LOB.

You are not interested in using Geography for hierarchy analysis.

- Store all Cost Center segments from all COAs in ACCOUNT\_SEG2\_CODE column in W\_GL\_ACCOUNT\_D.
- Store all Natural Account segments from all COAs in ACCOUNT\_SEG3\_CODE column in W\_GL\_ACCOUNT\_D.
- Store all LOB segments from all COAs in ACCOUNT\_SEG4\_CODE column in W\_GL\_ACCOUNT\_D.
- In W\_GL\_BALANCE\_A (where you store GL account balances at aggregated level), you want to store GL account balances at Company and Cost Center level instead of at GL Code Combination level.

The screenshot below shows how the file\_glacct\_segment\_config\_ora11i.csv would be configured to implement the business requirements specified above.

**Figure 10–3 Screenshot of file\_glacct\_segment\_config\_ora11i.csv in text editor**



**To Set Up Hierarchies With General Ledger Accounting Flex fields**

1. Use the /srcfile/file\_glacct\_segment\_configur\_ora.csv file to specify the segments that you want to analyze.
2. In the DAC, do the following:
  - a. For the Subject Area 'Financials – General Ledger', in the 'Configuration Tag' tab, make sure that:
    - \* The tag 'Oracle – Extract Value Set Hierarchies' is unchecked for the 'Inactive' checkbox.
    - \* The tag 'Oracle – Extract FSG Hierarchies' is checked in the 'Inactive' checkbox.
  - b. Click Assemble to redesign the subject area.
3. Run the ETL process for General Ledger Accounts.
4. In the RPD Physical Layer (using the Oracle BI Administration Tool), create additional aliases or change the names of the existing alias against the table W\_HIERARCHY\_D.

After the subject area has been redesigned, also redesign the execution plans which contain this subject area.

For example, if the existing aliases are numbered 1 to 3, name the new alias Dim\_W\_HIERARCHY\_D\_ValueSetHierarchy4, and rename the existing aliases as described in the table below:

Old alias name	New alias name
Dim_W_HIERARCHY_D_ValueSetHierarchy1	Dim_Company_Hierarchy_D

Old alias name	New alias name
Dim_W_HIERARCHY_D_ValueSetHierarchy2	Dim_CostCenter_Hierarchy_D
Dim_W_HIERARCHY_D_ValueSetHierarchy3	Dim_NaturalAccount_Hierarchy_D

5. Create a new alias against W\_HIERARCHY\_D and name the new alias Dim\_LOB\_Hierarchy\_D.
6. In the Physical Layer of the Oracle BI Analytics Warehouse, use the Oracle BI Administration Tool to create joins in the physical layer from the new aliases that you created in the previous step, as follows:
  - Company Hierarchy will join to the segment column in W\_GL\_ACCOUNT\_D that stores the Company segment.
    - \* Dim\_Company\_Hierarchy\_D.HIER20\_CODE = W\_GL\_ACCOUNT\_D.ACCOUNT\_SEG1\_CODE
    - \* Dim\_Company\_Hierarchy\_D.HIER\_CODE = W\_GL\_ACCOUNT\_D.ACCOUNT\_SEG1\_ATTRIB
  - Cost Center Hierarchy will join to the segment column in W\_GL\_ACCOUNT\_D that stores the Cost Center segment.
    - \* Dim\_CostCenter\_Hierarchy\_D.HIER20\_CODE = W\_GL\_ACCOUNT\_D.ACCOUNT\_SEG2\_CODE
    - \* Dim\_CostCenter\_Hierarchy\_D.HIER\_CODE = W\_GL\_ACCOUNT\_D.ACCOUNT\_SEG2\_ATTRIB
  - Natural Account Hierarchy will join to the segment column in W\_GL\_ACCOUNT\_D, that stores the Natural Account segment.
    - \* Dim\_NaturalAccount\_Hierarchy\_D.HIER20\_CODE = W\_GL\_ACCOUNT\_D.ACCOUNT\_SEG3\_CODE
    - \* Dim\_NaturalAccount\_Hierarchy\_D.HIER\_CODE = W\_GL\_ACCOUNT\_D.ACCOUNT\_SEG3\_ATTRIB
  - LOB Hierarchy will join to the segment column in W\_GL\_ACCOUNT\_D that stores the LOB segment.
    - \* Dim\_LOB\_Hierarchy\_D.HIER20\_CODE = W\_GL\_ACCOUNT\_D.ACCOUNT\_SEG4\_CODE
    - \* Dim\_LOB\_Hierarchy\_D.HIER\_CODE = W\_GL\_ACCOUNT\_D.ACCOUNT\_SEG4\_ATTRIB

---

**Note:** Hierarchies are linked to HIER20\_CODE, because it is leaf node of the hierarchy

---

7. In the Business Model Layer of the Oracle BI Analytics Warehouse, use the Oracle BI Administration Tool to create additional dimensions using the new alias.
 

For more information about example dimensions that are installed with Oracle BI Applications, refer to the following dimensions in the Oracle Business Analytics Warehouse:

  - Dim - GL ValueSetHierarchy1
  - Dim - GL ValueSetHierarchy2

- Dim - GL ValueSetHierarchy3
8. A HIER\_CODE filter should be specified in the Business Model Layer to restrain the output of logical table to be one Hierarchy only. To specify a filter in the Business Model Layer to restrain the output of logical table, do the following:
    - a. Right click logical table Dim\_IncomeStatement\_FSGHierarchy\_D.
    - b. Choose Properties.
    - c. Choose the Source tab.
    - d. Select "Dim\_IncomeStatement\_FSGHierarchy\_D".
    - e. Click Edit.
    - f. Choose the Content tab.
    - g. Insert the following code into the "use WHERE clause..." text box:
 

```
"Oracle Data Warehouse"."Catalog"."dbo"."Dim_W_HIERARCHY_
          D_FSG1"."HIER_CODE" =1001
```
  9. In the Business Model Layer of the Oracle BI Analytics Warehouse, use the Oracle BI Administration Tool to create additional hierarchies using the dimensions that you created in the previous step.
 

For more information about example hierarchies that are installed with Oracle BI Applications, refer to the following hierarchies in the Oracle Business Analytics Warehouse:

    - GL ValueSetHierarchy1
    - GL ValueSetHierarchy2
    - GL ValueSetHierarchy3
  10. In the Presentation Layer of the Oracle BI Analytics Warehouse, use the Oracle BI Administration Tool to drag the new hierarchies into the presentation folder.

---

**Note:** You can rename the hierarchies in the Presentation Layer if required.

---

**10.2.5.2.6 How to Configure General Ledger Account Hierarchies Using Financial Statement Generator (FSG) Report Definition** If you need to define GL account hierarchies based on multiple segments within a chart of accounts, you can use the Financial Statement Generator report definition UI in Oracle EBS to define them.

You should first use the Oracle EBS Financial Statement Generator (FSG) form to define a row set or a column set, then the Oracle BI Application will extract the row set or column set definition and convert them into hierarchies.

Oracle Financial Statement Generator hierarchies are extracted from following EBS source tables:

- RG\_REPORT\_AXIS\_CONTENTS
 

This table defines the relationship between the FSG report axis and GL code combinations. The GL code combinations with segment values within the value range defined for that axis are categorized as children of that axis.
- RG\_REPORT\_AXIS\_SETS

This table stores the information for each of the row set or column set you defined. There is one record in this table for each row or column set you defined. Each row includes an axis set identifier, a row or column set name, and a structure identifier to assign a specific chart of accounts to the row set or column set.

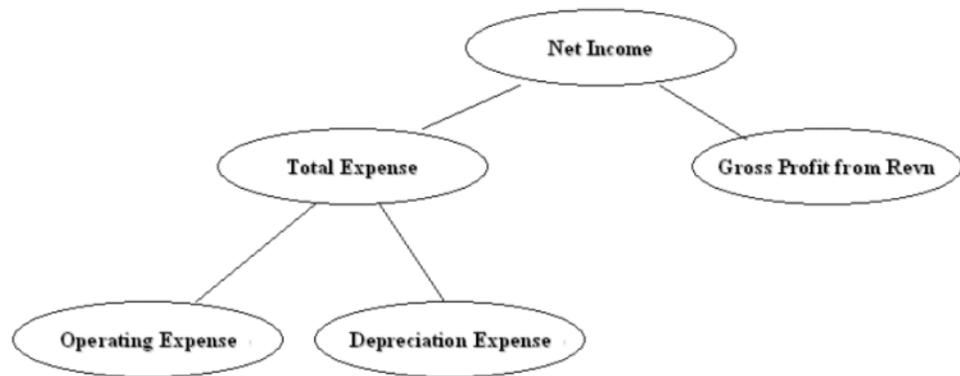
- **RG\_REPORT\_CALCULATIONS**

This table stores formulas for calculating each row or column in the row or column set. An example of a row calculation might be to sum up the amount from the previous five rows. An example of a columns calculation might be to calculate column five by subtracting column four from column three.

For example, in Income Statement, "Net Income" is the calculation result of "Gross Profit from Revenue" minus "Total Expense". When converting to hierarchy, Net Income becomes the parent of "Gross Profit from Revenue" and "Total Expense". Therefore, hierarchy can be derived based on the information in RG\_REPORT\_CALCULATION.

The diagram below shows an example hierarchy, with the top level Net Income node having two child nodes, Total Expense, and Gross Profit from Revn, and the Total Expense node having two child nodes, Operating Expense, and Depreciation Expense.

The diagram below shows how an income state is derived from a hierarchy.



The hierarchy above would be converted into a flattened hierarchy and stored in W\_HIERARCHY\_D in the following format:

**Table 10–6 Example flattened hierarchy stored in W\_HIERARCHY\_D**

HIER Name	HIER1	HIER2	HIER3	HIER4	HIER20
Income Statement	Net Income	Gross Profit...	Gross Profit...	Gross Profit...	Gross Profit...
Income Statement	Net Income	Total Expenses	Operating Expenses	Operating Expenses	Operating Expenses
Income Statement	Net Income	Total Expenses	Depreciation Expense	Depreciation Expense	Depreciation Expense

Fact tables join to the W\_HIERARCHY\_D table via the General Ledger Account dimension table (W\_GL\_ACCOUNT\_D).

The W\_GL\_ACCOUNT\_D table contains six fields (HIER1\_WID, HIER2\_WID, HIER3\_WID, ..., HIER6\_WID), which are foreign keys to the W\_HIERARCHY\_D.row\_wid. Therefore, each General Ledger Code combination can participate in up to six

different hierarchies. You can decide which of the six hierarchies to drill on based on the column you use to join to W\_HIERARCHY\_D.

For example, by using W\_GL\_ACCOUNT\_D.hier3\_wid = W\_HIERARCHY\_D.row\_wid. You are essentially drilling using the third hierarchy.

### About Configuring the ETL Process for Oracle Financial Statement Generator Report

Before you run the ETL process for General Ledger Accounts, you need to specify the hierarchies that you want to reference. To specify the FSG hierarchies that you want to reference, use the following ETL configuration file:

\$PMSERVER/srcfile/file\_gl\_hierarchy\_assignment\_ora.csv

**Figure 10–4** Screen Shot of file\_gl\_hierarchy\_assignment\_ora.csv File Opened in a Text Editor

	A	B	C	D	E	F
	CHART_OF_ACCOUNTS	Hierarchy1_AXIS_SET_ID	Hierarchy2_AXIS_SET_ID	Hierarchy3_AXIS_SET_ID	Hierarchy4_AXIS_SET_ID	Hierarchy5_AXIS_SET_ID
1						
2	101	1003	1922	1903	1924	1000
3	50194	1306		1245	2587	2589
4	50195	1754	1756	2001	2003	
5	50214		1744	1746	1749	1757
6	50234	1758				
7	50274	1758				
8	50317	2330	2331	2424		
9	50558		2107			
10	50560	2647	2648			
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						

In this file, for each chart of accounts, you can specify six FSG hierarchies (using axis\_set\_id; Axis\_set\_id is a column from RG\_REPORT\_AXIS\_SETS table. It is the unique ID of a row set or column set) you want to store in the GL account dimension table for the code combinations that belong to that chart of accounts.

The DATASOURCE\_NUM\_ID field specifies the data source to which the configurations apply. If you have multiple source systems, there might be a chart of accounts across the multiple source systems with the same ID. Therefore, you need to use the DATASOURCE\_NUM\_ID value to distinguish between them.

For example, we have an Income Statement FSG report and a Balance Sheet FSG report which we want to derive both of their hierarchy structures into OLAP.

Oracle BI Applications assumes that both reports are derived from same set of GL accounts with CHART\_OF\_ACCOUNTS=101. The axis\_set\_id of Income Statement is 1001, and for Balance Sheet, it is 1003. The DATASOURCE\_NUM\_ID for this application is 2.

In addition, for those GL ACCOUNTS that belong to the two reports (by which we used to calculate the reports), we want to associate their HIER1 column (in GL\_ACCOUNT\_D) with Income Statement hierarchy structure, HIER3 column with Balance Sheet hierarchy structure.

In this case, we would add one row into file\_gl\_hierarchy\_assignment\_ora.csv with the following fields set as below:

CHART\_OF\_ACCOUNTS - 101

HIER1\_AXIS\_SET\_ID - 1001

HIER3\_AXIS\_SET\_ID - 1003

DATASOURCE\_NUM\_ID - 2

(Leave the other row values blank.)

This row indicates that for all of the GL Accounts with CHART\_OF\_ACCOUNTS=101 and DATASOURCE\_NUM\_ID=2, assigning hierarchies with axis\_set\_id=1001, null, 1003, null, null, null to HIER1~HIER6 columns respectively. Therefore, after extraction and loading, for those affected GL Account rows, HIER1 column will be the foreign key to Income Statement hierarchy's row ID in W\_HIERARCHY\_D and HIER3 column will be the foreign key to Balance Sheet hierarchy's row ID in W\_HIERARCHY\_D.

**Note:** Financial Analytics will not load those hierarchies with axis\_set\_id not been specified in file\_gl\_hierarchy\_assignment\_ora.csv.

To Set Up Hierarchies With Financial Statement Generator Report Definition:

1. Use the /srcfile/ file\_gl\_hierarchy\_assignment\_ora.csv file to specify the hierarchies you want to reference for each CHART\_OF\_ACCOUNTS.
2. In the DAC, do the following:
  - a. Navigate to the Subject Area 'Financials – General Ledger'.
  - b. In the 'Configuration Tag' tab, make sure that the tag 'Oracle – Extract FSG Hierarchies' is unchecked for the 'Inactive' checkbox.
  - c. In the 'Configuration Tag' tab, make sure that the tag 'Oracle – Extract Value Set Hierarchies' is checked in the 'Inactive' checkbox.
  - d. Click Assemble to redesign the subject area.
  - e. After the subject area has been redesigned, also redesign the execution plans which contain this subject area.
3. Run the ETL process for General Ledger Accounts.
4. In the RPD Physical Layer (using the Oracle BI Administration Tool), create additional aliases or change the names of the existing alias against the table W\_HIERARCHY\_D.

For example, if you want to create an income statement hierarchy, create an additional alias "Dim\_IncomeStatement\_FSGHierarchy\_D" against the table W\_HIERARCHY\_D.

5. In the Physical Layer of the Oracle BI Analytics Warehouse, use the Oracle BI Administration Tool to create joins in the physical layer from the new aliases that you created in the previous step, as follows:
  - a. Income Statement Hierarchy will join to one of the HIER1~6 columns that you have specified in file file\_gl\_hierarchy\_assignment\_ora.csv for Income Statement.
  - b. In this case, we join it to HIER1 column. Dim\_W\_GL\_ACCOUNT\_D.HIER1\_WID = Dim\_IncomeStatement\_FSGHierarchy\_D.ROW\_WID

6. In the Business Model Layer of the Oracle BI Analytics Warehouse, use the Oracle BI Administration Tool to create additional dimensions using the new alias.

For the Income Statement hierarchy case, we create a new logical table "Dim\_IncomeStatement\_FSGHierarchy\_D", choose "Dim\_IncomeStatement\_FSGHierarchy\_D" in physical layer as source. Mapping ROW\_WID, HIER\_CODE, and HIER1~HIER20 (both name and code) from physical table to logical key.

Then, set HIER\_CODE=1001 (this is the Axis\_set\_id of Income Statement hierarchy) in logical table to restrain the output of logical table to be Income Statement Hierarchy only (right click logical table "Dim\_IncomeStatement\_FSGHierarchy\_D" – click "properties" – choose "Source" tab – select "Dim\_IncomeStatement\_FSGHierarchy\_D" – click "Edit" button – choose "Content" tab – fill ("Oracle Data Warehouse"."Catalog"."dbo"."Dim\_W\_HIERARCHY\_D\_FSG1"."HIER\_CODE" =1001) into "use WHERE clause..." textbox).

For more information about this process, please refer to the following pre-installed example logical table Dim - FSG Hierarchy 1 in the Oracle Business Analytics Warehouse

7. In the Business Model Layer of the Oracle BI Analytics Warehouse, use the Oracle BI Administration Tool to create new dimension based on the logical table that you created in the previous step.

Please refer "FSG Hierarchy 1" as example.

8. In the Presentation Layer of the Oracle BI Analytics Warehouse, use the Oracle BI Administration Tool to drag the new hierarchies into the presentation folder.

**Note:** You can rename the hierarchies in the Presentation Layer if required.

#### 10.2.5.2.7 How to Configure AP Balance ID for Oracle Payables Analytics

The AP Balance ID controls the level at which the balance in W\_AP\_BALANCE\_F is maintained.

This section contains configuration information for Oracle Payables Analytics that is specific to Oracle.

By default, the Accounts Payable (AP) Balance ID is maintained at the following granularity:

```
SET_OF_BOOKS_ID || '~' || CODE_COMBINATION_ID || '~' || VENDOR_SITE_ID || '~' || INP_ORG_ID
```

However, if you want to maintain your AP balance at a different grain, you can redefine the Balance ID value in the applicable mapplets.

#### To modify the Accounts Payable Balance ID

---



---

**Note:** To modify the Accounts Payable Balance ID, you must modify the following mapplets:

- mplt\_SA\_ORA\_APTransactionFact\_LiabilityDistribution
  - mplt\_SA\_ORA\_APTransactionFact\_ExpenseDistribution
  - mplt\_SA\_ORA\_APTransactionFact\_Payment
  - mplt\_SA\_ORA\_APTransactionFact\_PaymentSchedule
  - mplt\_SA\_ORA\_APTransactionFact\_DiffMan
- 
- 

1. In PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE\_ORA11510\_adapter).

2. In Mapplet Designer, open the mapplet (for example, `mplt_SA_ORA_ARTransactionFact_LiabilityDistribution`).
3. Double-click the Expression transformation to open the Edit Transformations window.
4. Click the Ports tab to edit the Balance ID definition in the `EXT_NU_BALANCE_ID` column.
5. Validate and save your changes to the repository.
6. Repeat steps 1 to 5 for each mapplet that is listed above.

#### 10.2.5.2.8 How to Configure AR Balance ID for Oracle Receivables Analytics and Oracle General Ledger and Profitability Analytics

The AR Balance ID controls the level at which the balance in `W_AR_BALANCE_F` is maintained.

By default, the AR Balance ID is maintained at the following granularity:

```
set_of_books_id || '~' || code_combination_id || '~' || customer_id || '~' ||
customer_site_use_id || '~' transaction_currency_code || '~' || org_id
```

However, if you want to maintain your AR balance at a different grain, you can redefine the Balance ID value in the applicable mapplets.

#### To modify the AR Balance ID

---



---

**Note:** To modify the AR Balance ID, you must modify the following mapplets:

- `mplt_SA_ORA_ARTransactionFact_Adjust`
  - `mplt_SA_ORA_ARTransactionFact_ARScheduleDerive`
  - `mplt_SA_ORA_ARTransactionFact_CreditMemoApplication`
  - `mplt_SA_ORA_ARTransactionFact_ReceivableApplication`
- 
- 

1. In PowerCenter Designer, open the appropriate Oracle Applications folder (for example, `SDE_ORA11510_adapter`).
2. In Mapplet Designer, open the mapplet (for example, `mplt_SA_ORA_ARTransactionFact_Adjust`).
3. Double-click the Expression transformation to open the Edit Transformations window.
4. Click the Ports tab to edit the Balance ID definition in the `EXT_NU_AR_BALANCE_ID` column.
5. Validate and save your changes to the repository.
6. Repeat steps 1 to 5 for each mapplet that is listed above.

**10.2.5.2.9 How to Configure the AR Adjustments Extract for Oracle Receivables Analytics** By default, Oracle Receivables Analytics extracts only approved adjustment entries against accounts receivable transactions. *Approved adjustments* are entries where the `AR_ADJUSTMENTS_ALL.STATUS = A`. If you want to extract additional types of AR adjustment entries, you can remove the filter in the Business Component mapplet. By modifying or removing the filter, you can extract other entries, such as those that require more research, those that are rejected, and those that are not accrued charges.

#### To modify the extract filter for Accounts Receivable adjustments

1. In PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE\_ORA11510\_adapter).
2. In Mapplet Designer, open the `mplt_BC_ORA_ARTransactionFact_Adjust` mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations window, and click the Properties tab.

In the SQL Query field and in the User Defined Join field, modify the statement:

```
AND AR_ADJUSTMENTS_ALL.STATUS = A
```

4. Validate and save your changes to the repository.
5. Repeat Step 2 to Step 4 for the `mplt_BC_ORA_ARTransactionFact_AdjustPrimary` mapplet.

**10.2.5.2.10 How to Configure the AR Schedules Extract** By default, Oracle Receivables Analytics extracts only completed schedules; that is, transactions where the `RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG(+) = Y`. If you want to extract additional types of AR schedule entries, you must remove the filter in the Business Component mapplet. By modifying or removing the filter, you can extract other entries, such as those that were marked as incomplete.

**To modify the extract filter for Accounts Receivable schedules**

1. In PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE\_ORA11510\_adapter).
2. In Mapplet Designer, open the `mplt_BC_ORA_ARTransactionFact_ARSchedules` mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations window.

In the User Defined Join field and in the SQL Query field, modify the statement:

```
AND RA_CUSTOMER_TRX_ALL.COMPLETE_FLAG(+) = Y
```

4. Validate and save your changes to the repository.

**10.2.5.2.11 How to Configure the AR Cash Receipt Application Extract for Oracle Receivables Analytics** By default, Oracle Receivables Analytics extracts only confirmed, cash-receipt application entries against accounts receivable transactions. *Confirmed receipts* are entries where the `AR_RECEIVABLE_APPLICATIONS_ALL.CONFIRMED_FLAG = Y OR NULL`. If you want to extract additional types of cash-receipt application entries, you can remove the filter in the Business Component mapplet. By modifying or removing the filter, you can extract other entries, such as unconfirmed applications.

You must modify both the regular mapplet (`mplt_BC_ORA_ARTransactionFact_ReceivableApplication`) as well as the primary extract mapplet (`mplt_BC_ORA_ARTransactionFact_ReceivableApplicationPrimary`).

**To modify the extract filter for AR cash receipt application**

1. In PowerCenter Designer, open the appropriate Oracle Applications folder (for example, SDE\_ORA11510\_adapter).
2. In Mapplet Designer, open the `mplt_BC_ORA_ARTransactionFact_ReceivableApplication` mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations window, and click the Properties tab.

In the User Defined Join field and in the SQL Query field, modify the statement:

```
AND NVL(AR_RECEIVABLE_APPLICATIONS_ALL.CONFIRMED_FLAG, 'Y') = Y
```

4. Validate and save your changes to the repository.
5. Repeat Step 2 to Step 4 for the `mplt_BC_ORA_ARTransactionFact_ReceiveableApplicationPrimary` mapplet.

#### 10.2.5.2.12 How to Configure the AR Credit-Memo Application Extract for Oracle Receivables

**Analytics** By default, Oracle Receivables Analytics extracts only confirmed, credit-memo application entries against accounts receivable transactions. *Confirmed credit memos* are entries where the `AR_RECEIVABLE_APPLICATIONS_ALL.CONFIRMED_FLAG = Y OR NULL`. If you want to extract additional types of AR credit-memo application entries, you can remove the filter. By modifying or removing the filter, you can extract other entries such as unconfirmed, credit memos.

You must modify both the regular mapplet (`mplt_BC_ORA_ARTransactionFact_CreditmemoApplication`), as well as the primary extract mapplet (`mplt_BC_ORA_ARTransactionFact_CreditmemoApplicationPrimary`). Repeat the following procedure for each mapplet.

#### To modify the extract filter for Accounts Receivable Credit-Memo Application receipts

1. In PowerCenter Designer, open the appropriate Oracle Applications folder (for example, `SDE_ORA11510_adapter`).
2. In Mapplet Designer, open the `mplt_BC_ORA_ARTransactionFact_CreditmemoApplication` mapplet.
3. Double-click the Source Qualifier to open the Edit Transformations window, and click the Properties tab.

In the User Defined Join field and in the SQL Query field, modify the statement:

```
AND NVL(AR_RECEIVABLE_APPLICATIONS_ALL.CONFIRMED_FLAG, 'Y') = Y
```

4. Validate and save your changes to the repository.
5. Repeat Step 2 to Step 4 for the `mplt_BC_ORA_ARTransactionFact_CreditmemoApplicationPrimary` mapplet.

**10.2.5.2.13 How to Specify the Ledger or Set of Books for which GL Data is Extracted** For If you have an Oracle EBS source system, you can specify which set of books or ledgers for which you extract the GL data.

Oracle BI Applications enables you to configure:

- The list of ledgers or set of books to extract the GL data
- The types of ledgers or set of books to extract the GL data for.

You can use either of these configuration points separately and combine them. When installed out-of-the-box, Oracle BI Applications extract all GL data for all ledgers or set of books.

For Oracle 11i customers, to configure the list of set of books to extract the GL data for, do the following:

1. Log into DAC and open the source container that you want to configure.
2. Display the Source System Parameters tab.
3. Use the Search facility to locate the parameter `$$FILTER_BY_SET_OF_BOOKS_ID`.

4. Set the value to 'Y'.  
Make sure that you put single quotes around the Y.
5. Save the changes.
6. Use the Search facility to locate the parameter `$$SET_OF_BOOKS_ID_LIST`.
7. Enter the IDs of the set of books for which you want to extract GL data for.  
Specify the list of set of book IDs separated by commas (do not specify single or double quotes).  
For example, if you want to extract GL data for set of books with IDs: 101, 207, and 303, then set the value of this parameter to the following:

101 , 207 , 303

8. Save the changes.

You can also specify the Set of Books type you want to extract GL data for. In Oracle 11i, there are three set of books types (this is based on the column `GL_SETS_OF_BOOKS.mrc_sob_type_code` column):

- P (Parent)
- R (Reporting)
- N (None)

For Oracle 11i customers, to configure the types of set of books to extract the GL data for, do the following:

1. Log into DAC and open the source container that you want to configure.
2. Display the Source System Parameters tab.
3. Use the Search facility to locate the parameter `$$FILTER_BY_SET_OF_BOOKS_TYPE`.
4. Set the value to 'Y'.  
Make sure that you put single quotes around the Y.
5. Save the changes.
6. Use the Search facility to locate the parameter `$$SET_OF_BOOKS_TYPE_LIST`.
7. Enter the types of set of books for which you want to extract GL data for.  
Specify the list of set of book IDs in single quotes and separated by commas.  
For example, if you want to extract GL data for all Parent set of books and all Reporting set of books, then set the value of this parameter as follows:

'P' , 'R'

8. Save the changes.

For Oracle 12i customers, to configure the list of ledgers to extract the GL data for, do the following:

1. Log into DAC and open the source container that you want to configure.
2. Display the Source System Parameters tab.
3. Use the Search facility to locate the parameter `$$FILTER_BY_LEDGER_ID`.
4. Set the value to 'Y'.

Make sure that you put single quotes around the Y.

5. Save the changes.
6. Use the Search facility to locate the parameter `$$LEDGER_ID_LIST`.
7. Enter the IDs of ledgers for which you want to extract GL data for.  
Specify the list of ledger IDs separated by commas (do not use single quotes).  
For example, For example, if you want to extract GL data for ledgers with IDs: 101, 207, and 303, then set the value of this parameter as follows:  
`101 , 207 , 303`

8. Save the changes.

You can also specify types of ledgers you want to extract GL data for. In Oracle R12, there are four types of ledgers (this is based on the column `GL_LEDGERS.ledger_category_code` column):

- PRIMARY
- SECONDARY
- ALC
- NONE

For R12 customers, to configure the types of ledgers to extract the GL data for, do the following:

1. Log into DAC and open the source container that you want to configure.
2. Display the Source System Parameters tab.
3. Use the Search facility to locate the parameter `$$FILTER_BY_LEDGER_TYPE`.
4. Set the value to 'Y'.

Make sure that you put single quotes around the Y.

5. Save the changes.
6. Use the Search facility to locate the parameter `$$LEDGER_TYPE_LIST`.
7. Enter the types of ledgers for which you want to extract GL data for.  
Specify the list of set of book IDs in single quotes and separated by commas.  
For example, if you want to extract GL data for all Primary ledgers and all Secondary ledgers, then simply set the value of this parameter to as follows:

`'PRIMARY' , 'SECONDARY'`

8. Save the changes.

#### 10.2.5.2.14 How to Setup Drill Down in Oracle BI Answers from General Ledger to Subledger

Oracle BI Applications enables you to trace a GL Journal to the Subledger transaction that created that journal. This ability (or drill down) is achieved through the "Navigation" feature in Oracle BI Answers. This feature is available AP if the source is Oracle EBS 11i and for all subledgers if the source is Oracle EBS R12.

To set up drill down in Oracle BI Answers from General Ledger to Subledger:

1. Create your subledger request from "Financials - AP Transactions" or "Financials - AR Transactions" catalog as applicable.
2. In your request, add a filter on the column "GL Journal ID" under the "Document Details" folder and the set the operator of the filter to "Is Prompted".

3. Build your GL Journal request from the "Financials - GL Detail Transactions" catalog.
4. To your request, add the column "GL Journal ID" under the "Document Details" folder.
5. Navigate to the Column Properties of this column, and set the Value Interaction property in the Column Format tab to "Navigate".
6. Add a navigation target and set the target location to the subledger request you created earlier.

---

**Note:** For COGS and Revenue, the "GL Journal ID" column is not exposed in any presentation catalogs. It is available in the business model layer of the RPD metadata under the logical tables "Dim - GL COGS Details" and "Dim - GL Revenue Details" respectively. As a workaround, you can create presentation catalogs to report of detail level transactions for COGS and Revenue and expose this column in the presentation catalog and use similar steps are above to setup a drilldown from GL to COGS/Revenue.

---

### 10.2.5.3 Configuration Steps for Financial Analytics For PeopleSoft

This section contains additional configuration steps that apply to PeopleSoft.

This section describes how to configure Oracle Business Intelligence Applications to use PeopleSoft versions 8.4 and 8.8 as data sources for Financial Analytics, and contains the following topics:

- [Section 10.2.5.3.1, "About PeopleSoft Trees in Oracle Financial Analytics"](#)
- [Section 10.2.5.3.2, "GL Account Dimension, Chartfields and Hierarchy for PeopleSoft"](#)
- [Section 10.2.5.3.3, "Group Account Configuration"](#)
- [Section 10.2.5.3.5, "General Ledger Chartfield Hierarchy Extraction Customization"](#)
- [Section 10.2.5.3.6, "Internal Organization Hierarchy Extraction Customization"](#)

Financial Analytics consists of three products: GL and Profitability Analytics, Payables Analytics, and Receivables Analytics. Financial Analytics is referred to in the Oracle Business Intelligence Applications as Financial Analytics. References to Oracle BI Applications can be read as Financial Analytics.

**10.2.5.3.1 About PeopleSoft Trees in Oracle Financial Analytics** For PeopleSoft, the Oracle Financial Analytics application sources data from a data structure, called PeopleSoft Trees, to get information about the organization's General Ledger hierarchies, Profit hierarchies, Cost Centers' hierarchies, and so on.

PeopleSoft Trees are a flexible, generic way of constructing hierarchical summarization of a particular database fields in PeopleSoft for reporting purposes. Typically, entities such as Chart of Account fields (Account, Dept, and Project, and so on), items, locations, and so on, are organized into user-defined trees.

[Table 10–7](#) lists the PeopleSoft Trees the Oracle Financial Analytics application sources.

**Table 10–7 PeopleSoft Trees for Oracle's Siebel Financial Analytics**

PeopleSoft Data Structure	Repository Source Name	Description
PeopleSoft Tree	ACCTROLLUP	Account Rollup
PeopleSoft Tree	AR_BUSINESS_UNITS	AR Business Unit Roll-Up
PeopleSoft Tree	DEPARTMENTS	Manufacturing Departments
PeopleSoft Tree	OPERUNIT	Operating Unit

Table 10–8 lists the mappings and sessions using PeopleSoft Trees.

**Table 10–8 List of Mappings And Sessions Using PeopleSoft Trees**

Repository Source Name	Mapping	Session
ACCTROLLUP	M_P_GL_ACCTS_ HIERARCHY_EXTRACT	S_M_P_GL_ACCTS_ HIERARCHY_EXTRACT
AR_BUSINESS_UNITS	M_P_BUSN_ORGS_ COMPANY_HIERARCHY_ EXTRACT	S_M_P_BUSN_ORGS_ COMPANY_HIERARCHY_ EXTRACT
DEPARTMENTS	M_P_PROFIT_CNTRS_ HIERARCHY_EXTRACT	S_M_P_PROFIT_CNTRS_ HIERARCHY_EXTRACT
OPERUNIT	M_P_COST_CENTERS_ HIERARCHY_EXTRACT	S_M_P_COST_CENTERS_ HIERARCHY_EXTRACT

**10.2.5.3.2 GL Account Dimension, Chartfields and Hierarchy for PeopleSoft** The GL Account Dimension in the Oracle Business Analytics Warehouse is at a granularity of a combination of chartfields. PeopleSoft Financials provides several chartfields for GL Accounts, such as Account, Alternate Account, Operating Unit, Department etc. The ETL program extracts all possible combinations of these chartfields that you have used and stores each of these chartfields individually in the GL Account Dimension. It extracts the combinations of chartfields used from the following PeopleSoft account entry tables:

- PS\_VCHR\_ACCTG\_LINES (Accounts Payable)
- PS\_ITEM\_DST (Accounts Receivable)
- PS\_BI\_ACCT\_ENTRY (Billings)
- PS\_CM\_ACCTG\_LINE (Costing)
- PS\_JRNL\_LN (General Ledger)

The GL Account Dimension (W\_GL\_ACCOUNT\_D) in the Oracle Business Analytics Warehouse provides a flexible and generic data model to accommodate up to 30 chartfields. These are stored in the generic columns named ACCOUNT\_SEG1\_CODE, ACCOUNT\_SEG2\_CODE and so on up to ACCOUNT\_SEG30\_CODE, henceforth referred to as segments. These columns store the actual chartfield value that is used in your PeopleSoft application.

#### Mapping PeopleSoft Chartfields

A CSV file has been provided to map the PeopleSoft Chartfields to the generic segments. Use this file to specify which PeopleSoft application chartfield should be populated in which segment. The file is called file\_glacct\_segment\_config\_psft.csv and is located in the <OracleBI>\dwwrep\Informatica\SrcFiles folder.

The first row in the file is a header row; do not modify this line. The second row in the file is where you specify how to do the mapping. The value for the column ROW\_ID is hard coded to "1"; there is no need to change this.

Note that the file contains 30 columns – SEG1, SEG2, up to SEG30. You will have to specify which chartfield needs to be populated in each of these columns by specifying one of the supported values for the chartfields. The chartfields currently supported for the PeopleSoft application are listed below.

---



---

**Note:** Values are case sensitive. You must specify the values exactly as shown below.

---



---

- Account
- Alternate Account
- Operating Unit
- Fund Code
- Department
- Program Code
- Class Field
- Budget Reference
- Product
- Project
- Affiliate
- Fund Affiliate
- Operating Unit Affiliate
- ChartField 1
- ChartField 2
- ChartField 3

---



---

**Note:** You only need to include the chartfields in the CSV file that you want to map.

---



---

### Hierarchy

Oracle Business Intelligence Application supports hierarchy for all these segments. If you have created a "tree" in PeopleSoft for any of these chartfields, you can extract these trees into Oracle Business Analytics Warehouse's hierarchy to analyze your facts at any levels of the hierarchy. Information is provided in the sections that follow on how to extract these trees and how to setup the repository file (RPD) to use the Oracle Business Analytics Warehouse's hierarchy.

### GL Balance Aggregate

The Oracle Business Analytics Warehouse data model has a fact table (W\_GL\_BALANCE\_F) that stores GL Balance for all your GL Accounts. To compress this fact table for better performance, it also provides an aggregate table built on top of this fact table that stores GL Balances for up to 6 segments of your choice. The number of

segments you want in the aggregate table and which ones, are totally configurable. This configuration is done at the third line of the file\_glacct\_segment\_config\_psft.csv file. Specify a value of "Y" under the segment column that you want to be included in the aggregate table.

---

**Note:** You can have only up to 6 "Y"s in the file. Also, it is not a requirement that you have to use all 6. If, for example you want only 3 segments in your aggregate, you need only to specify 3 'Y's.

---

### Example CSV File Configuration

As an example for the configuration of the file\_glacct\_segment\_config\_psft.csv file, consider the following scenario:

Your system uses 4 chartfields, Account, Alt Account, Operating Unit, and Dept. Assume that you are interested in analyzing your data by only 3 of the 4 chartfields: Account, Operating Unit, Department, and that you typically view your GL Balances at the combination of Account and Department. Only occasionally do you want to view your GL balances as the combination of all 3 chartfields. In this scenario, the CSV file should look like:

**Table 10–9 Example CSV Chartfield Mapping Values**

ROW_ID	SEG1	SEG2	SEG3
1	Account	Operating Unit	Department
AGGREGATION	Y		Y

With this configuration, W\_GL\_ACCOUNT\_D would store the "Account" chartfield value in SEGMENT1 column, "Operating Unit" chartfield value in SEGMENT2 column, and so forth. The GL Balance aggregate table, W\_GL\_BALANCE\_A would store GL Balances for each unique combination of "Account" chartfield and "Department" chartfield.

**10.2.5.3.3 Group Account Configuration** You can categorize your PeopleSoft General Ledger accounts into specific group account numbers. The GROUP\_ACCT\_NUM field denotes the nature of the General Ledger accounts.

---

**Note:** It is critical that the General Ledger Account Numbers are mapped to the Group Account Numbers (or domain values) as the metrics in the General Ledger reporting layer uses these values.

---

For example, Cash account, Payroll account, and so on. For a list of the Group Account Number domain values, see Oracle Business Analytics Warehouse Data Model Reference Version 7.9. The group account num configuration is used during data extraction as well as front-end reporting. For example the group account num configuration is used heavily in both Profitability Analysis (Income Statement) and General Ledger analysis. The logic for assigning the accounts is located in the file\_group\_acct\_codes\_psft.csv file. This file is located in the <OracleBI>\dwrep\Informatica\SrcFiles folder.

**Table 10–10 Layout of file\_group\_acct\_codes\_psft.csv File**

BUSINESS_UNIT	FROM_ACCT	TO_ACCT	GROUP_ACCT_NUM
AUS01	101010	101099	AP
AUS01	131010	131939	AR
AUS01	152121	152401	COGS
AUS01	171101	173001	OTHER
AUS01	240100	240120	REVENUE
AUS01	251100	251120	TAX

In the Table above, in the first row, all accounts within the account number range from 101010 to 101099 containing a Business Unit equal to AUS01 are assigned to AP. Each row maps all accounts within the specified account number range and with the given Business Unit. If you need to assign a new group of account numbers, you can then assign GL accounts to the new group of account numbers in the file\_group\_acct\_codes\_psft.csv file.

**10.2.5.3.4 GL ChartField Hierarchy and Internal Organization Hierarchy Customization Process**

Oracle Business Intelligence Application provides a set of mechanism to support various tree structures in the PeopleSoft application. The following tables illustrate the supporting structure types and level types.

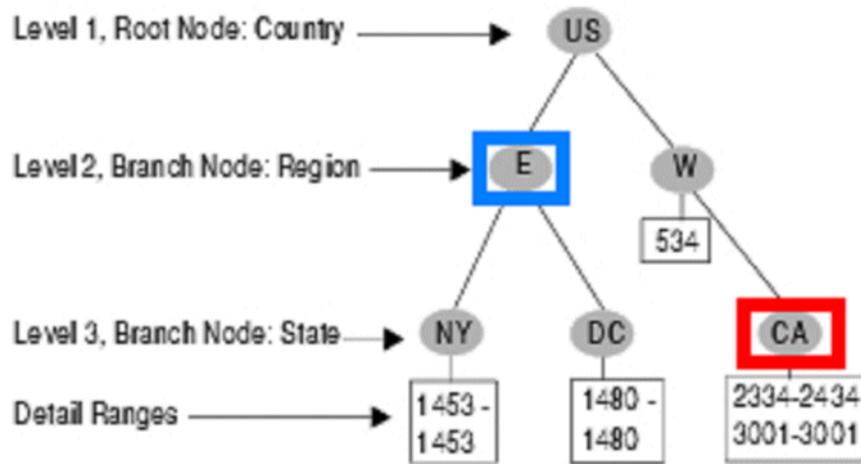
**Table 10–11 Structure Types**

Structure Type	Support
Winter Tree	Y
Detail Tree	Y
Summary Tree	Y

**Table 10–12 Level Types**

Level Type	Support
Strict-level Tree	Y
Loose-level Tree	Y

Figure 10–5 Example GL Chartfield hierarchy



For any nodes in the tree, except for the bottom nodes, W\_HIERARCHY\_D will have one corresponding record. As the "E" node in above tree (the one with blue square), you will have the following record in the database.

Table 10–13 Example CSV Chartfield Mapping Values

HIER1_CODE	HIER2_CODE	HIER3_CODE	HIER4_CODE	HIER5_CODE
US	E	E	E	E

If the actual hierarchy level in the tree ends earlier than the predefined level of W\_HIERARCHY\_D, then the lower hierarchy level in the table will repeat the same value as its next higher level. The predefined level is 20, which means if one path of the tree only has 5 level, so the level 6~20 will repeat the value of level 5's value.

For the leaf node in the detail tree shown above, W\_HIERARCHY\_D will have a record for every existing record within this detail range (The extraction process will find the single record in relevant table according to tree's structure). For example, in the CA node (the one with red square), the range is 2334-2434 and 3001-3001. The tree structure table (for example, State\_TBL) contains 2340, 3001 two records. When generating CA node, the extraction process will extract these two records, put them into W\_HIERARCH\_D as following structure. NOTE: This is purely for W\_HIERARCHY\_D, for W\_INT\_ORG\_DHS, the order of hierarchy is reversed.

Table 10–14 Example CSV Chartfield Mapping Values

HIER1_CODE	HIER2_CODE	HIER3_CODE	HIER4_CODE	HIER5_CODE
US	W	CA	2340	2340
US	W	CA	3001	3001

In both Summary Tree and Detail Tree, the lowest level node is always a node with specific value (as above example, the name and the code may be different). For the case of winter tree, the lowest level code would be the same as higher level.

**Figure 10–6 The Winter Loose-level Tree**



The record "Kuo" is stored in the database as follows:

**Table 10–15 Example CSV Chartfield Mapping Values**

HIER1_CODE	HIER2_CODE	HIER3_CODE	HIER4_CODE	HIER5_CODE
Smith	Cheng	Kuo	Kuo	Kuo

Oracle BI Applications uses the following tables to handle the extraction and load process of the tree. They are standard and not tree or tree-structure specific.

The General Ledger Chartfield Hierarchy is shown in the table below.

**Table 10–16 General Ledger Chartfield Hierarchy**

Sequence	Table Name
1	W_PSFT_TREE_TMP
1	W_PSFT_GLHIER_CF_DH_TMP
2	W_PSFT_GLHIER_DTLRGE_DH_TMP
2	W_PSFT_GLHIER_VERT_DH_TMP
3	W_PSFT_GLHIER_FLAT_DH_TMP
4	W_HIERARCHY_DS

The Internal Organization Hierarchy is shown in the table below.

**Table 10–17 Internal Organization Hierarchy**

Sequence	Table Name
1	W_PSFT_INT_ORG_TREE_TMP
1	W_PSFT_INT_ORG_GLBUDH_TMP
2	W_PSFT_INT_ORG_DTLRGE_DH_TMP
2	W_PSFT_INT_ORG_VERT_DH_TMP
3	W_PSFT_INT_ORG_FLAT_DH_TMP
4	W_INT_ORG_DHS
5	W_INT_ORG_DH

---



---

**Note:** The hierarchy related extraction and load tasks are deactivated in the standard application. To activate the hierarchy extractions and load tasks, you need to activate certain Configuration Tags in Oracle Data Warehouse Administration Console (DAC).

---



---

Configuration Tags is a DAC feature, which allows users to activate or deactivate a certain set of tasks in your ETL process. When deactivated, those set of tasks will not get executed when running the execution plan. When activated, those set of tasks will get executed when running the execution plan.

Oracle BI Applications includes two configuration tags for hierarchy extractions and load, as follows:

- "PeopleSoft - Populate Non Business Unit Related Hierarchies", which contains all but business unit related hierarchies' extractions and load tasks.
- "PeopleSoft - Populate Business Unit Related Hierarchies", which contains Business Unit related hierarchies' extractions and load tasks.

By simply activating and de-activating them in a Subject Area, you can control whether these hierarchy extraction and load tasks get executed in the execution plan.

To enable a set of tasks in your ETL process:

1. Open DAC, click "Design" button in navigation panel, and click "Configuration Tag".
2. De-select the "Inactive" checkbox of the specific hierarchy you want to enable.

For GL Chartfield, de-select the checkbox of "PeopleSoft - Populate Non Business Unit Related Hierarchies". For Business Unit, de-select the checkbox of "PeopleSoft - Populate Business Unit Related Hierarchies".

3. Save your changes
4. Click tab "Subject Area", check subject area one by one.

For every subject area, if the hierarchy-relevant item in its Configuration Tag (see the panel below subject area, there is a "Configuration Tag" tab) is inactive, de-select it, and save it.

As every tree in PeopleSoft is an identical item, each of them is extracted into data warehouse separately – that means it is required to create a new mapping if a new tree is to be extracted.

Oracle Business Intelligence Application provides a set of example mappings to extract Internal Organization Hierarchy and General Ledger Chartfield Hierarchy related trees. Users can customize their own extractions by simply copying some of those mappings and modifying one or two naming parameters.

**10.2.5.3.5 General Ledger Chartfield Hierarchy Extraction Customization** This section includes an example set of mappings for General Ledger Chartfield extraction. This example extracts a GL Account tree named "ACCTROLLUP" with SETID "SHARE" from the PeopleSoft application. To get other GL Chartfield related trees to be extracted, you need to customize DE\_PSFT\_Stage\_GLHierarchy\_AccountHierarchy\_Extract and SDE\_PSFT\_Stage\_GLHierarchy\_GetAccountChartField.

You do not need to change or copy the rest of the mappings. The example customization is provided as below.

**Table 10–18 General Ledger Chartfield Hierarchy Mappings**

Exec Sequence	Mapping Name	Needs Customization?
1	SDE_PSFT_Stage_GLHierarchy_AccountHierarchy_Extract	Y
1	SDE_PSFT_Stage_GLHierarchy_GetAccountChartField	Y
2	SDE_PSFT_Stage_GLHierarchy_GetHierarchyLevel	N
2	SDE_PSFT_Stage_GLHierarchy_DeriveRange	N
3	SDE_PSFT_Stage_GLHierarchy_Flatten	N
4	SDE_PSFT_GLHierarchy_Derive	N

For example, if a user wants to extract a tree named "DEPARTMENT\_HIERARCHY" with SETID "XYZ", which uses DEPT\_TBL as its structure's Record Name, DEPTID as Field Name (that means it builds up the hierarchy for Department chartfield from General Ledger), the following customization steps are required:

1. Create new mappings by copying the existing SDE\_PSFT\_Stage\_GLHierarchy\_AccountHierarchy\_Extract and SDE\_PSFT\_Stage\_GLHierarchy\_GetAccountChartField mappings.  
Name the new mappings SDE\_PSFT\_Stage\_GLHierarchy\_DepartmentHierarchy\_Extract and SDE\_PSFT\_Stage\_GLHierarchy\_GetDepartmentChartField.
2. Adjust Settings in SDE\_PSFT\_Stage\_GLHierarchy\_DepartmentHierarchy\_Extract, as follows:
  - a. In SDE\_PSFT\_Stage\_GLHierarchy\_DepartmentHierarchy\_Extract, open Source Definition PSFT\_TREE\_VERTICAL\_FLATTENING.
  - b. Under the "Properties" tab, change the Tree Name to "DEPARTMENT\_HIERARCHY", and Set ID to "XYZ".
3. Adjust Settings in SDE\_PSFT\_Stage\_GLHierarchy\_GetDepartmentChartField.  
In the SDE\_PSFT\_Stage\_GLHierarchy\_GetDepartmentChartField mapping, change the source table to "DEPT\_TBL" (also change the Source Qualifier). Connect SETID, DEPTID, EFFDT, EFF\_STATUS, and DESCR from Source Qualifier to next Filter "Filter\_GLACCT" corresponding fields (connect DEPTID to CHARTFIELD).  
In the last Expression Transformation of this mapping ("Exp\_TMPGLACCT"), change the expression field SOURCE\_TABLE\_NAME field to "DEPT\_TBL" (the same name as the source table).
4. Save all your changes in the Informatica folder.
5. Add Corresponding Sessions and Workflows in Informatica Workflow Manager, as follows:
  - a. Open Informatica Workflow Manager
  - b. Locate folder "SDE\_PSFT\_88\_adapter" (or "SDE\_PSFT\_84\_adapter").
  - c. Open Task Developer and create a new task for the that mappings you just created.  
Referring to the parameters in the original sample sessions (see table below, and leave other mappings set to the default value), modify these newly created sessions.

Parameter Configurations for SDE\_PSFT\_Stage\_GLHierarchy\_  
DepartmentHierarchy\_Extract Session are shown in the table below:

**Table 10–19 General Ledger Chartfield Hierarchy Mappings**

Tab Name	Parameter Name	Value
Properties	Parameter Filename	\$PMSourceFileDir\parameterfileOLTP.txt
Properties	\$Source connection value	\$DBConnection_OLTP
Properties	\$Target connection value	\$DBConnection_OLAP
Properties	DTM buffer size	32000000
Config Object	Default buffer block size	128000
Config Object	Stop on errors	1
Config Object	Override tracing	Normal
Mapping	ConnectionW_PSFT_TREE_TMP - Rational	\$DBConnection_OLAP
Mapping	ConnectionSQ_PSFT_TREE_VERTICAL_FLATTENING - Application	\$DBConnection_OLTP
Mapping	ConnectionW_PSFT_TREE_TMP1 - Rational	DBConnection_OLAP
Mapping	ConnectionLkp_Tree_ChartField - Rational	\$DBConnection_OLTP
Mapping	SDE_PSFT_Stage_GLHierarchy_ DepartmentHierarchy_Extract - Session\$Source Connection Value	\$DBConnection_OLTP
Mapping	SDE_PSFT_Stage_GLHierarchy_ DepartmentHierarchy_Extract - Session\$Target Connection Value	\$DBConnection_OLTP

---

**Note:** The SDE\_PSFT\_Stage\_GLHierarchy\_  
GetDepartmentChartField parameters are similar to this.

---

- d. Save your changes
  - e. Open Workflow Designer.
  - f. Create a Workflow with the same name as Session.
  - g. Drag the newly created session to the panel.
  - h. Connect the start with the session.
6. Save and Check-in all your changes in Informatica Designer and Workflow Manager.
  7. Re-configure the DAC settings, as follows:
    - a. Log in to the DAC and display the Task tab.
    - b. Add those two newly modified workflows as a task, right-click the task, and choose "synchronize" to update those newly added tasks.

- c. Add your tasks into Configuration Tags: navigate to Configuration Tag panel (click "Design", choose "Configuration Tag"). Select "PeopleSoft - Populate Non Business Unit Related Hierarchies" item. Select "Tasks" tab in the panel below. Click "Add/Remove" button; query the new mappings you just entered into DAC. Select mappings; add them into the right panel. Click OK, then click Save.
- d. Reassemble all the financial related areas: Click "Design" button at the top tool bar of DAC, select "Subject Areas" tab, for each of the financial-related subject areas (Cost of Goods Sold, General Ledger, Payables, Receivables, Revenue), right click on it, choose "assemble" item.
- e. Re-build related execution plan: Click "Execute" button at the top tool bar of DAC, select "Execution Plans" tab, find the executing plan for your PeopleSoft Financial Application (for example Financials\_PeopleSoft 8.8), right click on it, choose "build" item.

**10.2.5.3.6 Internal Organization Hierarchy Extraction Customization** Internal Organization Hierarchy follows a similar logic to the previous section. But since it only extracts the hierarchy related to Business Unit (unlike GL Chartfield, there are many different types of trees), only the Tree Name and SETID are required to change.

**Table 10–20 General Ledger Chartfield Hierarchy Mappings**

Exec Sequence	Mapping Name	Needs Customization?
1	SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_Extract	Y
1	SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_GetGLBU	N
2	SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_GetHierarchyLevels	N
2	SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_DeriveRange	N
3	SDE_PSFT_Stage_InternalOrganizationDimension_CompanyHierarchy_Flatten	N
4	SDE_PSFT_InternalOrganizationDimension_CompanyHierarchy_Derive	N

As the above table illustrates, only the SDE\_PSFT\_Stage\_InternalOrganizationDimension\_CompanyHierarchy\_Extract mapping needs to be modified (that is, there is no need to change or copy the rest mappings). For example, if you want to extract a tree named "BIZ\_UNIT\_HIER" with SETID "XYZ", which uses BU\_GL\_UPDAT2\_VW as its structure's Record Name, and BUSINESS\_UNIT as its field name (that means it builds up hierarchy for BUSINESS\_UNIT), do the following:

1. Create a new mapping by copying the existing SDE\_PSFT\_Stage\_InternalOrganizationDimension\_CompanyHierarchy\_Extract mapping.  
Name the new mappings SDE\_PSFT\_Stage\_InternalOrganizationDimension\_CompanyHierarchy2\_Extract.
2. Adjust Settings in SDE\_PSFT\_Stage\_InternalOrganizationDimension\_CompanyHierarchy2\_Extract, as follows:
  - a. In SDE\_PSFT\_Stage\_InternalOrganizationDimension\_CompanyHierarchy2\_Extract, open Source Definition PSFT\_TREE\_VERTICAL\_FLATTENING.

- b. Under the "Properties" tab, change the Tree Name to "BIZ\_UNIT\_HIER", and Set ID to "XYZ".
3. Add Corresponding Sessions and Workflows in Informatica Workflow Manager.
4. Save your changes in the Informatica Folder.
5. Add Corresponding Sessions and Workflows in Informatica Workflow Manager, as follows:
  - a. Open Informatica Workflow Manager, find folder "SDE\_PSFT\_88\_adapter" (or "SDE\_PSFT\_84\_adapter")
  - b. Open Task Developer, and create new task for the mappings that you just created.
  - c. Using the parameters in the original sample sessions as a guide (refer to the parameter settings for SDE\_PSFT\_Stage\_GLHierarchy\_DepartmentHierarchy\_Extract), modify these newly created session.
  - d. Save your changes.
  - e. Open Workflow Designer and create Workflow with the same name as the Session.
  - f. Drag the newly created session to the panel and connect the start with the session.
6. Save and Check-in all your changes in Informatica Designer and Workflow Manager.
7. Re-configure the DAC settings, as follows:
  - a. Log in to the DAC and display the Task tab.
  - b. Add those two newly modified workflows as a task, right-click the task, and choose "synchronize" to update those newly added tasks.
  - c. Add your tasks into Configuration Tags: navigate to Configuration Tag panel (click "Design", choose "Configuration Tag"). Select "PeopleSoft - Populate Non Business Unit Related Hierarchies" item. Select "Tasks" tab in the panel below. Click "Add/Remove" button; query the new mappings you just entered into DAC. Select mappings; add them into the right panel. Click OK, then click Save.
  - d. Reassemble all the financial related areas: Click "Design" button at the top tool bar of DAC, select "Subject Areas" tab, for each of the financial-related subject areas (Cost of Goods Sold, General Ledger, Payables, Receivables, Revenue), right click on it, choose "assemble" item.
  - e. Re-build related execution plan: Click "Execute" button at the top tool bar of DAC, select "Execution Plans" tab, find the executing plan for your PeopleSoft Financial Application (for example Financials\_PeopleSoft 8.8), right click on it, choose "build" item.

#### 10.2.5.4 Configuration Steps for Financial Analytics Chain For Universal

This section contains additional configuration steps that apply to Universal, and contains the following topics:

- [Section 10.2.5.4.1, "About the Customer Costs Lines and Product Costs Lines Tables for Oracle's Profitability Analytics"](#)
- [Section 10.2.5.4.2, "How to Configure the Customer Costs Lines and Product Costs Lines Tables for Oracle's Profitability Analytics"](#)

**10.2.5.4.1 About the Customer Costs Lines and Product Costs Lines Tables for Oracle's Profitability Analytics** This configuration is required only if you are implementing Oracle Profitability Analytics and you want to allocate your expenses by product or by customer dimensions. The out-of-the-box adapter of Oracle BI Applications does not capture the miscellaneous cost and expenses associated with generating revenue from a customer or from a product (for example, expenses such as marketing campaign expenses). Since the way that these miscellaneous expenses are allocated across products or customers can be very different between different companies, we do not perform this type of allocations out of box; we require customers to feed us the result of the allocations through Universal Adapter.

The Customer Costs Lines (W\_CUSTOMER\_COST\_LINE\_F) table stores cost details by customers. The total cost by customers include both direct cost that is captured in the financial system by products, and the allocated costs that are captured in the costing system. The table also stores the source of allocations. For example, the Sales and Marketing costs are not be captured as direct costs by customers. However, at a later point in time, the costs could be allocated from a combination of General Ledger accounts or Cost Centers into various customers. This table stores the source cost centers and General Ledger accounts as well. The Customer Costs can be categorized by Sales, Marketing, Service, and Operating costs. These could be further broken down to Salaries, Events, and Promotions. Oracle Business Analytics Warehouse provides a set of categories, and these can be changed depending on the user needs and cost components by which customers are tracked. The actual cost lines are also tracked in this table. Apart from Customers, there are number of other dimensions by which the costs can be tracked such as Sales Region, Sales Geography, Company, Business Area and the associated hierarchies. The different cost lines such as Sales, Marketing, and Operating costs, have different details and many dimensions are included in this table. Some of the dimension keys are not be applicable to certain cost components. It is important that an organization identifies the dimensional keys that are used for Product Cost Analysis for various components.

The Product Costs Lines (W\_PRODUCT\_COST\_LINE\_F) table stores cost details by products. The total cost by products include both direct cost that is captured in the financial system by products, and the allocated costs that are captured in the costing system. The table also stores the source of allocations. For example, the Sales and Marketing costs are not captured as direct costs by products. However, at a later point in time, the costs are allocated from a combination of General Ledger accounts and Cost Centers into various products. This table also stores the source cost centers and General Ledger accounts. The product costs can be categorized by Sales, Marketing, Service, and Operating costs. These could be further broken down into Salaries, Events, and Promotions. Oracle Business Analytics Warehouse provides a set of common categories, and these can be changed depending on the user needs and cost components by which products are tracked. The actual cost lines are also tracked in this table. Apart from products, there are number of other dimensions by which the costs are tracked such as Sales Region, Sales Geography, Company, Business Area, and the associated hierarchies. The different cost lines such as Sales, Marketing, and Operating costs, have different details and many dimensions are included in this table. Some of the dimension keys are not applicable to certain cost components. It is important that an organization identifies the dimensional keys that is used for Product Cost Analysis for various components.

**10.2.5.4.2 How to Configure the Customer Costs Lines and Product Costs Lines Tables for Oracle's Profitability Analytics** In Oracle's Profitability Analytics, the Customer Costs Lines and Product Costs Lines fact tables store the costing and expenses for the Profitability functional area. You need to use these tables with General Ledger Revenue and General Ledger COGS fact tables.

The General Ledger Revenue and General Ledger COGS fact tables are populated by the or Oracle 11i source system, but the Customer Costs Lines and Product Costs Lines fact tables are populated by the universal source system.

**To load the Customer Costs Lines and Product Costs Lines tables**

1. Open the `customer_cost_line.csv` file in the `$pmserver\Sample Universal Source Files` folder.
2. Insert a record into the file for each customer costing transaction you want to load into the Customer Cost fact table.

For the `DATASOURCE_NUM_ID` column, you need to provide the same source identifier value as in the `file_parameters_orall11i.csv` file.

3. Save the file in the `$pmserver\srcfiles` directory, then close the file.
4. Repeat steps 1 to 3 for the `file_product_cost.csv` file.



---

---

# Configuring Oracle Order Management and Fulfillment Analytics

This chapter describes how to configure Order Management and Fulfillment Analytics for particular sources to meet your business needs.

It contains the following topics:

- [Section 11.1, "Overview of Oracle Order Management and Fulfillment Analytics"](#)
- [Section 11.2, "Configuration Required Before A Full Load for Oracle Order Management and Fulfillment Analytics"](#)

## 11.1 Overview of Oracle Order Management and Fulfillment Analytics

The Oracle Order Management and Fulfillment Analytics application allows you to analyze the movement of sales orders through different stages of the sales cycle. This analysis includes insight into which items are booked, backlogged, and invoiced. It also provides you with information that allows you to evaluate the sales performance of individual sales representatives or departments. The Oracle Order Management and Fulfillment Analytics application contains the functional area, Orders and Revenue.

The Oracle Order Management and Fulfillment Analytics application consists of orders, invoices and backlogs. Sales orders are the entry point for the sales process. Invoices are the exit point from the fulfillment process. *Backlogs* are points of congestion in your fulfillment process.

In the Oracle Order Management and Fulfillment Analytics application, two main types of backlog exist:

- Operational
- Financial

The scheduled, unscheduled, delinquent, and blocked backlogs belong to the Operational backlog. The following sources can populate Orders and Revenue:

- Oracle 11i
- Oracle R12
- Universal source

The Oracle Order Management and Fulfillment Analytics application also requires post-load processing mappings to populate its tables.

## 11.2 Configuration Required Before A Full Load for Oracle Order Management and Fulfillment Analytics

This section contains configuration steps that you need to perform on Oracle Order Management and Fulfillment Analytics before you do a full data load, and contains the following topics:

- [Section 11.2.1, "Configuration Steps for Oracle Order Management and Fulfillment Analytics For All Source Systems"](#)
- [Section 11.2.2, "Configuration Steps for Oracle Order Management and Fulfillment Analytics For Oracle EBS"](#)
- [Section 11.2.3, "Configuration Steps for Oracle Order Management and Fulfillment Analytics For PeopleSoft"](#)
- [Section 11.2.4, "Configuration Steps for Oracle Order Management and Fulfillment Analytics For Universal"](#)

### 11.2.1 Configuration Steps for Oracle Order Management and Fulfillment Analytics For All Source Systems

This section contains configuration steps that apply to all source systems.

---

---

**Note:** For configuration steps that apply to all analytics modules (for example, Oracle Financial Analytics, Oracle HR Analytics, Oracle Sales Analytics, see [Chapter 8, "Configuring Common Areas and Dimensions"](#)).

---

---

### 11.2.2 Configuration Steps for Oracle Order Management and Fulfillment Analytics For Oracle EBS

This section contains configuration steps required before you do a full data load that apply to Oracle EBS.

- [Section 11.2.2.1, "About Configuring Domain Values and CSV Worksheet Files for Oracle Order Management and Fulfillment Analytics"](#).
- [Section 11.2.2.2, "How to Configure Invoice Type Domain Values"](#).
- [Section 11.2.2.3, "How to Configure Pick Types Domain Values"](#).
- [Section 11.2.2.4, "How to Configure Order Types Domain Values"](#)
- [Section 11.2.2.5, "How to Configure Pick Status Domain Values"](#)
- [Section 11.2.2.6, "How to Configure Invoice Status Domain Values"](#)
- [Section 11.2.2.7, "How to Configure Order Overall Status Domain Values"](#)
- [Section 11.2.2.8, "How to Configure Pay Method Domain Values"](#)

#### 11.2.2.1 About Configuring Domain Values and CSV Worksheet Files for Oracle Order Management and Fulfillment Analytics

The table below lists the CSV worksheet files and the domain values for Oracle Order Management and Fulfillment Analytics in the `$pmsserver\LkpFiles` folder.

**Table 11–1 Domain Values and CSV Worksheet Files for Oracle Order Management and Fulfillment Analytics**

Worksheet File Name	Description	Session
domainValues_InvoiceTypes_ora11i.csv	Lists the Invoice Document Type column and the corresponding domain values for the Oracle 11i or Oracle R12 application.  For more information about updating the values in this file, see <a href="#">Section 11.2.2.2, "How to Configure Invoice Type Domain Values"</a> .	SDE_ORA_TransactionTypeDimension_SalesInvoiceLines
domainValues_PickTypes_ora11i.csv	Lists the Picking Document Type column and the corresponding domain values for the Oracle 11i or Oracle R12 application.  For more information about updating the values in this file, see <a href="#">Section 11.2.2.3, "How to Configure Pick Types Domain Values"</a> .	SDE_ORA_TransactionTypeDimension_SalesPickLines
domainValues_OrderTypes_ora11i.csv	Lists the Order Document Type column and the corresponding domain values for the Oracle 11i or Oracle R12 application.  For more information about updating the values in this file, see <a href="#">Section 11.2.2.4, "How to Configure Order Types Domain Values"</a> .	SDE_ORA_TransactionTypeDimension_SalesOrderLines
domainValues_PickStatus_ora11i.csv	Lists the Picking Status Code and the Status Desc columns, and the corresponding domain values for the Oracle 11i or Oracle R12 application.  For more information about updating the values in this file, see <a href="#">Section 11.2.2.5, "How to Configure Pick Status Domain Values"</a> .	SDE_ORA_StatusDimension_SalesPickLines
domainValues_PayMethodCode_ora11i.csv	Lists the method code column and the corresponding domain value for the application.	SDE_ORA_PaymentMethodDimension
domainValues_InvoiceStatus_ora11i.csv	Lists the Invoice Status Code and the Status Desc columns, and the corresponding domain values for the Oracle 11i or Oracle R12 application.  For more information about updating the values in this file, see <a href="#">Section 11.2.2.6, "How to Configure Invoice Status Domain Values"</a> .	SDE_ORA_StatusDimension_SalesInvoiceLine

**Table 11–1 (Cont.) Domain Values and CSV Worksheet Files for Oracle Order Management and Fulfillment Analytics**

Worksheet File Name	Description	Session
DomainValue_ OrderOverallStatus_ ora11i.csv	List the Order Status Code column and the corresponding domain values for the Oracle 11i or Oracle R12 application.  For more information about updating the values in this file, see <a href="#">Section 11.2.2.7, "How to Configure Order Overall Status Domain Values"</a> .	SDE_ORA_ StatusDimension_ SalesOrderLineCycle

For general information about domain values with CSV worksheet files, see [Section 6.12, "About Domain Values"](#) and [Section 6.13, "Configuring the Domain Value Set with CSV Worksheet Files"](#).

### 11.2.2.2 How to Configure Invoice Type Domain Values

This section explains how to configure Invoice Type Domain Values using the domainValues\_InvoiceTypes\_ora11i.csv file.

To configure Invoice Type Domain Values:

1. Identify the Invoice Types in your Oracle 11i source system by using the following SQL:

```
SELECT DISTINCT RA_CUST_TRX_TYPES_ALL.TYPE
FROM RA_CUST_TRX_TYPES_ALL
ORDER BY 1;
```

2. Open the domainValues\_InvoiceType\_ora11i.csv file using a text editor in the \$pmserver\lkpfiles folder.
3. Copy the TYPE column to the XACT\_TYPE\_CODE column in the file.  
The data must be copied starting from the 2nd line. The first line is the column header.
4. Map each Transaction Type Code to one domain value.  
For more information on Transaction Type Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

### 11.2.2.3 How to Configure Pick Types Domain Values

This section explains how to configure Pick Types Domain Values using the domainValues\_PickTypes\_ora11i.csv file.

To configure Pick Types Domain Values:

1. Identify the Pick Types in your Oracle 11i source system.
2. Open the domainValues\_PickType\_ora11i.csv file using a text editor in the \$pmserver\lkpfiles folder.
3. Put 'STANDARD' in the XACT\_TYPE\_CODE column in the file.  
The data must be copied starting from the 2nd line. The first line is the column header.

4. Map each Transaction Type Code to one domain value.

For more information on Transaction Type Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

5. Save and close the file.

#### 11.2.2.4 How to Configure Order Types Domain Values

This section explains how to configure Order Types Domain Values using the domainValues\_OrderTypes\_ora11i.csv file.

To configure Order Types Domain Values:

1. Identify the Pick Types in your Oracle 11i source system by using the following SQL:

```
SELECT DISTINCT FND_LOOKUP_VALUES.LOOKUP_CODE
FROM FND_LOOKUP_VALUES
WHERE FND_LOOKUP_VALUES.VIEW_APPLICATION_ID = 660
AND FND_LOOKUP_VALUES.LANGUAGE = 'US'
AND FND_LOOKUP_VALUES.LOOKUP_TYPE = 'LINE_CATEGORY'
ORDER BY 1;
```

2. Open the domainValues\_OrderType\_ora11i.csv file using a text editor in the \$pmsserver\lkpfiles folder.
3. Copy the LOOKUP\_TYPE column to the XACT\_TYPE\_CODE column in the file. The data must be copied starting from the 2nd line. The first line is the column header.
4. Map each Transaction Type Code to one domain value.

For more information on Transaction Type Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

5. Save and close the file.

#### 11.2.2.5 How to Configure Pick Status Domain Values

This section explains how to configure Pick Status Domain Values using the domainValues\_PickStatus\_ora11i.csv file.

To configure Pick Status Domain Values:

1. Identify the Pick Statuses in your Oracle 11i source system by using the following SQL:

```
SELECT DISTINCT FND_LOOKUP_VALUES.LOOKUP_CODE
FROM FND_LOOKUP_VALUES
WHERE FND_LOOKUP_VALUES.LOOKUP_TYPE= 'PICK_STATUS'
AND FND_LOOKUP_VALUES.LANGUAGE = 'US'
AND FND_LOOKUP_VALUES.VIEW_APPLICATION_ID = 665
AND FND_LOOKUP_VALUES.SECURITY_GROUP_ID = 0
ORDER BY 1;
```

2. Open the domainValues\_PickStatus\_ora11i.csv file using a text editor in the \$pmsserver\lkpfiles folder.
3. Copy the LOOKUP\_CODE column to the STATUS\_CODE column in the file.  
The data must be copied starting from the 2nd line. The first line is the column header.
4. Map each Status Code to one domain value.  
For more information on Status Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

### 11.2.2.6 How to Configure Invoice Status Domain Values

This section explains how to configure Invoice Status Domain Values using the domainValues\_InvoiceStatus\_ora11i.csv file.

To configure Invoice Status Domain Values:

1. Identify the Invoice Statuses in your Oracle 11i source system by using the following SQL:
 

```
SELECT DISTINCT FND_LOOKUP_VALUES.LOOKUP_CODE
FROM FND_LOOKUP_VALUES
WHERE FND_LOOKUP_VALUES.LOOKUP_TYPE= 'INVOICE_TRX_STATUS' AND
FND_LOOKUP_VALUES.LANGUAGE = 'US'
AND FND_LOOKUP_VALUES.VIEW_APPLICATION_ID = 222
AND FND_LOOKUP_VALUES.SECURITY_GROUP_ID = 0
ORDER BY 1;
```
2. Open the domainValues\_InvoiceStatus\_ora11i.csv file using a text editor in the \$pmsserver\lkpfiles folder.
3. Copy the LOOKUP\_CODE column to the STATUS\_CODE column in the file.  
The data must be copied starting from the 2nd line. The first line is the column header.
4. Map each Status Code to one domain value.  
For more information on Status Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

### 11.2.2.7 How to Configure Order Overall Status Domain Values

This section explains how to configure Order Overall Status Domain Values using the domainValues\_OrderOverallStatus\_ora11i.csv file.

To configure Order Overall Status Domain Values:

1. Identify the Order Overall Statuses in your Oracle 11i source system by using the following SQL:
 

```
SELECT DISTINCT FND_LOOKUP_VALUES.LOOKUP_CODE
FROM FND_LOOKUP_VALUES
WHERE FND_LOOKUP_VALUES.LOOKUP_TYPE = 'LINE_FLOW_STATUS'
```

```

AND FND_LOOKUP_VALUES.LANGUAGE = 'US'
AND FND_LOOKUP_VALUES.VIEW_APPLICATION_ID = 660
AND FND_LOOKUP_VALUES.SECURITY_GROUP_ID = 0
ORDER BY 1;

```

2. Open the domainValues\_OrderOverallStatus\_ora11i.csv file using a text editor in the \$pmsserver\lkpfiles folder.
3. Copy the LOOKUP\_CODE column to the STATUS\_CODE column in the file.  
The data must be copied starting from the 2nd line. The first line is the column header.
4. Map each Status Code to one domain value.  
For more information on Status Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

### 11.2.2.8 How to Configure Pay Method Domain Values

This section explains how to configure Pay Method Status Domain Values using the domainValues\_PayMethodCode\_ora11i.csv file.

To configure Pay Method Domain Values:

1. Identify the Pay Methods in your Oracle 11i source system by using the following SQL:

```

SELECT DISTINCT FND_LOOKUP_VALUES.LOOKUP_CODE
FROM FND_LOOKUP_VALUES
WHERE LOOKUP_TYPE = 'PAYMENT TYPE'
AND VIEW_APPLICATION_ID = 660
AND LANGUAGE = 'US'
AND FND_LOOKUP_VALUES.SECURITY_GROUP_ID = 0
ORDER BY 1;

```

2. Open the domainValues\_PayMethodCode\_ora11i.csv file using a text editor in the \$pmsserver\lkpfiles folder.
3. Copy the LOOKUP\_CODE column to the METHOD\_CODE column in the file.  
The data must be copied starting from the 2nd line. The first line is the column header.
4. Map each Method Code to one domain value.  
For more information on Method Code domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.
5. Save and close the file.

## 11.2.3 Configuration Steps for Oracle Order Management and Fulfillment Analytics For PeopleSoft

This section contains configuration steps required before you do a full data load that apply to PeopleSoft.

Not applicable to Oracle BI Applications Version 7.9.4.

## 11.2.4 Configuration Steps for Oracle Order Management and Fulfillment Analytics For Universal

This section contains configuration steps required before you do a full data load that apply to Universal.

Not applicable to Oracle BI Applications Version 7.9.4.

## 11.2.5 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle Order Management and Fulfillment Analytics, and contains the following topics:

- [Section 11.2.5.1, "Configuration Steps for Oracle Order Management and Fulfillment Analytics For All Source Systems"](#)
- [Section 11.2.5.2, "Configuration Steps for Oracle Order Management and Fulfillment Analytics For Oracle EBS"](#)
- [Section 11.2.5.3, "Configuration Steps for Oracle Order Management and Fulfillment Analytics For PeopleSoft"](#)
- [Section 11.2.5.4, "Configuration Steps for Oracle Order Management and Fulfillment Analytics For Universal"](#)

### 11.2.5.1 Configuration Steps for Oracle Order Management and Fulfillment Analytics For All Source Systems

This section contains configuration steps that apply to all source systems.

**11.2.5.1.1 Tracking Multiple Attribute Changes in Bookings** When you modify the default `VAR_BOOKING_ID` column, the SQL statement is configured as follows for Oracle 11i and Oracle R12:

```
TO_CHAR(INP_LINE_ID) || '~' || TO_CHAR(INP_INV_ITEM_ID) || '~' || to_char(INP_WAREHOUSE_ID)
```

However, if you want to track changes based on more than one attribute, in the SQL statement you must concatenate the attribute column IDs in the `VAR_BOOKING_ID` column. For example, if you want to track changes in Salespersons and Customer, then concatenate the technical name IDs in the `VAR_BOOKING_ID` column as follows:

```
TO_CHAR(INP_LINE_ID) || '~' || TO_CHAR(INP_INV_ITEM_ID) || '~' || TO_CHAR(INP_WAREHOUSE_ID) || '~' || TO_CHAR(INP_SALESREP_ID) || '~' || TO_CHAR(INP_CUSTOMER_ID)
```

#### To track dimensional attribute changes in bookings

1. In PowerCenter Designer, open the `SDE_ORA115<ver>_adapter` or `SDE_ORAR12_adapter` folder.
2. Open the `mplt_SA_ORA_SalesOrderLinesFact` maplet.
3. Double-click the `EXP_SALES_ORDLNS` Expression transformation to open the Edit Transformation box.
4. In the Ports tab, edit the expression for the `VAR_BOOKING_ID` port, and enter the ID of the attribute for which you want to track changes.

If you want to track changes in multiple attributes, concatenate the IDs of all attributes and put the concatenated value in the `VAR_BOOKING_ID` column.

5. Validate and save your changes to the repository.

#### 11.2.5.1.2 Process of Aggregating Oracle Order Management and Fulfillment Analytics Tables

This section contains Oracle Order Management and Fulfillment Analytics configuration points for aggregating the Sales Invoice Lines and Sales Order Lines tables.

The aggregation processes use the following Teradata parameters:

- Hint\_Tera\_Pre\_Cast
- Hit\_Tera\_Post\_Cast

To aggregate the Sales Invoice Lines and Sales Order Lines tables, perform the following tasks:

- Configure the Sales Invoice Lines Aggregate Table
- Configure the Sales Order Lines Aggregate Table

#### About Configuring the Sales Invoice Lines Aggregate Table

The Sales Invoice Lines aggregate table (W\_SALES\_INVOICE\_LINE\_F\_A) is used to capture information about the invoices issued for your sales orders. You need to configure the Sales Invoice Lines aggregate table in order to run initial ETL and incremental ETL.

For your initial ETL run, you need to configure the TIME\_GRAIN parameter for the time aggregation level in the Sales Invoice Lines aggregate fact table.

For the incremental ETL run, you need to configure the time aggregation level.

To aggregate the Sales Invoice Lines table for your incremental run, you need to configure the TIME\_GRAIN parameter.

The TIME\_GRAIN parameter has a preconfigured value of Month. The possible values for the TIME\_GRAIN parameter are:

- 'DAY'
- 'WEEK'
- 'MONTH'
- 'QUARTER'
- 'YEAR'

The aggregation processes use the following Teradata parameters:

- Hint\_Tera\_Pre\_Cast
- Hit\_Tera\_Post\_Cast

The Sales Invoice Lines aggregate table is fully loaded from the base table in the initial ETL run. The table can grow to millions of records. Thus, the Sales Invoice aggregate table is not fully reloaded from the base table after each incremental ETL run. Oracle Business Analytics Warehouse minimizes the incremental aggregation effort, by modifying the aggregate table incrementally as the base table is updated. This process is described below.

- Oracle Business Analytics Warehouse finds the records to be deleted in the base table since the last ETL run, and loads them into the W\_SALES\_INVOICE\_LINE\_TMP table. The measures in these records are multiplied by (-1). The mapping responsible for this task is SIL\_SalesInvoiceLinesAggregate\_Derive\_

PreSoftDeleteImage, which is run before SIL\_SalesInvoiceLinesFact\_SoftDelete deletes the records from the base table.

- Oracle Business Analytics Warehouse finds the records to be updated in the base table since the last ETL run, and loads them into the W\_SALES\_INVOICE\_LINE\_TMP table. The measures in these records are multiplied by (-1). The mapping responsible for this task is SIL\_SalesInvoiceLinesFact\_Derive\_PreLoadImage, which is run before SIL\_SalesInvoiceFact deletes the records from the base table.
- Oracle Business Analytics Warehouse finds the inserted or updated records in the base table since the last ETL run, and loads them into the W\_SALES\_INVOICE\_LINE\_TMP table, without changing their sign. The mapping responsible for this task is SIL\_SalesInvoiceLinesFact\_Derive\_PreLoadImage, which is run before PLP\_SalesInvoiceLinesFact\_Derive\_PostLoadImage updates or inserts records in the base table.
- Oracle Business Analytics Warehouse aggregates the W\_SALES\_INVOICE\_LINE\_TMP table and load to W\_SALES\_INVOICE\_LINE\_A\_TMP, which has the same granularity as the W\_SALES\_INVOICE\_LINE\_A table.
- The PLP\_SalesInvoiceLinesAggregate\_Derive mapping looks up the W\_SALES\_INVOICE\_LINE\_A aggregate table to update existing buckets or insert new buckets in the aggregate table (the mapping is PLP\_SalesInvoiceLinesAggregate\_Load).

#### **How to Configure the Sales Invoice Lines Aggregate Table**

To load the Sales Invoice Lines aggregate table (W\_SALES\_INVOICE\_LINE\_A), you need to configure the parameterfileDW.txt file, and run the initial and then the incremental workflows.

##### **To configure the parameterfileDW.txt parameters file**

1. Open the parameterfileDW.txt file in a text editor.
  - This file is located in the OracleBI\DAC\Informatica\parameters\input folder.
2. Replace the default values with your new values.
3. Save and close the file.

##### **About Configuring the Sales Order Lines Aggregate Table**

The Sales Order Lines aggregate table (W\_SALES\_ORDER\_LINE\_A) is used to capture information about the order lines issued for your sales orders. You need to configure the Sales Order Lines aggregate table in order to run initial ETL and incremental ETL.

For your initial ETL run, you need to configure the TIME\_GRAIN parameter for the time aggregation level in the Sales Order Lines aggregate fact table.

For the incremental ETL run, you need to configure the time aggregation level.

To aggregate the Sales Invoice Lines table for your incremental run, you need to configure the TIME\_GRAIN parameter.

The TIME\_GRAIN parameter has a preconfigured value of Month. The possible values for the GRAIN parameter are:

- 'DAY'
- 'WEEK'
- 'MONTH'
- 'QUARTER'

- 'YEAR'

The aggregation processes use the following Teradata parameters:

- Hint\_Tera\_Pre\_Cast
- Hit\_Tera\_Post\_Cast

The Sales Order Lines aggregate table is fully loaded from the base table in the initial ETL run. The table can grow to millions of records. Thus, the Sales Order aggregate table is not fully reloaded from the base table after each incremental ETL run. Oracle Business Analytics Warehouse minimizes the incremental aggregation effort, by modifying the aggregate table incrementally as the base table is updated. This process is described below.

- Oracle Business Analytics Warehouse finds the records to be deleted in the base table since the last ETL run, and loads them into the W\_SALES\_ORDER\_LINE\_TMP table. The measures in these records are multiplied by (-1). The mapping responsible for this task is SIL\_SalesOrderLinesAggregate\_Derive\_PreSoftDeleteImage, which is run before SIL\_SalesOrderLinesFact\_SoftDelete deletes the records from the base table.
- Oracle Business Analytics Warehouse finds the records to be updated in the base table since the last ETL run, and loads them into the W\_SALES\_ORDER\_LINE\_TMP table. The measures in these records are multiplied by (-1). The mapping responsible for this task is SIL\_SalesOrderLinesFact\_Derive\_PreLoadImage, which is run before SIL\_SalesOrderFact updates the records from the base table.
- Oracle Business Analytics Warehouse finds the inserted or updated records in the base table since the last ETL run, and loads them into the W\_SALES\_ORDER\_LINE\_TMP table, without changing their sign. The mapping responsible for this task is SIL\_SalesOrderLinesFact\_Derive\_PreLoadImage, which is run before PLP\_SalesOrderLinesFact\_Derive\_PostLoadImage updates or inserts records in the base table.
- Oracle Business Analytics Warehouse uses the PLP\_SalesOrderLinesAggregate\_Derive mapping to aggregate the W\_SALES\_ORDER\_LINE\_TMP table and load to W\_SALES\_ORDER\_LINE\_A\_TMP, which has the same granularity as the W\_SALES\_ORDER\_LINE\_A table.
- W\_SALES\_ORDER\_LINE\_A\_TMP looks up the W\_SALES\_ORDER\_LINE\_A aggregate table to update existing buckets or insert new buckets in the aggregate table (the mapping is PLP\_SalesOrderLinesAggregate\_Load).

### How to Configure the Sales Order Lines Aggregate Table

To load the Sales Order Lines aggregate table (W\_SALES\_ORDER\_LINE\_A), you need to configure the post-load-processing parameter file and the source system parameter files, and run the initial and then the incremental workflows.

#### To configure the Sales Order Lines Aggregate Table

1. Open the parameterfileDW.txt file in a text editor.

This file is located in the OracleBI\DAC\Informatica\parameters\input folder.

2. Replace the default values with your new values.
3. Save and close the file.

**11.2.5.1.3 About Tracking Multiple Products for Oracle Order Management and Fulfillment Analytics** The Sales Order Lines table contains two columns, `ORDHD_KEY_ID` and `ORDLN_KEY_ID`, that track individual products when they are grouped and sold as a single package. These two columns allow you to analyze the relationship of all products sold as a single unit. The `ORDHD_KEY_ID` column stores the Order ID of the entire sales order. The `ORDLN_KEY_ID` column stores the Line Item ID of the parent product.

For example, assume a customer purchases a package that includes a computer, scanner, and printer. In addition, the customer purchases a monitor separately. In this case, there are two parent items: the package and the monitor. The computer, scanner, and printer are all child orders of the parent order *package*, while the parent order *monitor* is a single-item purchase.

Your data warehouse may store this sales information in the Sales Order Lines table as seen in [Table 11–2](#). The `ORDLN_KEY_ID` field contains the Line Item ID of the parent product in order to maintain the relationship between the parent and child products in a package. In this example, the `ORDLN_KEY_ID` field is `Line_1` for each of the three child products (A1, A2, A3) that were sold as a part of the parent package, Parent A.

**Table 11–2 Sales Order Table Columns With Parent/Child Relationships**

Key_ID	SALES_ORDER_NUM	PRODUCT_ID	ORDHD_KEY_ID	ORDLN_KEY_ID	Relationship (Not a column in the table.)
Line_1	1000	Package	1000	Line_1	Parent A
Line_2	1000	Computer	1000	Line_1	Child A1
Line_3	1000	Scanner	1000	Line_1	Child A2
Line_4	1000	Printer	1000	Line_1	Child A3
Line_5	1000	Monitor	1000	Line_5	Parent B (no children)

In contrast, if each of the four items described in [Table 11–2](#) were bought individually, the `ORDLN_KEY_ID` would have a different Line Item ID for every row. In this case, the Sales Order Lines table would look like [Table 11–3](#).

**Table 11–3 Sales Order Table Columns Without Parent/Child Relationships**

Key_ID	SALES_ORDER_NUM	PRODUCT_ID	ORDHD_KEY_ID	ORDLN_KEY_ID	Relationship (Not a column in the table.)
Line_1	1000	Computer	1000	Line_1	None
Line_2	1000	Scanner	1000	Line_2	None
Line_3	1000	Printer	1000	Line_3	None
Line_4	1000	Monitor	1000	Line_4	None

**11.2.5.1.4 Adding Dates to the Order Cycle Time Table for Post-Load Processing** To add more dates, you need to understand how the Order Cycle Times table is populated. Thus, if you want to change the dates loaded into the Order Cycle Time table (`W_SALES_CYCLE_LINE_F`), then you have to modify the `PLP_SalesCycleLinesFact_Load` and `PLP_SalesCycleLinesFact_Load_Full` mappings that take the dates from the `w_*` tables and load them into the Cycle Time table.

**To add dates to the Cycle Time table load**

1. In the PowerCenter Designer, open the Configuration for Post Load Processing folder.
2. In Warehouse Designer, modify the table definition for the target table to verify that it has a field to store this date.

For example, if you are loading the Validated on Date in the W\_SALES\_CYCLE\_LINE\_F table, then you need to create a new column, VALIDATED\_ON\_DT, and modify the target definition of the W\_SALES\_CYCLE\_LINE\_F table.

3. In Source Analyzer, modify the table definition of the source table to include this new column.

Continuing with the example, you would include the VALIDATED\_ON\_DT column in the W\_SALES\_CYCLE\_LINE\_F source table.

4. Modify the PLP\_SalesCycleLinesFact\_Load and PLP\_SalesCycleLinesFact\_Load\_Full mappings to select the new column from any of the following source tables, and load it to the W\_SALES\_CYCLE\_LINE\_F target table:

- W\_SALES\_ORDER\_LINE\_F
- W\_SALES\_INVOICE\_LINE\_F
- W\_SALES\_PICK\_LINE\_F
- W\_SALES\_SCHEDULE\_LINE\_F

5. Modify The Source Qualifier SQL Override for the mapping, and map the column in the Transformation to map it to the target table.

#### 11.2.5.1.5 About Configuring the Backlog Period Date for Oracle Order Management and Fulfillment Analytics

The Backlog table (W\_SALES\_BACKLOG\_LINE\_F) stores backlog data for the current month. In contrast, the Backlog History table (W\_SALES\_BACKLOG\_LINE\_F) stores snapshots of all previous months' historical backlog data. The periods for which the Backlog History table tracks backlog data is defined by the Backlog Period Date. By default, the date is set as the last calendar day of the month; however you may configure this date. You may want to view backlog history at a more detailed level, such as by day or by week, instead of by month. The following example describes how historical backlog data is stored and what the implications are for changing the backlog time period.

#### 11.2.5.1.6 Example of How Backlog Data Is Stored in the Backlog History Table

Assume you represent a manufacturing company where financial backlog is defined as any item that is ordered, but not invoiced. On February 1, 2001, you received an order (Sales Order #1) for 30 products. 20 were shipped and invoiced and 10 were shipped, but not invoiced. At the end of the day, there is an entry in the Backlog table and in the Backlog History table. The entry in the Backlog History table looks like that shown in [Table 11-4](#).

**Table 11-4 Oracle 11i and Oracle R12: Backlog History Table Entry as of February 1, 2001**

SALES_ORDER_NUM(Sales Order Number)	BACKLOG_DK(Backlog Date)	BACKLOG_PERIOD_DK(Backlog Period Date)	OPEN_QTY(Backlog Quantity)
1	02/01/2001	02/28/2001	10

On February 2, 5 of the 10 financial backlog items are invoiced and, thus, removed from the backlog. Thus, there is an update to the existing row in the Backlog History table, as shown in [Table 11-5](#).

**Table 11-5 Oracle 11i and Oracle R12: Backlog History Table Entry as of February 2, 2001**

SALES_ORDER_NUM (Sales Order Number)	BACKLOG_DK (Backlog Date)	BACKLOG_PERIOD_DK (Backlog Period Date)	OPEN_QTY (Backlog Quantity)
1	02/01/2001	02/28/2001	Old value: 10 New value: 5

No further activity happens until March 1st. On March 1st, the remaining 5 items on financial backlog are invoiced and removed from financial backlog. In addition, a new sales order (Sales Order #2) comes in for 50 new items. All of the items are put on financial backlog.

Even though all items from Sales Order #1 are cleared from financial backlog, the last backlog row remains in the Backlog History table. The purpose in retaining the last row is to indicate that there was backlog for this particular order. The quantity, in this case 5 items, does not tell you how many items were initially on backlog, which was 10.

For the 50 new financial backlog items, there is a new entry into the Backlog History table. So, as of February 28, 2001, the Backlog History table looks like the [Table 11-6](#).

**Table 11-6 Oracle 11i: Backlog History Table Entry as of February 28, 2001**

SALES_ORDER_NUM (Sales Order Number)	BACKLOG_DK (Backlog Date)	BACKLOG_PERIOD_DK (Backlog Period Date)	OPEN_QTY (Backlog Quantity)
1	Old value: 02/01/2001  New value: 02/02/2001	02/28/2001	Old value: 10 New value: 5

On March 1, 30 more items are ordered (Sales Order #3), all of which are on financial backlog. The resulting Backlog History table looks like [Table 11-7](#).

**Table 11-7 Oracle 11i and Oracle R12: Backlog History Table Entry as of March 1, 2001**

SALES_ORDER_NUM (Sales Order Number)	BACKLOG_DK (Backlog Date)	BACKLOG_PERIOD_DK (Backlog Period Date)	OPEN_QTY (Backlog Quantity)
1	Old value: 02/01/2001  New value: 02/02/2001	02/28/2001	5
2	03/01/2001	03/31/2001	50
3	03/01/2001	03/31/2001	30

Because backlog history is maintained at the monthly level, you have a partial history of your backlogs. Based on the latest state of the Backlog History table shown in

**Table 11–7**, you can see that sales order number 1 ended up with 5 financial backlogged items. You do not have visibility into what the initial financial backlogged item quantities were for the sales orders; you only have their ending quantities.

If you decide that you want to track more details on how the items moved out of backlog, then you'll have to maintain the history at a more granular level. For instance, if you want to know the number of items that were on backlog when the it was first opened, you would have to track the backlog history by day, instead of by month.

For example, if you maintained backlog history at the daily level you would be able to capture that sales order 1 had an initial backlog of 10 as of February 1 and the backlog quantity shrank to 5 as of February 2. So, by capturing history at the daily level, you could then compute cycle times on how long it took to move items out of backlog. However, if you decide to capture backlog history at a more detailed level, you may compromise performance because tracking backlog history at the daily level can increase the size of the Backlog History table exponentially.

If you choose to change the time period for which historical backlog data is kept, you must verify that all types of backlog are being stored at the same grain; which requires modification to multiple mappings. **Table 11–8** provides a list of all applicable mappings and their corresponding Expression transformations that you must modify.

**Table 11–8 Oracle 11i and Oracle R12: Backlog History Applicable Mappings and Expression Transformations**

Mapping	Expression Transformation
PLP_SalesBacklogLinesfact_LoadOrderLines	EXP_SALES_ORNLNS_BACKLOG
PLP_SalesBacklogLinesfact_LoadScheduleLines	EXP_SALES_SCHLNS_BACKLOG

The backlog history period is monthly by default. The default SQL statement in the Expression transformation for the port BACKLOG\_PERIOD\_DK is:

```
TO_DECIMAL(TO_CHAR(LAST_DAY(CALENDAR_DATE), 'YYYYMMDD'))
```

You can edit the backlog period date so that you can capture a more detailed backlog history with the following procedure. Possible periods include daily (CAL\_DAY\_DT), weekly (CAL\_WEEK\_DT), monthly (CAL\_MONTH\_DT), and quarterly (CAL\_QTR\_DT).

**11.2.5.1.7 About Configuring the Customer Status History Fact for Post-Load Processing In Oracle Order Management and Fulfillment Analytics** In Oracle Order Management and Fulfillment Analytics, W\_CUSTOMER\_STATUS\_HIST\_F is a fact table that tracks the status of customers based on the frequency of orders they place with the organization. Possible statuses are NEW, RECENT, DORMANT and LOST. The time duration for each status bucket is configurable, out of the box being a calendar year. The grain of this table is at a Customer, Customer Status and the Status Start Date level. This section explains the possible configurations available for this table, what they mean and how to implement them.

**11.2.5.1.8 Configuring the Customer Status History Fact table** This section talks about the following configurations that are available for the Customer Status History Fact table:

- Configure the Data Warehouse Identifier
- Configure the Period for each status bucket

**Configuring the Data Warehouse Identifier**

This table uses some of the Oracle Business Intelligence Applications defined statuses, like NEW, RECENT, DORMANT and LOST. These status data gets loaded into the Data Warehouse directly through an out of box pre-packaged CSV file. The data in the file is independent of any specific OLTP source systems where your Customer or Sales data resides. In order to differentiate between source-based statuses from the pre-packaged out of box Data Warehouse statuses, a definite identifier is required. The Informatica mapping parameter \$\$WH\_DATASOURCE\_NUM\_ID serves that purpose.

A pre-packaged value equal to 999 is set out of the box. Ideally you would not need to configure this value unless you have chosen to use this number (999) for a specific data source of yours, like Oracle EBS 11.5.10, etc.

For information about how to configure the \$\$WH\_DATASOURCE\_NUM\_ID value, see: [Section 11.2.5.1.9, "How to Configure the Customer Status History Fact table"](#).

### **Configuring the Period for each status bucket**

When a customer orders some products/services from your organization for the first time, Oracle Business Intelligence Applications sets the status for the customer as NEW. The customer maintains the same status if he/she shows a constant order pattern, as long as the duration between any two of his/her orders is less than a configurable/business defined period. The value (out of box being 365 days) of this Informatica parameter \$\$PERIOD is configurable. An use case for that would be a Fast Moving / Retail Goods company many define 30 days as their choice of period, whereas a Slow Moving company may be even happy with 730 days as period.

In case the customer is seen to have not ordered anything for more than one period, he/she is moved to the next status, RECENT. Similarly, no orders for one more period since he/she became RECENT would make him/her DORMANT. And lastly, he/she is set to LOST if no orders were seen for more than one more period after he/she became DORMANT.

However, if a customer orders while in DORMANT status, for instance, Oracle Business Intelligence Applications would upgrade his/her status back to RECENT. If the customer were in LOST status, and he/she orders, then he/she will be upgraded back to RECENT.

All these examples above illustrate how important it is for the business to set the right value for the period. Organizations would tend to launch different campaigns targeting to different customers based on their current status, or order-patterns, putting it in a different way.

For information about how to configure the \$\$PERIOD value, see: [Section 11.2.5.1.9, "How to Configure the Customer Status History Fact table"](#).

**11.2.5.1.9 How to Configure the Customer Status History Fact table** This section explains the steps involved in configuring the Customer Status History Fact table using the \$\$WH\_DATASOURCE\_NUM\_ID and \$\$PERIOD variables (for more information about these variables, see [Section 11.2.5.1.8, "Configuring the Customer Status History Fact table"](#)).

To modify the value of \$\$WH\_DATASOURCE\_NUM\_ID:

1. Log on to the DAC repository and display the Design tab.
2. Navigate to the appropriate container for your OLTP source system.
3. Display the Source System Parameters tab and locate the parameter \$\$WH\_DATASOURCE\_NUM\_ID.
4. In the Edit pane, use the **Value** field to specify a value.

5. Save the changes.

To modify the value of \$\$PERIOD:

1. Log on to the DAC repository and display the Design tab.
2. Navigate to the appropriate container for your OLTP source system.
3. Click on the Tasks tab and query for the following two tasks:
  - PLP\_CustomerStatusHistoryFact\_New\_Customers\_Load
  - PLP\_CustomerStatusHistoryFact\_Status\_Revalidate
4. For each task, display the Parameters tab, and use the **Value** field to specify a value.

Make sure you the set same value for both tasks.

5. Save the changes.

### 11.2.5.2 Configuration Steps for Oracle Order Management and Fulfillment Analytics For Oracle EBS

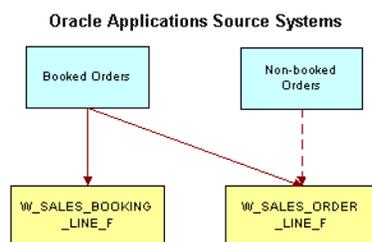
This section contains configuration steps that apply to Oracle EBS.

**11.2.5.2.1 How to Configure Sales Order Lines Data Storage** Sales order lines are the itemized lines that make up a sales order. This information is stored in the W\_SALES\_ORDER\_LINE\_F table. This topic describes how to modify the type of information stored in this table.

**11.2.5.2.2 About the Handling of Booked and Nonbooked Orders in the Order Lines and Bookings Table** By default, only booked orders are extracted from the Oracle source system as shown in Figure 11-1. Therefore, all orders loaded into the Sales Order Lines and Bookings tables are booked.

However, if you want to load nonbooked orders into the Sales Order Lines table, you have to configure the extract so that it does not filter out nonbooked orders. In Oracle 11i and Oracle R12, the OE\_ORDER\_LINES\_ALL.BOOKED\_FLAG = Y condition indicates that an order is booked; therefore, this statement is used to filter out nonbooked orders. To load all orders, including nonbooked orders, remove the filter condition from the WHERE clause in the SDE\_ORA\_SalesOrderLinesFact and SDE\_ORA\_SalesOrderLinesFact\_Primary mappings.

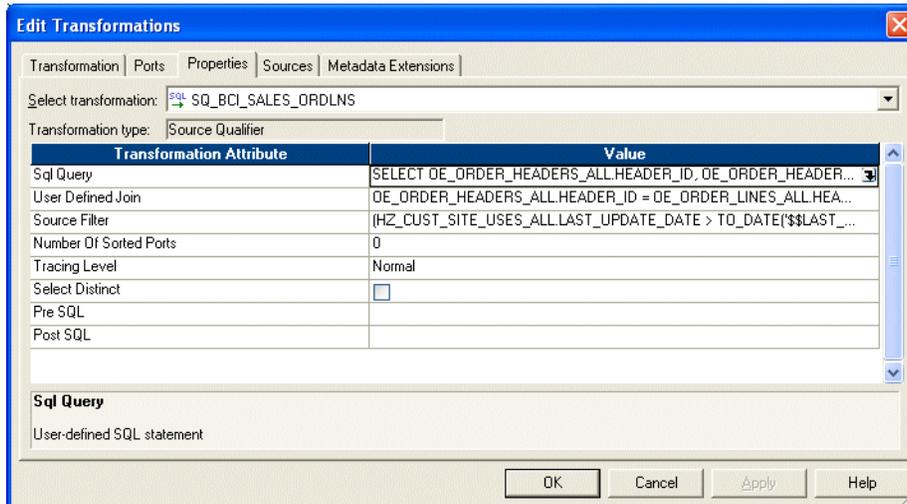
**Figure 11-1 Handling Booked and Nonbooked Orders**



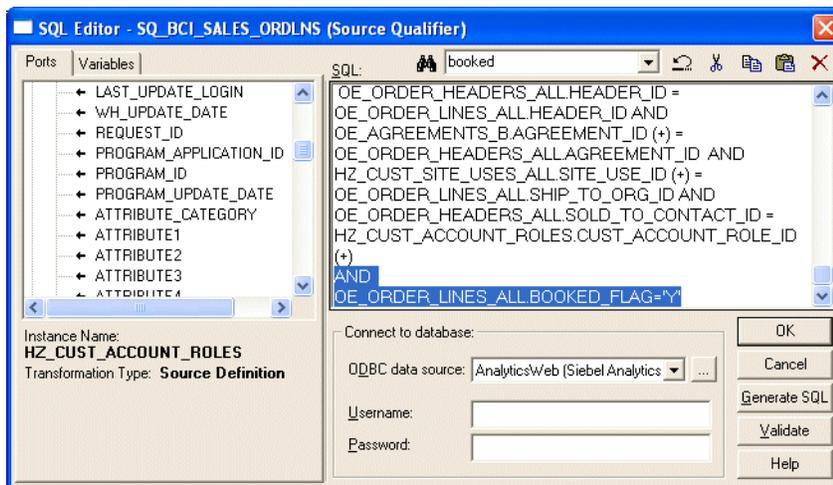
By default, only booked orders are loaded into the Sales Order Lines (W\_SALES\_ORDER\_LINES\_F) and Sales Booking Lines (W\_SALES\_BOOKING\_LINE\_F) tables. However, you can also load non-booked orders in Sales Order Lines (W\_SALES\_ORDERS\_LINES\_F).

**To include nonbooked orders in the Sales Order Lines tables**

1. In PowerCenter Designer, open the SDE\_ORA115<ver>\_adapter or SDE\_ORAR12\_ adapter folder.
2. Open the `mplt_BC_ORA_SalesOrderFact` maplet in the Maplet Designer.
3. Double-click the `SQ_BCI_SALES_ORDLNS` source qualifier to open the Edit Transformations box.
4. Display the Properties tab.



5. For both the Sql Query Transformation Attribute and the User Defined Join Transformation Attribute, do the following:
  - a. Select the down arrow in the Value field to display the SQL Editor box.



- b. In the SQL box, remove the line 'AND OE\_ORDER\_LINES\_ALL.BOOKED\_FLAG='Y''.
  - c. Select OK to save the changes.
6. Validate and save your changes to the repository.
7. Repeat steps 3 - 5 for the `SDE_ORA_SalesOrderLinesFact_Primary` mapping.

**11.2.5.2.3 How to Configure Sales Schedule Lines Data Storage** Sales schedule lines detail when each order's items are slated for shipment. Each sales order is broken into sales order lines, and each sales order line can have multiple schedule lines.

For example, you might not have enough stock to fulfill a particular sales order line, therefore you create two schedules to fulfill it. One schedule ships what you currently have in stock, and the other schedule includes enough time for you to manufacture and ship the remaining items of the sales order line. This information is stored in the `W_SALES_SCHEDULE_LINE_F` table. This topic describes how to modify the type of information stored in this table.

**11.2.5.2.4 About the Handling of Booked and Nonbooked Orders in the Sales Schedule Lines Table** By default, all orders loaded into the Sales Schedule Lines are booked.

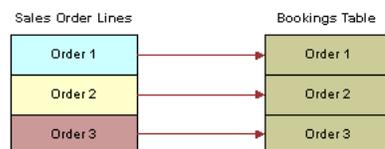
However, if you want to load nonbooked orders into the Sales Schedule Lines table, you have to configure the extract so that it does not filter out nonbooked orders. In Oracle 11i and Oracle R12, the `OE_ORDER_LINES_ALL.BOOKED_FLAG = Y` condition indicates that an order is booked; therefore, this statement is used to filter out nonbooked orders. To load all orders, including nonbooked orders, remove the filter condition from the WHERE clause in the `SDE_ORA_SalesScheduleLinesFact` and `SDE_ORA_SalesScheduleLineLines_Fact_Primary` mappings.

**To include nonbooked orders in the Sales Schedule Lines tables**

1. In PowerCenter Designer, open the `SDE_ORA115<ver>_adapter` or `SDE_ORAR12_adapter` folder.
2. Open the `mp1t_BC_ORA_SalesScheduleLinesFact` maplet in the Maplet Designer.
3. Double-click the `SQ_BCI_SALES_ORDLNS` source qualifier to open the Edit Transformations box.
4. Display the Properties tab.
5. For both the Sql Query Transformation Attribute and the User Defined Join Transformation Attribute, do the following:
  - a. Select the down arrow in the Value field to display the SQL Editor box.
  - b. In the SQL box, remove the line `'AND OE_ORDER_LINES_ALL.BOOKED_FLAG='Y'`.
  - c. Select OK to save the changes.
6. Validate and save your changes to the repository.
7. Repeat steps 3 - 5 for the `SDE_ORA_SalesScheduleLinesFact_Primary` mapping.

**11.2.5.2.5 About Loading Bookings at the Schedule Line Level** As initially configured for Oracle 11i and Oracle R12, bookings are recorded at the Sales Order Line level. For each booked order, there is at least one row in the Bookings table, as shown in the figure below.

**Figure 11–2 Sales Order Lines and Bookings Table**



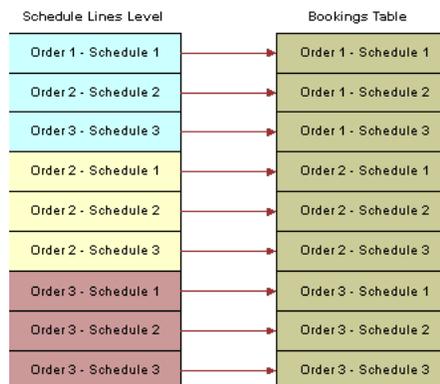
There are two subject areas in the SDE\_ORA115<ver>\_adapter or SDE\_ORAR12\_adapter containers:

- Enterprise Sales - Booking Lines & Order Lines
- Enterprise Sales - Booking Lines & Schedule Lines

The execution plans that are installed with Oracle BI Applications use the Enterprise Sales - Booking Lines & Order Lines subject area by default. If you want to load booking lines at the schedule line level, create a new execution plan and include the Enterprise Sales - Booking Lines & Schedule Lines subject area instead of Enterprise Sales - BookingLines & Order Lines.

Bookings may be recorded at the Sales Schedule Line level instead of the Sales Order Line level. At the Sales Schedule Line level, bookings provide a more granular view, as the orders are segmented by schedule line. Bookings recorded at the Schedule Line level provide one row in the Bookings table for each schedule line, as shown in the figure below. Oracle Applications schedule lines have the same granularity as order lines. Therefore, if you pull booking lines from schedule lines, the booking lines are limited to scheduled order lines.

**Figure 11–3 Bookings recorded at the Schedule Line level**

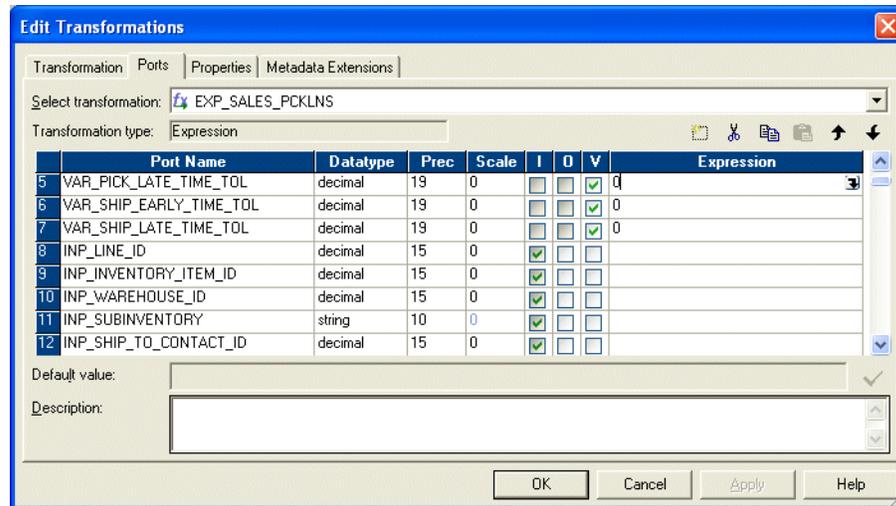


**11.2.5.2.6 How to Configure Early and Late Tolerances for Shipments** You configure the definition of early and late shipments by editing the EXP\_SALES\_PCKLNS expression in the mplt\_SA\_ORA\_SalesPickLinesFact mapplet. The mplt\_SA\_ORA\_SalesPickLinesFact mapplet is used by the SDE\_ORASalesPickLinesFact mapping.

This mapplet compares the pick date and ship date with the scheduled ship date to determine whether or not orders are late.

**To configure early and late tolerances for shipments**

1. In PowerCenter Designer, open the SDE\_ORA115<ver>\_adapter or SDE\_ORAR12\_adapter folder.
2. Open the mplt\_SA\_ORA\_SalesPickLinesFact mapplet in the Mapplet Designer.
3. Double-click the EXP\_SALES\_PCKLNS expression to open the Edit Transformation box.
4. Display the Ports tab.



5. Edit the Expression for the ports that you want to modify.

For example:

- If you want to allow two days after the scheduled pick date before you flag the pick as late, set the Expression value for the VAR\_PICK\_LATE\_TIME\_TOL port to 2.
- To set the number of days before a pick is flagged as early, set the Expression value the VAR\_PICK\_EARLY\_TIME\_TOL port.
- To set the number of days before a pick is flagged as late, set the Expression value for the VAR\_PICK\_LATE\_TIME\_TOL port.
- If you want to change the shipping tolerances, set the Expression values for the shipping ports (for example, VAR\_SHIP\_LATE\_TIME\_TOL, VAR\_SHIP\_EARLY\_TIME\_TOL).

6. Validate and save your changes to the repository.

**11.2.5.2.7 How to Configure Sales Invoice Lines Data Storage** Sales invoice lines are payments for items ordered by a customer. This information is stored in the W\_SALES\_INVOICE\_LINE\_F table. This topic describes how to modify the type of information stored in this table.

**11.2.5.2.8 How to Configure the Sales Invoice Extract** By default, the Oracle Order Management and Fulfillment Analytics application is configured to extract completed sales invoices when performing the Sales Invoice data extract. Oracle 11i and Oracle R12 use a flag to indicate whether a sales invoice is complete. In particular, completed sales invoices are those where the RA\_CUSTOMER\_TRX\_ALL.COMPLETE\_FLAG = Y in Oracle 11i and Oracle R12.

To extract incomplete sales invoices, as well as complete invoices, remove the extract filter statement.

**To remove the extract filter for sales invoices**

1. In PowerCenter Designer, open the SDE\_ORA115<ver>\_adapter or SDE\_ORAR12\_adapter folder.
2. Open the mplt\_BC\_ORA\_SalesInvoiceLinesFact mapplet in the Mapplet Designer.

3. Double-click the `SQ_BCI_SALES_IVCLNS` source qualifier to open the Edit Transformations box.
4. Display the Properties tab.
5. For the SQL Query Transformation Attribute, select the down arrow in the Value field to display the SQL Editor box.
6. In the SQL box, remove the line 'AND RA\_CUSTOMER\_TRX\_ALL.COMPLETE\_FLAG='Y''.
7. Validate and save your changes to the repository.
8. Repeat steps 2 - 7 for the `mp1t_BC_ORA_SalesInvoiceLinesFact_Primary` maplet.

**11.2.5.2.9 How to Configure the Order Lines Execution Plan and OTB ETL** The Backlogs and Cycle Lines (OTB ETL) component assumes that the Backlog, Picking and Cycle Lines tables have been updated with shipping and invoicing information (for example, using Oracle EBS Interface programs). If the Oracle Order Line Tables have not been updated with shipping and invoicing information, you need to update the OTB ETL and Execution Plan as follows:

**To configure the Order Lines Execution Plan and OTB ETL**

1. In PowerCenter Designer, open the PLP folder.
2. Open the `PLP_SalesCycleLinesFact` maplet in the Maplet Designer.
3. Double-click the `SQ_W_SALES_ORDER_LINE_F` source qualifier to open the Edit Transformations box.
4. Display the Properties tab.
5. For the SQL Query Transformation Attribute, select the down arrow in the Value field to display the SQL Editor box.
6. In the SQL box, replace the SQL text 'X.TOTAL\_SHIPPED\_QTY' with 'PICKLINE.TOTAL\_SHIPPED\_QTY'.
7. In the SQL box, replace the SQL text 'X.TOTAL\_INVOICED\_QTY' with 'IVCLINE.TOTAL\_INVOICE\_QTY'.
8. Validate and save your changes to the repository.
9. Log into the DAC (for more information, see [Section 4.10.2, "How to Log Into the DAC"](#)).
10. In the Design tab, expand the Configuration Points node, and select the Sales PLP Optional Tasks.
11. Activate the appropriate Subject Areas.
12. Open the Source Qualifier in `SDE_ORA_SalesPickLinesFact`.
13. Modify the SQL Query as follows
  - a. Add to the following join condition:
 

```
AND WSH_DELIVERY_DETAILS.DELIVERY_DETAIL_ID=WSH_DELIVERY_ASSIGNMENTS.DELIVERY_DETAIL_ID (+) AND WSH_DELIVERY_ASSIGNMENTS.DELIVERY_ID=WSH_NEW_DELIVERIES.DELIVERY_ID (+)
```
  - b. Nest the following filter condition within the parenthesis:
 

```
OR WSH_NEW_DELIVERIES.LAST_UPDATE_DATE > TO_DATE( '$$LAST_EXTRACT_DATE' , 'MM/DD/YYYY HH24:MI:SS' )
```

- c. Change `select OE_ORDER_LINES_ALL.ACTUAL_SHIPMENT_DATE` to `select WSH_NEW_DELIVERIES.INTIAL_PICKUP_DATE`.
- d. Select `WSH_NEW_DELIVERIES.LAST_UPDATE_DATE` and link it to `EXP_SALES_PCKLNS.LAST_UPDATE_DATE1`.

#### 11.2.5.2.10 How to Configure Supply Chain Analytics for Oracle Order Management and Fulfillment Analytics

The Oracle Order Management and Fulfillment Analytics application uses tables that are also used in Oracle's Supply Chain Analytics family of products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics).

For Oracle 11i and Oracle R12, you need to use the following configuration steps for Supply Chain Analytics to configure Oracle Order Management and Fulfillment Analytics:

- [Section 9.2.2.3, "How to Configure Quantity Types for Product Transactions"](#)
- [Section 8.2.2.1, "Configuring the Country Region and State Region Name"](#)
- [Section 8.2.2.2, "Configuring the State Name"](#)
- [Section 8.2.2.3, "Configuring the Country Name"](#)
- [Section 8.2.2.4, "Configuring the Configuring the Make-Buy Indicator"](#)

#### 11.2.5.2.11 How to Configure Oracle Financial Analytics for Oracle Order Management and Fulfillment Analytics

The Oracle Order Management and Fulfillment Analytics application uses tables that are also used in the Oracle Financial Analytics application.

For Oracle 11i and Oracle R12, you need to use the following configuration steps for Oracle Financial Analytics to configure Oracle Order Management and Fulfillment Analytics:

- [Section 10.2.5.2.1, "How to Configure How GL Balances Is Populated in Oracle EBS"](#)
- [Section 10.2.2.2, "How to Map Oracle General Ledger Natural Accounts to Group Account Numbers"](#)
- [Section 10.2.5.2.2, "How to Configure Oracle Profitability Analytics Transaction Extracts"](#)

#### 11.2.5.2.12 About Tracking Attribute Changes in Bookings

Changes in booked orders are tracked in the Booking Lines table (`W_SALES_BOOKING_LINE_F`), not in the Sales Order Lines table (`W_SALES_ORDER_LINE`). By default, the only changes tracked in the `W_SALES_BOOKING_LINE_F` table are changes in the ordered amount, ordered quantity, or Booking ID. By default, the Booking ID is defined as:

```
TO_CHAR(INP_LINE_ID) || '~' || TO_CHAR(INP_INV_ITEM_ID) || '~' || TO_CHAR(INP_WAREHOUSE_ID)
```

Any changes in these fields results in another row in the `W_SALES_BOOKING_LINE_F` table. However, changes in any other fields does not result in a new row; instead, the existing information are overwritten with the changed information. No history is kept for changes to these other field values. If you want to track other changes you can do so. For example, you may want to track changes to the sales representative who is handling the order. The ETL processes are prepackaged to overwrite sales representative changes; however, if you want to retain them, you must add the attribute to the Booking ID definition in the Booking ID expression in the Source Adapter mapplet (`mplt_SA_ORA_SalesOrderLinesFact`). The following section

describes what happens if you modify the Booking ID to include the sales representative.

**11.2.5.2.13 About Viewing the Data Warehouse Changes by Salesperson ID** Assume you want to track changes to the sales representative for bookings and debookings. You decide to do this to better evaluate each representative's sales performance. To track changes by Salesperson ID, you have to modify the VAR\_BOOKING\_ID to use the value:

```
TO_CHAR(INP_LINE_ID) || '~' || TO_CHAR(INP_INV_ITEM_ID) || '~' || to_char(INP_WAREHOUSE_ID)
```

For example, to edit the VAR\_BOOKING\_ID value, do the following:

1. In Informatica Designer, open the mplt\_SA\_ORA\_SalesOrderLinesFact mapplet in the Mapplet Designer.
2. Double click the MAPI\_SALES\_ORDLNS transformation to open the Edit Transformation box.
3. Display the Ports tab.
4. Select the EXP\_SALES\_ORDLNS transformation.
5. Edit the expression for the VAR\_BOOKING\_ID port.

The following paragraphs and tables describe what happens in the source system and the W\_SALES\_BOOKING\_LINE\_F table when you change sales representatives under this scenario.

Day 1: One order is placed with Salesperson 1001. The source system displays the information as shown in [Table 11–9](#).

**Table 11–9 Oracle 11i and Oracle R12: Source System Table Row After Day One Activity**

Sales Order Number	Sales Order Line Number	Salesperson ID	Quantity	Selling Price	Date
1	1	1001	100	25	1-June-2000

The row in [Table 11–9](#) is entered into the IA Bookings table (W\_SALES\_BOOKING\_LINE\_F) as shown in [Table 11–10](#).

**Table 11–10 Oracle 11i and Oracle R12: W\_SALES\_BOOKING\_LINE\_F Table Row After Day One Activity**

SALES_ORDER_NUM	SALES_ORDER_ITEM	SALESREP_ID	SALES_QTY	NET_DOC_AMT	BOOKED_ON_DT
1	1	1001	100	2500	1-June-2000

Day 2: Salesperson 1002 takes over this order, replacing Salesperson 1001. Thus, the salesperson associated with the order is changed from 1001 to 1002 in the source system. The row in the source system looks like the row shown in [Table 11–11](#).

**Table 11–11 Oracle 11i and Oracle R12: Source System Table Row After Day Two Activity**

Sales Order Number	Sales Order Line Number	Salesperson ID	Quantity	Selling Price	Date
1	1	1002	100	25	2-June-2000

The `SIL_SalesBookingLinesFact_Load_OrderLine_Credit`, which also writes to the booking table, now does a debooking for the old line and `SIL_SalesBookingLinesFact_Load_OrderLine_Debt` inserts a new row into the `W_SALES_BOOKING_LINE_F` booking table. On day two, the row in the `W_SALES_BOOKING_LINE_F` table looks like the row shown in the [Table 11–12](#).

**Table 11–12 Oracle 11i and Oracle R12: W\_SALES\_BOOKING\_LINE\_F Table Row After Day Two Activity**

SALES_ORDER_NUM	SALES_ORDER_ITEM	SALESREP_ID	SALES_QTY	NET_DOC_AMT	BOOKED_ON_DT
1	1	1001	100	2500	1-June-2000
1	1	1001	-100	-2500	2-June-2000
1	1	1002	100	2500	2-June-2000

**11.2.5.2.14 How to Configure Different Types of Backlog Calculations** Backlog information is stored in the `W_SALES_BACKLOG_LINE_F` and `W_SALES_BACKLOG_HISTORY_F` tables. This topic describes how to modify the type of information stored in these tables. Many types of backlog exist in the Oracle Order Management and Fulfillment Analytics application—financial backlog, operational backlog, delinquent backlog, scheduled backlog, unscheduled backlog, and blocked backlog. Each type of backlog is defined by two particular dates in the sales process; therefore, calculations of backlog hits multiple fact tables.

For example, financial backlog records which items have been ordered but payment has not been received. Thus, to calculate the number of financial backlog items, you use the Sales Order Lines table (to determine which items have been ordered) and the Sales Invoice Lines table (to see which orders have been paid for). Using these two tables, you can determine the number of items and the value of those items that are on financial backlog.

**11.2.5.2.15 Adding Closed Orders to Backlog Calculations** By default, the Oracle Order Management and Fulfillment Analytics application only extracts open sales orders from the Sales Order Lines (`W_SALES_ORDER_LINE_F`) table and Sales Schedule Lines table (`W_SALES_SCHEDULE_LINE_F`) for backlog calculations to populate the Backlog tables. *Open sales orders* are defined as orders that are not canceled or not complete. The purpose in extracting only open orders is that in most organizations those orders that are closed are no longer a part of backlog. However, if you want to extract sales orders that are marked as closed, you may remove the default filter condition from the extract mapping.

For example, assume your customer orders ten items. Six items are invoiced and shipped, but four items are placed on operational and financial backlog. This backlog status continues until one of two things happens:

- The items are eventually shipped and invoiced.
- The remainder of the order is canceled.

If you choose to extract sales orders that are flagged as closed, you must remove the condition in the Backlog flag. To do so, use the following procedure.

The `BACKLOG_FLAG` in the `W_SALES_ORDER_LINE_F` table is also used to identify which sales orders are eligible for backlog calculations. By default, all sales order types have their Backlog flag set to Y. As a result, all sales orders are included in backlog calculations.

**To remove open order extract filters**

1. In PowerCenter Designer, open the SDE\_ORA115<ver>\_adapter or SDE\_ORAR12\_adapter folder.
2. Open the `mplt_BC_ORA_SalesOrderLinesFact` maplet in the Maplet Designer.
3. Double click on the EXP\_SALES\_ORDLNS, and display the Ports tab.
4. Edit the VAR\_OPR\_BACKLOG\_FLG and remove the OPEN\_FLAG='Y'.
5. Edit the VAR\_FIN\_BACKLOG\_FLG and remove the OPEN\_FLAG='Y'.
6. Open the `mplt_BC_ORA_SalesScheduleLinesFact` maplet in the Maplet Designer.
7. Double click on the EXP\_SALES\_SCHLNS, and display the Ports tab.
8. Edit the VAR\_OPR\_BACKLOG\_FLG and remove the OPEN\_FLAG='Y'.
9. Edit the VAR\_FIN\_BACKLOG\_FLG and remove the OPEN\_FLAG='Y'.
10. Validate and save your changes to the repository.
11. Open the PLP folder.
12. Open the mappings `PLP_SalesBacklogLinesFact_LoadOrderLines` and `PLP_SalesBacklogLinesFact_LoadScheduleLines`.
13. Remove the condition `W_STATUS_CODE <> 'Closed'` from the SQL Query in the Source Qualifier.

**11.2.5.3 Configuration Steps for Oracle Order Management and Fulfillment Analytics For PeopleSoft**

This section contains configuration steps that apply to PeopleSoft.

Not applicable to Oracle BI Applications Version 7.9.4.

**11.2.5.4 Configuration Steps for Oracle Order Management and Fulfillment Analytics For Universal**

This section contains configuration steps that apply to Universal.

Not applicable to Oracle BI Applications Version 7.9.4.

---

---

## Configuring Oracle HR Analytics

This chapter describes how to configure Oracle Human Resources Operations and Compliance Analytics and Oracle Human Resources Compensation Analytics for particular sources to meet your business needs.

It contains the following topics:

- [Section 12.1, "Overview of Oracle HR Analytics"](#)
- [Section 12.2, "Configuration Required Before A Full Load for Oracle HR Analytics"](#)

### 12.1 Overview of Oracle HR Analytics

Oracle HR Analytics contains information for HR operations, employee events, and payroll.

The Oracle HR Analytics application has the following functional areas:

- **Compensation.** HR Compensation allows you to analyze the salaries, benefits, and rewards that comprise your employee compensation plan. The metrics provided as part of the application allow you to measure several areas of performance and perform a variety of comparative analyses at various levels of granularity.  
  
It provides your company with employee payroll information that can be vital to success in today's economy. Over or under-compensating employees can both have serious effects on your company's ability to maintain a competitive edge. The HR Compensation area provides the information your HR Management department needs to manage compensation costs, such as identifying emerging trends within the organization, or within specific areas of compensation, and evaluating the effectiveness of the level of compensation as an incentive.
- **Human Resource Performance.** The information stored in the Human Resource Performance area allows you to measure several areas of performance, including contribution and productivity, HR effectiveness, and trends analytics.
- **Retention.** Under the Retention functional area you can find the events that are the hallmarks of employees' professional life cycle. These events include their hiring information, their promotional opportunities realized and not realized, the quality of the employees' job performance as measured by performance ranking, their length of service, and the reasons for termination, both voluntary and involuntary. Monitoring retention rates within departments is useful in determining potential problem areas that may want to be addressed by senior management.
- **U.S. Statutory Compliance.** The U.S. Statutory Compliance functional area stores information that helps Human Resources departments prepare government-required reports.

- **HR Profile.** The HR Profile functional area provides you with the tools to separate sensitive from non-sensitive information, and to restrict access to sensitive data. Sensitive information includes such data as ethnicity, age, native language, marital status, and performance ratings. Non-sensitive information includes information such as job title, work location, and position status.

## 12.2 Configuration Required Before A Full Load for Oracle HR Analytics

This section contains configuration steps that you need to perform on Oracle HR Analytics before you do a full data load, and contains the following topics:

- [Section 12.2.1, "Configuration Steps for Oracle HR Analytics For All Source Systems"](#)
- [Section 12.2.2, "Configuration Steps for Oracle HR Analytics For Oracle EBS"](#)
- [Section 12.2.3, "Configuration Steps for Oracle HR Analytics For PeopleSoft"](#)
- [Section 12.2.4, "Configuration Steps for Oracle HR Analytics For Universal"](#)
- [Section 12.2.5, "Configuration Steps for Controlling Your Data Set"](#)

### 12.2.1 Configuration Steps for Oracle HR Analytics For All Source Systems

This section contains configuration steps required before you do a full data load that apply to all source systems.

---

---

**Note:** For configuration steps that apply to all analytics modules (for example, Oracle Financial Analytics, Oracle HR Analytics, Oracle Sales Analytics, see [Chapter 8, "Configuring Common Areas and Dimensions"](#)).

---

---

### 12.2.2 Configuration Steps for Oracle HR Analytics For Oracle EBS

This section contains configuration steps required before you do a full data load that apply to Oracle EBS, and contains the following topics:

- [Section 12.2.2.1, "About Domain Values and CSV Worksheet Files for Oracle HR Analytics"](#)
- [Section 12.2.2.2, "How to Configure the domainValues\\_Employee\\_Ethnic\\_Group\\_Code\\_ora11i.csv"](#)
- [Section 12.2.2.3, "How to Configure the domainValues\\_Employee\\_Sex\\_MF\\_ora11i.csv"](#)
- [Section 12.2.2.4, "How to Configure the domainValues\\_Employee\\_Veteran\\_Status\\_Code\\_ora11i.csv"](#)
- [Section 12.2.2.5, "How to Configure the domainValues\\_Employment\\_Cat\\_ora11i.csv"](#)
- [Section 12.2.2.6, "How to Configure the domainValues\\_Employment\\_Exempt\\_Flg\\_ora11i.csv"](#)
- [Section 12.2.2.7, "How to Configure the domainValues\\_Employment\\_Full\\_Time\\_Flg\\_ora11i.csv"](#)
- [Section 12.2.2.8, "How to Configure the domainValues\\_Employment\\_Status\\_ora11i.csv"](#)

- Section 12.2.2.9, "How to Configure the domainValues\_EventTypes\_ora11i.csv"
- Section 12.2.2.10, "How to Configure the domainValues\_HRPosition\_Active\_Pos\_Flg\_ora11i.csv"
- Section 12.2.2.11, "How to Configure the domainValues\_HRPosition\_Exempt\_Flg\_ora11i.csv"
- Section 12.2.2.12, "How to Configure the domainValues\_Job\_Eeo\_Cat\_Code\_ora11i.csv"
- Section 12.2.2.13, "How to Configure the domainValues\_Job\_Flsa\_Stat\_Code\_ora11i.csv"
- Section 12.2.2.14, "How to Configure the domainValues\_Pay\_Type\_Grp\_Code\_ora11i.csv"
- Section 12.2.2.15, "How to Configure the domainValues\_Pay\_Type\_Flg\_ora11i.csv"
- Section 12.2.2.16, "How to Configure Address Types for HR Profile"
- Section 12.2.2.17, "How to Configure Phone Types for HR Profile"
- Section 12.2.2.18, "How to Configure Education Degree Codes for Employee Dimension"
- Section 12.2.2.19, "About Configuring Key Flexfields"
- Section 12.2.2.20, "How to Configure the Key Flexfields for the Job Dimension"
- Section 12.2.2.21, "How to Configure the Key Flexfields for the HR Position Dimension"
- Section 12.2.2.22, "How to Configure the Key Flexfields for the Pay Grade Dimension"
- Section 12.2.2.23, "How to Configure multi-segmented Flexfields"
- Section 12.2.2.24, "How to Configure Flags for the Pay Type Dimension"
- Section 12.2.2.25, "How to Configure Classification Names for Payroll"

### 12.2.2.1 About Domain Values and CSV Worksheet Files for Oracle HR Analytics

You configure the CSV files in Oracle HR Analytics by mapping domain values to columns. The CSV worksheet files for Oracle HR Analytics are in the `$pmsserver\LkpFiles` folder.

For more information on configuring domain values with CSV worksheet files, see [Section 6.12, "About Domain Values"](#) and [Section 6.13, "Configuring the Domain Value Set with CSV Worksheet Files"](#).

For more information on domain values, see *Oracle Business Analytics Warehouse Data Model Reference*.

---



---

**Note:** Incorrect mappings result in inaccurate calculations of Oracle Business Intelligence metrics.

---



---

[Table 12-1](#) lists the CSV worksheet files and the domain values for Oracle HR Analytics in the `$pmsserver\LkpFiles` folder.

**Table 12–1 Domain Values and CSV Worksheet Files for Oracle HR Analytics**

<b>Worksheet File Name</b>	<b>Domain Value Table - Column</b>	<b>Description</b>	<b>Session</b>
domainValues_Employee_Ethnic_Group_Code_ora11i.csv	W_EMPLOYEE_D.W_ETHNIC_GRP_CODE	Lists the Ethnic codes and their corresponding domain values of "Ethnic Group Code" for the Oracle 11i Application.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employee_Sex_MF_ora11i.csv	W_EMPLOYEE_D.W_SEX_MF_CODE	Lists the Sex codes and their corresponding domain values of "Sex Code" for the Oracle 11i Application.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employee_Veteran_Status_Code_ora11i.csv	W_EMPLOYEE_D.W_VETERAN_STATUS_CODE	Lists the Veteran codes and their corresponding domain values of "Veteran Status Code" for the Oracle 11i Application.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employment_Cat_ora11i.csv	W_EMPLOYMENT_D.W_EMPLOYEE_CAT_CODE	Lists the User Person Types and their corresponding domain values of "Employment Category Code" for the Oracle 11i Application.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full
domainValues_Employment_Exempt_Flg_ora11i.csv	W_EMPLOYMENT_D.W_EXEMPT_FLG	Lists the FLSA Statuses and their corresponding domain values of "Exempt Flag" for the Oracle 11i Application.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full
domainValues_Employment_Full_Time_Flg_ora11i.csv	W_EMPLOYMENT_D.W_FULL_TIME_FLG	Lists the Employment Categories and their corresponding domain values of "Full Time Flag" for the Oracle 11i Application.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full
domainValues_Employment_Status_ora11i.csv	W_EMPLOYMENT_D.W_EMPLOYMENT_STAT_CODE	Lists the Per System Statuses and their corresponding domain values of "Employment Status" for the Oracle 11i Application.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full

**Table 12–1 (Cont.) Domain Values and CSV Worksheet Files for Oracle HR Analytics**

<b>Worksheet File Name</b>	<b>Domain Value Table - Column</b>	<b>Description</b>	<b>Session</b>
domainValues_EventTypes_ora11i.csv	W_EVENT_TYPE_ D.W_EVENT_CLASS, W_EVENT_TYPE_ D.W_EVENT_GRP_ CODE, W_EVENT_ TYPE_D.W_EVENT_ REASON_CODE, W_ EVENT_SUBG_CODE	Lists the Event Types, Event Codes and Meanings and their corresponding domain values of "Event Group", "Event Sub-Group" and "Event Reason" for the Oracle 11i Application.	SDE_ORA_ EventTypeDimension _AbsenceAttendance, SDE_ORA_ EventTypeDimension _AbsenceAttendance_ Full, SDE_ORA_ EventTypeDimension _AdditionalEvents_ FromFile, SDE_ORA_ EventTypeDimension _OtherHREvents, SDE_ORA_ EventTypeDimension _OtherHREvents_Full
domainValues_HRPosition_Active_Pos_Flg_ora11i.csv	W_HR_POSITION_ D.W_ACTIVE_ POSITION_FLG	Lists the Position Statuses and their corresponding domain values of "Active Position Flag" for the Oracle 11i Application.	SDE_ORA_ HRPositionDimension , SDE_ORA_ HRPositionDimension
domainValues_HRPosition_Exempt_Flg_ora11i.csv	W_HR_POSITION_ D.W_EXEMPT_FLG	Lists the FLSA Statuses and their corresponding domain values of "Exempt Flag" for the Oracle 11i Application.	SDE_ORA_ HRPositionDimension , SDE_ORA_ HRPositionDimension
domainValues_Job_Eeo_Cat_Code_ora11i.csv	W_JOB_D.W_EEO_ JOB_CAT_CODE	Lists the EEO Job Categories and their corresponding domain values of "EEO Job Category" for the Oracle 11i Application.	SDE_ORA_ JobDimension, SDE_ ORA_JobDimension_ Full
domainValues_Job_Flsa_Stat_Code_ora11i.csv	W_JOB_D.W_FLSA_ STAT_CODE	Lists the FLSA Statuses and their corresponding domain values of "FLSA Status Code" for the Oracle 11i Application.	SDE_ORA_ JobDimension, SDE_ ORA_JobDimension_ Full
domainValues_Pay_Type_Grp_Code_ora11i.csv	W_PAY_TYPE_D.W_ PAY_TYPE_GRP_ CODE	Lists the Classification Names, Element Names and their corresponding domain values of "Pay Type Group Code" for the Oracle 11i Application.	SDE_ORA_ PayTypeDimension, SDE_ORA_ PayTypeDimension_ Full

**Table 12–1 (Cont.) Domain Values and CSV Worksheet Files for Oracle HR Analytics**

<b>Worksheet File Name</b>	<b>Domain Value Table - Column</b>	<b>Description</b>	<b>Session</b>
domainValues_Pay_Type_Flg_ora11i.csv	W_PAY_TYPE_D.W_PAY_TYPE_FLG	Lists the Costing Debit or Credit values and their corresponding domain values of "Pay type Flag" for the Oracle 11i Application.	SDE_ORA_PayTypeDimension, SDE_ORA_PayTypeDimension_Full
domainValues_Employee_Ethnic_Group_Code_ora11i.csv	W_EMPLOYEE_D.W_ETHNIC_GRP_CODE	Lists the Ethnic codes and their corresponding domain values of "Ethnic Group Code" for the Oracle 11i Application.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employee_Sex_MF_ora11i.csv	W_EMPLOYEE_D.W_SEX_MF_CODE	Lists the Sex codes and their corresponding domain values of "Sex Code" for the Oracle 11i Application.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employee_Veteran_Status_Code_ora11i.csv	W_EMPLOYEE_D.W_VETERAN_STATUS_CODE	Lists the Veteran codes and their corresponding domain values of "Veteran Status Code" for the Oracle 11i Application.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employment_Cat_ora11i.csv	W_EMPLOYMENT_D.W_EMPLOYEE_CAT_CODE	Lists the User Person Types and their corresponding domain values of "Employment Category Code" for the Oracle 11i Application.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full
domainValues_Employment_Exempt_Flg_ora11i.csv	W_EMPLOYMENT_D.W_EXEMPT_FLG	Lists the FLSA Statuses and their corresponding domain values of "Exempt Flag" for the Oracle 11i Application.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full
domainValues_Employment_Full_Time_Flg_ora11i.csv	W_EMPLOYMENT_D.W_FULL_TIME_FLG	Lists the Employment Categories and their corresponding domain values of "Full Time Flag" for the Oracle 11i Application.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full
domainValues_Employment_Status_ora11i.csv	W_EMPLOYMENT_D.W_EMPLOYMENT_STAT_CODE	Lists the Per System Statuses and their corresponding domain values of "Employment Status" for the Oracle 11i Application.	SDE_ORA_EmploymentDimension, SDE_ORA_EmploymentDimension_Full

**Table 12–1 (Cont.) Domain Values and CSV Worksheet Files for Oracle HR Analytics**

<b>Worksheet File Name</b>	<b>Domain Value Table - Column</b>	<b>Description</b>	<b>Session</b>
domainValues_EventTypes_ora11i.csv	W_EVENT_TYPE_ D.W_EVENT_CLASS, W_EVENT_TYPE_ D.W_EVENT_GRP_ CODE, W_EVENT_ TYPE_D.W_EVENT_ REASON_CODE, W_ EVENT_SUBG_CODE	Lists the Event Types, Event Codes and Meanings and their corresponding domain values of "Event Group", "Event Sub-Group" and "Event Reason" for the Oracle 11i Application.	SDE_ORA_ EventTypeDimension _AbsenceAttendance, SDE_ORA_ EventTypeDimension _AbsenceAttendance_ Full, SDE_ORA_ EventTypeDimension _AdditionalEvents_ FromFile, SDE_ORA_ EventTypeDimension _OtherHREvents, SDE_ORA_ EventTypeDimension _OtherHREvents_Full
domainValues_HRPosition_Active_Pos_Flg_ora11i.csv	W_HR_POSITION_ D.W_ACTIVE_ POSITION_FLG	Lists the Position Statuses and their corresponding domain values of "Active Position Flag" for the Oracle 11i Application.	SDE_ORA_ HRPositionDimension , SDE_ORA_ HRPositionDimension
domainValues_HRPosition_Exempt_Flg_ora11i.csv	W_HR_POSITION_ D.W_EXEMPT_FLG	Lists the FLSA Statuses and their corresponding domain values of "Exempt Flag" for the Oracle 11i Application.	SDE_ORA_ HRPositionDimension , SDE_ORA_ HRPositionDimension
domainValues_Job_Eeo_Cat_Code_ora11i.csv	W_JOB_D.W_EEO_ JOB_CAT_CODE	Lists the EEO Job Categories and their corresponding domain values of "EEO Job Category" for the Oracle 11i Application.	SDE_ORA_ JobDimension, SDE_ ORA_JobDimension_ Full
domainValues_Job_Flsa_Stat_Code_ora11i.csv	W_JOB_D.W_FLSA_ STAT_CODE	Lists the FLSA Statuses and their corresponding domain values of "FLSA Status Code" for the Oracle 11i Application.	SDE_ORA_ JobDimension, SDE_ ORA_JobDimension_ Full
domainValues_Pay_Type_Grp_Code_ora11i.csv	W_PAY_TYPE_D.W_ PAY_TYPE_GRP_ CODE	Lists the Classification Names, Element Names and their corresponding domain values of "Pay Type Group Code" for the Oracle 11i Application.	SDE_ORA_ PayTypeDimension, SDE_ORA_ PayTypeDimension_ Full

**Table 12–1 (Cont.) Domain Values and CSV Worksheet Files for Oracle HR Analytics**

Worksheet File Name	Domain Value Table - Column	Description	Session
domainValues_Pay_Type_Flg_ora11i.csv	W_PAY_TYPE_D.W_PAY_TYPE_FLG	Lists the Costing Debit or Credit values and their corresponding domain values of "Pay type Flag" for the Oracle 11i Application.	SDE_ORA_PayTypeDimension, SDE_ORA_PayTypeDimension_Full
domainValues_Employee_Ethnic_Group_Code_ora11i.csv	W_EMPLOYEE_D.W_ETHNIC_GRP_CODE	Lists the Ethnic codes and their corresponding domain values of "Ethnic Group Code" for the Oracle 11i Application.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full
domainValues_Employee_Sex_MF_ora11i.csv	W_EMPLOYEE_D.W_SEX_MF_CODE	Lists the Sex codes and their corresponding domain values of "Sex Code" for the Oracle 11i Application.	SDE_ORA_EmployeeDimension, SDE_ORA_EmployeeDimension_Full

**12.2.2.2 How to Configure the domainValues\_Employee\_Ethnic\_Group\_Code\_ora11i.csv**

This section explains how to configure the domainValues\_Employee\_Ethnic\_Group\_Code\_ora11i.csv.

1. Identify the Ethnic Group Codes in your Oracle 11i source system by using the following SQL:

```
SELECT DISTINCT PER_INFORMATION1 FROM PER_ALL_PEOPLE_F
WHERE PER_INFORMATION1 in
('1','2','3','4','5','6','7','8','9','10','11','12','BA','BC','BO','C','I','O',
'P','W')
ORDER BY 1
```

2. Open the domainValues\_Employee\_Ethnic\_Group\_Code\_ora11i.csv file using a text editor in the \$pmsserver\lkpfiles folder.
3. Copy the PER\_INFORMATION1 to the ETHNIC\_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each Ethnic Code to one domain value.
5. Save and close the file.

**12.2.2.3 How to Configure the domainValues\_Employee\_Sex\_MF\_ora11i.csv**

This section explains how to configure the domainValues\_Employee\_Sex\_MF\_ora11i.csv.

1. Identify the Sex Codes in your Oracle 11i source system by using the following SQL:

```
SELECT DISTINCT SEX FROM PER_ALL_PEOPLE_F ORDER BY 1
```

2. Open the domainValues\_Employee\_Sex\_MF\_ora11i.csv file using a text editor in the \$pmserver\lkpfiles folder.
3. Copy the SEX column to the SEX column in the file. The data must be copied starting from the 6th line.
4. Map each Sex Code to one domain value.
5. Save and close the file.

#### 12.2.2.4 How to Configure the domainValues\_Employee\_Veteran\_Status\_Code\_ora11i.csv

This section explains how to configure the domainValues\_Employee\_Veteran\_Status\_Code\_ora11i.csv

1. Identify the Veteran Status Codes in your Oracle 11i source system by using the following SQL:

```
SELECT DISTINCT PER_INFORMATION5 FROM PER_ALL_PEOPLE_F
WHERE PER_INFORMATION5 in ('NOTVET', 'OTEDV', 'VET', 'VETDIS', 'VIETVET',
'VIETVETDIS')
ORDER BY 1
```

2. Open the domainValues\_Employee\_Veteran\_Status\_Code\_ora11i.csv file using a text editor in the \$pmserver\lkpfiles folder.
3. Copy the PER\_INFORMATION5 column to the VETERAN\_STATUS\_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each Veteran Status Code to one domain value.
5. Save and close the file.

#### 12.2.2.5 How to Configure the domainValues\_Employment\_Cat\_ora11i.csv

This section explains how to configure the domainValues\_Employment\_Cat\_ora11i.csv.

1. Identify the User Person Types in your Oracle 11i source system by using the following SQL:

```
SELECT DISTINCT SYSTEM_PERSON_TYPE, USER_PERSON_TYPE FROM
PER_PERSON_TYPES
WHERE SYSTEM_PERSON_TYPE IN
('EMP', 'OTHER', 'EMP_APL', 'EX_EMP', 'EX_EMP_APL', 'RETIREE', 'PRTN')
ORDER BY 1,2
```

2. Open the domainValues\_Employment\_Cat\_ora11i.csv file using a text editor in the \$pmserver\lkpfiles folder.
3. Copy the User Person Types to the USER\_PERSON\_TYPE column in the file. The data must be copied starting from the 6th line.
4. Map each User Person Type to one domain value. System Person Types are also extracted with User Person Type to help you map the domain values. Do not copy the System Person types in the CSV file.
5. Save and close the file.

#### 12.2.2.6 How to Configure the domainValues\_Employment\_Exempt\_Flg\_ora11i.csv

This section explains how to configure the domainValues\_Employment\_Exempt\_Flg\_ora11i.csv

1. Identify the FLSA Statuses in your Oracle11i source system by using the following SQL:

```
SELECT DISTINCT JOB_INFORMATION3 FROM PER_JOBS ORDER BY 1
```

2. Open the domainValues\_Employment\_Exempt\_Flg\_ora11i.csv file using a text editor in the \$pmserver\lkpfiles folder.
3. Copy the JOB\_INFORMATION3 to the FLSA\_STATUS\_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each FLSA\_STATUS\_CODE to one domain value.
5. Save and close the file.

### 12.2.2.7 How to Configure the domainValues\_Employment\_Full\_Time\_Flg\_ora11i.csv

This section explains how to configure the domainValues\_Employment\_Full\_Time\_Flg\_ora11i.csv.

1. Identify the Employment Categories in your Oracle11i source system by using the following SQL:

```
SELECT DISTINCT EMPLOYMENT_CATEGORY FROM PER_ALL_ASSIGNMENTS_F ORDER BY 1
```

2. Open the domainValues\_Employment\_Full\_Time\_Flg\_ora11i.csv file using a text editor in the \$pmserver\lkpfiles folder.
3. Copy the Employment Category to the EMPLOYMENT\_CATEGORY column in the file. The data must be copied starting from the 6th line.
4. Map each EMPLOYMENT\_CATEGORY to one domain value.
5. Save and close the file.

### 12.2.2.8 How to Configure the domainValues\_Employment\_Status\_ora11i.csv

This section explains how to configure the domainValues\_Employment\_Status\_ora11i.csv

1. Identify the Per System Statuses in your Oracle11i source system by using the following SQL:

```
SELECT DISTINCT PER_SYSTEM_STATUS FROM PER_ASSIGNMENT_STATUS_TYPES  
ORDER BY 1
```

2. Open the domainValues\_Employment\_Status\_ora11i.csv file using a text editor in the \$pmserver\lkpfiles folder.
3. Copy the PER\_SYSTEM\_STATUS to the PER\_SYSTEM\_STATUS column in the file. The data must be copied starting from the 6th line.
4. Map each PER\_SYSTEM\_STATUS to one domain value.
5. Save and close the file.

### 12.2.2.9 How to Configure the domainValues\_EventTypes\_ora11i.csv

This section explains how to configure the domainValues\_EventTypes\_ora11i.csv

1. Identify the Event Types in your Oracle11i source system by using the following SQL:

```
SELECT DISTINCT LOOKUP_TYPE, LOOKUP_CODE, MEANING
```

```

FROM FND_LOOKUP_VALUES
WHERE LOOKUP_TYPE IN
('EMP_ASSIGN_REASON',
'LEAV_REAS',
'PROPOSAL_REASON')
ORDER BY 1, 2, 3

```

2. Open the domainValues\_EventTypes\_ora11i.csv file using a text editor in the \$pmsserver\lkpfiles folder.
3. Copy the Lookup Type, Lookup Code and Meaning to the LOOKUP\_TYPE, LOOKUP\_CODE, and MEANING columns in the file respectively. The data must be copied starting from the 6th line. Use commas to separate the entries.
4. Map each Event Type (LOOKUP\_CODE) to one domain value for each of the 3 domain columns — W\_EVENT\_GRP\_CODE, W\_EVENT\_SUBG\_CODE, and W\_EVENT\_REASON\_CODE. Event Category (LOOKUP\_TYPE) and Event Description (MEANING) are also extracted with Event Type to help you map the domain values.
5. Save and close the file.

#### 12.2.2.10 How to Configure the domainValues\_HRPosition\_Active\_Pos\_Flg\_ora11i.csv

This section explains how to configure the domainValues\_HRPosition\_Active\_Pos\_Flg\_ora11i.csv.

1. Identify the Position Statuses in your Oracle11i source system by using the following SQL:

```

SELECT DISTINCT STATUS FROM HR_ALL_POSITIONS_F ORDER BY 1

```

2. Open the domainValues\_HRPosition\_Active\_Pos\_Flg\_ora11i.csv file using a text editor in the \$pmsserver\lkpfiles folder.
3. Copy the STATUS to the STATUS column in the file. The data must be copied starting from the 6th line.
4. Map each position STATUS to one domain value.
5. Save and close the file.

#### 12.2.2.11 How to Configure the domainValues\_HRPosition\_Exempt\_Flg\_ora11i.csv

This section explains how to configure the domainValues\_HRPosition\_Exempt\_Flg\_ora11i.csv.

1. Identify the FLSA Statuses in your Oracle11i source system by using the following SQL:

```

SELECT DISTINCT JOB_INFORMATION3 FROM PER_JOBS
ORDER BY 1

```

2. Open the domainValues\_HRPosition\_Exempt\_Flg\_ora11i.csv file using a text editor in the \$pmsserver\lkpfiles folder.
3. Copy the JOB\_INFORMATION3 to the FLSA\_STATUS\_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each FLSA\_STATUS\_CODE to one domain value.
5. Save and close the file.

### 12.2.2.12 How to Configure the domainValues\_Job\_Eeo\_Cat\_Code\_ora11i.csv

This section explains how to configure the domainValues\_Job\_Eeo\_Cat\_Code\_ora11i.csv

1. Identify the EEO Job Categories in your Oracle11i source system by using the following SQL:

```
SELECT DISTINCT JOB_INFORMATION1 FROM PER_JOBS
ORDER BY 1
```

2. Open the domainValues\_Job\_Eeo\_Cat\_Code\_ora11i.csv file using a text editor in the \$pmsserver\lkpfiles folder.
3. Copy the JOB\_INFORMATION1 to the EEO\_JOB\_CAT\_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each EEO\_JOB\_CAT\_CODE to one domain value.
5. Save and close the file.

### 12.2.2.13 How to Configure the domainValues\_Job\_Flsa\_Stat\_Code\_ora11i.csv

This section explains how to configure the domainValues\_Job\_Flsa\_Stat\_Code\_ora11i.csv.

1. Identify the FLSA Statuses in your Oracle11i source system by using the following SQL:

```
SELECT DISTINCT JOB_INFORMATION3 FROM PER_JOBS
ORDER BY 1
```

2. Open the domainValues\_Job\_Flsa\_Stat\_Code\_ora11i.csv file using a text editor in the \$pmsserver\lkpfiles folder.
3. Copy the JOB\_INFORMATION3 to the FLSA\_STAT\_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each FLSA\_STAT\_CODE to one domain value.
5. Save and close the file.

### 12.2.2.14 How to Configure the domainValues\_Pay\_Type\_Grp\_Code\_ora11i.csv

This section explains how to configure the domainValues\_Pay\_Type\_Grp\_Code\_ora11i.csv.

1. Identify the Pay Elements in your Oracle11i source system by using the following SQL:

```
SELECT DISTINCT CLASSIFICATION_NAME, ELEMENT_NAME
FROM
PAY_ELEMENT_TYPES_F,
PAY_ELEMENT_CLASSIFICATIONS
WHERE
PAY_ELEMENT_CLASSIFICATIONS.CLASSIFICATION_ID = PAY_ELEMENT_TYPES_
F.CLASSIFICATION_ID AND
CLASSIFICATION_NAME NOT LIKE '%Information%' AND
CLASSIFICATION_NAME NOT LIKE '%Employer%' AND
CLASSIFICATION_NAME NOT LIKE '%Balance%'
ORDER BY 1, 2
```

2. Open the domainValues\_Pay\_Type\_Grp\_Code\_ora11i.csv file using a text editor in the \$pmsserver\lkpfiles folder.

3. Copy the Classification Name and Element Name to the CLASSIFICATION\_NAME and ELEMENT\_NAME columns in the file respectively. The data must be copied starting from the 6th line. Use commas to separate the entries.
4. Map each Element Name to one domain value. Classification Names are also extracted with Element Names to help you map the domain values. If the element is not related to Payroll Pay Check, you can map the element to "OTHER".
5. Save and close the file.

### 12.2.2.15 How to Configure the domainValues\_Pay\_Type\_Flg\_ora11i.csv

This section explains how to configure the domainValues\_Pay\_Type\_Flg\_ora11i.csv.

1. Identify the Costing (Debit or Credit) in your Oracle11i source system by using the following SQL:

```
SELECT DISTINCT COSTING_DEBIT_OR_CREDIT FROM PAY_ELEMENT_CLASSIFICATIONS
ORDER BY 1
```

2. Open the domainValues\_Pay\_Type\_Flg\_ora11i.csv file using a text editor in the \$pmserver\lkipfiles folder.
3. Copy the COSTING\_DEBIT\_OR\_CREDIT to the COSTING\_DEBIT\_OR\_CREDIT column in the file. The data must be copied starting from the 6th line.
4. Map each \_DEBIT\_OR\_CREDIT to one domain value.
5. Save and close the file.

### 12.2.2.16 How to Configure Address Types for HR Profile

There are three address fields in the Employee dimension table:

- Permanent address
- Mail address
- Work address

For each of these, we use only the primary ones. In addition, the following logic is used to determine the various types of addresses:

- Permanent: Address Type = 'H' (This is also the home address, in other words)
- Mail: Address Type = 'M'. If this is not available, use Permanent address (which can be the best alternate for mailing address).
- Work: Address Type = 'Default'. If this is not available, use Permanent address (which can be the best alternate for mailing address).

You can modify this logic if required. For example, if your system tracks work address with a special address type 'W', then you should be able to modify the existing logic. Or, if you do not want to assume that the mail address (M) is same as the home address (H), you may want to remove the null evaluation check there.

#### To configure Address Type

1. In PowerCenter Designer, open the SDE\_ORA1158\_Adapter folder.
2. In Mapplet Designer, open the mplt\_SA\_ORA\_EmployeeDimension mapplet.
3. Locate the expression transformation Exp\_SA\_Employees\_Validate and scroll down the ports to get to the variables "HOME\_ADDRESS\_ID\_VAR", "MAIL\_ADDRESS\_ID\_VAR" and "DEFAULT\_ADDRESS\_ID\_VAR".

For example, if you have a specific address type for work addresses, (for example 'W'), you would modify the expression for the variable port DEFAULT\_ADDRESS\_ID\_VAR as follows:

From: :LKP.Lkp\_Address\_Id(INP\_PERSON\_ID, INP\_EFFECTIVE\_END\_DATE, 'Y', 'Default', INP\_DATASOURCE\_NUM\_ID)

To: :LKP.Lkp\_Address\_Id(INP\_PERSON\_ID, INP\_EFFECTIVE\_END\_DATE, 'Y', 'W', INP\_DATASOURCE\_NUM\_ID)

The output Address ID values are the next three ports "PERM\_ADDRESS\_ID", "MAIL\_ADDRESS\_ID" and "WORK\_ADDRESS\_ID". If you do not want to assume that the mail address (M) is same as the home address (H) in the event of mail address not being available, then you would modify the logic in the outgoing port "MAIL\_ADDRESS\_ID" as follows:

From: IIF(ISNULL(MAIL\_ADDRESS\_ID\_VAR), HOME\_ADDRESS\_ID\_VAR, MAIL\_ADDRESS\_ID\_VAR)

To: MAIL\_ADDRESS\_ID\_VAR

4. Validate and save changes to the repository. If you are using the version controlling for the Informatica repository, you will have to check in your changes as well.

### 12.2.2.17 How to Configure Phone Types for HR Profile

There are four phone related fields in the Employee dimension table, as follows:

- Fax
- Work phone
- Pager
- Mobile phone

The following logic is used to arrive at the various types of addresses:

- Fax: Phone Type = 'WF' (Work Fax)
- Work Phone: Phone Type = 'W1' (First work phone, if there are more than one)
- Pager: Phone Type = 'P' (Pager)
- Mobile: Phone Type = 'M' (Mobile)

You can modify this logic if required. For example, if your system tracks the primary work phone with a special phone type 'WP', instead of W1, then you should be able to modify the existing logic. The same applies for other phone types as well.

#### To configure Phone Type

1. In PowerCenter Designer, open the SDE\_ORA1158\_Adapter folder.
2. In Maplet Designer, open the mplt\_SA\_ORA\_EmployeeDimension maplet.
3. Locate the expression transformation Exp\_SA\_Employee and scroll down the ports to get to the output ports "EXT\_FAX\_PH\_NUM", "EXT\_WORK\_PHONE", "EXT\_PAGER\_NUM" and "EXT\_MOBILE\_NUM".

For example, if you have a specific phone type for the primary work phone, 'WP', you would modify the expression for the output port EXT\_WORK\_PHONE as follows:

From: :LKP.Lkp\_Phone\_Number(INP\_PERSON\_ID, 'W1', INP\_EFFECTIVE\_END\_DATE, INP\_DATASOURCE\_NUM\_ID)

To: :LKP.Lkp\_Phone\_Number(INP\_PERSON\_ID, 'WP', INP\_EFFECTIVE\_END\_DATE, INP\_DATASOURCE\_NUM\_ID)

4. Validate and save changes to the repository. If you are using the version controlling for the Informatica repository, you will have to check in your changes as well.

### 12.2.2.18 How to Configure Education Degree Codes for Employee Dimension

This section explains how to configure the categories that are used to evaluate the highest education degree code for an employee.

The session SDE\_ORA\_EmployeeDimension uses the Informatica parameter \$\$QUALIFICATION\_CATEGORY\_LIST to get the list of categories for the defined Education Degrees in Oracle E-Business Suite. The mapping SDE\_ORA\_EmployeeDimension uses this parameter to identify and populate Highest Education Degree attribute for the Employee Dimension. Incorrect setup of the parameter could cause the issues with quality of the data for Highest Education Degree attribute in Employee Dimension.

To configure the categories that are used to evaluate the highest education degree code for an employee:

1. Open the parameterfileOLTP.txt located on <DAC Install Root>\DAC\Informatica\Parameters\Input folder.
2. Navigate to the parameter section for session SDE\_ORA\_EmployeeDimension.
3. Set the value of the parameter \$\$QUALIFICATION\_CATEGORY\_LIST separated by comma and included in single quotes.

For example, mplt\_BC\_ORA\_EmployeeDimension.\$\$QUALIFICATION\_CATEGORY\_LIST = 'DEGREE', 'DT'.

---

**Note:** If you do not use the format specified above, you risk causing a SQL failure, which might prevent a successful ETL run. The parameter \$\$QUALIFICATION\_CATEGORY\_LIST is used in Lookup SQL override "WHERE" clause: < IN (\$\$QUALIFICATION\_CATEGORY\_LIST) >.

---

To obtain the list of the valid values for the education degree codes please login to Oracle E-Business Suite Instance using SQL\*Plus and execute the following SQL:

```
SELECT lookup_code, meaning FROM hr_lookups WHERE lookup_type = 'PER_CATEGORIES'
```

Select from the returned lookup\_code column values and decide which ones are used to identify Education Degrees.

### 12.2.2.19 About Configuring Key Flexfields

This section explains how to configure Key Flexfields in Oracle EBS applications.

In Oracle EBS Application module, a Flexfield is a field that allows a user to customize the Oracle applications. Each Flexfield in Oracle in turn consists of sub-fields called segments. Each segment is assigned a value set. The value set consists of values.

The main purpose of using Flexfields in Oracle EBS Applications is to customize the applications according to the business environment and practices. Oracle EBS

Application allows a user to validate values that are entered in these flexfields. A user can change the structure of a Flexfield depending on the data in the application.

Since Flexfield configurations vary from implementation to implementation, in order to accommodate all kinds of flexfield configurations done at the Oracle EBS Applications end, the attributes sourced from Flexfield columns is parameterized.

To configure Oracle HR Analytics to support flexfields, do the following:

- [Section 12.2.2.20, "How to Configure the Key Flexfields for the Job Dimension"](#)
- [Section 12.2.2.21, "How to Configure the Key Flexfields for the HR Position Dimension"](#)
- [Section 12.2.2.22, "How to Configure the Key Flexfields for the Pay Grade Dimension"](#)
- [Section 12.2.2.23, "How to Configure multi-segmented Flexfields"](#)

### **12.2.2.20 How to Configure the Key Flexfields for the Job Dimension**

This section is about configuring Job Code, Job Name and the Job Family Code columns.

To configure Job Code, Job Name and the Job Family Code columns using ParameterOLTP.txt

1. Open the parameterfileOLTP.txt from DAC\_HOME/Informatica/Parameters/Input directory and navigate to the task SDE\_ORA\_JobDimension.
2. Set the parameters \$\$JOBCODE\_FLXFLD\_SEGMENT\_COL with the Column Name of Job Code and \$\$JOBFAMILYCODE\_FLXFLD\_SEGMENT\_COL with the Column name of Job Group Code.
3. Navigate to the task SDE\_ORA\_CodeDimension\_Job.
4. Set the parameters for \$\$JOBCODE\_FLXFLD\_SEGMENT\_COL with the Column Name of Job Code and \$\$JOBNAME\_FLXFLD\_SEGMENT\_COL with the Column Name of the Job Name.

To find out what columns are defined in the Flexfield, do the following:

1. Login with Application Developer Responsibility.
2. Under the Key Flexfield, click on Segments.
3. Query for the Application Human Resources and locate the Flexfield Title relevant to Job.
4. Click on the Segments corresponding to the Flexfield that is being used in order to find the columns being used for the Job Code and Job Name.

### **12.2.2.21 How to Configure the Key Flexfields for the HR Position Dimension**

This section is about configuring the Position Number column.

To configure the Key Flexfields for the HR Position Dimension

1. Open the parameterfileOLTP.txt from DAC\_HOME/Informatica/Parameters/Input directory and navigate to the task SDE\_ORA\_HRPositionDimension.
2. Set the parameters \$\$POSITION\_NUM\_FLXFLD\_SEGMENT\_COL with the Column Name of Position Num.

3. Navigate to the task SDE\_ORA\_HRPositionDimension\_Full.
4. Set the parameters \$\$POSITION\_NUM\_FLXFLD\_SEGMENT\_COL with the Column Name of Position Num.

To find out what columns are defined in the Flexfield, do the following:

1. Login with Application Developer Responsibility.
2. Under the Key Flexfield, click on Segments.
3. Query for the Application Human Resources and locate the Flexfield Title relevant to Position.
4. Click on the Segments corresponding to the Flexfield that is being used in order to find the columns being used for the Position Num.

### 12.2.2.22 How to Configure the Key Flexfields for the Pay Grade Dimension

This section explains how to configure the Pay Level Name column.

To configure the Key Flexfields for the Pay Grade Dimension

1. Open the parameterfileOLTP.txt from DAC\_HOME/Informatica/Parameters/Input directory and navigate to the task SDE\_ORA\_CodeDimension\_Pay\_Level.
2. Set the parameter \$\$PAYLVNAME\_FLXFLD\_SEGMENT\_COL with the Column Name for the Pay level Name.

To find out what columns are defined in the Flexfield, do the following:

1. Login with Application Developer Responsibility.
2. Under the Key Flexfield, click on Segments.
3. Query for the Application Human Resources and locate the Flexfield Title relevant to Job.
4. Click on the Segments corresponding to the Flexfield that is being used in order to find the columns being used for the Job Code and Job Name.

### 12.2.2.23 How to Configure multi-segmented Flexfields

This section explains how to configure HR Analytics for multi-segmented Flexfields for any column.

In case your Flexfield configuration for a given business column involves more than one segment column, or in case you want to report a business column as a concatenation of more than one segment column separated by a 'dot' (say), then the corresponding parameter can be used to provide the information correctly. The example of configuring Job Name flexfield is provided below to illustrate the process.

When Oracle BI Applications is installed out-of-the-box, the Informatica parameter for the task 'SDE\_ORA\_CodeDimension\_Job' is set to the following:

```
$$JOBNAME_FLXFLD_SEGMENT_COL=SEGMENT6
```

If the Job Name in your deployment is configured using SEGMENT3 together with SEGMENT6 (or if you want to report Job Name as a concatenation of these two with 'dot' as the separator), you should set the Informatica parameter value as:

```
$$JOBNAME_FLXFLD_SEGMENT_COL=SEGMENT3 || SEGMENT6
```

Or, if you want a 'dot' in between, set the Informatica parameter value as:

```
$$JOBNAME_FLXFLD_SEGMENT_COL=SEGMENT3 || '.' || SEGMENT6
```

If your OLTP table already stores the concatenated value in some other column (like NAME), you could use it directly as follows:

```
$$JOBNAME_FLXFLD_SEGMENT_COL=NAME
```

**Note:** You must make sure you take a look at the SQL code (in Informatica designer Source Qualifier object) before setting the parameter values. For example, before configuring the values for the Informatica parameter \$\$JOBNAME\_FLXFLD\_SEGMENT\_COL (as in the above example), you should know which table the data will be pulled in. If necessary, use the following steps to obtain OLTP table names:

1. Open the parameterfileOLTP.txt from DAC\_HOME/Informatica/Parameters/Input directory and do a search on \$\$JOBNAME\_FLXFLD\_SEGMENT\_COL (against the current example).
2. Open the parameterfileOLTP.txt from DAC\_HOME/Informatica/Parameters/Input directory and do a search on \$\$JOBNAME\_FLXFLD\_SEGMENT\_COL (against the current example).  
In this case, the mapplet name that you need to derive is mplt\_BC\_ORA\_Codes\_Job.
3. Launch Informatica designer and log on to the repository and navigate to the appropriate folder.
4. Open the mapplet found in Step#2, open the Source Qualifier object, and see the SQL override. Get the physical OLTP table name from here. In this case, the table name is "PER\_JOB\_DEFINITIONS".

Once you know the OLTP table names, you can work out the column names to use in the parameter file (SEGMENT3 or SEGMENT6 or NAME and so on) that really exist in the table PER\_JOB\_DEFINITIONS.

#### 12.2.2.24 How to Configure Flags for the Pay Type Dimension

The Pay Type dimension W\_PAY\_TYPE\_D has three Flag columns namely COMP\_FLG, TAXABLE\_FLG and PENSION\_COMP\_FLG, apart from others. Each of these three flags can be either 'Y' or 'N'. These flag columns indicate whether the Pay Type mentioned in the current record is a Taxable Earning type, or Regular Compensation Earning type or a Pension Compensation Earning type. These Flag columns are chosen to contribute towards slowly changing dimension of type 2, and hence should be configured correctly.

Each of these three Flag columns are evaluated based on the Incoming Classification Name using a Flat File lookup. This Flat file needs to be configured to see the accurate values of the Pay Type Flags in the DataWarehouse.

To Configure Flags for the Pay Type Dimension

1. Open the file\_comp\_pension\_taxable\_flg\_ora11i.csv from Informatica Server\LkpFiles directory.
2. Add all possible Classification Names that are being used in your business.
3. Corresponding to the Classification Name, add the relevant values for the Pay Type Flags. Make sure to use either 'Y' or 'N'.

In Oracle HR Analytics, these flags apply to Earnings only. Therefore, you should only need to configure Classification Names related to Earnings. For other areas, the default Informatica mapping values should be acceptable.

### 12.2.2.25 How to Configure Classification Names for Payroll

This section explains how to configure classification names for Payroll.

The Aggregated items are loaded into the Payroll Fact table with a `DETAIL_FLG = 'N'`, apart from the line items. The Aggregated Line items that are loaded are: `TOTAL_GROSS`, `NET_PAY`, `TOTAL_DEDUCTIONS` and `TOTAL_TAXES`.

The following parameters need to be configured to get accurate results. Each Aggregated Line item is computed by grouping by elements that belong to a certain Classification name set.

**Table 12–2 Classification Name Parameters**

Parameter Name	Description
<code>\$\$GROSS_PAY_CLASSIFICATION_NAME</code>	Add all the Classification Names that add up to the Total Gross Pay.
<code>\$\$TOTAL_DEDUCTION_CLASSIFICATION_NAME</code>	Add all the Classification Names that correspond to any deduction.
<code>\$\$TOTAL_TAXES_CLASSIFICATION_NAME</code>	Add all the Classification Names that correspond to various Taxes.
<code>\$\$TOTAL_GROSS_FOR_NETPAY_EARNINGS</code>	Add all the Earnings Classification Names that are realized in Net pay. Note that some Classification Names are considered for Gross pay, but do not feature in the Net Pay (for example, Imputed Income). This is the major difference between this parameter and the <code>\$\$GROSS_PAY_CLASSIFICATION_NAME</code> parameter.
<code>\$\$TOTAL_GROSS_FOR_NETPAY_DEDUCTIONS</code>	This is same as the parameter <code>\$\$TOTAL_DEDUCTION_CLASSIFICATION_NAME</code> . However, in case some additional Classification Names are required to be added or removed for Net Pay, this parameter should be used for that.

The parameters are assigned default values when Oracle BI Applications is installed out of the box, but you can modify the values by following the steps below.

To configure the classification name parameters:

1. Open the `parameterfileOLTP.txt` under the `<DAC Install Dir>\Informatica\Parameters\Input` folder.
2. Navigate to the task `SDE_ORA_PayrollFact_Agg_Items`.
3. Add the Classification Names (as it appears in the OLTP) within quotes separated by commas.

---

**Note:** Oracle Business Intelligence Applications will use them within the IN clause of the SQL.

---

To find out the possible Classification Names existing in your system, please run the following SQL against the OLTP Database:

```
SELECT CLASSIFICATION_NAME FROM PAY_ELEMENT_CLASSIFICATIONS
WHERE LEGISLATION_CODE LIKE 'US'
```

The Classification Name of Elements can be obtained using the following SQL:

**Note:** Enter the Element Type IDs of the elements for which you need the Classification Name within the quotes. If there are multiple elements, separate them using commas:

```
SELECT E.ELEMENT_TYPE_ID, E.ELEMENT_NAME, C.CLASSIFICATION_NAME
FROM PAY_ELEMENT_TYPES_F E, PAY_ELEMENT_CLASSIFICATIONS C
WHERE E.CLASSIFICATION_ID = C.CLASSIFICATION_ID AND
E.ELEMENT_TYPE_ID IN ( )
```

### 12.2.3 Configuration Steps for Oracle HR Analytics For PeopleSoft

This section contains configuration steps required before you do a full data load that apply to PeopleSoft, and contains the following topics:

- [Section 12.2.3.1, "Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft 8.8 HCM"](#)
- [Section 12.2.3.2, "How to configure the domainValues\\_Employee\\_Ethnic\\_Group\\_Code\\_psft.csv"](#)
- [Section 12.2.3.3, "How to configure the domainValues\\_Employee\\_Sex\\_MF\\_psft.csv"](#)
- [Section 12.2.3.4, "How to configure the domainValues\\_Employee\\_Veteran\\_Status\\_Code\\_psft.csv"](#)
- [Section 12.2.3.5, "How to configure the domainValues\\_Employment\\_Category\\_psft.csv"](#)
- [Section 12.2.3.6, "How to configure the domainValues\\_Employment\\_Exempt\\_Flag\\_psft.csv"](#)
- [Section 12.2.3.7, "How to configure the domainValues\\_Employment\\_Full\\_Time\\_Flag\\_psft.csv"](#)
- [Section 12.2.3.8, "How to configure the domainValues\\_Employment\\_Status\\_psft.csv"](#)
- [Section 12.2.3.9, "How to configure the domainValues\\_Emp\\_EventType\\_Grp\\_Code\\_psft.csv"](#)
- [Section 12.2.3.10, "How to configure the domainValues\\_Emp\\_EventType\\_SubG\\_Code\\_psft.csv"](#)
- [Section 12.2.3.11, "How to configure the domainValues\\_HRPosition\\_Active\\_Pos\\_Flag\\_psft.csv"](#)
- [Section 12.2.3.12, "How to configure the domainValues\\_HRPosition\\_Exempt\\_Flag\\_psft.csv"](#)
- [Section 12.2.3.13, "How to configure the domainValues\\_Job\\_Eeo\\_Cat\\_Code\\_psft.csv"](#)
- [Section 12.2.3.14, "How to configure the domainValues\\_Job\\_Flsa\\_Stat\\_Code\\_psft.csv"](#)
- [Section 12.2.3.15, "How to configure the domainValues\\_Pay\\_Type\\_Grp\\_Code\\_psft.csv"](#)
- [Section 12.2.3.16, "How to configure the domainValues\\_Emp\\_EventType\\_EventReason\\_Code\\_psft.csv"](#)
- [Section 12.2.3.17, "How to configure the domainValues\\_Emp\\_EventType\\_EventClass\\_Code\\_psft.csv"](#)

- [Section 12.2.3.18, "How to configure the file\\_pension\\_comp\\_flg.csv"](#)
- [Section 12.2.3.19, "How to Configure Compensation Flag and Pension Compensation Flag for Pay Type Dimension"](#)
- [Section 12.2.3.20, "How to Configure the Performance Review Type ETL runtime parameter"](#)

### 12.2.3.1 Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft 8.8 HCM

The table below lists the CSV worksheet files and the domain values for PeopleSoft HR Analytics in \$pmsserver\LkpFiles folder located on Informatica Server box.

**Table 12–3 Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft 8.8 HCM**

Domain Value	Table	Column Description	Session
domainValues_Employee_Ethnic_Group_Code_psft.csv	W_EMPLOYEE_D.W_ETHNIC_GRP_CODE	Lists the Ethnic codes and corresponding domain values of "Ethnic Group Code" for PeopleSoft.	SDE_PSFT_EmployeeDimension
domainValues_Employee_Sex_MF_psft.csv	W_EMPLOYEE_D.W_SEX_MF_CODE	Lists the Sex codes and corresponding domain values of "Sex Code" for PeopleSoft.	SDE_PSFT_EmployeeDimension
domainValues_Employee_Veteran_Status_Code_psft.csv	W_EMPLOYEE_D.W_VETERAN_STATUS_CODE	Lists the Veteran codes and corresponding domain values of "Veteran Status Code".	SDE_PSFT_EmployeeDimension
domainValues_Employment_Category_psft.csv	W_EMPLOYMENT_D.W_EMPLOYEE_CAT_CODE	Lists the User Person Types and corresponding domain values of "Employment Category Code" for PeopleSoft.	SDE_PSFT_EmploymentDimension
domainValues_Employment_Exempt_Flg_psft.csv	W_EMPLOYMENT_D.W_EXEMPT_FLG	Lists the FLSA Statuses and corresponding domain values of "Exempt Flag".	SDE_PSFT_EmploymentDimension
domainValues_Employment_Full_Time_Flg_psft.csv	W_EMPLOYMENT_D.W_FULL_TIME_FLG	Lists the Employment Categories and corresponding domain values of "Full Time Flag".	SDE_PSFT_EmploymentDimension

**Table 12–3 (Cont.) Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft 8.8 HCM**

Domain Value	Table	Column Description	Session
domainValues_ Employment_ Status_psft.csv	W_ EMPLOYMENT_ D.W_ EMPLOYMENT_ STAT_CODE	Lists the Per System Statuses and corresponding domain values of "Employment Status".	SDE_PSFT_ EmploymentDimension
domainValues_ Emp_EventType_ Grp_Code_ psft.csv	W_EVENT_ TYPE_D.W_ EVENT_ CLASS,W_ EVENT_TYPE_ D.W_EVENT_ GRP_CODE,W_ EVENT_TYPE_ D.W_EVENT_ REASON_CODE	Lists the Event Types, Event Codes and Meanings and corresponding domain values of "Event Group" and Event Reason".	SDE_PSFT_ EventTypeDimension_Stage1 SDE_PSFT_ EventTypeDimension_Stage2
domainValues_ Emp_EventType_ SubG_Code_ psft.csv	W_EVENT_ SUBG_CODE	Lists the Event Types, Event Codes and Meanings and corresponding domain values of "Event Sub-Group"	SDE_PSFT_ EventTypeDimension_Stage1 SDE_PSFT_ EventTypeDimension_Stage2
domainValues_ HRPosition_ Active_Pos_Flg_ psft.csv	W_HR_ POSITION_D.W_ ACTIVE_ POSITION_FLG	Lists the Position Statuses and corresponding domain values of "Active Position Flag"	SDE_PSFT_ HRPositionDimension
domainValues_ HRPosition_ Exempt_Flg_ psft.csv	W_HR_ POSITION_D.W_ EXEMPT_FLG	Lists the FLSA Statuses and corresponding domain values of "Exempt Flag".	SDE_PSFT_ HRPositionDimension
domainValues_ Job_Eeo_Cat_ Code_psft.csv	W_JOB_D.W_ EEO_JOB_CAT_ CODE	Lists the EEO Job Categories and corresponding domain values of "EEO Job Category".	SDE_PSFT_JobDimension
domainValues_ Job_Flsa_Stat_ Code_psft.csv	W_JOB_D.W_ FLSA_STAT_ CODE	Lists the FLSA Statuses and Corresponding domain values of the "FLSA Status Code"	SDE_PSFT_JobDimension

**Table 12–3 (Cont.) Domain Values and CSV Worksheet Files for HR Analytics for PeopleSoft 8.8 HCM**

Domain Value	Table	Column Description	Session
domainValues_Pay_Type_Grp_code_psft.csv	W_PAY_TYPE_ D.W_PAY_ TYPE_GRP_ CODE	Lists the Classification Names, Element Names and corresponding domain values of "Pay Type Group Code"	SDE_PSFT_PayTypeDimension_Total, SDE_PSFT_PayTypeDimension_Earnings, SDE_PSFT_PayTypeDimension_FederalTaxes, SDE_PSFT_PayTypeDimension_Deductions, SDE_PSFT_PayTypeDimension_LocalTaxes1, SDE_PSFT_PayTypeDimension_LocalTaxes2, SDE_PSFT_PayTypeDimension_StateTaxes1, SDE_PSFT_PayTypeDimension_StateTaxes2
domainValues_Pay_Type_Flg_psft.csv	W_PAY_TYPE_ D.W_PAY_ TYPE_FLG	Lists the Costing Debitor Credit values and corresponding domain values of "Pay type Flag".	SDE_PSFT_PayTypeDimension_Earnings, SDE_PSFT_PayTypeDimension_FederalTaxes, SDE_PSFT_PayTypeDimension_Deductions, SDE_PSFT_PayTypeDimension_Total, SDE_PSFT_PayTypeDimension_LocalTaxes1, SDE_PSFT_PayTypeDimension_LocalTaxes2, SDE_PSFT_PayTypeDimension_StateTaxes1, SDE_PSFT_PayTypeDimension_StateTaxes2
domainValues_Emp_EventType_EventClass_Code_psft.csv	NA	TBD	SDE_PSFT_EventTypeDimension_Stage2
domainValues_Emp_EventType_EventReason_Code_psft.csv	NA	TBD	SDE_PSFT_EventTypeDimension_Stage2

### 12.2.3.2 How to configure the domainValues\_Employee\_Ethnic\_Group\_Code\_psft.csv

1. Identify the Ethnic Group Codes in your PeopleSoft source system by using the following SQL:

```
SELECT DISTINCT REG_REGION, ETHNIC_GRP_CD
FROM PS_DIVERS_ETHNIC
WHERE ETHNIC_GRP_CD NOT LIKE
ORDER BY 1
```

2. Open the domainValues\_Employee\_Ethnic\_Group\_Code\_psft.csv file text editor in the \$pmsserver\lkpfiles folder.
3. Copy the FIELDVALUE to the ETHNIC\_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each Ethnic Code to one domain value.
5. Save and close the file.

### 12.2.3.3 How to configure the domainValues\_Employee\_Sex\_MF\_psft.csv

1. Identify the Sex Codes in your PeopleSoft 8.8 source system by using the following SQL:

```
SELECT A.FIELDVALUE, A.XLATLONGNAME
FROM PSXLATITEM A
WHERE A.EFFDT = (SELECT MAX (C.EFFDT) FROM PSXLATITEM C WHERE
C.FIELDNAME = A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE)
AND A.FIELDNAME = 'SEX'
ORDER BY 1
```

2. Open the domainValues\_Employee\_Sex\_MF\_psft.csv file located in \$pmserver\lcpfiles folder on Informatica Server host using text editor.
3. Copy the FIELDVALUE column to the SEX column in the file. The data must be copied starting from the 6th line.
4. Map each Sex Code to one domain value.
5. Save and close the file.

### 12.2.3.4 How to configure the domainValues\_Employee\_Veteran\_Status\_Code\_psft.csv

1. Identify the Veteran Status Codes in your PeopleSoft 8.8 source system by using the following SQL:

```
SELECT A.FIELDVALUE, A.XLATLONGNAME
FROM PSXLATITEM A
WHERE A.EFFDT = (SELECT MAX (C.EFFDT) FROM PSXLATITEM C WHERE
C.FIELDNAME = A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE)
AND A.FIELDNAME = 'MILITARY_STATUS'
ORDER BY 1
```

2. Open the domainValues\_Employee\_Veteran\_Status\_Code\_psft.csv file located in \$pmserver\lcpfiles folder on Informatica Server host using text editor.
3. Copy the FIELDVALUE column to the VETERAN\_STATUS\_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each Veteran Status Code to one domain value.
5. Save and close the file.

### 12.2.3.5 How to configure the domainValues\_Employment\_Category\_psft.csv

1. Identify the User Person Types in your PeopleSoft 8.8 source system by using the following SQL:

```
SELECT DISTINCT LTRIM (RTRIM (EMPL_CLASS)) EMPL_CLASS
FROM PS_JOB
ORDER BY 1
```

2. Open the domainValues\_Employment\_Category\_psft.csv file located in \$pmserver\lcpfiles directory placed on Informatica Server host using text editor.

3. Copy the EMPL\_CLASS values to the EMP\_CAT\_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each User Person Type to one domain value.
5. Save and close the file.

#### 12.2.3.6 How to configure the domainValues\_Employment\_Exempt\_Flg\_psft.csv

1. Identify the FLSA Statuses in your PeopleSoft 8.8 source system by using the following SQL:

```
SELECT A.FIELDVALUE, A.XLATLONGNAME
FROM PSXLATITEM A
WHERE A.EFFDT = (SELECT MAX (C.EFFDT) FROM PSXLATITEM C WHERE
C.FIELDNAME = A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE)
AND A.FIELDNAME = 'FLSA_STATUS'
ORDER BY 1
```

2. Open the domainValues\_Employment\_Exempt\_Flg\_psft.csv file located in \$pmserver\lkpfiles directory placed on Informatica Server host using text editor.
3. Copy the FIELDVALUE to the FLSA\_STATUS column in the file. The data must be copied starting from the 6th line.
4. Map each FLSA\_STATUS to one domain value.
5. Save and close the file.

#### 12.2.3.7 How to configure the domainValues\_Employment\_Full\_Time\_Flg\_psft.csv

1. Identify the Employment Categories in your PeopleSoft 8.8 source system by using the following SQL:

```
SELECT A.FIELDVALUE, A.XLATLONGNAME
FROM PSXLATITEM A
WHERE A.EFFDT = (SELECT MAX (C.EFFDT) FROM PSXLATITEM C WHERE
C.FIELDNAME = A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE)
AND A.FIELDNAME = 'FULL_PART_TIME'
ORDER BY 1
```

2. Open the domainValues\_Employment\_Full\_Time\_Flg\_psft.csv file located in \$pmserver\lkpfiles directory placed on Informatica Server host using text editor.
3. Copy the FIELDVALUE to the FULL\_PART\_TIME column in the file. The data must be copied starting from the 6th line.
4. Map each EMPLOYMENT\_CATEGORY to one domain value.
5. Save and close the file.

#### 12.2.3.8 How to configure the domainValues\_Employment\_Status\_psft.csv

1. Identify the Per System Statuses in your PeopleSoft 8.8 source system by using the following SQL:

```
SELECT A.FIELDVALUE, A.XLATLONGNAME
FROM PSXLATITEM A
```

```
WHERE A.EFFDT = (SELECT MAX (C.EFFDT) FROM PSXLATITEM C WHERE
C.FIELDNAME = A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE)
AND A.FIELDNAME = 'EMPL_STATUS'
```

```
ORDER BY 1
```

2. Open the domainValues\_Employment\_Full\_Time\_Flg\_psft.csv file located in \$pmsserver\lcpfiles directory placed on Informatica Server host using text editor.
3. Copy the FIELDVALUE to the FULL\_PART\_TIME column in the file. The data must be copied starting from the 6th line.
4. Map each EMPLOYMENT\_CATEGORY to one domain value.
5. Save and close the file.

### 12.2.3.9 How to configure the domainValues\_Emp\_EventType\_Grp\_Code\_psft.csv

1. Identify the Event Types Group Codes in your PeopleSoft 8.8 source system by using the following SQL:

```
SELECT A.FIELDVALUE, A.XLATLONGNAME
FROM PSXLATITEM A
WHERE A.EFFDT = (SELECT MAX (C.EFFDT) FROM PSXLATITEM C WHERE
C.FIELDNAME = A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE)
AND A.FIELDNAME = 'ACTION'
ORDER BY 1
```

2. Open the domainValues\_EMP\_EventType\_Grp\_Code\_psft.csv file editor in \$pmsserver\lcpfiles folder using text.
3. Copy the FIELDVALUE to the LOOKUP\_TYPE\_CODE. The data must be copied starting from the 6th line. Use commas to separate the entries.
4. Map each Event Type (LOOKUP\_CODE) to one domain value for each domain column.
5. Save and close the file.

### 12.2.3.10 How to configure the domainValues\_Emp\_EventType\_SubG\_Code\_psft.csv

1. Identify the Event Types Subgroup Codes in your PeopleSoft 8.8 source system by using the following SQL:

MSSQL Server Syntax:

```
SELECT ACTION + ':' + ACTION_REASON SUBG_CODE, DESCR FROM PS_
ACTN_REASON_TBL ORDER BY 1
```

Oracle Syntax:

```
SELECT ACTION || ':' || ACTION_REASON SUBG_CODE, DESCR FROM PS_
ACTN_REASON_TBL ORDER BY 1
```

2. Open the domainValues\_EMP\_EventType\_SubG\_Code\_psft.csv file editor in \$pmsserver\lcpfiles folder using text.
3. Copy the SUBG\_CODE to the EVENT\_REASON. The data must be copied starting from the 6th line. Use commas to separate the entries.

4. Map each Event Type (EVENT\_REASON) to one domain value for each domain column.
5. Save and close the file.

#### 12.2.3.11 How to configure the domainValues\_HRPosition\_Active\_Pos\_Flg\_psft.csv

1. Identify the Position Statuses in your PeopleSoft 8.8 source system by using the following SQL:

```
SELECT A.FIELDVALUE, A.XLATLONGNAME
FROM PSXLATITEM A
WHERE A.EFFDT = (SELECT MAX (C.EFFDT) FROM PSXLATITEM C WHERE
C.FIELDNAME = A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE)
AND A.FIELDNAME = 'EFF_STATUS'
ORDER BY 1
```

2. Open the domainValues\_HRPosition\_Active\_Pos\_Flg\_psft.csv file using text editor in \$pmsserver\lkpfiles directory.
3. Copy the FIELDVALUE to the EFF\_STATUS column in the file. The data must be copied starting from the 6th line.
4. Map each position EFF\_STATUS to one domain value.
5. Save and close the file.

#### 12.2.3.12 How to configure the domainValues\_HRPosition\_Exempt\_Flg\_psft.csv

1. Identify the FLSA Statuses in your PeopleSoft 8.8 source system by using the following SQL:

```
SELECT A.FIELDVALUE, A.XLATLONGNAME
FROM PSXLATITEM A
WHERE A.EFFDT = (SELECT MAX (C.EFFDT) FROM PSXLATITEM C WHERE
C.FIELDNAME = A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE)
AND A.FIELDNAME = 'FLSA_STATUS'
ORDER BY 1
```

2. Open the domainValues\_HRPosition\_Exempt\_Flg\_psft.csv file using text editor in \$pmsserver\lkpfiles directory.
3. Copy the FIELDVALUE to the FLSA\_STATUS column in the file. The data must be copied starting from the 6th line.
4. Map each FLSA\_STATUS to one domain value.
5. Save and close the file.

#### 12.2.3.13 How to configure the domainValues\_Job\_Eeo\_Cat\_Code\_psft.csv

1. Identify the EEO Job Categories in your PeopleSoft 8.8 source system by using the following SQL:

```
SELECT 'EEO1CODE' CODE_TYPE,
LTRIM (RTRIM (EEO1CODE))
```

```

FROM PS_JOBCODE_TBL
UNION
SELECT 'EEO4CODE' CODE_TYPE,
LTRIM (RTRIM (EEO4CODE))
FROM PS_JOBCODE_TBL
UNION
SELECT 'EEO5CODE' CODE_TYPE,
LTRIM (RTRIM (EEO5CODE))
FROM PS_JOBCODE_TBL
UNION
SELECT DISTINCT 'EEO6CODE' CODE_TYPE,
LTRIM (RTRIM (EEO6CODE))
FROM PS_JOBCODE_TBL

```

2. Open the domainValues\_Job\_Eeo\_Cat\_Code\_psft.csv file in \$pmsserver\lkpfiles directory using text editor.
3. Copy the CODE\_TYPE to the EEO\_JOB\_CAT\_CODE\_TYPE column in the file. The data must be copied starting from the 6th line.
4. Map each pair of EEO\_JOB\_CAT\_CODE\_TYPE and EEO\_JOB\_CAT\_CODE\_VAL to one domain value.
5. Save and close the file.

#### 12.2.3.14 How to configure the domainValues\_Job\_Flsa\_Stat\_Code\_psft.csv

1. Identify the FLSA Statuses in your PeopleSoft 8.8 source system by using the following SQL:

```

SELECT A.FIELDVALUE, A.XLATLONGNAME
FROM PSXLATITEM A
WHERE A.EFFDT = (SELECT MAX (C.EFFDT) FROM PSXLATITEM C WHERE
C.FIELDNAME = A.FIELDNAME AND C.FIELDVALUE = A.FIELDVALUE)
AND A.FIELDNAME = 'FLSA_STATUS'
ORDER BY 1

```

2. Open the domainValues\_Job\_Flsa\_Stat\_Code\_psft.csv file in \$pmsserver\lkpfiles directory using text editor.
3. Copy the JOB\_INFORMATION3 to the FLSA\_STAT\_CODE column in the file. The data must be copied starting from the 6th line.
4. Map each FLSA\_STAT\_CODE to one domain value.
5. Save and close the file.

#### 12.2.3.15 How to configure the domainValues\_Pay\_Type\_Grp\_Code\_psft.csv

The table below shows pay type sources and their associated group codes.

**Table 12–4 Pay Elements for ETL**

PAYTYPE_SOURCE	W_PAY_TYPE_GRP_CODE
DEDUCTIONS	DEDUCTIONS
FEDERAL_TAXES	TAXES
LOCAL_TAXES	TAXES
STATE_TAXES	TAXES
EARNINGS	REGULAR
TOTAL	TOTAL

1. To configure the domainValues\_Pay\_Type\_Grp\_Code\_psft.csv
2. Open the domainValues\_Pay\_Type\_Grp\_Code\_psft.csv file using text editor in the \$pmsserver\lkpfiles directory.
3. Copy the Classification Name and Element Name to the PAYTYPE\_SOURCE. The data must be edited starting from the 6th line. Use commas to separate the entries.
4. Map each Element Name to one domain value. Classification Names are also extracted with Element Names to help you map the domain values. If the element is not related to Payroll Pay Check, you can map the element to "OTHER".
5. Save and close the file.

#### 12.2.3.16 How to configure the domainValues\_Emp\_EventType\_EventReason\_Code\_psft.csv

1. Identify the Costing (Debit or Credit) in your PeopleSoft 8.8 source system by using the following SQL:

```
SELECT ACTION + ':' + ACTION_REASON EVENT_REASON, DESCR FROM
PS_ACTN_REASON_TBL ORDER BY 1
```

Oracle Syntax:

```
SELECT ACTION || ':' || ACTION_REASON EVENT_REASON, DESCR FROM
PS_ACTN_REASON_TBL ORDER BY 1
```

2. Open the domainValues\_Emp\_EventType\_EventReason\_Code\_psft.csv file using a text editor in the \$pmsserver\lkpfiles folder.
3. Copy EVENT\_REASON to the EVENT\_REASON column in the file. The data must be copied starting from the 6th line.
4. Map each EVENT\_REASON to one domain value.
5. Save and close the file.

#### 12.2.3.17 How to configure the domainValues\_Emp\_EventType\_EventClass\_Code\_psft.csv

1. The shipped Domain Value for PeopleSoft 8.8 ETL source system is set:

- EVENT\_CLASS: W\_EVENT\_CLASS
- DEFAULT: HR\_EVENTS

2. Open the domainValues\_Emp\_EventType\_EventClass\_Code\_psft.csv file using a text editor in the \$pmsserver\lkpfiles folder.

3. Modify EVENT\_CLASS column in the file. The data must be copied starting from the 6th line.
4. Map each EVENT\_CLASS to one domain value.
5. Save and close the file.

#### **12.2.3.18 How to configure the file\_pension\_comp\_flg.csv**

1. Identify the Earnings Codes in your PeopleSoft 8.8 source system by using the following SQL:

```
SELECT DISTINCT ERNCD, DESCR FROM PS_EARNINGS_TBL ORDER BY 1
```

2. Open the file file\_pension\_comp\_flg\_psft.csv using a text editor in the \$pmsserver\lkpfiles folder.
3. Copy over the ERNCD and DESCR column into the file. The data must be copied starting from the sixth line.
4. Relate a Pension Compensation Flag and a Compensation Flag based on the nature of the Earnings Code.
5. Save and close the file.

The SQL can be further tuned if all the Earnings Code of your Organization is of a particular type. For e.g. If all the Earnings Code that you are using starts with "B" then the SQL can be modified as:

```
SELECT DISTINCT ERNCD, DESCR FROM PS_EARNINGS_TBL WHERE ERNCD LIKE 'B%' ORDER BY 1
```

#### **12.2.3.19 How to Configure Compensation Flag and Pension Compensation Flag for Pay Type Dimension**

The file "file\_pension\_comp\_flg\_psft.csv" is used to populate the Compensation flag and the Pension Compensation flag. The value of the Compensation Flag and the Pension Compensation flag is chosen based on the Input Earnings Code.

- Compensation Flag (COMP\_FLG)
- Pension Compensation Flag (PENSION\_COMP\_FLG)

##### **To modify Compensation Flag and Pension Compensation Flag**

1. Edit the file "file\_pension\_comp\_flg\_psft.csv" in the \$pmsserver\LkpFiles directory using in a text editor.
2. Enter the COMP\_FLG and PENSION\_COMP\_FLG value against the correct Earnings Code.
3. Save the file.

#### **12.2.3.20 How to Configure the Performance Review Type ETL runtime parameter**

As a part of the ETL configuration runtime parameter, value for Performance Review Type must be specified in Data Warehouse Administration Console (DAC). This parameter will be used to extract performance ranking information and related metrics for the Employee Snapshot Fact table.

##### **To configure the Performance Review runtime ETL parameter**

1. Start DAC and login to DAC Repository.
2. Select "Design" Mode and pick up "PeopleSoft 8.8" Container.

3. Navigate to TAB "Tasks" and select task "SDE\_PSFT\_EmployeeDailySnapshotFact\_Reviews\_Accruals" in upper pane.
4. Navigate to the Parameters TAB.
5. Set for the parameter \$\$PERFORMANCE\_REVIEW\_TYPE performance review type value.

SQL provided below can be used to find out Performance Review Types for review process in PeopleSoft 8.8 source system:

```
SELECT EP_REVIEW_TYPE, EP_REVW_DESCR FROM PS_EP_REVW_TYP_TBL
```

## 12.2.4 Configuration Steps for Oracle HR Analytics For Universal

This section contains configuration steps required before you do a full data load that apply to Universal.

Not applicable to Oracle BI Applications Version 7.9.4.

## 12.2.5 Configuration Steps for Controlling Your Data Set

This section contains additional configuration steps for Oracle HR Analytics.

- [Section 12.2.5.1, "Configuration Steps for Oracle HR Analytics For All Source Systems"](#)
- [Section 12.2.5.2, "Configuration Steps for Oracle HR Analytics For Oracle EBS"](#)
- [Section 12.2.5.3, "Configuration Steps for Oracle HR Analytics For PeopleSoft"](#)
- [Section 12.2.5.4, "Configuration Steps for Oracle HR Analytics For Universal"](#)

### 12.2.5.1 Configuration Steps for Oracle HR Analytics For All Source Systems

This section contains configuration steps that apply to all source systems, and contains the following topics:

- [Section 12.2.5.1.1, "How to Aggregate the Payroll Table for Oracle HR Analytics"](#)
- [Section 12.2.5.1.3, "How to control the rolling period for storing Employee Daily Snapshot data"](#)
- [Section 12.2.5.1.4, "How to control the granularity of the Employee Monthly Snapshot data"](#)

**12.2.5.1.1 How to Aggregate the Payroll Table for Oracle HR Analytics** You can aggregate the Payroll table to a different time levels, and aggregate levels of Employees, Jobs, and Payment Types dimensions. There are two time grain parameters to configure for this aggregate table and these parameters need to have the same value.

The GRAIN parameter has a preconfigured value of Month. The possible values for the GRAIN parameter are:

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

The Payroll aggregate table is fully loaded from the base table in the initial ETL run by the mapping "PLP\_PayrollAggregate\_Load\_Full". The table can grow to millions of records. The Payroll aggregate table is not fully reloaded from the base table after an incremental ETL run. Oracle HR Analytics minimizes the incremental aggregation effort, by modifying the aggregate table incrementally as the base table is updated. Oracle BI Applications looks for new records in the base table during the incremental ETL. This process is done in two steps:

1. There are new records in the W\_PAYROLL\_A table, which are inserted after the last ETL run. These new records are inserted into the W\_PAYROLL\_A\_TMP table. This step is part of the post load-processing workflow, and the mapping is called "PLP\_PayrollAggregate\_Extract".
2. Oracle HR Analytics aggregates the W\_PAYROLL\_A\_TMP table and joins it with the W\_PAYROLL\_A aggregate table to insert new or update existing buckets to the aggregate table. This step is part of the post load-processing workflow, and the mapping is called "PLP\_PayrollAggregate\_Load".

To load the Payroll aggregate table (W\_PAYROLL\_A), you need to configure the post-load processing parameters in the DAC (Data Administration Console).

To load the Payroll aggregate table (W\_PAYROLL\_A):

1. Log into the DAC.
2. Display the Design Tab.
3. Select the Task PLP\_PayrollAggregate\_Load under the Tasks Menu.
4. Add a parameter under the parameters tab with the Name \$\$GRAIN.
5. Depending upon the aggregation requirement define the value as one among 'DAY', 'WEEK', 'MONTH', 'QUARTER' or 'YEAR' and Choose the value to be Static.
6. Save the Task.

**12.2.5.1.2 About Configuring the Employee Snapshot Fact tables** The Employee Daily Snapshot Fact table stores employee-related metrics for all employees against their primary job function. You can configure the two fact tables that store Employee Snapshot data, namely W\_EMPLOYEE\_DAILY\_SNP\_F and W\_EMPLOYEE\_MONTHLY\_SNP\_F.

You can configure the Rolling Period and the Granularity for the Employee Snapshot data process.

### **Rolling Period**

The Employee Daily Snapshot Fact table is loaded everyday, the date being identified by the snapshot date column. For example, if your organization has 1000 employees, this table would load 1000 records everyday. If an employee's contract is terminated on a particular day, the following day's snapshot data would not have that record (that is, 999 records would be uploaded). Continuing this example, if you run ETL for 60 days, you will have 60 snapshot data sets. However, the number of days worth of snapshot data you want to store can be configured. In fact, you can configure to store only for a "rolling period of" some days, example 30. This is the out of box setting. If you start your ETL on January 1 and keep running for 40 days, you will have snapshot data sets from Jan 20 until Feb 10. If you configure to store only 7 rolling days, you will have snapshot data sets from Feb 3 until Feb 10, and so on.

The ETL mapping responsible for maintaining the rolling period choice you have indicated is "PLP\_EmployeeDailySnapshot\_Trim", which uses two Informatica parameters:

- **\$\$KEEP\_PERIOD**

Indicates your choice of the period type based on which you will be providing the rolling periods. The Valid values for this parameter are 'DAY', 'WEEK', 'MONTH', 'QUARTER', and 'YEAR' (values should include single quotes as indicated).

- **\$\$NUM\_OF\_PERIOD**

Indicates your choice of how many days/weeks/months/years (set by \$\$KEEP\_PERIOD) data you want to preserve in a rolling fashion. The valid value for this parameter is any integer.

For example, if you choose to preserve a rolling period of 45 Days, you would set \$\$KEEP\_PERIOD='DAY' and \$\$NUM\_OF\_PERIOD=45.

### Granularity

The Employee Monthly Snapshot Fact table stores the end-of-month snapshot data for all employees in your organization. Out of the box, this table stores end-of-month snapshot data, but this is configurable to store end-of-week, end-of-quarter or even end-of-year data. If you configure it to store end-of-month, as it is installed out-of-the-box, then you will have one snapshot data set for a given month, the set being refreshed everyday.

The out-of-box setting for the W\_EMPLOYEE\_MONTHLY\_SNP\_F table is to store end-of-month snapshot data set, on set for each calendar month. However, this can be configured. If you like to have this table store end-of-week, or end-of-quarter or even end-of-year data, your would need to configure the correct Informatica parameter definition in DAC. This parameter definition is as follows:

- **\$\$GRAIN** : Indicates the chosen granularity of the snapshot fact table that stores data at a interval higher than the base snapshot fact table. Valid values for this parameter are 'WEEK', 'MONTH', 'QUARTER', 'YEAR' (values should include single quotes as indicated).

### Notes:

- Do not set the value of \$\$GRAIN to 'DAY'.
- In the reporting logical metadata (RPD), the table W\_EMPLOYEE\_MONTHLY\_SNP\_F is joined with the Calendar Month dimension table - both in the physical layer as well as the logical layer. If you are change the \$\$GRAIN out-of-box value from 'MONTH' to anything else (like 'QUARTER'), you would need to modify the join in the RPD as well to the appropriate date dimension.
- The physical name of the table can still remain as W\_EMPLOYEE\_MONTHLY\_SNP\_F. You can change the logical name (or alias) appropriately. For example, if your \$\$GRAIN is 'QUARTER', you might change it to quarterly snapshot.

To configure the Employee Snapshot Fact tables, perform the following tasks:

- [Section 12.2.5.1.3, "How to control the rolling period for storing Employee Daily Snapshot data"](#)
- [Section 12.2.5.1.4, "How to control the granularity of the Employee Monthly Snapshot data"](#)

**12.2.5.1.3 How to control the rolling period for storing Employee Daily Snapshot data** For more information about the rolling period for the Employee Snapshot data, see [Section 12.2.5.1.2, "About Configuring the Employee Snapshot Fact tables"](#).

To configure the rolling period for storing Employee Daily Snapshot data:

1. Log in to the DAC repository, click the "Design" button and select your container from the drop down list.
2. Click on "Tasks" in the right pane and locate the task name = PLP\_EmployeeDailySnapshot\_Trim.
3. In the pane below, click on the "Parameters" tab.
4. Set the values of the two parameters \$\$KEEP\_PERIOD and \$\$NUM\_OF\_PERIOD.

**Note:** Although the \$\$NUM\_OF\_PERIOD requires an integer value, do not change the data type from "text".

**12.2.5.1.4 How to control the granularity of the Employee Monthly Snapshot data** For more information about granularity in the Employee Snapshot data, see [Section 12.2.5.1.2, "About Configuring the Employee Snapshot Fact tables"](#).

To configure the granularity of the Employee Snapshot data:

1. Log in to the DAC repository, click the "Design" button and select your container from the drop down list.
2. Click on "Tasks" in the right pane and locate the task name = PLP\_EmployeeMonthlySnapshot
3. In the pane below, click on the "Parameters" tab.
4. Set the value of the \$\$GRAIN parameter.

## **12.2.5.2 Configuration Steps for Oracle HR Analytics For Oracle EBS**

This section contains configuration steps that apply to Oracle EBS.

Not applicable to Oracle BI Applications Version 7.9.4.

## **12.2.5.3 Configuration Steps for Oracle HR Analytics For PeopleSoft**

This section contains configuration steps that apply to PeopleSoft, and contains the following topics:

- [Section 12.2.5.3.1, "How to Tune Performance for the Payroll table for PeopleSoft 8.8 HCM HR Analytics"](#)

### **12.2.5.3.1 How to Tune Performance for the Payroll table for PeopleSoft 8.8 HCM HR Analytics**

The Payroll mappings designed are designed to take care of the Payroll Adjustment and the Payroll Reversal process. These are complicated processes and normally take a reasonable amount of time to complete. Depending on your requirement, there are two ways you can achieve better overall performance in the Payroll Analytics. These techniques are as follows:

- Inactivate Adjustments and Reversals, if not required, for a given pay period
- Defer calculating Adjustments and Reversals for some time, until needed.

#### **To inactivate Adjustment and Reversal calculations for Payroll**

If you decide that there will be no Payroll Adjustment and Reversal for a Pay period, then you can choose to inactivate those tasks in DAC.

1. Go to Design Tab in the DAC Client.
2. Select Tasks.
3. Clear the check the box under the Inactive Column of the following tasks:
  - SDE\_PSFT\_PayrollFact\_Deductions\_Adjustments\_Reversals
  - SDE\_PSFT\_PayrollFact\_Earnings\_Adjustments\_Reversals
  - SDE\_PSFT\_PayrollFact\_EarningsOther\_Adjustments\_Reversals
  - SDE\_PSFT\_PayrollFact\_Total\_Adjustments\_Reversals
  - SDE\_PSFT\_PayrollFact\_Taxes\_Adjustments\_Reversals
4. Save the tasks.
5. Assemble the Subject Area (HR- Payroll) and build the Execution Plan.

**To defer Adjustment and Reversal calculations for Payroll**

1. inactivate Adjustment and Reversal calculations for Payroll and reassemble the Subject Area (Payroll) and Execution Plan, as described in the task above.
2. When you choose to run these again, reactivate the tasks, and reassemble Subject Area (Payroll) and Execution Plan.
3. Adjust the parameter `$$LAST_EXTRACT_DATE` so that it goes back to the date (preferably one day earlier) since when you decided not to run these tasks.

**12.2.5.4 Configuration Steps for Oracle HR Analytics For Universal**

This section contains configuration steps that apply to Universal.

Not applicable to Oracle BI Applications Version 7.9.4.



---

---

## Configuring Oracle Sales Analytics

This chapter describes how to configure Oracle Sales Analytics used with a Siebel CRM source system to meet your business needs.

It contains the following topics:

- [Section 13.1, "Overview"](#)
- [Section 13.2, "Configuration Required Before A Full Load"](#)

### 13.1 Overview

Oracle Sales Analytics is a comprehensive analytical solution that provides timely, fact-based insight into the entire sales process. It provides new levels of information richness, usability, and reach to sales professionals throughout the enterprise and in the field. All users, from sales executives to front-line sales representatives, get up-to-the-moment, complete, and in-context sales insight; insight that is personalized, relevant, and actionable. The benefits are faster and more informed decisions that help the sales organization compete more effectively, lower sales costs, and achieve better results.

### 13.2 Configuration Required Before A Full Load

This section contains configuration steps that you need to perform on Oracle Sales Analytics before you do a full data load, and contains the following topics:

- [Section 13.2.1, "How to Deploy Stored Procedures"](#)
- [Section 13.2.2, "Configuration Steps for Controlling Your Data Set"](#)

#### 13.2.1 How to Deploy Stored Procedures

Stored procedures are a group of SQL statements that perform particular tasks on the database. For example, stored procedures can help to improve the performance of the database. You deploy stored procedures by copying the stored procedure files from your Oracle BI Analytics installation and deploying them to the target data warehouse.

**Note:** Some sessions may fail if these procedures are not compiled in the database before running the workflows.

To deploy stored procedures:

1. Navigate to the OracleBI\dwrep\Informatica\Stored\_Procedure\_Scripts folder.

2. Open one of the folders based your database platform, and copy the source code from the file FIND\_AUDIT\_VALUES.sql into the target data warehouse schema.
3. Compile the stored procedures in the target data warehouse database.

**Note:** If you have problems deploying the stored procedures, see your database reference guide, or contact your database administrator.

## 13.2.2 Configuration Steps for Controlling Your Data Set

This sections contains additional configuration steps for Oracle Sales Analytics, and contains the following topics:

- [Section 13.2.2.1, "How to configure the snapshot frequency in Pipeline fact"](#)

### 13.2.2.1 How to configure the snapshot frequency in Pipeline fact

Even though the snapshot of Pipeline is captured every time ETL is run, you can choose to persist daily snapshots, or weekly snapshots or monthly snapshots or even yearly snapshots. There is one parameter you need to modify to configure for this. This parameter is called "GRAIN".

The GRAIN parameter has a preconfigured value of 'DAY'. The possible values for the GRAIN parameter are:

- DAY
- WEEK
- MONTH
- QUARTER
- YEAR

To configure the snapshot frequency in Pipeline fact:

1. Open the parameterfileDW.txt file using a text editor in the OracleBI\DAC\Informatica\parameters\input folder.
2. Locate [SIL\_PipelineFact] and \$\$GRAIN
3. Replace the default parameter value with your new value.  
**Note:** You need to use single quotes for the value.
4. Save and close the file.

---

---

# Configuring Oracle Contact Center Telephony Analytics

This chapter describes how to configure the Oracle Contact Center Telephony Analytics with Universal adapter to meet your business needs.

It contains the following topics:

- [Section 14.1, "Overview of Oracle Contact Center Telephony Analytics"](#)
- [Section 14.2.1, "About the Dimension Key Resolution Process for Universal Source"](#)

---

---

**Note:** For configuration steps that apply to all analytics modules (for example, Oracle Financial Analytics, Oracle HR Analytics, Oracle Sales Analytics, see [Chapter 8, "Configuring Common Areas and Dimensions"](#)).

---

---

## 14.1 Overview of Oracle Contact Center Telephony Analytics

Oracle Contact Center Telephony Analytics tracks incoming and outgoing contacts and their resulting activities, and the agents performing contact center tasks. The Oracle Contact Center Telephony Analytics application is made up of these functional areas:

- Representative Activities
- Contact Center and Agent Performance
- Contact Center Benchmark and Targets
- Interactive Voice Response History

## 14.2 Configuration Required Before A Full Load for Contact Center Telephony Analytics

This chapter describes how to configure Contact Center Telephony Analytics for particular sources to meet your business needs, and contains the following topics:

- [Section 14.1, "Overview of Oracle Contact Center Telephony Analytics"](#)
- [Section 14.2.1, "About the Dimension Key Resolution Process for Universal Source"](#)
- [Section 14.2.2, "How to Load the Dimension Table"](#)

- [Section 14.2.3, "How to Load the Fact Table"](#)
- [Section 14.2.4, "Setting Up The DATASOURCE\\_NUM\\_ID Parameter For The Universal Adapter"](#)
- [Section 14.2.5, "About Configuring the Event Type Columns"](#)
- [Section 14.2.7, "About Configuring Contact Status Type Column"](#)
- [Section 14.2.12, "How to Configure Dates and Times"](#)
- [Section 14.2.8, "Setting Up the Representative Activities Table"](#)
- [Section 14.2.9, "Setting Up the Contact Center Performance Table"](#)
- [Section 14.2.10, "Setting Up the Benchmarks and Targets Table"](#)
- [Section 14.2.13, "Configuring Flags for Oracle Contact Center Telephony Analytics"](#)
- [Section 14.2.14, "How to Exclude Representative Data for Post-Load Processing"](#)

## 14.2.1 About the Dimension Key Resolution Process for Universal Source

Dimension keys uniquely identify each record in a dimension table. The purpose of dimension keys is to relate a record in the dimension table to a record in the fact table. Therefore, the dimension key must be stored in both the dimension table and the fact table and resolved using the dimension table's `INTEGRATION_ID` and `DATASOURCE_NUM_ID` column values.

For universal business adapters, users supply the dimension `INTEGRATION_ID` and `DATASOURCE_NUM_ID` column values through a flat file interface. The same values for `INTEGRATION_ID` and `DATASOURCE_NUM_ID` are expected in both the dimension and fact business adapters so that the correct dimension key is resolved and loaded into the fact table.

The dimension key resolution process entails two steps:

1. Run the dimension table workflows to extract and load dimension records.

The dimension load mapping automatically creates a surrogate key for each record in the dimension table. This surrogate key value populates the dimension table's primary key column, which is referred to as the *dimension WID*. Similar to the `INTEGRATION_ID` column, which uniquely identifies the record within the source system, the dimension key uniquely identifies the record in the data warehouse dimension table.

2. Run the fact table workflows to extract and load fact records.

Records must contain the dimension ID column values for each fact record; these values must be the same values as the `INTEGRATION_ID` in the corresponding dimension tables.

The following sections describe these two steps in more detail by taking the example of one fact table (`W_REP_ACTIVITY_F`) and one dimension table (`W_EVENT_TYPE_D`). However, this process applies to all fact and dimension tables joined by a dimension key.

## 14.2.2 How to Load the Dimension Table

This task is explained by taking the example of the `W_EVENT_TYPE_D` table.

Loading the `W_EVENT_TYPE_D` table requires the following ETL processes:

1. The `SDE_Universal_EventTypeDimension` mapping extracts the data from `file_event_type.csv` and populates the `W_EVENT_TYPE_DS` staging table.
2. The `SIL_EventTypeDimension` mapping sources data from the staging table and passes it over to the Source Independent Load (SIL) mapping. The SIL mapping generates the surrogate key for each record in the staging table, then inserts it into `W_EVENT_TYPE_D` target table.

### 14.2.3 How to Load the Fact Table

This task is explained by taking the example of the `W_REP_ACTIVITY_F` fact table.

Loading the `W_REP_ACTIVITY_F` fact table requires the following ETL processes:

1. The `SDE_Universal_Rep_Activity` mapping extracts the data from `file_rep_activity.csv` and populates the `W_REP_ACTIVITY_FS` staging table.
2. The `SIL_Rep_ActivityFact` mapping sources the data from the staging table, and the fact SIL mapplet resolves the dimension key by doing a lookup on `W_EVENT_TYPE_D` using the values supplied in the `ACTIVITY_TYPE_ID` column and the `DATASOURCE_NUM_ID` column. Then, the SIL populates the `W_REP_ACTIVITY_F` fact table.

Since the dimension `*_ID` values are supplied through the Universal Interface flat file, it is critical that you supply the same value for the `INTEGRATION_ID` in the dimension table and the corresponding `*_ID` field in the joined fact table. In addition, you must verify that the `DATASOURCE_NUM_ID` column values match (for Universal Sources, the value that is defined out-of-the-box for the `DATASOURCE_NUM_ID` column is 3). If you supply different values for the two tables, the fact table load mapping is not able to resolve the dimension key. As a result, you cannot perform queries on the fact table using that dimension.

### 14.2.4 Setting Up The `DATASOURCE_NUM_ID` Parameter For The Universal Adapter

All the Source Dependent Extract mappings (SDE) for the Universal Adapter are designed to accept the value for the `DATASOURCE_NUM_ID` column, either from the input data file or from a predefined parameter. If the value for a record is not supplied in a file, then the value assigned to the parameter is picked up.

The `DATASOURCE_NUM_ID` parameter is defined in the Data Warehouse Administration Console (DAC) Universal container and is set to the internal DAC system variable '@DAC\_DATASOURCE\_NUM\_ID'. The value for this variable is dynamically set by DAC depending on the physical data source that is being used.

### 14.2.5 About Configuring the Event Type Columns

The `W_EVENT_TYPE_D` table is a dimension class table which tracks various types of events. This dimension table plays multiple roles in the Oracle Contact Center Telephony Analytics application. The various roles are described in the following section.

#### 14.2.5.1 The Activity Type Column

Contact center representatives engage in various activities, such as logging into the Automated Call Distributor (ACD) system to handle customer calls, taking a scheduled break, taking an unscheduled break, and so on. All such activities that you want to load into the warehouse should be provided in the `file_event_type.csv` source file to be stored in the `W_EVENT_TYPE_D` table with the `W_EVENT_CLASS` column set to the `REACTIVITY` domain value.

The `ACTIVITY_TYPE_WID` dimension key in the `W_REP_ACTIVITY_F` fact table identifies the nature of the activity. This key is resolved using the `W_EVENT_TYPE_D` table. To resolve the `ACTIVITY_TYPE_WID` dimension key in the `W_REP_ACTIVITY_F` table, the `W_REP_ACTIVITY_F` and `W_EVENT_TYPE_D` tables are joined through the `ACTIVITY_TYPE_ID` column and the `DATASOURCE_NUM_ID` column. For the `ACTIVITY_TYPE_WID` dimension key to resolve properly in the `W_REP_ACTIVITY_F` fact table, you must verify that the `ACTIVITY_TYPE_ID` column and the `DATASOURCE_NUM_ID` column values in `file_rep_activity.csv` file match with the `INTEGRATION_ID` column and the `DATASOURCE_NUM_ID` column values in the `file_event_type.csv` file. If the two columns do not match for a particular record, the fact load mapping cannot resolve the dimension key for that fact record.

### 14.2.5.2 Call Type (or Contact Type) Column

In a contact center there are various reasons for which customers contact an organization and organizations contact a customer. The contact can happen through various channels, such as phone, online chat, email, fax, and so on.

Some of the reasons for a customer to contact your organization include:

- Placing an order for a product or service.
- Lodging a complaint.
- Inquiring about a product offering.

Some of the reasons for your organization to contact a customer include:

- Performing a customer satisfaction survey.
- Following up on an inquiry.

The call types that you want to load into the Oracle Business Analytics Warehouse are provided in the `file_event_type.csv` source file to be stored in the `W_EVENT_TYPE_D` table with the `W_EVENT_CLASS` column set to the `CONTACT` domain value.

The `CALL_TYPE_WID` dimension key in `W_ACD_EVENT_F` fact table identifies the type of call. This key is resolved using the `W_EVENT_TYPE_D` table. To resolve the `CALL_TYPE_WID` dimension key in `W_ACD_EVENT_F` fact table, the `W_ACD_EVENT_F` and `W_EVENT_TYPE_D` tables are joined through the `CALL_TYPE_WID` column and the `DATASOURCE_NUM_ID` column. For the `CALL_TYPE_WID` dimension key to resolve properly in the `W_ACD_EVENT_F` fact table, you must verify that the `CALL_TYPE_WID` column and the `DATASOURCE_NUM_ID` column values in `file_acd_event.csv` file match with the `INTEGRATION_ID` column and the `DATASOURCE_NUM_ID` column values in the `file_event_type.csv` file. If the two columns do not match for a particular record, the fact load mapping cannot resolve the dimension key for that fact record.

The `CNTCT_TYPE_WID` dimension key in the `W_CNTCT_CNTR_PERF_F` fact table identifies the same information and it is resolved in a similar process. It requires the `CNTCT_TYPE_ID` column and the `DATASOURCE_NUM_ID` column values in the `file_cntct_cntr_perf.csv` file to match with the `INTEGRATION_ID` column and the `DATASOURCE_NUM_ID` column values in the `file_event_type.csv` file.

### 14.2.5.3 Call Event Type Column

In a call center, the Computer Telephony Integration (CTI) system tracks each activity associated with a call, including:

- Call placed in queue to be answered
- Call answered by a contact representative

- Call placed on hold by a contact representative
- Call transferred from one contact representative to another
- Call hung up by the customer

The call events that you want to load into the Oracle Business Analytics Warehouse are provided in the `file_event_type.csv` source file and stored in the `W_EVENT_TYPE_D` table with the `W_EVENT_CLASS` column set to `INTRA_CALL`.

To resolve the `CALL_EVENT_TYPE_WID` dimension key in `W_ACD_EVENT_F` fact table, the `W_ACD_EVENT_F` and `W_EVENT_TYPE_D` tables are joined through the `CALL_EVNT_TYPE_ID` column and the `DATASOURCE_NUM_ID` column. For the `CALL_EVENT_TYPE_WID` dimension key to resolve properly in the `W_ACD_EVENT_F` fact table, you must verify that the `CALL_EVNT_TYPE_ID` column and the `DATASOURCE_NUM_ID` column values in `file_acd_events.csv` file match with the `INTEGRATION_ID` column and the `DATASOURCE_NUM_ID` column values in the `file_event_type.csv` file. If the two columns do not match for a particular record, the fact load mapping cannot resolve the dimension key for that fact record.

#### 14.2.5.4 After-Call Work Activity (ACW) Type Column

On many occasions, contact representative are required to perform certain tasks after the initial contact. These tasks might include creating a follow-up action item list, dispatching the case from the contact to a particular group, and so on. These activities are known as after-call work (ACW) activities. The call events that you want to load into the Oracle Business Analytics Warehouse are provided in the `file_event_type.csv` source file and stored in the `W_EVENT_TYPE_D` table with the `W_EVENT_CLASS` column set to the `ACWACTIVITY` domain value.

To resolve the `ACW_ACT_TYPE_WID` dimension key in `W_CNTCT_CNTR_PERF_F` fact table, the `W_CNTCT_CNTR_PERF_F` table is joined with `W_EVENT_TYPE_D` table through the `ACW_ACT_TYPE_ID` column and the `DATASOURCE_NUM_ID` column. For the `ACW_ACT_TYPE_WID` dimension key to resolve properly in the `W_CNTCT_CNTR_PERF_F` fact table, you must verify that the `ACW_ACT_TYPE_ID` column and the `DATASOURCE_NUM_ID` column values in the `file_cntct_cntr_perf.csv` file match with the `INTEGRATION_ID` column and the `DATASOURCE_NUM_ID` column values in the `file_event_type.csv` file. If the two columns do not match for a particular record, the fact load mapping cannot resolve the dimension key for that fact record.

## 14.2.6 About Configuring The Contact Reason Column

The `W_REASON_D` table is a class dimension table that tracks various classes of reason such as those associated with the opportunity, opportunity status, quota, contact, contact transfer and so on. These classes are differentiated by the `W_REASON_CLASS` column.

For Oracle Contact Center Telephony Analytics, Contact Reason is an important reason class. All contacts made either by the customer to your organization, or by your organization to a customer, need to have a reason (for example, an inquiry or complaint).

The Contact Reasons that you want to load into the Oracle Business Analytics Warehouse are provided in the `file_reason.csv` source file to be stored in the `W_REASON_D` table with the `W_REASON_CLASS` column set to the domain value `CONTACT`. The Contact Transfer Reasons are provided with the `W_REASON_CLASS` set to `CONTACT_TRANSFER`.

To resolve the CNTCT\_REASON\_WID dimension key in W\_CNTCT\_CNTR\_PERF\_F fact table, the W\_CNTCT\_CNTR\_PERF\_F table is joined with W\_REASON\_D through the CNTCT\_REASON\_ID column and the DATASOURCE\_NUM\_ID column. For the CNTCT\_REASON\_WID dimension key to resolve properly in the W\_CNTCT\_CNTR\_PERF\_F fact table, you must verify that the CNTCT\_REASON\_ID column and the DATASOURCE\_NUM\_ID column values in file\_cntct\_cntr\_perf.csv match with the INTEGRATION\_ID column and the DATASOURCE\_NUM\_ID column values in the file\_reason.csv file. If the two columns do not match for a particular record, the fact load mapping cannot resolve the dimension key for that fact record.

## 14.2.7 About Configuring Contact Status Type Column

The W\_STATUS\_D is a dimension class table that tracks various classes of status such as those associated with the purchase orders, requisitions, shipments, inventory, customer cases, contact statuses, and so on. These classes are differentiated by the W\_STATUS\_CLASS column.

For the Oracle Contact Center Telephony Analytics application, Contact Status is an important status class. All contacts made either by the customer to your organization, or by your organization to a customer, are assigned a status. Examples include:

- Customer abandoned call before contact
- Customer abandoned call during contact
- Contact completed

The contact statuses that you want to load into the Oracle Business Analytics Warehouse are provided in the file\_status.csv source file to be stored in the W\_STATUS\_D table with the W\_STATUS\_CLASS column set to the CONTACT domain value.

The W\_STATUS\_CODE column in the W\_STATUS\_D table also contains domain values. The four domain values ABANDONED, RELEASE, DISCONNECTED, and HANGUP, are used in the computation of Contact Center Performance metrics. Therefore, it is critical that while you load all your Contact Statuses through the source file, the records are mapped into the appropriate W\_STATUS\_CODE domain value.

To resolve the CNTCT\_STATUS\_WID dimension key in W\_CNTCT\_CNTR\_PERF\_F fact table, the W\_CNTCT\_CNTR\_PERF\_F table is joined with W\_STATUS\_D through the CNTCT\_STATUS\_ID column and the DATASOURCE\_NUM\_ID column. For the CNTCT\_STATUS\_WID dimension key to resolve properly in the W\_CNTCT\_CNTR\_PERF\_F fact table, you must verify that the CNTCT\_STATUS\_ID column and the DATASOURCE\_NUM\_ID column values in file\_cntct\_cntr\_perf.csv match with the INTEGRATION\_ID column and the DATASOURCE\_NUM\_ID column values in the file\_status.csv file. If the two columns do not match for a particular record, the fact load mapping cannot resolve the dimension key for that fact record.

## 14.2.8 Setting Up the Representative Activities Table

When setting up the Representative Activities table you must consider the following:

- All events in the Representative Activities table are time span events. The events are not point in time events.
- The calculation of the Actual, Scheduled, Login, and Break durations are based on the event durations in the source-system data. To avoid duplication in a representative's time, the representative activity records must not overlap in time. For example, if the Login and Break activities overlap in time in the source-system data, then the time durations are counted towards both categories.

- The hourly aggregate is the lowest level of aggregation provided. Representatives are counted as present for an hourly bucket if they are present for any part of that hour. For example, if a representative activity starts at 9.45 A.M. and ends at 10.45 A.M., the representative is counted as present for 9-10 A.M. and 10-11 A.M. time buckets. No weight factor is used to indicate the representative is available for part of the hour. However, the duration of activities are apportioned into the two hourly buckets. If the Local Time Zone is different from the warehouse time zone, the duration of activities will be apportioned in all the local and warehouse hours that had part of that activity.
- The number of breaks a representative takes is calculated by counting the number of break records. There is one break record for each representative for each break (scheduled or actual). If a break is split into multiple records in the source system, then it is counted as multiple breaks in the Oracle Business Analytics Warehouse.
- If a representative's activity spans across the date boundary, then you must provide two different records for that representative for the same activity, with different activity start dates and times. The same applies to the local date boundary as well. Hence, if an activity crosses the day boundary in warehouse time and local time at different points in time (due to time-zone difference between local and warehouse), there should be three records for that particular activity, each having an Activity start day that is the same as the Activity end day and the Activity start local day the same as the Activity end local day. For example, if a representative logs on to the Automatic Call Distributor (ACD) system at 23:30 on January 4, 2006 (23:00 on January 4, 2006 in local time) and logs off from the ACD system at 01:30 on January 5, 2006 (01:00 on January 5, 2006 in local time) then create three records in the file\_rep\_activity.csv flat file interface, as shown in the following table.

**Table 14–1 Setting up activities and date boundaries**

Rep	Activity Type	Activity Start Date	Activity Start Time	Activity End Date	Activity End Time	Activity Start Local Date	Activity Start Local Time	Activity End Local Date	Activity End Local Time
Rep1	LOGIN	01/04/2006	23:30:00	01/04/2006	23:59:59	01/04/2006	23:00:00	01/04/2006	23:29:59
Rep1	LOGIN	01/05/2006	00:00:00	01/05/2006	00:29:59	01/04/2006	23:29:59	01/04/2006	23:59:59
Rep1	LOGIN	01/05/2006	00:30:00	01/05/2006	01:30:00	01/05/2006	00:00:00	01/05/2006	01:00:00

**To set up the Representative Activities table**

1. Open the file\_rep\_activity.csv file using a text editor in the \$PMRoot\SrcFiles folder.
2. Type in your data to load the Representative Activities table.
3. Save and close the file.

### 14.2.9 Setting Up the Contact Center Performance Table

The Contact Center Performance table stores information at a contact and representative grain. For example, if a call is handled by two representatives, there are two records for that contact in the Contact Center Performance table.

When setting up the Contact Center Performance table you must consider the following:

- The Abandoned Contact Count, Answered Contact Count, Hangup Contact Count, and Released Contact Count metrics are counts of contacts based on the Contact Status. The Contact Center Performance table is preconfigured to expect the Contact Status in the `file_cntct_cntr_perf.csv` file is to be at a Contact level. If you configure the Contact Status at the contact and representative level, you need to make sure that these aggregate metrics are defined at the contact and representative level in the appropriate workflows. You need to make any changes in the Select clause of the Source Qualifier SQL statement in the `PLP_ContactCenterPerformanceRepHourAggregate_Load` mapping. You also need to configure the metadata in the repository file. You need to change the definitions of these metrics in the Logical Table Source that maps to the `W_CNTCT_CNTR_PERF_F` fact table.

- *Answered contacts* are defined as the contacts whose status is not marked as ABANDONED. The Answered Contact Count metric is calculated as follows:

```
COUNT(DISTINCT (CASE WHEN W_STATUS_D.W_STATUS_CODE != 'ABANDONED' THEN
W_CNTCT_CNTR_PERF_F.CNTCT_NUM END)).
```

You can choose not to count calls which are completed in the Interactive Voice Response (IVR) as an answered call. You can exclude these contacts from the Answered Contact Count by assigning them a different or new Contact Status.

- As contacts are associated with a representative, a Contact Representative Identifier needs to be provided for each fact record. The IVR can be modeled as a virtual representative and assigned a Contact Representative Identifier. Calls waiting in a general queue, and not assigned to a representative, also need to be given their own Contact Representative Identifier.
- The majority of the data for the Contact Center Performance table is sourced from the data in the `file_acd_event.csv` file. You must make sure that the source data is consistent across the `file_acd_event.csv` and `file_cntct_cntr_perf.csv` files.

#### To set up the Contact Center Performance table

1. Open the `file_cntct_cntr_perf.csv` file using a text editor in the `$PMRootDir\SrcFiles` folder.
2. Type in your data to load the Contact Center Performance table.
3. Save and close the file.

### 14.2.10 Setting Up the Benchmarks and Targets Table

The benchmarks and targets subject area is used to establish targets for various metrics for your Contact Center and also capture industry benchmarks for the same metrics. The values of these metrics are calculated using the Contact Center Performance subject area and are compared to the Benchmarks and Targets table to evaluate the performance of your Contact Center.

When setting up the Benchmarks and Targets table you must consider the following:

- The `file_cntct_cntr_bnmrk_tgt.csv` file must supply the effective date range for each benchmark record. The date range is used to identify the appropriate benchmark to compare with the actuals and the determination of other metrics such as the Service Level. Actuals refers to the actual value of the metric (during the period) as opposed to the planned or targeted value of the metric.

- You need to supply an appropriate date range for the benchmark records. For example, if the benchmark records do not vary over time, a large date range can be used. An example is shown below:
  - PERIOD\_START\_DT 01/01/1899
  - PERIOD\_END\_DT 01/01/3714
- The Benchmarks and Targets table is preconfigured at the contact level. You can define other benchmarks and targets, for example, an Hourly-Total-Hold-Duration benchmark, and these can be added using the extension columns in the data warehouse. For more information on the methodology for storing additional data in the data warehouse, see [Chapter 6, "Customizing the Oracle Business Analytics Warehouse"](#).
- For each dimension in the Benchmark and Targets fact table, you can decide if a benchmark or target varies by that dimension or not. If you choose to keep a benchmark or target constant over a dimension, you need to supply a question mark (?) as the value for the dimension ID. In addition, the metric needs to be leveled in the repository (RPD) at the grand-total level of that dimension. This dimension ID also needs to be removed from the join in the SQL statement in the `PLP_ContactCenterPerformanceServiceLevelAggregate` mapping. If you choose to vary a benchmark or target by a dimension, you need to provide benchmark or target for each value of the dimension.
- The `FORECAST_CNTCT_CNT` table in the source file is preconfigured to forecast the number of calls for a day for a combination of dimensions.

The Benchmarks and Targets table is preconfigured with the smallest effective date range of a day. To changing the grain to be hourly, perform the following procedure.

#### To configure the grain of the Benchmarks and Targets table as an hourly level

1. Add the Hour WID to the `W_CNTCT_CNTR_BNCHMRK_TGT_F` fact table.
2. Modify the `SDE_Universal_ContactCenterBenchmarkTargetFact` and `SIL_ContactCenterBenchmarkTargetFact` mappings to populate the Hour WID based on the Period Start Date and Period End Date.  
  
These dates need to fall on the hour boundaries and not in the middle of an hourly interval.
3. Modify the `PLP_ContactCenterPerformanceRepHourAggregate_Load` mapping SQL statement to now join also on Hour WID to the Benchmarks and Targets table.
4. Modify the metadata in the repository to include the new physical and logical joins to the `W_HOUR_OF_DAY_D` dimension.
5. Set the content pane settings on the fact table to the newly added Hour (Time) dimension in the RPD.

#### To set up the Benchmarks and Targets table

1. Open the `file_cntct_cntr_bchmrk_tgt.csv` file using a text editor in the `$PMRootDir\SrcFiles` folder.
2. Type in your data to load the Benchmarks and Targets.
3. Save and close the file.

## 14.2.11 Logical Delete of Records From Fact Tables

If a fact record is deleted physically in the telephony data source, you can logically delete that record from the data warehouse. To do this, a target staging table called <FACT table>\_PE needs to be created for each fact table and a mapping needs to be created that can load this table.

For example in order to delete records from W\_ACD\_EVENT\_F, a new staging table needs to be created as W\_ACD\_EVENT\_F\_PE where PE stands for primary extract. The structure of this table will be same for any fact table as it will have only two columns: INTEGRATION\_ID and DATASOURCE\_NUM\_ID. A new mapping SDE\_Universal\_<Fact>\_Primary\_Extract needs to be created to load this the primary extract staging table with all the INTEGRATION\_ID and DATASOURCE\_NUM\_ID records that have not been deleted and are currently present in OLTP system.

The following example illustrates this process. In this example, there are five records in the telephony data source and in the data warehouse table W\_ACD\_EVENT\_F, as follows:

**Table 14–2 Example records in the W\_ACD\_EVENT\_F table**

INTEGRATION_ID	DATASOURCE_NUM_ID	CALLID_NUM	DELETE_FLG
1	0	20060101_C1	N
2	0	20060101_C2	N
3	0	20060101_C3	N
4	0	20060101_C4	N
5	0	20060101_C5	N

For any reason, records with an Integration\_Id = 3 and a Datasource\_num\_Id = 0 are deleted from the source system. Then Our primary extract mapping should load the remaining four records in the W\_ACD\_EVENT\_PE table, as follows:

**Table 14–3 Example records in the W\_ACD\_EVENT\_PE table**

INTEGRATION_ID	DATASOURCE_NUM_ID
1	0
2	0
4	0
5	0

This will be followed by the SIL\_IdentifyDelete and the SIL\_\*\_SoftDelete mappings, which are already available in the data warehouse repository, as installed out-of-the-box. The SIL\_ACDEventFact\_IdentifyDelete mapping will do a left outer join on W\_ACD\_EVENT\_F and W\_ACD\_EVENT\_F\_PE to find out the missing records in W\_ACD\_EVENT\_F\_PE, and to check whether the missing record is not already deleted (DELETE\_FLG='Y') in the W\_ACD\_EVENT\_F. If not, it will load those records in the staging table W\_ACD\_EVENT\_F\_DEL.

**Table 14–4 Example records in the W\_ACD\_EVENT\_F\_DEL table**

INTEGRATION_ID	DATASOURCE_NUM_ID
3	0

The SIL\_ACDEventFact\_SoftDelete mapping will read records from the W\_ACD\_EVENT\_F\_DEL and update the DELETE\_FLG of those records in W\_ACD\_EVENT\_F to 'Y', as follows:

**Table 14–5 Example records in the W\_ACD\_EVENT\_F table**

INTEGRATION_ID	DATASOURCE_NUM_ID	CALLID_NUM	DELETE_FLG
1	0	20060101_C1	N
2	0	20060101_C2	N
3	0	20060101_C3	Y
4	0	20060101_C4	N
5	0	20060101_C5	N

## 14.2.12 How to Configure Dates and Times

The Contact Center Telephony Analytics application supports analysis by both the local date/time and the Warehouse date/time. On the fact tables, the \_LDT columns refer to the local date and time while the \_DT fields refer to the warehouse date and time. If you want to capture the local date and time on a transaction, you will need to provide that data in the \_LDT columns through the flat file interface. You will also have to apply the appropriate transformation to compute the warehouse date/time and supply this in the \_DT columns through the flat file interface. Even if you choose not to support analysis based on local time, you will need to supply data for the \_DT (warehouse date/time) columns. The table below provides a list of the applicable date and local date columns for each fact table.

**Table 14–6 Date Columns and Flat Files**

Flat file	Applicable Date Column	Applicable Local Date Column	Table Using the Local Date
file_rep_activity.csv	ACTIVITY_START_DT, ACTIVITY_END_DT	ACTIVITY_START_LDT, ACTIVITY_END_LDT	W_REP_ACTIVITY_F
file_acd_event.csv	EVENT_START_DT, EVENT_END_DT	EVENT_START_LDT, EVENT_END_LDT	W_ACD_EVENT_F
file_cntct_cntr_perf.csv	CNTCT_START_DT, CNTCT_END_DT	CNTCT_START_LDT, CNTCT_END_LDT	W_CNTCT_CNTR_PERF_F
file_rep_activity_hour.csv (Alternate Load)	ACTIVITY_DT	ACTIVITY_LDT	W_REP_ACTIVITY_HOUR_A
file_cntct_cntr_perf_rep_hour.csv (Alternate Load)	CNTCT_DT	CNTCT_LDT	W_CNTCT_CNTR_PERF_REP_HOUR_A

### To provide local dates in the flat file interface

1. Open the applicable flat file interface.
2. In the flat file interface, input the new dates in the \*\_DT fields.
3. Input the corresponding local dates in the \*\_LDT fields.
4. Save your changes to the flat file interface.

5. Run a test load for 10 records to verify that your new dates are loaded into the applicable table.

### 14.2.13 Configuring Flags for Oracle Contact Center Telephony Analytics

Many of the fact and dimension tables within the Oracle Contact Center Telephony Analytics application use flag fields to provide value-added information pertaining to a contact or contact representative. These flag fields are configurable and include the following:

- CONSULT\_FLG
- CONFERENCE\_FLG
- PERTINENT\_INFO\_FLG
- CNTCT\_MTCH\_FLG
- IVR\_FLG

The possible values for these flag fields in the data warehouse tables are Y or N. The following table shows Configurable Flag Values and Descriptions.

**Table 14–7 Configurable Flag Values and Descriptions**

Flag	Flag Value	Description
CONSULT_FLG	Y	Indicates that the contact representative consulted with other contact representative during the course of the call or contact.
	N	Indicates that the contact representative did not consult with other contact representative during the course of the call or contact.
CONFERENCE_FLG	Y	Indicates that the contact representative conferenced with other contact representatives during the course of the call or contact.
	N	Indicates that the contact representative did not conference with other contact representative during the course of the call or contact.
PERTINENT_INFO_FLG	Y	Indicates that the pertinent information was available for the contact.
	N	Indicates that the pertinent information was not available for the contact.
CNTCT_MTCH_FLG	Y	Indicates that the contact was matched with the existing customer data using Customer Entered Digits (CED) such as PIN Numbers, Account Numbers, or Social Security Number.
	N	Indicates that the contact could not be matched with the existing customer data using Customer Entered Digits (CED) such as PIN Numbers, Account Numbers, or Social Security Number.
IVR_FLG	Y	Indicates that the call associated with the call was recorded in the Interactive Voice Response (IVR) system
	N	Indicates that the call associated with the call was not recorded in the Interactive Voice Response (IVR) system

## 14.2.14 How to Exclude Representative Data for Post-Load Processing

This section contains information about excluding representative data from contact representative and organization performance aggregate tables for post-load processing.

You may want to exclude certain contact representatives from the aggregation process for various reasons. For example, you may want to exclude representatives who are on vacation, or who are working on special projects.

The default configuration calculates contact-related information for all contact representatives in the enterprise. There are five aggregate tables supplied with the Oracle Contact Center Telephony Analytics application for improving the performance of the dashboards and reports:

- W\_CNTCT\_CNTRC\_PERF\_REP\_HOUR\_A
- W\_CNTCT\_CNTR\_PERF\_REP\_DAY\_A
- W\_CNTCT\_CNTR\_PERF\_REP\_MONTH\_A
- W\_CNTCT\_CNTR\_PERF\_ORG\_HOUR\_A
- W\_CNTCT\_CNTR\_PERF\_ORG\_DAY\_A

### To exclude data about specific representatives from the aggregation calculation

1. In PowerCenter Designer, open the Configuration for Post Load Processing folder.
2. Open the `PLP_ContactCenterPerformanceRepAggregate_Extract` mapping.
3. Double-click the `EXP_CNTCT_CNTR_PERF` transformation.
4. Locate the `EXCLUSION_IND_FLG` port.
5. This port is preconfigured with a value of N indicating that all rows are included in the aggregates. Change this logic to include your logic to determine which groups of records you want to exclude.
6. Validate and save the mapping to the repository.

---

**Note:** If you exclude data from an aggregate table, you also need to apply the same filter to the Logical Table Source corresponding to the `W_CNTCT_CNTR_PERF_F` base fact table in the repository metadata (Fact Service Contact Center Performance logical table). The metrics computed from the base fact tables are now consistent with those computed from the aggregate tables.

---

## 14.2.15 How to Configure Alternate Load Plan to Enable Direct Load to the Entry-Level Rep Activity Aggregates

The Representative Activity fact table stores information at an individual representative activity grain. In case the data is not available at that grain, you can alternately choose to provide summarized data at the hourly level and load the Rep Activity hour level aggregate table `W_REP_ACTIVITY_HOUR_A` directly.

This data will be used by the post-load process to load the following higher level aggregates:

- W\_REP\_ACTIVITY\_DAY\_A
- W\_REP\_ACTIVITY\_MONTH\_A

**To configure the alternate load plan to directly load aggregate table**

1. Login to the DAC repository
2. Select the Universal container
3. Choose Design, then Subject Area and query for the "Service – Rep Activity" subject area.
4. With this subject area selected, in the detail pane, click on Configuration Tags.
5. Uncheck the configuration point "ECC- Load into base fact and then aggregates".
6. Check the configuration point "ECC – Load directly into aggregate facts".
7. Redesign the subject area "Service – Rep Activity" and save.
8. Redesign the execution plan "Universal Contact Center Telephony Analytics".

**To provide hour level data**

1. Open the file rep\_activity\_hour.csv using a text editor from \$PMRootDir\SrcFiles folder.
2. Enter the data for Representative activities at the hourly level.
3. Save and close the file.

---



---

**Note:** If you directly load the entry-level aggregate table, the metrics in the Fact - Service - Rep Activities logical fact table that are mapped only to the base W\_REP\_ACTIVITY\_F LTS will not be populated.

---



---

## 14.2.16 How to Configure Alternate Load Plan to Enable Direct Load to the Entry-Level Contact Performance Aggregates

The Contact Center Performance table stores information at a contact and representative grain. In case the data is not available at a contact grain, you can alternately choose to provide summarized data at the hourly level and load the Contact Center Performance hourly aggregate table, W\_CNTCT\_CNTR\_PERF\_REP\_HOUR\_A, directly.

This data will be used by the post-load process to load the following higher level aggregates.

W\_CNTCT\_CNTR\_PERF\_REP\_DAY\_A  
 W\_CNTCT\_CNTR\_PERF\_REP\_MONTH\_A  
 W\_CNTCT\_CNTR\_PERF\_ORG\_HOUR\_A  
 W\_CNTCT\_CNTR\_PERF\_ORG\_DAY\_A

**To configure the alternate load plan to directly load aggregate**

1. Login to the DAC repository
2. Select the Universal container
3. Go to Design, then Subject Area and query for the "Service - Contact Center Performance" subject area
4. With this subject area selected, click on Configuration Tags
5. Uncheck the configuration point "ECC- Load into base fact and then aggregates"
6. Check the configuration point "ECC – Load directly into aggregate facts"

7. Redesign the subject area "Service - Contact Center Performance and save.
8. Redesign the execution plan "Universal Contact Center Telephony Analytics".

**To provide hour level data**

1. Open the file\_rep\_activity\_hour.csv using a text editor from \$PMRootDir\SrcFiles folder.
2. Enter the data for Contact Center Performance at the hourly level.
3. Save and close the file.

---

**Note:** The Configuration Tag for the Direct Load to both Rep Activity and Contact Center Performance Hourly Aggregates is the same and hence either both can be loaded at hourly level or both can be loaded at point-in-time grain. If you directly load the entry-level aggregate table, the metrics in the Fact - Service - Contact Center Performance logical fact table that are mapped only to the base W\_CNTCT\_CNTR\_PERF\_F LTS will not be populated.

---

### 14.2.17 How to Set Up the Integration\_Id of the Contact Center Telephony Facts in the Warehouse

The INTEGRATION\_ID by default is provided in the SrcFiles and they are unique for a DATASOURCE\_NUM\_ID in a fact. Alternately, they can also be formed in the warehouse in the SDE adapter mappings using a set of Dimension\_Ids and other columns of each record. The combination of Dimension Ids and other columns used for forming Integration\_id is unique within a DATASOURCE\_NUM\_ID and identifies the grain of the fact. For example, in the W\_ACD\_EVENT\_F fact table, INTEGRATION\_ID is formed as CALLID\_NUM||'~'||EVENT\_SEQUENCE\_NUM.

**To set up formation of Integration\_Id in SDE adapter**

1. Login to the DAC repository and select the Universal container.
2. Go to the SDE\_Universal task and click on the parameter tab.
3. Change the value of \$\$READ\_INTEGRATION\_ID\_FROM\_FILE parameter to FALSE.

The following table lists the expression of INTEGRATION\_ID for each fact and the SDE task that creates it.

**Table 14–8 Expressions and Task Names**

Table Name	INTEGRATION_ID Expression	Task Name
W_ACD_EVENT_F	CALLID_NUM  '~'  EVENT_SEQUENCE_NUM	SDE_Universal_ACDEventFact
W_CNTCT_CNTR_PERF_F	CNTCT_NUM  '~'  CNTCT_REP_ID  '~'  CNTCT_START_DT	SDE_Universal_ContactCenterPerformance Fact

**Table 14–8 (Cont.) Expressions and Task Names**

Table Name	INTEGRATION_ID Expression	Task Name
W_CNTCT_CNTR_ BNCHMRK_TGT_F	PERIOD_START_ DT  '~'  COMPANY_ORG_ ID  '~'  CNTCT_REP_ ID  '~'  REP_HR_ORG_ ID  '~'  CNTCT_REP_ORG_ ID  '~'  CNTCT_CNTR_ LOC_ID  '~'  CHNL_TYPE_ ID  '~'  CUSTOMER_ ID  '~'  PRODUCT_ID	SDE_Universal_ ContactCenterBenchmark TargetFact
W_CNTCT_CNTR_PERF_ REP_HOUR_A	CNTCT_CNTR_LOC_ ID  '~'  CNTCT_REP_ ID  '~'  CNTCT_DT	SDE_Universal_ ContactCenterPerformance RepHourAggregate
W_REP_ACTIVITY_F	CNTCT_REP_ ID  '~'  ACTIVITY_TYPE_ ID  '~'  ACTIVITY_START_ DT  '~'  ACTIVITY_ START_LDT	SDE_Universal_ RepActivityFact
W_REP_ACTIVITY_HOUR_ A	CNTCT_REP_ ID  '~'  ACTIVITY_TYPE_ ID  '~'  ACTIVITY_DT	SDE_Universal_ RepActivityHourAggregate

## 14.2.18 Setting Up the IVR Menu and IVR Navigation Profile Tables

The IVR Menu Dimension table W\_IVR\_MENU\_D stores information about each menu point in the Interactive Voice Response System. The Integration\_id in this table uniquely identifies a IVR MENU and is used as the dimension identifier IVR\_MENU\_ID in the ACD Event Fact fact table, W\_ACD\_EVENT\_F.

If a call has n number of consecutive IVR events without any transfer to a contact representative, the NAV\_PATH\_ID for that call is identified as MENU1-MENU2-MENU3-.....-MENU n where MENU m is the IVR\_MENU\_ID of the mth event in that call. The NAV\_PATH\_ID is used to determine the navigation profile dimension key for a call.

It is recommended that the Integration\_Id of the IVR MENU dimension W\_IVR\_MENU\_D does not exceed 10 characters in order to keep the NAV\_PATH\_ID within the defined size limit of 255 characters.

### 14.2.18.1 Determining Navigation Profile and Transfer Flag for a Call

If a call has multiple transfer from IVR to Rep or REP to IVR, a Navigation Profile is determined for each segment of the call that was completely within the IVR system. For example, suppose a call had eight events as given below.

**Table 14–9 Example call events in a navigation profile**

CallId	Event Type	Event Sequence Number	Menu Id
20060104-C1	IVR	1	MENU1
20060104-C1	IVR	2	MENU2
20060104-C1	IVR	3	MENU3
20060104-C1	REP	4	
20060104-C1	IVR	5	MENU4

**Table 14–9 (Cont.) Example call events in a navigation profile**

CallId	Event Type	Event Sequence Number	Menu Id
20060104-C1	IVR	6	MENU5
20060104-C1	IVR	7	MENU6
20060104-C1	IVR	8	MENU7

There will be two NAV\_PATH\_ID for both IVR segment of the call. Transfer Flag will be 'Y' for the first segment because the call was transferred to REP after this segment and it will be 'N' for the next segment where it completed in IVR itself.

**Table 14–10**

CallId	NAV PATH ID	Transfer Flag
20060104-C1	MENU1-MENU2-MENU3	Y
20060104-C1	MENU4-MENU5-MENU6	N

The mapping PLP\_IVRNavigation Profile Dimension will load two records in the IVR Navigation Profile Dimension table W\_IVR\_NAV\_PROFL\_D, one for NAV\_PATH\_ID "MENU1-MENU2-MENU3" and "MENU4-MENU5-MENU6" and create unique surrogate key for each of them. If any NAV\_PATH\_ID already exists in the table W\_IVR\_NAV\_PROFL\_D and is Effective on the day when the call was made, that record will not be inserted.

### 14.2.18.2 Slowly Changing Dimension Behavior of IVR Menu and IVR Navigation Profile Dimensions

IVR Menu dimension is configured as a slowly changing dimension. The history of any change in the following attributes will be maintained in the IVR Menu Dimension.

- PROMPT\_MSG
- PROMPT\_MESG\_DURN
- RSPNS\_ERROR\_MSG
- MENU\_BROADCAST\_MSG
- TIMEOUT\_DURN
- TIMEOUT\_MSG
- MENU\_ACTIVE\_FLG

When one or more of the above listed attributes change, the new record is inserted in W\_IVR\_MENU\_D and a new dimension key (ROW\_WID) is assigned to it. This change, in turn, result into a change in all the Navigation Profiles which has this MENU\_WID at any position from 1 to 15th. The history of this change is maintained in the IVR Navigation profile dimension table, W\_IVR\_NAV\_PROFL\_D.

### 14.2.19 How to View Duration Metrics in hh:mm:ss Format in Oracle BI Answers

The Oracle Business Intelligence Applications analytic repository (.rpd) has several duration metric in the "Fact – Service - Contact Center Performance" and "Fact – Service – Rep Activities" subject areas. Most of these duration metrics also have a version that is a pre-formatted number to be displayed in the hh:mm:ss format. In

order to view these duration metrics in hh:mi:ss format in an Oracle BI Answers report, do the following:

1. Select the duration in hh:mi:ss metric using the column selectors in the "Criteria" tab.
2. Click on "Format Column".
3. The Column Properties window will open. Go to Data format.
4. Check "Override Default data format".
5. Select Treat Number as "Custom" and give Custom Numeric Format as "0#:#:#:#".

### **14.2.20 Configuration Steps for Controlling Your Data Set**

Not applicable to Oracle BI Applications Version 7.9.4.

---

---

# Configuring the Oracle BI Applications Repository

This chapter describes how to configure the Oracle BI Repository for the Oracle Business Analytics Warehouse.

It contains the following topics:

- Section 15.1, "Modifying the Oracle BI Repository for Siebel version 6.3 and version 7.5 applications"
- Section 15.2, "Configuring the Oracle BI Repository Connections"
- Section 15.3, "Setting up Date Specific Metrics"
- Section 15.4, "Setting Up Additional Time Series Metrics for Oracle Business Analytics Warehouse"
- Section 15.5, "Setting Up Additional Dimension Tables for Oracle Business Analytics Warehouse"
- Section 15.7, "About Oracle BI Time Repository Variables"
- Section 15.8, "About User Authentication"
- Section 15.9, "About the Security or Visibility Configuration"
- Section 15.10, "About the Group Variable"
- Section 15.6, "About the Period Ago Keys for Oracle Business Analytics Warehouse"
- Section 15.11, "About Configuring Usage Tracking for Oracle Business Analytics Warehouse"
- Section 15.12, "About the Incremental Deployment of the Oracle BI Applications Repository"

---

---

**Note:** When you access the OracleBIAnalyticsApps.rpd repository in Informatica tools, you must log on as the username and password Administrator\SADMIN.

---

---

## 15.1 Modifying the Oracle BI Repository for Siebel version 6.3 and version 7.5 applications

Before you can enable Siebel version 6.3 and version 7.5 applications, you must modify the Oracle BI Applications Repository.

**To modify the Oracle BI Applications Repository for Siebel version 6.3 and version 7.5 applications**

1. Open the file RepositoryDifferences75\_63.xls in the \OracleBI\dwrep folder.
2. Using the Oracle BI Administration Tool, publish the information in the following tabs:
  - ITBlocks
  - LogicalCols
  - FinCols

## 15.2 Configuring the Oracle BI Repository Connections

Oracle BI Applications Repository uses four databases defined in the Physical layer, as follows:

- Oracle Data Warehouse
- Oracle EBS OLTP
- PeopleSoft OLTP
- Siebel OLTP

You need to configure the Oracle BI repository variables and connection pools to connect to your databases, as follows:

- For information about configuring connection pools, see [Section 15.2.2, "How to Configure Oracle BI Applications Repository Connections"](#).
- For information about configuring repository variables, see [Section 15.2.3, "How to Configure Oracle BI Applications Repository Variables"](#).

### 15.2.1 About The Predefined Connection Pools In The Oracle Business Analytics Warehouse

The Oracle Business Analytics Warehouse physical database has two predefined connection pools:

- **Oracle Business Analytics Warehouse Connection Pool.** The Oracle Business Analytics Warehouse Connection Pool is the main connection pool in the Oracle BI Applications Repository. You need to configure this connection pool to connect to your physical data warehouse. The connection is used by the session initialization blocks. You can use this connection pool to set up a dynamic data source name.
- **Oracle Business Analytics Warehouse Repository Initblocks Connection Pool.** You need to configure the Oracle Business Analytics Warehouse Repository Initblocks Connection Pool to connect to your physical data warehouse. The connection is used by the repository level initialization blocks. Repository level initialization blocks cannot be configured to use the dynamic data source name.

You can also set up dynamic data source names, which allow an Administrator to set one instance of Analytics server to connect to different data warehouses depending on the user. For more information about how to set up dynamic data source names, see [Section 15.2.1.1, "How to Configure Dynamic Data Source Names"](#).

You also need to configure the following Static variables:

- **OLAP\_DSN.** The value of the OLAP\_DSN static variable is set to the data source name for the warehouse database.
- **OLAP\_USER.** The value of the OLAP\_USER static variable is set to the database user name for the warehouse database.
- **OLAPTBO.** The value of the OLAPTBO static variable is set to the database table owner for the warehouse database.

The PeopleSoft OLTP, Siebel OLTP, and Oracle EBS OLTP databases have two predefined connection pools each. The actual databases in the RPD will depend on the modules licensed by the customer. The connection pools and their functions within each database are listed below.

- Oracle EBS OLTP:
  - **Oracle EBS OLTP DBAuth Connection Pool.** The Oracle EBS OLTP DBAuth Connection Pool is used if database authentication is required.
  - **Oracle EBS OLTP Connection Pool.** The Oracle EBS OLTP Connection Pool is used to connect to the Oracle EBS OLTP system.

You also need to configure the following Static variables:

- ORA\_EBS\_OLTP\_DSN. The value of the ORA\_EBS\_OLTP\_DSN static variable is set to the data source name for the Oracle EBS OLTP database.
- ORA\_EBS\_OLTP\_USER. The value of the ORA\_EBS\_OLTP\_USER static variable is set to the database user name for the Oracle EBS OLTP database.
- Siebel OLTP
  - **Siebel OLTP DBAuth Connection Pool.** The Siebel OLTP DBAuth Connection Pool is used if database authentication is required.
  - **Siebel OLTP Connection Pool.** The Siebel OLTP Connection Pool is used to connect to the Siebel OLTP system.

You also need to configure the following Static variables:

- OLTP\_DSN. The value of the OLTP\_DSN static variable is set to the data source name for the Siebel OLTP database.
- OLTP\_USER. The value of the OLTP\_USER static variable is set to the database user name for the Siebel OLTP database.

- PeopleSoft OLTP
  - **PeopleSoft OLTP DBAuth Connection Pool.** The PeopleSoft OLTP DBAuth Connection Pool is used if database authentication is required.
  - **PeopleSoft OLTP Connection Pool.** The PeopleSoft OLTP Connection Pool is used to connect to the PeopleSoft OLTP system.

You also need to configure the following Static variables:

- OLTP\_DSN. The value of the OLTP\_DSN static variable is set to the data source name for the PeopleSoft OLTP database.
- OLTP\_USER. The value of the OLTP\_USER static variable is set to the database user name for the PeopleSoft OLTP database.

### 15.2.1.1 How to Configure Dynamic Data Source Names

This sections explains how to create and configure dynamic data source names.

Dynamic data source names allow the Administrator to set one instance of Analytics server to connect to different data warehouses depending on the user. For this you need to have your user authentication based on an external system (like LDAP), and add the following to your repository:

1. Create new session variables: Session\_OLAP\_DSN and Session\_OLAP\_USER
2. Create a Session Init Block which uses "Oracle Business Analytics Warehouse Repository Initblocks Connection Pool" to populate these session variables based on the user login.
3. Add this Initialization Block to the Execution Precedence list of the Authorization Initialization block.
4. Modify the values of Data Source Name and User Name fields in "Oracle Data Warehouse Connection Pool" to be VALUEOF(Session\_OLAP\_DSN) and VALUEOF(Session\_OLAP\_USER) respectively.

For the Password field, you should have the user password the same as the user name.

5. Update the field password with the same value as of User Name.

## 15.2.2 How to Configure Oracle BI Applications Repository Connections

The section explains how to configure the Oracle BI connection physical databases and connection pools to connect to your database by doing the following:

- Configuring the Oracle Data Warehouse, Oracle EBS OLTP, PeopleSoft OLTP, and Siebel OLTP physical databases.
- Configuring the Oracle BI Applications Repository connection pools.

### To configure the Oracle Data Warehouse, Oracle EBS OLTP, PeopleSoft OLTP, or Siebel OLTP physical databases

1. Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd file in the \$SAHome\OracleBI\Server\Repository folder.
2. In the Physical pane, double-click the Oracle Data Warehouse object.
3. In the Database list, click your database type.
4. Save the repository.
5. Click Yes to Check Global Consistency.
6. Repeat steps 1-5 for the Oracle EBS OLTP, PeopleSoft OLTP, and Siebel OLTP data warehouse objects
7. Click OK when the Warnings are displayed.

### To configure the Oracle BI Applications Repository connection pools

1. Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd file in the \$SAHome\OracleBI\Server\Repository folder.
2. In the Physical pane:
  - a. Double-click the Oracle Data Warehouse Connection Pool within the Oracle Data Warehouse.
  - b. Type the database source name in the **Data source name** box.
  - c. Type your database User ID in the **User name** box.
  - d. Type your password in the **Password** box.

3. Repeat Steps a. to d. above for the other connection pools listed above.
4. Save the repository.
5. Click Yes to Check Global Consistency.
6. Click OK when the Warnings are displayed.

### 15.2.3 How to Configure Oracle BI Applications Repository Variables

The section explains how to configure the Oracle BI repository variables.

#### To configure the Oracle BI Applications Repository variables

1. Using the Oracle BI Administration Tool, open the OracleBIAnalyticsApps.rpd file in the \$SAHome\OracleBI\Server\Repository folder.
2. On the Manage menu, click Variables.
3. In the Variables Manager window, click Static.
4. Edit the OLAP\_DSN, OLAP\_USER, OLAPTBO, OLTP\_DSN, OLTP\_USER, ORA\_EBS\_OLTP\_DSN, ORA\_EBS\_OLTP\_USER variables, and close the Variables Manager window.
5. On the Manage menu, click Security
6. In the Security Manager window, click Users, and then:
  - a. Double-click on Administrator user, and type a new password.
  - b. Double-click on SADMIN, and type a new password.
  - c. Save and close the Security Manager.
7. Save the repository.
8. Click Yes to Check Global Consistency.
9. Click OK when the Warnings are displayed.

## 15.3 Setting up Date Specific Metrics

The time dimension in the Oracle BI repository for Oracle Business Analytics Warehouse is a standard or canonical time dimension that links to the important time role in each star schema. The Physical table alias used as a canonical time dimension is W\_DAY\_D\_Common.

If a fact table contains a distinct set of metrics that needs to be reported by different dates, the metadata is organized so that each metric is reported by its causal date.

For example, the Invoice fact table has three metrics called Invoice Amount, Fulfill Amount, and Paid Amount, and each of these metrics need to be reported by the corresponding date—Invoice Date, Fulfill Date, and Payment Date. Additional dates in a fact table that a metric could be queried by are known as Secondary dates. These are available to the end users inside a detailed presentation folder. The detailed presentation folder is typically called the Details folder.

In [Table 15–1](#) each of the metrics reflect the activity related to that event for the entire period, for example, Invoice Amount by Invoice Date, Fulfill Amount by Fulfill date, and Payment Amount by Payment Date.

**Table 15–1 Invoice Fact Table Example**

Date	Invoice Amount	Fulfill Amount	Payment Amount
January	4000	5000	4500

**To implement date specific metrics**

1. Open the OracleBIAnalyticsApps.rpd with the Oracle BI Administration Tool.  
The OracleBIAnalyticsApps.rpd file is located in the \OracleBI\server\Repository folder.
2. Right-click on Oracle Business Analytics Warehouse in the Physical layer, and create a new physical alias for the fact table.
3. Create Joins for the physical alias which are similar to the base fact table.  
The Join to the date dimension is changed to use the date role in question.
4. Create a new logical table source in the logical fact table that maps the metrics for the physical fact alias.  
The grain of the fact table is the same as the base fact table.

---

**Note:** You need to map each metric to one logical table source at the Detail Level.

---

## 15.4 Setting Up Additional Time Series Metrics for Oracle Business Analytics Warehouse

The Oracle BI repository provides a framework to add Period Ago metrics. The repository for Oracle Business Analytics Warehouse is preconfigured with pre-mapped period ago metrics, however you can map other metrics by using the following procedure.

**To set up additional time series metrics**

1. Open the OracleBIAnalyticsApps.rpd with the Oracle BI Administration Tool.
2. Right-click on Oracle Business Analytics Warehouse in the Physical layer, and create a new Period Ago physical alias table.
3. Create additional tables in the Physical Layer for each Period Ago alias required.  
For example, Quarter Ago, Year Ago, and so on.  
These aliases need to have the same joins as the base fact table, except for the date join, which you can change in the next step. Setting up this alias is easier to accomplish by copying the base table.
4. Change the join to the date dimension (W\_DAY\_D) to use the appropriate Period Ago Key.
5. Map the Period Ago metrics in the logical table using the new fact alias by creating a new logical table source under the fact table.
6. Set the content pane levels for the period ago logical table source, to specify the level of the source data.  
These settings are the same as the base fact table.
7. Save and close the OracleBIAnalyticsApps.rpd file.

## 15.5 Setting Up Additional Dimension Tables for Oracle Business Analytics Warehouse

Oracle Business Analytics Warehouse is preconfigured to map dimension tables required for analysis. The physical layer in the Oracle BI repository provides several other dimensional table keys that can be used for certain specific analysis. If you need to set up any of the additional dimensions tables to the physical layer, perform the following procedure.

### To set up additional dimension tables

1. Validate that the dimension table key is resolved appropriately for the data source that you are using.

---

---

**Note:** Dimension tables do not apply to every source system.

---

---

2. Open the OracleBIAnalyticsApps.rpd with the Oracle BI Administration Tool.
3. Add a dimension table alias in the physical layer.
4. Join the dimension table alias to the fact table alias using the appropriate keys.
5. Save and close the OracleBIAnalyticsApps.rpd file.

## 15.6 About the Period Ago Keys for Oracle Business Analytics Warehouse

The Period Ago Key fields are used to set up the time series metrics like Year Ago, Quarter Ago, and so on. The Period Ago Key fields represent metrics for a prior period, for example, Quarter Ago Revenue, Year Ago Revenue, and so on. Oracle Business Analytics Warehouse is preconfigured with a set of fields in the W\_DAY\_D table. These fields are:

- MONTH\_AGO\_WID
- QUARTER\_AGO\_WID
- TRIMESTER\_AGO\_WID
- WEEK\_AGO\_WID
- YEAR\_AGO\_WID

These fields are used in joins to Oracle Business Analytics Warehouse fact tables to achieve the period ago metrics. The joins in Oracle Business Analytics Warehouse uses the Period Ago fields in the W\_DAY\_D table.

## 15.7 About Oracle BI Time Repository Variables

The Oracle BI repository is preconfigured with variables that are used for both reporting and internal usage.

[Table 15–2](#) lists the Oracle BI repository date variables and their descriptions.

**Table 15–2 Oracle BI Repository Date Variables**

<b>Variable Name</b>	<b>Description</b>
CAL_MONTH_YEAR_AGO	Returns the value of Previous Year Month in the YYYY/MM format.
CURRENT_BALANCE_DK_AP	Returns the value of the last date key for the available Accounts Payable balance. It is used in Accounts Payable Account Balance Computation.
CURRENT_BALANCE_DK_AR	Returns the value of the last date key for the available Accounts Receivables balance. It is used in Accounts Receivable Account Balance Computation.
CURRENT_BALANCE_DK_GL	Returns the value of the last date key for the available General Ledger balance. It is used in General Ledger Account Balance Computation.
CURRENT_DAY	Returns the value of Current Date in the MM/DD/YYYY format.
CURRENT_FSCL_MONTH	Returns the value of Current Fiscal Month in the YYYY/MM format.
CURRENT_FSCL_QUARTER	Returns the value of Current Quarter in the YYYY Q n format.
CURRENT_FSCL_WEEK	Returns the value of Current Fiscal Week in the YYYY Week nn format.
CURRENT_FSCL_YEAR	Returns the value of Current Fiscal Year in the FYYYYY format.
CURRENT_JULIAN_DAY_NUM	Returns the value of Current Julian Date Number.
CURRENT_MONTH	Returns the value of Current Month in the YYYY/MM format.
CURRENT_QTR	Returns the value of Current Quarter in YYYY Q n format.
CURRENT_WEEK	Returns the value of Current Week in the YYYY Week nn format.
CURRENT_YEAR	Returns the value of Current Year in the YYYY format.
FSCL_MONTH_YEAR_AGO	Returns the value of Previous Year Fiscal Month in YYYY/MM format.
FSCL_QTR_YEAR_AGO	Returns the value of Previous Year Quarter in YYYY Q n format.
NEXT_FSCL_MONTH	Returns the value of Next Fiscal Month in the YYYY / MM format.
NEXT_FSCL_QUARTER	Returns the value of Next Quarter in the YYYY Q n.
NEXT_FSCL_WEEK	Returns the value of Next Fiscal Week in the YYYY Weeknn format.
NEXT_FSCL_YEAR	Returns the value of Next Fiscal Year in the FYYYYY format.
NEXT_MONTH	Returns the value of Next Month in the YYYY / MM format.
NEXT_QUARTER	Returns the value of Next Quarter in the YYYY Q n.
NEXT_WEEK	Returns the value of Next Week in the YYYY Weeknn format.
NEXT_YEAR	Returns the value of Next Year in the YYYY format.
PREVIOUS_FSCL_MONTH	Returns the value of Previous Fiscal Month in the YYYY/MM format.
PREVIOUS_FSCL_QUARTER	Returns the value of Previous Quarter in the YYYY Q n format.

**Table 15–2 (Cont.) Oracle BI Repository Date Variables**

Variable Name	Description
PREVIOUS_FSCL_WEEK	Returns the value of Previous Fiscal Week in the YYYY Weeknn format.
PREVIOUS_FSCL_YEAR	Returns the value of Previous Fiscal Year in the FYYYYY format.
PREVIOUS_MONTH	Returns the value of Previous Month in the YYYY/MM format.
PREVIOUS_QUARTER	Returns the value of Previous Quarter in the YYYY Q n.
PREVIOUS_WEEK	Returns the value of Previous Week in the YYYY Weeknn format.
PREVIOUS_YEAR	Returns the value of Previous Year in the YYYY format.
REF_JULIAN_DATE	Stores the start date of the Julian calendar and should not be changed.
REF_JULIAN_DATE_NUM	Stores the Julian number for the start of the Julian calendar and should not be changed.
TIME_OFFSET	Returns the difference between the current date and a given number of days value. It is primarily used for testing to simulate an earlier or later date. You could set the variable to the number of days you want the preceding date variables to be moved back.
YEAR_AGO_DAY	Returns the value of year ago date in the mm/dd/yyyy format.

## 15.8 About User Authentication

You need to configure the user authentication of the Oracle Business Analytics Warehouse repository depending on your requirements. The Oracle Business Analytics Warehouse supports various authentication modes, for example, Repository authentication, Database authentication, and LDAP.

For more information about configuring user authentication, see Oracle Business Intelligence Server Administration Guide.

## 15.9 About the Security or Visibility Configuration

The Oracle Business Analytics Warehouse repository is preconfigured with a set of user groups. These groups control the visibility of catalogs in the presentation layer.

For more information on adding a user to repository user group, see Oracle Business Intelligence Server Administration Guide.

[Table 15–3](#) lists the groups in the Oracle Business Analytics Warehouse repository.

**Table 15–3 Repository User Groups**

Repository User Group	Description
Administrators	The Administrators user group has all rights and privileges. It cannot be removed.
Agent Scorecard User	This user group is able to view Agent Scorecard application content.
AP Analyst	This user group is able to view application content for Oracle Payables Analytics.

**Table 15–3 (Cont.) Repository User Groups**

<b>Repository User Group</b>	<b>Description</b>
AP Manager	This user group is able to view high-level application content for Oracle Payables Analytics.
AR Analyst	This user group is able to view application content for Oracle Receivables Analytics.
AR Manager	This user group is able to view high-level application content for Oracle Receivables Analytics.
CFO	This user group is able to view most of the Oracle Financial Analytics application content.
Contact Center and Agent Performance Analyst	This user group is able to view Contact Center Telephony Analytics and Agent Performance application content.
Contact Center and Agent Performance User	This user group is able to view a subset of Contact Center Telephony Analytics and Agent Performance application content.
Contact Center Sales Analyst	This user group is able to view Contact Center Telephony Analytics and Order Management Analytics application content.
Contact Center Sales User	This user group is able to view a subset of Contact Center Telephony Analytics and Order Management Analytics application content.
Controller	This user group is able to view application content for Oracle General Ledger and Profitability Analytics and Siebel Profitability Analytics.
Customer Service Analyst	This user group is able to view Customer Service for Oracle Contact Center Telephony Analytics application content.
Customer Service User	This user group is able to view a subset of Customer Service for Oracle BI Contact Center Telephony Analytics application content.
Contact Center Telephony Analytics User	This user group is able to view Oracle BI Contact Center Telephony Analytics application content.
Financial Analyst	This user group is able to view Oracle Financial Analytics application content.
Human Resources Analyst	This user group is able to view Oracle HR Analytics application content.
Human Resources Vice President	This user group is able to view high-level application content for Oracle HR Analytics application.
Inventory Analyst	This user group is able to view application content for Oracle's Supply Chain Analytics family of products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics).
Inventory Manager	This user group is able to view high-level application content for Oracle's Supply Chain Analytics family of products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics).
Primary Owner-Based Security	Used for securing owner-based data elements that come from the transactional system.
Primary Position-Based Security	Used for securing position-based data elements that come from the transactional system.

**Table 15–3 (Cont.) Repository User Groups**

<b>Repository User Group</b>	<b>Description</b>
Purchasing Buyer	This user group is able to view Oracle's Supply Chain Analytics family of products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) content pertaining to purchasing.
Sales Executive Analytics	This user group is able to view high-level application content for the Order Management Analytics application.
Sales Manager	This user group is able to view most of the high-level application content for Oracle BI Contact Center Telephony Analytics application.
Sales Manager Analytics	This user group is able to view most of the high-level application content for Oracle BI Contact Center Telephony Analytics application.
Sales Operations Analytics	This user group is able to view operational application content for Oracle BI Contact Center Telephony Analytics application.
Sales Representative Analytics	This user group is able to view low-level application content for Oracle BI Contact Center Telephony Analytics application.
Sales Rev and Fulfill Analyst	This user group is able to view the content for Oracle BI Contact Center Telephony Analytics Revenue and Fulfillment application.
Sales Rev and Fulfill Exec	This user group is able to view the high-level application content for Oracle BI Contact Center Telephony Analytics Revenue and Fulfillment application.
Sales Rev and Fulfill Mgr	This user group is able to view most of the high-level application content for Oracle BI Contact Center Telephony Analytics Revenue and Fulfillment application.
Sales Rev and Fulfill Rep	This user group is able to view low-level application content for Order Management Analytics Revenue and Fulfillment application.
Sales Revenue Analyst	This user group is able to view the content for Oracle BI Contact Center Telephony Analytics Revenue application.
Sales Revenue Exec	This user group is able to view the high-level application content for Oracle BI Contact Center Telephony Analytics Revenue application.
Sales Revenue Mgr	This user group is able to view most of the high-level application content for Oracle BI Contact Center Telephony Analytics Revenue application.
Sales Revenue Rep	This user group is able to view low-level application content for Oracle BI Contact Center Telephony Analytics Revenue application.
Service Delivery and Costs Analyst	This user group is able to view Service Delivery and Costs for Oracle BI Contact Center Telephony Analytics application content.
Service Delivery and Costs User	This user group is able to view a subset of Service Delivery and Costs for Oracle BI Contact Center Telephony Analytics application content.
Supplier Performance Analyst	This user group is able to view Oracle's Supply Chain Analytics family of products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) content pertaining to supplier performance.
Supplier Performance Manager	This user group is able to view high-level content for Oracle's Supply Chain Analytics family of products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) content pertaining to supplier performance.

**Table 15–3 (Cont.) Repository User Groups**

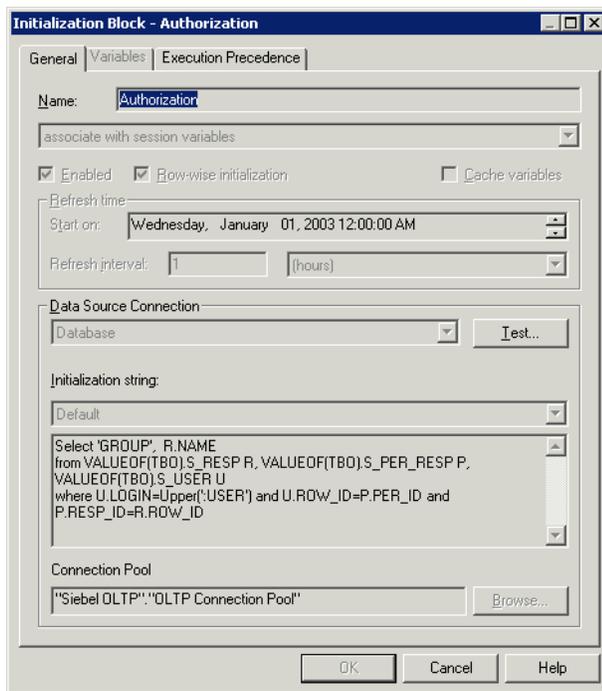
Repository User Group	Description
Supply Chain Executive	This user group is able to view Oracle's Supply Chain Analytics family of products (Oracle Inventory Analytics, Oracle Procurement and Spend Analytics, Oracle Supplier Performance Analytics) content.

## 15.10 About the Group Variable

The Group variable determines the membership of a user in the various security groups. You need to associate users to the appropriate groups defined in the OracleBIAnalyticsApps.rpd for the security filters to take effect.

If you are using the Oracle BI Applications with the Siebel CRM system, then you can leverage the Authorization session init block to populate the Group variable with the appropriate Siebel Responsibilities, which associates users to what they are allowed to see in the Oracle BI Application by using their Siebel Responsibility. For more information about configuring the Group variable, see *Oracle Business Intelligence Presentation Services Administration Guide*. The screen shot below shows an example of an initialization block that associates a user to a Group membership.

**Figure 15–1 Screenshot of the Initialization Block - Authorization screen**



## 15.11 About Configuring Usage Tracking for Oracle Business Analytics Warehouse

Oracle Business Analytics Warehouse supports the accumulation of usage tracking statistics. The Oracle BI repository for Oracle Business Analytics Warehouse is preconfigured with a connection pool to enable the population of the Usage Tracking log table.

You need to configure this connection pool to connect to the S\_NQ\_ACCT table. For more information the Usage Tracking application administering Usage Tracking, see the Oracle Business Intelligence Server Administration Guide.

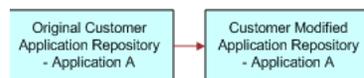
## 15.12 About the Incremental Deployment of the Oracle BI Applications Repository

Oracle Business Analytics Warehouse consist of various families of Oracle BI Applications, for example, Supplier Performance Analytics, Contact Center Telephony Analytics, General Ledger & Profitability Analytics, and so on. You can purchase these applications at different times. You can customize functionality and incrementally add new application or applications.

This section describes the procedure for deploying multiple applications. You can repeat the procedure to add applications incrementally.

The figure below shows a single Oracle Business Analytics Warehouse application environment. During installation, you will be asked to specify the application module(s) you have licensed, and the installer will extract the metadata project(s) corresponding to this module(s) into one repository file. You can then modify the Oracle BI repository to suit your business needs.

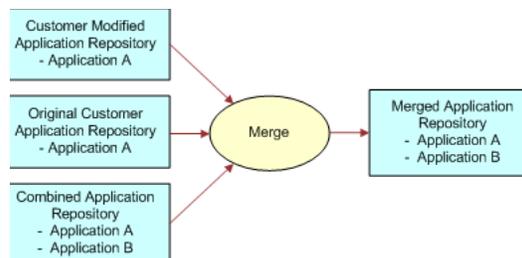
**Figure 15–2 Oracle Business Analytics Warehouse environment**



When you purchase another Oracle Business Analytics Warehouse application, you need to extract a new application repository, containing the metadata for all the modules that you have licensed. Use the Administration merge utility to perform a three-way merge of the original repository, the modified repository, and the combined repository. For more information on merging repositories, see *Oracle Business Intelligence Server Administration Guide*.

The merged repository preserves your modifications from the original Oracle BI repository and appends the information with the new Oracle BI repository, as shown in the figure below.

**Figure 15–3 Merging with an Oracle BI Applications Repository**



You can repeat this merging procedure to add more Oracle Business Analytics Warehouse applications to the Oracle BI repository



# Part V

---

---

## Appendices

Part V contains the following chapters:

- [Appendix A, "Installing the Usage Accelerator Application"](#)
- [Appendix B, "Localizing Oracle Business Intelligence Deployments"](#)
- [Appendix C, "Integrating Interactive Dashboards and Operational Applications Data"](#)
- [Appendix D, "Configuring Metadata for Analytics Applications"](#)
- [Appendix E, "Oracle Business Analytics Warehouse for Life Sciences Data Considerations"](#)
- [Appendix F, "Pharma Disconnected Analytics Administration - Supplemental Information"](#)
- [Appendix G, "Using Oracle Business Analytics Warehouse Exception Reports"](#)
- [Appendix H, "Troubleshooting Oracle Business Intelligence Applications"](#)

---

---

**Note:** For a high level road map for installation, configuration, and customization steps for Oracle BI Applications, see [Section 2.4, "Roadmap To Installing, Configuring, and Customizing Oracle BI Applications"](#).

---

---



---

---

# Installing the Usage Accelerator Application

This appendix provides instructions for installing the optional Usage Accelerator application.

This appendix contains the following topics:

- [Section A.1, "What is Usage Accelerator?"](#)
- [Section A.2, "About Installing the Usage Accelerator"](#)
- [Section A.3, "Usage Accelerator Application Integration with Siebel 7.5.3 CRM Application"](#)
- [Section A.4, "Usage Accelerator-Analytics Adapter Files"](#)

## A.1 What is Usage Accelerator?

Usage Accelerator was developed to enable sales organizations to measure and manage user adoption and effectiveness of Oracle's Siebel Sales. Every level of the sales organization benefits from the richer understanding of how each member of the team is contributing to the overall objective. Usage Accelerator provides a comprehensive set of management dashboards and pre-built metrics that deliver rich information to help accelerate organizational usage, drive alignment and increase the overall ROI of CRM implementations.

### A.1.1 About Configuring Usage Accelerator

This section explains how Usage Accelerator is configured.

#### A.1.1.1 Tracking of Targeted Users

In order to optimize the storage space required for the application data the Usage Accelerator has been designed to let you choose which users you want to measure. The Usage Accelerator determines the users that you want to measure through the use of a new responsibility called "Usage Accelerator – Tracking". This new responsibility will need to be associated to each user in the operational Siebel application.

#### A.1.1.2 Other Required Responsibilities

The Usage Accelerator application introduces eight new responsibilities. Four of the responsibilities will be used by all organizations except for FIN's customers. FIN's customers will use four responsibilities that are targeted specifically for their user groups. The responsibilities determine which dashboards a user will see when they use the Usage Accelerator application. These responsibilities need to be assigned to users in the operational application. The responsibilities are:

**Table A-1 User Accelerator Responsibilities**

Usage Accelerator Responsibilities	Usage Accelerator FIN's Responsibilities
Usage Accelerator – Sales Rep	Usage Accelerator - FINS Sales Rep
Usage Accelerator - Sales Manager	Usage Accelerator – FINS Sales Manager
Usage Accelerator - Sales Executive	Usage Accelerator – FINS Sales Executive
Usage Accelerator – Administrator	Usage Accelerator – FINS Administrator

---



---

**Note:** You must not remove a Responsibility for a User and later add the same Responsibility for that User in Siebel CRM Administration View "Responsibilities". If you do so, the related ETL will fail (for more information, see [Section H.7, "ETL Issues"](#)).

---



---

## A.1.2 About Deployment Date

Oracle’s Usage Accelerator Application needs to be calibrated with the Deployment Date variable UA\_DEPLOY\_DT, which determines the time that you want the Analysis to commence. This date will default to the date on which you create the Usage Accelerator Data Warehouse, but can be modified by changing the UA\_DEPLOY\_DT variable.

**Note:** If you set the Deployment Date earlier than the current calendar date, then metrics that rely on the Last Updated Date (e.g # of Updated Records, # of Logins, # of Sync’s, etc.) of a record might be inaccurate. For this reason Oracle recommends setting the Deployment Date to the current Calendar Date.

## A.1.3 About Metric Definitions and Variables

Usage Accelerator tracks user adoption and effectiveness metrics across the following metric categories:

- User Adoption Metrics
- Opportunity Metrics
- Information Completeness Metrics
- Coverage Metrics

### A.1.3.1 User Adoption Metrics

User Adoption Metrics provide visibility into the behavioral attributes and usage patterns of those using the system. These metrics enable you to quickly identify areas in need of improvement and individuals who require additional help to meet the company’s CRM standards.

**Table A–2 User Adoption Metrics**

<b>Metric Category</b>	<b>Description</b>	<b>Variables</b>
Active User	Identifies users that have not logged in or synced in the last seven day time period.	Number of logins needed is currently set to 2. Variable Name is: ACTIVE_NUMLOGINS. Number of Syncs needed is currently set to 0. Variable Name is: ACTIVE_NUMSYNCS.
Login	Measures the number of days the user's login. Note – Only one login a day will be captured for each user.	Not Applicable
Sync	Measures the number of days the user's sync. Note – Only one sync a day will be captured for each user.	Not Applicable
Records Created	Tracks the number of Accounts, Contacts, Opportunities, Activities, Quotes created by each user and group. FINS version also tracks number of Financial Accounts and Policies Created.	Not Applicable
Records Updated	Tracks the number of Accounts, Contacts, Opportunities, Activities, Quotes updated by each user and group. FINS version also tracks number of Financial Accounts and Policies Created.	Not Applicable

### A.1.3.2 Opportunity Updates Metrics

Opportunity Updates Metrics deliver reliable information to organizations that use Opportunity Management for tracking their pipeline and sales forecasts. The metrics provide a focus on opportunities by ensuring that current and accurate information is in place.

**Table A–3 Opportunity Updates Metrics**

<b>Metric Category</b>	<b>Description</b>	<b>Variables</b>
Opportunity Updates	Identifies Opportunities that have not been modified in the time period specified or have a close date that has passed. Opportunity Updates Needed is currently set to 90 Days.	Variable Name is: OPT_OUTDATED_NUMDAYS

### A.1.3.3 Information Completeness Metrics

Information Completeness Metrics assist in the completion of contact and account information by highlighting what is missing in an individual entry or in a set of entries.

**Table A-4 Information Completeness Metrics**

Metric Category	Description	Variables
Account Completeness	Looks for the presence of information in each of the primary Account Address fields.	Address Line 1, City, State, Zipcode, Phone
Contact Completeness	Looks for the presence of information in each of the primary Contact Address fields.	Address Line 1, City, State, Zipcode, Work Phone, Email Address, Title

### A.1.3.4 Coverage Metrics

Coverage Metrics monitor sales activity against Accounts, Contacts, Opportunities and Financial Accounts. It is not envisioned that you will need to use all of these metrics just the ones that are in alignment with your sales process.

**Table A-5 Coverage Metrics**

Metric Category	Description	Variables
Account Coverage	Determines whether an account has had a contact and activity associated to it in the timeframe specified. Activity Needed is currently set to 90 Days.	Variable name is: ACC_NEED_ACT_NUMDAYS. Contact Needed is currently set to 180 Days. Variable name is: ACC_NEED_CON_NUMDAYS.
Contact Coverage	Determines whether a Contact has had an activity associated in the time frame specified. Activity Needed is currently set to 90 Days.	Variable Name is: CON_NEED_ACT_NUMDAYS.
Opportunity Coverage	Determines whether an Opportunity has had a contact and activity associated to it in the timeframe specified. Activity Needed is currently set to 90 Days.	Activity Needed is currently set to 90 Days. Variable name is: OPT_NEED_ACT_NUMDAYS. Contact Needed is currently set to 180 Days. Variable name is: OPT_NEED_CON_NUMDAYS.
Financial Account Coverage & Insurance Policy Coverage (FINS)	Determines whether a Financial Account or Insurance Policy has had an Activity Associated to it in the time frame specified	Activity Needed is currently set to 90 Days. Variable name is: AST_NEED_ACT_NUMDAYS.

## A.2 About Installing the Usage Accelerator

If an execution plan in DAC metadata contains a Usage Accelerator Subject Area, it should include a preceding task "Create View For Usage Accelerator". All out of the box execution plans that contain Usage Accelerator subject area are configured with this preceding task. This step is required if you want to create a new execution plan or modify an existing execution plan to include "Usage Accelerator Subject area". This step is also required if you are using any customized subject area containing Usage accelerator warehouse tables for your execution plans. These tables are prefixed with "W\_UA\*".

---

---

**Note:** The following instructions in this appendix are required only if you are planning to use the Usage Accelerator analytics application integrated with Siebel CRM 7.5.3 applications. If you licensed the Usage Accelerator application versions 7.7 or 7.7.1, you should follow the installation instructions in Oracle Business Intelligence Platform Installation and Configuration Guide. No additional steps are required to install these versions of the Usage Accelerator application.

---

---

## A.3 Usage Accelerator Application Integration with Siebel 7.5.3 CRM Application

---

---

**Note:** The following steps are required only if you are planning to use the Usage Accelerator in an integrated environment with a Siebel Business application version 7.5.3.

---

---

The Usage Accelerator installation process consists of installing the Usage Accelerator application and importing seed data into the transactional database. You install the Usage Accelerator application using the standard Oracle Business Intelligence installer. The installation options that are available to you depend on the license key you enter.

### To integrate the Usage Accelerator application

1. Import repository objects to enable embedded Analytics.

Use Siebel tools to imports "SRW\_753\_UA\_Integration\_objects.sif" file into tools repository.

2. Lock the following projects:
  - Analytics
  - ERM
  - Siebel Field Service
  - Siebel Marketing Enterprise
  - Siebel Channel
  - Siebel Sales Enterprise
  - Siebel Service Enterprise
  - Siebel Universal Agent

---

---

**Note:** Do not apply the schema in the transactional database after importing these objects in tools repository. You might see some tables created in this step. These tables would have been applied in the transactional database.

---

---

3. Recompile the srf file, as follows:
  - a. In Windows Explorer, navigate to \tools\objects\enu, and create a copy of siebel.srf under a different name.

- b. In Oracle's Siebel Tools, navigate to Tools, then Compile Projects, and select the Locked Projects radio button.
- c. Under the Siebel Repository, select the new SRF file you created in Step a, and click Compile.

To see the Usage Accelerator screen in your application, the new SRF file must replace the siebel.srf file you were using.

4. Confirm the Analytics Server name to run Embedded Analytics.
  - a. Log in to the Siebel application as the administrator.
  - b. Navigate to Site Map, then Administration - Integration, then WI Symbolic URL List.
  - c. In the Administration - Integration screen, from the visibility filter, select Host Administration.
  - d. In the Host Administration list, locate the record Name=[AnalyticsServerName].
  - e. Change [AnalyticsServerName] to the name of the machine hosting your Analytics Server.
5. Import language-independent Siebel seed data, as follows:
  - a. Copy the appropriate seed data files:

**Table A-6 Seed data files**

File name	From	To
seed_753_UA.dat	OracleBI\dwrep	Local machine
seed_753_<xxx>_UA.inp where xxx corresponds to the database platform you are using	OracleBI\dwrep	Local machine

- b. Open a command line, and go to the same directory where you copied the .dat and .inp files.
- c. Replace each of the connection parameters with the values appropriate to your database environment, and then run the appropriate import command from the following table:

**Table A-7 Import commands**

Database Platform	Run Import Command
Oracle	<code>\$SIEBELSERVERROOT\bin\dataimp /u \$UserName /p Oracle \$Password /c "\$ODBCDataSource" /d \$DatabaseOwner /f seed_753_UA.dat /w y /q 100 /h Log /x f /i seed_753_ora_UA.inp /lseed_753_ora_UA.log</code>
DB2UDB	<code>\$SIEBELSERVERROOT\bin\dataimp /u \$UserName /p \$Password /c "\$ODBCDataSource" /d \$DatabaseOwner /f seed_753_UA.dat /w y /q 100 /h Log /x f /i seed_753_db2_UA.inp /lseed_753_db2_UA.log</code>
MSSQL	<code>\$SIEBELSERVERROOT\bin\dataimp /u \$UserName /p \$Password /c "\$ODBCDataSource" /d \$DatabaseOwner /f seed_753_UA.dat /w y /q 100 /h Log /x f /i seed_753_mssql_UA.inp /lseed_753_mssql_UA.log</code>

6. Associate the new Usage Accelerator responsibilities with the users whom you want to see the Usage Accelerator screen.

The Usage Accelerator responsibilities are as follows:

- Usage Accelerator Tracking
- Usage Accelerator - Sales Rep
- Usage Accelerator - Sales Manager
- Usage Accelerator - Sales Executive
- Usage Accelerator – Administrator

---



---

**Note:** The first responsibility has to be associated with any transaction application user whom you want to be tracked in the Usage Accelerator reports.

---



---

## A.4 Usage Accelerator-Analytics Adapter Files

The contents of each of the files included in the OracleBI\dwrep folder are shown in the table below.

**Table A-8 Usage Accelerator Analytics Adapter Files**

File	Description
seed_753_UA.dat	Siebel Seed Data file containing language- independent seed data records for the following tables: S_WI_SYMURL, S_WE_SYMURL_ARG, S_RESP, S_APP_VIEW, S_APP_VIEW_RESP.
seed_753_db2_UA.inp	Import file used to control the import of the seed_753_UA.dat file to a particular database platform.
seed_753_mssql_UA.inp	
seed_753_ora_UA.inp	



---

---

## Localizing Oracle Business Intelligence Deployments

Oracle Business Intelligence is designed to allow users to dynamically change their preferred language and locale preferences. This chapter contains the following topics on how to configure Oracle Business Intelligence Applications for deployment in one or more language environments besides English:

- [Section B.1, "Process of Maintaining Translation Tables for Oracle BI"](#)
- [Section B.2, "About Translating Presentation Services Strings"](#)
- [Section B.3, "Changing the Default Currency in Analytics Applications"](#)

### B.1 Process of Maintaining Translation Tables for Oracle BI

The Oracle Business Intelligence Presentation layer supports multiple translations for any column name. When working with Oracle BI Answers or rendering a dashboard, users see their local language strings in their reports. For example, English-speaking and French-speaking users would see their local language strings in their reports. There are two kinds of application strings requiring translation in Oracle Business Intelligence:

- Metadata

Metadata strings are Analytics-created objects in the Oracle Business Intelligence repository such as Subject Area, Metrics, and Dimensions.

- Presentation Services

Presentation Services objects are end-user created objects such as Reports, Dashboards, and Pages. Translations for Presentation Services strings are stored in the captions.xml file. For more information on accessing these strings and changing the translations, see Oracle Business Intelligence Web Administration Guide.

This process includes the following tasks:

- [Section B.1.1, "Upgrading Oracle Business Intelligence Seed Data for Non-English Locales"](#)
- [Section B.1.2, "Externalizing Customer Metadata Strings"](#)
- [Section B.1.3, "Adding Custom Translations to the W\\_LOCALIZED\\_STRING\\_G Table"](#)

## B.1.1 Upgrading Oracle Business Intelligence Seed Data for Non-English Locales

If Oracle Business Intelligence data in your deployment is to be viewed in a language other than English, you must also import Locale seed data into a data warehouse table called `W_LOCALIZED_STRING_G`. This process must be performed once for each language the application users might select to run their web client.

During the Oracle Business Intelligence installation, a folder named `$INSTALLDIR\SeedData` was created, which contains a sub folder for each language. Within each language sub folder is a `.dat` file (the data to be imported) and an `.inp` file (the `WHERE` clause governing the import).

### B.1.1.1 Importing Locale Seed Data Into The Translation Table (`W_LOCALIZED_STRING_G`)

If the primary language being used is not English, you may have to import additional locale seed data (depending on the number of languages you use) as shown in the following procedures.

---

---

**Note:** This procedure can be performed only by a BI Administrator.

---

---

#### To verify creation of Translation Table (`W_LOCALIZED_STRING_G`) and corresponding indexes:

1. Verify that in the setting up of the Business Analytics Warehouse, tables included the creation of the `W_LOCALIZED_STRING_G` table (see page 105, Process of Creating Data Warehouse Tables).
2. Lookup the definitions of the indexes in the DAC and create them manually in the Business Analytics Warehouse. The names of the indexes are as follows:
  - `W_LOCAL_STRING_G_U1`
  - `W_LOCAL_STRING_G_P1`
  - `W_LOCAL_STRING_G_M1`
  - `W_LOCAL_STRING_G_M2`

**Note:** It is better to add these indexes to `W_LOCALIZED_STRING_G` prior to importing the locale seed data in the next section, in order to safeguard against inadvertently duplicating the data in the table.

#### To import Locale seed data into the Translation Table (`W_LOCALIZED_STRING_G`)

1. Open a command window and navigate to `$INSTALLDIR\SeedData\bin` folder.
2. Run the import command in step 3 after replacing these connection parameters with the values appropriate to your database environment:
  - `UserName`
  - `Password`
  - `ODBCDataSource`
  - `DatabaseOwner`
3. Run the import command:

```
$INSTALLDIR\SeedData\bin\dataimp /u $UserName /p $Password /c "$ODBCDataSource"
```

```
/d $DatabaseOwner /f $INSTALLDIR\SeedData\<>XX>\analytics_seed_<XXX>.dat /w y /q
100 /h Log /x f /i $INSTALLDIR\SeedData\<>XX>\metadata_upgrade_<XXX>_
<DBPlatform>.inp /l metadata_upgrade_<XXX>.log
```

---

**Note:** Replace the XX with the Oracle Business Intelligence two-letter language code (\_fr, \_it) and the XXX with the Siebel Systems three-letter code (FRA, ITA).

---

4. When you have finished importing the Locale seed data into the Translation Table (W\_LOCALIZED\_STRING\_G), configure the Initialization block in the Oracle BI Administration tool to connect to the database where this table resides.

---

**Note:** Unicode connectivity can be used to access databases that do not support Unicode.

---

## B.1.2 Externalizing Customer Metadata Strings

Metadata Strings are loaded by the Oracle BI Server from a database table. In the case of Oracle Business Intelligence applications, this table is W\_LOCALIZED\_STRING\_G in the data warehouse. The initialization block 'Externalize Metadata Strings' loads the strings for the Server. It is recommended that you run a test to make sure that this initialization block runs successfully. An example of the translation table is shown in [Table B-1](#).

**Table B-1** Example of W\_LOCALIZED\_STRING\_G Translation Table

MSG_NUM	MSG_TEXT	LANG_ID
CN_Customer_Satisfaction	Customer Satisfaction	ENU
CN_Customer_Satisfaction	Kundenzufriedenheit	DEU
CN_Customer_Satisfaction	Satisfação do cliente	PTB

By default, the Oracle Business Intelligence repository is configured to run in English only. To deploy in any other language, you must externalize the metadata strings, as described in the following procedure.

### To externalize metadata strings in the Oracle Business Intelligence repository

1. Stop the Oracle BI Server.
2. Open OracleBIAnalyticsApps.rpd using the Oracle BI Administration Tool in offline mode.
3. Select the entire Presentation layer and right-click the mouse to display the menu.
  - From the pop-up menu, select Externalize Display Names. (A check mark appears next to this option the next time you right-click on the Presentation layer.)
  - Unselect the Presentation layer.

---

**Note:** When Externalize Display Names is checked, all metadata strings are read from the W\_LOCALIZED\_STRING\_G table in the data warehouse.

---

4. In the Physical layer, select the Externalized Metadata Strings database icon. Expand the tree.
5. Double-click Internal System Connection Pool.  
In the Connection Pool window General tab, the field Data source name should point to the data warehouse.
6. Click OK and exit the Oracle BI Administration Tool.
7. Restart the Oracle BI Server.

### B.1.3 Adding Custom Translations to the W\_LOCALIZED\_STRING\_G Table

When you add custom objects to the metadata and choose to externalize these objects (by right-clicking the object and checking the Externalize Display Name option), the Oracle BI Server looks for the translations (including those for the native language) in the W\_LOCALIZED\_STRING\_G table.

If you do not externalize the display names, you do not need to perform the following procedures.

---

---

**Note:** The custom Presentation layer objects show up only in the native language of the metadata (the language in which you added these new objects).

---

---

#### B.1.3.1 Adding String Translations for Analytics Metadata

The following procedure describes how to add string translations for Oracle Business Intelligence metadata to the W\_LOCALIZED\_STRING\_G table. This task occurs in any database administration tool, and in the Oracle BI Administration Tool.

##### To add string translations for Analytics metadata

1. Open a database administration tool and connect to your data warehouse database.
2. Query for the table named W\_LOCALIZED\_STRING\_G and add a new record to the table, as defined below in steps 4 to 8.
3. Obtain the Message Key from the Oracle BI Administration Tool as follows:
  - In the Oracle BI Administration Tool, right-click on the new Presentation layer metadata object and select Properties from the menu.
  - The Message key is displayed in the dialog box under Custom Display Name. The Message key is the part that starts with CN\_.

For example, double-click the Pipeline catalog folder in the Presentation layer. The Custom Display name is Valueof(NQ\_SESSION.CN\_Pipeline). CN\_Pipeline is the Message Key.
4. Enter your deployment language in the new record.
5. Enter the Message Type required (for example, Metadata, FINS\_Metadata).
6. Select the Message Level *AnalyticsNew*, then do the following:
  - In the Message Text column, add the translation of the object.
  - Check the flags (set to Yes) for the Translate and Active columns.
  - Set the Error Message # column to 0.

7. Enter the required Message Facility (for example, HMF, FIN).
8. Repeat Step 3 through Step 7 for each new metadata object string.
9. Exit the database administration tool, then restart the Oracle BI Server.

## B.2 About Translating Presentation Services Strings

The translations for such Presentation Services objects as report and page names are stored in the xxxCaptions.xml files available in OracleBIData\web\res\<language abbreviation>\Captions directories. In multiple language deployment mode, if you add any additional Presentation Services objects, such as reports and new dashboard pages, you also need to add the appropriate translations. Add these translations using the Catalog Manager tool. For more information on using this utility, see Oracle Business Intelligence Web Administration Guide.

## B.3 Changing the Default Currency in Analytics Applications

In Oracle Business Intelligence Applications, you may see a dollar sign used as the default symbol when amounts of money are displayed. In order to change this behavior, you must edit the currencies.xml file using the following procedure. The currencies.xml file is located in the following directories:

- Windows:  
\$INSTALLDIR\OracleBI\Web\config\
- UNIX:  
\$INSTALLDIR/OracleBI/web/config

### To change the default currency in Analytics Applications

1. In a text editor, open the currencies.xml file.
2. Look for the currency tag for the warehouse default (tag="int:wrhs"):

```
<Currency tag="int:wrhs" type="international" symbol="$" format="$#" digits="2"
displayMessage="kmsgCurrencySiebelWarehouse">
  <negative tag="minus" format="-$#" />
</Currency>
```

3. Replace the symbol, format, digits and negative information in the warehouse default with the information from the currency tag you want to use as the default.

For example, if you want the Japanese Yen to be the default, replace the contents of the warehouse default currency tag with the values from the Japanese currency tag (tag="loc:ja-JP"):

```
<Currency tag="loc:ja-JP" type="local" symbol="¥" locale="ja-JP" format="$#"
digits="0">
  <negative tag="minus" format="-$#" />
</Currency>
```

When you are finished, the default warehouse currency tag for Japanese should look like the following example:

```
<Currency tag="int:wrhs" type="international" symbol="¥" format="$#" digits="0"
displayMessage="kmsgCurrencySiebelWarehouse">
  <negative tag="minus" format="-$#" />
</Currency>
```

4. Save and close the currencies.xml file.

---

---

# Integrating Interactive Dashboards and Operational Applications Data

This appendix describes the additional configuration steps required for you to run an Oracle Business Intelligence application with one of Siebel (CRM) and Oracle E-Business Suite (operational) applications.

**Tip:** See Oracle Business Intelligence Server Administration Guide before performing any of the tasks in this section.

The integration of Oracle Business Intelligence with a Siebel operational application involves two general processes:

- [Section C.1, "Importing Oracle's Siebel Industry Applications Seed Data"](#)
- [Section C.2, "Completing the Initialization in the Siebel Operational Application"](#)

## C.1 Importing Oracle's Siebel Industry Applications Seed Data

Oracle Business Intelligence seed data is not installed with Oracle's Siebel Industry Applications. You must import the seed data into your database after the Oracle Business Intelligence installation is completed.

### To import Analytics seed data into a transactional database

1. Obtain the required language.inp and .dat files from the Oracle Business Intelligence language folder `\OracleBI\SeedData\bin\l_xx`, where *xx* is the two-letter code for the language you want to import.
2. Copy the .dat and corresponding .inp file from the language folder to the server installation `\bin` folder.
3. From the command prompt in `\OracleBI\SeedData\bin`, run the following command:

```
dataimp /u $USERNAME /p $PASSWORD /c "$ODBCDataSource" /d $Tableowner /f analytics_seed_<XXX>.dat /i metadata_upgrade_<XXX>_<DBPlatform>.inp /w y
```

Replace the *XXX* with the three-letter code (FRA, ITA) and the *DBPlatform* with the abbreviation for the database platform being used. For example:

```
dataimp /u sadmin /p sadmin /c JPN_CRMDEV1 /d siebel /f analytics_seed_JPN.dat /i metadata_upgrade_JPN_db2.inp /w y
```

For information about merging content into Oracle Business Intelligence Presentation Services, see the topics about using the Catalog Manager in Oracle Business Intelligence Web Administration Guide.

## C.2 Completing the Initialization in the Siebel Operational Application

Once you have configured your Oracle BI Server and are able to access the dashboards, you need to update the Siebel operational application to view Analytics dashboards from within the Siebel operational application. Completing the initialization in the Siebel operational application involves the following processes:

- Changing the operational application host name to the host name of the machine that runs Oracle Business Intelligence Presentation Services. See the following topics:
  - [Section C.2.1, "Updating the Siebel Operational Application"](#)
  - [Section C.2.2, "Reapplying Customized Style Sheets"](#)
  - [Section C.2.3, "How to Configure Oracle Business Intelligence with Oracle's Siebel Web Extension \(SWE\)"](#)
    - \* [Section C.2.4, "Creating a Virtual IP Address for the SWE and Oracle BI Presentation Services"](#)
    - \* [Section C.2.5, "Testing the Virtual IP Configuration for Oracle Business Intelligence and the SWE"](#)
  - [Section C.2.6, "How to Configure Oracle BI Action Links"](#)
    - \* [Section C.2.7, "Testing Action Links"](#)
- Customizing the operational application home page, dashboards, or content, or add views to a dashboard. See the following topics:
  - [Section C.2.8, "Process of Integrating Analytics and Oracle's Siebel Operational Applications"](#)
    - \* [Section C.2.9, "Accessing Optional Analytics Applications"](#)
    - \* [Section C.2.10, "Configuring the Browser Settings for Operational Applications"](#)
  - [Section C.2.11, "Process of Customizing Oracle BI Application Home Page and Dashboards"](#)
    - \* [Section C.2.12, "Customizing Oracle BI Content on the Siebel Operational Application Home Page"](#)
    - \* [Section C.2.13, "Determining the Oracle BI Report Path Argument"](#)
    - \* [Section C.2.14, "Adding Views for Custom Oracle BI Interactive Dashboards"](#)
    - \* [Section C.2.15, "Verifying NQHOST Setup for Pharma Disconnected Analytics Client"](#)
    - \* [Section C.2.16, "Integrating the Pharma Application Home Page with the Pharma Disconnected Analytics Client"](#)
    - \* [Section C.2.17, "Importing Pharma Application Target Lists to the Pharma Disconnected Analytics Client"](#)

### C.2.1 Updating the Siebel Operational Application

The following task changes the Siebel operational application host name to the host name of the machine that runs Oracle Business Intelligence Presentation Services.

#### **To update the Siebel operational application**

1. Open your Siebel operational application and login as SADMIN.
2. Navigate to View, then Site Map.
3. Click Integration Administration screen.
4. Click Host Administration view.
5. Query for NQHOST in the Virtual Name column:
  - In Windows, change the host name from *<AnalyticsServerName>* to the host name of the machine that runs Oracle BI Presentation Services.
  - In AIX or Solaris, change the host name from *<AnalyticsServerName>* to the name of the port.

For example, `servername.siebel.com:8080`

---

**Note:** For UNIX platforms only: You may need to add the domain name suffix to the server name in order to make sure that action links work on the Oracle Business Intelligence user interface.

---

6. Log out of the application and log back in.

## C.2.2 Reapplying Customized Style Sheets

For Oracle's Siebel Business Analytics versions 7.7 and later, new styles have been appended to the following style sheets:

- Go.css
- PortalBanner.css
- PortalContent.css
- Views.css

These new classes are identified in the style sheets. In this version of Oracle Business Intelligence, new styles and files must be added to the underlying style sheet (for example, to `s_Siebel7`). For complete functionality, any custom styles require similar updating. In addition, `views2.css` and some other files have been added to the `s_` directory. For custom column formatting to work properly, references to font sizes and families should also be removed from the `td` styles in `PortalBanner.css`, `PortalContent.css`, `Views.css`.

Because of these changes to styles, review your deployment's customizations manually, reapply them, and test them thoroughly to ensure that there are no problems.

After you have tested them, but before you copy the default `views.css` files back to the implementation server, perform the following task on the server to clear the server caches.

### To clear the server caches and restore your default views

1. Shut down Oracle Business Intelligence Server, Oracle BI Presentation Service and IIS.
2. Remove your custom `views.css` from the directory where it has been installed.

For example:

```
%INSTALL\Web\App\Res\s_Siebel7\b_mozilla_4
```

or

`$INSTALL\OracleBIData\Web.`

3. Clear the Oracle BI Presentation Services Server Cache.

In the `C:\WINNT\Temp` directory, delete the `nQs_*.temp` files.

4. Clear the Browser Cache.

From the Internet Explorer menu, navigate to Tools, then Internet Options, then Settings, then View Files, and delete all the files in these folder.

5. Restore the default views.css files to the appropriate folder.
6. Restart the Analytics Server, Oracle BI Presentation Services and IIS.

### C.2.3 How to Configure Oracle Business Intelligence with Oracle's Siebel Web Extension (SWE)

Whenever you run Oracle Business Intelligence and Siebel Web Extension (SWE) on separate machines, you must perform additional configuration steps in order for action links and interactive charts to work. If, for example, you plan to run the SWE and Oracle Business Intelligence Presentation Services on different Web servers, you must use some kind of networking or load balancing mechanism to create a single logical domain (or virtual IP address) for the two machines.

---

---

**Caution:** When Oracle BI Applications Server and Oracle BI Presentation Services are installed on different machines and load balancing software is used to create a single virtual IP address, the action links feature works only if you log onto the Siebel application using a virtual Web browser. Action links fail on a dedicated client if the SWE and Oracle BI Presentation Services servers are on separate machines. Either use a virtual Web browser, or install the Oracle BI Applications Web Server and Oracle BI Presentation Services on the same machine.

---

---

When one virtual IP address is created for two machines, the Web browser accesses one IP address and is still routed to different physical machines, based on the port accessed. From the browser, it appears that both servers are running on the same IP address.

You can use any of several physical methods to create a single logical domain, such as running SWE and Oracle Business Intelligence Presentation Services on a single machine if you are not load balancing the SWE, or using a router to do the mapping, or using load balancing software. Your company must determine the best mechanism to accomplish this routing given the topology being used.

Configuring Oracle Business Intelligence to work with Siebel Web Extension includes the following tasks:

- [Section C.2.4, "Creating a Virtual IP Address for the SWE and Oracle BI Presentation Services"](#)
- [Section C.2.5, "Testing the Virtual IP Configuration for Oracle Business Intelligence and the SWE"](#)

## C.2.4 Creating a Virtual IP Address for the SWE and Oracle BI Presentation Services

This task is part of the [Section C.2.3, "How to Configure Oracle Business Intelligence with Oracle's Siebel Web Extension \(SWE\)"](#).

You create a virtual IP address for the Siebel Web Engine (SWE) and Oracle BI Presentation Services in order to make it appear that all servers are running on the same virtual machine. The easiest way to do this is to configure Oracle BI Presentation Services to run on a different port (for example, port 84) from SWE (which usually runs on port 80).

For example, SWE is load-balanced across *<machine1>*:port 80 and *<machine2>*: port 80, and Oracle BI Presentation Services is running on *<machine3>*:port 84, and the virtual address is defined as `http://siebel.company.com`. Therefore, the network or load-balancing software should be configured to route requests like `http://siebel.company.com` to *<machine1>* and *<machine2>*, and to route requests like `http://siebel.company.com:84` to *<machine3>*.

### To create a virtual IP address for Siebel Web Engine and Oracle BI Presentation Services

1. On the network, set up CSS to direct requests from *<virtual domain>* to *<physical Siebel Web Server Extension machine>*:
  - Where the acronym CSS represents the load-balancer or router used to do the virtual IP configuration.
  - Where *<virtual domain>* is the virtual IP prefix that users enter to navigate to the Siebel applications (in the preceding example, this is `http://siebel.company.com`).
2. On the network, set up CSS to direct requests from *<virtual domain>*:84 to *<physical Oracle BI Presentation Services machine>*:84.
3. In the Siebel application, using the Siebel Administration screen, set the NQHost parameters for Oracle Business Intelligence Symbolic URLs to point to the *<virtual domain>*:84, instead of directly to the Oracle Business Intelligence Presentation Services server physical machine.

---

**Note:** If you are running Oracle's Siebel Business Analytics 7.5.3 instead of version 7.7 or later, perform the following additional step.

---

4. In the Siebel application on the Oracle Business Intelligence Presentation Services server machine, locate the registry setting `\SOFTWARE\Siebel Systems, Inc.\Siebel Analytics\Web\7.5\Charts`.
5. Add a new key, `ForceFileBasedPainter`, and enter TRUE into the Data string.

## C.2.5 Testing the Virtual IP Configuration for Oracle Business Intelligence and the SWE

This task is part of the [Section C.2.3, "How to Configure Oracle Business Intelligence with Oracle's Siebel Web Extension \(SWE\)"](#).

Use a client browser to verify that Oracle Business Intelligence and SWE work when accessed directly through a physical machine address, using the following procedure. For *<virtualdomain>*, substitute the Virtual IP address you created in [Section C.2.4, "Creating a Virtual IP Address for the SWE and Oracle BI Presentation Services"](#).

### To test the Virtual IP configuration with Oracle Business Intelligence and SWE

1. In a client browser, type `<virtual domain>:84/analytics`. The Oracle Business Intelligence logon appears.
2. In a client browser, type `<virtual domain>/callcenter` (or other Siebel application). The SWE appears.
3. Navigate to an Analytics screen within the Siebel application to see if Oracle Business Intelligence appears.
4. Interact with Oracle Business Intelligence charts and action links.

## C.2.6 How to Configure Oracle BI Action Links

Oracle Business Intelligence applications contain prebuilt action links in reports and dashboards. Action links are a way to navigate from a user's analytics dashboard to a record in a Siebel operational application, such as Oracle's Siebel Call Center.

For example, you can drill down directly from a Order Management Analytics dashboard to a specific record in a Sales view. The link is based on the row-ID column contained in a report. You can also create new action links for any existing report. For how to create Interactive Oracle Dashboards action links, see Oracle Business Intelligence Web Administration Guide.

Depending on your deployment, you may need to perform additional configuration of your Oracle Business Intelligence application in order for the action links feature to work properly. The following tasks are relevant to the configuration of your application for action links:

- [Section C.2.7, "Testing Action Links"](#)

## C.2.7 Testing Action Links

This task is part of the [Section C.2.6, "How to Configure Oracle BI Action Links"](#).

Use the following procedure to make sure that action links you create work correctly.

### To test an Oracle Business Intelligence action link

1. Log on using your system.  
Go to Service Analytics screen, then Employee Analytics, then To Do List.
2. Click the action link on the pie chart in Overdue Activities.  
You should be in Overdue Activities Detail in Analytics.
3. Click any ROW\_ID action link.  
You should be in the Employee screen in Oracle's Siebel Call Center operational application.

## C.2.8 Process of Integrating Analytics and Oracle's Siebel Operational Applications

The following tasks are described for the process of integrating your Oracle Business Intelligence application with your Siebel operational application:

- [Section C.2.9, "Accessing Optional Analytics Applications"](#)
- [Section C.2.10, "Configuring the Browser Settings for Operational Applications"](#)
- [Section C.2.11, "Process of Customizing Oracle BI Application Home Page and Dashboards"](#)

## C.2.9 Accessing Optional Analytics Applications

This task is part of [Section C.2.8, "Process of Integrating Analytics and Oracle's Siebel Operational Applications"](#)

Depending on the options you purchased with your Siebel operational application, you must perform additional steps in order to access the corresponding Oracle Business Intelligence options. [Table C-1](#) shows the additional options for Oracle Business Intelligence.

**Table C-1 Oracle Business Intelligence Applications Options**

Siebel Application	Options
Sales	Order Management Analytics, Usage Accelerator option.
Service	Service Analytics, Contact Center Telephony Analytics option.

By default, the dashboards and reports contained in these optional areas are hidden. If, for example, you purchased Order Management Analytics with your Sales application, you must perform the additional steps shown in the following procedure to access the Sales Analytics.

### To turn on options for Sales Analytics

1. Log in to Oracle Business Intelligence as Administrator.
2. Navigate to Answers, then Oracle BI Presentation Services Administration and select the option to manage Presentation Services groups and users.
3. Locate the Web Group corresponding to your option.

The Web Group options are shown in the following list.

- No Forecasting
  - No Forecasting Lite
  - No Universal Queuing
  - No Email Response
  - No Service Agreements
  - No Partner Marketing
  - No Partner ERM
  - No Partner ISS
4. Click on the Edit icon.
  5. Under the Group Membership section, click the delete icon (X) to delete Analytics Users from this group.
  6. Click Finished and log out of the application.
  7. Log in again to access the additional optional dashboards and reports.

## C.2.10 Configuring the Browser Settings for Operational Applications

---

**Note:** This topic applies only to Oracle's Siebel Business Analytics version 7.7.1 and Siebel operational applications version 7.7.

---

This task is part of [Section C.2.8, "Process of Integrating Analytics and Oracle's Siebel Operational Applications"](#).

If the Siebel Server and the Oracle Business Intelligence Server URLs are from different domains, you may need to change the browser settings for your Oracle Business Intelligence application. Perform the following procedure.

---

---

**Note:** You do not need to change browser settings if the Siebel Server and the Oracle Business Intelligence Server URLs are from the same domain.

---

---

#### **To change browser settings for Oracle Business Intelligence operational applications**

1. In your browser, navigate to Tools, then Internet options, then Security tab.
2. Select Internet zone and click Custom Level.
3. In the Miscellaneous > Access data sources across domains setting, select the Enable radio button and click OK.
4. Select Local Intranet zone and click Custom Level.
5. In the Miscellaneous > Access data sources across domains setting, select the Enable radio button and click OK.
6. Click OK to exit browser Tools.

### **C.2.11 Process of Customizing Oracle BI Application Home Page and Dashboards**

The process of customizing your Oracle Business Intelligence application's home page and dashboards may include the following tasks:

- [Section C.2.12, "Customizing Oracle BI Content on the Siebel Operational Application Home Page"](#)
- [Section C.2.13, "Determining the Oracle BI Report Path Argument"](#)
- [Section C.2.14, "Adding Views for Custom Oracle BI Interactive Dashboards"](#)
- [Section C.2.15, "Verifying NQHOST Setup for Pharma Disconnected Analytics Client"](#)
- [Section C.2.16, "Integrating the Pharma Application Home Page with the Pharma Disconnected Analytics Client"](#)
- [Section C.2.17, "Importing Pharma Application Target Lists to the Pharma Disconnected Analytics Client"](#)

For a list of Pharma Disconnected Analytics components that are installed with Oracle BI Applications, see [Appendix F, "Pharma Disconnected Analytics Administration - Supplemental Information"](#).

### **C.2.12 Customizing Oracle BI Content on the Siebel Operational Application Home Page**

This task is part of the [Section C.2.11, "Process of Customizing Oracle BI Application Home Page and Dashboards"](#).

Oracle Business Intelligence applications are integrated with Siebel operational applications using the symbolic URL infrastructure. The following task describes how

to use symbolic URLs to link a new report to a Siebel operational application home page and how to add new Analytics Dashboards to the Siebel operational application. The symbolic URL specifies how the HTTP request to the external application should be constructed and to defines any arguments and values to be sent as part of the request.

---

**Note:** For more information on the symbolic URL infrastructure, see Siebel Portal Framework Guide. This topic assumes that you have successfully built a symbolic URL to link some external content, using the instructions given in Siebel Portal Framework Guide.

---

For each Analytical report on a Siebel operational application home page, there is a symbolic URL record defined that links the home page to the Analytics report. If you have not already done so, you need to create a new home page and set it up to use a symbolic URL.

---

**Note:** For Oracle Business Intelligence Disconnected settings, see [Section C.2.15, "Verifying NQHOST Setup for Pharma Disconnected Analytics Client"](#).

---

The figure below shows example Inline and IFrame symbolic URL arguments.

**Figure C–1 Examples of Symbolic URL Arguments.**

Inline

Name	Required Argument	Argument Type	Argument Value	Append as Argument	Substitute in Text	Sequence #
Cmd	✓	Constant	Go	✓		1
Path	✓	Constant	/shared/Service/Service Requests	✓		2
Syndicate	✓	Constant	Siebel	✓		3
nqUser	✓	Command	UseSiebelLoginId	✓		4
nqPassword	✓	Command	UseSiebelLoginPassword	✓		5

IFrame

Name	Required Argument	Argument Type	Argument Value	Append as Argument	Substitute in Text	Sequence #
IFrameLogin:Cmd	✓	Constant	Logon	✓		1
Cmd	✓	Constant	PortalPages	✓		2
IFrameLogin:nqUser	✓	Command	UseSiebelLoginId	✓		3
IFrameLogin:nqPassword	✓	Command	UseSiebelLoginPassword	✓		4
PortalPath	✓	Constant	/shared/Service/_Portal/Activities	✓		5
PostRequest	✓	Command	PostRequest	✓		6
IFrameLogin:Syndicate	✓	Constant	Siebel	✓		7

**To configure a new Home Page to use a symbolic URL**

1. Make sure the symbolic URL has been set up.
2. Launch the Siebel operational application and navigate to the Integration Administration, then Symbolic URL Administration view.
3. Query for the symbolic URL that has been set up for Analytics.

The name of this symbolic URL should be exactly the same as the calculated value of the field that was added to the Business Component. For example, you may have a symbolic URL named HomePageAnalytics.

- In the URL field, enter the Web URL. For example:

`http://NQHOST/Analytics/saw.dll`

For the Host Name, choose the Analytics Server Name from the drop-down list. The following table shows the other parameters for reports.

Parameter	Value
NQHOST	A virtual name in the URL that points to the Oracle BI Presentation Services machine
Fixup Name	Inside Applet
SSO Disposition	Inline
Oracle BI Presentation Services application name	Select from the drop-down list

- Create the appropriate symbolic URL Arguments.

These arguments depend upon the particular Analytics report that you are trying to display. The Argument Values should be the same for any Analytics report you work with, except for the Path Argument Value. Use the procedure in [Section C.2.13, "Determining the Oracle BI Report Path Argument"](#) to determine the path to the Analytics report.

### C.2.13 Determining the Oracle BI Report Path Argument

This task is part of the [Section C.2.11, "Process of Customizing Oracle BI Application Home Page and Dashboards"](#).

The Path argument tells the symbolic URL the path to the report on the Oracle BI Presentation Services. (For example, `/shared/Sales/Pipeline/Overview/Top 10 Deals`.) Use the following procedure to determine the path to the Analytics report.

#### To determine the path to the report

- Log on to your Oracle BI Presentation Services as an Administrator.
- In the Siebel operational application, navigate to Answers, then Oracle BI Presentation Services Administration.
- Select Manage Analytics Catalog, and then navigate to your report.
- Add this path name to the Symbolic URL argument.

The following table shows the symbolic URL path arguments for reports.

Name	Type	Path Argument Value	Append	Sequence #
Cmd	Constant	GO	Y	1
Path	Constant	<code>/shared/Sales/Pipeline/Overview/Top 10 Deals</code>	Y	2
Syndicate	Constant	Siebel	Y	3
nQUser	Command	UseSiebelLoginId	Y	4
nQPassword	Command	UseSiebelLoginPassword	Y	5
PostRequest	Command	PostRequest	Y	6

## C.2.14 Adding Views for Custom Oracle BI Interactive Dashboards

This task is part of the [Section C.2.11, "Process of Customizing Oracle BI Application Home Page and Dashboards"](#) and is similar to that of adding Oracle Business Intelligence reports to the home page:

- Using Oracle's Siebel Tools, set up a new view.
- In the Siebel operational application, define a symbolic URL for that view.

For more information on how to set up a view to use Symbolic URLs, see Siebel Portal Framework Guide.

### To configure the symbolic URL for Analytics dashboards

1. Define a Symbolic URL.
  - a. Navigate to Site Map, then Integration Administration, then Symbolic URL Administration.
  - b. In the Symbolic URL Administration view, add a new record.
2. Define Symbolic URL arguments.
  - a. Navigate to Symbolic URL Administration.
  - b. In the Symbolic URL Administration list, select the Symbolic URL you want to configure.

## C.2.15 Verifying NQHOST Setup for Pharma Disconnected Analytics Client

This task is part of the [Section C.2.11, "Process of Customizing Oracle BI Application Home Page and Dashboards"](#).

Pharma Disconnected Analytics is a prebuilt Disconnected Analytics application for Oracle's Siebel Pharma Sales. When you run Oracle Business Intelligence Disconnected Client, your client application runs against a local instance of the database on your machine. You need to confirm that NQHOST is using the correct port number and DLL files to work with the Pharma Disconnected Analytics Client.

### To verify the NQHOST setup

1. Run the Disconnected Client application against your local database.
2. Log in as Administrator.
 

You must have access to the following Integration Administration Screen views:

  - WI Host Admin View
  - WI Symbolic URL Admin View
3. Navigate to Administration – Integration.
4. Click WI Symbolic URL List, and search for NQHOST in Virtual Name field.
 

The Host Name value should be localhost:9762.
5. Select the Host Administration View from the drop-down list and choose Symbolic URL Administration.
6. The following table shows Disconnected Client symbolic URLs to search for.

For each of the URLs, the URL field is:

```
http://NQHOST/Analytics/sawd.dll
```

The Disconnected Client Symbolic URL Names are:

- ePharmaHomePageAnalytics1
- ePharmaHomePageAnalytics2
- ePharmaHomePageAnalytics3
- ePharmaHomePageAnalytics4
- SISMLSDistrictManagerDashboard1
- SISMLSSalesRepDashboard1
- SiebelAnswers
- SiebelDelivers

The following table shows the symbolic URL path arguments for Oracle BI Answers and Oracle BI Delivers.

Name	Type	Path Argument Value	Append	Sequence #
Cmd	Constant	Answers for SiebelAnswers	Y	1
		Delivers for SiebelDelivers		
nQUser	Command	UseSiebelLoginId	Y	2
nQPassword	Command	UseSiebelLoginPassword	Y	3

### C.2.16 Integrating the Pharma Application Home Page with the Pharma Disconnected Analytics Client

This task is part of the [Section C.2.11, "Process of Customizing Oracle BI Application Home Page and Dashboards"](#).

If you have licensed both the Siebel Pharma operational application and Oracle Business Intelligence Disconnected Client, you must change part of the operational application configuration file in order to access Analytics reports on the LS Analytics Home Page and to allow Create Target List integration.

#### To allow integration of Pharma Analytics Home Page with Pharma Disconnected Analytics

1. On the machine where you have installed the Siebel operational application, navigate to `$INSTALLDIR\sea77\siebsrvr\BIN\ENU\epharma.cfg`.
2. Using a text editor, open the file `epharma.cfg`.
3. In the Local section, find the parameter `UseCachedExternalContent`.

The default value for this parameter is TRUE.

---

**Note:** If this parameter remains set to TRUE, Analytics reports on the Home Page return an error.

---

4. Set the `UseCachedExternalContent` parameter to FALSE.
5. Save and close the file.

## C.2.17 Importing Pharma Application Target Lists to the Pharma Disconnected Analytics Client

This task is part of the [Section C.2.11, "Process of Customizing Oracle BI Application Home Page and Dashboards"](#).

If you have licensed both the Siebel Pharma operational application and Oracle Business Intelligence Disconnected Client, you must change part of the operational application configuration file in order to allow you to import the contacts from the Pharma Analytics Create Target List function into the Disconnected Analytics Client.

### To allow importation of Target List contacts into Oracle Business Intelligence Disconnected Client

1. On the machine where you have installed the Siebel operational application, navigate to `$INSTALLDIR\sea77\siebsrvr\BIN\ENU\epharma.cfg`.
2. Using a text editor, open the file `epharma.cfg`.
3. In the Siebel Client section, find the parameter `EnableFQDN`.

The default value for this parameter is `TRUE`.

---

**Note:** If this parameter remains set to `TRUE`, the Create Target List functionality is disabled in the Pharma Disconnected Analytics Client.

---

4. Set the `EnableFQDN` parameter to `FALSE`.
5. Save and close the file.
  - a. In the Symbolic URL Arguments list, enter the arguments that need to be sent to the external host.

The following table shows the symbolic URL argument definitions for dashboards.

Parameter	Value
Fixup Name	Default
SSO Disposition	IFrame
Oracle BI Presentation Services application name	Select from the drop-down list.

The following table shows the symbolic URL path arguments for dashboards.

Name	Type	Value	Append	Sequence #
Cmd	Constant	Dashboard	Y	1
PortalPath	Constant	/	Y	2
Page	Constant	Sales	Y	3
Syndicate	Constant	Siebel	Y	4
nQPassword	Command	UseSiebelLoginPassword	Y	5
nQUser	Command	UseSiebelLoginId	Y	6

6. In Oracle Business Intelligence, create the dashboards.

---

---

**Note:** Only an Oracle BI Presentation Services administrator can perform this step.

---

---

7. In Oracle Business Intelligence, set up the dashboards to your requirements.

## C.3 Viewing Information About Preconfigured Dashboards

If your organization has prebuilt applications installed, you can use Catalog Manager to locate and view information about preconfigured dashboards. You may first have to expose the dashboards and requests.

### C.3.1 Exposing Dashboards and Requests

Depending on the Oracle Business Intelligence options your organization purchased, you may need to expose these options before the associated dashboards and requests can be viewed in Oracle BI Presentation Services and in Catalog Manager. This applies to sites that have the following prebuilt applications and options:

- Oracle Order Management Analytics, with the Forecasting and Incentive Compensation options.
- Oracle Service Analytics, with the Universal Queuing, Email Response, and Agreements options.
- Oracle Partner Analytics, with the Partner Marketing, Partner ERM, and Partner ISS options.

These options need to be exposed using the administration feature in Oracle BI Presentation Services.

### C.3.2 Locating Dashboard and Page Names

In Catalog Manager, the Presentation Catalog distributed with prebuilt applications has the following structure:

Presentation Catalog > shared folder > prebuilt application name > \_portal folder > dashboard name > dashboard page name

---

---

# Configuring Metadata for Analytics Applications

Oracle's Siebel applications are shipped with certain Oracle Business Intelligence metadata. These metadata repositories are included with Oracle's Siebel operational applications and Oracle's Siebel Industry Applications. All the rules for security, data modeling, aggregate navigation, caching, and connectivity for the Oracle Business Analytics Warehouse are stored in metadata repositories on the Oracle Business Intelligence Server.

This appendix describes configuration necessary for the Oracle Business Intelligence metadata for Oracle's Siebel operational applications, specifically those areas that a Siebel administrator may need to adjust. These administrative tasks generally cover two subareas:

- Dashboard content of Oracle's Siebel operational applications
- Metadata requirements for Oracle's Siebel operational applications

This appendix contains the following topics:

- [Section D.1, "What Are Oracle's Siebel Operational Applications?"](#)
- [Section D.2, "Updating Financial Analytics Logical Table Sources"](#)
- [Section D.3, "Externalizing Financial Services Metadata Translation Strings"](#)
- [Section D.4, "Disabling the Financial Analytics Logical Table Sources"](#)
- [Section D.5, "Developing and Deploying Predictive Scores"](#)
- [Section D.6, "Analytics Metadata Requirements for Oracle's Siebel Industry Applications"](#)

## D.1 What Are Oracle's Siebel Operational Applications?

Oracle's Siebel Operational Applications are built around a general business function, such as Oracle's Siebel Sales, Oracle's Siebel Service, and Oracle's Siebel Call Center.

Oracle's Siebel Industry Applications are built around specific industries' business practices:

- Financial Services
- Consumer Sector
- Communications, Media, and Energy (CME)
- Pharma Sales

---

---

**Note:** You can license one or more Siebel Industry Applications for Oracle Business Intelligence. Licensing allows access to particular dashboards, subject areas, and reports.

---

---

## D.2 Updating Financial Analytics Logical Table Sources

In the Oracle BI repository file, logical table sources are set by default to settings in Oracle's Financial Analytics family of products (Finance Sales Analytics, Finance Service Analytics, Finance Marketing Analytics, Finance Institutional Analytics, Finance Retail Analytics), the Healthcare Analytics family of products (Oracle Healthcare Sales Analytics, Oracle Healthcare Service Analytics, Oracle Healthcare Marketing Analytics, Oracle Healthcare Partner Manager Analytics), and the Oracle Insurance Analytics family of products (Oracle Insurance Partner Manager Analytics, Oracle Insurance Sales Analytics, Oracle Insurance Service Analytics, Oracle Insurance Marketing Analytics, Oracle Insurance Partner Manager Analytics). Before using these Financial Services Analytics applications, you must update the logical sources for two tables in the Oracle Business Intelligence repository file. These logical sources must be deactivated in order for Oracle's Financial Analytics family of products (Finance Sales Analytics, Finance Service Analytics, Finance Marketing Analytics, Finance Institutional Analytics, Finance Retail Analytics), the Healthcare Analytics family of products (Oracle Healthcare Sales Analytics, Oracle Healthcare Service Analytics, Oracle Healthcare Marketing Analytics, Oracle Healthcare Partner Manager Analytics), and one of the Oracle Insurance Analytics family of products (Oracle Insurance Partner Manager Analytics, Oracle Insurance Sales Analytics, Oracle Insurance Service Analytics, Oracle Insurance Marketing Analytics, Oracle Insurance Partner Manager Analytics) reports to point to the correct logical model and retrieve the correct data. Do this by deactivating the sources for the Fact-Asset logical table in the Core subject area and activating the FINS logical sources, as shown in the following procedure.

1. Launch Oracle BI Administration Tool and open the Analytics Repository (OracleBIAnalyticsApps.rpd).
2. Go to the Business Model and Mapping window (the logical layer window) and open the Core folder.
3. Scroll down to the Fact - CRM - Asset logical table and open its Sources folder.
4. In the list of logical table sources, right-click Fact\_ASSET\_F\_FINS.
5. Select Properties.
6. Click the General tab in the Properties window and make sure that the Active check box is checked. If it is not, check it.
7. In the list of logical table sources, right-click W\_ASSET\_F.
8. Select Properties.
9. Click the General tab in the Properties window and make sure that the Active check box is unchecked. If it is not, uncheck it.
10. Click OK and save the repository.
11. Restart Oracle BI Server.

## D.3 Externalizing Financial Services Metadata Translation Strings

The Financial Services applications use a different set of translation strings from other Siebel operational applications.

You must externalize the metadata strings in the Analytics repository.

### To externalize metadata strings in the Analytics repository

1. Open the OracleBIAnalyticsApps.rpd using the Oracle BI Administration Tool.
2. Select the entire presentation layer and right-click the mouse to display the menu.
3. Choose Display Names.

With this configuration, all metadata strings are read from an external Siebel operational application database, specifically from the table W\_LOCALIZED\_STRING\_G.

4. Make sure that the connection pool Externalized Metadata Strings points to the Siebel operational application database and is working correctly.
5. Go to the Manage Variables menu and locate the initialization block External Metadata Strings.

Double-click on the initialization block to open the Edit window.

6. In the Initialization SQL area, change the SQL:

from

```
select MSG_NUM, MSG_TEXT
from VALUEOF(OLAP_TBO).W_LOCALIZED_STRING_G
where MSG_TYPE = 'Metadata' and
LANG_ID = decode('VALUEOF(NQ_SESSION.WEBLANGUAGE)')... [more]
```

to

```
select MSG_NUM, MSG_TEXT
from VALUEOF(OLAP_TBO).W_LOCALIZED_STRING_G
where MSG_TYPE = 'FINS_Metadata' and
LANG_ID = decode('VALUEOF(NQ_SESSION.WEBLANGUAGE)')... [more]
```

7. Click OK to commit the changes and save the repository.
8. Restart the Oracle BI Server.

## D.4 Disabling the Financial Analytics Logical Table Sources

In the Analytics Repository file, logical table sources are set by default to settings in Oracle's Siebel Financial Analytics, the Healthcare Analytics family of products (Oracle Healthcare Sales Analytics, Oracle Healthcare Service Analytics, Oracle Healthcare Marketing Analytics, Oracle Healthcare Partner Manager Analytics), and the Oracle Insurance Analytics family of products (Oracle Insurance Partner Manager Analytics, Oracle Insurance Sales Analytics, Oracle Insurance Service Analytics, Oracle Insurance Marketing Analytics, Oracle Insurance Partner Manager Analytics). If you are using any *Siebel Industry application that is not* Oracle's Siebel Financial Services Analytics, you must first update the logical table sources in the Analytics Repository file.

These logical table sources must be deactivated in order for your Siebel Industry application Analytics reports to point to the correct logical model and retrieve the correct data. Do this by deactivating the FINS sources for the logical tables in the Core

subject area and activating the other logical sources, as shown in the following procedure.

---



---

**Note:** Before performing the following procedure, shut down the Oracle BI Server.

---



---

**To update Logical Table sources for Oracle's Siebel Industry Applications Analytics**

1. Launch Oracle BI Administration Tool and open the Analytics Repository (OracleBIAnalyticsApps.rpd).
2. Go to Business Model and Mapping window (the logical layer window) and open the Core folder.
3. Scroll down to the Fact - CMR - Asset logical table and open its Sources folder.
4. In the list of logical table sources, right-click Fact\_W\_ASSET\_F\_FINS.  
Select Properties.
5. Click the General tab in the Properties window and uncheck the Active check box.
6. In the list of logical table sources, right-click W\_ASSET\_F.  
Select Properties.
7. Click the General tab in the Properties window.  
Make sure that the Active check box is checked. If it is not, check it.  
Click OK and save the repository.
8. Restart Oracle BI Server.

## D.5 Developing and Deploying Predictive Scores

The Loyalty Management Dashboard and several Oracle Business Intelligence subject areas use customer scores generated from Oracle Real-Time Decisions. Oracle Real-Time Decisions uses mathematical models to predict customer behavior. For customer scoring to be made available for analysis in Oracle Business Intelligence, CME metadata is provided which maps these customer scores to dashboards and subject areas.

The following procedure describes the process of developing and deploying these predictive scores.

**To develop and deploy predictive scores**

1. Generate predictive scores using Oracle Real-Time Decisions.

---



---

**Note:** This is performed outside of the Siebel operational application.

---



---

2. Integrate the scores into the Oracle Business Analytics Warehouse.  
Once this is completed, scores may be viewed in the Siebel operational application by accessing the Accounts, then Profiles, then Loyalty Profile view.
3. Load the integrated scores into the Oracle Business Analytics Warehouse during the extraction, transformation, and loading (ETL) process.

4. After the scores are loaded into the Oracle Business Analytics Warehouse, map them to the following Oracle Business Intelligence metadata fields:
  - Churn Score
  - Customer Lifetime Value Score
  - Upsell Score
  - Cross-Sell Score
  - Financial Risk Score

In conjunction with other associated metadata, these fields are primarily used to populate the Loyalty Management dashboard.

## D.6 Analytics Metadata Requirements for Oracle's Siebel Industry Applications

Some metadata needs to be set up properly in the Oracle Business Analytics Warehouse for it to be displayed accurately in Oracle Business Intelligence. The following topics describe the metadata structure for each of the following Oracle's Siebel Industry Applications:

- [Section D.6.1, "Oracle's Siebel Consumer Sector Dashboards and Pages"](#)
- [Section D.6.2, "Oracle's Siebel Consumer Sector Data Requirements"](#)
- [Section D.6.3, "Oracle's CME family of products \(Oracle Communications, Media and Energy Sales Analytics\)"](#)
- [Section D.6.4, "Oracle's Siebel CME Dashboards and Pages"](#)
- [Section D.6.5, "Oracle's Siebel CME Data Requirements"](#)
- [Section D.6.6, "Oracle Pharma Sales Analytics Dimensions"](#)
- [Section D.6.7, "Dimensions Specific to Subject Areas in Oracle Pharma Sales Analytics and Oracle Pharma Marketing Analytics"](#)

### D.6.1 Oracle's Siebel Consumer Sector Dashboards and Pages

Oracle's Siebel Consumer Sector Sales Analytics extends the base Sales Analytics application to include Trade Funds Management Analytics, Trade Promotion Evaluation Analytics, Sales Performance Analytics, and Retail Audit Analytics.

All Consumer Sector specific metadata has been added to a single subject area. In the Oracle BI Administration Tool, this metadata is tagged with a red apple icon. The following topic covers each fundamental area of Consumer Sector Analytics and provides tips for data entry for effective analytics.

The Consumer Sector dashboards and pages available to the end user are described in [Table D-1](#).

**Table D-1 Consumer Sector Dashboards and Page Tabs**

Dashboard	Page	Function
Retail Audit	Last Audit	Shows the aggregated results of the last audit, defined the last time a product was audited at a specific account and merchandising location. The aggregated metrics can be drilled into to get a list of accounts or products to target future efforts.

**Table D-1 (Cont.) Consumer Sector Dashboards and Page Tabs**

Dashboard	Page	Function
	Trends	Displays key metrics captured in a retail audit over time across accounts or product categories.
	Position	Combines account and product performance with the individual representative responsible.
Promotion	Plan Year To Date	Displays both individual promotional performance with cumulative trends to provide overall perspective on meeting promotional targets.
	Key Accounts	Shows post promotion evaluation from a key account perspective across several levels of detail including plan, promotion, or promoted product detail.
	Corporate	Shows post promotion evaluation from a brand managers perspective, by evaluating corporate promotions.
Funds	Summary	Displays the key analyses for a fund manager, including Remaining Amounts which provides a status of all funds relative to all other funds.
	Accounts	Highlights status of funds and funds spent at a specific account and is targeted to assist key account managers in determining how to fund future promotions.
Sales Performance	Sales Volume Planning	Details the key metrics used in sales volume planning, including incremental, target, shipment and consumption volumes, over time and in comparison to one another. Baseline and estimated incremental volumes are stacked and compared to targets to assist in identifying future progress against targets. This analysis uses a global filter to assist end users in getting to the level of details they want in one selection.
	Hierarchy	Similar in content to Sales Volume Planning, however, this analysis is organized to promote exploration of data, allowing end users to freely drill up or down the account, time, or category product hierarchies. This page should be used to help sales managers identify where sales exceeded expectations.
	Trends	Depicts sales trends across accounts, channels, and categories as well as compares account performance in order to rank them.
	Growth	Displays key sales metrics versus the year ago and charts the rate of growth.
VP Sales	Business Overview	This page focuses on answering key business questions of a sales executive including where are my sales? How effective are my promotions by channel? Which account plans are top and bottom performers? How is my promoted volumes and funds spend trending as compared to last year? What are my top five accounts in each category?
	Product Overview	This page focuses on answering key product questions of a sales executive. For example, What percentage of total sales is in each category? What products are selling where by whom?
Key Account Manager	Account	This page focuses on answering key business questions of a key account manager. For example, How I am performing versus my account plan? What is my promotional forecast accuracy? What funds are available to plan additional retail activity?
	Category	This page focuses on answering key product questions of a key account manager. For example, Which category is best promoted at my accounts? How are my store conditions trending? Are out of stocks preventing me from reaching my targets?

## D.6.2 Oracle's Siebel Consumer Sector Data Requirements

The data requirements for the Consumer Sector–specific portion of the data model are detailed in [Table D–2](#). This includes the fund, promotion, performance and retail audit schema.

**Table D–2 Data Requirements for Consumer Sector Schema**

Page	Function
Funds	Trade Fund Management Analytics incorporates the latest Trade fund functionality including aggregation of approved adjustments, transfers, deal allocations, and payments at all statuses.
Promotion	In the transactional database, fund allocations and product movement (incremental volumes) can be assigned to a promotion at the promotion or the promoted product level. The ETL transfers this information at the promoted product level only. If you allocate funds at the promotion level and assign product detail such as shipment quantities at the promoted product level, the fund allocation data needs to be pushed to the promoted product level to be accurately stored in the Oracle Business Analytics Warehouse.
Performance	<p>Sales Performance Analytics is used primarily in conjunction with the category product hierarchy as defined by Oracle's Siebel Sales Volume Planning. To create the category-product hierarchy, the SVP adopt flag must be selected in order to aggregate product data up the category product hierarchy. This flag can be found by navigating to Catalog Administration, then SVP Category Details View, then Products list.</p> <p>The data warehouse and all prebuilt analyses are built from the end date of a period. To make sure data is accurate, the periods in the Oracle Business Analytics Warehouse tables must be of a single duration. If they are different, inaccurate data could be stored in the data warehouse. For example, if a week and a month end on the same date, the actual shipped quantities are combined during the building of the data warehouse.</p>
Retail Audit	<p>The Last Audit Flag is set during the ETL process. The flag is set to Yes for the most recent record for a product at an account and merchandising location. All other audit records for that combination of product, account, and merchandising location are set to No.</p> <p>The observation date is used to sort the retail audit records to determine which is the last audit. The observation date is populated when the status of a retail audit activity is changed to Done. The field observation date does not show up in the Oracle Business Analytics Warehouse user interface and does not have to be the same as the activity date.</p>

## D.6.3 Oracle's CME family of products (Oracle Communications, Media and Energy Sales Analytics)

Oracle's CME family of products (Oracle Communications, Media and Energy Sales Analytics,

Oracle Communications, Media and Energy Service Analytics, Oracle Communications, Media and Energy Marketing Analytics) makes use of order management functionality configured for CME. For Oracle's CME applications to fully reflect the information collected by CME order management functionality, some extensions to the Oracle CME Analytics application may be required. This topic explains these potential extensions.

Oracle's Siebel Sales Orders include complex products and simple products.

**Complex Products.** A series of products related by a product hierarchy. The highest product in the hierarchy is the root product, and the lower level products are the child products. In complex products, revenue figures are summed and roll up to the root product using the ROLLUP\_NET\_PRI field. For a complex product, Oracle Business

Intelligence examines only the root product when computing revenue. Child products are disregarded because their revenue is already reflected in the root.

**Simple Products.** A root product. Oracle Business Intelligence examines this root product when computing revenue, and nothing more.

Oracle's Siebel Communications, Media and Energy order management functionality supports products which have recurring charges over time (for example, \$20 per month for 12 months), one-time charges (for example, one-time purchase price of equipment), and usage charges (for example, 15 cents per minute).

The revenue attributed to a product with recurring charges is valued by taking the product's net price and multiplying it by the number of months that product is anticipated to be active, as represented by the Number of Revenue Occurrences field. This field, contained in Quote Item and Order Item records, is contained in the Oracle Business Analytics Warehouse by the following fields:

- W\_QUOTEITEM\_F.NUM\_OCCURRENCE
- W\_ORDERITEM\_F.NUM\_OCCURRENCE

In Oracle's CME family of products (Oracle Communications, Media and Energy Sales Analytics, Oracle Communications, Media and Energy Service Analytics, Oracle Communications, Media and Energy Marketing Analytics), revenue metrics do not automatically account for all recurring charges, and do not consider the NUM\_OCCURRENCE fields. Instead, Oracle's CME family of products revenue metrics incorporate one-time charges, one-month's worth of recurring charges, and no usage charges. To incorporate the anticipated value of all recurring charges, the W\_QUOTEITEM\_F.NUM\_OCCURRENCE and W\_ORDERITEM\_F.NUM\_OCCURRENCE fields may need to be incorporated into revenue calculations made during the Extraction, Transformation and Load (ETL) process for order item and line item records.

Alternatively, these fields in the Oracle Business Analytics Warehouse, representing the aggregated recurring and one-time product charges, may be used and incorporated into the ETL processes:

- S\_ORDERITEM.PER\_MTH\_CHG\_SUBTOT
- S\_ORDERITEM.ONETIME\_CHG\_SUBTOT
- S\_QUOTEITEM.PER\_MTH\_CHG\_SUBTOT
- S\_QUOTEITEM.ONETIME\_CHG\_SUBTOT

Each CME Order line item and Quote line item contains an Action Type of Add, Update, or Delete. Because Oracle Business Intelligence only looks at root product line items, only the Action Types associated with the root product are considered during analysis. Therefore, while all line items for a complex product may collectively include a combination of various Action Types, only the Action Type for the root product are considered during analysis. This is of special importance if a filter or query criteria in analysis is based on the Action Type field, which it is for most Account Management and Revenue Management dashboard reports.

Similarly, each CME Order line item and Quote line item is associated with a product of a particular Price Type. Because Oracle Business Intelligence considers root products only, only the Price Type associated with the root product is considered during analysis. Again, this is important if a filter or query criteria is based on Price Type. Such filter criteria apply to most Account Management and Revenue Management dashboard reports.

## D.6.4 Oracle's Siebel CME Dashboards and Pages

Oracle's Siebel Communications, Media and Energy (CME) Analytics contains corresponding industry-specific metadata. In the Oracle BI Administration Tool, industry-specific metadata is flagged with an icon picturing a telephone. Although this icon is visible in the Oracle BI Administration Tool, it is neither visible nor included within Oracle BI Answers. End users use Oracle BI Answers to access metadata for building queries and reports. For users of Oracle BI Answers to view and access CME metadata columns, they must log in using one of the CME responsibilities listed in [Table D-3](#). These responsibilities also determine what subject areas the user may access.

**Table D-3 Communications, Media and Energy Dashboards and Page Tabs**

Dashboard	Page	Function
Loyalty Management	Customer Lifetime Value	Segments customers based upon defined ranges of scores predicting customer lifetime value.
	Churn Propensity	Segments customers based on defined ranges of scores estimating churn propensity.
	Selling Propensity	Segments customers based on defined ranges of scores valuing the potential of up-sell and cross-sell opportunities.
	Financial Risk	Segments customers based on defined ranges of scores predicting financial risk.
	Actual Churn	Shows trending of actual customer churn, and actual customer acquisition, over time.
Revenue Management	Revenue Trends	Charts trends of order revenue and order volume over time, and identifies top products based on order revenue and volume.
	Service Activations	Charts trends of service activations over time, and indicates top service activations based on order revenue and volume.
	Service Modifications	Charts trends of service modifications over time, and indicates top service modifications based on order revenue and volume.
	Service Disconnections	Charts trends of service disconnections over time, and identifies services with the highest disconnection volume.
Account Management	Sales Portal	Identifies top accounts, and related top products, based upon order revenue and order volume.
	Service Activations	Charts trends of account service activations, and indicates top accounts based on service activation performance metrics.
	Service Modifications	Charts trends of account service modifications, and indicates top accounts based on service modification performance metrics.
	Service Disconnections	Charts trends of account service disconnections, and identifies accounts with the highest volume of disconnections.
	Trouble Tickets	Provides trouble ticket trending charts, and performance indicators, for particular accounts, and for accounts with selected common characteristics.
	Customer Satisfaction	Provides customer satisfaction trending charts, and performance indicators, for particular accounts, and for accounts with selected common characteristics.

## D.6.5 Oracle's Siebel CME Data Requirements

The data requirements for each of the Communications, Media and Energy dashboards are detailed in [Table D-4](#).

**Table D-4 Data Requirements for Communications, Media and Energy Dashboards**

Dashboard	Function
Loyalty Management	This dashboard uses customer scores generated from any third-party predictive modeling application offering the following predictive models: Customer Lifetime Value, Churn Propensity, Up-Sell Propensity, Cross-Sell Propensity, and Financial Risk Propensity. Scores must be generated for each Siebel Account, integrated into the Siebel Transaction Database, and then written to the Oracle Business Analytics Warehouse. This dashboard uses the Oracle Business Intelligence Customers subject area. See <a href="#">Section D.5, "Developing and Deploying Predictive Scores"</a> for more information.
Revenue Management	This dashboard uses data generated by Oracle's Siebel Communications, Media and Energy order management and interactive selling functionality. No specific data requirements are required beyond the data generated from these Siebel modules. This dashboard uses the Oracle Business Intelligence Orders and Products subject areas.
Account Management	This dashboard uses data generated by Oracle's Siebel Communications, Media and Energy order management, interactive selling, and service functionality. No specific data requirements are required beyond the data generated from these Siebel modules. This dashboard uses the Oracle Business Intelligence Orders, Products, Service Request and Customer Satisfaction subject areas.

## D.6.6 Oracle Pharma Sales Analytics Dimensions

Although the following dimensions are used in all subject areas, this topic describes the configuration necessary for Pharma Analytics applications. For more information, please refer to *Siebel Life Sciences Guide Version 8.0 Appendix B: Configuring Data for Siebel Pharma Analytics*.

### D.6.6.1 Positions Dimension

A sales territory is defined in Group Administration–Positions by a Siebel position. Creating parent positions creates the sales force hierarchy. Up to 10 levels of sales force hierarchy are supported by the application. Employees should be assigned to positions to populate employee hierarchy.

Position Types need to be set up according to compensation type (Rx or sales) only at the sales territory level. A district manager does not need to have a Position Type assigned to it. Sales Allocation needs to be exposed on the list to enter script compensation percentages (Rx or Sales) associated with each territory. For example, if all sales representatives receive 100% of the Rx on a ZIP Code, no action is needed or Position Type = Sales Representative can be assigned to the position.

Seed data on the Position Type list of values has been enhanced to include types for mirror, job share, and swat. Typically, both mirror and job share represent a position that receives less than 100% of the total scripts on a ZIP Code.

### D.6.6.2 Alignments Dimension

A sales territory alignment is the relationship of ZIP Code-to-territory or brick-to-territory. The alignment relationship is created in Oracle's Siebel Assignment Manager under Assignment Administration–Territories, as shown in [Table D-5](#).

**Table D-5 Sales Territory Alignment**

Relationship	Criteria	Comments
Contact ZIP to Territory	Contact ZIP Code	Use contact primary address ZIP Codes. Do not use ranges of ZIP Codes (that is, enter unique ZIP Codes as low and high values).  Do not enter duplicate ZIP Codes.
Account ZIP to Territory	Account ZIP Code	Do not use ranges of ZIP Codes (that is, enter unique ZIP Codes as low and high values).  Do not enter duplicate ZIP Codes.
Contact Brick to Territory	Contact Brick	Use contact primary address brick. Do not use ranges of bricks (that is, enter unique bricks as low and high values).  Do not enter duplicate ZIP Codes.
Account Brick to Territory	Account Brick	Do not use ranges of bricks (that is, enter unique bricks as low and high values).  Do not enter duplicate ZIP Codes.
Account to Territory	Account	Do not enter duplicate accounts.
Contact to Territory	Contact	Do not enter duplicate contacts.

### D.6.6.3 Products Dimension

The product hierarchy requires customer products (products of the company who licensed the software) to have predefined product types as shown in [Table D-6](#).

**Table D-6 Customer Products Predefined Product Types**

Product Level	Product Type	Example
3	Sample	Aracid 400 MG
2	Detail	Aracid
No Level	Sub Market	COPD
1	Market	Asthma

**Note:** Competitive products should use the product type Competitor. Competitor product hierarchies are set up using parent product relationships exclusively and should not have product levels assigned to them.

### D.6.6.4 Product Costs Dimension

Product costs for customer products (that is, products of the company that licensed the software) require population in the Product Administration, Product Form, as shown in [Table D-7](#).

**Table D-7 Product Costs For Customer Products**

Product Type	Field to be Populated
Sample	Sample Cost
Detail	Avg. Promo Cost

**Table D-7 (Cont.) Product Costs For Customer Products**

Product Type	Field to be Populated
Promotional Item Cost	Sample Cost

## D.6.7 Dimensions Specific to Subject Areas in Oracle Pharma Sales Analytics and Oracle Pharma Marketing Analytics

This section discusses the subject areas used by Pharma Analytics. For more information, please refer to *Siebel Life Sciences Guide Version 8.0 Appendix B: Configuring Data for Siebel Pharma Analytics*.

### D.6.7.1 Pharma Sales Effectiveness

This subject area is focused on syndicated data analytics.

The specific configuration required for the syndicated data depends on your data types, and the Analytics application and reports that you have licensed. The Data Loading Matrix table is the basis of prebuilt reports. The syndicated data loading matrix populates both base and derived metrics used in Pharmaceutical Sales Analytics.

### D.6.7.2 Pharma Product Categories

Oracle Pharma Sales Analytics and Oracle Pharma Marketing Analytics supports custom and prebuilt product category trees to allow roll-up of syndicated data by alternative hierarchies. To populate a custom category, first create a Catalog in Catalogue Administration, and create categories and subcategories as part of the catalogue. [Table D-8](#) lists the categories that need to have the Usage Type field populated in the Catalog Admin Category Detail list.

**Table D-8 Hierarchy Categories to be Populated in Pharma Analytics**

Usage Type Code	Hierarchy Category
ATC	Anatomical Therapeutic Class
Chemical	Chemical
Application Form	Product application
USC	User-defined codes and custom hierarchies

### D.6.7.3 Pharma Promotional Effectiveness

This subject area combines call activity data with syndicated data to analyze effectiveness of call activity.

Call Activity analysis records are derived from submitted call activity records stored in S\_EVT\_ACT in the Oracle Business Analytics Warehouse, where they are stamped with the ZIP Code or brick where the activity took place—that is, the Contact primary address's ZIP code/brick or the Account ZIP Code/brick. Allocation of these ZIP Code/brick records should be done by Assignment Manager rules to make sure that they are correctly allocated. Assignment Manager rules must match the Contact or Account primary address ZIP Codes or bricks. Otherwise, data integrity is not maintained.

Only calls that have status Submitted on the Pharma Professional Call Form are brought over from the Oracle Business Analytics Warehouse to the Oracle Business Analytics Warehouse.

**D.6.7.4 Pharma Medical Education Effectiveness**

This subject area combines measures from MedEd and Syndicated Data to measure effectiveness of medical education events used on Medical Education Analytics.

Only MedEd events with the status Completed on the Pharma ME Event List are extracted from Oracle Business Analytics Warehouse to populate the Oracle Business Analytics Warehouse.

MedEd Event costs are based on costs of activities in the Pharma ME Event Activity List. Costs are allocated based on MedEd Team cost allocation, and promoted products Cost Allocation on the MedEd event.

Costs are solely based on physician invitees with the status Attended in the Pharma ME Event Professional Invitee Session List.

Control groups are based on physicians who have the same contact ranking as attendee physicians within the same sales territory at the time of the event, but who did not attend the event.

**D.6.7.5 Pharma Objectives Achievement**

This subject is used to measure achievement and results for pharma call activity and Rx/sales targets. It is based on Pharma Objectives.

Objectives need to have a Unit populated in Retail Objective Form. Actual target numbers per contact and account need to be populated in the Pharma Campaign Target Account List or the Pharma Campaign Target Professional List Toggle.



---

---

# Oracle Business Analytics Warehouse for Life Sciences Data Considerations

This chapter contains the following topics:

- [Section E.1, "What's New for Pharma Analytics in Version 7.9"](#)
- [Section E.2, "Oracle Business Analytics Warehouse for Life Sciences Overview"](#)
- [Section E.3, "Importing Syndicated Data into Oracle Business Analytics Warehouse"](#)
- [Section E.4, "Syndicated Data Flat File Formats"](#)
- [Section E.5, "Life Sciences Data Loading Issues with Oracle Business Analytics Warehouse"](#)
- [Section E.6, "Incremental Updates in the Oracle Business Analytics Warehouse LS Dimension Tables"](#)

## E.1 What's New for Pharma Analytics in Version 7.9

This section describes the new features for Pharma Analytic, Version 7.9.

- New support for loading Syndicated Data with multiple Alignment Types.

Syndicated data can now be loaded with different Alignment Types to support the requirements of each Syndicated data type being loaded.

- Location of Alignment Version flag in DAC has been changed.
  - The creation of historical Alignment Dimension hierarchy versions is controlled using a flag set within the DAC console. In Pharma Analytics 7.9 the location of this Alignment Version flag has moved to the Source System Parameters screen within DAC's Design area.

## E.2 Oracle Business Analytics Warehouse for Life Sciences Overview

This section describes the key features and architecture of the Oracle Business Analytics Warehouse for Life Sciences Syndicated and Call Activity data.

- ETL for direct load of syndicated data.

Syndicated data can be directly loaded into the Oracle Business Analytics Warehouse.

A set of prebuilt processes against common external syndicated data types allows direct load of syndicated data, such as physician plan level Rx data, physician level Rx data, indirect sales data, and weekly Rx data.

- Syndicated data staging architecture.
  - Syndicated data is loaded against a staging architecture. The staging table architecture uses a denormalized time dimension that improves data loading by a factor of how many periods are loaded.
  - A set of cross-reference files for external key matching are supplied to resolve cross referencing and loading external syndicated data where the accounts, products, territories and contacts IDs are different from the Siebel IDs.
- Syndicated data and Call Activity data single grain fact tables.
  - Single grain syndicated data star schemas for power users and operational users.
  - Operational syndicated data star schemas for operational reporting.
  - Single grain call activity fact tables - account calls, account-attendee calls, and contact calls.
- Syndicated data Moving Annual Total (MAT) metrics calculated on ETL.
  - Calculation of rolling period calculations takes advantage of the syndicated data staging architecture to populate common pharmaceutical time aggregate metrics within the Oracle Business Analytics Warehouse.
- ETL for syndicated data market aggregation.
  - ETL process to load the Syndicated Data market fact tables based on aggregations on the product-market hierarchy. Minimizes the requirement to purchase pre-aggregated data by market.
- Set of prebuilt aggregation tables for high performance reporting.
  - Sales level aggregation tables for product-based syndicated data facts.
  - Sales level aggregation tables for market-based syndicated data facts.
  - Call activity aggregation tables for reports requiring top-level sales level activities.
  - De-duplication tables to de-duplicate data when the same contact is assigned to multiple territories for the district level for Physician Plan Level Rx syndicated data only.
- Mini dimensions.
  - Account and contact rank mini dimensions.
  - Representative specialty LOV dimensions.
  - W\_POSTN\_CON\_D used only for contact primary address.

## E.3 Importing Syndicated Data into Oracle Business Analytics Warehouse

Pharmaceutical companies purchase weekly and monthly sales and prescription data, known as syndicated data, from third-party vendors such as IMS, NDC, and CegeDim. Syndicated data vendors acquire data from drug wholesalers and retailers on a daily, weekly, and monthly basis and compile a master file of customers (wholesalers, pharmacies, hospitals, and doctors) and sales or prescription transactions for customers. Measures include indirect sales, indirect units, and prescriptions, and differ by vendor and periodicity. It is used for sales force analysis reporting and customer targeting.

The data is derived from panels of physicians, pharmacies, and so on, and projected nationally. Since the panels may change on a monthly basis, syndicated data suppliers tend to change their projections of sources on a monthly basis leading to full restatements of historical data. Thus, pharmaceutical companies are required to refresh fully the data in their data warehouses. In addition, weekly data requires incremental loading.

After it is delivered by the vendor, the syndicated data must be fully reloaded into the Oracle Business Analytics Warehouse in a timely manner and made available to users in order for them to make use of sales force analysis reporting.

This section includes the following topics:

- [Section E.3.1, "Syndicated Loading Definitions"](#)
- [Section E.3.2, "Data Types Supported in the Oracle Business Analytics Warehouse"](#)
- [Section E.3.3, "Loading Syndicated Data into the Oracle Business Analytics Warehouse"](#)

### E.3.1 Syndicated Loading Definitions

[Table E-1](#) provides terms and definitions related to syndicated loading.

**Table E-1 Syndicated Loading Definitions**

Term	Definition
Syndicated data	Third-party data from vendors that shows sales and prescription results for client defined markets.
Brick	Micro sales geographic area defined by vendor that contains prescription and sales trends for clients' product groupings or markets. Bricks do not exist in the US where the micro sales geographic area is commonly the postal code or zip code.
NRx	Abbreviation of new prescriptions. A new prescription is defined as dispensed prescriptions given a new number by the pharmacy, not necessarily new therapy for the patient.
TRx	Abbreviation of total prescriptions. $TRx = NRx + Refills$ . After the first time a prescription is filled, when it is refilled, the pharmacy refers back to the previous drug ID number and notes this as a refill.
Indirect sales	Total drug wholesalers product sales values to drug retailers (pharmacies) by brick or zip code. Sales values are calculated as units multiplied by client-selected price, where the price is the warehouse acquisition cost unless the client has chosen to have a different price applied to its direct sales or government depot sales.
Indirect units	Total drug wholesalers product sales units to drug retailers (pharmacies) by brick or zip code. Sales unit quantity is determined by multiplying package factor to obtain unit value (for example, one carton $\times$ 24 bottles = 24 units).

### E.3.2 Data Types Supported in the Oracle Business Analytics Warehouse

The Oracle Business Analytics Warehouse supports multiple data types defined by data source type and periodicity, as described in [Table E-2](#).

**Table E-2 Data Types Supported in Oracle Business Analytics Warehouse**

Data Type	Description
Prescription data by contact	Monthly NRx and TRx data for client defined markets that include competitor products data by physician.

**Table E–2 (Cont.) Data Types Supported in Oracle Business Analytics Warehouse**

<b>Data Type</b>	<b>Description</b>
Indirect sales brick level	Monthly wholesaler sales and unit values data for client defined markets that include competitor products aggregated by brick.
Indirect sales account level	Monthly wholesaler sales and unit values for company products by pharmacy.
Indirect sales zip level	Monthly wholesaler sales and unit values for client defined markets that include competitor products aggregated by postal code.
Direct sales account level	Monthly direct factory sales data and unit values to wholesalers for company's products by wholesaler.
Direct sales brick level	Monthly direct factory sales data and unit values to wholesalers for company's products by brick.
Direct sales zip level	Monthly direct factory sales data and unit values to wholesalers for company's products by postal code.
Weekly prescription data	Weekly NRx and TRx data for client defined markets that include competitor products by physician.
Plan level prescription data	Monthly prescription data by managed care plan for client defined markets that includes competitor products.
Sales market	<b>Incentives.</b> Monthly incentive compensation data sourced from internal incentive compensation systems and loaded as a new data type in the Oracle Business Analytics Warehouse.
Sales market	<b>Modified.</b> Monthly incentive compensation data sourced from internal incentive compensation systems and loaded as a new data type in the Oracle Business Analytics Warehouse.
Physician plan level Rx data	Monthly prescription data for physicians associated with a managed care plan.
Prescription data by zip code	Monthly prescription data for client defined markets that includes competitor products aggregated by postal code.
Prescription data by brick	Monthly prescription data for client defined markets that include competitor products aggregated by brick.

### E.3.3 Loading Syndicated Data into the Oracle Business Analytics Warehouse

The following options are available for loading syndicated data into the Oracle Business Analytics Warehouse:

- [Section E.3.3.1, "Loading From a Flat File Source Using Syndicated Data Staging Tables and Cross-Referencing Tables"](#)
- [Section E.3.3.2, "Loading From a Flat File Source Using Syndicated Market Staging Tables"](#)

#### E.3.3.1 Loading From a Flat File Source Using Syndicated Data Staging Tables and Cross-Referencing Tables

This option is supported in Oracle Business Intelligence Applications Version 7.8.3. To load syndicated data with this option, you have to prepare external data files for syndicated data and cross-reference data as described in tables [Table E–4](#) to [Table E–19](#) in section [Section E.4, "Syndicated Data Flat File Formats"](#).

The ETL process will load syndicated data source files as a full load each time, but cross-reference data files will be loaded incrementally. Cross-reference data files must contain only new and updated information each time to support proper incremental

loading. When the cross-reference data files are prepared, the data must be already loaded into the transactional database so the proper Siebel row IDs and the corresponding external source keys are resolved and provided in the data files.

Setting the correct alignment type to be used with the syndicated data is critical to loading data through the flat files. The Alignment Type is set in the external file `AlignmentType.csv` which is installed during the Oracle Business Intelligence Applications installation in the folder. Before running the ETL, the Alignment Type must be set to one of the following options under `ALIGN_TYPE` field.

- Zipcode
- Account/Contact
- Brick

Then, `AlignmentType.csv` file should be copied into the `SrcFiles` folder on Informatica Server machine, for example, `C:\Program Files\Informatica PowerCenter 7.1.4\Server\SrcFiles`.

In previous releases, only one Alignment Type could be set for use with all Syndicated Data Types. In Release 7.9, multiple Alignment Types can be set for different Syndicated Data Types.

For example, Physician Rx data can be loaded with an Account/Contact Alignment Type while Brick level Indirect Sales data can be loaded with a Brick Alignment Type. In addition, the same target Syndicated data table (for example the Physician Rx table) can be loaded with data defined in multiple Alignment Types such as both Account/Contact and Zipcode.

The Syndicated Data Type and Alignment Type rule mappings are set in the `AlignmentType.csv` file and this data is loaded into the `W_ALIGN_TYPE_G` table at the start of the ETL process for use in the Syndicated data load. The format of the `AlignmentType.csv` file is shown below.

---



---

**Note:** Before running the ETL, the values in the "ALIGN\_TYPE" column in this file should be updated to reflect the correct Alignment Types to be used with the appropriate Syndicated Data Types listed in the "DATA\_SRC\_CD" column.

---



---

**Table E-3** *AlignmentType.csv Mapping File*

ROW_WID	DATA_SRC_CD	ALIGN_TYPE
1	RXBrk	Brick
2	RXEVM	Zipcode
3	RXPT	Account/Contact
4	RXPrf	Account/Contact
5	RXSMI	Zipcode
6	RXSMM	Zipcode
7	RXXPT	Account/Contact
8	RXZip	Zipcode
9	SlsDirAct	Account/Contact
10	SlsIndAct	Account/Contact

**Table E-3 (Cont.) AlignmentType.csv Mapping File**

ROW_WID	DATA_SRC_CD	ALIGN_TYPE
11	SlsIndBrk	Brick
12	SlsIndZip	Zipcode

You can change the Alignment Types only when you choose to run the full ETL load process.

Once the syndicated data and cross-reference source files are prepared, then the data can be loaded directly into Oracle Business Analytics Warehouse staging tables. If any invalid data source type is used in the syndicated data files, then the ETL will load as it is. You have to prepare the correct data source type in the data files. Only the data that should not be nullable but is null in data files will be filtered out separately so you can review the rejected data for the next ETL run.

When the syndicated data is loaded from the Oracle Business Analytics Warehouse staging tables to target base tables, such as W\_SYNDD\_RX\_F, the external source keys are replaced with Oracle Business Intelligence keys and the data is loaded in the normalized format.

Then, based on what you defined as a primary alignment type above, the syndicated data in base target tables is replicated into the proper owner position and loaded into the main target tables at the territory level, such as W\_SYND\_RX\_T\_F, which contains all base table attributes plus ranking and position information.

**E.3.3.1.1 Flat File Data Population Rules for Populating TRx Data** This section provides information about various major columns in the W\_SYNDD\_RX\_F table related to populating TRx data.

- INTEGRATION\_ID
  - Data type is varchar(30).
  - Value can be any unique number or combination of values coming from the external file source with the appended postfix -##, such as -1, if the data is corresponding to the first month of bulk syndicated data, and -2 if the data is corresponding to the second month of bulk data, and so on.
  - INTEGRATION\_ID manipulation is done by LS\_EXP\_FLATFILE\_GENERATE\_INTEGRATION\_ID transformation. It is required because the incoming INTEGRATION\_ID in flat file represents 26 different bulk loads in denormalized format. Therefore, the format must be changed to normalized for the Oracle Business Analytics Warehouse target table.
- DATASOURCE\_NUM\_ID
  - ETL run specific (a numeric value that indicates the source for the data).
  - Used in standard Siebel mappings to ensure each record carries a value indicating its source.
  - Used in source qualifiers to ensure that lookups on dimension tables when populating fact tables area referencing data that was loaded from a uniform data source.
  - Data source number 1 is reserved for the Siebel transactional database. The external source should use a number other than 1.
  - For syndicated data flat file loads, the external source number (other than 1) is loaded into staging tables. When the data is loaded from the staging tables to

target tables and external source keys are matched to Siebel IDs, the data source number is converted as 1 in the target table.

- Combination of INTEGRATION\_ID and DATASOURCE\_NUM\_ID must be a unique value (mandated by a unique index on both columns).

This means that the system will not accept the same record from the same source with identical unique IDs.

- DATA\_LEVEL\_WID

- Stores a foreign key in the W\_LOV\_UD table for the record corresponding to that syndicated data source type for the data.
- Value comes from DATA\_SRC\_CD in the flat file. For example, in the table W\_SYNDD\_RX\_F, the proper data source type defined in flat file must be one of the following:
  - \* RXPrf
  - \* RXZip
  - \* RXBrk
  - \* RXSMI
  - \* RXSMM
- The standard mapping uses the incoming DATA\_SRC\_CD code to do a lookup against the W\_LOV\_UD table to secure the ROW\_WID of that data source type and inserts it into the W\_SYNDD\_RX\_F table DATA\_LEVEL\_WID column.

- PAYER\_TYPE\_WID

- Stores a foreign key in the W\_INS\_PLAN\_D table for the record corresponding to that payer type of the data such as Cash, Total.
- Value comes from PAYER\_TYPE\_ID in the flat file.

- PERIOD\_TYPE\_WID

- Stores a foreign key in the W\_LOV\_D table for the record corresponding to the period for the data.
- Value comes from PERIOD\_CD in the flat file, such as Month, Week.
- The standard mapping uses the incoming PERIOD\_CD code to do a lookup against the W\_LOV\_D table to secure the ROW\_WID of that period type and inserts it into the W\_SYNDD\_RX\_F table PERIOD\_TYPE\_WID column

- GEO\_WID

- GEO\_WID is a foreign key to the W\_GEO\_D dimension, which contains a geographical hierarchy based on zip code as the lowest level of detail. It is preconfigured to allow users to roll up a TRx measure to levels such as city, state or province, county, country, and continent.
- If the flat file source of TRx data load is zip code level, then the ZIPCODE field in the flat file must be provided. Then, the value is performed using the following logic to get loaded in the target GEO\_WID:
  - \* MPLT\_GEO\_WID\_ZIPCODE takes zip code as the incoming value, and the literal NO\_DUP\_FLG = 'N' ports to do a lookup against W\_GEO\_D for any record where the ZIPCODE column matches the incoming zip code value and the DUP\_ZIPCODE column = 'N'. (Sample data in the W\_GEO\_

D table contains multiple city records per zip code, only one of which is marked as DUP\_ZIPCODE = 'N'). The mapping also ensures that even with multiple matches only the first record is retrieved, and with no matches, the data comes back with the appropriate Unspecified code.

- CON\_GEO\_WID
  - CON\_GEO\_WID is a foreign key to the W\_GEO\_D dimension, which contains a geographical hierarchy based on zip code as the lowest level of detail.
  - The following three attributes, city, country, and zip code, are the main keys to determine CON\_GEO\_WID, and are brought from flat file CON\_CITY, CON\_COUNTRY, and CON\_ZIPCODE fields.
  - If the flat file source of TRx data load is contact level, then CON\_CITY, CON\_COUNTRY, and CON\_ZIPCODE fields in flat file must be provided. Then, the value is performed the following logic to get loaded in target CON\_GEO\_WID:
    - \* LS\_EXP\_FLATFILE\_CITY\_COUNTRY\_ZIP\_CHKNULL and LS\_MPLT\_FLATFILE\_GEO\_WID use the combination of city, country, and zip code to retrieve the first available record from W\_GEO\_D even if duplicates exist and regardless of the NO\_DUP\_FLG.

- ALIGN\_WID
  - Depending on your selected alignment item type, the following transformation is performed:

- \* When alignment item type is Zipcode:

```
IIF(DATA_SRC_CD = 'RXPrf', IN_CON_GEO_WID,
    IIF(DATA_SRC_CD = 'RXBrk' OR DATA_SRC_CD = 'RXSMI' OR DATA_SRC_CD =
        'RXSMM' OR DATA_SRC_CD = 'RXZip', IN_GEO_WID,
        ETL_UNSPEC_NUM))
```

- \* When alignment item type is Account/Contact:

```
IIF(DATA_SRC_CD = 'RXBrk' OR DATA_SRC_CD = 'RXPrf' OR DATA_SRC_CD =
    'RXSMI' OR DATA_SRC_CD = 'RXSMM' OR DATA_SRC_CD = 'RXZip', IN_CONTACT_
    WID, ETL_UNSPEC_NUM)
```

- \* When alignment item type is Brick:

```
IIF(DATA_SRC_CD = 'RXPrf', IN_CON_AREA_WID,
    IIF(DATA_SRC_CD = 'RXBrk' OR DATA_SRC_CD = 'RXSMI' OR DATA_SRC_CD =
        'RXSMM' OR DATA_SRC_CD = 'RXZip', IN_AREA_WID,
        ETL_UNSPEC_NUM))
```

The resulting value is used to populate the ALIGN\_WID column in the fact table, which should also be found in W\_ALIGNMT\_DH table ALIGN\_WID column.

Based on this ALIGN\_WID column, Oracle Business Analytics Warehouse replicates proper territory and populates W\_SYND\_RX\_T\_F table as well.

---



---

**Note:** This section discusses the W\_SYNDD\_RX\_F table as an example only. In the Oracle Business Analytics Warehouse there are six tables used for the different syndicated data sources which are loaded in the same manner: W\_SYNDD\_DS\_F, W\_SYNDD\_IDS\_F, W\_SYNDD\_RX\_F, W\_SYNDD\_PT\_F, W\_SYNDD\_XPT\_F, and W\_SYNDD\_W\_F tables.

---



---

### E.3.3.2 Loading From a Flat File Source Using Syndicated Market Staging Tables

The Oracle Business Analytics Warehouse supports loading syndicated market data using flat files. You have to prepare flat files of syndicated market data source and cross-referencing data source files as described in tables [Table E-4](#) to [Table E-19](#) in section [Section E.4, "Syndicated Data Flat File Formats"](#).

By default this option is disabled. To use this feature, you must turn on the option manually in the DAC client. Turning on this option allows the flat file syndicated market data to be loaded directly into the Oracle Business Analytics Warehouse staging table W\_SYNDM\_RX\_FS, and then loaded into the target table W\_SYNDM\_RX\_F.

#### To load syndicated market data using flat files

1. Open the DAC client, and navigate to Design, then Task.
2. Enter the following query:

```
LS Load into SyndicatedMarket Fact for*
```

Eleven tasks are returned.

3. Deactivate the following tasks, which, by default, are turned on:
  - LS Load into SyndicatedMarket Fact for Calculated Exponent Plantrak
  - LS Load into SyndicatedMarket Fact for Calculated Indirect Sales
  - LS Load into SyndicatedMarket Fact for Calculated Plantrak
  - LS Load into SyndicatedMarket Fact for Calculated Rx
  - LS Load into SyndicatedMarket Fact for Calculated Weekly Early View

---



---

**Note:** Do not deactivate LS Load into SyndicatedMarket Fact for Calculated Direct Sales.

---



---

4. Activate the following tasks:
  - LS Load into SyndicatedMarket Fact for Exponent Plantrak
  - LS Load into SyndicatedMarket Fact for Indirect Sales
  - LS Load into SyndicatedMarket Fact for Plantrak
  - LS Load into SyndicatedMarket Fact for Rx
  - LS Load into SyndicatedMarket Fact for Weekly Early View
5. Save your work.

## E.4 Syndicated Data Flat File Formats

This section includes descriptions of the following flat files used in the syndicated data load process:

- [Section E.4.1, "Syndicated Data Flat File for Direct Sales Data"](#)
- [Section E.4.2, "Syndicated Data Flat File for Indirect Sales Data"](#)
- [Section E.4.3, "Syndicated Data Flat File for Prescription Data"](#)
- [Section E.4.4, "Syndicated Data Flat File for Plan Level Rx Data"](#)
- [Section E.4.5, "Syndicated Data Flat File for Physician Plan Level Rx Data"](#)
- [Section E.4.6, "Syndicated Data Flat File for Weekly Early View Data"](#)
- [Section E.4.7, "Syndicated Market Flat File for Indirect Sales Data"](#)
- [Section E.4.8, "Syndicated Market Flat File for Prescription Data"](#)
- [Section E.4.9, "Syndicated Market Flat File for Plan Level Rx Data"](#)
- [Section E.4.10, "Syndicated Market Flat File for Physician Plan Level Rx Data"](#)
- [Section E.4.11, "Syndicated Market Flat File for Weekly Early View Data"](#)
- [Section E.4.12, "Flat File Cross-Reference Data Population Rules"](#)

### E.4.1 Syndicated Data Flat File for Direct Sales Data

[Table E-4](#) provides information about the syndicated data flat file for Direct Sales data.

**Table E-4** *Syndicated Data Flat File - Direct Sales Data*

Field Name	Data Type	Field Required	Field Description
DATASOURCE_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	Enter the following Siebel data source type code: SlsDirAct
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
AMT_DT	VARCHAR2(15)		Exchange date for currency code. Date format YYYYMMDD.
ACCNT_ID	VARCHAR2(30)		External account ID. <b>Note:</b> Corresponding Siebel S_ORG_EXT.ROW_ID should be provided in cross-reference file.
ACCNT_AREA_ID	VARCHAR2(30)		External brick ID of account's primary address for account level. <b>Note:</b> Corresponding Siebel S_REGION_EXT.ROW_ID (S_CON_ADDR.BRICK_ID) should be provided in cross-reference file.
ACCNT_CITY	VARCHAR2(50)		Account city. Initial capital format; for example, Short Hills.

**Table E-4 (Cont.) Syndicated Data Flat File - Direct Sales Data**

Field Name	Data Type	Field Required	Field Description
ACCNT_COUNTRY	VARCHAR2(30)		Account country. Initial capital format; for example, France; or abbreviate format, such as USA.
ACCNT_ZIPCODE	VARCHAR2(30)		Account zip code.
AMT_CURCY_CD	VARCHAR2(30)		Currency code for Amount.
PAYER_TYPE_ID	VARCHAR2(30)		External payer type ID. This column is used to define payment type, such as Cash, Government, Third Party, Total, and so on. <b>Note:</b> Corresponding Siebel S_INS_PLAN.ROW_ID should be provided in cross-reference file.
PERIOD_CD	VARCHAR2(30)		Enter the following Siebel period code: Month
PROD_ID	VARCHAR2(30)		External product ID. <b>Note:</b> Corresponding Siebel S_PROD_INT.ROW_ID should be provided in cross-reference file.
S_AMT_M01	NUMBER(22,7)		Stores sales amounts of current month data.
S_AMT_M02	NUMBER(22,7)		Stores sales amounts of previous month data.
S_AMT_M03 - S_AMT_M06	NUMBER(22,7)		
S_UNIT_M01	NUMBER(22,7)		Stores sales units of current month data (for example, 20050301 data).
S_UNIT_M02	NUMBER(22,7)		Stores sales units of previous month data (for example, 20050201 data).
S_UNIT_M03	NUMBER(22,7)		Stores sales units of 20050101 data.
S_UNIT_M04 - S_UNIT_M026	NUMBER(22,7)		

## E.4.2 Syndicated Data Flat File for Indirect Sales Data

Table E-5 provides information about the syndicated data flat file for Indirect Sales data.

**Table E-5 Syndicated Data Flat File - Indirect Sales Data**

Field Name	Data Type	Field Required	Field Description
DATASOURCE_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	Enter the following Siebel data source type code: SlsIndAct (Indirect sales account level) SlsIndZip (Indirect sales zip code level) SlsIndBrk (Indirect sales brick level)

**Table E-5 (Cont.) Syndicated Data Flat File - Indirect Sales Data**

Field Name	Data Type	Field Required	Field Description
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
AMT_DT	VARCHAR2(15)		Exchange date for currency code. Date format YYYYMMDD.
ACCNT_ID	VARCHAR2(30)		External account ID. <b>Note:</b> Corresponding Siebel S_ORG_EXT.ROW_ID should be provided in cross-reference file.
ACCNT_AREA_ID	VARCHAR2(30)		External brick ID of account's primary address for account level. <b>Note:</b> Corresponding Siebel S_REGION_EXT.ROW_ID (S_CON_ADDR.BRICK_ID) should be provided in cross-reference file.
ACCNT_CITY	VARCHAR2(50)		Account city. Enter value if DATA_SRC_CD is account level. Initial capital format; for example, Short Hills.
ACCNT_COUNTRY	VARCHAR2(30)		Account country. Enter value if DATA_SRC_CD is account level. Initial capital format; for example, France; or abbreviate format, such as USA.
ACCNT_ZIPCODE	VARCHAR2(30)		Account zip code. Enter value if DATA_SRC_CD is account level.
AMT_CURCY_CD	VARCHAR2(30)		Applies to syndicated data only. Currency code for Amount.
AREA_ID	VARCHAR2(30)		External syndicate brick ID. Enter value if DATA_SRC_CD is brick level. <b>Note:</b> Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
PAYER_TYPE_ID	VARCHAR2(30)		External payer type ID. This column is used to define payment type, such as Cash, Government, Third Party, Total, and so on. <b>Note:</b> Corresponding Siebel S_INS_PLAN.ROW_ID should be provided in cross-reference file.
PERIOD_CD	VARCHAR2(30)		Enter the following Siebel period code: Month
PROD_ID	VARCHAR2(30)		Applies to syndicated data only. External product ID. <b>Note:</b> Corresponding Siebel S_PROD_INT.ROW_ID should be provided in cross-reference file.
ZIPCODE	VARCHAR2(30)		Syndicate zip code. Enter value if DATA_SRC_CD is zip code level.

**Table E-5 (Cont.) Syndicated Data Flat File - Indirect Sales Data**

Field Name	Data Type	Field Required	Field Description
S_AMT_M01	NUMBER(22,7)		Stores sales amounts of current month data.
S_AMT_M02	NUMBER(22,7)		Stores sales amounts of previous month data.
S_AMT_M03	NUMBER(22,7)		Stores sales amounts of 20050101 data.
S_AMT_M04 - S_AMT_M026	NUMBER(22,7)		
S_UNIT_M01	NUMBER(22,7)		Stores sales units of current month data.
S_UNIT_M02	NUMBER(22,7)		Stores sales units of previous month data.
S_UNIT_M03	NUMBER(22,7)		Stores sales units of 20050101 data.
S_UNIT_M04 - S_UNIT_M26	NUMBER(22,7)		

### E.4.3 Syndicated Data Flat File for Prescription Data

Table E-6 provides information about the syndicated data flat file for Prescription data.

**Table E-6 Syndicated Data Flat File - Prescription Data**

Field Name	Data Type	Field Required	Field Description
DATASOURCE_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	RXPrf (Prescription data by Contact) RXZip (Prescription data by Zip Code) RXBrk (Prescription data by Brick) RXSMI (Sales Market – Incentives) RXSMM (Sales Market – Modified)
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
AREA_ID	VARCHAR2(30)		External syndicate brick ID. Enter value if DATA_SRC_CD is brick level. <b>Note:</b> Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
CONTACT_ID	VARCHAR2(30)		External contact ID. <b>Note:</b> Corresponding Siebel S_CONTACT.ROW_ID should be provided in cross-reference file.
CON_AREA_ID	VARCHAR2(30)		External syndicate brick ID of contact's primary address. Enter value if DATA_SRC_CD is brick level. <b>Note:</b> Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.

**Table E-6 (Cont.) Syndicated Data Flat File - Prescription Data**

Field Name	Data Type	Field Required	Field Description
CON_CITY	VARCHAR2(50)		Contact city. Enter value if DATA_SRC_CD is contact level.  Initial capital format; for example, Short Hills
CON_COUNTRY	VARCHAR2(30)		Contact country. Enter value if DATA_SRC_CD is contact level.  Initial capital format; for example, France; or abbreviate format, such as USA.
CON_ZIPCODE	VARCHAR2(30)		Contact zip code. Enter value if DATA_SRC_CD is contact level.
PAYER_TYPE_ID	VARCHAR2(30)		External payer type ID. This column is used to define payment type, such as Cash, Government, Third Party, Total, and so on.  <b>Note:</b> Corresponding Siebel S_INS_PLAN.ROW_ID should be provided in cross-reference file.
PERIOD_CD	VARCHAR2(30)		Enter the following Siebel period code:  Month
PROD_ID	VARCHAR2(30)		External product ID.  <b>Note:</b> Corresponding Siebel S_PROD_INT.ROW_ID should be provided in cross-reference file.
ZIPCODE	VARCHAR2(30)		Syndicate zip code. Enter value if DATA_SRC_CD is zip code level.
CONV_FACTOR	NUMBER(22,7)		Applies to Rx data only. Used to convert Rx volume to measurable units which will be used to dollarize the Rx volume.
NRX_M01	NUMBER(22,7)		Stores New Rx of current month data.
NRX_M02	NUMBER(22,7)		Stores New Rx of previous month data.
NRX_M03	NUMBER(22,7)		Stores New Rx of 20050101 data.
NRX_M04 - NRX_M26	NUMBER(22,7)		
TRX_M01	NUMBER(22,7)		Stores Total Rx of current month data.
TRX_M02	NUMBER(22,7)		Stores Total Rx of previous month data.
TRX_M03	NUMBER(22,7)		Stores Total Rx of 20050101 data.
TRX_M04 - TRX_M26	NUMBER(22,7)		

#### E.4.4 Syndicated Data Flat File for Plan Level Rx Data

Table E-7 provides information about the syndicated data flat file for Plan Level Rx data.

**Table E-7 Syndicated Data Flat File - Plan Level Rx Data**

Field Name	Data Type	Field Required	Field Description
DATASOURCE_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	Enter the following Siebel data source type code: RXPT
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
ACCNT_ID	VARCHAR2(30)		External account and plan ID. <b>Note:</b> Corresponding Siebel S_ORG_EXT.ROW_ID should be provided in cross-reference file.
ACCNT_AREA_ID	VARCHAR2(30)		External brick ID of account's primary address. Enter value if DATA_SRC_CD is account level. <b>Note:</b> Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
ACCNT_CITY	VARCHAR2(50)		Account city. Enter value if DATA_SRC_CD is account level. Initial capital format; for example, Short Hills
ACCNT_COUNTRY	VARCHAR2(30)		Account country. Enter value if DATA_SRC_CD is account level. Initial capital format; for example, France; or abbreviate format, such as USA.
ACCNT_ZIPCODE	VARCHAR2(30)		Account zip code. Enter value if DATA_SRC_CD is account level.
PERIOD_CD	VARCHAR2(30)		Enter the following Siebel period code: Month
PROD_ID	VARCHAR2(30)		Applies to syndicated data only. External product ID. <b>Note:</b> Corresponding Siebel S_PROD_INT.ROW_ID should be provided in cross-reference file.
CONV_FACTOR	NUMBER(22,7)		Applies to Rx data only. Used to convert Rx volume to measurable units which will be used to dollarize the Rx volume.
NRX_M01	NUMBER(22,7)		Stores New Rx of current month data.
NRX_M02	NUMBER(22,7)		Stores New Rx of previous month data.
NRX_M03	NUMBER(22,7)		Stores New Rx of 20050101 data.
NRX_M04 - NRX_M26	NUMBER(22,7)		
TRX_M01	NUMBER(22,7)		Stores Total Rx of current month data.

**Table E-7 (Cont.) Syndicated Data Flat File - Plan Level Rx Data**

Field Name	Data Type	Field Required	Field Description
TRX_M02	NUMBER(22,7)		Stores Total Rx of previous month data.
TRX_M03	NUMBER(22,7)		Stores Total Rx of 20050101 data.
TRX_M04 - TRX_M26	NUMBER(22,7)		

### E.4.5 Syndicated Data Flat File for Physician Plan Level Rx Data

Table E-8 provides information about the syndicated data flat file for Physician Plan Level Rx data.

**Table E-8 Syndicated Data Flat File - Physician Plan Level Rx Data**

Field Name	Data Type	Field Required	Field Description
DATASOURCE_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	Enter the following Siebel data source type code: RXXPT
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
ACCNT_ID	VARCHAR2(30)		External account ID. <b>Note:</b> Corresponding Siebel S_ORG_EXT.ROW_ID should be provided in cross-reference file.
ACCNT_AREA_ID	VARCHAR2(30)		External brick ID of account's primary address. Enter value if DATA_SRC_CD is brick level. <b>Note:</b> Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
ACCNT_CITY	VARCHAR2(50)		Account city. Enter a value if DATA_SRC_CD is account level. Initial capital format; for example, Short Hills
ACCNT_COUNTRY	VARCHAR2(30)		Account country. Enter value if DATA_SRC_CD is account level. Initial capital format; for example, France; or abbreviate format, such as USA.
ACCNT_ZIPCODE	VARCHAR2(30)		Account zip code. Enter value if DATA_SRC_CD is account level.
CONTACT_ID	VARCHAR2(30)		External contact ID. <b>Note:</b> Corresponding Siebel S_CONTACT.ROW_ID should be provided in cross-reference file.

**Table E–8 (Cont.) Syndicated Data Flat File - Physician Plan Level Rx Data**

Field Name	Data Type	Field Required	Field Description
CON_AREA_ID	VARCHAR2(30)		External brick ID of contact's primary address. Enter value if DATA_SRC_CD is contact level.  <b>Note:</b> Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
CON_CITY	VARCHAR2(50)		Contact city. Enter value if DATA_SRC_CD is contact level.  Initial capital format; for example, Short Hills
CON_COUNTRY	VARCHAR2(30)		Contact country. Enter value if DATA_SRC_CD is contact level.  Initial capital format; for example, France; or abbreviate format, such as USA.
CON_ZIPCODE	VARCHAR2(30)		Contact zip code. Enter value if DATA_SRC_CD is contact level.
PERIOD_CD	VARCHAR2(30)		Enter the following Siebel period code: Month
PROD_ID	VARCHAR2(30)		Applies to syndicated data only. External product ID. <b>Note:</b> Corresponding Siebel S_PROD_INT.ROW_ID should be provided in cross-reference file.
CONV_FACTOR	NUMBER(22,7)		Applies to Rx data only. Used to convert Rx volume to measurable units which will be used to dollarize the Rx volume.
NRX_M01	NUMBER(22,7)		Stores New Rx of current month data.
NRX_M02	NUMBER(22,7)		Stores New Rx of previous month data.
NRX_M03	NUMBER(22,7)		Stores New Rx of 20050101 data.
NRX_M04 - NRX_M26	NUMBER(22,7)		
TRX_M01	NUMBER(22,7)		Stores Total Rx of current month data.
TRX_M02	NUMBER(22,7)		Stores Total Rx s of previous month data.
TRX_M03	NUMBER(22,7)		Stores Total Rx of 20050101 data.
TRX_M04 - TRX_M26	NUMBER(22,7)		

#### E.4.6 Syndicated Data Flat File for Weekly Early View Data

[Table E–9](#) provides information about the syndicated data flat file for Weekly Early View data.

**Table E-9 Syndicated Data Flat File - Weekly Early View Data**

Field Name	Data Type	Field Required	Field Description
DATASOURCE_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	Enter the following Siebel data source type code: RXEVM
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
CONTACT_ID	VARCHAR2(30)		External contact ID. <b>Note:</b> Corresponding Siebel S_CONTACT.ROW_ID should be provided in cross-reference file.
CON_AREA_ID	VARCHAR2(30)		External brick ID of contact's primary address. Enter value if DATA_SRC_CD is brick level. <b>Note:</b> Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
CON_CITY	VARCHAR2(50)		Contact city. Enter a value if DATA_SRC_CD is contact level. Initial capital format; for example, Short Hills
CON_COUNTRY	VARCHAR2(30)		Contact country. Enter value if DATA_SRC_CD is contact level. Initial capital format; for example, France; or abbreviate format, such as USA.
CON_ZIPCODE	VARCHAR2(30)		Contact zip code. Enter value if DATA_SRC_CD is contact level.
PAYER_TYPE_ID	VARCHAR2(30)		External payer type ID. This column is used to define payment type, such as Cash, Government, Third Party, Total, and so on. <b>Note:</b> Corresponding Siebel S_INS_PLAN.ROW_ID should be provided in cross-reference file.
PERIOD_CD	VARCHAR2(30)		Enter the following Siebel period code: Week
PROD_ID	VARCHAR2(30)		Applies to syndicated data only. External product ID. <b>Note:</b> Corresponding Siebel S_PROD_INT.ROW_ID should be provided in cross-reference file.
CONV_FACTOR	NUMBER(22,7)		Applies to Rx data only. Used to convert Rx volume to measurable units which will be used to dollarize the Rx volume.
NRX_W01	NUMBER(22,7)		Stores New Rx of current week data.

**Table E-9 (Cont.) Syndicated Data Flat File - Weekly Early View Data**

Field Name	Data Type	Field Required	Field Description
NRX_W02	NUMBER(22,7)		Stores New Rx of previous week data.
NRX_W03	NUMBER(22,7)		Stores New Rx of 20050101 data.
NRX_W04 - NRX_W13	NUMBER(22,7)		
TRX_W01	NUMBER(22,7)		Stores Total Rx of current week data.
TRX_W02	NUMBER(22,7)		Stores Total Rx of previous week data.
TRX_W03	NUMBER(22,7)		Stores Total Rx of 20050101 data.
TRX_W04 - TRX_W13	NUMBER(22,7)		

### E.4.7 Syndicated Market Flat File for Indirect Sales Data

Table E-10 provides information about the syndicated market flat file for Indirect Sales data.

**Table E-10 Syndicated Market Flat File - Indirect Sales Data**

Field Name	Data Type	Field Required	Field Description
DATASOURCE_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	Enter the following Siebel data source type code: SlsIndAct (Indirect sales account level) SlsIndZip (Indirect sales zip code level) SlsIndBrk (Indirect sales brick level)
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
MKT_AMT_DT	VARCHAR2(15)		Exchange date for currency code. Date format YYYYMMDD.
ACCNT_ID	VARCHAR2(30)		External account ID. <b>Note:</b> Corresponding Siebel S_ORG_EXT.ROW_ID should be provided in cross-reference file.
ACCNT_AREA_ID	VARCHAR2(30)		External brick ID of account's primary address for account level. <b>Note:</b> Corresponding Siebel S_REGION_EXT.ROW_ID (S_CON_ADDR.BRICK_ID) should be provided in cross-reference file.
ACCNT_CITY	VARCHAR2(50)		Account city. Enter value if DATA_SRC_CD is account level.  Initial capital format; for example, Short Hills.

**Table E-10 (Cont.) Syndicated Market Flat File - Indirect Sales Data**

Field Name	Data Type	Field Required	Field Description
ACCNT_COUNTRY	VARCHAR2(30)		Account country. Enter value if DATA_SRC_CD is account level.  Initial capital format; for example, France; or abbreviate format, such as USA.
ACCNT_ZIPCODE	VARCHAR2(30)		Account zip code. Enter value if DATA_SRC_CD is account level.
AREA_ID	VARCHAR2(30)		External syndicate brick ID. Enter value if DATA_SRC_CD is brick level.  <b>Note:</b> Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
MARKET_ID	VARCHAR2(30)		External market ID for the product.  <b>Note:</b> Corresponding Siebel S_PROD_INT.ROW_ID should be provided in cross-reference file.
MKT_AMT_CURCY_CD	VARCHAR2(30)		Currency code for Amount.
PAYER_TYPE_ID	VARCHAR2(30)		External payer type ID. This column is used to define payment type, such as Cash, Government, Third Party, Total, and so on.  <b>Note:</b> Corresponding Siebel S_INS_PLAN.ROW_ID should be provided in cross-reference file.
PERIOD_CD	VARCHAR2(30)		Enter the following Siebel period code: Month
ZIPCODE	VARCHAR2(30)		Syndicate zip code. Enter value if DATA_SRC_CD is zip code level.
MKT_S_AMT_M01	NUMBER(22,7)		Stores sales amounts of current month data.
MKT_S_AMT_M02	NUMBER(22,7)		Stores sales amounts of previous month data.
MKT_S_AMT_M03	NUMBER(22,7)		Stores sales amounts of 20050101 data.
MKT_S_AMT_M04 - MKT_S_AMT_M26	NUMBER(22,7)		
MKT_S_UNIT_M01	NUMBER(22,7)		Stores sales units of current month data.
MKT_S_UNIT_M02	NUMBER(22,7)		Stores sales units of previous month data.
MKT_S_UNIT_M03	NUMBER(22,7)		Stores sales units of 20050101 data.
MKT_S_UNIT_M04 - MKT_S_UNIT_M26	NUMBER(22,7)		

### E.4.8 Syndicated Market Flat File for Prescription Data

Table E-11 provides information about the syndicated market flat file for Prescription data.

**Table E–11 Syndicated Market Flat File - Prescription Data**

Field Name	Data Type	Field Required	Field Description
DATASOURCE_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	RXPrf (Prescription data by Contact) RXZip (Prescription data by Zip Code) RXBrk (Prescription data by Brick) RXSMI (Sales Market – Incentives) RXSMM (Sales Market – Modified)
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
AREA_ID	VARCHAR2(30)		External syndicate brick ID. Enter value if DATA_SRC_CD is brick level.  <b>Note:</b> Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
CONTACT_ID	VARCHAR2(30)		External contact ID.  <b>Note:</b> Corresponding Siebel S_CONTACT.ROW_ID should be provided in cross-reference file.
CON_AREA_ID	VARCHAR2(30)		External syndicate brick ID of contact's primary address. Enter value if DATA_SRC_CD is brick level.  <b>Note:</b> Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
CON_CITY	VARCHAR2(50)		Contact city. Enter value if DATA_SRC_CD is contact level.  Initial capital format; for example, Short Hills
CON_COUNTRY	VARCHAR2(30)		Contact country. Enter value if DATA_SRC_CD is contact level.  Initial capital format; for example, France; or abbreviate format, such as USA.
CON_ZIPCODE	VARCHAR2(30)		Contact zip code. Enter value if DATA_SRC_CD is contact level.
MARKET_ID	VARCHAR2(30)		External market ID for the product.  <b>Note:</b> Corresponding Siebel S_PROD_INT.ROW_ID should be provided in cross-reference file.
PAYER_TYPE_ID	VARCHAR2(30)		External payer type ID. This column is used to define payment type, such as Cash, Government, Third Party, Total, and so on.  <b>Note:</b> Corresponding Siebel S_INS_PLAN.ROW_ID should be provided in cross-reference file.

**Table E-11 (Cont.) Syndicated Market Flat File - Prescription Data**

Field Name	Data Type	Field Required	Field Description
PERIOD_CD	VARCHAR2(30)		Enter the following Siebel period code: Month
ZIPCODE	VARCHAR2(30)		Syndicate zip code. Enter value if DATA_SRC_CD is zip code level.
CONV_FACTOR	NUMBER(22,7)		Applies to Rx data only. Used to convert Rx volume to measurable units which will be used to dollarize the Rx volume.
MKT_NRX_M01	NUMBER(22,7)		Stores New Rx of current month data.
MKT_NRX_M02	NUMBER(22,7)		Stores New Rx of previous month data.
MKT_NRX_M03	NUMBER(22,7)		Stores New Rx of 20050101 data.
MKT_NRX_M04 - MKT_NRX_M26	NUMBER(22,7)		
MKT_TRX_M01	NUMBER(22,7)		Stores Total Rx of current month data.
MKT_TRX_M02	NUMBER(22,7)		Stores Total Rx of previous month data.
MKT_TRX_M03	NUMBER(22,7)		Stores Total Rx of 20050101 data.
MKT_TRX_M04 - MKT_TRX_M26	NUMBER(22,7)		

#### E.4.9 Syndicated Market Flat File for Plan Level Rx Data

Table E-12 provides information about the syndicated market flat file for Plan Level Rx data.

**Table E-12 Syndicated Market Flat File - Plan Level Rx Data**

Field Name	Data Type	Field Required	Field Description
DATASOURCE_ NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	Enter the following Siebel data source type code: RXPT
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
ACCNT_ID	VARCHAR2(30)		External account and plan ID. <b>Note:</b> Corresponding Siebel S_ORG_EXT.ROW_ID should be provided in cross-reference file.
ACCNT_AREA_ID	VARCHAR2(30)		External brick ID of account's primary address. Enter value if DATA_SRC_CD is account level. <b>Note:</b> Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.

**Table E-12 (Cont.) Syndicated Market Flat File - Plan Level Rx Data**

Field Name	Data Type	Field Required	Field Description
ACCNT_CITY	VARCHAR2(50)		Account city. Enter value if DATA_SRC_CD is account level.  Initial capital format; for example, Short Hills
ACCNT_COUNTRY	VARCHAR2(30)		Account country. Enter value if DATA_SRC_CD is account level.  Initial capital format; for example, France; or abbreviate format, such as USA.
ACCNT_ZIPCODE	VARCHAR2(30)		Account zip code. Enter value if DATA_SRC_CD is account level.
MARKET_ID	VARCHAR2(30)		External market ID for the product.  <b>Note:</b> Corresponding Siebel S_PROD_INT.ROW_ID should be provided in cross-reference file.
PERIOD_CD	VARCHAR2(30)		Enter the following Siebel period code:  Month
CONV_FACTOR	NUMBER(22,7)		Applies to Rx data only. Used to convert Rx volume to measurable units which will be used to dollarize the Rx volume.
MKT_NRX_M01	NUMBER(22,7)		Stores New Rx of current month data (for example, 20050301 data).
MKT_NRX_M02	NUMBER(22,7)		Stores New Rx of previous month data (for example, 20050201 data).
MKT_NRX_M03	NUMBER(22,7)		Stores New Rx of 20050101 data.
MKT_NRX_M04 - MKT_NRX_M26	NUMBER(22,7)		
MKT_TRX_M01	NUMBER(22,7)		Stores Total Rx of current month data (for example, 20050301 data).
MKT_TRX_M02	NUMBER(22,7)		Stores Total Rx of previous month data (for example, 20050301 data).
MKT_TRX_M03	NUMBER(22,7)		Stores Total Rx of 20050101 data.
MKT_TRX_M04 - MKT_TRX_M26	NUMBER(22,7)		

#### E.4.10 Syndicated Market Flat File for Physician Plan Level Rx Data

Table E-13 provides information about the syndicated market flat file for Physician Plan Level Rx data.

**Table E-13 Syndicated Market Flat File - Physician Plan Level Rx Data**

Field Name	Data Type	Field Required	Field Description
DATASOURCE_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.

**Table E-13 (Cont.) Syndicated Market Flat File - Physician Plan Level Rx Data**

Field Name	Data Type	Field Required	Field Description
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	Enter the following Siebel data source type code: RXXPT
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
ACCNT_ID	VARCHAR2(30)		External account ID. <b>Note:</b> Corresponding Siebel S_ORG_EXT.ROW_ID should be provided in cross-reference file.
ACCNT_AREA_ID	VARCHAR2(30)		External brick ID of account's primary address. Enter value if DATA_SRC_CD is brick level. <b>Note:</b> Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
ACCNT_CITY	VARCHAR2(50)		Account city. Enter a value if DATA_SRC_CD is account level. Initial capital format; for example, Short Hills
ACCNT_COUNTRY	VARCHAR2(30)		Account country. Enter value if DATA_SRC_CD is account level. Initial capital format; for example, France; or abbreviate format, such as USA.
ACCNT_ZIPCODE	VARCHAR2(30)		Account zip code. Enter value if DATA_SRC_CD is account level.
CONTACT_ID	VARCHAR2(30)		External contact ID. <b>Note:</b> Corresponding Siebel S_CONTACT.ROW_ID should be provided in cross-reference file.
CON_AREA_ID	VARCHAR2(30)		External brick ID of contact's primary address. Enter value if DATA_SRC_CD is contact level. <b>Note:</b> Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
CON_CITY	VARCHAR2(50)		Contact city. Enter value if DATA_SRC_CD is contact level. Initial capital format; for example, Short Hills
CON_COUNTRY	VARCHAR2(30)		Contact country. Enter value if DATA_SRC_CD is contact level. Initial capital format; for example, France; or abbreviate format, such as USA.
CON_ZIPCODE	VARCHAR2(30)		Contact zip code. Enter value if DATA_SRC_CD is contact level.

**Table E-13 (Cont.) Syndicated Market Flat File - Physician Plan Level Rx Data**

Field Name	Data Type	Field Required	Field Description
MARKET_ID	VARCHAR2(30)		External market ID for product. <b>Note:</b> Corresponding Siebel S_PROD_INT.ROW_ID should be provided in cross-reference file.
PERIOD_CD	VARCHAR2(30)		Enter the following Siebel period code: Month
CONV_FACTOR	NUMBER(22,7)		Applies to Rx data only. Used to convert Rx volume to measurable units which will be used to dollarize the Rx volume.
MKT_NRX_M01	NUMBER(22,7)		Stores New Rx of current month data (for example, 20050301 data).
MKT_NRX_M02	NUMBER(22,7)		Stores New Rx of previous month data (for example, 20050201 data).
MKT_NRX_M03	NUMBER(22,7)		Stores New Rx of 20050101 data (for example, 20050301 data).
MKT_NRX_M04 - MKT_NRX_M26	NUMBER(22,7)		
MKT_TRX_M01	NUMBER(22,7)		Stores Total Rx of current month data.
MKT_TRX_M02	NUMBER(22,7)		Stores Total Rx of previous month data.
MKT_TRX_M03	NUMBER(22,7)		Stores Total Rx of 20050101 data.
MKT_TRX_M04 - MKT_TRX_M26	NUMBER(22,7)		

### E.4.11 Syndicated Market Flat File for Weekly Early View Data

Table E-14 provides information about the syndicated market flat file for Weekly Early View data.

**Table E-14 Syndicated Market Flat File - Weekly Early View Data**

Field Name	Data Type	Field Required	Field Description
DATASOURCE_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
DATA_SRC_CD	VARCHAR2(30)	NOT NULL	Enter the following Siebel data source type code: RXEVM
INTEGRATION_ID	VARCHAR2(25)	NOT NULL	External integration ID.
START_DT	VARCHAR2(15)	NOT NULL	Syndicate data date format YYYYMMDD.
CONTACT_ID	VARCHAR2(30)		External contact ID. <b>Note:</b> Corresponding Siebel S_CONTACT.ROW_ID should be provided in cross-reference file.

**Table E-14 (Cont.) Syndicated Market Flat File - Weekly Early View Data**

Field Name	Data Type	Field Required	Field Description
CON_AREA_ID	VARCHAR2(30)		External brick ID of contact's primary address. Enter value if DATA_SRC_CD is brick level.  <b>Note:</b> Corresponding Siebel S_REGION.ROW_ID should be provided in cross-reference file.
CON_CITY	VARCHAR2(50)		Contact city. Enter a value if DATA_SRC_CD is contact level.  Initial capital format; for example, Short Hills
CON_COUNTRY	VARCHAR2(30)		Contact country. Enter value if DATA_SRC_CD is contact level.  Initial capital format; for example, France; or abbreviate format, such as USA.
CON_ZIPCODE	VARCHAR2(30)		Contact zip code. Enter value if DATA_SRC_CD is contact level.
PAYER_TYPE_ID	VARCHAR2(30)		External payer type ID. This column is used to define payment type, such as Cash, Government, Third Party, Total, and so on.  <b>Note:</b> Corresponding Siebel S_INS_PLAN.ROW_ID should be provided in cross-reference file.
PERIOD_CD	VARCHAR2(30)		Enter the following Siebel period code:  Week
CONV_FACTOR	NUMBER(22,7)		Applies to Rx data only. Used to convert Rx volume to measurable units which will be used to dollarize the Rx volume.
MKT_NRX_W01	NUMBER(22,7)		Stores New Rx of current data.
MKT_NRX_W02	NUMBER(22,7)		Stores New Rx of previous data.
MKT_NRX_W03	NUMBER(22,7)		Stores New Rx of 20050101 data.
MKT_NRX_W04 - MKT_NRX_W14	NUMBER(22,7)		
MKT_TRX_W01	NUMBER(22,7)		Stores Total Rx of current week data.
MKT_TRX_W02	NUMBER(22,7)		Stores Total Rx of previous week data.
MKT_TRX_W03	NUMBER(22,7)		Stores Total Rx of 20050214 data.
MKT_TRX_W04 - MKT_TRX_W13	NUMBER(22,7)		

### E.4.12 Flat File Cross-Reference Data Population Rules

The following tables provide information about flat file cross-reference data population rules:

- [Table E-15, " Flat File Cross-Reference Data Population Rules for Area \(Brick\) Data"](#)

- [Table E–16, " Flat File Cross-Reference Data Population Rules for Account and Plan Data"](#)
- [Table E–17, " Flat File Cross-Reference Data Population Rules for Payer Type Data"](#)
- [Table E–18, " Flat File Cross-Reference Data Population Rules for Contact Data"](#)
- [Table E–19, " Flat File Cross-Reference Data Population Rules for Product and Market Data"](#)

**Table E–15 Flat File Cross-Reference Data Population Rules for Area (Brick) Data**

Field Name	Data Type	Field Required	Field Description
EXT_DATASRC_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.
EXT_INTEGRATION_ID	VARCHAR2(30)	NOT NULL	External area (brick) ID. <b>Note:</b> This same ID should be used in ACCNT_AREA_ID, AREA_ID, AND CON_AREA_ID syndicated data and syndicated market flat files.
EXT_AREA_NAME	VARCHAR2(50)		Area name. <b>Note:</b> The same value should already be loaded in Siebel S_REGION.NAME.
EXT_PROVINCE	VARCHAR2(50)		Area province. <b>Note:</b> The same value should already be loaded in Siebel S_REGION.PROVINCE.
EXT_STATE	VARCHAR2(50)		Area state. <b>Note:</b> The same value should already be loaded in Siebel S_REGION.STATE. Use the format all capitals and abbreviated, such as NJ.
EXT_COUNTRY	VARCHAR2(30)		Area country. <b>Note:</b> The same value should already be loaded in Siebel S_REGION.COUNTRY. Use the initial capital format; for example, France.
INTEGRATION_ID	VARCHAR2(30)	NOT NULL	Siebel area (brick) ID. <b>Note:</b> This value should come from Siebel S_REGION.ROW_ID.

**Table E–16 Flat File Cross-Reference Data Population Rules for Account and Plan Data**

Field Name	Data Type	Field Required	Field Description
EXT_DATASRC_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1. <b>Note:</b> The same value should be used in the DATASOURCE_NUM_ID field of the syndicated data and syndicated market flat files.

**Table E-16 (Cont.) Flat File Cross-Reference Data Population Rules for Account and**

Field Name	Data Type	Field Required	Field Description
EXT_INTEGRATION_ID	VARCHAR2(30)	NOT NULL	External account and plan ID. <b>Note:</b> The same ID should be used in ACCNT_AREA_ID, AREA_ID, AND CON_AREA_ID syndicated data and syndicated market flat files.
EXT_ACCNT_NAME	VARCHAR2(100)		Account and plan name. <b>Note:</b> The same value should already be loaded in Siebel S_ORG_EXT.NAME.
EXT_ACCNT_NUM	VARCHAR2(30)		Account outlet number.
INTEGRATION_ID	VARCHAR2(30)	NOT NULL	Siebel account ID. <b>Note:</b> This value should come from Siebel S_REGION.ROW_ID.

**Table E-17 Flat File Cross-Reference Data Population Rules for Payer Type Data**

Field Name	Data Type	Field Required	Field Description
EXT_DATASRC_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1. <b>Note:</b> The same value should be used in the DATASOURCE_NUM_ID field of the syndicated data and syndicated market flat files.
EXT_INTEGRATION_ID	VARCHAR2(30)	NOT NULL	External payer ID. <b>Note:</b> The same value should be used in the PAYER_TYPE_ID field of the syndicated data and syndicated market flat files.
EXT_PLAN_TYPE	VARCHAR2(30)		Plan type, such as Payer, Plan, and so on. <b>Note:</b> The same value should already be loaded in Siebel S_INS_PLAN.PLAN_TYPE.
EXT_PAYER_NAME	VARCHAR2(100)		Payer type, such as Cash, Government, Third Party, Total, and so on. <b>Note:</b> The same value should already be loaded in Siebel S_INS_PLAN.NAME.
INTEGRATION_ID	VARCHAR2(30)	NOT NULL	Siebel payer ID. <b>Note:</b> This value should come from Siebel S_REGION.ROW_ID.

**Table E–18 Flat File Cross-Reference Data Population Rules for Contact Data**

Field Name	Data Type	Field Required	Field Description
EXT_DATASRC_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.  <b>Note:</b> The same value should be used in the DATASOURCE_NUM_ID field of the syndicated data and syndicated market flat files.
EXT_INTEGRATION_ID	VARCHAR2(30)	NOT NULL	External contact ID.  <b>Note:</b> The same value should be used in the CONTACT_ID field of the syndicated data and syndicated market flat files.
EXT_CON_FULL_NAME	VARCHAR2(102)		Contact name.  <b>Note:</b> The same value should already be loaded in Siebel S_CONTACT.LAST_NAME and S_CONTACT.FST_NAME.
EXT_ME_NUM	VARCHAR2(50)		Contact ME number.  <b>Note:</b> The same value should already be loaded in Siebel S_CONTACT.CSN.
INTEGRATION_ID	VARCHAR2(30)	NOT NULL	Siebel contact ID.  <b>Note:</b> This value should come from Siebel S_CONTACT.ROW_ID.

**Table E–19 Flat File Cross-Reference Data Population Rules for Product and Market Data**

Field Name	Data Type	Field Required	Field Description
EXT_DATASRC_NUM_ID	NUMBER(10)	NOT NULL	External data source number ID other than 1.  <b>Note:</b> The same value should be used in the DATASOURCE_NUM_ID field of the syndicated data and syndicated market flat files.
EXT_INTEGRATION_ID	VARCHAR2(30)	NOT NULL	External product and market ID.  <b>Note:</b> The same value should be used in the PAYER_TYPE_ID field of the syndicated data and syndicated market flat files.
EXT_PROD_TYPE	VARCHAR2(30)		Product and market type.  <b>Note:</b> The same value should already be loaded in Siebel S_PROD_INT.
EXT_PROD_NAME	VARCHAR2(50)		Product and market name.  <b>Note:</b> The same value should already be loaded in Siebel PROD_INT.NAME.
INTEGRATION_ID	VARCHAR2(30)	NOT NULL	Siebel product ID.  <b>Note:</b> This value should come from Siebel S_PROD_INT.ROW_ID.

## E.5 Life Sciences Data Loading Issues with Oracle Business Analytics Warehouse

This issue is specific to Analytics for Life Sciences and does not affect other products.

The ETL process updates the Oracle Business Analytics Warehouse for Life Sciences with a full refresh or using incremental updates. In the DAC, you can run a full load by selecting Tools, then ETL Management, then Reset Data Warehouse. This procedure is usually only used for the initial build. If you run the same execution plan without confirming the Reset Data Warehouse again, then the ETL incrementally updates the Oracle Business Analytics Warehouse.

Running a full refresh load by confirming the Reset Data Warehouse feature deletes all existing information stored in the fact and dimension tables. The following types of incremental update rules are supported on the fact tables.

- **Account Call Fact.** Adds new records (call activity for account) to the fact table.
- **Attendee Call Fact.** Adds new records (call activity for attendee) to the fact table.
- **Contact Call Fact.** Adds new records (call activity for contact) to the fact table
- **Syndicated Data - Base Direct Sales Fact, Territory Direct Sales Fact, Base Indirect Sales Fact, Territory Indirect Sales Fact, District Indirect Sales Fact, Base Physician Rx Fact, Territory Physician Rx Fact, Base Rx Plan Fact, Territory Rx Plan Fact, Base Weekly Physician Rx Fact, Territory Weekly Physician Rx Fact, Base Physician Rx Plan Fact, Territory Physician Rx Plan Fact.** ETL does not support incremental updates. When running the full refresh ETL, all records in the fact and dimension tables are deleted. To maintain a history in the dimension tables (such as multiple alignments), use the incremental ETL. If you need to incrementally update the syndicates data fact tables for incremental syndicated data loading, use one of the following strategies:
  - **For incremental insert.** Prepare flat file source data that has new data with the new INTEGRATION\_ID. Load the data directly into the staging table and modify the session not to truncate the fact tables. Then use the existing ETL to load data into fact tables.
  - **For incremental update.** Create new mappings that do a lookup against the fact tables and run the update. Make sure that INTEGRATION\_ID used in the flat file source and in the target tables are not identical. Because the incoming syndicated data in the flat file going to the target tables is in the normalized format, the INTEGRATION\_ID must be manipulated properly. However, the constant rule is applied when the INTEGRATION\_ID is manipulated during the ETL load. The syndicated data during the first bulk load has the "original in-coming INTEGRATION\_ID" || "-1"; the 26th bulk load has the "original in-coming INTEGRATION\_ID" || "-26".

### E.5.1 Known Issues with the Syndicated Data Fact Tables

The following are known issues with creation of aggregate measure columns in the Syndicated Data fact tables.

- With large volumes of syndicated data, the creation of aggregate measures in the Oracle Business Analytics Warehouse can take four times the amount of time needed to load the fact table. This may be unacceptable under circumstances when there are large amounts of syndicated data.
  - Incremental Updates in the Oracle Business Analytics Warehouse LS Dimension Tables.

- MAT aggregate measures are handled by the ETL process and not by metadata. All other aggregate measures are handled by metadata.

## E.6 Incremental Updates in the Oracle Business Analytics Warehouse LS Dimension Tables

This issue is specific to Oracle Business Analytics Warehouse for Life Sciences and does not affect other products. In the LS dimension tables the following incremental updates are supported. Some dimensions have more than one set of attributes.

### E.6.1 W\_ALIGNMT\_DH

**Base Dimensional Hierarchy Table.**

**Attribute.** Zip code, Brick, or Account/Contact Hierarchy depends on what alignment item type is selected.

---



---

**Note:** In release 7.9, all Zip Code, Brick and Account/Contact hierarchy alignment can be loaded.

---



---

**Description.** Contains the current version of alignment only.

### E.6.2 W\_ALIGNVER\_DH

**Slowly Changing Dimension Type.**

**Attribute.** Historical Zip code, Brick, or Account/Contact Hierarchy depends on what alignment item type is selected.

---



---

**Note:** In release 7.9, all Zip Code, Brick and Account/Contact hierarchy alignment can be loaded.

---



---

**Description.** Tracks historical alignments. Note the following:

- Depending on selected alignment type in AlignmentType.csv, only the selected alignment rules are extracted and loaded into the Oracle Business Analytics Warehouse. You can change the alignment type only when you run a full refresh ETL.

---



---

**Note:** In release 7.9, all Zip Code, Brick and Account/Contact alignment types can be extracted and loaded.

---



---

- When the alignment rule is loaded into the Oracle Business Analytics Warehouse, the current alignment rule is stored in the W\_ALIGNMT\_DH base table and the existing historical alignment rules are stored in W\_ALIGNVER\_DH table.
- The creation of Alignment Versions is controlled through the setting of the Alignment Version flag in DAC console. In Pharma Analytics 7.9 this flag is now set in the Source System Parameters screen within DAC's Design area. The parameter name is "\$\$ALIGNMENT\_VER" and takes a value of either Y or N.

- If the `$$ALIGNMENT_VER` parameter is set to N, then the ETL makes changes to existing alignments if there is a change on an assignment rule.
- If the `$$ALIGNMENT_VER` parameter is set to Y, it creates a new alignment version. Any existing alignment moves to the `W_ALIGNVER_DH` table as history. The `W_ALIGNVER_DH` table is used only when the `$$ALIGNMENT_VER` parameter is set to Y.

The dimension has these characteristics:

- The first alignment after Full load is 1.
- Incremental Updates in the Oracle Business Analytics Warehouse LS Dimension Tables
- A new version is created when a new assignment rule (new position or position relationship) or an assignment criteria is modified (change in postal code, brick, contact, or account) if the Alignment Version flag is set to Y.
- Assignment criteria:
  - Contact ZIP Code or Account ZIP Code cannot use ZIP Code ranges. Each ZIP Code assigned to a territory needs to be on a separate row, so the same value needs to be entered for both ZIP Code Low and ZIP Code High. For example, if ZIP Code 09654 is assigned to a territory, the value for both ZIP Code High and ZIP Code Low should be 09654. Also, the same ZIP Code should not be assigned twice to the same territory and a ZIP Code should be unique to a territory when using assignment criteria for Contact ZIP Code or Account ZIP Code.
  - Contact Brick or Account Brick require unique bricks assigned to a territory.
  - Every new version increases the counter by one and is a sequential number. Any pre-existing history version alignment data moves to `W_ALIGNVER_DH` historical alignment table whenever a new version is created. That is, the `W_ALIGNMT_DH` table always maintains the latest version while the `W_ALIGNVER_DH` table maintains the rest of the historical version if the `$$ALIGNMENT_VER` parameter is set to Y. So, when the historical alignment dimension table gets too large, your administrator can create a SQL query to delete all attributes for version numbers from the `W_ALIGNVER_DH` table.
  - Effective date of the alignment is assignment activation date.

---

---

# Pharma Disconnected Analytics Administration - Supplemental Information

This appendix describes additional configuration steps for Oracle's Pharma Disconnected Analytics, and contains the following sections:

- [Section F.1, "About Pharma Disconnected Analytics"](#)
- [Section F.2, "Sourcing Reports for Oracle's Pharma Disconnected Analytics"](#)

## F.1 About Pharma Disconnected Analytics

Pharma Disconnected Analytics is a preconfigured Disconnected Analytics application for Pharma Sales. Pharma components for Disconnected Analytics are installed during an Oracle Business Intelligence installation in the \OracleBIData\Disconnected\Pharma directory.

The following is a list of components that have been tailored for Pharma Disconnected Analytics and their locations:

- **PharmaDisconnect.rpd.** The Pharma Disconnected Analytics repository is tailored for Pharma Disconnected Analytics and downloaded to the laptop during synchronization. It is located in the \OracleBIData\Disconnected\Pharma\Application directory.
- **PharmaDisconnect.webcat.** Presentation Services tailored for Pharma Disconnected Analytics. It is located in the \OracleBIData\Disconnected\Pharma\Application directory.
- **Pharma.XML.** Application configuration file tailored for Pharma Disconnected Analytics. It is located in the \OracleBIData\Disconnected\Pharma directory.
- **SQL Files.** Series of predefined DDL files to create the Pharma Disconnected Analytics mobile database tables and indexes. The files are located in the \OracleBIData\Disconnected\Pharma\Application directory.
- **Messages XML Files.** Language specific message XML files for localized Presentation Catalog captions in the Pharma Disconnected Analytics application. The files are located in the \OracleBIData\Disconnected\Pharma\messages directory.
- **Disconnected Sourcing Reports.** Predefined Presentation Catalog reports used to extract data during the synchronization process to populate the tables in the Pharma Disconnected Application's mobile database. The reports are located in the Presentation Services catalog under the shared\disconnected folder.

The Pharma Disconnected Analytics components have been built to work against the standard server side Oracle Pharma Sales Analytics Application and Data Warehouse. The Pharma Disconnected Analytics components need to be customized to correctly reflect any changes and customizations made to the standard server side application before any initial synchronization.

To improve the performance of preprocessed synchronization by reducing query execution time on the Oracle BI Server, you can fine-tune the index strategy and sort the data during the ETL mapping process.

---

**Note:** For more detailed information on Disconnected Analytics Application configuration and deployment, see the *Oracle Business Intelligence Disconnected Analytics Administration and Configuration Guide*.

---

## F.2 Sourcing Reports for Oracle's Pharma Disconnected Analytics

Table F-1 contains a list of the sourcing reports (queries) for Oracle's Pharma Disconnected Analytics. These reports issue logical SQL to the Oracle BI Server to extract the data used to populate the local Disconnected client schema.

**Table F-1 Sourcing Reports for Pharma Disconnected Analytics**

Subject Area	Sourcing Report Name	Target Table
Pharma Promotional Effectiveness	Day Dimension	W_DAY_D
Pharma Promotional Effectiveness	Product Ranking Dimension	W_CON_RANK_D
Pharma Promotional Effectiveness	Brick Dimension	W_AREA_D
Pharma Promotional Effectiveness	Geography Dimension	W_GEO_D
Pharma Promotional Effectiveness	Payor Plan Dimension	W_INS_PLAN_D
Pharma Promotional Effectiveness	Product Hierarchy	W_PRODUCT_DH
Pharma Promotional Effectiveness	Position Hierarchy	W_POSITION_DH
Pharma Promotional Effectiveness	Call Priority List of Value Dimension	W_LOV_D
Pharma Promotional Effectiveness	Product Indication List of Value Dimension	W_LOV_D
Pharma Sales Effectiveness	Period Type List of Value Dimension	W_LOV_D
Pharma Promotional Effectiveness	Person Dimension	W_PERSON_D
Pharma Objective Achievement	Plan Promotion Dimension	W_PLAN_PROMO_D
Pharma Customer Demographics	Contact Primary Address Dimension	W_POSTN_CON_D
Pharma Promotional Effectiveness	Contact Call Activity Fact	W_CON_CALL_F

**Table F-1 (Cont.) Sourcing Reports for Pharma Disconnected Analytics**

Subject Area	Sourcing Report Name	Target Table
Pharma Promotional Effectiveness	Contact Call Activity Aggregate	W_CON_CALL_N_A
Pharma Objective Achievement	Contact Objective Fact	W_CON_OBJ_F
Pharma Sales Effectiveness	Indirect Sales Market Level Fact	W_MARKET_IDS_F
Pharma Sales Effectiveness	Indirect Sales Market Level Aggregate	W_MARKET_IDS_N_A
Pharma Sales Effectiveness	Prescription Market Level Fact	W_MARKET_RX_F
Pharma Sales Effectiveness	Prescription Market Level Aggregate	W_MARKET_RX_N_A
Pharma Sales Effectiveness	Direct Sales Fact	W_SYND_DS_F
Pharma Sales Effectiveness	Direct Sales Aggregate	W_SYND_DS_N_A
Pharma Sales Effectiveness	Indirect Sales Fact	W_SYND_IDS_F
Pharma Sales Effectiveness	Indirect Sales Aggregate	W_SYND_IDS_N_A
Pharma Sales Effectiveness	Prescription Sales Fact	W_SYND_RX_F
Pharma Sales Effectiveness	Prescription Sales Aggregate	W_SYND_RX_N_A

---

**Note:** The ATC hierarchy's ATC IV level should be linked to the lowest level of the Standard Product hierarchy and the lowest product level available for the Syndicated Data type loaded into the fact table. As a general rule, load data at the lowest level possible in the product hierarchy. For more information about best practices when using reports in Oracle's Pharma Analytics, see Appendix B in *Siebel Life Sciences Guide*, Version 8.0.

---



---

---

# Using Oracle Business Analytics Warehouse Exception Reports

---

---

**Note:** Exception Reports are only supported with Siebel CRM sources.

---

---

This chapter covers exception reports, which provide information about the source data used for ETL processes that can lead to erroneous results in the Oracle Business Analytics Warehouse or may cause data loss during the ETL process. These reports point out some of the known problematic areas, but they should not be relied upon to find all potential data inconsistencies in the source data.

This chapter includes the following topics:

- [Section G.1, "Understanding Oracle Business Analytics Warehouse Exceptions"](#)
- [Section G.2, "Executing Oracle Business Analytics Warehouse Exception Reports"](#)

## G.1 Understanding Oracle Business Analytics Warehouse Exceptions

Exception reports are defined for the following components:

- **List of Values.** Identifies gaps and overlaps for certain LOV types.
- **Cost Lists.** Identifies products for which the cost lists have not been defined. Identifies the cost lists which define costs for a product in a certain currency for overlapping periods of time.
- **Exchange Rates.** Currency Exchange rates that do not change over a period of time. If exchange rates are not defined for more than 30-day intervals, then they are flagged as an exception.
- **Hierarchies.** Entities that have circular references are flagged as exceptions. The Oracle Business Analytics Warehouse supports 10 levels of hierarchies. If there are entities that have more than 10 levels of hierarchies defined, they are flagged as exceptions.

### G.1.1 List of Values Exceptions

List of Values include High and Low values that can be used as bucket values in categories for effective analysis. If these values are not contiguous (such as gaps or overlaps in defined values), the ETL process cannot accurately categorize the values.

An example of List of Values exceptions is shown in [Table G-1](#).

**Table G-1 List of Values Exceptions (Example)**

Type	Name	Low	High
MY_TYPE	< 100	10	100
MY_TYPE	80 – 200	80	200
MY_TYPE	250 – 300	250	300
MY_TYPE	350 – 400	350	400

Notice that, in the example for Type MY\_TYPE, there are overlaps and gaps between records. There is an overlap of ranges between the first and second row. There is a gap between second and third row, and between third and fourth rows.

The following LOV types are analyzed for List of Value Exceptions:

- ACCNT\_REVENUE\_SIZE
- ACCNT\_EMP\_SIZE
- LEAD\_AGE\_DAYS
- OPTY\_REVENUE\_SIZE
- OPTY\_UNIT\_SIZE
- ACCNT\_REVENUE
- QUOTE\_AGE\_DAYS
- ACCNT\_REVN\_GROWTH
- APPROVAL\_AUTH\_SIZE
- SR\_CHART\_AGE
- ASSET\_COST\_CATEGORY

### G.1.2 Cost List Exceptions

Cost Lists for specified products and currency should not have overlapping time periods. If multiple cost lists are defined for a product and currency during a given time period, then the cost for the product may not be computed correctly in the Oracle Business Analytics Warehouse.

An example of Cost List exceptions is shown in [Table G-2](#).

**Table G-2 Cost List Exceptions (Example)**

Cost List	Product Name	Currency	Start Date (MM-DD-YYYY)	End Date (MM-DD-YYYY)	Cost
Cost List 1	Product 1	USD	01-01-2000	12-31-2000	10.00
Cost List 2	Product 1	USD	06-01-2000	06-01-2001	12.00
Cost List 3	Product 1	USD	06-01-2001	06-01-2002	13.00

In the example, Cost List 1 and 2 have definitions of cost overlapping over 06-01-2000 to 12-31-2000.

### G.1.3 Products Without a Cost List

During the ETL process, the costs of the products are calculated based on the Cost List table. If the cost lists are not defined correctly, the cost of the products cannot be calculated correctly in the Oracle Business Analytics Warehouse. This exception mapping queries the product table and looks for a minimum of one cost list to be defined. The products with no cost list definition are flagged as exceptions.

### G.1.4 Exchange Rate Exceptions

The Oracle Business Analytics Warehouse supports transactions in many different currencies. Oracle Business Intelligence converts all currencies in the Oracle Business Analytics Warehouse to a single currency for analysis purposes. The ETL Base Exchange Currency parameter in System Preferences, indicates the currency to which all the financial amounts will be converted. The Exchange rates are derived from the Exchange Rate tables in the Oracle Business Analytics Warehouse. If the currency exchange rates do not change for a period of 30 days, then Oracle Business Intelligence flags it as an exception.

If there are time period gaps in the exchange rate data, the ETL process defaults to the most recent recorded exchange rate. If the actual exchange rate is significantly more or less favorable than what is recorded in the database, the outdated exchange rate distorts the true value of currency amounts in the Oracle Business Analytics Warehouse.

---



---

**Note:** Exchange rates are derived from records that are of type "Daily" in the Oracle Business Analytics Warehouse. If any other types have been defined, they are not handled without some customization.

---



---

### G.1.5 Invalid Hierarchy Exceptions

Accounts, divisions, products, and opportunities can all have hierarchical relationships. These entities are denormalized within the Oracle Business Analytics Warehouse database to a fixed number of levels. Oracle Business Intelligence supports up to ten hierarchies levels in the Oracle Business Analytics Warehouse. If the depth of hierarchies extends beyond this number, results become inconsistent or incomplete.

### G.1.6 Circular Hierarchy Exceptions

Circular Hierarchies arise when the parent-child relationship has circular references. See [Table G-3](#) for an example.

**Table G-3** *Circular Hierarchy Example 1*

Child	Parent
A1	A2
A2	A1

Oracle Business Intelligence flags exceptions for two levels. Circular references over two hierarchies are not flagged. See [Table G-4](#) for an example.

**Table G-4** *Circular Hierarchy Example 2*

Child	Parent
A1	A2

**Table G-4 (Cont.) Circular Hierarchy Example 2**

Child	Parent
A2	A3
A3	A1

These produce infinite levels of hierarchies. The same records will be captured under the Invalid Hierarchy exceptions as their hierarchy depths will increase beyond 10 levels.

## G.2 Executing Oracle Business Analytics Warehouse Exception Reports

Before loading the Oracle Business Analytics Warehouse for the first time and for the subsequent refreshes, you should plan to spend time cleansing your transactional database data using the exception reports. The process is iterative, and requires coordination with other team members who have responsibility for data in the transactional database, such as the Siebel database administrator. After the initial cleansing, you should generate the exception reports on a scheduled basis to maintain the integrity of your data.

---



---

**Note:** Rerunning the exception reports overwrites all data contained in this view.

---



---

To cleanse the Oracle Business Analytics Warehouse, repeat these actions until the exception report is empty:

- In the DAC, run the Exception Reports execution plan.
- In Oracle's Siebel application that you are using, navigate to Analytics Administration, then Exception Reports.

---



---

**Note:** In Siebel Financial Services, this screen is called DataMart Administration.

---



---

- For every line in the ETL Exception Reports list, fix the cause of the problem. For information on fixing problems, see [Section G.2.1, "Cleansing Data"](#).

### G.2.1 Cleansing Data

Use a combination of the Exception Reports and the Diagnostic views to assess changes that need to be made external to the Oracle Business Analytics Warehouse, and changes to the Oracle Business Analytics Warehouse directly.

The ETL Exception Reports list contains one record for each exception flagged in the Oracle Business Analytics Warehouse. The ETL Exception Explanation form, located below the ETL Exception Reports list, describes the selected exception, its effect on the Oracle Business Analytics Warehouse building process, and offers suggestions for repairing the data.

#### To fix an exception

1. Select an exception record.
2. Read and understand the text in the ETL Exception Explanation form.

3. Click the report link.  
The object's data appears. (For example, if the object is an account, then the Account form appears. If the object is a cost list, then the Cost List list appears.)
4. Repair the problem, using the text in the ETL Exception Explanation form as a guide.
5. Return to the ETL Exception Reports list and place a check mark in the Fixed column to indicate to others that this exception has now been fixed.

## G.2.2 Using the List of Values View

Use the List of Values view, shown in the figure below, to visually compare how the list of values data extracted from the Oracle Business Analytics Warehouse coordinates with the values loaded into the Oracle Business Analytics Warehouse. The ETL process removes duplicates and overlaps and fills data gaps. Values are extended to span the List of Values (LOV) minimum and maximum values. Duplicates, Range Gaps, and Overlaps are flagged by the exception reports.

Figure G-1 List of Values View

Type	Language Independent Code	Display Value	Low	High	Order	Language Name
ABS_COST_FCT_AREA	Field Service	Field Service			1	English-American
ABS_COST_FCT_AREA	Professional Services	Professional Services			2	English-American
ABS_COST_FCT_AREA	Workforce Management	Workforce Management			3	English-American
ABS_COST_FCT_TYPE	Normal	Normal			1	English-American
ABS_COST_FCT_TYPE	Emergency	Emergency			2	English-American
ABS_COST_FCT_VARIABLE	Constraint Violation	Constraint Violation			1	English-American
ABS_COST_FCT_VARIABLE	FSE Overtime	FSE Overtime			2	English-American

Type	Language Independent Code	Display Value	Low	High	Order	Language Name
ABS_COST_FCT_AREA	Field Service	Field Service	0	9,999,999,999	1	English-American
ABS_COST_FCT_AREA	Professional Services	Professional Services	0	9,999,999,999	2	English-American
ABS_COST_FCT_AREA	Workforce Management	Workforce Management	0	9,999,999,999	3	English-American
ABS_COST_FCT_TYPE	Normal	Normal	0	9,999,999,999	1	English-American
ABS_COST_FCT_TYPE	Emergency	Emergency	0	9,999,999,999	2	English-American
ABS_COST_FCT_VARIABLE	Constraint Violation	Constraint Violation	0	9,999,999,999	1	English-American
ABS_COST_FCT_VARIABLE	FSE Overtime	FSE Overtime	0	9,999,999,999	2	English-American

The top List of Values list shows values from the Oracle Business Analytics Warehouse and the bottom List of Values (Data Warehouse) list shows the data that is to be used in ETL process. You can edit the Oracle Business Analytics Warehouse data directly in this view, but the Oracle Business Analytics Warehouse list is read-only.

---

**Note:** The List of Values is extracted into the Oracle Business Analytics Warehouse where the language is the same as the ETL Default Language set in the DAC Source System Parameters, or whose translate flag is set to "N," or those that are active. For more information about setting DAC Source System Parameters, see [Section 4.19, "How to Start the Informatica Server"](#).

---

## G.2.3 Using the Exchange Rates View

Use the Exchange Rates view to diagnose currency translation issues in the Oracle Business Analytics Warehouse. The ETL process removes duplicates, fills gaps, and

removes overlaps. The ETL process computes exchange rates based on commutative and associative properties, such as product and reverse rates.

The top Exchange Rates list shows currencies, the middle Exchange Rates list shows the Oracle Business Analytics Warehouse values for active currencies and their exchange rates, and the bottom Exchange Rates (Data Warehouse) list shows the values loaded into the Oracle Business Analytics Warehouse for the selected currency in the upper Exchange Rates list to the ETL Base Exchange Currency. The Exchange Rates (Data Warehouse) list is read-only.

- The Active Currencies predefined query restricts the list to the active currencies in the Oracle Business Analytics Warehouse.
- The exception reports flag any exchange rates to the ETL Base Exchange Currency that have not been defined within a specified period (30 days) in the DAC Source System Parameters.

### G.2.4 Using the Cost List View

Use the Cost List view to display the cost lists from the Oracle Business Analytics Warehouse from the point of view of the product, and a read-only view of the values to be loaded into the Oracle Business Analytics Warehouse. The ETL process removes duplicates, overlaps, and fills gaps.

The Cost List list (top) shows products, and the Cost List Line Items list (middle) shows the cost lists associated with the selected product. The Cost Lists (Data Warehouse) list (bottom) shows the data as it is transformed for the Oracle Business Analytics Warehouse.

- The exception reports flag products that do not appear in the Cost List list or have Cost List time gaps and overlaps.
- The Oracle Business Analytics Warehouse contains only one Cost List for a product and a currency at a time.

### G.2.5 Using the ETL History View

After all of the exceptions are corrected, the building of the data warehouse can be initiated. This view lists the history of the ETL processes and their statuses. When each ETL batch starts, the name of the process along with the timestamp is set, the status is set to STARTED. When the batch completes, its status is updated to COMPLETED.

### G.2.6 Additional Exceptions

The above mentioned exceptions are not an exhaustive list of all possible exceptions. Other exceptions are:

- The Analysis start and end date in the DAC Source System Parameters must span the entire period of time during which the transactions have occurred. For example, you may want to choose an early and late date range to cover the entire time period you are analyzing. These dates in the DAC Source System Parameters are crucial for the building of Day Dimension, flattening of Exchange Rates, Cost Lists, and KPI (Key Performance Indicator fact) calculations.
- The DAC Source System Parameters — ETL Date Format, ETL Analysis Start, ETL Analysis End parameters, and the List of Values — ETL\_UNSPEC\_DATE must be defined in the same data format. If one is changed, the others must be changed accordingly.

- List of Values must be defined appropriately. If there is no appropriate entry in List of Values, the strings that depend on List of Values in the Oracle Business Analytics Warehouse will not be translated.
- There must be exchange rates defined for the currencies your organization deals with. If the appropriate exchange values are not found, the ETL process uses the ETL Unknown Exchange Rate defined in the DAC Source System Parameters.



---

---

# Troubleshooting Oracle Business Intelligence Applications

This chapter details some common problems related to the Oracle Business Analytics Warehouse and suggests possible solutions. It contains the following topics:

- [Section H.1, "Oracle Business Analytics Warehouse Installation and Initial Configuration Issues"](#)
- [Section H.2, "Informatica and Loading Issues"](#)
- [Section H.3, "Unicode Issues with Informatica and the Oracle Business Analytics Warehouse"](#)
- [Section H.4, "UNIX Installation Issues with the Oracle Business Analytics Warehouse"](#)
- [Section H.5, "Oracle Business Analytics Warehouse Upgrade Issues"](#)
- [Section H.6, "Oracle BI Interactive Dashboard Issues"](#)
- [Section H.7, "ETL Issues"](#)

**Note:** When you troubleshoot installation and configuration issues, use the following log information:

On Windows:

- the Informatica Server Log (check the Informatica Event Viewer)
- the Informatica Rep Server Log (check the Informatica Event Viewer)
- the Informatica Repagent Log (check the Informatica Event Viewer)
- the workflow log (located in ..\Informatica\Server\WorkFlowLogs)
- the session log for the session that fails (located in ..\Informatica\Server\SessLogs)
- the (Teradata) loader logs (located in ..\Informatica\Server\Tgtfiles)
- the Data Warehouse Application Console logs (located in the \DAC\log directory)

On UNIX:

- the Informatica Server Log (located in ..informatica/server/server.log)
- the Informatica Rep Server Log (located in ..informatica/repserver/repserver.log)
- the Informatica Repagent Log (located in ..informatica/repserver/reagent.log)
- the workflow log (located in ../informatica/Server/WorkFlowLogs)

- the session log for the session that fails (located in ../informatica/Server/SessLogs)
- the (Teradata) loader logs (located in ../informatica/Server/Tgtfiles)
- the Data Warehouse Application Console logs (located in the /DAC/log directory)

## H.1 Oracle Business Analytics Warehouse Installation and Initial Configuration Issues

[Table H-1](#) provides information about problems and solutions related to the installation and initial configuration of the Oracle Business Analytics Warehouse.

**Table H-1 Oracle Business Analytics Warehouse Installation and Configuration Issues**

Symptom/Error Message	Probable Cause/Solution
Cannot connect to the Oracle Business Analytics Warehouse running on DB2.	Make sure that the DB2 configuration matches the requirements listed in <a href="#">Section 3.4, "IBM DB2 UDB-Specific Database Guidelines for Oracle Business Analytics Warehouse"</a> .
After installing Informatica's ODBC driver (V3.5) on a Windows computer, you cannot open the Query Analyzer, and the Enterprise Manager shows an error message.	The ODBC driver installation replaces the odbcbcp.dll file with an older version. The Informatica documentation contains the procedure for correcting this problem.
Cannot connect to Oracle Business Analytics Warehouse from Informatica.	Make sure that the server running the Informatica Server software has an ODBC connection to the Oracle Business Analytics Warehouse using a Siebel ODBC driver and an ODBC connection to the Oracle Business Analytics Warehouse using the Merant Closed 3.5 32-bit driver.
Error: ORA-12541: TNS:no listener.	Check the Compatibility tab in Informatica service configuration. It should be Oracle 8. If it is not set to Oracle 8, change it and restart the Informatica server.
Error 2140 (Informatica service failed to start).	The server may not be started or the Informatica Service may not be started. See Informatica's installation and configuration guide on Siebel eBusiness Third-Party Bookshelf for detailed information.
Informatica installation fails with an Unhandled Exception error and displays a message similar to this: "Error Number: 0x80040707. Description: Dll function call crashed: ISRT._DoInstall."	The computer is probably running out of virtual memory. Restart the computer and reinstall Informatica.
After selecting "Warehouse Default Amount (Yen)" as the currency symbol, the dollar sign (\$) is still displayed.	The currencies.xml file must be edited. For instructions, see <a href="#">Appendix B.3, "Changing the Default Currency in Analytics Applications"</a> .

## H.2 Informatica and Loading Issues

[Table H-2](#) provides information about problems and solutions related to issues with Informatica and loading. To view the Informatica log file details, double-click the workflow.

**Table H-2 Informatica and Loading Issues**

Symptom/Error Message	Probable Cause/Solution
Double-clicking the workflow yields a Workflow Manager error message: "The system cannot find the file specified."	The session log files are not set up properly. You also may need to change the text editor.
Using Oracle, some mappings hang while running when performance statistics are switched on.	When running some Informatica mappings for loading the Oracle Business Analytics Warehouse, turning on the Performance Statistics can cause the mapping to hang. The only workaround is to increase the values of the LMSHaredMemory and MaxSessions variables in Informatica. The risk of increasing the LMSHaredMemory too much is that it may start to have a serious effect on overall performance of the machine that the Informatica server is running on.
When you execute a workflow on the Informatica Workflow Manager, Informatica returns the following error message: "Request to start workflow (workflow name) on server (server name) not successful."	This can happen due to a server time-out property that is usually set to 20 or 40 seconds. When you try to run a large workflow, every session in that workflow is fetched into the server's memory. If this takes longer than the server time-out property, the server returns a message that the workflow was unable to run. However, the workflow is running, but the server just needs time to complete fetching the sessions into memory. Double-click the workflow to view the log file details.
Informatica (RDBMS is DB2) gives the following error message: Error occurred unlocking [SDE_ServiceRequestDimension1]. An error occurred while accessing the repository[[IBM]][CLI Driver][DB2/6000] SQL0955C Sort memory cannot be allocated to process the statement. Reason code = "". SQLSTATE=57011 DB2 Fatal Error[FnName: ExecuteDirect -- SQLSTATE=57011 [IBM][CLI Driver][DB2/6000]	The DB2 parameter "SHEAPTHRES" is too small.
When loading the data warehouse, Informatica reports a lock problem.	Either someone has a session open or there is a dead session. Make sure no one has any open sessions. If no sessions are open, then follow the Informatica documentation on removing locks caused by dead sessions.
Session SDEINC_RecordExtractStart fails due to unique constraint error while executing a Refresh workflow.	This could be because the previous load or refresh did not complete successfully. Fix the problem that caused the previous refresh session to fail. Make sure you start the process from last entry of %RestartNextWorkflow # before the failed session, and restart the workflow process from that point.
The session fails and you receive the following error code: Error "TE_7042 Aggregate Error: File Operation Error	This is due to a disk space limitation. Check the /Informatica/PowerMart/Cache/Check directory for available disk space, also check the limits (ulimit) of the account used to start PowerMart.

**Table H-2 (Cont.) Informatica and Loading Issues**

Symptom/Error Message	Probable Cause/Solution
Informatica sessions get deadlocked and eventually fail when they try to do a "select" from the repository table OPB_OBJECT_LOCKS. This problem sometimes occurs on MSSQL server databases.	<p>This is possibly caused by a limited number of resources on the MSSQL Database Server. The workaround is to execute the following MSSQL specific SQL command on the Oracle Business Analytics Warehouse:</p> <pre>DROP INDEX OPB_OBJECT_LOCKS.OPB_OBJ_LOCKS_IDX DROP INDEX OPB_OBJECT_LOCKS.OPB_OBJ_LOCKS_IDX2 DROP INDEX OPB_OBJECT_LOCKS.OPB_OBJ_LOCKS_IDX3</pre> <p>Upon completion of executing these commands, continue executing the workflow processes to load the Oracle Business Analytics Warehouse.</p>
An error may occur when trying to send a post session email notification using MS Outlook 2000. Refer to Informatica release notes for further information.	<p>After installing Informatica Server on Windows, copy the file mapi32.dll from winnt\system32 to the bin folder where the Informatica Server is installed, overwriting the existing mapi32.dll in that directory. Start the Informatica Server so that the Informatica Server can use the new mapi32.dll.</p> <p>The Extended MAPI Error. MAPILogonEx failed[2147746065] error indicates that the logon is not configured correctly. Check the following:</p> <ol style="list-style-type: none"> <li>1. Choose Services, then Informatica, then Logon, and make sure the login (domain\username) and password are correct.</li> <li>2. Choose Control Panel, then Mail (it may also be called Mail and Fax or Exchange), then Services, then Show Profiles, and make sure the mail profile is correct.</li> <li>3. Choose Programs, then Informatica Server, then Informatica Server Setup, then Miscellaneous, and make sure the MS Exchange profile is correct.</li> </ol>
While creating a custom session, bulk load mode does not work properly with SQL Server.	Change the mode to "normal" in Informatica repository for the session. The "normal" mode must be used everywhere for SQL Server in all of your custom sessions.
Outlook closes when sending out a notification of finishing the ETL process.	Informatica is closing Outlook. This issue is known to Informatica and is scheduled to be resolved in an upcoming release. Until then, create a second profile in Outlook and add that profile name to the Informatica server setup.

**Table H-2 (Cont.) Informatica and Loading Issues**

Symptom/Error Message	Probable Cause/Solution
<p>Oracle 8i sessions running in bulk mode fail and return an error message similar to:</p> <pre>WRITER_1_1_1&gt; CMN_1022 Database driver error... CMN_1022 [ ORA-00600: internal error code, arguments: [kpodpmop_01], [2], [], [], [], [], [], [] Or MAPPING&gt; TE_7022 TShmWriter: Initialized MAPPING&gt; Sat Jan 26 13:54:45 2002 MAPPING&gt; TE_7001 Internal error: Failed to allocate a target slot. Sat MAPPING&gt; Jan 26 13:54:45 2002 TE_ 7017 Failed to Initialize Server MAPPING&gt; Transformation BLK_ALL_DATATYPES1 Sat Jan 26 13:54:45 2002MAPPING&gt; TM_6006 Error initializing DTM for session...MAPPING&gt; TM_6020 Session [s_BULK_LONG] completed at [Sat Jan 26 13:54:45 2002]</pre> <p>During an ETL execution, when Informatica and DAC servers use DB2 Connect version 7 to talk to DB2/390 version 7 OLTP and data warehouse databases, you receive an error message similar to the following:</p> <pre>SEVERE: [IBM][CLI Driver][DB2] SQL0191N Error occurred because of a fragmented MBCS character. SQLSTATE=22504 103 SEVERE Tue May 11 21:37:29 CDT 2004 [IBM][CLI Driver][DB2] SQL0191N Error occurred because of a fragmented MBCS character. SQLSTATE=22504</pre>	<p>This problem is an Oracle 8i defect. It has been resolved in Oracle 9i. The workaround is to run the session in Normal mode. To do so, in Workflow Manager navigate to the Targets window, and change the Target Load type to Normal.</p> <p>This problem is a DB2 Connect version 7 (IBM) defect related to code page conversion. The problem has been resolved in DB2 Connect version 8.</p> <p>To correct the problem, do the following:</p> <ol style="list-style-type: none"> <li>1. Download the file IBM01140.ucs from <a href="ftp://ftp.software.ibm.com/ps/products/db2/fixes/english/siebel/siebel7/Conversion_Files">ftp://ftp.software.ibm.com/ps/products/db2/fixes/english/siebel/siebel7/Conversion_Files</a> to the /sqlib/conv directory.</li> <li>2. Make a copy of the file and rename it to IMB05348.ucs.</li> </ol>

**Table H-2 (Cont.) Informatica and Loading Issues**

Symptom/Error Message	Probable Cause/Solution
<p>When an ETL process is running and tasks fail, Informatica returns an error similar to the following:</p> <p>Error while running Workflow Description: ERROR : TM_6292 : (3040 4868) Session task instance REP_12400 [Repository Error ([REP_51055] Repository agent connection failed. [System Error (errno = 121): The semaphore timeout period has expired. (Cannot read message. Read 5824 bytes.)])]</p>	<p>This issue is related to the network. The workaround is to increase the Timeout parameter values in the Informatica Repository Server Administration Console.</p> <ol style="list-style-type: none"> <li>1. In the left pane of the Repository Server Administration Console window, right click your repository and select Edit Connection.</li> <li>2. In the Network tab, enter 9 as the value for the parameters MessageReceiveTimeout and MessageSendTimeout.</li> <li>3. Stop and start the Informatica Repository Server.</li> <li>4. Start the Informatica Server.</li> </ol>
<p>[IBM][CLI Driver] CLI0102E Invalid conversion. SQLSTATE=07006 sqlstate = 07006 [IBM][CLI Driver][DB2/AIX64] SQL0303N A value cannot be assigned to a host variable in the SELECT, VALUES, or FETCH statement because the data types are not compatible. SQLSTATE=42806 sqlstate = 42806 This issue comes up when you use 'Unicode' data movement in Informatica.</p>	<p>The target table that the mapping is inserting into is a non-unicode table. Make sure the table is created as an Unicode table.</p>
<p>The ELT run fails with one of the following errors:</p> <pre>FATAL ERROR : Caught a fatal signal [11]. Will abort this server process shortly. INFO : CMN_1131 : (3644 1) System shared memory [360454] for [Load Manager Shared Memory] removed.  FATAL ERROR : Aborting this server process due to a fatal signal [11].</pre>	<p>If you want to perform ETL runs in Informatica on HP-UX Itanium 2 64-bit, you need to install one of the following patches:</p> <p>HP-UX 11.11 Itanium 64-bit systems require Patch PHSS_33033 or higher.</p> <p>HP -UX 11.23 Itanium 64-bit systems require Patch PHSS_33349 or higher.</p>

### H.3 Unicode Issues with Informatica and the Oracle Business Analytics Warehouse

[Table H-3](#) provides information about problems and solutions related to Unicode issues with Informatica and the Oracle Business Analytics Warehouse.

**Table H-3 Unicode Issues with Informatica and Oracle Business Analytics Warehouse**

Symptom/Error Message	Probable Cause/Solution
<p>Task fails and generates the error message: "TRANSF_1_1_1_1&gt; TE_7073 Aggregate Error: Expecting keys to be ascending." Tasks known to fail are SDE_DtlForecastFact and SDE_CostList</p>	<p>In Informatica Mapping Designer, open the corresponding mapping and in the Aggregator transformation, remove the check from the Sortedinput check box.</p>

**Table H-3 (Cont.) Unicode Issues with Informatica and Oracle Business Analytics**

Symptom/Error Message	Probable Cause/Solution
<p>In the Windows 2000 environment, with a double-byte language, such as Japanese, when you try to input the path for the directory where Informatica is installed for the value \$PMRootDir in the Workflow Manager, the double-byte characters are converted to question marks. This causes the ETL to fail because it cannot find the Informatica Server folder and its subdirectories.</p>	<p>Share the folder under a new name with no double-byte characters, and use the network share as the value for \$PMRootDir.</p>
<p>When Configuring the Informatica Server or Informatica Repository Server, there are major truncations in some of the tabs.</p> <p>For example, in the Server tab and the Repository tab in the Configure Informatica Service screen (choose Start, then Programs, then Informatica Server, then Informatica Server Setup, then Configure Informatica Service), the field names are truncated.</p>	<p>Informatica 6.x does not support non-English messages and resources. Delete the following files to display all messages in English.</p> <p>PC/PM client folder\*411.dll</p> <p>PMServer folder\bin\*411.dll</p> <p>PMRepServer folder\bin\*411.dll</p>
<p>Unable to start the Informatica repository with the Informatica Repository Server on UNIX.</p> <p>The Oracle_BI_DW_Base (MS Windows Japanese and superset of Shift-JIS) is not compatible with that of pmrepagent (UTF-8 encoding of Unicode).</p>	<p>Make sure Japanese locales are installed on the UNIX machine. The environment variables on the UNIX machine are not complete. Set the environment variable PM_CODEPAGE to MS932. For other languages, set the variable appropriately.</p>
<p>The database connection code page is incorrect for Unicode/Non-Latin code pages.</p>	<p>The database connection code page has to be set manually through the Informatica Workflow Manager.</p> <ol style="list-style-type: none"> <li>1. Start the Workflow Manager.</li> <li>2. Connect to the Oracle_BI_DW_Base using Administrator as the user.</li> <li>3. Click Menu Connection, then Select Relational.</li> <li>4. Select the appropriate OLTP connection (DB2_OLTP for DB2, ORA_OLTP for Oracle, MSSQL_OLTP or ODBC_OLTP for MSSQL).</li> <li>5. For the code page select the UTF-8 encoding of Unicode or the appropriate code page, and then click OK.</li> <li>6. Select the OLAP connection (DB2_OLAP for DB2, ORA_OLAP for Oracle, MSSQL_OLAP or ODBC_OLTP for MSSQL).</li> <li>7. For the code page select the UTF-8 encoding of Unicode or the appropriate code page, and then click OK.</li> </ol>

**Table H-3 (Cont.) Unicode Issues with Informatica and Oracle Business Analytics**

Symptom/Error Message	Probable Cause/Solution
Unable to install Informatica PowerMart on non-English UNIX locales (for example, ESN, DEU, FRA, ITA). The following error message appears: "FATAL ERROR during initialization, Invalid language specified, Application terminated."	This error will occur when the locale environment variables are set inconsistently or not at all. These settings are determined by entering the command <code>locale</code> at the UNIX command prompt. Make sure the correct locales are installed on the UNIX machine. Set the environment variable <code>PM_CODEPAGENAME</code> correctly (For example set it to <code>MS932</code> for Japanese).

## H.4 UNIX Installation Issues with the Oracle Business Analytics Warehouse

Table H-4 provides information about issues related to installing the Oracle Business Analytics Warehouse on UNIX.

**Table H-4 UNIX Installation Issues with Oracle Business Analytics Warehouse**

Symptom/Error Message	Probable Cause/Solution
When starting Informatica in <code>pmserver.log</code> , the following error messages appear:  FATAL ERROR: LM_36023 [Tue Mar 11 23:12:41 2003]: (9968 1) The IP address [172.0.129.141] for this machine did not match the IP address [172.20.94] in the repository for server [ORACLE_BI_DW_SERVER].  FATAL ERROR: SF_34014 [Tue Mar 11 23:13:41 2003]: (9968 1) Server initialization failed.  INFO: SF_34014 [Tue Mar 11 23:13:41 2003]: (9968 1) Server shut down.	Launch Workflow Manager, then right-click on <code>Oracle_BI_DW_Base</code> and click More. Make sure the Host Name is the machine name where the Informatica Repository Server resides.

**Table H-4 (Cont.) UNIX Installation Issues with Oracle Business Analytics Warehouse**

Symptom/Error Message	Probable Cause/Solution
<p>When connecting to the Informatica Repository Server from the Repository Administration Console, the following error messages appear:</p> <ul style="list-style-type: none"> <li>■ Oracle databases running on Solaris or HP: Database Connection Error: Database driver event...Error occurred loading library [Bad magic number for shared library: /export/home/oracle/9.0.1.3/lib/1ibclntsh.sl]Database driver event...Error occurred loading library [libpmora8.sl]</li> <li>■ DB2 databases running on AIX: (17764 1) Database Connection Error: Database driver event...Error occurred loading library [No such file or directory] Database driver event...Error occurred loading library [libpmb2.a]</li> </ul>	<p>Use Oracle 32-bit library.</p> <ol style="list-style-type: none"> <li>1. Set the environment variable as follows: LD_LIBRARY_PATH=/export/home/oracle/9.0.1.3/lib32</li> <li>2. Restart the Informatica Repository Server service.</li> </ol> <p>DB2 databases running on AIX: Database Connection Error: Database driver event...Error occurred loading library [No such file or directory] Database driver event...Error occurred loading library [libpmb2.a]</p> <ul style="list-style-type: none"> <li>■ Set the environment variable as follows: For DB2 v8: setenv LIBPATH \${LIBPATH}:/usr/opt/db2_08_01/lib</li> <li>■ For DB2 v7: setenv LIBPATH \${LIBPATH}:/usr/lpp/db2_07_01/lib</li> <li>■ Restart the Informatica Repository Server service, after setting the variable.</li> </ul>
<p>When restoring the Informatica Repository, the following error message appears in the dwdb.log:</p> <p>ERROR: OBJM_54543 [Tue Mar 25 17:47:16 2003]: (164 2484) DataBase error:</p> <p>ORA-01461: can bind a LONG value only for insert into a LONG column</p>	<p>The database is Unicode and the environment variable is needed.</p> <ol style="list-style-type: none"> <li>1. For Unicode UTF8, set the environment variable to: American_America.UTF8</li> <li>2. For Unicode AL32UTF8, set the environment variable to: American_America.AL32UTF8</li> </ol>
<p>When starting pmserver on UNIX, the following error message appears:</p> <p>INFO: LM_36039 [Thu Mar 27 11:40:29 2003]: (6772 1) The maximum number of sessions that can run simultaneously is [10].</p> <p>FATAL ERROR: CMN_1011 [Thu Mar 27 11:40:29 2003]: (6772 1) Error allocating system shared memory of [3000000] bytes for [Load Manager Shared Memory]. Error is [0]: []</p> <p>FATAL ERROR: SF_34004 [Thu Mar 27 11:40:29 2003]: (6772 1) Server initialization failed.</p> <p>INFO: SF_34014 [Thu Mar 27 11:40:29 2003]: (6772 1) Server shut down.</p>	<p>In the pmserver.cfg file, change MaxSessions to 15 and change LMSharedMem to 3000000.</p> <p><b>Note:</b> For each 10 count increase of MaxSessions, increase the LMSharedMem by 2000000 bytes.</p>

**Table H-4 (Cont.) UNIX Installation Issues with Oracle Business Analytics Warehouse**

Symptom/Error Message	Probable Cause/Solution
A session fails (any session), and the following error message appears:  TM_6227 Error: codepage incompatible in session [SDE_RecordExtractStart]. The source database DB2_OLTP and server ORACLE_BI_DW_SERVER do not have compatible code pages. (One way compatibility is required).	For Informatica servers running on UNIX, in the pmserver.cfg file, set ValidateDataCodePages to No.  For Informatica servers running on Windows, navigate to Informatica Server setup, then Configuration, and remove the check from the Validate Data CodePages check box.
When trying to start the Informatica Server, the following error message appears:  sdc50a125{qauser7a}/> pmserver exec(): 0509-036 Cannot load program pmserver because of the following errors:  0509-150 Dependent module libpmcef.a could not be loaded.  0509-022 Cannot load module libpmcef.a.  0509-026 System error: A file or directory in the path name does not exist.	Set the environment parameters as follows:  setenv PATH /export/home/informatica/pm:~:\${PATH}  setenv LD_LIBRARY_PATH /export/home/informatica/pm:\${LD_LIBRARY_PATH}

## H.5 Oracle Business Analytics Warehouse Upgrade Issues

[Table H-5](#) provides information about problems and solutions related to issues with upgrading the Oracle Business Analytics Warehouse.

**Table H-5 Upgrade Issues**

Symptom/Error Message	Probable Cause/Solution
When upgrading from Oracle Business Analytics Warehouse version 7.5.3 to version 7.7, duplicate rows are inserted into a mini dimension table from the dimension table.	<p>Before upgrading, all leading and trailing spaces need to be removed from the data.</p> <p>Before running the workflow for the mini dimension that appears in the upgrade folder, modify the data to remove leading and trailing spaces by using the RTRIM and LTRIM functions. For example:  <code>set(col)=rtrim(col)</code></p> <p>In cases where you find leading and trailing spaces, modify the SQL override for the mini dimension mapping in the main folder as shown in the example below:</p> <pre> SELECT DISTINCT   CASE WHEN G1.VAL IS NULL THEN RTRIM(LTRIM(DS.AREA)) ELSE G1.VAL END AS AREA,   CASE WHEN G1.NAME IS NULL THEN RTRIM(LTRIM(DS.AREA)) ELSE G1.NAME END AS AREA_I,   CASE WHEN G2.VAL IS NULL THEN RTRIM(LTRIM(DS.SUB_AREA)) ELSE G2.VAL END AS SUB_AREA   CASE WHEN G2.NAME IS NULL THEN RTRIM (LTRIM(DS.SUB_ AREA)) ELSE G2.NAME END AS SUB_AREA_I   CASE WHEN G3.VAL IS NULL THEN RTRIM(LTRIM(DS.SEV_CD)) ELSE G3.VAL END AS SEV_CD   CASE WHEN G3.NAME IS NULL THEN RTRIM (LTRIM(DS.SEV_CD)) ELSE G3.NAME END AS SEV_CD_I   CASE WHEN G4.VAL IS NULL THEN RTRIM(LTRIM(DS.PRIO_CD)) ELSE G4.VAL END AS PRIO_CD   CASE WHEN G4.NAME IS NULL THEN RTRIM (LTRIM(DS.PRIO_ CD)) ELSE G4.NAME END AS PRIO_CD_I FROM   W_SRVREQ_DS DS,   W_LOV_G G1,   W_LOV_G G2,   W_LOV_G G3,   W_LOV_G G4 WHERE   W_SRVREQ_DS LEFT OUTER JOIN V_LOV_G G1 ON   DS.AREA = G1.NAMEVAL AND   G1.TYPE = 'SR_AREA' LEFT OUTER JOIN W_LOV_G G2 ON   DS.SUB_AREA = G2.NAMEVAL AND   G2.TYPE = 'SR_AREA' LEFT OUTER JOIN W_LOV_G G3 ON   DS.SEV_CD = G3.NAMEVAL AND   G3.TYPE = 'SR_SEVERITY' LEFT OUTER JOIN W_LOV_G G4 ON   DS.PRIO_CD = G4.NAMEVAL AND   G4.TYPE = 'SR_PRIORITY' }</pre>

## H.6 Oracle BI Interactive Dashboard Issues

Table H-6 provides information about problems and solutions related to Oracle BI Interactive Dashboards.

**Table H-6 Oracle BI Dashboard Issues**

Symptom/Error Message	Probable Cause/Solution
Error Codes: OPR4ONWY:U9IM8TAC:OI2DL65P File: ./project/webodbcaccess/odbcstatement impl.cpp, Line: 186 State: S1000. Code: 10058. [NQODBC] [SQL_STATE: S1000] [nQSError: 10058] A general error has occurred. [nQSError: 17001] Oracle Error code: 4030, message: ORA-04030: out of process memory when trying to allocate 246284 bytes (hash-join subh,killcqs:killslbta) at OCI call OCISmtFetch. [nQSError: 17012] Bulk fetch failed. (S1000)	This is caused by a shortage of RAM on a dedicated (non shared server) environment. Depending on the Oracle version, the options for DBAs are: <ul style="list-style-type: none"> <li>▪ Increase pga_aggregate_target.</li> <li>▪ Decrease sort_area_size and/or hash_area_size.</li> <li>▪ Move to multi-threaded server (a.k.a. MTS or shared servers).</li> </ul>

## H.7 ETL Issues

[Table H-7](#) provides information about problems and solutions related to ETL in Oracle BI Applications.

**Table H-7 ETL Issues**

Symptom/Error Message	Probable Cause/Solution
ETL failure with error: unable to create unique indexes over the W_UAEMP_ TMP and W_UAPOS_TMP tables.	Duplicate records exist in the WS_PER_RESP table, probably caused by Responsibility changes in the Siebel CRM Administration application.  For example, in Siebel CRM Administration View "Responsibilities", you remove a Responsibility for a particular User, and later re-associate that Responsibility to that User.

## Symbols

---

### \$\$PERIOD

- about, 11-16
- configuring, 11-17

### \$\$WH\_DATASOURCE\_NUM\_ID

- about, 11-16
- configuring, 11-16

## A

---

### action links

- about, C-6
- testing, C-6

### alignments dimension, configuring, D-10

### ANSI NULL option, setting, 3-9

### Applications

- Oracle Contact Center Telephony Analytics, 9-1, 13-1, 14-1
- Oracle Financial Analytics, 10-1
- Oracle HR, 12-1
- Oracle Order Management and Fulfillment Analytics, 11-1
- Oracle Sales Analytics, 13-1
- Oracle Supply Chain Analytics, 9-1, 13-1

### attributes

- historically significant attributes, identifying, 6-31

## B

---

### browsers

- settings, configuring, C-7

### bulkload mode, error while creating a custom session, H-4

## C

---

### charts

- local seed data, importing, B-2
- translation tables, process of maintaining, B-1
- W\_LOCALIZED\_STRING\_G table, adding custom translations, B-4
- Web catalog strings, about translating, B-5

### circular hierarchy exceptions, about and example, G-3

### code lookup

### about, 6-35

### code mappings, 6-35

### codes mapplets, 6-36

### sessions for lookups, configuring, 6-36

### W\_CODES table, 6-35

### Communications, Media, and Energy Analytics

### dashboards and pages, D-9

### data requirements, D-10

### order management analytics, about, D-7

### predictive scores, developing and deploying, D-4

### security settings, 7-36

### complex products defined, D-7

### configuration steps

### all source systems, 8-1

### controlling data set

#### Oracle Contact Center Telephony Analytics, 14-1

#### Oracle Financial Analytics, 10-1, 10-12

#### Oracle HR Analytics, 12-1

#### Oracle Sales Analytics, 13-1

#### Oracle's Supply Chain, 9-1

#### Order Management and Fulfillment Analytics, 11-1

#### source independent, 8-1

### Oracle BI Repository, 15-1

### Oracle EBS-specific, 8-7

### PeopleSoft-specific, 8-13

### required before full load

#### Oracle Contact Center Telephony Analytics, 14-1

#### Oracle Financial Analytics, 10-1

#### Oracle HR Analytics, 12-1

#### Oracle Sales Analytics, 13-1

#### Oracle's Supply Chain Analytics, 9-1

#### Order Management and Fulfillment Analytics, 11-1

#### source independent, 8-1

### configuring, D-11

### DAC client, 4-4, 4-20

### DAC config.bat, 4-20

### DAC metadata repository, 4-26

### email recipients in DAC client, 4-36

### fiscal calendar, 8-4

### Oracle Business Analytics Warehouse database, 4-30

### SSE role, 4-31

- conformed dimensions
  - universal source, configuring for, 6-43
- Contact Center Telephony Analytics
  - dashboards and pages, D-5
  - data requirements, D-7
- Cost List exceptions
  - See also* exception reports
  - about and example, G-2
  - Cost lists, using to display, G-6
  - products without a cost list, G-3
- CRM and EBS schema changes
  - verifying, 4-58
- cross-module configuration, performing
  - codes lookup, 6-35
  - conformed dimensions, configuring, 6-43
  - dimension keys, resolving, 6-37
  - domain values, working with, 6-38
  - loads, configuring, 6-28
  - records, filtering and deleting, 6-28
  - slowly changing dimensions, changing, 6-31
  - stored lookups, configuring, 6-35
- currency, diagnosing translation issues, G-5
- Customer Status History Fact table
  - about, 11-15
  - configuring, 11-16

## D

---

- DAC
  - configuring config.bat, 4-20
  - configuring metadata repository, 4-26
  - Source System Parameters, 4-54
  - starting the server, 4-55
- DAC config.bat
  - configuring, 4-20
- dashboards
  - CME dashboards and pages, D-9
  - consumer sector dashboards and pages, D-5
  - customizing, about, C-8
  - exposing dashboards and requests, C-14
  - information, about viewing information, C-14
  - locating dashboard and page names, C-14
  - views, adding, C-11
- data import
  - syndicated data loading, overview, E-2
- data loading
  - Oracle's Siebel Analytics for Life Sciences, issues
    - when loading, E-30
  - troubleshooting, Informatica and data loading
    - issues, H-2
- Data Source NUM ID
  - configuring \$\$WH\_DATASOURCE\_NUM\_ID, 8-6
- data warehouse
  - database connections, 4-27
- Data Warehouse Administration Console (DAC)
  - about, 2-4
  - configuring on UNIX, 5-19
  - configuring the client, 4-4, 4-20
  - setting up system properties, 4-26

- database connections
  - transactional, creating, 4-27
- database connections, creating
  - DB2 installations, creating database connections
    - for, 4-16
  - Oracle installation, creating database connections
    - for, 4-17
- DATASOURCE\_NUM\_ID
  - about, 8-6
  - configuring, 8-6
- DB Library Options setting, modifying, 3-9
- DB2
  - database connections, creating for, 4-16
  - Oracle Business Analytics Warehouse
    - requirements, 3-5
  - Oracle Business Analytics Warehouse, cannot connect to, H-2
- DB2 databases
  - DB2CODEPAGE, 3-25
  - environment variable, 3-25
- DB2-UDB
  - creating stored procedures, 4-59
- dimensions
  - alignments dimension, configuring, D-10
  - key resolution, using a lookup, 6-37
  - keys, resolving, 6-37
  - pharmaceuticals, configuring dimensions specific to, D-12
  - positions dimension, configuring, D-10
  - product costs, configuring, D-11
  - products dimension, configuring, D-11
- domain values
  - domain value conversion process,
    - understanding, 6-38
  - domain value set, configuring, 6-42
  - domain value set, extending, 6-41
  - importance, understanding, 6-40
  - working with, 6-38

## E

---

- email
  - configuring Email in the DAC Server, 4-53
  - configuring email recipients in the DAC
    - Client, 4-30
  - email settings on UNIX, 5-21
  - notification in Outlook error when sending post
    - session email, H-4
- environment variables
  - DB2CODEPAGE, 3-25
  - NLS\_LANG, 3-24
  - setting on UNIX, 5-16
  - setting up on Windows, 4-18
- ETL
  - error with W\_UAEMP\_TMP and W\_UAPOS\_TMP
    - table, H-12
  - History view, G-6
  - index errors, H-12
  - running with Non-English OLTP Data
    - Source, 3-29

- troubleshooting, A-2, H-12
- ETL failure, H-12
- exception reports
  - additional exceptions, list of, G-6
  - circular hierarchy exceptions, about and example, G-3
  - cleansing the data, process and diagram, G-4
  - ETL History view, using to view the history of the ETL processes and statuses, G-6
  - exception report components, list of, G-1
  - exception, fixing, G-4
  - exchange rate exceptions, about, G-3
  - Exchange Rate view, using to diagnose currency translation issues, G-5
  - executing the report, cleansing steps, G-4
  - invalid hierarchy exceptions, about, G-3
  - List of Values exceptions, G-1
  - List of Values view, using to compare values with Oracle Business Analytics Warehouse, G-5
  - See also* Cost List exceptions
- system preferences
  - exchange rate exceptions, about, G-3
- Exchange Rate view, using to diagnose currency issues, G-5
- extracts, configuring
  - extracting additional data, about, 6-25
  - extracting data from new source table, 6-26
  - extracting new data from existing source table, 6-26

## F

---

- fiscal calendar
  - configuring, 8-4
- fiscal month
  - configuring, 8-4
- fiscal week
  - configuring, 8-4
- for, 14-12, 14-13

## H

---

- Healthcare Analytics
  - security settings, 7-37
- hierarchical relationships, invalid hierarchy exceptions, about, G-3
- historically significant attributes
  - identifying, 6-31
- home page
  - content, customizing, C-8
  - customizing, about, C-8
  - path argument, determining, C-10
- Horizontal applications
  - indexing for performance, 3-28

## I

---

- Incentive Compensation Analytics, turning on, C-7
- indexing
  - S\_\* tables, 3-28
  - to improve performance, 3-28

- Informatica
  - copying source files, 4-11
  - default passwords, 4-8
  - fatal errors 11, H-6
  - fatal errors 3644-1 or 360454, H-6
  - installing patches on UNIX, 5-16
  - license files, updating on UNIX, 5-15
  - lookup files, 4-11
  - username and password, 15-1

- Informatica client software
  - Enable Data Code Pages Validation, checking if conditions met, 4-12

- Informatica repository
  - configuring, 4-39
  - configuring in Workflow Manager, 4-44
  - creating, 4-39
  - dropping, 4-47
  - restoring, 4-39
  - verifying configuration, 4-50

- Informatica Repository Server

- installing on UNIX, 5-7
- password, 4-8
- registering, 4-41

- Informatica Server

- installing on other than default location, 4-45
- password, 4-8
- registering, 4-36
- setting up, 4-10
- starting, 4-55
- starting on UNIX, 5-21

- Informatica Server and repository, configuring

- Unicode mode, configuring under
  - Windows, 4-18
- Windows, setting up Informatica service under, 4-55

- Initialization Blocks, 3-27

- installation

- database connectivity, 4-21
- Informatica Server, installing on other than default directory, 4-45
- Java SDK, 4-4, 4-20
- JDBC drivers, 4-21
- Oracle BI Applications installation process task list, 4-2
- wizard, Oracle BI Applications, 4-4

- installing

- Oracle BI repository, setting up, 15-2

- Insurance Analytics family of products

- security settings, 7-37

- IP address

- virtual IP address, creating, C-5
- virtual IP address, testing, C-5

## J

---

- Java SDK

- installation, 4-4, 4-20

- JDBC drivers

- installation, 4-21

## L

---

- List of Values
  - exceptions, about and example, G-1
  - view, using to compare values with the Oracle Business Analytics Warehouse, G-5
- loads, configuring
  - about, 6-28
  - Source Adapter mapplet. creating, 6-27

## M

---

- Mappings
  - SA System Subject Area, 3-26
- metadata
  - externalizing strings, B-3
  - object level security, repository groups, 7-25
  - Web Catalog group security, 7-26
  - Web Catalog security, 7-26
- metadata object level security
  - repository groups, 7-25
- Microsoft Outlook
  - closes when finishing ETL process, probable cause, H-4
  - error when sending post session email, H-4
- Migrating Seed Data from Siebel OLTP Source Systems, 4-61
- Moving From Test to Production, 4-61

## N

---

- NLS\_LANG environment variable, 3-24
- Non-English OLTP Data Source
  - running an ETL with, 3-29

## O

---

- OPB\_OBJECT\_LOCKS Informatica session
  - deadlocked, probable cause, H-4
- Operational Applications
  - dashboards, adding views, C-11
  - home page content, customizing, C-8
  - home page path argument, determining, C-10
- Oracle
  - database connections, creating for, 4-17
  - mappings hang, probable cause, H-3
- Oracle BI Applications
  - action links, about, C-6
  - action links, testing, C-6
  - browser settings, configuring, C-7
  - completing initialization, C-2
  - dashboards, adding views, C-11
  - home page and dashboards, about
    - customizing, C-8
  - home page content, customizing, C-8
  - Incentive Compensation Analytics, turning on, C-7
  - initialization, completing, C-2
  - installation and configuration process task list, 4-2
  - installation process, 5-3
  - installation wizard, 4-4
  - installing, UNIX, 5-6
  - operational applications, about integrating within, C-6
  - Optional Analytics Applications, C-7
  - UNIX, deploying, 5-2
  - virtual IP address, creating, C-5
  - Web Extension, configuring to work with, C-4
- Oracle BI Applications, configuring
  - home page, determining path argument, C-10
  - virtual IP address, testing, C-5
- Oracle BI Charts
  - metadata strings, externalizing, B-3
  - Oracle BI seed data, updating for non-English locales, B-2
- Oracle BI Presentation layer
  - process of maintaining translation tables, B-1
- Oracle BI repository
  - externalizing metadata strings, B-3
  - repository groups, security, 7-25
- Oracle BI seed data
  - updating for non-English locales, B-2
- Oracle BI Supply Chain Analytics, 9-1
  - domain values and CSV worksheet files, 9-7
  - universal source, configuring expenses, 9-24
- Oracle Business Analytics Warehouse
  - architecture components, 2-3
  - database parameters and requirements, 3-3
  - database, configuring, 4-30
  - DB2 requirements, 3-5
  - operating system, driver, connectivity requirements, 3-2
  - Oracle requirements, 3-7
  - SQL server-specific requirements, 3-8
  - tables, creating, 4-31
  - Teradata requirements, 3-11
- Oracle Contact Center Telephony Analytics
  - Benchmarks and Targets table, setting up, 14-8
  - Contact Representative Snapshot table, setting up, 14-7
  - Contact Status Type columns, about configuring for, 14-6
  - dates and times, configuring, 14-11
  - dimension key resolution, 14-2
  - dimension table, loading, 14-2
  - Event Type columns, about configuring for, 14-3
  - fact table, loading, 14-3
  - overview, 14-1
  - Representative Activities table, setting up, 14-6
- Oracle databases
  - NLS\_LANG, 3-24
- Oracle Financial Analytics
  - configuration steps before full load, 10-2
  - disabling logical table sources, D-3
  - externalizing metadata translation strings, D-3
  - logical table sources, disabling, D-3
  - logical table sources, updating, D-2
  - metadata translation strings, externalizing, D-3
  - overview, 10-1
  - updating logical table sources, D-2

- Oracle HR Analytics, 12-31
    - domain values and CSV worksheet files, 12-3
    - overview, 12-1
  - Oracle Order Management and Fulfillment Analytics
    - options, 11-1
    - overview, 11-1
  - Oracle Partner Analytics
    - role-based analytics dashboard mapping, 7-30
  - Oracle Pharma Sales Analytics applications
    - security settings, 7-39
  - Oracle Supply Chain Analytics
    - configuration steps before full load, 9-4
    - overview, 9-1
  - Oracle's Siebel Industry Applications
    - Oracle Pharma Sales Analytics security settings, 7-39
    - Oracle's Siebel Financial Services security settings, 7-37
  - Oracle's Siebel Life Sciences
    - data loading issues, E-30
  - Oracle's Siebel Operational Applications
    - Oracle Partner Analytics security settings, 7-30
  - Oracle's Siebel Industry Applications
    - alignments dimension, configuring, D-10
    - Analytics applications, about metadata requirements, D-5
    - CME dashboards and CME Analytics applications dashboards and pages, D-9
    - CME data requirements, D-10
    - CME security settings, 7-36
    - consumer sector dashboards and pages, D-5
    - consumer sector data requirements, D-7
    - consumer sector security settings, 7-36
    - dimensions, configuring specific to pharmaceuticals, D-12
    - Pharma sales effectiveness, configuring subject area, D-12
    - product costs dimension, configuring, D-11
    - Usage Accelerator Analytics security settings, 7-41
  - Oracle's Siebel Industry applications
    - alignments dimension, configuring, D-10
    - Analytics applications, about metadata requirements, D-5
    - CME dashboards and pages, D-9
    - CME data requirements, D-10
    - consumer sector dashboards and pages, D-5
    - consumer sector data requirements, D-7
    - dimensions, configuring specific to pharmaceuticals, D-12
    - Pharma medical education effectiveness, configuring subject area, D-13
    - Pharma product categories, configuring subject area, D-12
    - Pharma promotional effectiveness, configuring subject area, D-12
    - Pharma Sales Analytics, about metadata requirements, D-10
    - Pharma sales effectiveness, configuring subject area, D-12
  - positions dimension, configuring, D-10
  - predictive scores, developing and deploying, D-4
  - product costs dimension, configuring, D-11
  - Oracle's Siebel Operational Applications
    - integration with
      - data types, C-1
      - integration with Industry Applications, C-1
      - security
  - OracleBIAnalyticsApps.rpd
    - Informatica username and password, 15-1
  - order management analytics, about, D-7
  - Order Management and Fulfillment Analytics
    - multiple products, about tracking, 11-12
  - Outlook
    - closes when finishing ETL process, probable cause, H-4
    - error when sending post session email, H-4
- ## P
- 
- page names, locating and dashboards, C-14
  - parent-child relationship, circular hierarchy exceptions, G-3
  - password
    - Informatica administrator, 4-8
    - OracleBIAnalyticsApps.rpd, 15-1
  - Payroll table, about aggregating, 12-31
  - PeopleSoft Trees, 10-30
  - PerfIndex\_Horizontal.sql, 3-28
  - PerfIndex\_Industry.sql, 3-28
  - performance
    - components configured to optimize performance, 3-9
    - implementing indexes to improve, 3-28
  - Pharma objectives achievement
    - configuring subject area, D-13
    - medical education effectiveness, configuring subject area, D-13
  - Pharma product categories
    - configuring subject area, D-12
  - Pharma promotional effectiveness, configuring subject area, D-12
  - Pharma Sales Analytics
    - alignments dimension, configuring, D-10
    - dimensions, configuring specific to pharmaceuticals, D-12
    - medical education effectiveness, configuring subject area, D-13
    - metadata requirements, about, D-10
    - objectives achievement, configuring subject area, D-13
    - Pharma medical education effectiveness, configuring subject area, D-13
    - Pharma objectives achievement, configuring subject area, D-13
    - Pharma product categories, configuring subject area, D-12
    - Pharma promotional effectiveness, configuring subject area, D-12
    - Pharma Sales effectiveness, configuring subject

- area, D-12
- positions dimension, configuring, D-10
- product categories, configuring subject area, D-12
- product costs dimension, configuring, D-11
- products dimension, configuring, D-11
- promotional effectiveness, configuring subject area, D-12
- sales effectiveness, configuring subject area, D-12
- security settings, 7-39
- Pharma Sales Analytics applications
  - dimensions, configuring specific to pharmaceuticals, D-12
  - metadata requirements, about, D-10
  - Pharma medical education effectiveness, configuring subject area, D-13
  - Pharma objectives achievement, configuring subject area, D-13
  - Pharma product categories, configuring subject area, D-12
  - Pharma promotional effectiveness, configuring subject area, D-12
  - Pharma Sales effectiveness, configuring subject area, D-12
  - positions dimension, configuring, D-10
  - product costs dimension, configuring, D-11
  - products dimension, configuring, D-11
- Pharma Sales Analytics, about metadata requirements, D-10
- positions dimension, configuring, D-10
- predictive analytics metadata
  - predictive scores, developing and deploying, D-4
- Presentation Catalog, administering
  - dashboards, about viewing information, C-14
  - dashboards, exposing and requests, C-14
  - dashboards, locating and page names, C-14
- primary extract and delete mapping
  - delete and primary extract sessions, enabling, 6-31
  - source-archived records, deletion configuration, 6-30
  - working with, 6-30
- PRM Analytics
  - data-level visibility, 7-35
  - portal-based analytics dashboard mapping, 7-32
  - subject area mapping, 7-33
  - subject area visibility, 7-34
- product costs dimension, configuring, D-11
- products dimension, configuring, D-11

## Q

---

- Query, H-2
- Query Analyzer, can not open after installing ODBC driver, H-2

## R

---

- RDBMS error message, probable cause, H-3
- records

- source-archived records, deletion configuration, 6-30
- records, filtering and deleting
  - about, 6-28
  - primary extract and delete mappings, understanding, 6-29
  - primary extract and delete mappings, working with, 6-30

- repository
  - Oracle BI repository, setting up, 15-2
- repository, setting up, 15-2
- requests
  - exposing dashboards and requests, C-14
- responsibilities
  - in Usage Accelerator, A-2, H-12

## S

---

- S\_\* tables
  - indexing, 3-28
- SA System Subject Area
  - mapping, 3-26
- schema definitions
  - transactional database, updating, 4-56
- SDEINC\_RecordExtractStart session fails, probable cause, H-3
- security
  - CME security settings, 7-36
  - CME settings, 7-36
  - Communications, Media, and Energy Analytics, 7-36
  - consumer sector security settings, 7-36
  - data-level security, about, 7-52
  - data-level security, implementing, 7-53
  - default security settings, 7-3
  - embedded analytics support for changing positions (Siebel App), 7-49
  - employee/position based security, about, 7-26
  - initialization blocks, 3-27
  - integrating data security with EBS, 7-3
  - integrating data security with Oracle's PeopleSoft Enterprise Apps, 7-14
  - metadata object level security (repository groups), about, 7-25
  - operational applications security
  - Oracle Financial Analytics settings, 7-37
  - Oracle Partner Analytics security settings, 7-30
  - Oracle Pharma Sales Analytics security settings, 7-39
  - Oracle's Siebel Usage Accelerator Analytics security settings, 7-41
  - primary position based security for CRM vertical applications, about, 7-30
  - security categories, about, 7-1
  - team based visibility, 7-42
  - team based visibility, DAC metadata, 7-48
  - types of security, about, 7-1
  - user responsibilities, checking, 7-2
  - Web Catalog security, 7-26
- seed data

- importing, B-2
- seed data, importing, C-1
- server components, installation requirements
  - See also* server component software
- setting up
  - DAC system properties, 4-26
- Siebel Financial Analytics
  - PeopleSoft Trees, 10-30
- simple products defined, D-8
- slowly changing dimensions, configuring
  - about, 6-31
  - effective dates, 6-33
  - historically significant attributes,
    - identifying, 6-31
  - Type I changing dimension, 6-32
  - Type II slowly changing dimension, 6-32
- Source Adapter mapplet
  - creating, 6-27
- Source System Parameters
  - about, 4-54
  - setting, 4-54
- Source System Parameters (also known as DAC ETL Preferences), 4-54
- source-archived records
  - deletion configuration, 6-30
- SQL Server
  - DB Library Options setting, modifying, 3-9
  - Oracle Business Analytics Warehouse requirements, 3-8
- SSE role
  - creating, 4-31
- starting the DAC Server, 4-55
- stored lookups
  - configuring, 6-35
- stored procedures for DB2-UDB
  - creating, 4-59
- string translations
  - adding for metadata, B-4
- SWE
  - See* Web Extension
- syndicated data
  - importing, overview, E-2
- system preferences
  - See also* exception reports

## T

---

- Teradata
  - activating join indexes, 4-30
  - avoiding arithmetic overflows, 3-11
  - code page file property, setting, 4-50
  - configuration checklist, Windows, 4-60
  - creating and dropping tables, 4-35
  - data warehouse requirements, 3-11
  - database hints, 3-11
  - general guidelines, 3-11
  - Hosts file, setting up, 4-50
  - installation checklist for UNIX, 5-23
  - installing patches on UNIX, 5-16
  - setting up Hosts file on UNIX, 5-18

- Transactional database
  - connection error, H-2
  - Cost lists, displaying, G-6
  - importing locale seed data, B-2
- translation tables
  - process of maintaining, B-1
- troubleshooting
  - ETL error, H-6, H-12
  - ETL failure, A-2, H-12
  - general, H-1
  - Informatica and data loading issues, H-2
  - Oracle Business Analytics Warehouse
    - installation, H-2
  - Unicode, H-6, H-11, H-12
  - UNIX installation, H-8
  - W\_UAEMP\_TMP, H-12
  - W\_UAPOS\_TMP, H-12

## U

---

- Unicode
  - deploying Informatica and Oracle Business Analytics Warehouse in, 3-15
  - local seed data, importing, B-2
  - metadata strings, externalizing, B-3
  - Oracle BI seed data, updating for non-English locales, B-2
  - string translations, adding for metadata, B-4
  - translation tables, process of maintaining, B-1
  - W\_LOCALIZED\_STRING\_G table, adding custom translations, B-4
  - Web catalog strings, about translating, B-5
  - Windows, configuring under, 4-18
- universal source
  - conformed dimensions, configuring, 6-43
  - Oracle BI Supply Chain Analytics, configuring expenses, 9-24
- UNIX
  - deploying Oracle BI Applications, 5-2
  - updating license files, 5-15
- Upgrade Issues, 4-61
- URL
  - dashboards, configuring for, C-11
  - home page, configuring for, C-8
- Usage Accelerator
  - ETL failure, A-2, H-12
- Usage Accelerator Analytics security settings, 7-41
- username
  - OracleBIAnalyticsApps.rpd, 15-1

## V

---

- verifying schema changes in Oracle's Siebel (CRM) and EBS, 4-58
- Vertical applications
  - indexing for performance, 3-28
- views, adding to dashboards, C-11
- virtual IP address
  - creating, C-5
  - testing, C-5

## W

---

- W\_LOCALIZED\_STRING\_G table, adding custom translations, B-4
- W\_UAEMP\_TMP
  - ETL errors, H-12
- W\_UAPOS\_TMP
  - ETL errors, H-12
- Web Catalog groups
  - Analytics applications, security in, 7-26
- Web Catalog strings
  - about translating, B-5
- Web Extension
  - configuring to work with, C-4
  - virtual IP address, creating, C-5
  - virtual IP address, testing, C-5
- Windows
  - Informatica Server, setting up under, 4-55
  - Unicode mode, configuring under
    - Windows, 4-18
- workflows
  - error message when executing on the Informatica Server Manager, probable cause, H-3
  - Server Manager error message, probable cause, H-3